

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Systém 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: 36
Přílohy: 1 flash disk
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, Palacky University Olomouc
Year of defense: 2023
Study program: Computer Science, Specialization: Programming and Software Development
Supervisor: doc. Mgr. Jan Outrata, Ph.D.
Page count: 36
Supplements: 1 flash drive
Thesis language: Czech

Anotace

Tato bakalářská práce se zaměřuje na řešení problému s nabídkou 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.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1 Úvod	8
1.1 Zadání práce	8
1.2 Existující řešení	8
1.3 Funkcionalita aplikace	9
1.3.1 Témata	9
1.3.2 Autentifikace	10
1.3.3 Notifikace	10
1.3.4 Lokalizace	10
1.3.5 Funkcionalita z pohledu uživatelů	10
2 Použité technologie	11
2.1 C#	11
2.2 ASP.NET	12
2.3 PostgreSQL	12
2.4 Entity Framework	12
2.5 Razor views	12
2.6 HTML	12
2.7 CSS	12
2.8 Javascript	12
2.9 Bootstrap	13
2.10 Použité knihovny	13
2.11 Další tecnologie	13
3 Programátorská příručka	13
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 Univerzity 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	19
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	22
3.4 Posílání e-mailů	22
3.5 Lokalizace	23
3.6 Konfigurace aplikace	23

4 Uživatelská příručka	24
4.1 Student	24
4.2 Vyučující	24
4.3 Externího uživatel	24
4.4 Administrátor	24
5 Možná rozšíření	24
5.1 Napojení na SSO univerzity	24
5.2 Funkcionalita	24
6 Závěr	24
6.0.1 Testování	24
6.1 Sazba literatury	24
6.1.1 Sazba bibliografie přes BIBL ^A TEX	24
6.1.2 Manuální sazba bibliografie	25
6.2 Drobná makra	25
6.3 Sazba rejstříku	25
6.4 Sazba zdrojových kódů	27
Závěr	29
Conclusions	30
A První příloha	31
B Druhá příloha	31
C Obsah přiloženého datového média	31
Seznam zkratek	33
Literatura	34
Rejstřík	36

Seznam tabulek

1	Odstavce v tabulkách	26
---	--------------------------------	----

1 Úvod

Každého studenta na vysoké škole čeká výběr závěrečné práce a ať už 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 ani nezapomíná na externí uživatele, kteří mohou být například zástupci firem či zástupci státních organizací, kterým aplikace umožní navrhovat téma 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 kdokoliv jiný, např. zástupci firem, s přístupem. Tato druhá témata bude moct vyučující klasifikovat (bakalářská vs. diplomová práce, vhodnost pro studijní program apod.), "adoptovat" (jako potenciální vedoucí práce) a "svoje" témata pak přidělit studentům. Student bude moct 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 musíme 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 téma 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 vedoucí práce a odevzdá sekretářce katedry.

Studenti by si měli vybírat téma vhodné pro svůj studijní program/specializaci. Studentům specializace 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 Janoštík

Vizualizace komunikace virtuální sítě v rodině protokolů TCP/IP

Vhodné pro studenty programu/specializace: 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/specializace: Učitelství

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

Můžeme vidět, že aktuální řešení nenabízí velkou funkcionalitu. Jedná se v podstatě o statickou stránku s nabízenými tématy vyučujícími, kde daná téma obsahují název, stručný popis a doporučené studijní programy.

Pro studenta, který by si vybíral své budoucí téma, může stránka působit nepřehledně. Jelikož například vidí téma, která nemusí být určená pro jím studovaný program nebo mezi všemi tématy se nachází i ta, která jsou již zabraná. Současné řešení také neposkytuje možnost vyhledávání, které lze ale částečně nahradit funkcionalitou prohlížeče a zkratkou ctrl + F.

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

1.3 Funkcionalita aplikace

Tato sekce 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. Proto tady ta sekce shrnuje všechny požadavky na výslednou aplikaci.

1.3.1 Témata

Ačkoliv hlavním účelem této práce je vytvořit systém pro nabídku závěrečných prací, tak aplikace bude podporovat i správu úkolů a projektů týkajících se předmětů vyučovaných na katedře informatiky. Tyto téma 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 Autentifikace

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 vhodných notifikací e-mailem uživatelům, kteří se přihlásí k jejich odběru. Uživatelé mohou být notifikovaní 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 můžeme rozdělit na základě toho, jaký uživatel ji bude využívat. Uživatele tohoto systému můžeme rozdělit do následujících skupin:

- Všichni uživatelé
 - Prohlížení, vyhledávání, 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

- Externí uživatelé
 - Navrhovat téma závěrečných prací a projektů
 - Změnit své heslo a e-mail
- Studenti
 - Navrhovat téma závěrečných prací a projektů
 - Projevit zájem o téma
 - Vytvářet anonymní příspěvky v diskuzích
- Vyučující
 - Spravovat svá téma závěrečných prací, úkolů a projektů
 - Klasifikovat a případně adoptovat navrhovaná téma externími uživateli a studenty
 - Mazat jakékoliv příspěvky v diskuzích
- Administrátoři
 - Přiřazovat role uživatelům a potvrzovat registrace externím uživatelům
 - Editovat text notifikací
 - Spravovat studijní programy
 - Spravovat veškerá téma
 - Mazat jakékoliv příspěvky v diskuzích

2 Použité technologie

K vytvoření jakékoliv aplikace je potřeba využít nějakých technologií. Tato kapitola popisuje technologie, které byly použity pro naprogramování výsledné webové aplikace.

2.1 C#

C# 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 je vývojová platforma složena 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 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 persistentní ukládání dat s kterými aplikace pracuje. Jedná se zejména o data nabízených témat a uživatelů.

2.4 Entity Framework

Entity framework, je open source ORM (object-database mapper). ORM 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 se typicky používají pro zobrazení soubory s příponou .cshtml. Tyto soubory obsahují značkovací jazyk HTML rozšířený o Razor kód. Razor kód nám umožní v souborech view používat jazyk C#. Pokud si uživatel vyžádá některou 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, zejména sémantický význam.

2.7 CSS

Kaskádové styly slouží k popisu způsobu zobrazení elementů na webových stránkách napsaných v jazyce HTML.

2.8 Javascript

Javascript je programovací jazyk, který slouží k vykonávání kódu na straně koncového uživatele. Finální aplikace využívá Javascript na 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 před připravených 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
 - Jedná se o knihovnu, která usnadňuje práci s javascriptem kódem. Je také vyžadována pro funkčnost Bootstrapu.
 - Licence: MIT
- dselect
 - Knihovna, podporující Bootstrap 5, která slouží k zprovoznění vyhledávání ve výběru uživatelů ve formulářích. Knihovnu jsem mírně upravil k potřebám aplikace.
 - Licence: MIT
- System.DirectoryServices.Protocol
 - Knihovna součástí .NET, která umožňuje práci s protokolem LDAP, který je využíván při autentizaci studentů a vyučujících.
 - Licence: MIT
- Npgsql
 - Slouží k propojení Entity Framework a databáze PostgreSQL.
 - Licence: PostgreSQL

2.11 Další tecnologie

- Jetbrains Rider TODO
- Github TODO

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 využití 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 (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 naší 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 reaguje tím, že z Modelu vybere žádaná data a odešle je příslušnému View.

3.1.2 Dependency injection

Aplikace využívá pro určité služby Dependency injection, jedná se o návrhový vzor, ve kterém objekt dostává jiné objekty, na kterých je závislý. Závislý objekt sám neví jak dané závislosti získávat, to má na starosti injector.

V ASP.NET MVC se tento vzor využívá následovně:

1. Vytvoří se služba/třída, z pravidla implementující rozhraní popisující danou funkčnost
2. Tato služba se zaregistrouje 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 instancí
3. Pro využití dané služby se přidá služba do konstruktoru třídy závislého

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. (Jaké modely aplikace používá si ukážeme později)

- 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, s 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 funkctionalitu, která se týká témat např.: vytváření, přidání komentáře, vyhledávání témat.

- GroupController

Obsluhuje vytváření, editaci, mazání a zobrazování skupin (k čemu se využívají skupiny popíšeme v následující kapitole).

- AdministrationController

Obsluhuje veškerou administraci stránky.

- wwwroot

Obsahuje statický obsah webové stránky. Kaskádové styly, obrázky, apod..

- Areas/Identity

V této složce nalezneme soubory týkající se ASP.NET Identity. Tento systém je zodpovědný za autentizaci a autorizaci.

- Services

Zde nalezneme nezávislé služby, které aplikace využívá.

- Resources

Soubory sloužící pro lokalizaci aplikace.

- Data

Tato složka obsahuje důležitý soubor ApplicationDbContext.cs, ve kterém se definuje interakce mezi aplikací a databází a soubor Seed.cs sloužící k vytvoření potřebné struktury databáze a záznamů.

3.1.4 Struktura databáze a popis modelů

Tato sekce 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 můžeme chápat jako název a datový typ sloupce v tabulce). Tyto vlastnosti mohou být rozšířeny o další informace sloužící například k validaci.

Pří 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.

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 tabulky.

Pro lepší představu jak celkový model databáze vypadá slouží následující diagram.



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 ke příspěvkům v diskuzích. Pro ukládání informací ohledně zájmu o téma slouží model UserInterestedTopic.

Programy

Aplikace podporuje přidávání doporučených studijních programů k tématům závěrečných prací. K této funkcionality slouží Model Programme, který reprezentuje studijní program a model TopicRecommendedProgram, který zodpovídá za přiřazení studijních programů k tématům.

Skupiny

Model Group slouží k 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 své předměty, které vyučují.

Uživatelé a role

Aplikace využívá ApplicationUser, ApplicationRole a ApplicationUser tyto modely dědí z již před vytvořených 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 můžeme pro účely této aplikace rozdělit na 2 skupiny: Autentizace uživatelů z Univerzity Palackého a Autentizace uživatelů mimo Univerzitu Palackého.

3.2.1 Autentizace uživatelů mimo Univerzity Palackého

Tito uživatelé jsou v aplikaci chápáni zejména jako externí uživatelé. Pro autentizaci si nejprve musí vytvořit účet a až poté se mohou přihlašovat. Mohou také ověřit svou e-mailovou adresu pro dokázání, že se skutečně jedná o jejich e-mail. Proces této autentizace je zcela nezávislý na službách třetích stran.

Pro plnou aktivaci potřebují schválení administrátorem, který jim musí přiřadit typický roli externího uživatele.

- 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 heslo odpovídá tomu uloženému.

3.2.2 Autentizace uživatelů z Univerzity Palackého

Tito uživatelé jsou studenti nebo vyučující na Univerzitě Palackého. Pro autentizaci nevyužívají registrace, ale rovnou se přihlašují pomocí stejných údajů, které používají pro autentizaci v systému STAG nebo v doméně Univerzity Palackého. Přihlašovaní těchto uživatelů je závislé na službách univerzity.

- Při prvním přihlášení se zkontroluje zda uživatel zadal správné přihlašovací údaje pomocí Active Directory Univerzity Palackého. Pokud dané údaje jsou správné tak aplikace získá osobní informace o uživateli (jméno, příjmení, e-mail) a také zjistí zda-li je uživatel student nebo vyučující a přiřadí mu odpovídající roli. Tyto informace ze získávají pomocí Stag Services.
- Při dalším přihlašování se už pouze kontroluje správnost přihlašovacích údajů pomocí Active Directory.

3.2.3 Využití služeb univerzity pro přihlašování

Tato sekce 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, který slouží k ukládání a přístupu k datům na adresářovém severu.

Stag services

STAG je univerzitní informační systém. Aplikace z něj získává osobní informace uživatelů při jejich prvním přihlášení.

Ke komunikaci aplikace používá REST API přístupné pouze z vnitřní sítě univerzity, které poskytuje stagservices.upol.cz.

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. Tento systém je funkční sám o sobě, ale zle ho jednoduše rozšířit o další funkcionality. Veškeré soubory, ze kterých se Identity skládá se nacházejí ve adresáři Areas/Identity/-Pages/Account. Je důležité podotknout, že tento systém v aplikaci není postaven

technologií ASP.NET MVC ale používá jinou technologii ASP.NET frameworku a to Razor Pages.

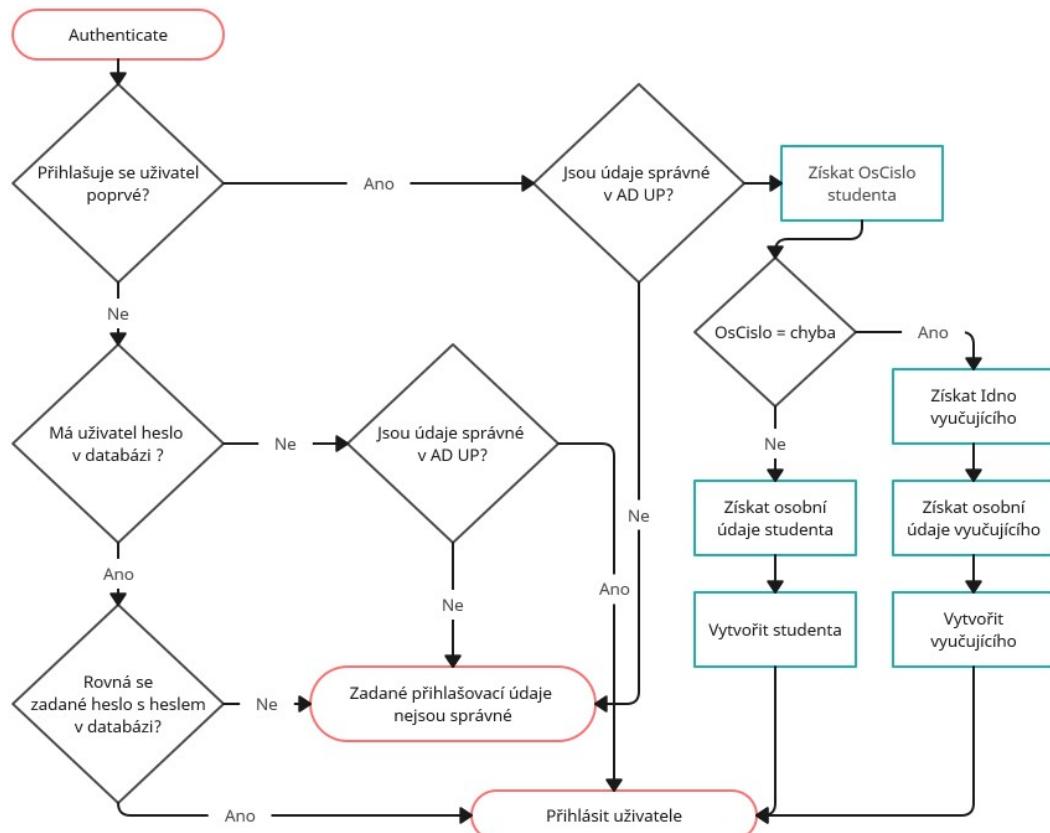
Aplikace nepotřebuje veškerou funkcionalitu, kterou Identity sám o sobě nabízí. Proto jsou v souboru ApplicationDbContext.cs určité vlastnosti modelů ignorovány a některé stránky smazány. Aplikace také ignoruje celé modely, které jsou zmíněny v sekci Struktura databáze a popis modelů.

3.2.5 Služba UpolAuthentication

Hlavní rozšíření systému Identity můžeme vidět v souborech Login.cshtml.cs a Register.cshtml.cs. Kde v obou souborech 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 tři parametry: přihlašovací jméno, heslo a zda si uživatel přeje zůstat přihlášen po zavření webové stránky.

Chování této služby můžeme rozdělit na dva případy na základě toho jestli je uživatel z Univerzity Palackého či nikoliv. Následující diagram abstraktně popisuje chování služby.



3.2.6 Role

Aplikace pro autorizaci využívá na pozadí následující role. Administrátor je ale 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 týkající se dané role.

- Administrator, Teacher, Student, External

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

- Topic, AnyTopic

Umožní vytvářet, editovat, klasifikovat/adaptovat a mazat téma, úkoly a projekty.

- ProposeTopic

Umožní uživateli navrhnout závěrečné téma nebo projekt.

- 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 diskusi 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 nahrať 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. Tato služba se v kódu využívá pomocí Dependency Injection.

3.3.1 Odesílání notifikací

Služba po zavolení 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 SendNotificacation a ta pro každou notifikaci provede následující.
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. Tyto klíčová slova jsou poté nahrazovány podle skutečných hodnot, které se dané notifikace týkají. K tady tomu procesu slouží metoda NotificationSubstitution.

3.4 Posílání e-mailů

Odesílání všech e-mailů má na starost třída SmtpEmailSender, která implementuje rozhraní IEmaiI Sender s metodou SendEmailAsync, která je zodpovědná za odeslání e-mailů. Odesílání e-mailů v této metodě je řešeno pomocí SMTP protokolu použitím třídy SmtpClient.

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

3.5 Lokalizace

Aplikace podporuje dva jazyky češtinu a angličtinu. Témata, skupiny a celkově veškerá datová pole, která se v životním cyklu 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 tady toho faktu je lokalizace provedena jednodušeji na úkor obtížnosti přidání nového jazyka.

Pro použití lokalizace se ve view souborech používá IHmlLocalizer a v kontrollerech IStringLocalizer. Instance se potom 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.

3.6 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 severu univerzity.
- 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

4.1 Student

4.2 Vyučující

4.3 Externího uživatel

4.4 Administrátor

5 Možná rozšíření

5.1 Napojení na SSO univerzity

5.2 Funkcionalita

6 Závěr

6.0.1 Testování

```
1 % Tiskne český závěr práce.  
2 \begin{kiconclusions}  
3 Závěr práce v \uv{českém} jazyce.  
4 \end{kiconclusions}  
5  
6 % Tiskne anglický závěr práce.  
7 \begin{kiconclusions}[english]  
8 Thesis conclusions written in \uv{English}.  
9 \end{kiconclusions}
```

Zdrojový kód 1: Sazba závěrů

6.1 Sazba literatury

Pro sazbu literatury má uživatel dvě možnosti. Může použít služeb balíků BIBLATEX, který je pro **kistyles** zapnutý, či lze použít manuální sazbu bibliografie.

6.1.1 Sazba bibliografie přes BibLATEX

Při použití tohoto balíku se data o použité literatuře ukládají do dedikovaného textového souboru, ukázku najdete i v tomto stylu pod jménem `bibliografie.bib`.

Formát daného souboru je nad rámec této dokumentace a je na každém uživateli, aby si jej nastudoval. Bibliografie se tiskne makrem `\printbibliography`.

Taktéž v preambuli dokumentu je třeba definovat, který soubor data bibliografie obsahuje, tedy například `\bibliography{bibliografie.bib}`.

Dokument, který využívá BIBI^ATEX je následně nutné přeložit jak pomocí překladače zvoleného ovladače, tak pomocí aplikace biber. Více informací poskytne soubor `Makefile` z distribuce tohoto stylu.

Výhodou tohoto přístupu je, že bibliografie se vysází automaticky a (obvykle) není třeba manuální úprava formátování.

6.1.2 Manuální sazba bibliografie

Manuální sazba obnáší vysázení prostředí `thebibliography` ručně. To je nad rámcem tohoto dokumentu. Ukázku tohoto přístupu lze samozřejmě nalézt ve zdrojovém souboru tohoto dokumentu nebo také [zde](#).

Pro aktivaci manuální sazby bibliografie je třeba volat třídu `kidiplom` s parametrem `biblatex=false`. Mějte, prosím, na paměti, že v tomto módu jsou makra `\bibliography` a `\printbibliography` nedostupná.

6.2 Drobná makra

Základní styl definuje hned několik maker pro usnadnění práce. Například makro `\buno` vysází řetezec „bez újmy na obecnosti“. Je k dispozici i verze s prvním velkým písmenem, `\Buno`.

Je rovněž možno přidávat položky do seznamu zkratek. K tomu slouží makro `\newacronym`, které lze použít například jednoduše jako `\newacronym{UPOL}{UPOL}{\kitextunivcz}`. Na danou zkratku se pak lze odkazovat jednoduše, `\gls{UPOL}`.

Sazba uvozovek respektuje nastavení částí dokumentu, a proto se doporučuje používat makro `\uv`. V anglické závěru práce toto platí taky, viz tato PDF ukázka.

Styl podporuje sazbu odstavců v tabulkách, více obsahuje tabulka 1.

K dispozici jsou také makra pro sazbu C# (`\csharp`) či C++ (`\cpp`).

6.3 Sazba rejstříku

Sazba rejstříku sestává z několika kroků:

1. Je třeba přes volbu `index=true` rejstříkování povolit.
2. Použítím makra `\index` rejstříkovat vybrané pojmy.
3. Kompilovat s použitím utility `makeindex`. Pro specifika tohoto kroku si stačí prohlédnout soubor `Makefile`.

Makro `\index` je redefinováno tak, že sází klikací odkaz na výraz v rejstříku. Je doporučeno jej použít ihned za výrazem^{1}.

Tabulka 1: Odstavce v tabulkách

<p>Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.</p>	<p>Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.</p>	<p>Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.</p>
--	---	---

Omezení redefinovaného makra \index: klikací odkaz nefunguje, pokud použijete konstrukci \index{výraz|makro} (resp. \index{výraz|(makro)}), např. \index{výraz|textit}.

Rejstřík lze vysázet pomocí makra \printindex.

6.4 Sazba zdrojových kódů

Styl nabízí dva způsoby sazby zdrojových kódů:

1. Sazbu řádkových kódů, například **background-color:** white;. K tomu slouží makro formátu \kiinlinecode{jazyk}{separátor}{kód}. Za separátor je vhodné volit jakýkoliv znak, který se nevyskytuje v samotném sázeném zdrojovém kódu. Za jazyk je nutno dosadit jeden z těchto: C, TeX, PHP, HTML, Lisp, SQL, TeX, Python, Java, TutorialD, text, csharp, cpp, JavaScript, CSS.
2. Sazbu zdrojových kódů do separátních prostředí. Takto vytiskněný kód se objeví v seznamu zdrojových kódů. Ukázka například zdrojový kód 2. Ukázku sazby najeznete ve zdrojovém kódu tohoto dokumentu.

```
1 int main("cs acsa") // komentar
2 int main("cs acsa") // komentar
3 int main("cs acsa") // komentar
4 int main("cs acsa") // komentar
5 int main("cs acsa") // komentar
```

Zdrojový kód 2: C++

```
1 new object() // komentar
```

Zdrojový kód 3: JS

```
1 public static int main("cs acsa") // komentar
```

Zdrojový kód 4: C#

```
1 SELECT * FROM table_1; /* komentár */
```

Zdrojový kód 5: SQL

```
1 table_1 AND table_2;
```

Zdrojový kód 6: TutorialID

Závěr

Závěr práce v „českém“ jazyce.

Conclusions

Thesis conclusions in “English”.

A První příloha

Text první přílohy

B Druhá příloha

Text druhé přílohy

C Obsah přiloženého datového média

Na samotném konci textu práce je uveden stručný popis obsahu přiloženého datového média (CD/DVD, flash disk apod.), tj. jeho závazné adresářové struktury, důležitých souborů apod.

bin/

Instalátor INSTALATOR programu, popř. program PROGRAM, spustitelné přímo z média. / Kompletní adresářová struktura webové aplikace WEBOVKA (v ZIP archivu) pro zkopirování na webový server. Adresář obsahuje i všechny runtime knihovny a další soubory potřebné pro bezproblémový běh instalátoru a programu z média / pro bezproblémový provoz webové aplikace na webovém serveru.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty programu PROGRAM / webové aplikace WEBOVKA se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu / adresářové struktury pro zkopirování na webový server.

readme.txt

Instrukce pro instalaci a spuštění programu PROGRAM, včetně všech požadavků pro jeho bezproblémový provoz. / Instrukce pro nasazení webové aplikace WEBOVKA na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Navíc médium obsahuje:

data/

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

install/

Instalátory aplikací, runtime knihoven a jiných souborů potřebných pro provoz programu PROGRAM / webové aplikace WEBOVKA, které nejsou standardní součástí operačního systému určeného pro běh programu / provoz webové aplikace.

literature/

Vybrané položky bibliografie, příp. jiná užitečná literatura vztahující se k práci.

U veškerých cizích převzatých materiálů obsažených na médiu jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na médiu, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.

Literatura

- [1] Bratková, Eva (sest.). *Metody citování literatury a strukturování bibliografických záznamů podle mezinárodních norem ISO 690 a ISO 690-2: metodický materiál pro autory vysokoškolských kvalifikačních prací* [online]. Verze 2.0, aktualiz. a rozšíř. Praha: Odborná komise pro otázky elektronického zpřístupňování vysokoškolských kvalifikačních prací, Asociace knihoven vysokých škol ČR, 2008 [cit. 2011-2-2]. 60 s. Dostupný z: <<http://www.evskp.cz/SD/4c.pdf>>.
- [2] Borgman, Christine L. *From Gutenberg to the global information infrastructure: access to information in the networked world*. First. Cambridge (Mass): The MIT Press, 2003. xviii, 324 s. ISBN 0-262-52345-0.
- [3] Greenberg, David. Camel drivers and gatecrashers: quality control in the digital research library. In Hawkins, B.L; Battin, P (ed.). *The mirage of continuity: reconfiguring academic information resources for the 21st century*. Washington (D.C.): Council on Library and Information Resources; Association of American Universities, 1998, s. 105–116.
- [4] LYNCH, C. Where do we go from here?: the next decade for digital libraries. *DLib Magazine* [online]. 2005, vol. 11, no. 7/8 [cit. 2005-8-15]. Dostupný z: <<http://www.dlib.org/dlib/july05/lynch/07lynch.html>>. ISSN 1082-9873.
- [5] Národní knihovna. A big paper. *The journal of big papers*. 1991, vol. 12, no. 3. ISSN 2232-332X.
- [6] Děta, Hugh; Rychlík, Tomáš. *A big paper: Podtitul* [online]. Druhé vyd. Praha: Academia, 1991 [cit. 2011-1-12]. 550 s. Pokusná edice. Dostupný z: <<http://pokus.cz>>. ISBN 978-44-55-X.
- [7] Děta, Hugh; Rychlík, Tomáš; Další, Pepa aj. *Úplně úžasná knížka*. Třetí vyd., 1991.
- [8] Děta, Hugh; Rychlík, Tomáš; Další, Pepa, et al. *Úplně úžasná knížka*. 3rd ed. Praha: MIT Press, 1991. 332 s.
- [9] Freely, I.P. A small paper: Podtitulek. *The journal of small papers*. 1997, roč. 1, č. 3, s. 2–5. to appear.
- [10] Jass, Hugh. A big paper. *The journal of big papers*. 1991, roč. 23.
- [11] Černý, Hugh. Titulek. *The journal of big papers*. 1991, roč. 12, č. 2, s. 22–44. Dostupný také z: <<http://dx.doi.org/10.112.22/jkn>>.
- [12] Kollmannová, Ludmila; Bubeníková, Libuše; Kopecká, Alena. *Angličtina pro samouky*. 5. vyd. Praha: Státní pedagogické nakladatelství, 1977. 525 s. Učebnice pro samouky. ISBN 80-04-25663-5.
- [13] Kollmannová, Ludmila; Bubeníková, Libuše; Kopecká, Alena. *Angličtina pro samouky*. 5. vyd. Novotná, Pepina. Praha: Státní pedagogické nakladatelství, 1977. 2. Podkapitola, s. 22–29. ISBN 80-04-25663-5.

- [14] *TUGBoat*. 1980-. Dostupný také z: <<http://tugboat.tug.org/>>. ISSN 1222-3333.
- [15] Knuth, Donald. Journeys of T_EX. *TUGBoat*. 2003, vol. 17, no. 3, s. 12–22. Dostupný také z: <<http://tugboat.tug.org/kkk.pdf>>. ISSN 1222-3333.
- [16] Geniální, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007. ISBN 978-222-626-222-2.
- [17] Vlašťovka, Josef. Velmi zajímavý článek. In Geniální, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007, s. 22–45. ISBN 978-222-626-222-2.

Rejstřík

výraz, [25](#)