

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

System pro nabídku témat závěrečných prací na VŠ



2023

Vedoucí práce:
doc. Mgr. Jan Outrata, Ph.D.

Filip Martiník

Studijní program: Informatika,
Specializace: Programování a vývoj
software

Bibliografické údaje

Autor:	Filip Martiník
Název práce:	Systém pro nabídku témat závěrečných prací na VŠ
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2023
Studijní program:	Informatika, Specializace: Programování a vývoj software
Vedoucí práce:	doc. Mgr. Jan Outrata, Ph.D.
Počet stran:	37
Přílohy:	elektronická data v úložišti katedry informatiky
Jazyk práce:	český

Bibliographic info

Author:	Filip Martiník
Title:	System for offering topics of diploma theses
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2023
Study program:	Computer Science, Specialization: Programming and Software Development
Supervisor:	doc. Mgr. Jan Outrata, Ph.D.
Page count:	37
Supplements:	electronic data in the storage of department of computer science
Thesis language:	Czech

Anotace

Bakalářská práce se zaměřuje na řešení problému týkající se nabídky témat závěrečných prací na katedře informatiky Univerzity Palackého. Hlavním cílem práce je vytvořit webovou aplikaci, která umožní nabízet a přiřazovat daná témata studentům. Aplikace je naprogramována na frameworku ASP.NET a poskytuje uživatelům přihlášení pomocí již existujících univerzitních účtů. Výsledkem práce je funkční webová aplikace, která může nahradit stávající řešení.

Synopsis

This bachelor thesis focuses on solving the problem of offering topics of diploma theses at the Department of Computer Science of Palacký University. The main goal of the thesis is to create a web application that allows to offer and assign topics to students. The application is programmed on the ASP.NET framework and allows users to log in using existing university accounts. The result of the thesis is a functional web application that can replace the existing solution.

Klíčová slova: webová aplikace; nabídka závěrečných prací; ASP.NET

Keywords: web application; offering topics of diploma theses; ASP.NET

Děkuji panu doc. Mgr. Janu Outratovi, Ph.D. za veškeré rady, připomínky a odborné vedení bakalářské práce.

Odevzdáním tohoto textu jeho autor/ka místopřísežně prohlašuje, že celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

Obsah

1	Úvod	8
1.1	Zadání práce	8
1.2	Existující řešení	8
1.3	Funkcionalita aplikace	9
1.3.1	Témata	10
1.3.2	Autentizace	10
1.3.3	Notifikace	10
1.3.4	Lokalizace	10
1.3.5	Funkcionalita z pohledu uživatelů	10
2	Použité technologie	12
2.1	C#	12
2.2	ASP.NET	12
2.3	PostgreSQL	12
2.4	Entity Framework	12
2.5	Razor views	12
2.6	HTML	13
2.7	CSS	13
2.8	Javascript	13
2.9	Bootstrap	13
2.10	Použité knihovny	13
3	Programátorská příručka	14
3.1	Architektura aplikace	14
3.1.1	MVC	14
3.1.2	Dependency injection	14
3.1.3	Struktura aplikace	15
3.1.4	Struktura databáze a popis modelů	16
3.2	Autentizace a Autorizace	18
3.2.1	Autentizace uživatelů mimo Univerzitu Palackého	18
3.2.2	Autentizace uživatelů z Univerzity Palackého	19
3.2.3	Využití služeb univerzity pro přihlašování	19
3.2.4	ASP.NET Identity	20
3.2.5	Služba UpolAuthentication	20
3.2.6	Role	21
3.3	Notifikace	22
3.3.1	Odesílání notifikací	22
3.3.2	Nahrazování klíčových slov	23
3.4	Posílání e-mailů	23
3.5	Diskuze	23
3.6	Přílohy	24
3.7	Zájem o téma	24
3.8	Lokalizace	25

3.9	Konfigurace aplikace	25
4	Uživatelská příručka	26
4.1	Lokalizace	26
4.2	Registrace, přihlášení a nastavení účtů	26
4.2.1	Registrace	26
4.2.2	Přihlášení	26
4.2.3	Zapomenuté heslo	27
4.2.4	Nastavení uživatelského účtu	27
4.3	Témata	28
4.3.1	Operační tlačítka	28
4.3.2	Přiřazení tématu	28
4.3.3	Vyhledávání a filtrování témat	28
4.3.4	Vytváření, editace a navrhování témat	29
4.3.5	Navrhování témat závěrečných prací	29
4.3.6	Detail tématu	30
4.3.7	Diskuze	30
4.3.8	Moje témata	31
4.4	Skupiny	31
4.5	Administrace	32
4.5.1	Uživatelé	32
4.5.2	Programy	32
4.5.3	Notifikace	32
5	Rozšíření aplikace	33
5.1	Napojení na SSO univerzity – Shibboleth	33
5.2	Odevzdávání vypracovaných témat	33
	Závěr	34
	Conclusions	35
	A Obsah přiloženého datového média	36
	Literatura	37

Seznam obrázků

1	Aktuální řešení nabídky témat	9
2	Diagram databáze	17
3	Diagram popisující chování služby UpolAuthentication	21
4	Formulář pro přihlášení uživatelů	27
5	Odebírání notifikací v nastavení	28
6	Prohlížení a vyhledávání témat	29
7	Detail tématu z pohledu studenta	30
8	Stránka „Moje témata“	31
9	Detail skupiny z pohledu vyučujícího	31
10	Správa uživatelů v administraci	32

Seznam zdrojových kódů

1	Funkce CommentsDepths v jazyce C#	24
---	---	----

1 Úvod

Každého studenta na vysoké škole čeká výběr závěrečné práce a zda se student rozhodne vypracovat navrhnuté téma od vyučujícího, z praxe nebo své, tak webová aplikace, která vznikla s touto prací, se mu pokusí s tímto procesem pomoci.

Práce pojednává o vzniku webové aplikace, která se snaží výběr závěrečné práce co nejvíce zpříjemnit, jak pro studenty, tak i vyučující na katedře informatiky Univerzity Palackého. Zároveň ale nezapomíná na externí uživatele, kterými mohou být například zástupci firem či zástupci státních organizací, kterým aplikace umožní navrhovat témata závěrečných prací.

1.1 Zadání práce

Student vytvoří webovou aplikaci pro nabídku témat závěrečných a jiných prací studentů na Katedře informatiky PřF UP. Témata do ní budou moci samostatně zadávat (a editovat) nejen vyučující katedry, ale i kdokoli jiný, např. zástupci firem, s přístupem. Tato druhá témata bude moci vyučující klasifikovat (bakalářská vs. diplomová práce, vhodnost pro studijní program apod.), "adoptovat" (jako potenciální vedoucí práce) a "svoje" témata pak přidělit studentům. Student bude moci vyznačit svůj zájem o téma.

- text, aktivace/deaktivace a přílohy témat (zadavateli), komentáře/diskuze k tématům (zadavateli i studenty), filtrace a seskupování zobrazení a vyhledávání témat
- klasifikace: výše plus např. projekt, úloha apod. v rámci předmětu aj.
- notifikace o zadání (editaci), aktivaci/deaktivaci, komentování, projevení zájmu o a přidělení tématu (zájemcům, emailem)

1.2 Existující řešení

Jako existující řešení se nabízí vzít v úvahu to aktuální, které katedra informatiky právě používá.

Témata bakalářských prací pro akademický rok 2022/2023

Výběr tématu probíhá ve 4. semestru v následujících krocích:

1. Na začátku března jsou na webu katedry zveřejněna témata bakalářských prací.
2. Do konce dubna si studenti zvolí vedoucího práce a téma. Zpravidla si volí téma vypsání vyučujícím katedry informatiky, je možné si však vybrat i téma vlastní.
3. Do konce května vloží student do STAGu název tématu v českém a anglickém jazyce, anotaci a seznam literatury.
4. Ze STAGu pak vytiskne podklad pro zadání, nechá ho podepsat vedoucím práce a odevzdá sekretářce katedry.

Studenti by si měli vybírat téma vhodné pro svůj studijní program/speciálizaci. Studentům speciálizace Obecná informatika doporučujeme zvolit si téma některého z vyučujících katedry. Pokud nebude student vědět, jaké téma si vybrat, kontaktuje zástupce vedoucího katedry pro výuku.

Radek Janošník

Vizualizace komunikace virtuální sítě v rodině protokolů TCP/IP

Vhodné pro studenty programu/speciálizace: Programování a vývoj software Informační technologie Učitelství

Student navrhne a naprogramuje jednoduchý simulátor počítačové sítě, ve kterém bude přehlednou formou vizualizována komunikace pomocí protokolů z rodiny TCP/IP. Důležitou částí bude přehledné zobrazení komunikace v čase - krokování a zpomalení.

Miroslav Kolařík

Program na podporu výuky – téma již není volné

Vhodné pro studenty programu/speciálizace: Učitelství

Po vzájemné dohodě si student vybere vhodné téma, které zpracuje jako interaktivní výukový program (spolu s učebním textem adekvátního rozsahu).

Obrázek 1: Aktuální řešení nabídky témat

Aktuální řešení (obr. 1) nenabízí velkou funkcionalitu. Jedná se v podstatě o statickou stránku s nabízenými tématy vyučujícími, kde daná témata obsahují název, stručný popis a doporučené studijní programy.

Pro studenta, který by si vybíral své budoucí téma, může stránka působit nepřehledně. Jelikož například vidí témata, která nemusí být určena pro jeho studijní program nebo mezi všemi tématy se nachází i ta, která jsou již zabraná. Současné řešení také neposkytuje možnost vyhledávání, filtrování, souborových příloh, podrobnějšího popisu a diskuze.

Hlavní výhodou tohoto řešení je integrace přímo do webové stránky katedry a jeho jednoduchost.

1.3 Funkcionalita aplikace

Tato podkapitola se zaměřuje na funkcionalitu, kterou by měla výsledná webová aplikace nabízet. Požadovaná funkcionalita zejména vyplývá ze samotného zadání práce, ale samotné zadání nepopisuje všechny funkce, které by byly vhodné, aby aplikace nabízela. Rozhodně není zbytečné mít ucelený popis všech požadavků na aplikaci. Poskytne čtenáři lepší přehled o vyvíjené aplikaci a může také sloužit jako seznam požadavků pro ověření funkcionality výsledné aplikace.

1.3.1 Témata

Ačkoliv hlavním účelem této práce je vytvořit systém pro nabídku témat závěrečných prací, tak aplikace bude podporovat i správu úkolů a projektů týkajících se předmětů vyučovaných na katedře informatiky. Všechna témata budou moci být vytvářena, editována, mazána, prohlížena, vyhledávána a filtrována. Témata závěrečných prací budou klasifikována buď na bakalářské, nebo diplomové práce a bude k nim možno přiřadit doporučený studijní program. Témata týkající se předmětů budou moci být klasifikována na projekty a úkoly. Ke všem tématům bude umožněno nahrávat souborové přílohy a přihlášeným uživatelům také bude zpřístupněna sekce diskuze pod každým tématem.

1.3.2 Autentizace

Aplikace umožní studentům a vyučujícím přihlašovat se pomocí univerzitních přihlašovacích údajů, tím se myslí údaje, které uživatelé používají při přihlašování do systému STAG nebo k doméně Univerzity Palackého.

1.3.3 Notifikace

Aplikace bude podporovat zasílání notifikací e-mailem uživatelům, kteří se přihlásí k jejich odběru. Uživatelé mohou být notifikováni při:

- Změně tématu, se kterým je uživatel v nějakém vztahu
- Přiřazení tématu
- Schválení navrhnutého tématu
- Novém komentáři
- Projevení zájmu u tématu

1.3.4 Lokalizace

Aplikace bude lokalizována do českého a anglického jazyka.

1.3.5 Funkcionalita z pohledu uživatelů

Funkcionalitu aplikace lze rozdělit na základě toho, jaký uživatel ji bude používat.

- Všichni uživatelé
 - Prohlížení, vyhledávání a filtrování témat závěrečných prací, úkolů a projektů
 - Prohlížet a stahovat souborové přílohy u témat
- Přihlášení uživatelé

- Prohlížet a přidávat příspěvky do diskuzí u jednotlivých témat
- Mazat vlastní příspěvky v diskuzích
- Odebírat notifikace
- Změnit e-mailovou adresu a jméno
- Externí uživatelé
 - Navrhovat témata závěrečných prací a projektů
 - Změnit své heslo
- Studenti
 - Navrhovat témata závěrečných prací a projektů
 - Projevit zájem o témata
 - Vytvářet anonymní příspěvky v diskuzích
- Vyučující
 - Spravovat svá témata závěrečných prací, úkolů a projektů
 - Klasifikovat a adoptovat navrhovaná témata externími uživateli a studenty
 - Mazat jakékoliv příspěvky v diskuzích
- Administrátoři
 - Přiřazovat role uživatelům a registrovat uživatele
 - Editovat text notifikací
 - Spravovat studijní programy
 - Spravovat veškerá témata
 - Mazat jakékoliv příspěvky v diskuzích

2 Použité technologie

Tato kapitola popisuje technologie, které byly použity pro naprogramování výsledné webové aplikace.

2.1 C#

C#[1] je moderní vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý společností Microsoft. Používá se zejména pro vývoj desktopových aplikací, her a webových aplikací běžících na platformě .NET.

.NET[2] je vývojová platforma složená z nástrojů, programovacích jazyků a knihoven sloužící k vývoji různých aplikací.

Aplikace využívá C# ve verzi 11 na multiplatformní open-source platformě .NET 7. Tyto specifické verze aplikace využívá z toho důvodu, že byly v době vývoje nejnovější.

2.2 ASP.NET

ASP.NET[3] je multiplatformní open-source framework pro vývoj webových aplikací s .NET a C#. Rozšiřuje .NET o další nástroje a knihovny specifické pro vývoj webových aplikací.

ASP.NET poskytuje více technologií pro vývoj webových aplikací, výsledná aplikace využívá ASP.NET MVC a Razor Pages.

2.3 PostgreSQL

PostgreSQL je open-source relační databázový systém. V aplikaci slouží pro perzistentní ukládání dat, s kterými aplikace pracuje.

2.4 Entity Framework

Entity Framework[4], je open-source ORM (object-database mapper) pro platformu .NET.

ORM[5] je programovací technika, která umožňuje automatickou konverzi dat mezi relační databází a objektově orientovaným jazykem.

Díky ORM není nutné v kódu aplikace psát složité SQL příkazy, ale místo toho umožňuje používat objekty a pracovat s nimi pomocí C# příkazů.

2.5 Razor views

V ASP.NET MVC[6] se typicky používají pro zobrazení soubory s příponou *.cshtml*. Tyto soubory obsahují značkovací jazyk HTML rozšířený o Razor kód. Razor kód umožňuje v souborech View využívat jazyk C#. Pokud si uživatel vyžádá nějakou stránku, tak se nejprve na straně serveru vykoná Razor kód a odešle se finální HTML soubor.

2.6 HTML

HTML je značkovací jazyk sloužící k vytváření webových stránek. Dodává textu, zobrazovaném na dané stránce, strukturu a sémantický význam.

2.7 CSS

Kaskádové styly slouží k úpravě vzhledu stránky napsané v jazyce HTML.

2.8 Javascript

Javascript je skriptovací jazyk, který slouží k vykonávání kódu na straně koncového uživatele. Finální aplikace využívá Javascript pro asynchronní volání kódu na straně serveru a upravování vzhledu stránky na základě akcí uživatele.

2.9 Bootstrap

Bootstrap je framework, pomocí kterého lze rychle a snadno vytvořit responzivní vzhled webových stránek. Obsahuje spoustu předpřipravených komponent, které lze využít pomocí atributu *class* v HTML elementech. Aplikace využívá verzi Bootstrap 5.3 na celkový vzhled aplikace, v době vývoje se jednalo o nejaktuálnější verzi.

2.10 Použité knihovny

- jQuery
 - Knihovna, která usnadňuje práci s Javascript kódem. Je také vyžadována pro funkčnost Bootstrap.
 - Licence: MIT
- dselect
 - Knihovna, která umožňuje vyhledávání v rozbalovacích seznamech. Knihovnu jsem mírně rozšířil k potřebám aplikace. [7]
 - Licence: MIT
- System.DirectoryServices.Protocol
 - Knihovna součástí .NET, která umožňuje práci s protokolem LDAP.
 - Licence: MIT
- Npgsql
 - Slouží k propojení Entity Framework a databáze PostgreSQL.
 - Licence: PostgreSQL

3 Programátorská příručka

Tato kapitola se podrobně zabývá architekturou, strukturou a funkcionalitou celé aplikace. Má za cíl čtenáře seznámit s vnitřním fungováním aplikace a může také sloužit jako úvodní návod pro programátora, který se rozhodne kód aplikace rozšířit.

3.1 Architektura aplikace

Architektura hraje zásadní roli při vyvíjení aplikací. Zjednodušeně stanovuje určitá pravidla, která když se budou dodržovat, tak usnadní budoucí vývoj a udrží řád ve struktuře a kódu aplikace.

3.1.1 MVC

ASP.NET MVC využívá jako hlavní návrhový vzor architektury MVC[8] (Model-View-Controller). Tento vzor rozděluje aplikaci do tří částí, kde každá část má svůj vlastní účel a zodpovědnost.

Model

Model má na starost data, zejména jejich získávání a ukládání z datového úložiště, v této aplikaci z databáze.

View

View má na starost uživatelské rozhraní. Vytváří výsledné webové stránky, které jsou odesílány koncovým uživatelům.

Controller

Controller řídí tok dat mezi Modelem a View. Dostává HTTP dotazy, na které většinou reaguje tím, že z Modelu vybere žádaná data a odešle je příslušnému View.

3.1.2 Dependency injection

Aplikace využívá návrhový vzor Dependency injection (česky vkládání závislostí). Tento vzor odděluje závislosti od třídy a umožňuje díky tomu snadno změnit závislost, bez potřeby úpravy samotné třídy. V třídě se nevytváří nové instance závislostí (pomocí operátoru new), ale instance jsou do dané třídy vkládány při vytvoření. To znamená, že třída využívající DI, je nezávislá na konkrétní implementaci závislostí a může pracovat s libovolnou implementací, která splňuje dané rozhraní.

ASP.NET MVC[8] nabízí vestavěnou podporu pro tento vzor a využívá se následovně:

1. Vytvoří se služba/třída, zpravidla implementující rozhraní popisující danou funkčnost
2. Tato služba se zaregistruje v hlavním souboru *Program.cs* s jedním z následujících životních cyklů:
 - Transient: služba se vytvoří při každé žádosti, vhodné pro jednoduché bezstavové služby
 - Scoped: služba se vytvoří jednou pro každé připojení klienta
 - Singleton: služba se vytvoří pouze jednou (první žádost) a poté aplikace pracuje s jedinou vytvořenou instancí
3. Využití dané služby lze rozdělit na dva způsoby, na základě typu souboru:
 - Controller: závislost se přidá přes konstruktor třídy
 - View: přidáním direktivy *@inject* na začátek souboru

3.1.3 Struktura aplikace

Následující seznam se zaměřuje na adresářovou strukturu projektu. Jsou zde zmíněné důležité adresáře s popisem jejich účelu.

- *Models/*

Obsahuje soubory reprezentující modely. Modely, které aplikace používá, jsou popsány v následující podkapitole [3.1.4](#).

- *Views/*

Veškeré View soubory, které aplikace obsahuje se nachází v tomto adresáři. Pro každý Controller existuje stejnojmenný podadresář, který obsahuje soubory s příponou *.cshtml*. Jedná se o soubory View, se kterými Controller pracuje.

V adresáři *Views/* se nachází zajímavý podadresář *Shared/*. Ten obsahuje sdílené Views, Partial Views (zobrazení, která lze vkládat do jiných zobrazení) a důležitý soubor *__Layout.cshtml*, který slouží jako šablona pro všechny ostatní Views.

- *Controllers/*

Aplikace se skládá z následujících 4 Controllerů:

- HomeController

Stará se o úvodní stranu a přepínání jazyku lokalizace.

- TopicController

Jedná se o klíčovou část celé aplikace. Obsahuje funkcionalitu, která je zodpovědná za všechny operace týkající se témat.

- `GroupController`
Obsluhuje vytváření, editaci, mazání a zobrazování skupin (využití skupin je popsáno v následující podkapitole 3.1.4).
- `AdministrationController`
Obsluhuje veškerou administraci stránky.
- *wwwroot/*
Obsahuje statický obsah webové stránky (CSS a Javascript soubory, obrázky, přílohy témat).
- *Areas/Identity/*
Soubory týkající se ASP.NET Identity. Tento systém je zodpovědný za autentizaci a autorizaci.
- *Services/*
Služby a statické třídy, které aplikace využívá.
- *Resources/*
Soubory sloužící pro lokalizaci aplikace.
- *Data/*
Tento adresář obsahuje důležitý soubor *ApplicationDbContext.cs*, ve kterém se definuje interakce mezi aplikací a databází. Také se zde nachází soubor *Seed.cs* sloužící k vytvoření potřebné struktury databáze a záznamů.

3.1.4 Struktura databáze a popis modelů

Tato podkapitola popisuje s jakými modely aplikace pracuje a k čemu vlastně slouží.

Typický model vypadá jako běžná třída v jazyce C#, která se skládá z properties, česky vlastností (vlastnost lze chápat jako název a datový typ sloupce v tabulce). Tyto vlastnosti mohou být rozšířeny o další informace, sloužící například k validaci.

Při vývoji aplikace jsem postupoval „Code First“ přístupem, to znamená, že se databáze a její tabulky vytvářely pomocí migrací z naprogramovaných modelů. To ale neznamená, že se každý model rovná jedné tabulce v databázi. K vytvoření tabulky je potřeba přidat záznam do souboru *ApplicationDbContext.cs*.

Ačkoliv tabulky v databázi používají jako cizí klíče pouze hodnoty Id, tak Entity Framework umožňuje použít navigační vlastnosti u modelů. Pomocí těchto navigačních vlastností je velmi ulehčen přístup k odkazovaným objektům.

Pro relace, které jsou ve vztahu M:N, je nutné explicitně vytvořit dekompoziční modely[6].

Pro lepší představu, jak celkový model databáze vypadá, slouží následující diagram (obr. 2).



Obrázek 2: Diagram databáze

Témata

K uchovávání témat slouží model *Topic*. To ale není jediný model, který se témat týká. Model *Attachment* uchovává informace o nahraných přílohách a model *Comment* slouží k příspěvkům v diskuzích. Pro ukládání informací ohledně zájmu o téma slouží model *UserInterestedTopic*.

Hodnota *Type* v modelu *Topic* označuje, zda se jedná o téma závěrečné práce, projektu nebo úkolu.

Programy

Aplikace podporuje přidávání doporučených studijních programů k tématům závěrečných prací. K této funkcionalitě slouží Model *Programme*, který reprezentuje studijní program a model *TopicRecommendedProgram*, který zodpovídá za přiřazení studijních programů k tématům. Hodnota *Type* u modelu *Programme* označuje, zda se jedná o bakalářský nebo magisterský program.

Skupiny

Model *Group* slouží ke shlukování témat, které spolu nějak souvisí. Mezi základní skupiny, které aplikace využívá jsou Bakalářské, Diplomové a Nezařazené závěrečné práce. Další skupiny mohou vytvářet vyučující zejména pro předměty, které vyučují.

Uživatelé a role

Aplikace využívá následující modely *ApplicationUser*, *ApplicationRole* a *ApplicationUserRole*. Tyto modely dědí z předvytvořených modelů systému ASP.NET Identity za účelem rozšíření daného modelu pro požadavky aplikace. Starají se o správu uživatelů a rolí.

Dále ASP Identity využívá následující modely: *AspNetRoleClaims*, *AspNetUserClaims*, *AspNetUserLogins*, *AspNetUserTokens*, které aplikace nevyužívá, ale pro zachování funkčnosti musí existovat.

Notifikace

Notification reprezentuje notifikace v aplikaci, tyto notifikace mají vlastní tabulku v databázi z toho důvodu, aby bylo možné upravovat jejich text. Pokud se uživatel bude chtít přihlásit k odběru některé notifikace, tak se tento údaj zaznamená do *UserSubscribedNotification*.

Migrace

Pro správnou funkčnost migrací je také potřeba, aby v databázi existovala tabulka *EFMigrationsHistory*, která si uchovává informace o aplikovaných migracích.

3.2 Autentizace a Autorizace

Autentizaci lze pro účely této aplikace rozdělit na dvě skupiny: Autentizace uživatelů mimo Univerzitu Palackého a Autentizace uživatelů z Univerzity Palackého.

3.2.1 Autentizace uživatelů mimo Univerzitu Palackého

Tito uživatelé jsou v aplikaci chápáni zejména jako externí uživatelé. Pro autentizaci musí nejprve administrátor vytvořit jejich účet a až poté se mohou

přihlašovat. Mohou také ověřit svou e-mailovou adresu pro prokázání, že se skutečně jedná o jejich e-mail. Proces této autentizace je zcela nezávislý na službách třetích stran.

- Při registraci si aplikace uloží veškeré zadané údaje včetně hesla.
- Při přihlašování aplikace vyhledá v databázi daného uživatele na základě přihlašovacího jména (e-mailu) a zkontroluje, zda zadané heslo odpovídá uloženému heslu.

3.2.2 Autentizace uživatelů z Univerzity Palackého

Tito uživatelé jsou studenti nebo vyučující na Univerzitě Palackého. Rovnou se přihlašují pomocí stejných údajů, které používají pro autentizaci v doméně Univerzity Palackého. Přihlašování těchto uživatelů je závislé na službách univerzity.

Aplikace nemá žádné údaje o konkrétních univerzitních uživateli bez jejich prvního přihlášení. Z tohoto důvodu není například možné přiřadit studenta, který se nikdy nepřihlásil, k některému tématu.

- Při prvním přihlášení se zkontroluje, zda uživatel zadal správné přihlašovací údaje pomocí Active Directory Univerzity Palackého. Pokud jsou dané údaje správné, tak aplikace získá osobní informace o uživateli (jméno, příjmení, tituly, e-mail) a také zjistí, zda-li je uživatel student nebo vyučující a přiřadí mu odpovídající roli. Tyto informace se získávají pomocí Stag Services. Uživatel může používat pro oba systémy různá hesla, proto může aplikace vyžádat při prvním přihlášení také heslo k informačnímu systému STAG.
- Při dalším přihlašování se už pouze kontroluje správnost přihlašovacích údajů pomocí Active Directory.

3.2.3 Využití služeb univerzity pro přihlašování

Tato podkapitola popisuje služby, které se využívají při autentizaci uživatelů z Univerzity Palackého.

Active directory

Active Directory je adresářová služba navrhnutá společností Microsoft. Aplikace komunikuje s Active Directory Univerzity Palackého za účelem autentizace uživatelů.

Ke komunikaci aplikace používá protokol LDAP[9], který slouží k ukládání a přístupu k datům na adresářovém serveru.

Stag Services

STAG je univerzitní informační systém. Aplikace z něj získá roli (student nebo vyučující) a osobní informace uživatele při jeho prvním přihlášení.

Ke komunikaci aplikace používá REST API přístupné pouze z vnitřní sítě univerzity, které poskytuje `stagservices.upol.cz`. Z tohoto důvodu je nutné, aby aplikace byla připojena k vnitřní síti Univerzity Palackého.

Uživatelé pro přihlášení využívají přihlašovací jméno k doméně univerzity. Konkrétního uživatele lze identifikovat v systému STAG pomocí atributu *ExternalLogin*, který je stejný jako přihlašovací jméno v doméně. Aplikace se dotazuje Stag Services (<https://stagservices.upol.cz/ws/services/rest2/>) na následující operace:

- `users/getOsobniCislaByExternalLogin?login=`
- `users/getUcitIdnoByExternalLogin?externalLogin=`
- `student/getStudentInfo?osCislo=`
- `ucitel/getUcitelInfo?ucitIdno=`

3.2.4 ASP.NET Identity

Jedná se o systém vyvinutý společností Microsoft pro autentizaci a autorizaci uživatelů v aplikacích postavených na frameworku ASP.NET[10]. Tento systém je funkční sám o sobě, ale lze ho jednoduše rozšířit o další funkcionalitu.

Veškeré soubory, ze kterých se Identity skládá, se nacházejí v adresáři *Areas/Identity/Pages/Account*. Je důležité podotknout, že tento systém není postaven na technologii ASP.NET MVC, ale používá technologii Razor Pages.

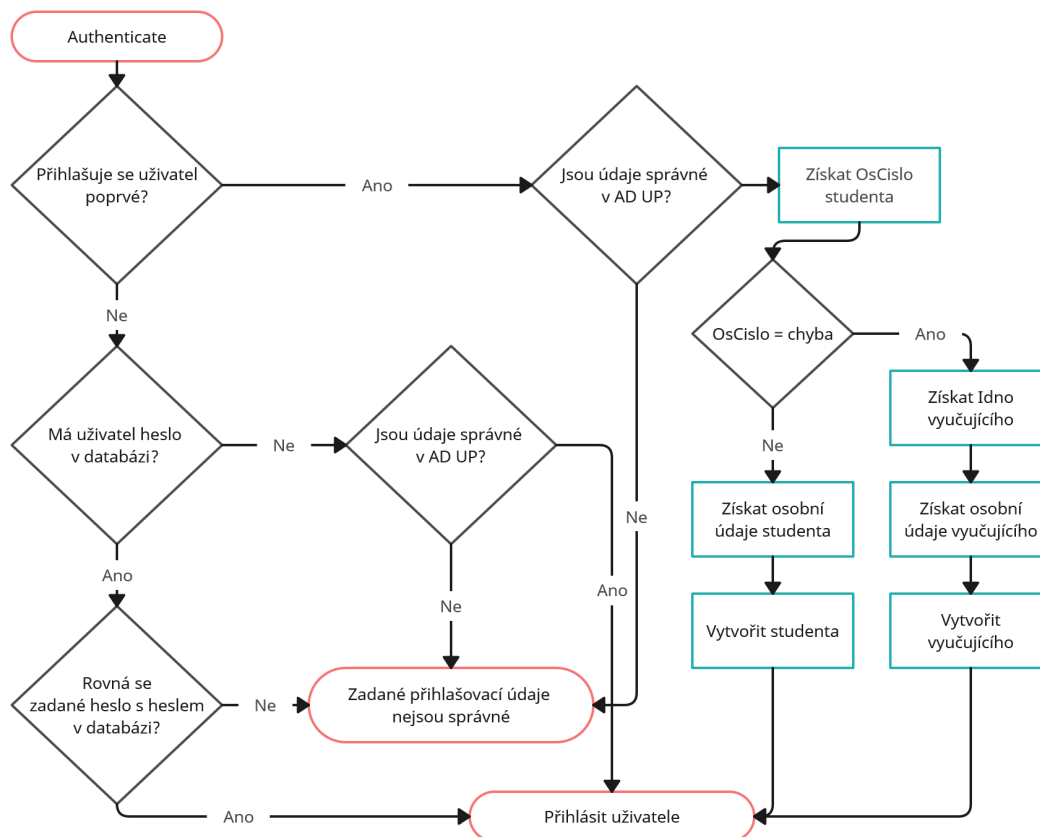
Aplikace nepotřebuje veškerou funkcionalitu, kterou systém Identity nabízí. Proto jsou v souboru *ApplicationDbContext.cs* určité vlastnosti modelů ignorovány a některé stránky smazány. Aplikace také ignoruje celé modely, které jsou zmíněny v podkapitole Uživatelé a role 3.1.4.

3.2.5 Služba UpolAuthentication

Hlavní rozšíření systému Identity se nachází v souboru *Login.cshtml.cs*, kde je využita služba *UpolAuthentication* implementující *IAuthentication*. Tato služba řídí celý proces autentizace a vytváření profilů univerzitních uživatelů.

Jedinou veřejnou metodou služby je metoda *Authenticate*, která vyžaduje čtyři parametry: přihlašovací jméno, heslo k Active Directory, heslo ke STAGu a zda si uživatel přeje zůstat přihlášen po zavření webové stránky.

Následující diagram (obr. 3) abstraktně popisuje chování služby.



Obrázek 3: Diagram popisující chování služby UpolAuthentication

3.2.6 Role

Aplikace pro autorizaci využívá na pozadí následující role. Administrátor je od těchto rolí distancován a rozhoduje pouze o tom, zda je uživatel student, vyučující, administrátor nebo externí uživatel. Předpony Any umožní uživateli pracovat s jakýmkoliv položkami, které se týkají dané role.

- Administrator, Teacher, Student, External

Jedná se o role, které slouží ke zjednodušení systému. Role Administrator zpřístupní sekci administrace a umožní editování popisu u základních skupin.

- Topic, AnyTopic

Umožní vytvářet, editovat, klasifikovat/adoptovat a mazat témata, úkoly a projekty.

- ProposeTopic

Umožní uživateli navrhnout téma závěrečné práce nebo projektu.

- InterestTopic
Umožní uživateli projevit zájem o téma.
- AssignedTopic, SuperviseTopic
Umožní, aby uživatel byl označen jako přiřazený student nebo vedoucí tématu.
- Group, AnyGroup
Umožní uživateli vytvářet, editovat a mazat skupiny.
- Comment, AnyComment
Umožní uživateli vidět diskuzi a také přidávat a mazat vlastní příspěvky.
- AnonymousComment
Umožní uživateli přidávat anonymní příspěvky do diskuze. Pouze administrátor a vyučující uvidí autora příspěvku.
- Attachment
Umožní uživateli nahrávat a mazat přílohy.

Pro zjednodušení práce s rolemi slouží statická třída *RoleHelper*, která přiřadí uživateli všechny role na základě čtyř hlavních rolí.

3.3 Notifikace

Notifikace jsou v aplikaci implementovány pomocí služby *NotificationManager*, která implementuje rozhraní *INotificationManager*.

3.3.1 Odesílání notifikací

Služba po zavolání metody na odeslání některé notifikace postupuje následovně:

1. Získá objekt notifikace z databáze.
2. Zvolí potenciální uživatele, kteří by mohli dostat danou notifikaci.
3. Odstraní se autor akce, kvůli které se notifikace odesílá.
4. Zavolá se metoda *SendNotification*, která:
5. Vytvoří řetězce subject a text, které reprezentují předmět a text notifikace. Dané řetězce vzniknou dle hodnot z databáze a nahrazením klíčových slov.
6. Odeberou se uživatelé, kterým má přijít notifikace dle toho, jestli jsou přihlášení k odběru dané notifikace.
7. Notifikace se odešle uživatelům e-mailem.

3.3.2 Nahrazování klíčových slov

Notifikace mohou obsahovat klíčová slova v hranatých závorkách. Tato klíčová slova jsou poté nahrazována podle skutečných hodnot, které se dané notifikace týkají. K tomuto procesu slouží metoda *NotificationSubstitution*.

3.4 Posílání e-mailů

Odesílání všech e-mailů má na starost třída *SmtplibEmailSender*, která implementuje rozhraní *IMailSender* s metodou *SendEmailAsync*, která je zodpovědná za odeslání e-mailů. Odesílání e-mailů v této metodě je řešeno pomocí SMTP protokolu použitím třídy *SmtplibClient*.

Aplikace odesílá dva druhy e-mailů: notifikace a e-maily týkající se správy uživatelského účtu (např. při zapomenutí hesla, potvrzení e-mailové adresy).

3.5 Diskuze

K práci s příspěvky slouží metody *AddComment* a *DeleteComment* nacházející se v *TopicController.cs*.

AddComment

Uživatel může přidat příspěvek do diskuze dvěma způsoby. Oba tyto způsoby využívají pro přidání příspěvku PartialView *AddComment.cshtml*.

- Nový příspěvek
Při přidání nového příspěvku se nastaví hodnota *ParentCommentId* na null.
- Odpověď na příspěvek
Při odpovědi se nastaví hodnota *ParentCommentId* na odpovídaný příspěvek.

Při přidávání anonymních příspěvků se nastaví hodnota *Anonymous* na pravdu. Díky této hodnotě se při vykreslování diskuze skryje autorovo jméno.

DeleteComment

Tato metoda využívá statickou metodu *DeleteComment* v souboru *CommentHelper.cs*, která má na starost mazání příspěvků.

- Mazání příspěvků bez odpovědí
Tyto příspěvky se celé vymažou.
- Mazání příspěvků s odpovědí
Text těchto příspěvků se vymaže a příspěvek se nastaví na anonymní.

Pokud smazaný příspěvek neměl odpověď a byl odpovědí na příspěvek (ParentComment), který již byl smazán, tak se metoda *DeleteComment* zavolá rekurzivně na ParentComment. Tím se docílí celkového smazání, již smazaných příspěvků, které nemají žádnou nesmazanou odpověď.

Vykreslení diskuze

Jelikož příspěvky mohou odpovídat na jiné příspěvky, je potřeba dané příspěvky vykreslovat tak, aby bylo možné přiřadit příspěvek a jeho odpověď nezaměnitelně k sobě. K tomuto aplikace využívá odsazení příspěvků, pomocí jejich hloubky (zanořenost příspěvku), která se počítá pomocí rekurzivní funkce *CommentsDepths* v souboru *Details.cshtml*.

```
1 private static IEnumerable<Comment> CommentsDepths (IEnumerable<  
    Comment> rootComments, int depth)  
2 {  
3     foreach (var comment in rootComments)  
4     {  
5         comment.Depth = depth;  
6         yield return comment;  
7         foreach (var r in CommentsDepths(comment.Replies, depth + 1))  
8         {  
9             yield return r;  
10        }  
11    }  
12 }
```

Zdrojový kód 1: Funkce CommentsDepths v jazyce C#

Každý příspěvek obsahuje tlačítko „Odpovědět“, které po stisknutí zobrazí odpovídající skrytý formulář pro odpověď pomocí Javascript kódu.

3.6 Přílohy

Všechny přílohy se ukládají na fyzické úložiště do adresáře *wwwroot/files/Id-tématu/Název-přílohy* a zároveň se záznam o souboru uloží do databáze. Metoda *CreateFiles* je zodpovědná za vytváření příloh. Metoda *DeleteAttachment* slouží k mazání jedné přílohy a je volána asynchronně. Tyto metody se nachází v souboru *TopicController.cs*.

3.7 Zájem o téma

Metoda *Interest* v souboru *TopicController.cs* je volána asynchronně, pokud student projeví nebo odebere zájem o téma.

Při projevení zájmu metoda přidá záznam do tabulky *UserInterestedTopic* s aktuálním časovým údajem a odešle notifikaci. Při odebrání zájmu metoda vymaže daný záznam.

3.8 Lokalizace

Aplikace podporuje dva jazyky – češtinu a angličtinu. Témata, skupiny a celkově veškerá datová pole, která se v životním cyklu aplikace vytvářejí, obsahují textový vstup jak pro češtinu, tak angličtinu. Všechny ostatní statické řetězce jsou lokalizovány pomocí souborů v adresáři *Resources/*, které pracují na bázi klíč:hodnota.

Aplikace nepředpokládá rozšíření o další jazyk. Kvůli tohoto faktu je lokalizace provedena jednodušeji na úkor obtížnosti přidání nového jazyka.

Pro použití lokalizace se ve View souborech používá *IHtmlLocalizer* a v Controllerech *IStringLocalizer*. Instance se poté používají tak, že na zadané klíčové slovo vrátí hodnotu z příslušného souboru.

Změna jazyku je řešena pomocí Partial View *__Culture.cshtml* a metody *CultureManagement* v *HomeController.cs*. Informace o zvoleném jazyku se ukládají jako soubory cookie.

Pokud datové pole nebo statický řetězec má překlad pouze v jednom jazyku, použije se ten, ačkoliv nemusí odpovídat nastavené lokalizaci.

3.9 Konfigurace aplikace

Konfigurace aplikace se nachází v souboru *appsettings.json*.

- *ConnectionString* slouží k připojení k databázovému serveru.
- *EmailSettings* slouží ke konfiguraci SMTP serveru.
- *UpolActiveDirectory* slouží k nastavení adresy Active Directory serveru university.
- *StagServices* slouží k nastavení URL adres k přístupu ke Stag Services.
- Pro správnou funkčnost příloh je také nutné nastavit vhodná oprávnění pro adresář *wwwroot/files/*.

4 Uživatelská příručka

Tato kapitola slouží jako uživatelská příručka. Popisuje hlavní funkce a základní ovládání webové aplikace.

4.1 Lokalizace

Při prvním přístupu k aplikaci se jazyk vybírá automaticky na základě používaného jazyka operačním systémem. Pro změnu jazyka aplikace slouží rozbalovací seznam v pravé části navigačního menu. Zvolení položky způsobí obnovení stránky, přepnutí lokalizace do zvoleného jazyka a daný výběr se uloží pro příští použití.

4.2 Registrace, přihlášení a nastavení účtů

Nepřihlášený uživatel vidí pouze odkaz k přihlášení na pravé straně navigačního menu (obr. 4). Přihlášený uživatel vidí místo daného odkazu své uživatelské jméno a možnost pro odhlášení se (obr. 5).

4.2.1 Registrace

Registrace slouží pro registrování uživatelů, kteří nemají univerzitní účty. V administraci stránky v záložce „Uživatelé“ se nachází akce „Vytvořit uživatele“, která umožňuje registrovat uživatele. Na této stránce se nachází formulář, ve kterém jsou povinná pole: e-mail, křestní jméno, příjmení a role. Pokud registrace proběhne bez chyby, registrovaný uživatel dostane e-mail obsahující přihlašovací údaje a odkaz pro potvrzení e-mailu.

4.2.2 Přihlášení

Při kliknutí na odkaz „Přihlásit“ aplikace přesměruje uživatele na stránku s formulářem sloužícím k přihlášení.

Přihlášení uživatele bez univerzitního účtu

Uživatel se přihlašuje pomocí přihlašovacích údajů z e-mailu. Případně si může tyto přihlašovací údaje změnit v nastavení účtu.

Přihlášení uživatele s univerzitním účtem

Pro úspěšné první přihlášení je potřeba, aby uživatel zadal stejné přihlašovací údaje, které používá pro přihlášení k doméně (Active Directory) Univerzity Palackého.

Přihlašovací jméno může zvolit buď jako své uživatelské jméno (např. martfi00) nebo uživatelské jméno@upol.cz (např. martfi00@upol.cz).

Pokud uživatel používá různá hesla pro Active Directory a informační systém STAG, tak aplikace také vyžádá heslo ke STAGu, aby mohla získat osobní údaje a zjistit zda-li je uživatel studentem nebo vyučujícím.

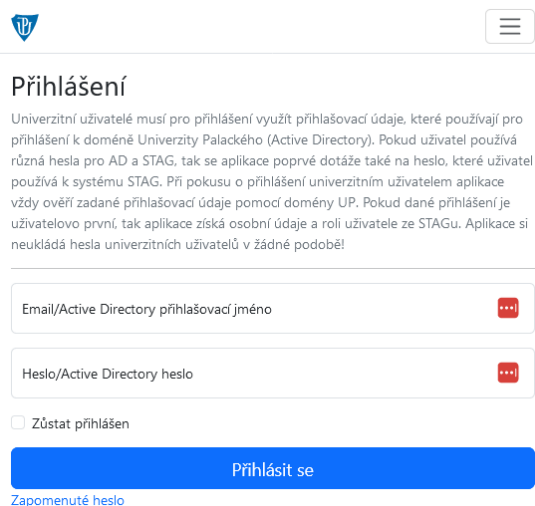
Pro další přihlášení může uživatel zvolit již zmíněné způsoby nebo svou e-mailovou adresu (např. filip.martinik01@upol.cz). Ve všech případech používá už jen heslo k doméně.

Zůstat přihlášen

Pokud uživatel potvrdí zaškrtnuté políčko „Zůstat přihlášen“, tak se jeho přihlášení uloží na následujících 14 dní.

4.2.3 Zapomenuté heslo

Na stránce s přihlášením se nachází odkaz „Zapomenuté heslo“. Tato položka slouží pro uživatele bez univerzitních účtů, kteří zapomněli své heslo. Po zvolení bude uživatel muset zadat e-mailovou adresu, na kterou bude zaslán odkaz pro obnovení hesla.



Přihlášení

Univerzitní uživatelé musí pro přihlášení využít přihlašovací údaje, které používají pro přihlášení k doméně Univerzity Palackého (Active Directory). Pokud uživatel používá různá hesla pro AD a STAG, tak se aplikace poprvé dotáže také na heslo, které uživatel používá k systému STAG. Při pokusu o přihlášení univerzitním uživatelem aplikace vždy ověří zadané přihlašovací údaje pomocí domény UP. Pokud dané přihlášení je uživatelské první, tak aplikace získá osobní údaje a roli uživatele ze STAGu. Aplikace si neukládá hesla univerzitních uživatelů v žádné podobě!

Email/Active Directory přihlašovací jméno

Heslo/Active Directory heslo

☐ Zůstat přihlášen

Přihlásit se

[Zapomenuté heslo](#)

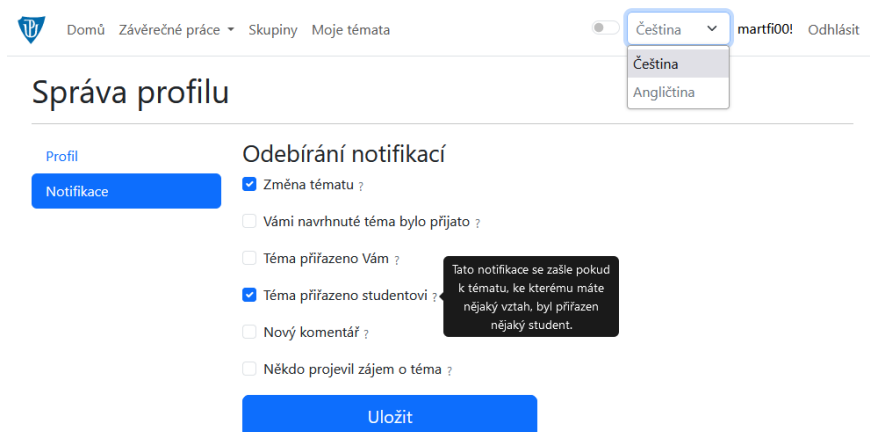
Obrázek 4: Formulář pro přihlášení uživatelů

4.2.4 Nastavení uživatelského účtu

Přihlášený uživatel může kliknout na své uživatelské jméno v navigační liště. Tento odkaz přesměruje uživatele do nastavení jeho účtu.

Univerzitní uživatel si v nastavení může změnit zobrazované jméno, e-mail a přihlásit se k odběru notifikací, které chce dostávat.

Uživatelé bez univerzitního přihlášení si mohou změnit zobrazované jméno, e-mail (přihlašovací jméno), heslo a také se mohou přihlašovat k odběru notifikací.



Obrázek 5: Odebírání notifikací v nastavení

4.3 Témata

K seznamu témat se lze dostat vícero způsoby. K tématům závěrečných prací se uživatel dostane po rozbalení nabídky „Závěrečné práce“ v navigační liště a zvolení jedné položky. K projektům a úkolům se uživatel dostane pomocí odkazu „Skupiny“ a vybrání konkrétní skupiny.

4.3.1 Operační tlačítka

Pod hlavním nadpisem skupiny se nachází operační tlačítka. Uživatel uvidí pouze taková, která jsou mu jeho rolí zpřístupněná. Tato tlačítka mohou být: Navrhnout téma/projekt, Vytvořit téma/projekt/úkol, Editovat skupinu. Po kliknutí na dané tlačítko aplikace uživatele přesměruje na stránku umožňující danou akci.

4.3.2 Přiřazení tématu

Univerzitní uživatel se musí alespoň jednou přihlásit, aby bylo možné jej přiřadit jako vedoucího/studenta k nějakému tématu. Bez prvního přihlášení nemá aplikace o daném uživateli žádné informace.

4.3.3 Vyhledávání a filtrování témat

Uživatel může využít následující prvky pro vyhledávání témat:

- Vyhledávací pole hledá témata, která obsahují v názvu vyhledávaný výraz. Také vyhledávací výraz může obsahovat jméno vedoucího tématu.
- Zaškrtnutí políčko „Zahrnout zabanované“ slouží k zahrnutí témat, která mají přiřazeného studenta.
- Rozbalovací seznam s programy slouží k zobrazení témat, které mají jako doporučený studijní program právě ten vybraný.

- Rozbalovací seznam řazení slouží k seřazení vyhledaných témat.
- Vyučující mají také možnost zvolit „Zahrnout skryté“ nebo „Pouze navrhnuté“ pomocí zaškrťovacích políček.

Domů Závěrečné práce Skupiny Moje témata

Čeština martf00! Odhlásit

Bakalářské závěrečné práce

[Navrhnout téma](#)

1. Na začátku března jsou na webu katedry zveřejněna témata bakalářských prací.
2. Do konce dubna si studenti zvolí vedoucího práce a téma. Zpravidla si volí téma vypsané vyučujícím katedry informatiky, je možné si však vybrat i téma vlastní.
3. Do konce května vloží student do STAGu název tématu v českém a anglickém jazyce, anotaci a seznam literatury.
4. Ze STAGu pak vytiskne podklad pro zadání, nechá ho podepsat vedoucím práce a odevzdá sekretářce katedry.

Systém ☒ Zahrnout nedostupné Informatika - Vývoj software Řazení Vedoucího [Hledat](#)

doc. Mgr. Jan Outrata, Ph.D.
[Systém pro nabídku témat závěrečných prací na VŠ](#)

Přiřazené Informatika - Vývoj software 1

Student vytvoří webovou aplikaci pro nabídku témat závěrečných a jiných prací studentů na Katedře informatiky PÍF UP.

Obrázek 6: Prohlížení a vyhledávání témat

4.3.4 Vytváření, editace a navrhování témat

Při vytváření, editování nebo navrhování témat se uživateli zobrazí formulář, který obsahuje datová pole, která uživatel může vyplnit. Pole označená hvězdičkou jsou povinná. Pro nahrání přílohy může uživatel stisknout tlačítko „Procházet“, které otevře průzkumníka souborů pro výběr nahrávaných souborů.

4.3.5 Navrhování témat závěrečných prací

Pokud student nebo externí uživatel navrhne téma, tak dané téma není veřejně přístupné. Nejprve jej musí vyučující potvrdit a případně upravit. Uživatel, který téma navrhl, může dané téma upravovat do doby jeho přijetí některým vyučujícím.

K těmto navrhnutým tématům se lze dostat třemi způsoby. Vyučující může zvolit položku „Navrhnuté“ z navigačního menu nebo při prohlížení bakalářských/diplomových prací může zaškrtnout pole „Pouze navrhnuté“. Autor tématu jej uvidí v záložce „Moje témata“.

Vyučující může téma potvrdit tím, že jej označí za veřejné. Takové téma již bude veřejně přístupné a bude označeno jako téma bez vedoucího v příslušné skupině. Vyučující může také téma adoptovat tím, že se zvolí jako vedoucí tématu.

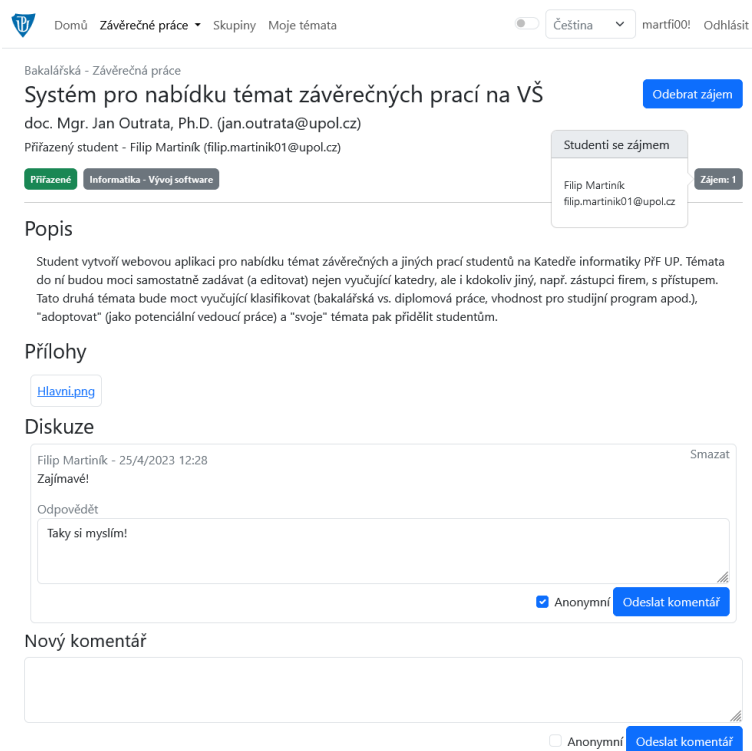
4.3.6 Detail tématu

Pro zobrazení detailu tématu musí uživatel kliknout na název tématu.

V zobrazení detailu tématu může uživatel, pokud má příslušná oprávnění, provést následující akce:

- Projevení/odstranění zájmu o téma
- Editovat téma

Přihlášený uživatel má možnost se podívat na studenty, kteří projeví zájem o dané téma. Kliknutím na tlačítko zobrazující počet studentů se zájmem se zobrazí seznam, který obsahuje studenty seřazené dle času, kdy projeví zájem. Na první (nejvyšší) pozici se nachází student, který projevil zájem jako první.



Obrázek 7: Detail tématu z pohledu studenta

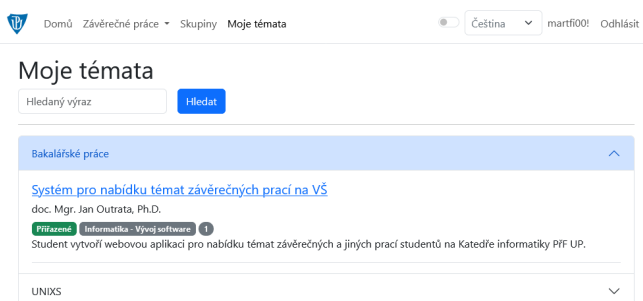
4.3.7 Diskuze

V detailu tématu se nachází sekce diskuze, ve které může uživatel přidávat příspěvky. V každém příspěvku se kromě autora, data a textu nachází dvě tlačítka „Odpovědět“ a „Smazat“, které slouží k příslušným akcím.

Studenti mají možnost zveřejnit příspěvek anonymně pomocí zaškrtnutí pole. Autor anonymního příspěvku je viditelný pro vyučující a administrátory.

4.3.8 Moje témata

Pod odkazem „Moje témata“ se skrývá seznam všech témat, se kterými má uživatel nějaký vztah.

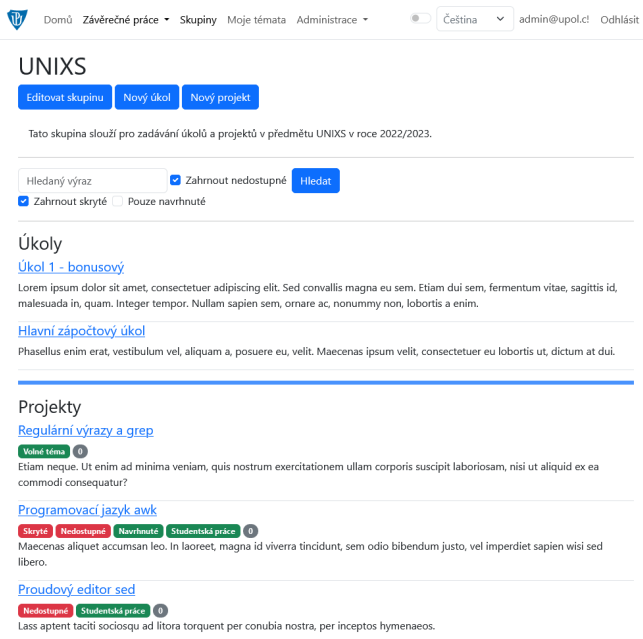


Obrázek 8: Stránka „Moje témata“

4.4 Skupiny

Při zvolení odkazu „Skupiny“ z navigačního menu se uživatel dostane na stránku zobrazující seznam všech veřejných skupin.

Pokud má uživatel dostatečná oprávnění, uvidí pod nadpisem Skupiny tlačítko „Vytvořit skupinu“, které přesměruje uživatele na formulář sloužící k vytvoření skupiny. Pokud uživatel nezadá anglický název skupiny, nastaví se anglický název na hodnotu českého názvu. Anglický název skupiny musí být unikátní.



Obrázek 9: Detail skupiny z pohledu vyučujícího

4.5 Administrace

Pro přístup k administraci musí mít uživatel roli Administrátor. Každý administrátor uvidí v navigačním menu položku „Administrace“. Administrace umožňuje následující tři operace:

4.5.1 Uživatelé

Pro vyhledání uživatele musí administrátor využít vyhledávací pole.

Kliknutím na odkaz „Role“ u uživatele se zobrazí stránka s přepínači daných rolí, která umožní změnu role uživatele.

Kliknutím na odkaz „Smazat“ se uživatel odstraní. Uživateli se smažou veškeré příspěvky v diskuzích a vlastnictví témat a skupin se předá administrátorovi.

Akce „Vytvořit uživatele“ přesměruje administrátora na formulář, který slouží k registraci uživatelů bez univerzitních účtů.

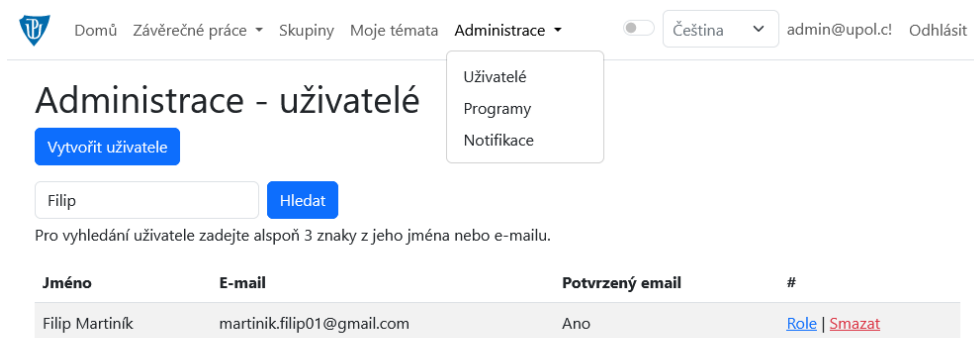
4.5.2 Programy

Tato sekce slouží k úpravě programů, které mohou být přiřazovány jako doporučené studijní programy pro témata. Tyto programy lze vytvářet, editovat a mazat.

4.5.3 Notifikace

Administrátor má možnost upravovat obsah notifikací. Po vybrání notifikace ze seznamu aplikace přesměruje administrátora na stránku obsahující dvě textová pole, která reprezentují obsah zvolené notifikace.

Do těchto textových polí může uživatel zadávat klíčová slova, která se nahradí za odpovídající reálné hodnoty. Návod pro klíčová slova je dostupný pod tlačítkem „Návod na editaci notifikace“.



Obrázek 10: Správa uživatelů v administraci

5 Rozšíření aplikace

Ačkoliv je aplikace plně funkční a plní všechny požadavky zadání, tak se zde nachází určitá rozšíření, která by kvalitu aplikace pozvedla.

5.1 Napojení na SSO univerzity – Shibboleth

Jako hlavní rozšíření aplikace považuji napojení přihlašování univerzitních uživatelů přes „Jednotné přihlášení Univerzity Palackého“. Při tomto způsobu přihlášení by uživatelé nemuseli vkládat svou důvěru v implementovaný systém a jeho zacházení s uživatelskými údaji. Také by se nemuseli přihlašovat vůbec, pokud by již byli někde přes tento systém přihlášení. Tento systém jsem bohužel nemohl použít z technických důvodů.

5.2 Odevzdávání vypracovaných témat

Dalším užitečným rozšířením je možnost odevzdávat vypracovaná témata jak závěrečných prací, úkolů či projektů. Tato funkcionalita nebyla součástí zadání a z části se jedná se o téma jiné bakalářské práce. Z tohoto důvodu nebyla implementována, ačkoliv by z mého pohledu rozšířila aplikaci velice vhodným a zásadním způsobem.

Závěr

Výsledkem této bakalářské práce je funkční webová aplikace, která může nahradit aktuální řešení nabídky závěrečných témat na katedře informatiky Univerzity Palackého. Aplikace také nabízí správu projektů a úkolů sloužící pro vyučované předměty. Ačkoliv byla aplikace vyvíjena zejména pro katedru informatiky, tak výsledná aplikace je jednoduše nasaditelná i pro jiné katedry a univerzity.

Conclusions

The result of this bachelor thesis is a functional web application that can replace the current solution of the offering of theses at the Department of Informatics of Palacký University. The application also provides project and assignment management used for the taught courses. Although the application was developed specifically for the Department of Informatics of Palacký University, the final application is easily deployable to other departments and universities.

A Obsah přiloženého datového média

text/

Adresář s textem práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech (textových) příloh, a všechny soubory potřebné pro bezproblémové vytvoření PDF dokumentu textu.

src/

Zdrojový kód aplikace.

bin/

Aplikace připravená k nahrání na webový server.

README.txt

Návod pro zprovoznění aplikace na webovém serveru.

Literatura

- [1] *C Sharp*. [online]. [cit. 2023-3-30]. Dostupný z: https://cs.wikipedia.org/wiki/C_Sharp.
- [2] *.NET documentation*. [online]. [cit. 2023-3-30]. Dostupný z: <https://learn.microsoft.com/en-us/dotnet/>.
- [3] *ASP.NET / Open-Source web framework*. [online]. [cit. 2023-3-30]. Dostupný z: <https://dotnet.microsoft.com/en-us/apps/aspnet>.
- [4] *Entity framework documentation*. [online]. [cit. 2023-3-30]. Dostupný z: <https://learn.microsoft.com/en-us/ef/>.
- [5] *Object-relational mapping*. [online]. [cit. 2023-3-30]. Dostupný z: https://en.wikipedia.org/wiki/Object-relational_mapping.
- [6] Delamater, Mary; Murach, Joel. *Murach's ASP.NET Core MVC*. 2020. 760 s. ISBN 978-1-943872-49-7.
- [7] *Dropdown select box for bootstrap 5*. [online]. [cit. 2023-3-30]. Dostupný z: <https://github.com/jarstone/dselect>.
- [8] *Overview of ASP.NET Core MVC*. [online]. [cit. 2023-3-30]. Dostupný z: <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-7.0>.
- [9] *Lightweight Directory Access Protocol*. [online]. [cit. 2023-3-30]. Dostupný z: https://en.wikipedia.org/wiki/Lightweight_Directory_Access_Protocol.
- [10] *Introduction to ASP.NET Identity*. [online]. [cit. 2023-3-30]. Dostupný z: <https://learn.microsoft.com/en-us/aspnet/identity/overview/getting-started/introduction-to-aspnet-identity>.