



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

**MODULY IS VUT PRO ZÁVĚREČNÉ PRÁCE A STÁTNÍ
ZÁVĚREČNÉ ZKOUŠKY**

BUT IS MODULES FOR FINAL THESES AND STATE FINAL EXAMS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ROMAN BÁRTL

VEDOUcí PRÁCE

SUPERVISOR

Ing. JAROSLAV DYTRYCH, Ph.D.

BRNO 2021

Zadání diplomové práce



Student: **Bártl Roman, Bc.**
Program: Informační technologie a umělá inteligence
Specializace: Informační systémy a databáze
Název: **Moduly IS VUT pro závěrečné práce a státní závěrečné zkoušky**
BUT IS Modules for Final Theses and State Final Exams
Kategorie: Informační systémy
Zadání:

1. Seznamte se s jazyky PHP a SQL a s technologiemi využívanými při rozvoji webové části centrálního informačního systému VUT.
2. Prostudujte problematiku projektových prací, semestrálních projektů, závěrečných prací a státních závěrečných zkoušek v IS FIT a dostupné moduly IS VUT, které lze pro tyto agendy využít.
3. Navrhněte nové moduly a úpravy stávajících modulů webové části IS VUT pro projektové praxe, semestrální projekty, správu závěrečných prací a pro komise pro státní závěrečné zkoušky. Nové a upravené moduly musí pokrýt veškerou aktivně využívanou funkcionalitu odpovídajících částí IS FIT.
4. Implementujte navržené řešení.
5. Zhodnoťte dosažené výsledky a vytvořte stručný plakát prezentující výsledky práce.

Literatura:

- Dle doporučení vedoucího

Při obhajobě semestrální části projektu je požadováno:

- Body 1, 2 a 3.

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

Vedoucí práce: **Dytrych Jaroslav, Ing., Ph.D.**
Konzultant: Tomšů Roman, Ing., VUT v Brně
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2021
Datum odevzdání: 18. května 2022
Datum schválení: 1. listopadu 2021

Abstrakt

Cílem této práce je navrhnout a implementovat moduly IS VUT pro projektové praxe, závěrečné práce a státní závěrečné zkoušky v rámci přechodu fakulty informačních technologií na tento systém. Součástí práce je navrhnout nová přístupová práva včetně parametrizace pro úpravu chování modulů. Práva i parametrizace budou nastavovány v informačním systému Apollo. Pro implementaci je využit framework Zend pro jazyk PHP. Moduly budou využívat jak zaměstnanci fakulty pro výpis zadání, přijímání studentů atp., tak i samotní studenti pro zobrazování seznamu prací, na které se mohou přihlašovat.

Abstract

The aim of this work is to design and implement BUT IS modules for project practise, final theses and state final exams within the transition of the Faculty of Information Technology to this system. Part of the work is to design new access rights, including parameterization to modify the behavior of modules. Rights and parameterization will be set in the Apollo information system. The PHP Framework Zend will be used for implementation. Modules will be used by faculty staff to create topics, accept students, etc., as well as by students themselves to display a list of thesis which they can apply for.

Klíčová slova

Závěrečné práce, státní závěrečné zkoušky, projektové praxe, IS VUT, web, PHP, Zend

Keywords

Final thesis, state final exams, project practise, BUT IS, web, PHP, Zend

Citace

BÁRTL, Roman. *Moduly IS VUT pro závěrečné práce a státní závěrečné zkoušky*. Brno, 2021. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Jaroslav Dytrych, Ph.D.

Moduly IS VUT pro závěrečné práce a státní závěrečné zkoušky

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Jaroslava Dytrycha, Ph.D. Další informace mi poskytli zaměstnanci Centra výpočetních a informačních služeb VUT. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Roman Bártl
11. května 2022

Poděkování

Rád bych touto cestou vyjádřil poděkování svému vedoucímu Ing. Jaroslavu Dytrychovi, Ph.D. za jeho pomoc, cenné rady a trpělivost při vedení tohoto projektu. Současně bych rád poděkoval všem zaměstnancům CVIS, zejména Ing. Romanu Tomšů a Ing. Marku Strakošovi za odborné konzultace a rady, které mi pomohly lépe pochopit strukturu centrálního systému. A v neposlední řadě bych rád také poděkoval všem svým nejbližším, kteří mě podporovali během této práce i po dobu celého studia.

Obsah

1	Úvod	3
2	Popis životního cyklu závěrečných prací	5
2.1	Popis projektové praxe	5
2.2	Popis závěrečných prací a SZZ	6
3	Použité technologie	9
3.1	IS VUT Apollo	9
3.2	IS VUT na webu	9
4	Ukládání dat v centrální databázi	11
4.1	Způsob ukládání předmětů	11
4.2	Způsob ukládání informací spojených se studiem	12
4.3	Limity vedených prací	12
4.4	Informace o závěrečných pracích	12
4.5	Příhlášky studentů k závěrečným pracím	14
4.6	Hodnocení závěrečných prací	14
4.7	Ukládání komisí obhajob	15
4.8	Ukládání témat zkoušených u SZZ	16
4.9	Ukládání státních závěrečných zkoušek	16
5	Návrh modulů k projektovým praxím	18
5.1	Návrh databáze	18
5.2	Návrh práv pro modul v aplikaci Teacher	19
5.3	Návrh parametrizace pro nové moduly	20
5.4	Návrh modulu pro StudIS	21
5.5	Návrh modulu pro Teacher	22
6	Návrh modulů k závěrečným pracím	25
6.1	Návrh práv pro moduly závěrečných prací	25
6.2	Návrh modulu pro přehled závěrečných prací v aplikaci Teacher	26
6.3	Návrhy úprav stávajícího modulu pro vypisování a editaci zadání	26
6.4	Návrhy změn modulů v aplikaci StudIS	30
6.5	Veřejný seznam závěrečných prací	31
7	Návrh modulů pro správu komisí projektových praxí a semestrálních projektů	32
7.1	Návrh úpravy databáze	33

7.2	Návrh práv pro modul	35
7.3	Návrh datových modelů	35
7.4	Návrh modulu pro Teacher	35
7.5	Návrh modulu pro StudIS	37
8	Návrh modulů pro státní závěrečné zkoušky	39
8.1	Návrh úprav modulu pro posudky závěrečných prací	39
8.2	Návrh úprav modulu pro správu komisí SZZ	40
8.3	Návrh úprav modulu pro zadávání výsledků SZZ	42
9	Implementace agend	46
9.1	Základní struktura agend	46
9.2	Manipulace s daty v databázi	47
9.3	Datagrid	48
9.4	Implementace agendy projektové praxe	49
9.5	Implementace agendy závěrečných prací	52
9.6	Implementace agendy pro správu komisí	53
10	Testování a nasazení agend	59
10.1	Projektové praxe	59
10.2	Závěrečné práce	59
10.3	Správa komisí pro semestrální práce a projektové praxe	60
11	Závěr	61
	Literatura	62

Kapitola 1

Úvod

Doposud využívala fakulta informačních technologií především svůj vlastní webový informační systém WIS, který se ale postupem času stal náročným na údržbu. Neumožňuje využívat centrální mobilní aplikaci, která se aktuálně vyvíjí. Dalším problémem je synchronizace výsledků v jednotlivých předmětech mezi WISem a centrálním systémem. Proto bylo rozhodnuto o přechodu fakulty na centrální systém VUT, v rámci kterého je potřeba navrhnout úpravy či nové moduly tohoto systému tak, aby vyhovovaly stávajícím procesům fakulty.

Tato práce se zabývá návrhem a implementací modulů spojených s projektovými pracemi, závěrečnými pracemi a státními závěrečnými zkouškami. Projektové praxe zahrnují skupinu předmětů, v rámci kterých se mohou nadaní studenti zapojit do řešení praktických projektů. Typicky projektová praxe předchází závěrečné práci a studenti tento projekt nakonec rozšíří a udělají z něj svoji bakalářskou či diplomovou práci. Závěrečné práce a státní závěrečné zkoušky jsou jedním z nejdůležitějších procesů jakékoliv univerzity. Závěrečnou práci studenti prokazují, že jsou schopni řešit a ústně a písemně prezentovat zadaný problém a nakonec své vlastní přístupy k řešení i obhájit v rámci státní závěrečné zkoušky. S tím souvisí i sestavování komisí pro obhajoby a výsledné hodnocení studentů.

Kapitola číslo 2 popisuje jednotlivé procesy fakulty související s implementovanými moduly, jak probíhají a jak na sebe navazují. Kapitola tedy popisuje průběh projektu v rámci projektových praxí s návazností na semestrální projekt a následně na závěrečnou práci a její obhajobu u státní závěrečné zkoušky. Dále popisuje proces sestavování komisí a způsob hodnocení studenta.

Kapitoly číslo 3 a 4 se zaměřují na technologie využívané v centrálním systému VUT. Popis používaných technologií je nezbytný pro pochopení následného návrhu a implementace jednotlivých modulů.

V kapitolách 5 až 8 jsou popsány návrhy jednotlivých modulů informačního systému VUT. Kapitoly obsahují návrhy jak pro aplikaci **Teacher**, tak i pro aplikaci **StudIS**, včetně návrhu databázových entit, práv, parametrizace a uživatelského rozhraní.

A v neposlední řadě kapitoly 9 a 10 popisují implementaci jednotlivých agend včetně jejich testování a nasazení.

Centrální systém spravuje Centrum výpočetních a informačních služeb, samostatná součást VUT, proto veškerá práce na tomto modulu byla konzultována s jejími zaměstnanci. Pro webovou část informačního systému bylo vytvořeno bezpočet komponent, z nichž některé jsou v tomto projektu využity.

Doposud se v ostrém provozu na centrálním systému začal využívat nový modul pro projektové praxe, který je spuštěn od začátku listopadu roku 2021. Další nový modul, který

se v době psaní textu již používá, slouží pro zobrazování seznamu vypsaných závěrečných prací (podobný modul žádná fakulta doposud nepotřebovala). Současně s ním byl upraven stávající modul pro vypisování a editaci zadání závěrečných prací, pomocí kterého již zaměstnanci fakulty vypisují zadání na následující akademický rok. Agenda pro komise semestrálních projektů a projektových prací byla implementována během letního semestru v akademickém roce 2021/2022. V navazující práci bude upraven dle návrhu také modul pro výpis posudků závěrečných prací.

Kapitola 2

Popis životního cyklu závěrečných prací

Tato kapitola popisuje, jakým způsobem probíhají závěrečné práce od vypsání zadání až po jejich obhajobu u státní závěrečné zkoušky na Fakultě informačních technologií. Kapitola obsahuje i popis projektových prací, které typicky předcházejí závěrečné práci a studentovi je umožněno na zadaném tématu v rámci závěrečné práce pokračovat.

2.1 Popis projektové praxe

Projektová praxe zahrnuje skupinu předmětů, která umožňuje nadaným studentům podílet se během studia v rámci výuky na praktických projektech a seznámit se při tom s nejnovějšími technologiemi. Studenti si sami vybírají projekt, na kterém by chtěli spolupracovat, a následně jsou na projekty přijímáni školiteli. Tento projekt poté řeší v rámci 1–3 předmětů dle svého studijního programu. Aby mohl student s projektem pokračovat v navazujícím předmětu, musí v aktuálním předmětu dosáhnout určitých výsledků.

Pro bakalářské studium jsou to předměty IP1, IP2 a IP3, pro magisterské PP1 a PP2, přičemž se předpokládá, že student na vybrané zadání navazuje závěrečnou prací. Projektové praxe jsou vypisovány i pro Erasmus (předměty IP1e, IP2e, PP1e a PP2e) jako alternativa ke kvalifikační práci, která je vždy na dva semestry, zatímco projektovou praxi mohou zahraniční studenti řešit pouze jeden semestr [4].

Projektové praxe jsou zahájeny vypsáním různých témat školiteli (vedoucími projektů). Termíny vypisování zadání se pro různé studijní programy typicky liší. Po vypsání je projekt ve stavu *příprava* a začíná proces schvalování. Zadání mohou být schválena či zamítnuta pouze vedoucími ústavů, pro které bylo zadání vypsáno, jejich zástupci, nebo osobami s přidělenými nejvyššími právy (viz kapitola 5.2). Jakmile je zadání schváleno, mohou se začít přihlašovat studenti. Student se může v určitém časovém období přihlásit na jeden či více libovolných projektů v rámci studijního programu, který právě v aktuálním studiu studuje, nebo je zaregistrován po domluvě některým ze školitelů. Přijímání (příhláška může být i zamítnuta) probíhá taktéž v předem stanoveném termínu, přičemž platí, že student může být přijat pouze na jediný projekt. K přihlášce školitel navíc přidává své vyjádření.

V bakalářském studiu se studenti typicky přihlašují v letním semestru na předmět IP1 vypsáný následující akademický rok, v magisterském studiu zase během zkouškového období za zimní semestr na předmět PP1 na následující letní semestr.

2.2 Popis závěrečných prací a SZZ

Závěrečné práce a státní závěrečné zkoušky jsou jedním z nejdůležitějších procesů jakékoliv univerzity. Závěrečnou prací studenti prokazují, že jsou schopni samostatně řešit a ústně a písemně prezentovat zadaný problém a nakonec své vlastní řešení i obhájit. Bakalářská práce se typicky řeší ve třetím ročníku v rámci dvou předmětů Semestrální projekt (ITT) a Bakalářská práce (IBT). Diplomová práce se pak řeší ve druhém ročníku navazujícího magisterského studia v rámci předmětů Semestrální projekt (SEP) a Diplomová práce (DIP) [2, 3].

V případě, že studenti nemají již registrovaný projekt v rámci projektových praxí, na kterém se rozhodli pracovat i v rámci své závěrečné práce, je typický postup registrace na zadání následující.

Přibližně od listopadu, v ročníku před předpokládaným ukončením studia, si mohou studenti předběžně vybírat ze zadání zobrazených v informačním systému, na kterém tématu by chtěli pracovat. Na závěrečné práci tak student může pracovat i s více než půlročním předstihem. Na každé zadání smí být registrován pouze jediný student, ale pokud by se stalo, že více studentů má zájem o podobné téma, smí vyučující vypsát další velmi podobné zadání.

Každý rok začátkem září se nabídka zadání pro studenty skryje a vedoucí prací vypisují, případně aktualizují zadání na příští akademický rok. Po určité době se nabídka studentům opět zobrazí. Nejpozději od poloviny září je studentům doporučeno začít si pro aktuální akademický rok shánět zadání, pokud nemají doposud žádné přihlášené. Následně probíhá elektronická registrace studentů v informačním systému na zbývající zadání pro ty, kteří se doposud nedomluvili s některým z vedoucích, aby je přihlásil. Přibližně týden po ukončení registrací lze ještě zadání doladit dle společné vize studenta i vedoucího. Na poslední úpravy má vedoucí čas ještě přibližně dva týdny, protože koncem října zasedá komise pro zadání na jednotlivých ústavech a rozhoduje se, která zadání budou schválena, a která bude nutné upravit. Položky ukládané k závěrečným pracím jsou blíže popsány v kapitole 4.4.

Vedoucí ústavů následně schvalují jednotlivá zadání, přičemž jsou dva základní postupy, jak tento proces probíhá. Prvním z postupů je, že vedoucí ústavu schválí všechna zadání vypsána pro jeho ústav (studenti je však zatím k dispozici nemají, protože je před oficiálním termínem zadání). Následně vytisknou zadání ve formátu *PDF* a navrhnou úpravy těch zadání, která nejsou zcela v pořádku. Poté jsou do systému nahrána oskenovaná zadání, systém je identifikuje podle čárového kódu a následně nastaví jako neschválená, přičemž je potřeba uložit aktuální stav textových položek pro pozdější dohledání změn. Tím jsou zadání opět odemčena pro jejich vedoucí, kteří pak mají čas na zapracování připomínek. Vedoucí ústavu si poté zobrazí rozdíl mezi starou uloženou verzí zadání a novou verzí, a pokud již nemá k zadání výhrady, schválí je. Druhý postup je, že si vedoucí ústavu vytiskne všechna zadání a naskenuje do systému ta, která jsou určena ke schválení. Zbýlá zadání pak postupně schvaluje po zapracování připomínek.

Po schválení zadání, typicky do 1. listopadu, dojde ke zveřejnění zadání studentům, kteří si je mohou ve formátu *PDF* stáhnout pro vložení do technické zprávy.

V lednu studenti magisterského studia odevzdávají technickou zprávu, která by měla mít rozsah přibližně poloviny diplomové práce. Následně probíhají obhajoby dosavadní práce v rámci předmětu Semestrální projekt, které se již týkají všech studentů. Délka obhajoby semestrálního projektu je přibližně 5 minut, poté následuje cca 5 minut diskuse. Student musí být přiřazen ke komisi, u které bude práci obhajovat. Zápočet za předmět ITT/SEP uděluje vedoucí na základě doporučení stanovené obhajovací komise, přičemž studentovi

současně k zadání doplňuje doporučení, která vyplynula z obhajoby. Při případném neúspěchu u obhajoby si student zruší navazující předmět IBT/DIP a odhlásí se ze státní závěrečné zkoušky. Podrobnosti o uložených datech ke komisím jsou popsány v kapitole 4.7.

V květnu probíhá odevzdávání finální verze práce. Nejprve se tato práce odevzdává v elektronické podobě v IS FIT. Student nahraje dokumentaci ve formátu PDF a zdrojové kódy. Společně s nimi se navíc do systému ukládá abstrakt a klíčová slova česky i anglicky. Tyto položky se následně zobrazují na webu po zveřejnění práce. V neposlední řadě se také ukládá adresa URL na cloudové úložiště, kam studenti vkládají přílohy k práci, které se taktéž mj. vkládají i na paměťové médium přiložené k vytištěnému textu. Dále má student, kterému byl povolen odklad zveřejnění (viz níže), možnost zadat počet let, po kterém dojde ke zveřejnění práce. Následně se odevzdává vytištěná práce. K identifikaci práce slouží čárový kód, který se nachází v zadání, ten je naskenován a následně se kontrolují náležitosti listinné podoby práce včetně toho, zda student přiložil paměťové médium. Pokud jsou všechny náležitosti splněny a potvrzeny, udělí se studentovi zápočet za předmět IBT/DIP. Pokud z jakéhokoli důvodu nebude studentovi udělen zápočet, musí se student do týdne odhlásit ze SZZ, aby nedošlo ke ztrátě řádného termínu.

Dle novely VŠ zákona 137/2016 Sb. musí být text práce zveřejněn na webových stránkách univerzity [1]. Zveřejnění lze ale odložit až o tři roky z důvodu ochrany duševního vlastnictví nebo obchodního tajemství, přičemž žádost musí být podána v elektronickém studijním oddělení přibližně měsíc před odevzdáním práce. K žádosti se vyjadřují všichni účastníci řízení dle zadání práce (vedoucí, vedoucí ústavu, zástupce dotčené firmy apod.) a následně o odkladu rozhoduje příslušný proděkan. V případě, že žádost schválí, vyplní také v systému důvod odkladu česky a anglicky, čímž studentovi umožní nastavení počtu let, po kterém dojde ke zveřejnění.

Student také může zažádat o odklad odevzdání práce o týden nebo na konec července. V tomto případě na základě rozhodnutí posune studijní oddělení termín pro odevzdání práce. Pokud student odkládá odevzdání na konec července, samotné SZZ se posouvají na srpen.

Nejpozději týden před konáním státních závěrečných zkoušek vedoucí a oponent práce zadávají jejich hodnocení v IS FIT. Tyto posudky jsou převedeny do PDF, které si student následně zobrazí v systému. Na základě posudků a otázek od oponenta se připravuje k obhajobě práce. V červnu probíhá samotná SZZ, pro kterou jsou jako v případě semestrálního projektu vytvořeny komise. Ke komisím se v systému ukládají jednotliví členové včetně jejich funkce – předseda, místopředseda, člen, tajemník nebo jejich náhradníci pro případ, že některý z členů komise nebude přítomen. Dále se ke komisi přiřazuje seznam studentů, ke kterým se ukládá například pořadí a informace, zda budou vykonávat celou SZZ, nebo pouze jednu z částí, pokud opakují.

SZZ se skládá ze dvou částí. Nejprve probíhá obhajoba závěrečné práce, která trvá buď 5 minut pro bakalářský program, nebo 8 minut pro magisterský program či Erasmus, a poté následuje diskuse. Po ní se přechází k ústní části státní závěrečné zkoušky. Při případném neúspěchu u SZZ již student neopakuje předměty ITT/SEP a IBT/DIP, ale pouze se pro formální evidenci odevzdání práce přihlašuje předmět IBTo/DIPo za 0 kreditů.

Výsledky SZZ se následně zadávají do systému. Každá část zkoušky se hodnotí zvlášť, přičemž systém nabízí předvyplněné texty v závislosti na udělené známce. V případě bakalářského studia obdrží student hodnocení za ústní zkoušku (jeden tématický okruh) a obhajobu. Studenti magisterského studia jsou zkoušeni ze dvou okruhů (každý okruh je hodnocen opět zvlášť). Otázky k obhajobě jsou předvyplněny podle posudku oponenta a případné další otázky komise mohou být doplněny u komise tajemníkem. Pro celkový průběh SZZ se

opět předvyplňují texty v závislosti na pozitivním nebo negativním hodnocení. Dále může komise zaznamenat návrh na ocenění. V případě neúspěchu v některé části zkoušky musejí být uvedeny podmínky opakování, a také zda-li bude student opakovat obě části, pouze jednu z částí, nebo, že nemůže opakovat, protože se jedná o druhý pokus.

Kapitola 3

Použité technologie

Tato kapitola popisuje zavedené technologie a praktiky v centrálním systému VUT a jakým způsobem se nastavují a používají přístupová práva a parametrizace. Na tuto kapitolu navazují návrh a implementace. IS VUT využívá dvě nezávislé aplikace – webový IS a desktopového klienta Apollo, jejichž části potřebné pro tuto práci budou blíže popsány.

3.1 IS VUT Apollo

Apollo je desktopová aplikace napsaná pomocí technologie Delphi. Obsahuje moduly pro všechny agendy, které se na VUT provozují, například agendy spojené s výukou, závěrečnými pracemi, ale i administrátorské nástroje pro přidělování práv, nebo také dokumentaci všech databázových struktur.

Jedním z nástrojů používaných pro tuto práci je tzv. parametrizace. Pomocí ní se spouštějí moduly IS VUT a také nastavují různé parametry/atributy, které upravují jejich chování. Pro každý modul musí existovat minimálně jedna skupina parametrů nazývaná též jako maska. Z těchto skupin se vytvářejí instance, z nichž každá se obvykle vztahuje k určité fakultě. Tímto způsobem je možné spouštět moduly každé fakultě zvlášť s rozdílným nastavením. Skupiny parametrů pro StudIS mají obvykle více instancí pro jednu fakultu. Je to z toho důvodu, aby bylo možné spouštět moduly jenom pro určité studijní programy nebo mít u programů rozdílné nastavení.

Dalším z nástrojů, které Apollo poskytuje a které bude potřeba pro tento projekt používat, je přidělování práv. Práva se vztahují k jednotlivým modulům a popisují, jaké operace může přihlášený uživatel provádět. Dále jsou práva omezena rolemi. Role říká, s jakými daty může uživatel v rámci dané součásti VUT pracovat. Tedy například lze právo omezit pouze na konkrétní ústav. S právy je nutné zmínit také profese. Profese je skupina práv a souvisejících rolí, která se přiřazuje uživatelům v závislosti na pracovní pozici nebo vykonávané činnosti.

3.2 IS VUT na webu

IS VUT má také svoji webovou podobu, která již neobsahuje takové množství modulů jako Apollo a má dvě části zaměřené především na studenty a učitele. Aplikace StudIS umožňuje studentům provádět operace spojené se studiem jako například registraci předmětů, projektů a jiných termínů. Učitelé skrze Teacher mohou spravovat předměty, které vedou, vypisovat termíny registrací k projektům a jiné informace pro studenty.

Pro webovou část IS již bylo vytvořeno velké množství komponent, které se používají napříč všemi aplikacemi. Jedná se například o komponentu pro jednoduché vytváření a obsluhu formulářů nebo tabulku `Datagrid`, která umožňuje dynamické vykreslování informací z databáze pomocí technologie `AJAX`. Dále obsahuje také funkce pro získání hodnot parametrů z parametrizace a ověření práv.

Všechny aplikace jsou napsané programovacím jazykem `PHP` verze 7.2 [5]. Zatímco `Teacher` je postaven na frameworku `Zend` verze 1 [8] využívajícím architektury `MVC`, `StudIS` je implementován funkcionálním `PHP`. Jak `Apollo`, tak i webový `IS` jsou postaveny nad databází `Oracle 19c` [6].

Kapitola 4

Ukládání dat v centrální databázi

Tato kapitola popisuje způsob ukládání informací v centrálním systému VUT, jako jsou předměty, studijní programy, informace o studentech přihlášených k projektům. Dále pak informace spojené se závěrečnými pracemi, tedy zadáními a způsoby registrace studentů, komisemi pro obhajoby a v neposlední řadě s hodnocením státních závěrečných zkoušek.

Podle filosofie centrálního IS VUT musí každá tabulka obsahovat minimálně pět atributů (kromě primárního klíče) – ID uživatele, který vložil záznam, datum, kdy byl záznam vložen, ID uživatele, který záznam naposledy upravil, a datum této úpravy. Poslední je status, jehož číselná hodnota označuje stav záznamu (**9** – platný, **-1** – neplatný/smazaný). Tyto atributy nebudou z úsporných důvodů zobrazovány v žádném z následujících ER diagramů.

4.1 Způsob ukládání předmětů

Informace o předmětech budou využívány mimo jiné v modulech pro projektové praxe. Předměty se ukládají především do dvou tabulek *st01.predmet* a *st01.aktualni_predmet*. První z tabulek obsahuje základní informace o předmětu jako například jeho název, zkratku, ID fakulty, která jej vyučuje, a podobně. Aby ale bylo možné zjistit, na které akademické roky byl předmět vypsán, je nutné vytvářet „instance“ tohoto předmětu s informacemi pro aktuální akademický rok – záznamy v tabulce *st01.aktualni_predmet*. Zde je uloženo, který rok byl předmět vypsán, který semestr, ID garanta, zkratka, nebo-li atributy, které se mohou každý rok měnit. Tyto záznamy se poté využívají také v elektronickém indexu studenta (zaregistrované předměty). Kvůli modulům spojeným s projektovými praxemi (viz kapitola 5) byl navíc vytvořen nový číselník *st01.ucel_predmetu*. Na jeho záznamy se odkazuje z tabulky *st01.predmet* a slouží k rozřazení předmětů do různých skupin jako například právě projektové praxe, závěrečné práce atp. Díky této informaci bude jednodušší dotazovat se v modulu na správné předměty patřící k projektové praxi, aniž by bylo nutné pracovat se seznamem zkratek či ID.

U projektových praxí je také důležité zjišťovat navazující předměty, pokud bude studentovi umožněno pokračovat na projektu v dalším semestru. K tomu slouží spojovací tabulka *st01.prerekvizita* s atributy *predmet_id_1* – předmět, který má prerekvizitu, a *predmet_id_2* – předmět, který je prerekvizitou. Prerekvizity mohou být i hodnocené, tedy student musí splňovat minimální hodnocení, aby si mohl zapsat navazující předmět.

4.2 Způsob ukládání informací spojených se studiem

Na rozdíl od WIS FIT se informace o studijních programech v IS VUT duplikují pro každý rok, tedy každý rok má stejný program jiné ID. Při vytváření nového záznamu se ovšem ukládá ID loňského programu, ze kterého se kopie vytváří. Tato skutečnost je zde zmíněna, protože bude nutné zjišťovat předměty projektové praxe (z tabulky *st01.aktualni_predmet*) vypsané pro následující rok pro stejný program. Detailnější vysvětlení této problematiky je v kapitole 5.3.

Společně s přihláškou u projektové praxe bude také nutné ukládat, ke kterému studiu se vztahuje (viz kapitola 5.1), aby mohla aplikace zobrazit požadované informace o studentech. Jedná se o tabulku *st01.studium*. Ta obsahuje například předpočítaný vážený průměr, zda-li je studium aktivní, nebo také cizí klíč na studijní program, který aktuální akademický rok studuje.

4.3 Limity vedených prací

Na fakultě je dle směrnice č. 69/2017 o standardech studijních programů zapotřebí kontrolovat, kolik prací může každý vedoucí daný akademický rok vést [7]. Z tohoto důvodu byl vytvořen nový modul, pomocí kterého je možné tyto limity nastavovat. Ačkoliv modul není součástí zadání této práce, je nezbytné se o něm zmínit, protože na něj tato práce navazuje. Ukázka tohoto modulu je na obrázku 4.1.

Pro modul byla vytvořena i databázová tabulka, která ukládá, kolik bakalářských, diplomových a disertačních prací mohou jednotliví vyučující vést každý rok. Dále ukládá i maximální počet všech vedených prací pro danou osobu za rok. Kromě této tabulky byl vytvořen i materializovaný pohled *st01.v_zav_prace_osoba_pocet_praci*. Do něj se pomocí triggeru při každé změně stavu přihlášky dopočítává, kolik prací každého typu již osoba vede, a kolik jich ještě zbývá. Díky těmto předpočítaným hodnotám se zjednoduší kontrola limitů vedených prací v modulech pro výpis a editaci zadání (viz kapitola 6.3).

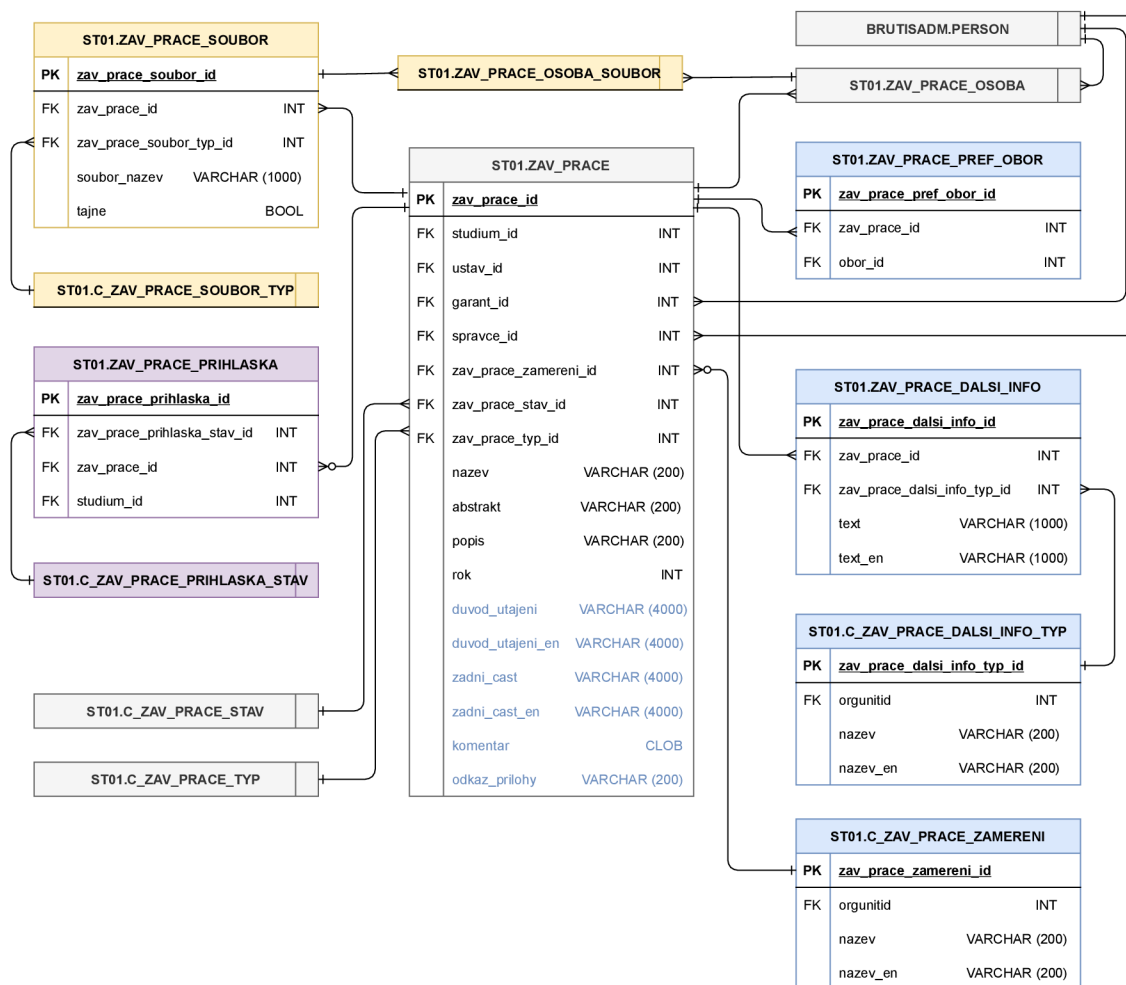
NASTAVENÍ LIMITŮ UIFS - ÚSTAV INFORMAČNÍCH SYSTÉMŮ

Osoba	Zařazení	Vedeno (vypsáno)				Limit			
		BAK	DIP	DIZ	Celkem	BAK	DIP	DIZ	Celkem
	Doktorand (K - 2. rok)	0 (0)	0 (0)	0 (0)	0 (0)	0	0	0	0
	Akademický pracovník	0 (0)	0 (1)	0 (0)	0 (1)	25	15	7	20
	Doktorand (P - 2. rok)	0 (0)	0 (0)	0 (0)	0 (0)	4	2	0	6
	Doktorand (P - 4. rok)	0 (0)	0 (0)	0 (0)	0 (0)	6	4	0	10

Obrázek 4.1: Ukázka nového modulu pro nastavování limitů jednotlivým vyučujícím.

4.4 Informace o závěrečných pracích

Způsob ukládání závěrečných prací je zobrazen na ER diagramu na obrázku 4.2. Z úsporných důvodů jsou zobrazeny pouze atributy nutné pro pochopení základního konceptu ukládání závěrečných prací. Dále jsou barevně odlišeny tabulky, které spolu logicky souvisejí.



Obrázek 4.2: ER diagram závěrečných prací a s nimi souvisejících informací o přihláškách studentů, uložených souborech a preferenčních oborech. Všechny tabulky ve skutečnosti obsahují mnohem více atributů, ale z úsporných důvodů jsou zobrazeny pouze ty důležité.

Samotná zadání závěrečných prací se ukládají do tabulky *st01.zav_prace*. K zadáním se kromě textových atributů jako název práce, abstrakt a znění zadání ukládají také například informace o ústavu obhajoby a osobách, které mohou se zadáním manipulovat. Jedná se o *garanta* a *správce údajů*. Vedoucí a oponent se dříve ukládali také přímo k zadání, později ale bylo zapotřebí na některých fakultách ukládat více oponentů, a proto se postupně přešlo k používání tabulky *st01.zav_prace_osoba*.

Se zadáními také úzce souvisí číselník *st01.c_zav_prace_typ*, ve kterém jsou uloženy všechny možné typy závěrečných prací, jako například *bakalářská práce*, *diplomová práce*, atp. Dalším důležitým číselníkem je *st01.c_zav_prace_stav* ukládající všechny stavy, kterých může zadání nabývat – *volné*, *přiděleno*, *odevzdáno*, *ohodnoceno*, *obhájeno*, *nevypracováno*, *neobhájeno*. Stav *volné* znamená, že práci nemá žádný student přidělenou. Na IS VUT je možné na jednu závěrečnou vytvářet přihlášky více studentům. Takovíto studenti jsou označováni jako *přihlášení*. Pakliže si daný vedoucí vybere jednoho z těchto studentů, mění se stav práce na *přiděleno*. Následně je možné práci vytisknout, a tím následně zakázat další úpravu zadání.

K závěrečné práci se ukládají také různé soubory, ať už se jedná o samotný elektronicky odevzdávaný text studenta nebo o posudky vedoucího či oponenta. Informace o těchto souborech jsou uloženy v tabulce *st01.c_zav_prace_soubor*, kde se kromě jeho názvu ukládá např. to, zda-li je soubor utajený (viz kapitola 2.2). K souboru se také musí ukládat jeho typ z číselníku *st01.c_zav_prace_soubor_typ* – např. *text práce*, *příloha* nebo *posudek*. Pokud se jedná o posudek je navíc do spojovací tabulky *st01.c_zav_prace_osoba_soubor* uloženo, která osoba soubor vložila.

Veškeré nově přidané atributy a tabulky související s přechodem fakulty jsou v diagramu vyznačeny modrou barvou. Jejich podrobnější popis se nachází v kapitole 6.3.

4.5 Přihlášky studentů k závěrečným pracím

Aktuálně se v IS VUT na zadání závěrečných prací přihlašují studenti podobně jako je tomu u projektové praxe (viz kapitola 2.1). Vedoucí může zaregistrovat více studentů (případně se mohou studenti zaregistrovat sami) a následně vedoucí vybere jednoho ze studentů, kterému práci přidělí. Je také možné přihlášku studenta zamítnout. Navíc musí být definován maximální možný počet přihlášek. Ukázka přihlašování je na obrázku 4.3. Datové struktury spojené s přihláškami jsou naznačeny v ER diagramu na obrázku 4.2. V případě, že je zadání přiděleno nějakému studentovi, změní se nejenom stav dané přihlášky, ale zároveň se uloží ID studia do tabulky *st01.zav_prace* k odpovídajícímu záznamu.

Maximálně přihlášek: 1

Přiděleno studentům: Zadání dosud nebylo přiděleno žádnému studentovi

Registrovaní studenti: 1. [blurred name] [trash] [plus] [refresh]

Přidělit studentovi

Zamítnuto studentům: Zadání dosud nebylo zamítnuto žádnému studentovi

Obrázek 4.3: Ukázka aktuálního řešení přihlášek studentů na zadání závěrečných prací. Registrované studenty je možné přidělovat k zadání případně zamítnout. Dále je možné například ke každé přihlášce napsat poznámku.

4.6 Hodnocení závěrečných prací

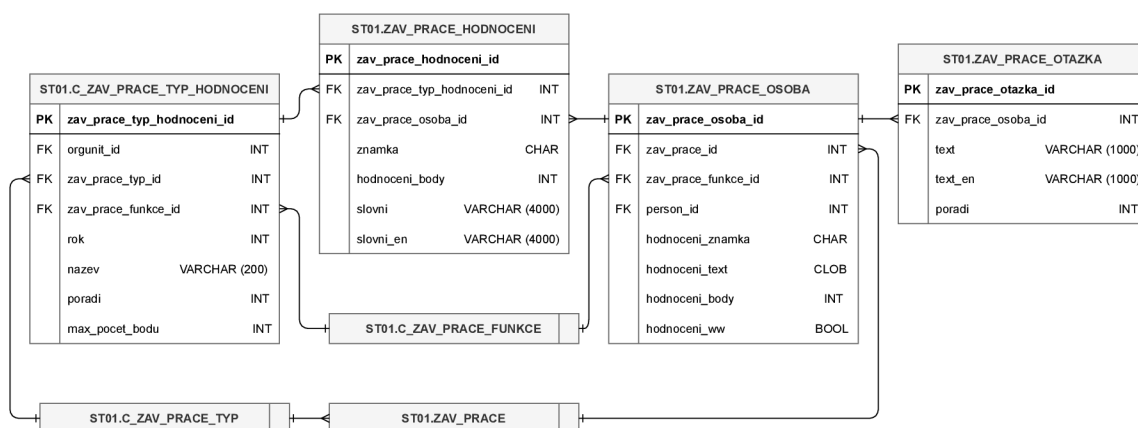
Součástí závěrečných prací je samozřejmě i jejich hodnocení, které vedoucí a oponent před obhajobou zadávají do systému. Pro tyto účely již ve webovém IS existuje modul využívaný některými součástmi VUT.

Každá součást má různá kritéria hodnocení, a proto je zapotřebí je od sebe navzájem rozlišovat. K tomu slouží číselník *st01.c_zav_prace_typ_hodnoceni* (viz obrázek 4.4). Ke každému kritériu se zaznamenává jeho název, pořadí, ve kterém se zobrazuje na webové stránce nebo v souboru daného posudku ve formátu PDF, a také maximální počet bodů, které mohou být uděleny. Jednotlivá kritéria jsou přiřazena nějaké součásti, přičemž se ještě navíc rozlišují podle funkce hodnotící osoby (vedoucí/oponent) a typu závěrečné práce.

Jinými slovy, vedoucí na práci hodnotí jiné atributy než oponenti – například vedoucí na rozdíl od oponenta hodnotí spolupráci s daným studentem.

Konkrétní hodnocení se zaznamenávají do tabulky *st01.c_zav_prace_hodnoceni*. Ke každé osobě je tedy v rámci jedné práce vytvořeno několik hodnocení v závislosti na počtu hodnocených kritérií práce. Daná část práce může být ohodnocena známkou, body nebo slovním popisem. Celkové hodnocení práce je ukládáno přímo k hodnotící osobě do tabulky *st01.c_zav_prace_osoba*.

Osoby hodnotící práci mohou také položit otázky, na které student v rámci obhajoby odpovídá. Otázky se ukládají do tabulky *st01.c_zav_prace_otazka*. Na Fakultě informačních technologií pokládají otázky pouze oponenti.

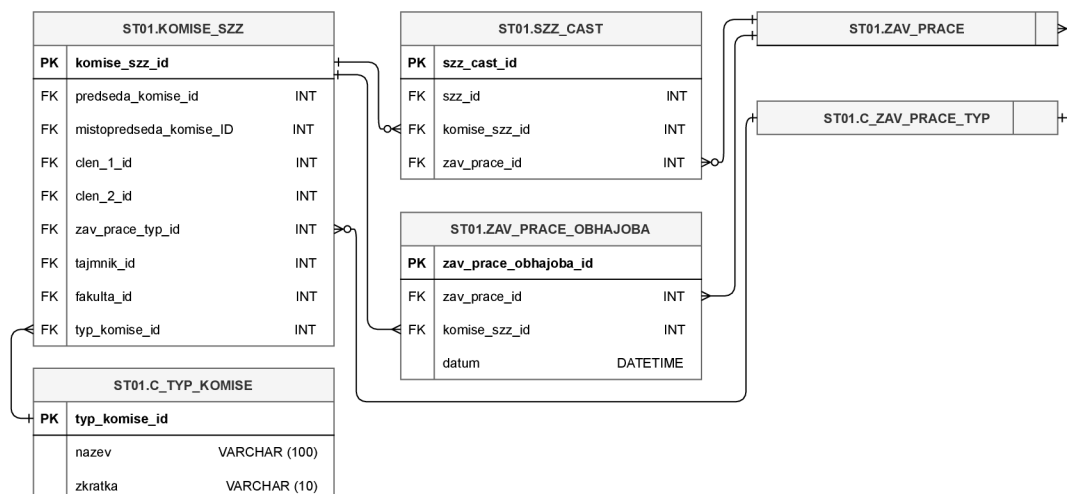


Obrázek 4.4: ER diagram znázorňující ukládání hodnocení závěrečných prací. Pro potřeby vysvětlení konceptu byly vybrány jen ty nejdůležitější atributy.

4.7 Ukládání komisí obhajob

Na centrálním systému se všechny komise ukládají do jedné tabulky *st01.komise_szz* (viz obrázek 4.5). Opět kvůli úspoře místa nejsou v ER diagramu znázorněny veškeré atributy. K samotné komisi jsou pomocí cizích klíčů přiřazeni jednotliví členové (až 12 členů komise). Stejným způsobem se ukládá předseda, místopředseda i tajemník. Další informace v tabulce jsou například ID fakulty, ID místnosti konání, nebo datum a čas začátku obhajob.

Na základě číselníku *st01.c_typ_komise* se poté rozlišuje, o jaký typ komise se jedná (například *bakalářská práce*, *diplomová práce* nebo také *semestrální práce*). Z toho také vyplývá potřeba rozlišovat různé typy obhajob studentů. Pokud se jedná o obhajobu v rámci SZZ, ukládají se veškeré informace o konkrétní obhajobě do tabulky *st01.szz_cast*. Podrobnější informace o SZZ jsou popsány v kapitole 4.9. Druhou možností je obhajoba v rámci semestrální práce, kterou v centrálním systému doposud využívala Fakulta elektrotechniky a komunikačních technologií (ale pouze v rámci IS Apollo). Informace s ní spojené jsou uloženy v tabulce *st01.zav_prace_obhajoba*. Ta na rozdíl od předchozí tabulky obsahuje pouze cizí klíče na závěrečnou práci a samotnou komisi. Přes obě tyto tabulky lze samozřejmě zjistit i to, který student u dané komise obhajuje, přes tabulku *st01.zav_prace*, kde je uloženo ID studia.



Obrázek 4.5: ER diagram znázorňující ukládání komisí v centrálním systému VUT. Opět jsou z úsporných důvodů uvedeny pouze ty nejdůležitější atributy.

4.8 Ukládání témat zkoušených u SZZ

Před samotnými státními závěrečnými zkouškami musí také proběhnout výběr okruhů, které budou daný rok zkoušeny při ústní části zkoušky. Pro fakultu informačních technologií byl vytvořen nový modul ve webovém IS umožňující garantům předmětů spravovat okruhy, které navrhují v rámci daného předmětu. Okruhy se vypisují pouze pro povinné předměty. Z těchto navržených okruhů následně garant programu či specializace vybírá ty, které budou daný akademický rok zkoušeny. Pro navržené okruhy slouží tabulka *st01.tema_k_szz*. Ke každému okruhu se ukládá znění otázky česky i anglicky a jeho relativní pořadí v rámci daného předmětu.

Výběr z vypsání okruhů provádí garanti programů či specializací v modulu informačního systému Apollo, kde však zatím nelze jednoduše vybírat z navržených okruhů z nového modulu vytvořeného pro FIT. Tyto okruhy se následně ukládají do tabulky *st01.szz_tematicka_oblast*. Na některých fakultách mají studenti možnost vybírat si předměty (a tím i dané tématické okruhy), ze kterých budou skládat zkoušku. Pro tyto účely existují další datové struktury, které ale nebudou dále popisovány, protože studenti na Fakultě informačních technologií dostávají neměnný seznam zkoušených okruhů pro daný akademický rok. To, ze kterého tématického okruhu byl student zkoušen, je zaznamenáno pomocí cizího klíče v rámci dané části SZZ (viz kapitola 4.9).

4.9 Ukládání státních závěrečných zkoušek

Průběhy samotných státních závěrečných zkoušek se ukládají zejména do dvou tabulek. První z nich, *st01.szz* zaznamenává celkové průběhy zkoušek. Nese v sobě informace o zkoušeném studentovi (cizí klíč na tabulku *st01.studium*), datu konání zkoušky, celkovém výsledku SZZ (známka a/nebo slovní hodnocení), slovním návrhu na ocenění a v neposlední řadě o celkovém hodnocení studia (známka a výsledný vážený průměr).

Druhou důležitou tabulkou je *st01.szz_cast*. Jak již název napovídá, ukládá v sobě jednotlivé části zkoušky. Může se jednat o ústní zkoušku nebo obhajobu závěrečné práce, přičemž se tyto typy částí ukládají do samostatného číselníku. Každá z částí je přes cizí

klíč propojena se samotným záznamem o SZZ, dále pak s komisí a případně se závěrečnou prací¹ (viz obrázek 4.5). Každá část může být hodnocena známkou a/nebo slovním popisem. Zaznamenává také, o kolikátý pokus se jedná, a pro případ, kdy by student v dané části neuspěl, také slovní popis podmínek reprobace.

¹Cizí klíč pro závěrečnou práci je nepovinný a je vyplněný pouze v případě, že se jedná o záznam typu *obhajoba závěrečné práce*.

Kapitola 5

Návrh modulů k projektovým praxím

V této kapitole bude popsán návrh funkcionality a uživatelského rozhraní nových modulů pro projektové praxe v aplikacích **Teacher** a **StudIS**, včetně databáze, práv a parametrizace. Protože žádná jiná součást VUT nemá podobný typ výběrových předmětů, je nutné vytvořit nové moduly, které by vyhovovaly požadavkům FIT. Návrh řešení se bude opírat o již existující moduly v informačním systému FIT.

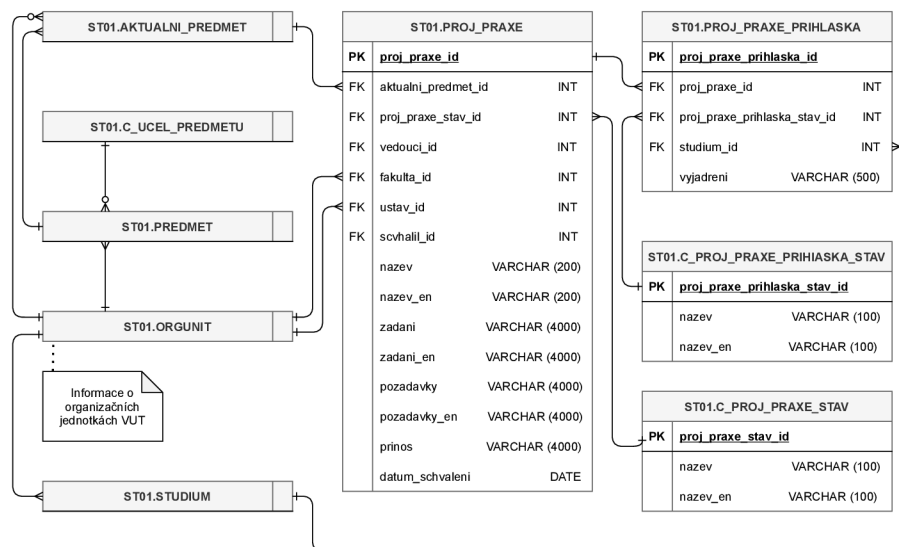
5.1 Návrh databáze

ER diagram zobrazující tabulky používané pro nový modul je na obrázku 5.1. Samotné projekty se budou ukládat do tabulky *st01.proj_praxe*. To, pro který ústav se zadání vypisuje, bude uloženo v cizím klíči *ustav_id*, který odkazuje do již existující tabulky centrálního systému, obsahující všechny organizační jednotky na VUT. To stejné platí i pro atribut *fakulta_id*. Ačkoliv je možné pomocí ID ústavu získat ID nadřazené jednotky, na CVIS se podobné redundance často zavádějí z důvodu urychlení SQL dotazů. Atributy *vedouci_id* a *schvalil_id* jsou unikátní číselné identifikátory osob v rámci VUT (*PER ID*) označující vedoucího (školitele) a osobu, která schválila zadání. Pro který předmět se bude zadání vypisovat, se bude ukládat pomocí cizího klíče *aktualni_predmet_id*. Jak se ukládají předměty v centrálním systému je popsáno v kapitole 4.1. K projektu se navíc ukládá několik textových položek – název (česky a anglicky), popis zadání (česky a anglicky), požadavky na studenta (česky a anglicky) a jaký přínos bude tento projekt pro fakultu mít, podle čehož se rozhoduje, zda-li bude schválen nebo zamítnut.

Zadání může nabývat tří stavů:

- **příprava** – zadání je nově vypsáno a čeká se na rozhodnutí o schválení nebo zamítnutí (podle parametrizace se nastaví, zda se budou zadání v tomto stavu zobrazovat studentům či nikoli),
- **schváleno** – zadání bylo schváleno a studenti se smí na zadání přihlašovat,
- **zamítnuto** – zadání bylo zamítnuto a nesmí být zobrazeno studentům.

Přihlášky se budou ukládat do zvláštní tabulky *st01.proj_praxe_prihlaska*, čímž se zajistí, že se k jednomu projektu bude moci přihlašovat více studentů. Kromě cizího klíče



Obrázek 5.1: Návrh databázových tabulek pro uložení dat o projektových praxích.

odkazujícího na projekt a textového atributu *vyjadreni* bude tabulka obsahovat ještě cizí klíč na studium (viz kapitola 4.2, kde je popsáno ukládání studií).

Přihláška může nabývat čtyř stavů:

- **přihlášeno** – student je přihlášen na daný projekt (může být přihlášen na více projektů současně),
- **přiděleno** – studentovi je přidělen projekt, jeden projekt může být přidělen jednomu studentovi a zároveň student nemůže mít přiděleno více projektů,
- **zamítnuto** – studentova přihláška byla zamítnuta,
- **odhlášeno** – student se sám odhlásil ze zadání.

Všechny dříve popsané stavy se budou ukládat do dvou číselníků – *st01.proj_praxe_stav* ukládající stavy, kterých může nabývat vypsané zadání, a *st01.proj_praxe_prihlaska_stav* pro stavy přihlášek studentů.

5.2 Návrh práv pro modul v aplikaci Teacher

Proto, aby mohl školitel vypisovat zadání, potřebuje mít přidělené právo. Pokud se k tomuto právu přidá role vedoucího ústavu, může takovýto uživatel vytvářet nová zadání a přiřazovat je jakémukoliv školiteli v rámci jeho ústavu. Tato zadání může i upravovat a schvalovat, přičemž se na něj nevztahují žádná časová omezení jako například časové období, po které je možné vypisovat nová zadání (viz kapitola 2.1).

Správce modulu má potom stejná práva jako vedoucí ústavu s tím, že se vztahují na celou fakultu.

Modul tedy bude využívat tři přístupových práv:

- **Právo pro čtení** – je přiřazeno všem osobám, které mají přístup k modulu a mohou číst všechna zadání.
- **Právo pro zápis** – přiřazuje se školitelům, kterým je dovoleno vytvářet nová zadání. Toto právo je dále omezeno atributy popsanými v kapitole 5.3. Pokud je toto právo zkombinováno s rolí vedoucího ústavu, neplatí pro něj žádná omezení spojená s parametrizací, tedy má stejné možnosti jako správce (viz níže), ovšem omezené pouze na zadání vypsána pro daný ústav.
- **Právo pro správce** – přiřazuje se například studijním poradcům a umožňuje spravovat jakákoliv zadání na fakultě. Jediným omezením je, že nelze zpětně upravovat stará zadání kromě těch, na kterých jsou přihlášení studenti se stále aktivním studiem.

Tato práva se budou nastavovat v systému Apollo. Práva pro čtení a zápis budou hromadně přiřazena pomocí profese *vyučující FIT* a právo správce profesím *děkan, proděkan* a *studijní poradce*. Případně se tato práva dají přidělit individuálně.

5.3 Návrh parametrizace pro nové moduly

Pro modul pro projektové praxe budou zapotřebí tři skupiny parametrů.

Skupina parametrů pro Teacher i skupina parametrů pro StudIS musí obsahovat parametry pro nadpis modulu (česky a anglicky), text v menu (česky a anglicky), datum od-do, kdy bude modul spuštěn, a přepínač pro zobrazování odkazu v menu.

Skupina parametrů pro Teacher kromě výše zmíněných obsahuje navíc parametr, který nastavuje, zda se bude posílat informativní e-mail studentovi o změně stavu jeho přihlášky.

Skupina parametrů pro StudIS bude obsahovat následující parametry:

- OP Fakulta ID
- OP Program ID
- OP Číslo ročníku
- OP Aktivní studium – parametr s pravdivostní hodnotou
- Program ID pro registraci předmětů
- Začátek přihlašování studentů
- Konec přihlašování studentů
- Začátek přijímání studentů
- Konec přijímání studentů
- Začátek vypisování zadání
- Konec vypisování zadání
- Minimální počet kreditů pro přihlášení studenta

- Zobrazovat zadání ve stavu příprava – parametr s pravdivostní hodnotou

Parametry, jejichž název začíná „OP“, se nazývají omezující podmínky. Jak již bylo naznačeno v kapitole 3.1, v aplikaci StudIS se obvykle vytváří více instancí jedné skupiny parametrů, a to proto, aby se modul mohl zobrazovat pouze studentům určitých studijních programů, ročníků, oborů atp. v rámci jedné fakulty. Parametr OP Program ID pouze nastavuje, kterému studijnímu programu se bude modul spouštět aktuální rok. Jak již ale bylo zmíněno v kapitole 2.1, studenti některých programů se mohou přihlašovat na zadání vypsána na následující akademický rok. Proto musí být aplikace schopná zjistit ID předmětů pro ten daný rok (databázová tabulka *st01.aktualni_predmet*). To ale nelze jednoduše zjistit z ID aktuálního programu, protože databázová tabulka pro program neobsahuje cizí klíč na následující studijní program, a nelze tedy jednoduše zjistit ID předmětů vypsanych pro daný program. Z tohoto důvodu byl přidán další parametr *Program ID pro registraci předmětů*.

Ačkoliv jsou parametry *Začátek vypisování zadání* a *Konec vypisování zadání* určeny pro aplikaci Teacher, byly přidány do této skupiny, jelikož termíny výpisu zadání se liší pro různé studijní programy.

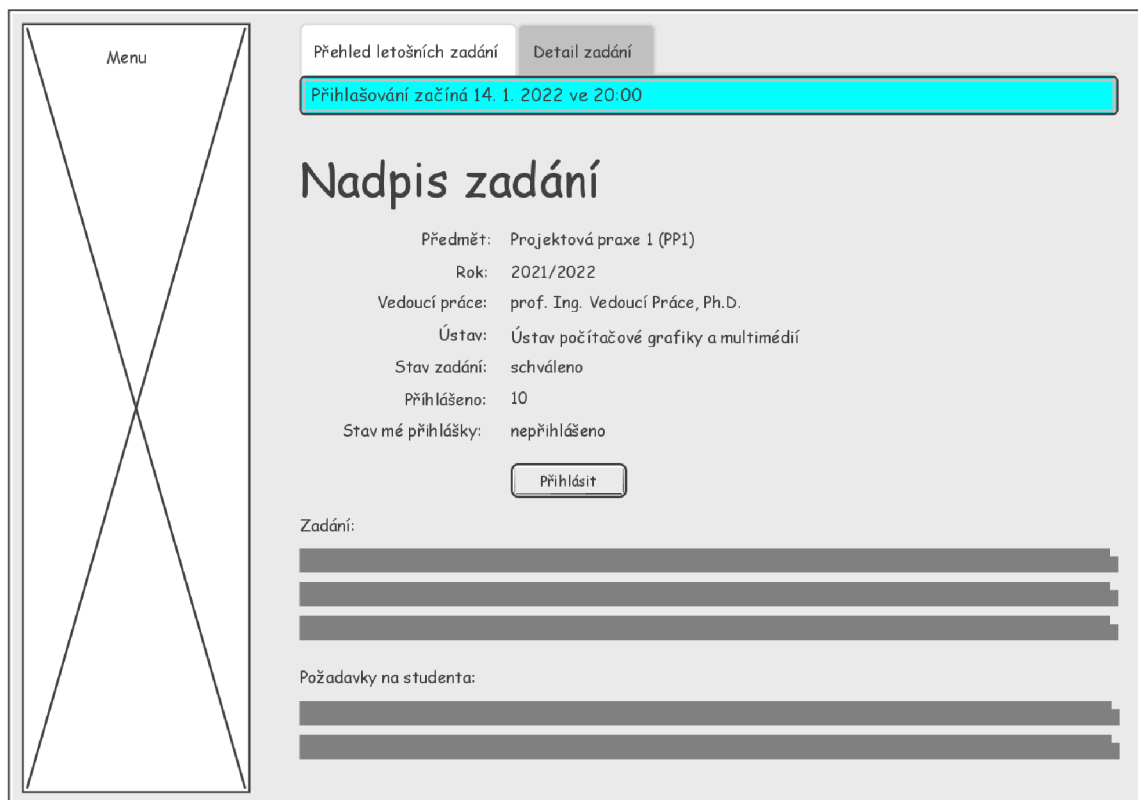
Podmínkou, aby se mohl student přihlásit, je mít splněné povinné předměty z předchozího roku/semestru. Ovšem platí, že se počet povinných předmětů může následující rok změnit. Dále se například pro studenty v magisterském studiu nezapočítávají některé povinné předměty, především TIN a MSP. Proto je nejjednodušší přihlašování nově omezit pouze na počet získaných kreditů.

Pro každý program, ve kterém se studenti budou moci přihlašovat na zadání, bude vytvořena instance této skupiny parametrů. Ovšem pokud by modul viděli pouze studenti, kteří se mohou přihlašovat, nedostali by se k němu studenti, kteří již mají zadání přihlášené z předchozího roku, nebo pouze pokračují v navazujícím předmětu. Proto bude potřeba vytvořit ještě jednu „výchozí“ instanci pro ostatní studijní programy a ročníky, aby i těmto studentům byl umožněn přístup k modulu. V tomto případě bude modul zobrazovat pouze studentem přihlášená zadání (bez možnosti přihlašování či odhlašování).

Třetí skupina parametrů bude společná pro obě aplikace. Bude obsahovat parametr se seznamem ID aktuálních předmětů, které budou viditelné pro studenty v seznamu vypsanych zadání. Zadání se totiž vypisují pouze pro předměty IP1 a PP1 či jejich varianty v angličtině. Ostatní předměty jsou navazující, tudíž student v rámci těchto předmětů pokračuje na stejném projektu. Z tohoto důvodu zadání vypisovaná přímo pro předměty IP2, IP3, PP2 apod. není vhodné zobrazovat všem studentům, jelikož takto vypsána zadání bývají přidělena pouze po udělení výjimky ve studiu. Další parametr je chybová hláška (česky a anglicky), která se bude zobrazovat školitelům, pokud při vypisování zadání zvolí jiný předmět než z předchozího seznamu. Ačkoliv bude tato hláška využita pouze v aplikaci Teacher, rozhodl jsem se ji přidat do této skupiny parametrů kvůli přehlednosti.

5.4 Návrh modulu pro StudIS

Tato část modulu pro projektové praxe bude studentům zobrazovat seznam vypsanych projektů a dále pak jejich detaily. Zadání, která studenti mohou vidět, souvisí s tím, na které předměty se mohou v rámci svého studia přihlašovat (viz kapitola 2.1). Modul bude dále rozlišovat seznam všech vypsanych projektů a seznam přihlášených projektů. Je to z toho důvodu, že student má možnost pracovat na zadání více než v jednom ročníku a tato zadání pak nebudou ve výše zmíněném seznamu viditelná. Zároveň se počítá s tím, že bude



Obrázek 5.2: Detail zadání v aplikaci StudIS.

student přihlášený v rámci jednoho studia na více zadání – například nebyl na některý projekt přijat a přihlásil se tedy na jiný.

U detailu projektu (viz obrázek 5.2) budou vypsány položky jako název projektu, jméno vedoucího, přesné znění zadání, požadavky na studenta, ústav, pro který je zadání vypsáno, počet přihlášených studentů, a pokud bude přihláška přijata či zamítnuta, tak i vyjádření školitele. Součástí detailu bude i formulář obsahující tlačítka pro přihlášení či odhlášení ze zadání. Student se může přihlašovat na více zadání současně. Pokud nebude umožněno přihlašování, budou tato tlačítka skryta.

5.5 Návrh modulu pro Teacher

Zobrazování seznamu projektů (viz obrázek 5.3) bude stejné jako v případě StudISu s tím rozdílem, že se všem, kdo mají k modulu přístup, se budou zobrazovat veškerá zadání vypsaná pro fakultu bez jakéhokoliv dalšího omezení.

Stránka pro vypisování zadání bude obsahovat pouze formulářové prvky. K ní budou mít přístup pouze uživatelé s právem zápisu (viz kapitola 5.2). Prvním důležitým prvkem je školitel (vedoucí) – jeho hodnotu budou moci měnit pouze vedoucí ústavů a správci, jinak jeho výchozí hodnota bude nastavena na aktuálně přihlášeného uživatele. Pokud vedoucí ústavu vypisuje zadání pro školitele z jiného ústavu, zobrazí se ve formuláři upozornění, že toto zadání nebude moci po vypsání upravovat. Toto upozornění se ještě jednou zobrazí v modálním okně po stisknutí tlačítka pro uložení. Název ústavu se ve formuláři bude měnit dynamicky podle vybraného školitele. Dalším důležitým elementem formuláře je výběr



Obrázek 5.3: Seznam vypsaných zadání pro projektové praxe v aplikaci Teacher.

předmětu. Výběr předmětů bude omezen podle parametru *Program ID pro registraci předmětů* (viz kapitola 5.3). Správcům nebo vedoucím ústavů bude umožněno vypisovat zadání pro jakékoliv předměty z aktuálního a následujícího akademického roku kvůli případným výjimkám ve studiu. Zbylé elementy budou pouze editory pro jednotlivé texty k zadání, které jsou ukládány v jazyce HTML.

K detailu již vypsaného zadání bude moci opět přistupovat kdokoli, kdo má alespoň právo pro přístup k modulu. Stejně jako stránka pro výpis zadání, bude i tato stránka obsahovat formulářové prvky. Ty budou nastaveny pouze pro čtení, pokud nebude mít přihlášená osoba možnost zadání upravovat – není školitel, správce nebo vedoucí ústavu nebo se jedná o staré zadání. V tomto případě se skryjí i tlačítka pro ukládání a mazání. Navíc se bude zobrazovat informace o stavu projektu (*příprava*, *schváleno* či *zamítnuto*) včetně data schválení, respektive zamítnutí a jména osoby, která zadání schválila, respektive zamítla. Uživatelům s právem správce se v této části rozhraní zobrazí i tlačítka pro možnost změny stavu (viz obrázek 5.4).

Dále se v detailu zadání bude zobrazovat seznam přihlášených studentů s informacemi o jejich studiích jako jsou login, vážený průměr, obor atp. V případě, že jsou na projekt přihlášení studenti, již nelze upravovat předmět, na který je zadání vypsáno. Bude zde zobrazen také jednoduchý formulář s vyhledávacím elementem, který našeptává jména studentů. Ten se nevykreslí u zadání zamítnutých nebo vypsaných pro předchozí akademické roky a v případě, že přihlášená osoba nemá právo zadání upravovat. Bude se provádět kontrola, zda-li může být student na projekt přihlášen (student má aktivní studium v programu, který odpovídá programu předmětu, a dostatečný počet získaných kreditů). V případě nějakého problému zobrazí aplikace upozornění. Aplikace ovšem dovolí přihlásit i studenta, který nespĺňuje minimální počet kreditů, pokud mu byla udělena výjimka ve studiu – pro tyto situace modul zobrazí modální okno s možnostmi potvrdit nebo zrušit akci.

Ze seznamu studentů lze přistupovat k detailu přihlášky. Kromě informací uvedených v seznamu zde bude navíc informace o předmětech, které má v rámci tohoto projektu student zapsané. Dále je v detailu možno studenta přijmout nebo odmítnout pomocí dvou tlačítek formuláře, přičemž musí být vyplněno textové pole s vyjádřením školitele. V případě, že uživatel nemá dostatečné oprávnění s přihláškou manipulovat, bude vyjádření nastaveno pouze pro čtení. Přihláška může nabývat následujících stavů – *přihlášeno*, *přiděleno*, *zamítnuto*, *odhlášeno*. Modul bude umožňovat i smazání přihlášky z databáze, protože se

může stát, že uživatel omylem přidá nesprávného studenta. Mazání přihlášky bude omezeno pouze na uživatele, který přihlášku do databáze vložil.

The screenshot displays the 'Detail zadání' (Assignment Details) page in the Teacher application. On the left is a vertical menu labeled 'Menu'. At the top, there is a breadcrumb trail with four items: 'Nové zadání', 'Přehled všech zadání', 'Mnou vytvořená zadání', and 'Detail zadání'. The main content area is titled 'Název zadání' (Assignment Name). Below this, the 'Stav zadání' (Assignment Status) is shown as 'příprava' (preparation). The 'Stav změnil(a):' (Status changed by) and 'Datum:' (Date) fields are empty. Two buttons are present: 'Schválit zadání' (Approve assignment) and 'Zamítnout zadání' (Reject assignment). The 'Detail zadání' section contains several form fields: 'Vedoucí:' (Supervisor) with the value 'prof. Ing. Vedoucí Projektů, Ph.D.', 'Ústav:' (Department) with 'Ústav počítačové grafiky a multimédií', 'Předmět:' (Subject) with a dropdown menu showing 'IPI/2022Z - Projektová praxe 1', 'Název tématu:' (Assignment Name) with 'Název zadání', 'Název tématu anglicky:' (Assignment Name in English) with 'Name of assignment', and 'Zadání práce:' (Assignment text) which is currently empty.

Obrázek 5.4: Detail zadání v aplikaci Teacher. Na obrázku je z úsporných důvodů zobrazena jenom část formulářových prvků. Následně obrázek zobrazuje možnost změny stavu zadání pro uživatele s patřičným právem.

Kapitola 6

Návrh modulů k závěrečným pracím

Tato kapitola se zabývá návrhem modulů spojených se závěrečnými pracemi. Pro závěrečné práce bude vytvořen nový modul s přehledem vypsáných závěrečných prací, dále pak bude upraven modul pro výpis a editaci zadání v aplikaci **Teacher** a dále modul pro výpis detailu pro studenty v aplikaci **StudIS** pro potřeby fakulty. Kapitola popisuje také návrh nových práv pro moduly závěrečných prací a kontrolu limitů vedených prací. V době psaní textu se již moduly v aplikaci **Teacher** využívají na ostrém provozu.

6.1 Návrh práv pro moduly závěrečných prací

Pro moduly závěrečných prací musejí být vytvořena nová přístupová práva tak, aby byly splněny požadavky fakulty. Doposud mohli k zadáním přistupovat pouze jejich vedoucí (neboli vyučující, kteří dané zadání vypsali), dále pak osoby označené u zadání jako správce údajů nebo garant. Na Fakultě informačních technologií je ale zapotřebí, aby mohli k libovolným zadáním přistupovat například vedoucí ústavů kvůli schvalování (případně zamítání) zadání nebo také studijní poradci. Tato práva budou využívána jak modulem pro výpis/editaci zadání, tak i novým modulem se seznamem vypsáných zadání (viz kapitola 6.2). Všechna tato práva se přiřazují na základě profesí – tato problematika je vysvětlena v kapitole 3.1.

- **Právo na přístup k seznamu zadání** – umožňuje přístup k novému modulu (viz kapitola 6.2). Na základě profese je přiřazeno například všem vyučujícím, děkanovi, studijním poradcům nebo také tajemníkovi fakulty.
- **Právo na zobrazování všech vypsáných zadání** – omezuje, ke kterým zadáním budou mít uživatelé přístup. Pokud toto právo uživatel nemá přiděleno, bude moci přistupovat pouze ke svým zadáním. Právo je také přiřazeno všem vyučujícím, děkanovi, studijním poradcům i tajemníkovi fakulty. Vyučujícím je právo přiřazeno z toho důvodu, aby se mohli případně inspirovat od zkušenějších.
- **Právo na vypisování zadání** – umožňuje přístup k modulu pro vypisování nových zadání. Pokud má uživatel toto právo, zobrazí se v seznamu vypsáných zadání také odkaz na tento modul (viz obrázek 6.2, karta vlevo). Toto právo je přiřazeno například všem vyučujícím, děkanovi, tajemníkovi fakulty, ale i studijním poradcům.

- **Právo pro možnost editace jakéhokoliv vypsaného zadání** – je přiřazeno profesím jako například studijní poradce, děkan, tajemník fakulty, ale už není přiřazeno uživatelům pouze s profesí vyučujícího.

6.2 Návrh modulu pro přehled závěrečných prací v aplikaci Teacher

Jedná se o zcela nový modul, který umožňuje zobrazit seznam všech zadání na fakultě. Aktuálně se v centrálním systému používá modul zobrazující pouze zadání, u nichž je přihlášený uživatel veden jako vedoucí, případně garant nebo správce údajů (viz obrázek 6.1). Dalo by se sice upravit tento modul zakomponováním nově vytvořených práv, ale nakonec se na CVIS rozhodlo, že bude jednodušší vytvořit nový modul, který se bude postupně nahrazovat u všech fakult. Navíc bude mnohem přehlednější díky tomu, že budou zadání zobrazena v tabulce na rozdíl od stávajícího modulu. Podobný modul má například i IS Apollo, který zobrazuje každému uživateli všechna zadání na fakultě a editaci mohou provádět jen oprávnění uživatelé.

Nový modul bude využívat práva popsaná v kapitole 6.1. Podle nich se bude řídit, jaká zadání přihlášený uživatel uvidí a jaké operace se zobrazí ve sloupci *akce* (viz obrázek 6.2). Bude se jednat o operace jako například stažení zadání ve formátu PDF pro tisk nebo smazání zadání. Aby mohlo být zadání smazáno, musí být splněny určité podmínky – přihlášený uživatel je vedoucí dané práce, atribut *studium_id* v tabulce *st01.zav_prace* je prázdný (práce je ve stavu *volná*) a práce není vytištěna. Do budoucna se ještě zvažuje možnost, že kromě vedoucích budou moci zadání mazat i uživatelé s *právem editace jakéhokoliv vypsaného zadání*. Další operace zobrazená ve sloupci bude například možnost převést práci na následující akademický rok. Tato možnost je tu pro to, aby vedoucí nemusel zdlouhavě vyplňovat zadání znovu na následující akademický rok v případě, že se na něj nepřihlásil žádný student. Modul bude také umožňovat hromadné převádění zadání – vedoucí pomocí zaškrtačacích políček bude moci vybrat práce, které chce vypsát na následující akademický rok. S tím samozřejmě souvisí i nutnost vyhledat v databázi ID oborů vypsaných na následující akademický rok. Podobně jako pro programy, tak i pro obory se vytváří každý rok nová kopie záznamu (bližší popis této problematiky je v kapitole 4.2).

6.3 Návrhy úprav stávajícího modulu pro vypisování a editaci zadání

Modul pro vypisování a editaci zadání je v centrálním systému již dlouhodobě využíván několika fakultami, které mají ovšem rozdílné postupy a/nebo informace, které se k zadáním ukládají. Chování modulu pro jednotlivé fakulty je upravováno parametrizací (viz kapitola 3.1). Pro FIT bude zapotřebí navrhnout nové parametry a zakomponovat nová práva popsaná v této kapitole. K detailu zadání bude navíc přidán odkaz na výsledky SZZ pro jednodušší práci na webu.






Modul je řízen jediným radičem a poskytuje dva základní pohledy – vypisování zadání a editaci zadání. Radič inicializuje dvě zvláštní formulářové komponenty pro každý z těchto pohledů. Jinými slovy pro vypisování zadání je vytvořena jiná formulářová komponenta než pro editaci. Z tohoto důvodu bude nutné provádět změny v obou formulářích současně.

Ak. rok: *	Typ oborů:	
2021/2022	- Vše -	
Student:	Od firmy:	Zobrazení (předběžných) zadání:
Vše	Vše	Jen nezpracovaná
<input type="checkbox"/> Mimo správce		Vyhledat

SEZNAM ZADÁNÍ

Výchozí řazení Přiděleno A-Z Přiděleno Z-A

« 1 » 10 25 50 100 1000

test (volné)	 Editovat	 Tisk	 Kopírovat	 Smazat
Ved. / Spr. / Op.:	/ / 0			
Kategorie / Typ:	Diplomové práce RRTES / diplomová práce			
Registrovaní studenti:	0 / 1			
Ke schválení:				

« 1 » 10 25 50 100 1000

Obrázek 6.1: Modul pro zobrazování seznamu zadání vypsaných přihlášeným uživatelem, který se aktuálně používá na několika fakultách VUT.

Přístupová práva

Jak již bylo zmíněno, na Fakultě informačních technologií má přístup k zadáním větší množství lidí než je tomu na ostatních fakultách. Je to především z toho důvodu, aby mohli vedoucí ústavů snadno schvalovat/zamítat zadání. Dalším důvodem je také například možnost vyučujících inspirovat se od zkušenějších kolegů.

Modul bude využívat *právo na vytváření nových zadání*, kterým bude řízen přístup k pohledu s vypisováním zadání. Další využívané právo bude *právo na zobrazování všech vypsaných zadání*. To bude řídit přístup k pohledu s detailem zadání. Pokud uživatel nebude mít toto právo, bude moci přistupovat pouze k detailům jím vypsaných zadání. *Právo pro možnost editace jakéhokoliv vypsaného zadání* bude řídit formulář pro editování zadání, konkrétně to, jestli budou položky pouze pro čtení, nebo jestli bude moci uživatel položky měnit.

Přímé přihlašování studentů

Další změna oproti stávajícímu řešení spočívá v přímém přidělování zadání studentům. Jak již bylo řečeno v kapitole 4.5, doposud všechny fakulty využívající centrální systém používaly koncept přihlášek k zadáním.

Na Fakultě informačních technologií je ale možné mít přihlášeného pouze jediného studenta s tím, že přihlášení znamená automatické přidělení práce. Jinými slovy odpadá veškerý koncept přihlášek. Přesto ale bude potřeba využívat datové struktury pro přihlášky (viz obrázek 4.2) kvůli dodržení filosofie centrálního systému.

Obrázek 6.2: Návrh vzhledu nového modulu pro přehled vypsaných zadání. Na obrázku je naznačena možnost vyhledávání podle roku, ústavu a zaměření práce. Dále modul zobrazuje informaci pro uživatele o tom, kolik může vést závěrečných prací.

Místo položek zobrazených na obrázku 4.3 zde bude jediné pole pro vyhledávání studentů na fakultě, a tím budou pro vyučující přihlášky odstíněny. Z tohoto důvodu bude muset být v parametrizaci pro závěrečné práce vytvořen nový parametr *přímé přiřazení studenta*. Ten bude jednak řídit, jaké formulářové prvky se budou zobrazovat, a jednak jakým způsobem se bude vytvářet záznam o přihlášce v databázi.

Dále bude potřeba vyřešit odebrání studenta ze zadání. Ačkoliv tento proces není na fakultě dovolen, může nastat situace, kdy vedoucí omylem přiřadí zadání nesprávnému studentovi. Z tohoto důvodu by měl mít systém možnost i tak přihlášku odebrat. Proto před uložením práce bude uživateli zobrazeno modální okno s upozorněním. S tím souvisí výměna studenta. Systém musí aktuálně přidělenou přihlášku nastavit jako zamítnutou a nově přidávanému studentovi vytvořit přihlášku novou. Také se může stát, že zadání bude znovu přiděleno studentovi s existující přihláškou. V tomto případě bude pouze změněn stav přihlášky.

Kontrola limitů

Dalším z požadavků fakulty je kontrolovat u jednotlivých vedoucích počet vedených prací (viz kapitola 4.3). Tento limit nemá vliv na počet vypsaných zadání, který je teoreticky neomezený. To, kolik prací osoba vede daný akademický rok, se odvíjí od počtu přidělených přihlášek studentů. Jak již bylo zmíněno, limity se nastavují zvlášť pro bakalářské, diplomové a disertační práce. Dále má vedoucí omezený ještě celkový počet vedených prací.

Kontrola se bude muset provádět zvlášť v šabloně pro vypisování a editaci zadání. Při vypisování nového zadání má vedoucí možnost rovnou přidělit zadání studentovi, a proto v tomto případě musí systém ještě před uložením zkontrolovat, zda-li vedoucí nepřekračuje

povolený limit, přičemž se musí brát v potaz typ závěrečné práce (BP/DP/DZ). Pokud limit překračuje, systém zobrazí upozornění a nedovolí práci uložit, dokud studenta neodebere. V opačném případě se s vložením zadání vytvoří nová přihláška a automaticky nastaví jako *přidělená*. U editace zadání bude potřeba opět kontrolovat případ, kdy bude chtít vedoucí přiřadit zadání studentovi. Pokud je již nějakému studentovi zadání přiřazeno, není možné zpětně měnit typ závěrečné práce, proto kontrola limitů spojená se změnou typu není nutná.

Úprava položek formulářů

Součástí požadavků fakulty jsou i některé změny ve formulářových prvcích pro vypisování a editaci zadání. S každou zmíněnou položkou bude navíc vytvořen i nový parametr v parametrizaci, který bude řídit její zobrazování respektive skrývání.

Jedná se o tyto položky:

- **Garant** – je položka zobrazovaná na všech součástech VUT, které využívají modul závěrečných prací. Pro FIT ale bude skryta.
- **Charakteristika problematiky úkolu** – je doplňující textová informace k zadání, která bude pro FIT také skryta.
- **Komentář k zadání** – je nově přidaná položka pro FIT. Obsahuje textovou informaci, která se bude zobrazovat studentům v aplikaci **StudIS**. Tento text se netiskne na zadání.
- **Body k prvnímu semestru** – souvisí se semestrálními projekty (viz kapitola 7). Bude se jednat o dvě položky (v češtině a angličtině), do kterých vedoucí vkládají informaci o tom, které z bodů zadání závěrečné práce musejí být splněné pro obhajobu semestrálního projektu. Budou se ukládat přímo k zadání do tabulky *st01.zav_prace*.
- **Zaměření práce** – popisuje konkrétní odvětví, kterého se závěrečná práce týká. Jedná se například o *bioinformatiku*, *databáze*, *počítačovou grafiku* nebo *operační systémy*. Tyto hodnoty budou uloženy v novém číselníku *st01.c_zav_prace_zamereni*, který je znázorněn v ER diagramu na obrázku 4.2.
- **Preferenční obor** – je způsob, jak práci doporučit studentům některých oborů (viz kapitola 6.4), přičemž každé zadání může být přiřazeno více preferenčním oborům. Tato položka ale nijak neomezuje to, kteří studenti se mohou na zadání přihlašovat na rozdíl od některých fakult, kde k práci je uložena informace o oboru, který určuje, jakým studentům může být zadání zapsáno. Proto, aby se tyto dvě informace nepletly dohromady, bude zapotřebí vytvořit pro preferenční obory novou tabulku *st01._zav_prace_pref_obor*, která je zobrazena v ER diagramu na obrázku 4.2. Preferenční obor bude obsahovat cizí klíč na studijní obor.
- **Doplňující informace** – zahrnuje nově přidanou skupinu položek pro FIT, která bude navržena tak, aby jejich počet mohl být proměnlivý pro každou součást VUT. Pro tyto položky bude vytvořen nový číselník *st01.c_zav_prace_dalsi_info_typ*, pomocí kterého se bude určovat, jaké položky se budou zobrazovat součástí dané pomocí atributu *ID součásti VUT*. Do tabulky *st01.zav_prace_dalsi_info* se budou ukládat textové hodnoty těchto položek. Obě tyto databázové tabulky jsou také zobrazeny v ER diagramu na obrázku 4.2.

- **Důvod utajení** – je dvojice textových položek (česky a anglicky) přidaných pro FIT. Aktuálně je na centrálním systému možné zadat počet let utajení odevzdané práce, ale už nikoliv text odůvodnění, který se na fakultě používá. Text vyplňuje proděkan pro studium.
- **Semestrální část práce** – odkaz na vložený soubor obsahující technickou zprávu pro semestrální projekt v rámci magisterského studia. V aplikaci StudIS bude existovat odpovídající položka s možností soubor nahrávat. Pro tyto soubory se bude využívat tabulka *st01.zav_prace_soubor*, přičemž bude potřeba ukládat informaci o tom, že se jedná o jiný typ práce pomocí číselníku (viz kapitola 4.4).
- **Odkaz na datové úložiště** – je textová položka obsahující odkaz na fakultní cloudové úložiště, kam studenti budou vkládat přes aplikaci StudIS zdrojové soubory a další přílohy k závěrečné práci. Z pohledu aplikace Teacher bude needitovatelná.

6.4 Návrhy změn modulů v aplikaci StudIS

V aplikaci StudIS již existují dva moduly pro závěrečné práce. První z nich, „Registrace závěrečné práce“ slouží k tomu, čemu jeho název napovídá. Zobrazuje seznam vypsaných závěrečných prací, ke kterým se student může přihlašovat. Dále je možné zobrazit i detail zadání. Pro FIT bude zapotřebí upravit řazení zobrazovaných prací v seznamu, které bude nově záviset na preferenčním oboru (viz kapitola 6.3). Práce s preferenčním oborem odpovídajícím oboru studenta budou řazeny na začátek seznamu. Další změnou bude možnost filtrování prací na základě zaměření, vedoucího nebo názvu práce. Do detailu zadání práce se budou přidávat další položky, stejně jako v případě modulu pro aplikaci Teacher – informace o preferenčním oboru, zaměření práce, komentář k zadání, doplňující informace a body k prvnímu semestru. Systém bude umožňovat přihlašování na zadání práce. Zde bude potřeba stejně jako v případě modulu v aplikaci Teacher upravit registraci zadání na přímé přihlašování (viz kapitola 6.3). Další úpravou tohoto modulu bude také kontrola limitů – v případě, že daný vedoucí již nemůže vést další práce, nebude studentovi umožněno se na zadání přihlásit.

Druhým modulem souvisejícím se závěrečnými pracemi je „Moje závěrečné práce“. Ten zobrazuje podobně jako IS FIT kromě detailu zadání i další položky, které student vyplňuje sám – abstrakt práce a klíčová slova (česky a anglicky). Student zde má možnost stáhnout si zadání nebo vzor desek ve formátu PDF. Posudky vedoucího a oponenta může modul podle nastavení parametrizace zobrazovat buď na webové stránce, nebo si je může student opět stáhnout jako soubor PDF. Pro potřeby fakulty bude nastaven parametr pro zobrazování souborů v PDF. Navíc se budou do modulu přidávat položky do detailu zadání jako v předchozím případě. Také se bude přidávat odkaz na modul s detailem přiřazené komise semestrálního projektu, případně komise pro státní závěrečné zkoušky (viz kapitola 7.5).

Položku *počet let utajení* si studenti mohou v IS FIT nastavovat sami. To ale znamená, že pokud ji před finálním odevzdáním práce zapomenou nastavit (což se občas stává), tak ji dále již nebudou schopni upravit. Proto v centrálním systému bude tuto položku (včetně důvodu utajení) mít právo nastavovat pouze děkan, proděkan, případně vedoucí studijního oddělení.

Dále se bude přidávat položka pro uložení adresy na cloudové úložiště *nextcloud*, do kterého studenti vkládají jak elektronickou podobu dokumentace, tak i zdrojové soubory (případně další přílohy). Služba *nextcloud* umožňuje sdíleným souborům nastavit omezený

přístup, což způsobuje problémy. Občas se totiž stane, že vedoucí prací nemohou dané soubory zobrazit. Naštěstí lze ale toto omezení zachytit v URL. Proto bude modul navíc kontrolovat formát adresy pomocí regulárního výrazu, aby se studentům omylem nepodařilo uložit neveřejné soubory.

Dále bude modul umožňovat odevzdávat dokumentace k semestrálnímu projektu ve formátu PDF. Pro všechny studenty magisterského studia bude ovšem povinná, přičemž neodevzdání znamená neudělení zápočtu za předmět SEP a nemožnost pokračovat k SZZ. Položka se bude zobrazovat i studentům bakalářského studia, protože někteří vedoucí to vyžadují. Pro záznam vložených souborů slouží tabulka *st01.zav_prace_soubor* (viz kapitola 4.2). Do číselníku *st01.zav_prace_soubor_typ* pak bude přidán nový typ souboru k závěrečné práci, který bude sloužit k tomu, aby se odlišily texty odevzdávané v rámci semestrálního projektu od finálních verzí závěrečné práce.

Stejně jako v případě modulů v aplikaci **Teacher** budou i v těchto modulech všechny položky zobrazované pro FIT řízeny parametry.

6.5 Veřejný seznam závěrečných prací

Jak již bylo zmíněno v kapitole 2.2, vysoké školy jsou povinny zveřejňovat text práce na svých webových stránkách. V centrálním systému je možné upřesňovat výsledky vyhledávání pomocí filtru. Ten obsahuje několik položek – fakulta, ústav, akademický rok, kdy byla práce obhajována, typ a jazyk práce. Další položkou je textové pole, pomocí kterého lze vyhledávat práce pomocí abstraktu, klíčových slov, vedoucího nebo autora práce.

Dalším z požadavků Fakulty informačních technologií je rozdělit poslední zmíněnou položku na tři nezávislá vyhledávací pole¹. Díky tomu bude uživatel vědět, podle čeho přesně vyhledává (viz obrázek 6.3). Z tohoto požadavku vyplývá také nutnost opravy SQL dotazu.

The image shows two versions of a search form titled "Závěrečné práce".

The top version (original) has the following fields:

- Fakulta (dropdown)
- Ústav (dropdown)
- Akademický rok: 2020/2021 (dropdown)
- Typ práce (dropdown)
- Jazyk práce (dropdown)
- Search field: "Název, autor, vedoucí, abstrakt..."
- Search button: "HLEDAT"

The bottom version (updated) has the following fields:

- Fakulta (dropdown)
- Ústav (dropdown)
- Akademický rok: 2021/2022 (dropdown)
- Jazyk práce (dropdown)
- Typ práce (dropdown)
- Search fields: "Název práce", "Abstrakt, klíčová slova, ...", "Autor práce", "Vedoucí práce"
- Search button: "HLEDAT"

Obrázek 6.3: Formulář pro filtrování vyhledávaných závěrečných prací na veřejné stránce centrálního systému VUT. Horní obrázek zobrazuje původní formulář. Na dolním obrázku je znázorněna úprava dle požadavku FIT.

¹Tato změna se již v době psaní textu využívá v ostrém provozu.

Kapitola 7

Návrh modulů pro správu komisí projektových praxí a semestrálních projektů

Tato kapitola se zabývá návrhem nových modulů pro správu komisí projektových praxí a semestrálních projektů, včetně úpravy stávajících datových struktur, které budou pro ukládání komisí využity. Dále je popsán návrh přístupových práv a uživatelského rozhraní.

Nadanější studenti mají možnost se v rámci studia podílet na praktických projektech. Pro tento účel se využívá skupina předmětů, pojmenovaných jako *projektová praxe*, díky kterým mohou studenti získávat kredity. Aby si však mohli zapsat navazující předmět, a tím získat větší množství kreditů, musí být daná prerekvizita ohodnocena předem stanoveným minimálním počtem bodů, přičemž práci musí obhájit před komisí. Student může na tomto projektu pokračovat nadále jako na své závěrečné práci.

V zinném semestru musejí studenti posledních ročníků svoji dosavadní činnost na závěrečné práci obhájit před komisí. Na rozdíl od projektových praxí, kde hodnotí pouze vedoucí, získávají studenti od komise doporučený počet bodů a slovní ohodnocení. Na základě tohoto doporučení následně obdrží od vedoucího body pro zápočet za předmět semestrální projekt, jenž je povinná prerekvizita pro nadcházející předmět Bakalářská/Diplomová práce (IBT/DIP) v letním semestru.

Doposud existují moduly pro správu komisí pouze v IS Apollo. Fakulta informačních technologií by uvítala možnost mít obdobný modul i ve webovém informačním systému. Pro zatím bude modul umožňovat pouze správu komisí pro semestrální projekty a projektové praxe. Komise pro SZZ budou i nadále pouze v IS Apollo.

Často však před stejnou komisí obhajují studenti své práce jak v rámci semestrálního projektu, tak i projektové praxe, což právě IS FIT neumí zaznamenat. Bylo by proto zřejmě velkou výhodou, kdyby toto nově vytvořený modul umožňoval. Nakonec ale došlo k rozhodnutí, že bude výhodnější tyto komise ukládat a spravovat zvlášť. Prvním důvodem je zkušenost. I dříve byla snaha vytvářet univerzální moduly, které by uměly spravovat více různých agend, které spolu nějakým způsobem souvisely. Postupem času se některé takové moduly staly těžko udržitelné, když bylo zapotřebí je na základě rozdílných požadavků jednotlivých fakult postupně upravovat. Kdykoliv v budoucnu se může stát, že některá ze součástí VUT bude chtít používat pouze semestrální projekty případně projektové praxe, což by mohlo vést ke komplikacím spojeným s úpravou tohoto modulu. Dalším důvodem je

i vyšší výpočetní náročnost v případě, kdy by bylo zapotřebí vypisovat odděleně obhajoby semestrálních projektů a projektových prací.

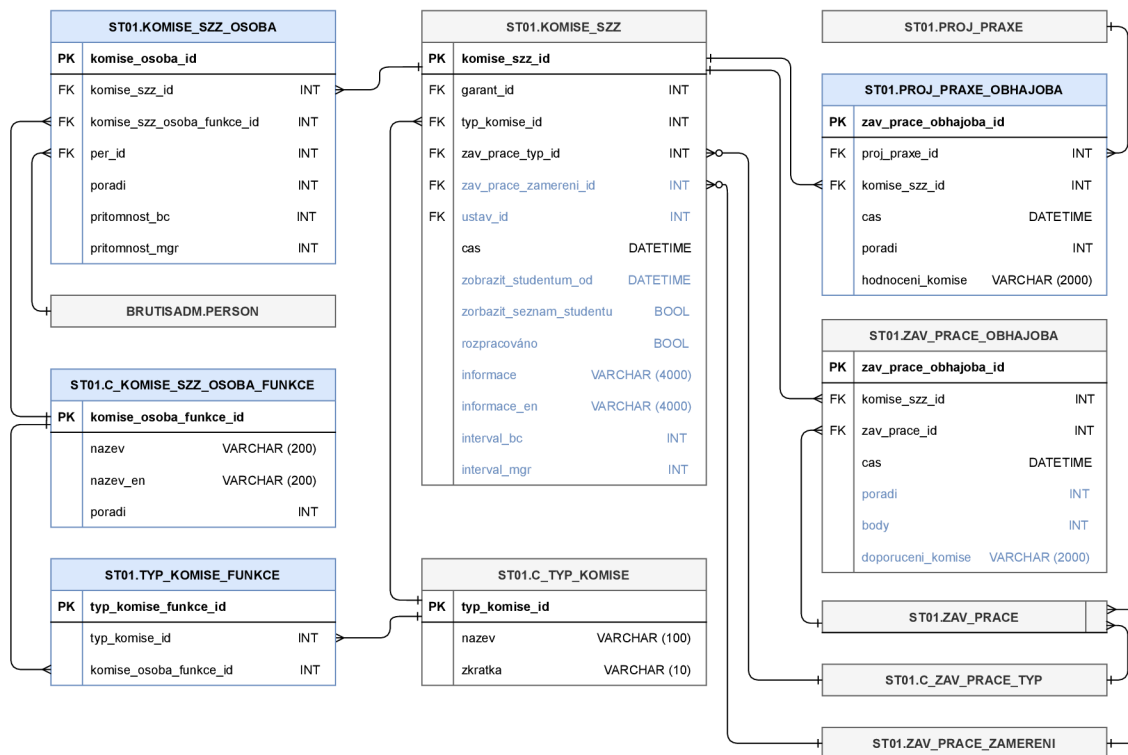
7.1 Návrh úpravy databáze

Ačkoliv se jedná o zcela novou agendu, budou moduly spojené se správou obhajob semestrálních projektů a projektových prací využívat již existující databázové tabulky popsané v kapitole 4.7. Ty ale bude nutné upravit nejen kvůli požadavkům fakulty, ale také z toho důvodu, že se nevhodným způsobem ukládají členové komisí. Tento způsob mimo jiné porušuje i normální formu. ER diagram návrhu změn se nachází na obrázku 7.1. Veškeré přidané atributy či tabulky jsou opět zvýrazněny modrou barvou.

Do tabulky *st01.komise_szz* bude přidáno několik nových atributů, které vyplývají z požadavků fakulty. *Garant komise* je osoba, která má na starosti správu dané komise. Typicky se jedná o uživatele, který komisi založil. V IS FIT se ke komisi přidává také informace o zaměření obhajovaných prací. To ovšem nebude mít vliv na chování modulu jako například omezení studentů podle zaměření atp. Zaměření prací je detailněji popsáno v kapitole 6.3. Dále se budou zvlášť ukládat délky trvání jednotlivých obhajob podle typu studia (atributy *inverval_bc*, *inverval_mgr*). V případě semestrálních projektů budou atributy označovat délku trvání obhajob bakalářských respektive magisterských studentů. Naopak pro komisi projektových prací délku trvání obhajob v rámci předmětů *IP* a *PP*. Délka trvání obhajob (interval) projektových prací se v uživatelském rozhraní může zadávat pro všechny obhajoby stejně, a to tak, že se zadaná hodnota v databázi nastaví obhajobám jak bakalářských, tak i magisterských studentů. Dalším atributem bude datum a čas, kdy budou informace o obhajobě zveřejněny studentům přiřazeným k této komisi. Atribut *zobrazit seznam studentům* určuje, zda-li se bude studentům zobrazovat seznam spolužáků s názvy témat jejich prací, kteří budou v rámci této komise obhajovat s nimi. Dalším rozšířením je atribut *rozpracováno*. Ten dokud není nastaven na hodnotu 0, nebude moci být komise studentům zveřejněna – bude viditelná pouze uživatelům v aplikaci **Teacher**. V neposlední řadě bude tabulka nově ukládat textové informace pro studenty (česky a anglicky).

Druhá změna zahrnuje přepracování aktuálního způsobu ukládání členů komise. Dopusud se členové ukládali přímo v tabulce pro komise, což ale postupem času začalo být neúnosné. Při každé změně v maximálním počtu možných členů u komise bylo nutné přidat nový sloupec. Aktuálně tabulka obsahuje 12 cizích klíčů na členy komise, přičemž předseda, místopředseda a tajemník se ukládají ještě do zvláštních atributů. Další problém, který musí být vyřešen, je možnost rozlišovat různé typy členů. Důvodem jsou například rozdílné finanční odměny v závislosti na typu člena, které zaměstnanci Fakulty informačních technologií dostávají. Pro ukládání samotných členů bude vytvořena tabulka *st01.komise_szz_osoba*. Pro každého člena se bude navíc zaznamenávat i jeho přítomnost na jednotlivých obhajobách a také pořadí, v jakém se bude vykreslovat v tabulce uživatelského rozhraní. Přítomnost komisaře na různých typech obhajob semestrálních projektů či projektových prací bude ukládána pomocí číselných atributů pojmenovaných jako *přítomnost*. Tato číselná hodnota udává, na kolika obhajobách byl komisař přítomen – pokud je hodnota nastavena na 0, byl přítomen na všech obhajobách. Na základě této hodnoty se bude vypočítávat peněžní odměna, přičemž nelze dostat odměnu za studenty, které komisař sám vede. Detaily o komisích spojených se státními závěrečnými zkouškami jsou popsány v následující kapitole.

Funkce členů se budou ukládat do tabulky *st01.komise_szz_osoba_funkce*. Aby se daly odlišit typy funkcí v závislosti na typu komise, bude vytvořena spojovací tabulka *st01.typ_komise_funkce*. Ta navíc zajistí, že se nebudou muset duplikovat funkce komisařů



Obrázek 7.1: Upravený ER diagram ukládání komisí pro obhajoby. Nově přidané atributy a tabulky jsou vyznačeny modrou barvou.

v případě, že některé typy komisí budou využívat stejné funkce. Zatím bude pro komise semestrálních projektů přiřazeno více funkcí, než je pro FIT potřeba (např. předseda, tajemník aj.), protože jsou pro tento typ komise používány na Fakultě elektrotechniky a komunikačních technologií. Pokud by toto do budoucna způsobovalo problémy, bude funkcím navíc přiřazeno ID fakulty pro jejich oddělení. FIT bude typicky využívat dvě funkce – *člen* a *host*. Rozdíl mezi *členem* a *hostem* je v tom, jestli dostane odměnu za přítomnost v komisi, nebo ne. Pro komise v rámci SZZ se bude jednat o funkce jako například *předseda*, *místopředseda*, *tajemník* aj.

Jak již bylo řečeno v kapitole 4.7, záznamy o jednotlivých obhajobách studentů se ukládají ve dvou tabulkách. Pro ukládání záznamů obhajob v rámci semestrálního projektu slouží tabulka *st01.zav_prace_obhajoba*. Obhajoby u SZZ jsou ukládány zvlášť do tabulky *st01.szz_cast*. Tabulka *st01.zav_prace_obhajoba* bude nově zaznamenávat pořadí, v jakém se budou studenti vykreslovat v tabulce uživatelského rozhraní. Dále pak body navržené komisí a textovou položku *doporučení komise*. Kvůli zachování konzistence s aktuálním řešením se budou zvlášť uchovávat záznamy o obhajobách projektových praxí v nové tabulce *st01.proj_praxe_obhajoba*. Má stejný význam jako předchozí tabulka pro semestrální práce.

7.2 Návrh práv pro modul

Protože se jedná o nový modul, bude zapotřebí vytvořit i nová přístupová práva. Tato práva budou mít podobný význam jako v případě závěrečných prací (viz kapitola 6.1), proto nebudou podrobněji popisována. I tato práva budou přidělována na základě profesí.

- **Právo na přístup k modulu** – umožňuje uživateli k tomuto modulu přistupovat a zobrazovat seznam komisí, ve kterých byl přiřazen jako komisař, případně garant. Právo bude přiřazeno například všem vyučujícím, děkanovi nebo studijním poradcům.
- **Právo na zobrazování všech komisí** – dává uživateli možnost zobrazit seznam všech vypsanych komisí. S tímto právem bude moci zobrazovat i jejich detaily. Toto právo budou mít v rámci fakulty přiřazeny stejné profese jako předchozí právo. Právo je navrženo pro případ, že by některá z fakult začala tento modul používat, ale nechtěla dovolit přístup všem vyučujícím ke všem komisím.
- **Právo na vytváření komisí** – umožní danému uživateli založit novou komisi, u které bude automaticky nastaven jako garant.
- **Právo na úpravu jakékoliv komise** – bude přiřazeno například studijním poradcům nebo děkanovi. Bez tohoto práva bude moci danou komisi upravovat garant nebo jakýkoliv její člen.
- **Právo na krádež studentů z přidělené komise** – bude přiřazeno především studijním poradcům. Na fakultě je totiž možné „ukrást“ studenta z komise a přiřadit ho do jiné. Typicky se toto provádí v případě, že student dopředu vzdává obhajobu semestrálního projektu. Takový student je přeřazen do virtuální komise sloužící jako přehled studentů, kteří daný akademický rok nebudou u obhajob přítomni.

7.3 Návrh datových modelů

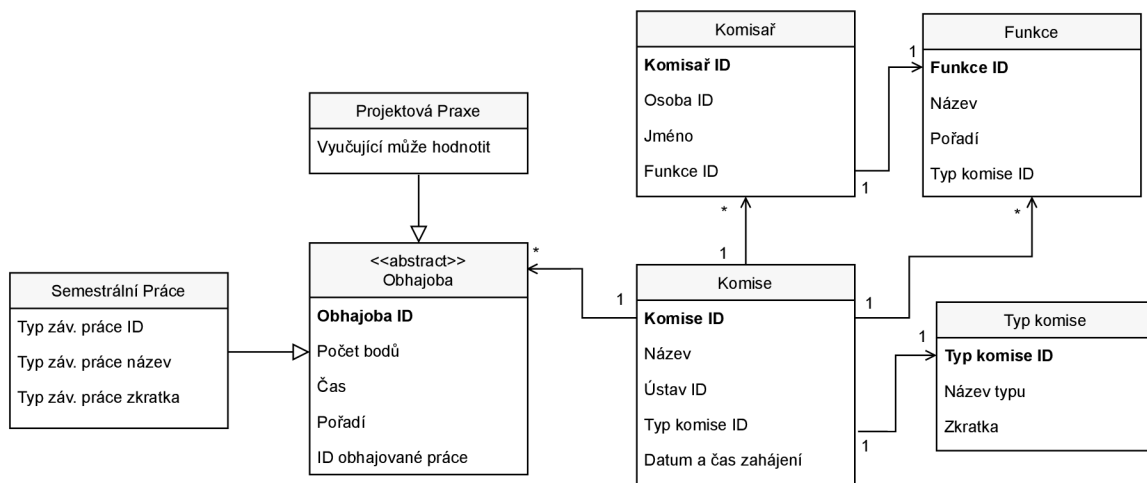
Vztahy mezi datovými objekty jsou znázorněny na obrázku 7.2. Každá komise si drží kolekci obhajob. Konkrétní typ jednotlivých obhajob se odvíjí od typu komise. Pro semestrální práce se ukládají především informace o typu závěrečné práce a pro projektové praxe příznak „vyučující může hodnotit“. Ten se bude označovat zda-li vyučující může hodnotit daného studenta. Na základě jeho hodnoty se buď bude u daného studenta v šabloně zobrazovat pole pro zadání bodů, nebo informace, že nemá právo hodnotit. Bližší informace o hodnocení projektových prací jsou popsány v kapitole 9.6. Komise si také drží kolekci všech funkcí, které mohou být komisařům přiděleny. Funkce se opět odvíjí od typu komise.

7.4 Návrh modulu pro Teacher

Modul pro aplikaci Teacher bude řízen jediným řadičem, a podobně jako řadič modulu projektových prací bude i tento spravovat více pohledů.

Dva z pohledů budou zobrazovat seznam všech komisí, respektive seznam komisí, kde je aktuálně přihlášený uživatel členem. Pohled se seznamem všech komisí bude omezen *právem pro přístup ke všem komisím*.

Stránky pro vytváření a úpravu komisí budou obsahovat formulářové prvky jako například *garant komise*, *ústav*, *název komise* či různé časové údaje (viz obrázek 7.3). Přiřazovat



Obrázek 7.2: Diagram tříd popisující vztahy mezi datovými objekty. Diagram zobrazuje pouze nejpodstatnější atributy, ve skutečnosti obsahují objekty atributů více.

jednotlivé komisaře a studenty bude ale možné až po vytvoření komise – až bude určeno, zda-li se jedná o komisi v rámci semestrálního projektu nebo v rámci projektové praxe.

Pro přidání člena komise se bude zobrazovat dialogové okno s vyhledávacím polem pro jeho snadné nalezení. Modul bude umožňovat nastavit u jednotlivých komisařů jejich funkci (*host* nebo *člen*) a přítomnost na jednotlivých obhajobách (viz kapitola 7.1).

V IS FIT nelze přihlašovat ke komisi studenta, dokud není jeho zadání schváleno. Protože se ale komise mohou vytvářet už od října a termín, do kterého musejí být zadání schválena, je začátkem listopadu, musí se často čekat, než všichni vedoucí ústavů zadání schválí. Bylo by proto vhodné, aby modul v centrálním systému uměl přiřazovat studenty ke komisi v případě, že mají přidělené zadání a zároveň mají zapsaný odpovídající předmět (toto platí jak pro obhajoby spojené s projektovými praxemi, tak i pro obhajoby semestrálních projektů). U studentů, kteří obhajují v rámci semestrálního projektu, se musí kontrolovat, zda-li mají zapsaný předmět IBT/DIP. Je to z toho důvodu, že student může prodlužovat studium a mít již splněný ITT/SEP z předchozího roku. V takovém případě nemá student obhajobu povinnou, ale doporučenou.

Po kliknutí na tlačítko *vložit studenta* se zobrazí dialogové okno se seznamem studentů s přidělenou prací, přičemž budou studenti filtrováni podle typu komise. Aby se dali studenti hromadně přidělovat ke komisi, bude seznam obsahovat zaškrťovací pole. V seznamu bude také vidět, zda-li je student přidělen k nějaké komisi. V případě, že ano, bude moci přeřadit studenta do nové komise pouze uživatel s příslušným právem (viz kapitola 7.2).

Seznam vybraných studentů v detailu komise bude obsahovat kromě jejich jmen, názvů prací a vedoucích také informace o tom, jestli mají zapsaný předmět semestrální projekt (případně některý z předmětů projektové praxe). Bude zobrazena i informace o typu závěrečné práce, pořadí v tomto seznamu a předpokládaný čas obhajoby. U obhajob semestrálních prací bude navíc možnost přistupovat k detailu obhajoby, kde budou moci komisaři studentovi navrhnout bodové ohodnocení včetně doporučení k dalšímu postupu. Udělené body budou zobrazeny také v seznamu studentů.

Dosud bylo na FIT zvykem posílat hodnocení obhajob studentů projektových praxí např. za pomoci e-mailu. Vedoucí práce poslal toto hodnocení vedoucímu ústavu či jiné pověřené osobě, která jej zadala do IS FIT k odpovídajícímu předmětu. Bylo to takto

z toho důvodu, že IS FIT neumí zaznamenávat studenty projektové praxe na těchto komisích. V centrálním systému bude pro větší přehlednost možné zadávat k daným studentům ohodnocení, které budou vyplňovat příslušní vedoucí.

V případě, že studenti se zapsaným předmětem *Semestrální projekt* neobhájí práci, budou hodnoceni známkou F a nebudou mít možnost daný akademický rok pokračovat k SZZ. Podobně je tomu i u studentů se zapsanou projektovou praxí. Pokud neobdrží dostatečný počet bodů, nebudou si moci zapsat navazující předmět (viz kapitola 4.1).

Dále bude modul umožňovat změnu pořadí studentů i komisařů. Na základě pořadí komisařů (vedoucích závěrečných prací) bude možné přeskládat pořadí studentů. Bude také možné pořadí studentů měnit ručně. V neposlední řadě bude modul umožňovat automaticky dopočítat každému studentovi čas začátku jeho obhajoby. Pro každou komisi navíc bude nutné umět rezervovat místnost, aby nedocházelo ke kolizím. V případě, že bude již místnost na daný termín komise obsazena, upozorní modul uživatele a nedovolí provést rezervaci.

V případě, že uživatel systému bude moci k detailu komise přistupovat, ale nebude mít právo ji upravovat, budou položky nastaveny pouze pro čtení.

Aby mohl být modul v rámci aplikace *Teacher* spouštěn, bude potřeba pro něj vytvořit novou skupinu parametrů. Pro tuto skupinu bude vytvořen jediný parametr upřesňující, kterým fakultám se bude modul spouštět. Pro zatím žádné jiné speciální parametry upravující chování nebudou vytvářeny. Může se sice časem ukázat, že některá z fakult začne tento modul využívat a bude chtít mít rozdílné informace o komisích, jako je tomu v případě závěrečných prací, ale v tomto případě budou nové parametry vytvořeny dodatečně dle potřeb dané fakulty.

Protože se budou vypisovat zvlášť komise pro semestrální projekty a projektové praxe, tak pro zjednodušení bude uživatelům umožněno vytvářet kopie komisí. Typicky bude tato funkce používána v případě, kdy u dané komise budou obhajovat studenti v rámci projektové praxe i semestrálního projektu. Pak bude moci uživatel změnit typ komise a čas zahájení obhajob bude automaticky nastaven na předpokládaný čas ukončení předchozích obhajob.

7.5 Návrh modulu pro StudIS

Modul v aplikaci *StudIS* bude sloužit pro zobrazování informací o komisích, ke kterým byl student přiřazen. Bude obsahovat dva pohledy – seznam přidělených komisí a detail dané komise. Pokud má student přidělenou závěrečnou práci nebo projektovou praxi a je přidělen ke komisi, která ale zatím nebyla zveřejněna, bude v pohledu se seznamem zobrazena informace o čase, kdy dojde ke zveřejnění.

V detailu budou veškeré základní informace jako například název, datum a čas zahájení obhajob, místnost konání, seznam jednotlivých členů komise a pokyny pro studenty. Seznam studentů, kteří v rámci stejné komise také obhajují své práce, se bude zobrazovat podle nastavení atributu *zobrazit seznam studentů* (viz kapitola 7.1). Pokud bude seznam skryt, uvidí student pouze informaci o začátku jeho vlastní obhajoby. Pro snadnou orientaci bude možné se na detail komise odkázat i z detailu zadání závěrečné práce případně projektové praxe a naopak. Detail zadání projektové praxe bude zobrazovat seznam všech komisí, u kterých student tento projekt obhajoval (pakliže na projektu pracoval v rámci více předmětů).

Podobně jako pro modul v aplikaci *Teacher* bude i pro tento muset být vytvořena skupina parametrů, aby se dal modul spouštět. Ani zde nebudou vytvářeny žádné speciální parametry upravující chování. Ovšem bude zapotřebí vytvářet více instancí této skupiny,

aby se pomocí omezujících podmínek (viz kapitola 5.3) upřesnily například to, kterým ročníkům se bude modul spouštět.

Menu

Nová komise
Přehled všech komisí
Mé komise
Detail komise

Název komise

Detail komise

Garant komise:

Typ komise:

Ústav:

Název komise:

Zaměření:

Místnost:

Datum a čas zahájení:

Zobrazit studentům od:

Intervaly BP/DP (min.):

Zobrazovat seznam studentů ve StudISu

Informace pro studenty:

Informace pro studenty (anglicky):

Členové komise

Pořadí	Jméno	Ústav	Typ	Přítomnost BP	Přítomnost DP	Akce

Studenti ke komisi

Pořadí	Čas	Body	Typ	ITT/SEP	Jméno	Login	Název práce	Vedoucí

Obrázek 7.3: Návrh uživatelského rozhraní detailu komise pro semestrální projekty.

Kapitola 8

Návrh modulů pro státní závěrečné zkoušky

Tato kapitola popisuje návrh úprav existujících modulů spojených se státními závěrečnými zkouškami. Jedná se o moduly využívané jak ve webovém IS, tak i v samotném IS Apollo.

8.1 Návrh úprav modulu pro posudky závěrečných prací

Při popisu posudků vedoucích a oponentů v kapitole 4.6 bylo mj. zmíněno, že pro webový informační systém VUT již existuje modul pro zadávání hodnocení. Tento modul umožňuje hodnotit na práci různá kritéria v závislosti na hodnotící osobě (vedoucí/oponent), typu závěrečné práce a nebo také rozlišovat tato kritéria mezi různými součástmi VUT. Každému z těchto dílčích hodnocení je možné udělit bodové ohodnocení, známku nebo slovní popis (případně lze využít všechny možnosti zároveň).

Tento způsob hodnocení ale není pro potřeby fakulty dostačující. IS FIT umožňuje vyučujícím vybírat z číselníku různé ohodnocení pro každé kritérium. Tedy například pro kritérium *náročnost zadání* může být vybráno z jedno následujících ohodnocení – *jednoduché zadání*, *méně obtížné zadání*, *průměrně obtížné zadání*, *obtížnější zadání* nebo *značně obtížné zadání*. Některá kritéria jsou ohodnocena pomocí zmíněného číselníku, jiná pouze udělením bodů. Navíc ke všem dílčím hodnocením může být vyplněný dodatečný komentář.

Změny tohoto modulu zasáhnou především databázi. V ER diagramu na obrázku 8.1 je modrou barvou vyznačen nově přidaný číselník *st01.c_zav_prace_hod_vycet*. Ten bude obsahovat veškerá možná ohodnocení všech kritérií včetně pořadí, v jakém se budou zobrazovat v uživatelském rozhraní. Kterému kritériu dané ohodnocení náleží, určuje cizí klíč na tabulku *st01.c_zav_prace_hodnoceni_typ*. Aby bylo možné nastavit, jaká ohodnocení se v rámci typu hodnocení (kritéria) budou udělovat, přidají se do této tabulky čtyři atributy pravdivostního datového typu (zvýrazněny modrou barvou).

V IS FIT existuje jedno kritérium, u kterého se nastavuje hodnocení pomocí dvou číselníků. U *rozsahu splnění požadavků* lze vybrat jedno ze 6 možných ohodnocení, například *zadání splněno*, *zadání nesplněno* atd. Dá se však upřesnit druhým číselníkem s hodnotami *s drobnými výhradami* nebo *s vážnějšími výhradami*. Pokud tato hodnota není vyplněna, má význam *bez výhrad*. Celkově je tedy možných 18 různých kombinací, z nichž však některé nedávají smysl. Všechny smysluplné možnosti budou uloženy v nově přidaném číselníku *st01.c_zav_prace_hod_vycet*. Jinými slovy, pro toto kritérium bude nově existovat 12 stupňů hodnocení – kombinací předchozích hodnot ze zmíněných dvou číselníků.

Výsledné stupně budou:

- *zadání splněno se značnými rozšířeními*
- *zadání splněno*
- *zadání splněno s drobnými výhradami*
- *zadání splněno s vážnějšími výhradami*
- *student se odůvodněně odchýlil od zadání*
- *zadání téměř splněno*
- *zadání téměř splněno s drobnými výhradami*
- *zadání téměř splněno s vážnějšími výhradami*
- *zadání splněno pouze částečně*
- *zadání splněno pouze částečně s drobnými výhradami*
- *zadání splněno pouze částečně s vážnějšími výhradami*
- *zadání nesplněno*

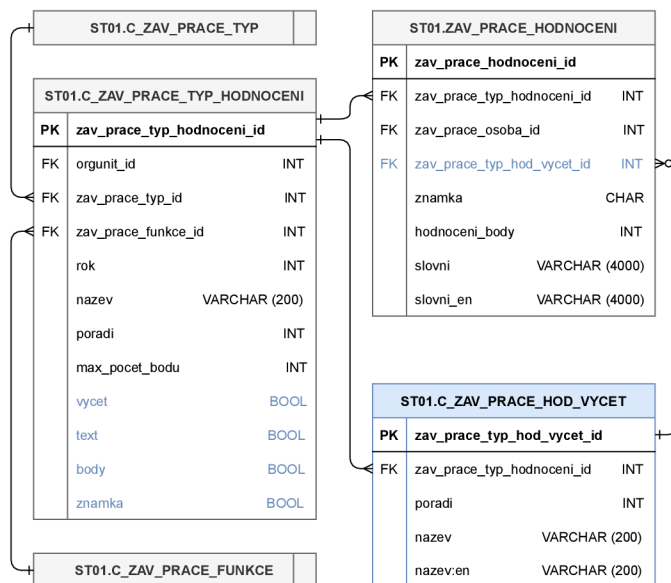
Tato úprava se samozřejmě týká i samotného uživatelského rozhraní. Jedinou změnou však bude přidání pole pro výběr z možných stupňů hodnocení. V závislosti na nastavení atributů u daného kritéria v číselníku *st01.c_zav_prace_hodnoceni_typ* zmíněném výše se budou zobrazovat různé formulářové prvky. Například pro ohodnocení *prezentační úrovně technické zprávy* bude pro zadání bodů zobrazen číselný vstupní prvek společně s textovým vstupem pro zadání komentáře. Další menší úpravou bude také například vzhled souborů typu PDF obsahujících posudky, které si následně studenti budou moci stáhnout v upraveném modulu „Moje závěrečné práce“ (viz kapitola 6.4).

Na otázky k obhajobě může být využita stávající tabulka *st01.zav_prace_otazka*. Protože oponenti na Fakultě informačních technologií vypisují otázky pouze do jediného textového pole v odrážkách, budou všechny tyto otázky v rámci jedné závěrečné práce ukládány jako jeden záznam v atributu *text*.

8.2 Návrh úprav modulu pro správu komisí SZZ

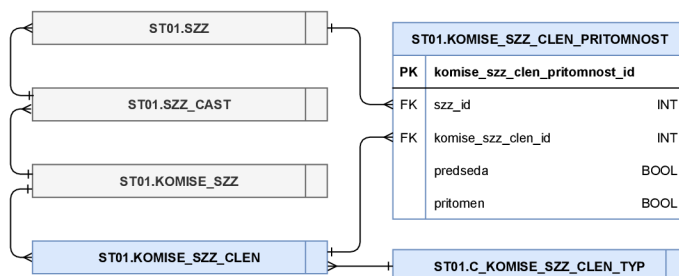
Jak již bylo řečeno dříve, pro všechny typy komisí jsou využity stejné datové struktury. Součástí této práce je také návrh nového způsobu zaznamenávání jednotlivých členů komise, který již byl popsán v návrhu modulu pro správu semestrálních komisí v kapitole 7.1. Proto jsou v ER diagramu na obrázku 8.2 tabulky s touto částí pouze naznačeny pro doplnění kontextu. Tento modul bude využíván zatím pouze v rámci IS Apollo.

Novinkou u komisí pro státní závěrečné zkoušky bude nutnost zaznamenávání přítomnosti jednotlivých členů na každé z obhajob. Ta se podobně jako pro obhajoby semestrálních prací bude ukládat ve zvláštní spojovací tabulce *st01.komise_szz_clen_pritomnost*. Důvod, proč se používá další tabulka, je logické oddělení spolu nesouvisejících dat, a také především kvůli tomu, co se pro každého člena zaznamenává. V tomto případě bude uložena pouze přítomnost na dané zkoušce pomocí pravdivostní hodnoty a zda-li člen byl v rámci



Obrázek 8.1: ER diagram popisující změny v ukládání dat pro modul hodnocení závěrečných prací. Nově přidáné atributy a tabulky jsou zvýrazněny modrou barvou.

dané komise předsedou. IS FIT podobně jako v případě hodnocení kritéria *rozsah splnění požadavků* (viz kapitola 8.1) využívá i pro označování funkcí členů dva číselníky (druhý pro náhradníky). Opět pro zjednodušení budou všechny možné kombinace načítány z jediného číselníku – *st01.komise_szz_clen_typ*. Jedná se o funkce *předseda*, *místopředseda*, *zastupující místopředseda*, *člen*, *zastupující člen*, *tajemník* a *zastupující tajemník*.



Obrázek 8.2: Zjednodušený ER diagram komisí pro státní závěrečné zkoušky. Modře zvýrazněné tabulky jsou nově přidáné oproti stávajícímu řešení. Podrobnější popis fungování komisí je popsán v kapitolách 4.7 a 7.

Pro samotné komise SZZ se bude zaznamenávat obor či specializace¹, číslo komise, datum a místnost konání zkoušek, čas zahájení a ukončení v rámci dané komise, maximální počet zkoušených studentů, které lze ke komisi přiřadit, datum, od kterého budou detaily o komisi zveřejněny přihlášeným studentům, a také doplňující informace pro studenty. Na rozdíl od komisí v rámci obhajob semestrálních projektů odpadá povinnost zadávat v uživatelském rozhraní intervaly jednotlivých typů obhajovaných prací (BP, DP, PP), ústav,

¹Pro programy, které nemají žádnou specializaci, bude vytvořen „virtuální obor“, reprezentující daný program. Studentům z takovýchto programů se ale bude muset zobrazovat pouze název programu namísto názvu „virtuálního oboru“.

kategorii a garanta záznamu. Studenti v aplikaci StudIS vždy uvidí seznam spolužáků v rámci dané komise.

Poslední změnou v detailu komise SZZ je přihlašování studentů k dané komisi. Přihlášení studenti budou zobrazováni v seznamu, přičemž u každého z nich se zobrazí jeho/její jméno, login, název práce, kategorie, do které závěrečná práce spadá, obor či specializace, ročník, počet získaných kreditů za dosavadní studium, vážený studijní průměr, informace, zda-li je student přihlášen k SZZ, a také pořadí v daném seznamu, které se stejně jako v případě modulu pro správu obhajob semestrálních projektů dá měnit. V IS FIT se ke každému studentovi také manuálně nastavuje typ zkoušky (*vše, závěrečná práce* nebo *ústní*), od čehož se odvíjí celková doba trvání SZZ, přičemž platí, že se pro každý typ musí počítat s časovou rezervou pro zhodnocení státní závěrečné zkoušky jako celku a celého studia. V centrálním systému ovšem bude tento typ nastavován automaticky v závislosti na pořadí termínu a části SZZ, ve které student neuspěl.

Jedná-li se o studentův řádný termín, nastaví systém délku zkoušky v plném rozsahu. Opakuje-li však některou z částí zkoušky, bude čas roven délce trvání dané části. V tomto případě se také navíc připočítá čas potřebný pro zhodnocení státní závěrečné zkoušky jako celku a studia.

Délka intervalů pro jednotlivé typy zkoušky se bude moci nastavovat pomocí tří nových parametrů v parametrizaci, tedy délka pro řádnou SZZ, délka v případě opakování ústní části a délka v případě opakování obhajoby závěrečné práce. Aby však bylo možné zpětně získávat informace o délce trvání celé zkoušky, bude tato informace uložena do nově vytvořeného atributu *délka zkoušky* v tabulce *st01.szz*. Všechny tyto parametry budou mít přednastavenou hodnotu, aby nastavení parametrizace na následující akademický rok bylo snazší.

8.3 Návrh úprav modulu pro zadávání výsledků SZZ

Jedná se o další z modulů, který bude pro zatím využívaný pouze v rámci IS Apollo. Pro hodnocení SZZ budou využity datové struktury popsané v kapitole 4.9, které však bude potřeba upravit pro splnění požadavků Fakulty informačních technologií. Tyto úpravy jsou znázorněny v ER diagramu na obrázku 8.3. Veškeré změny týkající se tohoto modulu budou samozřejmě řízeny nově přidanými parametry v parametrizaci.

Hodnocení obhajoby závěrečné práce

První změna v centrálním systému se týká průběhu obhajob závěrečné práce. Součástí hodnocení je jednak slovní popis průběhu obhajoby, a jednak samotná známka. Jak již bylo řečeno dříve, na fakultě je zvykem mít podobné texty předdefinované. K tomu bude sloužit nový číselník *st01.c_szz_cast_prubeh*. Ten bude podle typu části zkoušky (obhajoba nebo ústní část zkoušky) nabízet několik možností hodnocení.

V případě obhajoby závěrečné práce bude nabízet čtyři možnosti – *pozitivní bez otázek*, *pozitivní s otázkami*, *negativní bez otázek* a *negativní s otázkami*². Pro každou z těchto možností se bude zobrazovat v uživatelském rozhraní tlačítko. Podle vybrané možnosti (stisknutého tlačítka) se předvyplní text k hodnocení a v případě negativního hodnocení se automaticky vybere známka **F**. Protože předdefinovaný text slovního hodnocení musí hodnotící doplnit, bude potřeba tento text pokaždé kopírovat do již stávajícího atributu

²V IS FIT se liší text v protokolu podle toho, zda-li oponent zadal studentovi k obhajobě otázku nebo ne.

hodnoceni_slovni v tabulce *st01.szz_cast* k odpovídajícímu záznamu. Známkou bude taktéž ukládána do již existujícího atributu v této tabulce.

Dále se k obhajobě práce zaznamenávají otázky od komise, proto bude potřeba přidat nové vstupní textové pole do tohoto modulu. Toto pole bude předvyplněné z otázek položených oponentem, které se ukládají do tabulky *st01.zav_prace_otazka* (viz kapitola 4.6). Členové komise samozřejmě mohou v rámci obhajoby položit i další otázky kromě těch sepsaných oponentem práce. Aby se tedy mohly zaznamenat všechny tyto otázky dohromady, bude do tabulky *st01.szz_cast* přidán nový atribut.

Je důležité, aby slovní hodnocení obsahovalo informaci, zda-li byly studentovi položeny otázky nebo ne. Z tohoto důvodu bude potřeba před uložením ve slovním hodnocení zkontrolovat přítomnost textu odpovídající tomu z číselníku *st01.c_szz_cast_prubeh*. Tedy například pokud je pole s otázkami vyplněno, bude muset slovní hodnocení obsahovat předvyplněný text *pozitivní s otázkami* nebo *negativní s otázkami*.

V případě, že student práci neobhájí a bude klasifikován známkou **F**, je potřeba také zadat do systému podmínky opakování. Ty se na fakultě opět zadávají pomocí číselníku, proto bude zapotřebí vytvořit další novou tabulku *st01.szz_cast_podm_opak*. V ní budou uloženy hodnoty *doplnit práci*, *znovu práci*, *nové zadání* a *nelze opakovat*. Opakovat obhajobu nelze, pokud je toto studentův druhý pokus (zaznamenáno v atributu *poradi_terminu* v tabulce *st01.szz_cast*).

Hodnocení ústní části zkoušky

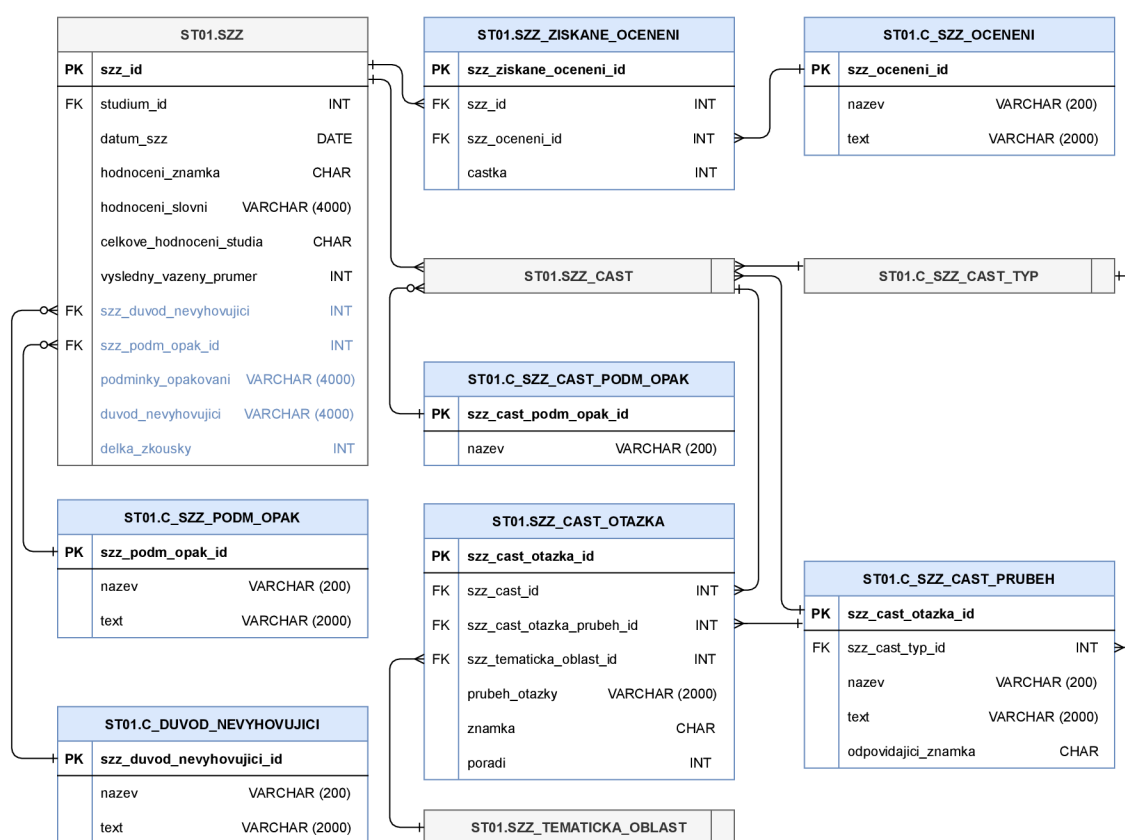
Další rozšíření tohoto modulu spočívá v přidání nových datových struktur pro oddělené hodnocení jednotlivých otázek položených při ústní zkoušce. Doposud se totiž na všech fakultách ústní část SZZ hodnotila dohromady. Proto bude vytvořena také nová tabulka *st01.szz_cast_otazka*. Ke každé otázce se bude zaznamenávat tematický okruh, ze kterého byl daný student zkoušen (viz kapitola 4.8), udělená známka, slovní popis průběhu zkoušky a pořadí, v jakém byly otázky položeny. Popis průběhu zkoušení každé otázky bude opět předdefinován pomocí číselníku *st01.c_szz_cast_prubeh*. Aby se však nepletly texty související s průběhem obhajoby a ústní části zkoušky, budou tyto hodnoty omezeny pomocí číselníku *st01.c_szz_cast_typ*. Na bakalářském studiu jsou studenti zkoušeni z jednoho okruhu a na magisterském ze dvou. Toto nastavení se bude ukládat do parametru v parametrizaci. Každá skupina parametrů bude různá pro jiný program. U ústní části zkoušky se podmínky opakování zadávají až do výsledného hodnocení SZZ.

Po úspěšném složení státní závěrečné zkoušky může být studentovi navrženo ocenění spojené s udělením mimořádného stipendia. Ačkoliv je pro tento účel vyhrazený zvláštní atribut v tabulce *st01.szz*, bude pro potřeby fakulty potřeba navrhnout nový číselník, pomocí kterého se bude na základě uděleného ocenění opět do daného textového pole v uživatelském rozhraní předvyplňovat text, který ale bude možno měnit. Z tohoto důvodu se tento text bude ukládat do výše zmíněného atributu. Samotná ocenění pak budou uložena v číselníku *st01.c_szz_oceneni* zobrazeném v ER diagramu na obrázku 8.3. Opět budou do uživatelského rozhraní přidána tlačítka, pomocí kterých se bude vybírat dané ocenění. Protože lze získaná ocenění za SZZ kombinovat, budou se ukládat ve spojovací tabulce *st01.szz_ziskane_oceneni*. Jedním z požadavků fakulty na centrální systém je možnost počítat stipendia na základě uděleného ocenění. Kvůli tomu, že se mohou částky udělované za dané ocenění v průběhu let měnit, bude spojovací tabulka obsahovat i částku, kterou student získal za danou SZZ. Tyto částky bude možné nastavovat pomocí nové skupiny parametrů v parametrizaci. Dalším z požadavků fakulty je také jednoduché filtrování výsledků zkoušek

podle daného typu ocenění – pro tyto účely bude v IS Apollo upraven seznam proběhlých zkoušek tak, aby se v něm dalo snadno pomocí této hodnoty vyhledávat.

Pro celkové hodnocení SZZ a studia budou využity stávající atributy v tabulce *st01.szz*. Pokud student úspěšně složil obě části zkoušky, musí být i závěrečná známka za SZZ lepší než **F**. V opačném případě musí být do protokolu uvedeny *podmínky opakování* a *odůvodnění hodnocení nevyhovující*. Pro obě tyto hodnoty opět budou vytvořeny číselníky s předdefinovanými texty. Tyto texty se navíc zkopírují do nových atributů, protože je opět možné předdefinovaný text z číselníku upřesnit. Číselník *st01.c_szz_podm_opakovani* udává, kterou část zkoušky bude student opakovat, případně že již opakovat nelze. Budou v něm uloženy možnosti *ústní, práci, ústní i práci, nelze opakovat*. Předdefinované texty *odůvodnění nevyhovujícího ohodnocení* se budou ukládat v číselníku *st01.c_szz_duvod_nevyhovujici* – bude obsahovat možnosti *neobhájil práci a nedostatečné znalosti*.

V případě, že se jedná o druhý pokus studenta, bude se v tomto modulu vyplňovat pouze ta část protokolu o SZZ, kterou opakuje, a celkové hodnocení SZZ a studia.



Obrázek 8.3: ER diagram zobrazující návrh úprav datových struktur pro hodnocení SZZ. Modře označené tabulky budou nově přidány do centrálního systému.

Přítomnost komisařů na zkoušce

Se zaznamenáváním přítomností komisařů na jednotlivých zkouškách souvisí také kontrola různých situací. Například se musí zajistit, aby byl vždy přítomný předseda (či místopředseda, který pro daného studenta převezme roli předsedy). Také nesmí nastat, aby se předsedou stal člověk, který nebyl na dané zkoušce přítomen. Dále by měl systém hlídat

usnášeníschopnost komise, tedy minimální počet členů přítomných v komisi. Pro tyto účely bude vytvořen nový číselný parametr v parametrizaci, který bude určovat minimální počet členů, kteří musejí být přítomni v komisi (na FIT se jedná o minimálně tři členy).

Kapitola 9

Implementace agend

V této kapitole bude popsán průběh celkového vývoje a implementace důležitých součástí modulů projektových praxí, závěrečných prací a správy komisí pro semestrální projekty a projektové praxe. Kapitola se bude zaměřovat především na nové moduly spojené s projektovými praxemi a správou komisí, protože v případě závěrečných prací byly využity již dříve implementované modely a řadiče, které bylo zapotřebí upravit pro potřeby fakulty. V prvních podkapitolách jsou popsány vlastnosti společné pro všechny agendy a následně jsou do jednotlivých podkapitol rozčleněny detaily týkající se jednotlivých agend.

9.1 Základní struktura agend

Všechny implementované moduly jsou rozděleny mezi dvě aplikace – aplikací pro vyučující **Teacher** a aplikaci pro studenty **StudIS**. Zdrojové soubory každé aplikace na webu jsou ve zvláštním adresáři. Existuje také adresář `_base`, který obsahuje komponenty společné pro více aplikací (jinak bývá zvykem, že komponenty, které se používají pouze v jedné aplikaci, se umísťují do adresáře dané aplikace). Ten obsahuje například komponenty pro zpracování formulářů, manipulaci s daty v databázi nebo také různé knihovny pro JavaScript.

Aplikace **Teacher** je postavena na frameworku **Zend** využívajícím architekturu **MVC**. Každý z implementovaných modulů je řízen jediným řadičem.

V případě projektových praxí se v konstruktoru odpovídajícího řadiče načítají práva z databáze a kontroluje se, zda-li přihlášený uživatel má právo přistupovat k modulu. Řadič pracuje s pěti pohledy – seznam všech projektových praxí, seznam projektových praxí vypsaných přihlášeným uživatelem, vypisování nového zadání, úprava zadání a detail přihlášky studenta¹.

V průběhu implementace ovšem došlo ke změně způsobu práce s datovými modely, a proto řadič modulu správy komisí pracuje rozdílným způsobem. Oproti projektové praxi se ve správě komisí nyní využívá nové třídy, nazývané služba, která poskytuje rozhraní pro práci s modely určené pro tuto agendu. Navíc automaticky načítá a kontroluje práva uživatele. Veškeré operace jsou tedy z řadiče delegovány na tuto službu, která dále deleguje práci na datový model *komise*. Řadič poskytuje čtyři pohledy – seznam všech komisí, seznam „mých“ komisí, vytváření nové komise a detail komise. V seznamu „mých“ komisí, jsou filtrovány ty komise, kterých je uživatel členem a/nebo garantem. Seznam všech komisí

¹Seznam všech zadání a seznam zadání vypsaných přihlášeným uživatelem je rozdělen do dvou pohledů především kvůli tomu, aby měly rozdílnou adresu a bylo možné si vytvořit záložku.

bude uživateli zobrazen pouze tehdy, má-li právo pro zobrazování všech komisí na fakultě. V opačném případě bude výchozím pohledem seznam „mých“ komisí.

Aplikace **StudIS** je psána procedurálním PHP. Na **CVIS** je ale zvykem odlišovat logiku a vzhled stránky do různých souborů, podobně jako je tomu v případě architektury MVC. Proto jsou zdrojové soubory s logikou a samotným vzhledem umístěny v rozdílných podadresářích.

Modul projektové praxe pro **StudIS** se skládá z jednoho souboru s logikou a třemi „pohledy“ (šablonami) – šablona pro zobrazování seznamu vypsaných zadání, na která se může student přihlašovat (odpovídají studijnímu programu a akademickému roku), seznam přihlášených zadání a detail zadání. Šablona pro zobrazování seznamu přihlášených zadání zůstává skryta, pokud student nemá žádná zadání přihlášená.

Naopak modul pro výpis přiřazených komisí v aplikaci **StudIS** je pouze informativní – zobrazuje seznam všech komisí, ke kterým je student v rámci studia přidělen, a detail dané komise.

Všechny agendy je navíc potřeba v obou aplikacích nastavit v konfiguračních souborech, aby byly viditelné v menu. V konfiguračních souborech je také možno nastavit, na které pozici v menu bude zobrazen odkaz na výchozí stránku modulu, případně zda-li budou moduly přístupné pouze z testovacího/vývojářského serveru nebo i z veřejného.

9.2 Manipulace s daty v databázi

Protože se budou stejné datové modely používat jak ve **StudISu** tak i v **Teacheru**, jsou soubory určené pro manipulaci s daty umístěny v adresáři `_base`, ve kterém jsou umístěné komponenty společné pro více částí systému.

Jak již bylo naznačeno v předchozí kapitole, agenda projektových prací používá ještě starý způsob práce s datovými modely. V době, kdy byla zahájena implementace agendy pro správu komisí, se začala využívat nová verze datových modelů, která je blíže popsána v následujících odstavcích včetně srovnání se starou verzí. V rámci dalšího vývoje probíhá průběžná refaktorizace této agendy a postupně budou modely přepsány na novou verzi, aby byly do budoucna lépe udržitelné.

Moduly pro správu komisí jsou jedny z prvních modulů, které využívají nově zavedený způsob pro práci s modely². Dříve totiž byla snaha oddělovat umístění modelů podle aplikace, ve které se používaly, což způsobilo nepřehlednost – odtud budou všechny nově vznikající modely v globálním adresáři `_base`. Dalším majoritním rozdílem je, že třídy obecně zvané jako repozitáře (viz text níže) již nebudou obsahovat statické metody, jako tomu bylo doposud, ale budou vždy instanciovány, aby bylo možné jednodušeji provádět jednotkové testy. V neposlední řadě budou modely nově využívat jmenných prostorů (namespace) kvůli jednoduššímu pojmenování tříd. Stávající třídy mají názvy odvíjející se od hierarchického uspořádání v adresářích, což v některých případech vede k dlouhým a nepřehledným názvům. Také byl přidán do struktury nový objekt **Builder** popsáný níže v textu.

Typicky je pro každé společné schéma databáze³ používáno 6 tříd rozdělených do zvláštních souborů.

- **Abstract** – jedná se o abstraktní třídu s vlastnostmi odpovídajícími atributům tabulky (nebo souvisejících tabulek) a jejich „getter“ a „setter“.

²Kromě tohoto modulu je tento nový způsob práce s modely využit v novém submodule pro hodnocení zadání, který jsem vytvořil v rámci své projektové praxe.

³Schéma databáze Oracle je zjednodušeně kolekce databázových objektů, které spolu logicky souvisejí.

- **Builder** – je instanciován v Mapperu a využívá se například pro vytváření instancí subjektů s nastavenými implicitními hodnotami, aby se v Mapperu nemusel kód duplikovat.
- **Iterator** – reprezentuje kolekci subjektů, přičemž obsahuje i metody pro provádění hromadných akcí nad těmito subjekty, např. pro transformaci dat, typicky pro vytváření asociativního pole pro prvek formuláře `select`, kde klíč odpovídá ID záznamu a hodnota názvu, který se zobrazuje uživatelům. Jeho instanci vrací metody Repozi-táře respektive Mapperu.
- **Mapper** – obsahuje metody s dotazy v jazyce SQL. Metody pro čtení dat obvykle buď jako parametr přebírají ID hledaného záznamu, nebo v případě získávání seznamu bývá parametrem asociativní pole pro upřesnění podmínek vyhledávání. Metodám pro ukládání se jako parametry předávají instance subjektu s nastavenými vlastnostmi a objekt reprezentující přihlášenou osobu, aby bylo možné ukládat například ID osoby, která záznam vložila. Každý Mapper v sobě ukládá instanci třídy Builder a Connector sloužící pro připojení k databázi.
- **Repository** – nese v sobě instanci Mapperu a obsahuje stejně nazvané operace. Každá z těchto metod pak volá odpovídající metodu Mapperu a ukládá navíc načtené záznamy z databáze do paměti cache.
- **Subject** – je potomkem abstraktu a jeho instance se buď předává jako parametr metodám repozi-táře pro ukládání, nebo ji vrací metody pro získávání jednotlivých záznamů. Navíc obsahuje metody související s business logikou.

V systému je implementována továrna (**factory**) pro rekurzivní vytváření instancí, která ve své metodě jako parametr bere třídu. Pokud daná třída má v konstruktoru definované argumenty, které jsou taktéž třída, budou automaticky továrnou instanciovány. Tohoto je využito například právě u Repozi-táře – ten v konstruktoru přebírá jako argument odpoví-dající Mapper, a ten v konstruktoru přebírá odpovídající Builder a Connector. Tímto způsobem lze pomocí zavolání jedné metody této továrny instanciovat zmíněné 4 objekty.

9.3 Datagrid

Datagrid je komponenta centrálního systému VUT, která slouží pro zobrazování seznamů. Obdobně jako formuláře, i **Datagrid** se nejprve inicializuje ve třídě, kde se definuje, jaké informace se budou k jednotlivým záznamům zobrazovat. Pomocí vygenerované tabulky je možné zasílat požadavky technologií AJAX například pro řazení podle specifického sloupce nebo pro filtrování záznamů podle zadané hodnoty atributu (viz obrázek 9.1) – tabulka se poté překreslí bez nutnosti obnovovat celou stránku.

Tato komponenta se využívá jak ve **StudISu**, tak i v **Teacheru** napříč všemi moduly pro zobrazování všech seznamů, například seznamu všech zadání/komisí, seznamu přihlášených studentů v **Teacheru** na daný projekt projektové praxe, seznamu všech zveřejněných zadání ve **StudISu** aj.

Datagrid má aktuálně druhou verzi, která je nasazena v několika nových modulech včetně všech modulů popisovaných v této práci kvůli testování v ostrém provozu.

Nové zadání Přehled všech zadání

PŘEHLED VŠECH ZADÁNÍ

Název	Předmět	Rok	Semestr	Jméno vedoucího	Stav	Ústav	Datum schválení	Přijatý student	Přihlášení studenti
		>= 2022							
Efektivní algoritmy pro rozhodování logických formulí	IP1	2022	zimní		příprava	UITS			
Efektivní manipulace s variantami konečných automatů	IP1	2022	zimní		příprava	UITS			
Efektivní metody pro práci s Büchho automaty	IP1	2022	zimní		příprava	UITS			
Experimenty s neuronovými sítěmi na Androidu	IP1	2022	zimní		příprava	UPGM			
Nástroj pro verifikaci programů s dynamickými datovými strukturami	IP1	2022	zimní		příprava	UITS			
Nástroj pro verifikaci systémů pomocí regulárního model checkingu	IP1	2022	zimní		příprava	UITS			
Nástroje pro sběr datové sady videí z internetu	IP1	2022	zimní		příprava	UPGM			
Rozvoj nástroje Spectra v platformě Testes pro automatizované testování softwaru	IP1	2022	zimní		příprava	UITS			
Rozvoj prostředí ANaConDA pro testování a dynamickou analýzu paralelních programů	IP1	2022	zimní		příprava	UITS			

50

Obrázek 9.1: Seznam vypsaných zadání pro projektové praxe včetně filtrování podle akademického roku.

9.4 Implementace agendy projektové praxe

Tato podkapitola popisuje implementační detaily modulů agendy projektových praxí.

Vypisování a úprava zadání

Pro vypisování a úpravu zadání byla vytvořena jediná třída pro správu formulářů, protože se v obou případech upravují stejné položky. Tato třída rozšiřuje komponentu `Form2`, která obsahuje sadu předdefinovaných metod pro generování formulářových elementů, pro zpracování odeslaných dat od uživatele a vykreslování formuláře v šabloně. Elementy, které jsou editory HTML kódu (viz kapitola 5.5), využívají JavaScriptovou komponentu `TinyMCE`. Je nutné předem zmínit, že předměty, pro které je povolen výpis nových zadání, se ukládají do pole v této třídě.

Modul musí omezovat výběr předmětů, na které se mohou vypisovat nová zadání. V případě, že je uživatel správce, případně vedoucí ústavu, může vypisovat zadání pro libovolný předmět související s projektovými praxemi aktuálního a následujícího akademického roku. Pro ostatní uživatele s právem k zápisu se omezí seznam předmětů podle parametrizace. Pokud je pro daný program aktuálně spuštěno vypisování zadání (parametry „Začátek výpisu zadání“ a „Konec výpisu zadání“), zkontroluje se parametr „Program ID pro registraci předmětů“ (viz kapitola 5.3), který říká, pro jaký program je dovoleno vypisování zadání. Podle ID programu se určí, které předměty související s projektovou praxí budou do seznamu zahrnuty. V případě, že není spuštěno vypisování zadání pro žádný program, bude na tuto skutečnost uživatel upozorněn. Uživateli, který nemá právo pro zápis, bude přístup k této šabloně zamítnut řadičem.

Úprava zadání je umožněna opět pouze oprávněným uživatelům – vedoucí zadání, vedoucí ústavu a uživatel s právem správce. Dokud není zadání schváleno, může vedoucí projektu libovolně upravovat veškeré položky. Po schválení může zadání upravovat pouze vedoucí ústavu nebo uživatel s právem správce. Pokud je k zadání přihlášen student, nelze již nikým upravovat předmět a ani nelze změnit stav zadání na „zamítnutý“.

Modul umožňuje také kopírovat zadání. Kopírování je umožněno opět správcům nebo vedoucím ústavů bez omezení nebo uživatelům s právem zápisu v případě, že je spuštěno vypisování zadání pro některý program. Může ovšem nastat situace, kdy se kopíruje staré

Nové zadání Přehled všech zadání Mnou vytvořená zadání **Detail zadání**

NÁSTROJ PRO VERIFIKACI SYSTÉMŮ POMOCÍ REGULÁRNÍHO MODEL CHECKINGU

STAV ZADÁNÍ

Stav: příprava

Stav změnil(a): -

Datum: -

Změna stavu: Schválit zadání Zamítnout zadání

DETAIL ZADÁNÍ

Vedoucí: * X

Ústav: Ústav inteligentních systémů

Předmět: * ▾

Název tématu: *

Název tématu anglicky: *

Zadání práce: *

↶ ↷ **B** *I* U ~~S~~ x^2 x_2 ☰ ☷ 🔗 🗑️ <>

1. Seznámit se s tématikou regulárního model checkingu (RMC).
2. Práce na nástroji pro regulární model checking používajícím některý z přístupů k RMC, např. abstraktní RMC nebo učení

Obrázek 9.2: Detail zadání – formulářové prvky včetně možnosti změny stavu zadání.

zadání nebo zadání vypsané pro program, pro který již vypisování skončilo. V takovémto případě je nutné změnit předmět nově vytvořeného zadání. Modul automaticky nastaví první položku v poli předmětů (viz začátek kapitoly) a v šabloně vykreslí upozornění uživateli. Je zřejmé, že pokud je uživatel správce nebo vedoucí ústavu, nebude u zadání vypsaných aktuální akademický rok předmět změněn.

V editaci také může nastat situace, kdy zadání není schváleno a není na něm ani přihlášený žádný student (tedy lze měnit předmět bez omezení i vedoucím projektu), ale vypisování zadání pro program tohoto zadání již skončilo. Proto je nutné explicitně do pole předmětů vždy přidávat předmět, pro který je zadání vypsané, aby se zobrazoval jako možnost v elementu `select`.

SEZNAM PŘIHLÁŠENÝCH STUDENTŮ

Osobní číslo	Jméno studenta	Login studenta	Vážený průměr	Rok studia	Obor studenta	Stav přihlášky	Datum přihlášení	Přijat na jiný projekt	Předmět zapsán
...	2	NISD	přiděleno	24. 11. 2021		Ano
...	2	NSEN	přihlášeno	24. 11. 2021		Ano

Obrázek 9.3: Seznam přihlášených studentů k projektu.

Načítání práv

Načítání práv probíhá především v konstruktoru řadiče. Nejprve jsou pomocí předprogramované funkce z centrální databáze načtena základní práva popsána v kapitole 5.2 včetně informace, zda je přihlášený uživatel vedoucí ústavu. Následně jsou předzpracovány pod-

mínky, které by se jinak opakovaně vyhodnocovaly v různých komponentách (formulář, datagrid). Tyto podmínky jsou kvůli jednoduchému předávání uloženy v asociativním poli. Vyhodnocuje se například, zda-li je aktuálně spuštěno vypisování zadání pro některý program – pokud není, bude šablona pro vypisování nových zadání zobrazovat upozornění. Pokud má uživatel přiřazena práva správce, má samozřejmě na vše právo a všechny podmínky jsou vyhodnoceny jako pravdivé.

Před načtením detailu projektu se dále vyhodnocují podmínky s ním spojené. Například zda-li má uživatel právo dané zadání upravovat – uživatel má přiřazené právo správce nebo je vedoucí ústavu a zadání je vypsáno pro daný ústav nebo je vedoucí projektu, ale zadání zatím nebylo schváleno.

Přihlašování studentů

Jak již bylo uvedeno v kapitole 3.1, v aplikaci StudIS se automaticky vybírá ta instance skupiny parametrů, jejíž všechny „omezující podmínky“ odpovídají parametrům studia právě přihlášeného studenta. Z této instance jsou poté načteny parametry související s přihlašování studentů k projektům – termín přihlašování a minimální počet získaných kreditů. Pokud některá z těchto podmínek není splněna, nebude dovoleno přihlašování (tlačítka formuláře se skryjí) a modul vypíše studentovi upozornění.

V aplikaci Teacher musí školitel nejprve studenta vyhledat, k čemuž slouží textové pole s našeptáváním. Tento element zobrazuje seznam všech studentů na fakultě (jejichž jméno, případně ID odpovídá napsanému textu) a předem ho nefiltruje pomocí podmínek z parametrizace. Pokud by se totiž školitel spletl a pokusil se přihlásit studenta na nesprávný projekt a modul by jej nezobrazoval, vytvářelo by to dojem, že modul nefunguje správně – školitel by nedostal v tomto případě žádnou zpětnou vazbu, proč nemůže studenta vyhledat. Z tohoto důvodu se kontrola provádí až ex-post. Po potvrzení studenta se pošle pomocí technologie AJAX na server požadavek, a pokud student splňuje podmínky, je rovnou vytvořena přihláška a uživatel je přesměrován na stránku s detailem přihlášky. V opačném případě se v odpovědi požadavku vrací odůvodnění, proč student nebyl zaregistrován. Buď se zobrazí chybová hláška, v případě, že není projekt vypsán pro odpovídající studijní program, nebo se může zobrazit modální okno s upozorněním, že student nesplňuje minimální počet požadovaných splněných kreditů. Takovýto student může být z výjimky přihlášen, proto dostane školitel možnost volby. Přidal jsem navíc možnost vyhledávat studenty pomocí loginu, protože je to na fakultě oblíbený způsob identifikace.

Přes Teacher je možnost přihlašovat studenty i na neschválená zadání (neplatí pro zadání zamítnutá), a to z toho důvodu, že se student mohl domluvit s vedoucím osobně ještě před schválením zadání. Takto přihlášená zadání se musí zobrazit daným studentům ve StudISu bez ohledu na to, zda-li je v parametrizaci nastaveno, že se studentům mohou zobrazovat pouze schválená zadání. Oprávnění uživatelé mohou přidělit zadání studentovi (přijmout přihlášeného studenta) až v případě, že je zadání schváleno.

Může se stát, že uživatel omylem přihlásil nesprávného studenta. Aby se tomuto studentovi nezobrazovala v systému neplatná informace o tom, že měl přihlášen a následně odhlášen projekt, lze přihlášku smazat, ale může to udělat pouze uživatel, který přihlášku vytvořil. Položky se v databázi centrálního systému nikdy přímo nemažou, ale nastaví se atribut `status` na hodnotu -1.

PROJEKTOVÁ PRAXE

Přehled letošních zadání Detail zadání

i Přihlašování začíná 14.01.2022 20:00:00

[CVIS] MODULY IS VUT PRO FIT

Předmět:	Projektová praxe 1 (PP1)
Rok:	2021/2022
Vedoucí práce:	[redacted]
Ústav:	Ústav počítačové grafiky a multimédií
Stav zadání:	schváleno
Přihlášeno:	2
Stav mé přihlášky:	nepřihlášeno

Zadání:

1. Prostudujte vybrané moduly Informačního systému FIT, ke kterým IS VUT nemá ekvivalenty dostatečně pokrývající využívanou funkcionalitu.
2. Ve spolupráci s CVIS VUT navrhnete rozšíření stávajících modulů IS VUT nebo nové moduly pro pokrytí potřebné funkcionality daných modulů IS FIT.
3. Implementujte navržené řešení.

Požadavky na studenta:

- Znalost databázových systémů, nejlépe Oracle.
- Zkušenosti s programováním v PHP (ideálně framework Nette).

Obrázek 9.4: Detail zadání projektu pro aplikaci StudIS

V detailu přihlášky se navíc zobrazují informace o zapsaných předmětech v rámci přiděleného projektu a dále odsud oprávněný uživatel může vstoupit do modulu `detail studia`⁴. Navíc je ošetřena situace, kdy se uživatel snaží přihlásit studenta, jehož program není vypsán v parametrizaci. V takovémto případě zobrazí modul upozornění.

Hromadné schvalování zadání

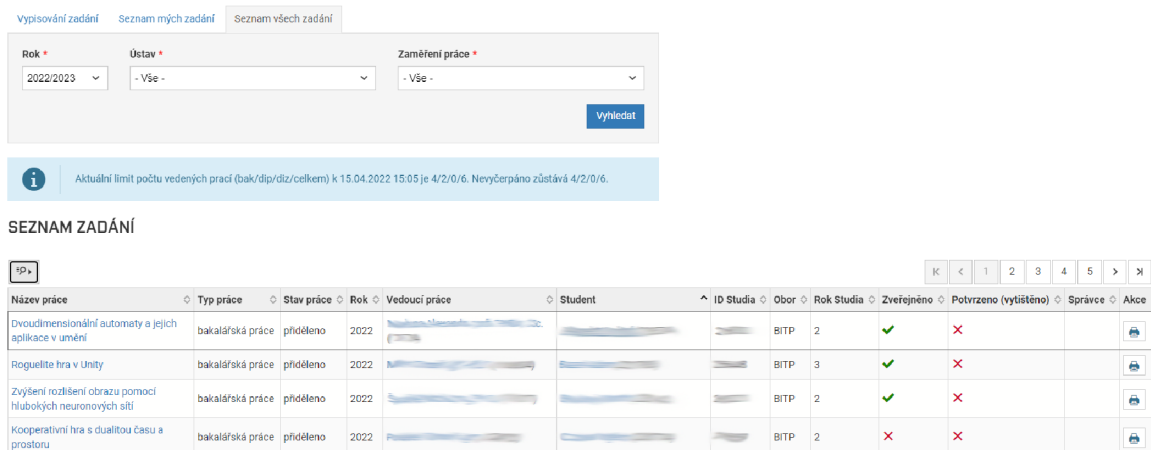
Po nasazení agendy na ostrý provoz byl na modul pro aplikaci `Teacher` vytvořen požadavek na hromadné schvalování zadání. Jedná se o jednoduchý pohled obsahující jediný `DataGrid` se seznamem zadání ve stavu *příprava* nebo *neschváleno*. Přístup k tomuto pohledu je řízen právem správce a rolí vedoucího ústavu. Pokud má uživatel právo správce, jsou zobrazeny veškerá zadání dané fakulty. Pokud je vedoucí ústavu, zobrazují se zadání vypsaná pouze pro daný ústav. Následně uživatel pouze vybere zadání a potvrdí akci. V případě neúspěchu je jeho důvod zobrazen v modálním okně.

9.5 Implementace agendy závěrečných prací

Protože agenda závěrečných prací se v IS VUT používá již dlouhodobě různými fakultami, budou v této podkapitole popsány pouze nejrelevantnější informace spojené s úpravou těchto modulů. Veškerý souhrn změn, které bylo potřeba pro Fakultu informačních technologií implementovat, je popsán v kapitole 6.

V případě editace závěrečných prací bylo vždy nutné kontrolovat, zda-li právě implementovaná změna nezpůsobila chybu na jiné fakultě. Bohužel párkrát (především zpočátku) se právě toto stalo. Především proto, že v době implementace byl modul hojně využíván ostatními fakultami pro přidělování zadání studentům a na rozdíl od FIT jej nepoužívali pouze pro vypisování zadání. Proto jsem dostal přístupová práva i pro ostatní fakulty, abych

⁴Jedná se také o nový modul, který byl vytvořen v rámci přechodu FIT na centrální informační systém, který zobrazuje výsledky studenta v jednotlivých předmětech v rámci zvoleného studia.



Obrázek 9.5: Nový modul pro zobrazování seznamu zadání, který postupně začne nahrazovat stávající nepřehledný seznam zadání používaný několika fakultami. Modul umožňuje stahovat zadání ve formátu v PDF nebo také smazat dané zadání.

mohl snáze opravovat případné chyby a především, abych mohl lépe otestovat, zda-li změna nezpůsobila nečekanou chybu i u jiných fakult.

Tento modul také používá starý způsob práce s datovými modely, a proto je kontrola práv integrována přímo v datovém modelu *zadání*. Právě proto byly do tohoto modelu integrovány kontroly na nová práva pro práci s modulem, která budou využívána Fakultou informačních technologií.

V průběhu implementace vyvstal například požadavek na vkládání více konzultantů a oponentů i mimo VUT (osoby nejsou definovány v databázi). K tomuto účelu se začala využívat nová JavaScriptová komponenta pro vkládání více textových položek zároveň. Tudiž bylo zapotřebí také upravit stávající způsob ukládání těchto osob na backendu. Společně s touto úpravou jsem provedl i refaktorizaci a optimalizaci kódu stávajícího ukládání konzultantů a oponentů. Refaktorizaci jsem se snažil provést v rámci většiny změn provedených na tomto modulu.

Další práce na tomto modulu spočívala v především řešeních nedostatků a chyb, na které se postupně narazilo. Například v případě přímého přiřazení studenta jsem neupravil atribut označující maximální počet studentů, kteří mohou být na zadání přihlášení, což způsobilo, že se tento atribut automaticky nastavil na hodnotu 9999. Chyba byla opravena a hodnota atributu u doposud vypsáných zadání byla upravena na 1. Nebo také byla potřeba například pro Fakultu stavební schovávat různé informace nebo tlačítka v závislosti na parametrech.

9.6 Implementace agendy pro správu komisí

Tato kapitola popisuje implementační detaily modulů spojených s agendou pro správu komisí semestrálních prací a projektových prací.

Kontrola práv

Veškerá kontrola práv probíhá ve zmíněné třídě – službě – která je instanciována v konstruktoru řadiče pomocí továrny (*factory*). Tento *singleton* je postupně předáván do dalších částí systému (například třída pro zpracování formulářů), aby se předešlo duplicitě

kódu. Poskytuje rozhraní pro nalezení detailu komise, úpravy komise a pro operace nad komisaři a studenty (přidat, odebrat, upravit atributy). Služba si udržuje objekt `komise` jako privátní proměnnou, aby se nemusela opakovaně vyhledávat v databázi v různých částech systému. Stejným způsobem si i objekt `komise` drží kolekci členů a obhajob. Tyto operace po zkontrolování práv delegují práci na uložený objekt `komise` (viz kapitola 7.3).

Vytváření a úprava komisí

Stejně jako u modulu projektových prací (viz kapitola 9.4) je i pro vytváření a editaci komisí vytvořena jediná třída pro správu formulářů. Na základě přidělených práv jsou v této třídě generovány buď klasické vstupní položky, nebo položky v režimu pouze pro čtení. Položky pouze pro čtení jsou zobrazeny v případě, že má uživatel právo číst všechny komise, ale danou komisi nemá právo upravovat (není člen ani garant ani nemá právo na úpravu jakékoliv komise).

V průběhu editace komise může nastat situace, kdy uživatel ztratí právo editovat komisi – například se odstraní ze seznamu komisařů nebo nastaví jiného garanta než sebe (a současně uživatel nemá právo editovat všechny komise) – modul tuto skutečnost hlídá a uživatele upozorní před uložením změn pomocí modálního okna. Změna vstupního pole pro zadávání garanta je hlídána pomocí události v JavaScriptu. V případě, že se uživatel odstraní ze seznamu členů, pošle server frontendu pomocí technologie AJAX odpověď a stránka je pomocí JavaScriptu znovu načtena.

Formulář musí kontrolovat zadávané časy. Nesmí například nastat situace, kdy by zveřejnění komise studentům proběhlo až po samotném zahájení obhajob. Toto je řešeno nastavením atributu `max` vstupnímu poli pro zadávání datumu zveřejnění studentům. Protože je ale tato maximální hodnota závislá na datu konání komise, je pomocí JavaScriptu dynamicky měněna.

Pokud jsou již ke komisi přiděleni studenti a proběhne změna času zahájení komise směrem dopředu, jsou všechny obhajoby automaticky posunuty tak, aby první obhajoba začínala ve stejný čas, jako je nové zahájení komise. Je to takto z toho důvodu, že první obhajoba nesmí začínat před zahájením komise. Pro posunutí časů je zavolána jedna z funkcí využívající SQL dotaz `MERGE INTO`, které jsou popsány v kapitole 9.6. Uživatel je před odesláním formuláře upozorněn pomocí modálního okna, pomocí kterého může operaci uložení potvrdit nebo zrušit.

V průběhu implementace byl do tabulky `st01.komise_szz` přidán nový atribut „rozpracováno“, který označuje, že je komise v rozpracovaném stavu a studenti ji nesmí v aplikaci *StudIS* vidět. Pro lepší pochopení tohoto atributu uživateli, je tento příznak ve formuláři označen jako „komise připravena ke zveřejnění ve *StudISu*“ – dokud nebude pole zaškrtnuto, nebude komise pro studenty viditelná. Při zpracovávání formuláře je pouze hodnota před uložením do databáze invertována. Dokud nebude komise připravena ke zveřejnění, bude se na začátku stránky zobrazovat upozornění na tuto skutečnost.

Duplikování komisí

Kvůli rozhodnutí o rozdělení komisí pro různé typy prací (viz úvod kapitoly 7) byla přidána možnost duplikovat komise. Po stisknutí tlačítka pro duplikování komise, je uživatel přesměrován na pohled vytvoření nové komise. Navíc je do URL adresy přidán parametr metody GET obsahující ID komise, která je duplikována. Pokud je takovýto parametr v URL obsažen, je v řadiči tato komise načtena a předána komponentě zpracovávající formulář a ta následně nastaví odpovídající atributy komise jednotlivým prvkům formuláře. Uživatel

Nová komise Přehled mých komisí

NOVÁ KOMISE

✖ Komise s ID = 19117, podle které chcete vytvořit kopii, nebyla nalezena.

Název komise: *

Typ komise: *

Organizátor komise:

Obrázek 9.6: Chybová hláška, která se zobrazí v případě, že se uživatel snaží duplikovat komisi, která buď neexistuje, nebo ji nemá právo zobrazit.

může samozřejmě tyto předvyplněné hodnoty měnit a po úpravě vložit novou komisi. Po vložení nové komise do databáze jsou k ní přiřazení i stejní komisaři.

Nutností je také kontrola práv uživatele vzhledem k duplikované komisi. Uživatel může zadat takovéto ID do URL ručně a eventuálně tak zobrazit informace o komisi, ke které by jinak neměl mít přístup. Případně může také být zadáno nevalidní ID. V těchto případech je v šabloně zobrazeno upozornění o neexistující komisi, jak je znázorněno na obrázku 9.6, a žádné hodnoty nejsou předvyplněny – pohled se chová stejně jako v případě vypisování nové komise.

Hromadné přihlašování komisařů a studentů

Pro všechny seznamy je v rámci této agendy taktéž využita komponenta **Datagrid** (viz kapitola 9.3). Ta se zde využívá mj. v kombinaci s dialogovým oknem k hromadnému výběru komisařů a studentů obhajujících práci. Na obrázku 9.7 je zobrazena ukázka výběru studentů ke komisi. Na začátku seznamu jsou zvýrazněni ti studenti, kteří jsou k dané komisi již přiřazení. Podobným způsobem jsou zobrazeni i studenti již přiřazení k jiné komisi daný akademický rok. Ti jsou řazeni na konec seznamu a zvýrazněni naopak červenou barvou. Tyto studenty lze přeradit z komise pouze v případě, že má daný uživatel přiřazené *právo na krádež studentů* (viz kapitola 7.2).

Řádek	ITT/SEP	Typ	Komise	Jméno studenta	Login studenta	Název záv. práce	Jméno vedoucího
ano	BP	komise KNOT 1				Vytvoření deepfake datové sady pro testování lidské schopnosti rozpoznávat deepfakes	
ano	BP	komise KNOT 1				Počítač jako inteligentní spolupráce ve slovně-asociační hře krycí jména	
ano	DP	komise KNOT 1				Detekce objektů na robotickém pracovišti pomocí dvojice Kinectů	
ano	BP	komise KNOT 1				Informační systém pro testování e-learningových kurzů	
ano	BP	komise KNOT 1				Blender add-on	
ano	DP	komise KNOT 1				RT - pokusná práce 1 - SMÁZAT	
	BP	-				Hodnocení lidské schopnosti rozpoznávat syntetickou řeč – deepfakes	
ano	DP	-				Platforma pro realizaci výpočtu v kapalném krystalu	
ano	DP	-				Simulované vkládání chyb v síťové komunikaci	
ano	BP	-				Konstrukce a řízení audio zesilovače	

Obrázek 9.7: Dialogové okno zobrazující **Datagrid** pro výběr studentů ke komisi pro obhajoby semestrálních prací. Studenti, kteří jsou přiděleni ke komisi, jsou zvýrazněni zeleným pozadím a řazeni na začátek seznamu.

V závislosti na typu komise (semestrální práce / projektová praxe) se zobrazují jiní studenti s různými informacemi. U komisí pro semestrální práce se zobrazují všichni studenti, kteří mají na daný akademický rok přidělené zadání závěrečné práce a zároveň mají zapsaný předmět s účelem bakalářská nebo diplomová práce. V případě komisí pro projektové praxe se zobrazuje seznam studentů, kteří daný akademický rok a semestr mají zapsaný některý z předmětů projektové praxe a mají přidělený některý ze schválených projektů.

Pro vkládání nového studenta do databáze se používá operace `INSERT INTO SELECT`. Pomocí operace `SELECT` se nejprve nalezne pro danou komisi poslední obhajoba (podle nastaveného pořadí) kvůli automatickému doplnění pořadí a času pro nově vkládanou obhajobu. Pokud operace `SELECT` nenalezla žádnou obhajobu (vkládá se první obhajoba), vrací implicitně pořadí „1“ a čas obhajoby odpovídá času zahájení komise.

Aby nedocházelo při každém přidání nových studentů k překreslení celé stránky, umožňuje `Datagrid` zasílat požadavky na server pomocí technologie `AJAX`. Následně obdrží od serveru odpověď a dojde k jeho překreslení. Díky tomu zobrazuje aktuální stav komise. Zároveň s ním se aktualizuje i `Datagrid` se seznamem přidělených studentů, který je součástí detailu komise.

Veškeré požadavky technologií `AJAX` odeslané z `Datagridů` jsou zpracovávány řadičem a ten dále deleguje práci na zmíněnou službu, kde před samotnou operací proběhne kontrola práv. Pokud dojde k chybě, je oznámena uživateli skrze dialogové okno.

Díky tomu, že je `Datagrid` pro výběr studentů/komisařů generovaný dynamicky zasíláním požadavku technologií `AJAX`, v případě, že uživatel v průběhu editace komise přijde o možnost komisi nadále upravovat, daný `Datagrid` nebude uživateli ani zobrazen.

Podobným způsobem, jako přidávání studentů, je řešeno i přidávání komisařů. Opět jsou na začátku seznamu řazeni komisaři přidělení ke komisi. Protože se ale mohou komise různě prolínat a komisař může být teoreticky přiřazen ke dvěma takovými prolínajícími komisím, lze přiřazovat komisaře k libovolným komisím bez jakékoliv další kontroly. V seznamu se prozatím zobrazují pouze zaměstnanci Fakulty informačních technologií (respektive té fakulty, pro kterou je modul aktuálně spuštěn a používán). V průběhu navazujících prací bude možnost přidat i komisaře, kteří nemají definovaný vztah k VUT (například hosté z externích firem).

Rozlišení databázových tabulek v závislosti na typu komise

Podobně jako datové modely (viz kapitola 7.3) existují i různé Repoziáře pro různý typ obhajoby – Repoziář pro obhajoby semestrálních prací a Repoziář pro obhajoby projektových prací. Oba dědí z abstraktního Repoziáře pro zajištění stejného rozhraní. Stejným způsobem jsou navrženy i Buildery, Iterátory a Mappery. Při inicializaci komise se podle jejího typu nastaví jeden z těchto Repoziářů. V závislosti na konkrétní instanci Repoziáře se volají rozdílné Mappery, což umožňuje pracovat s různými databázovými tabulkami nad jedinou business logikou. Podobně pracuje modul i s jedinou třídou pro zpracovávání formuláře obhajob.

Hromadné akce nad studenty

Jak již bylo zmíněno v kapitole 7.4, bude modul v aplikaci `Teacher` umožňovat hromadné operace jako seřazení studentů podle pořadí vedoucích, nastavit časy obhajob studentů podle jejich aktuálního pořadí a přecíslovat pořadí. Veškeré tyto operace volají odpovídající metodu v řadiči pomocí požadavku technologií `AJAX`. Následně proběhne opět delegování na službu a kontrolu práv. Samotné operace jsou prováděny každá pomocí jediného `SQL`

dotazu `MERGE INTO`. Ta nejprve provede výběr dat pomocí operace `SELECT` a následně provádí operaci `UPDATE` nebo `INSERT` na základě splnění/nesplnění dané podmínky. Jedná se o atomickou operaci, díky čemuž nedojde k nekonzistentnímu stavu databáze v případě, že by danou operaci u stejné komise provedli dva uživatelé současně. Například pro seřazení obhajob podle komisařů se provede operace `SELECT`, která pomocí `JOIN` spojí obhajoby dané komise s jejich vedoucími a následně provede seřazení výsledku podle pořadí těchto vedoucích (komisařů), přičemž pomocí pseudosloupce `ROW_NUMBER` je spočítáno nové pořadí obhajob. Ve stejném SQL dotazu je také dopočítán i nový čas obhajoby na základě jejího nového pořadí a délky trvání. Vzhledem k tomu, že modul umožňuje zadat rozdílnou délku trvání obhajob bakalářských a diplomových prací, je potřeba pro každý řádek zjistit délku trvání předchozí obhajoby. K tomuto je využita funkce databáze `LAG`. Pokud vedoucí dané práce není přiřazen ke komisi jako komisař, jsou obhajoby těchto prací řazeny na konec seznamu. Následně probíhá `UPDATE` záznamů v tabulce s obhajobami podle odpovídajícího primárního klíče. Podobným způsobem jsou řešeny i ostatní zmíněné hromadné akce.

Dále je možné manuálně měnit nastavení jednotlivých obhajob (časy, pořadí) a hromadně tyto informace poslat pomocí formuláře. Komponenta pro zpracování formulářů při každém zpracování formuláře implicitně vytváří databázovou transakci, díky čemu je i tato hromadná úprava atomická a nedojde k nekonzistenci dat. Formulář na frontendu a backendu samozřejmě validuje vstupní data – například čas žádné obhajoby nesmí předcházet času zahájení komise.

Udělování bodů studentům

Udělování bodů studentům se opět liší v závislosti na typu komise. Pro obhajoby semestrálních prací se udělené body ukládají k dané obhajobě do tabulky `st01.zav_prace_obhajoba`. Dále s těmito body modul nijak nepracuje. V případě obhajob projektových prací jsou body připisovány přímo k danému předmětu, který má student zapsaný. Pro každý takovýto předmět musí být vypsáné zadání, kam mj. může student v aplikaci `StudIS` odevzdávat zprávu.

Pro hodnocení zadání vznikl pro Fakultu informačních technologií nový submodul, který jsem implementoval v rámci své projektové praxe a v době psaní textu je již nasazen a používán v ostrém provozu. Pro tento submodul vznikly také nové modely a služba, které již respektují novou filozofii, která se na `CVIS` zavedla (viz kapitola 9.2). Zmíněná služba je využívána při hodnocení projektové praxe. Kontroluje, zda-li má uživatel práva hodnotit daného studenta. Vyučující musí například být garantem (zástupcem garanta) daného předmětu nebo mít právo udělovat body všem studentům v rámci předmětu nebo udělovat body studentům, kteří jsou přihlášení na zadání, u kterých je daný vyučující vedoucím. Veškerá tato práva se dají nastavit v `IS Apollo`. Pro projektovou praxi bude potřeba všem školitelům nastavit právo udělovat body všem studentům v rámci daných předmětů, aby nemuselo být pro každého školitele vypisováno zvláštní zadání.

Pokud vyučující nemá právo některé studenty hodnotit, jsou odpovídající vstupní pole s body skryta a nahrazena varovnými ikonami s tooltipem (viz obrázek 9.8). Dále může nastat situace, kdy daný student není registrován na žádné zadání v rámci předmětu – i v tomto případě je pole s body skryto a zobrazena varovná ikona. Kontrola práv a registrace na zadání (kvůli vykreslení správných informací) probíhá již v databázi při dotazování se na seznam přiřazených studentů ke komisi.

Po potvrzení výsledků jsou body zapsány do tabulky *st01.evidence* k odpovídajícímu zadání předmětu. Následně musí být zavolána uložená procedura, která provede přepočítání materializovaných hodnot v elektronickém indexu studenta.

Veškeré dříve zmíněné hromadné úpravy atributů obhajob (udělování bodů, manuální úprava časů a pořadí) jsou zpracovávány pomocí formuláře. V době psaní textu se pracuje na rozšíření funkcionality *Datagridu* o editaci buněk. To bude umožňovat zpracovávání formuláře pomocí technologie *AJAX* bez nutnosti obnovovat stránku.

SEZNAM STUDENTŮ

Vizit studenty

Hodnocení studentů bude propisováno přímo do elektronického indexu k vypsanému zadání daného předmětu.

Někteří ze studentů nejsou registrováni na zadání v rámci svého předmětu - takovým nelze zapsat body.

Řádek	Pořadí	Čas	Body	Předmět	Jméno studenta	Login studenta	Název proj. praxe	Jméno vedoucího	Akce
	10	09:20	75	IP2			Transformace informačního systému FIT na platformu VUT		
	20	09:35	85	IP2			Transformace informačního systému FIT na platformu VUT II.		
	30	09:57	56	IP2			Testování a optimalizace FPGA radiče pro externí DDR4 paměť		
	40	10:05		PP1			Webové rozhraní pro stabilizační platformu		Odebrat studenta
	50	10:20		PP1			HW platforma pro strojové učení		
	60	10:35		PP1			Rozpoznání zánětlivého ložiska z termosnímků		

Student není přiřazen na žádné zadání v rámci předmětu PP1. Není možné mu udělit body.

Vybrané (0): Seřadit podle vedoucích Nastavit časy podle pořadí Přecíslovat pořadí Odebrat vybrané studenty z komise

50 strana: 1/1, řádků: 6

Uložit nastavení obhajob

Udělit body studentům

Obrázek 9.8: Na obrázku je zobrazen seznam studentů přiřazených ke komisi typu *projektová praxe*. Jsou zde znázorněny tooltipy pro chybové hlášky a mazání jednotlivých studentů. Tlačítka pro změnu atributů a bodů jednotlivých obhajob nejsou prozatím součástí *Datagridu*. Po dokončení rozšíření, které bude umožňovat editovat buňky a zpracovávat formuláře pomocí technologie *AJAX*, budou tlačítka přidána pro větší přehlednost vedle ostatních hromadných akcí.

Kapitola 10

Testování a nasazení agend

Ve vývojovém prostředí byly moduly testovány pomocí testování identit, což je způsob, pomocí kterého je vývojář schopný přihlásit se do systému pod účtem jiného uživatele. Díky tomu bylo jednoduché testovat u různých uživatelů funkcionalitu nových modulů v závislosti na přidělených právech. Pro účely testování vyučujícími byla pro každou agendu na testovacím serveru vytvořena nová preview verze.

10.1 Projektové praxe

Agenda projektových prací byla s předstihem dostatečně otestována a následně nasazena již v listopadu roku 2021, kdy se využívala pouze část pro vyučující, kteří mohli začít vypisovat nová zadání. Od ledna roku 2022 je využíván i modul v aplikaci StudIS, kde se studenti mohou přihlašovat na vypsaná zadání. V průběhu zimního a letního semestru byly objeveny drobné nedostatky, které byly následně opraveny. Například při duplikování zadání původně modul rovnou vytvořil v databázi nový záznam se stejnými hodnotami jako mělo původní zadání. Většina uživatelů operaci duplikování pochopila tak, že pokud v právě duplikovaném zadání změni hodnotu, promítnou se i tyto změny do nového zadání. Nakonec bylo duplikování upraveno tak, aby bylo konzistentní napříč všemi moduly – pokud chce uživatel duplikovat zadání, zobrazí se šablona pro vytváření nových zadání s předvyplněnými hodnotami, které může upravit a následně vložit jako nové zadání. Toto řešení je pro uživatele mnohem přívětivější. Dále byl vytvořen požadavek na hromadné schvalování zadání, což vedoucím ústavů zjednodušilo práci. Dalším rozšířením navrženým uživateli bylo odeslání žádosti vedoucímu ústavu o schválení zadání. Díky tomuto nemusejí uživatelé posílat tuto žádost například pomocí e-mailu, ale modul odešle danému vedoucímu VUT zprávu. I toto rozšíření bylo závčas implementováno.

10.2 Závěrečné práce

Agenda závěrečných prací začala být Fakultou informačních technologií využívána taktéž od listopadu roku 2021. Bohužel pro podrobné testování nezbyvalo před nasazením příliš času, proto byl modul na vývoji odladěn a následně byly veškeré připomínky a nedostatky upravovány již v ostrém provozu. Například se jednalo o možnost stažení zadání v souboru formátu PDF, aby mohl vyučující zkontrolovat správnost jeho formátování, nebo filtrování podle preferenčních oborů prací v modulu *seznam závěrečných prací*. Dále vyvstal požadavek na možnost duplikování zadání, protože někteří vyučující často vypisují více podobných

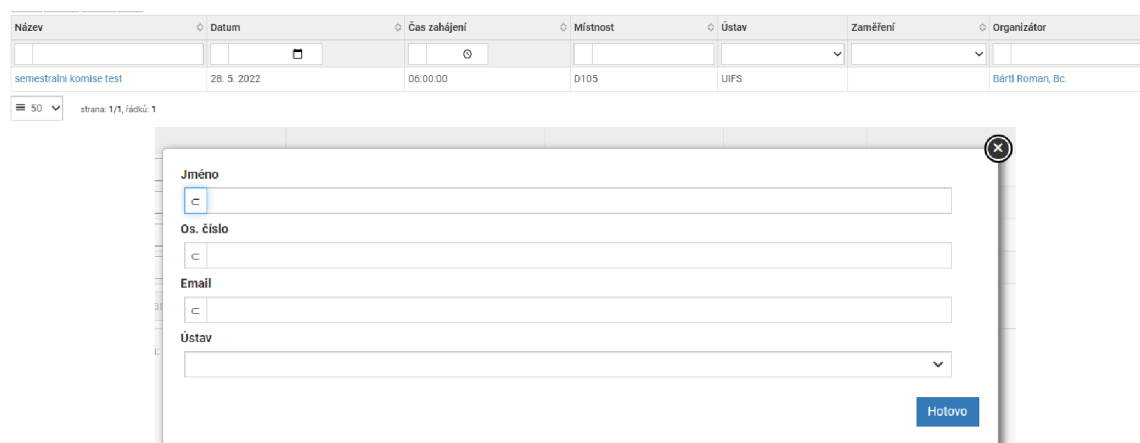
zadání s menšími obměnami, a tudíž by bylo časově náročné a nepohodlné vypisovat vícekrát stejné informace. Toto rozšíření bylo následně otestováno a přijato.

10.3 Správa komisí pro semestrální práce a projektové praxe

Agenda pro správu komisí bude taktéž hromadně využívána až od akademického roku 2022/2023. K testování této agendy byla vybrána úzká skupina deseti vyučujících, kteří se zaměřovali především na testování komisí pro projektové praxe. Na konci letního semestru v akademickém roce 2021/2022 bude totiž modul využit v ostrém provozu pro výpis komisí pro přibližně 30 studentů, kteří v daném semestru obhajují své projekty. Toto testování proběhne, aby se modul řádně otestoval na menším vzorku studentů přímo v provozu a odladily se pokud možno všechny problémy, které nesmějí nastat následující semestr, kdy bude modul využíván pro obhajoby semestrálních prací.

Většina nahlášených nedostatků v rámci předběžného testování se týkala **Datagridu**. Jak bylo zmíněno výše v textu, zatím tato komponenta není uzpůsobena pro editování buněk a pro možnost zadávat body, měnit pořadí nebo čas obhajob se tedy musí využít formulář. Většina testerů si stěžovala na nepohodlnost. Dále se projevovalo překreslování stránky po každém kliku na jakékoliv tlačítko generované **Datagridem**. Všechna tato tlačítka jsou generovaná pomocí značky `button` jazyka HTML, a proto docházelo k odeslání formuláře.

Dále se ukázalo, že dosavadní způsob filtrování v **Datagridu** v dialogovém okně je nevyhovující a nepochopitelný pro většinu testerů. V případě, že je **Datagrid** vykreslován v dialogovém okně, nejsou pole pro zadávání filtrovaných hodnot v hlavičce tabulky, ale je potřeba tyto filtry zobrazit pomocí speciálního tlačítka. Na obrázku 10.1 je zobrazeno srovnání zmiňovaného nevyhovujícího způsobu filtrování a klasického filtrování. Důvodem pro tento způsob je úspora místa kvůli omezené výšce dialogového okna. V případě zobrazení na mobilním zařízení se stává tabulka nepřehlednou. Nakonec ale byla na CVISu domluvena úprava, a to tak, že v případě větších monitorů bude mít **Datagrid** v dialogovém okně klasické filtrování jako v ostatních případech a schovávat se bude pouze pro menší obrazovky. Další drobnější nedostatek se týkal absence ústavu vedoucího zadání v seznamu obhajob.



Obrázek 10.1: Na horním obrázku je zobrazeno klasické filtrování, kde uživatel pomocí vstupních polí může jednoduše a přehledně zadávat hledané hodnoty. Ve spodní části obrázku je zobrazen způsob filtrování v **Datagridu** vykresleném v dialogovém okně. Původní tabulka je schována a nahrazena tímto seznamem se vstupními poli.

Kapitola 11

Závěr

Cílem této práce bylo nastudovat problematiku projektových prací, závěrečných prací a státních závěrečných zkoušek na Fakultě informačních technologií VUT a na základě požadavků fakulty navrhnout nové moduly nebo úpravy stávajících modulů v centrálním informačním systému VUT tak, aby fakulta mohla přejít z vlastního informačního systému na centrální. Následně byly jednotlivé agendy podle schváleného návrhu implementovány.

Projektové praxe jsou v IS VUT zcela nová agenda – proto jsem pro ně ve webové části IS VUT navrhl nové moduly včetně databázových struktur, práv a parametrizace. Tato agenda sestává ze dvou modulů pro aplikaci **Teacher** a pro aplikaci **StudIS**. V současné době je již implementována a využívána na ostrém provozu vyučujícími fakulty, kteří již postupně přihlašují zainteresované studenty na své projekty.

Moduly pro správu závěrečných prací jsou v centrálním systému VUT používány již delší dobu. Pro ně proto bylo potřeba navrhnout změny korespondující s požadavky fakulty. Tyto změny jsou řízeny nově vytvořenými parametry v parametrizaci, aby jimi nebyly ovlivněny ostatní fakulty VUT. Pro jednoduchou úpravu chování v závislosti na roli daného uživatele byla přidána i nová práva. Dále byl vytvořen nový modul pro přehled všech závěrečných prací s možností filtrování. Všechny tyto změny jsou taktéž implementovány a používány vyučujícími v ostrém provozu ve webovém IS VUT.

Agenda pro správu komisí je ve webovém IS VUT také novinkou. Pro tuto agendu byly využity stávající databázové struktury, které ale byly rozšířeny o nové tabulky. Tyto tabulky se týkají především nových číselníků a také nového způsobu ukládání členů, který začne být postupně využíván ve všech modulech spravujících komise. Agenda byla taktéž úspěšně implementována a následně úzkým kruhem vyučujících otestována, přičemž bude pro přesnější testování použita na vytvoření komisí projektových prací na konci letního semestru akademického roku 2021/2022 pro menší počet studentů, aby se zavčas stihly odladit veškeré nedostatky. Agenda začne být hromadně používána od poloviny zimního semestru roku 2022/2023.

V rámci navazující práce započne implementace navržených úprav posudků závěrečných prací a úpravy agendy SZZ v IS Apollo dle popsaného návrhu. Dále proběhne rozsáhlejší testování agendy pro správu komisí před jejím nasazením pro všechny vyučující v akademickém roce 2022/2023. V neposlední řadě budu průběžně provádět refaktorizaci agendy projektových prací tak, aby používané modely korespondovaly s nově zavedenou politikou CVIS pro práci s datovými modely.

Literatura

- [1] ČESKO. *Zákon č. 137 ze dne 2. března 2016 o vysokých školách. In: Sběrka zákonů České republiky.* [online]. Březen 2016 [cit. 2021.12.28]. Dostupné z: <https://www.msmt.cz/file/37522/>.
- [2] DYTRYCH, J. *Státnice a diplomky* [online]. Fakulta informačních technologií Vysoké učení technické v Brně, 7. prosince 2021 [cit. 2021.12.2]. Dostupné z: <https://www.fit.vut.cz/study/theses/master-theses/.cs>.
- [3] DYTRYCH, J. *Harmonogram zpracování závěrečných prací* [online]. Fakulta informačních technologií Vysoké učení technické v Brně, 10. ledna 2022 [cit. 2021.12.2]. Dostupné z: <https://www.fit.vut.cz/study/theses/schedule/.cs>.
- [4] HRUŠKA, T. *Projektová praxe 1* [online]. Fakulta informačních technologií Vysoké učení technické v Brně, 2021 [cit. 2021-11-29]. Dostupné z: <https://www.fit.vut.cz/study/course/IP1/.cs>.
- [5] PRETTYMAN, S. *Learn PHP 7: object oriented modular programming using HTML5, CSS3, Javascript, XML, JSON, and MySQL.* 1. vyd. Berkeley, CA New York, NY: Apress, Distributed to the Book trade worldwide by Springer, 2016. ISBN 978-1484217290.
- [6] SHARMA, J. *Oracle Database Database Development Guide, 19c* [online]. A. Kumar, L. Morin, C. Murray, T. Kyte, D. Adams, L. Ashdown, S. Moore, E. Paapanen, R. Strohm, R. Ward. červenec 2019 [cit. 2021.11.20]. Dostupné z: <https://docs.oracle.com/en/database/oracle/oracle-database/19/adfns/database-development-guide.pdf>.
- [7] ŠTEPÁNEK, P. *Směrnice č. 69/2017 - Standardy studijních programů VUT, čl. 2, odst. 3* [online]. Vysoké Učení Technické v Brně, 26. září 2017 [cit. 2021.12.28]. Dostupné z: [https://www.vut.cz/uredni-deska/vnitрни-predpisy-a-dokumenty/-d154567/uz-smernice-69-2017-do-\(d1-d2\)-p192547](https://www.vut.cz/uredni-deska/vnitрни-predpisy-a-dokumenty/-d154567/uz-smernice-69-2017-do-(d1-d2)-p192547).
- [8] ZEND TECHNOLOGIES INC.. *Programmer's Reference Guide Zend Framework* [online]. Listopad 2007 [cit. 2021.11.20]. Dostupné z: <https://framework.zend.com/manual/1.0/en/index.html>.