



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**

FACULTY OF INFORMATION TECHNOLOGY

**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**MODULY INFORMAČNÍHO SYSTÉMU  
PRO PLÁNOVÁNÍ ROZVRHŮ**

MODULES OF INFORMATION SYSTEM FOR TIMETABLE PLANNING

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JANA STOPKOVÁ**

**VEDOUcí PRÁCE**

SUPERVISOR

**Ing. JAROSLAV DYTRYCH, Ph.D.**

BRNO 2022

## Zadání bakalářské práce



Studentka: **Stopková Jana**  
Program: Informační technologie  
Název: **Moduly informačního systému pro plánování rozvrhů**  
**Modules of Information System for Timetable Planning**  
Kategorie: Informační systémy

### Zadání:

1. Seznamte se s problematikou plánování rozvrhů zkoušek a výuky na FIT VUT v Brně a s programovacími jazyky využitými v nástrojích pro jeho podporu.
2. Prostudujte dostupné programy pro podporu této činnosti, zejména systém Rozvrhy IS, a vstupy a výstupy plánování z předchozích semestrů.
3. Navrhněte nové moduly systému Rozvrhy IS určeného pro plánování rozvrhů, které umožní rychlou a efektivní přípravu plánování, zejména předzpracování požadavků, nastavení parametrů předmětů a přednáškových skupin apod. Zaměřte se při tom na rychlost zobrazení rozsáhlých tabulek a upozornění na chyby ve vstupních datech.
4. Implementujte navržené řešení.
5. Zhodnoťte dosažené výsledky a vytvořte stručný plakát prezentující výsledky práce.

### Literatura:

- Dle doporučení vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a 3.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Dytrych Jaroslav, Ing., Ph.D.**

Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.

Datum zadání: 1. listopadu 2021

Datum odevzdání: 11. května 2022

Datum schválení: 1. listopadu 2021

## Abstrakt

Tato práce se zabývá plánováním rozvrhů pro výuku a zkoušky na Fakultě informačních technologií Vysokého učení technického v Brně (FIT). Jejím cílem je navrhnout způsob zrychlení modulů aktuálně používaných pro přípravu plánování a tyto moduly následně integrovat do jiné aplikace, která je na FIT používána pro plánování. První část práce se zabývá uvedením do problematiky plánování rozvrhů na FIT. V druhé části je popsán návrh aplikace a následná implementace zakončená testováním během skutečného plánování rozvrhů.

## Abstract

This thesis deals with planning teaching and examination schedules at the Faculty of Information Technology of Brno University of Technology (FIT). It aims to propose a way to accelerate the modules currently used for scheduling and then integrate them into another application used for scheduling at FIT. The first part of the thesis deals with an introduction to the scheduling issues at FIT. The second part describes the application design and subsequent implementation, concluding with testing during actual scheduling.

## Klíčová slova

informační systém, moduly, plánování rozvrhů, HTML, CSS, JavaScript, PHP, optimalizace, REST API

## Keywords

information system, modules, timetable planning, HTML, CSS, JavaScript, PHP, optimization, REST API

## Citace

STOPKOVÁ, Jana. *Moduly informačního systému pro plánování rozvrhů*. Brno, 2022. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Dytrych, Ph.D.

# Moduly informačního systému pro plánování rozvrhů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Jaroslava Dytrycha, Ph.D. Další informace ohledně průběhu plánování rozvrhů mi poskytla Bc. Kristýna Zaklová. Uvedla jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpala.

.....  
Jana Stopková  
6. května 2022

## Poděkování

Ráda bych poděkovala Ing. Jaroslavu Dytrychovi, Ph.D. za odborné vedení této práce, konzultace a cenné informace. Dále bych také chtěla poděkovat Ing. Marku Kuchynkovi, Ing. Aleně Tesařové a Bc. Kristýně Zaklové za jejich ochotu věnovat se mým dotazům souvisejícím s plánováním a samotnou implementací. Díky také patří mé rodině, která mi byla během celého studia velkou oporou a také mé kamarádce, která mě mnohdy dokázala nasměrovat správným směrem.



# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Plánování rozvrhů na FIT</b>	<b>4</b>
2.1	Metoda plánování využívaná na FIT . . . . .	5
2.2	Organizace studia na FIT . . . . .	6
2.3	Průběh plánování . . . . .	7
2.4	Přechod z IS FIT na IS VUT . . . . .	11
2.5	Aktuálně využívaný software . . . . .	12
2.6	Požadavky na nové moduly . . . . .	16
<b>3</b>	<b>Využití technologie</b>	<b>19</b>
3.1	HTML a CSS . . . . .	19
3.2	JavaScript . . . . .	19
3.3	PHP . . . . .	20
3.4	MySQL . . . . .	20
<b>4</b>	<b>Návrh nových modulů aplikace</b>	<b>21</b>
4.1	Úpravy databáze . . . . .	21
4.2	Návrh modulu pro správu období . . . . .	22
4.3	Návrh modulu pro správu požadavků na výuku a zkoušky . . . . .	23
4.4	Návrh modulu pro správu příznaků k předmětům . . . . .	24
4.5	Návrh modulu pro import dat z REST API . . . . .	25
<b>5</b>	<b>Implementace</b>	<b>27</b>
5.1	Správa období . . . . .	27
5.2	Požadavky na výuku a zkoušky . . . . .	28
5.3	Příznaky u předmětů . . . . .	29
5.4	Správa předmětů . . . . .	30
5.5	Import dat . . . . .	32
<b>6</b>	<b>Testování</b>	<b>35</b>
6.1	Průběžné testování při vývoji . . . . .	35
6.2	Nasazení při plánování . . . . .	37
6.3	Měření času importu dat . . . . .	37
<b>7</b>	<b>Závěr</b>	<b>39</b>
	<b>Literatura</b>	<b>40</b>



# Kapitola 1

## Úvod

Plánováním rozvrhů se pravidelně musí zabývat mnoho akademických institucí. Osoby, které mají plánování rozvrhů na Fakultě informačních technologií Vysokého učení technického v Brně (FIT) na starosti, několikrát za rok sestavují rozvrhy a aby nemuseli používat papír a tužku, jsou jim k dispozici programy, které jim s tím pomohou. Plánovat s tužkou v ruce výuku na vysoké škole zní možná nereálně, ale i toto je stále jeden z běžně využívaných způsobů popsaných v kapitole 2.

Kapitola 2.1 vysvětluje důvody, proč se právě metoda plánování s podporou počítače na FIT používá. Následují kapitoly 2.2 a 2.3, které přibližují organizaci studia na FIT a také popisují, jakým způsobem plánování na FIT probíhá a co vše mu musí předcházet. V rámci těchto kapitol je také definováno, na co vše je nutné pamatovat, aby byl výsledkem plánování použitelný rozvrh. Vzhledem k tomu, že do procesu plánování vstupuje velké množství omezení, je zapotřebí i aplikace, ve které by bylo možné před samotným plánováním všechna potřebná data připravit. Takováto aplikace již existuje a je jí spolu s aktuálně používanou aplikací pro plánování věnována kapitola 2.5. Přípravu plánování ovšem ztěžují některé její negativní faktory.

Cílem práce je tedy do webové aplikace pro plánování rozvrhů, kterou vytvořil Ing. Marek Kuchynka, integrovat moduly pro rychlou a efektivní přípravu plánování rozvrhů. Moduly týkající se předzpracování dat pro jejich využití při plánování se budou zaměřovat na rychlost prezentace rozsáhlých dat uživateli. Modul pro import dat potřebných pro plánování se též bude zabývat rychlostí, ovšem z trochu jiného úhlu. U importu dat je nutné zredukovat jeho časovou náročnost, a to v některých případech o více jak hodinu. Více o požadavcích kladených na výslednou aplikaci je možné najít v kapitole 3.

Navržený způsob řešení jednotlivých modulů bude představen v kapitole 4. Tato kapitola se také věnuje způsobu, jak budou jednotlivé implementované části do již existující aplikace integrovány.

V kapitole 5 budou výsledné moduly popsány a též zde budou uvedeny některé implementační detaily. Na tuto část naváže kapitola 6 pojednávající o testování, které 2x proběhlo při reálném plánování rozvrhů. Ukázalo se, že implementované moduly dokáží rozvrhářům výrazně ušetřit čas, zvláště při importování dat.

Poslední část – kapitola 7 – shrnuje výsledky, kterých se podařilo dosáhnout. Také je zde nastíněno několik způsobů, jak by bylo možné do budoucna plánování a jeho přípravu rozšířit a usnadnit.

## Kapitola 2

# Plánování rozvrhů na FIT

Plánování rozvrhů je problémem, kterému čelí různé akademické instituce. Tento problém spočívá v hledání způsobu přiřazení vyučujících, studentů a místností k přednáškám, zkouškám, či jiným aktivitám. Cílem je sestavit rozvrh tak, aby byl uskutečnitelný a aby vyhovoval co nejvíce lidem, kteří se jím budou poté řídit.

Omezení, která do procesu plánování vstupují, bývají obecně rozdělována do dvou kategorií [3, 21]:

- **Tvrdá omezení** jsou omezení, která musí být splněna vždy. Zaručují totiž proveditelnost výsledného rozvrhu. Proveditelností se rozumí, že např. nenastane situace, kdy by měl být jeden učitel přítomen na dvou různých přednáškách zároveň.
- **Měkká omezení** oproti těm tvrdým splněna nutně být nemusí. S počtem splněných měkkých omezení ale roste kvalita navrženého rozvrhu. Jako příklad takového omezení mohou být uvedeny preference jednotlivých učitelů, případně i studentů.

## Metody plánování rozvrhů

Existuje několik způsobů, jak může být plánování prováděno. Tato podkapitola se bude věnovat třem možným způsobům a v podkapitole následující (2.1) bude uveden způsob plánování, který se aktuálně na *Fakultě informačních technologií Vysokého učení technického v Brně* (dále jen FIT) využívá.

### Automatizované plánování

Naplánování rozvrhu může být plně ponecháno v režii počítačového programu. Osoba zodpovědná za plánování rozvrhů (dále jen rozvrhář) se tedy nemusí na tvorbě rozvrhu ve větší míře podílet. Je však nutné, aby byly přesně zadané všechny požadavky, které má při tvorbě rozvrhu program zohlednit.

Rozvrh vytvořený počítačem nemusí být vždy úplně vyhovující a může být potřeba provedení manuálních úprav. Bylo totiž dokázáno (např. [8]), že plánování rozvrhů je NP-úplný problém. To znamená, že neexistuje žádný deterministický algoritmus, který by našel optimální řešení v polynomiálním čase. K optimálnímu řešení se tedy můžeme pouze přiblížit. Způsobů jak toho dosáhnout je hned několik. V minulosti byly k efektivnímu řešení tohoto problému použity např. genetické algoritmy [4, 14].

## Manuální plánování

V případě **manuálního plánování** rozvrhář využívá pouze papír a tužku a případně jiné kancelářské potřeby. Alternativou také může být tabulkový editor, např. Excel. Počítač do plánování a kontroly splnění omezení nijak nezasahuje a rozvrhář si tedy musí tvrdá a měkká omezení hlídat sám (ideálně s každým oknem přidaným do rozvrhu). Když dojde k porušení některého z omezení, může být složité tento nedostatek odhalit a chybu opravit. Pokud se tak stane v pozdějších fázích plánování, nemusí být možné kolizní položku přesunout tak, aby nebylo nutné spolu s ní pozměnit i další části rozvrhu.

## Plánování s podporou počítače

V pomyslném středu mezi automatizovaným a manuálním plánováním stojí plánování s podporou počítače. U této metody sehrává svou roli člověk i počítač. Před zahájením plánování jsou programu zadána vstupní data, na základě kterých program rozvrháři poskytuje zpětnou vazbu – kontroluje kolize (např. přednášky z dvou předmětů povinných pro jeden ročník ve stejném čase), zobrazuje relevantní informace k plánovaným úlohám, navrhuje možnosti apod. Úkolem rozvrháře je na základě poskytnuté podpory rozvrh sestavit.

### 2.1 Metoda plánování využívaná na FIT

Aktuálně využívanou metodou plánování na FIT je metoda plánování s podporou počítače. V následující sekci budou popsány důvody pro zvolení této metody. Rovněž budou rozebrány vlastnosti ostatních metod, které z nich činí nevhodné kandidáty pro použití na FIT.

Vzhledem velkému počtu přednášek, cvičení a jiných akcí, které je nutné naplánovat, můžeme z metod plánování připadajících v úvahu vyloučit manuální plánování. Tato metoda je pomalá a náročnější než ostatní.

Automatizované plánování by mohlo přinést uspokojivý výsledek v kratším čase. Existuje ale důvod, proč se nepoužívá. Tento důvod pramení z faktu, že automatizované plánování, jak již bylo zmíněno, vyžaduje přesně zadaná data. Některá potřebná data ale nemusí být vyučujícími vždy zadána. Absence těchto dat ovšem nemůže být brána tak, že učitel žádné požadavky nemá.

Další překážkou pro tento způsob plánování jsou požadavky ukládané v textové podobě. Takovéto požadavky by bylo velmi náročné zpracovávat strojově. Tímto způsobem se na FIT ukládají primárně poznámky k požadavkům. Tyto poznámky nemohou být opomíjeny, protože mohou obsahovat specifická upřesnění požadavků, která nelze do systému zadat jiným způsobem. Jako příklad je možné uvést odstup mezi zkouškami. Díky možnosti vyplnit textovou poznámku se také může stát, že učitel vypíše všechny své požadavky právě do textové poznámky, ale ostatní data už nevyplní.

Kromě jiného se také občas může objevit chyba v požadavku, anebo může jít o přehnaný požadavek, kterému nebude možné vyhovět. Některé problémy by počítač dokázal zachytit snadno, jiné ale dokáže odhalit jen osoba dobře seznámená s plánováním rozvrhů a organizací studia na FIT.

A na závěr, některá data nemusí být v době plánování vůbec známa (např. počet zapsaných studentů v předmětech, či počet studentů, kteří v předmětu získají zápočet a budou moci přistoupit ke zkoušce), případně dosud nejsou nikde evidována (např. časy potřebné pro přesuny mezi místnostmi).

Na základě důvodů popsaných výše zůstává plánování s podporou počítače i nadále tím nejvhodnějším způsobem, jak rozvrhy na FIT plánovat.

## 2.2 Organizace studia na FIT

Pro správné pochopení některých částí následujících sekcí je třeba zmínit určité informace ohledně organizace studia na FIT.

Akademický rok se skládá ze dvou semestrů (zimní a letní), přičemž každý z nich je rozdělen na výukové a zkuškové období. Výuka probíhá v prvních 13 týdnech a dalších 5 týdnů je věnováno zkouškám. Studium na FIT je rozděleno do tří stupňů: bakalářský (3 roky), magisterský (2 roky) a doktorský (4 roky). Plánování zkoušek se týká až na výjimky pouze bakalářského a magisterského studia. Proč tomu tak je bude objasněno v následujícím odstavci.

Studenti jsou zařazováni do různých **přednáškových skupin**, pro které se vytváří rozvrhy. Ke každému ročníku a stupni studia je přiřazena jedna skupina. Výjimku tvoří bakalářské studium, kde se studenti v prvních dvou ročnících rozdělují na dvě skupiny (*BIA* a *BIB*). Pro první ročník jsou tedy skupiny *1BIA* a *1BIB*, pro druhý ročník *2BIA* a *2BIB*. Pro třetí ročník je už pouze jedna přednášková skupina, a to skupina *3BIT*. Názvy přednáškových skupiny magisterského studia v českém jazyce též začínají ročníkem, ale navazuje na něj přípona *MIT*. U magisterského studia v anglickém jazyce je to přípona *EIT*. Dále je na FIT i skupina *INTE*, do které jsou zařazeni zahraniční studenti. Doktorského studia se dělení do skupin netýká z důvodu, že se pro tyto studenty sestavují individuální studijní plány [22]. V rámci doktorského studia je ale možné studovat několik předmětů, které mají přednášky a studují se typicky v prvním ročníku. Takovéto předměty jsou přiřazovány do rozvrhu skupiny *1DVI*.

### Výuka předmětů

Výuka předmětů může probíhat ve formě přednášek, cvičení (numerické, počítačové, laboratoře atd.) a demonstračních cvičení. Z hlediska pravidelnosti může výuka probíhat každý týden, anebo pouze v sudé, či liché týdny.

Pro jednotlivé specializace či dříve obory studia se liší **povinnost jednotlivých předmětů**. Na FIT existují tři kategorie povinností předmětů:

- **Povinné (P)** předměty musí student absolvovat v semestru a ročníku, který specifikuje studijní plán specializace či oboru, který student studuje.
- **Povinně volitelné (PV)** předměty vždy patří do určité skupiny. Z této skupiny si musí student vybrat minimálně předepsané množství předmětů (typicky minimálně jeden předmět) a tyto absolvovat.
- **Volitelné (V)** předměty mohou být absolvovány v libovolném ročníku. Výběr předmětů z této kategorie je zcela ponechán na studentovi. Do této kategorie spadají i předměty doporučené (D), které figurují v tabulce 4.1 a na FIT se využívají typicky ve studijních plánech pro Erasmus.

Aby student úspěšně zakončil studium, je spolu se získáním dostatečného množství kreditů (180 kreditů pro bakalářské studium, 120 kreditů pro magisterské studium) nutné absolvovat všechny povinné předměty a předepsaný počet povinně volitelných předmětů z každé skupiny.

V případě neúspěchu v povinném předmětu je možné tento předmět jednou opakovat, a to v nejbližším semestru, kdy se uskuteční výuka předmětu [1]. Pokud student neuspěje v povinně volitelném předmětu, může jej buď opakovat, anebo si může zapsat jiný předmět z dané skupiny povinně volitelných předmětů [2].

Pro správné zobrazování rozvrhů studentům je nutné vědět, pro jaké přednáškové skupiny mají předměty jakou povinnost. Právě z toho důvodu jsou k předmětům a přednáškovým skupinám přiřazovány různé **příznaky**:

- **Příznak oddo** je přiřazován k povinným předmětům. Jeho název vychází z historického dělení přednáškových skupin do několika dalších skupin, kdy tento příznak značil, že je předmět povinný pro všechny tyto podskupiny.
- **Příznak xx** je určen pro povinně volitelné a volitelné předměty. Dále je využíván také pro povinné předměty, pro které je přiřazován k ročníkům, ve kterých může být předmět opakován. Pokud je tedy předmět povinný pro první ročník, příznak *xx* je mu přiřazen k přednáškovým skupinám *2BIA* a *2BIB*.
- **Příznak obor** se přidává k těm předmětům, které jsou povinné pro některý magisterský obor/specializaci. U předmětu, který má tento příznak přiřazený, je v rozvrhu specifikováno, vůči kterému z oborů je povinný.

V přiřazování příznaků existuje několik výjimek, které upravují přiřazování příznaku *xx*:

- Předměty IZG a ITU je v určitých případech možné si zapsat o rok dříve. Proto je těmto předmětům v ročníku před ročníkem, ve kterém jsou povinné, přiřazen příznak *xx*.
- Předměty IKK a ISC jsou určeny pouze pro 1. ročník bakalářského studia. Příznak *xx* budou mít tedy pouze skupiny *1BIA* a *1BIB*.
- Předmět ISM je volitelný pouze pro první ročník bakalářského studia. Pokud ale student v prvním ročníku nezakončí úspěšně předmět IDM, může si předmět ISM zapsat i v druhém ročníku. Z toho důvodu bude mít předmět ISM příznak *xx* i pro přednáškové skupiny *2BIA* a *2BIB*.

## Zkoušky

Předměty vyučované na FIT mohou být zakončeny různými způsoby, a to udělením zápočtu, udělením klasifikovaného zápočtu, vykonáním kolokvia, vykonáním zkoušky nebo vykonáním zkoušky po předchozím udělení zápočtu [1].

V předmětech, ve kterých se koná zkouška, může student dle pravidel o organizaci studia na FIT<sup>1</sup> tuto zkoušku konat maximálně třikrát. Mohou být vypsány buď tři termíny zkoušky, kdy se prvním termínem rozumí *řádný termín* a dalšími poskytnutými termíny *opravné termíny*, anebo pět variantních termínů.

## 2.3 Průběh plánování

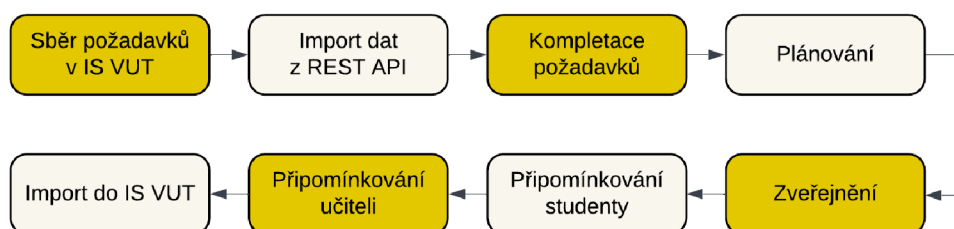
Procesem plánování rozvrhů, který naznačuje obrázek 2.1, si fakulta projde několikrát do roka. Pro každý semestr (zimní a letní) musí být vytvořen rozvrh výuky a zkoušek.

<sup>1</sup><https://www.fit.vut.cz/fit/info/smernice/sm2018-13-uz.pdf>



Výuka se obvykle plánuje v měsíci dubnu pro zimní semestr a v průběhu května pro letní semestr. Zkoušky po výuce v zimním semestru poté bývají plánovány v říjnu a zkoušky po výuce v letním semestru zase v březnu.

Podporu pro plánování aktuálně poskytují dvě různé aplikace. Příprava plánování nového rozvrhu společně s kompletací požadavků probíhá v aplikaci vytvořené Ing. Alenou Tesařovou. Samotné plánování rozvrhu poté pokračuje v aplikaci Ing. Marka Kuchynky. Obě aplikace budou blíže popsány v kapitole 2.5.



Obrázek 2.1: Aktuální proces plánování

## Příprava období a základních nastavení

Aby bylo s čím pracovat, musí učitelé nejprve v IS VUT vyplnit požadavky k předmětům, které vyučují. Na vyplnění svých požadavků mají učitelé, dle zápisů z kolegií děkana FIT<sup>2</sup>, 14 dní. Během tohoto období je již možné v aplikaci Ing. Tesařové vytvořit nové období a připravit prázdný rozvrh. U období se specifikuje rok, semestr, typ období (výuka, nebo zkoušky) a také datum začátku a konce. K novému rozvrhu je vždy nutné přidat různé texty (např. text před a za rozvrhem). Tyto texty se později zobrazí u finálního rozvrhu, který je zveřejněn studentům.

Vyplněné požadavky a další potřebná data, která jsou rozepsána níže v části věnované vstupním datům, jsou poté rozvrhářem importovány s pomocí REST API<sup>3</sup> z IS VUT.

## Vstupní data

Kromě požadavků učitelů na výuku a zkoušky (popsány později v této kapitole) je navíc nutné importovat tato data, bez kterých by plánování nebylo možné:

- Studijní programy, k nim patřící obory a jejich studijní plány
- Zařazení oborů do přednáškových skupin
- Rozsahy studijních skupin pro jednotlivé přednáškové skupiny
- Seznam předmětů, přičemž ke každému předmětu potřebujeme znát základní informace, kterými jsou: název a zkratka, počet kreditů, počty přednášek a cvičení (numerických, laboratorních, počítačových, příp. jiných), kapacita předmětu (kolik studentů si může předmět zapsat) a fakulta, která výuku daného předmětu zajišťuje. Dále potřebujeme také znát způsob zakončení předmětu, povinnost v ročnících jednotlivých

<sup>2</sup>Zápis č. 13/2022 z jednání rozšířeného kolegia děkana FIT dne 7. 4. 2022 je dostupný zde: <https://www.fit.vut.cz/fit/info/zkd/2022/zkd13-220407.pdf>

<sup>3</sup>Representational State Transfer Application Programming Interface



specializací/oborů a aktuální počet registrovaných a zapsaných studentů. Další důležitou informací u předmětu je to, zdali jsou jeho zkoušky plánovány centrálně, nebo si je plánuje sám učitel.

- Učitelé a jejich vazby na předměty, tedy například jaký učitel je přednášejícím v jakých předmětech.
- Počet společných studentů u dvou předmětů
- Počty studentů zapsaných minulý rok na zkoušky a počty registrovaných studentů jednotlivých oborů v jednotlivých předmětech. Tyto hodnoty jsou používány pro odhadování počtů studentů na výuku a zkoušky v aktuálním roce.
- Místnosti, ve kterých může probíhat výuka, či zkoušky
- Přiřazení typů hodnocení, které jsou v předmětech používány (např. projekt, půlse-mestrální zkouška, závěrečná zkouška atd.) k předmětům.
- Seznam registrovaných studentů v předmětech

### **Požadavky učitelů**

Požadavky, které učitelé mají možnost vyplnit, jsou preference, omezení a návaznosti, které byly nově zavedeny od roku 2022. Preference pro výuku a zkoušky se odlišují. Oproti níže uvedeným vyplnitelným údajům je navíc vždy možné přidat textovou poznámku.

K preferencím na výuku i zkoušky je také možné přidat požadavek na počet paralelně probíhajících bloků výuky či zkoušek, počet navazujících bloků a přestávku mezi navazujícími bloky. Takto učitel může zadat třeba preference typu 3 cvičení naráz probíhající v několika místnostech, anebo 2 cvičení, mezi kterými bude přestávka 50 minut.

#### **1. Preference pro zkoušky**

- Termín zkoušky a k němu se vážící další informace:
  - Rozsazení během zkoušky, kde je na výběr rozsazení studentů vedle sebe, ob jednoho, nebo ob dva.
  - Počet hodin, které je na zkoušku potřeba vyhradit.
  - Počet kol, ve kterých má zkouška probíhat. Kolem je zde myšlen počet skupin, do kterých budou studenti rozděleni, přičemž každá skupina bude mít zkoušku v jiném čase.
  - Preferovaná místnost, či místnosti, pro konání zkoušky.
- Preferované období, pro konání zkoušky je možné upřesnit následujícími údaji:
  - Počáteční a koncové datum
  - Časové rozmezí
  - Den v týdnu
  - Nízká, střední, anebo vysoká priorita požadavku

#### **2. Preference pro výuku**

- Typ výuky a k němu se vážící další informace:
  - Počet hodin potřebných pro výuku daného typu

- Počet skupin, do kterých mají být studenti rozděleni (např. na cvičení)
- Požadovaná kapacita místnosti
- Specifikace týdnů výuky ve smyslu, kolikátý týden má výuka začínat a končit a zdali se má konat každý týden, anebo pouze v lichých, či sudých týdnech
- Preferovaná místnost či místnosti pro výuku
- Období preferované pro výuku nelze, na rozdíl od období preferovaného pro zkoušku, specifikovat za pomoci konkrétního data.

### 3. Omezení

- Druh omezení je možné vybrat z několika předdefinovaných možností. Jsou jimi například zasedání orgánu fakulty, služební cesta nebo zdravotní důvody.
- Období, ke kterému se omezení váže. Konkretizuje se stejným způsobem, jako preferované období pro zkoušku.
- Frekvence opakování omezení udávaná v týdnech.

### 4. Návaznosti (od roku 2022 nově)

- Typ návaznosti (např. navazovat, ve stejný den, předchází)
- Předmět, kterého se návaznost týká (má být navázán na daný předmět, nebo naopak nemá být ve stejný den)
- Typ výuky
- Nízká, střední, anebo vysoká priorita

Příkladem návaznosti může být například to, že by učitel některého předmětu chtěl, aby mu hned na přednášku navazovalo demonstrační cvičení z téhož předmětu.

## Kompletace požadavků

Po úspěšném importu požadavků a dalších potřebných dat z REST API je nutné prověřit validitu požadavků zadaných učiteli. Může se totiž stát, že třeba nesprávně zadali některý z údajů, anebo zadali takové požadavky, které nejsou splnitelné.

Další situací, která často nastává, je ta, že učitelé předpokládají, že jejich požadavky jsou již v systému zadané z minulých let a znovu je nevyplňují. Rozvrhář tedy musí tyto požadavky manuálně načíst a poté k aktuálně plánovanému období přidat pouze ty, které jsou stále platné. Z toho vyplývá, že by rozvrhář měl mít všeobecný přehled o fungování fakulty. Měl by se minimálně částečně orientovat ve struktuře učeben, aby například nedocházelo k schválení požadavku na rozsazení studentů, které není z důvodu počtu míst v učebně možné. Dále by měl mít povědomí o pravidelných školních aktivitách učitelů (kolegia, schůze atd.).

Období, ze kterých jsou požadavky načítány, ukazuje tabulka 2.1, jak ji ve své bakalářské práci [22] navrhla Ing. Tesařová. Jak vypadá tabulka s požadavky ukazuje obrázek 2.3, který konkrétně zachycuje tabulku s požadavky na výuku.

## Plánování

Plánování obecně probíhá tak, že se do rozvrhu nejprve vkládají ty předměty, k nimž je nejvíce požadavků. Ještě předtím se ale musejí do rozvrhu vložit akce, které mají pevně

Typ	Sem.	1. období (modrá)			2. období (fialová)			3. období (žlutá)		
		Sem.	Rok	Typ	Sem.	Rok	Typ	Sem.	Rok	Typ
Výuka	ZS	ZS	min.	výuka	LS	min.	výuka	LS	min.	zk.
	LS	LS	min.	výuka	ZS	akt.	výuka	LS	min.	zk.
Zk	ZS	ZS	min.	zk	LS	min.	zk	ZS	akt.	výuka
	LS	LS	min.	zk	ZS	akt.	zk	LS	akt.	výuka

Tabulka 2.1: Období požadavků, která se používají pro kompletaci. Zkratky: min. – minulý, akt. – aktuální, sem – semestr, LS – letní semestr, ZS – zimní semestr, zk – zkouška

danou dobu konání a tím pádem je není možné umístit do rozvrhu na jiné místo (typicky mezifakultní výuka). Další postup při plánování se již liší v závislosti na tom, zda jde o plánování výuky, či zkoušek.

Při **plánování výuky** se následně přidávají předměty v sestupném pořadí dle počtu zapsaných studentů. Během plánování je nutné kontrolovat, zda jsou splněna minimálně všechna tvrdá omezení. Z tvrdých omezení se kontroluje, zda je výuka naplánovaná tak, aby měl v jeden čas učitel pouze jednu výuku. Dále je hlídáno, aby se studentovi nepřekrývala výuka z povinných předmětů. Tato situace by totiž znamenala, že by se výuky některých předmětů nemohl účastnit. Pamatovat se musí také na požadavky učitelů, zvláště na jimi zadaná omezení, aby nedošlo k naplánování výuky na čas, kdy učitel není přítomen. Kontroluje se také, zda-li se počet studentů zapsaných v předmětu vejde do vybrané místnosti. Tyto kontroly zastává aplikace pro plánování vytvořená Ing. Markem Kuchynkou.

U **plánování zkoušek** se oproti tomu začíná s bakalářskými předměty, z nichž se nejprve vybírají povinné předměty. Předměty se postupně umísťují do rozvrhových oken v sestupném pořadí dle počtu registrovaných studentů. U zkoušek se stejně jako u výuky kontroluje, zda dle rozvrhu nemá být učitel na dvou místech zároveň a zda nejsou porušena omezení učitelů. Dále je kontrolováno také to, zda studenti nemají na jeden den naplánováno více zkoušek z povinných předmětů. V potaz je brán také rozestup mezi jednotlivými zkouškami. Výsledky zkoušky totiž musí být zveřejněny do pěti pracovních dní od konání zkoušky [1]. Neformálním doporučením je zveřejnit výsledky zkoušky alespoň dva dny před konáním dalšího termínu. Veškeré nástroje pro zajištění těchto kontrol opět poskytuje aplikace Ing. Kuchynky.

Tato podkapitola vychází z postupu plánování popsáno v práci Ing. Tesařové [22].

## 2.4 Přejít z IS FIT na IS VUT

IS VUT jako zdroj, odkud jsou data stahována, se stal v roce 2022 náhradou dříve používaného IS FIT. S postupem času by se měl IS VUT stát plnohodnotnou náhradou původního systému.

Přejít na nový systém je realizován na základě rozhodnutí děkana dle doporučení Rady pro využití informačních technologií a vybavení na FIT VUT. Zatím není úplný a jeho dokončení je v horizontu několika let. Již s sebou ale přinesl různé změny, které budou níže v této kapitole zmíněny.

Důvodem k přechodu bylo jednak sjednocení systémů využívaných na jednotlivých fakultách Vysokého učení technického v Brně, ale také nedostatky původního systému. IS FIT se totiž pomalu stával neudržitelným, určité akce musely být prováděny manuálně a začínaly se objevovat problémy s jeho fungováním.

Využívaná datová struktura se se změnou systému dočkala hned několika změn (zpracovány v ER diagramu na obrázku 2.6):

- V IS VUT každým rokem dochází k duplikování programů. To, znamená, že pokud by byl program otevřen např. od roku 2020 do roku 2022, tak v databázi budou pro tento program tři záznamy, každý s jiným rokem. Duplikování programů má za následek duplikování oborů.
- Byly rozšířeny učitelské požadavky o návaznosti a další informace.
- K oboru byla přidána informace, zdali se jedná o hlavní obor či nikoli.
- Byly přidány tabulky **zkouska\_projekt** (používaná při importu rozvrhu zkoušek do IS VUT) a **typy\_hodnoceni** (obsahující typy hodnocení jako je třeba písemný test, projekt, zkouška a jejich zkratky).

## 2.5 Aktuálně využívaný software

Tato kapitola se bude zaměřovat na aktuálně používané aplikace z pohledu využitých technologií, architektury a vlastností. Důležité je také zmínit, že obě aplikace používají stejnou databázi (MySQL). ER diagram, na základě kterého byla databáze vytvořena, bude uveden na konci této kapitoly (obrázek 2.6).

### Aplikace Ing. Aleny Tesařové

Aplikace Ing. Aleny Tesařové je napsána v jazyce PHP s využitím frameworku Nette, který je postaven na návrhovém vzoru MVC<sup>4</sup>. Aplikace využívá objektově relační mapování (ORM<sup>5</sup>), které zajišťuje knihovna Doctrine 2, kterou do Nette integruje knihovna Kdyby/Doctrine. Na straně klienta je používán JavaScript a jQuery a dále také CSS framework Bootstrap.

Tato aplikace se primárně věnuje třem krokům plánování (celý proces ukázán na obrázku 2.1). Její část pro plánování rozvrhů se momentálně nepoužívá. V rámci využívaných částí umožňuje importovat data z REST API, provádět kompletaci požadavků a také importovat rozvrhy do IS VUT. Import rozvrhu zkoušek do IS VUT je prováděn pomocí REST API, pro výuku však toto API na straně IS VUT chybí, a tak je exportován pouze soubor ve formátu JSON, který musí být poté zpracován.

Import dat je prováděn ze stránky, která v tabulce zobrazuje všechny rozvrhy. U všech rozvrhů je možné po kliknutí na první ikonu ve sloupci (mimo sloupec *Rozvrh*) importovat:

- a) kolize (společné studenty v předmětech),
- b) předměty a preference.

V rámci bodu a) se kromě předmětů a preferencí učitelů importují také programy, obory, povinnosti předmětů, požadavky na návaznosti, omezení atd. U rozvrhů pro zkoušky je navíc možné importovat počty studentů, kteří se účastnili jednotlivých termínů zkoušek minulý rok. Barva jednotlivých ikon slouží k informování o tom, zda byla data již importována

<sup>4</sup>Model-view-controller – <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/>

<sup>5</sup>Object-relational mapping – <https://www.altexsoft.com/blog/object-relational-mapping/>

(zelená), nebo k jejich importu musí teprve dojít (červená). Oranžová ikona, která je vidět ve sloupci *Předměty* poukazuje na to, že jsou pro zvolený rozvrh importované předměty, ale preference zatím chybí.

Před spuštěním importu dat je nutné zadat přihlašovací údaje k IS VUT a zvolit, zda chceme vymazat data a nahrát nová, anebo přidat neexistující záznamy a aktualizovat původní data.

Negativem této aplikace, co se importů týče, je jednoznačně jejich rychlost. Čas potřebný pro import dat před plánováním zkoušek, se dle Ing. Dytrycha může i v případě, že během něj nenastane žádná chyba, pohybovat okolo 90 minut.

Název rozvrhu	Předměty   Preference	Kolize	Počty na zk.	Rozvrh
2021 LS - výuka (vyuka 2021 LS)				
2021 LS zkoušky (zkoušky 2021 LS)				
2021 ZS - výuka (vyuka 2021 ZS)				

Obrázek 2.2: Část tabulky s rozvrhy a nabízené možnosti importu

Další část týkající se přípravy plánování je kompletace požadavků. Tabulky na stránkách *Požadavky na výuku* a *Požadavky na zkoušku* zobrazují všechna data, která jsou ohledně požadavků učitelů na výuku či zkoušky evidována. Umožňují v případě potřeby požadavky upravovat nebo přidávat nové. Nově přidané řádky ale nejsou uloženy automaticky. Nejprve je nutné uložit požadavek učitele a až poté preferenci či omezení. Tabulky s požadavky na výuku a zkoušky se dělí na několik částí (naznačeny písmeny na obrázcích 2.3 a 2.4), které budou rozepsány níže. Části *Preference* a *Omezení* mohou být souhrnně označovány jako *požadavky na termín*.

- Informace o předmětu (část A)
- Požadavek učitele (část B)
- Preference (část C)
- Omezení (část D)
- Požadavek učitele na návaznost výuky (část E)

Požadavky na výuku

A										B										C										D										E									
Přec.	Fys	Fys	Pro	Max	Garant	Předmět	Vyučující	Typ	Poc	Kap	Hod	Výdan	Od	Do	Místnost	Poznámka	Den	Od	Do	Pro	Sout.	Navaz.	Prez.	Poznamka	Datum od	Datum do	Den	Od	Do	Úruh	Poznamka	Název st	Priority	Prez	Recoz	Poznámka													
2021	INBY	MIS	11	90	Zkouš	Zkouš	Zkouš	pr	1	20	2	výu			N205																																		

Obrázek 2.3: Kompletace požadavků na výuku

Většinu buněk tabulky je také možné obarvovat, přičemž každá barva má svůj význam. Červená barva slouží k označení nesplněného požadavku a zelená naopak značí splněný požadavek. Sytě růžová barva se používá pro označení požadavků, ze kterých učitelé slevili.

Obrázek 2.4: Kompletace požadavků na zkoušku

Výrazně žlutou barvou se označují předměty, které se neplánují. Trojice barev modrá, fialová a žlutá slouží pro označení požadavků, které byly načteny z minulých období, a světle oranžová barva značí poznámku.

Rozvrhář může buňky obarvit buď privátně, tedy že barvu uvidí pouze on sám, nebo veřejně.

I v případě kompletace požadavků má aplikace nedostatek v rychlosti. Načítání stránek je pomalé a bylo ještě více zpomalené rozšířením tabulek o nová data. Průměrný čas požadovaný pro kompletní načtení stránek *Požadavky na výuku* a *Požadavky na zkoušku* je rozepsán v tabulce 2.2. Během načítání stránek také dochází k opakovanému vypršení limitu pro načtení stránky (zvláště u stránky *Požadavky na výuku*).

Požadavky	Zobrazení
Zkoušky ZS 2021/2022	54 s
Zkoušky LS 2021/2022	46 s
Výuka ZS 2021/2022	114 s
Výuka LS 2021/2022	107 s

Tabulka 2.2: Rychlost zobrazení některých stránek s požadavky. Zkratky: ZS – zimní semestr, LS – letní semestr.

Aplikace řeší i správu příznaků a jejich generování. Příznaky pro předměty a přednáškové skupiny jsou zobrazené v tabulce, kde v záhlaví jsou přednáškové skupiny a v prvním sloupci předměty. Barevně jsou označeny řádky s těmi předměty, jejichž příznaky byly minulý rok manuálně změněny. Možnost generování příznaků je „schovaná“ v okně, které se zobrazí po kliknutí na ikonu nápovědy. Příznaky se zde generují na základě pravidel, která jsou zakotvena ve zdrojovém kódu a některé příznaky se negenerují správně.

## Aplikace Ing. Marka Kuchynky

Jelikož budou nové moduly, jejichž implementace je cílem této práce, přímo integrovány mezi ostatní části zdrojového kódu aplikace Ing. Marka Kuchynky, bude v této části mimo jiné popsána i struktura existujících zdrojových kódů.

Aplikace Ing. Kuchynky nepoužívá žádný PHP framework, ale pouze samotné PHP, konkrétně verzi PHP 7.4. Na straně klienta používá JavaScript společně s knihovnami jQuery a jQuery UI.



Zdrojové kódy jsou strukturované a rozdělené do několika složek dle toho, jakou funkciionalitu soubory ve složkách obsažené zajišťují. Práce s databází MySQL je řešena v rámci **repozitářů**. Jedná se o třídy obsahující metody, které k práci s databází využívají rozhraní PDO<sup>6</sup>. V případě, kdy se v rámci repozitářů získávají data, jsou tato data ukládána v rámci instancí tříd definovaných pro uchovávání konkrétních dat. Z důvodu potřeby rychlých reakcí na uživatelské vstupy je část logiky implementována na straně klienta [11]. Tyto části implementace jsou poskytovány několika předdefinovanými třídami jazyka JavaScript a jejich metodami. Na straně klienta je realizováno např. vykreslování rozvrhu a některé kontroly dodržení tvrdých a měkkých omezení. Z důvodu potřeby pracovat s databází bez nutnosti aktualizace stránky je součástí implementace i jednoduché **API**. K jeho asynchronnímu volání se používá AJAX. Jednotlivé žádosti jsou směrovány na jeden skript v jazyce PHP, který dle předaných parametrů zavolá příslušné funkce implementované v jazyce PHP, které následně žádost zpracují. Po dokončení zpracování je klientovi vrácena odpověď.

K celému plánování slouží pouze jediná stránka, která v sobě obsahuje mnoho různých nástrojů, které rozvrhářům jejich práci usnadňují. Rozvrh je možné verzovat a lze se i pohybovat tam i zpět napříč historií změn oken vložených do rozvrhu. Při plánování automaticky dochází ke kontrolám tvrdých a měkkých omezení, které již byly dříve definovány. Porušení některého omezení je indikováno barevným trojúhelníkem v pravém horním rohu rozvrhového okna. Obrázek 2.5 několik takových porušení zachycuje. Aplikace rozlišuje různé závažnosti problémů:

- Méně závažné problémy jsou značeny fialovou barvou. Do této kategorie se řadí např. porušení preferencí učitelů.
- Mezi středně závažné problémy značené žlutě patří kupříkladu porušení požadavku na délku zkoušky či výuky.
- Závažné problémy na sebe upozorňují červenou barvou a k těmto problémům se řadí zejména porušení tvrdých omezení. Ohlášena je tedy např. situace, kdy dojde k porušení omezení učitele.

Dochází tedy k detekování různých problémů, ale aplikace poskytuje i sadu nástrojů umožňující jejich předcházení, či jinak usnadňujících plánování. Společnými nástroji pro plánování výuky a zkoušek jsou jmenovitě nástroje: *Zbývá naplánovat*, *Požadavky*, *Kapacity*, *Kolize* a *Tranzitivní kolize*. Pro plánování zkoušek se navíc používají i nástroje *Rozestupy* a *Zkoušky v týdnu*.

Nástroj *Zbývá naplánovat* má informativní charakter. Zobrazuje totiž počty studentů přihlášených v předmětech a také to, kolik hodin jednotlivých typů výuky ještě zbývá naplánovat. *Požadavky* umožňují zobrazit požadavky na výuku, či zkoušky od jednotlivých vyučujících předmětu. Právě v této části se zobrazují i textové poznámky, které mohou rozvrhářům v některých případech výrazně pomoci. *Kapacity místností* zase informují o tom, kolik procent potřebných míst pro realizaci výuky či zkoušky ze zadaného předmětu poskytnou vybrané místnosti. Nástroj *Kolize* zobrazuje počty společných studentů pro zadané předměty. *Tranzitivní kolize* slouží k podobnému účelu. Zobrazují se kombinace předmětů, které mají společné studenty a také počty těchto studentů. Nástroj *Zkoušky v týdnu* zobrazuje počty studentů, kteří by dle aktuálního rozvrhu měli zkoušku z více předmětů (jsou vypsaní) v jednom týdnu. A nakonec, nástroj *Rozestupy* zobrazuje odhadované počty studentů na jednotlivých termínech zkoušky spolu s odhadovaným počtem zkoušek, které bude

<sup>6</sup>PHP data objects – <https://www.php.net/manual/en/book.pdo.php>

nutné ohodnotit za hodinu pracovní doby, aby bylo možné výsledky zveřejnit dva dny před dalším termínem zkoušky.

Z popisu jednotlivých částí aplikace, které jsou používány pro zajištění realizovatelnosti rozvrhu, vyplývá, že je vždy zapotřebí mít aktuální data. Jelikož aplikace kontroluje i požadavky zadané učiteli, je zde potřeba určité předzpracování, aby byla zajištěna jejich korektnost a byly tak pro plánování skutečně prospěšné. A právě tyto dva faktory zajišťuje aplikace Ing. Tesařové, bez které by plánování v aplikaci Ing. Kuchynky nebylo možné.

The screenshot displays the 'Rozvrhy IS' application for the 2021 ZS semester. The main view is a timetable for course ID 1365, showing a weekly schedule from 7:00 to 20:00. The timetable is color-coded by subject: green for IEL, blue for IUS, yellow for IPT, and red for IZP. A sidebar on the right lists subjects and their respective student counts (Společně studenti). The interface includes various filters and controls at the top, such as 'Akce s rozvrhem' and 'Filtr: Místnosti'.

Obrázek 2.5: Plánování rozvrhů v aplikaci Ing. Kuchynky

## 2.6 Požadavky na nové moduly

Analýza požadavků na nové moduly vychází z nedostatků stávající aplikace pro podporu plánování rozvrhů. Nedostatky byly definovány při kompletaci požadavků pro zkoušky v zimním semestru 2021, které se autorka účastnila.

### Rychlost

Jak již bylo zmíněno dříve, velkým nedostatkem aktuálně používané aplikace je právě rychlost. Při implementaci modulů by tedy měl být kladen velký důraz na:

- rychlost zobrazení dat** uživateli a urychlení importu dat z REST API – při zobrazování rozsáhlých tabulek by nemělo docházet k vypršení časového limitu pro načtení stránky,
- urychlení importu dat** z REST API – čas potřebný pro import by se měl být co nejkratší. Časové nároky dosud využívané aplikace jsou příliš velké, zvláště v případě, když během importování dat dojde k chybě a import se přeruší.



## Spolehlivost

Moduly požadavků na výuku a zkoušky by dále měly poskytovat funkci automatického ukládání dat. Data v tabulkách je potřeba automaticky ukládat nejen při změnách již existujících buněk, ale též po vložení dat do buněk v nově přidaném řádku. Nové požadavky by tak nemusely být manuálně ukládány s nutností uložení jednotlivých částí ve specifickém pořadí.

Takto by mělo být převedeno situaci, kdy rozvrhář před aktualizací stránky nebo ukončením přípravy pro plánování rozvrhu zapomene uložit nová data a tím pádem o ně přijde. Rozvrhář by také již neměl narážet na upozornění, že požadavek na termín nelze uložit před uložením samotného požadavku pro předmět.

## Použitelnost

Je také třeba zajistit, aby při každém plánování nebylo nutné procházet a opravovat větší množství příznaků k předmětům. Až na výjimky (předměty, které je možné si v určitých případech zapsat o rok dříve, a předměty, které si nelze zapsat ve vyšších ročnících), by měly být příznaky správně vygenerovány na základě uživatelsky nastavitelných pravidel.

Díky pravidlům, která by nebyla zakotvena ve zdrojovém kódu, by bylo možné generování příznaků snadno přizpůsobit např. nově vzniklému studijnímu programu bez nutnosti změny implementace.

## Přehlednost

S novými daty vážícími se k požadavkům učitelů na výuku a zkoušky dojde k rozšíření již tak rozsáhlých tabulek na stránkách s výpisem požadavků. Je třeba zajistit, aby tyto tabulky zůstaly i nadále v rámci možností přehledné. Nemělo by se stát, že buňky, které mají obsahovat větší množství textu, budou na výšku příliš roztažené kvůli jejich omezené šířce.



## Kapitola 3

# Využití technologie

Technologie, které budou využity při implementaci aplikace jsou z větší části dány technologiemi, které byly použity v rozšiřované aplikaci Ing. Marka Kuchynky, o které hovoří sekce 2.5.

### 3.1 HTML a CSS

Značkovací jazyk HTML (Hypertext markup language) je používán pro tvorbu webových stránek. Konkrétně slouží k definování významu obsahu stránek. Vznikl na počátku 90. let 20. století a jeho autorem je Tim Bernes-Lee, který založil a aktuálně stále vede konsorcium W3C [7]. Toto konsorcium se společně se skupinou WHATWG věnuje tvorbě specifikace HTML. Aktuální verzí je HTML5 (specifikace HTML Living Standard [24]).

Zatímco se HTML, jak již bylo uvedeno, používá k popsání obsahu webové stránky, k určení vzhledu stránky slouží jazyk CSS (Cascading Style Sheets). Koncept CSS se poprvé objevil v roce 1994, kdy jej publikoval Håkon Wium Lie [23]. Dva roky poté, v roce 1996, byla konsorciem W3C standardizována první verze CSS1 [12]. Od té doby se standard CSS postupně vyvíjel a jeho nejnovější verzí je CSS3.

### 3.2 JavaScript

JavaScript je slabě typovaný událostmi řízený multiparadigmatický skriptovací jazyk. V roce 1995 jej pod názvem Mocha vytvořil Brandan Eich, který byl za najat společností Netscape. Později byl název tohoto jazyka změněn právě na JavaScript. V roce 1997 byl společností Ecma International vytvořen standard ECMAScript a JavaScript se tak stal pouze jednou z jeho implementací. Uvedené informace o historii tohoto jazyka byly čerpány z: [10]. Nejnovější verzí je ECMAScript 2021 a jedná se již o 12 verzi [15].

Dříve byl tento jazyk používán převážně na straně klienta, kde zajišťoval interaktivitu a změny webové stránky bez nutnosti načítat data ze serveru. Existují ale i řešení, která využívají JavaScript na serverové straně, např. Node.js<sup>1</sup>

V rámci implementace také budou využity dvě knihovny založené na jazyce JavaScript, kterými jsou **jQuery** a **jQuery UI**. Knihovna jQuery usnadňuje manipulaci s objektovým modelem dokumentu a jeho procházení. Dále poskytuje funkce pro správu událostí, animace a také jednoduché API pro používání technologie AJAX (Asynchronous JavaScript and XML). AJAX umožňuje asynchronní komunikaci se serverem. Po odeslání žádosti

---

<sup>1</sup><https://nodejs.org/en/about/>

na server tedy prohlížeč nečeká na odpověď a při příchodu odpovědi nemusí být stránka aktualizována.

JQuery UI je knihovna stavící na jQuery a zaměřující se na uživatelské rozhraní. Přináší různé efekty použitelné při interakci uživatele s prvky stránky, způsoby interakce (např. přesun elementu myší po stránce), sadu ikon a několik nastavitelných prvků (angl. *widgets*).

### 3.3 PHP

PHP je slabě typovaný jazyk využívající dynamickou typovou kontrolu. V roce 1994 jej vytvořil Rasmus Lerdorf za účelem sledování návštěv svého online životopisu. Jazyk PHP byl postupně rozšiřován a v červnu 1995 byly zveřejněny zdrojové kódy. Aktuálně jej vyvíjí skupina *The PHP Group*. [18] Nejnovější verzí, vydanou 25. listopadu 2021, je verze PHP 8.1. Protože je cílem práce rozšířit implementaci aplikace Ing. Marka Kuchynky, bude místo nejnovější verze využita verze PHP 7.4, kterou tato aplikace používá.

Jedná se o multiplatformní jazyk, který je dle [20] podporován vícero webovými servery. Kromě Apache, jednoho z nejpopulárnějších webových serverů, jej podporují např. i nginx nebo další webové servery, který podporují standard CGI<sup>2</sup>.

### 3.4 MySQL

MySQL je otevřený systém řízení báze dat (SŘDB). Vznikl z databázového systému Unireg, ke kterému v roce 1995 jeho autor Michael Widenius připojil rozhraní SQL [6]. V tomtéž roce založili Michael Widenius, David Axmark a Allan Larsson společnost MySQL AB, která MySQL vyvíjela [6]. Aktuálně se vývojem MySQL zabývá společnost Oracle [9]. Nejnovější verzí je MySQL 8. Tato verze MySQL nebude využita z důvodu, že produkční databáze využívá starší verzi – MySQL 5.7.

MySQL podporuje více typů úložišť, které se dají obecně rozdělit na transakční a netransakční. Produkční databáze používá transakční úložiště *InnoDB*. Toto úložiště podporuje kromě transakcí i uzamykání na úrovni řádků při souběžném přístupu a použití cizích klíčů. Tento odstavec byl převzat z: [5]

---

<sup>2</sup>Common Gateway Interface – protokol pro propojení externích aplikací s webovým serverem

## Kapitola 4

# Návrh nových modulů aplikace

Tato kapitola se bude zabývat návrhem modulů pro přípravu plánování a návrhem jejich integrace do aplikace Ing. Marka Kuchynky. Podkapitola 4.1 bude pojednávat o změnách, které bude nutné provést v databázi. Na ni poté naváží další podkapitoly věnující se návrhu jednotlivých modulů.

### 4.1 Úpravy databáze

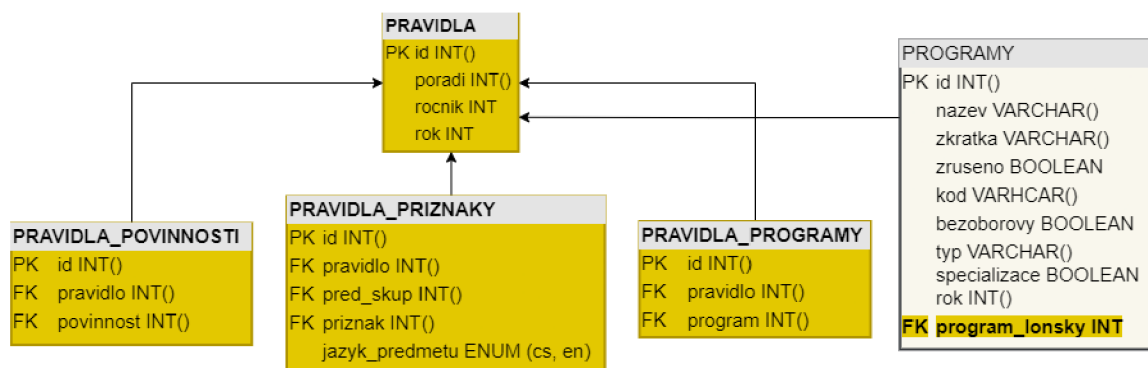
Existující databázovou strukturu bude stačit rozšířit o tabulky pro ukládání pravidel pro generování příznaků. Dále dojde k přidání několika nových atributů do již existujících tabulek. Všechny změny jsou v ER diagramu databáze 2.6 označeny výrazně žlutou barvou. Nově přidané tabulky pro ukládání pravidel pro generování příznaků jsou znovu uvedeny níže (obrázek 4.1).

Hlavní část tvoří tabulka **pravidla**, na kterou je navázáno několik dalších tabulek: **pravidla\_povinnosti**, **pravidla\_priznaky** a **pravidla\_programy**.

Atribut *poradi* tabulky *pravidla* určuje, v jakém pořadí bude pravidlo při generování příznaků aplikováno. Kvůli duplikaci programů pro každý rok je v této tabulce přidán i atribut *rok*, který určuje, pro jaký rok je pravidlo platné.

Propojovací tabulky *pravidla\_povinnosti* a *pravidla\_programy* určují, na jaké programy s jakou povinností předmětů v jejich oborech, bude pravidlo aplikováno.

Atributy *pred\_skup* a *priznak* tabulky *pravidla\_priznaky* určují, které přednáškové skupině bude přiřazen jaký příznak. Pro omezení přiřazení příznaku pouze na předměty vyučované v určitém jazyce slouží atribut *jazyk\_predmetu*.



Obrázek 4.1: Změny ER diagramu týkající se pravidel pro generování příznaků

S těmito tabulkami bude možné zachytit pravidla, která jsou rozepsána v tabulce 4.1. Pokud by ovšem byla databáze ponechána v tomto stavu, bylo by nutné kvůli již zmíněné duplikaci programů vytvářet pro další rok pravidla nová. Z toho důvodu byl do tabulky **programy** přidán atribut *program\_lonsky*, který bude obsahovat identifikátor programu pro minulý rok. Tento atribut bude tedy pro program *Informační technologie a umělá inteligence (MITAI)* pro rok 2021 obsahovat identifikátor programu *Informační technologie a umělá inteligence (MITAI)* pro rok 2020. Díky tomu bude možné automatické vytvoření pravidel platných pro další roky.

Poslední provedenou úpravou struktury databáze bylo doplnění atributu *predmet* do tabulky **pozad\_na\_navaznost\_vyuk**, bez kterého by nebylo přesně jasné, jaká výuka má navazovat. Učitel totiž může chtít, aby mu cvičení navazovalo na přednášku ze stejného předmětu, ale také může chtít na přednášku z jednoho předmětu navázat přednáškou z jiného.

Prog.	Povin.	Roč.	Jazyk	Před. skup.	Příznak
Bc.	P	-	-	1BIA, 1BIB, 2BIA, 2BIB, 3BIT	oddo
Bc.	P	1	-	1BIA, 1BIB	oddo
			-	2BIA, 2BIB	xx
Bc.	P	2	-	2BIA, 2BIB	oddo
			-	3BIT	xx
Bc.	P	3	-	3BIT	oddo
Bc.	PV, V	1	-	1BIA, 1BIB, 2BIA, 2BIB, 3BIT	xx
Bc.	PV, V	2	-	2BIA, 2BIB, 3BIT	xx
Bc.	PV, V	3	-	3BIT	xx
NMgr.	P	-	-	1MIT, 2MIT	obor
			en	1EIT, 2EIT	xx
NMgr.	P	1	-	1MIT	obor
			-	2MIT	xx
			en	1EIT	obor
			en	2EIT	xx
NMgr.	P	2	-	2MIT	obor
			-	1MIT	xx
			en	2EIT	obor
			en	1EIT	xx
NMgr.	PV, V	-	-	1MIT, 2MIT	xx
			en	1EIT, 2EIT	xx
Erasmus	P, PV, V, D	-	en	INTE	xx
Ph.D.	P, PV, V	-	-	1DVI	xx

Tabulka 4.1: Pravidla pro generování příznaků. Znak „-“ značí libovolný ročník, nebo jazyk.

## 4.2 Návrh modulu pro správu období

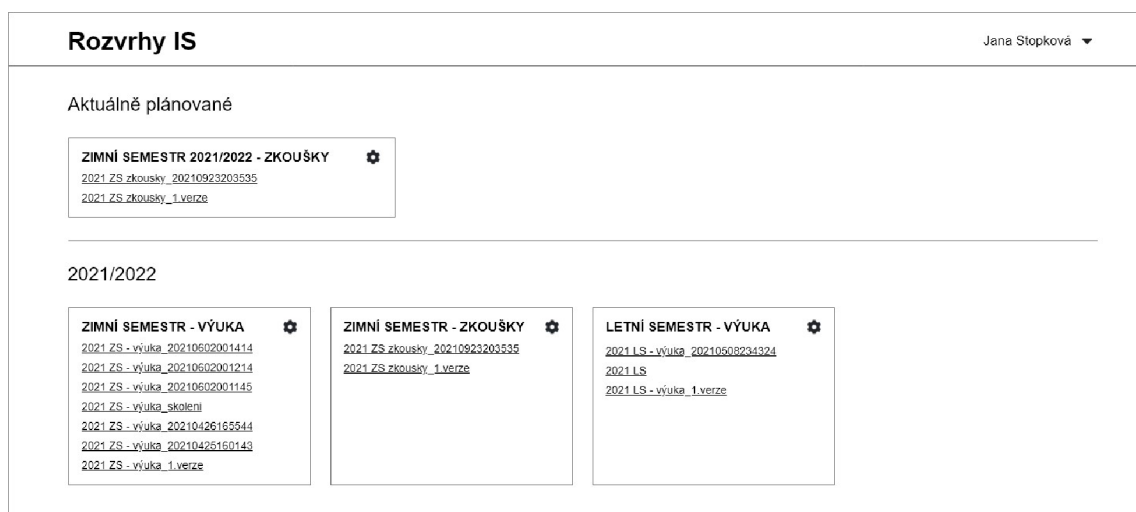
První z hlavní stránky viditelnou úpravou budou nové ikony pro přístup ke správě období, které budou přiřazeny ke každému ze zobrazených období. Přístup ke správě období bude samozřejmě omezen a přidávané ikony budou viditelné pouze pro administrátory. Takovýto



způsob přístupu ke správě období, odkud jsou dostupné jednotlivé implementované moduly, je rychlý a jednoduchý. Navrženou úpravu hlavní stránky zobrazuje obrázek 4.2.

Samotná stránka správy období si zachová přibližně takovou podobu, jakou jí dala Ing. Alena Tesařová. Nově zde ale přibude tlačítko pro úpravu zvoleného období. Bude možné zvolit, pro jaké podobdobí (výuka a zkoušky) má být úprava provedena. Dostupné úpravy budou ale omezené jen na změnu začátku a konce zvoleného období. Semestr, akademický rok, společně s typem podobdobí, by neměly být měněny z toho důvodu, že definují, o jaké období se jedná. Změna těchto hodnot by zapříčinila nesprávnou interpretaci dat a tím pádem špatné chování aplikace.

Součástí správy období by také měla být možnost vytvořit období a k němu rozvrh. Tato část by měla být dostupná z nově přidaného menu aplikace.



Obrázek 4.2: Návrh zakomponování přístupu ke správě jednotlivých období na hlavní stránku

### 4.3 Návrh modulu pro správu požadavků na výuku a zkoušky

Správa učitelských požadavků na výuku je prováděna na stránce, na níž se nachází tabulka, která požadavky zobrazuje. Stejně tak je tomu u požadavků na zkoušky. Obě tyto tabulky bude nutné rozšířit tak, aby obsah tabulek reflektoval změny ve zdrojových datech. V tabulce požadavků na zkoušku se nově objeví priorita preference, počet současně probíhajících bloků zkoušek, počet bloků navazujících na jiný blok, přestávka mezi navazujícími bloky udávaná v minutách a také druh omezení a frekvence opakování omezení. K požadavkům na výuku bude přidán počáteční a koncový týden výuky, počet současně probíhajících bloků výuky, počet bloků navazujících na jiný blok, přestávka mezi navazujícími bloky, druh omezení, frekvence opakování omezení a pět dalších sloupců pro návaznosti výuky. S takovýmto nárůstem počtu sloupců v tabulkách již přestává být realizovatelné, aby se tabulka, co se týče šířky, vešla na stránku celá. Z tohoto důvodu budou nové sloupce implicitně skryté a pro jejich zobrazení bude nutné stisknout tlačítko. Po zobrazení skrytých sloupců se tabulka rozšíří tak, že její šířka bude převyšovat šířku stránky.

Další navržená úprava se týká zobrazování barev jednotlivých buněk tabulky. Aktuálně uživatel vždy vidí jak veřejné, tak privátní barvy, přičemž privátní barva má přednost před

veřejnou. Takto není jasné, které barvy vidí jen on sám a které jsou viditelné i pro ostatní rozvrháře. Z tohoto důvodu by měla přibýt **nová zobrazení barev**. Pokud bude mít uživatel nastavené veřejné obarvování buněk (značí ikona otevřeného zámku), uvidí pouze veřejné barvy. V případě privátního obarvování buněk (značí ikona zavřeného zámku) bude mít na výběr mezi dvěma zobrazeními

- Pouze privátní barvy
- Veřejné i privátní barvy, přičemž privátní barvy opět budou mít při zobrazování přednost, ale budou navíc nějakým způsobem zvýrazněny, aby je bylo snadné odlišit od veřejných barev.

Změnou by také mělo projít ukládání požadavků v nových řádcích. Nebude již nutné každý nově přidaný řádek ukládat kliknutím na ikonu připínáčku. Po vložení první hodnoty do libovolné buňky bude tato hodnota ihned uložena do databáze a při úpravách dalších buněk téhož řádku už bude pouze docházet k aktualizaci existujícího záznamu.

Vylepšeno bude i načítání požadavků z minulých období. Aktuálně se požadavky načítají ze tří období (detaily v tabulce 2.1). Po úpravě by stále měla být využívána stejná období, ale například k preferencím na výuku nebudou načítány preference na zkoušku. Rovněž se nebudou načítat požadavky, které mají uvedené počáteční a koncové datum a rozsah těchto dat nezasahuje do plánovaného období. Tímto způsobem budou rozvrháři nabízeny relevantnější požadavky.

## 4.4 Návrh modulu pro správu příznaků k předmětům

Způsob zobrazení příznaků přiřazených jednotlivým předmětům a přednáškovým skupinám zůstane stejný, tzn., že přednáškové skupiny budou v záhlaví tabulky a předměty v prvním sloupci. Na stránku dále bude nutné umístit tlačítko, po jehož stisknutí se otevře okno, ve kterém bude možné spravovat pravidla pro jejich generování.

Každé jedno pravidlo se bude skládat ze šesti částí: studijní program, povinnost předmětu, ročník, jazyk výuky předmětu, přednášková skupina a přiřazovaný příznak. V částech program a přednášková skupina bude možné zvolit více hodnot.

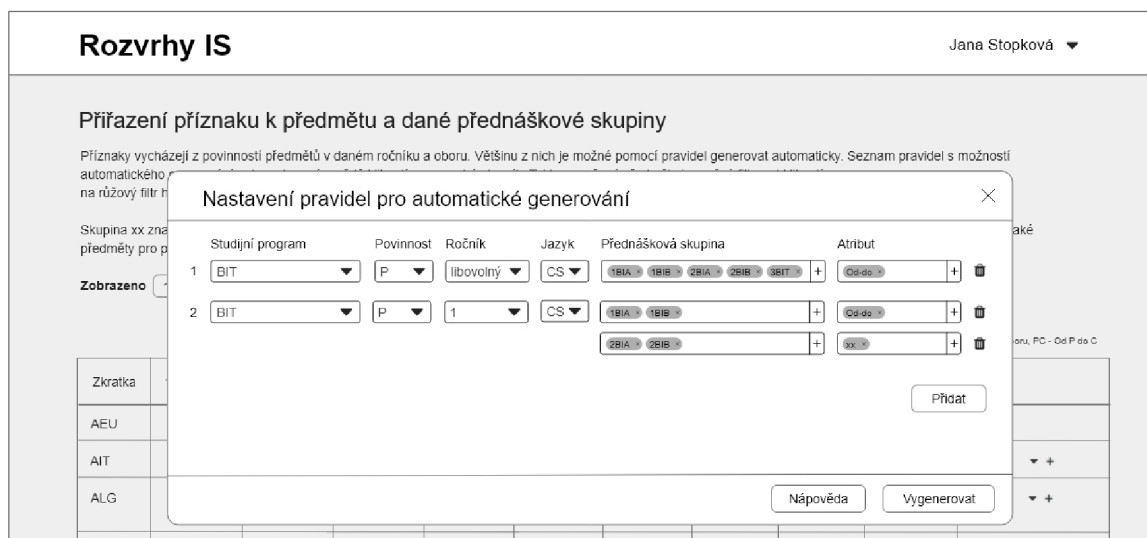
V rámci pravidla bude ke každé množině  $A = \{\text{studijní programy, povinnost, ročník}\}$  možné přiřadit více množin  $B = \{\text{jazyk, přednáškové skupiny, příznaky}\}$ . Pokud by tomu tak nebylo, docházelo by ke zbytečnému duplikování hodnot z množiny  $A$ .

Součástí tvorby pravidel je také jejich uspořádání. Jako nejvhodnější způsob, jak toto realizovat, bylo zvoleno přesunutí pravidla nahoru či dolů způsobem „drag and drop“ („chycením“ řádku pomocí myši a následným přesunutím do cílového místa).

Změna pořadí pravidel bude do databáze ukládána automaticky po přesunutí pravidla. Ostatní úpravy v jednotlivých částech pravidla ale budou muset být ukládány manuálně. Pokud by totiž uživatel nastavil nejprve přednáškové skupiny, dosud by neexistovalo pravidlo, ke kterému by vybrané skupiny byly přiřazeny. Způsobem, jak dosáhnout automatického ukládání, by mohlo být vytvoření pravidla s implicitně přiřazenými hodnotami. To ale není příliš dobrým řešením, protože v případě, že by uživatel další části pravidla nevyplnil a pravidlo nesmazal, zůstala by v databázi nesprávná data. Absence automatického ukládání bude tedy kompenzována alespoň podbarvením upravené části pravidla do té doby, než jej uživatel uloží.



Pokud v následujících letech nedojde ke změnám pravidel, bude je stačit vytvořit pouze pro jeden rok a do dalších let se budou pouze převádět, jak o tom hovoří podkapitola 4.1. Převod pravidel bude realizován pomocí tlačítka.



Obrázek 4.3: Návrh způsobu definování pravidel pro generování příznaků

## 4.5 Návrh modulu pro import dat z REST API

Část uživatelského rozhraní týkající se importu dat bude přepracována pravděpodobně nejvíce ze všech implementovaných modulů. Všechna data, která proces plánování vyžaduje, bude možné importovat z jedné stránky. Uživatel tak bude mít vše na jednom místě a přesně si nastaví, co bude chtít importovat. U všech částí bude možné nastavit, zda chceme provést aktualizaci záznamů (případně přidání nových záznamů), odstranění záznamů a nahrání nových, pouze odstranění, nebo nechceme provést nic.

Aby měl uživatel přehled o tom, jak import více zvolených částí postupuje, bude na stránce zobrazen i ukazatel průběhu importu jednotlivých částí. Dále bude během importu dat docházet k zaznamenávání chyb, které nenaruší proces natolik, že by v něm nebylo možné pokračovat. Zaznamenané chyby budou po dokončení importu uživateli předloženy ve formě stáhnutelného textového souboru obsahujícího detaily o nastalých problémech.

Vzhledem k tomu, že je jedním z cílů této práce redukce času potřebného k importování dat, zaměří se následující část této kapitoly právě na tuto problematiku. Konkrétněji na faktory ovlivňující rychlost ukládání dat do databáze, respektive na to, co ovlivňuje proces importu dat od části, kdy jsou do proměnných načtena všechna data, až do části, kdy jsou všechna tato data uložena do databáze.

Rychlost operace `INSERT` je ovlivňována různými faktory a existují různé způsoby, jak tuto operaci urychlit [17, 13]. Některé z těchto faktorů byly prověřeny několika pokusy měření. Výsledky tohoto měření ukazuje tabulka 4.2, pod kterou jsou jednotlivé testované varianty blíže popsány. Uvedené časy jsou průměrem ze šesti pokusů.

Pro veškeré testované varianty posloužila jako úložiště jednoduchá tabulka obsahující primární klíč, dva cizí klíče, tři atributy typu `INT` a jeden atribut typu `VARCHAR` s délkou omezenou maximálně na 10 znaků. Tato testovací tabulka svým rozložením odpovídá tabulce `ma_registrovane` z 4.2. V rámci všech pokusů bylo do tabulky vkládáno 20000 řádků.

Pokud není v tabulce 4.2 uvedeno jinak, bylo vkládání dat do databáze prováděno v rámci transakce, naráz byl vkládán pouze jediný řádek a vše proběhlo v rámci jednoho spojení s databází.

Test č.	Popis	Vložení dat
1	Referenční test	3,799 s
2	Bez využití transakce	11,419 s
3	Vytváření nového spojení po vložení každého řádku	>15 min
4	Odstranění cizích klíčů	3,349 s
5	Vložení 18000 řádků v jednom dotazu s následným vložím zbylých 2000 řádků	<b>0,532 s</b>

Tabulka 4.2: Výsledky testů měření rychlosti operace INSERT

V rámci pokusu č. 2 se ukázalo, že využívání transakcí dokáže, kromě zajištění integrity dat, také zrychlit ukládání dat do databáze. Odstranění cizích klíčů v testu č. 4 ukázalo určité zlepšení, v rámci prováděných testů ale nepříliš významné. Z hlediska implementace by navíc takovýto zásah do struktury databáze nebyl realizovatelným vzhledem k tomu, že používaná databáze je sdílená s jinými aplikacemi. Test č. 3, v rámci kterého bylo vyzkoušeno opakované vytváření a ukončování spojení s databází, ukázal, že navázání co nejmenšího počtu spojení je správná cesta.

Poslední test, tedy test č. 5, představuje variantu, která bude použita při samotné implementaci. Vkládání většího množství řádků v jednom jediném dotazu již má určitý pozitivní vliv na rychlost ukládání dat.

Při optimalizaci ale také bude nutné brát na zřetel určité limity, na které by se mohlo narážet:

- Maximální množství paměti, kterou může proces použít je určeno v konfiguračním souboru `php.ini` v položce `memory_limit`. Na produkčním serveru je momentálně nastavena hodnota 16000M.
- Délka řetězce v PHP je pro 32 bitové sestavy omezena na 2 GB, u 64 bitových sestav žádný limit délky řetězce neexistuje.
- Maximální délka jednoho SQL dotazu je daná hodnotou položky `max_allowed_packet` v konfiguračním souboru databázového serveru. Na produkčním serveru je nastavena hodnota 16M.

## REST API

Nové REST API umožňuje pomocí metody GET získávat širokou škálu různých dat, ale ne všechna poskytovaná data budou při implementaci modulů využita. V příloze A budou uvedeny pouze ty adresy, které budou použity pro import jednotlivých částí potřebných dat. Z některých URL bude nutné číst data opakovaně a měnit pouze parametry. Pro získání dat z kterékoli uvedené URL musí být uživatel nejprve autentizován a autorizován.

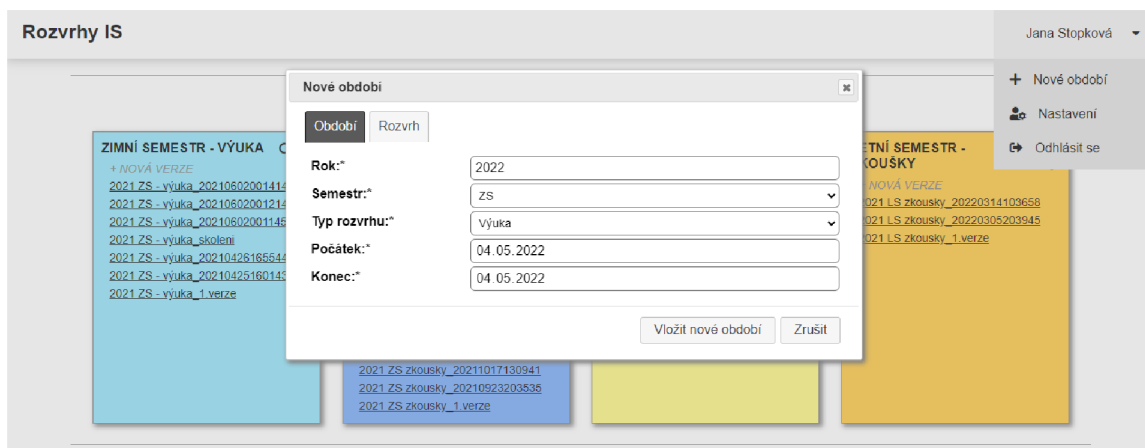
# Kapitola 5

## Implementace

Moduly informačního systému pro plánování rozvrhů byly implementovány dle předem vytvořeného návrhu (kapitola 4). Tato kapitola stručně jednotlivé implementované moduly popisuje a detailněji se zmiňuje o těch částech řešení, které by pro čtenáře mohly být zajímavé.

### 5.1 Správa období

Na hlavní stránku aplikace byly k jednotlivým obdobím přidány ikony pro přístup ke správě období přesně dle návrhu 4.2. Z menu aplikace je nyní dostupná možnost přidání období (obrázek 5.1), se kterým je možné případně rovnou vytvořit nový prázdný rozvrh.



Obrázek 5.1: Okno pro tvorbu nového období

Oproti návrhu byla na stránku se správou období přidána ještě další funkcionalita. Kromě zobrazení názvů rozvrhů patřících k vybranému období a možnosti přejít na jejich editaci je z této stránky nově možné přidat nový rozvrh a verzi vybraného rozvrhu. Dále se odsud dá změnit nastavení vybraného rozvrhu (název a texty, které se zobrazují např. před a za rozvrhem při jeho zveřejnění).


## 5.2 Požadavky na výuku a zkoušky

Implementované moduly pro správu požadavků na výuku a zkoušky poskytují v základu stejnou funkcionalitu jako v aplikaci Ing. Tesařové. Dále bylo zapracováno na nedostacích zmíněných v návrhu a byly také přidány kontroly uživatelských vstupů.

Pro tabulky zobrazující požadavky jsou dostupné dva typy zobrazení: základní a rozšířené. V základním zobrazení tabulka obsahuje ty sloupce, které byly využívány před přechodem na IS VUT, a ostatní sloupce jsou skryté. Toto zobrazení je implicitní. Rozšířené zobrazení uživateli nabízí pohled na všechny sloupce tabulky za cenu, že se tabulka roztáhne do šířky a vybočuje z toku stránky.

Co se týče práce s daty v tabulce, k ukládání dat dochází při změně jakékoli buňky. Při přidání nového požadavku učitele k předmětu a vyplnění např. jeho preferencí již není nutné řešit, že je potřeba nejprve uložit požadavek a až poté preferenci. V tomto směru je uživateli usnadněna práce a nemusí na manuální ukládání vůbec myslet. Jediná situace, kdy uživatel musí kliknout, aby uložil nějaká data, je přidání požadavků z minulých období. Části tabulky obsahující data načtená z minulých období jsou od ostatních požadavků vizuálně odlišena šedým textem. Jejich další odlišností je to, že je není možné upravit do té doby, než jsou uložena prostřednictvím ikony připínáčku. Zobrazené požadavky z minulých období, které rozvrhář nebude chtít uložit, je možné odebrat pomocí ikony koše. K odfiltrování načtených požadavků se stejným obsahem byla použita jednoduchá funkce převzatá z [19].

Implementované řešení také nabízí **rozšířené možnosti zobrazování barev**. V režimu privátního zobrazení barev se současně vybranou možností *Všechny barvy* je možné privátní a veřejné barvy snadno odlišit. Veřejné barvy jednak není možné odstranit, o čemž uživatele informuje i vyskakovací hláška v případě, že se u některé z těchto buněk pokusí barvu odebrat. Hlavním vizuálním způsobem odlišení privátních a veřejných barev je ale to, že pozadí veřejně obarvených buněk je pruhované, což je možné vidět na obrázku 5.2. U privátně obarvených buněk má jejich pozadí jednolitou barvu.

Zobrazit všechny sloupce Paleta  Všechny barvy

Předmět	Zkouší	Roz	Poznámka	Vyhovoval by mi termín					Nemohu zkoušet												
				Termín	Hod	Kol	Pref. místnost	Datum od	Datum do	Den	Čas od	Čas do	Poznámka	A	Datum od	Datum do	Den	Čas od	Čas do	Poznámka	A
IZO (S18 stud) Španěl Michal Ing., Ph.D. Milet Tomáš, Ing.	Španěl Michal Ing., Ph.D.	ob0	☺	vše	2	1	D105 - D0206 + D0207 (dohromady dostatečná kapacita pro IZO)	2020-05-11	2020-05-15	-	07:00	20:50	1 termín by bylo lepší plánovat spíše ke konci prvního týdne nebo na začátku druháho týdne zkouškového (já mám to čas na ukončení hodnocení projektů)	+			Út	08:00	09:50	Graph meeting	+
				vše	2	1	U105? (bohatě by měla stačit kapacita do 200 míst)	2020-05-20	2020-05-29	-	07:00	20:50	Na opravě 1 termínu prosím počítejte alespoň 1 týden	+			Po	07:00	18:50	Prosím přednášky mi naplánujte mimo pondělí, díky.	+
IZU (481 stud) Zbořil František doc. Ing. Ph.D. Zbořil František, doc. Ing. CSc.	Zbořil František, doc. Ing., Ph.D.	ob0	☺	Nekolidovat s termíny v FRL, dostatečné (alespoň 5 pracovních dnů pro opravu a realizaci) časové rozestupy mezi termíny	vše							+			-					+	

Obrázek 5.2: Tabulka požadavků na zkoušky

Podařilo se také **snížit čas potřebný k načtení stránek s požadavky**. Porovnání s aplikací Ing. Tesařové ukazuje tabulka 5.1. Uvedené hodnoty představují průměrnou dobu načítání požadavků k uvedenému období. Významné zlepšení je možné vidět zvláště u požadavků na zkoušky.

K dosažení rychlejšího zobrazení a kompletního načtení stránky dopomohly dva faktory. Prvním z nich je postupné vykreslování tabulky. Po příchodu na stránku je nejprve zobrazeno prvních 10 předmětů společně s požadavky k nim se vážícími. Poté se postupně vykreslí i zbytek předmětů. Druhým faktorem je snaha o zvýšení počtu elementů naráz vkládaných do DOM. Místo toho, aby se části tabulky vkládaly do DOM po jednotlivých buňkách, jsou vkládány po jednotlivých řádcích.

Požadavky	Aplikace Ing. Tesařové	Implementovaná aplikace
Zkoušky ZS 2021/2022	54 s	13 s
Zkoušky LS 2021/2022	46 s	8 s
Výuka ZS 2021/2022	114 s	50 s
Výuka LS 2021/2022	107 s	45 s

Tabulka 5.1: Porovnání rychlosti zobrazení některých stránek s požadavky s aplikací vytvořenou Ing. Tesařovou. Zkratky: ZS – zimní semestr, LS – letní semestr

### 5.3 Příznaky u předmětů

Tabulka zobrazující příznaky k předmětům a přednáškovým skupinám (obrázek 5.3) byla implementována dle návrhu a výsledné zobrazení je kompaktnější, než v aktuálně používané aplikaci. Je zde možné přidávat, měnit, a mazat příznaky. Všechny změny se ukládají dynamicky. Příznaky jsou přidávány přes tlačítka „+“. K jejich odstranění zde není žádná ikona, ale jsou odebrány vybráním implicitní hodnoty „-“ značící, že není zvolen žádný příznak. Barevně zvýrazněné řádky znamenají, že příznaky pro daný předmět byly minulý rok manuálně upraveny.

Rozvrhy IS
Příznaky u předmětů ZS 2022
Jana Stopková ▾

Příznaky vycházejí z povinností předmětů v daném ročníku a oboru. Většinu z nich je možné pomocí pravidel **generovat automaticky**. Pokud v následujícím formuláři změníte příznak, uloží se příznak změny do databáze a příští rok bude tento předmět označený žlutou barvou. Takto označené předměty je možné filtrovat kliknutím na ikonu filtru vedle vyhledávání.

Skupina xx znamená něco jako "ostatní". Jsou v ní zařazeny přednášky z volitelných předmětů, cvičení, kde si lze vybrat 1 z N termínů, apod. V xx jsou zařazeny také předměty pro předchozí ročník (pro opakující) a v 1BIT také IZG a ve 2BIT ITU, které si studenti s lepšími výsledky mohou zapsat o rok dříve.

Zobrazeno 50 záznamů Hledat:

Zkratka	1BIA	1BIB	2BIA	2BIB	3BIT	1MIT	2MIT	1DVI	1EIT	2EIT	INTE
AGS	- ▾	- ▾	- ▾	- ▾	- ▾	XX ▾ OBOR ▾ +	XX ▾ +	- ▾	- ▾	- ▾	- ▾
ATGD	XX ▾ +	XX ▾ +	XX ▾ +	XX ▾ +	XX ▾ +	- ▾	- ▾	- ▾	- ▾	- ▾	- ▾
BC_MOG1	- ▾	- ▾	- ▾	- ▾	- ▾	XX ▾ OBOR ▾ +	XX ▾ +	- ▾	- ▾	- ▾	- ▾
GAL	- ▾	- ▾	- ▾	- ▾	- ▾	XX ▾ OBOR ▾ +	XX ▾ OBOR ▾ +	- ▾	- ▾	- ▾	- ▾

Obrázek 5.3: Správa příznaků k předmětům

Nad tabulkou se nachází tlačítka pro generování příznaků, které otevírá okno (obrázek 5.4), odkud je možné vytvářet pravidla pro generování příznaků (dále jen pravidla) a nechat dle pravidel příznaky vygenerovat. V případě, že pro aktuální rok dosud nejsou vytvořena žádná pravidla, vedle tlačítka *Přidat* se navíc zobrazí tlačítka, které spouští převod příznaků z minulých let.



Pravidla se vypisují do jednoduché tabulky, která odpovídá návrhu, a jsou od sebe oddělena pomocí vodorovné čáry. Aby bylo dosaženo navrženého způsobu řazení pravidel (drag and drop), byla využita funkce `sortable` z knihovny jQuery UI. Díky použití této funkce je možné všechna pravidla přesouvat v rámci tabulky, ve které se nachází. Tato funkcionalita je uváděna do chodu „chycením“ ikony pro přesun pravidla (poslední z nabízených) a posunutím myši nahoru, nebo dolů. Jelikož ale jedno pravidlo může být přes více řádků, jak je to např. u posledního pravidla na obrázku níže, je nutné společně přesouvat více řádků. Z toho důvodu jsou jednotlivé řádky jednoho pravidla obaleny elementem `<tbody>`. Tabulka tedy obsahuje více elementů `<tbody>` a při přesouvání pravidel jsou to právě tyto elementy, které mění svou pozici v DOM.

Samotné generování příznaků je řešeno cyklem, který prochází jednotlivá pravidla seřazená dle jejich pořadí. Pro každé pravidlo jsou z databáze vybrány ty předměty, které pravidlu odpovídají, a k těmto předmětům se přidávají příznaky pro přednáškové skupiny z daného pravidla.

V případě provedení převodu pravidel z minulých let se pravidla překopírují a změní se pouze identifikátory programů a rok platnosti pravidla. Starý identifikátor programu je novým nahrazen tak, že se v databázi vyhledá program, jehož atribut `program_lonsky` je roven identifikátoru starého pravidla. Pokud takový program není nalezen, nebude u převedeného pravidla vyplněn.



Obrázek 5.4: Část vytvořených pravidel pro generování příznaků

## 5.4 Správa předmětů

Správa předmětů (obrázek 5.5) je určena jednak pro zobrazení předmětů pro vybrané období, ale také pro manuální zásahy do informací o předmětech. Úprava veškerých hodnot, které je možné změnit, je dynamická. *Id* v tabulce je vždy odkazem na kartu předmětu na webu FIT. Barevně zvýrazněné řádky značí, že byly zkoušky z daného předmětu minulý rok plánovány centrálně, ale tento rok nikoli. Z tabulky je také možné upravit učitele učící předmět, povinnosti předmětu v oborech a studenty, kteří mají zaregistrovaný předmět. Tyto úpravy jsou dostupné přes ikony v příslušných sloupcích, tedy ve sloupcích *Učitelé*, *Povinnost* a *Studenti*. Po kliknutí na ikonu se otevře modální okno s další tabulkou a formulářem, díky kterému je možné přidávat a odebírat záznamy.

Kromě základních údajů u předmětu evidujeme také povinnosti, učitele a registrované studenty. Pokud je předmět obarvený žlutou barvou, jedná se o předmět, který byl minulý rok centrálně plánovaný a tento rok již není. Takto označené předměty je možné filtrovat kliknutím na ikonu filtru vedle vyhledávání.

Zobrazeno 1000 záznamů Hledat: als

Id	Zkratka	Název	Kredity	Kapacita	Přihlášeno	Před	Sem./Cv.	Lab. cv.	Poc. cv.	Jiná cv.	Fakulta	Zakon.	Jazyk	Centrálně	Učitelé	Povinnosti	Studenti	Akce
#244813	GALe	Grafové algoritmy (v angličtině)	5	20	0	39	0	0	0	13	FIT VUT	Zk	en	<input type="checkbox"/>				
#244926	IALe	Algoritmy (v angličtině)	5	40	8	39	0	0	0	13	FIT VUT	ZaZk	en	<input type="checkbox"/>				
#244929	PCG	Paralelní výpočty na GPU	5	60	34	26	0	0	12	14	FIT VUT	Zk	cs	<input checked="" type="checkbox"/>				
#235510	PDD	Aplikace paralelních počítačů	0	20	1	39	0	0	0	0	FIT VUT	Zk	cs	<input type="checkbox"/>				

Obrázek 5.5: Tabulka pro správu předmětů

Když chceme k předmětu přidat učitele, je nutné zadat jeho *jméno* a *rolí*, jakou ve výuce předmětu zastává (např. přednášející, cvičící, nebo garant). Z důvodu velkého množství možných vyučujících nebylo vhodné vložit všechna jména jako možnosti do elementu `<select>`. Namísto této možnosti byla při implementaci využita funkce `autocomplete` z knihovny jQuery UI. Této metodě je předloženo pole vstupních objektů s klíči `value` (hodnoty obsahují id učitelů) a `label` (hodnoty obsahují jména učitelů s tituly). Když uživatel začne psát do vstupního pole formuláře jméno učitele, dojde k prohledání pole. Vyhledávání probíhá tak, že se kontroluje, zda některá z hodnot objektů patřící ke klíči `label` neobsahuje zadaný text. Uživateli jsou následně zobrazeny pouze ty hodnoty, které toto splňují. Počet zobrazených hodnot je omezen na 10, aby byla zajištěna rychlá odezva a uživatel nemusel vybírat z dlouhého seznamu třeba o 50 jménech. Když uživatel zvolí některé z nabízených jmen, dojde k uložení identifikátoru učitele do skrytého prvku formuláře.

V okně pro přidání povinností je ve formuláři možné vyplnit obor, povinnost a ročník, pro který bude povinnost v oboru platná. Zadávání oboru funguje na stejném principu jako zadávání jména učitele. Namísto jména oboru se ovšem zobrazuje řetězec ve tvaru „*Program – Obor*“.

Při implementaci správy registrovaných studentů musela být použita jistá optimalizace, aby bylo docíleno zobrazení v přijatelném čase. V prvotní implementaci se postupně každý řádek, tedy záznam o studentovi obsahující jeho *id* a *login*, přidával do DOM. Pro předměty s méně studenty se všechny řádky zobrazily rychle. Problém ovšem nastal u předmětů s několika stovkami studentů (např. předmět IDM, který měl v roce 2021 v zimním semestru 903 studentů). Kompletní načtení všech řádků trvalo více jak minutu. Z toho důvodu byl způsob implementace přehodnocen a v aktuálním řešení se nejprve vygeneruje 60 řádků ve formě řetězce a až poté se naráz vloží do DOM. Všechny záznamy o studentech v předmětu IDM v již zmíněném období se nyní zobrazí za méně než 8 vteřin.

Rozvrhy IS Předměty ZS 2021 Jana Stopková

Kromě základních údajů u předmětu evidujeme také povinnosti, učitele a registrované studenty. Pokud je předmět obarvený žlutou barvou, jedná se o předmět, který byl minulý rok centrálně plánovaný a tento rok již není. Takto označené předměty je možné filtrovat kliknutím na ikonu filtru vedle vyhledávání.

Zobrazeno 50 záznamů Hledat: idm

Id	Zkratka	Název	Kredity	Kapacita	Přihlášeno	Před	Sem./Cv.	Lab. cv.	Poc. cv.	Jiná cv.	Fakulta	Zakon.	Jazyk	Centrálně	Učitelé	Povinnosti	Studenti	Akce
#244836	IDM	Diskrétní matematika	5	20	0	39	0	0	0	13	FIT VUT	Zk	en	<input type="checkbox"/>				

**Povinnosti předmětu IDM - Diskrétní matematika**

Hledat:

Obor	Ročník	Povinnost	Akce
BIT (IT-BC-3)	1	P	
BITP (BIT)	1	P	

Obor:

- IT-BC-3 - BIT
- IT-MGR-2 - MBI
- IT-MGR-2 - MBS
- IT-BC-1H - BCH
- BIT - BITP
- MITAI - NBIO
- MITAI - NEMB

Obrázek 5.6: Ukázka modálního okna pro přidání povinnosti předmětu pro určitý obor

## 5.5 Import dat

Modul pro import dat, jehož podobu ukazuje obrázek 5.7, byl realizován dle návrhu. Data, která lze importovat, jsou rozdělena do tří kategorií: *rok*, *semestr* a *ostatní*. Každá kategorie obsahuje několik položek, přičemž u každé z nich je možné kliknutím na symbol „X“ otevřít nabídku akcí proveditelných s vybranou položkou. Těmito akcemi jsou *A* (aktualizovat záznamy), *O* (odstranit záznamy a nahrát nové), *S* (pouze odstranit záznamy) a *X* (bez akce). Význam jednotlivých možností je popsán v legendě nacházející se v pravé části stránky pod tlačítkem *Importovat*.

Do kategorie *rok* spadají ta data, která ve většině případů stačí importovat jednou ročně. Jsou jimi programy, obory, studijní plány a přednáškové a studijní skupiny. Do další kategorie, tedy do kategorie *semestr*, spadá většina zbylých dat, které je nutné importovat:

- Předměty a jejich povinnosti
- Počty studentů v předmětu – počty zaregistrovaných a zapsaných studentů
- Registrovaní studenti – identifikátor studenta, jeho login a registrovaný předmět společně s jeho povinností pro studenta
- Společní studenti – počty společných studentů pro dvojici předmětů a typ kolize (*B* – dva bakalářské předměty, *N* – dva magisterské předměty, *BN* – bakalářský a magisterský předmět)
- Počty na výuku a počty na zkoušku
- Požadavky na výuku a požadavky na zkoušku
- Tabulka zkouška projekt – data používaná při importu rozvrhu zkoušek do IS VUT
- Vyučované předměty – předměty a jejich vyučující společně s informací o tom, zdali je daný vyučující např. garant, cvičící nebo přednášející

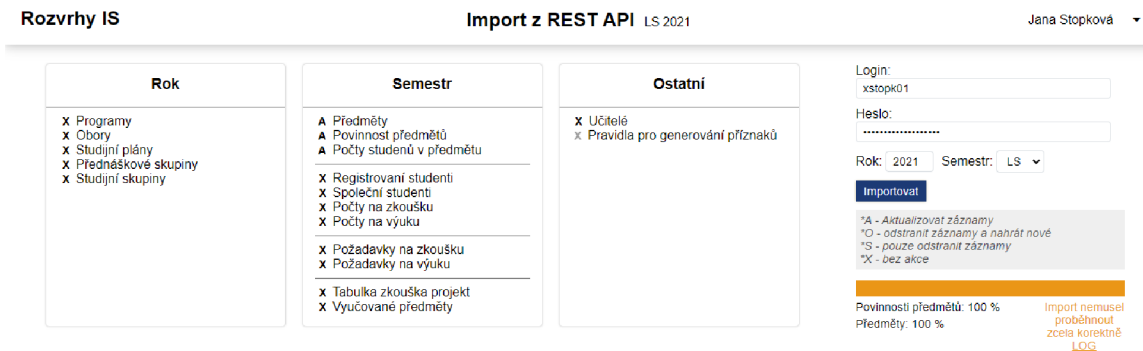
Do kategorie *ostatní* pak patří data, která se neváží k žádnému konkrétnímu roku, ani semestru. Tato kategorie má pouze dvě položky, kterými jsou učitelé a pravidla pro generování příznaků.

Pravidla pro generování příznaků jsou do této kategorie přiřazena i přes to, že není možné je z REST API importovat, a to z důvodu potřeby informovat uživatele. Pokud totiž uživatel odstraní programy pro určitý rok, odstraní se spolu s nimi i pravidla pro generování příznaků, která k danému roku přísluší. Uživatel je o této skutečnosti informován tak, že po vybrání akce *O* nebo *S* u možnosti *Programy*, dojde u možnosti *Pravidla pro generování příznaků* k automatickému vybrání akce *S*.

Některé akce vyžadují, aby spolu s nimi byly provedeny jiné. Takováto situace nastává např. při odstranění předmětů, kdy musí zároveň dojít i k odstranění veškerých dat, která se na předměty váží. V této situaci dojde k vybrání akce *O* nebo *S* (v závislosti na akci, která byla vybrána u předmětů) u všech možností, které se na předměty váží. K odstranění souvisejících dat by sice mohlo dojít automaticky bez toho, aby byly další zmíněné možnosti vybírány, ale uživatel by si nemusel uvědomit, o jaká všechna data by s odstraněním programů přišel.

Kromě výběru dat, která budou importována, nabízí modul i výběr roku a semestru, pro který má import proběhnout.





Obrázek 5.7: Modul pro import dat

Po zadání přihlašovacích údajů k REST API a spuštění importu dojde k postupnému volání funkcí pro import vybraných částí dat. Celý import dat je prováděn v rámci transakce, což zajišťuje zachování integrity dat a přispívá též k rychlosti ukládání dat do databáze.

Data jsou stahována pomocí cURL<sup>1</sup> a ukládána do pomocného pole. Některé části importu vyžadují drobnou úpravu části získaných dat. K uložení do databáze dojde teprve poté, co jsou do pomocného pole uloženy všechny záznamy k aktuálně zpracovávané části dat. Jednotlivé řádky nejsou do databáze vkládány po jednom, nýbrž po  $x$  řádcích, kde  $x = 100.000/\text{počet sloupců tabulky}$ . S tímto nastavením je pro zimní semestr 2021 možné do každé tabulky uložit všechna data v jediném dotazu. V návrhu uvedený limit na velikost SQL dotazu porušen nebyl. Nejrozsáhlejším dotazem pro uvedené období bylo vložení dat do tabulky *ma\_registrovane*. Vkládalo se skoro 18 tisíc řádků, ale velikost výsledného dotazu byla pouze přibližně 0,54 MB, čímž stále zůstává velká rezerva.

Čas, který import zabere, se podařilo značně zredukovat. Pokud by byl import dat před plánováním zkoušek realizován pomocí nově implementované aplikace, vše by trvalo průměrně 6 minut a 10 sekund namísto původních 60 až 90 minut. V rámci implementované aplikace je třeba před plánováním zkoušek aktualizovat předměty a jejich povinnosti, počty studentů v předmětech, registrované a společné studenty, počty na zkoušku, požadavky na výuku, tabulku zkouška projekt a učitele společně s předměty, které vyučují.

Informování uživatele o postupu importu dat je řešeno následujícím způsobem. Před spuštěním importu je na straně klienta vygenerován název souboru, do kterého budou záznamy o postupu importu ukládány. Záznamy jsou vždy zapisovány ve tvaru: *pořadí importované části~název importované části~procento dokončení*. K zapisování těchto záznamů dochází průběžně. Pro bližší specifikaci je vhodné uvést příklad: Při importu povinností předmětů jsou z databáze nejprve načteny všechny předměty pro vybrané období a poté jsou pro každý předmět stahovány z REST API jeho povinnosti. K uložení záznamu (přidání řádku do souboru) dojde po dokončení zpracování předmětu. Na straně klienta je obsah souboru čten v pravidelných intervalech. Dle procenta dokončení uvedeného v posledním záznamu dochází k aktualizaci ukazatele postupu importu. O stavu dokončení jednotlivých částí je uživatel též informován pomocí textu zobrazujícího se pod ukazatelem stavu, který se rovněž průběžně aktualizuje. Tento text má tvar *Název importované části:procento dokončení*. V případě importu více částí současně se tyto informativní řádky postupně přidávají. Nejnovější informace se vždy nachází nejvýše.

<sup>1</sup><https://www.php.net/manual/en/book.curl.php>

V rámci importu dochází také k zaznamenávání chyb, které nebyly kritické a bylo možné i přes jejich výskyt v importu pokračovat. Příkladem takové chyby může být chyba v importu povinností předmětů, kdy povinnost vůči některému oboru/specializaci nebylo možné uložit z důvodu neexistence tohoto oboru či specializace v databázi. Neexistence oboru v databázi buď může znamenat, že tento obor dosud nebyl importován, nebo že se jedná o obor jiné fakulty, který se může v datech z REST API také vyskytnout. Při zaznamenání chyby je vždy uložen její popis spolu s dalšími daty, která by mohla uživateli pomoci identifikovat problém. Veškeré zaznamenané chyby jsou po dokončení importu nabídnuty uživateli ke stažení ve formě textového souboru. Stažení tohoto souboru bylo implementováno na základě [16]. V případě, že žádné problémy nenastaly, zobrazí se pod ukazatelem průběhu importu text „Import dokončen“.

# Kapitola 6

## Testování

Nedílnou součástí vývoje bylo i postupné testování. Nejprve byly jednotlivé moduly testovány lokálně (podkapitola 6.1) a po vytvoření prvního funkčního prototypu došlo k jeho nasazení na produkční server a k otestování v rámci plánování zkoušek (podkapitola 6.2). Po dokončení implementace byly v rámci plánování výuky otestovány i další moduly, o čemž pojednává podkapitola 6.2. Kromě testování aplikace došlo též k měření času potřebného pro import dat, kdy byl zkoumán vliv počtu vkládaných řádků na rychlost importu (sekce 6.3).

### 6.1 Průběžné testování při vývoji

Každý z vyvíjených modulů byl v průběhu vývoje testován lokálně. K ověření správné funkcionality byla využívána reálná data exportovaná z produkční databáze společně s lokálně spuštěnou aplikací Ing. Tesařové (dále jen porovnávanou aplikací). Některé chyby tak bylo možné odhalit pouhým pohledem na dvě aplikace, které by měly zobrazovat totéž.

Vizuální a jiné rozdíly ale ne vždy značily chybu, v mnohých případech indikovaly spíše opak, tedy správnost implementovaného řešení. Některé neshody s porovnávanou aplikací, ať už po vizuální stránce, anebo po stránce datové struktury, byly konzultovány s Ing. Dytrychem, aby došlo k vyjasnění důvodu neshody.

#### Požadavky na výuku a zkoušky

Mezi moduly nejnáročnější na testování patřily požadavky na výuku a zkoušky, jejichž implementaci popisuje podkapitola 5.2. Hlavní částí těchto modulů jsou rozsáhlé tabulky s uživatelsky upravitelným obsahem většiny buněk. Správné ukládání veškerých hodnot bylo nutné postupně odladit. Stejně tak udržování konzistence tabulky při manipulaci s řádky (jejich přidávání a odstraňování).

Z důvodu opakovaně se vyskytujících problémů s konzistencí tabulky byla navržena testovací sada, díky které se podařilo eliminovat veškeré problémy. Mezi opakující se problémy patřilo například:

- nesprávné nastavování atributu `rowspan` jazyka HTML (obrázek 6.1),
- vkládání požadavků z minulých období na nesprávné místo,
- neuložení obsahu buňky při její změně.

Testovací sada, která tyto problémy vyřešila, je rozdělena na tři části a každá z nich obsahuje několik testovacích případů:

### 1. Požadavky

- Přidat dva řádky pro požadavky a smazat vrchní. Totéž provést v situaci, kdy k předmětu existuje pouze jeden požadavek.
- Přidat dva řádky pro požadavky a smazat spodní.
- Přidat řádek pro požadavek, zvolit učitele a přidat více než jeden řádek pro požadavek na termín. Poté požadavek odstranit.

### 2. Požadavky na termín

- Přidat řádek požadavku na termín zkoušky pod 1. existující požadavek a změnit některou z hodnot v novém řádku. Následně pod nový řádek přidat ještě jeden a v něm opět upravit některou z hodnot.
- Přidat řádek požadavku na termín zkoušky pod poslední existující požadavek a změnit v něm některou z hodnot. Následně pod nový řádek přidat ještě jeden a v něm opět upravit některou z hodnot.
- Odstranit nově přidaný řádek požadavku na termín.
- Pokusit se odstranit kladný požadavek na termín (preferenci) zkoušky přičemž je v řádku ještě záporný požadavek (omezení). Do některé buňky, jejíž hodnota byla tímto odstraněna, vložit hodnotu znovu.

### 3. Požadavky z minulých období

- Vložit požadavky na termín k poslednímu požadavku u předmětu.
- Připnout přidaný požadavek.
- Vložit požadavky všech učitelů k předmětu
- Připnout libovolný požadavek některého z učitelů

Vyhovoval by mi termín														Nemohu zkusit								
Předmět	Zkouší	Roz	Poznámka	Termín	Hod	Kol	Pref. místost	Datum od	Datum do	Den	Čas od	Čas do	Poznámka	A	Datum od	Datum do	Den	Čas od	Čas do			
PDB (26 stud) Kolář Dušan, doc. Dr. Ing. Rychlý Marek, RNDr. Ph.D.	+	ob2	Prosím bych PDB a PDBe plánovat dotromady do jedné učebny. Také budu hodnovit častěčně zkušby z UPA - Marek Rychlý. Kdyby šlo mi týden ve	řadná	2	1		00207/G202				07:00	20:50	Čt	07:00	12:50		KD				
				1 oprar	2	1		E1041105						07:00	20:50	-	17:00	20:50		moc pozdě		
				2 oprar	2	1		A112						07:00	20:50	-	07:00	07:50		moc brzy (na 7:00 nedějdu učně, od 8:00 velmi nerac)		
				vše													Út	13:00	20:50		senát, komise	
				vše													Pá	07:00	20:50		dost nerad v pátek - mám setkání se studenty...	
				vše													-	12:00	12:50		oběd	
				vše																		
PDBe (4 stud)	Kolář Dušan, doc. Dr. Ing.	ob2	prosím bych PDB a PDBe plánovat dotromady											Čt	07:00	12:50		KD				
vše														-	17:00	20:50		moc pozdě				
vše														-	07:00	07:50		moc brzy (na				

Obrázek 6.1: Nekonzistentní zobrazení tabulky požadavků na zkoušku

## 6.2 Nasazení při plánování

### Zkoušky v letním semestru 2021/2022

První ostré testování prototypu aplikace bylo možné uskutečnit 4. března 2022. Konkrétně byly při přípravě plánování prováděné Ing. Jaroslavem Dytrychem, Ph.D. a Bc. Kristýnou Zaklovou otestovány dvě části: správa příznaků k předmětům a správa požadavků na zkoušku.

Generování příznaků k předmětům proběhlo až na pár výjimek, kdy se k některým předmětům nesprávně přidal jeden příznak navíc, bez potíží. Celkově nebylo nutné dělat manuální zásahy do většího množství předmětů, čímž došlo k ušetření času.

Významnější nedostatky neukázalo ani testování správy požadavků. Jediným problémem bylo to, že se v některých buňkách tabulky nezobrazovaly správně hodnoty. Konkrétně šlo o elementy `<select>`. Z databáze sice byly vyčteny správné hodnoty, ale poté byla nesprávně nastavována vybraná hodnota. Šlo o chybu, jejíž příčina byla snadno nalezena a samotná oprava zabrala pouze několik málo minut.

O chybách autorku informovala Bc. Kristýna Zaklová, která zároveň navrhla drobné vylepšení. Jelikož je nutné texty v hlavičce tabulky požadavků zkracovat (např. slovo *Přestávka* je zkráceno na *Přes*), přišla s návrhem zobrazovat po najetí myši nezkrácený text. Tato drobná úprava by měla přinést o něco snazší orientaci v tabulce.

### Výuka v zimním semestru 2022/2023

Druhé ostré testování, tentokrát již hotové aplikace, se uskutečnilo v pátek 22. dubna 2022 a v sobotu 23. dubna 2022. Přípravu opět prováděla Bc. Kristýna Zaklová a Ing. Jaroslav Dytrych, Ph.D. Byla otestována správa příznaků k předmětům, správa požadavků na výuku a také import dat z REST API.

Generování příznaků jako takové tentokrát zafungovalo tak, jak mělo, až na předem známé výjimky (viz konec kapitoly 2.2), kde se počítá s manuální úpravou příznaků. Ukázal se však jiný problém, který ovšem nesouvisel s implementací řešení. Některé příznaky byly vygenerovány špatně z důvodu chybějících dat v databázi. Nešlo ani o problém s importem dat, jelikož se tato data z REST API nestáhla z důvodu, že je REST API vůbec neposkytlo.

## 6.3 Měření času importu dat

V závěru testování aplikace proběhlo také měření reálného importu dat. Cílem bylo porovnání doby potřebné pro import, když budou data do databáze vkládána po jednom řádku a když budou všechna data (v rámci jedné tabulky) do databáze vložena najednou.

Testován byl import více různých částí, vždy s nastavenou možností *O* – odstranit záznamy a nahrát nové. Výsledky tohoto testování ukazuje tabulka 6.1. Ve sloupci *Import* je průměrný čas potřebný pro dokončení importu (od stažení dat z REST API po jejich úspěšné uložení do databáze). Sloupec *Vložení dat* poté zachycuje pouze dobu potřebnou pro samotné uložení dat.

Můžeme si všimnout, že při importu dat zabírá nejvíce času komunikace s REST API a případná úprava hodnot. Rozdíly mezi vkládáním jednoho a *N* řádků nejsou příliš významné. Největší rozdíl můžeme vidět u části *Registrovaní studenti*. Zde byla data vkládána do tabulky **ma\_registrovane**. Do této tabulky bylo vkládáno nejvíce řádků (skoro 18000) ze všech testovaných tabulek.

---

<sup>1</sup>Požadavky na zkoušky jsou ukládány do dvou tabulek.

<b>Importovaná část</b>	<b>Současně vložené řádky</b>	<b>Import</b>	<b>Vložení dat</b>
Požadavky na zkoušky <sup>1</sup>	1	29,6 s	5 s
	N	28,8 s	4 s
Registrovaní studenti	1	55 s	4,6 s
	N	46 s	1,4 s
Vyučované předměty	1	46,4 s	0,32 s
	N	44,4 s	0,12 s

Tabulka 6.1: Výsledky testů měření rychlosti importu dat. N současně vložených řádků zde značí vložení všech řádků (v rámci jedné tabulky) v jednom SQL dotazu.

# Kapitola 7

## Závěr

V rámci této práce byl popsán způsob plánování rozvrhů pro výuku a zkoušky na Fakultě informačních technologií Vysokého učení technického v Brně (FIT). Blíže byl popsán zejména sběr požadavků a příprava plánování (kompletace dat), a také samotné plánování. Byly také popsány aplikace, které se k tomuto účelu na FIT využívají a byly definovány nedostatky aplikace používané pro přípravu plánování.

Implementované řešení umožňuje spravovat příznaky u předmětů a také tyto příznaky generovat na základě uživatelsky nastavitelných pravidel. Je tedy možné je snadno změnit, aby odpovídala momentálním potřebám. Vyřešeno je i převádění definovaných pravidel do následujícího roku, aby nemusela být uživatelem zadávána znovu. V rámci stránek s rozsáhlými tabulkami umožňujícími správu požadavků na výuku a zkoušky dochází k automatickému ukládání veškerých dat. Samotné načítání těchto stránek se podařilo optimalizovat a tím zrychlit přibližně o polovinu. Zkrátit se podařilo i čas potřebný pro import dat. Úspora času je již dosti významná. Potřebný čas totiž klesl z původních 60–90 minut na průměrných 6 minut (obě hodnoty se vztahují k importu dat potřebných pro plánování zkoušek).

Probíhalo také průběžné testování aplikace při vývoji. Řešení bylo navíc dvakrát otestováno v ostrém provozu, přičemž druhé testování v rámci plánování výuky na zimní semestr akademického roku 2022/2023 dokázalo odhalit chybu v API IS VUT používaném pro import dat.

Aplikace by mohla být dále rozšířena o možnost vytváření samostatných pravidel pro generování příznaků k těm předmětům, na které se nevztahují běžná pravidla. Další vylepšení by se mohlo týkat generování přehledové tabulky s časovým plánem zkoušek pro jednotlivé přednáškové skupiny, která po zveřejnění rozvrhu poskytuje studentům přehled, kdy jsou jaké zkoušky. Obsah této tabulky je v současné době nutné generovat manuálně tak, že uživatel postupně prokliká všechny nabízené přednáškové skupiny, čímž se data uloží do databáze. Tento způsob není ideální. Vylepšení by tedy spočívalo v generování obsahu této tabulky již při tvorbě rozvrhu, anebo lépe ve vytvoření algoritmu, který vše potřebné vygeneruje až při zobrazení či exportu dat.



# Literatura

- [1] Úplné znění studijního a zkušebního řádu VUT [online]. Září 2018 [cit. 2022-04-26]. Dostupné z: <https://www.vut.cz/uredni-deska/vnitri-predpisy-a-dokumenty/-d149085/uplne-zneni-studijniho-a-zkusebniho-radu-od-1-9-2018-p163172>.
- [2] Úplné znění směrnice č. 13/2018 Pravidla o organizaci studia na FIT [online]. Březen 2021 [cit. 2022-04-26]. Dostupné z: <https://www.fit.vut.cz/fit/info/smernice/sm2018-13-uz.pdf>.
- [3] AL YAKOOB, S. M. a SHERALI, H. D. Mathematical models and algorithms for a high school timetabling problem. *Computers & operations research*. Elsevier Ltd. 2015, sv. 61, s. 56–68. ISSN 0305-0548.
- [4] BELIGIANNIS, G. N., MOSCHOPOULOS, C. a LIKOTHANASSIS, S. D. A genetic algorithm approach to school timetabling. *The Journal of the Operational Research Society*. London: Taylor & Francis. 2009, sv. 60, č. 1, s. 23–42. ISSN 0160-5682.
- [5] BODNAR, J. MySQL storage engines. *ZetCode* [online]. Červenec 2020 [cit. 2022-04-27]. Dostupné z: <https://zetcode.com/mysql/storageengines/>.
- [6] CABRAL, S. a MURPHY, K. *MySQL administrator's bible*. 1st ed. Indianapolis, IN: Wiley Pub, 2009. Bible. ISBN 0470416912.
- [7] CASTRO, E. *HTML5 a CSS3 : názorný průvodce tvorbou WWW stránek*. 1. vyd. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
- [8] EVEN, S., ITAI, A. a SHAMIR, A. On the Complexity of Timetable and Multicommodity Flow Problems. *SIAM journal on computing*. Philadelphia: Society for Industrial and Applied Mathematics. 1976, sv. 5, č. 4, s. 691–703. ISSN 0097-5397.
- [9] EXADEL MEDIA. Old Reliable: A History of MySQL. *Enterprise Software Development Company | Exadel* [online]. Říjen 2017 [cit. 2022-04-26]. Dostupné z: <https://exadel.com/news/old-reliable-mysql-history/>.
- [10] HOSFORD, A. L., KIRKBRIDE, P., LABRECQUE, J., LOVE, J., MEHLA, G. et al. *The JavaScript Workshop*. Packt Publishing, 2019. ISBN 9781838641917.
- [11] KUCHYNKA, M. *Informační systém pro plánování rozvrhů*. Brno, CZ, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Dostupné z: <https://www.fit.vut.cz/study/thesis/23408/>.
- [12] LIE, H. W. a BOS, B. Cascading Style Sheets, level 1. *World Wide Web Consortium (W3C)* [online]. W3C, prosinec 1996 [cit. 2022-04-25]. Dostupné z: <https://www.w3.org/TR/REC-CSS1-961217>.



- [13] ORACLE. MySQL 5.7 Reference manual. *MySQL* [online]. [cit. 2022-04-29]. Dostupné z: <https://dev.mysql.com/doc/refman/5.7/en/insert-optimization.html>.
- [14] RAGHAVJEE, R. a PILLAY, N. A Study of Genetic Algorithms to Solve the School Timetabling Problem. In: *Advances in Soft Computing and Its Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, sv. 8266, s. 64–80. Lecture Notes in Computer Science. ISBN 3642451101.
- [15] ECMA INTERNATIONAL. ECMAScript® 2021 Language Specification. *ECMA International* [online]. [cit. 2022-04-29]. Dostupné z: <https://262.ecma-international.org/12.0/>.
- [16] JOSEPH SILBER, MATKO, LUDOVIC FELTZ, KORAY TUGAY. *How to create a file in memory for user to download, but not through server?* [Stack Overflow]. Srpen 2021 [cit. 2022-05-04]. Dostupné z: <https://stackoverflow.com/a/33542499>.
- [17] SEO EXPLORER. Twenty Ways To Optimize Slow MySQL for Faster Insert Rate. *SEO Explorer* [online]. [cit. 2022-04-29]. Dostupné z: <https://seo-explorer.io/blog/twenty-ways-to-optimize-mysql-for-faster-insert-rate/>.
- [18] THE PHP GROUP. History of PHP. *PHP: Hypertext Preprocessor* [online]. [cit. 2022-04-27]. Dostupné z: <https://www.php.net/manual/en/history.php.php>.
- [19] TRAVIS, CHICKENS. *How to remove all duplicates from an array of objects?* [Stack Overflow]. Únor 2022 [cit. 2022-05-04]. Dostupné z: <https://stackoverflow.com/a/56757215>.
- [20] SKLAR, D. *PHP 7 : praktický průvodce nejrozšířenějším skriptovacím jazykem pro web*. Vydání první. Brno: Zoner press, 2018. ISBN 978-80-7413-363-3.
- [21] TASSOPOULOS, I. X. a BELIGIANNIS, G. N. Solving effectively the school timetabling problem using particle swarm optimization. *Expert systems with applications*. Elsevier Ltd. 2012, sv. 39, č. 5, s. 6029–6040. ISSN 0957-4174.
- [22] TESAŘOVÁ, A. *Program pro podporu plánování rozvrhů*. Vysoké učení technické v Brně. Fakulta informačních technologií, 2018.
- [23] W3C. CSS1 Fact Sheet. *World Wide Web Consortium (W3C)* [online]. W3C, prosinec 1996 [cit. 2022-04-25]. Dostupné z: <https://www.w3.org/Press/CSS1-fact.html>.
- [24] W3C a WHATWG. *HTML Living Standard* [online]. WHATWG, duben 2022 [cit. 2022-04-22]. Dostupné z: <https://html.spec.whatwg.org/multipage/>.

# Příloha A

## REST API

Obsahem této přílohy je přehled adres URL, které budou využity pro importování různých částí dat z REST API.

- Seznam programů:  
<https://api.vut.cz/api/fit/predmety/v3>
- Seznam oborů/specializací v programu:  
[https://api.vut.cz/api/fit/program/{id\\_programu}/obory/v3](https://api.vut.cz/api/fit/program/{id_programu}/obory/v3)
- Seznam studijních plánů v programu:  
[https://api.vut.cz/api/fit/program/{id\\_programu}/plany/v3](https://api.vut.cz/api/fit/program/{id_programu}/plany/v3)
- Seznam přednáškových skupin a studijních skupin:  
<https://api.vut.cz/api/fit/skupiny/v3>
- Seznam předmětů:  
<https://api.vut.cz/api/fit/predmety/v3>
- Detail předmětu:  
[https://api.vut.cz/api/fit/aktualni\\_predmet/{id\\_predmetu}/v3](https://api.vut.cz/api/fit/aktualni_predmet/{id_predmetu}/v3)
- Učitelé učící předmět:  
[https://api.vut.cz/api/fit/aktualni\\_predmet/{id\\_predmetu}/vyucujici/v3](https://api.vut.cz/api/fit/aktualni_predmet/{id_predmetu}/vyucujici/v3)
- Studenti zapsaní v předmětu:  
[https://api.vut.cz/api/fit/aktualni\\_predmet/{id\\_predmetu}/studenti/zapsani/v3](https://api.vut.cz/api/fit/aktualni_predmet/{id_predmetu}/studenti/zapsani/v3)
- Studenti zaregistrovaní v předmětech:  
<https://api.vut.cz/api/fit/spolecni-studenti/zaregistrovani/seznam/v3>
- Studenti zapsaní v předmětech:  
<https://api.vut.cz/api/fit/spolecni-studenti/zapsani/seznam/v3>
- Přiřazení typů hodnocení k předmětům:  
[https://api.vut.cz/api/fit/aktualni\\_predmet/{id\\_predmetu}/zkouska-projekt/v3](https://api.vut.cz/api/fit/aktualni_predmet/{id_predmetu}/zkouska-projekt/v3)
- Počty společných zaregistrovaných studentů v předmětech:  
<https://api.vut.cz/api/fit/spolecni-studenti/zaregistrovani/pocty/v3>

- Povinnost předmětu v oborech/specializacích:  
[https://api.vut.cz/api/fit/aktualni\\_predmet/{id\\_predmetu}/obory/v4](https://api.vut.cz/api/fit/aktualni_predmet/{id_predmetu}/obory/v4)
- Počty studentů jednotlivých oborů/specializací v předmětech:  
<https://api.vut.cz/api/fit/spolecni-studenti-obory/zapsani/pocty/v3>
- Počty studentů na jednotlivých termínech zkoušek:  
<https://api.vut.cz/api/predmety-termíny/rok/{rok}/v3>
- Detail organizační jednotky:  
[https://api.vut.cz/api/fit/org/{id\\_organizacni\\_jednotky}/v3](https://api.vut.cz/api/fit/org/{id_organizacni_jednotky}/v3)
- Požadavky na zkoušku a preference:  
[https://api.vut.cz/api/aktualni\\_predmet/{id\\_predmetu}/pozadavky/zkousky/v3](https://api.vut.cz/api/aktualni_predmet/{id_predmetu}/pozadavky/zkousky/v3)
- Požadavky na výuku:  
[https://api.vut.cz/api/aktualni\\_predmet/{id\\_predmetu}/pozadavky/vyucovani/v3](https://api.vut.cz/api/aktualni_predmet/{id_predmetu}/pozadavky/vyucovani/v3)
- Preference termínů na výuku:  
[https://api.vut.cz/api/rozvrh\\_pozadavek\\_vyucovani/{id\\_pozadavku\\_na\\_vyuku}/preference/v3](https://api.vut.cz/api/rozvrh_pozadavek_vyucovani/{id_pozadavku_na_vyuku}/preference/v3)
- Požadavky na návaznosti výuky:  
[https://api.vut.cz/api/rozvrh\\_pozadavek\\_vyucovani/{id\\_pozadavku\\_na\\_vyuku}/navaznosti/v3](https://api.vut.cz/api/rozvrh_pozadavek_vyucovani/{id_pozadavku_na_vyuku}/navaznosti/v3)
- Omezení vyučujícího:  
[https://api.vut.cz/api/zamestnanec/{id\\_vyucujiciho}/pozadavky/omezeni/v3](https://api.vut.cz/api/zamestnanec/{id_vyucujiciho}/pozadavky/omezeni/v3)