

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Webová aplikace pro tvorbu online testů



2022

Vedoucí práce: RNDr. Bartl Eduard, Ph.D.

Veronika Gábová

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor: Veronika Gábová
Název práce: Webová aplikace pro tvorbu online testů
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2022
Studijní obor: Aplikovaná informatika, prezenční forma
Vedoucí práce: RNDr. Bartl Eduard, Ph.D.
Počet stran: 41
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Veronika Gábová
Title: Web application for online tests creation
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2022
Study field: Applied Computer Science, full-time form
Supervisor: RNDr. Bartl Eduard, Ph.D.
Page count: 41
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Cílem bakalářské práce je vytvoření webové aplikace pro tvorbu online testů. Umožňuje registrovat jednotlivé organizace včetně lektorů, kteří vytvářejí testy a distribují je buď jednotlivým studentům, nebo skupinám studentů. Studenti přistupují ke svým zadaným testům přes přihlášení do aplikace. Po absolvování testu mají přístup i ke všem výsledkům z absolvovaných testů.

Synopsis

The part of the bachelor thesis is creating a web application for creating online tests. It allows to register individual organisations including lecturers who create tests and distribute them either to individual students or to groups of students. Students access their assigned tests through logging into the application. They also have access to all results from the tests once they have completed them.

Klíčová slova: webová aplikace; testy; testování znalostí

Keywords: web application; tests; knowledge testing

Děkuji panu RNDr. Bartl Eduard, Ph.D. za 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
2	Obecné pojmy	9
2.1	Test	9
2.2	Webová aplikace	9
3	Podobné aplikace	10
3.1	EDUMO	10
3.2	BookWidgets	11
3.3	Testedit	13
3.4	iTester	14
4	Použité Technologie	16
4.1	C#	16
4.2	.NET	16
4.3	ASP.NET Core	17
4.3.1	Castle Windsor	18
4.4	HTML	19
4.5	CSS	19
4.6	JavaScript	20
4.7	AJAX	20
4.8	Boostrap	20
4.9	Microsoft SQL Server	21
5	Autentizace	22
5.1	Autentizace HTTP	22
5.2	Ověřování třetí stranou	22
5.2.1	Standard OpenID	22
5.2.2	Standard OAuth	23
6	Autorizace	24
7	Programátorská dokumentace	25
7.1	DataAccessLayer	25
7.2	ModelLayer	25
7.3	ServiceLayer	25
7.4	WebApplicationForOnlineTests	26
7.4.1	Properties	26
7.4.2	wwwRoot	26
7.4.3	Controllers	26
7.4.4	Infrastructure	26
7.4.5	Views	27
7.4.6	appsettings.json	27
7.4.7	Program.cs	27

7.4.8	Startup.cs	27
7.5	Struktura databáze	28
7.6	Autentizace a autorizace	29
8	Uživatelská dokumentace	30
8.1	Uživatelská role administrátor	30
8.1.1	Položka kurzy	30
8.1.2	Položka lektori	30
8.1.3	Položka uživatelé	31
8.2	Uživatelská role lektor	31
8.2.1	Položka kurzy	31
8.2.2	Položka vytvořené testy	32
8.2.3	Položka uživatelé	33
8.2.4	Položka statistiky	33
8.3	Uživatelská role uživatel	34
8.3.1	Položka kurzy	34
8.3.2	Položka testy	34
8.3.3	Položka statistiky	35
9	Testování aplikace	36
	Závěr	37
	Conclusions	38
A	Obsah příloženého CD/DVD	39
	Literatura	40

Seznam obrázků

1	Náhled aplikace EDUMO: vytváření testu	10
2	Náhled aplikace EDUMO: spuštěný test	11
3	Náhled aplikace BookWidgets: výběr typu otázky I.	11
4	Náhled aplikace BookWidgets: výběr typu otázky II.	12
5	Náhled aplikace BookWidgets: vytváření testu	12
6	Náhled aplikace Testedit: vytváření testu	13
7	Náhled aplikace Testedit: spuštěný test	13
8	Náhled aplikace iTester: vytváření testu	14
9	Náhled aplikace iTester: spuštěný test	15
10	Náhled aplikace iTester: sekce reporting	15
11	Náhled na seznam kurzů	32
12	Náhled vytváření testu	33
13	Náhled na seznam uživatelů	34
14	Náhled vypracování testu	35
15	Náhled na uživatelskou statistiku	35

Seznam zdrojových kódů

1 Úvod

Cílem mé bakalářské práce je vytvořit webovou aplikaci pro tvorbu online testů. Aplikace bude umožňovat registrovat jednotlivé organizace včetně svých správců, kteří budou jednotlivé organizace spravovat. Ti budou přidávat primárně lektory, ale budou moci přidávat i případné studenty a kurzy. Každý lektor bude vytvářet testy, kurzy a jednotlivé studenty. Kurz představuje skupinu studentů, kde jsou přiřazeny testy podobných témat. Kurzy, studentům ulehčí přehled tím, že budou seskupovat testy do jednotlivých kategorií například kurz Biologie. Podobné zřehlednění bude i pro lektora, kdy do určitého kurzu přidají nové testy, které se následně přiřadí k jednotlivým studentům.

Student bude mít o přiděleném testu dostupné informace jako například název a popis testu, zpřístupnění a ukončení testu, časový limit, bodování nebo pokusy. Pokud student z nějakého důvodu přeruší test, bude se brát jako absolvovaný. Během testu má student k dispozici znázornění časového limitu a možnost přeskokovat otázky. Nakonec bude studentovi dostupná statistika o jeho celkových výsledcích.

Samotné testy budou mít k dispozici různé typy otázek (jedna správná odpověď, více správných odpovědí, doplnění vynechaného slova nebo vlastní studentova odpověď) a množství nastavení, které bude moci lektor specifikovat (časový limit, přiřazení bodů k jednotlivým otázkám, odečet bodů za špatné odpovědi apod.).

2 Obecné pojmy

2.1 Test

Test [14], obecně řečeno, znamená nástroj, techniku nebo metodu, která je určena k měření znalostí studenta nebo jejich schopnosti splnit určitý úkol. Lze tedy testování považovat za formu hodnocení. Existují různé typy testů:

- rozřazovací testy (rozřazení studentů do určité úrovně nebo sekce),
- diagnostické testy (pomáhají učitelům a žákům určit silné a slabé stránky),
- znalostní testy (měří úroveň např. jazykových znalostí),
- prospěchové testy (určeny k měření dovedností a znalostí, které si žák osvojil po určitém druhu výuky).

Jednotlivé testy mohou mít více typů otázek [9], typické typy jsou například

- výběr z více možností (jedna správná odpověď),
- více správných odpovědí,
- pravda nebo lež,
- vlastní odpověď,
- přiřazení výrazu k jinému výrazu.

Ovšem existuje mnoho dalších typů otázek, se kterými se můžeme v testu setkat.

2.2 Webová aplikace

Webová aplikace [28, 16] neboli web app je aplikační software, který běží na webovém serveru. Na rozdíl od počítačových programů, které jsou spuštěny lokálně v operačním systému zařízení. K webovým aplikacím přistupuje uživatel prostřednictvím webového prohlížeče s aktivním síťovým připojením. Tyto aplikace jsou naprogramovány na bázi klient-server.

Webová aplikace oproti jiným typům aplikací se staví do kontrastu s nativními aplikacemi, což jsou aplikace vyvinuté speciálně pro určitou platformu nebo zařízení a nainstalované v daném zařízení. Nativní aplikace mohou také využívat hardware specifický pro dané zařízení například GPS nebo fotoaparát v mobilní nativní aplikaci. Dále existují programy, které kombinují oba přístupy ozn. Hybridní aplikace. Takové aplikace fungují podobně jako webové aplikace, ale do zařízení se instalují stejně jako nativní aplikace a na rozdíl od nativních aplikací nemohou tyto aplikaci fungovat offline.

3 Podobné aplikace

V této kapitole si uvedeme aplikace, které jsou na obdobné bázi jako vytvářené aplikace bakalářské práce. Uvedené aplikace v následujících podkapitolách jsou převážně samostatné webové aplikace určené pro vytváření testů nikoli aplikace obsahující pouze určitý modul pro vytváření testů.

3.1 EDUMO

Aplikace EDUMO [6] je určená pro testování uživatelů. Tato aplikace vytváří libovolné množství online testů. Nabízejí více variant testů a jejich následné vyhodnocení je okamžité. Obsahuje i statistiky testovaných uživatelů, a tím poukazuje na jejich slabiny, či nedostatky v testech. Následující obrázek prezentuje stránku, kde si uživatel může vytvořit test. Z prvního dojmu, není zcela jasné, co přesně znamená položka Aktivní. Mezi další nevýhodu patří počet otázek, které se musí předem nastavit na určitou hodnotu. V případě, pokud lektor zjistí v průběhu vytváření otázek, že chce o jednu otázku navíc, musí vytvořit test nový. Dále bych zmínila, že daný test obsahuje jednoduché hodnocení, a to buď úspěš, nebo neúspěš.

Název	Počet otázek	Čas testu [min]	Min.pož.úspěšnost %	Upravil
Nový test				Suchel Mirek

Název:

Aktivní:

Počet otázek:

Čas testu:

Min.pož.úspěšnost:

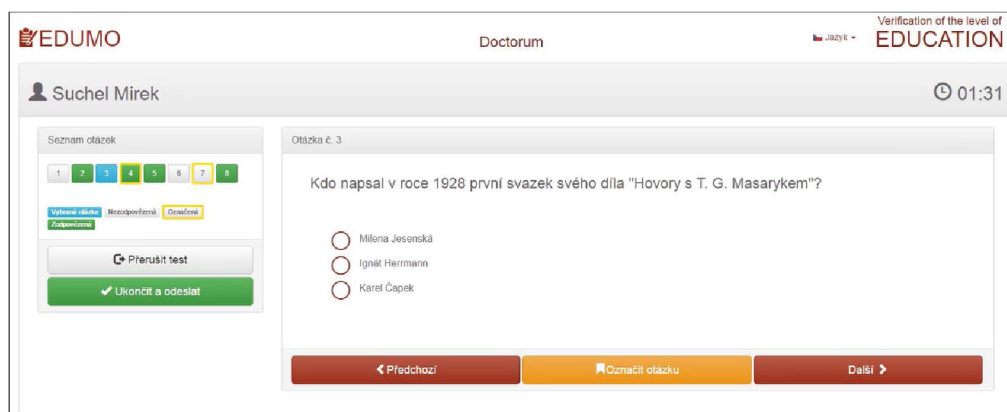
Text pro studenta:

Interní poznámka:

Obrázek 1: Náhled aplikace EDUMO: vytváření testu

Samotný spuštěný test vypadá přehledný, jen s malou výtkou, jedná se o „Označit otázku“, kdy není zcela jasné, k čemu daná funkce slouží. Zde bych zmínila jen úvahovou otázku nad tím, zda je výhodou či nikoliv přerušování testu. V případě, že lze otázky přeskakovat a zcela všechny otázky si může uživatel zobrazit, nepřijde mi vhodné test přerušit. Uživatel si tímto může při opětovném spuštění přilepšit v hodnocení a test pak není objektivní.

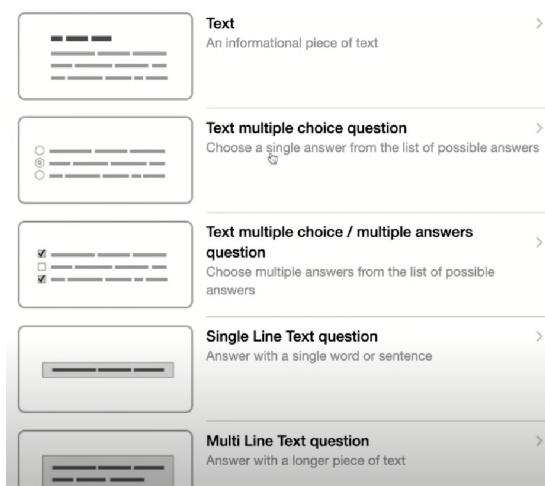
Aplikace je placená na základě množství uživatelů a délky předplatného. V ceně je zahrnuta neomezená aktualizace po celou dobu užívání. Co se týká technické podpory, tak jsou zde zahrnuty opravy chyb, které mohou v aplikaci nastat. Uživatelé si mohou aplikaci ještě před zakoupením vyzkoušet zdarma.



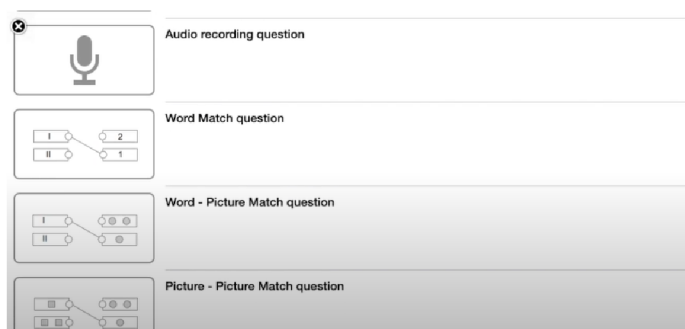
Obrázek 2: Náhled aplikace EDUMO: spuštěný test

3.2 BookWidgets

Celosvětově známá webová aplikace zvaná BookWidgets [3] nabízí možnost vytvořit více než 40 typů testů. Tato aplikace je velmi konkurenceschopná a nabízí uživatelům opravdu mnoho funkcí. Od základních typů otázek, až po rozmanitější, které zde můžete vidět v obrázcích níže.

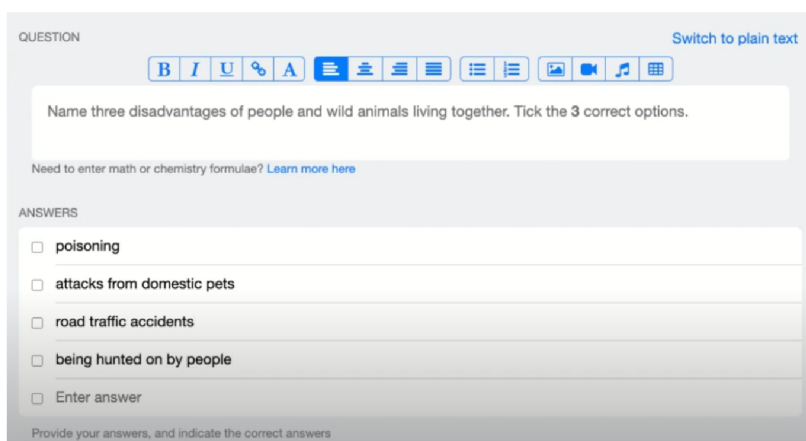


Obrázek 3: Náhled aplikace BookWidgets: výběr typu otázky I.



Obrázek 4: Náhled aplikace BookWidgets: výběr typu otázky II.

Aplikace nabízí i v samotném vytváření testu editor textu, který vylepšuje samotnou webovou aplikaci o úroveň výše a nabízí uživatelům, o to příjemnější stylování otázky.

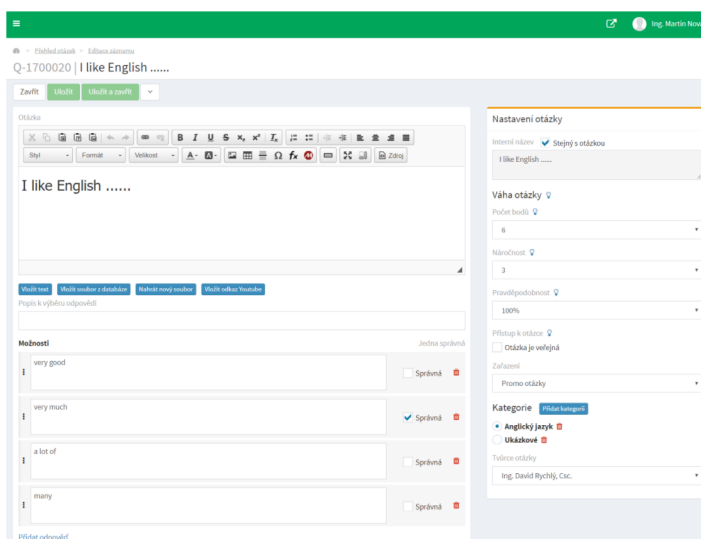


Obrázek 5: Náhled aplikace BookWidgets: vytváření testu

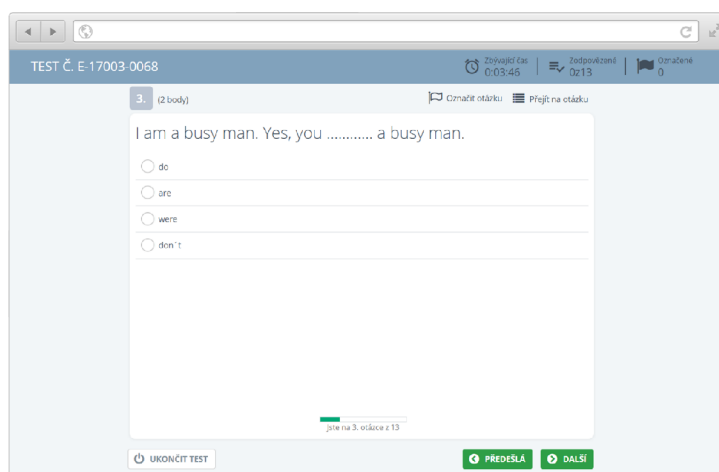
Zajímavostí této aplikace je, že nenabízí jen vytváření testů, ale i vytváření výukových her (Hangman, Crosswords) nebo obsahuje i matematickou část (Arithmetic, ActivePlot). Aplikace je responsivní, takže si uživatel může vyplnit svůj test jednoduše i na mobilním telefonu nebo tabletu. Ovšem za takovou vyspělou aplikaci si člověk připlatí. Aplikace je nabízena ve čtyřech variantách, a to varianta učitel, skupina učitelů, standard a profesionál. Aplikace nabízí též možnost trial verze, tedy než dojde k samotné koupi může uživatel zjistit, zda si tuto aplikaci chce pořídit. Daná aplikace nabízí návod jak s ní pracovat, a to ve formě videí.

3.3 Testedit

Webová stránka testedit.cz [17] nabízí webovou aplikaci pro tvorbu, distribuci a vyhodnocení testu. Test je možný vypracovat i offline (tisk zadání a záznamového archu). Aplikace uživateli umožňuje přihlašování k testům, vypracování a zobrazení výsledků. Administrativní část umožňuje tvorbu různých typů otázek, schvalování otázek a následnou distribuci testů a přehledy vypracovaných testů. Nabízí také strukturované členění otázek (kategorie, náročnost nebo tvůrce testu) a variabilitu vypracování testu (časové omezení, počet pokusů).



Obrázek 6: Náhled aplikace Testedit: vytváření testu



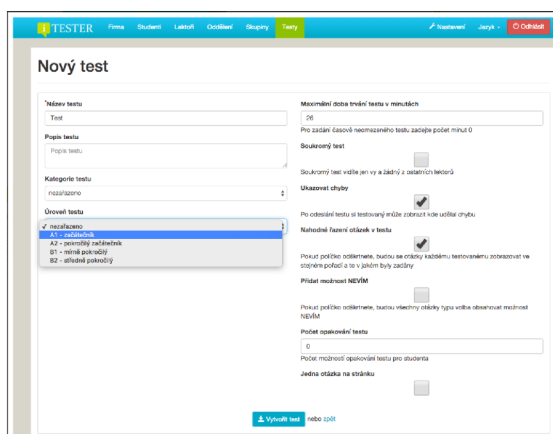
Obrázek 7: Náhled aplikace Testedit: spuštěný test

Další funkce, které uživatelé ocení jsou export testů do XLSX, poznámky k testům nebo e-mailová notifikace při odeslání testu. Aplikace nabízí český a slovenský jazyk, a aplikaci také nechybí responzivní design. Ovšem cena aplikace se člení dle balíčku, který si můžete vybrat. Nejlevnější balíček na činí 1390 Kč na měsíc, kdy obsahuje:

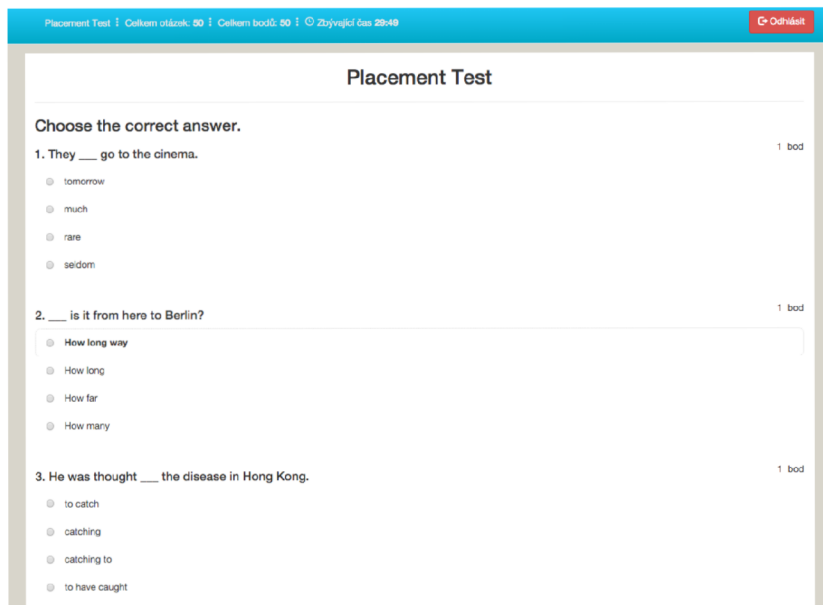
- komplexní zabezpečení provozu systému na jejich cloudové platformě,
- pravidelné zálohování,
- zabezpečení systému bezpečnostním certifikátem,
- údržbu systému a jeho aktualizace,
- garantovanou technickou a uživatelskou podporu.

3.4 iTester

Profesionální webová aplikace iTester [7] umožňuje vytvářet testy na míru pro použití ve firmách, školách a vzdělávacích nebo personálních agenturách. Umožňuje rychlé vytváření testů, kdy otázky jsou různých typů jako výběr z možností, doplnění nebo přiřazení. Testové otázky mohou mít také nastavenou různou váhu pro následné vyhodnocení výsledků. Přiřazení testů je možné definovaným skupinám uživatelů, jednotlivé testované osoby můžete dále seskupovat do skupin. Administrace všech uživatelů a jejich organizace do čtyř dostupných skupin rozdělených na základě přístupových práv. Součástí aplikace je také komplexní správa uživatelů a reporting, který obsahuje veškeré statistiky, grafy, přehledy výsledků testování. Data získaná z vypracovaných testů je možné filtrovat a třídit podle zadaných parametrů přímo v aplikaci.

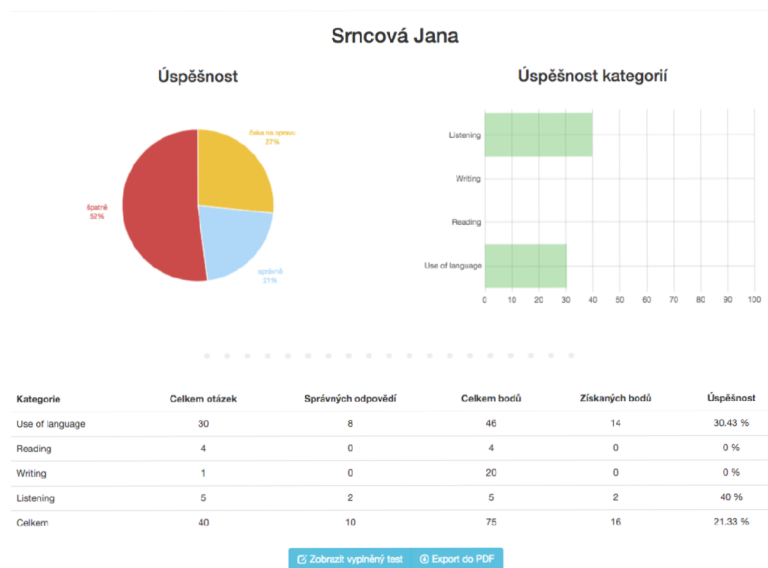


Obrázek 8: Náhled aplikace iTester: vytváření testu



Obrázek 9: Náhled aplikace iTester: spuštěný test

Hodnocení testu: A2



Obrázek 10: Náhled aplikace iTester: sekce reporting

4 Použité Technologie

4.1 C#

C# [23] je objektově orientovaný programovací jazyk vyvinutý firmou Microsoft spolu s vývojovou platformou .NET. Tento programovací jazyk vychází z programovacího jazyka C/C++, ale v mnoha ohledech je bližší programovacímu jazyku Java. Od začátku byl navržen jako multiplatformní programovací jazyk. Tento programovací jazyk lze navíc využít k tvorbě databázových programů, webových aplikací, webových služeb, formulářových aplikací ve Windows a softwaru pro mobilní zařízení.

Překladače programovacího jazyka C# jsou case sensitive. V jazyce C# bylo zavedeno několik konvencí jako například to, že jména balíčků, tříd, rozhraní a většiny dalších položek začínají velkým písmenem. Naopak malým písmenem začínají modifikátory, proměnné a parametry. Nyní si uvedeme základní charakteristiku daného jazyka [12, 2]:

- programovací jazyk C# je čistě objektově orientovaný,
- obsahuje pouze jednoduchou dědičnost s možností násobné implementace rozhraní,
- správa paměti je automatická, tudíž o korektní uvolňování zdrojů aplikace stará garbage collector,
- podporuje zpracování chyb pomocí výjimek,
- vedle členských dat a metod přidává vlastnosti a události,
- zajišťuje typovou bezpečnost,
- zajišťuje zpětnou kompatibilitu se stávajícím kódem jak na binární, tak na zdrojové úrovni.

4.2 .NET

.NET [27], dříve pod názvem .NET Core, je bezplatná, open-source platforma spravovaná počítačovým softwarovým frameworkem pro operační systémy Windows, Linux a macOS. Jedná se o multiplatformního nástupce .NET frameworku a je vyvíjen především zaměstnanci ze společnosti Microsoft prostřednictvím nadace .NET Foundation a následně vydáván pod licencí MIT.

Současná verze je .NET 5 [11], která je další verzí .NET Core následující po verzi 3.1. Tato nová verze je pojmenovaná .NET 5 místo .NET Core 4.0 ze dvou důvodů. Prvním důvodem je přeskočením čísla verzí 4.x, abych nedocházelo k nejasnostem .NET Frameworku 4.x a druhý důvod je takový, že bylo vypuštěno „Core“, aby se zdůraznilo, že se jedná o hlavní implementaci .NET do budoucna. .NET 5.0 podporuje více typů aplikací a více platform než .NET Core nebo .NET Framework.

4.3 ASP.NET Core

ASP.NET Core [21] je nová verze webového frameworku ASP.NET od společnosti Microsoft, která je určena především pro platformu .NET Core. Jedná se o bezplatný/svobodný, open-source a multiplatformní Framework pro vytváření cloudových aplikací, jako například webová aplikace. Aplikace jsou navrženy tak, aby mohly běžet v cloudu i na lokálních počítačích.

Technologie je založená na architektuře klient-server, kdy aplikace je program a jehož výstupem je webová stránka. Microsoft nabízí několik způsobů jak webové stránky vytvářet. Nejstarším způsobem byly WebForms, kdy jejich nástupcem je ASP.NET MVC. Nicméně již uvedené ASP.NET MVC je v současné době nahrazováno technologií ASP.NET Core, který navíc obsahuje architektonický vzor MVC [18], který rozděluje webovou aplikaci do tří komponent různých typů, a to Modely (Model), Pohledy (View) a Kontrolery (Control).

- **Model** – zodpovídá za správu dat a obecně charakterizuje třídy, například obsahuje výpočty, databázové dotazy nebo validace.
- **Pohled** – stará se o prezentaci modelu v určitém formátu.
- **Kontroler** – přijímá vstupní údaje, volitelně je ověřuje a poté je předává modelu. Jedná se o tzv. prostředníka, se kterým komunikuje uživatel, model i pohled. Obecně lze říci, že všechny komponenty propojuje.

ASP.NET Core se od ASP.NET MVC liší pouze tím, že na server kromě naší aplikace nahráváme také i samotný Framework ASP.NET, který už nemusí být na serveru nainstalován. Z toho plyne, že aplikace může být nahrána i tam, kde není operační systém Windows nebo ASP.NET. Nyní se dozvíme, proč se vyplatí využívat zrovna ASP.NET Core.

- **Multiplatformnost** – aplikace lze spustit v různých systémech a nemusíme vytvářet jednotlivé aplikace pro různé platformy pomocí různých frameworků.
- **Rychlost** – na této platformě lze pracovat rychle, z důvodu nezávislosti na Systém.Web.dll pro komunikaci mezi prohlížečem a serverem. ASP.NET nám umožňuje zahrnout balíčky, které potřebujeme pro naši aplikaci, a tím se snižuje množství požadavků a zlepšuje výkon a škálovatelnost.
- **Kontejner IoC** – slouží pro automatické vkládání závislostí (Dependency Injection), což umožňuje údržbu a testování.
- **Integrace s moderními frameworky uživatelského rozhraní** – umožňuje používat a spravovat moderní frameworky uživatelského rozhraní, jako jsou AngularJS, ReactJS, Ember, Bootstrap atd. pomocí Bower (správce balíčků pro web).

- **Hostování** – webové aplikace ASP.NET Core umožňuje hostovat na různých platformách s libovolným webovým serverem, jako je IIS, Apache atd. Není závislá pouze na IIS jako standardní .NET Framework.
- **Sdílení kódu** – umožňuje vytvořit knihovnu tříd, kterou lze použít s jinými frameworky .NET, jako je .NET Framework 4.x.
- **Verzování aplikací vedle sebe** – ASP.NET Core běží na jádře .NET Core, které podporuje souběžné spouštění více verzí aplikací.

ASP.NET Core podporuje návrhový vzor DI [24] tzv. Dependency Injection (přeloženo do češtiny jako vkládání závislostí), který implementuje princip IoC. Jedná se o techniku pro vkládání závislostí mezi jednotlivými komponentami programu tak, aby mohla komponenta používat jinou komponentu bez jakékoli reference v době sestavování, díky čemuž jsou lépe testovatelné, udržovatelné a rozšiřitelné. Právě takové automatické vkládání závislostí je implementováno frameworkem IoC kontejnerem, který řídí vytváření objektů a jejich životnost a navíc vkládá závislosti do jednotlivých tříd. Obecně lze říci, že kontejner IoC [22] vytvoří objekt zadané třídy a prostřednictvím konstruktoru, vlastnosti nebo metody za běhu vloží všechny závislé objekty, a ve vhodné chvíli je navíc uvolní. Provádí se to proto, abychom nemuseli objekty vytvářet a spravovat ručně.

4.3.1 Castle Windsor

V této práci jsem použila Castle Windsor [10] kontejner. Je druhým nejstarším DI kontejnerem pro .NET a také jedním z nejnávštěvnějších kontejnerů.

Castle projekt [5] se vyvinul z projektu Apache Avalon v polovině roku 2003 jako pokus o vytvoření velmi jednoduchého kontejneru pro inverzi řízení. Obecně měl projekt snahu zjednodušit vývoj podnikových nebo webových aplikací. Později však nastala otázka, co by měl užitečný kontejner řešit, a na to právě vznikl první návrh Castle Windsor. Do současné podoby došlo k několika změnám.

4.4 HTML

HTML [4] neboli Hypertext Markup Language je značkovací jazyk, který popisuje obsah a strukturu dokumentu pomocí značení nebo tagování různých prvků dokumentu. Navíc díky termínu hypertext je patrné, že podporuje propojení dokumentů pomocí hypertextových odkazů.

V první letech využívání této technologie neexistovala žádná organizace, která by definovala pravidla nebo syntaxi jazyka HTML. Nicméně problémem bylo, že jednotlivé webové prohlížeče podporovaly různé „příchutě“ jazyka HTML, a tak se webová stránka vykreslovala v každém prohlížeči jinak. Nakonec skupina webových návrhářů a programátorů nazvaná World Wide Web Consortium (W3C) dohodla na určitých standardech a specifikacích, které musí dodržovat všichni výrobci prohlížečů. Stále však existovalo mnoho nekompatibilit v tom, jak byl jazyk HTML implementován v různých prohlížečích. Kvůli tomu se W3C snažilo převzít kontrolu nad tím vším a prosadit přísnější sadu standardů v jiné verzi webu zvaném XHTML (Extensible Hypertext Markup Language). Jenže časem se zjistilo, že má jeden velký problém a to, že nebylo zpětně kompatibilní s HTML a v důsledku toho nebylo možné snadno převést starší webové stránky do nového standardu. Díky tomuto se vytvořila v roce 2004 pracovní skupina WHATWG (Web Hypertext Application Technology Working Group), jejímž úkolem bylo vyvinout konkurenční verzi k XHTML zvanou HTML5, která byla kompatibilní s dřívějšími verzemi jazyka HTML. V současné době existuje mnoho starších prohlížečů a aktivně používaných zařízení, které avšak HTML5 plně nepodporují.

4.5 CSS

CSS [4] neboli Cascading Style Sheets (přeloženo do češtiny jako kaskádové styly) používající se k popisu prezentace dokumentu napsaného ve značkovacím jazyce, jako například HTML, XML nebo XHTML. Je navržen tak, aby umožňoval oddělení prezentace a obsahu, to napomáhá ke zlepšení přístupnosti obsahu, poskytnutí větší flexibility. Vzhledem k tomu umožňuje více webovým stránkách sdílet formátování tím, že příslušné CSS specifikují v samostatném souboru .css, což snižuje složitost a opakování strukturálního obsahu. Rovněž umožňuje ukládání souboru .css do mezipaměti, aby se zvýšila rychlost načítání stránek, které sdílejí tento soubor a jeho formátování.

Jak jsme výše zmínili s příchodem HTML5 je spjat i příchod CSS3, který slouží pro stylování internetových dokumentů. Navíc CSS spravuje World Wide Web Consortium (W3C).

4.6 JavaScript

JavaScript [25, 13], ozn. JS, je multiplatformní, multiparadigmatický skriptovací jazyk, který odpovídá specifikaci ECMAScript. Je to univerzální programovací jazyk standardizovaný mezinárodní organizací Ecma, který se běžně používá pro skriptování na straně klienta na World Wide Webu a stále častěji se používá pro psaní serverových aplikací a služeb pomocí Node.js. Programy napsané v tomto jazyce se nazývají skripty a zapisují se přímo do HTML kódu.

V současné době je JavaScript součástí většiny webů a jeho popularita stoupá díky javascriptovým frameworkům, jako například Angular nebo React. Javascript si ovšem rozšířil působnost i na mobilní aplikace pomocí technologie React Native, umožňující současnou tvorbu aplikací pro více platform (Android a iOS).

4.7 AJAX

AJAX [20] je zkratkou pro asynchronní JavaScript a XML. Běžné webové aplikace totiž přenášejí informace na server a z něj pomocí synchronních požadavků. To znamená, že po stisku tlačítka odeslat se dostaneme na novou stránku s novými informacemi ze serveru. Jenže při použití AJAXu po stisknutí tlačítka odeslat JavaScript provede požadavek na server, interpretuje výsledky a aktualizuje pořadovanou část stránky bez nutnosti přenačtení celého obsahu. Je to technika, která vytváří rychlejší a interaktivnější webové aplikace pomocí XML, HTML, CSS a JavaScriptu. Jazyk HTML se používá pro obsah, CSS pro prezentaci spolu s Document Object Model, JavaScript pro dynamické zobrazování obsahu a XML se běžně používá jako formát pro příjem dat ze serveru.

4.8 Bootstrap

Bootstrap [19, 8] je front-end framework HTML, CSS a JavaScriptu pro tvorbu responzivních a mobilních webových stránek. Původně pojmenovaný Twitter Blueprint, která byl vyvinut Markem Ottem a Jacobem Thorntonem ve společnosti Twitter jako Framework. Časem se do projektu zapojovalo mnoho vývojářů a následně byl přejmenován na Bootstrap v roce 2011. Používá se pro snadnější a rychlejší vývoj webových stránek a navíc je zcela zdarma ke stažení a používání. Dále si uvedeme hlavní výhody, proč by se uživatel měl rozhodnout pro tento framework:

- velmi jednoduchý na používání, pokud má uživatel základní znalosti HTML a CSS,
- ušetření času pomocí předdefinovaných šablon a tříd Bootstrap,
- usnadňuje uživatelům tvorbu responzivních webových stránek,
- kompatibilní s většinou prohlížečů.

4.9 Microsoft SQL Server

Microsoft SQL Server [26] je relační systém pro správu databází vyvinutý společností Microsoft. Jakožto databázový server má za funkci ukládání a načítání dat podle požadavků jiných softwarových aplikací, které mohou běžet buď na stejném počítači, nebo na jiném počítači v síti. Společnost Microsoft nabízí ovšem SQL Server v několika edicích s různými sadami funkcí a pro různé uživatele, jako například Express, Standard nebo Azure.

Microsoft v roce 1986 ještě neměl svůj vlastní databázový systém, avšak to se časem změnilo roku 1992, kdy představil na databázovém trhu produkty Microsoft Access a Microsoft FoxPro. Po velkém úspěchu se později Microsoft obrátil k softwaru Sybase, jehož kód produktu Sybase SQL Server se stal základnou pro vytvoření Microsoft SQL Serveru, a tím se stal první databází od Microsoftu. Tento produkt konkuroval systémům Sybase, Oracle i společnosti IBM. Nicméně první databáze, kterou Microsoft vyvinula bez cizí pomoci, byla v roce 1995 databáze s názvem Microsoft SQL Server 6.0. Postupem času byla představena první verze Microsoft SQL Serveru 2000, který byl postaven na architektuře klient/server. Navíc má integrovanou podporu jazyka XML a bezpečný systém. Následné verze se většinou během dvou let vylepšovaly až k současné verzi Microsoft SQL Server 2019.

5 Autentizace

Autentizace [15](neboli ověřování) je běžným požadavkem webových aplikací. Takové ověření je potřeba právě k ochraně důvěrných nebo soukromých informací. Máme různé typy autentizace, a to například autentizace http nebo ověřování třetí stranou.

5.1 Autentizace HTTP

Autentizace HTTP poskytuje dva mechanismy ověřování. První mechanismus ověřování je nejjednodušší, ale za to nejméně bezpečný. K ověření uživatele se používá kombinace uživatelského jména a hesla, které jsou zakódovány algoritmem Base64 a odeslány zpět v hlavičce Authorization. Takové heslo lze snadno dekodovat, proto by se mělo při základní ověřování používat pouze přes zabezpečené připojení (HTTPS). V praxi se často používá k zajištění nízké úrovně ochrany zdrojů.

Druhý mechanismus je iniciována stejným způsobem, ale namísto odeslání uživatelského jména a hesla klientský prohlížeč vygeneruje jednosměrný hash (např. pomocí md5 algoritmu), a to na základě uživatelského jména, hesla, unikátního řetězce a odesílaného požadavku. Tento hash je odeslán jako součást dalšího požadavku spolu s uživatelským jménem. Prohlížeč má stejné informace jako klient a může si vypočítat stejný jednosměrný hash a porovnat obdrženou hodnotu.

5.2 Ověřování třetí stranou

Nedávným vývojem v oblasti webových aplikací bylo poskytování ověřovacích služeb třetími stranami. Jde o to, aby nová webová aplikace nespravovala vlastní uživatelská pověření, ale aby se obrátila na jinou webovou aplikaci, která zkontroluje uživatelské jméno a heslo. Následně na to informuje klientskou aplikaci o uživateli. Přináší to mnoho výhod jak pro vývojáře, tak pro uživatele. Například vývojář se nemusí starat o správu uživatelských údajů, zejména o bezpečnost hesel, protože si je neukládá. Existují dva hlavní standardy, které si uvedeme níže.

5.2.1 Standard OpenID

Standard OpenID umožňuje použití celé řady poskytovatelů identit. Identitou uživatele je webová adresa, kterou může webová aplikace použít k ověření identity uživatele. Transakce se účastní tři strany – uživatel, poskytovatel OpenID a spoléhající strana (webová aplikace, která se bude spoléhat na výsledek autentizace).

5.2.2 Standard OAuth

Standard OAuth, používáme aby webová aplikace umožňovala přístup k datům nebo službám poskytovaný jinou webovou aplikací jménem daného uživatele. Tento standard je autorizační a autentizační mechanismus, který umožňuje přístup uživatele naší aplikace k poskytovateli. Při úspěšné odpovědi obdržíme token, s jehož pomocí můžeme jménem uživatele přistupovat k určitým rozhraním API. Je to výhodné v případě, že váš obchodní případ použití vyžaduje některé určité API směrem k uživateli, například přístup k Disku Google. Nicméně může to být docela komplikované, pokud používáte více poskytovatelů OAuth k ověřování uživatelů nad místním systémem ověřování.

6 Autorizace

Autorizace [1] je funkce určování přístupových práv či privilegií ke zdrojům. Formálněji lze říci, že autorizovat znamená definovat určitou politiku přístupu.

Jednoduše můžeme danou definici ilustrovat na jednoduchém příkladu z reálného života, a to vlastnictví domu. Vlastník má plná přístupová práva k nemovitosti, jenž může udělit právo přístupu i dalším lidem. Z toho lze říct, že přístup k domu je oprávnění neboli akce, kterou můžeme se zdrojem provést. Dalšími oprávněními k domu může být jeho vybavení, úklid, oprava apod. Oprávnění se stává tímto privilegiem nebo právem, jenž je někomu přiděleno. Pokud tedy přidělíme například svému interiérovému designerovi oprávnění k vybavení domu, udělujeme mu právě privilegium. Na druhou stranu může být oprávnění do jisté míry spojeno s identitou, například v reálném životě je to při nástupu do letadla. Právě palubní vstupenka spolu s dokladem totožnosti cestujícího uděluje přístupová práva ke vstupu do letadla.

7 Programátorská dokumentace

Aplikace se skládá z několik vrstev, tyto si níže popíšeme včetně toho k čemu slouží a co případně obsahují.

7.1 DataAccessLayer

Tato vrstva slouží ke komunikaci s databázovými prostředky.

- **DaoClasses** – třídy obsahující jednotlivé SQL databázové dotazy, které slouží k reprezentaci dat uložených v databázi do tříd v modelové vrstvě a naopak k zápisu dat z modelových tříd do databáze,
- **Interfaces** – zde se nachází rozhraní pro přístup jednotlivým DAO třídám,
- **SQL** – nachází se zde inicializační skript, který slouží k vytvoření databáze, obsahuje jednotlivé tabulky a inicializaci základních datových struktur nutných k běhu aplikace,
- **Transactions** – nachází se zde třídy sloužící k transakční manipulaci s daty v databázi.

7.2 ModelLayer

V této modelové vrstvě se nacházejí třídy, které zobrazují popisující objekty.

- **Classes** – nacházejí se zde třídy reprezentující nejjednodušší datové struktury obsažené v aplikaci, a dále se zde nachází podadresář Views, kde jsou uloženy složitější datové struktury. Většina těchto tříd slouží jako modelové třídy pohledů.
- **Extensions** – adresář obsahuje statické třídy, ve kterých jsou implementovány statické metody sloužící k převodu mezi složitějšími datovými typy a strukturami,
- **Struct** – obsahuje datovou strukturu reprezentující uživatelské role

7.3 ServiceLayer

Tato vrstva slouží jako zprostředkovatel komunikace mezi aplikací a databázovou vrstvou.

- **Classes** – zde se nacházejí servisní třídy zprostředkovávající přenos dat,
- **Interfaces** – obsahuje rozhraní zprostředkovávající jednotlivé servisní třídy.

7.4 WebApplicationForOnlineTests

V následných podkapitolách si popíšeme jakou má projekt WebApplicationForOnlineTests adresářovou strukturu. Význam modelu MVC (Model-Views-Controller) si znovu popisovat nebudeme, vypíšeme jen jaké třídy obsahuje.

7.4.1 Properties

Ve složce properties se nachází soubor launchSetting.json, který popisuje způsob spuštění projektu například na jakém portu localhostu se má aplikace lokálně spustit.

7.4.2 wwwRoot

Složka wwwRoot je vylepšení v ASP.NET Core oproti MVC, která obsahuje samostatné složky pro různé typy statických souborů jako jsou JavaScript, CSS anebo obrázky.

7.4.3 Controllers

- **CourseController** – stará se o vytváření, editaci kurzů a jejich zobrazení do jejich pohledů,
- **ErrorController** – kontroler, který má na starost správu v případě chybového kódu 404,
- **HomeController** – je výchozí kontroler dané aplikace,
- **LectorController** – kontroler se stará o vytváření a zobrazení lektorů,
- **OrganizationController** – stará se o vytváření, editaci a zobrazení organizace,
- **StatisticsController** – stará se o vytváření a zobrazení statistiky uživatele a lektora,
- **TestController** – stará se o vytváření, editaci testů a jejich zobrazení do jejich pohledů (včetně otázek, které obsahuje),
- **UserController** – kontroler se stará o vytváření, editaci uživatelů/lektorů a jejich zobrazení do jejich pohledů.

7.4.4 Infrastructure

Nachází se zde složka IoC, která obsahuje soubor DefaultWindsorInstaller.cs. Registrují se zde jednotlivé komponenty do kontejneru, a tím se implementuje princip Dependency Injection.

7.4.5 Views

Zde jsou uloženy pohledy pro aplikaci, každá množina pohledů má svou vlastní složku pojmenovanou podle kontroleru (např. Course, Lector) Metody akcí budou vykreslovat pohledy ve složce svého kontroleru.

7.4.6 appsettings.json

Slouží k ukládání informací jako jsou například připojovací řetězce (connection strings) nebo specifická nastavení aplikace. Ve výchozím nastavení obsahuje šablona kromě souboru appsettings.json také soubor appsettings.Development.json. Nastavení definovaná v tomto souboru se načítají pouze do vývojového prostředí.

7.4.7 Program.cs

Tento soubor obsahuje statickou metodu Main(), která je vstupním bodem naší webové aplikace. Jediným účelem této metody je definovat hostitele a poté předat řízení třídě Startup.cs, kde se dokončí konfigurace aplikace. Hostitel je objekt, který zapouzdřuje prostředky aplikace, jako například vkládání závislostí (DI), logování nebo konfigurační nastavení.

7.4.8 Startup.cs

Z popisu výše a ukázky kódu lze vidět, že soubor Startup.cs je nakonfigurován pomocí rozšiřující metody UserStartup<T>. Daný název Startup je podle konvence ASP.NET Core, ale samozřejmě ji můžeme přejmenovat a vložit do metody jakýkoliv jiný název souboru. Daná třída obsahuje dvě metody Configure (nastavuje middleware pipeline pro náš projekt) a ConfigureServices (do ní registrujeme všechny potřebné služby v kontejneru DI a definujeme konfigurační nastavení pro některé z nich).

7.5 Struktura databáze

V této aplikaci je použita databáze Microsoft SQL Server, která obsahuje níže uvedené tabulky:

- **Course** – obsahuje informace o jednotlivých kurzech
- **Evaluation** – obsahuje hodnocení navázané k jednotlivým testům,
- **Organization** – slouží k uložení jednotlivých organizací,
- **Question** – obsahuje otázky navázané k jednotlivým testům,
- **Question_type** – v této tabulce jsou uloženy typy otázek, tabulka slouží pouze k poskytování dat,
- **Relation_test_course** – relační tabulka pro uchování vazby mezi tabulkou Test a Course,
- **Relation_user_course** – relační tabulka sloužící k uchování vazby mezi tabulkou User_ a Course,
- **Relation_user_test** – relační tabulka, která uchovává vazbu mezi tabulkou User_ a Test,
- **Response** – tabulka, kde jsou uloženy odpovědi navázané k jednotlivým otázkám,
- **Result** – obsahuje odkazy na uživatelem zvolené odpovědi a nebo samotnou textovou odpověď,
- **Test** – obsahuje informace o jednotlivých testech,
- **Test_result** – výsledková tabulka, kde jsou uloženy informace k vyhodnoceným testům,
- **User_** – obsahuje informace o všech uživateli včetně lektora a správce organizace,
- **User_role** – tabulka obsahuje uživatelské role, která slouží pouze ke čtení.

7.6 Autentizace a autorizace

Autentizace v této práci je pomocí kombinace uživatelského jména a hesla, kdy heslo je následně zakódováno algoritmem sha-256 a uloženo do tabulky v databázi. Díky zakódování není heslo z databáze zpětně čitelné a ověření uživatele se provádí pomocí porovnání zakódovaného řetězce v databázi a aktuálně zadaného hesla, které je opět zakódováno stejným algoritmem. Každé heslo, pod kterým se bude kterýkoliv uživatel přihlašovat musí tvořit minimálně osmimístní řetězec.

Autorizace má tři typy uživatelských rolí, a to administrátor, lektor a uživatel. Každá uživatelská role má jiná přístupová práva a umožňuje rozdílnou práci s aplikací.

8 Uživatelská dokumentace

V této kapitole si představíme uživatelskou správu aplikace, která je cílem této bakalářské práce. Tato aplikace se nazývá *Webová aplikace pro tvorbu online testů*. Na úvodní stránce uživatel nalezne menší informaci o aplikaci včetně náhledu některých bloků aplikace. Na stránce je k dispozici i samotná registrace organizace nebo přihlášení. Po přihlášení uživatele můžete aplikace mít z pohledu několika uživatelských rolí administrátora, lektora nebo uživatele.

8.1 Uživatelská role administrátor

Administrátor, neboli správce organizace, je uživatel, který zaregistroval svou organizaci a má tímto nejvyšší práva. Po přihlášení se mu zobrazí jeho uvítací stránka, na které se nachází naposledy přidání lektorů a informace o své organizaci. V této uživatelské roli je v samotném záhlaví ukázáno menu s přihlášením. Položky menu se nazývají *Kurzy*, *Lektoři* a *Uživatelé* a přihlašovací rozevírací se tlačítko s názvem svého uživatelského jména. Po kliknutí na toto tlačítko se uživateli zobrazí seznam s položky *Můj profil* a *Odhlášení*.

8.1.1 Položka kurzy

Po stisknutí položky *Kurzy* se uživateli následně ukáže seznam kurzů, které buď vytvořil nebo jej vytvořili lektorů pod touto organizací. Dále je uživateli nabízeno přidat nový kurz, vyhledání kurzu podle názvu nebo seřazení záznamu podle nabízejících možností. Pro každý záznam v seznamu je zobrazen název kurzu, jeho lektora, počet obsahujících testů a počet uživatelů, kteří jsou rozděleni do skupin. Jednotlivé záznamy lze smazat nebo si je prohlédnout podrobněji. Pokud se tak uživatel rozhodne a zobrazí si určitý kurz, může si jej i upravit (změna názvu kurzu, převedení lektora na jiného nebo přidání/odebrání uživatele z organizace. Nechybí ani tlačítko otevření samotného kurzu, kde se mu zobrazí jednotlivé testy.

8.1.2 Položka lektori

Po kliknutí na položku v menu *Lektoři* je na první pohled vidět seznam vytvořených lektorů. Tento seznam obsahuje ikonu lektora, příjmení, jméno, uživatelské jméno a počet kurzů, kde jsou zařazeni. Každý záznam tohoto seznamu lze zobrazit podrobněji nebo ho trvale smazat. Dále se tu nachází tlačítko vyhledat lektora v seznamu nebo dokonce seznam lektorů seřadit podle nabízejících se parametrů, například podle příjmení. Nechybí tu ani tlačítko vytvořit lektora, kdy se po kliknutí otevře formulář s položky vytvořit jméno, příjmení, uživatelské jméno, gerenovat heslo, informace o lektorovi a vybrání obrázku, který reprezentuje daného lektora. Generování hesla slouží k vytvoření prvotního hesla, které se předá novému lektorovi s uživatelským jménem. Po prvním přihlášení by si ovšem měl toho heslo změnit ve svém profilu. Nastane-li situace, kdy lektor zapomene

heslo, kontaktuje správce organizace. Ten má možnost nové heslo opětovně vygenerovat.

8.1.3 Položka uživatelé

Zbývající položka *Uživatelé* se stará o správu všech vytvořených uživatelů dané organizace. Na stránce je zobrazen seznam uživatelů, kde můžeme jednotlivé uživatele podrobně zobrazit, vytvořit nebo smazat. Nechybí tu ani tlačítko seřazení jednotlivých uživatelů nebo samotné vyhledávání uživatele, které vyhledává jak ve slově příjmení, tak ve slově jméno. Vytvoření uživatele pracuje obdobně jako vytvoření lektora, jen navíc obsahuje seznam kurzů, které mu lze rovnou přiřadit. Uživatel má stejnou možnost v případě zapomenutí hesla jak lektor.

8.2 Uživatelská role lektor

Tato uživatelská role má menší práva než administrátor, ale vyšší než běžný uživatel. Po přihlášení tento uživatel uvidí na stránce své naposledy vytvořené testy a své kurzy.

8.2.1 Položka kurzy

Po stisknutí položky *Kurzy* se uživateli zobrazí seznam jednotlivých kurzů, které spravuje. Seznam jednotlivých kurzů lze seřadit podle přípustných možností anebo si vyhledat název určitého kurzu. Jednotlivý záznam kurzu zobrazuje název kurzu, počet testů a počet uživatelů, které jsou přiřazeny ke kurzu. Každý záznam kurzu obsahuje tlačítko podrobnější informace a tlačítko smazání. V podrobnějším zobrazení se zobrazí uživateli název kurzu, název testů a název uživatelů přiřazujících ke kurzu. Nachází se i možnost upravit přiřazení buď uživatele, nebo testu ke danému kurzu. Po vstupu do kurzu se uživateli ukáže seznam přiřazených testů. Zde může lektor přímo v daného kurzu vytvořit nový test, aniž by musel do položky vytvořené testy. Jednotlivé testy obsahují tlačítka zobrazení testu, odebrání z kurzu, přiřadit uživatele (patřící do kurzu) k testu nebo přejít do vyhodnocení samotného testu. Přesměrujeme-li se do vyhodnocení testu, uvidí zde uživatel jednotlivé uživatele, kteří mají test přiřazený a test mají absolvovat. Pokud test obsahující určitý typ odpovědi, a to vlastní odpověď, musí lektor sám potvrdit zda je odpověď správná. Navíc existuje-li i k testu hodnocení, tak se následně po potvrzení odpovědi test sám vyhodnotí podle pravidel, které lektor při vytváření testu zadal.

#	Název kurzu	Počet testů	Počet uživatelů
1	Dějepis	2	2
2	Literatura II.	1	2
3	Literatura základy	1	3

Obrázek 11: Náhled na seznam kurzů

8.2.2 Položka vytvořené testy

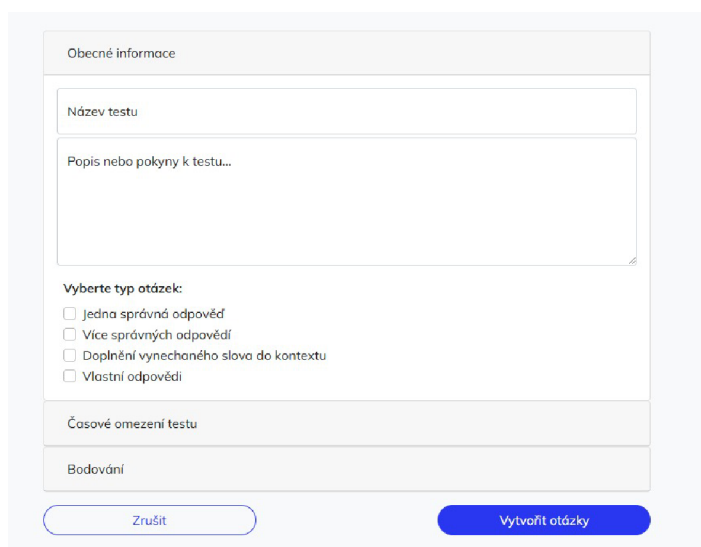
Položka *Vytvořené testy* jak od samotného názvu napovídá, obsahuje seznam jednotlivých vytvořených testů uživatelem. Každý jednotlivý test lze zobrazit, upravit dle zobrazených možností nebo trvale smazat. Na stránce kromě seznamu testů je i tlačítko na seřazení seznamu testů a vyhledávací pole, kde si můžeme najít libovolný název testu z našeho seznamu. Nachází se tu i tlačítko vytvořit nový test, které nás přeměruje na stránku obsahující formulář o třech částech.

První část se nazývá obecné informace, a to název testu, popis nebo pokyny k testu pro uživatele a typ otázek. Nachází se tu čtyři typy otázek:

- jedna správná odpověď,
- více správných odpovědí,
- doplnění vynechaného slova do kontextu,
- vlastní odpovědi.

Druhá část se nazývá časové omezení testu. Chceme-li nastavit časový limit k testu, právě zde potřebujeme přepnout tlačítko na časový limit a zapsat si příslušnou hodnotu minutách. Nachází se tu i zpřístupnění testu a uzavření testu, kdy zpřístupnění nesmí být datum starší než datum dnešní, poslední položka je počet pokusů.

Třetí část je bodování, tu uživatel může editovat nebo přímo přeskočit a přejít na vytváření jednotlivých otázek. V této třetí části může uživatel přepnout položku bodování stejným počtem bodů a automatické vyhodnocení. Pokud nepřepne uživatel bodování stejným počtem bodů, bude ke každé odpovědi psát libovolný počet bodů za správnou a špatnou odpověď. Rozhodne-li se přepnout, stačí pouze zapsat hodnotu za správnou a špatnou odpověď. Další možnost, kterou si uživatel může přepnout je automatické vyhodnocení (nepřepne-li nebude se automaticky vyhodnocovat). Po přepnutí jsou výchozí dvě možnosti hodnocení s procentuálním rozložením (100% - hodnota, hodnota - 0%), ovšem můžete přidat nejvýše další tři položky hodnocení. Jednotlivé hodnocení si může uživatel pojmenovat například na úspěš a neúspěš. A při pokračování na tlačítko vytvořit



Obrázek 12: Náhled vytváření testu

otázky, se nám zobrazí možnost jaký typ otázky (námi zvolené), chceme přidat a povrdíme. Zobrazí se nám formulář na vytvoření otázky, kde si zadáme název otázky, její odpovědi a správnou odpověď vybereme. Pokud uživatel nepřepnul bodování stejným počtem bodu, zobrazí se nám ještě bodování této otázky. Po vytvoření několika otázek si uživatel jednotlivé otázky může seřadit nebo případně některou z nich trvale vymazat. Ke konci už stačí vše uložit a vytvořený test se vloží do našeho seznamu vytvořených otázek.

8.2.3 Položka uživatelé

Daná položka *Uživatelé* je zcela totožná jako u uživatelské role admin v položce *Uživatelé*. Obecně jsou zde uživatelé, kterým bude cíleně vytvořen test a následně budou k němu přiřazeni.

8.2.4 Položka statistiky

Po stisknutí položky statistika si může lektor zobrazit výsledky jednotlivých uživatelů nebo skupiny uživatelů z daného kurzu pro určitý test. Každé zobrazení statistiky zobrazí graf, který ukazuje procentuální hodnoty správných a špatných odpovědí. Pokud si lektor zobrazí výsledky všech uživatelů z daného testu a určitého kurzu, může si všechny uživatele procházet včetně zobrazení grafu výsledků a zobrazí se mu i průměrná úspěšnost testu přiřazených uživatelů.

PORTÁL online testy

Kurzy Vytvořené testy Uživatelé Statistika **jesti.razickova**

Uživatelé

Vyhledej uživatele...

#	Příjmení	Jméno	Uživatelské jméno	Počet kurzů	Počet testů	
1	Kopecká	Jitka	jitka.kopeckova	6	4	<input type="button" value="Upravit"/> <input type="button" value="Smazat"/>
2	Koutrný	Jadřich	koutrny.jadrich	5	3	<input type="button" value="Upravit"/> <input type="button" value="Smazat"/>
3	Dubovská	Rieneta	dub.rieneta	4	1	<input type="button" value="Upravit"/> <input type="button" value="Smazat"/>

© PORTÁL Online testy 2021. Všechna práva vyhrazena.

Obrázek 13: Náhled na seznam uživatelů

8.3 Uživatelská role uživatel

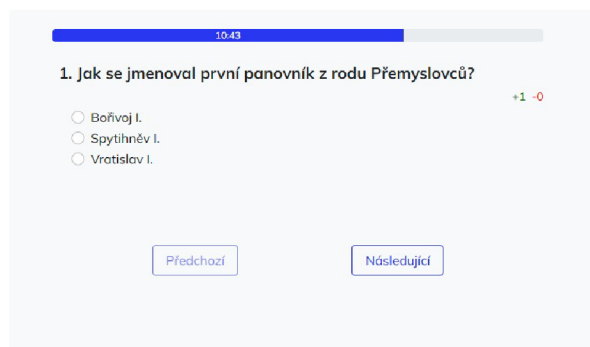
Uživatelská role uživatel je běžný uživatel, pro kterého je tato stránka především určena. Po přihlášení se mu zobrazí úvodní stránka, na kterém uvidí doposud nespustěné testy a své kurzy, do kterých byl přiřazen. V této uživatelské roli je v samotném záhlaví ukázáno menu (s položky kurzy, testy a statistika) s přihlášením.

8.3.1 Položka kurzy

Po stisknutí této položky se uživateli zobrazí bloky jednotlivých kurzů, které slouží k lepší přehlednosti než seznam. Každý blok jednotlivého kurzu obsahuje název kurzu, název lektora, počet nespustěných testů a počet celkových testů. Samozřejmě obsahuje i tlačítko vstup do kurzu, kde následně vidíme seznam testů, které kurz obsahuje. Daný seznam testů lze filtrovat dle nespustěného testu nebo test s možností zopakování testu.

8.3.2 Položka testy

Po kliknutí na položku v menu *Testy* je zobrazen seznam přiřazených testů. Jednotlivé testy lze si vyfiltrovat na nespustěné testy nebo testy k možnosti zopakování. Tento seznam obsahuje název testu, název kurzu (zařazení testu), zpřístupnění testu, uzavření testu, počet pokusů a hodnocení. Jednotlivé testy obsahují tlačítko spustit test, který nás přeměruje na stránku zobrazující pokyny před samotným vypracováním testu. Takové pokyny obsahují navíc informace k testu, jako například počet pokusů nebo časový limit. Pokud je v testu nastaveno vyhodnocení nebo neobsahuje určitý typ otázky, zobrazí se mu po vypracování testu vyhodnocení. Je-li uživateli umožněno test zopakovat, ukládá se mu do vyhodnocení testu vždy poslední pokus.



Obrázek 14: Náhled vypracování testu

8.3.3 Položka statistiky

Položka *Statistika* slouží uživateli k zobrazení souhrnných výsledků testů. Na této stránce si uživatel zvolí, podle kterého kurzu se mu budou zobrazovat testy. Následně se mu zobrazí graf s procentuálními výsledkem správných odpovědí ku špatným. Uživatel si může vybrat i všechny testy z kurzu, které si může jednotlivě procházet včetně zobrazení grafu. V takovém případě se mu navíc objeví i jeho průměrná úspěšnost kurzu.



Obrázek 15: Náhled na uživatelskou statistiku

9 Testování aplikace

Aplikace byla vytvářena, testována a odlaďována především na lokálním webovém serveru IIS zprostředkovaném programem Microsoft Visual Studio 2019. Pro tyto účely byla využita lokální databáze Microsoft SQL Server 2019. Lokální prostředky byly spouštěny na notebooku HP Elitebook 840 G6 s procesorem Intel Core i7-8565U, 16GB RAM a s monitorem v rozlišení FHD tedy 1920x1080 px. V lokálním prostředí aplikace běžela bez problémů a hardwarové prostředky byly pro její běh dostatečné.

Aplikace byla též testována na free hostingovém webovém serveru, kde je nyní stále vyvěšena pro potřeby jednodušší kontroly bez nutnosti nahrávání souborů, přiložených u této bakalářské práce, na webový server. Stránka se nachází na webové adrese www.portalonlinetests.aspifyhost.cz, přístupové údaje k testovacím datům jsou případně k dispozici v souboru Readme.txt na přiloženém CD.

Pro bezproblémovou funkci webové aplikace je zapotřebí Microsoft SQL Server 2019 a dále Microsoft IIS 2019 s podporou ASP .NET s podporou .NET5.

Zobrazení programu bylo testováno na nejrozšířenějších webových prohlížečích, a to Google Chrome a Microsoft Edge, kde se aplikace zobrazovala a fungovala bez problémů.

Závěr

Cílem mé bakalářské práce je webová aplikace pro tvorbu online testů. Webová aplikace umožňuje zaregistrovat jakoukoliv organizaci, kterou nemusí být konkrétně škola. Během registrace se zároveň registruje i správce organizace, se kterým je spjata uživatelská role administrátor.

Ve webové aplikaci se můžete přihlásit pod uživatelskými rolemi administrátor, lektor nebo běžný uživatel. Uživatelská role s nejvyššími právy tj. administrátor umožňuje vytvářet příslušné kurzy, lektory a uživatele pro danou organizaci, kterou může zároveň spravovat. Tato uživatelská role nemůže vytvářet testy, jednotlivé testy vytváří pouze lektor, a tudíž administrátor nemůže ani jednotlivé testy přiřazovat uživatelům v kurzu. Uživatelská role lektor vytváří kurzy, testy a uživatele. Ke každému kurzu jsou přiřazeny testy a uživatelé, kteří mohou být ještě dále přiřazeny k jednotlivým testům v kurzu. Lektor vytváří testy se čtyřmi možnostmi otázek (jedna správná odpověď, více správných odpovědí, vyplnění chybějícího slova do kontextu a vlastní odpověď uživatele). Každý test má při vytváření volitelně nastavené možnosti například časový limit, časové zpřístupnění/uzavření testu, počet pokusů, způsob bodování anebo automatické vyhodnocení. Třetí uživatelskou rolí je běžný uživatel, takový uživatel absolvuje testy, které mu přiřadili jednotliví lektoři pod příslušným kurzem. Před samotným spuštěním testu je uživatel obeznámen s pokyny k testu, jeho časovým limitem nebo zbývajícím počtu pokusů. Po vyplnění testu se zobrazí uživateli stránka, kde se dozví informace o svém výsledku z testu a případně známce (nastaví-li lektor při vytváření testu). Pokud v testu, který lektor vytvoří je obsažena otázka vlastní odpověď, uživatel je informován o této situaci a čeká na manuální vyhodnocení odpovědi od lektora.

V uživatelské roli lektor a uživatel nalezneme v menu položku statistika, kde se zobrazí, jak si uživatel nebo uživatelé vedli v jednotlivých testech pod určitým kurzem. Každému uživateli je zobrazen graf a stručný popis informací o výsledku z daného testu (například úspěšnost testu, počet získaných bodů nebo počet správných/špatných odpovědí).

Conclusions

The part of the bachelor thesis is creating a web application for creating online tests. The web application allows you to register any organization, which does not have to be a school specifically. During the registration process, the administrator of the organization is also registered, with which the user role of administrator is associated.

In the web application you can log-in as administrator, lecturer or regular user. The user role with the highest privileges, i.e. administrator, allows you to create relevant courses, lecturers and users for a given organisation, which you can also manage. This user role cannot create tests, only the lecturer can create individual tests and therefore the administrator cannot assign individual tests to users in a course. The lecturer user role creates courses, tests, and users. Tests and users are assigned to each course and can be further assigned to individual tests in the course. The lecturer creates tests with four question options (one correct answer, multiple correct answers, fill in the missing word in context, and the user's own answer). Each test has optional options when created, such as time limit, time access/closure, number of attempts, scoring method, or automatic scoring. The third user role is the regular user, such a user completes the tests assigned by the individual lecturer under the relevant course. Before running the test itself, the user is familiar with the test instructions, the time limit or the number of attempts remaining. After completing the test, the user is presented with a page where the user is informed of own test result and, if necessary, the grade (if set by the instructor when creating the test). If the test the lecturer creates contains a self-answer question, the user is informed of this situation and waits for the lecturer to manually evaluate the answer.

In the lecturer and user roles, the menu item statistics will show how the user or users performed in each test under a particular course. Each user is shown a graph and a brief description of the result information from that test (e.g., pass rate, number of points earned, or number of right/wrong answers).

A Obsah příloženého CD/DVD

bin/

Instalátor `INSTALATOR.ZIP` programu, ZIP soubor obsahující vypublikovanou webovou stránku a inicializační skript pro vytvoření databáze.

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 `WEBOVÁ APLIKACE PRO TVORBU ONLINE TESTŮ` se všemi potřebnými zdrojovými texty, knihovnamy a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu, zabalené v ZIP archivu. Obsahuje také projekt Microsoft Visual Studio 2019 v ZIP archivu.

readme.txt

Instrukce pro instalaci a spuštění programu `WEBOVÁ APLIKACE PRO TVORBU ONLINE TESTŮ`, včetně všech požadavků pro jeho bezproblémový provoz.

Literatura

- [1] AUTH0, introduction to identity. What is Authorization [online]. [cit. 29.11.2021]. Dostupné z: <https://auth0.com/intro-to-iam/what-is-authorization/>
- [2] BĚHÁLEK Marek. Programovací jazyk C# [online]. [cit. 29.10.2021]. Dostupné z: <http://www.cs.vsb.cz/behalek/vyuka/pcsharp/text.pdf>
- [3] BOOKWIDGETS. Webová aplikace na tvorbu testů [online]. [cit. 29.11.2021]. Dostupné z: <https://www.bookwidgets.com>
- [4] CAREY P.: HTML5 and CSS3. Cengage Learning 2017
- [5] CASTLE PROJECT, Build your .NET projects on a rock solid foundation. Castle Project [online]. [cit. 29.11.2021]. Dostupné z: <http://www.castleproject.org>
- [6] EDUMO. Webová aplikace na tvorbu testů [online]. [cit. 29.11.2021]. Dostupné z: <https://test.edumo.eu/>
- [7] ITESTER. Webová aplikace na tvorbu testů [online]. [cit. 29.11.2021]. Dostupné z: <https://www.itester.cz>
- [8] JAVATPOINT. What is Bootstrap [online]. [cit. 29.11.2021]. Dostupné z: <https://www.javatpoint.com/what-is-bootstrap>
- [9] LAPTEVA Anna. 14 Types of Quiz Questions to Use in Your eLearning Course [online]. [cit. 29.10.2021]. Dostupné z: <https://www.ispringsolutions.com/blog/6-types-of-quiz-questions-to-use-in-your-e-courses>
- [10] LIVEBOOK MANNING, dependency injection in dot net. Chapter 10. Castle Windsor [online]. [cit. 29.11.2021]. Dostupné z: <https://livebook.manning.com/book/dependency-injection-in-dot-net/chapter-10/>
- [11] MICROSOFT, documentation. What's new in .NET 5 [online]. [cit. 29.11.2021]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/core/dotnet-five>
- [12] NAGEL Christian, Bill Evjen, Jay Glynn, Morgan Skinner, Karli Watson: C# 2008 Programujeme profesionálně. Computer Press, 2009
- [13] RASCASONE. JavaScript pro začátečníky: co to je a jak funguje [online]. [cit. 29.11.2021]. Dostupné z: <https://www.rascasone.com/cs/blog/co-je-javascript-pro-zacatecniky>
- [14] RHALMI Mohammed. Definition of testing, assessment and evaluation [online]. [cit. 29.10.2021]. Dostupné z: <https://www.myenglishpages.com/blog/definition-of-testing-assessment-and-evaluation/>

- [15] SCIENCE, Macquarie University Australia. Authentication for Web Applications [online]. [cit. 29.11.2021]. Dostupné z: http://web.science.mq.edu.au/matr/courses/web_applications/9_Authentication/notes.html
- [16] TECHTARGET. Web application [online]. [cit. 29.11.2021]. Dostupné z: <https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>
- [17] TESTEDIT. Webová aplikace na tvorbu testů [online]. [cit. 29.11.2021]. Dostupné z: <http://www.testedit.cz>
- [18] TRELSEN A., Japikse P.: Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming. Apress 2021
- [19] TUTORIALREPUBLIC. Bootstrap Tutorial [online]. [cit. 29.11.2021]. Dostupné z: <https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/>
- [20] TUTORIALSPPOINT, simply easy learning. What is AJAX? [online]. [cit. 29.12.2021]. Dostupné z: https://www.tutorialspoint.com/ajax/what_is_ajax.htm
- [21] TUTORIALSTEACHER. ASP.NET Core Tutorials [online]. [cit. 29.11.2021]. Dostupné z: <https://www.tutorialsteacher.com/core/>
- [22] TUTORIALSTEACHER. IoC Container [online]. [cit. 29.11.2021]. Dostupné z: <https://www.tutorialsteacher.com/ioc/ioc-container>
- [23] WIKIPEDIA, the free encyclopedia (comp.). C Sharp [online]. [cit. 29.10.2021]. Dostupné z: [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [24] WIKIPEDIA, the free encyclopedia (comp.). Dependency injection [online]. [cit. 29.11.2021]. Dostupné z: https://en.wikipedia.org/wiki/Dependency_injection
- [25] WIKIPEDIA, the free encyclopedia (comp.). JavaScript [online]. [cit. 29.11.2021]. Dostupné z: <https://en.wikipedia.org/wiki/JavaScript>
- [26] WIKIPEDIA, the free encyclopedia (comp.). Microsoft SQL Server [online]. [cit. 29.11.2021]. Dostupné z: https://en.wikipedia.org/wiki/Microsoft_SQL_Server
- [27] WIKIPEDIA, the free encyclopedia (comp.). NET Framework [online]. [cit. 29.11.2021]. Dostupné z: https://en.wikipedia.org/wiki/.NET_Framework
- [28] WIKIPEDIA, the free encyclopedia (comp.). Web application [online]. [cit. 29.11.2021]. Dostupné z: https://en.wikipedia.org/wiki/Web_application