

Návrh aplikace pro nabídku on-line kurzů

Dalibor Ďuriš



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav počítačových a komunikačních systémů

Akademický rok: 2021/2022

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení:	Dalibor Ďuriš
Osobní číslo:	A19019
Studijní program:	B3902 Inženýrská informatika
Studijní obor:	Softwarové inženýrství
Forma studia:	Prezenční
Téma práce:	Návrh aplikace pro nabídku on-line kurzů
Téma práce anglicky:	Design of an Application for Online Courses

Zásady pro vypracování

1. Proveďte rešerši existujících řešení.
2. Vypracujte stručný rozbor technologií, které budou použity k návrhu.
3. Proveďte rozbor a analýzu požadavků na zvolené řešení.
4. Zpracujte aplikaci na základě výsledků analýzy.
5. Věnujte pozornost zabezpečení aplikace.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. PAZ, Guay a Jose ROLAO. Introduction to azure cosmos db. In: Microsoft Azure Cosmos DB Revealed. Apress, Berkeley, CA, 2018. p. 1-23. ISBN 9781484233504.
2. HILLAR, Gaston C.; YÖNDEM, Daron. Guide to NoSQL with Azure Cosmos DB: Work with the massively scalable Azure database service with JSON, C#, LINQ, and .NET Core 2. Packt Publishing Ltd, 2018. ISBN 1789612896.
3. Dokumentace k Azure Cosmos DB, 2020. Dokumentace k Azure Cosmos DB [online]. USA: Microsoft [cit. 2020-11-25]. Dostupné z: <https://docs.microsoft.com/cs-cz/azure/cosmos-db/>
4. OWLER, Adam. NoSQL for dummies. John Wiley & Sons, 2015. ISBN 9788126554904.
5. FREEMAN, Adam. Putting ASP.NET Core in Context. In: Pro ASP.NET Core 3. Apress, Berkeley, CA, 2020. p. 3-7. ISBN: 9781484254400.

Vedoucí bakalářské práce:

doc. Ing. Petr Šilhavý, Ph.D.
Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce: **3. prosince 2021**

Termín odevzdání bakalářské práce: **23. května 2022**

doc. Mgr. Milan Adámek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
garant oboru

Ve Zlíně dne 24. ledna 2022

Jméno, příjmení: Dalibor Ďuriš

Název bakalářské práce: Návrh aplikace pro nabídku on-line kurzů

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 23.5.2022

Dalibor Ďuriš v.r.
podpis studenta

ABSTRAKT

Cieľom bakalárskej práce je vytvorenie ukážkovej aplikácie pre ponuku online kurzov. V teoretickej časti je popis existujúcich riešení dostupných aplikácií, ďalej obsahuje popis technológií ktoré boli použité pre návrh aplikácie. Praktická časť sa venuje aplikačnému návrhu a implementácii návrhu. Aplikácia je vytvorená pomocou frameworku Asp.Net s využitím MS SQL. Aplikácia používa návrhový vzor MVC.

Kľúčová slova: Asp.Net, MVC, Webová aplikácia, Razor pages, Online kurzy

ABSTRACT

The aim of the bachelor thesis is to create a sample application for offering online courses. The theoretical part is a description of existing solutions of available applications, it also contains a description of the technologies that were used to design the application. The practical part deals with application design and design implementation. The application is created using the Asp.Net framework using MS SQL. The application uses the MVC design pattern.

Keywords: Asp.Net, MVC, Web application, Razor pages, Online courses

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD.....	9
I TEORETICKÁ ČÁST	10
1 ROZBOR PROBLEMATIKY	11
1.1 APLIKÁCIE PRE PONUKU ONLINE KURZOV	11
2 POPIS POUŽITÝCH TECHNOLOGIÍ.....	14
2.1 ASP.NET.....	14
2.1.1 ASP.NET WEB Forms.....	14
2.1.2 ASP.NET Web Pages.....	15
2.1.3 ASP.NET MVC.....	15
2.1.4 Model	16
2.1.5 View	16
2.1.6 Controller	17
2.2 BOOTSTRAP.....	17
2.3 HTML.....	17
2.4 CSS.....	18
2.5 FRAMEWORK ENTITY	18
2.6 ZABEZPEČENIE ASP.NET	18
2.6.1 Autentifikácia / Autorizácia	19
2.6.2 Ochrana údajov	19
2.6.3 Útoky Cross-Site request forgery (XSRF/CSRF)	20
2.6.4 Uplatnenie HTTPS v ASP.NET	20
2.6.5 CORS v ASP.NET	20
2.6.6 Zabránenie cross site skriptovaniu	21
2.7 MICROSFT VISUAL STUDIO	21
2.8 SQL	22
2.9 UML	22
II PRAKTICKÁ ČÁST.....	25
3 NÁVRH APLIKÁCIE.....	26
3.1 FUNKČNÉ POŽIADAVKY	27
3.2 NEFUNKČNÉ POŽIADAVKY.....	28
3.3 MODEL PRÍPADU UŽITIA	28
3.3.1 Scenáre	29
3.4 E-R DIAGRAM DATABÁZOVÝ MODEL.....	35
3.5 VYTVORENIE KURZU	37
4 TVORBA APLIKÁCIE	38
4.1 NÁHEAD APLIKÁCIE	38
4.2 PRIHLÁSENIE DO APLIKÁCIE	39

4.3	REGISTRÁCIA DO APLIKÁCIE	40
4.4	ZOBRAZENIE KURZOV	41
4.4.1	Zobrazenie detailu kurzu	42
4.4.2	Vytvorenie kurzu	43
4.4.3	Editácia kurzu	44
4.5	LEKTORI	44
4.6	VYHĽADÁVANIE V APLIKÁCIÍ	45
4.7	SPRÁVA UŽÍVATEĽOV	45
4.8	NÁKUPNÝ KOŠÍK	46
4.8.1	Vybavenie objednávky	47
4.9	SPRÁVA OBJEDNÁVOK	49
5	ZABEZPEČENIE APLIKÁCIE	50
	ZÁVĚR	52
	SEZNAM POUŽITÉ LITERATURY	54
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	56
	SEZNAM OBRÁZKŮ	57
	SEZNAM TABULEK	58
	SEZNAM PŘÍLOH	59

ÚVOD

V dnešnej dobe je online forma vzdelávania veľmi populárna a za posledné roky sa veľmi rýchlo popularita tejto formy vzdelávania zvyšuje. Online kurzy poskytujú formu vzdelávania z pohodlia domova. Pre aplikácie ktoré ponúkajú množstvo online kurzov je potrebné mať prehľad o užívateľoch ktorý využívajú ich služby. Cieľom tejto bakalárskej práce je navrhnúť a zrealizovať webovú aplikáciu pre ponuku online kurzov

Úlohou teoretickej časti tejto bakalárskej práce je získanie prehľadu o problematike aplikácií pre ponuku online kurzov. Po dokončení prehľadu problematiky budú porovnané existujúce riešenia. Ďalšou časťou bude opis nástrojov ktoré budú použité pre vývoj a návrh aplikácie pre ponuku online kurzov. Rozbor použitých nástrojov pre vývoj aplikácie sa zameria na technológie použité pri tvorbe návrhu aplikácie, zameria sa na Asp.Net, jeho popis a zabezpečenie.

Praktická časť bakalárskej práce bude obsahovať návrh aplikácie ktorý sa zameria na slovný popis návrhu aplikácie. Bude obsahovať návrh požiadavkou pre danú aplikáciu. Obsahom návrhu bude model prípadu užitia ktorý bude zahrňovať scenáre jednotlivých prípadov. Ďalšou časťou je návrh entitne relačného diagramu a databázovej schémy aplikácie ktorý bude popisovať jednotlivé vzťahy medzi entitami. Posledným bodom bude samotná tvorba aplikácie ktorá bude vychádzať z návrhu a požiadavku aplikácie. Tvorba aplikácie bude zahrňovať popis funkcionality aplikácie, zameria sa na popis a postup tvorby webovej aplikácie. Záverom práce bude vyhodnotenie vytvorenej aplikácie a prípadne jej návrhy na zlepšenie.

I. TEORETICKÁ ČÁST

1 ROZBOR PROBLEMATIKY

1.1 Aplikácie pre ponuku online kurzov

Na internete je k dispozícii mnoho aplikácií ktoré sa zaoberajú ponukou online kurzov. Svoju ponuku kurzov ponúkajú pre firmy ale aj pre jednotlivcov. Tieto aplikácie ponúkajú vzdelávanie z pohodlia domova pomocou sérií videí alebo návodov poprípade aj ich kombinácie. Výstupom z týchto online kurzov sú vo väčšinou certifikáty ktoré sú zasielané účastníkom kurzov pomocou emailu. V aplikáciách vieme nájsť kurzy ktoré sú platené ale tak isto sú v aplikáciách k dispozícii aj online kurzy ktoré sú platené buď jednorazovo. Niektoré z webových aplikácií ponúkajú aj kurzy zadarmo, poprípade ponúkajú predplatné ktorým si je možné sprístupniť určitú sériu kurzov. Pre porovnanie som si vyčlenil rôzne aplikácie ktoré ponúkajú rôzne kategórie online kurzov.

Seduo

Spoločnosť LMC s.r.o prevádzkuje aplikáciu seduo. Na ich stránke je aj ponuka vzdelávacích programov. Ich kurzy ponúkajú jednoduchosť možnosť vybrať si tempo vzdelávania. Umožňujú získanie záverečného certifikátu pomocou úspešného dokončenia kurzu. Ponúkajú skúšobnú verziu pre firmy a je možnosť si vybrať z mnoho kategórií kurzov. Obsahuje modul zobrazenia detailu kurzu kde je v krátkosti uvedený popis kurzu. Zobrazenie detailu kurzu po prihlásení zobrazí aj jeho stručnú osnovu ktorá opisuje čo daný kurz obsahuje. Táto aplikácia obsahuje aj hodnotenia samostatného kurzu, umožňuje užívateľom pridávať komentár. [1]

Skillmea

Aplikácia Skillmea prevádzkovaná spoločnosťou Learn2Code o.z. ponúka rôzne typy kurzov oproti ostatným aplikáciám je možné si zvoliť aj typ ťažnosti kurzu. Takisto je možné za poplatok získať predplatné ktoré sa delí na tri rôzne úrovne predplatného, výhodou je možnosť získania celoživotného predplatného. Taktiež obsahuje možnosť získania certifikátu o obsahovaní kurzu a jeho detail. Po prihlásení si môže užívateľ vybrať z množstva kurzov. Každý kurz obsahuje svoj popis, hodnotenie, jazyk v ktorom je kurz vedený a level akým je kurz vedený. Taktiež obsahuje zoznam kapitol ktoré sú súčasťou daného kurzu. Táto webová aplikácia umožňuje vyskúšať online kurzy zdarma. Ďalej umožňuje zobrazenie dokončených kurzov a certifikátov spojených s nimi, objednávky a úpravy profilu. [2]

Itlearning

Spoločnosť It learning slovakia, s.r.o. prevádzkuje webovú aplikáciu Itlearning, tá ponúka rôzne typy kurzov z oblasti informatiky pre jednotlivcov alebo firmy. Jedná sa o aplikáciu ktorá ponúka kurzy online je možné zobrazit' jeho popis, cenu. Obsahuje modul ktorý zobrazuje v akom stave je online kurz a je na výber z ôsmich možností, napríklad skupinový kurz, zobrazuje aký typ kurzu sa jedná online alebo denný. Je možné si zobrazit' hodnotenia kurzu taktiež aj osnovu ktorá popisuje čo daný kurz obsahuje [3]

Vita

VITA Academy s.r.o prevádzkuje webovú aplikáciu Vita academy. Aplikácia ponúka širokú ponuku kurzov z oblasti informatiky. Obsahuje modul ktorý zobrazuje dané kategórie kurzov. Ponúka kurzy pre jednotlivcov ale tak isto aj pre firmy a väčšiu skupinu ľudí. Každý kurz obsahuje svoj popis cenu, a obsah ktorý popisuje čo sa v danom kurze nachádza. Kurzy ktoré ponúka táto aplikácia v popise obsahujú aj potrebné znalosti ktoré sú potrebné pre zvládnutie kurzu. V tejto aplikácii je možné získať ročné predplatné ktoré umožní prístup ku všetkým kurzom. Obsahuje moduly ktoré zobrazujú získané certifikáty, vlastné kurzy. Jeden z ďalších modulov tejto aplikácie je zobrazenie štatistík užívateľa. [19]

Udemy

Udemy inc. Ponúka webovú aplikáciu udemy ktorá je na trhu už od roku 2010 a ponúka rôzne typy online kurzov. Obsahuje moduly pre zobrazenie kategórií online kurzov, zobrazenie ceny, popis kurzu, zobrazenie obsahu. Je možné získať prístup k kurzom aj bezplatne. Obsahuje modul ktorý zobrazuje užívateľovi absolvované kurzy a získané certifikáty.[18]

Vybrané webové aplikácie som zhodnotil podľa požiadavkou:

- Obsahuje registráciu/prihlásenie
- Možnosť predplatného
- Vyúčtovanie
- Obsahuje zobrazenie kategórie kurzov
- Zobrazenie certifikátov

- Zobrazenie obsahu kurzu

Všetky aplikácie ktoré sú spomenuté v tejto práci sú spoplatnené a ponúkajú možnosť registrácie takže som čerpal z informácií ktoré boli dostupné na ich weboch. Zo všetkých softvérov vyšla najlepšie aplikácia Udemy ktorá splnila najviac požiadavkou a ponúka skvelé možnosti a ponuky online kurzov a ponúka množstvo možnosti ako pracovať s ich aplikáciou.

2 POPIS POUŽITÝCH TECHNOLOGIÍ

2.1 ASP.NET

ASP.NET sa prvýkrát objavil v roku 2002. Bol súčasťou .NET Frameworku a nástupca za ASP Classic. ASP.NET je postavený na základoch technológie Common Language Runtime. Táto technológia umožňuje vyvíjať kód v ASP.NET s využitím ľubovoľného programovacieho jazyka ktorý je podporovaným .NET jazykmi. Podporované programovacie jazyky .Net sú napríklad F#, C#, VisualBasic atď.[12]

ASP.NET je opensource framework vyvinutý pre tvorbu webových aplikácií. Tento framework slúži pre tvorbu dynamických webových aplikácií. Súčasťou ASP.NET frameworku sú tri ďalšie frameworky s ktorými je možný vývoj webových aplikácií a nimi sú[12]:

- ASP.NET WEB Forms
- ASP.NET MVC
- ASP.NET Web Pages

2.1.1 ASP.NET WEB Forms

ASP.NET WEB Forms sú súčasťou frameworku ASP.NET a je to jeden z troch frameworkov ktoré sú určené k vývoju webových aplikácií. Ich základ je postavený na frameworku od spoločnosti Microsoft ASP.NET, pre fungovanie je kód vykonávaný na strane serveru a vytvára webové stránky dynamicky. Sú kompatibilné a ponúkajú využitie všetkých programovacích jazykov ktoré ponúka ASP.NET. Výhodou obsiahnutia v ASP.NET je že ponúkajú všetky funkcie ktoré ponúka tento framework Princíp fungovania ASP.NET WEB forms spočíva vo vyžiadaní požiadavky pre zobrazenie stránky od užívateľa internetovým prehliadačom. Web forms využívajú pre tvorbu stránok HTML súbory, správu servera, klientske skripty. Po vyžiadaní pre zobrazenie od užívateľa sa odošle požiadavka na server kde prejde kompiláciou a prebehne vytvorenie HTML súboru ktorý je predaný internetovému prehliadaču ktorý ho vykreslí. Toto umožňuje oddeliť logiku aplikácie, kód, súbory HTML od používateľského rozhrania. ASP.NET WEB forms a ASP.NET MVC využívajú pre zobrazenie obsahu užívateľovi Razor Pages. [20]

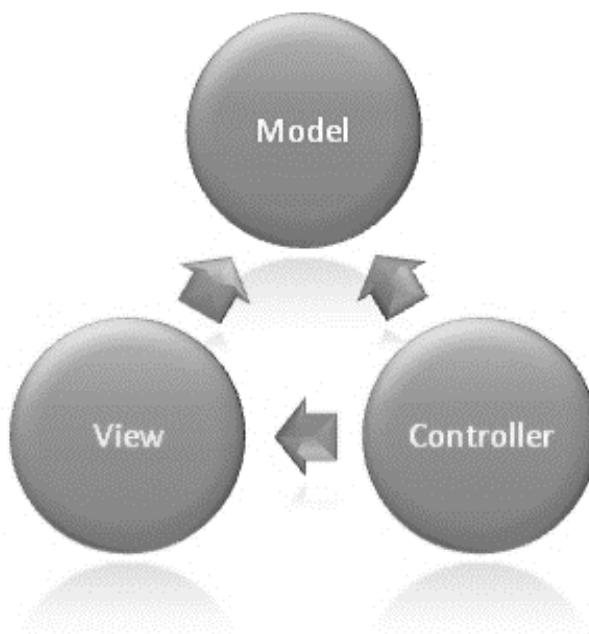
2.1.2 ASP.NET Web Pages

ASP.NET Web pages je jeden z frameworkov ktorý sa využíva pre tvorbu webových aplikácií kde je využitý ASP.NET. Tento Framework je využívaný pre vývoj webových stránok ktoré sú tvorené dynamicky. Klasické jednoduché webové stránky sú vytvárané staticky využívajú iba značkovací jazyk HTML a nevyužívajú žiadnu logiku kódu na strane servera. Webové aplikácie vytvorené pomocou ASP.NET používajú stránky ktoré sú vytvárané pomocou kódu. Jednou z výhod ASP.NET web pages je spracovanie vstupu od užívateľov, práca s databázami a ďalšie rôzne úkony ktoré umožňujú dynamické webové aplikácie čo umožňuje interagovať napríklad s rôznymi službami ktoré sú dostupné na internete. [21]

2.1.3 ASP.NET MVC

Technológia ASP.NET MVC využíva model ktorý sa nazýva model-view-controller. Tento model tvorí tvar webovej aplikácie a interakcie s komponentami ktoré webová aplikácia využíva. Technológia ASP.NET MVC je súčasťou frameworku ASP.NET od spoločnosti Microsoft ktorý slúži pre vývoj webových aplikácií. Architektonický vzor Model-View-Controller rozdeľuje aplikáciu na tri hlavné časti a nimi sú [11]:

- Model
- View
- Controller



Obrázok 1 : znázornenie modelu MVC.[11]

Tento model umožňuje rozdeliť logiku aplikácie. Vstupné požiadavky od používateľa spracováva controller ktorý pracuje s modelom. Controller vyberie view ktorý následne zobrazí všetky dáta modelu ktoré používateľ potrebuje.[11]

Výhodou webových aplikácií založených na ASP.NET MVC architektúre je že rozdeľuje zložitosť aplikácie na tri časti a tými sú view, controller a model. Webové aplikácie využívajúce MVC architektúru nevyužívajú formuláre založené na serveri a poskytuje úplnú kontrolu nad správaním aplikácie. Poskytujú lepšiu podporu pre testom riadený vývoj aplikácie. [11]

2.1.4 Model

Model je súčasťou vzoru MVC, zodpovedá všetkej logike ktorá súvisí s dátami s ktorými používateľ pracuje. Môžu sa prenášať medzi komponentami view a controller. Pre popis modelu sa používa trieda v C# údaje z modelu sa ukladajú do databázy. Model obsahuje údaje ktoré sú zobrazované užívateľovi.[11]

2.1.5 View

View je súčasťou vzoru MVC, predstavuje zobrazenie obsahu modelu používateľovi. View umožňuje používateľovi taktiež zobrazený obsah upravovať. Pre zobrazenie obsahu sa využívajú Razor view engine pre zobrazenie kódu pre používateľa. Vo views by mala byť minimálna logika kódu a mali by slúžiť iba pre zobrazenie obsahu. Views zasielajú

požiadavky pre vykonanie akcie controlleru ktorý ich následné obslúži a vráti príslušne view.[11]

2.1.6 Controller

Controller je súčasťou vzoru MVC, spracováva požiadavky od používateľa riadi logiku aplikácie. Pracuje s modelom aplikácie, zodpovedá za výber modelu s ktorým chce aplikácia pracovať, vyberá view pre zobrazenie obsahu pre používateľa. Ďalšou úlohou controlleru je pracovanie s modelom, to znamená pridávanie, úprava, mazania dát z modelu. [11]

2.1.7 Razor Pages

V ASP.NET Core vieme nájsť framework Razor Pages, tento framework vie zjednodušiť vytvorenie a úpravu jednotlivých stránok aplikácie ako je použitie controlleru alebo view. Tento framework je obsiahnutý v frameworku .NET, razor pages umožňujú vytvárať dynamické webové stránky. Razor Pages súbory končia na príponu .cshtml a sú súčasťou frameworku ASP.NET MVC. Razor Pages sú dizajnové pre jednoduchú implementáciu pri vývoji webových aplikácií. [6]

Kód v razor pages je podobný ako kód ktorý sa používa v aplikáciách ktoré používajú ASP.NET kde je využitý controller a view. Rozdiel spočíva využitím @page, ktorá spracováva požiadavky okamžite a nečaká na controller aplikácie. V razor pages sa musí nachádzať @page na začiatku súboru, taktiež @page ma vplyv na správanie konštruktora razor.[6]

2.2 Bootstrap

Bootstrap je open-source framework ktorý umožňuje rýchlo vytvárať responzívne rozhranie pre webové aplikácie. Tento framework je bezplatný a bol použitý prvýkrát v roku 2011 pre vývoji Twitteru. Je koncipovaný pre rýchle dizajnovanie vzhľadu webových aplikácií ktoré sú responzívne čo zabezpečuje že všetky prvky webovej aplikácie sa správne zobrazia v rozličných rozlíšeníach. Momentálne je najnovšia verzia frameworku bootstrap verzia 5. [14]

2.3 HTML

HTML inak nazývaný HyperText Markup Language, je značkovací jazyk pre vývoj web stránok, slúži k štruktúrovaniu obsahu stránok ako sú obrázky, text, odkazy a ďalší obsah. HTML stránky sú zobrazované klasickými webovými prehliadačmi. HTML sa používa ako

značkovací jazyk pre tvorbu základného skeletu web stránky. K tvorbe stránky sa využívajú HTML značky, tagy ktoré oddeľujú text a obsah ktorý má byť vykreslený na stránke. Tieto značky, tagy, sú tvorené začiatočnou (<TAG>) a koncovou značkou (</TAG>). HTML značky umožňujú tiež úpravu textu, napríklad farebné zvýraznenie, hrubé písmo, podčiarknutie textu a podobne. [13]

2.4 CSS

CSS alebo inak nazývané kaskádové štýly (Cascading Style Sheets) umožňujú vývojárovi prezvať kontrolu nad rozvrhnutím, dizajnom webových stránok. Kaskádové štýly umožňujú popisovať ako sa budú jednotlivé elementy značkovacieho jazyka HTML zobrazovať. Pomáha vytvárať web stránky ktoré budú vizuálne spracované dobre. S jazykom CSS je možné vytvárať rozne vizualne úpravy ako sú okraje, zmena veľkosti písma, zvýraznenia a podobne. Kaskádové štýly CSS sa nachádzajú v oddelenom súbore od HTML súboru čo znižuje duplicitnosť kódu uľahčuje údržbu webov, umožňuje zdieľanie štýlov medzi stránkami.[7]

2.5 Framework Entity

Entity framework je open source framework vyvinutý pre aplikácie .NET od spoločnosti Microsoft. Tento framework uľahčuje tvorbu a väzby medzi objektami v aplikácii do tabuliek v databázovom systéme. Výhodou entity frameworku je jednoduchosť, umožňuje napísať kód a systém automaticky vytvorí objekty z kódu [8]

Entity framework umožňuje vývojárom vytvárať objekty s ktorými môžu pracovať a následne ich ukladať do databázy. Vylučuje potrebu kódu pre prístup k dátam ktorý je potrebné napísať. [8]

Základom entity frameworku je model pomocou ktorého sa pristupuje k údajom. Model je zložený z entít a objektu ktoré reprezentujú reláciu medzi databázou a modelom. Umožňuje vytvoriť model už vytvorenej databáze, je možné ho ručne nakódovať alebo pomocou entity framework migrácií vytvoriť model databáze. Tento model je ďalej možné upravovať a vyvíjať s aplikáciou.[8]

2.6 Zabezpečenie ASP.NET

ASP.NET framework poskytuje vývojárom nástroje vďaka ktorým si môžu vývojári konfigurovať, nastavovať a spravovať zabezpečenie pri vývoji webovej aplikácie. ASP.NET

umožňuje zabezpečenie aplikácie pomocou autentifikácie, kontrolu oprávnení, ochranu údajov pred ich zneužitím. Obsahuje vynútenie HTTPS, ochranu pred správami CORS. Tieto nástroje umožňujú vytvoriť dostatočne zabezpečené ASP.NET webové aplikácie. ASP.NET obsahuje viacero nástrojov ktoré zabezpečujú aplikácie ako sú zabudovaný poskytovateľ identity ale je možné použiť aj služby tretích strán ktorými sú napríklad služba Facebook, LinkedIn a Twitter.[9]

2.6.1 Autentifikácia / Autorizácia

Autentifikácia je proces v ktorom sa overuje identita užívateľa, užívateľ poskytne overenia a tie sú následné porovnané s povereniami ktoré sú uložené v databázovom systéme alebo aplikácií. ASP.NET zaobstaráva funkciu autentifikácie služba `IAuthenticationService`, táto služba využíva registrované autentifikačné postupy k overeniu akcie s overovaním. Príkladom s overovacou akciou je overenie užívateľa a odpoveď keď sa neoverený užívateľ pokúša o prístup k obmedzenému zdroju.[9]

Autorizácia zahŕňa proces ktorý overuje to čo je užívateľovi umožnené vykonávať. Užívateľ ktorý má práva môže vytvárať dokumenty, pridávať dokumenty vykonávať ich úpravu alebo ich vymazať. ASP.NET poskytuje deklaratívnu rolu, model ktorý je založený na zásadách. Oprávnenie pre užívateľa je vyjadrené pomocou požiadavkou ktoré následne rutinu obslúžia a vyhodnotia identitu užívateľa a vlastnosti prostriedkov kam sa užívateľ pokúša získať prístup.[9]

2.6.2 Ochrana údajov

ASP.NET poskytuje kryptografické rozhranie (API) ktorými poskytuje ochranu dát. To zahŕňa správu kľúčov a rotáciu. Webové aplikácie potrebujú ukladať citlivé údaje. ASP.NET data protection k tomuto poskytuje jednoduché kryptografické rozhranie. Framework ASP.NET data protection stack je usporiadaný tak že je využívaný ako náhrada `<machineKey>` prvku v ASP.NET verzii 1-4. Táto funkcia bola vyvinutá tak aby upravili nedostatky starej verzie a zároveň aby poskytovala vývojárom už hotové riešenie pre použitie s ktorým sa môžu aplikácie stretnúť. Vo všeobecnosti sa odporúča v dnešnej dobe podporovať protokol HTTPS namiesto predchodcu http vo všetkých internetových webových aplikáciách. Aplikácie ktoré používajú ASP.NET používajú HTTPS redirection middleware na presmerovanie starých protokolov HTTP na zabezpečenú verziu HTTPS. [9]

2.6.3 Útoky Cross-Site request forgery (XSRF/CSRF)

Útoky cross-site request forgery, falšovanie žiadostí medzi webovými aplikáciami tiež označované ako XSRF alebo CSRF predstavujú útok na webovú aplikáciu tak že oklame užívateľa aby urobil nežiaduce kroky, tým môže ovplyvniť interakciu medzi klientom a webovou aplikáciou. Takéto útoky sú možné z dôvodu odosielania overovacích údajov automaticky z webových prehliadačov. Takáto forma útoku sa nazýva niekedy aj ako útok jedným kliknutím (one click attack). [9]

Príkladom útoku CSRF je keď sa užívateľ prihlási do webovej aplikácie, overí sa pomocou overovacieho formulára, následne užívateľ overí server a ten vráti odpoveď ktorá obsahuje overené súbory. Takáto lokalita je zraniteľná pred útokmi kvôli platnému overovaciemu súboru pretože dôveruje všetkým žiadosťami ktoré obdrží. Ďalej môže užívateľ navštíviť web ktorý obsahuje škodlivý formulár, ak užívateľ klikne na tlačidlo tak prehliadač automaticky vytvorí požiadavku a pridá overený súbor požadovanej domény. Následne požiadavok beží na serveri s overovacím súborom užívateľa a môže robiť zmeny na ktoré má užívateľ povolenia. [9]

Okrem takéhoto scenára môže škodlivá webová aplikácia spustiť skript ktorý automaticky odošle takýto škodlivý formulár, môže skryt škodlivý formulár pomocou CSS. Takéto scenáre nevyžadujú interakciu používateľa stačí ak navštíví škodlivú webovú aplikáciu. Používanie protokolu HTTPS nezabráni útoku typu CSRF. ASP.NET razor pages automaticky vkladajú tokeny proti antifalšovaniu do súborov HTML. [9]

2.6.4 Uplatnenie HTTPS v ASP.NET

ASP.NET framework uplatňuje jeden z internetových protokolov a ním je protokol hyper text transfer protokol secure inak nazývaný HTTPS. Tento protokol je vyžívaný na internete pre bezpečnú komunikáciu. Protokol HTTPS je vylepšenou verziou jeho predchodcu protokolu hyper text transfer protokol nazývaného HTTP. V dnešnej dobe sa odporúča využívať protokol HTTPS namiesto jeho predchádzajúcej verzie HTTP. Pri vývoji ASP.NET aplikácií sa odporúča používať HTTPS Redirection middleware pre presmerovanie z protokolu HTTP na novšiu zabezpečenejšou verziu HTTPS. [9]

2.6.5 CORS v ASP.NET

ASP.NET framework obsahuje funkciu ktorou je možné povoliť cross origin resource sharing (CORS) a umožňuje zdieľanie zdrojov webovej aplikácie pre inú aplikáciu ktorá využíva inú doménu. Zabezpečenie internetových prehliadačov zabraňujú vytvoreniu

požiadavku a jeho odoslaniu na inú doménu ako bola tá ktorá obsluhovala danú webovú aplikáciu. Takéto obmedzenie zo strany webového prehliadača sa nazýva politika rovnakého pôvodu a bráni v získavaní citlivých údajov z inej webovej aplikácie. CORS je súčasťou štandardu W3C, CORS nie je súčasťou zabezpečovacej funkcie iba rozvoľňuje zabezpečenie webovej aplikácie. Cors umožňuje povoliť a zbierať údaje z rôznych zdrojov a zároveň umožňuje údaje z iných zdrojov odmietnuť. [9][10]

2.6.6 Zabránenie cross site skriptovaniu

Cross site scripting (XSS), je skriptovanie medzi webovými aplikáciami. XSS je bezpečnostná chyba ktorá ohrozuje zabezpečenie webovej aplikácie a umožňuje útočníkovi umiestniť užívateľovi skripty do jeho webových aplikácií. Takéto skripty ktoré používajú útočníci sú vo väčšine vytvorené v jazyku JavaScript. Ak užívateľ načíta takto upravenú stránku skript od útočníka sa spustí čo umožní útočníkovi získať údaje od užívateľa jeho tokeny, umožní mu upravovať stránky webovej aplikácie cez DOM alebo presmerovať sa na inú stránku webovej aplikácie v internetovom prehliadači. Takýto útok sa vyskytuje vtedy ak aplikácia preberá vstup od užívateľa ktorého výstup vstupuje na stránku bez, kódovania alebo overovania.[9]

ASP.NET obsahuje modul Razor pages, stránky razor pages automaticky zakódujú výstup z aplikácie, používa kódovanie atribútov HTML vždy keď sa použije direktíva @. [9]

2.7 Microsoft Visual Studio

Microsoft Visual Studio je vývojové prostredie od spoločnosti Microsoft. Môže byť použitý pre vývoj konzolových aplikácií s grafickým rozhraním spolu s aplikáciami Windows Forms, webovými aplikáciami, webovými stránkami, a webovými službami ako v zdrojovom kóde, tak v riadiacom kóde na platformách Microsoft Windows, Windows Mobile, Windows CE, .NET, .NET Compact Framework a Microsoft Silverlight. Aktuálne je na trhu Visual Studio 2022. Visual Studio obsahuje editor kódu podporujúci Integrovaný debugger. Pracuje tak na úrovni kódu, ako aj na úrovni stroja. Je možné pridávať rozšírenie, čo vylepšuje funkčnosť na takmer každej úrovni. Microsoft Visual Studio 2022 podporuje programovanie v jazykoch C, C++, VisualBasic, C#, F#, JavaScript atď. Ruby, Node.js a M okrem iných, je dostupná prostredníctvom doplnkov. Java (a J#) boli podporované v minulosti. Microsoft Visual Studio je plne zadarmo pre študentov a individuálnych vývojárov.[15]

2.8 SQL

Jazyk SQL je najpoužívanější jazyk speciálně vytvorený pre tvorbu a návrh databázových systémov. Technológia SQL sa používa pre tvorbu relačných databázových systémov ktoré sa považujú za jednu z mnoha ďalších dôležitých technológií ktoré sa využívajú v softvérovom priemysle. Umožňuje ukladať dáta do tabuliek ktoré môžu byť prepojené spoločným údajom. SQL umožňuje vývojárom vytvárať databázové systémy do ktorých je možné vkladať údaje, editovať ich a získať iba určitú časť údajov. Táto technológia bola vyvinutá spoločnosťou IBM v sedemdesiatych rokoch dvadsiateho storočia, odvtedy však technológia pokročila dopredu a stala sa priemyselným štandardom. [4][5]

V dnešnej dobe je možné nájsť množstvo databázových systémov ktoré podporujú technológiu SQL a sú prevádzkované na osobných počítačoch alebo iných zariadeniach. [5]

SQL poskytuje vývojárom funkcie s ktorými môžu pristupovať do databázového systému ktoré zahrňujú, definíciu dát ktoré umožňujú vývojárovi vytvárať a štrukturovať údaje do tabuliek pomocou príkazu create, ďalej umožňuje navrhnuť vzťahy medy uloženými údajmi v tabuľkách. Ďalšou funkciou je načítanie dát z databázového systému, čo umožňuje vývojárovi načítať údaje a následne ich použiť, príkladom použitia funkcie pre vyhľadávanie je príkaz select. Ďalšou z funkcií jazyka SQL je manipulovanie z uloženými údajmi tým je myslená ich úprava, odstránenie z databázového systému alebo pridanie nového údaju do databázového systému. Príkladom využitia týchto funkcií sú príkazy delete, update. Jednou z funkcií je Riadenie prístupu do databázového systému to umožňuje vývojárovi obmedziť prístup pre používateľa k získavaniu, úprave dát čím ich ochraňuje pred neoprávneným prístupom príkladom využitia tejto funkcie je príkaz revoke. Zdieľanie dát je ďalšia funkcia ktorú poskytuje jazyk SQL, umožňuje zmeny ktoré učiní užívateľ nevymažú zmeny ktoré učinil súčasne s iným užívateľom. Poslednou z funkcií ktorú ponúka jazyk SQL je integrita dát, to znamená ochranu dát pred poškodením v kombinácii z aktualizáciami systému alebo ich zlyhaním.[5]

2.9 UML

UML (Unifitied modeling language) je jazyk ktorý slúži pre modelovanie softvérových systémov a systémov obecné. Často sa jazyk UML spája s modelovaním prednostne objektovo orientovaným softvérovým aplikáciám, je možné oveľa väčšie jeho využitie k modelovaniu softvéru. Pre lepšie pochopenie systému je dobré sa naň pozrieť z rôznych uhlov pohľadu. Týmito pohľadmi sú model požiadavkou, dynamické alebo statické modely

systému. Jazyk UML uľahčuje vývoj aplikácií a pomáha pochopiť softvér z rôznych pohľadov. Jazyk UML sa stal jedným z štandardov pri modelovaní softvérov. Po viacerých úpravách jazyka UML bol predložený návrh UML 1.1 a bol prijatý ako modelovací štandard v roku 1977. V roku 2005 vznikla vylepšená verzia jazyku UML ktorá sa nazýva UML2. [17]

Štruktúra jazyka UML sa rozdeľuje do troch častí tou sú stavebné bloky, spoločné mechanizmy a architektúra





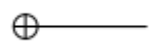

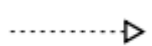
stavebné bloky obsahujú základne prvky modelovania napríklad diagramy a vzťahy.

Spoločné mechanizmy sú základné spôsoby ktorými vieme v jazyku UML splniť cieľ ktorý sme si stanovili na začiatku.

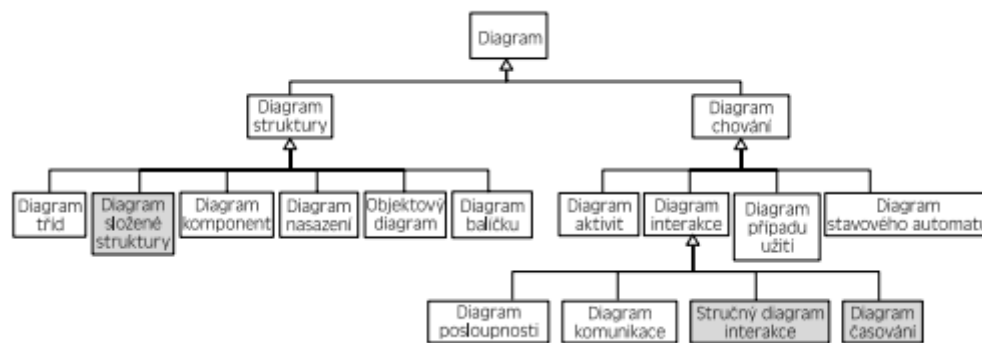
Architektúra je to samotný pohľad jazyka UML na navrhnutý softvér.

Jazyk UML umožňuje vytvoriť vzťahy medzi predmetmi v navrhovanom modeli. Vzťahy medzi predmetmi v návrhu nazývane tiež relácie umožňujú zachytiť vzťah medzi predmetmi. Jazyk UML poskytuje niekoľko typov vzťahov ktoré môžeme použiť pri tvorbe návrhu softvéru [16]

Tabuľka 1. Typy vzťahov UML

Názov Vzťahu	Syntax jazyka UML	Popis
Dependency		Zmena v predmete ovplyvňuje význam závisleho predmetu
Association		Spojenie medzi objektmi
Agregation		Druhý prvok je súčasťou prvého prvku
Composition		Lepšia forma agregácie s viac obmedzeniami
Containment		Prvý objekt obsahuje druhý objekt
Generalization		Prvý objekt je špecializáciou iného prvku a je možné ho nahradiť iným prvkom
Realization		Vzťah medzi klasifikátormi, jeden určuje dohodu, uskutočnenie obsluhuje druhý klasifikátor

V jazyku UML sa nachádzajú aj rôzne typy diagramov. Samotné diagramy sú pohľady na navrhovaný model. V jazyku UML existuje celkom trinásť rôznych typov diagramov. Diagramy môžeme rozdeliť na dva rôzne typy a to sú dynamické a statické. Dynamické zobrazujú ako jednotlivé predmety na seba navzájom pôsobia, kdežto statický model zobrazuje predmety a ich vzťahy medzi nimi [16]



Obrázok 2 : Schéma diagramov UML. [16]

II. PRAKTICKÁ ČÁST

3 NÁVRH APLIKÁCIE

Aplikácia pre ponuku online kurzov navrhnutá v tejto bakalárskej práci má rôzne druhy špecifikácií a požiadavkou. Základnou funkcionalitou je vytvorenie kurzu. Vytvorenie kurzu sa skladá:

- Vyplnenie informácií ako sú názov, popis, cena
- Vytvorenie obsahu kurzu
- Vyplnenie kategórie kurzu
- Nastavenie lektora kurzu

Ďalšou funkcionalitou je pridanie užívateľa do kurzu, registrácia užívateľa, prihlásenie užívateľa. Jednou z ďalších funkcionalít je spravovanie kurzov užívateľom ktoré pozostáva:

- Vyplnenie Emailovej adresy
- Zadanie používateľského mena
- Zadanie hesla

Ďalšou funkcionalitou aplikácie je vyhľadávanie kurzu podľa názvu pomocou vyhľadávača v aplikácii.

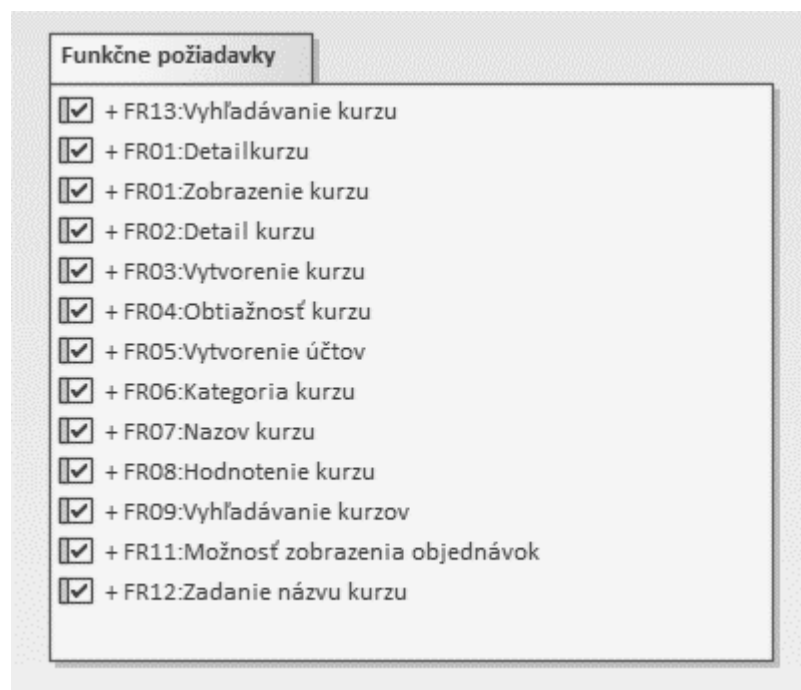
Všetky aktivity ktorými sú vytváranie, schvaľovanie a úprava sa vykonávajú pomocou webovej aplikácie a sú zapísané do databáze.

Predpokladom pre využitie takejto aplikácie je jednotlivец alebo firma ktorá chce poskytovať online kurzy. Predpokladom je že administrátor priradí lektora ku kurzu. Certifikáty o úspešnom dokončení kurzu sú predávané užívateľom ktorý úspešne zvládnu kurz. Takouto štruktúrou a spracovaním aplikácie pre ponuku online kurzov sa líši táto aplikácia od ostatných a je vyvinutá na základe takýchto požiadavkou.

3.1 Funkčné požiadavky

Pri tvorbe a návrhu softvéru je funkčný požiadavkou niečo čo definuje funkcionality samotného softvéru. Funkčné požiadavky definujú chovanie medzi vstupmi a výstupmi. Opisujú čo by mal daný softvér robiť ako by mal pracovať. Ako funkčné požiadavky som zvolil:

1. Aplikácia umožní vytvoriť nový online kurz
2. Aplikácia umožní zobrazit' detail kurzu
3. Aplikácia umožní vybrať kategóriu kurzu
4. Aplikácia umožní pridať hodnotenie kurzu
5. Aplikácia umožní zobrazit' kurzy
6. Aplikácia umožní správu účtov
7. Aplikácia umožní editáciu kurzu
8. Aplikácia umožní priradenie lektora kurzu
9. Aplikácia umožní vyhľadávanie kurzov podľa názvu

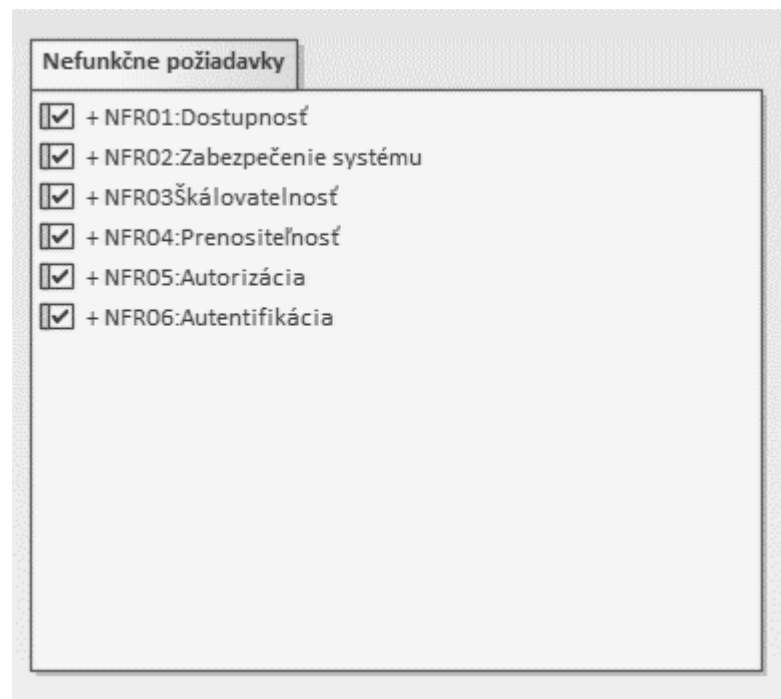


Obrázok 3 : Funkčné požiadavky

3.2 Nefunkčné požiadavky

Nefunkčné požiadavky pri návrhu softvéru definujú obmedzenia softvéru. Obmedzením softvéru môže byť výkon, zabezpečenie softvéru, zabezpečenie. Všetky tieto obmedzenia softvéru definujú obmedzenia výkonu softvéru. Pri návrhu tejto webovej aplikácie som ako nefunkčné požiadavky zvolil:

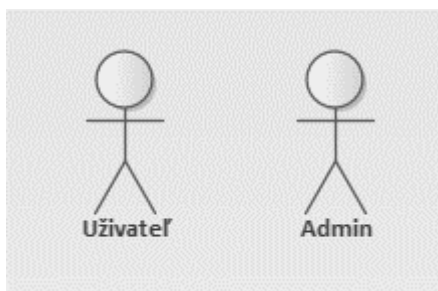
1. Škálovateľnosť aplikácie
2. Bezpečnosť aplikácie
3. Overenie používateľa
4. Dostupnosť
5. Prenositel'nosť



Obrázok 4 : Nefunkčné požiadavky

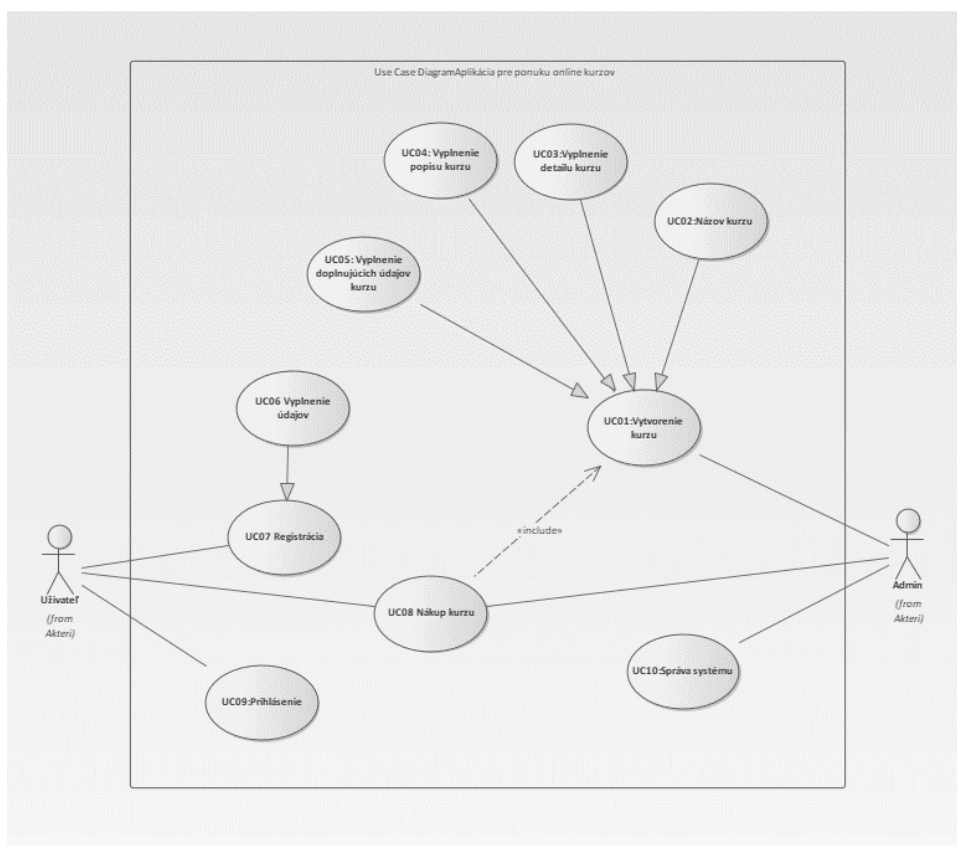
3.3 Model prípadu užitia

Model prípadu užitia opisuje činnosť softvéru obsahuje časti ako sú aktéri, modely use casey ktoré popisujú činnosť. Tieto modely umožňujú zobrazit' vzťahy medzi jednotlivými use case. Use case model tejto aplikácie obsahuje dvoch aktérov. Nimi sú admin a používateľ. Používateľ je osoba ktorá využíva aplikáciu vytvára objednávky, prihlasuje sa a pod. Admin je osoba ktorá spravuje aplikáciu. Vytvára kurzy, spravuje kurzy, upravuje ich. [22]



Obrázok 5 : Aktéri prípadu užitia

Model prípadu užitia (Use case) zachycuje chovanie systému z pohľadu užívateľa. Vzťahy medzi aktérmi a systémom sú znázornené vzťahmi. Model prípadu užitia na obrázku graficky znázorňuje ako daný systém vidí užívateľ zo svojho pohľadu.



Obrázok 6 : Model prípadu užitia

3.3.1 Scenáre

Scenáre popisujú jednotlivé use case modely pomocou scenárov detailnejšie. Každý scenár obsahuje jeho názov, názov use modelu, aktérov vstupné, výstupné podmienky a jednotlivé kroky scenárov ktoré sú vykonávané aktérom systému. Každý krok obsahuje popis. Scenáre use case modelu navrhovanej aplikácie:

Tabuľka 2. Scenár vytvorenia kurzu.

Názov Vytvorenie kurzu		
ID: UC01		
Charakteristika: Use case popisuje vytvorenie kurzu		
Primárny aktér: Admin		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Admin je prihlásený a vytvára kurz		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Admin	Vyplní názov kurzu > UC02
2	Admin	Vyplnenie detailu kurzu > UC2
3	Admin	Vyplnenie popisu kurzu > UC04
4	Admin	Vyplnenie doplňujúcich údajov kurzu > UC05
5	Admin	Admin uloží kurz do databáze

Tabuľka 3. Scenár zadania názvu kurzu.

Názov: Zadanie názvu kurzu		
ID: UC02		
Charakteristika: Use case popisuje zadanie názvu kurzu		
Primárny aktér: Admin		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Užívateľ je prihlásený ako admin		
Výstupní podmienky: Názov bude zadaný v kurze		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Admin	Zadá názov kurzu

Tabuľka 4. Scenár vyplnenia detailu kurzu.

Názov: Vyplnenie detailu kurzu		
ID: UC03		
Charakteristika: Use case popisuje vyplnenie detailu kurzu		
Primárny aktér: Admin		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Užívateľ je prihlásený ako admin		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Admin	Admin zadá detail kurzu

Tabuľka 5. Scenár vyplnenia popisu kurzu.

Názov: Vyplnenie popisu kurzu		
ID: UC04		
Charakteristika: Use case popisuje vyplnenie popisu kurzu		
Primárny aktér: Admin		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Užívateľ je prihlásený ako admin		
Výstupné podmienky: Popis kurzu bude zadaný v kurze		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Správca	Admin zadá popis kurzu

Tabuľka 6. Scenár vyplnenia doplňujúcich údajov.

Názov: Vyplnenie doplňujúcich údajov kurzu		
ID: UC05		
Charakteristika: Use case popisuje vyplnenie doplňujúcich údajov		
Primárny aktér: Admin		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Užívateľ je prihlásený ako admin		
Výstupné podmienky: Doplňujúce údaje kurzu bude zadané v kurze		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Admin	Admin zadá lektora kurzu
2	Admin	Admin zadá kanál kurzu
3	Admin	Admin zadá obtiažnosť kurzu
4	Admin	Admin zadá cenu kurzu

Tabuľka 7. Vyplnenie údajov

Názov: Vyplnenie údajov		
ID: UC06		
Charakteristika: Use case popisuje vyplnenie údajov používateľa		
Primárny aktér: užívateľ		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: /		
Výstupné podmienky: Údaje budú vyplnené		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Užívateľ	Užívateľ vyplní potrebné údaje

Tabuľka 8. Scenár registrácie.

Názov: Registrácia		
ID: UC07		
Charakteristika: Use case popisuje registráciu užívateľa		
Primárny aktér: Užívateľ		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Zadanie vyplnených údajov		
Výstupné podmienky: Užívateľ bude zaregistrovaný		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Užívateľ	Užívateľ skontroluje vyplnenie údajov
2	Užívateľ	Užívateľ odošle registračný formulár

Tabuľka 9. Scenár nákupu kurzu.

Názov Nákup kurzu		
ID: UC08		
Charakteristika: Use case popisuje nákup kurzu		
Primárny aktér: Užívateľ		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Zadanie prihlasovacích údajov		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Užívateľ	Užívateľ pridá kurz do košíka
2	Užívateľ	Užívateľ skontroluje košík
3	Užívateľ	Užívateľ vyberie matnosť platby
4	Užívateľ	Užívateľ zadá platobné údaje
8	Užívateľ	Užívateľ prevedie platbu

Tabuľka 10. Scenár prihlásenia.

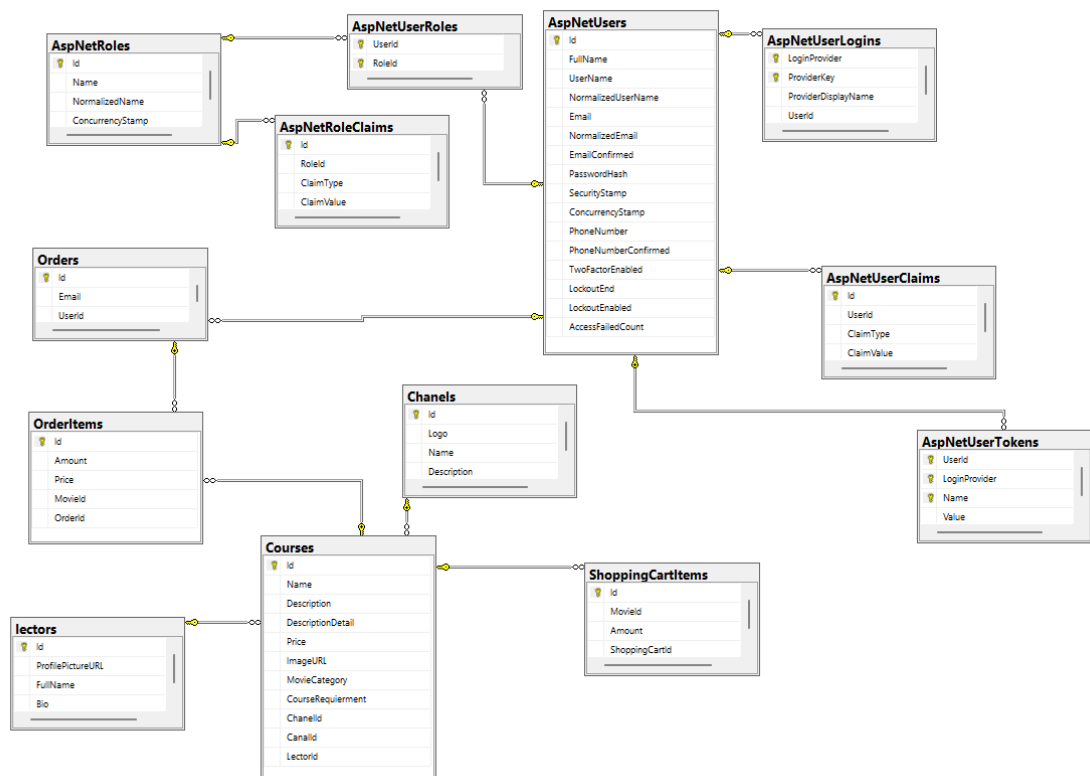
Názov: Prihlásenie		
ID: UC09		
Charakteristika: Use case popisuje prihlásenie užívateľa		
Primárny aktér: Užívateľ		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Užívateľ je zaregistrovaný		
Výstupné podmienky: Užívateľ je úspešne prihlásený		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Užívateľ	Užívateľ vyplní prihlasovací formulár
2	Užívateľ	Užívateľ potvrdí prihlasovací formulár
3	Užívateľ	Užívateľ sa prihlási

Tabuľka 11. Scenár správy systému.

Názov: Správa systému		
ID: UC10		
Charakteristika: Use case popisuje správu systému admina		
Primárny aktér: Admin		
Vedľajší aktéri: Nie sú		
Vstupné podmienky: Užívateľ je prihlásený ako admin		
Výstupné podmienky: Užívateľ vykoná zmeny v systéme		
Hlavní scenár:		
Krok	Aktér/Systém	Popis
1	Admin	Admin prevedie úpravu v systéme
2	Admin	Admin uloží úpravu systému

3.4 E-R diagram databázový model

E-R diagram nazývaný entitno relačný diagram ktorý popisuje vzťahy medzi entitami údajov v tabuľkách ktoré sú súčasťou databáze. Modelovaním databáz môžeme skrátiť vývoj aplikácie. Modelovanie E-R diagramov používa rôzne nástroje ktorými môžeme znázorniť dôležité prepojenie údajov medzi tabuľkami a tým tak navrhnuť ich vzťahy. Databázová schéma aplikácie obsahuje trinásť tabuliek, entít, v ktorých sú uložené údaje. Vzťahy medzi tabuľkami sú reprezentované pomocou čiar.



Obrázok 7 : Databázová schéma.

Pre vytvorenie tabuliek a celkovo databázovej schémy bol využitý Entity framework a migrácie. Model je vytvorený ako trieda (class) obsahuje informácie o entitách ktoré majú byť obsiahnuté v tabuľkách. Obsahujú informácie o primárnych, cudzích kľúčoch a pod. Pomocou techniky code first s využitím entity frameworku sa z modelu pomocou migrácií sa automaticky z modelu tried vygeneruje databáza.

```

<# references
public class Course:IEntityBase
{
    //id primarny kluč
    [Key]
    19 references
    public int Id { get; set; }
    //nazov kurzu
    16 references
    public string Name { get; set; }
    //popis kurzu
    12 references
    public string Description { get; set; }
    //podrobný popis kurzu
    9 references
    public string DescriptionDetail { get; set; }
    //cena
    16 references
    public double Price { get; set; }
    //nastavenie grafiky
    11 references
    public string ImageURL { get; set; }
    //kategorie kurzov
    11 references
    public CourseCategory MovieCategory { get; set; }
    // poziadavky pre kurz
    9 references
    public CourseRequierment CourseRequierment { get; set; }
    //Prepojenie tabuliek
    //Tvorca prepojenie na tabulku
    10 references
    public int ChanelId { get; set; }
    [ForeignKey("CanalId")]
    4 references
    public Canal Canal { get; set; }
    //Lektor prepojenie na tabulku
    10 references
    public int LectorId { get; set; }
    [ForeignKey("LectorId")]
    3 references
    public Lector Lector { get; set; }
}

```

Obrázok 8 : Model triedy kurzu.

Tabuľky vytvorené pomocou ASP.NET majú na začiatku názvu ASPNET, využívajú rozhranie od ASP.NET Core Identity ktorá podporuje rozhranie pre prihlasovanie a funkcionality s tým spojenú. To zahŕňa správu užívateľov, hesiel, emailov a pod. V návrhu aplikácie sú modely tried:

- AspNetRoles – obsahuje údaje o rolách a oprávneniach k funkcionalitám
- AspNetRolesClaims - ukladá rolu priradenú užívateľovi
- AspNetUserRoles – prepája užívateľa a rolu v systéme
- AspNetUserClaims – ukladá opravenia užívateľa
- AspNetUsers – údaje o užívateľoch registrovaných v systéme
- AspNetUserLogins – ukladá údaje pre prihlásenie z tretích strán
- AspNetUserTokens – ukladá overovacie tokeny užívateľa
- Lectors – obsahuje údaje spojené s lektormi kurzu

- ShoppingCartItems – obsahuje údaje o kurzoch ktoré sa nachádzajú v nákupnom košíku
- Orders – obsahuje údaje o objednávke a užívateľovi ktorý objednávku vytvoril
- OrderItems – obsahuje údaje o kurze ktorý bol objednaný
- Courses – obsahuje údaje o online kurzoch
- Channels – obsahuje údaje o kanáloch online kurzov

3.5 Vytvorenie kurzu

Vytvorenie online kurzu v aplikácii pozostáva z niekoľkých krokov. Pre vytvorenie kurzu je potrebné byť prihlásený ako admin inak nie je možné vytvárať nové online kurzy. Admin zadá potrebné informácie ktoré sú potrebné pre vytvorenie kurzu ktorými sú názov, popis a cena. Ďalej je potrebné vyplniť grafiku online kurzu ktorá bude zobrazená ako náhľad kurzu. Jedným z ďalších krokov pri vytváraní kurzu je vyplnenie popisu obsahu kurzu ktorý bude obsahovať. Ďalším krokom je priradenie lektora zo zoznamu ku kurzu, priradenie kanálu ktorý kurz spravuje, nastavenie ťažnosti kurzu a výber kategórie kurzu. Posledným krokom je vytvorenie kurzu.

4 TVORBA APLIKÁCIE

Aplikácia bolo vyvinutá na základe požiadavkov a návrhu aplikácie

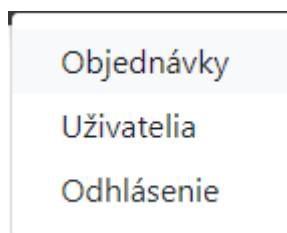
4.1 Náhľad aplikácie

Navrhnutú aplikáciu môžeme rozdeliť na tri základné časti. Prvou je Navigačný navbar ktoré obsahuje prepojenie na hlavnú stránku, vyhľadávacie pole pre vyhľadanie kurzu a ikonu nákupného košíku. Odkazy v navbare stránky sa menia na základe role pod ktorou je užívateľ prihlásený.



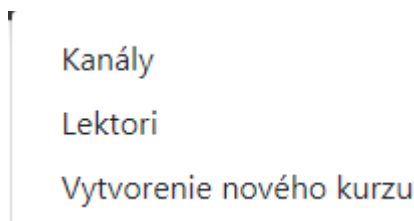
Obrázok 9 : Navbar stránky.

Po prihlásení do systému aplikácie sa zobrazí užívateľovi v pravom hornom rohu jeho údaje a ponuka v ktorej si môže zobrazit' z ponuky výberu, objednaných kurzov a odhlásenia.



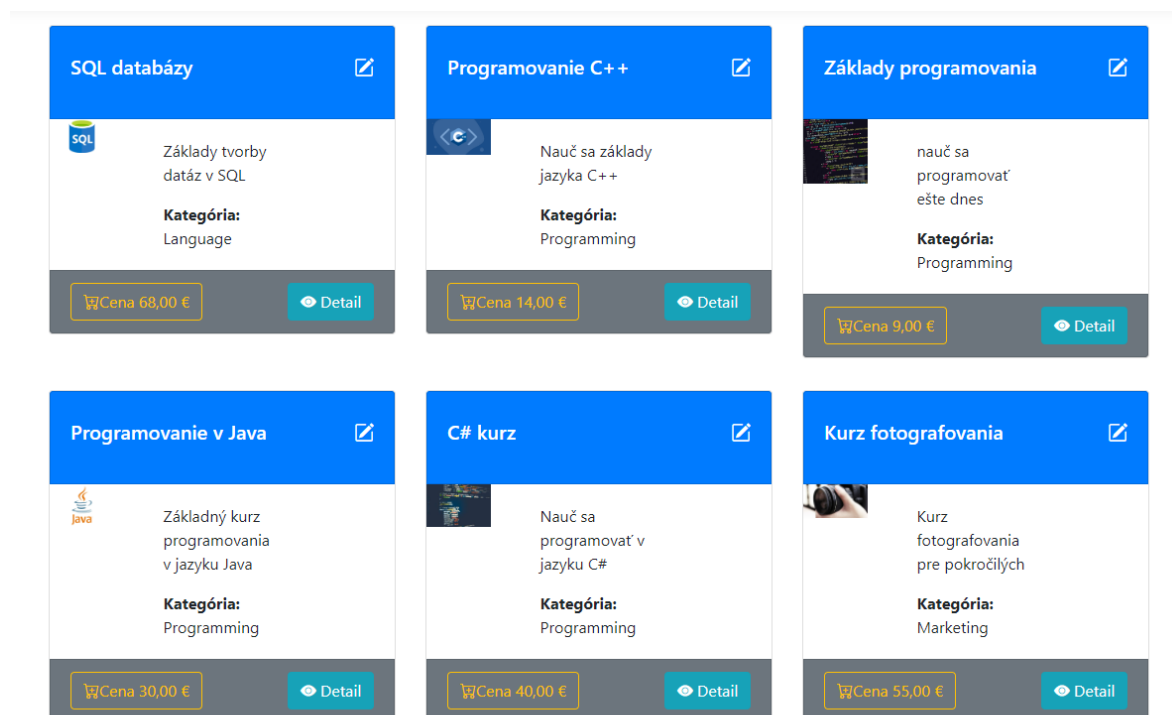
Obrázok 10 : Výber admina.

Ak je užívateľ prihlásený ako admin stránky tak sa v navbare stránky zobrazí odkaz správa stránky ktorá ponúka adminovi možnosti úprav kanálov ktoré dane kurzy obsahujú, možnosť pridať a editovať lektorov a možnosť pridať a editovať kurzy stránky.



Obrázok 11 : Výber správy stránky.

Druhou časťou stránky je zobrazenie tela stránky ktoré sa líši od zvoleného odkazu. Hlavným zobrazením tela stránky je zobrazenie obsahu kurzov.



Obrázok 12 : Zobrazenie kurzov stránky

Tret'ou a poslednou časťou stránky je pätička stránky.



Obrázok 13 : Pätička stránky.

4.2 Prihlásenie do aplikácie

Prihlásenie do aplikácie je potrebné pre užívateľa aby mohol pracovať s údajmi ktoré obsahuje stránka a mal možnosť vytvárať objednávky. Prihlásenie do aplikácie prebieha pomocou jednoduchého formulára pre prihlásenie kde užívateľ zadá email ktorým sa prihlasuje do aplikácie a následne zadá svoje heslo. Po prihlásení je užívateľ presmerovaný na hlavnú stránku stránky.

Prihlásenie k účtu

Email address

Password

Log in

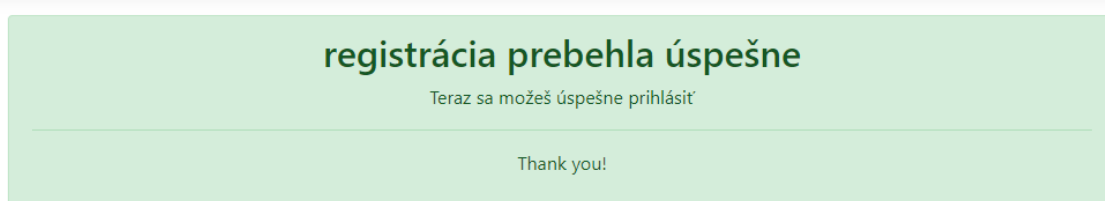
Are you here for the first time? Create a free account

[Register](#)

Obrázok 14 : Prihlasovací formulár.

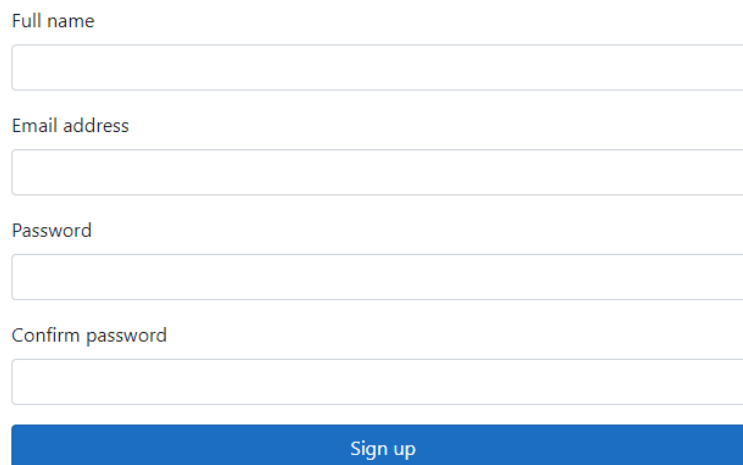
4.3 Registrácia do aplikácie

Registrácia do aplikácie prebieha pomocou ASP.NET Core Identity. Pre správne vytvorenie účtu pri registrácii je potrebné vyplniť správne registračný formulár ktorý obsahuje meno, priezvisko, email, heslo. Podmienkou registrácie je vytvorenie silného hesla, ktoré musí obsahovať požiadavky ako sú špeciálne znaky, veľké písmená a čísla. Ak sa pri registrácii heslá nezhodujú užívateľ je na túto skutočnosť upozornený. Ďalším ošetrením je ak sa už daná emailová adresa používa tak je užívateľ upozornený že daná emailová adresa je už použitá. Po úspešnej registrácii sa užívateľovi zobrazí upozornenie o úspešnej registrácii. Registračný formulár a upozornenie o úspešnej registrácii sú zobrazené na obrázkoch nižšie.



Obrázok 15 : Zobrazenie úspešnej registrácie.

Registrácia



Full name

Email address

Password

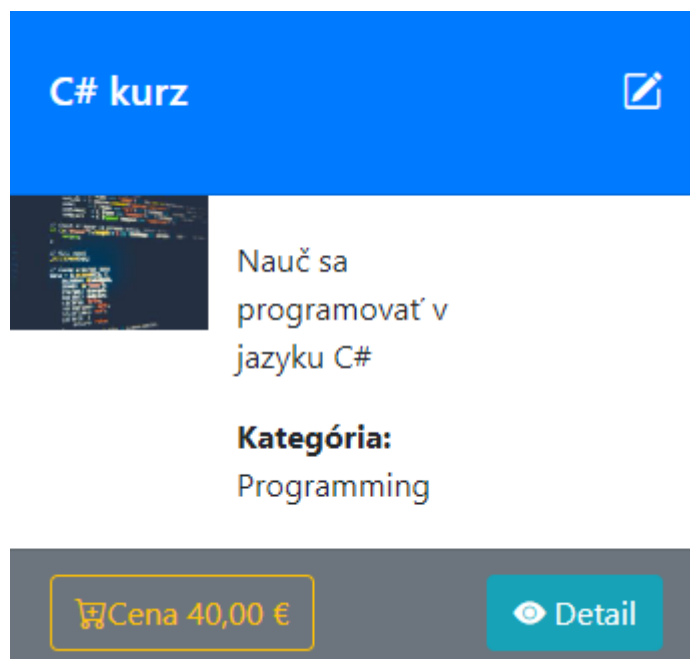
Confirm password

Sign up

Obrázok 16 : Registračný formulár.

4.4 Zobrazenie kurzov

Hlavná stránka aplikácia zobrazuje všetky online kurzy. Všetci užívatelia si môžu zobrazit' všetky kurzy. Kurz zobrazuje základné informácie o kurze, jeho jednoduchý popis a kategóriu do ktorej je kurz zaradený. Zobrazenie karty kurzu môžeme rozdelit' na tri časti. V hlavičke kurzu je uvedený jeho názov. Telo kurzu obsahuje krátky popis ktorý oboznamuje užívateľa čo kurz ponúka, kategóriu kurzu. Pätička karty kurzu zobrazuje cenu a detail kurzu, obsahuje dve tlačidlá jedno je pridanie kurzu do nákupného košíka a druhým si môžeme zobrazit' detailný obsah kurzu.



Obrázok 17 : Karta kurzu.

4.4.1 Zobrazenie detailu kurzu

Stránka obsahuje zobrazenie detailu kurzu. Po kliknutí na tlačidlo detail v karte kurzu sa zobrazí stránka detailu kurzu. Stránka detailu kurzu obsahuje všetky vyplnené údaje ktoré boli zadane pri tvorbe kurzu. Obsahuje základný popis. V detaile kurzu sa užívateľovi zobrazí lektor kurzu na ktorý môže užívateľ kliknúť a zobraziť si údaje o lektorovi kurzu, kategória kurzu. Užívateľovi sa zobrazí podrobnejší popis kurzu ktorý popisuje detailnejšie obsah kurzu. Zobrazí sa grafika kurzu. V pätičke karty detailu kurzu je možnosť vrátiť sa späť na všetky kurzy, zobrazenie ceny a pridanie kurzu do košíku. Obrázok nižšie zobrazuje detail karty kurzu.

Programovanie C++



Nauč sa základy jazyka C++

Lektor: Michal Ovenos



Kategória: Programming

Požiadavky pre zvládnutie kurzu: Zaciatočník

Popis obsahu kurzu:
Kurz C++ Elementary je online kurz určený pre každého, kto chce začať programovať, alebo aspoň pre toho, kto sa chce učiť vyššie programovacie jazyky, medzi ktoré patrí aj jazyk C++. Programovanie v jazyku C++ je náročnejšie, preto to nemožno zhrnúť v jednom online kurze. Práve preto pre teba pripravujeme ďalšie kurzy jazyka C++, ktoré budú postupne na seba nadväzovať. Jazyk C++ je stále veľmi populárnym jazykom. Od obdobia, kedy boli publikované prvé štandardy jazyka C++, sa veľmi rýchlo posunul na top pozície a tam si drží akúsi rovnakú líniu dlhdo. V oblasti vyšších programovacích jazykov mu dobre konkuruje len jazyk Java. C++ je jazyk objektovo orientovaný, ktorý podporuje sekvenčné, procedurálne i generické programovanie. Medzi jeho prednosti patria výkonný kompilátor, rýchly spustiteľný a ľahko udržiavateľný kód. Tento jazyk je súčasťou každého štúdia, ktoré je zamerané na informačné technológie. Učí sa aj na iných odboroch technického zamerania. Používa sa v oblasti ekonomiky, videnia počítača, počítačovej grafike, počítačových hrách, vstavaných systémoch, komunikačných systémoch, tiež v automatizácii a robotike. Jazyk C++ bude pre teba prínosom. Môže sa stať dokonca tvojím hlavným programovacím jazykom. Myslím si, že každý dobrý programátor by mal vedieť aspoň jeden vyšší programovací jazyk, ak nie dokonca viac. Tento kurz sa snaží pozeráť na problematiku od základov a veľmi detailne, preto neľutuješ, že si ho absolvoval.

Cena 14,00 €

Späť

Obrázok 18 : Karta detailu kurzu.

4.4.2 Vytvorenie kurzu

Vytvorenie nového kurzu prebieha pomocou formulára ktorý je potrebné vyplniť. Vytvorenie nového kurzu je možné po kliknutí na odkaz vytvorenie nového kurzu. Tvorba kurzu pozostáva z vyplnenia všetkých potrebných údajov o kurze ako sú popis, detailný popis, cena, vyplniť grafiku kurzu, vybrať obťažnosť kurzu z troch možných kategórií ktorými sú začiatocník, pokročilý a expert. Rovnaký postup je pre vyplnenie kanálu kurzu, vyplnenie lektora kurzu a kategórie z možností na výber. Po vyplnení týchto údajov je kurz možné úspešne vytvoriť pomocou create. Vytvorenie kurzu obsluhuje metóda create ktorá spolu s view modelom newCourseVM obsluhujú tvorbu nového kurzu. NewCourseVM popisuje ktoré údaje sú povinné pre vyplnenie pomocou parametru „Required“ pomocou parametru „Display“ zobrazujú názov do view.

Vytvorenie online kurzu

Názov kurzu	Kanál kurzu
<input type="text"/>	Online Kurzy
Cena	Kategória
<input type="text"/>	Programming
Náhľad kurzu	Požiadavky pre zvládnutie kurzu
<input type="text"/>	Zaciatocnik
	Lektor kurzu
	Dalibor Ďuriš
Popis online kurzu	
<input type="text"/>	
Popis obsahu kurzu	
<input type="text"/>	
<input type="button" value="Show all"/>	<input type="button" value="Create"/>

Obrázok 19 : Formulár tvorby kurzu.

4.4.3 Editácia kurzu

Editácia kurzu je možné z hlavnej stránky aplikácie po prihlásení k účtu admina. V pravom hornom rohu karty kurzu sa zobrazí ikona pre editáciu kurzu. Editácia kurzu prebieha podobne ako tvorba kurzu. Admin po editácii kurzu môže uložiť údaje. Po editácii kurzu je admin presmerovaný na hlavnú stránku aplikácie.

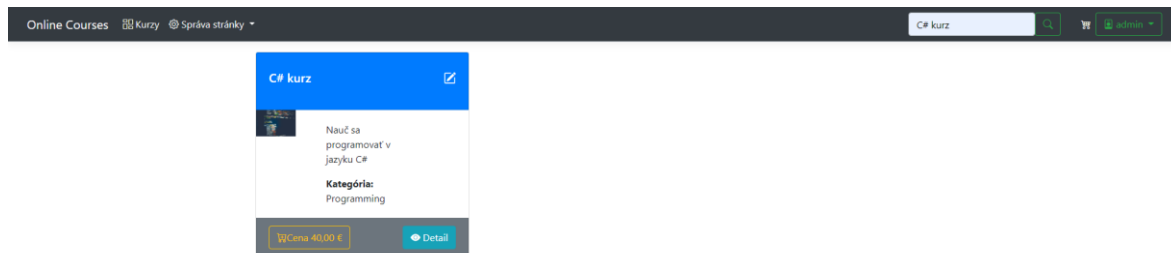
4.5 Lektori

Lektori sú časť stránky ktorú si môže zobrazit' užívateľ po kliknutí karty detailu kurzu. Lektor je priradený ku kurzu a užívateľ si môže zobrazit' základné informácie o lektorovi kurzu, ktorými sú meno a krátky popis lektora.

Pridávanie lektorov do aplikácie má na starosti admin ktorý môže pridať lektorov do aplikácie pomocou odkazu lektori a následne pridať. Admin vyplní informácie o lektorovi a tým vytvorí lektora ktorý môže byť priradený kurzu. Admin môže editovať a odstraňovať lektorov z kurzov a zobrazit' si ich detail. Pre vymazanie lektora z kurzu sa zobrazí adminovi stránka ktorá vyžaduje potvrdenie pre vymazanie lektora z databázy aby nedošlo k nechcenému odstráneniu lektora.

4.6 Vyhľadavanie v aplikácií

Aplikácia umožňuje vyhľadavanie kurzov podľa zadaných parametrov. Jedným z týchto parametrov je vyhľadavanie podľa názvu kurzu. Vyhľadavanie prebieha pomocou search baru ktorý je umiestnený v navbare stránky, akýkoľvek užívateľ zadá názov kurzu do vyhľadávača a prebehne vyhľadanie kurzu v databáze. Ak daný kurz nie je v databáze nedôjde k zobrazeniu kurzu ak sa kurz v databáze nachádza kurz sa zobrazí na stránke.



Obrázok 20 : Vyhľadavanie v aplikácií.

Vyhľadavanie v aplikácií obsluhuje metóda „Filter“ ktorá využíva ako vstupný parameter ktorý zadal užívateľ do vyhľadávacieho poľa (String) ktorý porovná s názvom kurzov ktoré sú uložené v databáze a následne vráti view kde je vyfiltrovaný kurz.

```
public async Task<IActionResult> Filter(string searchString)
{
    var courses = await _service.GetAllAsync(n => n.Canal);

    if (!string.IsNullOrEmpty(searchString))
    {
        var filter = courses.Where(n => string.Equals(n.Name, searchString, StringComparison.CurrentCultureIgnoreCase) ||
            string.Equals(n.Description, searchString, StringComparison.CurrentCultureIgnoreCase)).ToList();

        return View("Index", filter);
    }

    return View("Index", courses);
}
```

Obrázok 21 : Kód metódy „Filter“.

4.7 Správa užívateľov

Aplikácia umožňuje správu a zobrazenie užívateľov adminovi zobrazíť užívateľov ktorý sa nachádzajú. Každý novo registrovaný užívateľ sa zobrazí v sekcii užívateľa. Prístup do tejto sekcie je obmedzený pre iba admina.

List of all users

Full name	UserName	Email
Admin	admin	admin.com
Test	test	test@email.com

Obrázok 22 : Správa užívateľov.

4.8 Nákupný košík

Aplikácia obsahuje modul nákupného košíka. Pre prístup do nákupného košíka musí byť užívateľ prihlásený. Ak užívateľ nie je prihlásený a chce pridať kurz do košíku je automaticky presmerovaný na prihlasovací formulár. Po prihlásení užívateľ môže pridať kurz do košíku pomocou tlačidla rohu karty kurzu, poprípade v karte detailu kurzu. Po pridaní kurzu do košíku sa kurz zobrazí v košíku. Pridávanie kurzov do košíku ma na starosti metóda „AddItemToShoppingCart“. Metóda získa Id kurzu ktorý následne predá metóde „addItemToCart“ a ako posledné sa zavolá metóda ShoppingCart ktorá zobrazí kurz v nákupnom košíku.

```
0 references
public async Task<IActionResult> AddItemToShoppingCart(int id)
{
    var item = await _moviesService.GetMovieByIdAsync(id);

    if (item != null)
    {
        _shoppingCart.AddItemToCart(item);
    }
    return RedirectToAction(nameof(ShoppingCart));
}
```

Obrázok 23 : Kód pridania kurzu do košíku.

Zobrazenie nákupného košíka zobrazuje počet kurzov ktoré sú v košíku, názov kurzu, cenu kurzu a celkovú sumu.

```

public IActionResult ShoppingCart()
{
    var items = _shoppingCart.GetShoppingCartItems();
    _shoppingCart.ShoppingCartItems = items;

    var response = new ShoppingCartVM()
    {
        ShoppingCart = _shoppingCart,
        ShoppingCartTotal = _shoppingCart.GetShoppingCartTotal()
    };

    return View(response);
}

```

Obrázok 24 : Kód zobrazenia kurzov v košíku.

Nákupný košík umožňuje odstrániť kurzy z košíku túto funkciu obsluhuje metóda „RemoveItemFromSchoppingCart“ ktorá získa identifikátor produktu v košíku, následne zavolá metódu „RemoveItemFromCart“ tá načíta údaje z databázy skontroluje či sa daný kurz nachádza v databáze košíku a vymaže ho pomocou „Remove“.

```

0 references
public async Task<IActionResult> RemoveItemFromShoppingCart(int id)
{
    var item = await _courseService.GetMovieByIdAsync(id);

    if (item != null)
    {
        _shoppingCart.RemoveItemFromCart(item);
    }
    return RedirectToAction(nameof(ShoppingCart));
}

```

Obrázok 25 : Funkcia pre odobratie kurzu.

Nákupný košík			
Počet	Kurz	Cena	Spolu
1	Programovanie v Java	30,00 €	30,00 €
		Total:	30,00 €




Obrázok 26 : Nákupný košík.

4.8.1 Vybavenie objednávky

Objednávku je možné vybaviť pomocou zaplattenia pomocou PayPal pre „zaplatenie“ objednávky užívateľ využije tlačidlo PayPall. Po kliknutí na toto tlačidlo je užívateľ presmerovaný na okno ktoré ponúka dve možnosti a to je platba pomocou PayPol. Užívateľ

si môže vybrať z dvoch možností, môže sa prihlásiť k svojmu účtu PayPal alebo si ho môže vytvoriť.

PayPal

Pay with PayPal

Enter your email or mobile number to get started.

Next

or

Pay with Debit or Credit Card

[Cancel and return to Test Store](#)

[English](#) | [Français](#) | [Español](#) | [中文](#)

[Contact Us](#) | [Privacy](#) | [Legal](#) | [Policy Updates](#) | [Worldwide](#)

Obrázok 27 : Platba pomocou PayPal.

Platbu pomocou playpol obsluhuje skript na stránke ktorý generuje tlačidlo a následne presmeruje užívateľa na stránku kde sa môže uskutočniť platbu za kurz dvoma možnosťami. Táto funkcia využíva rozhranie api ktorú poskytuje PayPal. Po úspešnej platbe pomocu tohto scriptu sa objednávka presunie do uskutočnených objednávok.


```
@section Scripts{  
  
    <script src="https://www.paypalobjects.com/api/checkout.js"></script>  
  
    <script>  
  
        let _total = @Html.Raw(Model.ShoppingCartTotal);  
  
        paypal.Button.render({  
            env: 'sandbox',  
            client: {  
                sandbox: 'AcKRTgL4i68EZ2ZYUx3iIihAg2YhL5dtF_m19A8MjBWVOn6ewNS4M_uV8gA3xi98zTcUq38gUHbVFX'  
            },  
  
            locale: 'en_US',  
            style: {  
                size: 'small',  
                color: 'gold',  
                shape: 'pill'  
            },  
            commit: true,  
  
            //Set up a payment  
            payment: function (data, actions) {  
                return actions.payment.create({  
                    transactions: [{  
                        amount: {  
                            total: _total,  
                            currency: 'USD'  
                        }  
                    }]  
                });  
            },  
  
            //Execute the payment  
            onAuthorize: function (data, actions) {  
                return actions.payment.execute().then(function () {  
                    var url = '@Url.Action("CompleteOrder", "Orders", new { })';  
                    window.location.href = url;  
                });  
            }  
        }, '#paypal-btn')  
    </script>  
}
```

Obrázok 28 : Funkcia pre platbu cez PayPol.

4.9 Správa objednávok

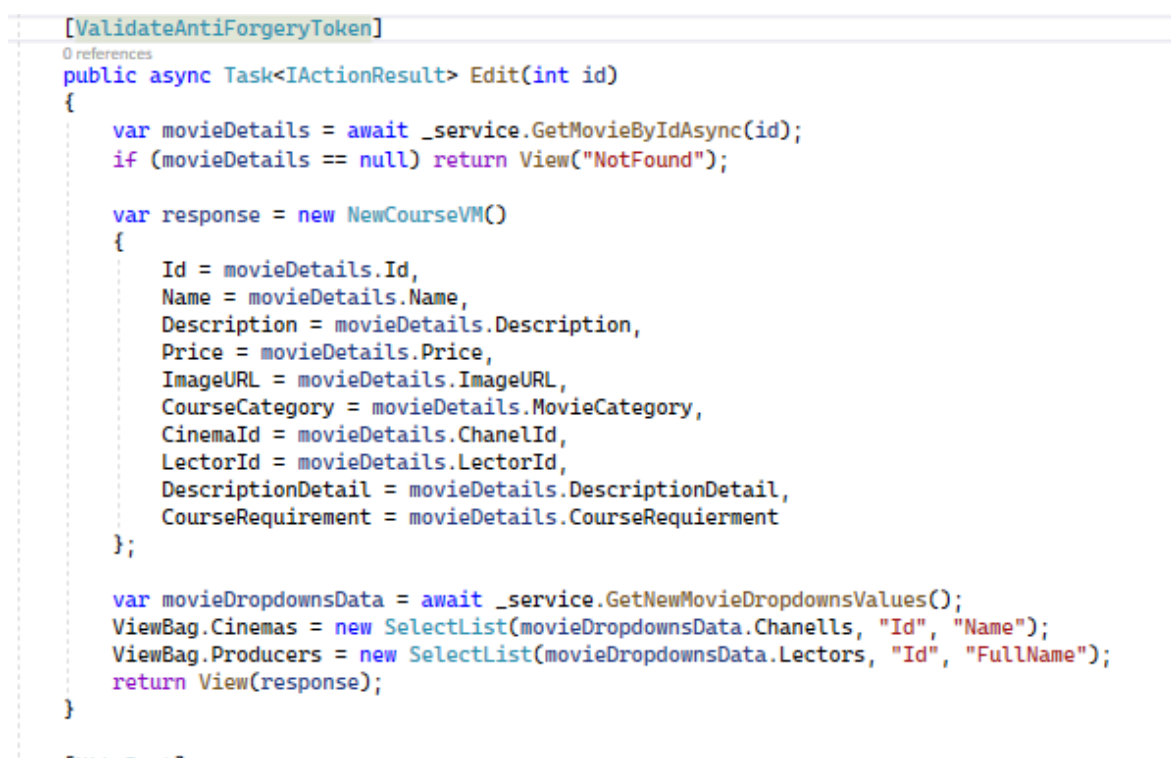
Každý užívateľ má možnosť zobrazit' svoje uskutočnené objednávky. Po prihlásení do svojho účtu je má možnosť zobrazenia svojich objednávok. Ich zobrazenie obsahuje číslo objednávky, názov kurzu a sumu.

5 ZABEZPEČENIE APLIKÁCIE

Pri vývoji aplikácie bolo jedným z bodov zabezpečenie aplikácie ASP.NET ponúka niekoľko techník ktorými vieme zabezpečiť aplikáciu.

5.1 Zabezpečenie CSFR útoku

Pre zabezpečenie proti útokom typu CSFR je možné použiť funkcie ktoré ponúka framework ASP.NET. Jednou z možností ako zabrániť útokom typu CSRF je pridanie tokenov proti falšovaniu, v ASP.NET nazývaných „Anti-Forgery Tokens“. Tieto tokeny zablokujú akcie ktoré obsahujú tokeny ktoré nie sú validné. Využitie Anti-Forgery Tokens je možné nastaviť v html súbore kde stačí využiť atribút post. Ďalšou z možnosti ochrany proti útokom typu CSFR je využitie autentifikácie. Príkladom využitia tokenov pre zabránenia falšovania je pridanie atribútu „ValidateAntiForgeryToken“ pred metódy kódy ktoré upravujú stav, údaje.



```
[ValidateAntiForgeryToken]
0 references
public async Task<IActionResult> Edit(int id)
{
    var movieDetails = await _service.GetMovieByIdAsync(id);
    if (movieDetails == null) return View("NotFound");

    var response = new NewCourseVM()
    {
        Id = movieDetails.Id,
        Name = movieDetails.Name,
        Description = movieDetails.Description,
        Price = movieDetails.Price,
        ImageURL = movieDetails.ImageURL,
        CourseCategory = movieDetails.MovieCategory,
        CinemaId = movieDetails.ChanelId,
        LectorId = movieDetails.LectorId,
        DescriptionDetail = movieDetails.DescriptionDetail,
        CourseRequirement = movieDetails.CourseRequierment
    };

    var movieDropdownsData = await _service.GetNewMovieDropdownsValues();
    ViewBag.Cinemas = new SelectList(movieDropdownsData.Chanells, "Id", "Name");
    ViewBag.Producers = new SelectList(movieDropdownsData.Lectors, "Id", "FullName");
    return View(response);
}
```

Obrázok 29 : Príklad využitia „ValidateAntiForgeryToken“.

5.2 Zabezpečenie hesiel užívateľov

Ukladanie údajov do databáze môže byť tiež jednou z foriem bezpečností aplikácie. Ak ukladáme údaje o užívateľovi, napríklad ich hesla bez zabezpečenia riskujeme že ktokoľvek kto sa dostane do databáze získa prístup k ich účtu. ASP.NET ponúka funkciu `AspNetUsers` ktorá sa stará o užívateľov a ponúka ich správu a zabezpečenie. Stará sa o hashovanie hesiel užívateľov a následne ich ukladá do databáze tak aby ak niekto získa prístup k databáze tak získa zahashované heslo užívateľa.

ZÁVĚR

Hlavným cieľom tejto bakalárskej práce bolo navrhnúť a vytvoriť webovú aplikáciu pre ponuku online kurzov. K úspešnému splneniu cieľa bolo potrebné vytvoriť rešerš existujúcich riešení, zozbierať požiadavky pre návrh aplikácie, vytvoriť samotný návrh webovej aplikácie z ktorej následne vzišla implementácia webovej aplikácie.

Prvá kapitola tejto práce sa zaoberá technológiami ktoré boli použité pri návrhu a implementácii webovej aplikácie. Technológie pre návrh a implementáciu webovej aplikácie boli vybraté na základe výberu autora.

Druhá kapitola bakalárskej práce obsahuje rešerš už existujúcich riešení. Obsahuje základný popis online kurzov, opisuje základnú funkcionality aplikácií. Pre vytvorenie rešerše boli vybrané už existujúce aplikácie Seduo, Skilmea, Vita, Udemy a Itlearning. Každá zo spomenutých aplikácií obsahuje svoj popis kde sú spomenuté výhody aj nevýhody danej webovej aplikácie. Záverom je krátke zhodnotenie rešerše existujúcich riešení kde je porovnanie už existujúcich aplikácií. Druhá kapitola ukončuje zároveň teoretickú časť tejto práce.

Praktická časť tejto bakalárskej práce obsahuje tri kapitoly. Prvou z nich je návrh webovej aplikácie. Ten obsahuje slovný popis návrhu implementácie aplikácie, obsahuje funkčné požiadavky pre aplikáciu ktoré popisujú funkcionality aplikácie, obsahuje nefunkčné požiadavky aplikácie ktoré zahŕňajú, škálovateľnosť, zabezpečenie, výkon aplikácie. Návrh aplikácie obsahuje diagram prípadu užívania a ich popis pomocou scenárov. Súčasťou je aj E-R diagram a databázové schéma navrhovanej aplikácie.

Druhá kapitola praktickej časti popisuje tvorbu a samotnej webovej aplikácie. Je v nej popísaná funkcionality webovej aplikácie prepojenie z databázou. Popisuje základné funkcie webovej aplikácie a venuje sa popisu funkcií ktoré boli použité pri implementácii webovej aplikácie.

Tretia kapitola popisuje zabezpečenie implementovanej aplikácie. Obsahuje funkcie ktoré sú použité pre zabezpečenie webovej aplikácie.

Webová aplikácia zrealizovaná v tejto bakalárskej práci má jednoduchý vzhľad. Webová aplikácia je zameraná na funkcionality, jednou z hlavných funkcionalít aplikácie je vytvorenie nového online kurzu. Online kurzy sú jednou z hlavných funkcionalít sú

zobrazované na hlavnej stránke webovej aplikácie zobrazujú ich detail. Prihlásenie, registrácia, správa užívateľom sú ďalšie z funkcionalít aplikácie ktoré sú rovnako dôležité.

Pri vývoji webovej aplikácie som sa stretol s mnohými problémami, ktoré boli spôsobené základnou znalosťou frameworku Asp.Net. Samotný vývoj aplikácie v tomto frameworku ma veľmi bavil tento framework je intuitívny a rád by som v ňom vyvíjal aj v budúcnosti.

SEZNAM POUŽITÉ LITERATURY

- [1] Seduo [online]. Bratislava: LMC s.r.o, 2022 [cit. 2022-03-09]. Dostupné z: <https://www.seduo.sk/>
- [2] Skillmea [online]. Bratislava: Learn2Code o.z., 2022 [cit. 2022-03-09]. Dostupné z: <https://skillmea.sk/>
- [3] Itlearning [online]. Bratislava: IT LEARNING SLOVAKIA, 2021 [cit. 2022-03-09]. Dostupné z: <https://www.itlearning.sk/ponuka-kurzov/>
- [4] SQL For Dummies. 9th. Hoboken, New Jersey: John Wiley & Sons, Inc., 111, 2019. ISBN 978-1-119-52707-7.
- [5] GROFF, James R. a Paul N. WEINBERG. SQL: The Complete Reference. New York: Osborne/McGraw-Hill, 1999. ISBN 0072118458.
- [6] Introduction to Razor Pages in ASP.NET Core. Microsoft [online]. Microsoft, 2022 [cit. 2022-03-10]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/razor-pages/?view=aspnetcore-6.0&tabs=visual-studio>
- [7] CSS: The Missing Manual. Fourth Edition. Sebastopol: O'Reilly Media, Inc., 2015. ISBN 978-1-491-91805-0.
- [8] Entity Framework Core. Microsoft [online]. New York: Microsoft, 2022 [cit. 2022-03-11]. Dostupné z: <https://docs.microsoft.com/en-us/ef/core/>
- [9] ASP.NET Core security topics. Microsoft [online]. New York: Microsoft, 2022 [cit. 2022-03-13]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/security/?view=aspnetcore-6.0>
- [10] Cross-Origin Resource Sharing (CORS). Resources for Developers, by Developers [online]. Mountain View: mozilla.org, 2022 [cit. 2022-03-13]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>
- [11] Overview of ASP.NET Core MVC. Microsoft [online]. New York: Microsoft, 2022 [cit. 2022-03-13]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/core/mvc/overview?view=aspnetcore-6.0>
- [12] FREEMAN, Adam. Pro ASP.NET MVC 5. Fifth edition. New York: Springer Science+Business Media, 2015. ISBN 978-1-4302-6529-0.

- [13] HARRIS, Andy. *HTML5 and CSS3 All-in-One For Dummies*. 3rd Edition. New Jersey: John Wiley & Sons, Inc., 2015. ISBN 978-1-118-28938-9.
- [14] JAKOBUS, Benjamin a Jason MARAH. *Mastering Bootstrap 4*. Second Edition. Birmingham: Packt Publishing, 2018. ISBN 978-1-78883-490-2.
- [15] *Welcome to the Visual Studio IDE*. Microsoft [online]. New York: Microsoft, 2022 [cit. 2022-03-13]. Dostupné z: <https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2022>
- [16] ARLOW, Jim Arlow a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací Objektově orientovaná analýza a návrh prakticky*. 2. aktualizované a doplněné vydání. Brno: Computer Press, 2011. ISBN 978-80-251-1503-9.
- [17] GOMAA, Hassan Gomaa. *SOFTWARE MODELING AND DESIGN*. Madrid: Cambridge University Pres, 2011. ISBN 978-0-521-76414-8.
- [18] Udemy. Udemy [online]. San Francisco: Udemy, 2022 [cit. 2022-04-13]. Dostupné z: <https://www.udemy.com/>
- [19] Vita. Vita [online]. Bratislava: VITA Academy s.r.o, 2022 [cit. 2022-04-13]. Dostupné z: <https://www.vita.sk/>
- [20] *What-is-web-forms*. Microsoft [online]. 2022 [cit. 2022-03-15]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/web-forms/what-is-web-forms>
- [21] *Introducing-aspnet-web-pages*. Microsoft [online]. 2022 [cit. 2022-03-15]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/web-pages/overview/getting-started/introducing-aspnet-web-pages-2/getting-started>
- [22] *Uml-use-case-diagram* [online]. 2022 [cit. 2022-04-25]. Dostupné z: <https://www.lucidchart.com/pages/uml-use-case-diagram>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secured
SQL	Structured Query Language
UML	Unified Modeling Language

SEZNAM OBRÁZKŮ

Obrázok 1 : znázornenie modelu MVC.[11]	16
Obrázok 2 : Schéma diagramov UML. [16]	24
Obrázok 3 : Funkčné požiadavky	27
Obrázok 4 : Nefunkčné požiadavky.....	28
Obrázok 5 : Aktéri prípadu užitia	29
Obrázok 6 : Model prípadu užitia	29
Obrázok 7 : Databázová schéma.....	35
Obrázok 8 : Model triedy kurzu.....	36
Obrázok 9 : Navbar stránky.....	38
Obrázok 10 : Výber admina.....	38
Obrázok 11 : Výber správy stránky.....	38
Obrázok 12 : Zobrazenie kurzov stránky.....	39
Obrázok 13 : Pätička stránky.....	39
Obrázok 14 : Prihlasovací formulár.....	40
Obrázok 15 : Zobrazenie úspešnej registrácie.....	40
Obrázok 16 : Registračný formulár.....	41
Obrázok 17 : Karta kurzu.....	42
Obrázok 18 : Karta detailu kurzu.....	43
Obrázok 19 : Formulár tvorby kurzu.....	44
Obrázok 20 : Vyhľadávanie v aplikácii.....	45
Obrázok 21 : Kód metódy „Filter“.	45
Obrázok 22 : Správa užívateľov.....	46
Obrázok 23 : Kód pridania kurzu do košíku.....	46
Obrázok 24 : Kód zobrazenia kurzov v košíku.....	47
Obrázok 25 : Funkcia pre odobratie kurzu.....	47
Obrázok 26 : Nákupný košík.....	47
Obrázok 27 : Platba pomocou PayPal.....	48
Obrázok 28 : Funkcia pre platbu cez PayPol.....	49
Obrázok 29 : Príklad využitia „ValidateAntiForgeryToken“.	50
Obrázok 30 : View model course.....	60
Obrázok 31 : Diagram aktivít tvorby kurzu.....	61

SEZNAM TABULEK

Tabuľka 1. Typy vzťahov UML	23
Tabuľka 2. Scenár vytvorenia kurzu.....	30
Tabuľka 3. Scenár zadania názvu kurzu.	30
Tabuľka 4. Scenár vyplnenia detailu kurzu.	31
Tabuľka 5. Scenár vyplnenia popisu kurzu.	31
Tabuľka 6. Scenár vyplnenia doplňujúcich údajov.	32
Tabuľka 7. Vyplnenie údajov	32
Tabuľka 8. Scenár registrácie.	33
Tabuľka 9. Scenár nákupu kurzu.	33
Tabuľka 10. Scenár prihlásenia.	34
Tabuľka 11. Scenár správy systému.	34

SEZNAM PŘÍLOH

Príloha PI: View model course

Príloha PII: Diagram aktivít

Príloha PIII: Obsah CD s kódom

PRÍLOHA P I: VIEW MODEL COURSE

```
3 references
public int Id { get; set; }

[Display(Name = "Názov kurzu")]
[Required(ErrorMessage = "Povinné pole")]
9 references
public string Name { get; set; }

[Display(Name = "Popis online kurzu")]
[Required(ErrorMessage = "Povinné pole")]
9 references
public string Description { get; set; }

[Display(Name = "Popis obsahu kurzu")]
[Required(ErrorMessage = "Povinné pole")]
8 references
public string DescriptionDetail { get; set; }

[Display(Name = "Požiadavky pre zvládnutie kurzu")]
[Required(ErrorMessage = "Povinné pole")]
5 references
public CourseRequierment CourseRequirement { get; set; }

[Display(Name = "Cena")]
[Required(ErrorMessage = "Povinné pole")]
9 references
public double Price { get; set; }

[Display(Name = "Náhľad kurzu")]
[Required(ErrorMessage = "Povinné pole")]
9 references
public string ImageURL { get; set; }

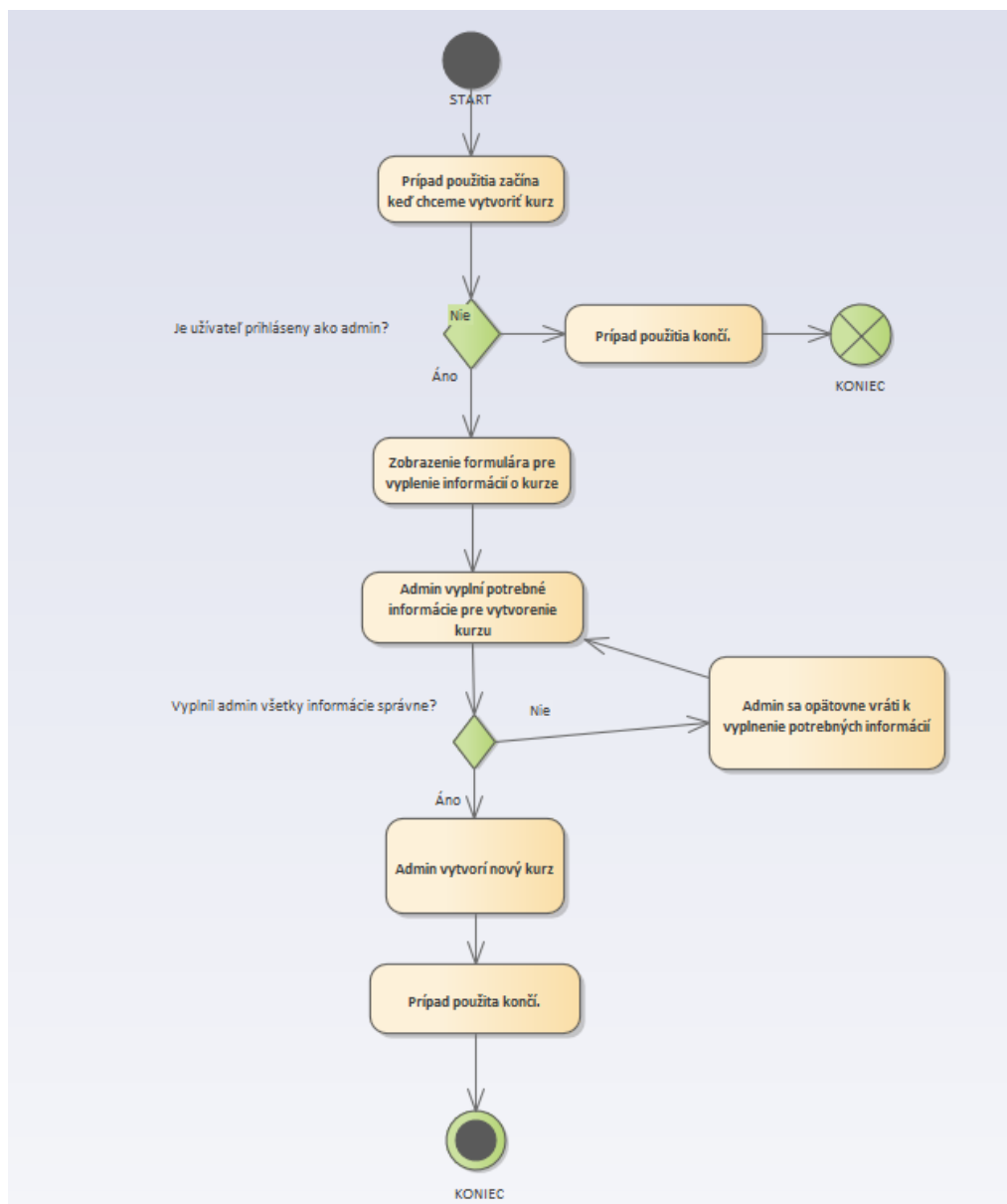
[Display(Name = "Kategória")]
[Required(ErrorMessage = "Povinné pole")]
9 references
public CourseCategory CourseCategory { get; set; }

[Display(Name = "Kanál kurzu")]
[Required(ErrorMessage = "Povinné pole")]
9 references
public int CinemaId { get; set; }

[Display(Name = "Lektor kurzu")]
[Required(ErrorMessage = "Povinné pole")]
9 references
public int LectorId { get; set; }
```

Obrázok 30 : View model course.

PRÍLOHA P II: DIAGRAM AKTIVÍT



Obrázok 31 : Diagram aktivít tvorby kurzu.

PRÍLOHA PIII: OBSAH CD

- Fulltext.pdf
- Webapp.zip