

ZÁPADOČESKÁ UNIVERZITA V PLZNI  
FAKULTA PEDAGOGICKÁ  
KATEDRA VÝPOČETNÍ A DIDAKTICKÉ TECHNIKY

**TVORBA WEBOVÉHO SYSTÉMU PRO VEDENÍ  
PEDAGOGICKÝCH PRAXÍ**  
BAKALÁŘSKÁ PRÁCE

**Luboš Hubáček**

*Přírodovědná studia, Informatika se zaměřením na vzdělávání*

Vedoucí práce: PhDr. Tomáš Jakeš, Ph.D.

**Plzeň 2018**

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně  
s použitím uvedené literatury a zdrojů informací.

V Plzni, 29. června 2018

.....  
vlastnoruční podpis

## Poděkování

Chtěl bych na tomto místě poděkovat vedoucímu této bakalářské práce PhDr. Tomáši Jakešovi, Ph.D. za odborné vedení a pomoc při zpracování této práce.

ZDE SE NACHÁZÍ ORIGINAL ZADÁNÍ KVALIFIKAČNÍ PRÁCE.

## OBSAH

SEZNAM ZKRATEK .....	3
ÚVOD .....	4
1 SYSTÉM PEDAGOGICKÝCH PRAXÍ NA FPE ZČU .....	5
1.1 SOUČASNÝ STAV .....	5
1.2 TYPY PEDAGOGICKÝCH PRAXÍ .....	5
1.2.1 Náslechové praxe .....	5
1.2.2 Výstupové praxe .....	7
1.2.3 Společenské praxe .....	8
1.3 ROLE VYSTUPUJÍCÍ V PROCESU PRAXÍ A JEJICH HIERARCHIE .....	9
1.3.1 Role u náslechových praxí .....	9
1.3.2 Role u výstupových praxí .....	9
1.4 VÝSTUPY ZE STUDENTSKÝCH PRAXÍ .....	10
1.5 PROBLÉMY SOUČASNÉHO SYSTÉMU .....	10
2 ZVOLENÉ TECHNOLOGIE .....	11
2.1 VÝBĚR FRAMEWORKU .....	11
2.2 FRAMEWORK LARAVEL .....	12
2.2.1 Základní informace .....	12
2.2.2 O frameworku .....	13
2.2.3 Systém pro správu šablon .....	15
3 POPIS ARCHITEKTURY VYTVOŘENÉ APLIKACE .....	16
3.1 PŘEDSTAVENÍ ZÁKLADNÍ ARCHITEKTURY .....	16
3.2 DATABÁZOVÝ MODEL .....	18
3.2.1 Systémové tabulky .....	18
3.2.2 Uživatelé a jejich role .....	19
3.2.3 Předměty .....	20
3.2.4 Rozvrhové akce .....	21
3.2.5 Reflexe z hodin .....	22
3.3 PŘEDSTAVENÍ VYBRANÝCH ČÁSTÍ KÓDU .....	23
3.3.1 Mechanismus ověřování přístupnosti URL adresy .....	23
3.3.2 Vytvoření databázové tabulky s uživateli .....	25
3.3.3 Ukázka práce se směrovacím systémem .....	26
3.3.4 Zpracování odeslaného formuláře .....	27
4 INSTALACE APLIKACE A JEJÍ ZPROVOZNĚNÍ NA SERVERU .....	28
4.1 MINIMÁLNÍ POŽADAVKY PRO BĚHOVÉ PROSTŘEDÍ .....	28
4.2 INSTALACE NA SERVER .....	29
5 POPIS UŽIVATELSKÉHO ROZHŘANÍ .....	31
5.1 ZÍSKÁNÍ UŽIVATELSKÉHO ÚČTU .....	31
5.2 ÚVODNÍ STRÁNKA A PŘIHLÁŠENÍ DO SYSTÉMU .....	32
5.2.1 Přihlašovací stránka .....	32
5.2.2 Nástěnka (dashboard) .....	32
5.3 ROZHŘANÍ ADMINISTRÁTORA .....	33
5.3.1 Seznam uživatelů .....	33
5.3.2 Vytvoření nového učitele .....	35
5.3.3 Vytvoření nového didaktika .....	36
5.3.4 Založení nového předmětu .....	36
5.3.5 Práce s rozvrhovými akcemi .....	37

---

5.3.6	Vytvoření rozvrhu pro následové praxe .....	40
5.3.7	Seznam (spolupracujících) škol.....	40
5.4	ROZHRANÍ DIDAKTIKA .....	41
5.4.1	Seznam dozorovaných studentů .....	41
5.4.2	Vyučovací hodiny u následové praxe .....	42
5.5	ROZHRANÍ UČITELE ZE ŠKOLY.....	43
5.5.1	Seznam vedených studentů.....	43
5.5.2	Moje vyučovací hodiny .....	43
5.6	ROZHRANÍ STUDENTA .....	44
5.6.1	Vytvoření vyučovacích hodin (výstupová praxe) .....	44
5.6.2	Vyplnění reflexe k předmětu .....	44
	ZÁVĚR.....	45
	RESUMÉ .....	46
	SEZNAM LITERATURY .....	47
	SEZNAM OBRÁZKŮ A TABULEK .....	49
	PŘÍLOHA 1 – DOCHÁZKOVÝ LIST NÁSLECHOVÝCH PRAXÍ .....	I
	PŘÍLOHA 2 – ZÁZNAM O BLOKOVÉ PEDAGOGICKÉ PRAXI .....	II
	PŘÍLOHA 3 – HOSPITAČNÍ ZÁZNAM HODINY A JEJÍ REFLEXE .....	III
	PŘÍLOHA 4 – DATABÁZOVÉ SCHÉMA.....	V

**SEZNAM ZKRATEK**

AJAX	Asynchronous JavaScript And XML
CSS	Cascading Style Sheets
DI	Dependency Injection
FPE	Fakulta pedagogická
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
ID	Identifier
JS	JavaScript
KČJ	Katedra českého jazyka a literatury
KHI	Katedra historie
KMT	Katedra matematiky, fyziky a technické výchovy
KNJ	Katedra německého jazyka
KVD	Katedra výpočetní a didaktické techniky
MVC	Model-view-controller
MVP	Minimum Viable Product
ORM	Object Relational Mapping
PHP	PHP: Hypertext preprocessor
ROAK	Rozvrhová akce
SPP	Středisko pedagogické praxe
SQL	Structured Query Language
URL	Uniform Resource Locator
ZČU	Západočeská univerzita

## Úvod

Pedagogické praxe patří neodmyslitelně ke studiu učitelství stejně jako studium vybraného teoretického základu. Na Pedagogické fakultě Západočeské univerzity v Plzni jsou tyto praxe zařazeny jak v bakalářském, tak i na navazujícím magisterském stupni.

V minulosti jsme se zcela jistě jako žáci základních škol setkali s budoucími pedagogy, kteří si přišli na naši školu vyzkoušet roli vyučujícího, která se pro mnohé z nich stala rolí celoživotní. Málokdo si už ovšem uvědomuje nakolik je problematika působení posluchačů pedagogických fakult na jednotlivých školách komplexní a kolik hodin přípravy a administrativních prací za tím stojí.

Hlavním cílem této práce bylo vytvořit elektronický webový systém pro vedení pedagogických praxí na základě informací získaných z pečlivé analýzy současného systému. Vedlejším cílem pak bylo vytvořit celý webový portál tak, aby byl univerzální, přehledný a v budoucnu, pokud možno snadno rozšiřitelný.

V první části této práce jsou sepsány poznatky zjištěné při analýze současného systému. Seznámíme se zde se základní kategorizací praxí, zainteresovanými stranami, celkovým fungováním systému a jeho výstupy. V závěru tohoto bloku je nástin některých problémů současného systému, které jsme se snažili v novém systému eliminovat.

Dalších několik částí je ryze technologických. Nachází se zde krátký popis procesu výběru softwarových nástrojů pro tvorbu systému, jejichž pečlivý výběr umožní tomuto systému se v následujících letech snadno a rychle rozšiřovat. Tato část plynule přechází v popis architektury vytvořené aplikace, která obsahuje jak seznámení s programovým kódem, tak i databázovou strukturou.

Poslední kapitoly obsahují podobný popis uživatelského rozhraní vytvořené aplikace a návod na jeho ovládání. Kapitoly jsou zde strukturovány do logických celků tak, aby popisovaly některé z činností, které budou uživatelé při práci se systémem provádět.



## 1 SYSTÉM PEDAGOGICKÝCH PRAXÍ NA FPE ZČU

### 1.1 SOUČASNÝ STAV

Systém pedagogických praxí na FPE ZČU v současné době prochází překotnými změnami, které jsou způsobeny tím, že zde v současnosti paralelně běží (a do skončení platnosti starých akreditací budou běžet) dva systémy praxí, což do velké míry ovlivňuje i vytvářený softwarový nástroj pro jejich organizaci.

Kromě stávajícího systému, který fungoval řadu let, zde nově začíná existovat i systém nový, obsahující inovované plány vzdělávání studentů na fakultě. Nově zavedený systém bude postupně nabíhat u všech oborů, na které se začnou uplatňovat pravidla z nové akreditace.

### 1.2 TYPY PEDAGOGICKÝCH PRAXÍ

Pedagogické praxe lze rozdělit do třech základních skupin. První skupinou jsou náslechové praxe, druhou výstupové a třetí skupinou jsou tzv. společenské praxe. Každá z těchto skupin je zaměřená na jiné aspekty a předává studentovi rozličné druhy informací a zkušeností.

#### 1.2.1 NÁSLECHOVÉ PRAXE

Náslechové praxe může student studovat jak v bakalářském, tak i v magisterském studijním programu (1). V současné době existuje velké množství různých předmětů, které se zabývají tímto druhem praxí. Zpravidla má každá katedra, nebo oddělení, svoje předměty, jejichž plnění kontroluje didaktik z dané katedry. Jejich náplň se totiž může navzájem odlišovat v závislosti studovaného oboru. Například studenti, kteří mají jako jednu z oborových kombinací obor výpočetní technika se v předmětu KVD/NASPN<sup>1</sup> zúčastní výuky zejména informatických předmětů.

Tyto praxe sestávají z několika vyučovacích hodin, při nichž student navštíví školská pracoviště (několik tříd), kde vystupuje v roli pozorovatele, který analyzuje probíhající výuku. Do hodin nijak nezasahuje. Vyučovací hodiny navštěvují menší skupinky studentů (cca. 10) a jsou celou dobu pod dozorem přiděleného didaktika. V případě, že by ve skupině bylo více žáků, tak dochází k rozdělení na menší skupiny. Není výjimkou, že praxe se konají v průběhu několika dní, při nichž studenti navštíví více škol a pozorují výuku v různých

---

<sup>1</sup>KVD: Náslechová praxe – navazující

třídách pod taktovkou pedagogických pracovníků z jednotlivých spolupracujících školských zařízení.

Pokud bychom generalizovali cíle náslechové praxe, tak jejich úkolem je ukázat studentům reálnou školní výuku související s oborem, který studují.

Následující ukázka pochází z anotace předmětu KVD/NASPN (2):

***Škola není jen teorie, ale také praxe!***

*V tomto předmětu vám bude umožněno nahlédnout do skutečných hodin zkušených pedagogů na příslušném typu školy, uvidíte aplikaci teorie v praxi, budete moci posoudit použité metody, klima třídy, znalosti studentů, osobnost učitele a další aspekty z celé řady faktorů vyučování.*

***Výhodou náslechové praxe je, že nemusíte vyučovat, ale jen pečlivě naslouchat! Díky tomu toho uvidíte a objevíte mnohem více.***

Svou účast na jednotlivých hodinách studenti stvrzují vlastnoručním podpisem (ukázka prezenčního listu je v příloze 1). Mimo toho mají povinnost ještě přiložit vyplněný hospitační dotazník a reflexi z hodin (y).

Hospitační dotazník slouží pro popis struktury vyučovací hodiny a jejích událostí. Student je tak donucen k aktivnímu sledování průběhu hodiny a průběžné tvorbě poznámek do výše zmíněného dotazníku. První částí reflexe je obsahová a metodická úroveň hodiny, druhou profesně a osobnostně kompetenční hledisko. Tyto vyplněné materiály pak student následně předává příslušnému didaktikovi. Ten zkontroluje jejich obsah a rozhodne o udělení zápočtu z předmětu. Učitel v tomto případě provádí klasickou výuku ve svých hodinách, aniž by ho ovlivňovaly náslechy studentů.

Pro ukázkou ještě několik předmětů spadajících do bloku náslechové praxe:

- KMT/NAPFN<sup>2</sup>
- KHI/NASPD<sup>3</sup>
- KVD/NASPN<sup>4</sup>

<sup>2</sup> KMT: Náslechová praxe – navazující (F)

<sup>3</sup> KHI: Náslechová praxe – navazující

<sup>4</sup> KVD: Náslechová praxe – navazující

### 1.2.2 VÝSTUPOVÉ PRAXE

Tento typ praxí zavádí do studia předměty, pro jejichž splnění se student musí aktivně zúčastnit výuky na školách pod vedením místních pedagogických pracovníků. Cílem je, aby si student osvojil roli učitele a získal cenné zkušenosti s organizací činnosti ve třídě a komunikaci s žáky. Postupně by tak mělo docházet k identifikaci studenta s rolí učitele. Během tohoto období je student osvobozen od ostatní výuky na fakultě.

Výstupové praxe se pak dělí na dvě podskupiny v závislosti na délce jejich trvání na krátkodobou a dlouhodobou.

Mezi předměty **krátkodobé výstupové praxe** patří například:

- KNJ/PEPZ<sup>5</sup>
- KČJ/PEZŠ<sup>6</sup>
- KVD/PRAXE<sup>7</sup>

**Dlouhodobou výstupovou praxí** reprezentuje předmět SPP/PPS<sup>8</sup>.

Pro výstupovou praxi se hledá učitel se stejnou nebo podobnou aprobačí jako student. O hodnocení výstupových praxí se stejně jako u náslechové praxe stará příslušný didaktik nebo středisko pedagogické praxe. Vyučovací hodiny jsou zpravidla rozloženy do dvou kalendářních týdnů.

U **krátkodobých** výstupových praxí je studentovi učitel přidělen.

Naopak u **dlouhodobých** výstupových praxí si student hledá učitele sám. Často tak vyučuje na škole v místě svého bydliště. Studentův výběr samozřejmě podléhá kontrole ze strany SPP. V případě, že student studuje učitelství pro 2. stupeň ZŠ nebo učitelství pro SŠ, tak absolvuje jednu hodinu náslechu a dalších 20 hodin samostatné výuky pro každý z aprobačních předmětů. Pokud je studentem učitelství pro 1. stupeň ZŠ, tak absolvuje 2 hodiny náslechu a odučí 40 hodin v určených třídách. V obou případech tedy dojde k absolvování 2 + 40 hodin na dané škole (3).

---

<sup>5</sup> KNJ: Pedagogická praxe

<sup>6</sup> KČJ: Výstupová praxe ZŠ

<sup>7</sup> KVD: Praxe informatiky

<sup>8</sup> SPP: Pedagogická praxe souvislá

### 1.2.3 SPOLEČENSKÉ PRAXE

Anotace postavení společenské praxe v portfoliu předmětů (4):

*Tento typ pedagogické praxe absolvují studenti učitelství na základě dobrovolnosti s cílem navázat na teoretickou přípravu, na znalosti získané v předmětech didaktika, pedagogika a psychologie. Při vedení zájmových útvarů ve vybraných organizacích (viz Seznam organizací, ve kterých je možné realizovat pedagogickou praxi v rámci mimoškolní práce s dětmi a mládeží – dále jen Seznam) si ověří vědomosti získané studiem teoretických předmětů.*

Jedná se o nově zavedený, nepovinný předmět. Rozvíjejí se v něm zejména studenti, kteří mají větší zájem o samotné učitelství a pracují tedy se studenty i nad rámec povinných předmětů. Ve velké míře se jedná o vedení kroužků v domech dětí a mládeže v blízkosti města Plzně (v blízkosti pedagogické fakulty) (5).

**Poznámka:** Jelikož se tento typ praxe diametrálně odlišuje od zbylých dvou typů praxí, rozhodli jsme se, že v první funkční verzi projektu (MVP) podporovaný nebude.

### 1.3 ROLE VYSTUPUJÍCÍ V PROCESU PRAXÍ A JEJICH HIERARCHIE

V celém systému vystupují 4 základní skupiny – studenti, didaktici, vyučující na školách a Středisko pedagogické praxe. Jak u následových, tak i u výstupových praxí mají všechny role víceméně obdobné funkce a pravomoci, až na učitele ze škol, jejichž role je u každého z těchto typů odlišná.

#### 1.3.1 ROLE U NÁSLECHOVÝCH PRAXÍ

Na začátku roku domlouvají **didaktici** z jednotlivých kateder následové praxe u učitelů z jednotlivých škol. Poté sestaví plán následových praxí, kde je napsáno, na jaké škole, u jakého učitele, kdy, v kolik hodin a v jaké třídě se budou konat hospitace. Tento plán poté zašlou **studentům**, kteří se hospitací zúčastní za doprovodu příslušného didaktika. Po proběhnutí následů studenti vypracují hospitační dotazníky a reflexi hodiny, kterou odevzdají příslušnému didaktikovi, který vše vyhodnotí a vyvodí z toho příslušné závěry.

**Učitelé ze škol** u následových praxí vystupují z pohledu systému pouze pasivně, jelikož se nevměšují do záležitostí univerzity, ale pouze provádějí klasickou výuku svých předmětů.

#### 1.3.2 ROLE U VÝSTUPOVÝCH PRAXÍ

U krátkodobých výstupových praxí je škola a konkrétní učitel studentovi přidělen (viz. Kapitola 1.2.2). U dlouhodobých výstupových praxí však musí student vystupovat aktivně již od samého počátku a praxi si dojednat s příslušným vyučujícím a poté si nechat svoji volbu potvrdit od SPP.

Student poté musí odevzdat seznam svých hodin didaktikovi, aby měl přehled o svých studentech a mohl na jejich hodinách řádně vykonávat hospitace. Bohužel má každý z didaktiků přiděleno mnoho studentů a není tak možné vykonat hospitaci u každého. Doposud tak byla volba návštěvy u jednotlivých studentů nahodilá.

Při průběhu praxe samotné se pak student stará o vyplňování záznamu o blokové pedagogické praxi (ukázka v příloze č. 2). Tento formulář je pak vodítkem pro didaktika, který podle něj rozhoduje o udělení zápočtu studentovi (nachází se na něm totiž i vyjádření vedoucího učitele).

## 1.4 VÝSTUPY ZE STUDENTSKÝCH PRAXÍ

Obsah této kapitoly slouží jako přehled rozdílnosti výstupů z obou hlavních systémů.

Náslechové praxe:

- Podepsaný docházkový list (příloha 1)
- Hospitační záznam hodiny a její reflexe (příloha 3)

Výstupové praxe:

- Záznam o blokové pedagogické praxi (příloha 2)
- Hospitační záznam hodiny a její reflexe (příloha 3)

U výstupových praxí vykonává hospitaci vyučující z dané školy.

## 1.5 PROBLÉMY SOUČASNÉHO SYSTÉMU

Největším problémem současného systému je jeho malá transparentnost a velké množství papírových podkladů, které kolují mezi zúčastněnými subjekty.

Nežádka kdy se stane, že si student domluví s učitelem na škole výstupovou praxi, ale ve stanoveném termínu na ní nenastoupí. Většinou to bývá způsobeno ukončením studia studentem, který pak již neinformuje učitele ze školy, že na praxi nenastoupí.

Kvůli uzavřenosti celého systému se zejména u výstupových praxí nedaří v konvergentním čase zjišťovat a promptně reagovat na problémy, které zde nastanou. Může se jednat například o nepříliš dobře spolupracujícího učitele ze školy nebo na nedostatečnou přípravu studenta na výuku ve třídě.

Dva výše uvedené problémy by měl pomoci odstranit elektronický systém pro vedení pedagogických praxí, který bude pomáhat těmto stavům předcházet.

## 2 ZVOLENÉ TECHNOLOGIE

Cílem této práce bylo vytvořit, v budoucnu snadno rozšiřitelnou, webovou aplikaci pro správu pedagogických praxí, do které bude snadné doimplementovat nové funkcionality dle aktuálních požadavků fakulty. Logicky tak padl výběr na programovací jazyk PHP, především kvůli tomu, že se jedná o zralý open source projekt, tudíž není problém s jeho využitím ve školství. Druhým důvodem je také jeho rozšířenost a možnosti, které poskytuje.

Co se velikosti samotného projektu týká, tak po základní analýze (kapitola 2) bylo zjištěno, že se jedná o středně velký softwarový projekt. Tomu se uzpůsobila kritéria pro výběr frameworku, ve kterém se bude celý systém vyvíjet.

### 2.1 VÝBĚR FRAMEWORKU

Vybíralo se celkem ze 4 frameworků, z nichž všechny se umístily na vysokých příčkách v hodnocení oblíbenosti z předešlých let (6). Výčet těchto frameworků se nachází v tabulce 1.

Tabulka 1: Základní informace o porovnávaných frameworkcích

Název frameworku	Webové stránky	Poslední verze*
<b>Laravel</b>	<a href="https://laravel.com/">https://laravel.com/</a>	24. 1. 2017 (5.4)
<b>Symfony</b>	<a href="https://symfony.com/">https://symfony.com/</a>	6 / 2017 (3.3)
<b>Nette</b>	<a href="https://nette.org/cs/">https://nette.org/cs/</a>	3. 5. 2017 (2.4)
<b>Yii</b>	<a href="https://www.yiiframework.com/">https://www.yiiframework.com/</a>	5. 6. 2017 (2.0.12)

\*Jedná se o poslední verzi v době výběru frameworku (červen 2017)

Prvním vyřazeným frameworkem bylo **Nette**, jelikož je všeobecně známé, že jeho dokumentace je poměrně špatně aktualizovaná a nacházejí se v ní mnohdy neaktuální informace (7).

**Symfony** je framework vytvořený agenturou SensioLabs<sup>9</sup>, který určuje současný směr vývoje ostatních frameworků, a ze kterého vychází například i v přehledu zmíněný framework Laravel. Jedná se však o komplexní nástroj, jehož využívání již vyžaduje některé pokročilejší znalosti programování. Z tohoto důvodu byl z výběru vyřazen, abychom zajistili,

<sup>9</sup> Kromě Symfony vytvořila například i systém pro správu šablon systém Twig

že v budoucnu bude pro kohokoliv snadné navázat na vývoj systému pro vedení praxí (8) (9).

V porovnání frameworků nakonec zvítězil **Laravel** – jednak proto, že se jedná o nejmladší framework mezi nejvýše umístěnými, ale i proto, že se tento nástroj poměrně rychle vyvíjí, má kvalitní dokumentaci a existuje rozsáhlá komunita, se kterou lze konzultovat problémy a architekturu vytvářené aplikace.

Naopak **Yii** bylo z úzkého kruhu vyřazeno zejména pro svou menší komunitu a pomalejší způsob vývoje (10).

## 2.2 FRAMEWORK LARAVEL

### 2.2.1 ZÁKLADNÍ INFORMACE

Laravel je moderní webový framework implementovaný v jazyce PHP, u jehož zrodu stál webový vývojář Taylor Otwell. První verze frameworku (Laravel 1) vyšla 6. 6. 2011 (11). V počátcích se Otwell věnoval vývoji frameworku při zaměstnání a až od počátku roku 2015 se jeho vývoji věnuje naplno. V současnosti se na full-time vývoji Laravelu podílí dva programátoři (12). Logo frameworku se nachází na obrázku 1.



Obrázek 1: Oficiální logo Laravelu

Anotace frameworku z jeho oficiálního GitHub repozitáře (13):

*Laravel is a web application framework with expressive, elegant syntax. We believe development must be an enjoyable and creative experience to be truly fulfilling. Laravel attempts to take the pain out of development by easing common tasks used in the majority of web projects.*

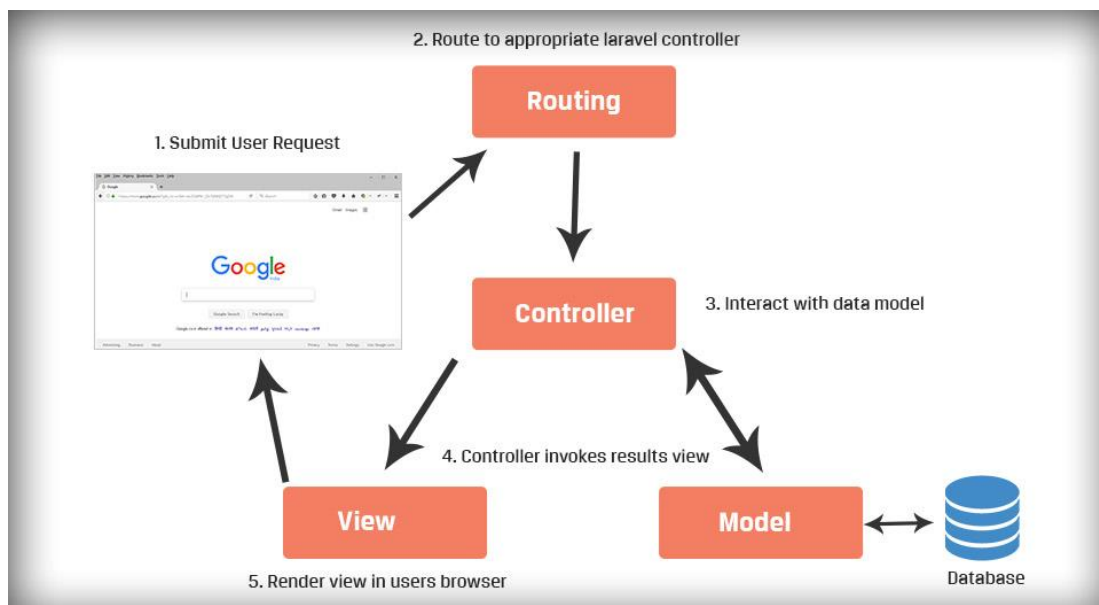


### 2.2.2 O FRAMEWORKU

Jedná se o MVC framework, umožňující rychlý vývoj webových aplikací a který vede vývojáře k dobrým návykům v oblasti návrhu software. Je dostupný všem zdarma jako open-source projekt na jehož vývoji se může podílet každý pomocí tzv. pull request prostřednictvím oficiálního GitHub<sup>10</sup> repozitáře (13).

Jednou z hlavních deviz patří zejména rozsáhlá mezinárodní komunita, se kterou lze komunikovat například přes komunikátor Slack v reálném čase a která je přívětivá k novým uživatelům frameworku. Co se technologií týče, tak Laravel do vínku dostal mj. mocné nástroje pro práci s databází, pokročilý směrovací systém (anglicky routing system), nativní podporu zabezpečení formulářů a příkazovou řádku Artisan.

Na obrázku 2 je zobrazeno základní schéma, jak Laravel reaguje a požadavky přicházející do aplikace a odkrývá základní schéma jeho fungování. Po odeslání požadavku (stisk tlačítka, odeslání formuláře, ...) dojde ke zpracování požadavku směrovacím systémem, který mj. pomocí middlewarů vyhodnocuje, zdali je daný uživatel oprávněný k požadované akci. Pokud jsou podmínky splněny, tak je požadavek postoupen příslušné metodě v controlleru, jenž získá data pomocí práce s modely a následně vkládá získaná data do pohledů (anglicky view), které jsou v podobě HTML stránky posílány zpět k uživateli.



Obrázek 2: Schéma principu práce aplikace (14)

<sup>10</sup> Portál pro správu verzí softwarových projektů využívající git

Velkou výhodou frameworku jsou nástroje pro práci s databází, které usnadňují celkový vývoj a umožňují programátorovi aplikace, aby v ní udržel pořádek, čímž je zajištěna případná snadná migrace projektu k jinému vývojáři. Laravel v sobě obsahuje tzv. migrační skripty (**migrate**), což je množina skriptů, které se spustí po zadání příkazu `php artisan migrate` **potážmo** `php artisan migrate:fresh`. Tyto skripty vytvoří kompletní databázovou strukturu nutnou pro běh aplikace – není tak potřeba přenášet soubory obsahující strukturu databáze mimo aplikaci. Další komponentou usnadňující práci s databází jsou tzv. **seedery**, které mají za úkol vložit do příslušných tabulek výchozí data (například vytvoří administrátorský účet nebo vytvoří seznam předmětů vyučovaných na FPE ZČU).

Výchozím způsobem, jak pracovat s vytvořenou databázovou strukturou je používání vestavěné komponenty **Eloquent ORM**. Pro každou databázovou tabulku je vytvořena třída, která se stará o interakce s danou tabulkou. Mezi těmito třídami vytvoříme relace, které nám umožňují snadno pracovat s daty z databáze bez vytváření klasických SQL dotazů. Při výběru pak můžeme využívat tzv. fluent syntax<sup>11</sup>, pomocí které můžeme jednotlivé příkazy řetězit za sebou.

Dalším pomocníkem je příkazová řádka **Artisan**, jejíž používání umožňuje tvůrci aplikace napůl automatizované provádění některých monotónních a často se opakujících operací. Umí například vytvářet skelety databázových migrací, modelů a tzv. request, čímž programátorovi odpadá značná část práce, kterou by musel vynaložit na psaní skeletů tříd, které jsou stále stejné a registraci těchto tříd do vnitřních konfiguračních souborů frameworku. Pokud je to nutné, tak si můžeme vytvořit i vlastní příkazy (15).

Za zmínku ještě stojí automatické vkládání závislostí, pomocí něhož můžeme vytvářet instance tříd, aniž bychom museli vytvářet instance některých objektů, se kterými třídy pracují (například služeb – services). Stačí pouze přidat odkaz na umístění třídy mezi jmenné prostory a požadovanou třídu jako parametr konstruktoru dané třídy. O vytvoření instance takové třídy se pak postará DI kontejner při volání kódu třídy.

---

<sup>11</sup> Fluent interfaces

### 2.2.3 SYSTÉM PRO SPRÁVU ŠABLON

Systém pro správu šablon (anglicky template engine) je systém, jenž slouží ke zvýšení přehlednosti webových aplikací oddělováním logických operací od pouhé prezentace dat. Jedná se vlastně o speciální jazyk pro tvorbu tzv. šablon, pomocí kterých zobrazujeme data. V MVC architektuře by to mělo být jediné přípustné místo pro psaní HTML kódu. Se samotným HTML kódem bychom si však nevystačili, proto tyto systémy obsahují ještě sadu speciálních konstrukcí, pomocí kterých vkládáme do šablon proměnné a řídíme tok dat (for, if, ...).

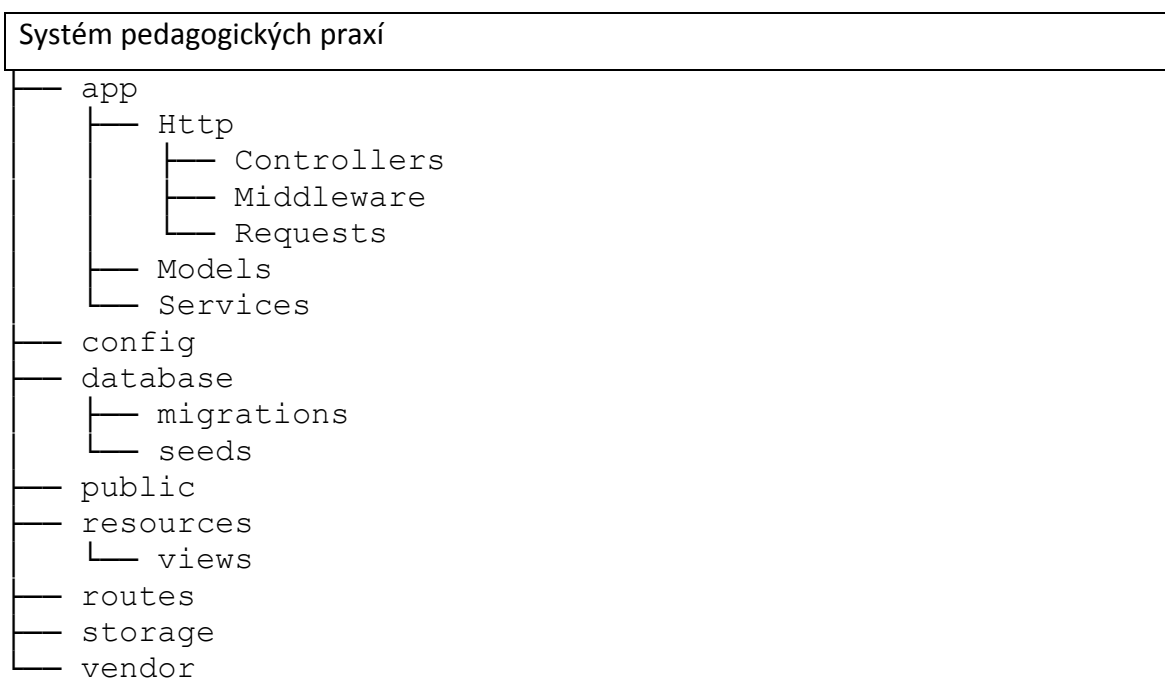
Výchozím systémem pro správu šablon pro Laravel je **Blade**, který ovšem kromě speciální syntaxe umožňuje do zdrojového kódu šablony vkládat i klasický PHP kód (16). Z tohoto důvodu jsem hledal alternativu, kterou se stal konkurenční systém **Twig**. Tento nástroj je vytvořený společností SensioLabs, obsahuje všechny potřebné konstrukce stejně jako Blade, ale jde ještě dál a nedovoluje do šablon žádným způsobem vkládat klasický PHP kód. Tím se docílí větší přehlednosti aplikace, protože některé operace by se v šablonách opravdu nacházet neměly (17).

### 3 POPIS ARCHITEKTURY VYTVOŘENÉ APLIKACE

Celý systém je vyvinutý za pomoci základní instalace Laravel frameworku, čemuž většinou odpovídá i adresářová struktura výsledné aplikace. Více informací o důležitých aspektech adresářové struktury jsou popsány v kapitole 3.2.3.

#### 3.1 PŘEDSTAVENÍ ZÁKLADNÍ ARCHITEKTURY

Na níže vyobrazeném schématu je zobrazena základní adresářová struktura webového systému pro správu pedagogických praxí. Z větší části kopíruje základní adresářovou strukturu frameworku Laravel, která se vytvoří automaticky při jeho instalaci (18). Jediným, na první pohled viditelným, rozdílem je existence složky *app/Models*, která v základu neexistuje, protože se počítá s tím, že modely budou ukládány rovnou ve složce *app*. Při větším množství modelů by se obsah této složky mohl stát nepřehledným.



Obrázek 3: Zjednodušené schéma adresářové struktury

Adresář **app** je jádrem vytvářeného systému. Složka **Controllers** slouží pro ukládání řídicích struktur aplikace, jejichž jednotlivé metody jsou volány prostřednictvím směrovače (neboli routeru<sup>12</sup>). Jednotlivé soubory ve složce **Middleware** slouží pro filtrování HTTP požadavků směrem do aplikace. V případě, že chceme zabezpečit některé URL pomocí vlastního algoritmu, tak toho docílíme přes námi vytvořený a zaregistrovaný middleware. Poslední

<sup>12</sup> soubor *routes/web.php*

důležitější složkou ve složce app je složka **Requests**, do níž jsou ukládány filtry na příchozí HTTP požadavky směrem do aplikace. Pomocí těchto souborů můžeme snadno ošetřit validaci formuláře na straně serveru bez psaní dlouhých podmínek. Tím zpřehledníme jednotlivé metody volané směrovacím systémem tak, aby se staraly pouze o výběr správných dat, a nikoliv o podružné funkce jako je například validace formulářů.

Ve složce **database** se nacházejí dva adresáře – složka migrations a složka seeds. Jejich význam byl už popsán v kapitole 2.2.2.

Adresář **public** slouží pro ukládání souborů, které mají být veřejně přístupné a u kterých nehrozí jejich zneužití. Tato složka je tedy vhodným místem pro ukládání obrázků a souborů s příponou js a css.

Složka **resources** může obsahovat jednak soubory pro lokalizaci do jiných jazyků, ale především obsahuje složku **views**, do které jsou ukládány jednotlivé šablony pro reprezentaci dat. Pro přehlednost jsou tyto šablony tříděny do dalších adresářů, z nichž každý zastupuje jeden z modulů systému.

Poslední v seznamu je složka **vendor**, která slouží k ukládání souborů externích knihoven, jež jsou stahovány pomocí utility **composer** (logo je na obrázku 4). Jedná se o manažera závislostí pro PHP, s jehož pomocí si zajistíme stahování potřebných balíčků a knihoven, které jsou nutné pro běh naší aplikace (19). Při instalaci si composer prohlédne soubor composer.json, kde jsou reference na externí knihovny, jenž posléze začne stahovat včetně jejich závislostí. Pokud by se některá závislost opakovala, tak se stáhne pouze jednou. Nakonec se vytvoří soubor **autoload.php**, který obsahuje reference na všechny externí knihovny.



Obrázek 4: Logo nástroje Composer

## 3.2 DATABÁZOVÝ MODEL

Srdcem celé aplikace je databázový model, který je navržen s ohledem na aktuální potřeby Pedagogické fakulty ZČU. Tvorba samotná byla velmi složitá, jelikož jsme se snažili brát zřetel na potřeby jednotlivých pracovišť.

Výsledná databáze je dobrým kompromisem mezi nároky kladenými zadáním práce, dodržováním dobrých zvyklostí při návrhu zdravých databází (zejména splnění normálních forem), její rozšiřitelností a poměrně snadnou použitelností.

Strukturu kompletního databázového modelu je možné spatřit v příloze 4. Jeho další popis se pak nachází v následujících kapitolách, kde je vše pro lepší přehlednost demonstrováno na částech zmiňovaného databázového modelu.

### 3.2.1 SYSTÉMOVÉ TABULKY

Mezi systémové tabulky patří tabulka **migrations**. K vytvoření této tabulky dojde při instalaci aplikace, konkrétně po zadání příkazu `php artisan:migrate`. Jedná se o pomocnou tabulku, která si uchovává informace o provedených migracích (vytvoření databázových tabulek).

Z porovnání výpisu proběhnutého migračního skriptu, který je k vidění na obrázku 10 a obsahu tabulky `migrations`, kterou můžeme spatřit na obrázku 5 je zřejmé, že se jedná o identické informace. Při zavádění nových tabulek tak framework pozná, odkud má pokračovat v instalaci.

id	migration	batch
1	2014_10_12_000000_create_users_table	1
2	2014_10_12_100000_create_password_resets_table	1
3	2018_01_07_182149_create_schools	1
4	2018_01_07_182550_create_teachers	1
5	2018_01_07_182845_create_logs	1
6	2018_01_07_183355_create_students	1
7	2018_01_07_183614_create_didactics	1
8	2018_01_12_190146_create_subjects	1
9	2018_01_12_190558_create_roaks	1
10	2018_01_12_191309_create_mn_roak_didactics	1
11	2018_01_12_192000_create_mn_roak_students	1
12	2018_03_26_152035_roak_hours	1
13	2018_04_04_181555_create_settings	1
14	2018_06_05_212459_create_reflection_questions_tabl...	1
15	2018_06_05_215423_create_mn_question_hours	1

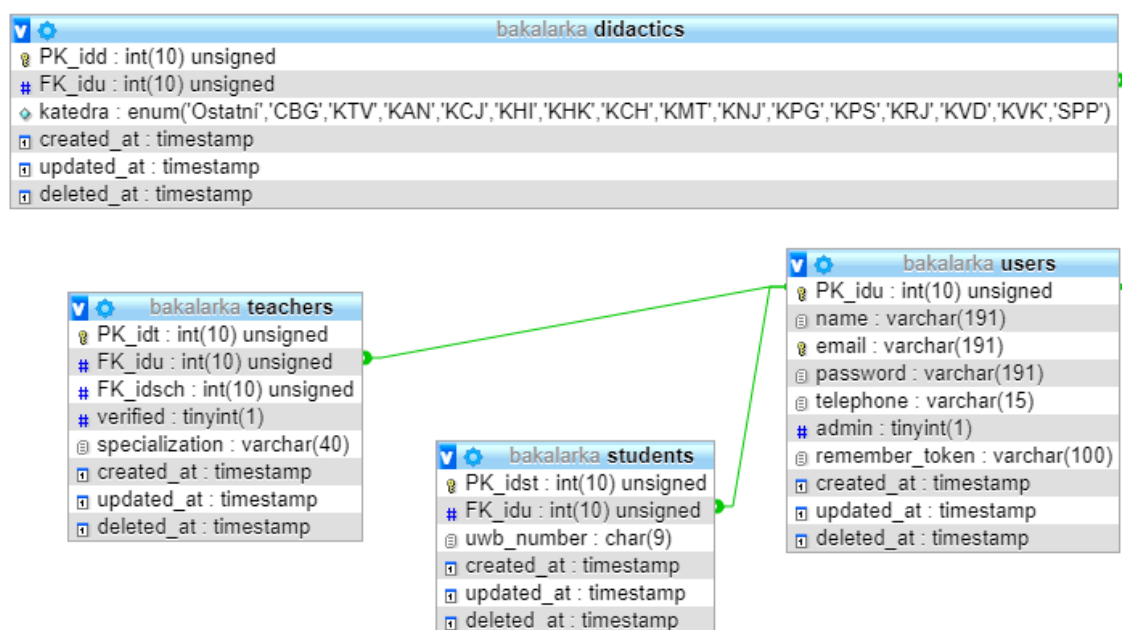
Obrázek 5: Obsah tabulky migrations

### 3.2.2 UŽIVATELÉ A JEJICH ROLE

Při analýze jsme zjistili, že každý uživatelský účet v systému může nabývat celkem čtyř unikátních rolí (kapitola 1.3):

- Administrátor
- Didaktik
- Učitel na škole
- Student

S ohledem na výše uvedené role je navrženo i jádro systému. Středem tohoto modelu je obecná tabulka `users`, která sdružuje obecné informace o všech uživateli. Na tuto tabulku se vážou další tabulky, z nichž každá jedna tabulka představuje jednu z uživatelských rolí (kromě role administrátora). Vztahy mezi tabulkou uživatelů a ostatními tabulkami si tedy lze dát do paralely s objektovým návrhem (specifičtější tabulky od tabulky `users` dědí – obrázek 6).



Obrázek 6: Databázové tabulky pro role uživatelů

Tento návrh je pro systém velmi výhodný a poskytuje zde možnost souběhu jednotlivých uživatelských rolí. Pokud by se například stalo, že osoba, která je didaktikem, je současně i učitelem na škole, tak to systém umožní a dle toho danému uživateli zobrazí přístupné portlety.

Jedinou rolí, pro kterou se v systému nenachází pomocná tabulka, je role **administrátora**. U administrátora není potřeba uchovávat žádná data navíc, tj. v tabulce `users` se nachází sloupec `is_admin`, který může nabývat pravdivostních hodnot.

Pro přiřazení uživatelské role didaktika slouží tabulka `didactics`, která obsahuje sloupec `katedra`, dle kterého určujeme působnost daného didaktika, například párujeme didaktika s předměty z jednotlivých kateder.

Tabulka `teachers` slouží pro přiřazení uživatelské role učitele ze školy, který je následně přiřazen k jedné ze škol, které jsou uloženy v tabulce `schools`. U učitele je dále z evidenčních důvodů sloupec pro ukládání textové informace o specializaci daného učitele. Tato informace poslouží zejména v období, kdy bude docházet k tvorbě nových rozvrhových akcí, abychom správné studenty dokázali snadno spárovat se správnými učiteli ze škol, aby se, pokud možno střetly jejich oborové kombinace.

Studenti jsou ukládáni do tabulky `students`. Zde je kromě údajů odkazujících na konkrétního uživatele (jméno, příjmení, ...) uchováváno i univerzitní číslo studenta, které pomůže didaktikům při dohledávání údajů o daném studentovi v aplikaci Portál ZČU například za účelem zapsání zápočtu do tohoto systému.

### 3.2.3 PŘEDMĚTY

K uchování dat o předmětech slouží databázová tabulka `subjects`. Záznamy o jednotlivých předmětech se do tabulky naplní jednou a následně už jsou jenom každý rok párovány k příslušné rozvrhové akci. Jelikož při změně parametrů předmětu není vhodné upravovat data u stávajícího předmětu, ke kterému již v minulosti byly párované rozvrhové akce, tak je nutné založit předmět nový. V opačném případě, při neodborném zásahu do dat v této tabulce by mohlo dojít k neočekávaným stavům u spárovaných rozvrhových akcí.

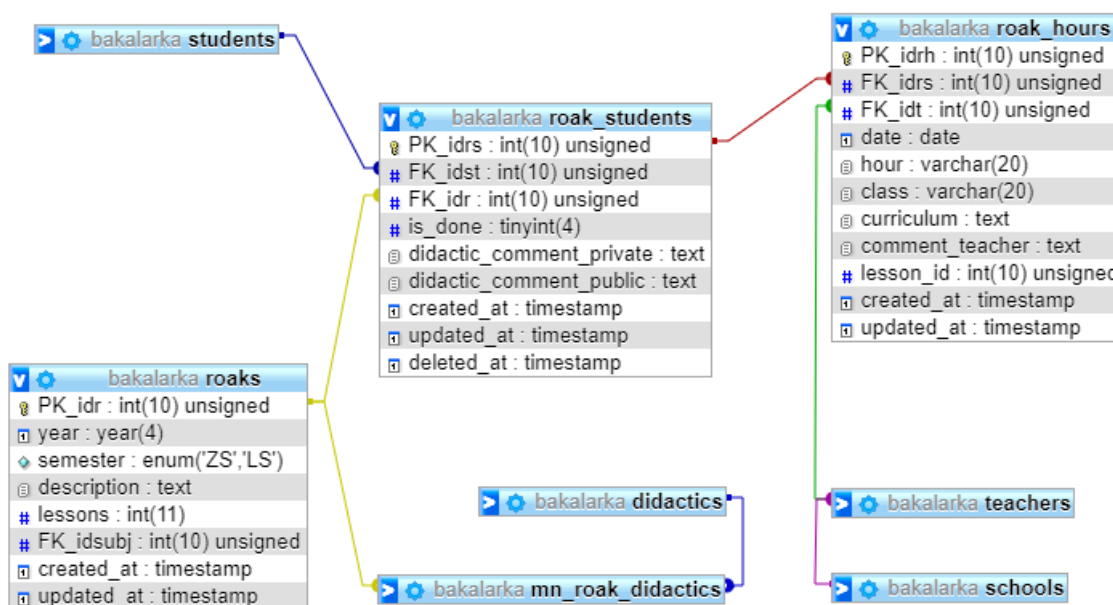


### 3.2.4 ROZVRHOVÉ AKCE

Každý další akademický rok v jednotlivých předmětech studují noví studenti. Z tohoto důvodu je vytvořena tabulka rozvrhových akcí – `roaks`, která reprezentuje instance jednotlivých předmětů pro konkrétní časové období. K vytvořeným rozvrhovým akcím se poté párují studenti.

Při vytváření záznamu v tabulce `roaks` dochází ke kopírování některých údajů z tabulky `subjects`, protože pokud by se meziročně změnil například počet hodin, tak by to mohlo v krajním případě znamenat nekonzistentnost v systému. Hodnota číslovky uložené ve sloupci `lessons` v tabulce `roaks` indikuje počet jednotlivých hodin pro každého studenta rozvrhové akce. V případě náslechovéch praxí tyto řádky generuje programově systém při vyplnění formuláře pro zadání rozvrhu. U výstupových praxí si student generuje hodiny samostatně, ale jejich počet taktéž koresponduje s výše zmíněnou hodnotou. Výhoda tohoto modelu je zřejmá – ke každé vyučovací hodině lze párovat jiného učitele. Toto je výhodné především u náslechovéch praxí, kdy je ke každé hodině většinou přiřazen jiný učitel.

K rozvrhové akci se také váže jeden nebo více didaktiků, kteří mají následně právo zobrazovat data příslušných studentů a kontrolovat jejich výstupy.



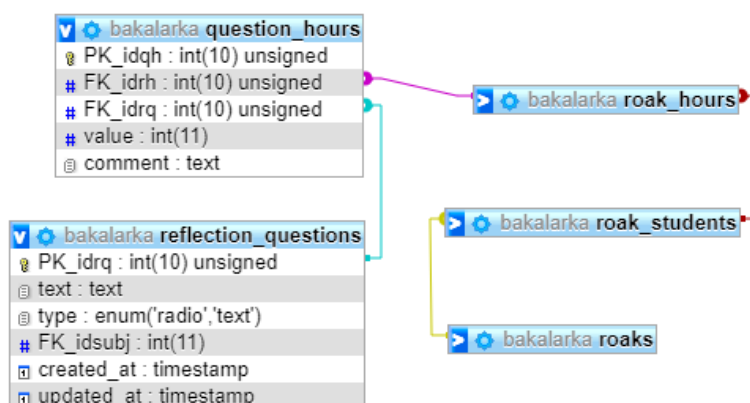
Obrázek 7: Organizace rozvrhových akcí

### 3.2.5 REFLEXE Z HODIN

Reflexe jsou reprezentovány jako souhrn otázek, jež jsou párovány ke konkrétní hodině příslušné rozvrhové akce. U následových praxí tyto otázky zodpovídá student, v opačném případě, jedná-li se o praxe výstupové, tak odpovědi ukládá vyučující ze školy.

Pro ukládání otázek slouží tabulka `reflection_questions` (obrázek 8). Pro zjednodušení databázového modelu je zde pospolu uložena jak část A (obsahová a metodická úroveň hodiny), tak i část B (profesně a osobnostně kompetenční hledisko). Odpovědi na první část těchto otázek jsou ukládány jako výběr ze čtyř možností a komentář. Na dotazy z části B se však odpovídá pouze komentářem (příloha 3).

Nevýhodou je tedy horší využití sloupce `value` z tabulky `question_hours`, která ukládá odpovědi na uložené otázky. Devizou tohoto návrhu je hlavně jeho jednoduchost zajištěná menším počtem databázových tabulek.



Obrázek 8: Databázové tabulky pro ukládání reflexí

### 3.3 PŘEDSTAVENÍ VYBRANÝCH ČÁSTÍ KÓDU

Následujících několik ukávek zdrojových kódů bylo pečlivě vybráno tak, aby zde byla představena některá specifika zvoleného frameworku a bylo zde tak popsáno, jak fungují a jak s nimi pracovat v případě úprav některých z částí systému.

#### 3.3.1 MECHANISMUS OVĚŘOVÁNÍ PŘÍSTUPNOSTI URL ADRESY

Základním zabezpečovacím mechanismem je restrikce přístupu k jednotlivým URL adresám. S tímto základním kódem dokážeme zabezpečit, aby se někdo nepokoušel například o odesílání formuláře, ke kterému nemá přístup, ale z dřívějšíka zná jeho strukturu a vstupní pole.

Zde bylo využito konstrukce zvoleného frameworku nazvané middleware. Ten lze použít pro nativní zabezpečení URL adres přímo ve směrovacím systému. Pro ověřování, zdali má, či nemá daný uživatel některou z rolí, byla vytvořena třída `CheckRole`.

Na níže uvedeném úryvku kódu ze směrovacího systému můžeme spatřit, jak je zabezpečena URL adresa `/subjects/list`. V tučně zvýrazněné řádce se nachází volání zabezpečovacího systému. Jako parametr se mu předává řetězec obsahující název<sup>13</sup> middleware následovaný dvojtečkou, za kterou může být několik parametrů oddělených čárkami.

```
<?php
    Route::group(['prefix' => 'subjects'], function() {
        Route::middleware('role:admin, didactic')->group(function() {
            Route::get('/list', 'SubjectController@getSubjects')
                ->name('subjects.list.get');
                // zde se nachází další zdrojový kód
        });
    });

    // zde se nachází další zdrojový kód
?>
```

Předávané parametry jsou poslány do metody `handle` ve třídě `CheckRole`. Každý middleware musí akceptovat přijetí minimálně dvou parametrů. První parametr slouží pro předávání dat, druhý pro odkaz na kód, který chceme spustit v případě splnění podmínek a v každém dalším se nachází parametry volání (v našem případě názvy rolí). K načtení více parametrů do pole slouží PHP konstrukce tří teček (`...$roles`).

<sup>13</sup> Zde se ovšem nepíše celý název třídy „`CheckRole`“, ale jen „`role`“. O převedení na název konkrétní třídy se postarají interní algoritmy frameworku.

Konkrétní implementace je v úryvku kódu níže. Algoritmus je nastaven tak, že výchozím stavem je zamítnutí přístupu. Poté následnou iterací vstupních rolí se rozhoduje o udělení přístupu. Pokud uživatel vlastní alespoň jednu roli, je mu přístup povolen.

```
<?php
namespace App\Http\Middleware;

use Illuminate\Support\Facades\Auth;
use App\Models\User;
use Closure;

class CheckRole
{
    /**
     * Handle an incoming request.
     *
     * @param  \Illuminate\Http\Request $request
     * @param  \Closure $next
     * @param  array $role
     * @return mixed
     */
    public function handle($request, Closure $next, ...$roles)
    {
        $validRoles = [
            'student',
            'teacher',
            'didactic',
            'admin',
        ];

        $statement = false;
        foreach ($roles as $role) {
            $isRole = 'is' . ucfirst($role);
            $statement |= (
                in_array($role, $validRoles) &&
                User::find(Auth::id())->{$isRole}
            );
        }

        if (!$statement) {
            return redirect()->route('index')->with(
                'warning',
                'Pro tuto operaci nemáte dostatečná oprávnění!'
            );
        }

        return $next($request);
    }
}
?>
```

### 3.3.2 VYTVOŘENÍ DATABÁZOVÉ TABULKY S UŽIVATELI

Pro vytváření databázových tabulek je využita technologie migrací, která byla blíže popsána v kapitole pojednávající o frameworku Laravel. V následujícím zdrojovém kódu je ukázka konkrétní implementace tabulky pro ukládání uživatelů.

Při instalaci se zavolá metoda `up` a v případě deinstalace dojde k zavolání příkazů v metodě `down`. Odinstalace se provádí v případě, že je migrace nahrazena v pozdější době novou migrací.

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('PK_idu');
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password');
            $table->string('telephone', 15)->nullable();
            $table->boolean('admin')->default(false);

            $table->rememberToken();
            $table->timestamps();
            $table->softDeletes();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
?>
```

### 3.3.3 UKÁZKA PRÁCE SE SMĚROVACÍM SYSTÉMEM

Směrovací neboli routovací systém je srdcem celé aplikace. Tento modul má na starosti správné směrování příchozích požadavků ke konkrétním metodám, které se starají o provedení jednotlivých funkcionalit.

Na níže uvedeném úryvku kódu se nachází obsah souboru `routes/web.php`, konkrétně pak část, která se stará o požadavky týkající se operací s předměty (`subjects`).

URL adresy je možné sdružovat do skupin, což je k vidění na první řádce. Při sjednocování do skupin lze předávat více parametrů, jako například název middlewaru, pro zabezpečení nebo prefixu dané URL. `Auth` je název middlewaru vygenerovaného Artisanem, který vpustí jenom autentizované uživatele.

Popis dalších konstrukcí směrovacího systému se nachází v tabulce 2.

Tabulka 2: Výběr konstrukcí směrovacího systému a jejich popis

Konstrukce	Popis funkce
<code>Route::get</code>	Zpracování HTTP požadavků typu GET (zobrazení dat).
<code>Route::post</code>	Zpracování HTTP požadavků typu POST (odeslání formuláře).
<code>Route::group</code>	Sdružování URL adres do skupin, pro aplikaci stejných pravidel.
<code>-&gt;name</code>	Vlastní pojmenování URL adresy (využívané ve Twigu).

V této ukázce není přesná kopie zdrojového kódu z programu z důvodu vyšší přehlednosti.

```
<?php
Route::group(['namespace' => 'Web', 'middleware' => 'auth'], function () {

    // subjects
    Route::group(['prefix' => 'subjects'], function() {
        Route::middleware('role:admin,didactic')->group(function() {
            Route::get('/edit/{id}', 'SubjectController@getEdit')
                ->name('subjects.edit.get.id');
            Route::post('/edit/{id}', 'SubjectController@postEdit')
                ->name('subjects.edit.post.id');
        });

        Route::middleware('role:teacher')->group(function() {
            Route::get(
                '/student/{studentId}',
                'RoakController@getStudentSubjects'
            )->name('subjects.student.get.id');
        });
    });
});
?>
```

### 3.3.4 ZPRACOVÁNÍ ODESLANÉHO FORMULÁŘE

Pro zpracovávání odeslaných formulářů máme v Laravelu připravenou velkou plejádu pomocných nástrojů a doporučených postupů (tzv. best practise).

Například na zpracování formuláře pro úpravu předmětů se podílejí následující soubory (v následujícím číslovaném pořadí):

1. resources/views/subjects/edit.twig
2. routes/web.php
3. app/Http/Requests/SubjectRequest.php
4. app/Http/Controllers/SubjectController.php

V šabloně (1) se nachází HTML formulář a u tagu `<form>` v atributu `action` se nachází jméno směrovacího pravidla a ID editovaného předmětu. Po odeslání dojde k dotazu do směrovacího systému (2), kde se o zpracování tohoto požadavku postará řádek:

```
Route::post('/edit/{id}', 'SubjectController@postEdit')
    ->name('subjects.edit.post.id');
```

Tento řádek nám říká, že danou URL má zpracovávat metoda `postEdit` ze třídy `SubjectController` (4) a kromě formulářových dat dojde ještě k předání ID upravovaného předmětu.

Metoda `postEdit` má dva parametry – prvním z nich je objekt obsahující formulářová data a druhý je ID předmětu, s nímž manipulujeme. `SubjectRequest` (3) je třída vytvořená na míru konkrétnímu formuláři starající se o validaci formuláře na straně serveru. V této metodě již provádíme zpracování formuláře – v tomto případě úpravu existujícího záznamu v databázi.

```
public function postEdit(SubjectRequest $request, int $subjectId)
{
    $subject = Subject::findOrFail($subjectId);

    $this->subjectService->createOrUpdateSubject($subject, $request);

    return redirect("/subjects/edit/{$subjectId}")->with(
        'success',
        "Předmět byl upraven."
    );
}
```

## 4 INSTALACE APLIKACE A JEJÍ ZPROVOZNĚNÍ NA SERVERU

Příloha k této bakalářské práci (webový portál) je implementovaná v Laravelu ve verzi 5.5, od něhož se určují minimální běhové požadavky v závislosti na extenzích a verzi PHP, který tato verze frameworku vyžaduje.


### 4.1 MINIMÁLNÍ POŽADAVKY PRO BĚHOVÉ PROSTŘEDÍ

Pro zprovoznění portálu musí být na Vašem severu k dispozici tyto komponenty:

- PHP o minimální verzi 7.0.0 s extenzemi:
  - OpenSSL PHP Extension
  - PDO PHP Extension
  - Mbstring PHP Extension
  - Tokenizer PHP Extension
  - XML PHP Extension
- Composer pro instalaci balíčků

Dostupnost těchto komponent snadno ověříme pomocí příkazu:

```
<?php
    phpinfo();
?>
```

PHP Version 7.2.2 	
System	Windows NT DELL 10.0 build 17134 (Windows 10) i586
Build Date	Jan 31 2018 19:27:55
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-snap-build\deps_aux\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\xampp72\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718,TS,VC15
PHP Extension Build	API20170718,TS,VC15

Obrázek 9: Ukázka části výpisu příkazem phpinfo

Na výše uvedeném obrázku je ukázka části nastavení PHP na stroji, kde byl systém vyvíjen (PHP 7.2.2). Aplikace by však měla být funkční i na nižší verzi PHP 7.0.0.



## 4.2 INSTALACE NA SERVER

5. Naklonování GIT repozitáře nebo rozbalení archivu obsahujícího aplikaci na webovém serveru s nainstalovaným balíčkem Apache2.
6. Instalace balíčků třetích stran pomocí příkazu `composer install --no-dev`
7. Konfigurace Virtual Hosts<sup>14</sup> pomocí následující syntaxe:

```
<VirtualHost *:80>
    ServerName bp.localhost
    DocumentRoot "{CESTA_K_PROJEKTU}/public"

    <Directory "{CESTA_K_PROJEKTU}/public">
        Options Indexes Multiviews FollowSymLinks
        RewriteEngine On

        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^ index.php [L]

        Require all granted
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

Dle vlastního serveru je nutné upravit řádky `ServerName` (URL adresa) a `Document Root` (cesta, do složky `public` projektu).

8. Restart webového serveru Apache. Na operačních systémech odvozených od distribuce Debian GNU/Linux lze použít následující dva příkazy. Prvním z nich provedeme kontrolu správnosti konfigurace a druhým se postaráme o restart webového serveru, při kterém dojde k načtení nové konfigurace.

```
apache2ctl -t
service apache2 restart
```

9. Přejmenování souboru `.env.example`, který se nachází v hlavním adresáři projektu na `.env` a změnit v něm hodnoty příslušných klíčů. Zejména půjde o nastavení připojení na databázový a e-mailový server.
10. Vygenerování aplikačního klíče pro Laravel příkazem:

```
php artisan key:generate
```

Hodnota vygenerovaná tímto příkazem by se měla objevit jako hodnota klíče `APP_KEY` v souboru `.env`.

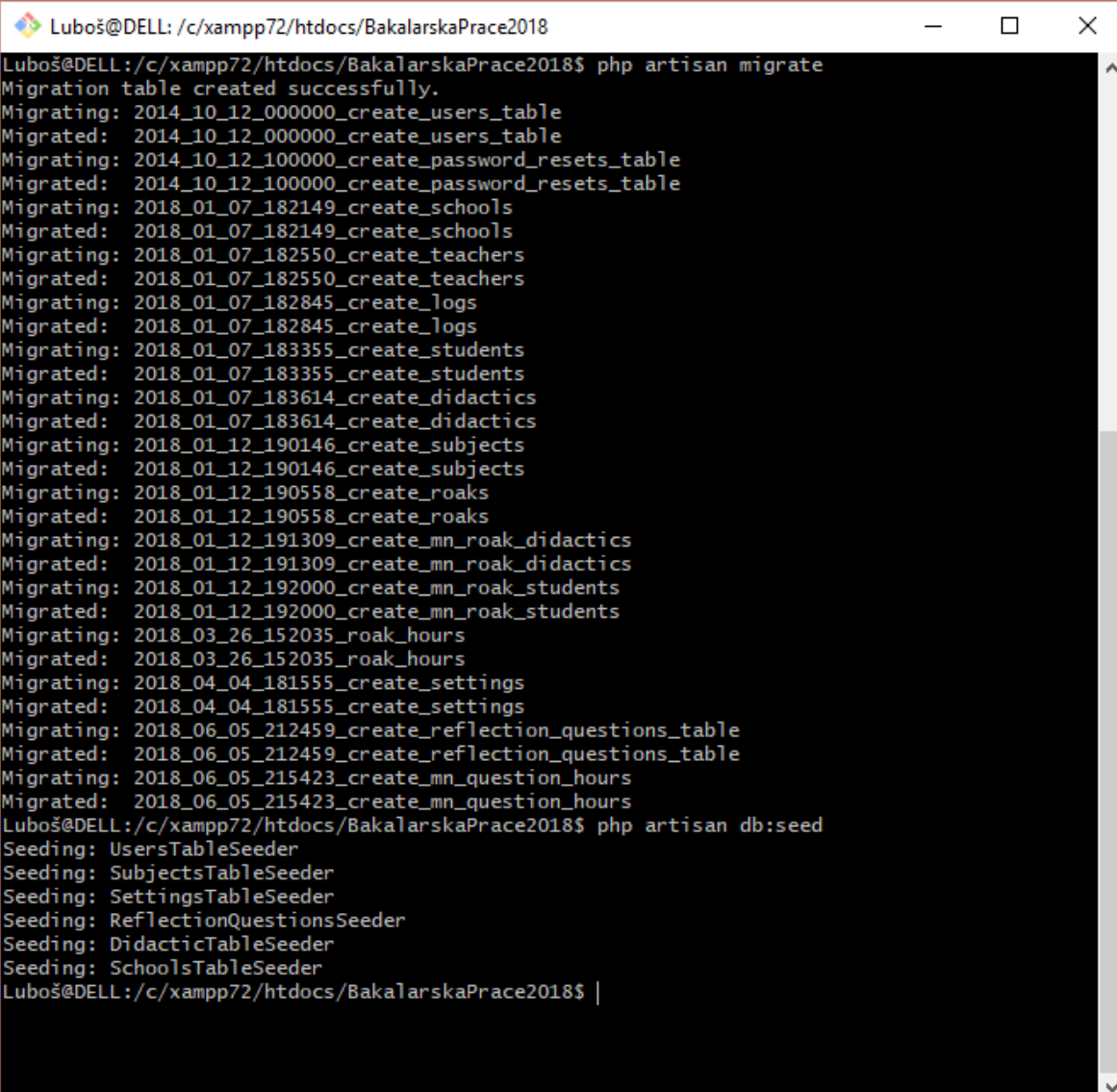
11. Založení databáze s porovnáním `utf8mb4_unicode_ci`.
12. Provedení výchozí inicializace databáze (vytvoření databázových tabulek a naplnění tabulek základními daty):

```
php artisan migrate
```

---

<sup>14</sup> Virtual hosts nám umožní definovat několik virtuálních strojů na jednom fyzickém stroji. Je zadán neplatný pramen.

```
php artisan db:seed
```



```
Luboš@DELL: /c/xampp72/htdocs/BakalarskaPrace2018
Luboš@DELL:/c/xampp72/htdocs/BakalarskaPrace2018$ php artisan migrate
Migration table created successfully.
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table
Migrating: 2018_01_07_182149_create_schools
Migrated: 2018_01_07_182149_create_schools
Migrating: 2018_01_07_182550_create_teachers
Migrated: 2018_01_07_182550_create_teachers
Migrating: 2018_01_07_182845_create_logs
Migrated: 2018_01_07_182845_create_logs
Migrating: 2018_01_07_183355_create_students
Migrated: 2018_01_07_183355_create_students
Migrating: 2018_01_07_183614_create_didactics
Migrated: 2018_01_07_183614_create_didactics
Migrating: 2018_01_12_190146_create_subjects
Migrated: 2018_01_12_190146_create_subjects
Migrating: 2018_01_12_190558_create_roaks
Migrated: 2018_01_12_190558_create_roaks
Migrating: 2018_01_12_191309_create_mn_roak_didactics
Migrated: 2018_01_12_191309_create_mn_roak_didactics
Migrating: 2018_01_12_192000_create_mn_roak_students
Migrated: 2018_01_12_192000_create_mn_roak_students
Migrating: 2018_03_26_152035_roak_hours
Migrated: 2018_03_26_152035_roak_hours
Migrating: 2018_04_04_181555_create_settings
Migrated: 2018_04_04_181555_create_settings
Migrating: 2018_06_05_212459_create_reflection_questions_table
Migrated: 2018_06_05_212459_create_reflection_questions_table
Migrating: 2018_06_05_215423_create_mn_question_hours
Migrated: 2018_06_05_215423_create_mn_question_hours
Luboš@DELL:/c/xampp72/htdocs/BakalarskaPrace2018$ php artisan db:seed
Seeding: UsersTableSeeder
Seeding: SubjectsTableSeeder
Seeding: SettingsTableSeeder
Seeding: ReflectionQuestionsSeeder
Seeding: DidacticTableSeeder
Seeding: SchoolsTableSeeder
Luboš@DELL:/c/xampp72/htdocs/BakalarskaPrace2018$ |
```

Obrázek 10: Ilustrační výpis vytvoření a naplnění tabulek

Po úspěšném provedení všech předchozích kroků by vše mělo fungovat a v prohlížeči by se po zadání příslušné URL měla zobrazit úvodní stránka vytvořené webové aplikace.

## 5 POPIS UŽIVATELSKÉHO ROZHRANÍ

Následující kapitola může posloužit jako uživatelská příručka pro všechny uživatele systému (administrátor, didaktik, učitel ze školy a student).

Pro kompletní pochopení všech funkcionalit popsaných v této kapitole je nutné přečtení kapitoly č. 1, která vysvětluje systém fungování pedagogických praxí.

Celé uživatelské rozhraní je koncipováno tak, aby bylo přehledné a snadno se používalo i méně zdatným uživatelům počítačů. Samozřejmostí je i responsivní návrh, který zajistí adekvátní zobrazení i na mobilních telefonech a tabletech.

### 5.1 ZÍSKÁNÍ UŽIVATELSKÉHO ÚČTU

Uživatelský účet lze, v tomto okamžiku, získat několika způsoby. Prvním způsobem je být na seznamu importovaných studentů k dané rozvrhové akci. Tento import provádí vždy na začátku školního roku SPP stažením seznamu studentů k danému předmětu. Při tomto importu dojde k vytvoření (pokud již neexistuje) nového uživatelského účtu. Heslo ke každému účtu je náhodně generováno a je zasláno, v případě, že jde o studenta, na jeho studentský e-mail. Dalším ze způsobů je se registrovat samostatně a počkat na přidělení uživatelských rolí od střediska pedagogické praxe nebo některého z administrátorů. Poslední způsob je využíván při zakládání učitelů ze škol. Ti se neregistrují, ani nejsou odnikud importováni, ale vytvářejí jim uživatelské účty přímo administrátoři (v případě náslechoových praxí) nebo studenti (u výstupových praxí) a těm taktéž chodí heslo na e-mail, který je k účtu zadaný.

## 5.2 ÚVODNÍ STRÁNKA A PŘIHLÁŠENÍ DO SYSTÉMU

V následujících podkapitolách jsou popsány sekce, které jsou společné všem uživatelům systému (tj. jejich využívání není ovlivněno přidělenými uživatelskými rolemi).

### 5.2.1 PŘIHLAŠOVACÍ STRÁNKA

Přihlašovací stránka je graficky pojata tak, aby korespondovala se zavedenou univerzitní identitou a mohla tak být všemi příchozími návštěvníky, pokud možno, považována za důvěryhodnou.

Pedagogické praxe

Přihlásit se

Zaregistrovat se

Přihlášení do systému

Uživatelé Západočeské univerzity se přihlašují svým univerzitním Orion účtem přes WebAuth. Klikněte na obrázek přihlašovací stránky WebAuthu.

**Orion WebAuth**

Orion login:

Heslo:

[Nápověda](#) | [Nechci se přihlásit](#)

**Kde to jsem? Kam jsem se to zase dostal?**  
 Webový server, na který se snažíte přihlásit, byl zařazen do domény jednotného přihlášení (single sign-on, SSO), a vyžaduje ověření vaší identity platným uživatelským jménem a heslem. Stránka, na které se právě nacházíte, je vstupním bodem k webovým serverům ZČU zařazeným pod systém jednotného přihlašování. To znamená, že svoji identitu prokážete skrze tuto stránku prvnímu serveru, na který chcete přistupovat, a tím jste automaticky přihlášen(a) i k ostatním serverům v doméně.

**Výhody**  
 Větší pohodlí pro uživatele (heslo zadávají jen jednou) a technicky vyšší bezpečnost: mezi prohlížečem a webovým serverem se neposílá heslo, ale jen autentizační token. Platnost tokenu je navíc časově omezena.

**Důležité upozornění**  
 Nikdy nezadávejte Orion jméno a heslo do webových formulářů, pokud se nejedná o přihlašovací stránku systému Webauth (tato stránka) nebo oficiální webový nástroj pro změnu hesla. Obě aplikace jsou provozovány na stroji webkdc.zcu.cz!!!

V jiném případě se můžete přihlásit přes tento formulář:

Přihlašovací e-mail

Obrázek 11: Přihlašovací stránka do systému

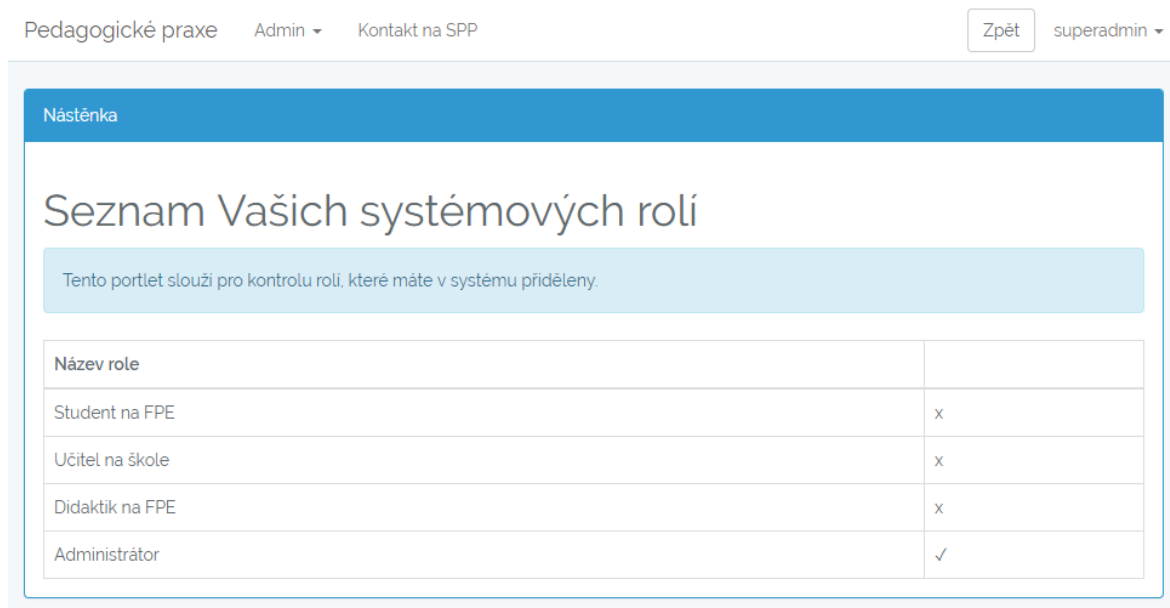
Přihlášení do systému je realizováno pomocí e-mailu a hesla. V případě, že uživatel zapomene heslo pro přihlášení, tak má možnost si vygenerovat nové heslo prostřednictvím sekce zapomenuté heslo.

### 5.2.2 NÁSTĚNKA (DASHBOARD)

Po úspěšném přihlášení je každý uživatel přesměrován na stránku s nástěnkou<sup>15</sup> (obrázek 12), kde je zobrazen seznam všech rolí, jež jsou vázané s daným uživatelským účtem. Toto

<sup>15</sup> Často se pro označení nástěnky používá anglického výrazu dashboard

lze použít pro rychlou kontrolu, zdali mám přidělené správně všechny uživatelské role, které bych měl mít nebo si mám o jejich přidělení případně napsat SPP.



Obrázek 12: Nástěnka portálu

### 5.3 ROZHRAŇÍ ADMINISTRÁTORA

Další podkapitoly popisují ovládání rozhraní, které je přístupné pouze administrátorům systému, mezi které patří zejména SPP.

#### 5.3.1 SEZNAM UŽIVATELŮ

Tato sekce poskytuje stránkovaný seznam všech uživatelů, ve kterém se dá vyhledávat pomocí dvou základních filtrů – dle role, jména nebo kombinace. U vyfiltrovaných uživatelů lze přejít do detailu uživatele (obrázek 14), nebo se za daného uživatele přihlásit a používat tak systém jeho jménem. Druhá z nabízených možností slouží zejména pro řešení problémů s jednotlivými uživateli. Například v případech, kdy nefunguje některá z funkcionalit, která by měla fungovat. Snadno tak lze tuto skutečnost ověřit a postoupit její řešení například systémovému administrátorovi. Pokud jsme takto přihlášení za některého uživatele, tak se v pravé části horní lišty zobrazí červené tlačítko s nápisem „Zpět do administrace“. Po stisku tohoto tlačítka proběhne přepnutí uživatele na původní účet, ze kterého proběhlo administrátorské přihlášení.

Pedagogické praxe Admin Kontakt na SPP Zpět superadmin

Seznam všech uživatelů evidovaných v systému

Vyberte roli:

Zadejte jméno:

Jméno	Telefon	E-mail	Role	
superadmin		admin@zcu.cz	admin	<input type="button" value="Detail uživatele"/> <input type="button" value="Přihlásit se za"/>
Ing. Jan Novák, Ph.D.		didaktik1@kvd.zcu.cz	didaktik, admin	<input type="button" value="Detail uživatele"/> <input type="button" value="Přihlásit se za"/>
Mgr. Jana Nováková		didaktik2@khi.zcu.cz	didaktik	<input type="button" value="Detail uživatele"/> <input type="button" value="Přihlásit se za"/>

Obrázek 13: Seznam uživatelů

Na obrázku 14 níže je zobrazený náhled sekce „Detail uživatele“, ve kterém lze ručně přiřazovat nebo odebírat uživatelům jejich jednotlivé role. S každou zde zobrazenou rolí se pojí jeden portlet. Jsou zde všechny role s výjimkou administrátora, o jehož přidělení se postará systémový administrátor prostřednictvím ruční úpravy této volby v databázi. Portlet ke zvolené roli se zobrazí (skryje) po zaškrtnutí (odškrtnutí) příslušného pole. Každý z takto zobrazených portletů disponuje konkrétním obsahem v závislosti na parametrech, vztahujících se k rolím.

Pedagogické praxe Admin Kontakt na SPP Zpět superadmin

Detail uživatele Ing. Jan Novák, Ph.D.

Jméno: Ing. Jan Novák, Ph.D.

E-mail: didaktik1@kvd.zcu.cz

Telefon:

Student  Učitel  Didaktik

Uživatelská role didaktik

Pracoviště:

Obrázek 14: Ruční úprava uživatele – formulář

### 5.3.2 VYTVOŘENÍ NOVÉHO UČITELE

Formulář pro vytvoření učitele je přístupný jak administrátorům, tak i studentům. V případě, že jde o náslechovou praxi, tak se o tvorbu rozvrhu k dané ROAK stará SPP, což znamená, že středisko páruje konkrétní vyučující ze škol ke konkrétním vyučovacím hodinám náslechoových praxí.

Pokud se ovšem jedná o dlouhodobou výstupovou praxi, vytváří si plán vyučovacích hodin každý student samostatně. Zde je taktéž ke každé vyučovací hodině párován jednotlivý učitel. Z tohoto hlediska je podstatné, aby měl student možnost vytvářet uživatelské účty pro učitele ze škol.

Vytvoření učitele je poměrně snadné. Stačí vyplnit všechny pole ve formuláři zobrazeném na obrázku 15 a vybrat příslušnou školu, kde učitel působí z výběrového pole. V případě, že FPE se školou, kde působí právě přidávaný učitel, doposud nespolečně pracovala, tak je nutné danou školu založit kliknutím na tlačítko „Škola není v seznamu“. Po zvolení této možnosti dojde k zobrazení dalších formulářových polí pro vyplnění základních údajů o škole (především kontaktní údaje na školu).

The screenshot shows a web interface for creating a new teacher. At the top, there are navigation links: 'Pedagogické praxe', 'Admin', and 'Kontakt na SPP'. On the right, there are buttons for 'Zpět' and 'superadmin'. The main form is titled 'Vytvořit nového učitele' and contains the following fields:

- Jméno učitele:** Text input containing 'Mgr. Jiří Dvořák'.
- E-mail učitele:** Empty text input.
- Telefon na učitele:** Empty text input.
- Oborová kombinace:** Text input with a placeholder '(Zadávejte oddělené pomlčkami - příklad: INF-CHE)'.
- Zvolte školu:** A dropdown menu is open, showing 'Zvolte' (selected), 'Plzeň', and '10. základní škola Plzeň Masarykovo gymnázium Plzeň'. A button 'Škola není v seznamu' is next to it.

A green button labeled 'Vytvořit učitele' is located at the bottom left of the form.

Obrázek 15: Formulář pro založení nového učitele

### 5.3.3 VYTVOŘENÍ NOVÉHO DIDAKTIKA

Stránka s formulářem pro vytváření nových didaktiků je přístupná pouze uživatelům s rolí administrátora.

Po vyplnění formuláře a jeho následném odeslání dojde k vytvoření didaktika. Pokud se ovšem didaktik přihlásí, tak se mu nezobrazí téměř nic, alespoň do doby, dokud nebude napojen na konkrétní rozvrhové akce. Ukázka formuláře s vyplněnými fiktivními daty je na obrázku 16.

The screenshot shows a web interface for creating a new didactic unit. At the top, there are navigation links: 'Pedagogické praxe', 'Admin', and 'Kontakt na SPP'. On the right, there is a 'Zpět' button and a user profile 'superadmin'. The main content area is titled 'Vytvořit nového didaktika' and contains a form with the following fields:

- Jméno:** Mgr. Lubomir Svoboda
- E-mail:** lubomir.svoboda@knj.zcu.cz
- Telefon:** 222333999
- Pracoviště:** KNJ

At the bottom of the form is a green button labeled 'Vytvořit didaktika'.

Obrázek 16: Formulář pro založení nových didaktiků

### 5.3.4 ZALOŽENÍ NOVÉHO PŘEDMĚTU

Formulář pro vytváření nových předmětů nalezneme jako pátou položku administrátorského menu. Dle zde nastavených parametrů poté ovlivníme, jak se bude předmět chovat při vytváření rozvrhových akcí.

Kromě základních atributů jako je název předmětu a jeho popis (obrázek 17), což jsou data získaná z aplikace Portál ZČU (konkrétně ze sylabu určitého předmětu), zde vyplňujeme i počet vyučovacích hodin, což je atribut, který později bude rozhodovat o počtu vytvářených hodin pro každého studenta zvolené rozvrhové akce. Pole výchozí didaktik slouží pro vybrání jednoho didaktika, jehož bude aplikace párovat s vytvářenými rozvrhovými akcemi tohoto předmětu. Ve výběrovém poli typ praxe máme na výběr mezi náslechovou a výstupovou praxí, což je volba ovlivňující především kdo bude vybírat učitele zvolených hodin.



Zaškrťovací pole „učitele vyplňuje student“ je vhodné zaškrtnout u předmětů, kde si studenti hledají vyučujícího pro praxi samostatně. Z toho vyplývá, že je vhodné toto pole zaškrtnout, pokud se jedná o dlouhodobou výstupovou praxi.

Pedagogické praxe Admin ▾ Kontakt na SPP Zpět superadmin ▾

**Vytvořit nový předmět**

Název předmětu:

Popisek předmětu:

Počet vyučovacích hodin:

Odkaz na Courseware:   
(Zadávejte včetně protokolu - http:// nebo https://)

Zvolte semestr:

Výchozí didaktik:   
(Hlavní didaktik této ROAK)

Typ praxe:

Je povinná reflexe:  (U každé hodiny se zobrazí formulář pro zadání reflexe - následovně pro studenty a výstupově pro učitele)

Učitele vyplňuje student:  (Pokud je zaškrtnuto, tak učitele k dané hodině vyplňuje příslušný student)

Předmět je aktivní:  (Pokud je předmět aktivní lze pro něj vytvořit rozvrhovou akci)

Obrázek 17: Formulář pro založení nového předmětu

Všechny vytvořené předměty pak lze nalézt v sekci „Seznam předmětů“ odkud je možné přejít k jejich úpravě.

### 5.3.5 PRÁCE S ROZVRHOVÝMI AKCEMI

Na začátku každého akademického roku je potřeba vytvořit rozvrhové akce a propojit je s konkrétními studenty, didaktiky a vyučujícími ze škol.

Prvním krokem je výběr předmětu z roletky obsahující seznam všech aktivních předmětů. Poté pokračujte kliknutím na tlačítko „Načíst informace“, které načte výchozí informace

o předmětu do následujících polí. Je však možné si tyto údaje dále přizpůsobit, když se budou lišit od výchozích údajů zapsaných u předmětu.

Po vyplnění formuláře máme na výběr dvě možnosti. První (zelené) tlačítko vytvoří rozvrhovou akci se zadanými parametry a přesměruje uživatele na jinou stránku, kde se nachází formulář pro import seznamu studentů. Druhé (modré) tlačítko vytvoří rozvrhovou akci a vrátí uživatele do téhož formuláře, kde může pokračovat zadáváním dalších rozvrhových akcí (obrázek 18).

The screenshot shows a web interface for creating a course schedule (ROAK). At the top, there are navigation links: 'Pedagogické praxe', 'Admin', and 'Kontakt na SPP'. On the right, there is a 'Zpět' button and a user profile 'superadmin'. The main form is titled 'Založit rozvrhovou akci (ROAK)' and contains the following fields:

- Název předmětu:** A dropdown menu with 'KVD/NASPN' selected and a 'Načíst informace' button.
- Popisek:** A text area containing 'Náslechová praxe - navazující'.
- Počet vyučovacích hodin:** A text input field with the value '5'.
- Zadejte začátek ŠR:** A text input field with '2018' and a note 'Př. pro ŠR 2018/2019 zadejte 2018'.
- Zvolte semestr:** A dropdown menu with 'Zimní semestr' selected.
- Zvolte didaktiky:** A list box showing 'KVD', 'KHI', and 'KNJ'. The 'KVD' option is expanded to show a list of didactics: 'Ing. Jan Novák, Ph.D.', 'Mgr. Jana Nováková', and 'Mgr. Lubomír Svoboda'. A note below says 'Více didaktiků zvolíte držením klávesy CTRL a kliknutím myši na příslušné osoby.'

At the bottom of the form, there are two buttons: 'Vytvořit ROAK (a import studentů)' (green) and 'Vytvořit ROAK (a další nová)' (blue).

Obrázek 18: Formulář pro vytvoření rozvrhové akce

Dalším krokem je vložení seznamu studentů k vytvořeným rozvrhovým akcím. Do sekce pro vložení seznamu studentů se dostaneme buď stiskem zeleného tlačítka ve formuláři výše, nebo ze sekce se seznamem rozvrhových akcí (obrázek 19). U každého předmětu na této stránce je tlačítko označené „Studenti“ jehož zbarvení nám napoví, zda jsou u předmětu uvedeni studující (zelená barva tlačítka) nebo nikoliv (červená barva tlačítka).

Pedagogické praxe Admin Kontakt na SPP Zpět superadmin

Seznam všech rozvrhových akcí Založit novou ROAK

Zadejte rok:

Zvolte semestr:

Vyberte předmět:

[Filtrovat rozvrhové akce](#)

ID	Název předmětu	Studentů	Rok (semestr)	Dotace	
1	KVD/NASPN	0	ZS 2018	5 hodin	<a href="#" style="background-color: #3498db; color: white; padding: 2px 5px;">Upravit</a> <a href="#" style="background-color: #e74c3c; color: white; padding: 2px 5px;">Studenti</a> <a href="#" style="background-color: #f1c40f; color: #34495e; padding: 2px 5px;">Vyučovací hodiny</a>
2	SPP/PPS	0	LS 2018	2 hodin	<a href="#" style="background-color: #3498db; color: white; padding: 2px 5px;">Upravit</a> <a href="#" style="background-color: #e74c3c; color: white; padding: 2px 5px;">Studenti</a>

Obrázek 19: Seznam rozvrhových akcí

Importovat studenty lze pomocí přesně strukturovaného souboru s příponou csv, který lze získat pouze v aplikaci Portál ZČU v detailu příslušné rozvrhové akce – zvýrazněná ikonka pro stažení souboru je na obrázku 20. Pro usnadnění se zde také nachází odkaz pro přímé stažení souboru z portálu.

**Předmět** Tisk/export:

Hledání předmětu podle zadaných parametrů

Zkratka pracoviště:  Zkratka předmětu:  Název:  Rok:

Vyučováno v jazyce:  Způsob zakončení:  Interní akreditace:  Fakulta:

[URL na seznam](#)

Zkratka  Název  Varianta

SPP / PPS  Pedagogická praxe souvislá  2018/2019

Obrázek 20: Získání datového CSV souboru se studenty

Po úspěšném importu by se studenti měli zobrazit v tabulce, která se nachází pod formulářovým polem na vložení studentů a u každého z nich se zobrazí stav předmětu „nesplněno“.

Pokud bychom vkládali studenty k následovým praxím a měli bychom potřebu studenty rozdělit do více skupin, musíme vytvořit více rozvrhových akcí a do každé z nich vložit jenom část studentů, což vyžaduje ruční úpravu csv souboru v tabulkovém procesoru.

### 5.3.6 VYTVOŘENÍ ROZVRHU PRO NÁSLECHOVÉ PRAXE

Výchozím bodem pro vytvoření vyučovacích hodin následových praxí je seznam rozvrhových akcí (obrázek 19). Kliknutím na oranžové tlačítko přejdeme do správy vyučovacích hodin předmětu. Ve spodní části stránky, pod tabulkou se seznamem studentů se nachází tlačítko pro vytvoření vyučovacích hodin. Po poklepání se zobrazí tabulka, jež můžeme spatřit na obrázku 21. Tato tabulka má přesně danou strukturu a počet řádek je určený podle počtu vyučovacích hodin. Po odeslání formuláře dojde k vytvoření hodin u každého napojeného studenta.

#### Vytvořit vyučovací hodiny

Datum	Třída	Hodina	Vyučující	Škola
dd.mm.rrrr	Třída	Vyučovací hodina	Mgr. Jiří Dvořák ▼	
dd.mm.rrrr	Třída	Vyučovací hodina	Mgr. Jiří Dvořák ▼	
dd.mm.rrrr	Třída	Vyučovací hodina	Mgr. Jiří Dvořák ▼	
dd.mm.rrrr	Třída	Vyučovací hodina	Mgr. Jiří Dvořák ▼	
dd.mm.rrrr	Třída	Vyučovací hodina	Mgr. Jiří Dvořák ▼	

Vytvořit vyučovací hodiny

Obrázek 21: Formulář pro zadání termínů následu

### 5.3.7 SEZNAM (SPOLUPRACUJÍCÍCH) ŠKOL

Poslední položka v administrátorském menu obsahuje odkaz na seznam spolupracujících škol. Jako spolupracující škola je označena každá škola, jejíž alespoň jeden vyučující se podílel na následové nebo výstupové praxi.

## 5.4 ROZHRAŇÍ DIDAKTIKA

Následujících několik podkapitol popisuje uživatelské rozhraní pro didaktiku.

### 5.4.1 SEZNAM DOZOROVANÝCH STUDENTŮ

Na této stránce (obrázek 22) se nachází přehledný seznam všech studentů ze všech rozvrhových akcí, ke kterým je daný didaktik přidělen. Tito studenti jsou řazeni dle let sestupně a u každého z nich je uveden údaj, jestli má předmět splněný či nikoliv. Může nabývat celkem tří stavů – nehodnoceno, splněno, nesplněno.

Pedagogické praxe Didaktik ▾ Kontakt na SPP Ing. Jan Novák, Ph.D. ▾

Rozvrhové akce studentů které mám nastarosti

Název předmětu	Student	Rok (semestr)	Má splněno?	Hodinová dotace	Akce
KVD/NASPN	Tereza	ZS 2018	Nehodnoceno	5 hodin	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 3px;">Detail ROAK</a>
KVD/NASPN	Luboš	ZS 2018	Nehodnoceno	5 hodin	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 3px;">Detail ROAK</a>
KVD/NASPN	Michaela	ZS 2018	Nehodnoceno	5 hodin	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 3px;">Detail ROAK</a>
KVD/NASPN	Kristýna	ZS 2018	ANO	5 hodin	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 3px;">Detail ROAK</a>
KVD/NASPN	Anežka	ZS 2018	Nehodnoceno	5 hodin	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 3px;">Detail ROAK</a>
KVD/NASPN	Jan	ZS 2018	Nehodnoceno	5 hodin	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 3px;">Detail ROAK</a>
KVD/NASPN	Barbora	ZS 2018	Nehodnoceno	5 hodin	<a href="#" style="background-color: #007bff; color: white; padding: 5px 10px; border-radius: 3px;">Detail ROAK</a>

Obrázek 22: Seznam přidělených studentů

Z tohoto seznamu vede odkaz do detailu předmětu jednotlivých studentů. V tomto přehledu je kromě základních informací o rozvrhové akci k dispozici ještě seznam všech instancí vyučovacích hodin daného studenta včetně informací o vyučovací hodině, třídě, jménu učitele a škole, kde se bude tato hodina konat.

Didaktik má v tomto rozhraní možnost si vést ke studentovi privátní i veřejné poznámky a je zde také zaškrťovací pole, po jehož zaškrtnutí se označí, že student splnil předmět (tzn. zelenou barvu u sloupečku „Má splněno?“ ve výše uvedeném seznamu studentů.

Detail jedné z vyučovacích hodin studenta je k vidění na následujícím obrázku. Rozhraní studenta vypadá analogicky, jenom je zde možnost nahrát hospitační dotazník a vyplnit reflexi. Učitel si zde může zobrazit vložené hospitační dotazníky a odpovědi na reflexi, kterou student odeslal.

Pedagogické praxe Didaktik ▼ Kontakt na SPP Zpět do administrace Zpět Ing. Jan Novák, Ph.D. ▼

**Detail hodiny č.1 (KVD/NASPN)**

Číslo hodiny	Datum	Třída	Hodina č.	Vyučující	Škola	Obsah hodiny	Komentář vyučujícího
1	20. 3. 2018	7. B	4	Mgr. Tomáš Černý, DiS.	Masarykovo gymnázium Plzeň	Nevyplněný	Nevyplněný

**Záznamový arch z dané hodiny**

### Nahrané soubory

Doposud nebyly nahrány žádné soubory!

**Reflexe hodiny**

Student/učitel:	Zcela:	Dostatečně:	Nedostatečně:	Vůbec:	Zdůvodnění:
Probíranou látku podává věcně správně.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	nezadáno

Obrázek 23: Detail hodiny náslechové praxe

#### 5.4.2 VYUČOVACÍ HODINY U NÁSLECHOVÉ PRAXE

Didaktik si může zobrazit vyučovací hodiny u každého předmětu náslechové praxe, ke kterému je spárován. Pro zobrazení tohoto přehledu stačí přejít ze seznamu rozvrhových akcí pomocí tlačítka „Vyučovací hodiny“.

## 5.5 ROZHRANÍ UČITELE ZE ŠKOLY

### 5.5.1 SEZNAM VEDENÝCH STUDENTŮ

V této sekci se nachází seznam všech studentů, kteří mají vazbu na přihlášeného učitele. Z tohoto přehledu se poté dále přechází do detailu konkrétního studenta, kde dojde k zobrazení předmětů, které jsou vázány s konkrétním studentem (obrázek 24).

Pedagogické praxe Učitel Kontakt na SPP [Zpět do administrace](#) [Zpět](#) Mgr. Tomáš Černý, DiS. ▾

Rozvrhové akce studenta (Jan), kterého mám nastarosti

ID	Název předmětu	Rok (semestr)	Má splněno?	Hodinová dotace	Akce
6	KVD/NASPN	ZS 2018	Nehodnoceno	5 hodin	Pro následovné praxe nejsou k dispozici žádné akce
13	SPP/PPS	LS 2018	Nehodnoceno	2 hodin	<a href="#">Detail ROAK</a>

Obrázek 24: Rozvrhové akce studenta

Pokud klikneme na tlačítko „Detail ROAK“ tak se dostaneme na základní informace o této rozvrhové akci a seznamu vyučovacích hodin studenta. Odtud pak vede odkaz na jednotlivé hodiny (obrázek 23).

V případě, že jde o výsledkové praxe, tak vyučující vyplňuje hodnocení studentovy výuky ve třídě prostřednictvím formuláře na odeslání reflexe.

### 5.5.2 MOJE VYUČOVACÍ HODINY

Stránka obsahuje sestupně seřazené vyučovací hodiny daného učitele (obrázek 25).

Pedagogické praxe Učitel Kontakt na SPP [Zpět](#) Mgr. Jiří Dvořák ▾

Vyučovací hodiny, ke kterým jsem přiřazen (řazeno sestupně dle datumu)

Datum konání	Hodina číslo	Třída
21. 3. 2018	2	2. B
21. 3. 2018	5	2. A

Obrázek 25: Vyučovací hodiny přihlášeného učitele

## 5.6 ROZHRANÍ STUDENTA

### 5.6.1 VYTVOŘENÍ VYUČOVACÍCH HODIN (VÝSTUPOVÁ PRAXE)

U dlouhodobých výstupových praxí si studenti hledají učitele pro praxi sami. Z důvodu časové úspory SPP je umožněno studentům vytvářet učitele a ty pak párovat se svými vyučovacími hodinami.

Vytvořit hodiny je možné v detailu rozvrhové akce (obrázek 26). Stiskem tlačítka se zobrazí tabulka podobná tabulce na obrázku 21. Po jejím vyplnění dojde k vytvoření jednotlivých hodin.

Vyučovací hodiny k ROAK (student: Luboš)

Pro vyplnění hospitačních dotazníků zvolte tlačítkem detail hodiny.

Číslo hodiny	Datum	Třída	Hodina č.	Vyučující	Škola	Obsah hodiny	Komentář vyučujícího	Akce
Výuka zatím nebyla zadána! <input type="button" value="Zadat výuku"/>								

Obrázek 26: Poloha tlačítka pro zadání výuky výstupových praxí

V průběhu praxe se zde poté průběžně vyplňuje příprava na každou hodinu, která by měla být do systému vkládána s přijatelným časovým předstihem.

Dle vyplněné přípravy na jednotlivé hodiny a výstupy z hospitací dozorujícího vyučujícího ze školy se pak didaktik rozhoduje, zdali je nutná jeho fyzická návštěva školského zařízení, kde student vykonává praxi.

### 5.6.2 VYPLNĚNÍ REFLEXE K PŘEDMĚTU

Studenti vyplňují reflexi zpravidla u náslechových praxí. Ta se skládá z množiny otázek, na které lze zodpovědět jednou ze 4 možností a doplnit krátkým komentářem, kde dojde k vysvětlení studentovy volby.

Formulář na vyplnění reflexe se nachází v detailu konkrétní vyučovací hodiny dané rozvrhové akce, pro které je vyplnění studentských reflexí vyžadováno.



## ZÁVĚR

V práci byla řešena především tvorba nového systému pro vedení pedagogických praxí. Po důkladné analýze proběhnul návrh nového systému. Toto bylo velmi složité, jelikož jsme neměli žádný stávající elektronický systém, ze kterého bychom mohli vycházet. Vytvořená aplikace je tedy průkopníkem pro budoucí systémy.

V teoretické části práce byly popsány principy fungování současného systému pedagogických praxí. Velká pozornost byla kladena na podrobnou analýzu výstupových praxí, které jsou dvojího typu a vzájemně se v některých aspektech odlišují. Kromě toho byly odhaleny některé slabiny současného systému, které nový systém eliminuje. Díky tomuto systému budou mít všechny zúčastněné strany k dispozici aktuální informace takřka okamžitě a nebudou muset čekat na stávající papírové výstupy.

Hlavním smyslem práce bylo vytvořit přehledný webový systém, který se bude snadno používat a nebude problémem do něj implementovat nové funkcionality. Domnívám se, že se nám podařilo vytvořit kvalitní systém postavený na nejnovějších technologiích, které jsou dobře dokumentovány jak v oficiální dokumentaci, tak přímo ve zdrojovém kódu zhotovené aplikace.

Již při tvorbě práce jsme narazili na značné množství možných vylepšení, které nejspíše budou implementovány v budoucnosti. Jako možná úprava se nabízí zavedení možnosti uzpůsobovat výběr reflexních otázek v závislosti na konkrétním předmětu. Dalším možným vylepšením je vytvořit nativní podporu pro vytváření skupin studentů u náslechových praxí, aby odpadla nutnost při větším počtu studentů vytvářet více rozvrhových akcí.

**RESUMÉ**

This bachelor work deals with the analysis and creation of a new system for administration of teaching practices at FPE ZČU. The objective of this work was to analyze the current system of teaching practices and to design new solution that will eliminate weaknesses of the original system. In this bachelor work there is a complete summary of received information, information about the selected device for the development of the system, the description of selected parts of the source code and finally the work includes the description of the user interface.

## SEZNAM LITERATURY

1. PAEDDR. JOSEF KEPKA, C. Fakulta pedagogická. *Harmonogram pedagogických praxí 2017 /2018* [online]. 2017 [cit. 2018-06-24]. Dostupné z: <https://fpe.zcu.cz/spp/harmonogram.html>
2. CourseWARE. *KVD/NASPN* [online]. 2009 [cit. 2018-06-24]. Dostupné z: <https://courseware.zcu.cz/portal/studium/courseware/kvd/naspn>
3. PAEDDR. JOSEF KEPKA, C. Fakulta pedagogická. In: *Pokyny pro souvislou pedagogickou praxi studentů závěrečných ročníků* [online]. 2017 [cit. 2018-06-24]. Dostupné z: <https://fpe.zcu.cz/export/sites/fpe/spp/pokynySPP/praxeprub.pdf>
4. PAEDDR. JOSEF KEPKA, C. Fakulta pedagogická - pokyny pro praxe. In: *Pokyny k vykonání praxe studentů v organizacích, které zajišťují mimoškolní činnost dětí a mládeže* [online]. 1. 10. 2014 [cit. 2018-06-24]. Dostupné z: <https://fpe.zcu.cz/export/sites/fpe/spp/pokynySPP/Pokyny-mimokoln-organizace-1.doc>
5. PAEDDR. JOSEF KEPKA, C. [Seznam organizací a institucí, ve kterých je možné realizovat pedagogickou praxi v rámci mimoškolní ...] In: *Fakulta pedagogická - pokyny pro praxe* [online]. [cit. 2018-06-24]. Dostupné z: <https://fpe.zcu.cz/export/sites/fpe/spp/pokynySPP/Seznam-organizac-a-instituc.xls>
6. sitepoint. *The Best PHP Framework for 2015: SitePoint Survey Results* [online]. 30. 4. 2015. Dostupné také z: <https://www.sitepoint.com/best-php-framework-2015-sitepoint-survey-results/>
7. ITnetwork. *Lekce 1 - Úvod do Nette frameworku pro PHP* [online]. 23. 3. 2015. Dostupné také z: <https://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette/>
8. Symfony. *What is Symfony* [online]. [cit. 2018-06-10]. Dostupné z: <http://symfony.com/what-is-symfony>
9. Symfony. *About Symfony Project* [online]. [cit. 2018-06-10]. Dostupné z: <http://symfony.com/about>
10. yiisoft/yii2. *Roadmap* [online]. 13. 5. 2018. Dostupné také z: <https://github.com/yiisoft/yii2/wiki/Roadmap>
11. MAKSIM SURGUY: PERSONAL WEBSITE AND BLOG. *History of Laravel PHP framework, Eloquence emerging* [online]. 27. 6. 2013. Dostupné také z: <https://maxoffsky.com/code-blog/history-of-laravel-php-framework-eloquence-emerging/>
12. Medium. *Laravel is Hiring Again* [online]. 9. 4. 2018. Dostupné také z: <https://medium.com/@taylorotwell/laravel-is-hiring-again-246b502f7dcb>
13. GitHub. *About Laravel* [online]. 1. 6. 2018. Dostupné také z: <https://github.com/laravel/laravel>
14. Save Contact Form 7. *Everything You Need to Know About Laravel Framework* [online]. 7. 2. 2017 [cit. 2018-06-10]. Dostupné z: <http://www.savecontactform7.com/everything-you-need-to-know-about-laravel-framework>

15. Laravel. *Artisan Console* [online]. [cit. 2018-06-10]. Dostupné z: <https://laravel.com/docs/5.5/artisan>
16. Laravel. *Blade Templates* [online]. [cit. 2018-06-11]. Dostupné z: <https://laravel.com/docs/5.5/blade>
17. medium. *Comparing Blade and Twig templates in Laravel* [online]. 19. 9. 2015 [cit. 2018-06-11]. Dostupné z: <https://medium.com/@barryvdh/comparing-blade-and-twig-templates-in-laravel-187fde7fcac9>
18. Laravel. *Directory Structure* [online]. 30. 8. 2018. Dostupné také z: <https://laravel.com/docs/5.5/structure>
19. Composer. *Dependency Manager for PHP* [online]. [cit. 2018-06-10]. Dostupné z: <https://getcomposer.org/>
20. Fakulta pedagogická. *Středisko pedagogické praxe (SPP)* [online]. 13. 12. 2017. Dostupné také z: <https://fpe.zcu.cz/spp/index.html>
21. interval.cz. *Jak přežít Apache – Virtual Hosts* [online]. 17. 2. 2002 [cit. 2018-06-01]. Dostupné z: <https://www.interval.cz/clanky/jak-prezit-apache-virtual-hosts/>

**SEZNAM OBRÁZKŮ A TABULEK**

Obrázek 1: Oficiální logo Laravelu .....	12
Obrázek 2: Schéma principu práce aplikace (14).....	13
Obrázek 3: Zjednodušené schéma adresářové struktury .....	16
Obrázek 4: Logo nástroje Composer .....	17
Obrázek 5: Obsah tabulky migrations .....	18
Obrázek 6: Databázové tabulky pro role uživatelů .....	19
Obrázek 7: Organizace rozvrhových akcí .....	21
Obrázek 8: Databázové tabulky pro ukládání reflexí .....	22
Obrázek 9: Ukázka části výpisu příkazem phpinfo .....	28
Obrázek 10: Ilustrační výpis vytvoření a naplnění tabulek .....	30
Obrázek 11: Přihlašovací stránka do systému .....	32
Obrázek 12: Nástěnka portálu .....	33
Obrázek 13: Seznam uživatelů .....	34
Obrázek 14: Ruční úprava uživatele – formulář.....	34
Obrázek 15: Formulář pro založení nového učitele.....	35
Obrázek 16: Formulář pro založení nových didaktiků .....	36
Obrázek 17: Formulář pro založení nového předmětu .....	37
Obrázek 18: Formulář pro vytvoření rozvrhové akce .....	38
Obrázek 19: Seznam rozvrhových akcí .....	39
Obrázek 20: Získání datového CSV souboru se studenty .....	39
Obrázek 21: Formulář pro zadání termínů náslehu .....	40
Obrázek 22: Seznam přidělených studentů .....	41
Obrázek 23: Detail hodiny náslechové praxe .....	42
Obrázek 24: Rozvrhové akce studenta .....	43
Obrázek 25: Vyučovací hodiny přihlášeného učitele .....	43
Obrázek 26: Poloha tlačítka pro zadání výuky výstupových praxí .....	44
Tabulka 1: Základní informace o porovnávaných frameworkcích.....	11
Tabulka 2: Výběr konstrukcí směrovacího systému a jejich popis .....	26



**PŘÍLOHA 2 – ZÁZNAM O BLOKOVÉ PEDAGOGICKÉ PRAXI**



EVROPSKÁ UNIE  
Evropské strukturální a investiční fondy  
Operační program Výzkum, vývoj a vzdělávání



Zvyšování kvality pregraduálního vzdělávání  
na Fakultě pedagogické ZČU v Plzni

CZ.02.3.68/0.0/0.0/16\_038/0006677

Fakulta pedagogická Západočeské univerzity v Plzni Školní rok: .....

Středisko pedagogické praxe Semestr: .....

Plzeň, Chodské náměstí 1, 306 19

Záznam o blokové pedagogické praxi

Studující: ..... Ročník: ..... Aprob. předmět: .....

Škola praxe: .....

Samostatné výstupy:

Datum zadání přípravy	Datum konzul. přípr.	Datum výstupu	Tř.	Učivo hodiny	Výsledek rozboru – hodnocení	Podpis učitele

Vzájemné hospitace:

Datum	Tř.	Učivo hodiny – závěr rozboru	Podpis učitele

Doporučuji – nedoporučuji udělení zápočtu ..... Datum podpis  
Započteno dne ..... podpis

**PŘÍLOHA 3 – HOSPITAČNÍ ZÁZNAM HODINY A JEJÍ REFLEXE****Hospitační záznam hodiny**

Škola:

Třída:

Počet žáků:

Datum a čas:

Předmět:

Vyučující:

Téma vyučovací hodiny:

zaměření třídy:

/z toho žáků s SVP...../s OMJ.....

**1. Struktura vyučovací hodiny**

Čas (čas změny činnosti)	Popis činnosti

Edukační cíle:

Použité metody:

Organizační formy:

Pomůcky:



**2. Reflexe hodiny:****A) Obsahová a metodická úroveň hodiny:**

Student/učitel:	zcela	dostatečně	nedostatečně	vůbec	Zdůvodnění:
Probíranou látku podává věcně správně.					
Probíranou látku podává přiměřeně (ke stupni vzdělávání, vstupním znalostem žáků).					
Probíranou látku podává srozumitelně (výukové metody odpovídají cílům a obsahu (učivu), žákům s SVP, OMJ apod.).					
Probíranou látku podává v případě potřeby individualizovaně (ve vztahu k žákům s SVP, OMJ apod.).					
Hodinu strukturuje funkčně (členění do logických celků, návaznost a gradace učiva).					
Volí vhodné organizační formy (ve vztahu k cílům, metodám a učivu, žákům s SVP, OMJ apod.).					
Vytváří prostředí pro aktivní učení žáků (žák navrhuje a zdůvodňuje postupy řešení úloh, pracuje samostatně, kooperuje*).					
Probíranou látku uvádí v kontextu (propojuje ji s již známou látkou, zasazuje do souvislosti mezi jednotlivými tématy, včetně průřezových, s dalšími předměty, dokládá příklady z reálného života).					
Adekvátně využívá učební pomůcky pro názornost a vizualizaci (obrázky, schémata, modely, videa, audio, mapy, atlasy, ...).					
Poskytuje žákům formativní popisnou zpětnou vazbu.					
Vede žáky k seberefektovanému učení.					

\* žák vysvětluje, kategorizuje, uvádí příklady, formuluje poučky a vyvozuje závěry, navrhuje a zdůvodňuje vlastní postupy řešení...

**B) Profesně a osobnostně kompetenční hledisko:**

Verbální projev učitele (tempo, hlasitost, srozumitelnost, spisovný jazyk, parazitní slova)	
Neverbální komunikace (gesta, posturologie...)	
Interakce učitel žák	
Další:	

Celkové zhodnocení studenta (jeho silné a slabé stránky a doporučení pro další rozvoj):

PŘÍLOHA 4 – DATABÁZOVÉ SCHÉMA

