

Sem vložte zadání Vaší práce.



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

## **E-learningový portál Mentica - modul kurz**

***Bc. Jiří Matějka***

Vedoucí práce: Ing. Jiří Chludil

4. května 2015



---

## Poděkování

Tímto bych chtěl poděkovat vedoucímu své diplomové práce Ing. Jiřímu Chludilovi za veškerou pomoc a cenné rady při tvorbě této práce. Dále bych chtěl poděkovat ostatním členům týmu, Bc. Olze Budník a Bc. Jaroslavu Tesařovi za vzájemnou podporu a trpělivost. Na závěr bych rád poděkoval své rodině za možnost vystudování vysoké školy a podporu po celou dobu studia.



---

## 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.

V Praze dne 4. května 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Jiří Matějka. 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**

Matějka, Jiří. *E-learningový portál Mentica - modul kurz*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.



---

## Abstrakt

Práce se zabývá návrhem a implementací modulu pro správu výukových kurzů v týmovém projektu Mencia. Tento projekt je sestaven z několika dalších modulů pro podporu tvorby e-learningových materiálů. Součástí práce je rovněž analýza, návrh, implementace a testování modulu Kurz. Modul Kurz umožňuje vytvářet a spravovat studijní materiály různých druhů, které jsou zařazeny do kapitol a lekcí. Výstupem práce je modulární aplikace napsaná v programovacím jazyku PHP a frameworku Symfony2.

**Klíčová slova** Výukový kurz, e-learning, modulární architektura, online vzdělávání, výukový systém, přednáška, video lekce

---

## Abstract

The thesis deals with the design and implementation of educational course management module inside Mencia team project. This project consists of few more modules for e-learning materials creation support. Analysis, design, implementation and testing of Course module are part of this thesis as well. Course module enables creation and administration of several types of educational materials, that are categorized in chapters and lessons. Output of the thesis is an application, which is implemented in PHP framework Symfony2.

**Keywords** Learning course, e-learning, modular architecture, online learning, learning system, lecture, video lesson

---

# Obsah

<b>Úvod</b>	<b>1</b>
Rozdělení projektu . . . . .	2
<b>1 Analýza</b>	<b>3</b>
1.1 Výukový kurz . . . . .	3
1.2 Cílová skupina uživatelů . . . . .	6
1.3 Analýza konkurenčních řešení . . . . .	7
1.4 Souborový systém . . . . .	15
1.5 Výběr databázového systému . . . . .	18
1.6 Funkční požadavky . . . . .	19
1.7 Nefunkční požadavky . . . . .	22
1.8 Uživatelské role - aktéři . . . . .	22
1.9 Případy užití . . . . .	25
<b>2 Návrh</b>	<b>33</b>
2.1 Doménový model . . . . .	33
2.2 Databázový model . . . . .	34
2.3 Diagramy aktivit . . . . .	35
2.4 Návrh uživatelského rozhraní . . . . .	37
<b>3 Implementace</b>	<b>43</b>
3.1 Použité technologie . . . . .	43
3.2 Databáze . . . . .	45
3.3 Propojení modulů . . . . .	47
3.4 Lekce . . . . .	48
3.5 URL adresy . . . . .	50
3.6 Souborový systém . . . . .	51
3.7 Instalační příručka . . . . .	55
<b>4 Testování</b>	<b>57</b>

4.1	Testování s uživateli . . . . .	59
4.2	Testování funkčnosti . . . . .	60
<b>Závěr</b>		<b>63</b>
	Splněné a nesplněné cíle . . . . .	63
	Budoucí práce . . . . .	65
<b>Literatura</b>		<b>67</b>
<b>A Seznam použitých zkratk</b>		<b>69</b>
<b>B Screener</b>		<b>71</b>
B.1	Veřejná část screeneru . . . . .	71
B.2	Neveřejná část screeneru . . . . .	72
<b>C Pre-test dotazník</b>		<b>75</b>
<b>D Post-test dotazník</b>		<b>77</b>
<b>E Testování s uživateli - popis úkolů</b>		<b>79</b>
E.1	Vyučující . . . . .	79
E.2	Student . . . . .	80
<b>F Funkční testy - scénáře</b>		<b>81</b>
F.1	Scénář 1 - vytvořit kurz . . . . .	81
F.2	Scénář 2 - vytvořit studijní materiály . . . . .	83
F.3	Scénář 3 - publikovat kurz . . . . .	84
<b>G Uživatelská příručka</b>		<b>85</b>
<b>H Screenshoty aplikace</b>		<b>87</b>
<b>I Obsah příloženého CD</b>		<b>89</b>

---

## Seznam obrázků

1.1	Ukázkové schéma materiálů v kurzu . . . . .	3
1.2	Uživatelské role a jejich vztahy . . . . .	23
1.3	Případy užití - administrace kurzů . . . . .	26
1.4	Případy užití - administrace studijních materiálů . . . . .	29
1.5	Případy užití - studium kurzu . . . . .	30
1.6	Případy užití - student před přihlášením do kurzu . . . . .	31
2.1	Doménový model modulu kurz . . . . .	34
2.2	Databázový model modulu kurz . . . . .	36
2.3	Diagram aktivit - postup studenta kurzem . . . . .	37
2.4	Diagram aktivit - vytvoření a publikace nového kurzu . . . . .	37
2.5	Formulář pro vytvoření nového kurzu . . . . .	39
2.6	Administrační rozhraní pro správu studijních materiálů . . . . .	40
2.7	Administrační rozhraní pro přidání nové lekce . . . . .	41
2.8	Zobrazení lekce z pohledu studenta . . . . .	42
3.1	Diagram komponent modulů Mentica . . . . .	49
H.1	Editace kurzu . . . . .	87
H.2	Přidání nové lekce . . . . .	88
H.3	Seznam kurzů . . . . .	88



---

## Seznam tabulek

1.1	Hodnocení konkurence podle kritérií . . . . .	15
3.1	Skládání parametrů . . . . .	53
F.1	Testovací případ: Vytvořit kurz - validní data . . . . .	81
F.2	Testovací případ: Vytvořit kurz - nevalidní data . . . . .	82
F.3	Testovací případ vytvořit studijní materiály . . . . .	83
F.4	Testovací případ - publikovat svůj kurz s připojením kurzu . . . . .	84
F.5	Testovací případ - publikovat svůj kurz bez připojení kurzu . . . . .	84





---

# Úvod

Mentica.com je internetový portál poskytující uživatelům online vzdělávání jako alternativu klasické prezenční výuky. Vzdělávat se je možné pomocí osobního počítače a Internetu online. Tato moderní forma výuky přináší v dnešní uspěchané době jednu zásadní výhodu v podobě nezávislosti na místě a času, umožňuje tak uživateli zvolit si individuální rozvržení výuky. Mnoho lidí, zejména mladší generace používá svůj počítač nebo notebook každý den. Také tablety se staly alternativou knih a chytrými mobilními telefony jsme již téměř obklopeni a používáme je pořád. Díky nim můžeme být online kdekoliv a vyřizovat tak emaily, brouzdat po webu či se dorozumívat s přáteli pomocí zpráv na Internetu. V blízké budoucnosti bude tento nastolený trend pokračovat a výpočetní technika nás obklopí ještě více.

Dnešní společnost klade velký důraz na vzdělání, což dokazuje neustálý růst počtu vysokoškolských studentů v České Republice. Z dat [1] nasbíraných Českým statistickým úřadem vyplývá, že se počet absolventů vysokých škol za poslední dekádu ztrojnásobil. V roce 2013 bylo 91 600 absolventů, oproti tomu v roce 2003 bylo absolventů pouze 31 000. Celkový počet studentů vysokých škol v roce 2013 byl 368 000. Potřebu lidí vzdělávat se v každém věku dokazuje neustále rostoucí počet vzdělávacích kurzů a workshopů. Úřady práce po celé České republice nabízí mnoho rozmanitých rekvalifikačních kurzů, různé vzdělávací nebo zájmové kurzy jsou rovněž pořádány pro seniory a každý může studovat některou z univerzit 3. věku.

Základní myšlenkou pro vytvoření online vzdělávacího portálu je propojení moderních technologií a narůstající poptávky po vzdělávání. Podnět vytvořit takový portál vznesl BcA. Petr Mentberger, který je externím zadavatelem projektu. Projekt má přinášet řadu výhod v osobnostním rozvoji, například stimulovat motivaci uživatelů ke studiu pomocí gamifikace nebo možnosti porovnávání svých znalostí a výsledků s ostatními uživateli. Z těchto dat jsou uživatelům předkládány statistiky a dostávají za ně odznaky, které mohou sbírat. Výsledkem tedy má být moderní portál, který nabízí uživateli vzdělání jako službu v kombinaci se zábavou a s prvky sociální sítě.

## Rozdělení projektu

Projekt Mentica je rozsáhlý portál zahrnující mnoho funkcionalit. Na začátku projektu byl sestaven tříčlenný tým a projekt byl rozdělen do několika samostatně nezávislých modulů, které jsou realizovány a popsány v 3 diplomových pracích. Tato práce se zabývá modulem kurzu. Dalšími členy týmu jsou Bc. Olga Budnik, která má na starosti modul správa testů [2] a Bc. Jaroslav Tesař, který se zabývá uživatelským modulem [3].

Modul test pokrývá rozhraní pro studenta i učitele. Základní činností učitele je vytvářet a spravovat testy. Oproti tomu student testy vyplňuje a systém je poté opraví a výsledek zobrazí studentovi i mentorovi. Modul test poskytuje různé druhy testových otázek (klasické zaškrtování jedné či více odpovědí, doplňovací kvízy, přiřazovací kvízy, logické otázky typu pravda/nepravda a jiné). Při tvorbě testů uživatel vybírá z široké palety nastavení, například z druhů bodování testů, kdy studenti získávají body za kompletně správnou otázku, částečně správnou otázku a jiné.

Dalším modulem je modul uživatele. Trendem moderních projektů je podpora či zaintegrovaní různých prvků sociálních sítí. V případě Menticy tomu není jinak. Mentica může v budoucnu být sociální sítí, ve které mají uživatelé svůj veřejný/neveřejný profil, který je tvořen na základě informací vyplněnými samotnými uživateli. Profil je doplněn o data nasbíraná v průběhu studia v rámci portálu. Jako příklad lze uvést studijní výsledky či nasbírané odznaky. Výstupem profilu je životopis, který si může každý uživatel vyrobit ze svých dat. Mentica rovněž integruje nejpoužívanější sociální sítě, především facebook.

Další z modulů je modul gamifikace a motivace, který není realizován v rámci diplomových prací. Motivační systém je velmi propracovaný. Uživatelé získávají zkušenostní body (experience points) za aktivitu na webu. Po splnění daných kritérií pak získávají odznaky a ocenění (badges, achievements). Ze všech důležitých aktivit se vytvářejí žebříčky a statistiky. Uživatelé si mohou naplánovat svůj rozvrh učení v plánovači, který jim bude připomínat, že je čas na další lekci. Lákadlem pro mnohé uživatele je porovnání svých výsledků s ostatními uživateli a snaha překonávat je či soutěžit se svými přáteli. Za absolvování kurzů a testů získávají uživatelé zkušenostní body, které mohou být později aplikovány na odemknutí některých kurzů. Jako příklad lze uvést kurz přírodovědy pro pokročilé, který se uživateli otevře až po absolvování stejného kurzu pro začátečníky.

Vlastnosti modulu kurz jsou popsány v kapitole číslo 1.1. Analýzou, návrhem a implementací modulu se zabývá celá tato práce.

# Analýza

Kapitola analýza je zaměřena na sestavení podrobného zadání projektu, podle kterého bude později projekt realizován. Důležitou částí analýzy je stavení funkčních a nefunkčních požadavků a případů užití. Analýza je spolu s rešerší první fází vývoje projektu.

## 1.1 Výukový kurz

Výukovým kurzem rozumíme rozsáhlou posloupnost výukových materiálů strukturalizovanou v kapitolách, které obsahují lekce. Informace předávané v lekcích mohou být prezentovány různými formami. Díky tomu nemusí být obsah pro studenty stereotypní a může je více zaujmout. Prostor tak kromě klasických textových lekcí dostanou obrázky, videa, grafy či různé grafické vizualizace obsahu nebo testy. Jednotlivé prvky na sebe navazují a tvoří tak pomyslný spojový seznam, kterým lze procházet v obou směrech.

Posloupnost jednotlivých lekcí definuje autor kurzu - mentor. U kapitol lze nastavit volitelné podmínky určující odemknutí navazujících kapitol. V jednom kurzu jsou například zveřejněny všechny přednášky ihned a lze mezi nimi od začátku libovolně přeskakovat, u jiného kurzu mohou studenti přistupovat k materiálům dalších kapitol až po úspěšném absolvování testu právě otevřené kapitoly. Příklad průchodu kurzem je vidět na obrázku číslo 1.1.



Obrázek 1.1: Ukázkové schéma materiálů v kurzu

Délka studia kurzu není ve výchozím stavu omezena. Po přihlášení do kurzu může student pracovat se studijními materiály libovolně dlouho. Obtížnost i průměrná délka studia kurzů se bude velmi lišit v závislosti na konkrétním tématu a studijních materiálech.

Kurzy mohou být analogicky přirovnány k produktům na aukčních portálech, kterými jsou například [www.aukro.cz](http://www.aukro.cz) známé v Čechách či [www.ebay.com](http://www.ebay.com), který funguje po celém světě. Uživatelé v roli mentora mohou nabízet své produkty studentům, kteří jsou v roli zákazníků a přihlašují se ke studiu. Kurzy mají podobně jako produkty na aukčním serveru svoji úvodní produktovou stránku.

Uživatelé přihlášení do kurzů mohou kurzy hodnotit pomocí hvězdiček. Každý kurz má vlastní diskuzi, do které mohou uživatelé vyjádřit svůj názor na kurz. Uživatelé mohou kurzy sdílet na sociálních sítích nebo jim mohou zaslat pozvánku na kurz prostřednictvím emailu.

Jako modelový příklad lze uvést kurz pojmenovaný Adobe Photoshop pro začátečníky. Kurz bude obsahovat 4 tematické okruhy po 10 přednáškách. Po každém okruhu (kapitole) následuje nepovinný test. Kurz je zakončen povinným závěrečným testem, úspěšný řešitel kurzu musí dosáhnout 80 % bodů.

### 1.1.1 Rozdělení kurzů

Kurzy lze dělit na základě mnoha kritérií. Nejvýznamnějším kritériem je rozdělení na základě typu kurzu, dalšími kritérii jsou obtížnost, viditelnost nebo zveřejnění kurzu.

#### Typ kurzu

Kurzy dělíme podle typu na online kurz a prezenční kurz. Tato práce se zabývá online, neboli e-learningovým kurzem, který kompletně probíhá v elektronické podobě na webu. Studenti se učí z materiálů dostupných online a vyplňují testy. Na konci čeká uživatele závěrečný test, na jehož základě se určí výsledné hodnocení kurzu. Studenti nemusí znát mentora osobně a případná komunikace probíhá online.

Druhou variantou je prezenční kurz, ve kterém výuka může probíhat online nebo klasicky v učebně s mentorem, případně i kombinovaně. Na webu jsou umístěny veškeré materiály týkající se kurzu. U prezenčního kurzu musí být uvedeny doplňující informace ohledně místa a času konání. Závěrečný test může probíhat opět jak online, tak prezenčně. U prezenčních testů vyplňují mentoři výsledků testů do systému.

#### Viditelnost kurzu

Z hlediska viditelnosti můžeme kurzy rozdělit na veřejné a soukromé. Základní variantou kurzu je nechat viditelnost kurzu nastavenou na hodnotu veřejný. Veřejný znamená, že je kurz volně dostupný pro všechny uživatele a lze volně vyhledat nebo najít procházením v kategoriích nebo vyhledáváním podle názvu. Veřejný kurz se nachází na pevně definované volně dostupné URL adrese.

Soukromý kurz je speciální varianta kurzu, který nelze nijak vyhledat ani se do něj přihlásit. Jedinou cestou jak se do kurzu přihlásit je potvrdit přihlášku zaslou od mentora. Více informací o přihláškách a uživatelských skupinách je popsáno v práci [3] Bc. Jaroslava Tesaře. Praktické využití privátních kurzů je například pro třídy na základní škole nebo kruhy na vysoké škole, se stálou skupinou studentů. URL adresa privátního kurzu je přístupná pouze pro uživatele přihlášené do kurzu. Pro ostatní uživatele je URL adresa nedostupná.

### **Zveřejnění kurzu**

Kurzy lze rozdělit na publikované a nepublikované. Publikovaný kurz je veřejný pro všechny uživatele, kteří se do něj mohou začít přihlašovat. Rozpracovaný kurz, do kterého mentor přidává nové materiály nebo upravuje stávající je ve stavu nepublikovaný. Vlastníci kurzů si sami určují, kdy kurzy zveřejní.

### **Obtížnost**

Kurzy půjde filtrovat nejenom podle kategorií, ale rovněž podle obtížnosti. Mentori si u svých kurzů mohou zvolit některou z přednastavených obtížností (začátečník, pokročilý, expert).

## **1.1.2 Rozdělení obsahu v lekcích**

### **Nadpis**

Nadpis je pouze obyčejný text, bez možnosti jakéhokoliv formátování. Nadpis by se měl vejít na jeden řádek.

### **Textová lekce**

Textová lekce je strukturalizovaný text vytvořený přímo na webu pomocí textového editoru, který může být formátován. Přípustné formátování je odstavec, tučné písmo, kurzíva, podtržení, vložení speciálních znaků, a číslovaný nebo odrážkový seznam.

### **Obrázek**

Obrázek či fotografie připojená k lekci. Kvůli autorským právům a citacím lze k obrázkům připojit informace o jejich a zdroji a autorovi. Každý obrázek může být také doplněn o krátký popis.

### **Video lekce**

Uživatelé připojovat k lekcím svá videa. Stejně jako u obrázků mohou být u videí doplněny informace o autorovi, zdroji či popis videa.

### **Tabulka**

Tabulkou je myšleno zobrazení dat pomocí struktury řádků a sloupců. Jedná se však pouze o uspořádání textu, nad daty nepůjdou provádět funkce či jiné operace známe z tabulkových procesorů.

### **Matematický výraz**

Uživatelé zadávají matematické výrazy a rovnice či jiné objekty související s matematickou syntaxí.

### **Citace**

Citace je obyčejné textové pole bez možnosti formátování textu. Smyslem citace je zdůraznění krátké informace. V zobrazení lekce budou citace oproti obyčejnému textu zvýrazněny.

### **Příloha**

Přílohami jsou myšleny soubory nahrané od uživatelů, které nespádají do kategorie obrázků nebo video. Aplikace přílohy nezobrazuje a pouze je nabízí ke stažení. Příloha je zavedena převážně kvůli souborům s příponou pdf nebo souborům z kancelářského balíku MS Office.

### **1.1.3 Zakončení kurzu**

Jak již bylo zmíněno, každý kurz musí mít definovaný způsob zakončení. Standardní výstup z kurzu je absolvování závěrečného testu nebo složení zkoušky, v případě prezenčních kurzů. Mentori musí vždy připojit ke svému kurzu závěrečný test. U závěrečného testu lze omezit počet pokusů, čas při vyplňování testu, minimální hranice pro splnění testu, možnosti zobrazování otázek atd.

Při úspěšném absolvování závěrečného testu je kurz označen jako splněný, systém vypočte body, které uživateli přidělí. Přidělené body závisí na obtížnosti kurzu a na dosaženém výsledku v porovnání s ostatními uživateli pomocí percentilu. Přidělování zkušenostních bodů bude mít na starosti modul gamifikace.

## **1.2 Cílová skupina uživatelů**

Jednou z hlavních konkurenčních výhod projektu by měla být rozmanitost v rámci nabízených produktů. Mentica se nebude zaměřovat na jednu specifickou oblast, například jazyky nebo matematika, naopak se bude snažit pokrýt co nejširší skupinu předmětů a témat. K dispozici budou materiály jak pro děti na základní škole, tak pro vysokoškoláky nebo seniory. Kromě předmětů z oblasti vzdělání bude na portálu Mentica prostor pro kurzy volnočasových aktivit a sportů nebo na téma osobního rozvoje, který je čím dál více populárnější. Různorodost nabízených produktů je založena na faktu, že kurzy může vytvářet kterýkoliv registrovaný uživatel. O tom, který z kurzů je kvalitní a který nikoliv, rozhodnou sami uživatelé pomocí hlasování.

Portál Mentica by měl být atraktivní pro uživatele všech věkových kategorií, stejně jako pro uživatele s různým vzděláním. Své uplatnění zde najde laická i odborná veřejnost. Přestože projekt není zaměřen na určitou skupinu lidí, zadavatel specifikoval cílovou skupinu od 10 do 30 let, protože pro mladou generaci je běžné pracovat s počítači a trávit u nich volný čas.

## 1.3 Analýza konkurenčních řešení

Před samotnou analýzou konkurenčních řešení je nejdříve nutné specifikovat seznam pojmů, na základě kterých budou již existující portály porovnávány. Téměř totožné služby nabízené různými společnostmi mají různé názvosloví. Typickým příkladem je pojem kurz, který se dá použít v různých významech. Pro jeden subjekt znamená kurz dlouhodobý cyklus online přednášek realizovaných například pomocí videí. Na jiném webu se můžeme setkat s termínem kurz jako s jednorázovým několikahodinovým školením, do kterého mohou účastníci aktivně zasahovat pomocí mikrofonů. Služby nabízené konkurencí budou jednotně pojmenovány podle následujícího názvosloví.

### 1.3.1 Definice názvosloví nabízených produktů

#### **Kurz**

Pojem kurz je v této práci chápán jako dlouhodobá posloupnost přednášek a testů. Forma kurzů může být prezenční nebo online.

#### **Workshop**

Workshopem je myšlena krátká přednáška, která je externí, nikoliv online. U workshopů je maximálně kladen důraz na praxi a výklad teorie bývá omezen na nutné minimum. Účastníci workshopu si probírané postupy ihned zkouší a procvičují. V rámci Menticy se s workshopy počítá do budoucna. Potenciální zájemci se budou přihlašovat na termíny workshopů. Mentoři budou workshopy prezentovat na webu a spravovat přihlášené uživatele.

#### **Doučování**

Doučování je chápáno jako krátkodobé, většinou pravidelné vysvětlování látky studentovi. Doučování je z pravidla individuální a účastní se ho jen mentor a student. Mentica s doučováním nepočítá, i když v úvahu by přicházela možnost inzerce nabídky a poptávky doučování.

#### **Webinář**

Pojem webinář vznikl spojením slov web a seminář. Jedná se o seminář, který neprobíhá osobně, ale online. Vyučující vykládá účastníkům látku pomocí web kamery a mikrofonu. Dle definice webináře na inflow [4] by se účastníci měli aktivně podílet na obsahu webináře. Neměla by tedy nastat situace, kde studenti jsou v roli pasivního odběratele obsahu. Některé společnosti nabízí pro své klienty audiovizuální záznam z webinářů.

### 1.3.2 Online vzdělání v České Republice

#### **Webináře**

V České republice není mnoho webů specializujících se na poskytování webinářů pro veřejnost. V poslední době se situace začíná zlepšovat a nové weby

vznikají. Většina z nich je zaměřena převážně na ekonomické obory (účetnictví, obchodování, podnikání, na burze nebo na problematiku okolo daní). Dalším častým okruhem webinářů je téma osobního rozvoje (motivace, sebezdokonalování, atd.).

### **Doučování**

Doučování zprostředkované online je v České republice velmi silně zaměřeno na cizí jazyky. Trh je plný jazykových škol a agentur, které zprostředkují online doučování. Kromě jazyků se lze setkat s doučováním matematiky a fyziky pro základní až střední školy. Podíl matematiky je ovšem oproti cizím jazykům minoritní. Ostatní témata nejsou zahrnuta téměř vůbec.

Ve formě inzerce dobře na webu funguje nabídka doučování pro jednotlivce soukromými lektory. Nejedná se ale o online vzdělávání, ale o klasické prezenční doučování.

### **1.3.3 Výběr konkurence**

Pro analýzu existujících řešení budou porovnávány konkurenční portály nabízející kurzy na základě několika kritérií. Jednotlivým konkurentům bude každé kritérium ohodnoceno body podle toho, jak kritérium splňuje. Bodová stupnice je v rozsahu 0 až 5 bodů, kde 5 je nejvíce.

Konkurence bude porovnávána pomocí následujících kritérií:

#### **Druhy lekcí**

Typy obsahů byly rozděleny do 5 kategorií. Za každý splněný typ bude konkurent ohodnocen jedním kladným bodem. Prvním druhem hodnoceného obsahu jsou textové lekce, které mohou být pouze jako prostý text, obohacený text nebo HTML. V případě možnosti vkládání pouze prostého textu bude konkurent ohodnocen půl bodem, formátovaný text bude ohodnocen jedním celým bodem. Za typ lekce video lze dosáhnout rovněž až 1 bodu, který je rozdělen na dvě poloviny. První půl bod lze získat, pokud server umožňuje nahrávat uživatelům videa a druhý půl bod, pokud řešení dovoluje připojovat videa pomocí serverů třetích stran, například pomocí serveru [www.youtube.com](http://www.youtube.com). U studijních materiálů ve formě obrázků je možné získat půl bodu, pokud systém nahrávání obrázků podporuje a druhou polovinu bodu, pokud systém podporuje alespoň 3 z následujících přípon: **jpg**, **png**, **bmp**, **svg**, **gif**. Další bod lze získat, pokud systém podporuje vyplňování testů a poslední pátý bod je vyhrazen pro ostatní doplňující typy lekcí, například pro nápovědu, použití interaktivních prvků například map nebo 3D zobrazení tvarů.

#### **Vybrané funkcionality**

Body v tomto kritériu konkurenti získají, pokud jejich portál umožňuje následující funkcionality. Dva body v případě, pokud uživatel může být v roli vyučujícího. Až 1 bod za hodnocení kurzu (půl bodu za číselné hodnocení a půl bodu, pokud lze o kurzech diskutovat v přílehlé diskuzi). Další bod mohou



uživatelé získat, pokud systém umožňuje zobrazovat a upozorňovat na aktivity a novinky týkající se kurzů (například formou zdi na sociálních sítích). Poslední bod je rovněž rozdělen na dvě poloviny a první polovinu lze získat, pokud existuje systém pro správu uživatelů v kurzu a druhý půl bod, pokud mentoři vidí informace o studiu svých studentů.

#### **Gamifikace a motivace**

Každý vzdělávací portál nemusí mít implementovanou gamifikaci, či ji může úspěšně využít pouze v omezené míře. U gamifikace je důležité promyslet jednotlivé části a stanovit si jejich cíle. Pouze poté může gamifikace správně fungovat a jednotlivé druhy gamifikace se mohou vhodně doplňovat.

Konkurentům budou přiděleny body, pokud jsou v jejich portálu následující systémy gamifikace. Až 2 body za možnost získání odznaků (badges) a ocenění (achievements). Za implementaci zkušenostních bodů (experience points) také až 2 body, první bod je za možnost získávání zkušenostních bodů a druhý bod je za možnost jejich pozdějšího uplatnění. Poslední bod bude přidělen, pokud se na portálu objevuje motivační infografika, například procentuální počítadla říkající uživateli, kolik procent testu či formuláře už je vyplněno atd.

#### **Subjektivní ohodnocení**

V posledním hodnoceném kritériu nejsou body přidělovány striktně podle předem definovaných pravidel, ale podle subjektivního zhodnocení portálu.

Hodnocena byla originalita a unikátnost konkurenta, uživatelské rozhraní a celkový dojem z portálu.

#### **1.3.4 Český trh**

Portál, který by rozsahem odpovídal projektu Mentica, zatím v České republice ani na Slovensku není. Nicméně v oblasti vzdělávání u nás existuje mnoho webů. Některé z nich nabízí kvalitní služby, jiné méně. Žádná z nich není dostatečně univerzální, aby nabízela vzdělání v mnoha oborech najednou. Navíc je většina z nich poskytována pouze jako služba pro studenty, protože uživatel nemůže tvořit obsah.

#### **kurzyproradost.cz**

Portál [www.kurzyproradost.cz](http://www.kurzyproradost.cz) vznikl v roce 2012 a je orientován na kurzy z oblasti volnočasových aktivit či sebezdokonalování. Podle zaměření kurzů na hlavní stránce odhaduji cílovou skupinou webu ženy ve věku 30 až 50 let. Typickým příkladem kurzu je zde kurz vyšívání. Kurzy jsou rozděleny do několika dílů, většinou 3 - 10 přednášek, které se liší svou délkou. Po jednoduché registraci a přihlášení se uživatelé si mohou uživatelé prohlížet kurzy a do některých se přihlásit.

#### **Druhy lekcí**

Jednotlivé lekce kurzu jsou vždy video přednášky umístěné na serveru youtube. Systém umožňuje pod video lekci doplnit také přílohu ve formě zip

## 1. ANALÝZA

---

archivu. Videá na sebe navazují a lze skrz ně procházet tam i zpět. Textové přednášky, testy ani interaktivní prvky k dispozici nejsou.

Hodnocení: 1 bod

### **Vybrané funkcionality**

Uživatel je po registraci na portálu pouze v roli studenta a nemůže tak tvořit obsah. V patičce webu je odkaz na formulář pro uživatele, kteří by chtěli publikovat vlastní videa. Pokud by se chtěl uživatel stát mentorem, musí projít výběrovým řízením. Z tohoto důvodu nemohlo být zjištěno, jestli má systém rozhraní pro správu uživatelů. Pro studenty nejsou k dispozici žádné notifikace týkající se kurzu. V polovině roku 2014 byly k dispozici ke každému kurzu dvě diskuze. První byla realizovaná přímo na portálu pouze pro registrované uživatele a druhá přístupná komukoliv a realizovaná pomocí Facebook API. V současné době funguje pouze Facebook diskuze.

Hodnocení: 0,5 bodů (*část hodnocení nelze posoudit*)

### **Gamifikace a motivace**

Jelikož nejsou uživatelé na webu nijak testováni, nemohou se mezi sebou porovnávat. Nejsou tu žádné veřejné uživatelské profily, ve kterých by byly k vidění absolvované kurzy. Jakákoliv motivace či gamifikace na webu chybí.

Hodnocení: 0 bodů

### **Subjektivní ohodnocení**

Jako konkurenční výhodou považují jednoduchost a přímočarost webu, kde po registraci pomocí dvou kliknutí můžeme sledovat první lekci kurzu. V oboru volnočasových aktivit v Čechách nemají Kurzyproradost konkurenci. Nejvíce mě zaujala velikost diskuzí u většiny kurzů. Lze z toho vyvodit, že komunita uživatelů je velmi aktivní. Kurzy jsou zaměřeny převážně pro začátečníky.

Všechny kurzy mají placenou a neplacenou část. Neplacená část je spíše jen ukázkou. Negativně hodnotím záměrně skrytou cenu v přehledu kurzů, kde je u každého kurzu napsáno „vstoupit do kurzu zdarma“. Uživatele může odradit, že osnova placené části kurzu je vidět pouze po přihlášení se do kurzu.

Hodnocení: 2 body

### **Onlinejazyky.cz**

Portál [www.onlinejazyky.cz](http://www.onlinejazyky.cz) jsou zajisté nejzajímavějším z webů zabývajících se výukou cizích jazyků v České republice. Od roku 2007 využilo služby této společnosti více než 8500 studentů. Společnost se prezentuje webem s velmi moderním interaktivním designem a nabízí výuku 4 cizích jazyků pomocí přednášek, videí, kvízů a doplňovaček. Cena ročního kurzu libovolného jazyka je 3 000 Kč. Analýza portálu byla provedena ve verzi zdarma. Očekávám, že v plné verzi jsou k dispozici další funkcionality.

### **Druhy lekcí**

Základním stavebním kamenem jsou videa, která jsou sestříhaná tak, aby

byla co nejvíce poutavá. Textové přednášky jsou kratší, obsahují zformátovaný text a hojně se v nich vyskytují obrázky a zvuky. Například výuka slovíček je interaktivní, při kliknutí na obrázek se přehraje dané slovíčko se správnou výslovností. Uživatel si tak slovíčko přečte, vidí ho na obrázku a navíc ho i slyší. Výukou provází několik lektorů, studenti se setkají kromě českých lektorů i s rodilým mluvčím, takže získají představu o písemné i mluvené formě jazyka.

Mezi jednotlivými lekcemi se objevují různé testy a kvízy, které jsou velmi rozmanité (doplňování slov, křížovka, test, přiřazování správných možností atd). Vyhodnocení kvízů je vidět ihned včetně vyznačení chyb a procentuální úspěšnosti. Každá kapitola kurzu je zakončena závěrečným testem. Portál byl hodnocen pouze v demo verzi, která je zdarma. V této verzi nebyl závěrečný test dostupný, takže nemůže být bohužel posouzen.

Hodnocení: 5 bodů

#### **Vybrané funkcionality**

Web funguje jako výuka jazyků – služba pro studenty. Uživatelé nemohou být v roli mentorů a vytvářet vlastní kurz, takže není k dispozici ani správa uživatelů. Formát webu je postaven na individuálním samostatném studiu, ke kurzům nenáleží žádné diskuzní vlákno. Uživatelům jsou předkládány studijní výsledky s notifikacemi na zopakování slovíček či neaktivitu.

Hodnocení: 1 bod

#### **Gamifikace a motivace**

Po splnění kurzu si lze vyžádat certifikát o absolvování, který ale neslouží jako oficiální zkouška z cizího jazyka. Přesto vidina certifikátu může být pro uživatele motivační.

Hratelnost nabízí web v podobě vtipných videí po absolvování kvízů, jelikož lektori sdělují studentům výsledky vtipnou formou v závislosti na úspěchu či neúspěchu kvízu. Připraveny jsou desítky videí, které se střídavě spouští podle dosaženého výsledku. Ocenění ve formě pochvalného videa lze považovat za obdobu odznaků. Studenti mají přehledně viditelné statistiky se svou úspěšností v testech. Zkušební body portál implementovány nemá.

Hodnocení: 3 body

#### **Subjektivní ohodnocení**

Portál je velmi originální a lze ho uvést jako ukázkový příklad moderního vyučování v 21. století. Rozhraní webu je rovněž moderní. Všechny důležité funkcionality portálu jsou na začátku ukázány pomocí krátkého videa a interaktivní animace. Díky tomu může uživatel ihned začít využívat portálu naplno. Zajímavá a originální funkcionalita je vytváření vlastního slovníku, do kterého si uživatelé mohou vkládat slovíčka, která si nedokáží zapamatovat.

Jako jediné negativum zmíním úzkou specializaci na 4 cizí jazyky. Očekávám, že do budoucna se bude projekt rozrůstat a přibudou i ostatní cizí jazyky.

## 1. ANALÝZA

---

Pokud by takovýto web pokrýval více odlišných odvětví kromě jazyků, byl by pro mě ideální pomůckou při vzdělávání.

Hodnocení: 3,5 bodů

### 1.3.5 Zahraniční trh

Zahraniční trh, zejména anglicky hovořící, je oproti českému nesrovnatelně větší. Ve světě existuje celá řada projektů zaměřených na online vzdělávání. Z této obrovské množiny byly vybrány dva projekty, které jsou typově nejpodobnější portálu Mentica. Bohužel se tak do této analýzy nedostaly vzdělávací kurzy a přednášky poskytované zdarma nejprestižnějšími zahraničními univerzitami, jako je například Harvard, Cambridge, MIT, Yel či Berkley. Kurzy předních univerzit jsou k dispozici na vzdělávacích portálech [www.edx.org](http://www.edx.org) a [www.coursera.org](http://www.coursera.org).

#### Open2study.com

Původně australský projekt [www.open2study.com](http://www.open2study.com) místních vysokých škol, které na tomto webu publikují své kurzy a testy. V porovnání s dalším konkurentem [www.khanacademy.org](http://www.khanacademy.org) se jedná o mnohem menší projekt, který má celkově jen 5% návštěvnosti<sup>1</sup>, které dosahuje khanacademy. V současné době se seznam lektorů rozšířil i o průmyslové partnery.

#### Druhy lekcí

Základními výukovými materiály jsou textové přednášky s obrázky nebo PDF soubory. Uživatelé mohou shlédnout také výuková videa, která jsou spouštěna přímo ze serveru open2study. Interaktivní obsah není poskytován, uživatelé mohou však vyplňovat testy, které jsou velmi dobře zpracované a k dispozici jsou u testů například nápovědy. Jednotlivé části kurzu, které se zde nazývají moduly (modules) jsou vždy ukončeny testem, na který má uživatel celkem 3 pokusy. Po prvním pokusu lze zobrazit špatné odpovědi. Některý z pokusů uživatel musí odeslat, čímž se test uzavře a výsledek již nelze změnit. Konečné skóre se započítá jako výstup z modulu.

Hodnocení: 4 body

#### Vybrané funkcionality

Uživatelé se mohou stát mentorem, pouze pokud je administrátoři akreditují, běžný uživatel být mentorem nemůže. Nelze hodnotit, jestli systém nabízí správu uživatelů a možnost sledovat postup v jejich učení. Uživatelé mohou kurzy hodnotit a psát o nich recenze. Systém se v mnoha ohledech podobá sociální síti, k dispozici je zeď, na které uživatelé sledují upozornění a novinky z obsahu, který se jich týká.

---

<sup>1</sup>Zdroj dat návštěvností: [www.similarweb.com](http://www.similarweb.com)

Hodnocení: 2 body (*část hodnocení nelze posoudit*)

#### **Gamifikace a motivace**

Koncept webu je tematicky zasazen do prostředí skautingu. Uživatelé jsou motivováni plněním úkolů podobných táborovým bobříkům, ze kterých kompletují sbírky. Například, pokud se uživatel přihlášený do některého z kurzů spojí s ostatními členy kurzu, získá odznak táborového ohně. Nalezení vzácné mince je odměna za 100% úspěšnost ve všech testech v rámci kurzu. Za ověření svého profilu získá uživatel po registraci odznak stanu, symbolizující nováčka v táboře. Gamifikace je velmi originální a působivá. Motivační infografiku ani zkušenostní body web nenabízí.

Hodnocení: 2 body

#### **Subjektivní ohodnocení**

Online vzdělávací portál open2study považuji za typově nejpodobnější chystanému portálu Mentica, zejména v poskytování vzdělávacích materiálů. Líbí se mi koncept portálu, přehledné uživatelské rozhraní, zaujala mě 9 kroková nápověda ihned po registraci, která portál představí. Bohužel i na tomto portálu postrádám možnost vytvářet vlastní obsah a poskytovat ho ostatním studentům.

Hodnocení: 4,5 bodů

#### **Khanacademy.org**

Velmi úspěšný mezinárodní projekt [www.khanacademy.org](http://www.khanacademy.org) nabízí vzdělání dostupné komukoliv, protože poskytuje studijní materiály zdarma, což deklaruje i do budoucnosti. Na stránkách projektu je uvedeno, že část obsahu je přeložena do více než 40 cizích jazyků a do výukového systému je zaregistrován téměř 25 000 000 studentů (údaje z dubna 2015). Celosvětově je projekt Khanacademy dominující na trhu online vzdělávání.

V České republice vznikl projekt [www.khanovaskola.cz](http://www.khanovaskola.cz), ve kterém je část videí z Khanacademy přeloženo nebo doplněno o titulky. Na českém portálu ale chybí prvky profilu, gamifikace a web zatím funguje pouze pro přehrávání videí.

#### **Druhy lekcí**

Základní formát přednášek je buď text (doplněný obrázky různých formátů) nebo video lekce. Video lekce jsou umístěny na serverech [www.youtube.com](http://www.youtube.com). Výukou je proloženo časté testování znalostí pomocí krátkých kvízů. Positivem je stejný koncept přednášek z různých oborů. Všechny přednášky mají podobný styl výkladu a stejnou informační grafiku ve videích.

Uživatelé mohou vyplnit kvízy buď jako součást kurzu, nebo samostatně v rámci cvičení. V kvízech jsou k dispozici rozsáhlé nápovědy, které bývají velmi názorné. Uživatel může čerpat nápovědu v několika krocích. První krok je jen malá rada, kdežto poslední krok bývá vysvětlení celého řešení. Portál

## 1. ANALÝZA

---

nabízí i interaktivní lekce, ve kterých je například cílem studenta vynést do grafu průběh funkce podle zadání.

Hodnocení: 4,5 bodů

### **Vybrané funkcionality**

Khanacademy nabízí uživateli hned 3 role. Kromě klasického studenta může být uživatel ještě v roli rodiče nebo učitele. Cílem rodiče je dohlížet na své dítě v roli studenta, zadávat a plánovat mu práci a sledovat jeho studijní výsledky. Učitel vytváří třídy a přidává do nich studenty. Učitelé nemohou vytvářet vlastní materiály, ke každé třídě připojují již hotové kurzy a testy, kterých je na portálu nepřeberné množství. Diskuze ani hodnocení kurzů na webu není k dispozici.

Hodnocení: 4 body

### **Gamifikace a motivace**

Uživatelům jsou nehledě na roli předkládány podrobné soukromé statistiky vizualizující průběh studia. Uživatelé zároveň sbírají různé ocenění a odznaky. Veškeré tyto prvky jsou do systému dobře zapracované, pokud uživatele nezajímají, tak ho neruší, ale zároveň jsou dobře dostupné a uživatel si svoji sbírku může prohlížet. Kurzy se dají plnit na různé úrovni, které jsou odlišeny barvami. Výsledky jdou mezi uživateli snadno porovnávat, protože studenti vidí, kolik procent ostatních uživatelů splnilo kurz na úroveň master. Téměř za každou činnost na webu získávají studenti zkušenostní body (experience points). Gamifikace a motivace je na velmi vysoké úrovni.

Hodnocení: 5 bodů

### **Subjektivní ohodnocení**

Hlavními stavebními kameny Khanacademy jsou kurzy zdarma, vysoká úroveň gamifikace a role trenéra (coach). Díky těmto vlastnostem je Khanacademy unikátní a nesmírně populární. Uživatelé zcela jistě ocení jednotný formát všech videí – podobnou grafiku poznámek a podobný styl vykládání látky, takže si uživatelé nemusí zvykat na různé přístupy učitelů. Khanacademy funguje díky notifikacím jako sociální síť.

Hodnocení: 4,5 bodů

### **1.3.6 Zhodnocení konkurence**

Pro přehlednost a snazší vyhodnocení konkurenčních řešení jsou hodnocená kritéria všech konkurentů vyneseny do společné tabulky číslo 1.1.

Vítězem provedené analýzy konkurenčních řešení se stala Khanacademy, která obdržela 18,5 z maximálních 20 bodů. Rozdílové body navíc získalo Khanacademy díky možnosti vybrat si mezi rolemi student, učitel a trenér. Z českých konkurenčních řešení vyšel jako poražený portál Kurzyproradost, jeli-

Hodnocení konkurence podle kritérií				
	Kurzyporadost	Onlinejazyky	Open2Study	KhanAcademy
Druhy lekcí	1	5	4	4,5
Vybrané funkcionality	0,5 *	1	2 *	4
Gamifikace a motivace	0	3	2	5
Subjektivní ohodnocení	2	3,5	4,5	4,5
<b>Celkem</b>	<b>3,5</b>	<b>12,5</b>	<b>12,5</b>	<b>18,5</b>

\* - část hodnocení nelze kompletně posoudit

Tabulka 1.1: Hodnocení konkurence podle kritérií

kož se jedná pouze o placená videa bez dalších prvků e-learningového portálu. Druhý porovnávaný portál dopadl o poznání lépe, Onlinejazyky jsou moderní výukovou pomůckou a zaostávají za světovými projekty pouze úzkou specializací a možností pouze odebírat obsah. Obě tyto vlastnosti však vyplývají z konceptu portálu, který nabízí studium cizích jazyků jako službu.

Cílem portálu Mentica je vytvořit projekt osobitý podobně jako Open2study s funkcionalitami, které nabízí Khanacademy a moderními prvky z webu Onlinejazyky. Oproti konkurenci má mít portál Mentica navíc možnost být v roli mentora a svobodně vytvářet obsah pro ostatní uživatele.

## 1.4 Souborový systém

Ve všech implementovaných modulech Menticy je potřeba pracovat se soubory. Uživatelé nahrávají své profilové obrázky či obrázky do skupin, u testů je možné doplnit otázku o soubor. Zdaleka nejvíce pracuje se soubory modul kurz, protože každá lekce se skládá z několika souborů (texty, obrázky nebo videa). Součástí této práce je najít a aplikovat řešení pro jednotnou práci se soubory.

Vhodným řešením je použít souborový systém, který by měl být co nejvíce univerzální, aby mohl být nasazen a snadno použitelný napříč všemi moduly.

Souborový systém musí být přenositelný napříč úložišti, případná migrace webového serveru do cloudového úložiště musí být co nejjednodušší. Dalším požadavkem je schopnost pracovat se soubory různých formátů, kromě zobrazení obrázku či textu například streamovat videa.

Nezávislost na cílovém úložišti je v souborových systémech řešena pomocí takzvaných adaptérů. Adaptér je rozhraní mezi souborovým systémem a cílovým úložištěm. Adaptéry řeší připojení a přenos souborů do úložiště. Změna úložiště je možná díky adaptérům pouze změnou v konfiguraci.

### 1.4.1 Výhody souborového systému

Souborový systém přináší řadu výhod, zde je výčet těch nejdůležitějších:

- Streamování videí a velkých souborů
- Jednotný přístup k různým souborům
- Oddělení práce se soubory do vlastní abstraktní vrstvy
- Kešování pro rychlejší přístup k často používaným souborům
- Nezávislost na úložišti (localhost, cloud, apod.)
- Vyšší bezpečnost

### 1.4.2 Výběr souborového systému

#### The Filesystem Component

Do výběru byla zařazena komponenta filesystem, která je přímo zaintegrovaná v Symfony frameworku a je užitečným nástrojem pro práci se soubory. Oproti statickým metodám, které jsou součástí jazyka PHP, je tato komponenta navržena objektově, takže je práce se soubory jednotná, přehledná a znovupoužitelná. Komponenta je však nedostačující, protože OOP zaručuje pouze částečnou abstrakci a k dispozici nejsou adaptéry pro úložiště souborů.

#### Gaufrette

Gaufrette je knihovna poskytující abstraktní vrstvu souborového systému. Knihovna je vyvíjena skupinou KNPLabs<sup>2</sup>, která je velmi uznávaná komunitou uživatelů Symfony frameworku díky vytvoření oficiální dokumentace k frameworku. Velkou výhodou je integrace Gaufrette do Symfony frameworku pomocí GaufretteBundle. Samozřejmostí knihovny je OOP implementace. Soubory jsou ukládány a načítány pomocí principu klíč – hodnota, takže uživateli stačí vědět název k souboru a Gaufrette nezávisle na úložišti soubor vrátí. K dispozici je zhruba 10 adapterů pokrývajících nejrozšířenější cloudová řešení.

---

<sup>2</sup>KNPLabs je skupina Symfony2 expertů, která vytvořila mnoho open source projektů. [www.knplabs.com](http://www.knplabs.com)



Knihovna umí kešovat a streamovat soubory. Na verzovacím úložišti Github<sup>3</sup> jsou uvedeny základní návody s příklady.

### Filicious

Souborový systém Filicious je nový projekt, který byl spuštěn v srpnu roku 2014 v alfa verzi a zdaleka není hotov. V současné době (únor 2015) nemá projekt ani dokumentaci, ani adaptéry pro jakékoliv cloudové řešení a k dispozici je pouze localhost.

Z těchto důvodů je Filicious pro projekt Mentica nepoužitelný, nicméně pokud bude vydána plnohodnotná verze se všemi slibovanými funkcionalitami, které autoři připravují a prezentují na oficiálních stránkách projektu, bude Filicious plnohodnotný souborový systém s mnoha výhodami.

### Flysystem

Flysystem je úspěšný moderní souborový systém, který byl v březnu 2015 oceněn skupinou SensioLabs, která vytvořila Symfony framework a další světově úspěšné webové projekty. Flyweb podporuje kešování souborů, zvládá streamovat velké soubory a jako jediný zkoumaný souborový systém snadno zvládá přenos souborů mezi dvěma odlišnými úložišti. Stejně jako Gaufrette je zaintegrovan do Symfony Frameworku přes zásuvný balíček OneupFlysystem-Bundle<sup>4</sup>. Od stejného autora je navíc k dispozici druhý balíček pro klientskou část aplikace, který spolupracuje s několika nejrozšířenějšími javascriptovými nástroji pro vícenásobné nahrávání souborů pomocí drag and drop<sup>5</sup>.

Flysystem podporuje adaptéry pro různá úložiště, namátkou Amazon S3, Dropbox, Google drive a je v tomto ohledu srovnatelný se souborovým systémem Gaufrette.

### 1.4.3 Použitý souborový systém

Z výše popsaných souborových systémů se výběr zúžil na dva rovnocenné kandidáty – Gaufrette a Flysystem. Obě dvě knihovny jsou velmi podobné funkcionalitami a splňují všechny požadavky kladené na souborový systém. Nakonec byla vybrána knihovna Gaufrette, díky bohatší dokumentaci s více příklady. V její prospěch také rozhodl vývojový tým KNPLabs, jelikož je v projektu použito více rozšíření od této skupiny, například paginatorBundle<sup>6</sup>.

---

<sup>3</sup>Repositář projektu KNPLabs/KnpGaufretteBundle <https://github.com/KNPLabs/KnpGaufretteBundle>

<sup>4</sup>Repositář projektu 1up-lab/OneupFlysystemBundle <https://github.com/1up-lab/OneupFlysystemBundle>

<sup>5</sup>Drag and drop je technika používaná v grafických prostředích pro přesun objektů

<sup>6</sup>Repositář projektu KNPLabs/KnpPaginatorBundle <https://github.com/KNPLabs/KnpPaginatorBundle>

### 1.5 Výběr databázového systému

V rámci rozdělení projektu připadl do této práce výběr databázového systému včetně výběru technologie pro práci s databázovým systémem. Z diplomové práce [3] vyplývá, že aplikace bude naprogramována v PHP frameworku Symfony 2.6. Výběr nástrojů pro práci s databázovým systémem tak vychází z tohoto předpokladu.

#### 1.5.1 MySQL

Výkon a zejména vlastnosti databázového systému MySQL jsou značně ovlivněny použitím různých databázových úložišť (storing engine). Výchozím databázovým úložištěm je od verze 5.1 InnoDB, které nahradilo úložiště MyISAM používané jako výchozí v předchozích verzích. Za zmínku stojí také úložiště Archiv, které lze využít zejména k uchovávání velkých objemů dat.

#### Vlastnosti databázových úložišť

- MyISAM
  - Jednodušší na návrh a vytvoření pro začátečníky, protože nemá vztahy
  - Díky jednodušší struktuře rychlejší zápis
  - Zvláště rychlá při častém opakovaném čtení[5]
- InnoDB
  - Použití transakcí
  - Fulltextové vyhledávání ve verzi 5.6.4 a novější
  - Cizí klíče
- Archiv
  - Nejlepší komprese
  - Nepoužívá indexy
  - Povolené pouze příkazy INSERT a SELECT (DELETE a UPDATE neumožňuje)

Z vlastností databázových úložišť MySQL vyplývá, že archiv lze použít jen tam, kde se data po uložení v budoucnu nikdy nemění. Archiv je dobře využitelný například pro evidenci aktivity v aplikaci nebo pro uchování historických dat. MyISAM preferuje vyšší rychlost na úkor nižšího zabezpečení a konzistenci dat, InnoDB jde opačnou cestou. Dříve byla výhoda MyISAM například v možnosti fulltextového vyhledávání, které dnes má i InnoDB. Pro většinu případů považuji za výhodnější použít InnoDB, protože transakce a

cizí klíče jsou mocné nástroje. V určitých specifických případech však má MySQL smysl, například u dat, která se velmi často a nepravidelně mění. MySQL má řádově větší rychlost [6] při mnohonásobných vykonávání příkazu INSERT. U InnoDB lze docílit rychlejšího zpracování INSERT příkazů vložením více záznamů v rámci jednoho příkazu.

### 1.5.2 Porovnání PostgreSQL a MySQL

Porovnat mezi sebou PostgreSQL a MySQL využívající InnoDB je velmi obtížné. Oba databázové systémy jsou odlišné a hodí se k jiným účelům. Například nelze jednoznačně říci, který databázový systém je výkonnější. MySQL hojně používá kešování záznamů, které může výsledný čas dotazu zrychlit, ale v případě špatně zvolené velikosti keše i velmi zpomalit. Podrobný článek o srovnání databází PostgreSQL a MySQL na serveru wikivs [7] se shoduje se závěrem článku [8] zabývajícím se zátěžovými testy databázových systémů. Z obou článků vyplývá, že rychlost obou databázových systémů je srovnatelná. Obecně však platí, že MySQL dosahuje lepších výsledků u jednodušších dotazů na menší data a PostgreSQL naopak při větším objemu dat a složitějších SQL dotazech.

Z hlediska pokročilých funkcí z porovnání obou databázových systémů jasně vítězí PostgreSQL, jelikož umožňuje pokročilejší použití transakcí a triggerů. MySQL je ale mnohem více rozšířená. Jedním z důvodů masivního rozšíření MySQL je velké množství open source projektů, které pracují právě s tímto databázovým systémem. Dalším důvodem je jistě nepřehledné množství návodů pro začátečníky, které se zabývají právě MySQL.

S vědomím kvalit a výhod PostgreSQL avšak s přihlédnutím k současným znalostem a zkušenostem celého týmu jsme se rozhodli využít databázový systém MySQL. Díky použití Symfony frameworku a Doctrine 2 je však aplikace nezávislá na databázovém systému a do budoucna přichází v úvahu změnit databázový systém a použít například PostgreSQL, který bezpochyby nabízí pokročilejší nástroje pro práci s databází.

## 1.6 Funkční požadavky

Seznam funkčních požadavků je přehled požadovaných funkcionalit systému, kterými má systém disponovat. Na základě definice funkčních požadavků je patrné chování systému v konkrétních situacích.

Při sestavování funkčních požadavků bylo vycházeno ze specifikace systému od zadavatele popsané v úvodní studii 1.1. Funkční požadavky byly upraveny a doplněny na základě analýzy konkurenčních řešení 1.3.

### Z pohledu mentora – tvorba kurzu

#### F1.1 - Administrační rozhraní pro správu kurzů

Každý uživatel může vytvářet své vlastní kurzy pro ostatní uživatele. Pro

tento účel bude přístupné administrační rozhraní. Aplikace bude umožňovat editaci parametrů kurzů, jako je například cena kurzu, popis, obrázek, nastavení privátnosti (viditelnosti) kurzu, zařazení do kategorií, publikování kurzu a další.

### **F1.2 - Rozhraní pro tvorbu studijních materiálů**

Studijní materiály jsou uloženy v lekcích, které vždy spadají do kapitol. Součástí aplikace bude administrační rozhraní pro tvorbu, editaci a mazání lekcí. Lekce mohou být typu: nadpis, text, obrázek, video, graf, tabulka, citace, příloha.

### **F1.3 - Zobrazení kurzu jako student**

Při tvorbě kurzu a jeho lekcí vidí mentor vytvářený obsah v administračním rozhraní. Například při vytváření nové textové lekce - přednášky vidí obsah v textovém editoru. Aplikace mentorovi umožní přepnutí zobrazení kurzu do módu student, ve kterém uvidí svůj vytvořený obsah tak, jak ho uvidí studenti.

### **F1.4 – Publikování a stažení kurzu**

Při splnění podmínek pro publikaci (kurz obsahuje alespoň 1 kapitolu a má nastaven závěrečný test) umožní aplikace mentorovi kurz zveřejnit. Pokud je kurz ve stavu zveřejněný, je přístupný studentům, kteří se do něj mohou přihlásit. Mentor má možnost zrušit publikování kurzu a nastavit kurz zpět do stavu nezveřejněný.

### **F1.5 – Připojení a odpojení testu ke kurzu**

Ke každému kurzu musí být připojený závěrečný test. Bez splnění této podmínky nelze kurz publikovat. Aplikace poskytne rozhraní pro napojení testu z modulu test. V kurzu existují dva typy testů. Závěrečný test celého kurzu, který je povinný a nepovinné dílčí testy ke každé kapitole. Připojené testy může mentor od kurzu odebrat. Odpojením závěrečného testu se zruší zveřejnění kurzu.

### **F1.6 - Verzování studijních materiálů**

Všechny verze souborů nahrávaných od uživatelů i textových přednášek vytvářených ve webovém rozhraní jsou ukládány, neboli verzovány. Editace lekce způsobí přidání nové verze k lekci. Uživatelé se mohou ke starším verzím vracet.

## **Z pohledu studenta – studium kurzu**

### **F2.1 – Zobrazení studijních materiálů**

Pro každého studenta je připraveno studijní rozhraní, ve kterém lze zobrazovat jednotlivé lekce uspořádané v kapitolách. Při průchodu materiálu se lze pohybovat v obou směrech. Pokud závěr kapitoly obsahuje test, student je na něj přesměrován.

### **F2.2 – Kontrola přístupu k uzamčeným materiálům**

Aplikace umožní zobrazit studentům studijní materiály pouze u kapitol, které

má student odemčené. Kapitoly se odemykají splněním závěrečných testů předchozích kapitol.

### **F2.3 – Vizualizace průchodu kurzem**

Aplikace bude studentům graficky odlišovat lekce, které již shlédli od těch, které zatím neviděli. Rozděleny budou také dokončené kapitoly od nedokončených. Pro lepší přehled studentům poslouží také poměrová počítadla již shlédnutých lekcí a předchozí procentuální úspěšnost v kurzu.

### **F2.4 – Přidávání hodnocení ke kurzům**

Uživatelé mohou přidávat hodnocení kurzů, ke kterým jsou přihlášení. Hodnocení je číslo na stupnici od 0 do 5, kde pět je nejvíce. Výsledné hodnocení je zobrazeno jako průměr všech hodnocení zaokrouhlených na jedno desetinné místo.

## **Z pohledu studenta – před přihlášením do kurzu**

### **F3.1 – Vyhledávání kurzů**

Kurzy a ostatní produkty budou vyhledatelné pomocí fulltextového vyhledávání.

### **F3.2 – Vyhledávání kurzů pomocí procházení v kategoriích a podkategoriích**

Každý kurz musí být zařazen do některé z kategorií. Kategorie a podkategorie jsou reprezentovány ve stromové struktuře. Čím hlubší úroveň kategorie má kurz nastavenou ve stromové struktuře, tím lépe je zařazen.

### **F3.3 – Náhled kurzu**

Každý kurz má svoji produktovou stranu, podle které se uživatelé rozhodují o přihlášení se do kurzu. Na produktové straně je obrázek, popis, osnova kurzu a další informace.

### **F3.4 – Přihlašování se do kurzů**

Uživatelé se mohou přihlásit do kurzu a začít tak kurz studovat. Přihlásit se je možné na produktové stránce kurzu.

## **Ostatní požadavky související s kurzy**

### **F4.1 – Nahlášení závadného obsahu v kurzu**

Obsah na webu je tvořen uživateli, takže může být závadný a porušovat pravidla webu, autorská práva či zákony. Uživatelé mají možnost nahlásit závadný obsah stisknutím tlačítka, které otevře formulář, do kterého může uživatel připojit krátkou zprávu.

### **F4.2 – Diskuze ke kurzům**

Uživatelé mají možnost konverzovat v diskuzích u kurzů. Každý kurz má své vlastní diskuzní vlákno.

### 1.7 Nefunkční požadavky

Nefunkční požadavky definují vlastnosti a omezující podmínky na systém. Nefunkční požadavky se obvykle vztahují na celý systém a stejně jako funkční požadavky vyplývají ze zadání projektu.

#### **N1.1 - Dostupnost přes web**

Aplikace bude pro uživatele přístupná pomocí Internetového prohlížeče a protokolu HTTP.

#### **N1.2 - Serverová část aplikace bude naprogramována v PHP frameworku Symfony2**

Z analýzy diplomové práce [3] vyplynulo, že pro implementaci projektu bude použit programovací jazyk PHP a aplikace bude vyvinuta ve frameworku Symfony2, který disponuje MVC architekturou a má v sobě zaintegrovanou nástavbu modelové vrstvy ORM Doctrine 2. Symfony2 framework napomáhá udržovat kód znovupoužitelný a klade velký důraz na zabezpečení.

#### **N1.3 – Unifikované webové rozhraní**

Aplikace musí být přístupná nezávisle na zařízení, kterými se uživatelé budou připojovat (chytré telefony, tablety, osobní počítače). Zobrazované stránky budou optimalizované pro všechny významné internetové prohlížeče (Internet Explorer, Mozilla Firefox, Chrome, Safari).

#### **N1.4 - Aplikace bude klást důraz na snížení zátěže serveru a přenesením výpočtů na klienta.**

Předzpracování dat, validace formulářů, vykreslování statistik a další funkcionality budou řešeny na straně klienta pomocí jazyka JavaScript.

#### **N1.5 - Přístup k datům pomocí souborového systému**

Veškeré soubory bude aplikace ukládat pomocí jednotného souborového systému, který bude nezávislý na cílovém úložišti.

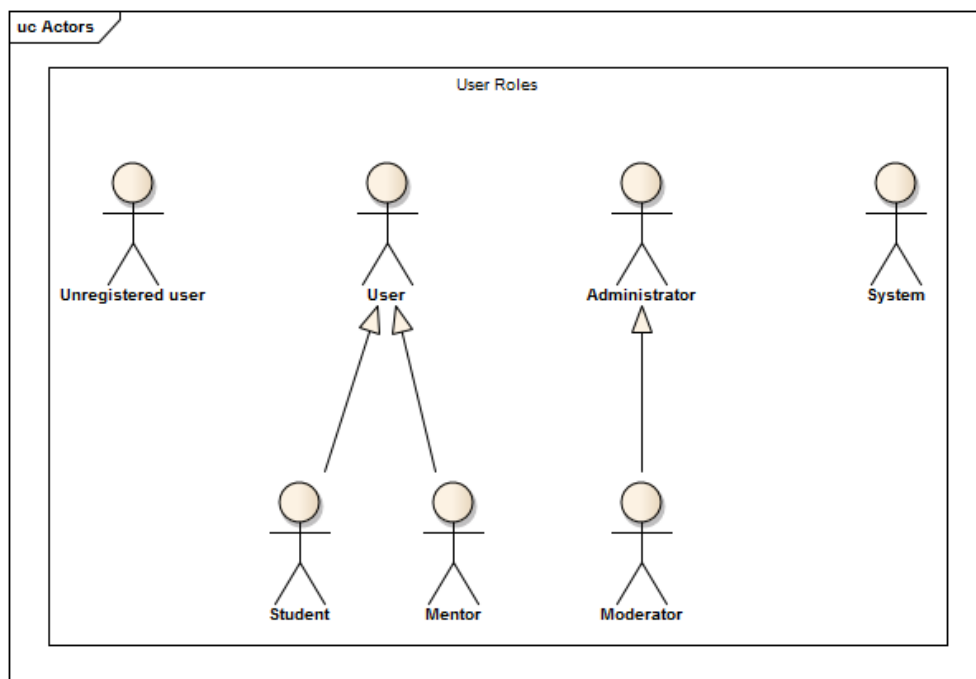
#### **N1.6 - Použití hezkých URL**

Aplikace bude využívat hezké URL (nice URLs), které jsou přívětivé pro uživatele.

### 1.8 Uživatelské role - aktéři

Definování uživatelských rolí neboli aktérů umožňuje rozdělit uživatele webu do různých skupin. Každý aktér má definovány určité funkcionality aplikace. Aktéři nepatří přímo do systému, ale jsou se systémem v interakci.

Aktéři vyplývají ze zadání projektu, které je popsáno pomocí funkčních požadavků. Díky nim můžeme pro jednotlivé uživatelské role definovat hranice systému. Základním aktérem je registrovaný a přihlášený uživatel, který může být zároveň v roli studenta i mentora. Následující seznam uživatelských rolí se vztahuje převážně ke kurzům, nicméně pro úplnost rolí jsou uvedeny popisy



Obrázek 1.2: Uživatelské role a jejich vztahy

rolí, které přímo s kurzy nesouvisí. Vizualizace uživatelských rolí je zobrazena na obrázku číslo 1.2.

### 1.8.1 Nepřihlášený uživatel

Každý návštěvník webu vystupuje alespoň v roli nepřihlášeného uživatele, který se může buď zaregistrovat pokud nemá na webu účet anebo přihlásit, pokud již účet má. Nepřihlášený uživatel je rolí s nejmenším vztahem k aplikaci a má značně omezené pravomoci, protože vytvářet nebo čerpat obsah mohou pouze uživatelé přihlášení do svého účtu. Hlavní akcí nepřihlášeného uživatele je registrace nebo přihlášení do systému.

Jedním z cílů projektu je konvertovat nové návštěvníky na aktivní uživatele. Aby tak návštěvníci učinili, musí pro ně být web atraktivní a přínosný. Nepřihlášení návštěvníci proto mohou procházet kurzy nebo ostatní produkty a vybírat si je podobně jako v katalogu nebo produkty vyhledávat. Ke každému kurzu je k dispozici náhled a informace. Nepřihlášení uživatelé mají přístupné veřejné části webu (často kladené otázky, kontakty, podmínky, o nás).

### 1.8.2 Uživatel

Aktér uživatel popisuje registrovaného uživatele, který je přihlášený na webu do svého profilu. Přihlášený uživatel má dostupné všechny funkcionality nepřihlášeného uživatele (kromě registrace či přihlášení).

Z důvodu přehlednosti bude role uživatel v této práci rozdělena na dvě samostatné role. Jednou z rolí je student a druhou mentor. Je ale důležité zdůraznit, že uživatel nabývá obou rolí najednou. Mezi rolemi se uživatelé nemusí nijak přepínat, protože fungují automaticky společně. Systém ví, jestli pro jeden konkrétní kurz je uživatel v roli mentora, studenta nebo k tomuto kurzu nemá žádnou vazbu a podle toho mu zpřístupní obsah. Každý uživatel má kromě svého profilu i vlastní statistiky a seznam získaných ocenění a odznaků. Uživatelé mohou své dosažené úspěchy (ať už jako studenti nebo jako vyučující) přidat a spravovat ve svém profilu.

### 1.8.3 Student

Student je majoritní uživatelskou rolí. Pro většinu uživatelů role jedinou, protože stejně jako v reálné třídě náleží skupině studentů jeden učitel. Hlavní činností studentů je získávání vědomostí pomocí zobrazování studijních materiálů a následně aplikování vědomostí při vyplňování testů.

### 1.8.4 Mentor

Mentorem, neboli vyučujícím se uživatel stává v okamžiku, jakmile začne vytvářet na webu kurzy nebo testy pro své budoucí studenty. Tyto dvě činnosti jsou hlavní náplní mentora. Mentoři mají k dispozici administrační rozhraní pro své kurzy, ve kterém mohou spravovat materiály a studenty svých kurzů. Správou uživatelů se zabývá uživatelský modul popsáný v diplomové práci [3].

### 1.8.5 Administrátor

Administrátor je uživatelská role s veškerými právy. V rámci modulu kurz mohou administrátoři prohlížet všechny kurzy a všechny materiály nehledě na to, jestli jsou již materiály publikované pro veřejnost. Administrátoři mohou mazat závadný obsah či blokovat celý kurz nebo uživatele. Role administrátora je globální pro celý web. Do budoucna je v plánu roli administrátora rozdělit na menší segmenty a zmenšit tak pravomoci administrátorů. Nová role se bude nazývat moderátor a bude mít na starosti například jen správu kurzů.

V rámci diplomové práce není zatím role administrátora použita.

### 1.8.6 Systém

Ačkoliv systém není rolí, kterou by uživatelé webu mohli nabývat, nelze ho opomenout, protože spouští a kontroluje procesy. Po úspěšně zvládnutém kurzu



system vypočítá a přidělí uživateli zkušenostní body Pomocí plánovačů úloh (cronů) jsou spouštěny skripty, které například definitivně mažou obsah, kterému se při smazání pouze nastavil příznak smazaný. Záznamy starší než 7 dní s tímto příznakem se systém smaže definitivně. Do uplynutí této doby lze ještě záznamy obnovit.

## 1.9 Případy užití

Jim Arlow definuje ve své knize Unifikovaný proces vývoje aplikací [9] případy užití jako zachycení funkčních požadavků. Při vytváření případů užití jsou nejprve pevně vytyčeny hranice systému a stanovení aktéři případů užití. Poté jsou případy užití vyspecifikovány včetně alternativních chování. Nakonec jsou popsány relace mezi případy užití a aktéry. Rozdělení případů užití do skupin kopíruje skupiny funkčních požadavků.

### Případy užití z pohledu mentora

#### **Vytvořit kurz**

Mentor může přidat nový kurz.

#### **Editovat kurz**

Vlastnosti kurzů mohou být editovány v administračním rozhraní pro správu kurzů.

#### **Připojit ke kurzu test**

Mentor může v administračním rozhraní kurzu přidat ke kurzu zakončení formou připojení závěrečného testu.

#### **Odpojit test od kurzu**

Závěrečný test může být mentorem smazán (odpojen) od kurzu.

#### **Publikovat kurz**

Mentor může zveřejnit své kurzy, pokud jsou splněny všechny podmínky pro zveřejnění.

#### **Stáhnout publikaci kurzu**

Mentor může kdykoliv zrušit zveřejnění kurzu a nastavit ho do stavu nezveřejněný.

#### **Zobrazit kurz**

Umožňuje mentorovi zobrazit úvodní stránku svého kurzu.

#### **Scénář**

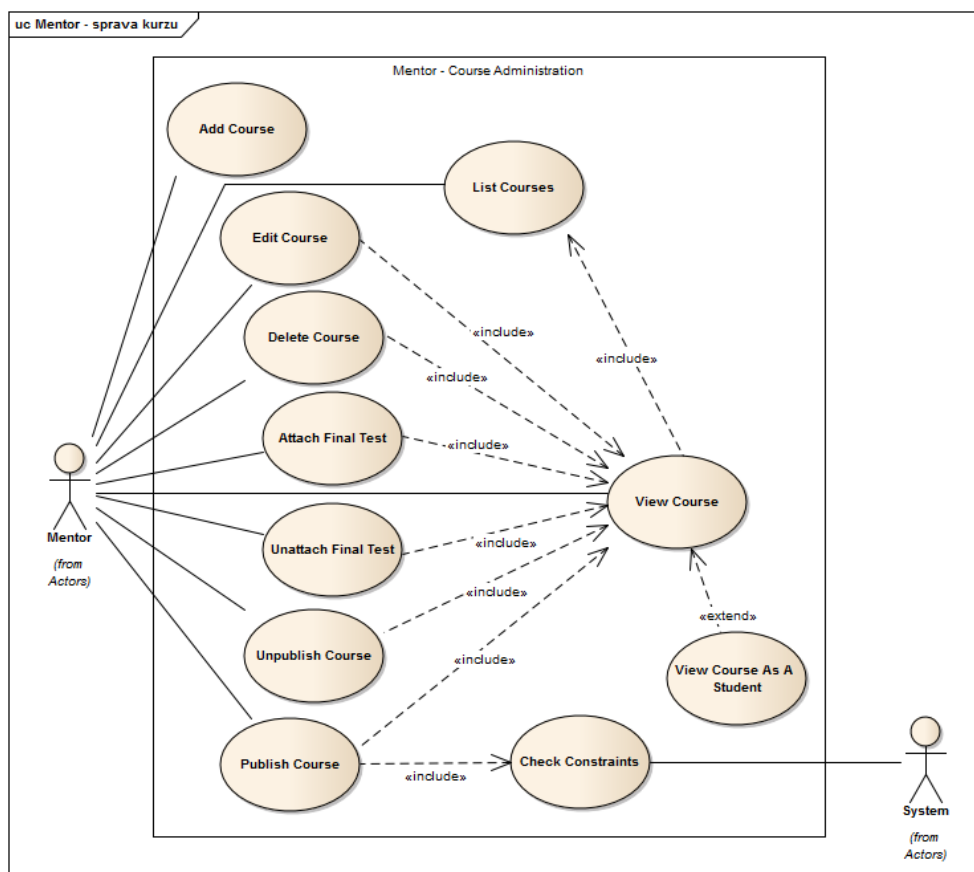
Hlavní průchod

1. Příklad užití začíná, když chce mentor zobrazit svůj kurz.
2. `INCLUDE` Příklad užití: Seznam kurzů - mentor vybere kurz který chce zobrazit

## 1. ANALÝZA

3. Systém zobrazí mentorovi kurz

4. EXTEND Příklad užití: Zobrazit kurz jako student



Obrázek 1.3: Případy užití - administrace kurzů

### Přidat lekci

Umožňuje mentorovi vytvořit novou lekci do některé kapitoly.

#### Scénář

Hlavní průchod

1. Příklad užití začíná, když chce mentor přidat lekci do některé z kapitol.
2. INCLUDE Příklad užití: Zobraz seznam lekcí
3. Systém zobrazí mentorovi uživatelské rozhraní pro přidávání obsahů do lekce.
4. Uživatel vytvoří nový obsah lekce nebo připojí soubory a stiskne tlačítko vytvořit novou lekci.

5. Systém zvaliduje odeslaná data a vrátí uživateli informaci, že lekce byla úspěšně přidána.

6. INCLUDE Případ užití: Přidej verzi

Výjimka

5b. Systém zvaliduje a vrátí uživateli chybovou hlášku, že jsou data nevalidní, kde se stala chyba a jak ji napravit

6. Případ užití pokračuje 4. bodem hlavního scénáře

### **Uprav lekci**

Mentor může v administračním rozhraní upravit všechny části lekce.

### **Smaž lekci**

Lekce může být smazána. U sousedních lekcí dojde k přepsání vazeb další a předchozí.

### **Případy užití z pohledu studenta – studium kurzu**

Nejdůležitějším funkčním požadavkem studenta je zobrazení studijních materiálů. Z toho vyplývá nejdůležitější případ užití pro studenta – zobrazit lekci.

#### **Zobrazit lekci**

Případ užití zobrazit lekci umožní zobrazit na jedné stránce všechny bloky studijních materiálů, z kterých se lekce skládá. Pokud chce uživatel zobrazit lekci, musí ji nejprve vybrat ze seznamu lekcí. Stejným způsobem musí vybrat rovněž kapitolu a kurz. Případy užití, které zajišťují výběr lekce, není díky jejich jednoduchosti třeba rozepisovat.

#### **Přidat hodnocení**

Student může přidat hodnocení ke kurzu, do kterého je zapsán. Hodnocení je na stupnici 1 -5, kde 5 je nejvíce.

#### **Upravit hodnocení**

Student může upravit hodnocení, které v minulosti kurzu udělal.

#### **Nahlásit závadný obsah**

Pokud student shledá některý z materiálů jako chybný nebo porušující zákony či zásady dobrého chování, může využít jednoduchého formuláře pro nahlášení závadného obsahu. Formulář se odešle emailem administrátorům.

### **Případy užití z pohledu studenta – před přihlášením do kurzu**

Uživatel, který nemá ke kurzu vztah studenta nebo mentora, může procházet kurzy jako v katalogu pomocí stromu kategorií, nebo je může vyhledávat. U každého kurzu si může rozkliknout jeho náhled a na produktové stránce kurzu se může v případě zájmu do kurzu přihlásit a začít ho studovat. Před přihlášením do kurzu může student provést následující akce:

#### **Vyhledat kurzy**

## 1. ANALÝZA

---

Pomocí vyhledávání lze zobrazit seznam kurzy podle vyhledávaného kritéria.

### **Zobrazit seznam kategorií**

Zobrazit seznam kategorií ve stromové struktuře, kterou lze postupně procházet.

### **Zobrazit kategorii**

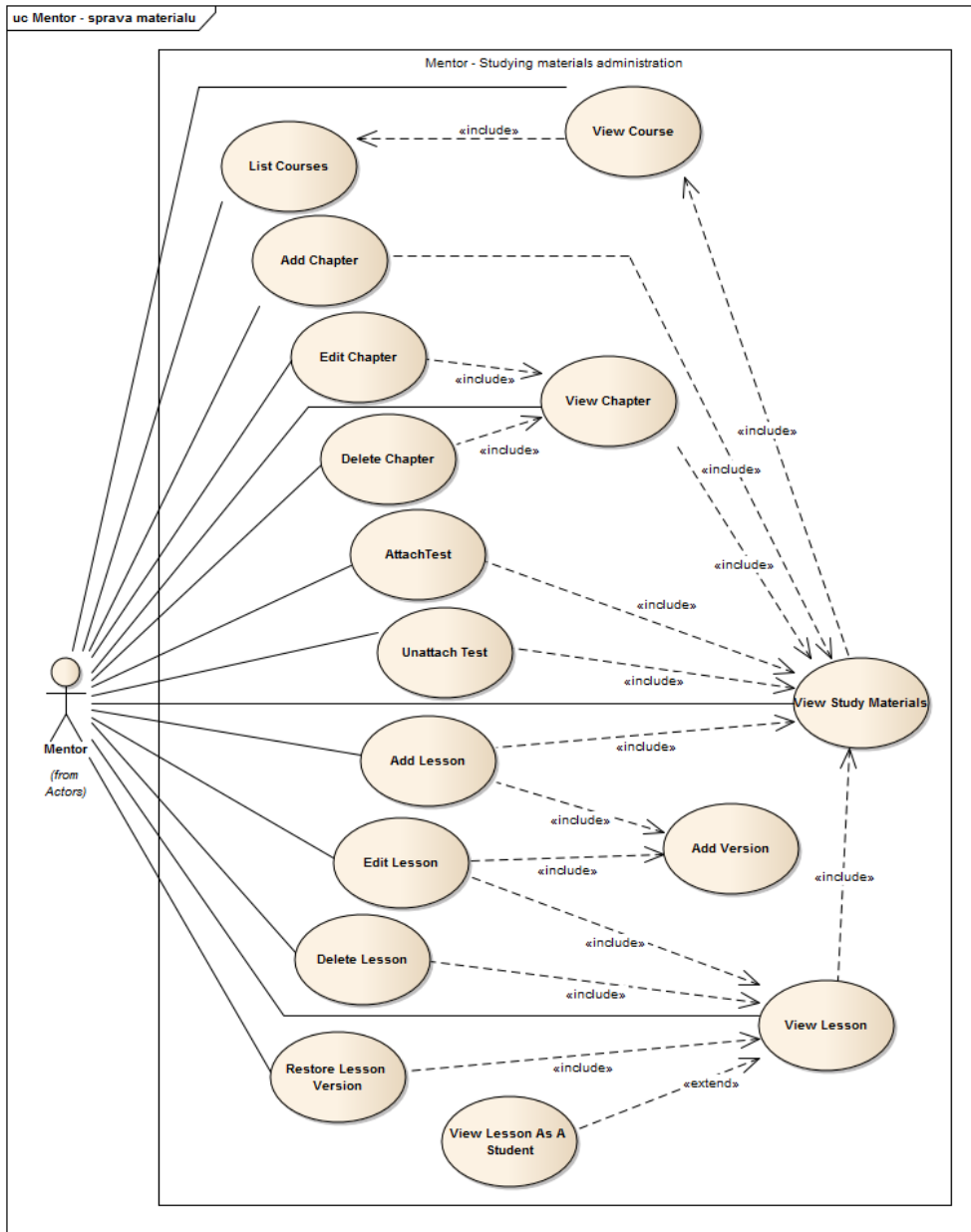
Zobrazit informace a popis o dané kategorii

### **Zobrazit náhled kurzu**

Zobrazit náhled kurzu nebo jiného produktu. Na stránce náhledu je popis a ostatní informace o kurzu.

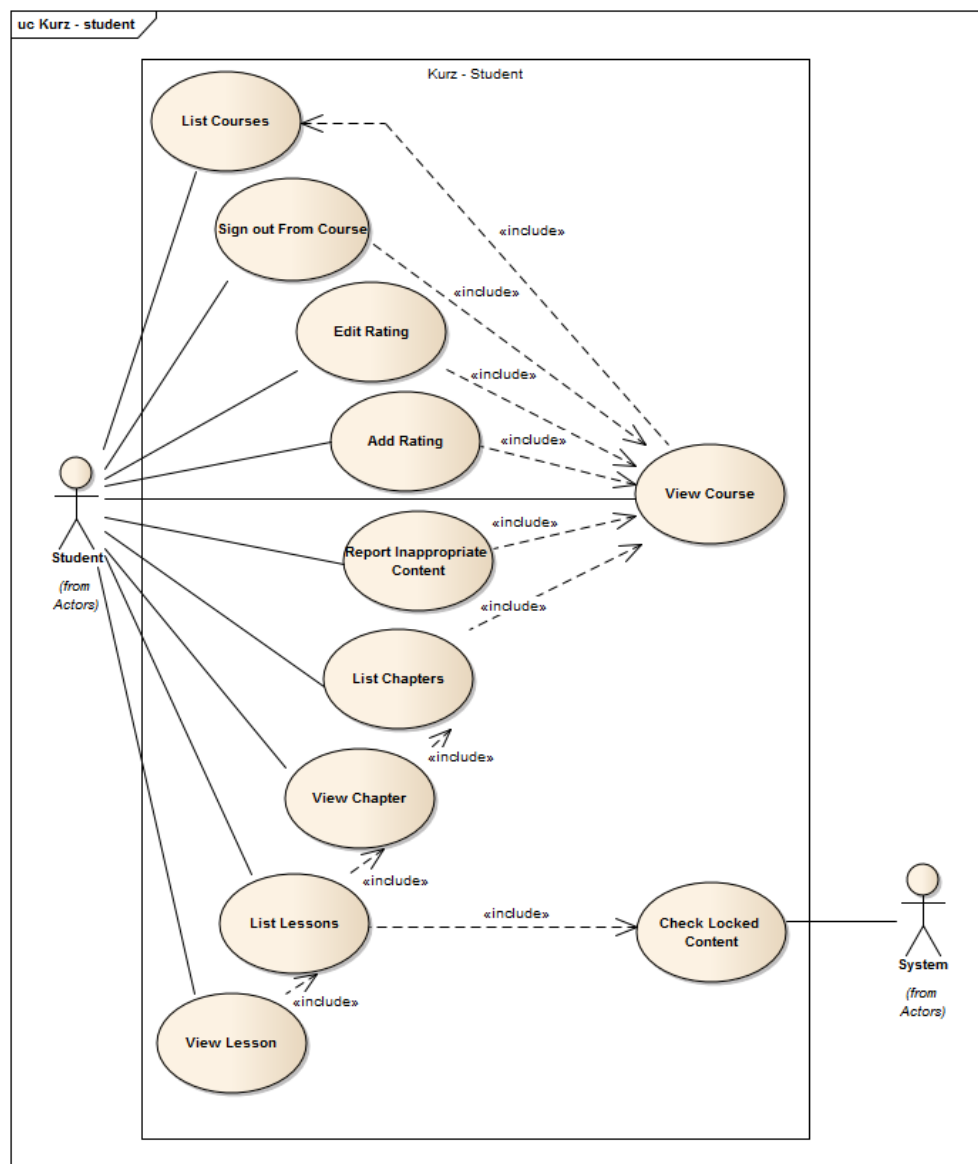
### **Přihlásit se do kurzu**

Ze stránky náhledu kurzu se lze přihlásit do kurzu. Byznys logiku přihlášek řeší modul Uživatel.

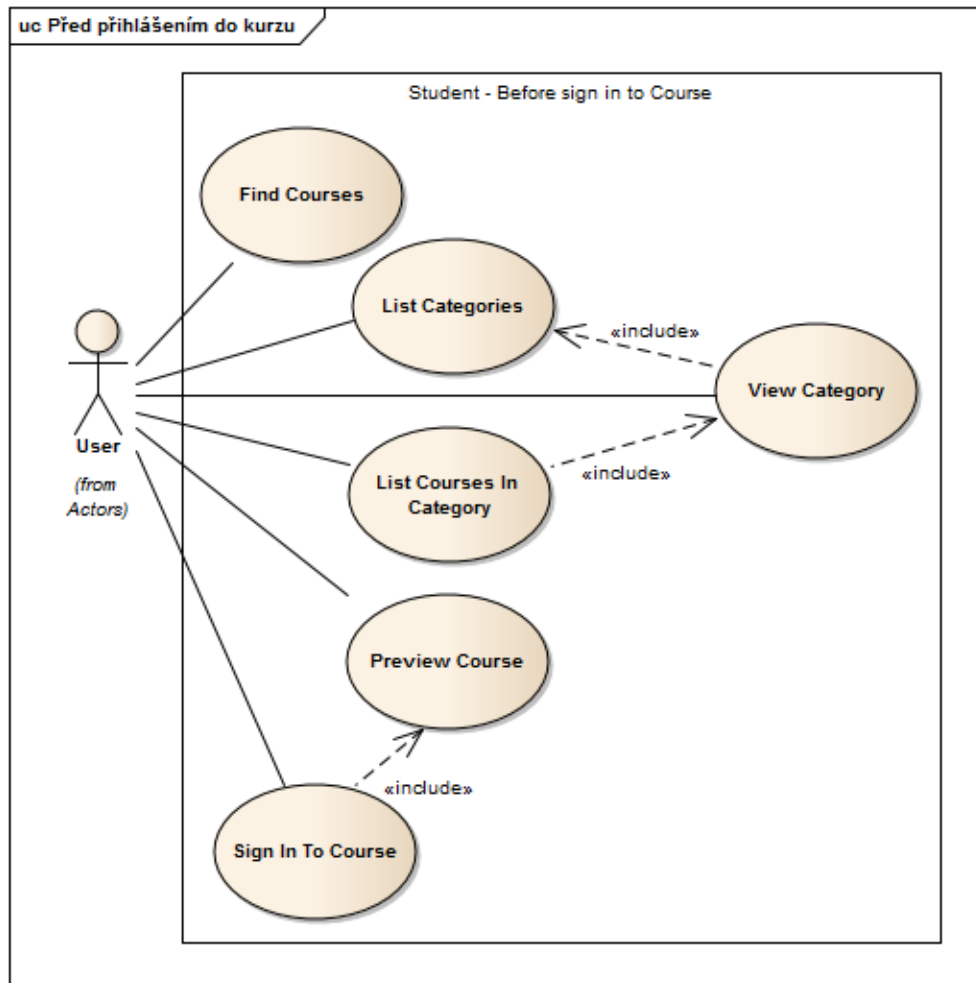


Obrázek 1.4: Případy užití - administrace studijních materiálů

# 1. ANALÝZA



Obrázek 1.5: Případy užití - studium kurzu



Obrázek 1.6: Případy užití - student před přihlášením do kurzu





---

# Návrh

Návrh řešení nutně vychází z funkčních požadavků a případů užití popisovaných v kapitole 1.6. Návrh aplikace pokrývá všechny její vrstvy, jak uživatelské rozhraní, tak strukturu aplikace. Stěžejními částmi návrhu je doménový a databázový model.

## 2.1 Doménový model

V doménovém modelu jsou třídy zobrazeny do bloků podle modulů. Bloky Test, Uživatel a Skupina nejsou na obrázku 2.1 detailně rozkresleny, protože nejsou realizovány v této práci. Modulem test se zabývá diplomová práce [2] a moduly Uživatel a Skupina se zabývá diplomová práce [3].

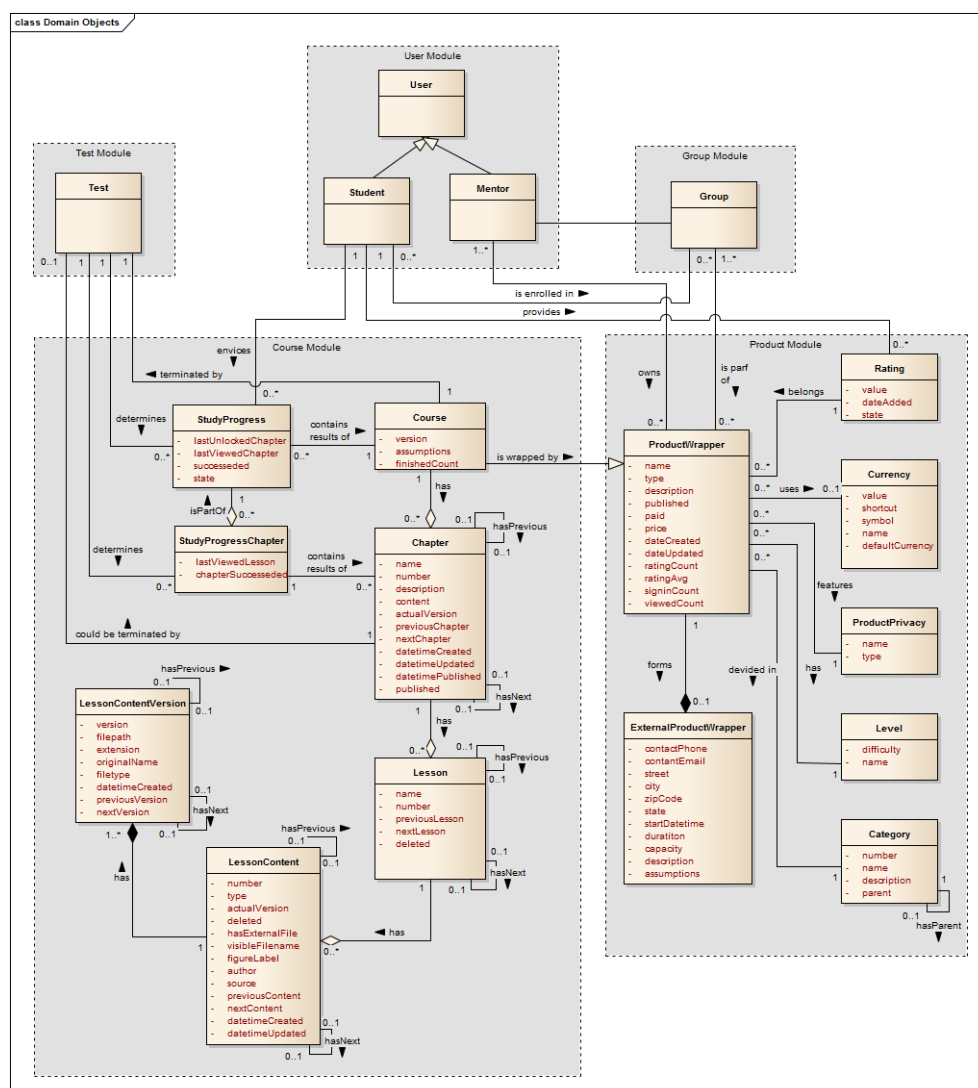
Součástí této práce je návrh a realizace modulu pro tvorbu a studium kurzů. Kurz je jedním z nabízených produktů portálu. Kvůli budoucí snadné rozšiřitelnosti byl modul rozdělen na dvě části a vznikl modul kurz a modul produkt.

Modul kurz zajišťuje pouze funkcionality týkající se studia, tedy tvorbu materiálů a jejich čerpání. Oba moduly jsou vzájemně propojeny dědičností. Kurz je potomek entity `ProductWrapper`, která poskytuje univerzální obálku pro všechny druhy produktů, nejenom kurzů. Příkladem jiného produktu může být workshop, který zatím není realizován. Entita `ProductWrapper` zajišťuje komunikaci s ostatními moduly, zejména s modulem skupina, sloužícím pro přihlašování a správu uživatelů v kurzu. Entita `ProductWrapper` má rovněž na starosti publikování studijních materiálů, jejich nastavení a shromažďuje informace, kolik uživatelů produkt vidělo, studuje nebo absolvovalo.

Z doménového modelu je patrné, že kurz se skládá z kapitol, kapitoly z lekcí a lekce z více obsahů lekcí. Každý obsah je verzovaný.

Každá z entit `Chapter`, `Lesson`, `LessonContent` a `LessonContentVersion` má dvě vazby sama na sebe. Všechny tyto entity totiž fungují jako obousměrný spojový seznam a vazby sama na sebe znamenají odkazy na předchozí, respektive další prvek v seznamu. Spojový seznam umožňuje snadné přidávání

## 2. NÁVRH



Obrázek 2.1: Doménový model modulu kurz

nového prvku, ať už nakonec, nebo doprostřed seznamu a díky němu lze například snadno měnit pořadí lekcí.

## 2.2 Databázový model

Všechny moduly aplikace jsou postaveny na jednom společném databázovém úložišti. Díky tomu je možné plně využít cizí klíče. Stejně jako v doménovém modelu jsou i na obrázku číslo 2.2 v databázovém modelu znázorněny tabulky z cizích modulů (test a uživatel), protože zejména do tabulky uživatel vedou

cizí klíče z mnoha tabulek. Uživatel je spojen se všemi částmi materiálů kurzů – produkt, kapitola, lekce, obsah lekce i verze obsahu lekce, protože při vytvoření nebo editaci (přidání nové verze) se vždy eviduje, který uživatel změnu provedl. Ke kurzům a kapitolám se v aplikaci připojují testy, tyto dvě tabulky tedy mají cizí klíč do tabulky `mtc_test`, která byla vytvořena v modulu `test`.

Databázový model je navržen a normalizován podle prvních třech normálních forem [10] včetně Boyce-Coddovi normální formy. Všechny vazby mezi tabulkami jsou typu jedna případně nula ku mnoha. Vazba mnoho ku mnoho nebyla nikde využita.

Při návrhu databázového úložiště bylo vytvořeno v tabulce `mtc_product` několik sumarizačních sloupců (`rating_avg`, `rating_count`, `signin_count`, `viewed_count`, `finished_count`). Hodnoty těchto sloupců lze dopočítat z ostatních tabulek pomocí SQL funkcí `COUNT`, `SUM`, `AVG` a sloupce jsou tedy nadbytečné. Sloupce přesto mají opodstatnění kvůli optimalizaci, jelikož data ze zmíněných sloupců jsou často využívána a nemusí být tedy přepočítávána pokaždé, když některý z uživatelů zobrazí náhled kurzu. Pokud například proběhne přidání, editace nebo smazání záznamu v tabulce hodnocení pro některý z produktů, pomocí databázových triggerů je u daného produktu automaticky změněn záznam, aby databáze zůstala vždy konzistentní.

## 2.3 Diagramy aktivit

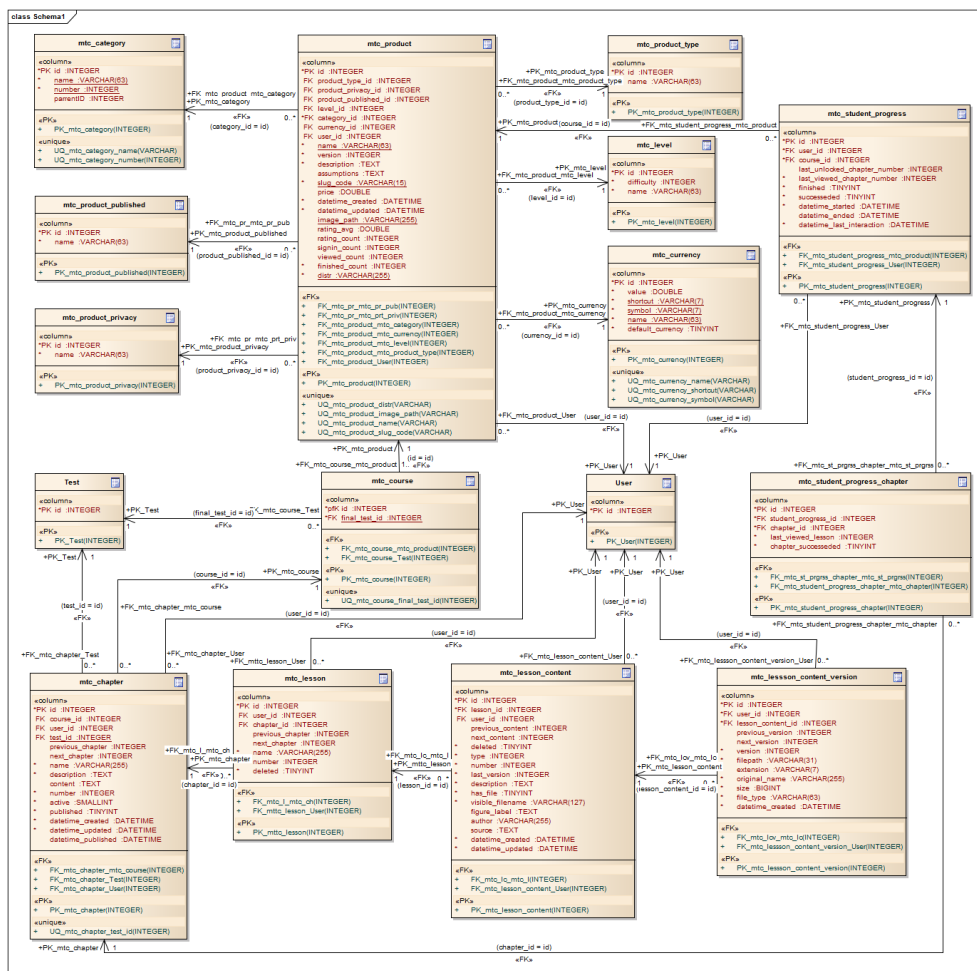
Diagramy aktivit slouží ke znázornění akcí v časové posloupnosti. Dle definice [11] je cílem diagramu srozumitelně popsat procedurální logiku byznys procesů. Diagram je horizontálně nebo vertikálně rozdělen na menší pruhy neboli plavecké dráhy (swimlines) podle aktérů, kteří souvisí s byznys procesem.

Aktivity jsou zařazeny do plaveckých drah podle aktérů, kterým náleží a jsou mezi sebou propojeny orientovanými hranami. Pořadí jednotlivých aktivit je definováno časem. Tok diagramu aktivit by měl jít vždy jedním směrem. Diagram aktivit musí vždy začínat počátečním uzlem a končit koncovým uzlem. Koncových uzlů může být více, protože uvnitř diagramů aktivit lze vytvářet větvení.

### Postup studenta kurzem

Prvním popisovaným procesem je studium kurzu z pohledu studenta zobrazené na obrázku číslo 2.3. Vstupním bodem do procesu je kladně potvrzená přihláška do kurzu, díky které je uživatel ve stavu student. Uživatel musí nejdříve projít všechny studijní materiály. Následně mu systém otevře přístup k závěrečnému testu. Od této chvíle se může uživatel kdykoliv pokusit absolvovat závěrečný test. Po vyplnění a odeslání testu systém test opraví. Pokud byl pokus úspěšný, systém označí studenta jako absolventa kurzu a přidělí mu zkušenostní body. V případě, že student úspěšný nebyl, systém vyvolá akci na základě omezení testu.

## 2. NÁVRH

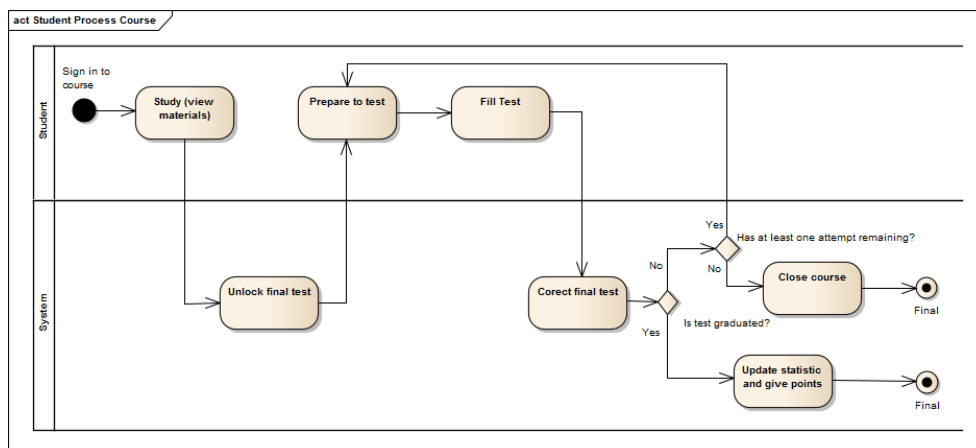


Obrázek 2.2: Databázový model modulu kurz

Systém zkontroluje, jestli je test omezený na počet pokusů a v případě že je, jestli uživateli ještě nějaké pokusy zbývají. Pokud ano, uživateli je umožněno test opakovat. Pokud uživatel už další pokus nemá, kurz je uzavřen jako nesplněný.

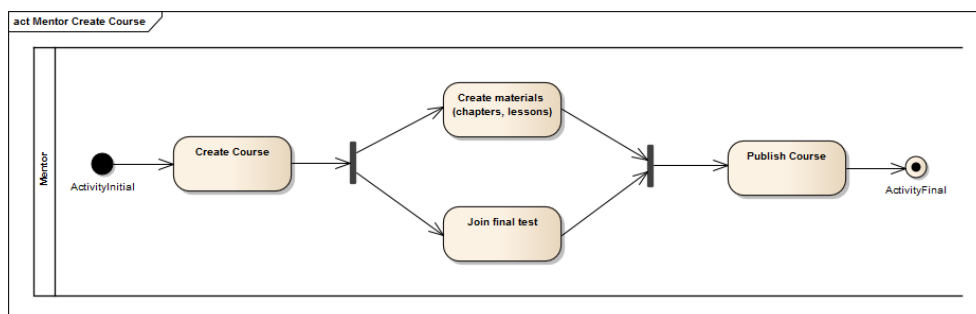
### Vytvoření a publikování kurzu

Druhým popisovaným byznys procesem je publikování kurzu mentorem zobrazené na obrázku číslo 2.4. Vstupním bodem je stav, kdy mentor chce založit a zveřejnit kurz. Cílem aktivity diagramu je naznačit, že kurz lze publikovat pouze za současného splnění dvou podmínek. Po vytvoření kurzu musí mentor v kurzu vytvořit alespoň jednu kapitolu a v ní alespoň jednu lekci a zároveň musí definovat způsob zakončení kurzu. V současné



Obrázek 2.3: Diagram aktivit - postup studenta kurzem

verzi lze kurz zakončit pouze pomocí závěrečného testu. Pokud jsou obě podmínky splněny, mentorovi se přístupné tlačítko publikovat, kterým kurz zveřejní.



Obrázek 2.4: Diagram aktivit - vytvoření a publikace nového kurzu

## 2.4 Návrh uživatelského rozhraní

Návrh uživatelského rozhraní je vždy velmi důležitý a měla by mu být věnována značná pozornost. Uživatelé komunikují s aplikací právě přes uživatelské rozhraní. Proto musí být uživatelské rozhraní co nejjednodušší a intuitivní, aby uživatelé nemuseli dlouze hledat ovládací prvky. Existuje mnoho případů, kdy optimalizace uživatelského rozhraní zvýšila společností zisk. Nejznámějším případem změny rozhraní [12] je

změna jediného tlačítka, díky které zvýšila společnost svůj obrat o 300 milionů dolarů za první rok poté, co odstranila tlačítko registrovat se z nákupního procesu v elektronickém obchodě v Americe.

Při návrhu uživatelského rozhraní musí být kladen maximální důraz na použitelnost. Web je dobře použitelný, pokud uživatelé nemusí přemýšlet, co mají v dalším kroku udělat nebo pokud musí hledat, kde to lze udělat. Rozvržení uživatelského rozhraní je v případě Menticy standardní. V horní části obrazovky je horizontální menu, které obsahuje nejdůležitější odkazy a pomocí kterých lze web ovládat. Horizontální menu je použito například u služeb společnosti Google nebo na sociální síti Facebook.

V následující kapitole budou představeny návrhy uživatelského rozhraní nejdůležitějších stránek v modulu kurz.

### 2.4.1 Návrh wireframů

Kapitola se zabývá tvorbou a popisem wireframů. Wireframy jsou jednoduché skici uživatelského rozhraní, jejichž cílem je srozumitelně rozmístit ovládací prvky stránky tak, aby uživatelé nemuseli hledat, kde se který prvek nachází a přemýšlet nad navigační strukturou webu. V této kapitole jsou představeny a zobrazeny wireframy nejdůležitějších stránek modulu kurz, zejména tvorba kurzu a materiálů.

#### Vytváření nového kurzu

Založení nového kurzu (viz obrázek číslo 2.5) je prvním stěžejním krokem každého mentora. Nový kurz se zakládá pomocí formuláře, jehož položky se musí povinně vyplnit. Jedinou nepovinnou položkou je připojení náhledového obrázku kurzu. Při nesplnění validační podmínky některé z položek se položka červeně zvýrazní a uživatel je upozorněn varovnou hláškou, která říká, jak problém vyřešit. Varovné hlášky jsou umístěny vždy napravo od nevyhovující položky.

Na formuláři je zajímavý výběr kategorie, který je kvůli přehlednosti vícekový. Stromová struktura kategorií bude mít 4 úrovně. V prvním kroku uživatel zvolí ve výběracím listu první dvě úrovně, například kategorii Angličtina zařazenou v kategorii Cizí jazyky. Pokud má kategorie Angličtina další podkategorie, uživateli se zobrazí další výběrací list, ve kterém může uživatel vybrat kategorii pro kurz přesněji, například ho zařadit do kategorie Nepravidelná slovesa, která je v kategorii Gramatika.

Formulářové prvky radio button jsou zobrazeny vedle sebe jako tlačítka či posuvné přepínače. Toto řešení se v současné době hojně používá v uživatelském rozhraní chytrých telefonů a tabletů a umožňuje uživa-

The screenshot shows a web browser window with the URL `http://mentica.com/course/add`. The page title is "Mentica.com: Add new Course". The browser's address bar shows the URL. The page header includes the "Mentica" logo, navigation links for "Courses" and "Tests", a search bar, and a user profile icon labeled "TestUser" with a "Settings" link. The main content area is titled "Create new course" and contains the following form elements:

- Course Name:** A text input field.
- Category:** A dropdown menu with "Select Category" as the placeholder.
- Type:** Two radio buttons labeled "Online" and "Full-time".
- Difficulty:** Three radio buttons labeled "Basic", "Advanced", and "Master".
- Accessibility:** Two radio buttons labeled "Public" and "Private".
- Description:** A large text area with the placeholder text "Write a full description of course ...".
- Assumptions:** A text area with the placeholder text "Define assumptions and prerequisites needed to pass the course ...".
- Paid:** Two radio buttons labeled "Free" and "Paid".
- Price:** A text input field followed by a dropdown menu set to "CZK".
- Image:** A button labeled "Choose and upload an image".

At the bottom of the form, there are two buttons: "Create course" and "Cancel". A small note at the bottom left indicates "\* - required".

Obrázek 2.5: Formulář pro vytvoření nového kurzu

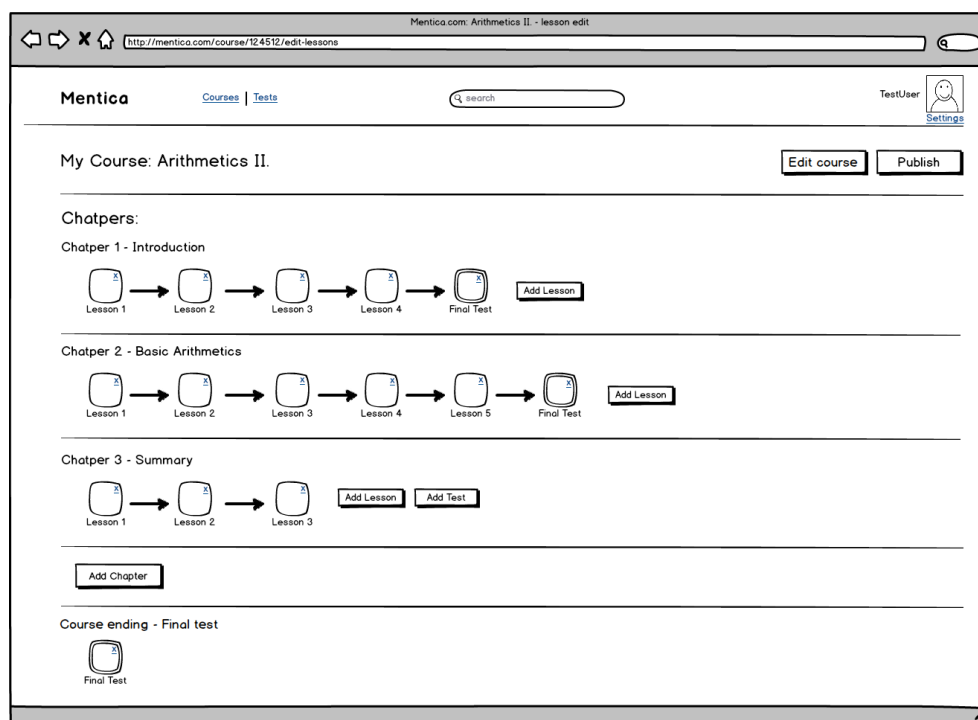
teli rychle vybírat z nabízených možností. Vždy je potřeba jednoznačně zvýraznit vybranou volbu od nevybraných. Formulář se odesílá tlačítkem vytvořit kurz, nebo se může případně zrušit tlačítkem zrušit.

### Stránka s výpisem studijních materiálů

Stránka pro výpis a editaci studijních materiálů zobrazená na obrázku číslo 2.6 je jedna z nejčastěji navštěvovaných stránek mentory. Mentoři zde vytváří a upravují kapitoly a lekce. Kapitoly jsou vypisovány vertikálně pod sebou, lekce v kapitolách jsou naopak vypsány horizontálně vedle sebe a jsou mezi nimi naznačeny propojující hrany. Do každé kapitoly lze přidat lekci pomocí tlačítka nová lekce, nebo kliknutím přímo na lekci lze editovat studijní materiály. V pravém horním rohu je křížek, pomocí kterého lze lekci smazat. Smazání lekce vždy předchází potvrzovací okno. Návrh uživatelského rozhraní pro mentory zachovává tok informací od shora dolů stejně jako zobrazování informací studentům, kteří vidí chronologicky seřazené kapitoly a pod nimi zakončení kurzu pomocí závěrečného testu. Mentoři tedy připojují závěrečný test ve spodní části stránky.

Ve vrchní části jsou tlačítka pro ovládání celého kurzu, tlačítko edi-

## 2. NÁVRH



Obrázek 2.6: Administrační rozhraní pro správu studijních materiálů

tovat kurz pro úpravy a nastavení kurzu a tlačítko zveřejnit kurz pro zveřejnění kurzu. Pokud je kurz zveřejněný, na stejném místě je tlačítko stáhnout zveřejněný kurz, který kurz opět zneaktivní.

### Přidávání nové lekce

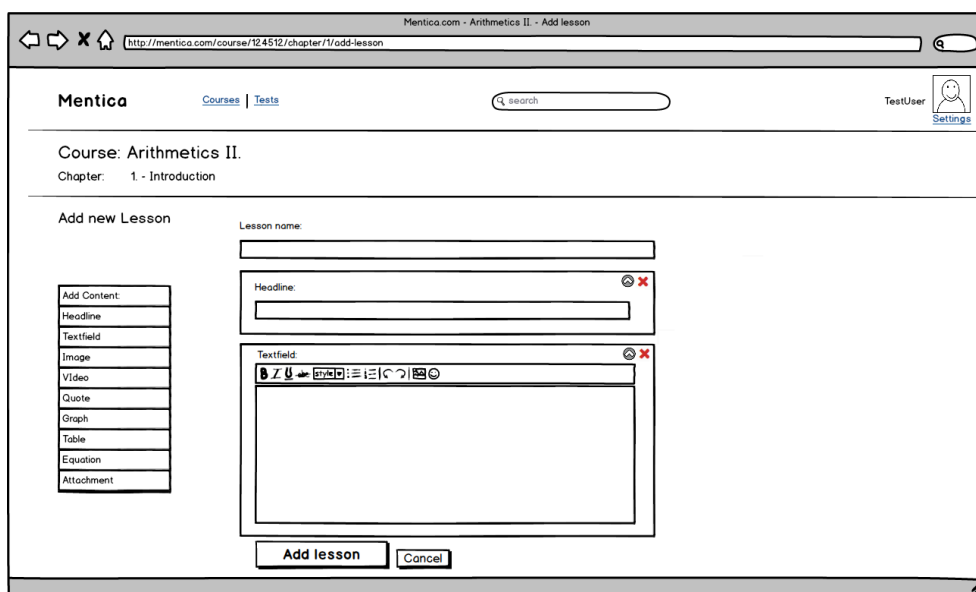
K přidávání nových lekcí slouží dynamický formulář, který má na začátku pouze jedno povinné pole pro zadání názvu lekce. Přidání nové lekce je zobrazeno na obrázku číslo 2.7. V levé části je menu pro přidávání různých druhů obsahů lekce. Po kliknutí na některou položku menu přibudou pod název lekce příslušné formulářové prvky, které jdou opět odstranit. Pokud mentor klikne na odstranění bloku lekce, blok se ihned odstraní, pokud nebyl editován (upraven text nebo přidán soubor). V opačném případě musí mentor potvrdit odstranění v dialogovém okně. Validace probíhá rovnou při vyplňování položek. Po založení nové lekce je uživatel přesměrován zpět na editaci studijních materiálů kurzu, viz obrázek číslo 2.6.

### Zobrazení lekce z pohledu studenta

Stránku zobrazení lekce z pohledu studenta zobrazenou na obrázku číslo



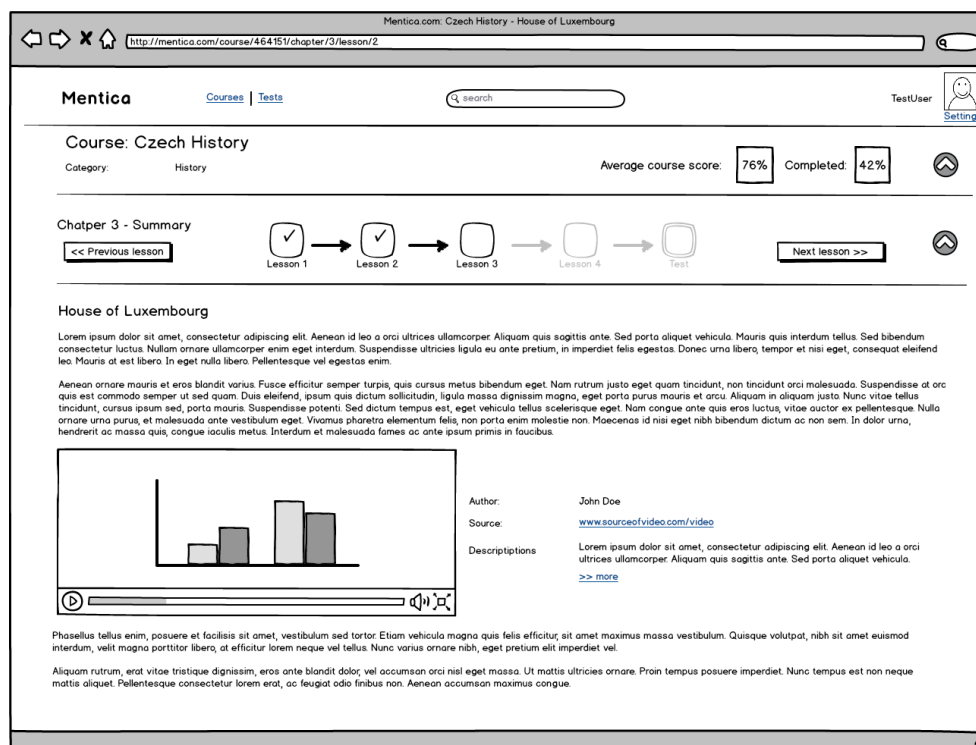
## 2.4. Návrh uživatelského rozhraní



Obrázek 2.7: Administrační rozhraní pro přidání nové lekce

2.8 lze rozdělit na tři části, které jsou pod sebou. První dvě části lze skrýt (minimalizovat), aby se zvětšil prostor pro studijní materiály. První část je pruh pro informace související s kurzem. Student zde vidí informace o kurzu a informace o svém postupu v kurzu. Druhá část je pruh se seznamem lekcí kapitoly, kterou právě student studuje. Uživatel se může rychle prokliknout do všech ostatních lekcí v rámci stejné kapitoly nebo využít tlačítko předchozí a další lekce. Poslední část je nejdůležitější, zobrazuje se v ní obsah lekce.

## 2. NÁVRH



Obrázek 2.8: Zobrazení lekce z pohledu studenta

---

# Implementace

Implementace projektu vychází z řešení navrhnutého v předchozí kapitole. V této kapitole budou popsány detaily týkající se implementace, zejména vytváření lekcí a ukládání souborů do souborového systému. V závěru kapitoly jsou rozepsané instalační a uživatelská příručka. Kapitulu uzavírá popis nasazení aplikace.

## 3.1 Použité technologie

### PHP

PHP (PHP: Hypertext Preprocessor) je nejrozšířenější [13] serverový skriptovací jazyk na světě a je použitý pro 82% webových stránek. Hlavními výhodami jazyka PHP je jeho jednoduchost a množství rozšíření, frameworků a open-source projektů, díky kterým mohou i uživatelé s malou znalostí programování vyrobit funkční dynamický web. Nevýhodou jazyka je nejednotnost jeho funkcí, které mají různé pojmenování i pořadí parametrů. Nevýhodou je také postupná transformace jazyka, protože v počátcích jazyk nebyl objektově orientovaný a nepodporoval mnoho dnes běžných funkcionalit, například výjimky. Tyto funkcionality byly přidány postupně během vývoje jazyka.

### Symfony2 Framework

Symfony2 je moderní modulární framework, který klade velký důraz na bezpečnost, flexibilitu, rychlost a nízké nároky aplikace na zdroje. Obrovskou výhodou frameworku je početná komunita uživatelů, kteří framework zásobují mnoha volně dostupnými moduly, pomocí kterých lze framework snadno rozšířit. Symfony2 je navrženo podle softwarové architektury MVC. Výběrem a porovnáním PHP frameworků se zabývá diplomová práce Bc. Jaroslava Tesaře [3].

#### **Objektově relační mapování**

Pod zkratkou ORM (objektově relační mapování) se skrývá objektově orientovaná programovací technika značně usnadňující práci s databází pomocí objektové abstrakce. Databázové tabulky jsou reprezentovány jako třídy a ORM zajišťuje převod řádků z databázové tabulky na instance třídy. Programátor tak namísto psaní SQL dotazů používá pro základní CRUD operace objekty. Díky tomu je v případě potřeby aplikace přenositelná na jinou databázi. Nevýhodou ORM je, že pro složitější úlohy nemusí generovat optimální dotazy, z čehož vyplývá větší časová náročnost aplikace.

ORM může být realizováno pomocí různých návrhových vzorů. Nejjednodušším z nich je Active Record popsáný v knize Patterns of Enterprise Application Architecture [14], ve kterém je pro každou tabulku vyrobena třída. Sofistikovanějším řešením je použití návrhového vzoru Data Mapper, který rozděluje práci s databází do několika vrstev, minimálně se používají mapovací objekty, které zachovávají strukturu databáze a doménové objekty, které jsou od databáze zcela oddělené. Všechny moduly aplikace využívají knihovnu objektově relačního mapování Doctrine2. Mapování je detailně popsáno v kapitole 3.2.1.

#### **Gaufrette**

Souborový systém Gaufrette je popsán v kapitole 1.4.2, která se věnuje jak souborovým systémům obecně, tak popisem vlastností knihovny Gaufrette.

#### **MySQL**

Popisem databázového systému MySQL se v úvodní analýze detailně zabývá kapitola 1.5.

#### **AJAX**

AJAX (Asynchronous JavaScript and XML) je asynchronní volání požadavků na serveru vyvolané klientem pomocí jazyka JavaScript. Hlavní výhoda AJAX požadavku je jeho zpracování a odpověď serveru bez nutnosti obnovení stránky a opětovného stažení všech zdrojů. Jednoduchou práci s technologií AJAX zajišťuje knihovna jQuery.

#### **JSON**

JSON (JavaScript Object Notation) je datový formát, pomocí kterého lze uchovávat a přenášet data mezi platformami. JSON je způsob, kterým lze libovolný datový typ zapsat pomocí jednoduché syntaxe závorek a oddělovačů. Výsledný řetězec je dobře čitelný pro člověka i počítače. Jednoduchosti zápisu dat JSON využívá například AJAX, když je třeba vrátit klientovi data ze serveru.

### JavaScript + jQuery

JavaScript je objektově orientovaný programovací jazyk, který je běžně používaný jako součást webových prohlížečů. Nejběžnější použití JavaScriptu je jeho začlenění do HTML kódu. V současné době lze ale JavaScript využít například pro vytvoření škálovatelného serveru pomocí JavaScriptové platformy `node.js`<sup>7</sup>. Primárním použitím JavaScriptu však stále je reagovat na akce uživatelů na webu, měnit HTML DOM nebo například validovat data.

Nejvýznamnější JavaScript knihovnou je jQuery, která je v dnešní době naprosto běžnou součástí mnoha webů, jelikož ulehčuje práci v jazyce JavaScript, zkracuje a zjednodušuje funkcionality jazyka. Pomocí jQuery lze snadno zachytávat akce uživatele, měnit části HTML stránek nebo asynchronně komunikovat se serverem pomocí volání AJAX dotazů.

### HTML

HTML neboli hypertextový značkovací jazyk slouží k tvorbě webových stránek. Značkovací jazyk je množina značek (tagů), pomocí kterých jsou popsány dokumenty na webu. Každá značka má vlastní význam. Značky mají atributy a jsou do sebe hierarchicky zanořené. Pořadí a syntaxe značek je definována jazykem DTD (Document Type Definition), který popisuje validitu HTML stránek nebo jiných XML dokumentů.

### CSS

Kaskádové styly (CSS) definují formu vzhledu zobrazení dat popsaných v jazyce HTML. Stejná data mohou být zobrazena úplně jinak pomocí dvou různých kaskádových stylů. Výhodou použití kaskádových stylů je oddělení vzhledu a obsahu zobrazovaných dokumentů do dvou nezávislých vrstev. Díky kaskádovým stylům je například možné upravit a vytvořit vzhled HTML stránky pro zobrazení na stolních počítačích nebo chytrých telefonech.

## 3.2 Databáze

V následující kapitole je popsána a názorně ukázána práce s ORM knihovnou Doctrine 2, automatická validace pomocí mapování entit a generování databázového úložiště vycházejícího z mapování entit.

### 3.2.1 Mapování a validace entit

Díky Doctrine ORM jsou všechny entity modelové vrstvy mapovány anotacemi. Mapování znamená, že u všech atributů každé entity jsou předem

<sup>7</sup>JavaScriptový framework `node.js` je dostupný na adrese: <https://nodejs.org>

nadefinovány datové typy, propojení s jinými atributy, chování v systému a omezení entit. Při vytváření entit je potřeba věnovat čas preciznímu nastavení všech atributů, což v budoucnu ušetří mnoho starostí. V praxi to znamená, že se programátor nemusí starat o validaci dat vstupujících do databáze, protože ORM samo hlídá datové typy, omezení a cizí klíče a data validuje automaticky, takže databáze je vždy v konzistentním stavu. Spojením PHP validace s validací databáze do jednoho validačního prvku vzniká jednotný robustní nástroj pro validaci.

Jednou z obrovských výhod Symfony frameworku je komponenta console, která značně zjednodušuje programátorům práci. Pomocí konzole může být například vygenerována třída, kontrolér nebo vytvořen a nastaven celý balíček (bundle). Symfony console umožňuje důmyslně pracovat se směřováním URL adres a usnadňuje práci s databází. Databázové úložiště lze vytvořit a poté efektivně spravovat pouze konzolí. Po nakonfigurování databáze a vytvoření entit a jejich mapování stačí provést v konzoli příkaz `php app/console doctrine:schema:update`, který zkontroluje současný stav databáze a porovná ho s mapováním. Následně vytvoří a provede SQL příkazy potřebné pro úpravu databázového schématu tak, aby odpovídal mapování. Pomocí jednoho příkazu tak lze reflektovat změny v kódu do struktury databáze. Příkaz nemění data v tabulkách a sloupcích, ve kterých nedochází ke změnám, takže je relativně bezpečné ho použít.

Následující ukázka části kódu entity `Lesson` ukazuje práci s mapováním pomocí anotací a definování validačních podmínek. V příkladu je zobrazen primární klíč, cizí klíč a atribut název, u kterého jsou nastaveny omezující podmínky. Anotace jsou uvedeny přímo v kódu třídy jako komentáře s prefixem `@ORM`, validace má prefix `@Assert`. Zapsání anotací je možné několika způsoby, kromě komentářů například do vlastního konfiguračního souboru XML.

Atribut `ID`, který je primárním klíčem tabulky lekce má tuto skutečnost anotovanou pomocí prefixu `@ORM\Id`. Při propojování entit a vytváření cizích klíčů lze nastavit jednosměrnou nebo obousměrnou vazbu. Obousměrná vazba znamená, že v obou entitách je atribut, ve kterém je uložen objekt respektive kolekce objektů druhé entity. Dle následující ukázky kódu je v entitě kapitola atribut `$lessons`, který je kolekcí objektů všech lekcí kapitoly. Vše o propojování entit je popsáno v dokumentaci ORM Doctrine 2 [15]. K propojení entit slouží anotace `@ORM\JoinColumn`. U atributu `$name` jsou definovány podmínky, pomocí kterých bude entita validována. Pomocí několika málo řádků komentářů lze docílit validace, která hlídá typ atributu řetězec a jeho délku v rozsahu 3-63 znaků. Navíc lze definovat i výjimky, které se budou zobrazovat

při porušení jednotlivých pravidel. Validační podmínky atributů se nastavují pomocí syntaxe `@Assert\TypPodminky()`. ORM Doctrine2 umí pracovat s nejdůležitějšími validačními datovými typy, například řetězec, číslo, datum soubor a jiné.

```

1 class Lesson
2 {
3 ...
4     /**
5      * @var integer
6      * @ORM\Column(name="id", type="integer")
7      * @ORM\Id
8      * @ORM\GeneratedValue(strategy="AUTO")
9      */
10    private $id;
11
12    /**
13     * @ORM\ManyToOne(targetEntity="Chapter", inversedBy="lessons")
14     * @ORM\JoinColumn(name="chapter_id", referencedColumnName="id")
15     */
16    protected $chapter;
17
18    /**
19     * @var string
20     * @Assert\Length(
21     *     min = 3,
22     *     max = 63,
23     *     minMessage = "Nazev musi mit delku alespon {{ limit }} znaku!",
24     *     maxMessage = "Nazev musi mit delku maximalni {{ limit }} znaku!"
25     * )
26     * @ORM\Column(name="name", type="string", length=255)
27     */
28    private $name;
29 ...
30 }

```

### 3.3 Propojení modulů

Kapitola propojení modulů popisuje technické řešení, pomocí kterého je realizována výměna informací napříč jednotlivými moduly (bundles) v rámci Symfony frameworku. V kapitole 3.3.2 je popsáno řešení kurzu jako samostatného produktu a integrace testů do modulu kurz.

#### 3.3.1 Propojení modulů pomocí services

I když mají všechny moduly aplikace jednu společnou databázi, s každou databázovou tabulkou může pracovat vždy pouze jeden ten mo-

dul, v jehož modelu jsou vytvořeny třídy entity a repositáře. Základní struktura entit a poté i databázové tabulky byly vygenerovány pomocí `Symfony console`. Díky tomuto pravidlu nemůžou v aplikaci existovat duplicitní třídy entit. Některé funkcionality jednoho modelu však musí být propojeny s jiným modelem, protože si potřebují vyměňovat informace a volat metody. K tomuto účelu je v `Symfony frameworku` vytvořen `Services Container`, který poskytuje modulu rozhraní pro spolupráci s jinými moduly. V každém modulu je vytvořena `Service`, která globálně zpřístupňuje funkcionality modelu pro jiné moduly.

#### 3.3.2 Propojení modulu produkt a test

Jak již bylo zmíněno, každý kurz musí být zakončen závěrečným testem. Každá kapitola může být zakončena testem. V zadání projektu byl požadavek, že portál má umět pracovat s testy jako se samostatnými produkty, do kterých se uživatelé přihlašují stejně jako do kurzu.

Z hlediska implementace byl tento požadavek vyřešen přidáním nové hodnoty do atributu `type` v entitě produkt. Samostatný test je v rámci aplikace kurz, který nemůže mít studijní materiály a má připojený závěrečný test. Tímto řešením se nemusí nijak měnit byznys logika ohledně vyplnění testu, přihlašování uživatelů k testu nebo publikování samostatného testu. Z pohledu uživatele se samostatný test jeví jako produkt, který nemá s kurzem nic společného.

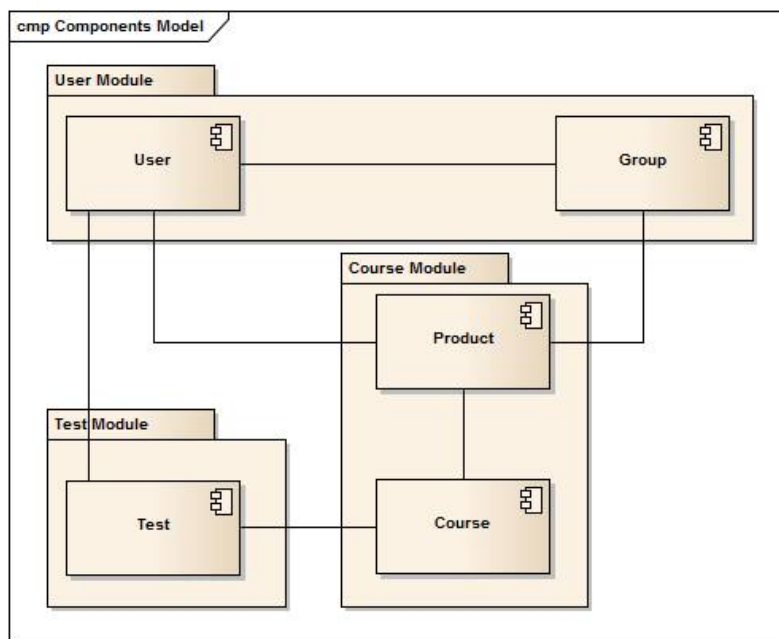
#### 3.3.3 Diagram komponent

Diagram komponent popisuje interakci mezi jednotlivými moduly napříč celou aplikací. Komponenty mají oddělenou funkcionality a komunikují mezi sebou pomocí `services`. Z obrázku číslo 3.1 vyplývá, že modul Kurz zabývající se tvorbou produktů a kurzů se nachází uprostřed mezi ostatními moduly. Do kurzu patří testy, které jsou díky produktu odstíněny od byznys logiky týkající se produktů (zveřejňování, vyhledávání, správa) a skupin (vytváření a správa pozvánek a uživatelů).

### 3.4 Lekce

Lekce jsou základním pilířem modulu kurz, protože uchovávají studijní materiály. Každá lekce se může skládat z více různých částí, například textu a několika obrázků. Analogií je textový editor, například MS Word, kde v rámci jedné stránky lze mezi text vkládat obrázky, tabulky atd. Lekce se skládá z 1 až n bloků, kde maximální počet je nastaven na 99 a lze v případě potřeb kdykoliv změnit v konfiguraci.





Obrázek 3.1: Diagram komponent modulů Mentica

Různé druhy obsahů lze rozdělit do dvou skupin podle toho, jestli obsahují externí soubory. První skupinou jsou ty bloky, které mají externí soubor. Patří do ní obrázky, videa a přílohy. Pokud uživatel nahraje skupinu obrázků, každému z nich se iterativně vytvoří vlastní obsah lekce. Naopak ve druhé skupině uživatelé externí soubory nenahrávají a pouze tvoří materiály pomocí webového rozhraní. Typickým příkladem je text nebo tabulka. Data jsou ukládána do textových souborů pomocí souborového systému. Ve výsledku má každý obsah lekce svůj soubor.

Implementačně je vše řešeno pomocí dědičnosti, kde předkem všech druhů obsahů je třída `LessonContent`, která poskytuje metody, pro práci s jednotlivými částmi lekce. Třída `LessonContent` má několik abstraktních atributů. Atribut `lessonContentType` definuje druh obsahu lekce a atribut `hasExternalContent` určuje, jestli obsah lekce pracuje s externím souborem od uživatele. Díky dědičnosti lze do budoucna snadno vytvářet nové druhy obsahů bez zásahu do logiky aplikace.

### 3.4.1 CKEditor

Lekce spojené s textem (text, tabulka, matematický vzorec, citace) jsou řešeny pomocí WYSIWYG<sup>8</sup> editoru CKEditor<sup>9</sup>. CKEditor je dostupný zdarma a jednou z jeho hlavních výhod je jeho přizpůsobivost, jelikož je plně konfigurovatelný podle potřeb vývojářů. Základní podoba a funkce CKEditoru jsou podobné textovým editorům, například MS Word. CKEditor nabízí definovat povolené formátování textu. Po odeslání formuláře je obsah zvalidován pomocí Advanced Content Filter, který kontroluje povolené formátování a přípustné HTML tagy a veškeré nepovolené řetězce ze vstupu smaže. Pro textové lekce bylo využito několik rozšíření, například TableTable je rozšíření umožňující tvorbu tabulek dostupné na: <http://ckeditor.com/addon/table> nebo MathJax<sup>10</sup>.

## 3.5 URL adresy

V kapitole jsou popsány důležité informace o URL adresách. Jak URL adresy vznikají, co je to směrování a jak probíhá a vliv URL adres na optimalizaci pro vyhledávače.

### 3.5.1 Přívětivé URL a optimalizace pro vyhledávače

Hezké neboli přívětivé URL adresy (nice URLs) jsou dnes běžným standardem poskytujícím výhody uživatelům i počítačům. Pro uživatele je přívětivější URL adresa ve tvaru [www.mentica.com/kurz/gemoetrie](http://www.mentica.com/kurz/gemoetrie) namísto adresy [www.mentica.com?kurz=123](http://www.mentica.com?kurz=123), protože mu přináší lepší představu, jaký obsah se na dané URL adrese nachází. Z hlediska optimalizace pro vyhledávače (SEO) pomáhá správný tvar URL dosáhnout lepšího hodnocení (ranking) stránky a díky tomu vyšší pozici ve vyhledávacích. Pro SEO je důležité, aby URL adresa obsahovala klíčové slovo stránky - zde kurz a název kurzu geometrie. Při zadání dotazu do vyhledávače Google je ve výsledcích vidět nadpis stránky, popis i cílová URL adresa. Slova z hledané fráze jsou navíc ve výsledcích tučně zvýrazněna, takže jsou pro uživatele lépe vidět.

Použití názvu článku nebo produktu k vytvoření hezké URL adresy je běžný postup. Je třeba však myslet na situaci, že mentoři své kurzy

---

<sup>8</sup>Způsob editace dokumentů, ve kterém je vzhled při editaci totožný s výslednou podobou

<sup>9</sup>CKEditor je HTML editor pro úpravu textů, dostupný na adrese: <http://ckeditor.com/>

<sup>10</sup>MathJax je rozšíření umožňující tvorbu matematických vzorců dostupné na: <http://ckeditor.com/addon/mathjax>

přejmenují. Pro zachování pozic ve vyhledávačích je při změnách URL adres nutné přeměřovat starou adresu na novou pomocí http kódu 301.

Systém přidělí každému kurzu unikátní identifikátor, podle kterého je v systému jednoznačně určen. Tento identifikátor není stejný s identifikátorem záznamu v databázi a má konstantní délku. URL adresa pro kurzy má tento tvar: `www.mentica.com/kurz/123456-nazev-kurzu`. Při změně názvu kurzu zůstává slug nezměněn. Systém se při výběru kurzu neřídí názvem, ale pouze slugem, takže přejmenování kurzu nemá vliv na změnu adresy.

### 3.5.2 Směrování

Nedílnou součástí moderních MVC frameworků je možnost použití routování (směrování) URL adres. Jinak tomu není ani u Symfony2 frameworku, routování je zde samostatnou vrstvou aplikace. Pod pojmem routování je myšlen oboustranný překlad mezi akcemi MVC vrstvy controller a URL adresami. Podle tvaru URL adresy je zavolána příslušná metoda v konkrétním controlleru podle routování a naopak, systém může vygenerovat URL adresu podle zadané akce.

Při vývoji aplikace se vývojář nemusí vůbec zabývat tvarem URL adres, protože všechny lze nastavit a editovat později. Celá konfigurace URL adres se nastavuje v jednom konfiguračním souboru, takže při změnách není třeba zasahovat do kódu.

## 3.6 Souborový systém

V úvodní analýze vznikl požadavek, že má aplikace obsahovat souborový systém, který umožní jednotným způsobem ukládat data nezávisle na úložišti. Pro účely vývoje diplomové práce je použito lokální úložiště, které se nachází na pevném disku webového serveru. V budoucnu se počítá se počítá se záměnou úložiště, protože správa lokálního systému by byla náročná na místo na pevném disku i údržbu. Daleko lepším řešením je použít namísto lokálního úložiště některé z cloudových řešení. V současné době je na trhu na výběr cloud od několika společností včetně technologických gigantů jako Google, Amazon nebo Microsoft.

Použití cloudů přináší řadu výhod, například zabezpečení a automatickou zálohu dat, téměř neomezený datový prostor s platbou pouze za využívanou část a vysokou dostupnost. Pomocí streamování souborů lze navíc převést datový tok ze serveru na cloudové servery. Jako cloudové úložiště bylo zvoleno Amazon 3S. Díky adaptérům stačí pro přepnutí lo-

kálního úložiště na cloud pouze změnit konfiguraci a nastavit bezpečností tokeny.

#### 3.6.1 Bezpečnost

Z hlediska bezpečnosti [16] je bezpodmínečně důležité, aby uživatelé neměli přímý přístup k souborům. Souborový systém Gaufrette je abstraktní vrstva, která umožňuje aplikaci přistupovat k souborům nehledě na jejich fyzickém umístění pouze pomocí konfigurace adaptéru a libovolnému nastavení kořenové složky. To znamená, že se soubory nemusí nacházet ve veřejném adresáři webu podobně jako kaskádové styly nebo JavaScripty. O ukládání a načítání souborů se vždy stará Gaufrette. V případě použití lokálního adaptéru je nutné zabezpečit kořenovou složku souborového systému pomocí souboru `.htaccess`, ve kterém je nastaveno odepření přímého přístupu.

Stejně jako při odesílání formuláře se nelze spoléhat na data od klienta, nelze důvěřovat ani původním názvům u souborů. Různé souborové systémy mohou mít na jména souborů odlišné požadavky týkající se povolených znaků, délky souborů nebo rezervovaných řetězců, které není pro název možné použít. Generování názvů musí podléhat pravidlům, aby byl výsledek nezpůsoboval chyby či bezpečnostní hrozby. Součástí názvů souborů jsou i přípony, které musí být kontrolovány také. Textové řetězce přípon jsou kontrolovány na délku, převáděny na malá písmena a porovnávány vůči povoleným typům souborů. Seznam povolených přípon:

- Obrázky: `bmp`, `jpg`, `jpeg`, `png`, `gif`, `svg`
- Videá: `avi`, `vmw`, `mp4`
- Ostatní: `doc`, `docx`, `xls`, `xlsx`, `ppt`, `pps`, `csv`, `pdf`

#### 3.6.2 Generování názvů souborů

Kromě bezpečnosti existuje ještě jeden významný důvod, proč přejmenovávat soubory nahrávané uživateli. Jména včetně adresářové struktury (cesty k souboru) mají zásadní vliv na výkon souborového systému. Soubory budou uloženy v souborovém systému `ext4`. Dle specifikace [17] tohoto souborového systému v něm může být uloženo maximálně 4 miliardy ( $2^{32}$ ) souborů v neomezeně mnoha adresářích. S velkým množstvím souborů lze pracovat efektivněji, pokud jsou hierarchicky členěny do adresářů a podadresářů. V článku [18] se píše, že při použití souborového

sytému `ext4` se operace jako `ls` nebo `find` citelně zpomalují, pokud je ve vypisovaném adresáři řádově tisíce souborů nebo více.

Generování názvů souborů proto musí být propracované, aby byly jednotlivé adresáře stejně velké a byly zaplňovány rovnoměrně. V úvahu připadá několik variant:

### Hashování ID

První variantou generátoru názvů souborů je použití hašovací funkce. Vhodnou funkcí je například MD5, která z libovolně dlouhého vstupu dělá konstantě dlouhý výstup složený ze znaků používaných v hexadecimální soustavě. Generování názvu souboru by vypadalo následovně. Vstupem do generátoru je ID řádku z databáze, ve kterém jsou uloženy informace o souboru. Z tohoto ID se pomocí hašovací funkce vytvoří výstupní řetězec, jenž se rozdělí na několik částí. Prvních 8 znaků se použije na 4 dvojice, které určují hierarchii adresářů, ve které je soubor uložen. Jako název souboru se použije celý výstup hašovací funkce. Výhodou tohoto řešení je, že každá konečná složka by mohla obsahovat nanejvýše 256 souborů. Nevýhodou je časté hašování pro přístup k souborům.

### Skládání parametrů

Druhou variantou jak sestavit název souborů je složení několika řetězců obsahujících pouze číslice. Po spojení všech částí by vznikl výsledný řetězec dlouhý 12 znaků, který by posloužil jak pro název souboru, tak i pro vygenerování adresářové struktury, na kterou by byly použity první 3 trojice znaků.

Generovaný název by se skládal z několika částí částí:

Typ	Pozice znaků	Popis
Typ souboru	0	Logické zařazení souborů dle číselníku
ID	1 - 8	Identifikátor konkrétní části lekce
Verze	9 - 11	Číslo verze souboru

Tabulka 3.1: Skládání parametrů

Příklad vygenerování názvu souboru ze vstupních parametrů:

ID: 00006412, verze: 2, přípona: jpg, typ: 1

Výsledná cesta a název souboru: 100/006/412/100006412002.jpg

Tento způsob generování názvů a cest souborů přináší výhodu v dobré čitelnosti pro lidské oko. Oproti tomu nevýhodou je nerovnoměrné zaplnění souborů v jednotlivých složkách. Výhodnější je změnit pořadí jednotlivých částí. Díky zachování postupu při generování a prohození identifikátoru a verze se docílí rovnoměrného a postupného zaplňování souborového systému. V každém podadresáři může být maximálně 1000 souborů a při zaplnění adresáře se vytvoří adresář další. Zároveň budou od sebe odděleny soubory pro kurz, test a uživatele, protože mají unikátní hodnotu v typu souboru. Drobnou nevýhodou je, že ve stejných složkách u sebe nebudou různé verze jednoho souboru, nicméně pro aplikaci to není rozdíl díky přístupu k souborům klíč – hodnota.

Názvy souborů jsou v aplikaci generovány pomocí posledního popísaného postupu.

#### 3.6.3 Použití souborového systému Gaufrette

Práce s knihovnou je jednoduchá a intuitivní. Stačí vytvořit objekt adaptéru, který má za úkol připojení do souborového systému podle konfigurace pomocí tohoto objektu vytvořit souborový systém, na kterém lze provádět metody jako čtení nebo zápis souboru. Knihovna pracuje stejně jako hašovací mapa na principu klíč a hodnota, kde klíčem je cesta k souboru. Příklad čtení a zápisu souborů je ukázán v následujícím kódu:

```
1 // include namespaces
2 use Gaufrette\Filesystem;
3 use Gaufrette\Adapter\Local as LocalAdapter;
4
5 // inicialize filesystem and adapter
6 $adapter = new LocalAdapter('filesystem');
7 $filesystem = new Filesystem($adapter);
8
9 //write file from form and create new file
10 $filename1 = "path/to/file.txt";
11 $filename2 = "path/to/file2.txt";
12
13 $file = file_get_contents($file->getPathname()); // file from form
14 $filesystem->write($filename, $file);
15
16 $file2 = "Lorem ipsum dolor sit amet."; // file from new string
17 $filesystem->write($filename2, $file2);
18
19 //read file
20 $filesystem->get($filename); // return Object Gaufrette/File
```

## 3.7 Instalační příručka

Pro spuštění aplikace na webovém serveru Apache je potřeba provést následující kroky:

1. Pomocí protokolu SFTP zkopírovat složku `mentica` se zdrojovými kódy do adresáře `www` v Apache serveru.
2. K dalšímu kroku je potřeba mít správně nainstalovaný `composer`<sup>11</sup>. Pro aktualizaci závislostí definovaných v konfiguračním souboru `composer` stačí spustit v terminálu příkaz `php composer.phar update`.
3. V konfiguračním souboru `httpd.conf` na serveru Apache nastavit kořenový adresář (`root`) do složky `web` zkopírované v kroku 1.
4. (Nepovinné) V terminálu lze otestovat konfiguraci serveru pro projekty implementované ve frameworku `Symfony2`. V terminálu stačí zadat testovací příkaz: `php app/check.php`.
5. Ve složce `app` povolit právo zápisu složkám `cache` a `logs`.
6. Vytvořit novou MySQL databázi, pojmenovanou `mentica` a nového databázového uživatele.
7. V konfiguračním souboru `app/config/parameters.yml` je třeba změnit údaje o připojení k databázi nadefinované v předchozím kroku.
8. Pomocí webového rozhraní (například `phpmyadmin` nebo `adminer`) nebo pomocí terminálu importovat SQL soubor `mentica.sql`. Soubor obsahuje všechna potřebná data pro spuštění aplikace.
9. Na URL adrese Apache serveru aplikaci spustit. Pokud byly předchozí kroky provedeny v pořádku, výsledkem by měla být úvodní stránka projektu.

Po úspěšné instalaci je potřeba zaregistrovat se a vytvořit si tak svůj účet. Poté již může uživatel začít portál využívat. Uživatelé mají k dispozici uživatelskou příručku, která jim usnadní práci s aplikací. Uživatelská příručka je popsána v příloze G.

---

<sup>11</sup>Composer je správce závislostí pro PHP dostupný na adrese:<https://getcomposer.org>





## Testování

Testování je jedním ze základních životních cyklů každého softwarového projektu, které přichází na řadu po implementaci. Otestování aplikace před jejím ostrým nasazením je důležité, protože může odhalit chyby aplikace, které mohly nastat při implementaci nebo návrhu.

Existuje celá řada technik testování, pomocí kterých lze otestovat některou část aplikace. Testování aplikace lze rozdělit do následujících skupin:

### **Funkční testy**

Funkční neboli akceptační testování ověřuje, zdali aplikace úspěšně dokončuje všechny naimplementované funkcionality. Úspěšnost funkčních testů má zásadní vliv na bezproblémový provoz aplikace. Před provedením funkčních testů musí být vytvořen návrh funkčních testů definující jeho podmínky. Funkční testy jsou navrženy v příloze F. Akceptační test se hodnotí podle kritéria splněno / nesplněno.

### **Testování použitelnosti**

Dobrá použitelnost webu je jedním ze základních pilířů úspěšnosti webu. Pouze spokojení uživatelé se budou na web vracet. Dobrá použitelnost znamená, že uživatelé nemusí při práci s webem přemýšlet nebo hledat navigační prvky a dokáží se snadno orientovat. Použitelnost webu úzce souvisí s návrhem uživatelského rozhraní. Testování použitelnosti lze rozdělit na několik základních skupin.

První skupinou je testování s uživateli, které je náročné na čas i zdroje. Testování probíhá ve speciální laboratoři, musí se důkladně připravit a musí se sehnat skupina participantů, kteří budou aplikaci testovat. Testování s uživateli se věnuje kapitola 4.1.

Kvalitně testovat použitelnost lze i bez uživatelů. Existují dvě základní metody. První z nich je heuristická evaluace se skládá z deseti

bodů (heuristik), které definoval ve své knize Usability inspection methods [19] Jacob Nielsen. Při použití heuristické evaluace se testovaná část aplikace podrobí všem deseti heuristikám. Případné nalezené problémy se klasifikují a sepíší do výstupního dokumentu. Problémy jsou poté odstraňovány podle priorit.

Druhou metodou testování bez uživatelů je kognitivní průchod, ve kterém se simuluje průchod aplikací uživatelem. V Kognitivním průchodu se specifikuje bod, který má být dosažen fiktivním uživatelem. Uživatel postupuje po krocích a v každém kroku zodpovídá několik otázek, které zjišťují, jestli si stanovil správný cíl pro další postup, jestli našel kudy postupovat, jestli je provedená akce správná a jestli uživatel rozumí zpětné vazbě po provedení akce. Odpovědi na všechny otázky v každém kroku se zapisují do tabulky, z které pak snadno vyplyne, kde nastaly během testování nejasnosti.

Zvýšit použitelnost aplikace může také AB testování, které klade velké nároky na počet testovaných uživatelů. Princip je jednoduchý, uživatelé jsou rozděleni na dvě zhruba stejně velké skupiny. Každé skupině je zobrazována jiná varianta webu (například jiná barva nebo velikost tlačítek). Během testu se měří akce obou variant, například proklik testovaného tlačítka. Po nasbírání dat se mohou obě varianty porovnat proti sobě a říci, která varianta funguje lépe. Nevýhodou testování je nutný dostatečně velký vzorek uživatelů a dlouhá doba měření, aby se mohlo měření prohlásit za statisticky věrohodné.

### **Testování korektnosti implementace**

K ověření správné funkcionality kódu se využívají jednotkové testy (Unit tests). Jednotkou testu je často metoda nebo třída a test má na starosti prozkoumat její funkčnost pomocí testů složených ze vstupů různých kombinací.

Správnost HTML kódu můžeme snadno zkontrolovat pomocí online validátorů, například <http://validator.w3.org>. Zkontrolována by také měla být kompatibilita HTML stránek napříč různými zařízeními a internetovými prohlížeči.

### **Testování zabezpečení aplikace**

Maximální zabezpečení by měla být prioritou každé aplikace, obzvláště pokud aplikace pracuje s daty svých uživatelů. Testování bezpečnosti by mělo být zaměřeno na autorizaci přístupu k datům a na známé webové útoky (SQL Injection, Cross site scripting, Cross Site Request Forgery, a další).

### **Testování zátěže**

Testování zátěže má za úkol zjistit, jak se aplikace chová při velkém zatí-

žení, které simuluje reálné uživatele. Při implementaci aplikace obvykle žádná zátěž není a nezatížený systém funguje bezproblémově. Při zátěži může dojít k snížení rychlosti aplikace a testování může objevit úzké hrdlo aplikace, kterým může být například zamykání tabulek v databázovém systému nebo špatně navržený SQL dotaz.

## 4.1 Testování s uživateli

Testování s uživateli probíhá ve speciální laboratoři použitelnosti, do které jsou postupně přivedeni předem vybraní participanti, kteří se snaží splnit zadané úkoly. Interakce a chování uživatelů jsou zaznamenávány pomocí audiovizuálních záznamů, které vyhodnocuje tým expertů ve vedlejší místnosti. Častou chybou při testování je poskytnutí nápovědy participantům, jelikož je tím zkeslen výsledek testování. Tým testerů vyhodnocuje chování a interakci participantů při plnění úkolů a snaží se odhalit chyby v použitelnosti. Chybou může být, pokud participanti v nějakém kroku nevědí jak pokračovat, například jim nedojde, že musí kliknout na tlačítko, které je špatně viditelné. S testováním souvisí několik dotazníků. První z nich, screener sloužící k vybrání množiny participantů. Participanti také dostanou před a po testování krátké dotazníky, které testování upřesňují.

### 4.1.1 Dotazník Screener

Screener je dotazník, který má za úkol vybrat vhodné kandidáty (participanty) pro testování s uživateli. Každému potenciálnímu uchazeči jsou položeny jednoduché otázky a podle odpovědí může být uchazeč vybrán k testování. Dotazník screener se skládá ze dvou částí. Kromě veřejné části, která je umístěna v příloze B.1 pro uživatele má každý screener neveřejnou část, v které je definována požadovaná skupina participantů. Neveřejná část screeneru je připojena k práci v příloze B.2.

Hlavním úkolem při testování bude otestovat uživatelské rozhraní pro tvorbu kurzů. Ve výsledné skupině participantů by proto měli být alespoň 3 participanti, kteří mají pedagogickou zkušenost. Z hlediska vzdělání by měli mít všichni participanti větší dosažené vzdělání než základní. Do množiny participantů není vhodné zařadit příliš zkušené uživatele, protože by mohli projít testováním bez problémů a potenciální chyby aplikace by nemusely být odhaleny. Z tohoto důvodu byli z přípustné množiny participantů vyřazeni kandidáti, kteří mají profesionální zkušenosti s informačními technologiemi nebo již aktivně využívají e-learningový portál.

### 4.1.2 Pre-test dotazník

Před samotným testováním participantů vyplňují dotazník, jehož cílem je ověřit, jaký mají participantů přehled o oboru, kterého se testovaná aplikace týká. Dotazník by měl participanty připravit na samotné testování. Informace z pre-test dotazníku jsou vyhodnocovány spolu se samotným testováním a pomáhají testerům pochopit chování participanta. Pre-test dotazník se přiložek k práci v příloze C.

### 4.1.3 Post-test dotazník

Posledním úkolem participantů je vyplnit post-test dotazník. Participantů dostanou dotazník po skončení testování. Cílem dotazníku je zjistit zpětnou vazbu, tedy jaký dojem na participanty testovaná aplikace udělala, jestli jim přišlo testování obtížné či nikoliv. Zde je na místě nechat participantům volný prostor pro vyjádření vlastního názoru. Stejně jako ostatní dotazníky i post-test dotazník je umístěn v příloze D.

## 4.2 Testování funkčnosti

Testování funkčnosti vychází z případů užití. Ke všem případům užití, které je potřeba otestovat se vyrobí scénáře testování. Každý scénář může zahrnovat několik testovacích případů, vždy však musí obsahovat alespoň jeden. Testovací případ se skládá z několika kroků, kde krok je vždy jedna akce, kterou může uživatel vykonat. Každý krok má svůj vlastní identifikátor, popis kroku, očekávaný výsledek, skutečný výsledek a výsledné ohodnocení. Testovací případy se pro přehlednost zapisují do tabulek.

### 4.2.1 Scénáře

V této kapitole jsou popsány vybrané testovací scénáře, které mají otestovat nejdůležitější případy užití. Cílem všech scénářů je otestovat veškeré chování, které může v systému při interakci s uživateli nastat.

#### **Scénář 1: Vytvoření nového produktu**

Založení nového kurzu je velmi důležitý úkol, protože všechny ostatní úkoly na něj navazují. Uživatelé mají možnost vytvořit nový vlastní kurz pomocí formuláře pro přidávání kurzů. Testovací případy musí pokrývat všechny varianty, které mohou při vytváření nového kurzu nastat. Testovací případy jsou uvedeny v příloze F.1.1 a F.1.2.

#### **Scénář 2: Vytvoření nové kapitoly**

Založení nové kapitoly je pro uživatele i systém jednoduchý úkol, který je však potřebný pro další scénáře. Pro založení kapitoly je třeba vyplnit jednoduchý formulář, do kterého uživatel zadá název a popis kapitoly.

**Scénář 3: Vytvoření nové lekce**

Vytváření lekcí je nejčastější a nejdůležitější scénář, který je třeba otestovat. Uživatelé mohou vytvářet lekce skládající se několika z bloků různých studijních materiálů. Lekce musí být bezpečně a správně uloženy v aplikaci. Pokud by nastal jakýkoliv problém, systém o něm musí přehledně informovat uživatele. Testovací případ je uveden v příloze F.2.1.

**Scénář 4: Publikování kurzu**

Zveřejnění kurzu pro ostatní uživatele je posledním krokem cyklu vytvoření nového kurzu. Kurz může být publikován, pouze pokud má studijní materiály a připojený závěrečný test. Testovací případy jsou uvedeny v příloze F.3.1 a F.3.2.

**Scénář 5: Editace kurzu**

Uživatelé musí mít možnost změnit parametry kurzu či jeho nastavení. Při pokusu o uložení nových chybných dat nesmí být špatná data uložena.

**Scénář 6: Editace studijní materiálů kurzu**

Každý studijní materiál musí jít v systému změnit. Lekce se skládají z bloků, které mohou být přidány, změněny nebo nahrazeny bloky jiných druhů. Aplikace musí zajistit bezproblémovou editaci studijních materiálů. Scénáře



---

# Závěr

Cílem práce bylo vytvoření modulu kurzů do modulární aplikace Mentica. Na začátku práce byly provedeny rozsáhlé analýzy požadavků na modul, konkurenčních řešení a technologií. Na základě získaných poznatků a požadavků od zadavatele bylo zhotoveno zadání projektu, které bylo zformulováno pomocí funkčních požadavků. Dalším krokem v životním cyklu projektu byl návrh modulu, který zahrnoval jak návrh uživatelského rozhraní, tak návrh softwarového řešení pomocí grafických notací UML. Implementace byla provedena v PHP frameworku Symfony2 a vycházela z diagramu tříd a diagramu databáze. Poslední kapitolou je testování, které bylo dle přání zadavatele pouze navrženo a zadavatel si aplikaci otestuje sám.

Kapitoly analýza a návrh lze považovat za hotové, stejně jako návrh testů použitelnosti a funkčního testování. V části implementace nebyly z časových důvodů dokončeny všechny funkcionality. Aplikace lze v současném stavu považovat za prototyp, v kterém jsou nejdůležitější části funkční, nicméně některé funkcionality chybí. Splněné a nesplněné cíle jsou podrobně rozepsány v následující kapitole.

## Splněné a nesplněné cíle

Každý cíl bude samostatně ohodnocen jako splněný, částečně splněný nebo nesplněný, podle funkcionalit, které jsou v současné době kompletní.

### **Administrační rozhraní pro správu kurzů**

Uživatelé (mentoři) mají možnost správy svých kurzů. Mentoři mohou vytvářet vlastní nové kurzy či editovat nebo mazat stávající. Předtím, než je kurz dostupný studentům, musí být mentorem zveřejněn. Správa a publikace kurzů jsou funkční.

Cíl splněn

### **Vytváření studijních materiálů**

V kurzech lze vytvářet, editovat a mazat kapitoly a lekce. Při smazání neprázdné kapitoly se kapitole pouze nastaví příznak smazaný a skutečně je vymazána až po 7 dnech. Při vytváření lekcí lze lekci složit z libovolného množství studijních bloků. Základními bloky jsou formátovatelný text, video a obrázek. U videa bylo v plánu přidávat privátní videa ze serveru youtube.com, v současné aplikaci lze pouze přidávat vlastní videa. Zatím nebyla naimplementována změna pořadí lekcí v rámci jedné kapitoly. Model aplikace je na změnu pořadí připravený. V klientské části zatím není v jazyku JavaScript naimplementován Drag and Drop sloužící pro změnu pořadí.

Cíl splněn částečně

### **Čerpání obsahu z pohledu studenta**

Výukový kurz je z pohledu studenta až na drobné detaily funkční. Uživatel může procházet kapitolami a lekcemi. Aplikace dovolí studentovi zobrazit pouze materiály kapitol, které má student odemčené pomocí splněných testů. Aplikace zatím kontroluje přístup k materiálům podle poslední odemčené kapitoly. Do budoucna je plánována pokročilejší kontrola.

Cíl splněn částečně

### **Integrace s ostatními moduly**

Modul kurz byl propojen s moduly uživatel a test. Integrace s modulem test spočívá v možnosti přidání testu ke kapitole nebo závěrečného testu ke kurzu. Splnění testu se promítne do studijního progresu studentů kurzu, například může odemknout zamčenou kapitolu. Díky propojení s modulem uživatel je zajištěna autorizace uživatelů v systému. Pomocí skupin, které jsou součástí modulu uživatel, se studenti mohou přihlásit ke studiu v kurzu a mentor může studenty spravovat.

Cíl splněn

### **Verzování studijních materiálů**

Ačkoliv verzování studijních materiálů nebylo v zadání práce, bylo rozhodnuto verzování všech studijních materiálů přidat. Verzování úzce souvisí se souborovým systémem, kterému se věnuje kapitola 3.6. V současném stavu aplikace umí verzovat studijní materiály. Při vytváření nebo editaci studijních materiálů je po uložení lekce vždy vytvořena nová verze



souboru pro každý blok lekce. Bohužel zatím není funkční rozhraní, v kterém by šly verze prohlížet, porovnávat a vracet se ke starším verzím souborů.

Cíl splněn částečně

## Budoucí práce

Ve všech modulech připadá v úvahu mnoho rozšíření, které by mohly portál vylepšit. Velmi důležitým rozšířením aplikace je přidání modulu gamifikace, který by přinesl studentům motivaci ke studiu a kreditový systém, díky kterému by mohli studenti čerpat výhody. V uživatelském profilu by mohl být vytvořen online chat pro komunikaci mezi uživateli. V úvahu také připadá implementace vztahů mezi uživateli realizovaná formou přátelství či sledování.

Modul kurz lze rozšířit o funkcionality při vytváření studijních materiálů, například přidáním nových druhů studijních materiálů při tvorbě lekcí, nebo vylepšení nástrojů pro práci se stávajícími materiály. Například by šlo zajistit snadnější práce s obrázky pomocí implementace knihovny umožňující editaci obrázků online. Profilové obrázky kurzů by mohly být stejně jako obrázky studijních materiálů v lekcích snadno vyhledatelné a připojitelné ke kurzu pomocí Flickr API či jiné podobné služby. Vkládání velkého množství souborů by mohlo být ulehčeno propojením modulu kurz s cloudem Google Drive či jiných podobných řešeních od konkurenčních společností.



---

## Literatura

- [1] Český statistický úřad: Studenti a absolventi vysokých škol v ČR. 2013. Dostupné z: [https://www.czso.cz/documents/10180/23202203/1\\_vs\\_studenti\\_celkem.pdf](https://www.czso.cz/documents/10180/23202203/1_vs_studenti_celkem.pdf)
- [2] Budnik, B. O.: *E-learningový portál Mentica - modul správa testů*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií.
- [3] Tesař, B. J.: *E-learningový portál Mentica - modul uživatelský modul*. Diplomová práce, České vysoké učení technické v Praze, Fakulta informačních technologií.
- [4] author =: .
- [5] stackoverflow.com: When to use MyISAM and InnoDB? 2013. Dostupné z: <http://stackoverflow.com/questions/15678406/when-to-use-myisam-and-innodb>
- [6] Vrána, J.: Rychlost vkládání do InnoDB tabulek. 2008. Dostupné z: <http://php.vrana.cz/rychlost-vkladani-do-innodb-tabulek.php>
- [7] Wikivs.com: MySQL vs PostgreSQL. 2015. Dostupné z: [https://www.wikivs.com/wiki/MySQL\\_vs\\_PostgreSQL](https://www.wikivs.com/wiki/MySQL_vs_PostgreSQL)
- [8] randombugs.com: MySQL vs PostgreSQL Benchmarks. 2009. Dostupné z: <http://www.randombugs.com/linux/mysql-postgresql-benchmarks.html>
- [9] Arlow, J.: *UML 2 and the Unified Process: Practical Object-Oriented Analysis and Design (2nd Edition)*. 2005, ISBN 978-0321321275.

- [10] studytonight.com: Normalization of Database. Dostupné z: <http://www.studytonight.com/dbms/database-normalization.php>
- [11] Rejnková, P.: Diagram aktivit. 2009. Dostupné z: [http://uml.czweb.org/diagram\\_aktivit.htm](http://uml.czweb.org/diagram_aktivit.htm)
- [12] SPOOL, J. M.: The \$300 Million Button. 2009. Dostupné z: [http://www.uie.com/articles/three\\_hund\\_million\\_button/](http://www.uie.com/articles/three_hund_million_button/)
- [13] <http://w3techs.com/>: Historical trends in the usage of server-side programming languages for websites. 2015. Dostupné z: [http://w3techs.com/technologies/history\\_overview/programming\\_language](http://w3techs.com/technologies/history_overview/programming_language)
- [14] FOWLER, M.: *Patterns of Enterprise Application Architecture*. 2002, ISBN 0-321-12742-0.
- [15] 2, D.: *Doctrine 2 ORM's documentation*. 2015. Dostupné z: <http://doctrine-orm.readthedocs.org/en/latest/>
- [16] stackoverflow.com: Stop people uploading malicious PHP files via forms. 2009. Dostupné z: <http://stackoverflow.com/questions/602539/stop-people-uploading-malicious-php-files-via-forms/602904#602904>
- [17] Štrauch, A.: Ext4: evoluční souborový systém. 2008. Dostupné z: <http://www.root.cz/clanky/ext4-evolucni-souborovy-system/>
- [18] stackoverflow.com: Storing & accessing up to 10 million files in Linux. 2011. Dostupné z: <http://stackoverflow.com/questions/5019371/storing-accessing-up-to-10-million-files-in-linux>
- [19] NIELSEN, J.: *Usability inspection methods*. 1994, ISBN 04-710-1877-5.

## Seznam použitých zkratk

- URL** Uniform Resource Locator
- HTML** HyperText Markup Language
- PHP** PHP: Hypertext Preprocessor
- OOP** Object-oriented programming
- ORM** Object-relational mapping
- MVC** Model–view–controller
- JSON** JavaScript Object Notation
- AJAX** Asynchronous JavaScript and XML
- MS** Microsoft
- CSS** Cascading Style Sheets
- SEO** Search engine optimization
- PDF** Portable Document Format
- DOM** Document Object Model
- WYSIWYG** What you see is what you get



## Screener

### B.1 Veřejná část screeneru

Vaše pohlaví?

- Muž
- Žena

Váš věk?

- Méně než 15 let
- 15 - 20 let
- 21 – 30 let
- 31 – 40 let
- Více než 40 let

Vaše nejvyšší dosažené vzdělání?

- Základní vzdělání
- Střední odborné vzdělání s výučním listem
- Střední odborné vzdělání s maturitou
- Vysokoškolské nebo vyšší odborné vzdělání

Vaše zkušenosti s počítačem?

- Nedostatečné – počítač nedovedu ovládat

- Základní – s počítačem pracuji zřídka, avšak dokážu si najít informace na Internetu
- Pokročilý – běžně pracuji s počítačem
- Profesionál – pohybuji se v oboru IT

**Máte zkušenost s pedagogickou činností?**

- Ano
- Ne

**Používáte některý z online výukových portálů?**

- Ano
- Ne

**Online vzdělávání hodnotím v porovnání s klasickou výukou jako:**

- Nezájímavé, raději preferuji klasickou formu vzdělávání
- Oba druhy jsou pro mě zajímavé zhruba stejně
- Zajímavé

## **B.2 Neveřejná část screeneru**

**Vaše pohlaví?**

- V cílové skupině mají být zastoupeny obě pohlaví.

**Váš věk?**

- Méně než 15 let nežádoucí, zbylé skupiny zastoupeny rovnoměrně

**Vaše nejvyšší dosažené vzdělání?**

- 3x vysokoškolské, zbytek alespoň střední odborné

**Vaše zkušenosti s počítačem?**

- Nedostatečné zkušenosti nežádoucí, profesionální zkušenosti maximálně 1x, zbytek libovolně

**Máte zkušenost s pedagogickou činností?**



- 3x ano, 3x ne

**Používáte některý z online výukových portálů?**

- Přijatelná pouze odpověď ne

**Online vzdělávání hodnotím v porovnání s klasickou výukou jako:**

- Odpověď nezajímavé je nežádoucí



## Pre-test dotazník

**Jste zaregistrovaný na některé sociální síti?**

- Ano
- Ne

**Který formát online přednášky preferujete?**

- Video přednáška
- Textová přednáška
- Interaktivní přednáška

**Kolik času týdně věnujete studiu nebo sebevzdělávání se?**

- Méně než 2 hodiny
- 2 až 5 hodin
- 5 až 15 hodin
- Více než 15 hodin

**Slyšel jste někdy pojem gamifikace? Dokážete ho popsat?**

- -----

**Co očekáváte od online vzdělávací aplikace?**

- -----

**Účastnil jste se někdy testování použitelnosti?**

- -----



## Post-test dotazník

**Přišlo Vám testování obtížné?**

- Ano
- Spíše ano
- Nevím
- Spíše ne
- Ne

**Je podle Vás testovaná aplikace dostatečně přehledná?**

- Ano
- Spíše ano
- Nevím
- Spíše ne
- Ne

**Naplnila aplikace Vaše očekávání?**

- Ano
- Spíše ano
- Nevím
- Spíše ne
- Ne

Co Vám přišlo na testování nejtěžší a proč?

- -----

Co se Vám na aplikaci nejvíce líbilo a nelíbilo?

- -----

Chyběla Vám v aplikaci nějaká funkce?

- -----

Využili byste v budoucnu podobnou aplikaci?

- -----

---

# Testování s uživateli - popis úkolů

## E.1 Vyučující

**Prerekvizity:** Uživatel je přihlášen do systému a je autorem alespoň jednoho testu. Na ploše počítače jsou předem připraveny vzorová data (texty, obrázky, videa). Přihlášený uživatel dříve vytvořil několik testů, které se dají připojit ke kurzům.

Jste v roli vyučujícího, který chce pro své studenty vyrobit online kurz, do kterého se budou moci studenti zapisovat.

1. Vytvořte nový kurz. Téma kurzu jsou české dějiny ze středověku.
2. Kurz se skládá ze dvou částí. Vytvořte 6 lekcí, které budou rozděleny do dvou skupin. Každá skupina bude mít 3 lekce. Ve složce data, která se nachází na ploše, jsou vzorové materiály lekcí (texty, obrázky, videa).
3. Do jedné lekce přidejte video soubor.
4. Z libovolné lekce smažte blok textu.
5. Libovolně změňte název kurzu.
6. Připojte ke kuru závěrečný test.
7. Publikujte kurz.

## E.2 Student

**Prerekvizity:** Uživatel je přihlášen do aplikace a zároveň aplikace obsahuje několik výukových kurzů.

Nyní budete v roli studenta, Vaším cílem bude vybrat si a vystudovat některý z nabízených kurzů.

1. Prohlédněte si veřejně dostupnou nabídku kurzů, některý kurz si vyberte a zobrazte jeho detailní informace.
2. Zapište se ke studiu kurzu.
3. Zkuste projít první studijní kapitolou a zobrazte všechny studijní materiály.



## Funkční testy - scénáře

### F.1 Scénář 1 - vytvořit kurz

#### F.1.1 Testovací případ 1 - validní data

Testovací případ: Vytvořit kurz - validní data				
Krok	Popis	Očekávaný výsledek	Skutečný výsledek	Hodnocení
1	Kliknout na přidat kurz.	Aplikace přesměruje na stránku s formulářem pro přidávání nového kurzu.		
2	Validně vyplnit všechna povinná data formuláře a odeslat formulář.	Po odeslání formuláře dojde k přesměrování na stránku, na které se dají vytvářet studijní materiály. V horní části obrazovky se zobrazí informační hláška, že kurz byl úspěšně vytvořen.		

Tabulka F.1: Testovací případ: Vytvořit kurz - validní data

**F.1.2 Testovací případ 2 - nevalidní data**

Testovací případ: Vytvořit kurz - nevalidní data				
Krok	Popis	Očekávaný výsledek	Skutečný výsledek	Hodnocení
1	Kliknout na přidat kurz.	Aplikace zobrazí formulář pro přidávání nového kurzu.		
2	Při vyplňování formuláře nechat některá povinná pole nevyplněná nebo zadat nevalidní data a odeslat formulář.	Aplikace zobrazí uživateli varovnou zprávu, že kurz nemohl být vytvořen, protože některá data nejsou validní. Prvky formuláře, které neprošly validací se označí červeně a vypíše se vpravo od nich varovná hláška, která uživateli radí, jak problém odstranit.		

Tabulka F.2: Testovací případ: Vytvořit kurz - nevalidní data

## F.2 Scénář 2 - vytvořit studijní materiály

### F.2.1 Testovací případ 1 - vytvořit studijní materiály

Testovací případ: Vytvořit studijní materiály				
Krok	Popis	Očekávaný výsledek	Skutečný výsledek	Hodnocení
1	Kliknout na odkaz Moje kurzy.	Aplikace zobrazí stránku s výpisem kurzů, které uživatel vytvořil.		
2	Vybrat kterýkoliv svůj kurz a kliknout na něj.	Aplikace zobrazí stránku kurzu, na které lze vytvářet nové nebo editovat stávající materiály.		
3	Kliknout na odkaz přidat kapitolu.	Aplikace zobrazí stránku s formulářem pro přidávání kapitoly.		
4	Vyplnit formulář přidat kapitolu validními daty.	Aplikace přesměruje uživatele zpět na stránku editace studijních materiálů. V horní části stránky se zobrazí informativní hláška oznamující úspěšné přidání nové kapitoly.		
5	Kliknout na tlačítko přidat lekci náležící kapitole přidané v předchozím kroku.	Aplikace zobrazí stránku s formulářem pro přidání nové lekce. Stránka obsahuje menu s jednotlivými druhy obsahů, které se dají do lekce přidat.		
6	Přidat a vyplnit alespoň 3 různé druhy obsahů.	Aplikace vytvoří na stejné stránce formulářové prvky pro přidání nových obsahů lekce.		
7	Smazat jeden libovolný obsah lekce.	Aplikace zobrazí varovnou hlášku, jestli chce uživatel opravdu smazat část lekce. Pokud uživatel odklikne že ano, část formuláře zmizí.		
8	Odeslat formulář.	Aplikace přesměruje uživatele zpět na stránku vytváření a editace studijních materiálů. V horní části stránky se zobrazí informativní hláška oznamující úspěšné přidání nové lekce.		

Tabulka F.3: Testovací případ vytvořit studijní materiály

### F.3 Scénář 3 - publikovat kurz

#### F.3.1 Testovací případ 1 - připojený test

Testovací případ: Publikovat svůj kurz - s připojením testem				
Krok	Popis	Očekávaný výsledek	Skutečný výsledek	Hodnocení
1	Kliknout na odkaz přidat závěrečný test.	Aplikace zobrazí formulář s výběrem testů.		
2	Vybrat libovolný test a kliknout na něj.	Aplikace přesměruje uživatele zpět na stránku vytváření a editace studijních materiálů. V horní části stránky se zobrazí informativní hláška oznamující úspěšné přidání nové kapitoly.		
3	Kliknout na tlačítko publikovat kurz.	Ve vrchní části stránky se objeví hláška informující, že kurz byl přidán.		

Tabulka F.4: Testovací případ - publikovat svůj kurz s připojením kurzu

#### F.3.2 Testovací případ 2 - nepřipojený test

Testovací případ: Publikovat svůj kurz - bez připojení testu				
Krok	Popis	Očekávaný výsledek	Skutečný výsledek	Hodnocení
1	Kliknout na tlačítko publikovat kurz.	Aplikace zobrazí chybovou hlášku, že kurz nemůže být publikován bez připojení testu. Součástí stránky je tlačítko umožňující zahájit akci připojení testu.		

Tabulka F.5: Testovací případ - publikovat svůj kurz bez připojení kurzu

---

# Uživatelská příručka

Uživatelská příručka popisuje akce při průchodu uživatelským rozhraním aplikace. Uživatelská příručka bude rozdělena podle uživatelských rolí. První část se zabývá rolí student a je zaměřena na výběr kurzu, přihlášení do kurzu a studium kurzu. Druhá část popisuje vytváření a správu studijních materiálů.

## Student

### Výběr kurzu

1. Uživatel klikne v menu na odkaz kurzy.
2. Uživateli se zobrazí seznam kurzů, z kterého si může některý kurz vybrat.
3. Uživatel klikne na tlačítko náhled kurzu.
4. Uživateli se zobrazí produktová stránka kurzu, na které jsou všechny informace o kurzu.

### Přihlášení se do kurzu

1. Uživatel si zobrazí produktovou stranu kurzu (viz výběr kurzu).
2. Uživatel klikne na tlačítko přihlásit se do kurzu.
3. Uživatel je přesměrován na úvodní stránku kurzu s výpisem studijních materiálů.

### Studium kurzu

1. Uživatel klikne v menu na odkaz kurzy.
2. Uživateli se zobrazí seznam kurzů, do kterých je přihlášený a uživatel si může některý kurz vybrat.
3. Uživatel klikne na některý z kurzů.
4. Uživateli se zobrazí úvodní stránka kurzu, na které jsou všechny stu-

dijní materiály a informace o kurzu.

5. Uživatel klikne na některou lekci.

6. Uživateli se zobrazí lekce, pomocí tlačítek předchozí nebo další se uživatel může mezi lekceami pohybovat.

### **Mentor**

#### **Vytvoření kurzu**

1. Uživatel klikne v menu na odkaz publikovat a vybere publikovat kurz.

2. Uživatel vyplní formulář a odešle.

3. Uživatel je informován o výsledku odeslání formuláře.

#### **Správa studijních materiálů**

1. Uživatel klikne v menu na odkaz kurzy.

2. Uživatel vybere ze seznamu jeden z kurzů, který vytvořil a klikne na něj.

3. Uživateli se zobrazí stránka se studijními materiály kurzu, které může editovat nebo smazat.

4a. Pokud chce uživatel editovat stávající studijní materiály, klikne na vybraný materiál.

4b. Pokud chce uživatel smazat některý studijní materiál, klikne na tlačítko X umístěné v pravém horním rohu materiálu, který má být smazán.

4c. Pokud chce uživatel přidat lekci, vybere kapitolu, v které chce lekci přidat a v této kapitole klikne na tlačítko přidat lekci.

#### **Publikování kurzu**

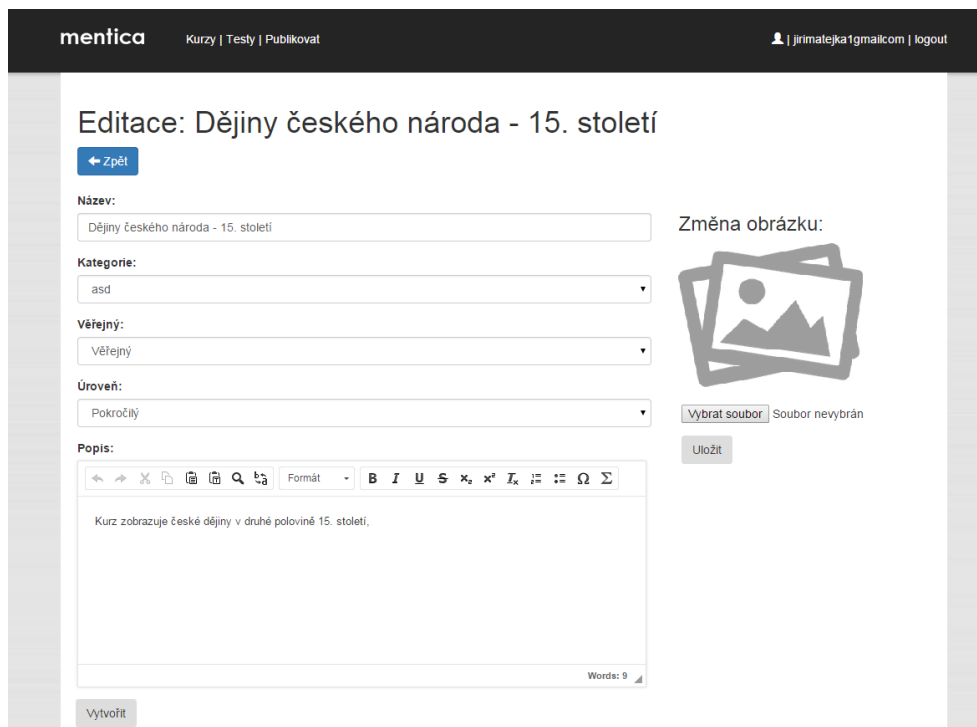
1. Uživatel klikne v menu na odkaz kurzy.

2. Uživatel vybere ze seznamu jeden z kurzů, který vytvořil a klikne na něj.

3. Uživatel vidí stránku správa studijních materiálů, v pravém horním rohu je tlačítko publikovat kurz, které uživatel může zmáčknout.

4. Pokud lze kurz publikovat, kurz se publikuje a uživateli se zobrazí informativní hláška, v opačném případě se uživateli zobrazí chybová hláška.

## Screenshots aplikace



Obrázek H.1: Editace kurzu

## H. SCREENSHOTY APLIKACE

The screenshot shows the 'Přidat lekci' (Add lesson) form in the Mentica application. The header includes the Mentica logo, navigation links for 'Kurzy | Testy | Publikovat', and a user profile with the email 'jirimalejka1gmail.com' and a 'logout' link. The main content area is titled 'Přidat lekci' and shows the course 'Dějiny českého národa - 15. století' and the chapter 'Úvodní kapitola'. A 'Menu' sidebar on the left lists content types: Nadpis, Text, Obrázek, Video, Citace, Tabulka, Rovnice, and Příloha. The main form fields include: 'Název lekce:' with a text input containing 'První lekce' and a 'Delete' button; a rich text editor with a toolbar and a 'Words: 0' counter; 'Obrázek:' with a 'Vybrat soubor' button, a selected image 'Obrázek c 1.JPG', and a 'Delete' button; 'Autor:' with a text input containing 'John Doe'; 'Viditelný název souboru:' with a text input containing 'historický obrázek'; 'Zdroj obrázku:' with an empty text input; 'Popis obrázku:' with an empty text input; and a 'Vytvorit' (Create) button at the bottom.

Obrázek H.2: Přidání nové lekce

The screenshot shows the 'Seznam kurzů' (Course list) page in the Mentica application. The header is identical to the previous screenshot. The main content area is titled 'Seznam kurzů' and lists three courses under the heading 'Kurzy:'. Each course card includes a thumbnail image, the course title, the provider 'Mentica team', a five-star rating, the type 'Typ: Kurz', and a 'Zdarma' (Free) label. The courses are: 1. 'Kontinent Austrálie' with a satellite map of Australia; 2. 'Jak napsat dobrý článek' with a paper airplane; 3. 'Vše okolo DNA' with a DNA double helix. Each card also features a 'Náhled' (Preview) button.

Obrázek H.3: Seznam kurzů



---

## Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kódy implementace
├─ init.....	inicializační soubory
├─ thesis .....	zdrojová forma práce ve formátu $\text{\LaTeX}$
text .....	text práce
├─ thesis.pdf .....	text práce ve formátu PDF