

Sem vložte zadání Vaší práce.



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



Bakalářská práce

## **Webový portál pro podporu výuky na středních školách**

*Lukáš Hamrla*

Vedoucí práce: Mgr. Michal Mikláš

11. května 2015



---

## Poděkování

Děkuji panu Mgr. Michalu Miklášovi za cenné rady při vedení této práce a za samotný nápad na její vytvoření. Děkuji všem, kteří mě po celou dobu tvorby podporovali.



---

# 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 11. května 2015

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2015 Lukáš Hamrla. 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**

Hamrla, Lukáš. *Webový portál pro podporu výuky na středních školách*. Bachelářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.



---

# Abstrakt

Tato bakalářská práce se věnuje analýze, návrhu a samotné implementaci webového portálu pro podporu výuky na středních školách. Cílem práce je zatraktivnit a zpříjemnit výuku studentům a vyučujícím tak, aby se studenti více zapojovali a byli více aktivnější a zjednodušit komunikaci mezi studenty a učitelem. První část se věnuje analýze požadavků, popisuje již existující webové portály pro podporu výuky. Na základě těchto analýz jsou v další části vybrány použité technologie. Závěrem obsahuje práce postupy při implementaci aplikace. Důraz je kladen především na rychlost používání aplikace a na anonymitu studentů při kladení dotazů. Z důvodu rozšiřitelnosti a snadné instalace pro co nejvíce škol je použit programovací jazyk PHP a databáze MySQL.

**Klíčová slova** webový portál, podpora výuky na středních školách, interaktivní výuka, Nette Framework, PHP, Foundation, AngularJS, SASS

---

# Abstract

This thesis describes analysis, design and implementation of web portal for high-school instruction support. The objective of the thesis is to attract and dulcify teaching for students and teachers so that students are going to more participate and show more activity. Application simplifies communication between students and teacher as well. The first part describes requirements analysis, analyse existing web portals for instruction support. The second part describes chosen technologies used in application based on the results of previous research. At the end, thesis contains procedures of implementation of the application. There is quick use of application emphasised and on student anonymity for questions asking too. Due to accessibility, PHP is used as a programming language and MySQL as a database engine.

**Keywords** web portal, high-school instruction support, interactive teaching, Nette Framework, PHP, Foundation, AngularJS, SASS

---

# Obsah

Úvod	1
<b>1 Analýza</b>	<b>3</b>
1.1 Definice pojmů . . . . .	3
1.2 Současná řešení . . . . .	3
1.3 Analýza procesů . . . . .	6
1.4 Analýza funkčních a nefunkčních požadavků . . . . .	10
1.5 Business doménový model . . . . .	11
<b>2 Návrh</b>	<b>13</b>
2.1 Relační databázový model . . . . .	13
2.2 Možnosti řešení a použité technologie . . . . .	13
2.3 Návrh webdesignu . . . . .	21
<b>3 Realizace</b>	<b>23</b>
3.1 Podpůrný software . . . . .	23
3.2 Struktura projektu . . . . .	23
3.3 Přihlášení do aplikace . . . . .	26
3.4 Evidence studentů a tříd . . . . .	27
3.5 Evidence úkolů a otázek . . . . .	29
3.6 Veřejná diskuze pro účastníky vyučování . . . . .	31
3.7 Chat mezi studenty a učiteli . . . . .	32
<b>Závěr</b>	<b>35</b>
<b>Použité zdroje</b>	<b>37</b>
<b>A Seznam použitých zkratk</b>	<b>41</b>
<b>B Instalační příručka</b>	<b>43</b>



---

## Seznam obrázků

1.1	Počet uživatelů jednotlivých LMS (zdroj: [2]) . . . . .	4
1.2	Ukázka prostředí aplikace Moodle a její využití na Fakultě informačních technologií ČVUT. . . . .	5
1.3	Ukázka prostředí webové služby Edmodo. . . . .	6
1.4	Příprava na školní rok . . . . .	8
1.5	Příprava, zadávání, odevzdávání a hodnocení otázek a úkolů . . . . .	9
1.6	Uzavření školního roku . . . . .	10
1.7	Business doménový model . . . . .	12
2.1	Relační databázový model . . . . .	14
2.2	Statistika využití jednotlivých verzí PHP 5 (zdroj: [8]) . . . . .	14
2.3	Architektura MVC (zdroj: [11]) . . . . .	16
2.4	Výsledný webdesign včetně ukázky responzivního zobrazení . . . . .	21
3.1	Přehled odpovědí včetně vizualizace do grafu . . . . .	30
3.2	Ukázka nekonečného rolování (zdroj: [34]) . . . . .	32



---

# Seznam tabulek

3.1	Tvar exportovaného souboru z informačního systému Bakaláři . . .	28
-----	--	----





---

## Seznam zdrojových kódů

2.1	Dependency injection pomocí anotací . . . . .	18
3.1	Použití autowiringu . . . . .	26
3.2	Snippets v šabloně . . . . .	27
3.3	Práce se snippets v PHP skriptu . . . . .	27
3.4	Dynamické kontejnery u formulářů . . . . .	29
3.5	Použití třídy Multiplier . . . . .	31



---

# Úvod

V dnešní době se studenti čím dál méně aktivně zapojují do vyučování. Snaží se dobu vyučování nějak „přetrpět“ a poté usednout zpět k počítačům, tabletům a mobilům, hlavně aby je učitel neobtěžoval svým výkladem.

Cílem této bakalářské práce je proto vytvořit webový portál pro podporu výuky na středních školách, který bude stimulovat studenty k větší aktivitě při vyučování a bude velmi jednoduchý na používání. Velký důraz je kladen také na rychlost a pohodlnost obsluhy aplikace napříč všemi zařízeními jako jsou počítače, tablety a mobily. Aplikace implementovaná v této práci bude použita v reálném vyučování na Gymnáziu a Jazykové škole s právem státní jazykové zkoušky ve Zlíně.

Aplikace zjednodušuje komunikaci mezi studenty a učitelem, je v ní možno zadávat úkoly a otázky, které budou studenti řešit. Zároveň umožňuje vizualizaci odpovědí do grafů a je možná také anonymní ohodnocení vypracovaných úkolů mezi samotnými studenty. Dále také nabízí možnost konverzace pomocí chatu mezi studenty a učiteli.

Přesný popis požadavků se nachází v kapitole Analýza, stejně tak analýza procesů a doménový model. Zároveň jsou zde porovnány již existující řešení pro podporu výuky.

V kapitole Návrh je popsán relační model databáze, rozebrány možnosti řešení, popsány zvolené technologie a je v ní obsažen také návrh webdesignu.

Kapitola Realizace se zabývá samotnou implementací webového portálu, obsahuje ukázky kódu a funkčnosti aplikace.



---

# Analýza

## 1.1 Definice pojmů

Pro začátek je potřeba definovat některé pojmy vyskytující se dále v textu:

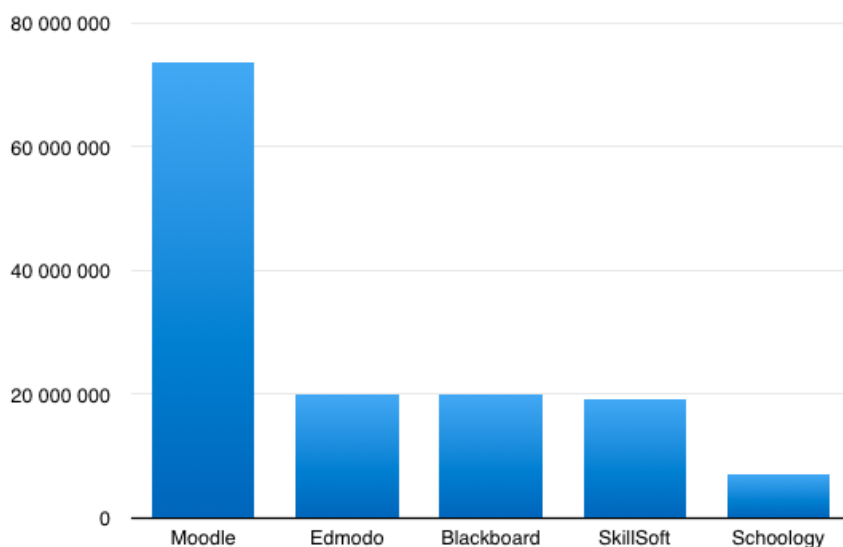
- *Třída* – je logický celek sdružující v sobě několik studentů. Každý student v jednom školním roce může být součástí pouze jedné třídy.
- *Vyučovací skupina* – je logický celek sdružující v sobě několik studentů. V jednom školním roce může student navštěvovat více vyučovacích skupin, v praxi jsou tím na mysli např. skupiny pro výuku jazyků, seminářů, aj.
- *Vyučování* – je logický prvek, který definuje vyučování ve škole. Nese v sobě informaci o vyučovaném předmětu, *třídě* nebo *vyučovací skupině*, stejně tak informaci o době vyučování a učitelích. Jako příklad poslouží vyučování informatiky třídy 4.B ve středu od 14:00 do 14:45.
- *Vyučovací hodina* (nebo též pouze *hodina*) – je logický prvek, který definuje konkrétní vyučovací hodinu v daný den a daný čas a je součástí jednoho vyučování. V analogii k předchozímu příkladu – vyučovací hodina informatiky 22. 4. 2015 od 14:00 do 14:45 třídy 4.B s názvem „Základy programování v PHP“.

## 1.2 Současná řešení

V této kapitole jsou rozebrána současná řešení webových portálů pro podporu výuky (zkráceně také LMS<sup>1</sup>).

---

<sup>1</sup>Learning Management System – systém pro řízení výuky v rámci e-learningu, organizace a poskytování vzdělávacích programů nebo kurzů [1]



Obrázek 1.1: Počet uživatelů jednotlivých LMS dle [2]

### 1.2.1 Moodle

Z grafu 1.1 lze vyčíst, že suverénně nejpoužívanějším LMS dle počtu uživatelů je portál Moodle<sup>2</sup>, který je velmi populární i v České republice. Tento open-source<sup>3</sup> software je „webová výuková platforma navržená pro vyučující, administrátory a studenty tak, aby s jedním robustním, bezpečným a integrovaným řešením mohli vytvářet personalizované výukové prostředí“ [3]. Tento systém umožňuje vytvářet kurzy, úlohy, přidávat materiály a mnoho dalšího. Jeho velká výhoda spočívá právě v jeho otevřenosti, což umožňuje nadšencům vytvářet pluginy pro rozšíření funkcionality. Tyto doplňky dovolují přidat např. možnost vytvářet a spravovat ankety.

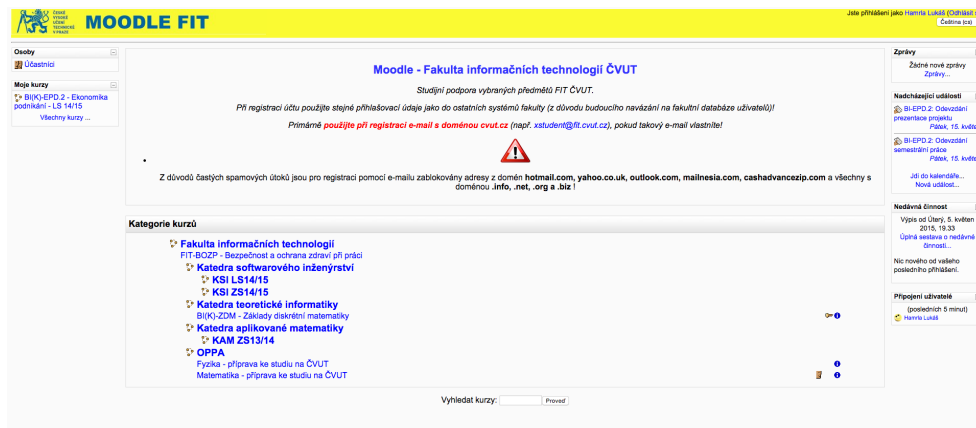
Komunikace mezi uživateli je zde možná pomocí pluginu pro chat, který ovšem umí vytvářet jen místnosti pro více lidí nebo je také dostupná možnost soukromých zpráv. Součástí jednotlivých kurzů je i diskuzní fórum.

Nevýhodou systému Moodle je pomalá rychlost s přibývajícím objemem dat a vysoké hardwarové nároky na server. Ve starších verzích, které jsou stále využívány, je navíc poměrně uživatelsky nepřívětivé prostředí. Zároveň se také nezaměřuje přímo na použití při hodině, ale spíše slouží jako doplněk pro domácí studium a také k využití pro všechna možná školicí centra, kurzy, aj.

---

<sup>2</sup><https://moodle.org>

<sup>3</sup>volně šířitelný



Obrázek 1.2: Ukázka prostředí aplikace Moodle a její využití na Fakultě informačních technologií ČVUT.

### 1.2.2 Edmodo

Dalším velmi populárním nástrojem je Edmodo<sup>4</sup>. Narozdíl od systému Moodle není open-source softwarem, nýbrž webovou službou, do které se mohou přihlásit učitelé, studenti a dokonce i rodiče. Obsahuje v sobě také mnohé prvky sociální sítě (přidávání fotek, jejich komentování, aj.). Jsou zde k dispozici i různé aplikace a výukové hry. Svým vzhledem, k prohlédnutí na obrázku 1.3, připomíná sociální síť Facebook<sup>5</sup>.

Největší výhodou je bezesporu právě skutečnost, že je webovou službou. To znamená, že není nutné, aby škola měla svůj vlastní server, dokonce nemusí mít žádného administrátora, který by musel nastavovat přístupy do databáze, aj. Všichni se totiž přihlašují a pracují přímo na WWW<sup>6</sup> stránkách <https://www.edmodo.com>. Na druhou stranu mohou školy mít problém s tím, že data o studentech jsou na cizím serveru. Edmodo, stejně jako Moodle, má i svou vlastní mobilní aplikaci dostupnou pro telefony a tablety s operačním systémem iOS a Android.

Nevýhodou je nedostupnost české lokalizace, což je nejspíše hlavní důvod, proč není v České republice tato služba tak rozšířená, jako již zmiňovaný Moodle.

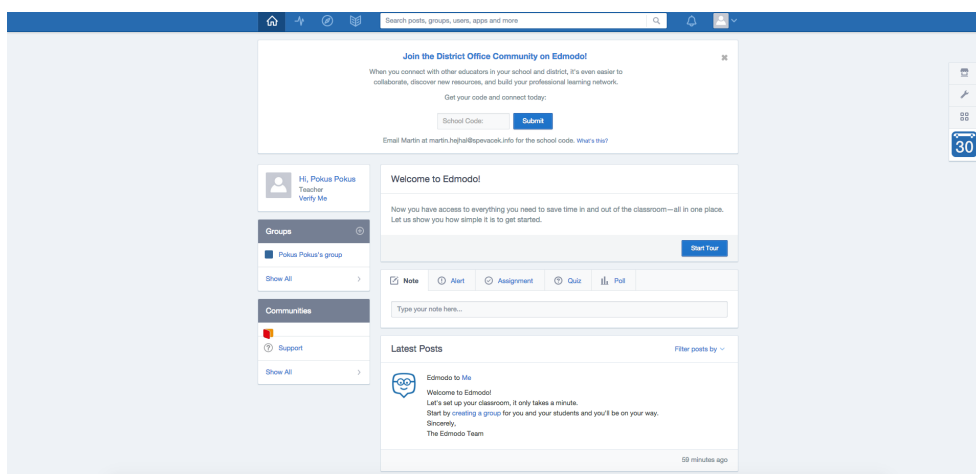
Tento systém se zaměřuje primárně na školy a na komunikaci mezi studenty, učiteli a rodiči.

<sup>4</sup><https://www.edmodo.com>

<sup>5</sup><http://www.facebook.com>

<sup>6</sup>World Wide Web

## 1. ANALÝZA



Obrázek 1.3: Ukázka prostředí webové služby Edmodo.

### 1.2.3 Blackboard

Blackboard<sup>7</sup> je webová aplikace napsaná v Javě. Nezaměřuje se jen na samotnou výuku, ale má také jednotlivé moduly pro analýzu dat, pro komunikaci, atd.

Tato aplikace je, narozdíl ode dvou předchozích systémů, placená a také nemá českou lokalizaci.

Velkou nevýhodou je také poměrně nepřehledná dokumentace a velmi složitá instalace, což dokazuje instalační příručka, která má přes 100 stránek [4], takže pro školy bez odborníka je tato aplikace nezprovoznitelná.

### 1.2.4 Závěr

Z výše uvedených informací je zřejmé, že jedinou možností pro české školy je využití systému Moodle, který je navíc kompletně zdarma. Větší rozmach zajímavě vyhlížející služby Edmodo sráží nepřítomnost lokalizace pro ostatní jazyky – aplikace je přeložena pouze do nejrozšířenějších světových jazyků [5].

## 1.3 Analýza procesů

Při používání aplikace probíhají 3 na sebe navazující procesy. Před školním rokem probíhá proces příprava na školní rok, v průběhu školního roku a zároveň nejčastěji opakovaným procesem je příprava, zadávání, odevzdávání a hodnocení otázek a úkolů a proces, který školní rok uzavírá, čímž je uzávěrka školního roku. Většiny procesů se účastní zejména vyučující, popř. adminis-

<sup>7</sup><http://www.blackboard.com>



trátor, studenti se zapojují pouze při odpovídání na otázky a odevzdávání úkolů v průběhu školního roku.

Jelikož tato aplikace byla vytvářena jako kompletně nový systém, analýza procesů je typu TO BE<sup>8</sup>.

### 1.3.1 Příprava na školní rok

Před začátkem školního roku si administrátor vyexportuje všechny *třídy* a *vyučovací skupiny* se studenty, které budou používat aplikaci. Exportovaný soubor může administrátor získat z informačního systému Bakaláři. Tento soubor ve formátu XLS nebo XLSX poté naimportuje do webové aplikace, která automaticky vytvoří nové profily studentům nebo již existující profily studentů, kteří aplikaci dříve používali, přiřadí do požadované *třídy* či *vyučovací skupiny*. Pokud škola tento informační systém nevyužívá, je možné všechny studenty, *třídy* a *vyučovací skupiny* vytvořit ručně. K jednotlivým *třídám* a *vyučovacími skupinám* administrátor vytvoří všechna *vyučování*. *Vyučování* se skládá z předmětu, učitele a doby výuky. Dle zadané doby výuky se automaticky vytvoří konkrétní *hodiny* v průběhu celého školního roku. Administrátor poté vše zkontroluje a po případné úpravě tok procesu končí.

### 1.3.2 Příprava, zadávání, odevzdávání a hodnocení otázek a úkolů

Aplikace se bude věnovat hlavně podpoře při samotném *vyučování*, a proto je tento proces analyzován nejdetailněji. Diagram aktivit pro tento proces je k nahlédnutí na obrázku 1.5. Tento proces končí s koncem *vyučovací hodiny*.

#### 1.3.2.1 Příprava hodiny

Vyučující si nejdříve připraví *vyučovací hodinu*. Tato *hodina* buď již automaticky vznikla při vytváření *vyučování* nebo ji učitel vytvoří manuálně. K *hodině* může přidat její název a popis. Do popisku může přidávat také obrázky a popřípadě jiné materiály potřebné pro *hodinu*. Zároveň ji také naplní otázkami a úkoly, které budou v *hodině* použity.

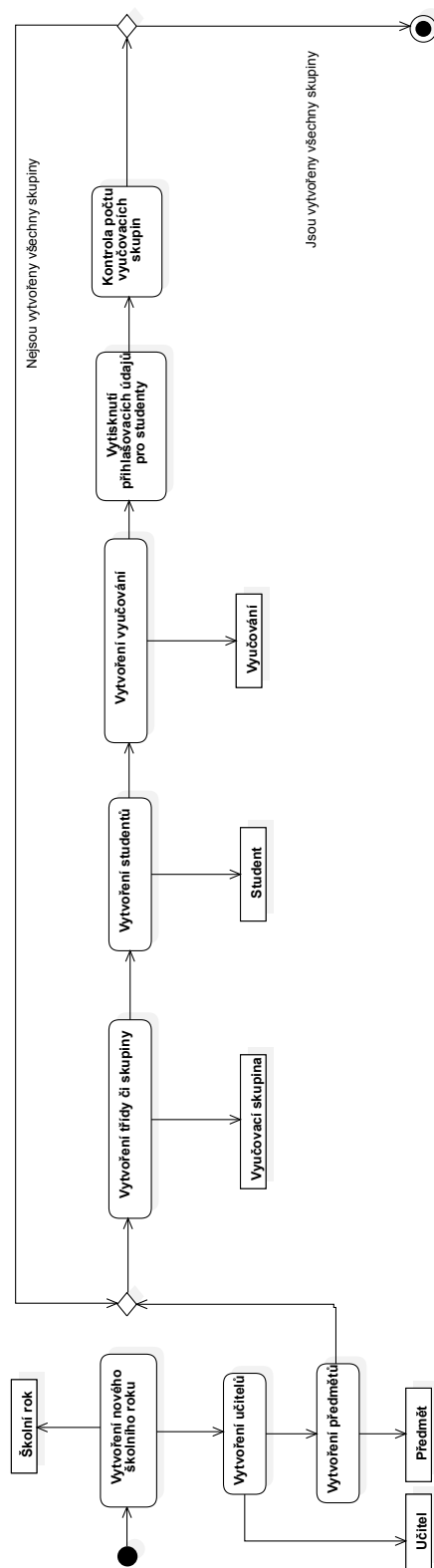
#### 1.3.2.2 Zadání a odpovězení na otázku

Učitel při *hodině* může zadat otázku z již předpřipravené sady otázek nebo může vytvořit otázku úplně novou. Při vytvoření zadá její text, nastaví uzavřenost či otevřenost otázky, přidá případné možnosti a označí správné odpovědi. Při uzavřené otázce také může požadovat po studentech textové zdůvodnění vybrané odpovědi. Studenti poté na zadanou otázku odpovídají a v případě

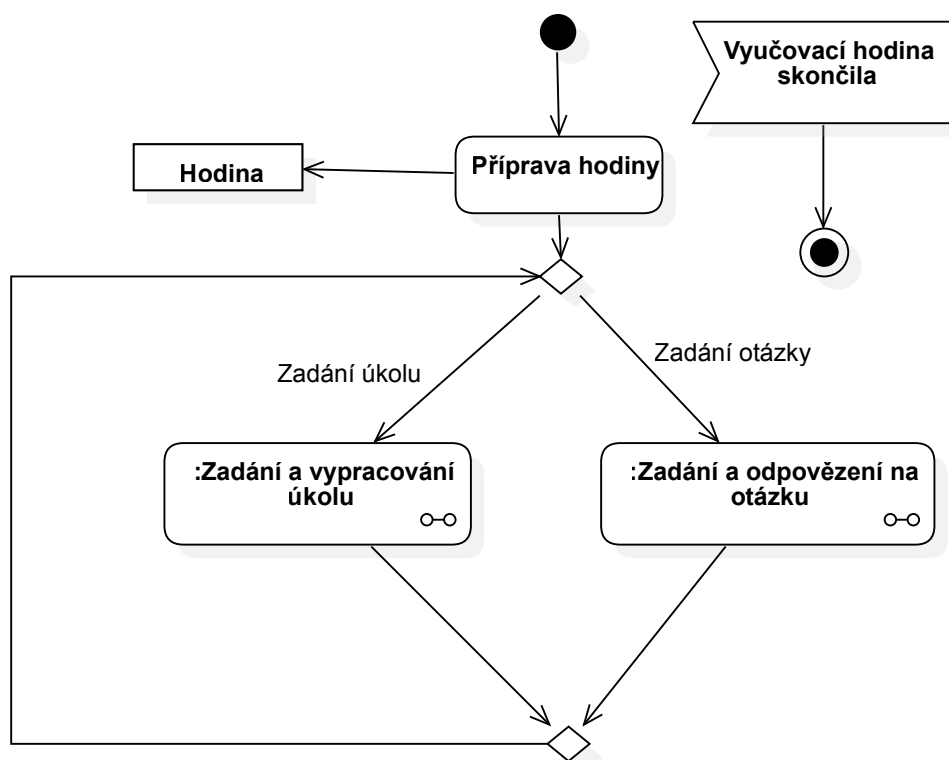
---

<sup>8</sup>budoucí stav

# 1. ANALÝZA



Obrázek 1.4: Diagram aktivit procesu příprava na školní rok



Obrázek 1.5: Diagram aktivit procesu příprava, zadávání, odevzdávání a hodnocení otázek a úkolů

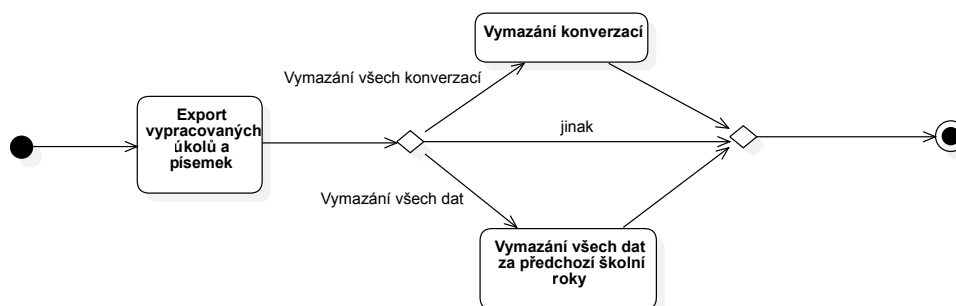
uzavřené otázky se ihned dovídají, jestli odpověděli správně. Učitel vidí průběžné výsledky odpovědí studentů, rovněž tak jejich hodnocení, které systém vypočítá na základě správně označených odpovědí.

### 1.3.2.3 Zadání a odevzdání úkolu

Učitel při *hodině* může zadat úkol z již předpřipravené sady úkolů nebo lze vytvořit úkol úplně nový. Při vytvoření popíše zadání úkolu, nastaví typ limitu, který může být striktní, volný nebo také žádný a zvolí, zda-li chce, aby si studenti mohli své odevzdané úkoly ohodnotit mezi sebou. Poté vidí průběžně odevzdané úkoly, které může hodnotit a nakonec vyrobí prezentaci z nejlepších prací.

### 1.3.3 Uzavření školního roku

Z důvodu často nedostatečné kapacity pevných disků na webových serverech škol a požadavku archivace odevzdaných úkolů, má administrátor po skončení školního roku tyto možnosti:



Obrázek 1.6: Diagram aktivit procesu uzavření školního roku

- export všech vypracovaných úkolů,
- vymazání všech konverzací v chatu,
- vymazání veškerých dat.

V případě dostatečného prostoru na serveru nemusí využít administrátor ani jednu z možností, pouze označí školní rok jako uzavřený.

## 1.4 Analýza funkčních a nefunkčních požadavků

V této kapitole jsou sepsány funkční a nefunkční požadavky, které vychází z obecné dostupnosti technologií a dohody se zadavatelem.

### 1.4.1 Funkční požadavky

- F1. Evidence studentů, tříd a vyučovacích skupin
  - systém bude evidovat studenty a jejich zařazení do tříd a vyučovacích skupin, u jednotlivých studentů bude evidováno hodnocení. Bude také možné provést import z informačního systému Bakaláři. Systém bude umět zobrazovat studenty v konkrétní hodině a vyučující jim bude moci přidělovat body za aktivitu.
- F2. Evidence úkolů a otázek
  - systém bude umožňovat co nejrychlejší vytváření úkolů a otázek. Studenti budou moci odpovídat na otázky a odevzdávat úkoly. Bude také možné nechat studenty anonymně ohodnotit navzájem své odevzdané úkoly. Systém bude schopný vyrobit prezentaci nejlepších prací dle hodnocení. Zároveň také bude umět spárovat stejné otázky a úkoly napříč hodinami pro možnou budoucí demonstraci nejlepších odevzdaných úkolů a porovnání výsledků z jiných tříd s aktuálními.

- F3. Veřejná diskuze pro účastníky *vyučování*
  - u jednotlivých *vyučování* a *hodin* bude možná veřejná diskuze. Na jednotlivé příspěvky budou moci členové *vyučování* reagovat a komentovat je. Uživatel rozhodne, zda-li jeho příspěvek bude anonymní či podepsaný.
- F4. Chat mezi studenty a učiteli
  - systém bude umožňovat komunikaci prostřednictvím chatu mezi studenty a učiteli. Vyučující bude moci chat kdykoliv zneaktivnit. Komunikace bude povolena pouze mezi studenty a učiteli ze stejných *tříd* a *vyučovacích skupin*.

#### 1.4.2 Nefunkční požadavky

- N1. Dostupnost přes webové rozhraní
  - systém bude dostupný přes webové rozhraní z důvodu přenositelnosti a snadného používání na všech platformách. Aplikace nemusí být optimalizována pro prohlížeče Internet Explorer, z důvodu zbytečných omezení tohoto programu vzhledem k dostupnosti jiných pokročilejších technologií zdarma, jako jsou např. Google Chrome, Mozilla Firefox, aj., které podporují mnohem více CSS<sup>9</sup> vlastností a HTML<sup>10</sup> elementů.
- N2. Responzivní design
  - design aplikace bude navržen v responzivním layoutu, který bude reagovat na různá rozlišení displeje vhodným přeuspořádáním bloků tak, aby bylo jednoduché aplikaci používat i na mobilních zařízeních.

### 1.5 Business doménový model

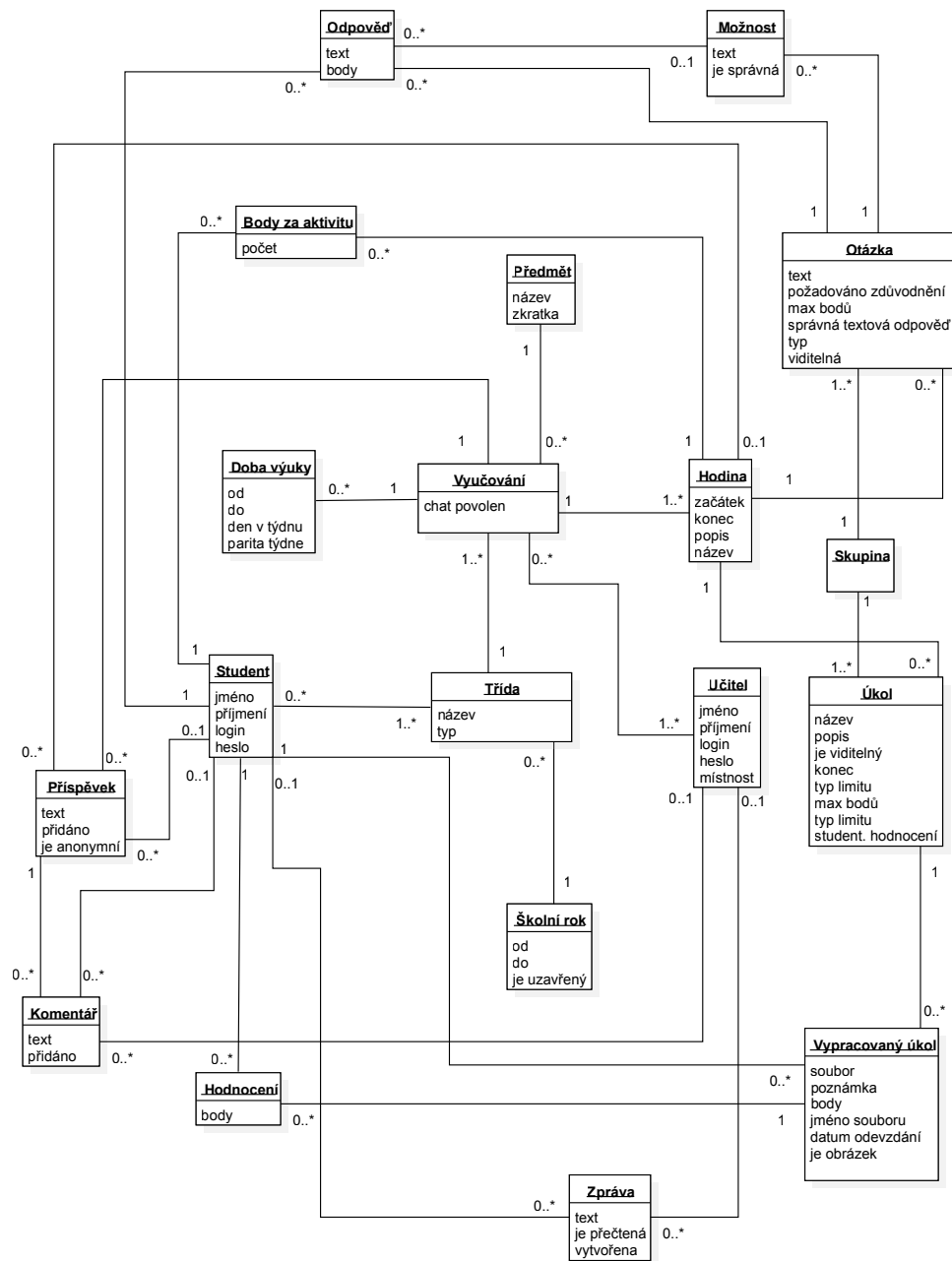
V business doménovém modelu jsou zaneseny veškeré entity, které mohou při práci v aplikaci vzniknout. Doménový model vychází z analýzy funkčních požadavků popsanych v předchozí sekci. Tento model je pak východiskem pro databázový model, který je popsán ve vlastní kapitole 2.1. Entita *skupina* je v tomto modelu pouze prostředkem pro spojení stejných otázek a úkolů zadaných v různých hodinách.

---

<sup>9</sup>Cascading Style Sheets

<sup>10</sup>HyperText Markup Language

# 1. ANALÝZA



Obrázek 1.7: Business doménový model

---

# Návrh

## 2.1 Relační databázový model

Relační model popisuje strukturu databáze v MySQL. Vychází z doménového modelu, oproti němu obsahuje navíc identifikátory jednotlivých záznamů v tabulkách, dekomponuje vazby M:N a obsahuje cizí klíče. Na obrázku 2.1 je pouze část databázového modelu, na kterém je vyobrazen způsob řešení přiřazování učitelů a studentů do jednotlivých tříd a hodnocení prací mezi studenty. Zároveň jsou zde entity učitel a student z doménového modelu spojeny do jedné entity `users` z důvodu jednodušší autentizace uživatelů. Učitel od studenta jsou na úrovni zdrojového kódu rozeznáváni dědičností ze společného předka z entity `users`. Celý databázový model je k dispozici na příloženém CD. V návrhu databáze je velmi důležité integritní omezení, a to, že jeden student může být v jednom školním roce součástí pouze jedné entity `class`, která je typu *třída*.

## 2.2 Možnosti řešení a použité technologie

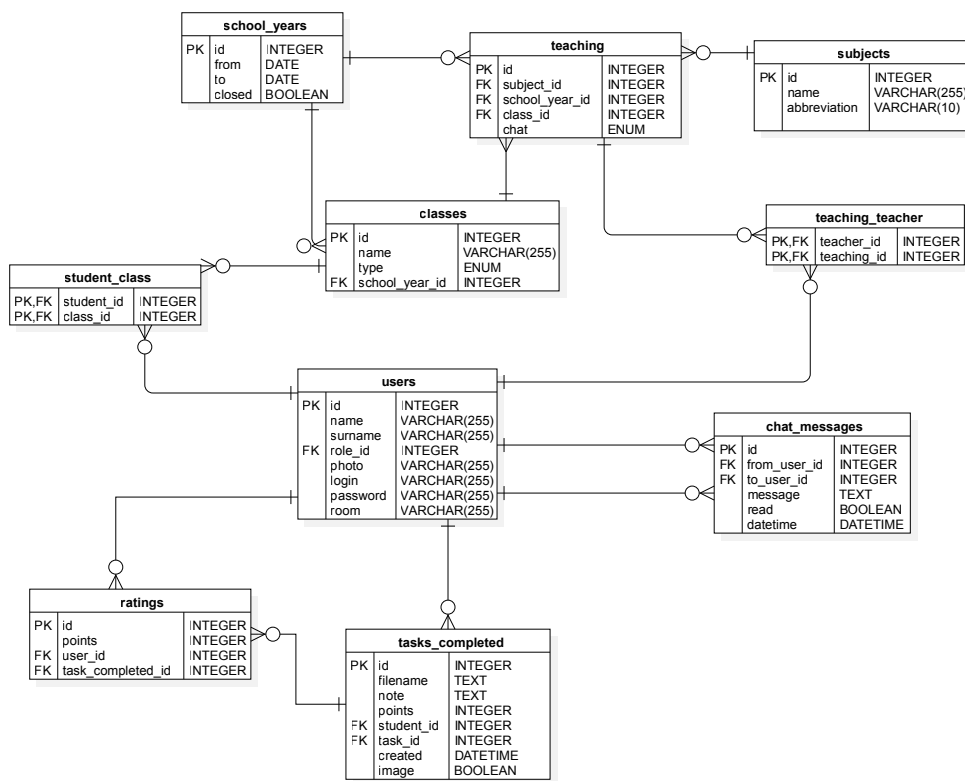
Na základě analýzy funkčních a nefunkčních požadavků bylo potřeba vybrat vhodné technologie pro implementaci. Na následujících řádcích jsou rozebrány možné alternativy a popsány zvolené technologie.

Jelikož hlavní požadavek byla dostupnost aplikace na webové platformě, jako programovací jazyk byl vybrán PHP<sup>11</sup>, a to hlavně z důvodu jeho rozšiřitelnosti a nízkých nároků na server. Jelikož střední školy nerady zasahují do toho, co funguje, je nutné se zaměřit i na kompatibilitu se staršími verzemi PHP. Jak je viditelné z grafu 2.2, téměř polovina webových stránek používá verzi 5.3, jejíž první vydání se datuje do roku 2010 [7] a celkem 82 % využívá verzi  $\geq 5.3$ . Proto je tato aplikace kompatibilní se všemi verzemi PHP od verze 5.3.

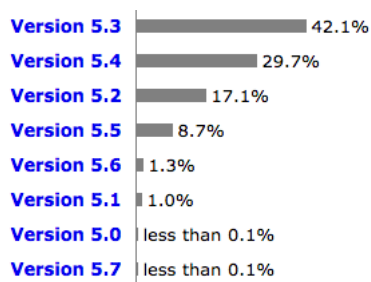
---

<sup>11</sup>Hypertext Preprocessor

## 2. NÁVRH



Obrázek 2.1: Relační databázový model



Obrázek 2.2: Statistika využití jednotlivých verzí PHP 5 [8]



### 2.2.1 PHP frameworky

Nad jazykem PHP je postaveno mnoho populárních a moderních frameworků, většina z nich ctí návrhový vzor MVC<sup>12</sup>.

#### 2.2.1.1 Symfony

Symfony<sup>13</sup> je framework vyvinutý převážně franouzskými vývojáři, který je velmi rozsáhlý, robustní a vhodný pro velké projekty. Skládá se z jednotlivých balíčků (tzv. bundles), které se mohou do projektu jednoduše doinstalovat a je jich dostupné nepřeborné množství. Nejedná se úplně tak o framework v pravém slova smyslu, ale jako spíš o knihovnu seskupující několik bundlů dohromady.

#### 2.2.1.2 CakePHP

CakePHP<sup>14</sup> je naopak vhodný spíše na menší projekty a to hlavně z důvodu jeho malé konfigurovatelnosti. Klade si za cíl co nejvíce zjednodušit a zrychlit programátorovi práci bez zbytečného nastavování. Jako jeden z mála větších frameworků má v sobě stále podporu pro PHP 5.2 [9] (platí pouze pro verzi 2, CakePHP je současně vyvíjen a podporován ve verzích 2 a 3).

#### 2.2.1.3 Nette

Pro tuto práci byl vybrán framework Nette<sup>15</sup>, který má velkou popularitu hlavně v České republice a na Slovensku, jak je vidno z průzkumu [10], v němž se umístil na 3. místě mezi nejpoblárnějšími frameworky, kde z celkového počtu 864 hlasů českých a slovenských vývojářů hlasovalo právě 686 pro Nette.

Jeho velká výhoda spočívá právě v aktivní české komunitě, rychlosti a jednoduchosti, která ovšem není na úkor nemožnosti konfigurovatelnosti frameworku. Stejně jako Symfony se Nette v nových verzích odchýlilo od myšlenky jednoho frameworku a tíhne spíše ke spojení samostatných komponent, které lze používat také samostatně bez nutnosti použití Nette. Framework má celou řadu vlastností a nástrojů, které jsou popsány dále.

**Architektura Model-View-Controller** Tento návrhový vzor vzniknul z nutnosti oddělit aplikační logiku od prezentační a z důvodu vymícení tzv. "špagetového kódu"<sup>16</sup>. Pokud systém správně dodržuje zásady této architektury, na stejném projektu může zároveň pracovat HTML kodér i backendový PHP vývojář zároveň bez toho, aniž by pracovali na stejných souborech.

---

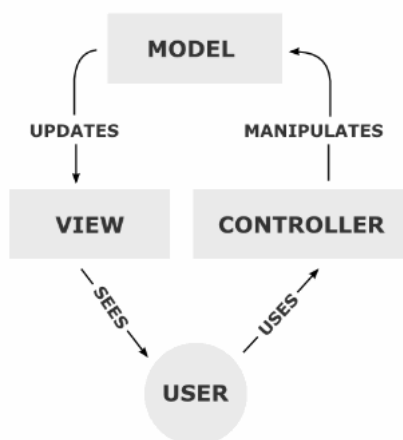
<sup>12</sup>Model-View-Controller

<sup>13</sup><http://symfony.com>

<sup>14</sup><http://cakephp.org>

<sup>15</sup><http://nette.org>

<sup>16</sup>Zdrojový kód obsahující PHP i HTML kód.



Obrázek 2.3: Architektura MVC [11]

Jak už název napovídá, tento návrhový vzor se skládá ze tří částí: **modelu**, **controllerů**, a **view**.

**Model** Model je soubor obsahující datový a funkční základ celé aplikace. Poskytuje rozhraní umožňující číst a zapisovat data. Tato část aplikace nemá ponětí o ostatních segmentech (View, Controller), jeho jediný úkol je zpracování a persistence dat.

Nette poskytuje rozhraní pro práci s databází, `Nette\Database`, které ovšem neumožňuje pracovat s jednotlivými entitami jako s objekty – je to pouze nástroj pro přečtení a nahrání dat do databáze. Proto je v této práci použita knihovna Doctrine<sup>17</sup>, která umožňuje namapovat jednotlivé třídy na databázové tabulky a poté využívat předností objektového návrhu. V Nette není tato knihovna přímo k dispozici, je proto nainstalováno rozšíření, které tuto knihovnu implementuje do aplikace, s názvem `Kdyby\Doctrine` [12].

**View** View (neboli pohled) je nejčastěji přístupný HTML kódérům. Je to ta část aplikace, která se vykreslí jako poslední a která se zobrazí přímo uživateli. Mnoho frameworků používá tzv. šablonovací systémy, které umožňují psát čistý kód bez nutnosti znalosti PHP. Nette má pro tyto účely svůj vlastní šablonovací systém, `Latte`<sup>18</sup>, který mimo jiné poskytuje i celou řadu filtrů a maker. Zároveň také myslí na bezpečnost, protože ve výchozím stavu escapuje<sup>19</sup> výpis jednotlivých proměnných, čímž eliminuje riziko XSS<sup>20</sup>.

<sup>17</sup><http://www.doctrine-project.org>

<sup>18</sup><http://doc.nette.org/cs/2.3/templating>

<sup>19</sup>Escapování je převod znaků majících v daném kontextu speciální význam na jiné odpovídající sekvence. [13]

<sup>20</sup>Narušení kontextu webových stránek podstrčením JavaScriptového kódu od útočníka.

**Controller** V Nette se tyto třídy nazývají presentery, což je názvosloví používané v příbuzném návrhovém vzoru, Model-View-Presenter. Presentery jsou jakýmsi spojovacím článkem mezi dvěma výše zmíněnými částmi. Dle návrhového vzoru MVC je důležité, aby model a view o sobě navzájem vůbec nevěděli, veškerou komunikaci zajišťuje presenter. Ten na základě akce uživatele volá patřičnou aplikační logiku.

**Dependency injection** Dependency injection (česky předávání závislostí) je technika, kdy si každá třída v kódu vyžaduje své závislosti. Např. servisní třída pro vytvoření objednávky vyžaduje přístup k databázi, aby mohla tuto objednávku v databázi uložit. Proto ve svém konstruktoru uvede tuto závislost. Při vytváření této třídy, např. v presenteru, je potřeba jí tuto závislost předat. Presenter má tedy také u svého konstruktoru uvedeno, že vyžaduje přístup k databázi. Samozřejmě někde v nejvyšším rodiči je potřeba tuto závislost vytvořit a předat. Nette pro tyto potřeby poskytuje tzv. dependency injection kontejner<sup>21</sup>. U jakékoliv třídy si stačí vyžádat své závislosti a tento kontejner se postará o samotné předání. Aby bylo možné závislost předat, je potřeba ji v konfiguračním souboru zaregistrovat jako službu.

Framework poskytuje čtyři metody předávání závislostí:

- **předání konstruktorem (tzv. constructor injection)** – metoda, která byla nastíněna výše – každá třída má ve svém konstruktoru uvedeny své závislosti, které se jí při vytváření instance předají,
- **předání setterem (tzv. setter injection)** – při této metodě se požadovaná závislost přidá jako vlastnost objektu a pro každou tuto závislost se vytvoří vlastní setter,
- **předání metodou inject\*** – opět se závislost přidá jako členská proměnná objektu a pro každou z nich je potřeba vytvořit metodu, jejíž název je ve tvaru `injectNázevSluzby`,
- **předání anotací @inject\*** – tato metoda je dostupná pouze pro presentery a spočívá v přidání služby jako veřejné členské proměnné a u ní označení anotací `@inject`, jak je ukázáno v kódu 2.1.

**Bezpečnost** Framework poskytuje širokou škálu nástrojů pro zvýšení zabezpečení a eliminování rizik. Některé z nich jsou zapnuty automaticky, jiné se musí povolit manuálně.

---

<sup>21</sup><http://doc.nette.org/cs/2.3/dependency-injection#toc-di-kontejner-a-sluzby>

```
class MyPresenter
{
    /**
     * @var MyService
     * @inject
     */
    public $myService;
}
```

---

Zdrojový kód 2.1: Dependency injection pomocí anotací

**SQL Injection** Tato technika spočívá v podstrčení škodlivého SQL<sup>22</sup> dotazu útočníkem do míst, kde je pravděpodobné, že aplikace bude provádět dotaz do databáze. Proto se vkládají např. do formulářových polí nebo do parametrů URL<sup>23</sup>. Pro dostatečnou eliminaci tohoto rizika je proto potřeba veškeré vstupy od uživatele správně odfiltrovat. Nette toto provádí automaticky, jak ve své vlastní `Nette\Database`, tak v samostatné knihovně `Kdyby\Doctrine`.

**Cross-Site Scripting (XSS)** O této technice je již zmíněno v kapitole 2.2.1.3 o šablonách. Nette toto riziko velmi dobře eliminuje<sup>24</sup> ve svém šablonovacím systému Latte.

**Cross-Site Request Forgery (CSRF)** „Cross-Site Request Forgery je útok spočívající v tom, že přimějeme uživatele navštívit stránku, která skrytě vykoná útok na webovou aplikaci, kde je uživatel zrovna přihlášen“ [14]. Většinou se útok nezaměřuje na prolomení přístupu, ale spíše zneužívá povolené akce právě přihlášeného uživatele. V Nette zabezpečení proti této hrozbě není zapnuto automaticky, je potřeba u každého formuláře zvlášť tuto ochranu zapnout.

### 2.2.2 Responzivní design

V posledních pár letech zažívají mobilní zařízení nebývalý rozmach. Protože mají oproti počítačům malé displeje, je potřeba těmto menším displejům přizpůsobit obsah na webu. Toto je docíleno pomocí Media Queries<sup>25</sup>, které „nám umožňují změnit prezentaci stránky na základě velikosti obrazovky, kterou používá návštěvník webu“ [18]. Pro implementaci responzivního designu existuje několik CSS frameworků, které využívají právě tuto vlastnost CSS3.

---

<sup>22</sup>Structured Query Language

<sup>23</sup>Uniform Resource Locator

<sup>24</sup><http://doc.nette.org/cs/2.3/templating#toc-context-aware-escaping>

<sup>25</sup><http://www.w3.org/TR/css3-mediaqueries>

### 2.2.2.1 Bootstrap

Bootstrap<sup>26</sup> (někdy také uváděno jako Twitter Bootstrap), je komplexní a moderní CSS framework vytvořený firmou Twitter vydávaný pod licencí MIT, která umožňuje tento framework používat ve všech možných projektech. Bootstrap je velmi populární knihovnou, pro kterou existuje nepřeberné množství dalších pluginů.

Pro responzivitu využívá 12 sloupcového layoutu, který umožňuje kóděrovi jednoduše definovat pozici bloků na různých zařízeních. Zároveň také v základu obsahuje velké množství komponent, jako jsou tlačítka, navigace, notifikační zprávy a mnoho dalšího.

### 2.2.2.2 Foundation

Foundation<sup>27</sup> je také velmi populární CSS framework pro vytváření responzivních webů. Stejně jako Bootstrap používá 12 sloupcový layout a množství jeho komponent je také srovnatelné. Nicméně jednotlivé elementy, jako např. tabulky jsou nadefinovány pouze v jednom stylu, kdežto Bootstrap definuje také třídy pro různě barevné tabulky, také pro jejich různé chování, atd.

Oproti Bootstrapu obsahuje některé komponenty navíc (formulářové validace, responzivní zdroje obrázků, aj.), ale jeho popularita zdaleka nedosahuje výše zmíněného konkurenta, o čemž svědčí dostupný počet různých pluginů pro tyto 2 frameworky.

Foundation, stejně jako Bootstrap, umožňuje také práci společně s CSS preprocesory<sup>28</sup>, jako jsou SASS<sup>29</sup> a v případě Bootstrapu i LESS<sup>30</sup>.

Jak vidno, v přímém porovnání těchto 2 frameworků jednoznačně nevítězí ani jeden. Protože se autor již s Twitter Bootstrapem dříve setkal, z touhy po nových znalostech se rozhodl pro svou práci využít služeb frameworku Foundation, stejně jako použití společně s CSS preprocesorem SASS.

### 2.2.3 JavaScript

Protože jedním z požadavků na aplikaci je její rychlost a obsluha aplikace bez nutnosti načítání celé stránky, v této práci je použito skriptovacího jazyku JavaScript<sup>31</sup>. Veškerý kód se vyhodnocuje na straně klienta, v prohlížeči.

**jQuery** „Knihovna jQuery klade důraz na interakci mezi značkovacím jazykem HTML a programovacím jazykem JavaScript“ [15] a velmi zjednodušuje

---

<sup>26</sup><http://getbootstrap.com>

<sup>27</sup><http://foundation.zurb.com>

<sup>28</sup>Preprocesory přidávají čistému CSS mnoho dalších vlastností, známých z programování, jako jsou funkce, proměnné, atd. Nevýhodou je nutná kompilace, jejíž výstupem je soubor v čistém CSS.

<sup>29</sup>Syntactically Awesome Style Sheets – <http://sass-lang.com>

<sup>30</sup><http://lesscss.org>

<sup>31</sup><http://www.w3schools.com/js>

syntaxi JavaScriptu. V současné době je dle [17] používána na 95 % webů, které používají nějakou JavaScriptovou knihovnu, což svědčí o její obrovské popularitě. Existuje pro ni také nespočet pluginů, jako např. jQuery UI<sup>32</sup>.

**AngularJS** Nad jazykem JavaScript je postaveno také několik frameworků, které slouží pro psaní rozsáhlých a složitých aplikací. Jedním z nich je framework s názvem AngularJS<sup>33</sup>, který je vyvíjen stejnou skupinou lidí, která vyvinula prohlížeč Google Chrome [16]. Tento framework je založen na MVC architektuře a obsahuje mnoho pokročilých technik, jako jsou ajaxové služby, dependency injection, správa historie prohlížeče a mnoho dalšího.

### 2.2.4 Chat v reálném čase

Pro implementaci chatu je možno použít v zásadě 2 postupy.

Prvním z nich je pomocí **WebSocketu**, nové vlastnosti HTML5. WebSocket „vytváří stavové spojení mezi prohlížečem a serverem“ [18], narozdíl od bezstavového HTTP<sup>34</sup> protokolu. Při používání této technologie začíná komunikace mezi serverem a klientem klasicky HTTP požadavkem a odpovědí, po navázání spojení se navíc otevře další, oboustranné spojení, které již obsluhuje WebSocket.

Druhou možností je využití pravidelné aktualizace stránky s chatem, pro příjemnější je určitě lépe využít služeb **AJAXu**<sup>35</sup>, takže se veškeré HTTP požadavky provádějí na pozadí.

Na první pohled se zdá, že WebSocket je pro tuto aplikaci jasná volba. Kromě správy chatu by se dal využít také např. na kontrolování nově zadaných otázek, úkolů, nově přidávaných příspěvků a mnoho dalšího. Problémem ovšem je, že se klient při otevření spojení pomocí WebSocketu musí připojit na nějaký server, který spravuje tuto komunikaci. Tím pádem by bylo nutné mít neustále spuštěný skript (ideálně napsaný v Node.js<sup>36</sup>), simulující právě tento server. Jelikož tato aplikace nefunguje jako webová služba, ale je nutné si ji nainstalovat na vlastní webový server, musel by se správce starat i o tento WebSocket server a v případě problémů reagovat. Pokud by totiž tento server přestal odpovídat, rázem by přestal fungovat celý chat. Protože každá škola nemusí mít ve svém středu člověka s požadovanými znalostmi i chutí mít tuto práci na starost, je pro tuto práci využita druhá varianta – **pravidelnou aktualizací stránky**. Tuto funkcionalitu obsluhuje výše zmíněný framework AngularJS.

---

<sup>32</sup><https://jqueryui.com>

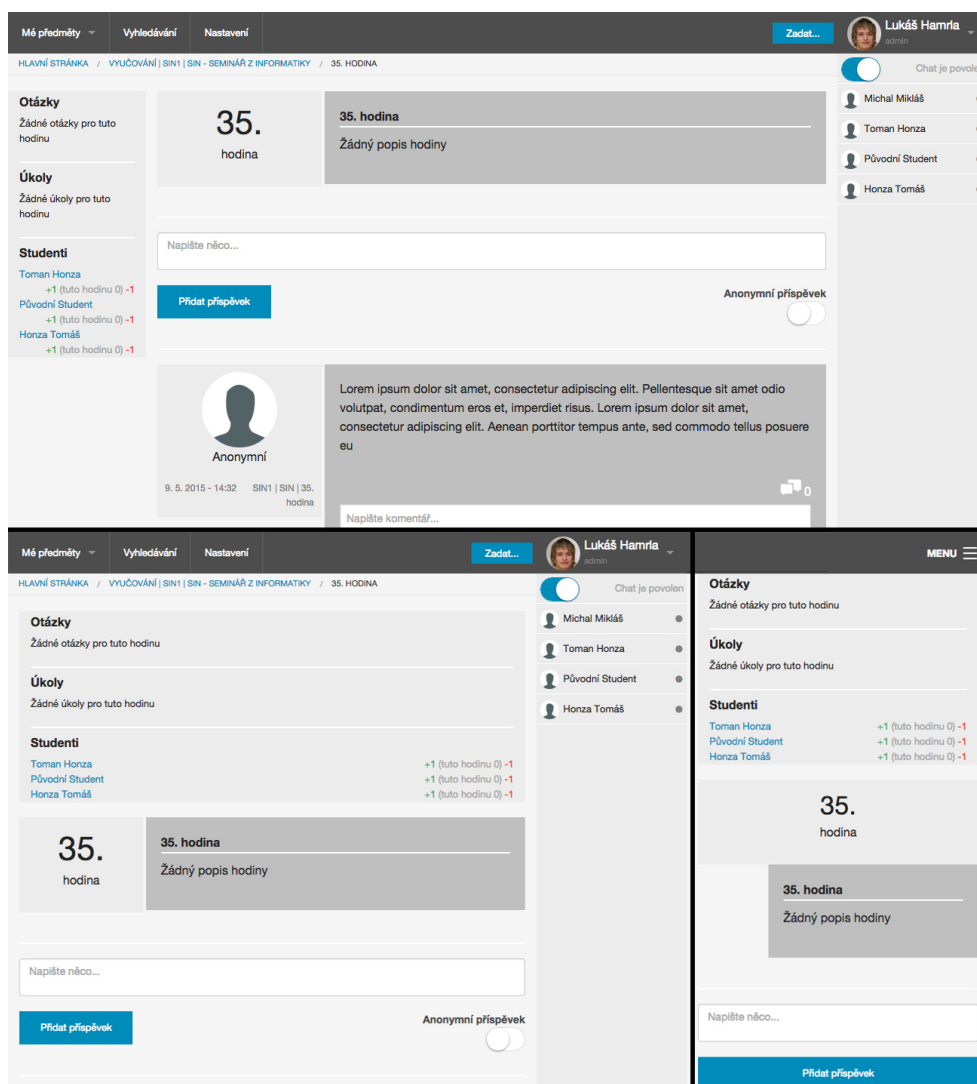
<sup>33</sup><https://angularjs.org>

<sup>34</sup>Hypertext Transfer Protocol

<sup>35</sup>Asynchronous JavaScript and XML

<sup>36</sup><https://nodejs.org>

## 2.3. Návrh webdesignu



Obrázek 2.4: Výsledný návrh webdesignu včetně ukázky responzivního zobrazení (nahore počítač, vlevo dole tablet, vpravo dole mobilní telefon)

## 2.3 Návrh webdesignu

Webdesign byl navržen v jednoduchém a minimalistickém stylu. Důležité také bylo zaměřit se na uživatelskou přívětivost a na co nejrychlejší používání. Proto akce, u kterých se předpokládá nejčastější používání, jsou dostupné pod ovládacími elementy, které jsou ihned viditelné. Při navrhování byla využita služba UXPin [19] s využitím elementů z frameworku Foundation a s ohledem na responzivitu. Výsledný návrh včetně zobrazení na různých zařízeních je viditelný na obrázku





---

## Realizace

V této kapitole je detailně popsána aplikace na úrovni zdrojového kódu. Konvence vychází ze zvolených technologií, primárně z frameworku Nette.

### 3.1 Podpůrný software

V projektu autor používal některé podpůrné programy pro instalaci knihoven a jejich závislostí, a to zejména Composer [20]. „Composer je nástroj pro správu závislostí v PHP. Nechá vás nadeklarovat závislé knihovny, které váš projekt potřebuje a poté je za vás nainstaluje“ [21]. Další nástroj, Bower [22], slouží také pro správu závislostí, tentokrát ale pro front-endové knihovny (např. již zmiňované jQuery) a v neposlední řadě je to nástroj s názvem Grunt [23], jehož doplňky umí mimo jiné také minifikovat<sup>37</sup> externí soubory, jako jsou skripty v JS<sup>38</sup> či CSS styly. Grunt zároveň umožňuje všechny tyto soubory sjednotit do jednoho, čímž se docílí kratší doby načítání, protože prohlížeč na straně klienta provede pouze jeden HTTP požadavek na server namísto několika dalších, které by se provedly pro každý soubor zvlášť.

### 3.2 Struktura projektu

#### 3.2.1 Adresářová struktura projektu

Adresářová struktura projektu vychází z předpřipravené šablony *nette-sandbox*<sup>39</sup>.

Zajímavostí je rozdělení konfiguračního souboru na globální, který obsahuje registraci jednotlivých tříd jako služby aplikace, a na lokální, který zahrnuje pouze informace o lokálním serveru, např. přístupové údaje k databázi. Zatímco dříve se používal jeden konfigurační soubor, který obsahoval přístu-

---

<sup>37</sup>Odstranění přebytečných netisknutelných znaků za účelem zmenšení velikosti souboru.

<sup>38</sup>JavaScript

<sup>39</sup><https://github.com/nette/sandbox>

app	..... zdrojové kódy aplikace
├─ config	..... globální i lokální konfigurační soubory ve formátu NEON
├─ controls	..... továrničky komponent a jejich šablony
├─ filters	..... nově definované filtry pro použití v šablonách
├─ forms	..... továrničky formulářů a jejich šablony
├─ model	..... skripty s datovou a business logikou, jako jsou namapované entity a servisní třídy
├─ presenters	..... presentery a šablony pro jednotlivé stránky
├─ router	..... továrnička pro vytvoření routeru, který mapuje URL na jednotlivé akce presenterů
├─ security	..... obsahuje skript, který nastavuje role a práva uživatelům v aplikaci
├─ bootstrap.php	..... spouštěcí skript celé aplikace
├─ log	..... složka s log soubory, jako jsou např. texty vyjímek
├─ node_modules	..... obsahuje pluginy použité v nástroji Grunt
├─ scss	..... zdrojové kódy pro CSS preprocesor SASS
├─ temp	..... složka používaná jako úložiště pro cache
├─ vendor	..... veškeré knihovny 3. stran
└─ www	..... jediná složka dostupná z webu, obsahuje CSS a JS soubory, obrázky, aj.

pové údaje jak na lokálním, tak na produkčním<sup>40</sup> serveru [24], jehož riziko spočívalo v možném (i neinvazivním) přístupu útočníka do počítače programátora a krádeži hesel k produkčnímu serveru, tímto rozdělením je riziko eliminováno.

#### 3.2.2 Presentery

Společným předkem všech presenterů v aplikaci je třída `BasePresenter`, která poskytuje všem svým potomkům instanci třídy `EntityManager` z knihovny `Kdyby\Doctrine`, skrz kterou mají přístup k databázi. Kromě třídy `LoginPresenter`, která se stará o vykreslení formuláře pro přihlášení a autorizaci uživatelů, dědí všechny ostatní presentery od třídy `AuthorizedBasePresenter`, která při každém načtení stránky kontroluje, zda je uživatel přihlášený.

#### 3.2.3 Model

Model je v aplikaci reprezentován servisními třídami pro business logiku, entitními třídami pro mapování objektů do databázových tabulek a repozitáři, které rozšiřují již existující repozitáře v knihovně `Kdyby\Doctrine` a přidávají jim další funkčnost.

---

<sup>40</sup>Server, kde běží „živá“ verze aplikace a je dostupná návštěvníkům.

### 3.2.4 Šablony

V aplikaci je jedna hlavní šablona (tzv. layout), která obsahuje odkazy na soubory se styly a JS skripty a definuje nutné bloky k vykreslení pro každou stránku. Obsahuje také blok s vertikálním a horizontálním menu, chatem a pro hlavní obsah. V blocích s vertikálním menu a chatem jsou vykresleny komponenty, které jsou podrobněji popsány v dalších kapitolách.

### 3.2.5 Komponenty

Komponenty jsou hlavním stavebním kamenem frameworku Nette a jsou nástrojem pro psaní znovupoužitelného kódu. Pro psaní komponent je dodržována konvence, kde je název třídy vždy `NazevKomponentyControl`, v případě formulářů je to `NazevFormulareForm`. Tyto komponenty jsou v aplikaci tvořeny tehdy, pokud se stejná logika nebo formulář používají na více místech nebo pokud je potřeba osamostatnit větší množství kódu, který obsluhuje jeden logický celek.

V projektu je vytvořeno několik samostatných komponent. Jsou to `QuestionSummaryControl`, starající se o přehled odpovědí na otázku a vykreslení grafu, `BreadcrumbsControl`, která má na starosti vykreslování drobečkové navigace, `MenuControl` vykreslující horizontální menu, `StudentsControl` a `StudentDetailControl`, které zobrazují detaily o konkrétním studentovi a umožňují mu přidávat body za aktivitu. Další používané komponenty jsou ty, které obsahují vykreslení a správu formulářů, např. pro přidání a editaci studenta. Komponenty nepopisované výše jsou představeny v dalších kapitolách. Komponenty nemusí být ve zvláštní třídě, mohou se také vytvořit přímo v těle metody `createComponentNazevKomponenty` v presenteru. Tato možnost je využívána tehdy, pokud se komponenta (typicky formulář), používá pouze v jednom presenteru.

Při psaní komponent je využívána velmi usnadňující vlastnost frameworku Nette a tou je tzv. `autowiring`<sup>41</sup>. Autowiring umožňuje automaticky vytvářet továrničky<sup>42</sup> pro jednotlivé komponenty a vkládat do nich závislosti dle `type hintů`<sup>43</sup>. Proto je ke každé komponentě ještě vytvořeno rozhraní, jehož název je ve tvaru `INazevKomponentyFactory` definující tvar továrničky. Pro příklad je v ukázce 3.1 uvedena komponenta `TaskForm`. Ta ve svém konstruktoru vyžaduje identifikátor *vyučovací hodiny* a instanci třídy `EntityManager`. Proto je vytvořeno rozhraní s názvem `ITaskFormFactory`, které obsahuje funkci `create`, s parametrem `$lessonId`. Jelikož je třída `EntityManager` zaregistrovaná jako služba, systémový kontejner Nette se postará o to, aby byla injektována automaticky. Rozhraní `ITaskFormFactory` je poté zaregistrováno

<sup>41</sup><http://doc.nette.org/cs/2.3/configuring>

<sup>42</sup>Továrnička (factory) je samostatná třída, která vytváří nové instance jiné třídy. Toto chování je popsáno návrhovým vzorem Factory.

<sup>43</sup>Vyžaduje, aby parametr ve funkci byl instancí určité třídy.

### 3. REALIZACE

---

```
class TaskForm extends Control
{
    public function __construct($lessonId,
        EntityManager $em)
    {...}
}

interface ITaskFormFactory {
    /**
     * @param int $lessonId
     * @return TaskForm
     */
    function create($lessonId);
}

// v presenteru pote staci pouze
return $this->taskFormFactory->create($lessonId);
```

---

Zdrojový kód 3.1: Použití autowiringu

v konfiguračním souboru také jako služba a Nette při vytvoření systémového kontejneru vytvoří továrničku, která implementuje toto rozhraní. V jakémkoliv presenteru je nyní možné si nechat injektovat objekt, který implementuje toto rozhraní a na ni zavolat metodu `create`, která vrátí instanci třídy `TaskForm`.

Na první pohled to vypadá jako více zbytečného kódu navíc, ale rozhodně tomu tak není. Pokud bude v budoucnu komponenta `TaskForm` vyžadovat např. službu `LessonService`, jediné, co stačí změnit, za předpokladu, že je tato služba registrovaná v konfiguračním souboru, je konstruktor třídy `TaskForm`. Bez použití této techniky by bylo nutné procházení celého kódu a hledání, kde všude se volá vytváření instance této komponenty. Navíc není potřeba jednotlivé závislosti komponenty injektovat i do presenterů, což by bez použití továrniček nebylo možné.

### 3.3 Přihlášení do aplikace

Obsah aplikace je dostupný pouze autentizovaným uživatelům, proto je jako první stránka aplikace přihlašovací formulář. Po přihlášení se provede ověření uživatele oproti databázi; toto ověření obsluhuje třída `UserService`, která implementuje rozhraní `IAuthenticator`. Hesla jednotlivých uživatelů jsou uložena v databázi v jejich zahashované podobě, jež je výsledkem PHP funkce `crypt`, která dle [25] používá hashovací algoritmy typu DES. Konkrétní algoritmus je vybrán na základě dostupnosti v operačním systému. Před hashová-

---

```

<input type="text" class="liveSearch"
  data-url="{link searchStudents!}"
  data-spinner="studentSpinner"/>
{snippet students}
  
  // vypis studentu
{/snippet}

```

---

Zdrojový kód 3.2: Snippets v šabloně

---

```

public function handleSearchStudent($query)
{
  $this->template->students =
    $repository->findByName($query);
  $this->redrawControl('students');
}

```

---

Zdrojový kód 3.3: Práce se snippets v PHP skriptu

ním se k samotnému heslu přidá náhodně generovaná sůl<sup>44</sup>.

## 3.4 Evidence studentů a tříd

Zasahovat do evidence studentů a *tříd* může pouze administrátor. Na samostatné stránce „Nastavení“ má k dispozici správu studentů, *tříd*, učitelů, předmětů a školních roků. Pro rychlé vyhledávání je zde využit live search<sup>45</sup>, který je implementován v JavaScriptu jednoduchou funkcí, která si načte z `data` atributů elementu `input` informace, na kterou URL má posílat AJAXový HTTP požadavek a které načítací kolečko má zobrazit. Překreslování jednotlivých částí na stránce je řešeno pomocí snippetů z Nette Frameworku. Snippets jsou jednotlivé výstřižky ze stránky, které se umí překreslovat bez nutnosti načtení celé stránky [26]. V ukázkách kódu 3.2 a 3.3 je pro příklad uvedeno vyhledávání mezi studenty.

### 3.4.1 Import studentů

Po vytvoření *třídy* nebo *vyučovací skupiny* je nutné ji naplnit studenty. Kromě ručního vytváření či přiřazování žáků je také možnost importu z informačního

---

<sup>44</sup>Soubor znaků, které se připojují k heslu pro zvýšení zabezpečení.

<sup>45</sup>Hledání v datech bez nutnosti odesílat formulář.

Název skupiny		
N žáků		
typ		
váha		
datum		
téma(poznámka)		
Hamrla	Lukáš	Č4.A
Novák	Jan	Č4.B
...	...	...

Tabulka 3.1: Tvar exportovaného souboru z informačního systému Bakaláři

systému Bakaláři. Jelikož tento IS<sup>46</sup> neposkytuje žádné API<sup>47</sup>, import se dá provést pouze ze souboru vygenerovaného z tohoto systému. Nejlepší možností by byla dostupnost jednoho souboru, ve kterém by byl seznam všech studentů a tříd a vyučovacích skupin, které navštěvují. Po konzultaci se samotnými autory programu, kteří potvrdili, že žádná taková možnost neexistuje, doporučili využít možnost exportu seznamu studentů ze sekce pro zápis známek u každé třídy či skupiny zvlášť. Tento soubor je ve formátu XLS a má předem definovanou strukturu, čehož je v aplikaci využito. Nebezpečím je rozdílný tvar tohoto souboru pro různé (i budoucí) verze systému Bakaláři. Aplikace implementovaná v rámci této práce umí pracovat se souborem vyexportovaným z verze programu Bakaláři 14/15, nicméně dle autorů by předchozí verze neměly mít rozdílný formát. Tvar souboru je viditelný v tabulce 3.1.

Kód, starající se o samotný import, je součástí třídy `StudentService` a k přečtení XLS souboru používá knihovnu `PHPEXcel` [27]. Tento kód se umí vypořádat i s drobnou odlišností ve struktuře souboru, jako jsou například prázdné buňky, aj. Dle jména studenta a názvu třídy umí také vyhledat výskyt tohoto studenta v minulém roce a nevytvářet stejnému studentovi více účtů. Tato funkcionality ovšem počítá s klasickým značením tříd na školách, tzn. název třídy obsahuje číslo, které se každý rok zvýší o 1. Není možné zjistit jednoznačný identifikátor již existujícího studenta z programu Bakaláři, protože v exportu není dostupný. Tato možnost importu je velká výhoda oproti Moodle, který sice má možnost importu vlastního souboru, ale tento soubor musí mít přesnou strukturu, která ovšem nevyhovuje struktuře exportovaného souboru ze systému Bakaláři.

### 3.4.2 Vytvoření vyučování

Aby studenti a učitelé mohli aplikaci využívat, je potřeba vytvořit vyučování. To se skládá z třídy nebo vyučovací skupiny, předmětu, učitelů a doby výuky.

---

<sup>46</sup>Informační systém

<sup>47</sup>Application Programming Interface

---

```

$form = new Form();
$form->addDynamic('teachers',
    function (Container $teacher) {
        $teacher->addText('name', 'Jmeno');
    }
);

```

---

Zdrojový kód 3.4: Dynamické kontejnery u formulářů

Učitelů a časů výuky může být samozřejmě více, proto je v tomto případě využito knihovny `Kdyby\Replicator` [28], která umožňuje přidání dynamických kontejnerů (dalších formulářů) do již existujícího formuláře; ukázka použití je viditelná ve zdrojovém kódu 3.4. S použitím AJAXu probíhá veškerá logika přidávání bez nutnosti aktualizace stránky. Při vytváření *vyučování* není nutné mít předem existující účet pro vyučujícího, může se vytvořit při samotné tvorbě *vyučování*. Pokud je k *vyučování* přidána doba výuky, automaticky se vytvoří jednotlivé hodiny napříč celým školním rokem. Zároveň se při vytváření kontrolují státní svátky v České republice, včetně Velikonoc, kde je použita zajímavá PHP funkce `easter_date`. *Vyučovací hodiny* také mohou manuálně vytvořit a spravovat vyučující patřící ke konkrétnímu *vyučování*.

## 3.5 Evidence úkolů a otázek

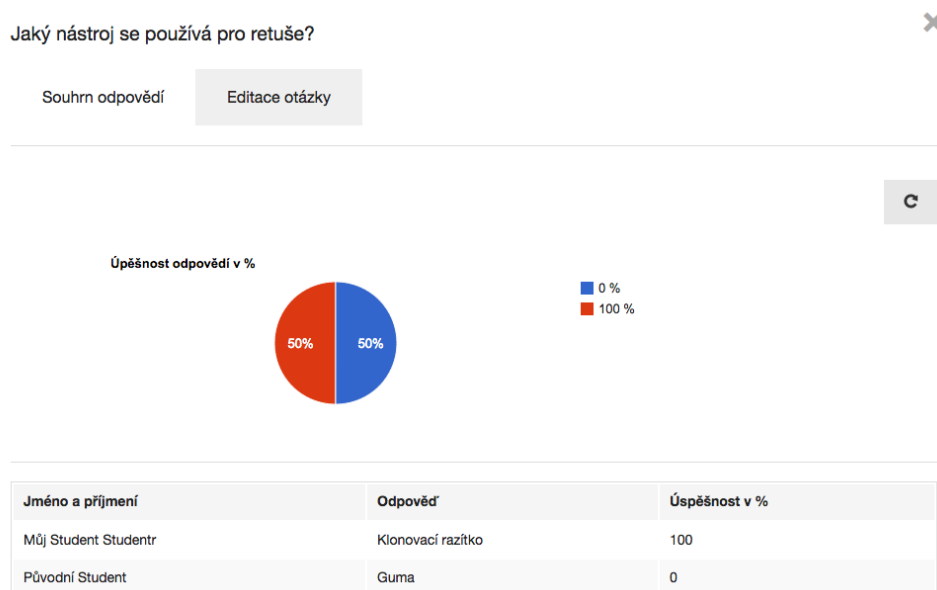
Na stránce s *vyučovací hodinou* může učitel vytvořit úkol či otázku. O tuto logiku se stará komponenta `NewActivityControl`, která v sobě obsahuje továrničky na další komponenty `QuestionForm` a `TaskForm`. Formulář pro vytvoření je dostupný v modálním okně po kliknutí na tlačítko v menu. Při vytváření si učitel může vybrat, zda-li chce vytvořit úplně novou entitu či zkopírovat již dříve vytvořenou a použitou. Pokud provede kopii, nová otázka či úkol se spáruje s již původní a tuto skutečnost bude možné poté využít např. při prezentaci nejlepších odevzdaných úkolů.

### 3.5.1 Vytvoření a souhrn otázky

Při vytvoření otázky má vyučující možnost položit otevřenou otázku, uzavřenou s jednou správnou odpovědí nebo uzavřenou s více možnými odpověďmi. Po uložení otázky vidí v levém menu přehled všech otázek a po jejich rozkliknutí se mu zobrazí modální okno, do kterého se načte lazy způsobem<sup>48</sup> aktuální informace o otázce, včetně grafů úspěšnosti odevzdaných odpovědí, jak je viditelné na obrázku 3.1. Pro grafy byl použit nástroj `Google Charts` [29]. Učitel může upravit hodnocení jakékoliv otázky pouhým kliknutím na toto číslo a přepsáním. Toto je umožněno díky novému atributu `contenteditable`,

<sup>48</sup>Metoda, kdy se data načítají až v době vyžádání, ne dříve.

### 3. REALIZACE



Obrázek 3.1: Přehled odpovědí včetně vizualizace do grafu

který je jednou z novou vlastností HTML5<sup>49</sup>, jež dovoluje jakémukoliv elementu v html kódu nastavit, jestli je přímo editovatelný.

#### 3.5.2 Vytvoření a souhrn úkolu

Pokud se vyučující rozhodne nastavit úkolu limit, má na výběr ze 2 typů:

1. **volný**, při kterém budou moci studenti odevzdávat vypracované úkoly i po jeho uplynutí, ale vyučující o tom bude informován a může na to reagovat např. stržením bodů,
2. **striktní**, po jehož uplynutí už nebude možnost odevzdat úkol vůbec anebo nemusí být nastaven žádný limit.

Učitel také může u úkolu nastavit, jestli si studenti budou moci navzájem anonymně ohodnotit odevzdané úkoly. Pokud bude tato možnost zvolena, vyučující poté u jednotlivých odevzdaných úkolů jejich průměrné hodnocení. Rozsah hodnocení je od 0 do 5 bodů a je vyřešeno systémem hvězdiček s pomocí pluginu RateIt [30]. Tuto logiku obstarává komponenta `RatingControl`. Pokud je navíc odevzdaný úkol obrázkem, uživatelé si jej můžou prohlédnout ve vyšším rozlišení s pomocí knihovny Fancybox [31], která také umožňuje učiteli vytvořit prezentaci nejlepších prací dle hodnocení studentů. V modálním okně se souhrnem otázky má učitel rovněž k dispozici odevzdané úkoly

<sup>49</sup>[http://www.w3schools.com/tags/att\\_global\\_contenteditable.asp](http://www.w3schools.com/tags/att_global_contenteditable.asp)



---

```

public function createComponentComments()
{
    $userId = $this->userId;
    $em = $this->em;
    return new Multiplier(
        function($postId) use ($userId, $em){
            $post = $em->find(Post::getClassName(), $postId);
            $comments =
                new CommentsControl($em, $post, $userId);
            return $comments;
        });
}

```

---

Zdrojový kód 3.5: Použití třídy Multiplier

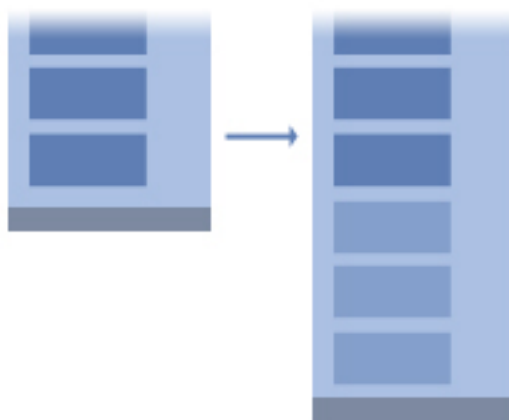
z minulých zadání tohoto úkolu, ze kterých si může vybrat určitý počet nejlepších a z nich také vytvořit prezentaci. Studenti navíc vidí na hlavní stránce přehled všech úkolů k vypracování.

## 3.6 Veřejná diskuze pro účastníky vyučování

Na každé stránce s *vyučováním* a s konkrétní *vyučovací hodinou* je dostupná veřejná diskuze formou příspěvků. Tuto funkcionalitu obstarává komponenta s názvem `PostsControl`, která v sobě obsahuje továrničku pro vytvoření formuláře pro příspěvek a také továrničku pro vytvoření formuláře s možností vložit komentář. Pro každý příspěvek je ovšem potřeba vlastní formulář pro přidání komentáře. Z pohledu rychlosti by určitě nebylo ideální před vykreslením stránky projít všechny příspěvky v cyklu a pro ty vytvořit zvláštní formulář, proto je zde využita třída `Nette\Application\UI\Multiplier`<sup>50</sup>, která „funguje na principu vnořených komponent – každá komponenta dědí od `Nette\ComponentModel\Container` může obsahovat další komponenty“ [32]. V šabloně poté stačí použít `{control comments-$post->id}`. Tento kód způsobí při prvním průchodu zavolání metody `createComponentComments` s potomkem `$post->id`. `Multiplier` název potomka přijme v argumentu callbacku, který dle něj vytvoří samostatnou novou komponentu. Přesněji je to uvedeno v ukázce kódu 3.5. V tomto případě se vytváří instance komponenty `CommentsControl`, která má na starosti přidávání a zobrazování komentářů.

Text příspěvku a komentáře zůstane po použití PHP funkce `strip_tags` bez html tagů, což je potřeba pro to, aby bylo možné při výpisu textu nalézt odkazy na web a tyto odkazy nahradit za hypertextové odkazy. Proto je použití funkce `strip_tags` nezbytné, protože se pak text příspěvku zobrazí

<sup>50</sup><http://api.nette.org/2.3.1/Nette.Application.UI.Multiplier.html>



Obrázek 3.2: Ukázka nekonečného rolování [34]

bez escapování tagů, čehož by mohli využít útočníci. Pro nalezení a nahrazení URL v textu za hypertextové odkazy je vytvořen nový filtr v šabloně, který pomocí regulárního výrazu [33] toto provede.

Uživatel poté na hlavní stránce vidí všechny příspěvky ze všech *vyučování*, do kterých je zapsán a může na ně přímo reagovat. S postupem přes konkrétní *vyučování* až po *vyučovací hodinu* se sada příspěvků zužuje dle navštívené stránky.

Při vkládání nového příspěvku si uživatel může zvolit, jestli jeho příspěvek bude anonymní či nikoliv. Je nutno podotknout, že příspěvek se jako anonymní tváří pouze na webu před uživateli, v databázi je u příspěvku stále uveden jeho autor, a to z důvodu možného nevhodného chování a textů, u kterých je potřeba zjistit pravého autora.

Protože příspěvků může být velké množství, bylo potřeba použít stránkování. Toto je vyřešeno použitím tzv. nekonečného rolování, které zaručuje to, že po kliknutí na další stránku se pouze přidají příspěvky z další stránky pod již aktuální příspěvky.

## 3.7 Chat mezi studenty a učiteli

Chat je jednou z největších výhod aplikace implementované v rámci této práce, protože žádný z výše popisovaných LMS systémů tímto druhem komunikace nedisponuje. Mají možnost pouze chatovací místnosti, ale není možné spustit chat pouze mezi dvěma uživateli.

O veškerou logiku spojenou s chatem se na straně serveru stará komponenta s názvem `ChatControl`. Tato komponenta vyžaduje v konstruktoru instanci aktuálně přihlášeného uživatele, školní rok a případně aktuální *vyučo-*

*vání*. Na základě těchto dat zjistí všechny uživatele, kteří se budou aktuálně přihlášenému v chatu zobrazovat. Zobrazují se všichni učitelé a studenti z *vyučování*, do kterých je přihlášený uživatel přiřazen.

Na straně klienta obsluhuje tuto logiku presenter `ChatController`, který je napsán v JavaScriptovém frameworku AngularJS.

### 3.7.1 Dostupnost uživatele pro ostatní přihlášené

Při přihlášení uživatele do aplikace je potřeba označit tohoto uživatele jako dostupného pro přijímání zpráv pro ostatní přihlášené. Toto je vyřešeno JS skriptem, který každou půl minutu odesílá HTTP požadavek na URL, která aktualizuje poslední aktivitu přihlášeného uživatele na aktuální datum a čas. Ostatní uživatelé vidí tohoto uživatele jako dostupného, pokud je rozdíl mezi aktuálním časem a časem poslední aktivity menší než 45 vteřin. V opačném případě se uživatel hlásí jako nedostupný. Pokud je uživatel nedostupný, i tak mu mohou ostatní posílat zprávy, na které bude příjemce upozorněn po nejbližším přihlášení.

### 3.7.2 Správa konverzací a vzhled chatu

Vzhled a funkčnost celého chatu je inspirován provedením v sociální síti Facebook<sup>51</sup>, kde je okno s uživateli na pravé straně obrazovky s fixovanou pozicí, což znamená že i při rolování stránky je toto okno stále na stejném místě. V této ploše jsou zobrazeni všichni uživatelé, se kterými je možné začít konverzaci. První problém vzniká při nutnosti řazení těchto uživatelů v okně. Je totiž žádoucí, aby od sebe byli odděleni studenti a učitelé. Nicméně v aplikaci jsou jednotliví uživatelé dostupní v poli, pod indexem s hodnotou jejich identifikátoru v databázi. AngularJS sám poskytuje filtr `orderBy` při použití s direktivou `ng-repeat`, nicméně funguje pouze na pole objektů, ne na asociativní pole objektů [36]. Proto bylo potřeba použít vlastní filtr, který toto seřazení provede. Poté stačí v šabloně použít `<li ng-repeat="user in users | orderBy: 'ordering'">`. Nicméně s pomocí tohoto filtru se poté nedá zjistit předchozí prvek v cyklu, takže v seznamu uživatelů nejsou učitelé od studentů odděleni žádným viditelným oddělovačem.

Po kliknutí na uživatele se otevře okno a načte se předchozí konverzace s uživatelem, kde je opět využito vlastnosti nekonečného rolování. Pokud je již okno s konverzací otevřené, pouze se přesune na první pozici před ostatní okna. Zavření okna proběhne kliknutím na ikonu s křížkem nebo stisknutím klávesy ESC. Okna se dají také minimalizovat kliknutím na lištu se jménem. Toto okno má pevně danou šířku a je plovoucí, takže se řadí vedle sebe tak dlouho, dokud již není místo na obrazovce. Z tohoto důvodu je chat dostupný až na zařízeních s šířkou větší než 641 px (toto rozlišení odpovídá malým tabletům).

---

<sup>51</sup><http://www.facebook.com>

Je také důležité, aby při přechodu na další stránku zůstaly uživateli otevřená stejná okna konverzace a nemusel je neustále znovu otevírat. Zde je využito technologie HTML5, konkrétně Local Storage<sup>52</sup>. Tato technologie umožňuje ukládat data přímo do uživatelova prohlížeče. Její použití je velmi jednoduché – pro vložení dat do tohoto úložiště stačí zavolat `localStorage.setItem('popups' + $scope.userId, angular.toJson($scope.popups))` a pro získání dat `localStorage.getItem('popups' + $scope.userId)`. Je důležité upozornit, že data musí být pouze v textové podobě, není možno uložit samotný objekt, proto je v tomto případě objekt konvertován do jeho reprezentace ve formátu JSON<sup>53</sup>. Dále také stojí za zmínku tvar klíče, pod kterým se ukládají data. Tento klíč musí obsahovat identifikátor přihlášeného uživatele, protože se může stát, že stejný prohlížeč používá více uživatelů a každý může mít otevřená jiná okna.

#### 3.7.3 Odesílání a přijímání zpráv

V otevřené konverzaci může uživatel poslat zprávu jinému uživateli. Stačí napsat text do textového pole a stisknout klávesu ENTER. Kód v AngularJS po stisknutí této klávesy provede HTTP požadavek na URL s parametry obsahující id příjemce a text zprávy. Server poté tuto zprávu uloží do databáze.

Zároveň se přihlášenému uživateli průběžně kontrolují nové přijaté zprávy. Časový interval mezi jednotlivými dotazy roste exponenciálně z důvodu úspory HTTP požadavků na server. Při přístupu na stránku začíná interval na 1 vteřině a s každým dotazem na nové zprávy se interval zvýší dvojnásobně až do hodnoty 32 vteřin. Tento interval se opět vyrestartuje na dobu 1 sekundy při odeslání zprávy, kdy se předpokládá, že s přibývajícím časem po odeslání zprávy se snižuje pravděpodobnost odpovědi od druhé strany. Pokud je nalezena nová zpráva, uživateli se otevře nové okno s konverzací s adresátem zprávy a dokud na toto okno neklikne, bude mu také svítit u jména adresáta počet nepřečtených zpráv. Tento počet se také ukládá do Local Storage, aby nedošlo k přehlédnutí nové zprávy po přechodu na jinou stránku.

#### 3.7.4 Zapínání a vypínání chatu

Učitel v daném *vyučování* může zapnout nebo vypnout chat pro všechny studenty, kteří jsou součástí tohoto *vyučování*. Tito studenti poté nemohou využívat služeb chatu nikde v aplikaci, ani na stránce s jiným *vyučováním*.

---

<sup>52</sup>[http://www.w3schools.com/Html/html5\\_webstorage.asp](http://www.w3schools.com/Html/html5_webstorage.asp)

<sup>53</sup>JavaScript Object Notation

---

## Závěr

V rámci této práce byl úspěšně implementován portál pro podporu výuky na středních školách. Samotné implementaci, která je detailně popsána v kapitole Realizace, předcházelo provedení analýzy procesů, funkčních a nefunkčních požadavků a vytvoření business analytického modelu, čemuž se věnuje kapitola Analýza.

Jedním z hlavních požadavků pro implementaci byla dostupnost aplikace na webovém rozhraní, proto bylo použito programovacího jazyku PHP a databázového stroje MySQL. Responzivní design webu byl vyřešen pomocí CSS frameworku Foundation s využitím CSS preprocesoru SASS. Dalším požadavkem, který je mezi ostatními portály tohoto typu unikátní, byla možnost chatu mezi studenty a učiteli. Pro tuto potřebu byl vybrán JavaScriptový framework AngularJS. Těmto a dalším použitým technologiím se podrobně věnuje kapitola Návrh, která obsahuje také návrh webdesignu a relační databázový model.

Aplikace implementovaná v rámci této práce bude využívána na Gymnáziu a Jazykové škole s právem státní Jazykové zkoušky ve Zlíně, s jehož zástupci z řad pedagogů a technických pracovníků proběhlo v průběhu práce několik konzultací.

V případě zájmu bude vývoj aplikace pokračovat i nadále, budou implementovány další vlastnosti, jako např. zadávání a hodnocení písemek. Pokud se aplikace osvědčí v reálném provozu na výše zmíněném gymnáziu, bude nabídnuta dalším školám ve Zlínském kraji.



---

## Použité zdroje

- [1] *Learning Management System (LMS)* [online]. ABZ.cz . [vid. 10.5.2015] Dostupné z: <http://slovník-cizích-slov.abz.cz/web.php/slovo/learning-management-system-lms>
- [2] *Top LMS Software* [online]. Capterra Inc. [vid. 7.5.2015] Dostupné z: <http://www.capterra.com/learning-management-system-software/?utf8=%E2%9C%93&v=4#infographic>
- [3] *About Moodle* [online]. Moodle Pty Ltd. [vid. 19.11.2014] Dostupné z: [https://docs.moodle.org/28/en/About\\_Moodle](https://docs.moodle.org/28/en/About_Moodle)
- [4] *Blackboard Learn - Installation Guide* [online]. Blackboard inc. [vid. 19.11.2014] Dostupné z: [http://library.blackboard.com/ref/df5b20ed-ce8d-4428-a595-a0091b23dda3/Content/Print\\_Output/Blackboard\\_Learn\\_91\\_Installation.pdf](http://library.blackboard.com/ref/df5b20ed-ce8d-4428-a595-a0091b23dda3/Content/Print_Output/Blackboard_Learn_91_Installation.pdf)
- [5] *Edmodo* [online]. Edmodo, Inc. [vid. 19.11.2014] Dostupné po registraci: <https://www.edmodo.com/home>
- [6] *Usage of server-side programming languages for websites* [online]. W3Techs. [vid. 19.11.2014] Dostupné z: [http://w3techs.com/technologies/overview/programming\\_language/all](http://w3techs.com/technologies/overview/programming_language/all)
- [7] *PHP: Releases* [online]. PHP Group. [vid. 21.2.2014] Dostupné z: <http://php.net/releases/>
- [8] Percentages of websites using various subversions of PHP 5. In: *W3Techs* [online]. Q-Success. [vid. 2.5.2015] Dostupné z: <http://w3techs.com/technologies/details/pl-php/5/all>
- [9] *Installation - CakePHP Cookbook 2.x documentation* [online]. Cake Software Foundation, Inc. [vid. 21.2.2014] Dostupné z: <http://book.cakephp.org/2.0/en/installation.html>

- [10] *Best PHP Framework for 2015 – SitePoint Survey Results* [online]. SitePoint Pty. Ltd. [vid. 28. 3. 2015] Dostupné z: <http://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
- [11] MVC Process. In: *Sitepoint* [online]. SitePoint Pty. Ltd. [vid. 5. 5. 2015] Dostupné z: <http://www.sitepoint.com/the-mvc-pattern-and-php-1/>
- [12] PROCHÁZKA, Filip. *Kdyby\Doctrine*. [software]. [vid. 5. 4. 2015]. Dostupné z: <https://github.com/Kdyby/Doctrine>
- [13] <http://phpfashion.com/escapovani-definitivni-prirucka> [online]. GRUDL, David. [vid. 10. 5. 2015] Dostupné z: <http://phpfashion.com/escapovani-definitivni-prirucka>
- [14] *Zabezpečení před zranitelnostmi* [online]. Nette Foundation. [vid. 2. 5. 2015] Dostupné z: <http://doc.nette.org/cs/2.3/vulnerability-protection#toc-cross-site-request-forgery-csrf>
- [15] MARGORÍN, Marián. *jQuery bez předchozích znalostí*. Brno: Computer Press, 2011. ISBN 978-80-251-3379-8
- [16] LERNER, Ali. *ng-book*. San Francisco: Fullstack.io, 2013. ISBN 978-0-9913446-0-4
- [17] *Usage statistics and market share of JQuery for websites* [online]. Q-Success. [vid. 1. 5. 2015] Dostupné z: <http://w3techs.com/technologies/details/js-jquery/all/all>
- [18] HOGAN, Brian P. *HTML5 a CSS3: výukový kurz webového vývojáře*. Přeložil Jakub URBAN. Brno: Computer Press, 2011. ISBN 978-80-251-3576-1
- [19] UXPIN SP. Z O. O. *UXPin*. [software]. [vid. 12. 3. 2015]. Dostupné z: <https://www.uxpin.com/>
- [20] ADERMANN, Nils a Jordi BOGGIANO. *Composer*. [software]. [vid. 25. 4. 2015]. Dostupné z: <https://getcomposer.org/download/>
- [21] *Getting Started*. [online]. ADERMANN, Nils a Jordi BOGGIANO. [vid. 25. 4. 2015]. Dostupné z: <https://getcomposer.org/doc/00-intro.md>
- [22] TWITTER, INC. *Bower*. [software]. [vid. 25. 4. 2015]. Dostupné z: <http://bower.io/>
- [23] ALMAN Ben. *Grunt*. [software]. [vid. 25. 4. 2015]. Dostupné z: <http://gruntjs.com/getting-started>
- [24] *Nette\Environment* [online]. Nette Foundation. [vid. 20. 8. 2014] Dostupné z: <http://doc.nette.org/cs/0.9/nette-environment>



- 
- [25] *PHP: crypt.* [online]. PHP Group. [vid. 5. 5. 2015]. Dostupné z: <http://php.net/manual/en/function.crypt.php>
- [26] *AJAX & snippety.* [online]. Nette Foundation. [vid. 5. 5. 2015]. Dostupné z: <http://doc.nette.org/cs/2.3/ajax>
- [27] BAKER Mark a Maarten BALLIAUW. *PHPExcel.* [software]. [vid. 1. 4. 2015]. Dostupné z: <http://phpexcel.codeplex.com/>
- [28] PROCHÁZKA, Filip. *Kdyby\Replicator.* [software]. [vid. 5. 4. 2015]. Dostupné z: <https://github.com/Kdyby/Replicator>
- [29] GOOGLE, INC. *Google Charts.* [software]. [vid. 20. 4. 2015]. Dostupné z: <https://developers.google.com/chart/>
- [30] JUNGE, Gideon. *RateIt.* [software]. [vid. 5. 4. 2015]. Dostupné z: <https://rateit.codeplex.com/>
- [31] SKARNELIS, Janis. *Fancybox 2.* [software]. [vid. 18. 4. 2015]. Dostupné z: <http://fancyapps.com/fancybox/>
- [32] *Multiplier.* [online]. DOBEŠ, Vojtěch. [vid 2. 4. 2015]. Dostupné z: <http://pla.nette.org/cs/multiplier>
- [33] *URL regular expression for links with or without....* [online]. O'RIORDAN, Matthew. [vid 5. 5. 2015]. Dostupné z: <https://tools.ietf.org/html/rfc6455>
- [34] Infinite scroll pattern. In: *Infinite Scroll / jQuery plugin* [online]. Paul Irish. [vid 5. 5. 2015]. Dostupné z: <http://www.infinite-scroll.com/wp-content/uploads/2008/07/infinite-scroll-pattern.gif>
- [35] *AngularJS Filter for Ordering Objects (Associative Arrays or Hashes) with ngRepeat.* [online]. KLEMM, Justin. [vid 18. 4. 2015]. Dostupné z: <http://justinklemm.com/angularjs-filter-ordering-objects-ngrepeat/>
- [36] GOOGLE, INC. *AngularJS.* [software]. [vid. 20. 6. 2014]. Dostupné z: <https://angularjs.org/>
- [37] ZURB, INC. *Foundation.* [software]. [vid. 3. 2. 2015]. Dostupné z: <http://foundation.zurb.com/>
- [38] BEŇO, Peter. *Foundation Datpicker.* [software]. [vid. 2. 4. 2015]. Dostupné z: <https://github.com/najlepsiwebdesigner/foundation-datpicker>

- [39] BEŇO, Peter. *Foundation Datetimepicker*. [software]. [vid. 20. 4. 2015]. Dostupné z: <https://github.com/najlepsiwebdesigner/foundation-datetimepicker>
- [40] LUEGG. *angular-scroll-glue*. [software]. [vid. 10. 4. 2015]. Dostupné z: <https://github.com/Luegg/angularjs-scroll-glue>
- [41] GOOGLE, INC. *Google Chrome*. [software]. [vid. 1. 7. 2014]. Dostupné z: <https://www.google.com/chrome/browser/desktop/>
- [42] JETBRAINS S.R.O. *PHP Storm*. [software]. [vid. 11. 6. 2014]. Dostupné z: <https://www.jetbrains.com/phpstorm/>
- [43] OLŠÁK, Petr. *vlna*. [software]. [vid. 1. 5. 2015]. Dostupné z: <http://ftp.linux.cz/pub/tex/local/cstug/olsak/vlna/>
- [44] SEQUEL PRO a COCOAMYSQL. *Sequel Pro*. [software]. [vid. 1. 7. 2015]. Dostupné z: <http://www.sequelpro.com/>
- [45] NETTE FOUNDATION. *Nette Framework*. [software]. [vid. 1. 2. 2015]. Dostupné z: <http://nette.org/>
- [46] PETKOV, Alexander. *Switchery*. [software]. [vid. 20. 3. 2015]. Dostupné z: <http://abpetkov.github.io/switchery/>
- [47] JACOB, Caleb a Louis AMELINE. *Tooltipster*. [software]. [vid. 20. 3. 2015]. Dostupné z: <http://abpetkov.github.io/switchery/>
- [48] DOBEŠ, Vojtěch. *nette.ajax.js*. [software]. [vid. 10. 1. 2015]. Dostupné z: <https://github.com/vojtech-dobes/nette.ajax.js>
- [49] VAN RIEL, Mike. *phpDocumentor*. [software]. [vid. 10. 1. 2015]. Dostupné z: <http://www.phpdoc.org/docs/latest/getting-started/installing.html>

## Seznam použitých zkratek

**AJAX** Asynchronous JavaScript and XML

**API** Application Programming Interface

**CSS** Cascading Style Sheets

**HTML** HyperText Markup Language

**HTTP** Hypertext Transfer Protocol

**IS** Informační systém

**JS** JavaScript

**JSON** JavaScript Object Notation

**LMS** Learning Management System

**MVC** Model-View-Controller

**PHP** Hypertext Preprocessor

**SASS** Syntactically Awesome Style Sheets

**SQL** Structured Query Language

**URL** Uniform Resource Locator

**WWW** World Wide Web



---

## Instalační příručka

1. Celý obsah adresáře `src` nahrejte do kořenového adresáře webového serveru.
2. Nastavte kořenovou složku webu do složky `www` (pokud to nelze provést, budete muset za url přidávat `/www/` nebo dle návodu na stránkách Nette<sup>54</sup> vytvořte požadovaný soubor `.htaccess`).
3. Složkám `temp` a `log` umožňete práva pro zápis všem uživatelům.
4. Vytvořte databázi s jakýmkoliv názvem.
5. Přejděte na adresu `www.vase-domena.cz/install` a zadejte požadované údaje. Pokud vše proběhne v pořádku, uvidíte zprávu s přístupovými údaji do aplikace.

Po provedení těchto kroků byste měli po přístupu na hlavní stránku `www.vase-domena.cz` vidět přihlašovací formulář. Nyní se můžete přihlašovacími údaji, které Vám byly poskytnuty při instalaci, přihlásit a používat aplikaci.

---

<sup>54</sup><http://pln.nette.org/cs/faq>



---

## Obsah přiloženého CD

analysis-design.....	výstupy z analytické a návrhářské části práce
readme.txt.....	stručný popis obsahu CD
src	
├ docs.....	API dokumentace k aplikaci
├ impl.....	zdrojové kódy implementace
└ thesis.....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
text.....	text práce
├ thesis.pdf.....	text práce ve formátu PDF
└ zadani.pdf.....	schválené zadání práce