

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Bakalářská práce**

# **Informační systém pro Student Support Centre**

Místo této strany bude  
zadání práce.

# Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 30. dubna 2020

Jakub Hlaváč

# Poděkování

Tímto bych chtěl poděkovat vedoucímu mé práce Doc. Ing. Liborovi Vášovi, Ph.D. za vedení mé práce a cenné připomínky. Dále Ing. Ladislavu Pešíčkovi za pomoc při problémech s aplikačním serverem. Velké díky si zaslouží také má rodina, která mě při psaní práce podporovala.

## **Abstract**

This bachelor's thesis is about the creation of an information system with web interface for easier administration and service of the Student Support Centre project. The aim of the project is to create a functional and suitable system that will serve as a powerful tool for scheduling Student Support Centre services. In addition to administrators, the application will also be available to tutors who will be using a tool to schedule own lectures. The system will allow registration of tutors and administrators will be able to monitor the statistics of Student Support Centre attendance and the numbers of lessons taught. Also the system will send automatic email notifications to users. In the system will be possibility to fill in questionnaires of satisfaction with SSC services and send them anonymously to the application. These questionnaires can be exported as a file. In the end of development, the whole application is tested and deployed to a real web server.

## **Abstrakt**

Tato bakalářská práce se zabývá doimplementací informačního systému a webového rozhraní pro snadnější správu a chod projektu Student Support Centre. Cílem projektu je vytvořit funkční systém, který bude sloužit jako nástroj pro sestavování služeb Student Support Centre. Tento nástroj budou moci kromě správců používat i tutoři, kteří si v něm budou zapisovat služby v SSC. Systém bude umožňovat registraci tutorů a administrátor bude moci sledovat statistiky návštěvnosti Student Support Centre a počet vyučovaných hodin. Prostřednictvím systému také půjde rozesílat emailové notifikace uživatelům. V systému bude možnost vyplňovat dotazníky spokojenosti s výukou SSC. Tyto dotazníky bude následně možné vyexportovat do souboru. V závěru je celá aplikace otestována podle scénáře a nasazena na reálný webový server.

# Obsah

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Úvod</b>  | <b>1</b>  |
| <b>2</b> | <b>Moderní webové aplikace</b>                       | <b>2</b>  |
| 2.1      | Cloudové řešení                                      | 2         |
| 2.1.1    | AWS  | 2         |
| 2.1.2    | Google   | 3         |
| 2.2      | ASP.NET Core   | 3         |
| 2.3      | MVC  | 4         |
| <b>3</b> | <b>Aktuální webový portál Student Support Centre</b> | <b>5</b>  |
| <b>4</b> | <b>Seznámení s částečnou implementací</b>            | <b>7</b>  |
| 4.1      | Úvod   | 7         |
| 4.2      | Rozdělení týmu                                       | 7         |
| 4.3      | Rozsah projektu                                      | 8         |
| 4.4      | Kontext systému                                      | 8         |
| 4.5      | Specifikace požadavků                                | 8         |
| 4.6      | Provozní prostředí                                   | 10        |
| 4.7      | Publikum   | 10        |
| 4.8      | Databáze   | 10        |
| 4.9      | Třídy uživatelů                                      | 11        |
| 4.10     | Implementované funkce systému                        | 13        |
| 4.10.1   | Integrace s webAuth                                  | 13        |
| 4.10.2   | Registrace   | 13        |
| 4.10.3   | Přihlášení   | 14        |
| 4.10.4   | Podání přihlášky na roli tutora                      | 14        |
| 4.10.5   | Vytváření vlastních rozvrhových akcí                 | 15        |
| 4.10.6   | Správa uživatelů a tutorů                            | 15        |
| 4.10.7   | Správa statického obsahu webu                        | 17        |
| <b>5</b> | <b>Přepsání aplikace do .NET Core</b>                | <b>18</b> |
| 5.1      | .NET Standard  | 18        |

|          |  |           |
|----------|--|-----------|
| 5.2      | .NET Core . . . . .                            | 18        |
| 5.3      | Rozdíly oproti .NET Frameworku . . . . .       | 19        |
| 5.4      | .NET Core výhody . . . . .                     | 19        |
| 5.5      | Struktura projektu . . . . .                   | 20        |
|          | 5.5.1 ISSSC . . . . .                          | 20        |
|          | 5.5.2 ISSSCResources . . . . .                 | 20        |
|          | 5.5.3 Statistiky . . . . .                     | 21        |
| 5.6      | Problémy při migraci . . . . .                 | 21        |
|          | 5.6.1 Databáze . . . . .                       | 21        |
| <b>6</b> | <b>Návrh databázového modelu . . . . .</b>     | <b>23</b> |
| 6.1      | Scis_user . . . . .                            | 23        |
| 6.2      | Scis_session . . . . .                         | 25        |
| 6.3      | Scis_param . . . . .                           | 25        |
| 6.4      | Scis_content . . . . .                         | 26        |
| 6.5      | Event . . . . .                                | 27        |
| 6.6      | Feedback . . . . .                             | 28        |
| 6.7      | Tutor_application . . . . .                    | 28        |
| 6.8      | Tutor_application_subject . . . . .            | 28        |
| 6.9      | Pomocné tabulky . . . . .                      | 29        |
|          | 6.9.1 Approval . . . . .                       | 29        |
|          | 6.9.2 Participation . . . . .                  | 29        |
| 6.10     | Číselníky . . . . .                            | 30        |
|          | 6.10.1 Enum_role . . . . .                     | 30        |
|          | 6.10.2 Enum_subject . . . . .                  | 30        |
| <b>7</b> | <b>Implementované případy užití . . . . .</b>  | <b>32</b> |
| 7.1      | Žádosti o extra lekci . . . . .                | 32        |
| 7.2      | Potvrzování lekcí . . . . .                    | 33        |
|          | 7.2.1 Grafická interpretace . . . . .          | 33        |
|          | 7.2.2 Stažení rozvrhu ve formátu CSV . . . . . | 35        |
| 7.3      | Sběr statistik k lekci . . . . .               | 36        |
|          | 7.3.1 Sběr zpětné vazby . . . . .              | 36        |
|          | 7.3.2 Sběr dat o návštěvnosti lekce . . . . .  | 38        |
| 7.4      | Prohlížení statistik . . . . .                 | 39        |
|          | 7.4.1 Statistiky lekcí . . . . .               | 39        |
|          | 7.4.2 Statistiky tutorů . . . . .              | 40        |
|          | 7.4.3 Stažení a zobrazení feedbacků . . . . .  | 41        |
| 7.5      | Emailové notifikace . . . . .                  | 42        |
| 7.6      | Admin login . . . . .                          | 44        |

|           |   |           |
|-----------|---|-----------|
| 7.7       | Úprava dříve implementovaných případů užití . . . . . | 45        |
| 7.7.1     | Administrace uživatelů . . . . .                      | 45        |
| 7.7.2     | Statické texty . . . . .                              | 47        |
| 7.7.3     | Pravý informační panel . . . . .                      | 47        |
| <b>8</b>  | <b>Nasazení informačního systému . . . . .</b>        | <b>49</b> |
| 8.1       | Databázový server . . . . .                           | 49        |
| 8.2       | Aplikační server . . . . .                            | 49        |
| 8.2.1     | Požadovaný software . . . . .                         | 49        |
| 8.2.2     | Nasazení aplikace v Internet Information Services . . | 50        |
| <b>9</b>  | <b>Testování . . . . .</b>                            | <b>51</b> |
| 9.1       | HTML validace . . . . .                               | 51        |
| 9.2       | Mobile responsivity . . . . .                         | 52        |
| 9.3       | Manuální testování . . . . .                          | 53        |
| 9.3.1     | Přihlášení administrátora . . . . .                   | 53        |
| 9.3.2     | Emailové notifikace . . . . .                         | 54        |
| 9.3.3     | Vyplnění feedbacku . . . . .                          | 55        |
| 9.3.4     | Prohlížení statistik . . . . .                        | 57        |
| 9.3.5     | Práva administrátorů . . . . .                        | 58        |
| <b>10</b> | <b>Rozšíření aplikace . . . . .</b>                   | <b>60</b> |
| 10.1      | Editace parametrů . . . . .                           | 60        |
| 10.2      | Mobilní aplikace . . . . .                            | 60        |
| 10.3      | Integrace s portal.zcu.cz . . . . .                   | 60        |
| <b>11</b> | <b>Závěr . . . . .</b>                                | <b>61</b> |
|           | <b>Literatura . . . . .</b>                           | <b>62</b> |
|           | <b>Seznam zkratk . . . . .</b>                        | <b>65</b> |
|           | <b>Přílohy . . . . .</b>                              | <b>66</b> |
| A         | Instalační příručka . . . . .                         | 66        |
| B         | Uživatelská příručka . . . . .                        | 68        |
| B.1       | Uživatel . . . . .                                    | 68        |
| B.2       | Ověřený uživatel . . . . .                            | 68        |
| B.3       | Tutor . . . . .                                       | 68        |
| B.4       | Administrátor . . . . .                               | 69        |



# 1 Úvod

Předmětem této bakalářské práce je doimplementovat a nasadit do produkčního prostředí informační systém pro správu a běh Student Support Centre, které běží pod záštitou Fakulty aplikovaných věd.

V rámci vývoje trendů moderní společnosti, kde je kladen důraz na snadnou a rychlou dostupnost dat, byl vznesen požadavek od vedoucího Student Support Centre na informační systém, který by usnadnil chod a správu celého projektu. Aktuální webové stránky projektu jsou na statickém přístupu, kdy jsou pravidelné výukové hodiny vypisovány na každý měsíc dopředu vedoucím profesorem, který musí přes komunikační kanály zjistit, kdy může student „tutor“ vyučovat. Cílem zavedení informačního systému má být tyto kroky pokud možno co nejvíce zautomatizovat nebo přenést rutinní a zjišťovací práci přímo na studenty, kteří jsou evidováni jako tutoři. Tito studenti si následně v systému budou sami zadávat hodiny, ve kterých mohou v SSC vyučovat a administrátor následně zadané lekce schválí do pravidelného rozvrhu nebo lekci zamítne. Žádosti o lekce lze díky grafické interpretaci (popisána v kapitole 7.2.1), která je strukturou totožná s Excelovou tabulkou využívanou v současné době, snadno přijímat. Zároveň je zde možnost kontroly počtu již schválených hodin u každého tutora na daný měsíc. Tyto pravidelné hodiny se následně automaticky propíše do pravidelného rozvrhu lekcí, který je dostupný online pro všechny návštěvníky webu. Důležitou součástí systému má být zároveň část pro sběr podnětů od studentů, kteří hodiny navštěvují a shromažďování statistik o vyučujících studentech a návštěvnosti hodin. Nové webové rozhraní má zobrazovat veškerá potřebná data pro chod Student Support Centre v responzivním designu, který je implementován použitím knihovny Bootstrap.

Vývoj tohoto informačního systému probíhal již v předmětu KIV/ZSWI, kde jsem byl součástí tříčlenného týmu, který si vývoj vzal na starosti. V průběhu vývoje jsme zjistili, že informační systém má být tak rozsáhlý, že nám hodinová dotace na tento předmět nedokázala pokrýt kompletní vývoj a byla nutná dohoda se zadavatelem a následné uvolnění práce jako bakalářské téma.

## 2 Moderní webové aplikace

Moderní webové aplikace jsou v dnešním rychlém světě velmi žádané. Největšími požadavky dnešní doby jsou bezpečnost uchovávaných dat a po vizuální stránce responzivní vzhled webu, jelikož stále větší procento lidí hledá informace při volných chvílích v autobuse či vlaku na zařízeních s malou zobrazovací plochou jako jsou např. mobily či tablety. Tyto aplikace pracují s uživatelskými daty a následně je koncovému uživateli v přívětivém grafickém rozhraní zobrazují. Kladen je i důraz na kompatibilitu s dalšími technologiemi a prostředími, levný provoz a snadnou údržbu celého systému.

### 2.1 Cloudové řešení

V dnešní době se velice rozšiřují služby takzvaného cloudového řešení, kdy není potřeba vlastnit fyzická zařízení jakou jsou výpočetní servery pro běh těchto webových aplikací či úložné prostory pro databáze a data v nich. Řešení v podobě fyzických zařízení je pro začínající podniky velkou počáteční investicí s potřebou vlastního dohledu nad hardwarem a aktualizacemi nainstalovaného softwaru. Naopak moderní cloudové řešení poskytuje volnost v podobě výběru různých dodavatelů, či výpočetních kapacit přímo na míru, kde se platí jen za výkon, který doopravdy potřebujete.

#### 2.1.1 AWS

Největším dodavatelem cloudových služeb je Amazon se svou službou Amazon Web Services. Tyto služby jsou dostupné po celém světě a jsou dodávány v řešení „pay-as-you-go“ (platíme pouze za to co využíváme). Amazon Web Services nabízejí rozsáhlou škálu možností, jak přenést svoji webovou aplikaci do cloudového řešení. Nejsnadnější možností je pronájem takzvaného virtuálního serveru, tedy stroje na kterém nám poběží celý operační systém podle našeho výběru a u kterého si dále můžeme konfigurovat jeho výkon podle počtu přiřazených jader procesoru či velikosti operační paměti RAM. Rozšiřující možností hostingu informačních systémů se na Amazonu stávají lambda funkce, které jsou levnější variantou k vlastnímu serveru a platí se pouze za volání každé jedné lambda funkce, která je spuštěná a je jim přiřazován výkon podle toho, jak jsou využívány a jaké paměťové nároky mají. Potřebnou databázi k informačnímu systému můžeme taktéž hostovat na Amazon Web Services, kde jsou nám nabízeny všechny známé rozšířené

verze databází. Na výběr máme ze standardních typů databází: relační databáze, databáze na bázi klíč-hodnota, dokumentační databáze a In-memory databáze, které uchovávají data v paměti pro rychlý přístup. Pro informační systémy jsou nejrozšířenější databáze relační, kde si můžeme vybrat ze všech známých databází jako jsou MySQL, Oracle, MariaDB či PostgreSQL až po profesionální řešení od Microsoftu v podobě SQL Serveru. Amazon Web Services garantují 99.999% dostupnost svých služeb po celém světě, což je umožněno díky výstavbě záložních datových center, kdy jednu oblast obsluhuje více takovýchto datových center [16].

### 2.1.2 Google

Google se svým Google Cloud řešením je druhým největším poskytovatelem těchto služeb. Stejně jako u Amazon Web Services nám Google nabízí pronájem virtuálních serverů, u kterých si můžeme konfigurovat jejich výkon podle našich potřeb a potřeb našeho informačního systému. Jejich databázové řešení je taktéž velice kvalitní a s garancí dostupnosti 99%. Najdeme zde nejrozšířenější databáze MySQL, PostgreSQL a SQL Server. Za všechny tyto databázové řešení se platí paušální poplatky za jednu hodinu spuštěné instance databáze stejně jako tomu je u Amazon Web Services. Řešení od Google se v kvalitě i ceně přibližuje řešení od Amazonu, kdy Google jako stále se rozvíjející společnost v oblasti IT rozšiřuje svoje služby a buduje nová datová centra po celé zemi.

## 2.2 ASP.NET Core

Jedná se o populární open-source framework pro vývoj webových aplikací na platformě .NET. Díky kompletnímu přepracování se pyšní větší výkonností než předcházející ASP.NET a je koncipován pro multiplatformní použití, jelikož může běžet na .NET Core. Zároveň s touto výhodou je možnost na jednom stroji spustit a provozovat vícero aplikací napsaných v ASP.NET Core, kdy je díky podpoře verzování možno mít tyto aplikace na rozdílných verzích ASP.NET Core. Jako u všech běžných frameworků je i u tohoto možnost volit mezi různými cykly podpory od vývojářů. Tudíž je jen na programátorovi, zda zvolí cestu nejnovější verze frameworku za cenu častějšího upgradování, nebo cestu vývoje aplikace na verzích s dlouhodobou podporou bez nových možností, které přišli s novější verzí [13].

## 2.3 MVC

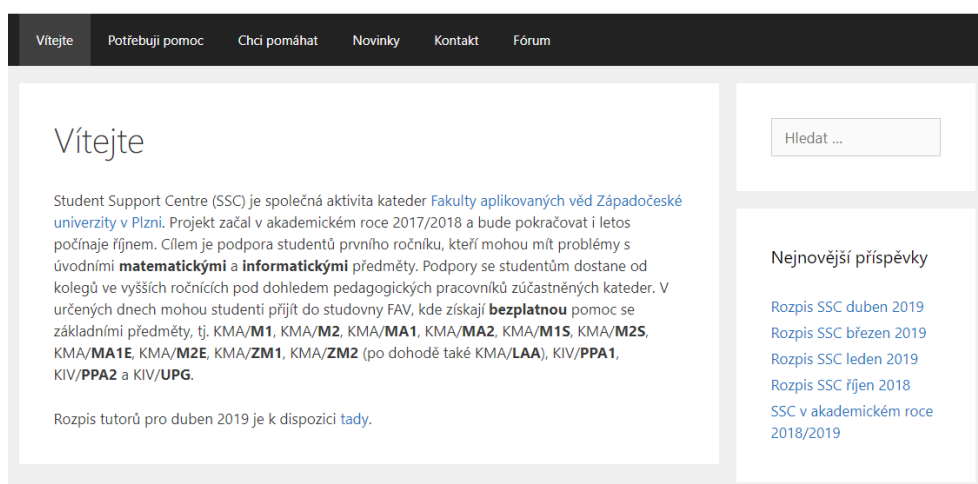
*Model-view-controller* neboli zkráceně znám pod zkratkou *MVC*. Jedná se o oblíbený architektonický vzor, který se používá zejména při implementování webových aplikací. Základní myšlenkou je oddělení logiky aplikace od výstupu, tudíž je architektura rozdělena do tří základních komponentů. Tento přístup řeší základní problém, kdy se nám v jednom souboru či třídě vyskytují najednou logické operace a renderování výstupu. Takovýto kód je následně špatně čitelný, protože jednotlivé části nejsou korektně formátovány a většina IDE prostředí si s tím nedokáže poradit. Komponenta *model* neví nic o výstupu a obsahuje veškerou logiku a práci s daty. V bakalářské práci bylo zároveň s MVC použito i *ORM* neboli *Objektově-Relační Mapování*, kdy modely přímo korespondují s databázovými tabulkami. *View* se následně stará o grafické zobrazování výstupu uživatelům. V souvislosti s ASP.NET se jedná o cshtml šablonu, která následně předaná data zobrazí. Obdobně jako *model* neví odkud data přišla. Komponenta *Controller* je posledním prvkem této architektury a jeho funkcionalitou je předávání dat mezi modelem a view [11].

# 3 Aktuální webový portál Student Support Centre

Aktuální webový portál, který používá Student Support Centre od svého založení je postaven na redakčním systému a rozhraní WordPress, které je uživatelsky přívětivé pro administrátora a nabízí mu mnoho možností ohledně úpravy textu či změn šablon vzhledu [19]. Toto aktuální řešení se dá na-

## Student Support Centre

...utopit vás nenecháme.



Obrázek 3.1: Aktuální webový portál Student Support Centra

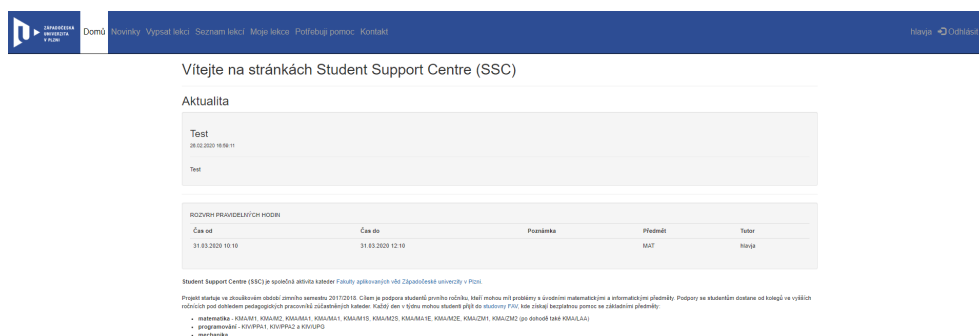
zvat jako statické, protože zde nedochází k žádné interakci mezi uživatelem a systémem, pouze pro přesun mezi adresami webu, viz obrázek 3.1, který zobrazuje aktuální stav webové prezentace využívané pro potřeby Student Support Centra. Tento portál neumožňuje automatické přihlašování nových zájemců o tutorství, kteří by měli zájem vyučovat své již absolvované předměty se známkami velmi dobře a výše. Jejich přihlášky probíhají pomocí emailové komunikace s administrátorem. Následné tvoření a zveřejňování rozvrhu na každý nový měsíc dopředu probíhá také pomocí emailové komunikace a následné zveřejnění probíhá zde na webu pomocí přidané novinky, ve které se nachází odkaz na PDF soubor, ve kterém jsou pravidelné hodiny následujícího měsíce vypsány do tabulky s jmény tutorů, kteří danou vyučující hodinu povedou. Dalším nedostatkem, který by se u WordPressového řešení webové aplikace musel řešit pomocí takzvaných pluginů, které

se ve WordPressu používají [9], je umožnit studentům ZČU zažádat o extra lekci daného předmětu, na kterou nemohou dorazit v hodinách vypsáných v pravidelném rozvrhu. Tento systém by požadoval propojení WordPressového řešení s jednotným přihlašovacím systémem Orion, který je na ZČU využíván. Dalším požadavkem na funkčnost informačního systému zůstává také sběr podnětů ke zlepšení hodin, který by zde musel být řešen přes moduly obsahující dotazníky či jiné možnosti odesílání dat do databáze. Vyřešení ukládání dat do databáze by v případě využití redakčního systému WordPress mohlo být usnadněno pouhým rozšířením již stávající databáze, na které celý redakční systém běží o potřebné tabulky a vazby. Toto řešení by odstranilo nutnost mít vlastní běžící databázový server mimo aplikaci a nutnost jeho údržby a správy. Jelikož je tento redakční systém psán v jazyce PHP s databází MySQL, je pro uživatele nutné využívat předem daný framework WordPressu pro doprogramování extra modulů či částí webu.

# 4 Seznámení s částečnou implementací

## 4.1 Úvod

Informační systém a webová aplikace pro Student Support Centre vznikala již jako semestrální týmový projekt v předmětu KIV/ZSWI. Zde jsme společně v týmu s Filipem Háchou a Lucií Tauchenovou shromáždili všechny potřebné požadavky na aplikaci, od našeho zadavatele, a následně zahájili vývoj na takzvané „zelené louce“. Jelikož jsme od zadavatele tohoto projektu nedostali zadané technologie, ve kterých máme informační systém vyvíjet, byly po debatě v týmu zvoleny technologie .NET a ASP a jako databázová vrstva aplikace bylo zvoleno řešení od Oracle. Programovací technologie .NET podporuje a je snadno použitelná pro architekturu odpovídající návrhovému vzoru MVC (Model-View-Controller). Následující obrázek 4.1 zobrazuje převzaté webové rozhraní.



Obrázek 4.1: Implementované webové rozhraní v předmětu ZSWI

## 4.2 Rozdělení týmu

V průběhu návrhu a implementace práce jsme v týmu měli rozdělené úkoly, abychom pracovali efektivněji a pokud možno využili naše vědomosti na správných místech. Již ve druhém ročníku jsme měli brigády v oboru, a tak jsme každý načerpali vědomosti v různých oblastech programování.

- *Lucie Tauchenová* - komunikace se zadavatelem a kontakty na CIV, administrativní práce v rámci předmětu ZSWI, návrh databázového

modelu, testování aplikace

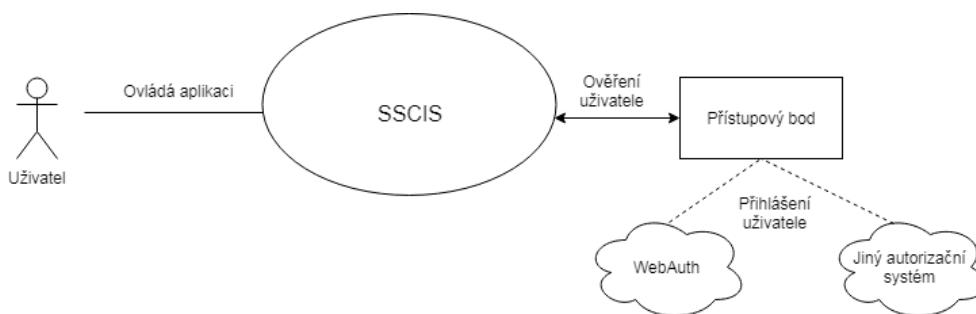
- *Filip Hácha* - návrh aplikace a architektury, návrh databázového modelu, hlavní programátor back-endu
- *Jakub Hlaváč* - návrh a implementace front-endu, implementace back-endu

### 4.3 Rozsah projektu

Celý informační systém se má skládat z vytvoření kompletní struktury databáze, která bude potřebná k zajištění bezproblémového chodu. Dále se jedná i o webové rozhraní a back-endovou část aplikace, která je implementovaná pomocí ASP.NET frameworku. Webové rozhraní bude vyvíjeno a podporováno primárně pro webový prohlížeč Google Chrome (na kterém bude také otestováno) a bude reflektovat nejmodernější trendy na vzhled a strukturu. Z možných programovacích jazyků frameworku .NET byl vybrán C#, který je s Javou jedním z nejrozšířenějších a nejmodernějších programovacích jazyků.

### 4.4 Kontext systému

Nový informační systém nahradí stávající webové stránky Student Support Centra a obsah těchto webových stránek se přesune do nového systému. Kontext systému je znázorněn na obrázku 4.2



Obrázek 4.2: Kontext systému

### 4.5 Specifikace požadavků

Specifikace požadavků na nový informační systém pro Student Support Centre spočívá v jednoduchém sestavování pravidelných měsíčních rozpisů ho-



din, následnou evidenci studentů, kteří na hodiny přijdou, statistiky odučených hodin tutorů a možnost zadávání zpětné vazby na příslušné hodiny ve formě feedbacků. Důležitým požadavkem od zadavatele byla integrace s jednotným přihlašovacím systémem Orion, který je využíván Západočeskou univerzitou v Plzni.

- Informační systém nebude uchovávat žádná hesla uživatelů.
- Základním požadavkem je registrace nových uživatelů přes přihlášení pomocí Orion účtu a následná možnost takto registrovaných uživatelů vyplnit dotazník s žádostí o přiřazení „tutorství“ daného předmětu. Každý úspěšně zaregistrovaný a následně přihlášený uživatel má možnost požádat o extra lekci nad rámec pravidelného rozvrhu, pokud mu žádná z pravidelných lekcí časově nevyhovuje.
- Tvorba pravidelného rozvrhu hodin pomocí jednoduchého schvalovacího mechanismu, kdy správce vybírá ze zadaných hodin tutorů ty, které se nejvíce hodí pro potřeby Student Support Centra a jejich následné uveřejnění ve webové aplikaci všem návštěvníkům. Do tohoto rozvrhu se budou následně promítat i veškeré přijaté extra lekce s poznámkou, jaký daný problém se na příslušné hodině bude probírat. Ve zveřejněném pravidelném rozvrhu hodin budou vyznačeny lekce, které jsou z určitých důvodů zrušeny, takovéto lekce budou vypsány tučným písmem a podbarveny červenou barvou.
- Odesílání notifikací emailem při různých definovaných akcích v systému. Mezi tyto akce spadá především zasílání notifikace administrátorům při vyplnění nové žádosti o tutorství, žádosti o extra lekci a její následné přijetí daným tutorem. Notifikace registrovaných uživatelů při schválení jejich přihlášek o tutorství či přijetí žádosti jejich vypsání lekce jedním z tutorů, kteří daný předmět mohou vyučovat. Implementace popsána v kapitole 7.5.
- Přihlášený uživatel musí mít možnost procházet veškeré svoje žádosti o extra lekce. Uživatel se statusem tutora je schopen procházet veškeré svoje vypsání hodiny, které byly odeslány do systému a čekají na schválení administrátorem.
- Systém pro sběr statistik z vyučovaných hodin. Zde se jedná především o upozorňování tutorů, kteří měli svoji pravidelnou hodinu v rozvrhu, aby zadali do systému počet studentů, kteří se na danou lekci dostavili a využili tak služeb, které nabízí Student Support Centre. Automatické

vypočítávání odučených hodin pro každého tutora a možnost vygenerovat takovýto seznam pro všechny tutory, kteří jsou v systému za předešlý měsíc pro potřeby následného finančního účetnictví.

- Možnost tutorů vygenerovat pro svoji pravidelnou lekci URL odkaz, na kterém mohou návštěvníci dané lekce zapsat svůj názor na poskytnuté služby. Tento URL odkaz je generován i v podobě QR kódu, přes který se uživatelé dostanou snadněji prostřednictvím svého chytrého telefonu k danému dotazníku. URL by nemělo být v prostém tvaru, aby nedocházelo ke zneužívání.
- Uživatel spadající do třídy administrátorů má možnost snadno přes webové rozhraní zadat novou aktualitu na web, či editovat již zadanou. Takovému uživateli je jako jedinému umožněno zrušit již schválenou lekci v systému. Po zrušení zůstane tato lekce v databázi se změněným statusem, aby se následně propala do rozvrhu jako zrušená.

## 4.6 Provozní prostředí

Celý informační systém má běžet na výpočetní technice, která se nachází na Západočeské univerzitě v Plzni jmenovitě na Katedře informatiky a výpočetní techniky. Na této katedře bude poskytnut server, na kterém poběží webová aplikace a příslušný databázový server uchovávající data celého informačního systému.

## 4.7 Publikum

Tento informační systém by měl sloužit především pro studenty Západočeské univerzity v Plzni, kteří navštěvují některý předmět Fakulty aplikovaných věd, který je v rámci Student Support Centra podporován jako jeden z předmětů, který se zde vyučuje. Dále je hlavním konzumentem dat z tohoto informačního systému administrátor aplikace, který má veliký zájem na vytvořených statistikách o tutorech či vyučovaných hodinách, ke kterým je umožněno sbírat zpětnou vazbu v podobě feedbacků a návštěvnost jednotlivých lekcí, kterou do systému po odučené hodině zadává tutor.

## 4.8 Databáze

Požadavky na použití daného systému řízení báze dat nebyl určen. Na začátku projektu v rámci předmětu ZSWI jsme databázi navrhli pro Oracle

Database 12c a bylo předpokládáno, že tato databázová instance poběží ve stejném prostředí jako informační systém. V rámci této bakalářské práce došlo k přesunutí databáze pod MySQL a restrukturalizaci databázového modelu pro použití v nových případech užití. Více v kapitole 6.

## 4.9 Třídy uživatelů

Celý informační systém je založen na přístupových právech, které se dělí do několika tříd. Každý návštěvník webové aplikace je rozřazen do jedné z těchto tříd a podle toho jsou mu nastavena práva a umožněny přístupy do systému, kde může provádět předem definované úkony náležící jeho skupinovému zařazení. Informační systém obsahuje čtyři třídy uživatelů:

- **Administrátor, správce**

Jedná se o vlastníka či správce celé aplikace. Přístupová práva k hlavnímu administrátorskému účtu jsou uložena v databázi a není potřeba registrace (popsáno v kapitole 7.6). Z tohoto účtu je možné administrátorská práva přidělit již přihlášenému uživateli, který k přihlášení využil Orion login a má záznam v databázi. Uživatel zařazený do této třídy má přístup ke všem částem informačního systému a je mu pomocí webové aplikace dovoleno komfortně zobrazovat, měnit či mazat povolené záznamy z databáze. Jeho hlavní náplň práce v systému je akceptace žádostí o tutorství a následné přijímání či zamítání pravidelných lekcí, které jsou zadány uživateli, kteří mají status tutora. Toto přijímání lekcí lze provádět v grafickém rozhraní, které se podobá rozvrhu a pro daný měsíc je také možnost vygenerovat soubor .csv, který obsahuje data reflektující tento rozvrh (více v kapitole 7.2.2). Aplikací je administrátorovi umožněn přístup do veškerých statistik, který informační systém nabízí, zobrazování výpisu feedbacků či jeho export z aplikace ve formátu .csv. Správce či administrátor má jako jediný právo na zrušení členství tutora a při zneužívání systému extra lekcí může celou část informačního systému, která tyto extra lekce obsluhuje, vypnout. Registrace/přiřazování práv administrátorů je možná jen přes účet v třídě administrátor.

- **Registrovaný uživatel**

Registrovaný uživatel získává přístup do částí systému, ve kterých může požádat o „tutorství“ pomocí dotazníku, ve kterém vyplní své absolvované předměty a z nich dosažené známky. Další zpřístupněnou sekcí tohoto uživatele je možnost, požádat si o extra lekci mimo pravidelný rozvrh hodin. Tato možnost je povolena pouze registrovaným

uživatelům z důvodu snadné dohledatelnosti zneužívání tohoto vstřícného systému Student Support Centra.

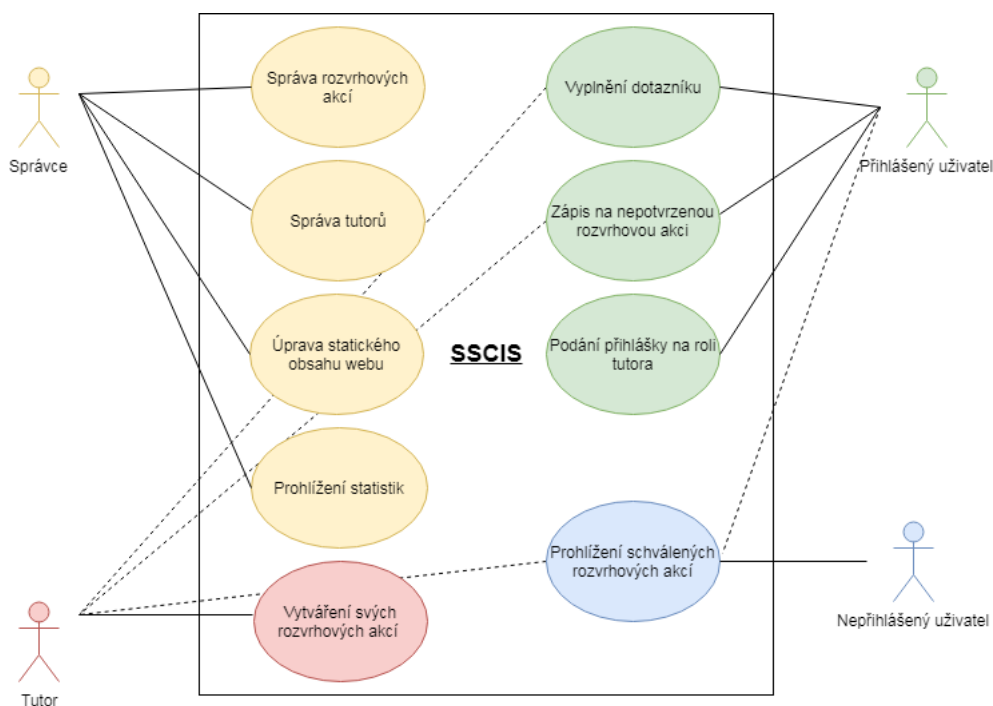
- **Tutor**

Jedná se o registrovaného uživatele, kterému byla administrátorem aplikace přidělena práva pro možnost vypisování vlastních lekcí z množiny předmětů, které mu byly schváleny v přihlášce. Tyto vypsané hodiny musejí stále projít schvalovacím procesem, který provádí administrátor, aby se co nejvíce zamezilo zneužívání systému.

- **Neregistrovaný uživatel**

Běžný uživatel, který navštíví stránky Student Support Centra. Je mu dovoleno procházet veškeré veřejné části webu, zobrazit rozvrh pravidelných lekcí a nejdůležitější novinky, které byly ohledně Student Support Centra zveřejněny.

Zobrazení příslušných funkcí a částí systému uživateli je znázorněno na obrázku 4.3 a následně jsou tyto případy použití popsány v kapitolách 4.10 a 7.



Obrázek 4.3: Kontextový diagram informačního systému

## 4.10 Implementované funkce systému

Celý projekt informačního systému pro Student Support Centre byl již z části vyvinut v předmětu KIV/ZSWI. Zde jsme dokázali implementovat zásadní části tohoto systému a zprovoznit ho v beta verzi, která byla schválena zadavatelem a následně bylo po domluvě rozhodnuto o doimplementování zbylých částí projektu v rámci této bakalářské práce.

### 4.10.1 Integrace s webAuth

Největším problémem při vývoji byl požadavek na propojení celého informačního systému s jednotným přihlašovacím portálem Orion, který využívá celá Západočeská Univerzita v Plzni. Po debatě se správci tohoto portálu nám bylo sděleno, že celý systém přihlašování je napsán v jazyce PHP a dosud se nikomu nepodařilo na tento systém napojit jiný, který je napsán v odlišném programovacím jazyce a při použití jiných programovacích technik. Tato skutečnost nebyla nikomu předem známá a s ohledem na již probíhající vývoj systému nebylo proveditelné již implementované části přepisovat do jiného programovacího jazyka. Jakožto dodavatel systému jsme byli nuceni na tuto skutečnost reagovat a po diskuzi se správci Orion portálu na ZČU jsme došli k velice uspokojujivému řešení, jak tuto situaci vyřešit nenásilným způsobem. Řešení celé situace spočívalo v tom, že nám bude na CIVu zpřístupněn proxy server, na který webová aplikace Student Support Centra bude přesměřovávat veškeré dotazy na přihlášení do systému. Tento proxy server bude běžet na HTTPS protokolu, abychom mohli jednoznačně v aplikaci určit, zda odpověď s daty přišla právě z určeného proxy serveru a zamezilo se případným pokusům o nabourání do systému pomocí podstrčení falešných údajů, které nám proxy server posílá. Jelikož pro naše účely běhu informačního systému nepotřebujeme znát žádné citlivé osobní údaje studentů, vystačíme si s emailovou adresou, jménem a skutečností, že se student do systému ZČU dokáže přihlásit a tím pádem jej lze považovat za studenta. Proto byl proxy server nastaven tak, že nám tyto údaje v souladu s nejnovějšími normami GDPR poskytuje přes zabezpečený komunikační kanál a informační systém Student Support Centre je schopen určit o jakého studenta se jedná a tyto data bezpečně uložit do databáze či v ní pro již zaregistrovaného studenta vyhledat patřičné záznamy.

### 4.10.2 Registrace

Registrace do systému tedy probíhá pomocí jednotného přihlašování pomocí Orion účtu, kdy se do databáze informačního systému Student Support Cen-

tra uloží jen potřebné záznamy o přihlášeném uživateli a při úspěšném přihlášení přes Orion počítáme s tím, že přihlašovaný uživatel je studentem Západočeské univerzity v Plzni a může využívat služby Student Support Centra.

### 4.10.3 Přihlášení

Přihlašování uživatelů je spravováno pomocí přihlašování přes Orion účet, kdy je uživatel přesměrován na proxy server s tímto přihlašovacím systémem a po úspěšném přihlášení je uživatel přesměrován zpět na naši webovou aplikaci a v headeru dotazu jsou nám po přesměrování sděleny nutné informace, pro validaci úspěšného přihlášení uživatele do systému ZČU. Následně je takovýto uživatel vyhledán v databázi Student Support Centra a pokud je úspěšně nalezen záznam odpovídající pokusu o přihlášení, jsou tomuto uživateli uloženy do session ve webovém prohlížeči následující data:

- UserID - obsahující id daného uživatele v databázi Student Support Centra
- Hash - vygenerovaný hash klíč používaný ke spárování záznamu v databázi se záznamem v session
- Role - role přihlášeného uživatele
- Login - login username přihlášeného uživatele

Dále je uložen záznam v tabulce **sscis\_session**, kam je uložen vygenerovaný hash klíč pro daného uživatele s časem přihlášení a je k němu dopočítán čas expirace. Doba expirace hash klíče je dána parametrem, který je uložený v databázové tabulce **sscis\_param** pod klíčem *SESSION\_LENGTH*.

Pokud takovýto uživatel v databázi není nalezen dojde k jeho registraci popsané v bodě 4.10.2

### 4.10.4 Podání přihlášky na roli tutora

Implementace žádosti o tutorství je provedena pomocí jednoduchého dotazníku, který je namapován na objekt v databázi. Tento dotazník obsahuje select box pro vybrání podporovaných předmětů Student Support Centrem a následně zadání známky do textového pole pod ním. Vespod dotazníku se nachází tlačítko na přidání dalšího záznamu do aktuální přihlášky o tutorství. Na jednu takto vytvořenou přihlášku je možno odeslat více absolvovaných předmětů, pokud má student zájem vyučovat několik vědních oborů.

K obsluze tohoto případu užití slouží controller `TutorApplicationController`, ve kterém jsou implementovány metody pro obsluhu veškerých operací, které jsou nad žádostí o tutorství volány. Controller obsluhuje vytvoření dané žádosti o tutorství a založení příslušného záznamu do databáze, následné zobrazení dat z databáze administrátorům, kteří danou žádost o „tutorství“ vyřizují a metody ke schválení či zamítnutí žádosti, které opět provedou dané změny v databázi aplikace. Je zde i metoda pro zobrazení chybové hlášky, která se zobrazuje při navštívení webové stránky s možností odeslání dotazníku pro žádost o roli tutora.

#### 4.10.5 Vytváření vlastních rozvrhových akcí

Zápis hodin je možný pro registrované uživatele spadající do třídy tutorů. Tito uživatelé mají přístupnou část webu, kde pomocí dotazníku, který je mapován na objekt v databázi, zadají potřebné údaje pro založení žádosti o pravidelnou lekci na tento měsíc. Dotazník obsahuje několik polí pro zadání nutných informací pro založení žádosti o lekci. V prvním poli, které je typováno na *Date*, je nutno zadat validní datum, které je větší než aktuální datum dne, ve kterém je dotazník vyplňován. Následuje pole typované na *Time*, do kterého je nutno vložit čas, od kterého daný tutor může vyučovat. Délka takovéto hodiny je dána pevným systémovým parametrem, který je uložen v databázové tabulce `sscis_param` pod klíčem `STANDARD_EVENT_LENGTH`. Následující select box dá přihlášenému uživateli na výběr předměty, které má schválené v přihlášce, a ten je povinen jeden z nabízených předmětů vybrat. Select box s přiřazováním tutora k dané lekci je zde jen pro uživatele s administrátorskými právy, kdy takovýto administrátor může na žádost kteréhokoliv tutora danou lekci vypsat místo něho a tudíž musí vybrat z tohoto select boxu, kde se zobrazí všichni aktivní tutoři toho, pro koho takovouto žádost o hodinu vyplňuje, viz obrázek 4.4.

#### 4.10.6 Správa uživatelů a tutorů

Tato sekce systému je přístupná pouze administrátorům a správcům, kteří zde spravují veškeré uživatele, kterým mohou přiřazovat třídy uživatelů, nabízené informačním systémem. Pod tento případ užití spadá také správa přihlášek uživatelů, kteří se hlásí do role tutorů.

Na správu veškerých uživatelů je ve webové prezentaci vytvořena speciální URL adresa, na které je zobrazen seznam všech uživatelů, kteří se do systému přihlásili přes svůj Orion účet a mají záznam v databázi Student Support Centra. Tento pohled spravuje controller s názvem `UserController`,

## Vytvořit

Vypsání lekce

|   |   |
|---|---|
| Datum   | <input type="text" value="10.04.2020"/> |
| Čas od  | <input type="text" value="12:00"/>      |
| Předmět   | <input type="text" value="INF"/>        |
| TUTOR   | <input type="text" value="hlavja"/>     |
| Počet opakování                                 | <input type="text" value="1"/>          |
| <input type="button" value="Vytvořit událost"/> |   |

### Legenda:

Počet opakování - kolikrát se stejná lekce do systému vypíše s týdenním opakováním ve stejnou dobu  
(Např: Počet opakování = 2, Datum vypsání = 1.1.2020 -> Lekce se mi vypíše dvakrát a to 1.1 a 8.1 ve stejný čas.)

Aktuální délka vypisované hodiny - je nastavena na 2 h

[Zpět](#)

Obrázek 4.4: Vytvoření nové hodiny do pravidelného rozvrhu.

který obsahuje veškeré potřebné metody k manipulaci s daty uživatelů v databázi. U těchto uživatelů je možno zobrazit informace, které jsou o každém z nich uloženy v databázi. Tyto údaje je možno následně upravovat pomocí formuláře, kdy z formuláře bude nejdůležitější select box s rolí uživatele (aktuální role je v select boxu vybrána vždy jako defaultní při otevření editačního formuláře). Tuto roli může administrátor změnit výběrem jiné role ze select boxu a tato možnost je zde z důvodu potřeby odebrat určitému uživateli práva tutora, pokud se proviní proti pravidlům Student Support Centra či ukončí své studium.

Správa přihlášek tutorů probíhá na speciálním view webové prezentace, které je k tomuto účelu určené a tento pohled obsluhuje controller s názvem `TutorApplicationsController`, který v sobě obsahuje metody pro schválení či zamítnutí dané přihlášky. Zde jsou v seznamu pod sebou vypsány veškeré evidované a nezpracované přihlášky registrovaných uživatelů, kteří se chtějí stát tutory. V základním seznamu žádostí je zobrazen datum, kdy byla daná žádost o tutorství podána a login uživatele který ji podal. Následně se proklikem přes tlačítko *Detail* dostaneme na detail dané žádosti. V detailu se nám zobrazí předměty, které daný uživatel při vyplňování dotazníku o tutorství zadal, ve spodní části obrazovky se administrátorovi zobrazí tlačítka pro schválení či zamítnutí žádosti. Je zde i tlačítko pro návrat na předchozí výpis veškerých žádostí o tutorství. Při schválení žádosti je volána metoda `Accept`, která v databázi v tabulce `sscis__user` u daného uživatele změní potřebné záznamy:



- Role - nastaví roli tutora
- AcceptedDate - aktuální datetime, kdy byla žádost přijata
- IsAccepted - nastaví tuto hodnotu na true
- AcceptedBy - nastaví id administrátora, který danou přihlášku schválil

Dále jsou provedeny změny v databázové tabulce Approval (6.9.1), kde se k uživateli přiřadí předměty, které může vyučovat. Při zamítnutí žádosti jsou v databázi provedeny změny jen v tabulce **sscis\_user** a to nastavení hodnoty *IsAccepted* na *false* a *AcceptedBy* je nastaveno na id daného administrátora či správce, který přihlášku vyřizoval.

#### 4.10.7 Správa statického obsahu webu

Statický obsah webu jako jsou pevné texty na stránkách či texty v zápatí stránek je možno upravovat pouze ve zdrojovém kódu. S touto skutečností byl zadavatel obeznámen a byla poskytnuta podrobná dokumentace jak tyto texty jednoduše změnit a udělat deploy aktualizované verze webové prezentace na hostingový server. Implementovaná úprava textů na webových stránkách přes webové rozhraní je možná pouze u Aktualit, které mají na webu svoji vlastní URL. Tyto úpravy obsluhuje controller `ContentsController` obsahující veškeré potřebné metody pro editaci či smazání dané aktuality. Jedná se jeden z hlavních požadavků na webové rozhraní, kdy je potřeba reagovat na změny v krátkém časovém intervalu a měnit tyto aktuality pokaždé ve zdrojovém kódu by bylo velice neefektivní, pracné a zdlouhavé. Proto se veškeré aktuality ukládají jako text do databáze, přesněji do tabulky **sscis\_content** popsané v kapitole 6.4. Následně jsou z databáze zobrazovány ve webovém rozhraní na příslušné stránce aktualit a nejnovější aktualita je také zobrazována na úvodní stránce webu. Tyto aktuality lze editovat či smazat pomocí tlačítek, která se zobrazí na stránce s výpisem všech aktualit pouze přihlášenému uživateli s administrátorskými právy.

# 5 Přepsání aplikace do .NET Core

Tématem této bakalářské práce bylo také po dohodě se zadavatelem požadavku na informační systém veškeré jeho již implementované části přepsat z architektury .NET Framework do .NET Core 1.0 a vyšší. Tato migrace celé aplikace umožňuje větší variabilitu při výběru softwaru, na kterém celý systém poběží díky multiplatformnosti technologie .NET Core.

## 5.1 .NET Standard

.NET Standard je vylepšená verze Portable Class Library, kterou by měl v nejbližších letech nahradit. Definuje standardizované rozhraní a to přímo mapuje na implementaci dané platformy. Tímto přístupem by mělo být zabráněno meziplatformní fragmentaci. Tudíž si vývojář může být jist, že všechno co předepisuje určitá verze .NET Standardu je implementována v příslušné verzi .NET Frameworku i .NET Core.

## 5.2 .NET Core

.NET Core je poměrně mladá technologie, která se na trhu objevila před dvěma lety. Přesněji 7.3.2017 byla oficiálně vydána první verze s označením .NET Core 1.0. Jedná se o open-source platformu, kterou si pod svá křídla vzal Microsoft a je jím podporována a vyvíjena. .NET Core si dalo za cíl vytvořit na platformě nezávislé prostředí, které by bylo spustitelné na libovolném počítači bez ohledu na nainstalovaný operační systém či softwarovou výbavu. V tuto chvíli jsou verzi 2.0 podporovány distribuce Windows, které jsou stále podporovány ze strany Microsoftu, podporován je i MacOS či nejrozšířenější a nejznámější distribuce Linuxu. Důležitou součástí .NET Core je také zpětná kompatibilita s technologiemi .NET Framework a Xamarin. Tato kompatibilita je umožněna díky knihovnam .NET Standard [27].

.NET Core aplikace lze psát ve vícero jazycích, mezi nejznámější patří programovací jazyk C#. Jde o jazyk vycházející z rodiny jazyků C a je velice podobný jazykům C, C++, Java či JavaScript. Jedná se o moderní, jednoduchý jazyk podporující objektově orientované programování. Podpora je zde i pro jazyky F# (jedná se o multiplatformní jazyk postavený na open

source pro programování v .NET) či jazyk Visual Basic [18].

Pro rozšíření funkcionality jsou v .NET Standardu přístupné takzvané balíčky NuGet, které rozšiřují možnosti .NET Core. Tyto balíčky jsou vyvíjeny přímo Microsoftem, či jinými společnostmi nebo jednotlivci, kteří těmito balíčky přidávají do .NET Core nové knihovny potřebné pro určité funkcionality. NuGet prostředí obsahuje přes 90 000 balíčků [12]. Nevýhoda oproti .NET Frameworku spočívá v tom, že je potřeba při updatu na novější verzi .NET Core aktualizovat i veškeré NuGet balíčky tak, aby jejich podporovaná verze odpovídala verzi aktuálního core. Zde se můžou vyskytnout problémy s uspěchaným přechodem na novou verzi, kdy námi využívané balíčky NuGet nebudou aktualizovány a při přechodu na novější verzi přestanou fungovat.

### 5.3 Rozdíly oproti .NET Frameworku

Největším rozdílem je jeho univerzálnost v podobě multiplatformnosti, kdy je možno aplikace napsané pro .NET Core hostovat či spustit na nejrozšířenějších systémech. Jedná se stále o vyvíjející se technologii, která je poměrně mladá, takže na rozdíl od .NET Frameworku je na internetu k dohledání velice málo návodů či tutoriálů. Pro programátora, který již pracoval s aplikacemi napsanými v .NET Frameworku to ale není žádný problém, protože jsou zde jen minimální změny co se programovací struktury týče a je k dispozici velice podrobná a dobře strukturovaná dokumentace, ze které jsou veškeré změny provedené v .NET Core zřejmé.

### 5.4 .NET Core výhody

K multiplatformnímu využití této technologie je zajisté potřeba přidat také jeho snadnou škálovatelnost. Je zde také výhoda v podobě mladé technologie, která se neustále vyvíjí a narozdíl od .NET Frameworku jsou zde opraveny některé technologické chyby, které v .NET Frameworku kvůli jeho zachování zpětné kompatibility opravit nelze. Další výhodou této technologie je také její snadná škálovatelnost, kterou určitě oceníme při potřebě mít vysoce výkonné systémy. Tuto technologii lze také spustit bez větších problémů v Docker kontejnerech.

Z programátorského hlediska a z hlediska výkonnostního lze tuto technologii doporučit pro začínající projekty, které je potřeba provozovat multiplatformně. Zároveň je .NET Core doporučován pro aplikace, u kterých je potřeba vysokého výkonu [10].

## 5.5 Struktura projektu

Celý projekt byl vyvíjen ve Visual Studio 2019 se studentskou licencí, kterou máme díky spolupráci s Microsoft dostupnou. V tomto IDE je vytvořeno jedno *řešení*, což je v podstatě kontejner pro vícero projektů a nese informace o sestavení. *Projekt* obsahuje veškeré soubory, které jsou zkompileovány do spustitelné aplikace. Obsahuje také konfigurační soubory a nastavení kompilátoru. V rámci implementace jsem založil dva projekty: **ISSSC** a **ISSSCResources**.

### 5.5.1 ISSSC

Hlavní projekt, ve kterém jsou zdrojové kódy celé aplikace. K jeho zkompileování je potřeba projektu **ISSSCResources**, který je přidán jako dependency a jsou z něj načítány konstanty. Struktura obsahuje soubory členěné do logických celků, podle kterých jsou pojmenovány složky, které jsou vypsány v tabulce 5.1.

| Soubor/složka    | Popis   |
|------------------|---|
| wwwroot          | obsahuje soubory důležité pro webové rozhraní               |
| wwwroot/images   | složka obsahující grafiku webového rozhraní                 |
| wwwroot/js       | veškeré javascript funkce                                   |
| wwwroot/lib      | složka obsahující webové knihovny (např. Bootstrap, jQuery) |
| Attributes       | třídy pro autorizaci  |
| Class            | definice tříd aplikace                                      |
| Controlllers     | definice veškerých controllerů aplikace                     |
| Database         | zakládací scripty databáze                                  |
| Models           | složka obsahující objekty mapované na tabulky databáze      |
| Models/Meta      | obsahuje pomocné objekty                                    |
| Views            | obsahuje veškeré layouty pohledů                            |
| appsettings.json | konfigurační soubor aplikace                                |

Tabulka 5.1: Struktura projektu ISSSC

### 5.5.2 ISSSCResources

Jedná se o projekt, uchovávající definice konstant pro celý projekt. Zvolení zvláštního projektu pro uchovávání těchto konstant bylo z důvodu možného

snadnějšího překládání webu do cizího jazyka, tak aby ho mohli využívat i zahraniční studenti.

### 5.5.3 Statistiky

Jelikož byl projekt přebírán již rozpracovaný, byly nad řešeními pro představu o rozšíření aplikace vypočítány statistiky. Ač se nejedná o velký nárůst délky výkonného kódu, jelikož při implementaci docházelo k přepisování metod tak, aby se v kódu nevyskytovaly duplicitní sekce, funkčnost celé aplikace se výrazně rozšířila a přibylo mnoho nových případů užití. Výsledné statistiky jsou zobrazeny v tabulce 5.2.

| Zkoumaní oblast             | ZSWI | Bakalářská práce |
|-----------------------------|------|------------------|
| počet řádek zdrojového kódu | 6790 | 15630            |
| počet nemapovaných objektů  | 6    | 15               |
| počet controllerů           | 10   | 10               |
| počet tříd                  | 14   | 19               |
| počet views                 | 59   | 68               |

Tabulka 5.2: Statistiky projektu v rámci ZSWI oproti bakalářské práci

## 5.6 Problémy při migraci

Při migraci již implementovaných částí aplikace jsem narazil na několik významných problémů. Jedním z hlavních problémů byla „mladost“ technologie, kdy bylo ve staré implementaci využíváno balíčků NuGet. Potřebné balíky pro .NET Core nebyly v době implementace dostupné či za ně bylo potřeba platit. Dále se zde našly problémy v podobě chybějících knihoven či úpravě metod v nich, kdy se názvy metod trochu obměnily oproti .NET Frameworku či bylo potřeba použít jiné parametry pro funkčnost.

### 5.6.1 Databáze

Informační systém byl navržen tak, aby pracoval s databází Oracle, ve kterém tato databáze byla vytvořena. Při migraci jsem narazil na problém, kdy byla databáze Oracle v aplikaci namapovaná pomocí NuGet balíčků vyvíjených přímo společností Oracle. Je zajímavé, že tyto balíčky jsou pro .NET Framework zcela zdarma, kdežto pro .NET Core jsou zpoplatněny nemalou částkou. Jelikož byly použité techniky již domluvené muselo zde dojít ke kompromisu a změně technologie.

Rozhodl jsem se využít databázového řešení MySQL, které je na trh do-  
dáváno pod bezplatnou licencí GNU GPL (přesněji pod verzí 2, která se  
značí GPLv2). GNU General Public License nazývaná jako „svobodný soft-  
ware“ umožňuje dané dílo modifikovat, kopírovat či dále rozšiřovat bez ja-  
kýchkoliv omezení při zachování GNU licence. [17] Díky rozšířenosti MySQL  
na Katedře informatiky a výpočetní techniky byl změněn požadavek ohledně  
hostingu databáze na stejném serveru jako je informační systém. Pro potřeby  
této bakalářské práce a následně pro chod informačního systému v produkč-  
ním prostředí byla databáze založena na databázovém serveru KIV. Na tento  
server se lze připojit jen ze školní sítě, což může pomoci odvrátit základní  
útoky na infrastrukturu.

Připojení do databáze a následné namapování celé této databáze do .NET  
Core je zprostředkováno pomocí balíčků NuGet, které jsou vyvíjeny přímo  
pro MySQL a s touto databází umí kompatibilně spolupracovat. Při prvním  
spuštění tohoto balíčku jsou vytvořeny veškeré potřebné modely tabulek  
a je vytvořena obsluhující třída, která mapuje veškeré entity databáze a při  
programování nám umožňuje s databází pracovat jako s objekty.

Pro načítání všech souvisejících tabulek z databáze bylo využito balíčku  
*Microsoft.EntityFrameworkCore.Proxies*, který umožňuje opožděné načítání  
dat z databáze. Toto řešení umožňuje takzvané *lazy-loading*, kdy je načí-  
tání dat z databáze odloženo až do doby kdy jsou potřeba. Pro použití  
tohoto přístupu je zapotřebí při konfiguraci připojení k databázi zavolat  
metodu `UseLazyLoadingProxies()` [20]. Tento přístup k databázové  
vrstvě může v jistých scénářích zrychlit chod aplikace a načíst veškerá re-  
levantní data až v době, kdy jsou potřeba. Zároveň se jedná o snadnější  
implementaci při přepisování stávajícího kódu.

# 6 Návrh databázového modelu

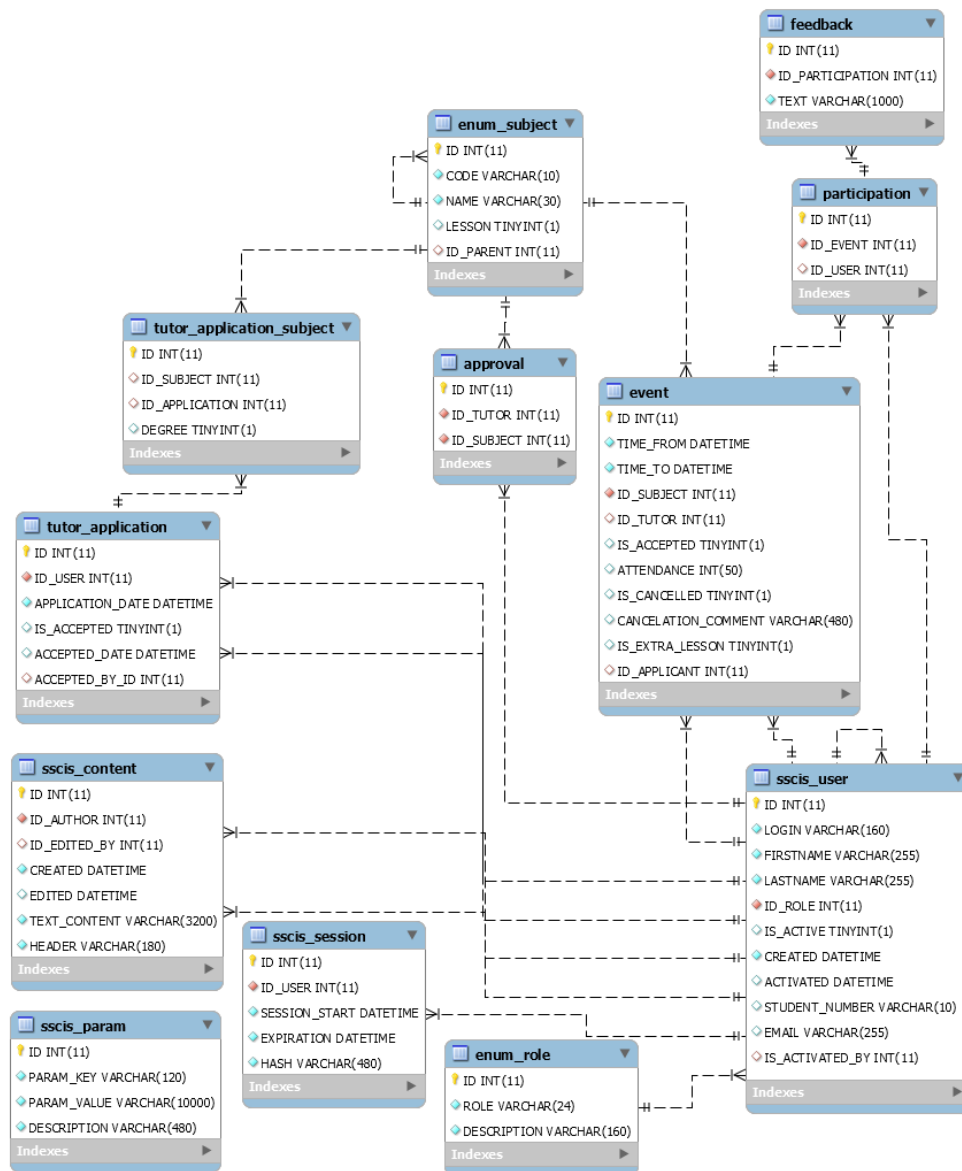
Databázový model bylo potřeba obměnit. S implementací chybějících částí systému došlo k výrazným požadavkům na změnu struktury databáze a na změny v tabulkách, kdy bylo potřeba přidat či odebrat některé sloupce. Při těchto změnách došlo také ke změně relací mezi tabulkami. Konečný model databáze je vyobrazen na obrázku 6.1. Kvůli migraci do řešení MySQL bylo potřeba celou databázi znova vytvořit přes MySQL Workbench, který umožňuje grafickou tvorbu tabulek a propojování přes grafické rozhraní aplikace. Databázi si lze v programu uložit lokálně a následně vygenerovat SQL scripty pro tvorbu databáze. Využil jsem možnosti připojit MySQL Workbench na vzdálenou databázi, která běží na školním serveru. Tato možnost softwaru mi umožnila vytvořit potřebnou databázi přímo na serveru, kde bude databáze nasazena do produkčního prostředí [21].

Aplikace počítá se správnou strukturou databáze, na kterou je připojena. Je zde také nutnost aby tabulka **sscis\_param** obsahovala základní systémové záznamy, které jsou potřeba pro spuštění aplikace a její následný chod. Povinné záznamy jsou vyjmenovány v kapitole 6.3. K bakalářské práci budou v přílohách přiloženy všechny potřebné zakládací scripty pro databázi, které budou obsahovat předem domluvené defaultní hodnoty parametrů nutné pro start aplikace.

V této kapitole jsou veškeré tabulky detailně popsány a jsou k nim uvedeny i vazby na ostatní. Každá tabulka v databázi obsahuje sloupec *ID*, které je nastaveno na *Auto Increment* a udává index záznamu v tabulce.

## 6.1 Sscis\_user

Tabulka uchovávající data o uživateli, kteří se úspěšně přihlásí přes Orion login na proxy serveru. Údaje o uživateli jsou využívány především k autorizaci, zda má přihlášený uživatel přístup do částí systému, kam se snaží dostat. Následně jsou data o uživateli používána také při vytváření žádosti o tutorství či při žádostech o extra lekci. Tabulka se váže relací 1:N k tabulkám **participation**, **event**, **enum\_role**, **sscis\_session**, **sscis\_content**, **tutor\_application** a **approval**. Tato tabulka je také vázaná rekurzivní relací sama k sobě, kdy je v tabulce tato vazba použita pro cizí klíč *IS\_ACTIVA-*



Obrázek 6.1: ERA diagram konečné verze databáze

*TED\_BY*, kam se zapisují *ID* uživatele, který daného uživatele přijal jako tutora.

- **LOGIN**: Login uživatele
- **FIRSTNAME**: Křestní jméno uživatele.
- **LASTNAME**: Příjmení zaregistrovaného uživatele.
- **IS\_ACTIVE**: Hodnota nabývající čísel 0 a 1 reprezentující logické hodnoty *false* a *true*.



- **CREATED**: Datum, kdy byl uživatel zaregistrován do databáze.
- **ACTIVATED**: Datum, kdy byl uživatel aktivován administrátorem.
- **STUDENT\_NUMBER**: Identifikační unikátní studentské číslo, odpovídající studentskému číslu používanému na Západočeské univerzitě v Plzni.
- **EMAIL**: Studentský email uživatele sloužící k zaslání notifikací.
- **ID\_ROLE**: Cizí klíč tabulky **enum\_role**, který přiřazuje uživateli jeho roli.
- **IS\_ACTIVATED\_BY**: Jedná se o rekurzivní relaci, která představuje spojení uživatele a správce, který schválil jeho přihlášku k tutorství. Daný správce je také uživatelem v systému a tudíž je entita **sscis\_user** v relaci **ACTIVATED\_BY** použita dvakrát [26].

## 6.2 Sscis\_session

Tabulka uchovávající data o přihlášených uživateli a jejich autorizace pomocí dat uložených v session, kdy dochází ke kontrole uloženého hash klíče v databázi s klíčem, který má uživatel uložený u sebe v session prohlížeče. Tabulka se váže vazbou N:1 k tabulce **sscis\_user**.

- **SESSION\_START**: Datetime vyjadřující vytvoření session hash klíče pro uživatele (tedy čas jeho přihlášení do systému Student Support Centra).
- **EXPIRATION**: Datetime určující, kdy vyprší platnost session hash klíče pro daného uživatele.
- **HASH**: Vygenerovaný hash klíč pro identifikaci session ve webovém prohlížeči.
- **ID\_USER**: Cizí klíč tabulky **sscis\_user**, který spojuje záznam session s určitým přihlášeným uživatelem.

## 6.3 Sscis\_param

Parametry využívané aplikací pro bezproblémový chod. Některé z těchto parametrů jsou kritické pro spuštění aplikace v konzistentním stavu. S aplikací jsou dodány scripty pro vytvoření databáze, které obsahují zároveň i script

pro naplnění tabulky **sscis\_param** potřebnými daty pro chod aplikace. Tabulka nemá žádné vazby na ostatní tabulky a stojí v datovém modelu databáze osamoceně.

- **PARAM\_KEY**: Jednoznačně identifikující klíč daného parametru, který je unikátní a je napsán velkými písmeny.
- **PARAM\_VALUE**: Hodnota daného klíče, která může být reprezentována textem či číslem, které se následně v aplikaci na back-endu převede do číselné podoby.
- **DESCRIPTION**: Popis daného parametru pro snadnější identifikaci a orientaci mezi parametry.

## 6.4 Sscis\_content

Tabulka uchovávající veškerá data k aktualitám, které se zobrazují na webu. Každý jeden záznam v této tabulce odpovídá právě jedné aktualitě zobrazené na webu. Editací těchto aktualit přes webové rozhraní dochází k úpravě záznamů v databázi. Tabulka má vazbu N:1 s tabulkou **ssci\_user**.

- **ID\_AUTHOR**: Cizí klíč tabulky **sscis\_user** jednoznačně určující uživatele, který danou novinku vytvořil přes webové rozhraní.
- **ID\_EDITED\_BY**: Pole defaultně nastavené na hodnotu *NULL*. Je využíváno v případě, že byl daný záznam upraven přes webové rozhraní. V tomto případě obsahuje cizí klíč tabulky **sscis\_user**. Slouží tedy ke spojení daného záznamu s uživatelem, který ho upravil.
- **CREATED**: Datetime uchovávající časovou značku, kdy byla novinka vytvořena.
- **EDITED**: Datetime defaultně nastavený na hodnotu *NULL*. K vyplnění sloupce dojde v případě úpravy dané novinky a to aktuální časovou značkou.
- **TEXT\_CONTENT**: Textový záznam celého těla novinky.
- **HEADER**: Textové pole sloužící pro uchování nadpisu novinky.

## 6.5 Event

Záznamy v této tabulce reprezentují lekce zadané do systému. Každý záznam odpovídá jedné zadané lekci přes webové rozhraní při vytváření vlastní rozvrhové lekce. Pro potřeby informačního systému jsou v tabulce nastaveny některé fieldy na defaultní hodnoty. Tyto hodnoty se nastaví při založení záznamu, který vznikne vyplněním dotazníku na stránkách SSC. Tabulka má vazbu N:1 s tabulkami **sscis\_user** a **enum\_subject**, dále vazbu 1:N s tabulkou **participation**.

- **TIME\_FROM**: Datetime určující od kdy je daná hodina vyučována.
- **TIME\_TO**: Datetime vypovídající do kdy je hodina vyučována. Je vypočítáván v controlleru pomocí parametru, který je zadán v tabulce **sscis\_param**.
- **IS\_ACCEPTED**: Nabývá hodnot 0 a 1, které reprezentují logické hodnoty *false* a *true*.
- **ATTENDANCE**: Celočíselné pole, které uchovává hodnoty reprezentující počet studentů, kteří navštívili danou hodinu. Je zadáván tutorem dané lekce po ukončení. Defaultní hodnota je NULL.
- **IS\_CANCELLED**: Hodnota nabývající čísel 0 a 1 reprezentující logické hodnoty *false* a *true*. Určuje zda je daná hodina zrušena. Defaultní hodnota je 0.
- **CANCELLATION\_COMMENT**: Pole používané pro zadávání informací proč byla daná lekce zrušena.
- **EXTRA\_COMMENT**: Pole používané pro popis látky, kterou chce student probírat na extra vyučující hodině.
- **IS\_EXTRA\_LESSON**: Hodnota nabývající čísel 0 a 1 reprezentující logické hodnoty *false* a *true*. Určuje zda je daná hodina extra lekcí vytvořenou na žádost či nikoli a jedná se o lekci, která čeká na schválení nebo je schválená do pravidelného rozvrhu.
- **ID\_APPLICANT**: Cizí klíč tabulky **sscis\_user**, který přiřazuje uživatele k hodině pokud se jedná o extra lekci. Defaultní hodnota je NULL.
- **ID\_TUTOR**: Cizí klíč tabulky **sscis\_user**, který slouží pro propojení dané lekce s vyučujícím tutorem.

## 6.6 Feedback

Nově vytvořená tabulka pro sběr zpětné vazby od studentů, kteří navštívili danou lekci. Ukládají se zde záznamy každé vyplněné zpětné vazby, která je zadána přes webové rozhraní. Z dat v této tabulce se následně vychází při tvorbě statistik o spokojenosti navštěvujících studentů a k případnému hodnocení tutorů. Tabulka obsahuje vazbu N:1 s tabulkou **participation**.

- **ID\_PARTICIPATION**: Cizí klíč tabulky **participation**, který slouží pro propojení zadané zpětné vazby s odpovídající lekcí v systému SSC.
- **TEXT**: Textové pole pro ukládání hodnocení či připomínek k lekci.

## 6.7 Tutor\_application

Tabulka obsahující jednotlivé přihlášky uživatelů systému, kteří se chtějí stát tutory a vyučovat v SSC. Záznam je v této tabulce vytvořen při úspěšném odeslání přihlášky přes webové rozhraní. Tabulka je propojena vazbou 1:N s tabulkou **tutor\_application\_subject** a 1:1 s tabulkou **sscis\_user**.

- **ID\_USER**: Cizí klíč tabulky **sscis\_user**, který propojuje danou přihlášku s registrovaným uživatelem.
- **APPLICATION\_DATE**: Datetime obsahující timestamp, kdy byla přihláška do systému odeslána.
- **IS\_ACCEPTED**: Pole pro uchování informace, zda byla daná přihláška již zpracována. Defaultní hodnota je nastavena na *NULL*, při přijetí přihlášky je nastavena na *1*, při zamítnutí na hodnotu *0*.
- **ACCEPTED\_DATE**: Datetime s defaultní hodnotou *NULL*, který je naplněn časovou značkou při interakci s přihláškou ve webovém rozhraní.
- **ACCEPTED\_BY\_ID**: Cizí klíč tabulky **sscis\_user**, který je defaultně nastaven na hodnotu *NULL*. Propojení s daným uživatelem je provedeno v případě přijetí přihlášky.

## 6.8 Tutor\_application\_subject

Jelikož je v systému umožněno na jednu přihlášku navázat více různých předmětů, viz kapitola 4.10.4, které by chtěl uživatel vyučovat, bylo potřeba pro

tato data o položkách na přihlášce vytvořit zvláštní tabulku a přes cizí klíč je na ni navázat. Tabulka tedy obsahuje vazbu 1:N s tabulkou **enum\_subject** a **tutor\_application**.

- **ID\_SUBJECT**: Jedná se o cizí klíč tabulky **enum\_subject**, který propojuje danou známku s předmětem, ze kterého ji uživatel podávající přihlášku obdržel.
- **ID\_APPLICATION**: Cizí klíč tabulky **tutor\_application**, který propojuje daný záznam s přihláškou, ke které se vztahuje.
- **DEGREE**: Celočíselné pole pro uchovávání známek z předmětu v rozsahu 1 až 4.

## 6.9 Pomocné tabulky

Při tvorbě databáze bylo potřeba uchovávat data, kdy pro jeden záznam v první tabulce je potřeba uchovávat odkaz na více záznamů v tabulce druhé. Takovéto vazby se označují jako vazby M:N a jejich realizování je možno pomocí spojovacích tabulek. Tyto pomocné tabulky mají speciální mezilehlou relaci 1:N s každou použitou tabulkou [2].

### 6.9.1 Approval

Rozkladová tabulka pro vazbu M:N mezi **sscis\_user** a **enum\_subject**. Obsahuje cizí klíče jako reference na záznamy v těchto tabulkách. Jedná se o **ID\_TUTOR** odpovídající primárnímu klíči tabulky **sscis\_user** a **ID\_SUBJECT**, které reflektuje odkaz na záznam v **enum\_subject**.

### 6.9.2 Participation

Tabulka vytvořená pro potřeby zadávání zpětné vazby k lekcím či tutorům. Rozklad vazby M:N mezi tabulkami **event** a **feedback**. Vazbu je nutné provést přes pomocnou tabulku, jelikož jeden přihlášený uživatel může zadat feedback k vícero lekcím a zároveň se očekává, že jedna lekce bude mít více než jeden záznam o zpětné vazbě. Obsahuje jen dvě položky (**ID\_EVENT** a **ID\_USER**), které jsou definovány jako cizí klíče výše zmíněných tabulek.

## 6.10 Číselníky

Číselníky jsou v současnosti přítomné snad v každém informačním systému, který je implementován. Jedná se o tabulky, které na základě přesně určeného identifikátoru identifikují určité skutečnosti (objekty, řídicí či popisná data atd.). Data si tedy lze v této tabulce představit jako klíč-hodnota, kdy za klíč je považován PK<sup>1</sup> záznamu a za hodnotu zbylé sloupce [4].

### 6.10.1 Enum\_role

Databázová tabulka sloužící k definici rolí v systému. Role jsou dále v systému používány jako cizí klíče v tabulce **sscis\_users**. Tabulka obsahuje pole **ROLE**, které je unikátní a definuje danou roli. Dále obsahuje pole **DESCRIPTION** využívané k popisu role.

### 6.10.2 Enum\_subject

Tabulka sloužící definici všech možných předmětů, které je možné v SSC vyučovat jako tutor. Nad touto tabulkou je postavena celá logika ohledně předmětů v informačním systému, kde se nacházejí dva druhy předmětů:

- nadřazený předmět - určuje předmět, který je možné v rámci SSC vyučovat. V defaultním nastavení se jedná o předměty *MAT*<sup>2</sup> a *INF*<sup>3</sup>. Tyto předměty lze vybrat ve formuláři při vypisování lekce, pokud má tutor v tabulce **approvals** alespoň jeden z podmiňujících předmětů. Aby byl aplikací předmět rozpoznán jako nadřazený, je potřeba mu v databázi nastavit následující parametry: **LESSON** = 0 a **ID\_PARENT** = *NULL*.
- podmiňující předměty - jedná se o předměty, které má student povoleny zadat při vyplňování přihlášky. Jsou nutné pro tabulku **approvals**, která uchovává informace jaké podmiňující předměty má daný tutor schválené a tudíž má povoleno vypisovat hodiny z k nim nadřazeného předmětu. V základních scriptech databáze jsou obsaženy základní předměty jako M1<sup>4</sup>, M2, PPA1<sup>5</sup> a PPA2. Pro aplikaci je nutné takovýto předmět nastavit následovně: **LESSON** = 1 a **ID\_PARENT** = *ID\_příslušného\_nadřazeného\_předmětu*.

---

<sup>1</sup>Primární klíč

<sup>2</sup>MAT - matematika

<sup>3</sup>INF - informatika

<sup>4</sup>M1 - matematika 1

<sup>5</sup>PPA1 - počítače a programování 1

Do tabulky je možné další záznamy přidávat přes webové rozhraní aplikace a před spuštěním ostrého provozu je tedy potřeba tabulku naplnit nutnými daty pro provoz.

- **CODE:** Sloupec představuje zkratku předmětu, která je zobrazována ve formulářích.
- **NAME:** Celé jméno předmětu pro snadnější orientaci v administraci.
- **LESSON:** Hodnota *1* přiřazuje záznamu význam podmiňujícího předmětu (jedná se o vyučovaný předmět na Fakultě aplikovaných věd). Když je toto pole vyplněno *0*, systém bere záznam o předmětu jako nadřazený předmět.
- **ID\_PARENT:** Rekurzivní relace v rámci tabulky, kde je defaultní hodnota nastavena na *NULL*. Pole je vyplněno v závislosti na **LESSON**, kdy v případě hodnoty *1* v tomto je hodnota tohoto pole nastavena přes webové rozhraní jako reference na nadřazený předmět.

# 7 Implementované případy užití

V rámci bakalářské práce jsem se zabýval implementací chybějících částí systému, které byly potřebné k jeho bezproblémovému využívání pro potřeby Student Support Centre. Veškeré implementované případy užití budou popsány v podkapitolách níže.

## 7.1 Žádosti o extra lekci

Tento implementovaný případ užití rozšiřuje systém o možnost zažádat si o vlastní „soukromou“ lekci, pokud daný student potřebuje urgentně probrat danou látku. Při návrhu řešení bylo potřeba upravit model stávající databáze, která na tuto možnost nebyla připravena. Došlo k zásahu do tabulky **event**, kdy bylo potřeba definovat sloupce pro uchovávání následujících informací:

- zda se jedná o extra lekci
- komentář žadatele, jakou látku potřebuje na dané lekci probrat
- cizí klíč k záznamu uživatele, který si o danou lekci zažádal

Bližší informace, jak jsou tyto informace uloženy v databázi, jsou k nalezení v kapitole 6.5. Grafické rozhraní žádosti o extra lekci bylo implementováno do stránky *Potřebuji pomoc*, kde se nachází ve formě dotazníku pod pravidelným rozvrhem hodin, viz obrázek 7.1. Dotazník je dostupný po úspěšném přihlášení přes Orion účet, a tedy pro běžného návštěvníka portálu není dostupný. Vyplývá to z požadavku, aby byly veškeré žádosti o extra lekce spárovány s konkrétním studentem, který využívá služeb SSC. Rozšiřující funkcionalitou tohoto případu užití bylo potřeba umožnit administrátorům kdykoliv možnost vypnutí extra lekcí vypnout či zapnout. Ovládání tohoto případu užití bylo implementováno pomocí nastavitelného parametru v databázové tabulce **sscis\_param**, ve které se nachází klíč *EXTRA\_LESSONS* nastavitelný na hodnoty *true* nebo *false*. Při vypnutí systému žádostí o extra lekce dojde na backendu k zakázání vykreslení příslušného dotazníku ve webovém rozhraní a nemožnosti odeslat žádost o extra lekce. Pokud, je v momentu vypnutí této funkcionality vypsána nějaká extra lekce, nedojde



k jejímu vymazání ze systému a je stále viditelná v rozvrhu nadcházejících akcí. Podobná logika se vztahuje i na extra lekce, které ještě nejsou přijaté žádným tutorem, ale v době vypnutí funkcionality jsou již zaneseny do systému. Jejich zrušení nebo vymazání musí provést administrátor explicitně.

Pomoc/konzultace je poskytována bezplatně, účast je interně evidována pro potřeby support centra, evidence však není poskytována vyučujícím předmětů a nemá žádný vliv na hoc

Pokud se Vám žádná z vypsanych pravidelných lekcí nehodí, je zde možnost domluvit si extra hodinu.

Vypsání lekce

|          |   |
|----------|---|
| Datum    | 27.03.2020  |
| Čas od   | 10:00   |
| Předmět  | INF   |
| Komentář | Lambda výrazy v <a href="#">LINQ</a> , <a href="#">iQuery</a> |

Zažádat o extra lekci

Obrázek 7.1: Grafická interpretace žádosti o extra lekci

## 7.2 Potvrzování lekcí

Funkcionalita zobrazení zadaných lekcí tutorů jako grafický rozvrh vzešla z podnětu správce informačního systému. Implementovaný způsob potvrzování lekcí, který již byl v aplikaci přítomný, se správci zdál při potvrzování velkého množství lekcí v jednom měsíci příliš složitý. Došlo k upřesnění definice tohoto případu užití, kdy bylo řečeno, že tutoři zadávají své možné vyučovací hodiny na celý měsíc dopředu v určitém časovém intervalu. Správce po tomto časovém úseku zadané hodiny potvrzuje. Při potvrzování se snaží kontrolovat celkový počet již potvrzených hodin každého z tutorů, aby se pokud možno hodiny rovnoměrně rozprostřely mezi všechny tutorů SSC. Zároveň byl vznesen požadavek, zda by bylo možné tento způsob hromadného či jednotkového potvrzování implementovat pomocí grafického rozhraní ve webovém prohlížeči.

### 7.2.1 Grafická interpretace

Návrh grafické interpretace vycházel z předlohy excelovského zobrazení, které nyní používají správci k administraci lekcí. Zobrazení ve webovém prohlížeči bylo navrženo tak, aby splňovalo veškeré definované požadavky. Vypsání lekce jsou zobrazovány v přehledné tabulce, kdy řádky reprezentují tutorů

a sloupce zobrazují časové úseky, na které byla daná hodina vypsána. Tabulka je generována dynamicky a reflektuje aktuální data v databázi. Členění tabulky bylo zvoleno pomocí HTML tagů `<thead>` a `<tbody>`, které následně umožňuje snadnější přístup k elementům při používání jQuery [22]. Nad rozvrhem vypsáných hodin je zobrazen select box, ve kterém je možno přepínat mezi jednotlivými měsíci v roce a zobrazovat pro ně příslušný rozvrh. Vybraný měsíc se na controller backendu přenáší pomocí parametru URL. Jako defaultní měsíc je při každé první návštěvě stránky vybrán aktuální kalendářní. Následná grafická implementace je zobrazena na obrázku 7.2.

Pro potřeby zbylých požadavků, t.j. zobrazování počtu přiřazených hodin k tutorům, počet již akceptovaných hodin ve stejnou dobu a možnost hromadně vybírat a potvrzovat více lekcí, je potřeba využít programovacího jazyka JavaScript. Pro snadnější práci s tímto jazykem jsem využil knihovnu jQuery, která je v současnosti nejrozšířenější knihovnou pro interakci mezi JavaScriptem a HTML. Tato knihovna je dostupná s MIT licencí, která umožňuje danou knihovnu svobodně používat při zachování kopie licence a jména autora [23]. Knihovna je využita pro navázání event handlerů na HTML elementy a jejich stylizaci při interakci. Jednotlivá pole tabulky odpovídají datům v databázi a jsou interpretovány podle následujících pravidel:

- červené pole - tutor nemá v odpovídající termín vypsanou lekci
- zelené pole - akceptovaná lekce tutora v daný termín
- oranžové pole - tutor má vypsanou lekci na daný termín, kterou je možno vybrat kliknutím
- oranžové šedě šrafované pole - vypsaná lekce tutora, která je vybrána k hromadnému potvrzení

Těmito pravidly se řídí backend při generování výsledného HTML. V tabulce je zároveň na buňky, které odpovídají vypsané lekci tutora, která nebyla doposud akceptována, přidán atribut *onclick*. Na tento atribut je následně přidána funkce `toggleSelected()`, která obsluhuje změnu stylu dané buňky při stisku a zároveň zařazení ID lekce do proměnné, která se odesílá na backend při hromadném potvrzování. Funkce je multifunkční a používá se i k následnému odstranění dané lekce z výběru. Pro přepočítání hodin u tutorů a daného časového úseku slouží funkce `calculateRow()` a `calculateColumn()`. Tyto dvě funkce jsou spuštěny pokaždé, když se provede interakce s tabulkou a jejich spuštěním dojde k přepočtu příslušných dat odpovídajících změně. Takto vybrané hodnoty lze následně hromadně potvrdit odesláním formuláře na backend pomocí tlačítka SEND.



Obrázek 7.2: Grafická interpretace potvrzování lekcí

### 7.2.2 Stažení rozvrhu ve formátu CSV

Pro potřeby administrace SSC a následné vykazování je potřeba data ze systému stáhnout. K tomu slouží tlačítko **DOWNLOAD CSV**. Po interakci s tlačítkem dojde k zavolání metody `controller.csvTimetableDownload()`. Tato metoda obsluhuje vytvoření požadovaného souboru a jeho vrácení pomocí metody `File`. Pro zápis dat do souboru byl použit návratový typ `FileContentResult`, který potřebuje tři parametry: pole bitů s daty, MIME typ souboru a jméno výsledného souboru [27]. Tento soubor je poté automaticky stáhnut přes webové rozhraní a má následující strukturu, viz obrázek 7.3.

Generování textových dat pro soubor obsluhuje implementovaná metoda `generateCsv()`, do které jako parametry vstupují seznamy hodin a tutorů. Pro snadné importování `.csv` souboru do tabulkového procesoru Excel jsou jednotlivé sloupce odděleny předem definovaným oddělovačem „;“.

```
Tutor;Hodiny
;13.02.2020 10:00:00;28.02.2020 10:00:00;28.02.2020 12:00:00;Součet
██████████;-;-;-;0
██████████;-;-;-;0
Jakub Hlaváč;-;X;-;1
```

Obrázek 7.3: Stáhnutý rozvrh v `.csv` formátu

## 7.3 Sběr statistik k lekci

Důležitou částí informačního systému je zajisté i systém pro sběr zpětné vazby k vyučovaným hodinám a shromažďování jejich statistik návštěvnosti. Ze specifikací požadavků na sběr těchto informací byl následně vytvořen provázaný systém, jak je efektivně a snadno od studentů zaznamenávat. Jelikož systém nefunguje na bázi přihlašování studentů na dané vypsání lekce, nelze generovat data o návštěvnosti automaticky a je nutná interakce ze strany tutora. Z tohoto důvodu byl tento požadavek rozdělen na dva podřazené. Následně byly postupně implementovány.

### 7.3.1 Sběr zpětné vazby

Sběr zpětné vazby k lekcím je záležitost studentů, kteří danou lekci navštíví. Zde je potřeba jejich interakce se systémem a celý tento případ užití je postaven na dobrovolnosti zadávání těchto podnětů. Vyplnit dotazník je možné na speciální URL adrese, která je vygenerována k příslušné lekci. Eliminování zneužívání tohoto formuláře je tedy dvouúrovňové, kdy je nejprve potřeba znát unikátní adresu a následně být v systému přihlášen. Unikátní adresu generuje vždy tutor dané lekce či správce a je možné ji vygenerovat i zpětně po odučené lekci a následně dotazník studentům, kteří lekci navštívili, rozeslat emailem.

Formulář pro zaslání příslušné zpětné vazby je naprosto jednoduchý a obsahuje jen jedno vyplnitelné pole - text zpětné vazby, viz obrázek 7.4. Předpokládá se, že student při vyplňování dotazníku ví, ke které lekci se vztahuje. Spárování s danou lekcí následně proběhne přes parsing URL adresy na backendu aplikace. Metoda pro rozklíčování dat z parametru **ResolveEventID** se nachází ve třídě `FeedbackUrlGenerator`. Tato metoda zjistí, zda je zadaný parametr validní a následně provede rozparsování tohoto parametru, ze kterého zjistí odpovídající ID lekce, ke které se dotazník vztahuje. Toto ID je pak vráceno jako návratová hodnota metody, pokud se nepovede parametr rozparsovat, je vrácena hodnota `null` [14], viz 7.1. Vrácení hodnoty `null` vede k zobrazení chybové hlášky ve webovém rozhraní, v opačném případě se vytvoří objekt `Feedback`, který je odeslán do view.

```
if (int.TryParse(code.Substring(6), out int result))
{
    return result;
}
return null;
```

Kód 7.1: Parsování URL parametru

Feedback

---

Text

[Zpět](#)

Obrázek 7.4: Vyplněný formulář pro odeslání feedbacku

### Vygenerování dotazníku pro sběr podnětů

Generování unikátní URL adresy tohoto dotazníku provádí tutor dané lekce či správce systému. Celý tento proces se provádí na detailu lekce stiskem odkazu *Generuj odkaz pro feedback*.

Vygenerování příslušné unikátní URL adresy je provedeno ve třídě `FeedbackUrlGenerator`. V této třídě je implementována metoda na vygenerování unikátního identifikátoru příslušné hodiny `generateURL`. Vygenerovaný parametr má následující formát:

```
Pro vyučovací hodinu 10.12.2020 s databázovým ID 256:  
?code=201210256
```

Jelikož většina studentů, kteří využívají služeb SSC, zvládá mobilní technologii, je při každém generování URL adresy vygenerován také unikátní QR code. Tento QR code v sobě obsahuje data potřebná k otevření unikátní URL adresy v prohlížeči. Poté ho stačí naskenovat mobilním zařízením, které si z něj tato data načte a v defaultním prohlížeči otevře příslušnou stránku s formulářem.

Pro snadnější generování jsem použil NuGet package *QRCoder*, který je dostupný pod MIT licencí [5]. Celé generování QR Code je provedeno přes tři třídy v tomto balíčku: `QRCodeGenerator`, `QRCodeData` a `QRCode`, kdy se pomocí generátoru naplní třída `QRCodeData` uchováující data. Tento objekt je dále použit pro vygenerování celého QR kódu přes třídu `QRCode`. Data pro zobrazení kódu na view jsou odeslány jako `Base64String` a následně v HTML tagu `<img>` zobrazeny graficky, viz obrázek 7.5.

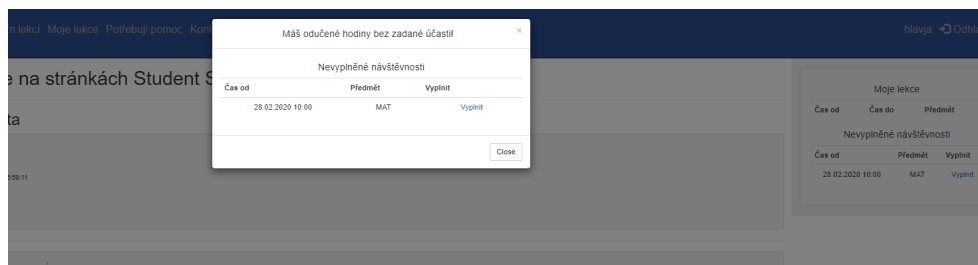
## Generování QR kódu



Obrázek 7.5: Stránka s vygenerovaným QR kódem a unikátní URL adresou feedbacku

### 7.3.2 Sběr dat o návštěvnosti lekce

Statistiky o návštěvnosti lekcí je nutno zadávat ručně. Příslušná data tedy budou manuálně vyplňovat tutoři, kteří danou hodinu vyučovali, pomocí formuláře, který jim bude po skončení lekce zpřístupněn. Pokud nedojde k vyplnění návštěvnosti je každý tutor, který má alespoň jednu hodinu s nevyplněnou návštěvností při každém přihlášení vyzván k jejímu vyplnění pomocí vyskakovacího modálního okna uprostřed obrazovky, viz obrázek 7.6. V tomto modálním okně jsou v přehledném formátu vypsány veškeré lekce, u kterých chybí zadat návštěvnost. U každé z takto vypsáných lekcí je zároveň přidán proklikávatelný text *Vyplnit*, který přesměruje tutora na stránku s formulářem. Podobný formulář je také renderován v pravém informačním panelu stránky.



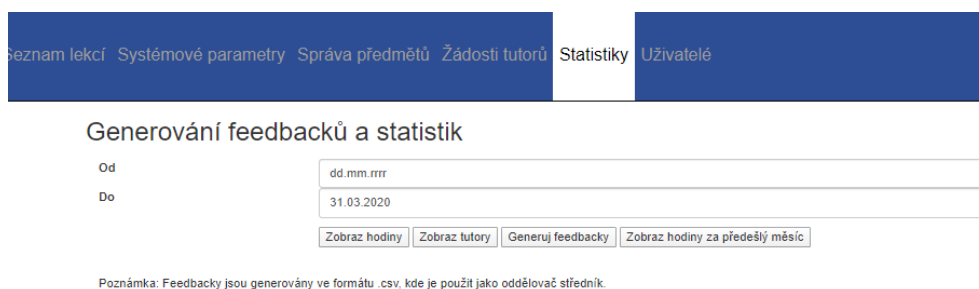
Obrázek 7.6: Vyskakovací okno pro sběr statistik návštěvnosti

## 7.4 Prohlížení statistik

Ze všech dat uchovávaných v systému je následně potřeba vygenerovat statistiky a umožnit administrátorům jejich zobrazení či stažení ve vhodném formátu. K jejich prohlížení slouží záložka *Statistiky* v horním menu, viz obrázek 7.7. Tato stránka obsahuje dvě pole pro zadávání dat od kdy do kdy chceme statistiky generovat. Při nevyplnění těchto polí zde nedojde k chybě, nýbrž k zobrazení všech statistik, které jsou v systému uloženy do aktuálního data. Zároveň je možné vyplnit jen jedno pole formuláře, kdy se systém zachová podle definovaných scénářů:

- *vyplněné pole od* - zobrazí statistiky v časovém intervalu mezi zadaným datem od až do aktuálního data
- *vyplněné pole do* - zobrazí statistiky od počátku jejich shromažďování až do zadaného data

Tato pravidla jsou aplikována na všechny interakce, které lze pod formulářem vykonat a zobrazované statistiky se jimi řídí.



Obrázek 7.7: Webové rozhraní pro zobrazování statistik.

### 7.4.1 Statistika lekcí

Statistika lekcí lze zobrazit tlačítkem *Zobraz hodiny*, které správce přesměruje na přehledný seznam všech akceptovaných hodin v daném časovém intervalu, viz obrázek 7.8. V tomto seznamu jsou zobrazeny veškeré informace o daných hodinách včetně počtu zpětných vazeb, které byly k dané hodině zadány. Zobrazovaná data jsou získávána controllerem `FeedbacksController`, který nejdříve z databáze načte do datové struktury `List<Event>` veškeré hodiny odpovídající časovému rozpětí a následně naplní příslušný model `Statistics` zbylými potřebnými informacemi. Tento model je controllerem odeslán na view `List`, kde proběhne sestavení příslušného HTML

pro zobrazení ve webovém prohlížeči. Pro zajištění bezpečnosti uchovávaných dat je možné metodu pro zobrazení statistik `List (MetaInterval model)` volat jen s platnými údaji v session na úrovni **ADMIN**. Ověření dat v session je prováděno pomocí třídy `SSCISAuthorize`, která obsahuje metody nutné pro autentizaci, které se vykonávají před zavoláním metody controlleru.

| Statistiky      |                     |                     |                 |                    |
|-----------------|---------------------|---------------------|-----------------|--------------------|
| Počet hodin: 23 | MATH: 19            | INF: 4              | MECH: 0         | Odučené hodiny: 46 |
|                 | MATH: 38            | INF: 8              | MECH: 0         |                    |
| Hodina          | Čas od              | Čas do              | Počet feedbacků | Tutor              |
| Matematika      | 20.11.2019 11:00:00 | 20.11.2019 13:00:00 | 0               | admin              |
| Matematika      | 26.11.2019 10:00:00 | 26.11.2019 12:00:00 | 0               | lvasa              |
| informatika     | 02.12.2019 9:00:00  | 02.12.2019 11:00:00 | 0               |                    |
| Matematika      | 03.12.2019 10:00:00 | 03.12.2019 12:00:00 | 0               | lvasa              |
| Matematika      | 05.12.2019 10:00:00 | 05.12.2019 12:00:00 | 0               |                    |
| Matematika      | 05.12.2019 10:00:00 | 05.12.2019 11:00:00 | 0               |                    |
| Matematika      | 05.12.2019 15:00:00 | 05.12.2019 17:00:00 | 0               |                    |
| informatika     | 09.12.2019 9:00:00  | 09.12.2019 11:00:00 | 0               |                    |
| Matematika      | 10.12.2019 10:00:00 | 10.12.2019 12:00:00 | 0               | lvasa              |
| Matematika      | 12.12.2019 10:00:00 | 12.12.2019 12:00:00 | 0               |                    |
| Matematika      | 12.12.2019 15:00:00 | 12.12.2019 17:00:00 | 0               |                    |
| informatika     | 16.12.2019 9:00:00  | 16.12.2019 11:00:00 | 0               |                    |
| Matematika      | 17.12.2019 10:00:00 | 17.12.2019 12:00:00 | 0               | lvasa              |
| Matematika      | 19.12.2019 10:00:00 | 19.12.2019 12:00:00 | 0               |                    |
| Matematika      | 19.12.2019 15:00:00 | 19.12.2019 17:00:00 | 0               |                    |
| informatika     | 31.12.2019 11:00:00 | 31.12.2019 12:00:00 | 0               | admin              |
| Matematika      | 22.01.2020 10:10:00 | 22.01.2020 12:10:00 | 0               | admin              |
| Matematika      | 29.01.2020 10:10:00 | 29.01.2020 12:10:00 | 0               | admin              |
| Matematika      | 05.02.2020 10:10:00 | 05.02.2020 12:10:00 | 0               | admin              |
| Matematika      | 13.02.2020 10:00:00 | 13.02.2020 12:00:00 | 0               |                    |
| Matematika      | 28.02.2020 10:00:00 | 28.02.2020 12:00:00 | 0               | hlavja             |
| Matematika      | 28.02.2020 12:00:00 | 28.02.2020 14:00:00 | 0               | hlavja             |
| Matematika      | 30.03.2020 10:10:00 | 30.03.2020 12:10:00 | 1               | hlavja             |

Obrázek 7.8: Webové rozhraní statistik lekcí

## 7.4.2 Statistika tutorů

Statistiky tutorů lze zobrazit dvěma způsoby, aby šlo snadněji vygenerovat nejdůležitější data pro výpočet mzdy tutorů. Automatické zobrazení statistik lze provést stisknutím tlačítka *Zobraz hodiny za předešlý měsíc*. Pro zobrazení statistik v libovolném časovém intervalu je zde tlačítko *Zobraz tutorů*, pro které jsou statistiky vygenerovány na základě vyplněných dat ve formuláři. Obě dvě interakce zobrazují stejně členěný seznam tutorů a jejich odučených hodin viz obrázek 7.9. V příslušné metodě na controlleru jsou do modelu `StatisticsTutor` uložena veškerá potřebná data pro zobrazení ve webovém prohlížeči. Následně je naplněný model poslán na view `TutorList`, kde dochází ke zpracování uložených dat a jejich grafickému zobrazení. Jako v předchozím případě je i volání této metody podmíněno autorizací na úrovni **ADMIN**.



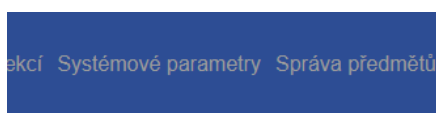
| Seznam lekcí   Systémové parametry   Správa předmětů   Žádosti tutorů   <b>Statistiky</b>   Uživatelé |                         |     |     |      |                     |  |
|---|-------------------------|-----|-----|------|---------------------|--|
| Statistiky  |                         |     |     |      |                     |  |
| Tutor   | Počet vyučovacích hodin | MAT | INF | MECH | Počet čistých hodin |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| hlava   | 3                       | 3   | 0   | 0    | 6                   |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| ██████████  | 5                       | 5   | 0   | 0    | 10                  |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |
| ██████████  | 0                       | 0   | 0   | 0    | 0                   |  |

Obrázek 7.9: Webové rozhraní statistik tutorů

### 7.4.3 Stažení a zobrazení feedbacků

Záznamy z formulářů zpětné vazby k lekci lze prohlížet dvěma způsoby. Způsob prohlížení zpětných vazeb ve webovém prohlížeči je možný po zobrazení statistik lekcí, které bylo popisováno v kapitole 7.4.1. Na této stránce je u každé lekce zobrazen počet zadaných podnětů k lekci, na který lze kliknout a následně jsou veškeré příslušné podněty zpětné vazby zobrazeny v prohlížeči, viz obrázek 7.10a.

Druhá možnost, jak získat data pro analýzu zpětné vazby, je jejich vygenerování do souboru `.csv` a následné stáhnutí do svého zařízení. Generování dat pro tlačítko *Generuj feedbacky* se řídí pravidly formuláře na zadávání časového intervalu popsány v kapitole 7.4. V tomto případě dojde na backendu aplikace k vygenerování příslušného souboru v kódování UTF-8, kdy jsou použity středníky jako oddělovače jednotlivých polí, viz obrázek 7.10b.



#### Feedbacky k lekci

| Login uživatele | Text            |
|-----------------|-----------------|
| admin           | Nejlepší hodina |

[Zpět](#)

```

1 Datum;Od;Do;Predmet;Text
2 30.3.2020;10:10;12:10;MAT;Nejlepší hodina
3

```

(a) Zobrazení feedbacku k lekci (b) Ukázka formátu `.csv` souboru feedbacků

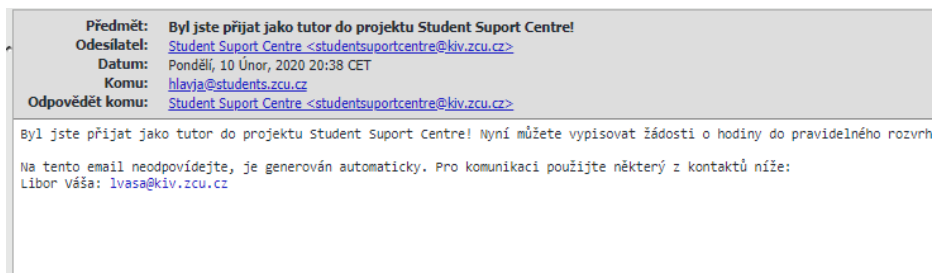
## 7.5 Emailové notifikace

Po implementaci všech případů užití odpadly některé rutinní kroky spojené s přijímáním tutorů, vypisováním lekcí a dalších nutných aktivit. Následně bylo potřeba implementovat systém, který by sloužil k informování tutorů, ohledně akcí, které jsou spojeny s jejich uživatelským jménem. Po analýze možností a jejich uplatnění v praxi mi vyšly jako použitelné jen dva možné způsoby jak důležité informace tutorům předávat.

Jedna z možností byla dané informace zobrazovat ve zvýrazněné formě po přihlášení do systému na hlavní obrazovce či jiných vhodně umístěných místech na webovém rozhraní. Tato možnost byla již částečně využita pro zadávání návštěvnosti lekcí, které je popsáno v kapitole 7.3.2. Dále byla tato možnost informování použita k zobrazování informací o lekcích, které náleží danému přihlášenému uživateli v pravém informačním panelu, který je popsán v kapitole 7.7.3. Tato metoda pro informování uživatelů stále předpokládá jejich aktivní přihlášení do systému na denní bázi.

Druhou metodou byla možnost při důležitých akcích v systému zasílat automatické emaily. Proto byl vyvinut systém pro zasílání takovýchto emailů při definovaných akcích v systému:

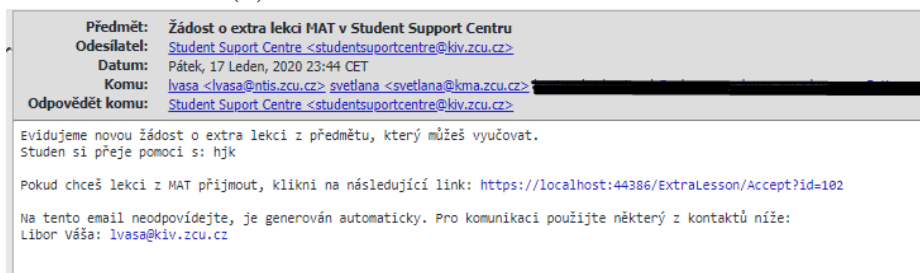
- schválení přihlášky na pozici tutora, viz obrázek 7.11a
- akceptování lekce do pravidelného rozvrhu, viz obrázek 7.11b
- žádost o extra lekci (email se odešle všem tutorům, kteří mají oprávnění vyučovat daný předmět), viz obrázek 7.11c



(a) Informační email o akceptování přihlášky



(b) Informační email o schválení lekce



(c) Informační email o nové žádosti o extra lekci

Pro usnadnění implementace byla využita externí multiplatformní knihovna MailKit[6] v podobě *NuGet* balíčku. Tuto knihovnu jsem následně využil ve třídě `EmailService`, která implementuje metodu pro odesílání emailu pomocí pomoci třídy `SmtpClient` z importované knihovny. Toto odesílání je díky možnostem knihovny vykonáváno asynchronně zavoláním metody `SendAsync`, viz kapitola 7.2.

```
await emailClient.SendAsync(message).ConfigureAwait(false)
    ↪ ;
```

Kód 7.2: Asynchronní odesílání emailu

Po úspěšném odeslání emailu je `SmtpClient` odpojen od SMTP služby, aby nedocházelo k množení konekcí na serveru. Metoda odesílání emailu je volána při každé výše definované akci a jako parametry jsou vkládána potřebná data k naplnění emailu. SMTP server pro odesílání pošty je uložen v databázové tabulce `sscis_param` pod unikátním klíčem `SMTP`. Pokud není tento klíč vyplněn, je následně použita hodnota z defaultního nastavení v souboru

*appsettings.json*. Tato hodnota je nastavena na SMTP server Google.com. Zároveň jsou v tomto souboru uloženy všechny definice textů emailů, které je tudíž jednoduché změnit. Jedná se o záznamy, které jsou uloženy pod klíčem *EmailMessageConfigs*.

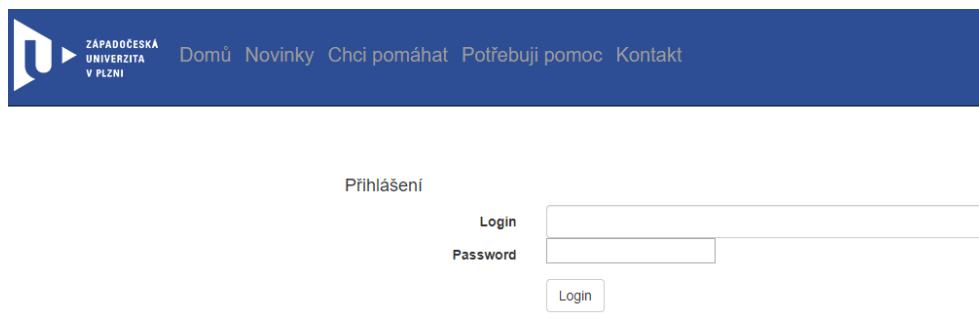
## 7.6 Admin login

Po nasazení systému do produkčního prostředí s novou databází a přihlašováním přes orion login nebyly v databázi žádné záznamy o uživateli. Tyto záznamy se vytvoří až s prvním přihlášením uživatele, ale takovýto uživatel má standardně nastavenou roli na User. Tedy v tuto dobu neexistoval nikdo, kdo by měl po přihlášení přes orion login roli Admin a mohl spravovat systém. Jediná možnost, jak přiřadit prvnímu správci roli Admin bylo přes databázi pomocí SQL scriptu.

Tento nedostatek systému byl vyřešen pomocí vytvoření administrátorského přístupu do aplikace i v době, kdy je spuštěné přihlašování přes orion login. Na stránce AdminLogin je zobrazen formulář pro přihlášení do systému pro uživatele *admin*, pro kterého je nutné zadat heslo, viz obrázek 7.12. Toto je jediné heslo, které je v databázi aplikace uloženo a slouží k autentizaci uživatele. Pokud zadané heslo v databázi neexistuje, je při prvním navštívení přihlašovací stránky vytvořen záznam s předem definovaným defaultním heslem: **VasaAdmin**. Vytvořené heslo je delší než 8 znaků a bylo zvoleno na základě doporučení pro minimální vlastnosti přístupového hesla [24]. Při jeho vytváření dochází k využití knihoven .NET Core System.Security.Cryptography, která obsahuje třídu Rfc2898DeriveBytes. Tato třída obsahuje metody pro zahashování hesla pomocí saltingu [25]. Počet opakování hashování jsem použil pevný a to 1000. Tento vytvořený hash je následně uložen do databáze jako parametr *ADMIN\_PASSWORD* do tabulky **sscis\_param**. Při dalším přihlášení přes formulář na stránce AdminLogin dojde tedy k vytažení hash hodnoty hesla z databáze, která se porovná s nově vytvořeným hashem zadaného hesla. Pokud jsou shodné dojde k přihlášení uživatele, kterému se vytvoří session.

Toto defaultní heslo je možné změnit, stejně jako ostatní parametry aplikace, na stránce SSCISParams. Při úpravě není zobrazeno původní heslo v plain textu, ale je potřeba vytvořit úplně nové heslo, které přepíše již existující záznam v databázi příslušnou hash hodnotou. Tedy při změně hesla z defaultního a následného zapomenutí nově vytvořeného je potřeba přihlášení přes jiný Admin účet a heslo změnit či zásahem do databáze záznam s heslem vymazat a při následném pokusu o přihlášení přes AdminLogin

bude opět vygenerován hash defaultního hesla.



Přihlášení

Login

Password

Login

Obrázek 7.12: Přihlašovací obrazovka pro administrátora

## 7.7 Úprava dříve implementovaných případů užití

V průběhu práce na této bakalářské práci jsem zároveň sbíral podněty od uživatelů, kteří systém testovali ve zkušebním provozu. Následně byly tyto podněty vyhodnoceny s vedoucím práce a užitečné připomínky následně zapracovány do systému. V hojně míře se již jednalo o implementované případy užití z předmětu ZSWI či drobné dodělávky webového rozhraní a jeho možností. V neposlední řadě se jednalo také o podněty samotných správců aplikace, kdy došlo k úpravě následujících případů.

### 7.7.1 Administrace uživatelů

Pro snazší administraci uživatelů byly provedeny úpravy na více místech aplikace.

Prvním podnětem byla žádost jednoho ze správců umožnit vypisovat hodiny uživatelům v systému. Tento požadavek vedl k úpravě již implementovaného formuláře pro vypisování lekcí, jelikož ten je pro všechny uživatele stejný. Zde došlo k úpravám na backendu aplikace, kdy se kontroluje session přihlášeného uživatele. Pokud je controllerem zjištěno přihlášení správce aplikace, je formulář na view naplněn všemi uživateli a možnými předměty, které lze vyučovat. Následně je možné vypsát libovolnou lekci libovolnému uživateli.

Druhým důležitým požadavkem byla možnost administrace jednotlivých uživatelů, kdy by daného uživatele mělo být možno ze systému smazat a mít možnost mu manuálně přiřazovat předměty, které může vyučovat. Implementace této funkcionality proběhla do již existujícího view na editaci uživatelů,

viz obrázek 7.13. Při mazání uživatelů bylo potřeba vyřešit konflikt, kdy má daný uživatel v systému zadanou vyučovací hodinu. Z tohoto důvodu byly předem nadefinovány případy, jak se má v těchto situacích systém zachovat:

- uživatel má vypsanou hodinu, která není schválená administrátorem - hodina se ze systému vymaže společně se záznamem uživatele
- uživatel má zadanou žádost o extra lekci, kterou neakceptoval nikdo z tutorů - hodina se ze systému vymaže
- uživatel má vypsanou a schválenou hodinu do pravidelného rozvrhu - tato hodina se označí jako *Zrušená* a v databázi se k ní přiřadí jako tutor správce, který mazání prováděl
- uživatel má akceptovanou žádost o extra lekci - tato lekce se označí jako *Zrušená* a v databázi je odstraněn žadatel

Mazání zbylých záznamů z databáze se provádí explicitně a u každé tabulky, kde je atribut uživatele použit jako cizí klíč, je podle nutnosti zachování dat rozhodnuto o zachování či smazání záznamu. Například u statistik dochází k nastavení takovýchto odkazů na defaultní hodnoty. Na druhé straně případné záznamy o přihlášce daného uživatele jsou smazány a pokud se uživatel do systému vrátí a bude se chtít stát tutorem, je potřeba přihlášku opětovně odeslat.

The screenshot shows a web interface for user management. At the top, there is a navigation bar with links: 'Seznam lekcí', 'Systémové parametry', 'Správa předmětů', 'Žádosti tutorů', 'Statistiky', and 'Uživatelé'. Below this is a header 'Upravit' (Edit) for a user named 'TUTOR - hlavja - hlavja@students.zcu.cz'. The form contains the following fields:

- Login:** hlavja
- Jméno:** Jakub
- Příjmení:** Hlaváč
- Role:** TUTOR
- Studentské číslo:** (empty)
- E-mail:** hlavja@students.zcu.cz
- MAT:**
- INF:**
- MECH:**

At the bottom of the form is a 'Uložit' (Save) button and a 'Zpět' (Back) link.

Obrázek 7.13: Obrazovka administrace uživatele

### 7.7.2 Statické texty

Při přebírání částečné implementace systému byly veškeré texty vyplněny již na úrovni kódu. Tato implementace se v průběhu testování ukázala jako nevhodná, protože i kvůli malé úpravě v textu je potřeba provést úpravu kódu. Následně je potřeba upravený kód zkompilovat a nasadit novou verzi na server. Tyto nutné kroky bylo možno zredukovat o nasazování aplikace, kdy by se již do běžící aplikace na serveru nahrály zkompilované views. Pokud by se jednalo jen o úpravu statického textu, aplikace by byla nadále plně funkční, ale takovýto přístup není vhodný pro ostrý provoz. Na základě tohoto zjištění byly statické texty přesunuty do databázové tabulky **sscis\_param**, kde má každý statický text svůj unikátní klíč a je následně zobrazován ve webovém prohlížeči. Pokud daný klíč v databázi neexistuje, je zobrazen předem definovaný univerzální text. V databázi je uložen HTML kód daného textu, který je při editaci zobrazován v prostém input fieldu, takže není nijak validován a při úpravách se počítá se znalostmi HTML jazyka a kompetencí správce. Před uložením do databáze se nad zadaným textem provede pomocí `WebUtility.HtmlEncode()` HTML enkódování [7], aby nebylo možné napadnout data v databázi pomocí SQL injection. Při zobrazování textu v editoru či na stránkách je nad hodnotou uloženou v databázi zavoláno dekodování.

### 7.7.3 Pravý informační panel

Tento informační panel byl ve webovém rozhraní částečně implementován již v předmětu ZSWI, kde docházelo jen k zobrazování tutorem vypsanych lekcí. Implementaci jsem následně rozšířil do stávající podoby, která je zachycena na obrázku 7.14. Nyní se menu skládá ze čtyř částí:

- *Moje lekce* - tato část byla již součástí aplikace, ale docházelo k výpisu veškerých lekcí, které měl tutor zadány v systému. Pro přehlednost informačního panelu byly provedeny nutné opravy pro zobrazování jen akceptovaných lekcí daného tutora, které spočívaly v rozšíření podmínek na backendu aplikace.
- *Přijaté lekce* - část informačního panelu, ve které se zobrazují veškeré přijaté extra lekce, které daný uživatel zadal do systému.
- *Žádosti o lekce* - v této části se zobrazují veškeré žádosti o extra lekci, které čekají na přijetím jedním z tutorů.

- *Nevyplněné návštěvnosti* - spodní část panelu souvisí se sběrem návštěvnosti lekcí, viz kapitola 7.3.2, kde se jedná o totožný formát informací.

hlavja [➔ Odhlásit se](#)

utor

lavja

stane od kolegů ve vyšších

| Moje lekce       |                  |         |
|------------------|------------------|---------|
| Čas od           | Čas do           | Předmět |
| 01.04.2020 10:00 | 01.04.2020 12:00 | MAT     |

Přijaté lekce

| Čas od | Čas do | Předmět |
|--------|--------|---------|
|        |        |         |

Žádosti o lekce

| Čas od           | Čas do           | Předmět |
|------------------|------------------|---------|
| 03.04.2020 10:10 | 03.04.2020 11:10 | MECH    |

Nevyplněné návštěvnosti

| Čas od           | Předmět | Vyplnit  |
|------------------|---------|--|
| 28.02.2020 10:00 | MAT     | <a href="#" style="color: #007bff; text-decoration: none;">Vyplnit</a> |
| 31.03.2020 10:10 | MAT     | <a href="#" style="color: #007bff; text-decoration: none;">Vyplnit</a> |
| 30.03.2020 10:10 | MAT     | <a href="#" style="color: #007bff; text-decoration: none;">Vyplnit</a> |

Obrázek 7.14: Pomocné menu na pravé straně



# 8 Nasazení informačního systému

Nasazení informačního systému proběhlo na výpočetní techniku Fakulty aplikovaných věd Katedry informatiky a výpočetní techniky. Bylo zažádáno o poskytnutí MySQL databáze a serveru pro hosting aplikace s minimálním úložným prostorem 50GB. Aplikační server měl mít předinstalovaný operační systém Windows minimální verze 7, Windows Server 2012 nebo novější či Linux Debian verze 9 a novější.

## 8.1 Databázový server

MySQL databáze byla vytvořena na serveru *147.228.63.10* s přístupem přes PhpMyAdmin na URL adrese *https://students.kiv.zcu.cz/phpmyadmin/*. Pomocí MySQL Workbench byly v předvytvořené databázi *lvasa* definovány tabulky, které byly následně naplněny daty důležitými pro start aplikace, viz kapitola 6.

## 8.2 Aplikační server

Nasazení aplikace proběhlo na Windows server *typhoon.zcu.cz* v následující konfiguraci, která je zobrazena v tabulce 8.1.

Aplikační server *typhoon.zcu.cz*

---

|                  |   |
|------------------|---|
| Operační systém: | Windows Server 2016 Standard 64-bit                         |
| CPU:             | Intel(R) Xeon(R) CPU E5-4620 v2<br>@ 2.60GHz (2 processors) |
| RAM:             | 32 GB   |
| Connection:      | CESNET  |

Tabulka 8.1: Specifikace aplikačního serveru

### 8.2.1 Požadovaný software

Pro spuštění .NET Core aplikace na Windows serveru je potřeba doinstalovat následující software:

- ASP.NET Core Runtime 2.2.8 (Hosting Bundle) <sup>1</sup>
- Internet Information Services 10.0 Express

Veškerý software je možné stáhnout z oficiálních online zdrojů a následně nainstalovat pomocí jednoduchého průvodce instalací, který je přítomen u každého z nich.

## 8.2.2 Nasazení aplikace v Internet Information Services

Výsledná aplikace byla po dokončení vývoje nasazena na reálný webový server, na který byl předinstalován požadovaný software. Pro snadnější publikování aplikace bylo využito možností Visual Studio 2019, které umožňuje celý projekt sestavit a publikovat do definovaného adresáře. Takto vytvořený adresář byl následně překopírován přes vzdálený přístup na server. Umístění adresáře na serveru je zapotřebí zvolit v souladu s přístupy uživatele, který bude k provozu služby používán. Podrobný postup, jak aplikaci spustit v IIS, bude popsán v Instalační příručce, viz kapitola A.

V rámci nasazení aplikace do ostrého provozu, bylo vyhověno moderním trendům pro bezpečnou komunikaci mezi webovým rozhraním a aplikací a v průběhu testování došlo k nasazení protokolu HTTPS, který využívá protokol HTTP a SSL. Tento bezpečný protokol zajišťuje šifrování přenášených dat, jejich integritu a také autentizaci s ověřením certifikátu [1]. Pro využití protokolu HTTPS bylo zapotřebí na aplikační server nahrát platný certifikát podepsaný certifikační autoritou, která je celosvětově uznávaná. K tomuto kroku bylo využito možností *Let's Encrypt Authority X3*, pro kterou je k dispozici volně dostupný software *win-acme* umožňující přiřadit certifikát k jednotlivým webovým aplikacím v IIS. Nástroj také nabízí nastavení automatického tasku ve Windows, pro prodloužení doby platnosti certifikátu [15].

---

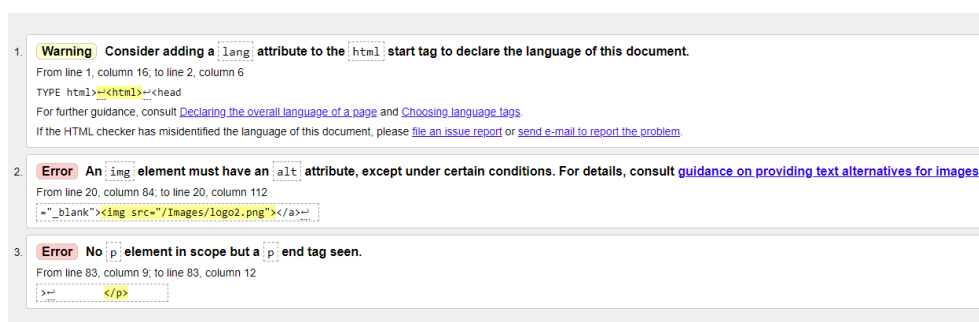
<sup>1</sup>Dostupné z: <https://dotnet.microsoft.com/download/dotnet-core/2.2>

# 9 Testování

Vývoj aplikace probíhal v lokálním prostředí pomocí vývojového prostředí Visual Studio 2019. V rané fázi projektu, při přepisování aplikace do .NET Core, bylo využíváno nástroje XAMPP, přes který lze spustit lokální MySQL databázi a webovým rozhraním phpMyAdmin. V průběhu vývoje byla tato databáze přesunuta na databázový server poskytnutý KIV. Aplikace byla nasazena do lokálního IIS a její deploy probíhal přes výše zmíněné vývojové prostředí. Následné testování aplikace probíhalo po jejím nasazení na reálný server, viz kapitola 8. V kapitolách níže jsou popsány jednotlivé metody testování a jejich výsledky.

## 9.1 HTML validace

V rámci testování byla provedena kompletní validace HTML kódu. Ke komplexnímu otestování takto rozsáhlého kódu bylo využito externího volně dostupného nástroje W3C validator, který je implementován ve webovém rozhraní na adrese <https://validator.w3.org/> [3]. Tento komplexní a intuitivní nástroj nabízí kontrolování kódu přístupem přes URL adresy. V tomto případě dochází k přístupu na definované stránky webové aplikace, které jsou následně testovány na valaditu zdrojového kódu. Pokud jsou nalezeny problematické sekce kódu jsou vypsány v podobě varovných či chybových hlášek, ke kterým je zapsáno jejich možné řešení. Takto vypsané chyby zobrazují zároveň číslo řádky a část chybového kódu, viz obrázek 9.1. Veškeré takto



Obrázek 9.1: Chybové hlášky v nástroji W3C validator

nalezené chyby byly po testování opraveny a na aplikační server nasazena opravená verze aplikace. Pro kontrolu veškerých důležitých sekcí webu bylo zapotřebí využít možnosti validovat i ručně nahraný HTML kód, jelikož vět-

šina obsahu je dostupná až po přihlášení do systému. Takovéto kritické sekce byly do validátoru nahrány ručně nebo zkontrolovány přes přístup k lokálně nasazené aplikaci s vypnutou autentizací a testovacími daty.

## 9.2 Mobile responsivity

Při testování byla provedena také kontrola responzivního designu aplikace ve webovém rozhraní. Responzivitou z velké části zajišťuje použitý framework Bootstrap, ale je důležité zachování předepsaných konvencí umísťování jednotlivých elementů do divů a containerů. Toto správné rozvrhnutí bylo otestováno manuálně pomocí reálných a virtuálních zařízení, viz tabulka 9.1. Pro emulaci zobrazení na mobilních zařízeních byla využita volně dostupná

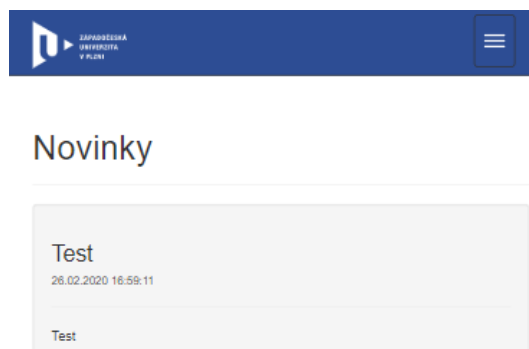
| Testovací zařízení | Typ zařízení       |
|--------------------|--------------------|
| iPhone X           | fyzické zařízení   |
| iPhone 6S Plus     | fyzické zařízení   |
| Galaxy Note 9      | virtuální zařízení |
| Galaxy S8          | virtuální zařízení |
| HTC One M7         | virtuální zařízení |

Tabulka 9.1: Mobilní testovací zařízení

webová aplikace <http://mobiletest.me/>. Tato stránka nabízí zobrazení webového rozhraní na široké škále různých mobilních zařízeních [8], ze kterých bylo vybráno několik modelů, viz tabulka 9.1. Na fyzických zařízeních bylo otestováno chování administrátora a na virtuálních zařízeních běžné akce, které vykonávají uživatelé.

Při testování administrátorských případů užití byly objeveny problémy při akcích spojených s upravováním parametrů či vypisováním nových aktualit na webu. Dané akce lze provést bez limitací i na těchto zařízeních, ale manipulace s nimi je dosti nepřehledná a zdlouhavá. Ve více jak 90% případech se pro tyto akce předpokládá využití desktopového rozlišení, proto byla oprava těchto nedostatků prozatím vynechána. Při nasazení aplikace do ostrého provozu a aktivním využívání je v plánu tyto nedostatky vyřešit a umožnit administrátorům komfortní zadávání i přes mobilní zařízení.

Virtuální zařízení byla použita pro běžné akce související s návštěvou stránek za účelem rychlého získání informací o aktuálním dění v SSC, viz obrázek 9.2. Při procházení webového rozhraní aplikace byly zaznamenány problémy s pravidelnými rozvrhy, kdy docházelo ke špatnému formátování tabulky a písmena byla neúměrně velká ve vztahu k rozlišení obrazovky.



Obrázek 9.2: Webové rozhraní na mobilním zařízení

Tento nedostatek byl vyřešen pomocí CSS, definicí parametrů použitých selektorů v návaznosti na rozlišení zařízení.

## 9.3 Manuální testování

V této kapitole je popsáno manuální testování aplikace, které nahrazuje Selenium testy. Jelikož se jedná o menší projekt, kde byla vydána jedna hlavní verze aplikace, rozhodl jsem se pro provedení testů tímto způsobem. Manuálním testováním jsem otestoval veškeré nově implementované či změněné případy použití, jejich scénáře a výsledky jsou zaznamenány níže. Zároveň byla aplikace uvolněna v konečné fázi implementace k otestování uživatelům SSC, kteří v hojné míře zasílali zpětnou vazbu, díky které se povedlo webové rozhraní odladit a opravit chyby.

### 9.3.1 Přihlášení administrátora

#### Popis scénáře

Aplikace umožňuje administrátorský přístup bez orion účtu, viz kapitola 7.6. Navštívíme URL adresu `ssc2.fav.zcu.cz/AdminLogin`, kde je zobrazen přihlašovací formulář. Vyplníme ho validními přihlašovacími údaji *Login*: admin a vyplníme defaultní hodnotu pro *Password*: VasaAdmin následně po stisknutí enteru či tlačítka *Login* dojde k přihlášení do systému pod uživatelem *admin*.

Zkusíme změnit defaultní heslo pro administrátorský účet v systémových parametrech, kam se dostaneme kliknutím na *Systémové parametry* v horní navigační liště. Zde v seznamu parametrů nalezneme `ADMIN_PASSWORD`, u kterého je zobrazen aktuální hash hesla. Klikneme na tlačítko *Upravit* u seznamu a zobrazí se nám prázdný formulář na zadání nového hesla pro tento

účet. Vyplníme pole **ParamValue** hodnotou **TestAdmin** a potvrdíme. Následně se z aplikace odhlásíme a pokusíme se přihlásit s defaultním heslem, tudíž opakujeme stejné kroky jako v prvním odstavci. Aplikace nás nepřihlásí a dojde k errorovému výpisu *Invalid login*. Po vyplnění správných změněných údajů a odeslání formuláře dojde k přihlášení do aplikace.

Pro otestování znovugenerování defaultního hesla přejdeme opět na systémové parametry a *ADMIN\_PASSWORD* tlačítkem *Smazat* vymažeme. Poté se odhlásíme z aplikace a opět přejdeme na login obrazovku */Admin-Login*. Do přihlašovacího formuláře zadáme defaultní hodnoty a přihlásíme se. V záložce *Systémové parametry* zkontrolujeme znovugenerování parametru *ADMIN\_PASSWORD*.

### Výsledek scénáře

Aplikace se chovala přesně podle popsaného scénáře. Heslo se správně do systému ukládalo zahashované a při pokusu o přihlášení s nesprávným heslem aplikace korektně vypsala chybovou hlášku a nepovolila přihlášení. Zresetování defaultního hesla proběhlo také v pořádku a bylo možno pozorovat, že i když bylo heslo totožné, jako při první návštěvě stránky s parametry, jejich hash hodnota se lišila.

## 9.3.2 Emailové notifikace

### Popis scénáře

Přihlásíme se do aplikace přes stránky *ssc2.fav.zcu.cz* pomocí svého orion loginu. V horní navigační liště vybereme záložku *Chci pomáhat*. Po načtení stránky zde vyplníme přihlášku. Předmět vybereme **MECH 1** a známku lepší jak tři, po úspěšném výběru formulář odešleme. Provedeme odhlášení přes odkaz v pravém horním rohu *Odhlásit se*. Po úspěšném odhlášení se přihlásíme na účet administrátora zadáním URL adresy */AdminLogin* s defaultními přihlašovacími údaji viz kapitola 9.3.1. Po úspěšném přihlášení přejdeme pomocí horního navigačního menu na záložku *Žádosti tutorů*, zde si v seznamu najdeme orion login našeho účtu a po kliknutí na *Detail* si ve spodním menu akcí vybereme *Schválit*.

Po této akci se opět přihlásíme pomocí svého orion účtu a vypíšeme lekci. Na formulář pro vypsání lekce se dostaneme přes horní menu *Vypsát lekci*. Vyplníme formulář validními daty (počet opakování necháme na 1, datum lekce vyplníme zítřejší) a odešleme. Následně se přihlásíme na admin účet (viz kroky z prvního odstavce pro přihlášení na účet administrátora) a zde se přes navigaci *Seznam lekcí* dostaneme na úplný výpis zadaných lekcí do

systému. Nalezneme naši vypsanou z předchozího kroku a tu pomocí *Schválit* potvrdíme. Zůstaneme přihlášení a navštívíme URL adresu */Home/HelpMe*. Na této stránce vyplníme formulář se žádostí o extra lekci, ve kterém jako předmět vybereme **MECH** a po vyplnění zbylých polí validními záznamy formulář odešleme.

Nyní navštívíme svoji emailovou schránku, ve které bychom měli najít tři nové emaily viz obrázek 9.3.

| Předmět   | Odesílatel             |
|---|------------------------|
| Žádost o extra lekci MECH v Student Support Centru                | Student Support Centre |
| Vaše vypsaná lekce z MECH dne 08.04.2020 12:00:00 byla schválena! | Student Support Centre |
| Byl jste přijat jako tutor do projektu Student Support Centre!    | Student Support Centre |

Obrázek 9.3: Odeslané emaily aplikací v průběhu testovacího scénáře

## Výsledek scénáře

Nebylo zaznamenáno žádné nestandardní chování aplikace v průběhu přihlašování uživatelů, ani při odesílání dotazníků. Emaily byly aplikací odesílány bez prodlev, hned jak byly dané akce, při kterých jsou emailové notifikace očekávány, provedeny přes webové rozhraní. Formátování doručených zpráv odpovídá předem definovanému formátu a při prohlížení emailu v emailovém klientovi nedochází k problémům s kódováním znaků. V rámci testování byl objeven překlep v názvu emailu, který je odesílán při schválení lekce. Tento nedostatek byl následně opraven.

### 9.3.3 Vyplnění feedbacku

#### Popis scénáře

Přihlásíme se do aplikace se svým orion účtem a následně se přes navigační panel přepneme na *Seznam lekcí*. Zde uvidíme svoji vypsanou lekci z předchozího testovacího scénáře. Otevřeme si detail této lekce pomocí odkazu *Detail* a následně provedeme vygenerování odkazu formuláře pro zadání feedbacku k dané lekci kliknutím na *Generuj odkaz pro feedback*. Dojde k zobrazení stránky, na které se nachází unikátní URL adresa dotazníku a nad ní QR Code, ve kterém je tato adresa uložena.

Nejdříve otestujeme správnost QR Code jeho načtením do mobilního zařízení (bylo využito mobilního telefonu iPhone X). Po načtení QR se otevřel prohlížeč, který nás automaticky přeměroval na stránku pro zadávání feedbacků. Zkontrolujeme, že nejsme přihlášení do systému svým orion loginem, aby se feedback uložil anonymně a vyplníme do zobrazeného pole

text **Super test** a následně odešleme. Zobrazí se nám obrazovka s textem *Zpráva odeslána*.

Otestování URL odkazu provedeme jeho otevřením v novém okně browseru. Zde si zkontrolujeme, že jsme stále přihlášení do systému a pokud tomu tak není, přihlásíme se svým orion účtem. Poté vyplníme pole pro text **Super test - přihlášený uživatel!** a odešleme. Počkáme na zobrazení stránky s nápisem *Zpráva odeslána* a okno browseru můžeme zavřít.

Pro zkontrolování uložení feedbacků se přihlásíme na účet administrátora, kde přejdeme na záložku *Statistiky* a vyplníme datum *Do* datem o sedm dní větším než je aktuální. Klikneme na tlačítko *Zobraz hodiny*. V seznamu uvidíme naši vypsanou lekci z bodu 7.5, u které zobrazíme zadané feedbacky kliknutím na číslovku zobrazující počet zadaných feedbacků u dané lekce (jedná se o čtvrtý sloupec zleva). V nové záložce se zobrazí tabulka s vypsanými zadanými feedbacky a u prvního z nich očekáváme ve sloupci *Login uživatele* vyplněnou hodnotu *ANONYM*, kdy byl tento feedback zadán nepřihlášeným uživatelem.

Dále zkusíme zadané feedbacky stáhnout na stránce *Statistiky*, kde zadáme stejný datum jako jsme zadávali v odstavci výše a klikneme na *Generuj feedbacky*. Dojde ke stažení souboru *feedback.csv*, ve kterém jsou uloženy dva záznamy viz obrázek 9.4.

```
1 Datum;Od;Do;Predmet;Text
2 8.4.2020;12:00;14:00;MECH;Super test
3 8.4.2020;12:00;14:00;MECH;Super test - přihlášený uživatel!
4
```

Obrázek 9.4: Soubor s vygenerovanými feedbacky

## Výsledek scénáře

Přepínání mezi webovými stránkami probíhalo plynule a v aplikaci nedošlo k žádným zásekům. Přihlášení pomocí orion účtu bylo vždy úspěšné. Zadané texty se do databáze uložili ve správném kódování a následně byly zobrazeny v čitelném formátu. Při anonymním zadávání podnětu k lekci nebyla v systému zaznamenána žádná data, která by danou akci mohla spojit s konkrétním uživatelem, a takovýto zadaný podnět byl následně ve webovém rozhraní zobrazen s uživatelem ANONYM. Při pokusu stáhnout soubor obsahující feedbacky, byl soubor backendem vygenerován v uspokojivém časovém intervalu a po jeho naimportování do tabulkového procesoru Excel byla veškerá data zobrazena korektně.



### 9.3.4 Prohlížení statistik

#### Popis scénáře

Scénář popisuje testování všech možných akcí, které vedou k zobrazení daných statistik, které umí systém generovat. Přihlásíme se na účet administrátora pomocí obrazovky */AdminLogin* a přes horní panel se přesuneme na obrazovku *Statistiky*. Otestování možností formuláře popsanych v kapitole 7.4 bude provedeno v rámci testování možných akcí.

První testovanou akcí je tlačítko *Zobraz hodiny*, do formuláře nevyplňujeme žádné datum a kliknutím na tlačítko se zobrazí několik předpřipravených lekcí, které jsou v systému uloženy. V tomto seznamu nenajdeme lekci, kterou jsme vytvářeli v bodě 9.3.2, protože je naplánována až na zítra. Vrátime se zpět přes horní navigaci na stránku *Statistiky*.

Nyní vyplníme datum do pole *Od* na dnešní den a opět klikneme na *Zobraz hodiny*. Zobrazí se prázdná stránka, protože vybraný interval odpovídá jen dnešnímu dnu a v tento den jsme žádnou lekci v systému nevypsali. Opět se vrátíme na stránku *Statistiky*.

Předposlední testovací akcí je zadání pole *Do* na den o sedm dní větší. Zde se zobrazí seznam hodin uložených v systému, které jsou vyučovány až do zadaného data. Tedy vidíme i naši vypsanou lekci se dvěma feedbacky. V horní liště tabulky se tato hodina přičte do statistik.

Poslední akcí je zadání časového intervalu vyplněním *Od* i *Do*. Nastavíme rozmezí ode dneška do zítra. Po kliknutí na *Zobraz hodiny* se ve výpisu objeví jen hodina, kterou jsme dříve v systému vypsali. V horní liště je započítána jen jedna hodina *MECH* a odučené hodiny z tohoto předmětu jsou 2. Zobrazování feedbacků jsme již otestovali v předcházejícím bodě testování, taktéž jejich stažení.

Stejná pravidla jako jsou popsána výše se aplikují i na tlačítko *Zobraz tutorý*. Zadáme interval ode dneška do zítra a klikneme na toto tlačítko. V přehledné tabulce jsou zobrazení všichni tutoři v SSC. Najdeme si svůj login a zjistíme, že zde máme jednu vyučující hodinu *MECH* a *Počet čistých hodin 2*.

Poslední akce, kterou lze zobrazit statistiky, je tlačítko *Zobraz hodiny za předešlý měsíc*. Po interakci s tlačítkem jsou zobrazené statistiky tutorů za předchozí měsíc. Zde jsou samé nuly, protože v systému není v předchozím měsíci vypsána žádná lekce.

## Výsledek scénáře

Systém reagoval přesně podle definovaných pravidel z kapitoly 7.4. Zobrazování příslušných dat je provedeno ve velmi krátkém časovém intervalu i při zobrazování většího počtu záznamů. A k tomuto formuláři je povolen přístup jen úspěšně přihlášenému uživateli s rolí *ADMIN*.

### 9.3.5 Práva administrátorů

#### Popis scénáře

Cílem tohoto testovacího scénáře je otestovat doimplementované možnosti administrátorů do aplikace viz kapitola 7.7. Přihlásíme se do administrátorského účtu a na obrazovce *Uživatelé* pod nadpisem zkontrolujeme přítomnost čtyř odkazů. Klikneme na *Email MECH tutors!*, dojde k otevření emailového klienta a v kolonce adresáta zkontrolujeme přítomnost svého studentského emailu.

V seznamu uživatelů najdeme svůj login a klikneme na *Upravit*. Zde vyplníme studentské číslo a zaškrtneme ve spodním řádku kolonku **INF** a klikneme *Uložit*. Přehlásíme se na vlastní orion účet a zkusíme vypsát novou hodinu na obrazovce *Vypsát lekci*, kde ve formuláři vyplníme zítřejší datum a jako předmět zvolíme **INF**. Dojde k úspěšnému vypsání lekce a přesměrování na obrazovku *Moje lekce*.

Pro otestování případu užití, kdy je potřeba uživateli odebrat či přiřadit roli „TUTOR“ se opět přihlásíme jako administrátor. V navigaci vybereme záložku *Uživatelé* a opět klikneme na *Upravit* u svého loginu. Zde upravíme položku *Role* tak, že vybereme ze seznamu možnost **USER** a uložíme. Na obrazovce zkontrolujeme, že došlo ke změně role u našeho loginu na „USER“. Pro kontrolu se přehlásíme do svého orion účtu a zkontrolujeme, že nám zmizely z horní nabídky možnosti *Vypsát lekci*, *Moje lekce atd.* Zároveň má uživatel možnost opětovně podat přihlášku přes *Chci pomáhat*.

Posledním krokem scénáře je ověřit funkčnost vymazání uživatele. Pro otestování opět navštívíme stránku *Uživatelé* pod administrátorským účtem a u svého loginu klikneme na *Smazat uživatele* a potvrdíme na zobrazené obrazovce. Ověříme, že smazaný účet zmizel ze seznamu uživatelů a pomocí *Domů* se přesuneme na uvítací stránku. Zde zkontrolujeme, že jsou námi vypsané lekce ve výše uvedených testovacích scénářích podbarveny červeně a jako poznámka je k nim uvedeno *Tutor fired!*

## **Výsledek scénáře**

Testované případy užití fungovaly správně a aplikace reagovala dle očekávání. Po zkontrolování záznamů v databázi po smazání uživatele nebyla nalezena žádná související data. Časté přehlašování mezi účty probíhalo plynule.

# 10 Rozšíření aplikace

V této kapitole jsou uvedena a popsána možná vylepšení a rozšíření systému, který je výstupním produktem této bakalářské práce.

## 10.1 Editace parametrů

Jednou z možností, kde stávající aplikaci vylepšit, je určitě na stránce s editací parametrů. Pro editaci je nyní používáno prosté textové pole či oblast pro delší parametry. Ve webovém rozhraní na desktopu by bylo možno doimplementovat editační okno pro parametry, kde se zadávají a upravují předdefinované texty na webu. Tyto texty jsou zadávány již s HTML značkami a jejich nynější editace je v prostém textovém poli poněkud nepřehledná a je nutno využívat externího softwaru.

## 10.2 Mobilní aplikace

Mobilní aplikace celého SSC by mohla přinést přehledněji veškeré potřebné informace na mobilní zařízení studentů. Zde by byla potřeba takovouto aplikaci vytvořit a následně do back-endu doimplementovat rozumné API, které by aplikaci poskytovalo nezbytná data pro fungování.

## 10.3 Integrace s portal.zcu.cz

Plné integrace se STAGem ZČU by aplikaci přinesla množství rozšíření v podobě přístupu k rozvrhům daných přihlášených uživatelů, kdy by se pravidelný rozvrh studenta promítl do pravidelného rozvrhu SSC. Dále by byla možnost napovídat tutorům, při vypisování hodin v SSC a na obrazovce jim zobrazovat předpřipravené lekce, které se jim nekryjí s jejich rozvrhem ve škole a zároveň jsou v časech, kdy je o služby SSC největší zájem. Tato interakce by mohla být opačného přístupu, kdy by se studentům ve STAGu mohly zobrazovat hodiny, které mají vyučovat v SSC či jejich přijaté extra lekce.

# 11 Závěr

V rámci této bakalářské práce proběhlo seznámení s částí již implementovaného informačního systému pro SSC, který má zlepšit jeho chod a snížit časovou náročnost na plánování provozu. Po analýze již napsaného kódu bylo provedeno jeho přepsání z .NET Frameworku do .NET Core, aby bylo možno aplikaci provozovat nezávisle na platformě. Při přepisování stávající aplikace byly postupně implementovány zbylé definované případy užití a po analýzách vylepšeny již stávající. Pro tyto nové případy užití bylo potřeba změnit databázový model, aby splňoval požadavky na nové funkčnosti systému. Zároveň došlo k migraci celé databáze z řešení Oracle do MySQL, která je nově vybraným frameworkem podporována bezplatně.

Výsledná aplikace a její webové rozhraní je především určena studentům Západočeské univerzity v Plzni, kteří chtějí využívat možností Student Support Centra v oblasti výuky. Její způsob užívání se odvíjí od role, kterou má daný uživatel přiřazenou v systému. Studenti, kteří budou chtít využít nadstandardních možností centra, mají možnost podání žádosti o vyučující hodinu nad rámec pravidelných lekcí. Dále mají možnost podání přihlášky, aby mohli v centru vyučovat předměty. Tutoři jsou v systému studenti, kterým bylo umožněno v rámci centra pomáhat ostatním. Tito uživatelé budou nejvíce využívat možnosti vypisování svých vlastních lekcí do pravidelného rozvrhu, které poté musí schválit administrátor. Zároveň mohou akceptovat žádosti o extra lekce. Administrátorům systém umožní efektivní metodu, jak připravovat pravidelný rozvrh na celý měsíc dopředu pomocí hromadného potvrzování vypsaných lekcí. Dále bude sloužit jako zdroj statistik návštěvnosti a pro sběr zpětné vazby od studentů, kteří využijí služeb centra. Administrátor bude také jako jediný spravovat ostatní uživatele, kterým bude moci přiřazovat předměty, které budou moci vyučovat.

V úvodu bakalářské práce je uvedeno, že by tato aplikace měla nahradit stávající web Student Support Centra. Pro tento krok byla celá aplikace v rámci bakalářské práce nasazena na reálný server a je dostupná na dočasné adrese <https://ssc2.fav.zcu.cz/>. Do budoucna je aplikace připravena na nasazení do ostrého provozu. V případě změny hostingu, je možno jako server použít řešení Windows či Debian s databází MySQL.

# Literatura

- [1] *What is SSL, TLS and HTTPS?* [online]. Symantec Corporation, 2018. [cit. 2020/03/18]. Dostupné z: <https://www.websecurity.digicert.com/security-topics/what-is-ssl-tls-https>.
- [2] *Databázové systémy* [online]. Vít Fábera, 2017. [cit. 2020/03/18]. Dostupné z: <https://www.fd.cvut.cz/personal/xfabera/DB/prednasky/prednaska3/prednaska-DATS-03-2017.pdf>.
- [3] *Markup Validation Service* [online]. W3C, 1996. [cit. 2020/03/27]. Dostupné z: <https://validator.w3.org>.
- [4] *Lookup table* [online]. ZIFF DAVIS, LLC. PCMAG DIGITAL GROUP, 1996. [cit. 2020/03/27]. Dostupné z: <https://www.pcmag.com/encyclopedia/term/lookup-table>.
- [5] *A pure C# Open Source QR Code implementation* [online]. Raffael Herrmann, 2018. [cit. 2020/03/27]. Dostupné z: <https://github.com/codebude/QRCoder/wiki>.
- [6] *Introduction* [online]. Jeffrey Stedfast, 2020. [cit. 2020/03/30]. Dostupné z: <http://www.mimekit.net/docs/html/Introduction.htm>.
- [7] *HttpServerUtility.HtmlEncode Metoda* [online]. Microsoft, 2020. [cit. 2020/03/31]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/api/system.web.httpserverutility.htmlencode?view=netframework-4.8>.
- [8] *Mobile test* [online]. Vangelis Bibakis, 2014. [cit. 2020/04/03]. Dostupné z: <http://mobiletest.me/>.
- [9] *Plugin Handbook* [online]. wordpress.org, 2020. [cit. 2020/04/06]. Dostupné z: <https://developer.wordpress.org/plugins/>.
- [10] *Choosing between .NET Core and .NET Framework for server apps* [online]. Microsoft, 2018. [cit. 2020/04/06]. Dostupné z: <https://docs.microsoft.com/cs-cz/dotnet/standard/choosing-core-framework-server>.
- [11] *MVC architektura* [online]. itnetwork.cz, 2018. [cit. 2020/04/14]. Dostupné z: <https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor>.

- [12] *An introduction to NuGet* [online]. Microsoft, 2019. [cit. 2020/03/18].  
Dostupné z: <https://docs.microsoft.com/cs-cz/nuget/what-is-nuget>.
- [13] *What is ASP.NET Core?* [online]. Microsoft, 2020. [cit. 2020/04/14].  
Dostupné z: <https://dotnet.microsoft.com/learn/aspnet/what-is-aspnet-core>.
- [14] *Nullable value types (C# reference)* [online]. Microsoft, 2020.  
[cit. 2020/04/14]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/language-reference/builtin-types/nullable-value-types>.
- [15] *Automatic renewal* [online]. win-acme, 2020. [cit. 2020/04/14]. Dostupné z: <https://www.win-acme.com/manual/automatic-renewal>.
- [16] *Amazon Compute Service Level Agreement* [online]. win-acme, 2019.  
[cit. 2020/04/20]. Dostupné z: <https://aws.amazon.com/compute/sla/>.
- [17] *What is GNU?* [online]. Free Software Foundation, Inc., 2020.  
[cit. 2020/03/18]. Dostupné z: <https://www.gnu.org/>.
- [18] *.NET Programming Languages* [online]. Microsoft, 2020. [cit. 2020/03/18].  
Dostupné z: <https://dotnet.microsoft.com/languages>.
- [19] *Student Support Centre* [online]. Fakulta aplikovaných věd ZČU, 2019.  
[cit. 2020/03/18]. Dostupné z: <http://ssc.fav.zcu.cz/>.
- [20] *Lazy Loading Related Data In Entity Framework Core* [online]. Learn Entity Framework Core, 2020. [cit. 2020/03/18]. Dostupné z: <https://www.learnentityframeworkcore.com/lazy-loading>.
- [21] *Creating A New MySQL Connection (Simple)* [online]. Oracle Corporation, 2020. [cit. 2020/03/18]. Dostupné z: <https://dev.mysql.com/doc/workbench/en/wb-mysql-connections-new.html>.
- [22] *Tabulky* [online]. Jak psát web píše Yuhů, Dušan Janovský, 2019.  
[cit. 2020/03/18]. Dostupné z: <https://www.jakpsatweb.cz/html/tabulky.html>.
- [23] *MIT License* [online]. Techopedia Inc., 2020. [cit. 2020/03/18]. Dostupné z: <https://www.techopedia.com/definition/3287/mit-license>.
- [24] *Minimum password length* [online]. Microsoft, 2020. [cit. 2020/03/18].  
Dostupné z: <https://docs.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/minimum-password-length>.

- [25] *Password Hashing using Rfc2898DeriveBytes* [online]. Cidean, LLC, 2029. [cit. 2020/03/18]. Dostupné z: <https://www.cidean.com/blog/2019/password-hashing-using-rfc2898derivebytes/>.
- [26] CONOLLY, C. E. B. a. R. H. T. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- [27] NEHA SHRIVASTAVA, R. V. *.NET Core 2.0 By Example*. Packt Publishing, 2018. ISBN 9781788395090.



# Seznam zkratek

- **SSC** - Student Support Centre
- **PDF** - Portable Document Format - univerzální souborový formát pro ukládání dokumentů
- **HTML** - Hypertext Markup Language - značkovací jazyk
- **JS** - JavaScript - objektově orientovaný skriptovací jazyk
- **CSS** - Cascading style sheets - jazyk pro stylizaci HTML elementů
- **SQL** - Structured Query Language - standartizovaný strukturovaný dotazovací jazyk
- **MVC** - Model, View, Controller - softwarová architektura
- **HTTP** - Hypertext transfer protocol - internetový protokol pro výměnu HTML dokumentů
- **URL** - Uniform resource locator - definovaná struktura odkazující na přesné umístění dokumentu na internetu
- **Tutor** - student s právy na vyučování v Student Support Centre
- **GDPR** - obecné nařízení o ochraně osobních údajů
- **IDE** - vývojové prostředí
- **API** - Application Programming Interface
- **IIS** - Internet Information Services
- **ZSWI** - Základy softwarového inženýrství

# Přílohy

## A Instalační příručka

Instalační příručka je napsána pro operační systém Windows, který disponuje nainstalovaným programem Visual Studio a nutným softwarem popsaným v kapitole 8.2.1. Zároveň se předpokládá, že stroj, na kterém se systém nasazuje, je připojen do internetové sítě Západočeské univerzity v Plzni. Tento požadavek je nutný pro správné připojení k databázi.

### Publikování aplikace

- Zkopírujeme složku **ISSSC** z příloženého CD na lokální disk.
- Přes navigaci vstoupíme do této složky a otevřeme projekt ve Visual Studio pomocí souboru *ISSSC.sln*
- V horní navigační liště vybereme záložku *Build*
- V otevřené nabídce vybereme *Publish ISSSC*
- Pod záložkou *Publish* zvolíme z výběrového okna možnost *Publish ISSSC* (výchozí nastavená cesta pro vytvoření aplikace je **C:\ISSSC**, pokud k této cestě nemáte přístup je nutno cestu pomocí možnosti *Edit* pod výběrem změnit na platnou, jedná se o pole *Target location*)
- Po zkontrolování nastavené cesty zmáčkneme tlačítko *Publish*
- Na zvolené cestě se vytvořila složka **ISSSC**, ve které je sestavená aplikace

### Nasazení aplikace do IIS

Pro úspěšné spuštění aplikace v IIS jsou nutná práva administrátora.

- Zkontrolujeme, zda běží IIS
  - Otevřeme *Ovládací panely* a následně v nabídce *Programy a funkce*
  - V levé panelu vybereme *Zapnout nebo vypnout funkce systému Windows* (pro tento krok jsou nutná práva administrátora)

- V otevřeném zkontrolujeme, za je u možnosti *Internetová informační služba* zobrazen černý čtvereček (pokud tomu tak není, klikneme na bílý čtvereček)
- Potvrdíme tlačítkem *OK* a zkontrolujeme běžící IIS pomocí webového prohlížeče, kde zadáme adresu *//localhost* (zobrazí se modrá obrazovka s nadpisem Internet Information Services)
- Pomocí *Ovládací panely - Nástroje pro správu* otevřeme aplikaci *Správce Internetové informační služby*
- V levém menu vybereme záložku *Weby* a pod ní *Defaultní web*
- V pravém panelu zvolíme možnost *Základní nastavení...*
- Zde zkontrolujeme, že je vybrán Fond aplikací: *DefaultAppPool* (pokud není, zvolíme tuto možnost přes tlačítko *Vybrat...*)
- Nastavíme cestu ke složce na publikovanou aplikaci (defaultně *C:\ISSSC*) a nastavíme *Připojit jako...*
- Po otevření nabídky vybereme možnost *Určitý uživatel* a klikneme na *Nastavit*, kde vyplníme přihlašovací údaje k účtu s právy administrátora
- Zkontrolujeme nastavení pomocí tlačítka *Test nastavení...* - v zobrazené tabulce jsou obě řádky zobrazeny se zelenou fajfkou
- Zavřeme okno kontroly a potvrdíme tlačítkem *OK*
- V pravém menu restartujeme webovou službu pomocí *Restartovat*
- Ověříme běžící webovou aplikaci přes webový prohlížeč na adrese *//localhost*

## B Uživatelská příručka

### B.1 Uživatel

Má možnost procházet veřejně přístupné části aplikace a prohlížet rozvrh vypsaných hodin (pravidelné i schválené nepravidelné lekce). Nemá přístup do sekcí, kde systém vyžaduje ověření přes Orion login. V těchto částech se mu zobrazuje upozornění, že se část systému odemkne po přihlášení.

### B.2 Ověřený uživatel

Uživateli se zpřístupní části systému, které jsou dostupné po ověření přes Orion login. Povolené případy užití:

- **Žádost o extra lekci** - na stránce *Potřebuji pomoc* je možnost vyplnit formulář pro žádost o extra lekci nad rámec pravidelného rozvrhu. Zde je potřeba vyplnit datum, předmět a látku, kterou daný uživatel chce na hodině probírat.
- **Žádost o tutorství** - jedná se o formulář na stránce *Chci pomáhat*, kde je potřeba vyplnit z předdefinovaných akceptovaných předmětů předmět, který daný uživatel ve svém studiu splnil a zadat z něho dosaženou známku. Předmětů lze přidat na přihlášku vícero. Pole pro zadání dalšího předmětu se zobrazí po kliknutí na „Přidat předmět“.

### B.3 Tutor

Ověřený uživatel, který úspěšně prošel přijímacím řízením SSC a byly mu pro tuto roli nastavena práva. Zpřístupní se mu části systému, které jsou potřeba pro plnění role tutora.

- **Vypsání lekce** - lze provést na stránce Vypsat lekci, která je dostupná z horního navigačního panelu. Na této stránce je potřeba vyplnit dotazník, který požaduje datum a čas začátku lekce. Je potřeba vybrat předmět, který byl danému uživateli v přihlášce schválen a na který chce lekci vypsát. Pole počet opakování umožňuje danou lekci vypsát s týdenním opakováním podle zadaného celého nezáporného čísla. Délka lekce je nastavována automaticky a její délka je zobrazena v legendě pod formulářem. Vypsát lekci jde i přes klikatelný odkaz na stránce Seznam lekcí, který se nachází pod nadpisem Lekce.
- **Žádost o extra lekci** - viz kapitola B.2 Žádost o extra lekci

- **Zrušení špatně vypsané lekce** - špatně vypsanou lekci lze smazat pouze v případě, že ještě nebyla přijata správcem. Takového zrušení lze provést na obrazovce Seznam lekcí přes klikatelný odkaz u příslušné lekce v pravé části obrazovky.
- Obrazovka **Moje lekce** slouží k zobrazení všech lekcí, které přihlášený tutor vypsál.
- **Zadávání návštěvnosti lekce** - tutor je po každé proběhlé lekci a následném přihlášení do systému vyzván k vyplnění dotazníku o návštěvnosti jeho odučené lekce.
- **Generování odkazu pro zpětnou vazbu** - Na obrazovce Seznam lekcí lze po schválení příslušné lekce pod detailem generovat odkaz pro zadávání zpětné vazby k dané lekci. Po kliknutí na *Generuj odkaz pro feedbacky* je zobrazena stránka, na které se nachází QR kód pro snadné načtení stránky v mobilním zařízení či přímý odkaz na formulář pro zadání zpětné vazby.

## B.4 Administrátor

- **Vytvoření novinky** - nová novinka se vytvoří na stránce Novinky kliknutím na odkaz Vytvořit novinku
- **Editace novinky** - novinky lze editovat přes odkaz Upravit, který se nachází u každé novinky ve spodní části šedivě podbarvené plochy.
- **Smazání novinky** - každou novinku lze také smazat pomocí odkazu Smazat ve spodní části šedivě podbarveného pozadí novinky.
- **Vypsání lekce** - administrátor má umožněno vypisovat lekce stejně jako může tutor, viz kapitola B.3. Administrátorovi je navíc umožněno vypsát lekci pro jakéhokoliv ověřeného uživatele v systému z libovolného předmětu nad rámec jeho schválených předmětů.
- **Akceptace vypsání lekcí** - administrátor může potvrzovat vypsání lekce tutorům pomocí klikatelného odkazu na obrazovce Seznam lekcí, který se nachází v pravé části příslušného řádku se záznamem o lekci. Potvrzovat lekce lze i hromadně přes grafické rozhraní, které se podobá rozvrhu. Toto rozhraní je dostupné pod klikatelným odkazem Grafický rozvrh na téže stránce, viz kapitola B.4

- **Grafický rozvrh** - na této stránce je interaktivně zobrazen rozvrh vypsaných lekcí. Na řádcích jsou zobrazeni tutoři a ve sloupcích veškeré vypsané termíny v daném měsíci, který je vybrán ze select boxu nad rozvrhem. Zeleně podbarvené pole znázorňuje již akceptovanou lekci, oranžově znázorňuje vypsanou lekci a šrafované pole reprezentuje vybranou lekci k potvrzení. Lekcí lze vybrat najednou více a všechny potvrdit tlačítkem SEND. Rozvrh je interaktivní a při klikání se automaticky obnovují hodnoty vybraných hodin u tutorů, tak hodin celkových v daný termín. Aktuálně vybraný měsíční rozvrh lze stáhnout v reprezentaci .csv, kterou lze snadno importovat do MS Excel a data ze systému dále zpracovávat a upravovat.
- **Změna systémových parametrů** - parametry nutné pro běh IS, které jsou konfigurovatelné. Jejich úprava probíhá přes odkaz Upravit v pravé části příslušného řádku parametru.
- **Správa předmětů** - do IS lze přidávat předměty či je odebírat. Existují dva druhy předmětů tzv. hlavní předmět a podřazený předmět. Hlavní předmět je ten předmět, který lze v SSC vypsat jako lekci (jedná se hlavně o MAT, INF). Takovýto nadřazený předmět má své podřazené, které je možné vyplňovat při přihlášce tutora a jsou dostatečné pro výkon tutora daného nadřazeného předmětu. Přidání hlavního předmětu probíhá nastavením parametru Lesson na „false“. Podřazený předmět má parametr Lesson „true“ a je mu nastaven nadřazený předmět.
- **Akceptace a zamítnutí přihlášek** - probíhá na obrazovce Žádosti tutorů, kde jsou zobrazeny veškeré aktuální přihlášky do SSC. Po kliknutí na detail dané přihlášky je možné takovouto přihlášku přijmout kliknutím na „Schválit“ či zamítnout kliknutím na „Zamítnout“.
- **Odesílání hromadných emailů** - IS umožňuje odesílat hromadné emaily tutorům přes odkaz na obrazovce Uživatelé. Tyto odkazy jsou zobrazeny přímo pod hlavním nadpisem a je možno odeslat email všem tutorům najednou či skupině tutorů, kteří mohou vyučovat daný předmět. Jedná se o prostý HTML *mailto*, který otevře emailového klienta a předvyplní adresy příjemců.
- **Práce s uživateli** - uživatele lze ze systému smazat či je editovat. Smazání probíhá přes odkaz „Smazat uživatele“ u daného záznamu uživatele na obrazovce Uživatelé. Editace se provádí pomocí odkazu

„Upravit“, který zobrazí formulář pro editaci daného uživatele. Uživateli lze měnit veškeré atributy a přidávat mu oprávnění na výuku dalších předmětů.

- **Prohlížení statistik** - statistiky lze prohlížet na stránce Statistiky. Tato stránka umožňuje zobrazovat statistiky pro tutorů B.4, statistiky odučených lekcí B.4 a prohlížet zadané podněty k lekcím B.4. Při odeslání nevyplněného formuláře se zobrazí veškerá příslušná data uložená v systému.
- **Statistiky tutorů** - lze zobrazit přes stránku Statistiky, kdy je možné zvolit dané časové rozpětí, za které se mají statistiky zobrazit při vyplnění formuláře a jeho odesláním pomocí „Zobraz tutorů“, či je zde možnost rychlejšího přístupu k datům o tutorech za předešlý měsíc pomocí „Zobraz hodiny za předešlý měsíc“. Obě varianty zobrazí obrazovku se všemi tutorů a v přehledné formě je zde vidět celkový počet vyučovacích hodin, kolik vyučovacích hodin daný tutor vyučoval, jaký předmět a dále pak čistý počet hodin, který je dán násobkem počtu vyučovacích hodin a standardní délkou jedné hodiny.
- **Statistiky hodin** - při vyplnění formuláře a odeslání pomocí tlačítka „Zobraz hodiny“ je zobrazena obrazovka, na které jsou vypsané veškeré lekce, které se v daném časovém rozpětí vyučovaly. Je zde zobrazen předmět, na který byla daná hodina vypsaná, informace o dané hodině a dále klikatelný odkaz na zobrazení všech podnětů, které k dané hodině byly zadány uživateli.
- **Prohlížení podnětů k lekci** - prohlížení zpětné vazby je také možné přes obrazovku Statistik hodin viz B.4 či přes vygenerovaný .csv soubor, který lze snadno importovat do tabulkového procesoru Excel a v tabulce zobrazit veškeré podněty k lekcím v období, které jsme zadali do formuláře před odesláním.