

České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra počítačů



Bakalářská práce

TestQuiz – webová aplikace pro testování

Vypracovala: BASHKOVA KSENIIA
Vedoucí práce: ING. BOŽENA MANNOVÁ, PH.D.
Studijní program: Softwarové inženýrství a technologie

3.01.2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Bashkova** Jméno: **Kseniia** Osobní číslo: **469974**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

TestQuiz - webová aplikace pro testování

Název bakalářské práce anglicky:

TestQuiz - Web application for testing

Pokyny pro vypracování:

Cílem práce je analýza, návrh, implementace a testování webové aplikace TestQuiz, která bude sloužit jako pomůcka studentům během studia. Aplikace bude sloužit k vytváření a editování testových otázek a odpovědí, k vytváření a spouštění testů, zobrazování výsledků testů a sdílení testů s ostatními studenty.

- 1) Seznamte se s problematikou a analyzujte existující systémy poskytující podobnou funkcionalitu.
- 2) Na základě vyhodnocení těchto poznatků, specifikujte požadavky na funkcionalitu navrhovaného systému.
- 3) Seznamte se s technologiemi potřebnými pro vytvoření aplikace.
- 4) Navrhněte a vhodně zvolenými nástroji implementujte aplikaci.
- 6) Otestujte aplikaci včetně uživatelských testů a výsledky vyhodnoťte.

Při zpracování využívejte prostředky SI.

Seznam doporučené literatury:

- [1] Roger S. Pressmann Bruce Maxim: Software Engineering: A Practitioner's Approach , ISBN-10: 9780078022128
- [2] H. Hudák, „VŠE TESTER,“ [Online]. Available: <http://www.vsetester.wz.cz/>
- [3] IBM, 'Rational Software Architect,' [Online]. Available: <https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=diagrams-use-case>.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Božena Mannová, Ph.D., kabinet výuky informatiky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **28.06.2021**

Termín odevzdání bakalářské práce: **04.01.2022**

Platnost zadání bakalářské práce: **19.02.2023**

Ing. Božena Mannová, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Studentka bere na vědomí, že je povinna vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

_____ Datum převzetí zadání

_____ Podpis studentky

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Bashkova Kseniia
V Praze, 3.01.2022

Poděkování

Ráda bych poděkovala zejména své vedoucí Ing. Boženě Mannové, Ph.D. za její odborné vedení, rady, ochotu a celkovou vstřícnost při konzultacích a vypracování této bakalářské práce. Dále bych také ráda poděkovala své rodině a přátelům za podporu a pomoc během studia

Abstrakt

Práce je zaměřena na vývoj webové aplikace TestQuiz, která slouží jako pomůcka studentům během studia. V rámci této práce byla provedena detailní analýza, návrh, implementace a testování webové aplikace.

Klíčová slova: testování, webová aplikace, Spring Boot, React, PostgreSQL, Heroku.

Abstract

The work is focused on the development of the web application TestQuiz, which serves as an utility for to students during the study. Within this work, a detailed analysis, design, implementation and testing of a web application was performed.

Keywords: testing, web application Spring Boot, React, PostgreSQL, Heroku.

Obsah

1. Úvod.....	1
1.1. Motivace.....	1
1.2. Cíle projektu a cílová skupina.....	1
2. Softwarová Analýza.....	2
2.1. Analýza existujících řešení.....	2
2.1.1. Quizizz.....	2
2.1.2. VŠE TESTER.....	4
2.1.3. Master-test.....	5
2.1.4. Vyhodnocení.....	6
2.2. Systémové požadavky.....	7
2.2.1. Funkční požadavky.....	7
2.2.2. Kvalitativní požadavky a omezení.....	11
2.3. Případy užití.....	12
2.3.1. Uživatelské role.....	12
2.3.2. Diagram případů užití.....	12
2.4. Analytický doménový model tříd.....	18
2.5. Shrnutí kapitoly.....	19
3. Volba technologií.....	20
3.1. Serverová část.....	20
3.1.1. Výběr programovacího jazyka.....	20
3.1.2. Výběr frameworku.....	22
3.2. Klientská část.....	22
3.2.1. Výběr programovacího jazyku.....	22
3.2.2. Výběr frameworku.....	23
3.3. Databáze.....	25
3.4. Hostinky.....	25
3.4.1. Heroku.....	26
3.4.2. OpenShift.....	26
3.4.3. Netlify.....	26
3.4.4. Vyhodnocení.....	27
3.5. Shrnutí kapitoly.....	27
4. Implementace.....	28
4.1. Fyzická architektura aplikace.....	28
4.2. Použité knihovny.....	29
4.3. Další nástroje.....	30
4.3.1. Buildovací nástroj.....	30
4.3.2. Verzovací nástroj.....	30

4.3.3.	Vývojové prostředí.....	30
4.4.	Zabezpečení.....	31
4.5.	Rozhraní webové služby.....	32
4.5.1.	REST.....	32
4.5.2.	Swagger.....	32
4.6.	Nasazení.....	34
4.6.1.	Serverová část.....	34
4.6.2.	Klientská část.....	34
4.7.	Shrnutí kapitoly.....	34
5.	Testování.....	36
5.1.	White-box testování.....	36
5.1.1.	Jednotkové testy.....	36
5.1.2.	Integrační testy.....	37
5.2.	Black-box testování.....	37
5.2.1.	Systémové testování.....	37
5.2.2.	Uživatelské testování.....	37
6.	Závěr.....	39
6.1.1.	Rozšíření systému z pohledu požadavků.....	39
6.1.2.	Rozšíření z pohledu implementace.....	40
	Citovaná literatura.....	41
	Seznam zkratk.....	44

Seznam příloh

Příloha A.	Případy užití.....	45
Příloha B.	Scénáře testování.....	60
Příloha C.	Výsledné ankety testování.....	68
Příloha D.	Seznam odkazů.....	70
Příloha E.	Snímky uživatelského rozhraní.....	71

Seznam obrázků

Obr. 1. Quizizz.....	3
Obr. 2. VŠE TESTER	5
Obr. 3. Mater-test.....	6
Obr. 4. Uživatelské role	12
Obr. 5. Diagram případů užití – Přihlášení	14
Obr. 6. Diagram případů užití – Téma/podtéma	15
Obr. 7. Diagram případů užití – Otázky/odpovědi	16
Obr. 8. Diagram případů užití – Testy	17
Obr. 9. Diagram případů užití – Reporty	17
Obr. 10. Analytický doménový model.....	18
Obr. 11. Diagram nasazení.....	28
Obr. 12. Proces registrace, přihlášení a autorizace uživatele.....	31
Obr. 13. Ukázka části uživatelského rozhraní projektu Swagger UI	33
Obr. 14. Použití šablony.....	36
Obr. 15. Webová aplikace. Přihlašovací stránka	71
Obr. 16. Webová aplikace. Domovská stránka	71
Obr. 17. Webová aplikace. Filtrace veřejných testů	72
Obr. 18. Webová aplikace. Zobrazení vytvořených testů	72
Obr. 19. Webová aplikace. Vytvoření otázky.....	73
Obr. 20. Webová aplikace. Nahlédnutí otázky	74
Obr. 21. Webová aplikace. Zobrazení testu	74
Obr. 22. Webová aplikace. Sdílení testu	75
Obr. 23. Webová aplikace. Spouštění testu	75
Obr. 24. Webová aplikace. Nahlédnutí výsledků testu	76

Seznam tabulek

Tab. 1. Výhody a nevýhody existujících řešení	7
Tab. 2. Rozdíly mezi NoSQL a SQL databázemi	25
Tab. 3. Vztah CRUD operací k HTTP metodám.....	32

1. Úvod

1.1. Motivace

Každý student hledá způsob, jak se učit efektivně, tak aby učení zabralo co možná nejméně času a co nejvíce si toho zapamatoval.

V dnešní době je již vynalezena spousta efektivních metod učení, například myšlenkové mapy, metody efektivního čtení, učení se s kartičkami a další.

Po dlouhých letech studia a zkoumání různých technik jsem přišla na to, že nejefektivnější metodou je testování sebe sama (self-testing).

Testování obecně neslouží (jak jsme zvyklí si myslet) jenom k tomu, aby učitel ověřil, jestli něco umíme nebo neumíme. Jedná se především o velice účinnou techniku, jak si informace efektivně zapamatovat. Čím častěji si budeme vyvolávat uložené informace z paměti, tím je snadnější si to vybavit i příště. Paměť se opakovaným vyvoláváním informací rozšiřuje a posiluje.

Z mého pohledu nejlepší způsob, jak zvládnout písemnou zkoušku je prostě vytvořit si soubor vlastních otázek ze zápisků, učebnic či skript a odpovědět si na ně. Pak už zbývá pouze zkontrolovat odpovědi a ty méně jisté podrobněji dále prostudovat.

Právě proto jsem se rozhodla v rámci bakalářské práce navrhnout a implementovat aplikaci která bude sloužit jako pomůcka studentům během studia.

1.2. Cíle projektu a cílová skupina

Cílem této bakalářské práce je analýza, návrh, implementace a testování webové aplikace TestQuiz, která bude sloužit jako pomůcka studentům během studia. Výstupem práce je fungující aplikace a detailní analýza.

Hlavními rysy aplikaci jsou:

- Vytvořit testové otázky s odpověďmi (i ve formě obrázků)
- Vytvořit témata a podtémata a přidávat k nim otázky
- Vytvořit a spustit test
- Spustit test pouze špatně zodpovězených otázek
- Zobrazování výsledků testu
- Sdílet test s ostatními

Viz podkapitola 2.2.1 *Funkční požadavky*.

2. Softwarová Analýza

Tato kapitola je důležitou fází vývoje aplikace TestQuiz.

Nejprve je provedena analýza existujících řešení. Na základě této analýzy jsou určeny požadavky na funkce a kvalitu systému, navržené případy užití včetně scénářů a vytvořen analytický doménový model tříd.

Všechny diagramy v této kapitole jsou modelované pomocí modelovacího jazyka UML¹ a nástroje pro systémovou analýzu a návrh – Enterprise Architect.

2.1. Analýza existujících řešení

Tato část se bude zabývat analýzou již existujících aplikací, které umožní vytvářet vlastní testy a cvičit je. Také byla určena vhodná struktura pro lepší a důkladnější analýzu:

- Základní informace
- Registrace
- Vytvoření testu
- Vytvoření otázek a odpovědí
- Spuštění testu
- Zobrazení výsledků
- Podporovaný jazyk (myslí se tím jazyk ve kterém je zobrazena aplikace, nikoliv jazyk, ve kterém je možné psát otázky s odpověďmi)

2.1.1. Quizizz

Základní informace.

Aplikaci založili Antik Gupta a Deepak Joy Cheenath v roce 2015 a od té doby se aktivně využívá ve školách a na univerzitách.

Aplikace je určena pro pedagogy, studenty, žáky a jejich rodiče. Quizizz slouží k vytvoření online testů. V databázi se již také nacházejí sdílené veřejné testy, které byly vytvořeny učiteli, studenty či jinými uživateli.

Quizizz umožňuje učitelům vytvářet skupiny, testy, otázky a následně zobrazit výsledky každého žáka. Quizizz umožňuje žákům vytvářet testové otázky, zúčastnit se testu do kterého byli přihlášení učitelem a také se zúčastnit veřejně dostupných testů, které vytváří učitelé po celém světě.

¹ Diagramy UML (Unified Modeling Language) ilustrují kvantifikovatelné aspekty systému, které lze popsat vizuálně, jako jsou vztahy, chování, struktura a funkčnost [2].

Jak již bylo zmíněno aplikace Quizizz je určená jak pro učitele, tak i pro studenty proto tato aplikace má odlišnou funkcionalitu pro tyto role. Protože cílem mé práce je vývoj aplikace za účelem samostudia, tak další analýza Quizizz bude zahrnovat pouze roli studenta.

Registrace.

Pro vytvoření testu – nutná.

Pro vyplnění zveřejněného testu – není nutná.

Vytvoření testu.

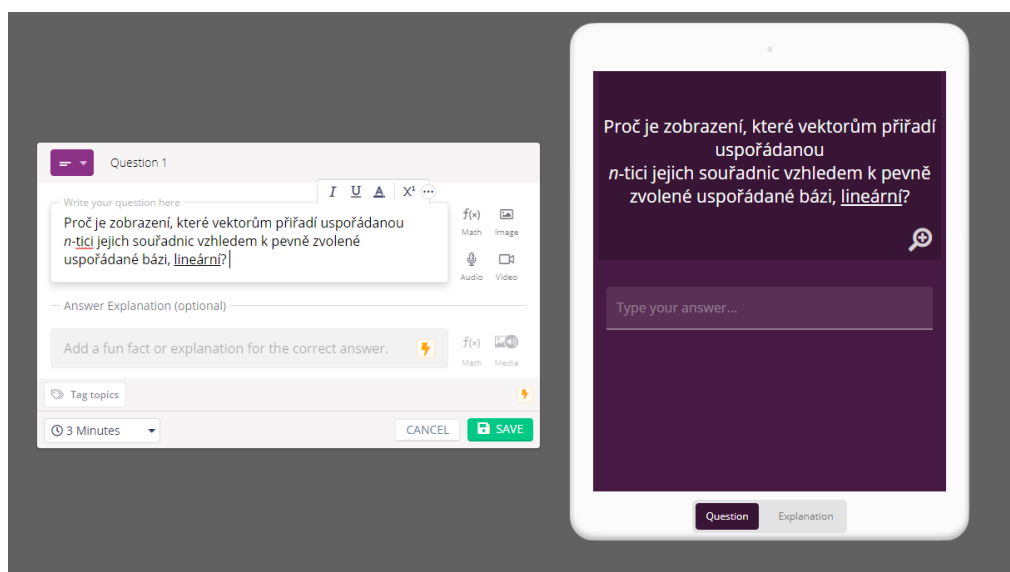
Pro vytvoření testu je nutné uvést název testu. Volitelnou možností je vložení profilového obrázku. Nakonec je možné určit, jestli test bude soukromý nebo veřejný.

Vytvoření otázek a odpovědí.

Po založení testu, aplikace umožní sestavit otázky a odpovědi. Aplikace má na výběr několik druhů otázek: multiple choice², fill-in-the-blank³ a open-ended⁴ otázky. Aplikace také podporuje vložení obrázků, audio a video a matematických symbolů k jednotlivým otázkám.

Další možnost, jak vytvořit otázky a odpovědi je import CSV⁵ souborů.

Vytváření testových otázek je znázorněno na *obrázku 1. Quizizz.*



Obr. 1. Quizizz

² Multiple Choice – otázka s více možnostmi. Umožňuje vybrat jednu nebo více správných odpovědí z řady potenciálních odpovědí.

³ Fill-in-the-blank – otázka kde se doplňují odpovědi do prázdných polí v textu.

⁴ Open-ended – otázka, která vyžaduje více než jednoslovnou odpověď.

⁵ Comma-Separated Values – jednoduchý souborový formát určený pro výměnu tabulkových dat. Takovýto soubor se skládá z řádků, ve kterých jsou jednotlivé sloupce v rámci jednoho řádku odděleny oddělovacím znakem. Nejčastěji to bývá čárka nebo středník [28].

Spuštění testu.

Po úspěšném vytvoření se test zobrazí v seznamu všech testů uživatele, kde se také spouští. Pro spuštění testu je možné nastavit základní parametry kvízu, mezi které patří například: promíchání otázek a odpovědí nebo časový limit u otázek. Jakmile jsou parametry nastavené, test může být spuštěn.

Zobrazení výsledků.

Po vyplnění testu je možnost zobrazit podrobnější informace, například celkovou úspěšnost otázek a čas strávený testem.

Podporovaný Jazyk

Angličtina.

2.1.2. VŠE TESTER

Základní informace.

VŠE TESTER je aplikace, která byla vytvořena Davidem Kadlckem pro spuštění testů uložených v textových souborech. Existují dvě verze, desktopová a android verze.

Aplikace je určená pro studenty. Je jednoduchá pro používání, ale neposkytuje možnost vkládat obrázky a vytvářet testy přímo v aplikaci. Tuto aplikaci jsem využívala osobně během studia, ale chyběla mi právě ta možnost vložení grafických objektů.

Registrace.

Není nutná.

Vytvoření testu.

Aplikace neumožňuje vytvoření testů. Místo toho je možné vybrat textový soubor s otázkami. Textový soubor hraje v tomto případě roli testu.

Vytvoření otázek a odpovědí.

Otázky se také vytváří v textovém souboru.

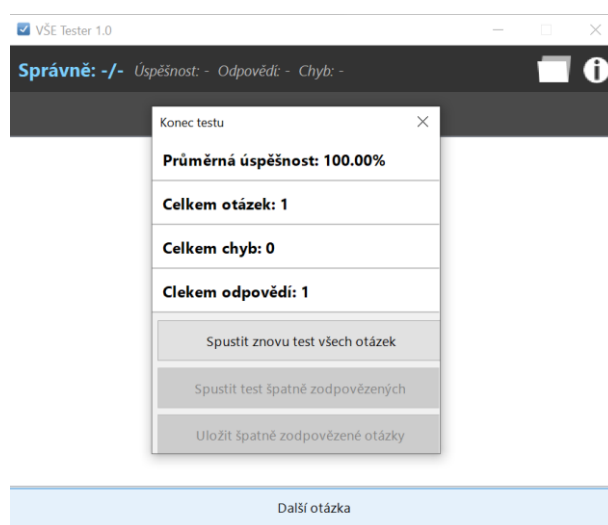
Požadovaný formát textových souborů s testy – jednotlivé otázky a odpovědi se mají nacházet na samostatných řádcích. Správné odpovědi jsou označeny symbolem "+", špatné symbolem "-" [1].

Spuštění testu.

Pro spuštění testu je potřeba vybrat textový soubor, který se musí nacházet lokálně v počítači uživatele. Po skončení testu, aplikace umožňuje spustit test znovu nebo pustit test pouze špatných odpovědí.

Zobrazení výsledků.

Výsledky se zobrazují jak během testu, tak i po dokončení. Zobrazení výsledků po dokončení testu je znázorněno na *obrázku 2. VŠE TESTER*.



Obr. 2. VŠE TESTER

Podporovaný Jazyk.

Čeština. Možnost vybrat jazyk, ve kterém bude napsán test, otázky a odpovědi.

2.1.3. Master-test

Základní informace.

Master-test je aplikace, která byla vytvořena Janem Chovanským v roce 2000. Hlavním cílem aplikace je provádění interaktivního testování znalostí studentů a žáků.

Registrace.

Registrace je nutná jak pro učitele, tak i pro studenta.

Vytvoření testu.

Pro vytvoření testu stačí pouze vybrat možnost vytvořit test. Název testu se uvádí po ukončení vytvoření testu.

Vytvoření otázek a odpovědí.

Po založení testu aplikace umožní sestavit otázky a odpovědi. Aplikace má na výběr několik druhů otázek: multiple choice a fill-in-the-blank. Také aplikace umožní přidat váhu otázce. Příklad vytváření otázek a odpovědí je znázorněn na *obrázku 3. Mater-test*.

The screenshot shows a web form for creating a question. At the top, there is a text input field for the question title, containing the text "At' A je čtvercová reálná matice. Vyberte pravdivé tvrzení." Below this is a checkbox labeled "Дополнительно" (Additional) which is currently unchecked. Underneath is a dropdown menu for "Тип вопроса:" (Question type) set to "Однозначный Ответ" (Single Answer). A section titled "Ответ" (Answer) contains four radio button options: "Pokud je A regulární, pak je A diagonalisovatelná.", "Pokud je A diagonalisovatelná, pak je A singulární.", "Matice A má nulové vlastní číslo právě tehdy, když je A singulární." (which is selected), and "Pokud má A mílové vlastní číslo, pak je A regulární." Below the options is a "Добавить Ответ" (Add Answer) button. At the bottom, there is a "Вес Вопроса:" (Question weight) dropdown set to "1" and a "Готово" (Done) button.

Obr. 3. Mater-test

Spuštění testu.

Po úspěšném vytvoření se test zobrazí v seznamu všech testů uživatele a tam se test spouští. Pro spuštění testu je nutné nastavit čas po kterém se test automaticky ukončí.

Zobrazení výsledků.

Po vyplnění testu se zobrazí podrobnější informace: celková úspěšnost otázek, celkový počet otázek, počet chybových otázek.

Podporovaný Jazyk.

Ruština.

2.1.4. Vyhodnocení

V *tabulce 1. Výhody a nevýhody existujících řešení* jsou uvedeny výhody a nevýhody každé aplikace.

Aplikace	Výhody	Nevýhody
Quizizz	<ul style="list-style-type: none"> - Lze vkládat obrázky. - Lze vytvářet různé druhy testů. - Lze vytvářet témata a podtémata. - Lze sdílet testy s uživateli. 	<ul style="list-style-type: none"> - Neintuitivní uživatelské rozhraní. - Určena především pro testování ve školách než pro samostatné studium. - Je komerční.
VŠE TESTER	<ul style="list-style-type: none"> - Jednoduchost v používání. - Není komerční. 	<ul style="list-style-type: none"> - Nelze vkládat obrázky. - Nelze vytvářet témata a podtémata. - Zastaralé grafické rozhraní. - Nelze sdílet testy s uživateli.
Master-test	<ul style="list-style-type: none"> - Jednoduchost v používání. - Není komerční. - Lze vytvářet témata a podtémata. 	<ul style="list-style-type: none"> - Nelze vkládat obrázky. - Zastaralé grafické rozhraní. - Nelze sdílet testy s uživateli.

Tab. 1. Výhody a nevýhody existujících řešení

Existuje mnoho dalších řešení kromě výše zmíněných jako například Kahoot, Educandy, SOCRATIVE apod. Většina těchto aplikací je zaměřena na používání ve školách a pro moje účely mají zbytečné funkcionality a příliš složité interaktivní rozhraní. Ve většině případů jsou tyto aplikace placené.

Z analýzy existujících projektů, jejich výhod a nevýhod, jsou odvozeny požadavky na systém TestQuiz, které jsou popsány v následující podkapitole.

2.2. Systémové požadavky

Pochopení požadavků patří k nejsložitějším úkolům, které řeší softwarový inženýr. Podle určených požadavků se dále vytváří celý systém [2].

2.2.1. Funkční požadavky

V této podkapitole je definováno, co systém bude uživatelům umožňovat. Pro přehlednost všechny požadavky rozdělené do pěti skupin: přihlášení, téma/podtéma, otázky/odpovědi, testy a reporty.

Požadavky mají následující atributy:

- Jednoznačný identifikátor/číslo
- Název
- Popis

FRQ2 – Přihlášení.

- SRQ 101 – Zaregistrovat se v systému.

System umožní nezaregistrovanému uživateli zaregistrovat se v systému. Registrace bude považovaná za úspěšnou, pokud uživatel vyplní přihlašovací formulář, kde zadá svůj unikátní login, heslo, jméno, příjmení a validní e-mailovou adresu.

- SRQ 102 – Přihlásit se do systému.
System umožní uživateli přihlásit se do systému pomocí loginu a hesla které byly uvedené při registraci.
- SRQ 103 – Odhlásit se ze systému.
System umožní uživateli odhlásit se ze systému, pokud je uživatel přihlášený.
- SRQ 104 – Změnit heslo.
System umožní přihlášenému uživateli změnit heslo.
- SRQ 105 – Změnit e-mail.
System umožní přihlášenému uživateli změnit e-mail.
- SRQ 106 – Změnit login.
System umožní přihlášenému uživateli změnit login.

Další požadavky se budou vztahovat pouze na přihlášeného uživatele.

FRQ2 – Téma/podtéma.

- SRQ 201 - Vytvořit téma/podtéma.
System umožní uživateli vytvořit nové téma nebo podtéma v rámci vybraného tématu. Pro vytváření je nutné uvést název a popis. U podtématu nemůže být vytvořené další podtéma.
- SRQ 202 - Odstranit téma/podtéma.
System umožní uživateli odstranit vytvořené téma nebo podtéma. Při odstranění se automaticky smažou i otázky v rámci tématu nebo podtématu.
- SRQ 203 - Změnit název tématu/podtématu.
System umožní uživateli změnit existující název tématu/podtématu.
- SRQ 204 – Zobrazit téma/podtéma.
System umožní uživateli zobrazit vytvořené téma/podtéma a otázky které se tam nachází.

FRQ3 – Otázky/odpovědi.

- SRQ 301 - Vytvořit otázku.
System umožní uživateli vytvořit otázku v rámci tématu/podtématu. Otázky mohou být ve formě textu nebo obrázku. Otázka může být i bez odpovědí.

- SRQ 302 – Odstranit otázku.
System umožní uživateli odstranit vytvořenou otázku. Při odstranění otázky se automaticky odstraní i všechny odpovědi. Pokud otázka se nachází v testu, pak při odstranění se tato otázka automaticky odstraní z testu.
- SRQ 303 – Odstranit nahlášenou otázku.
System umožní administrátoru odstranit veřejně dostupnou otázku, pokud tato otázka byla nahlášena uživatelem, viz SRQ 501 – Nahlásit příspěvek.
- SRQ 304 – Zobrazit otázku.
System umožní uživateli zobrazit vytvořenou otázku včetně všech odpovědí.
- SRQ 305 – Zobrazit seznam otázek.
System umožní uživateli zobrazit seznam všech vytvořených otázek v rámci vybraného tématu.
- SRQ 306 - Vytvořit odpověď.
System umožní uživateli vytvořit odpověď k vybrané otázce. Odpovědi mohou být ve formě textu nebo obrázku. Při vytváření odpovědi uživatel může označit odpověď jako správnou.
- SRQ 307 – Odstranit odpověď.
System umožní uživateli odstranit vytvořenou odpověď.

FRQ4 – Testy.

- SRQ 401 – Vytvořit test.
System umožní uživateli vytvořit test. Pro vytvoření testu je nutné uvést jeho jméno.
- SRQ 402 – Odstranit test.
System umožní uživateli odstranit test ze seznamu testů.
- SRQ 403 – Zobrazit test.
System umožní uživateli zobrazit test a otázky které se v něm nachází.
- SRQ 404 – Zobrazit seznam vytvořených testů.
System umožní uživateli zobrazit všechny testy, které jim byly vytvořené.
- SRQ 405 – Přidat otázky.
System umožní uživateli přidávat vytvořené otázky s odpověďmi do vytvořeného testu.
- SRQ 406 – Odebrat otázku.
System umožní uživateli odstranit otázku z vybraného testu.

- SRQ 407 – Spustit test.
 Systém umožní uživateli spustit vytvořený test. Při spuštění testu je nutné uvést čas, za který bude test automaticky ukončen a způsob kontroly odpovědí:
 - Kontrolovat odpovědi po každé otázce.
 - Zkontrolovat všechny odpovědi na konci testu.
- SRQ 408 – Ukončit test.
 Systém umožní uživateli ukončit spuštěný test, než nevyprší čas, který byl nastavený uživatelem při spuštění testu.
- SRQ 409 - Spustit test pouze špatně zodpovězených otázek.
 Systém umožní uživateli po dokončení testu spustit test špatně zodpovězených otázek.
- SRQ 410 – Sdílet test.
 Systém umožní uživateli sdílet svůj test s jinými uživateli. Uživatel, kterému byl test sdílen nemůže test editovat.
- SRQ 411 – Zveřejnit test.
 Systém umožní uživateli zveřejnit svůj test. Zveřejněný test se zobrazí ostatním uživatelům v menu “Veřejné testy”. Pro zveřejnění testu uživatel musí vybrat kategorii ze seznamu.
- SRQ 412 – Nahlédnout výsledky testu.
 Systém umožní uživateli po ukončení testu se podívat na počet špatně a správně zodpovězených otázek a čas za který byl test vyplněn.
- SRQ 413 – Nastavit časovač na test.
 Systém umožní uživateli nastavit časovač při spuštění testu. Po vypršení času test bude ukončen a zobrazí se výsledky.
- SRQ 414 – Ohodnotit test.
 Systém umožní uživateli ohodnotit veřejně dostupný test ve formě „líbí se“, „nelíbí se“. Test, který získá více než 100 špatných hodnocení bude odstraněn z veřejně dostupných testů. Uživatel může ohodnotit stejný test pouze jednou.
- SRQ 415 – Přidat veřejný test do záložek.
 Systém umožní uživateli přidat veřejně dostupný test do záložek. Test se pak zobrazí v seznamu uložených testů. Tento test uživatel nebude moct editovat.
- SRQ 416 – Zobrazit seznam přidanych do záložek testů.

System umožní uživateli zobrazit seznam přidanych do záložek testů. Pod pojmem „přidané do záložek testy“ se rozumí testy, které byly uloženy uživatelem z veřejných testů.

- SRQ 417 – Zobrazit seznam sdílených s uživatelem testů.

System umožní uživateli zobrazit testy, které mu sdíleli jiné uživatelé. Tyto testy uživatel nemůže test editovat.

- SRQ 418 – Uložit test.

System umožní uživateli uložit sdílený test nebo test přidany uživatelem do záložek do svých testů. Po přidání, tyto testy mohou být editované uživatelem.

FRQ5 – Reporty.

- SRQ 501 – Nahlásit příspěvek.

System umožní uživateli nahlásit veřejně dostupnou otázku. Pro nahlášení je nutné vybrat jeden z následujících důvodů: porušení práva duševního vlastnictví, nahota nebo sexuální aktivita, nenávistné slovní projevy nebo jiný důvod.

- SRQ 502 – Zobrazit reporty.

System umožní administrátoru prohlédnout si nahlášené příspěvky.

2.2.2. Kvalitativní požadavky a omezení

V této podkapitole budou uvedené omezení kladená na systém nebo proces vývoje.

NFR 1. RESTful API.

Aplikace bude vystavovat RESTful API umožňující veškeré CRUD operace. Existence REST API bude umožňovat komunikaci klientské a serverové části aplikace.

NFR 2. Podporované prohlížeče.

Aplikace bude podporována v nejnovějších verzích následujících prohlížečů: Internet Explorer, Google Chrome, Mozilla Firefox, Safari.

NFR 3. Doba zpracování požadavků.

Aplikace bude schopná zpracovat uživatelské požadavky do 2 sekund.

NFR 4. Bezpečnost.

Aplikace neumožní přístup k jednotlivým stránkám a informacím bez úspěšného přihlášení uživatele.

Aplikace musí být zabezpečena proti vkládání skriptů a HTML značek, které by mohly poškodit obsah webu nebo dokonce zničit záznamy v datovém úložišti.

NFR 5. Rozšiřitelnost.

Aplikace bude umožňovat snadné rozšíření o další funkcionality.

NFR 6. Dokumentace.

Aplikace bude poskytovat kvalitní dokumentace popisující nabízené funkce a zacházení s nimi.

NFR 7. Lokalizace.

Aplikace bude přizpůsobena pro použití ve více jazycích (čeština, angličtina, ruština).

2.3. Případy užití

V této podkapitole definování aktéři a případy užití.

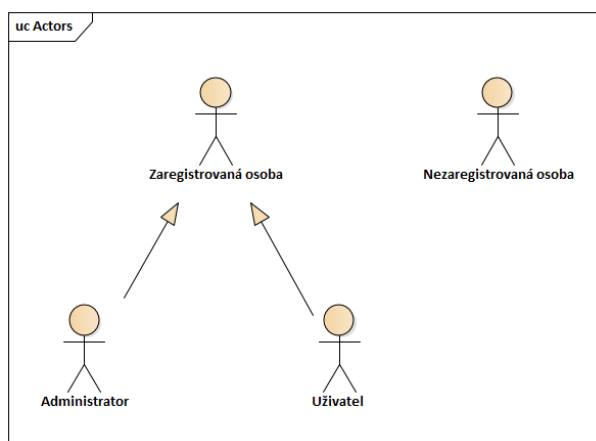
V UML diagramy případů užití (use-case diagramy) modelují chování systému a pomáhají zachytit požadavky na systém [3].

2.3.1. Uživatelské role

V systému jsou definované následující uživatelské role:

1. Zaregistrovaná osoba. Osoba, která již zaregistrována v systému. Od této role se dědí dvě další.
 - 1.1. Administrátor. Zaregistrovaná osoba, která je správcem veřejně dostupných otázek.
 - 1.2. Uživatel. Zaregistrovaná osoba, která má přístup ke všem funkcionalitám systému, kromě správy veřejně dostupných otázek.
2. Nezaregistrovaná osoba. Osoba, která není zaregistrována v systému. Má právo se pouze zaregistrovat.

Znázornění uživatelských rolí je na *obrázku 4 Quizizz*.



Obr. 4. Uživatelské role

2.3.2. Diagram případů užití

Cíl modelování diagramu – najít hranice systému [4].

Samotné případy užití jsou detailně popsány v *příloze A. Případy užití*. Pro přehlednost jsou všechny případy užití jsou rozdělené do pěti skupin: přihlášení, téma/podtéma, otázky/odpovědi, testy, reporty.

Případy užití mají následující atributy:

- Jednoznačný identifikátor/číslo
- Název
- Příslušný systémový požadavek
- Možnosti spuštění
- Hlavní scénář⁶
- Alternativní scénář⁷ (pokud existuje)
- Výjimečný scénář⁸ (pokud existuje)

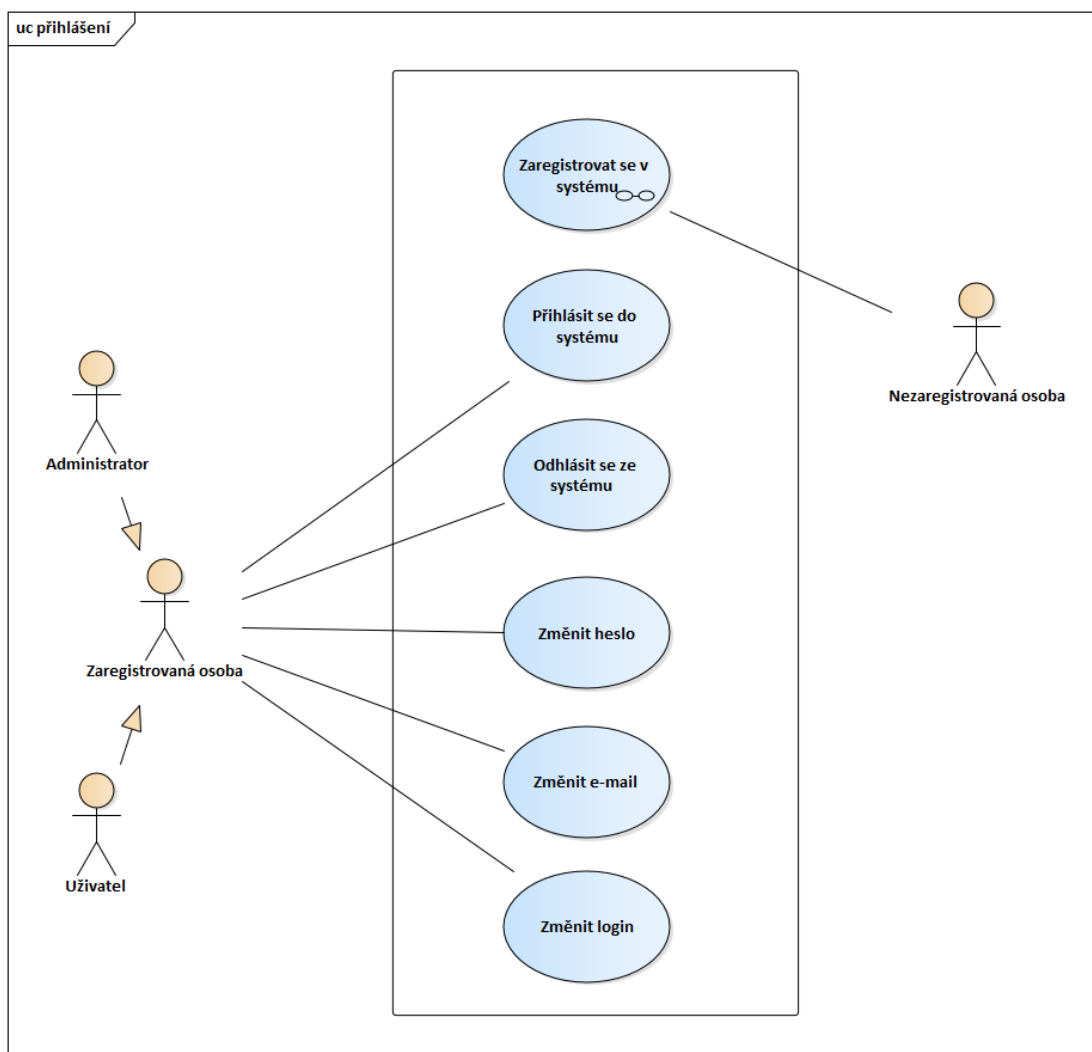
Případy užití – Přihlášení.

Na *obrázku 5. Diagram případů užití – Přihlášení* lze vidět případy užití pro skupinu přihlášení a uživatelské role pro které jsou tyto případy užití určené.

⁶ Hlavní scénář – popisuje kroky případu užití, pokud vše jde, jak má [4].

⁷ Alternativní scénář – zachycuje odchylky od hlavního scénáře [4].

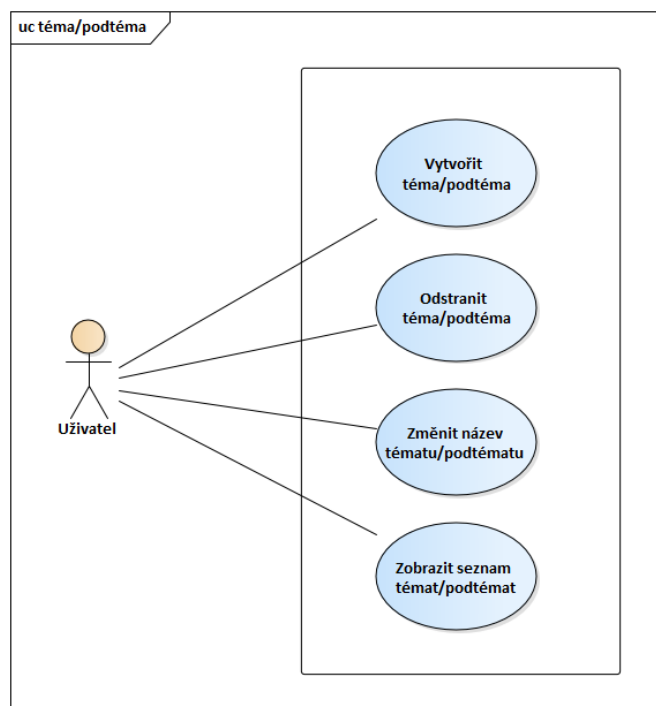
⁸ Výjimečný scénář – popisuje varianty, kdy se nepodaří dosáhnout cíle případu užití [4].



Obr. 5. Diagram případů užití – Přihlášení

Případy užití – Téma/podtéma.

Na obrázku 6. Diagram případů užití – Téma/podtéma lze vidět případy užití pro skupinu téma/podtéma a uživatelské role pro které jsou tyto případy užití určené.



Obr. 6. Diagram případů užití – Téma/podtéma

Případy užití – Otázky/odpovědi.

Na obrázku 7. Diagram případů užití – Otázky/odpovědi lze vidět případy užití pro skupinu otázky/odpovědi a uživatelské role pro které jsou tyto případy užití určeny.

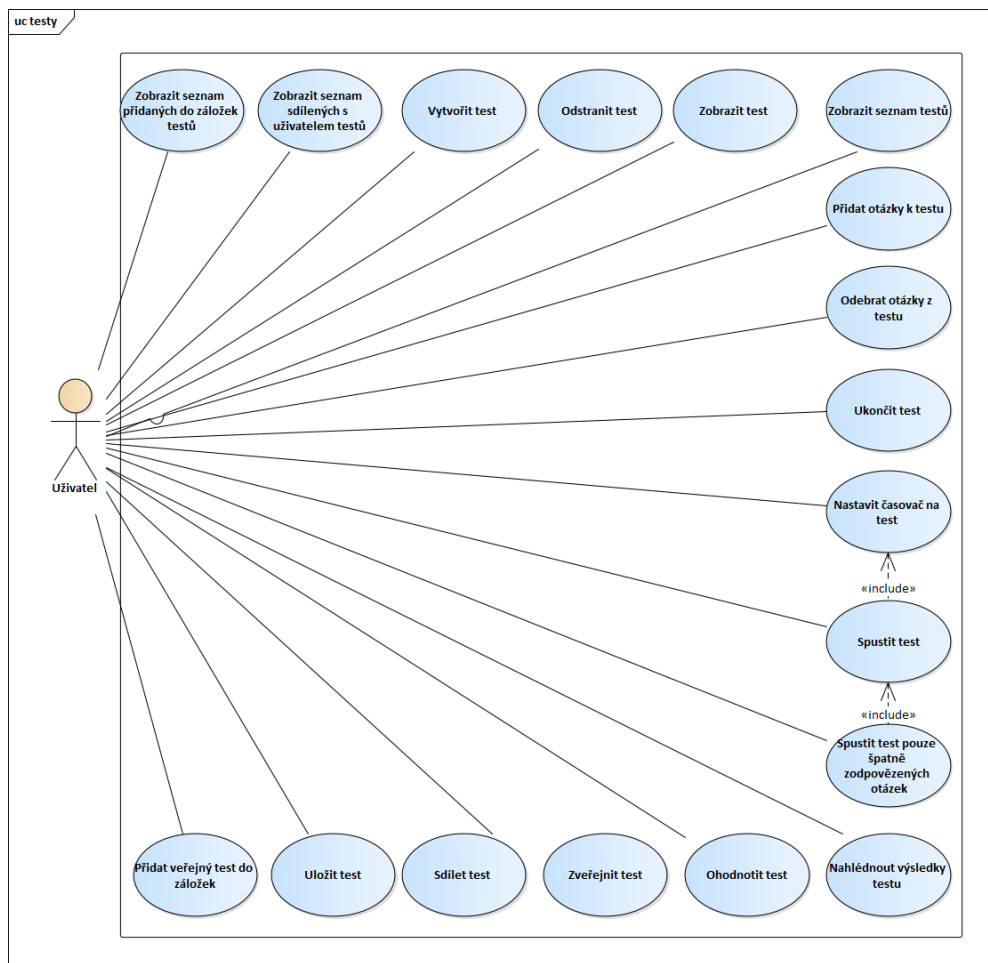


Obr. 7. Diagram případů užití – Otázky/odpovědi

Případy užití – Testy.

Na obrázku 8. Diagram případů užití – Testy lze vidět případy užití pro skupinu testy a uživatelské role pro které jsou tyto případy užití určeny.

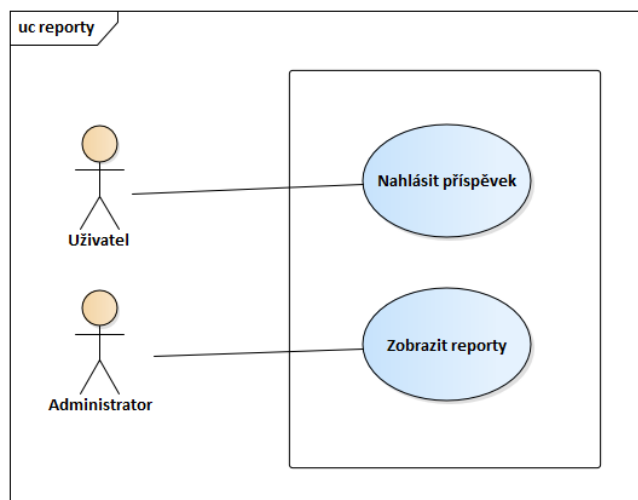
Případ užití napojený pomocí vazby <<include>> znamená že „includovaný“ případ užití je vyvolán vždy, když dojde k vyvolání „nadřazeného“ případu užití [4].



Obr. 8. Diagram případů užití – Testy

Případy užití – Reporty.

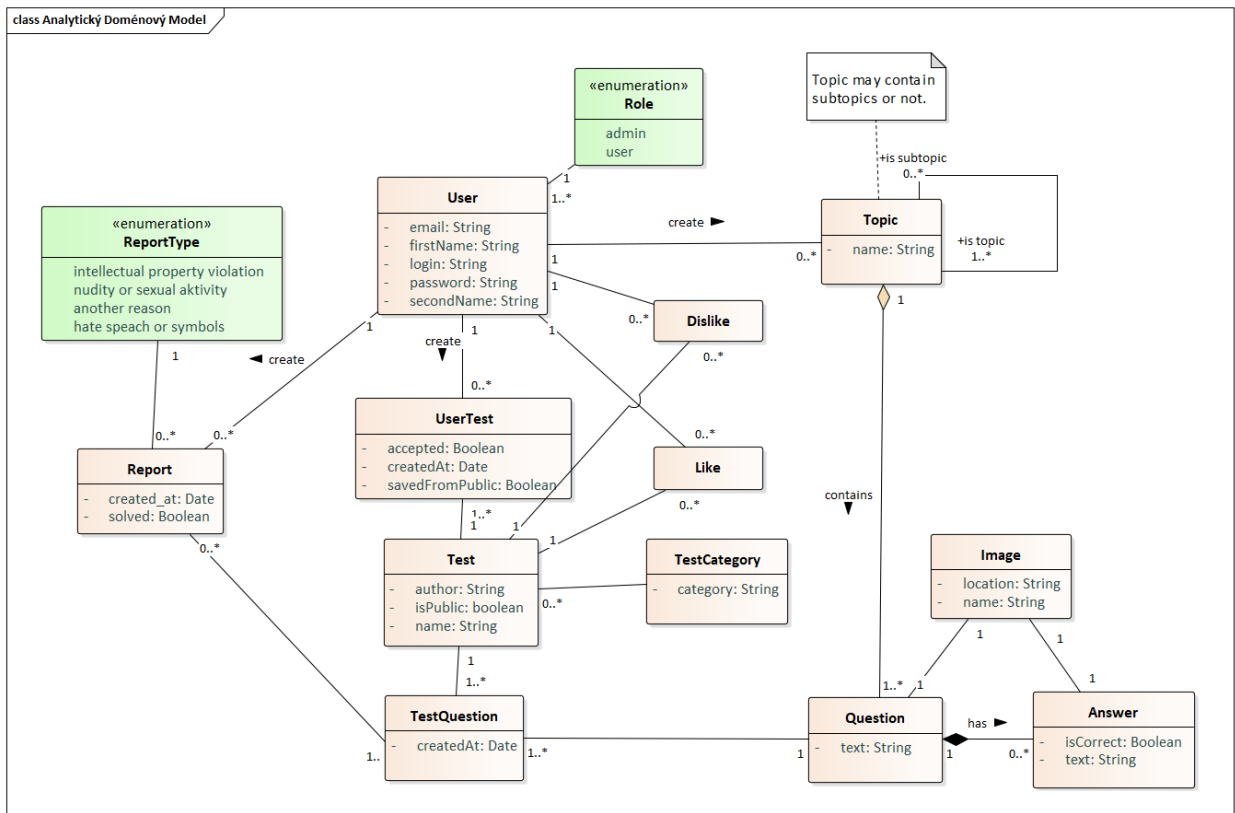
Na obrázku 9. Diagram případů užití – Reporty lze vidět případy užití pro skupinu testy a uživatelské role pro které jsou tyto případy užití určeny.



Obr. 9. Diagram případů užití – Reporty

2.4. Analytický doménový model tříd

Na základě vytvořených případů užití byl navržen analytický doménový model tříd. Model pomáhá definovat entity, jejich atributy a vztahy mezi nimi. Popisuje data, se kterými se bude pracovat v rámci tohoto systému. Analytický doménový model je znázorněn na obrázku 10. Analytický doménový model.



Obr. 10. Analytický doménový model

Seznam entit:

- *User* – uživatel systému. Buď to administrátor nebo obyčejný uživatel systému
- *Topic* – reprezentuje téma nebo podtéma. Vztah – rekurentní relace⁹ neboli reflexivní asociace
- *TestQuestion* – asociační třída¹⁰, která zprostředkovává vztah mezi dvěma entitami – *Test* a *Question*
- *Test* – reprezentuje test. Jelikož uživatel může vytvořit více testů a test může vlastnit více uživatelů (například po sdílení testu), bylo zapotřebí vytvořit asociační třídu – *UserTest*

⁹ Rekurentní relace – entita má vztah sama na sebe, speciální případ binárního vztahu [29].

¹⁰ Asociační třída dosgvoluje přiřadit atributy, operace a další rysy k asociační vazbě, která řeší vztah mezi třídami typu M:N.

- *TestCategory* – Reprezentuje kategorii testu
- *UserTest* – asociační třída, která zprostředkovává vztah mezi dvěma entitami Test a User
- *Question* – reprezentuje otázku. Jelikož v testu se může nacházet více otázek a otázka se může nacházet ve více testech, bylo zapotřebí vytvořit asociační třídu – TestQuestion
- *Answer* – reprezentuje odpověď na otázku
- *Image* – reprezentuje obrázek. Obsahovat obrázek může jak otázka, tak i odpověď.
- *Report* – reprezentuje nahlášení příspěvku uživatelem
- *Like* – asociační třída mezi entitami Test a User. Slouží k uchování dat ohledně kladného hodnocení testů
- *Dislike* – asociační třída mezi entitami Test a User. Slouží k uchování dat ohledně záporného hodnocení testů

2.5. Shrnutí kapitoly

V této kapitole byly uvedeny existující řešení a provedena jejich analýza. Na základě této analýzy byly definovány požadavky na systém a navrženy případy užití. Na konci byl vytvořen analytický doménový model tříd.

3. Volba technologií

Cílem této kapitoly je poskytnout základní přehled technologií používaných pro tvorbu webových aplikací a jejich porovnání. Pak, na základě této analýzy, vybrat nejvhodnější technologie pro aplikaci. Při výběru se brali v potaz požadavky na systém, cíl práce, jednoduchost řešení, osobní zkušenost s technologiemi a kvalita dokumentace.

3.1. Serverová část

Vývoj webových aplikací je v zásadě rozdělen na klientskou a serverovou část. Obě mají svoji příslušnou sadu nástrojů a technologií. Tato podkapitola se zaměřuje na programovací jazyky používané na straně serveru.

3.1.1. Výběr programovacího jazyka

Mezi vývojáři podle StackOverflow [4] nejpobulárnější jsou následující programovací jazyky a prostředí:

- Python
- Java
- Node.js
- C#
- PHP

Seznam je zobrazen sestupně podle popularity. Zdůrazním, že seznam není úplný a obsahuje pouze prvních 5 jazyků používaných na straně serveru.

Python.

Je vysokoúrovňový, interpretovaný programovací jazyk. Podporuje objektové, procedurální i funkcionální programování. Slouží především ke skriptování, vědeckým výpočtům, vývoji webových aplikací.

Pokud jde o vývoj webových aplikací obvykle se používají frameworky, nejpobulárnější jsou Django, Flask, Web2Py a TurboGears.

Hlavní výhody použití jazyka Python jsou: jednoduchá syntaxe a tím pádem snadná čitelnost a údržba kódu, velká uživatelská základna a dostupnost velkého množství knihoven a nástrojů pro vývoj.

Nevýhody jsou: jelikož Python je interpretovaný jazyk, většina chyb se ukáže až za běhu programu, na rozdíl od kompilovaných jazyků, kde se mnoho chyb objeví při kompilaci [5]. Navíc dynamické interpretované jazyky jako Python jsou pomalejší.

Java.

Obecný, vyšší, imperativní (procedurální) a objektově orientovaný programovací jazyk [6]. Java je univerzální jazyk, vhodný jak pro vývoj desktopových aplikací, tak i pro vývoji mobilních a webových aplikací.

Pokud jde o vývoj webových aplikací obvykle se používají frameworky, nejpobulárnější jsou Spring Boot, Blade a Play.

Hlavní výhody použití jazyka Java jsou: osobní zkušenost s vývojem webových aplikací, dostupnost velkého množství knihoven a nástrojů pro vývoj, velice silné zabezpečení a existence skvělého frameworku – Spring Boot.

Mezi nevýhody patří delší doba implementace než v Python.

Node.js.

Asynchronní JavaScriptový runtime (prostředí pro provádění scriptu). Kód je psaný v jazyce JavaScript. Node.js je postavený na enginu Chrome V8, takže základ tohoto prostředí je stejný jako ve webovém prohlížeči Google Chrome¹¹. Primární účel Node.js je tvorba webových serverů.

Hlavní výhody použití Node.js jsou: vývoji klientské a serverové části ve stejném jazyce, vysoká výkonost a vysoká škálovatelnost, tzn. schopnost obsloužit mnoho připojených klientů naráz [7].

Mezi nevýhody patří to, že prostředí není konzistentní. API se často mění a změny občas nejsou zpětně kompatibilní.

C#.

Moderní, objektově orientovaný a typově bezpečný programovací jazyk. Programy v jazyce C# běží na .NET, virtuálním systému spouštění s názvem Common Language Runtime (CLR) a sadě knihoven tříd [8]. C# lze použít k vytvoření téměř čehokoli, ale je většinou se používá pro vytváření desktopových a web aplikací a her pro Windows.

Hlavní výhody použití C# jsou: dostupnost velkého množství knihoven a nástrojů pro vývoj a podobnost syntaxe v C# a v Javě.

Mezi nevýhody patří to, že C# je interní součástí .NET frameworku, takže server, na kterém je aplikace spuštěna, musí být založen na Windows.

¹¹ V8 je název JavaScript enginu, který bere Javascriptový kód a spouští jej při procházení prohlížečem Chrome [40].

PHP.

PHP je skriptovací jazyk a výkonný nástroj pro vytváření dynamických a interaktivních webových stránek [9]. Pokud jde o vývoj obvykle se používají frameworky Nette, Symfony a Laravel.

Hlavní výhody použití PHP jsou: velká komunita, kratší doba implementace než například v Java a jednoduchá integrace do webových serverů.

Mezi nevýhody patří netypovost jazyka [10], i když to občas dává určitou svobodu.

Zvolené řešení.

Pro serverovou část aplikace byl zvolen jazyk Java.

Jazyk PHP nebyl zvolen především kvůli komplikované syntaxi (z osobního pohledu). Jazyk C# nebyl zvolen kvůli malé zkušenosti s tímto jazykem, stejně tak byl vyloučen i Node.js.

Zůstaly tedy Java a Python. Pro Javu nakonec rozhodla rychlost výsledné aplikace, osobní zkušenost s vývojem aplikací v tomto jazyce a vysoká kvalita dokumentace.

3.1.2. Výběr frameworku

Existuje celá řada frameworků pro vývoj webových aplikací v programovacím jazyce Java. Podle StackOverflow [4] nejpopulárnějším je Spring. S ohledem na tuto statistiku i na osobní dlouhodobou zkušenost s tímto frameworkem bylo rozhodnuto použít Spring Boot.

Spring Boot.

Spring Boot je open-source aplikační framework napsaný v jazyce Java. V podstatě to je nadstavba Spring Framework. Implementuje tedy veškeré funkce Springu, umožňuje nám navíc vytvořit samostatnou aplikaci s minimálními nebo nulovými konfiguracemi. Spring Boot automaticky konfiguruje třídy na základě požadavků. Je lepší jej použít, pokud chceme vyvíjet jednoduchou Spring-based aplikaci nebo RESTful služby [11].

Framework umožňuje snadné vytvoření a ovládání koncových bodů, zabezpečení aplikace a usnadňuje práci s datovou vrstvou.

3.2. Klientská část

3.2.1. Výběr programovacího jazyku

Hlavními nástroji podle StackOverflow [4], které se používají při vývoji klientské části jsou: Javascript, HTML a CSS a TypeScript.

Javascript.

Skriptovací interpretovaný jazyk, který se obvykle používá pro vytváření interaktivních webových stránek na straně klienta.

Výhodou Javascriptu je to, že nepotřebuje překladač, protože se spouští ve webovém prohlížeči.

Mezi nevýhody podle mého názoru patří to, že Javascript není objektově orientovaný jazyk a neposkytuje typovou kontrolu.

TypeScript.

Jazyk založený na JavaScriptu (má všechny funkce JavaScriptu), který ho rozšiřuje o statické typování¹², třídy, rozhraní a další věci známé z OOP [12].

Hlavní výhodou použití TypeScriptu podle mého názoru je to, že TypeScript je objektově orientovaný programovací jazyk.

Mezi nevýhody patří to, že TypeScript vyžaduje kompilátor TypeScript k převodu do souboru JavaScriptu.

Zvolené řešení.

S ohledem na předchozí zkušenosti a popularitu jazyku pro vývoj klientské části aplikace byl zvolen Javascript.

3.2.2. Výběr frameworku

Z důvodu nedostatku zkušeností s frameworky pro vytváření uživatelských rozhraní, při výběru frameworků byla brána v úvahu jejich popularita.

Podle [13] nejpopulárnějšími frameworky pro tvorbu klientské části aplikace v JavaScriptu jsou: React, Angular, a Vue.js.

React.

React je JavaScript knihovna pro vytváření uživatelských rozhraní [14]. Je k dispozici od roku 2013 a za tu dobu se komunita rozrostla. Navíc je stále podporována samotným Facebookem, jehož vývojáři neustále pracují na dalším rozvoji.

V React se tvoří struktura pomocí tzv. komponent. Tyto komponenty je možné opakovaně použít, a to jak v rámci jednoho projektu, tak i pro oddělené webové aplikace nebo webové stránky.

¹² Statické typování vyžaduje definovat typ proměnné a tento typ je dále neměnný.

Hlavní výhodou React je využívání virtuálních objektových modelů dokumentu – DOMů. Virtuální DOM zjišťuje, zda došlo ke změně a následně obnoví v klasickém DOMu pouze ty komponenty, které se změnily. Další výhodou je znovupoužitelnost komponent a možnost psaní komponent jako funkce místo tříd.

Hlavní nevýhodou je nemožnost použití komponent napsaných jako funkce v třídových komponentách.

Angular, AngularJS.

AngularJS je předchůdce frameworku Angular. Tyto 2 frameworky nejsou navzájem kompatibilní. Jelikož AngularJs již není podporován společností Google, dále se bude zabývat pouze frameworkem Angular.

Angular je framework od společnosti Google určený k tvorbě dynamických webů a webových aplikací. Angular stejně jako React tvoří svoji strukturu pomocí komponent. Základem komponenty je programová třída. Angular využívá principů objektově orientovaného programování.

Výhodou Angular je to, že architektura je postavena na komponentách a framework obsahuje všechny potřebné nástroje pro vývoj škálovatelných webových aplikací.

Nevýhodou Angular je to, že tento framework nemá virtuální DOM, čímž se výrazně zvyšuje doba implementace. Další nevýhodou je omezení svobody a flexibility při výběru knihoven (na rozdíl od React) a velká velikost projektu.

VueJS.

VueJS je framework pro vytváření webových uživatelských rozhraní pomocí jazyka JavaScript. Stejně jako React a Angular, tvoří svoji strukturu pomocí komponent. Jádro knihovny je stejně jako v React zaměřeno pouze na zobrazovací vrstvu (view).

Výhodou VueJS je využívání virtuálních objektových modelů dokumentu stejně jako v React a malá velikost frameworku.

Hlavní nevýhodou je to, že má zatím nejmenší komunitu podpory.

Zvolené řešení.

Pro klientskou část aplikace byla zvolena knihovna React.

Angular nebyl zvolen především protože nemá virtuální DOM, který zrychluje celý proces vývoje. I když VueJs má to nejlepší od React a Angular nebyl zvolen proto, že má menší uživatelskou základnu a je méně populární než React.

3.3. Databáze

Před výběrem konkrétní databáze je potřeba nejdřív vybrat vhodný databázový systém. V dnešní době existují dva dominantní databázové systémy:

- Relační databáze
- NoSQL databáze

Základní rozdíly jsou uvedené v *tabulce 2. Rozdíly mezi NoSQL a SQL databázemi* [15].

	NOSQL	RELAČNÍ
UKLÁDÁNÍ DAT	- Dokumenty JSON - Páry klíč/hodnota - Grafy	- Tabulky
SCHÉMA	Dynamická nebo flexibilní	Statická
ŠKÁLOVATELNOST	Horizontální	Vertikální
DATA	Nové vlastnosti lze přidávat za běhu	Přidání nové vlastnosti může vyžadovat alternativní schémata

Tab. 2. Rozdíly mezi NoSQL a SQL databázemi

Na základě určených systémových požadavků je možné říct, že každý záznam v databázi bude mít stejné vlastnosti a systém nebude potřebovat zpracovávat velké množství dat. Proto bylo rozhodnuto použít relační databázový systém.

Podle průzkumu Stack Overflow [4] nejpopulárnější databázové řešení jsou: MySQL, PostgreSQL a SQLite.

S ohledem na předchozí zkušenosti a širokou znalostní báze byla vybrána open source relační databáze – PostgreSQL. Mezi její další výhody patří jednoduchá rozšiřitelnost, multiplatformita a stabilita.

Zatímco aplikace je nastavena pro přístup k PostgreSQL databázi, testy používají in-memory H2 databázi. H2 je relativně nová relační databáze, napsaná v Javě.

In-memory databáze je mnohem rychlejší na vytvoření, takže jsou s ní testy mnohem rychlejší. Po každém testu pak můžeme databázi zahodit a vytvořit novou [16].

3.4. Hostiny

V této podkapitole byla provedena analýza dostupných hostingů. Na základě této analýzy byl vybrán vhodný pro nasazení hosting.

K dispozici existuje celá řada již existujících řešení, které usnadňují celý proces nasazení, například Heroku, Netlify nebo OpenShift.

Při výběru hostingu byly brány v úvahu pouze ty, které dovolují nasazení zadarmo.

3.4.1. Heroku

Heroku je PaaS služba od společnosti Salesforce pro nasazování a spouštění Full Stack aplikací. Podporuje jazyky Ruby, Python, Java, PHP, Go, Scala a platformu Node.js. Jako výchozí databáze nabízí PostgreSQL a Redis.

Výhody:

- Jednoduchý začátek, není potřeba řešit infrastrukturu
- Dovoluje nasadit jak serverovou, tak i klientskou část
- Skvělá dokumentace na hlavních stránkách
- Integrace s GitHub

Nevýhody:

- Aplikační server se po dobu nečinnosti 30 minut vypne a chvíli trvá, než se zapne
- Nelze zdarma přidat vlastní doménu

3.4.2. OpenShift

OpenShift je PaaS služba od společnosti Red Hat. Podporuje jazyky Java, Python, Perl, Ruby, PHP a platformách .NET Core a Node.js. Nabízí MySQL, PostgreSQL, Redis, MariaDB a MongoDB.

Výhody:

- Dovoluje nasadit jak serverovou, tak i klientskou část
- Integrace s GitHub

Nevýhody:

- Aplikační server se po dobu nečinnosti 30 minut vypne a chvíli trvá, než se zapne
- Vypnutí musí zabrat alespoň 18 hodin z každého 72-hodinového intervalu
- Umožňuje mít pouze 3 aplikace pro jeden uživatelský účet

3.4.3. Netlify

Netlify je platforma od společnosti Netlify, která pomáhá vývojářům vytvářet, testovat a nasazovat webové stránky. Podporuje frameworky React, Next.js, Vue.js, Svelte, WordPress a další.

Výhody:

- Dovoluje zobrazení aplikace na vlastní doméně
- Integrace s GitHub

Nevýhody:

- Neumožňuje hostování standardní serverové části

3.4.4. Vyhodnocení

Existuje mnoho dalších řešení kromě výše zmíněných jako například Amazon web service, Firebase, Google App Engine apod. Ale většinou jsou placené nebo nejsou pohodlné pro začátečníky.

Na základě provedené analýzy bylo rozhodnuto použít pro nasazení Heroku. Zaprvé dovoluje nasadit jak serverovou část, tak i klientskou (na rozdíl od Netlify). Výhodou před OpenShift bylo to, že Heroku nemusí být vypnutý 18 hodin z každého 72-hodinového intervalu.

3.5. Shrnutí kapitoly

V této kapitole byla provedena analýza nejčastěji používaných technologií. Pak, na základě této analýzy a softwarové analýzy byly vybrány nejvhodnější technologie pro aplikaci TestQuiz.

Pro serverovou část byla vybrána kombinace – Java a Spring Boot. Pro klientskou část – Javascript a React. Pro manipulaci s daty byla zvolena relační databáze PostgreSQL. Nakonec byl vybrán vhodný hosting pro nasazení – Heroku.

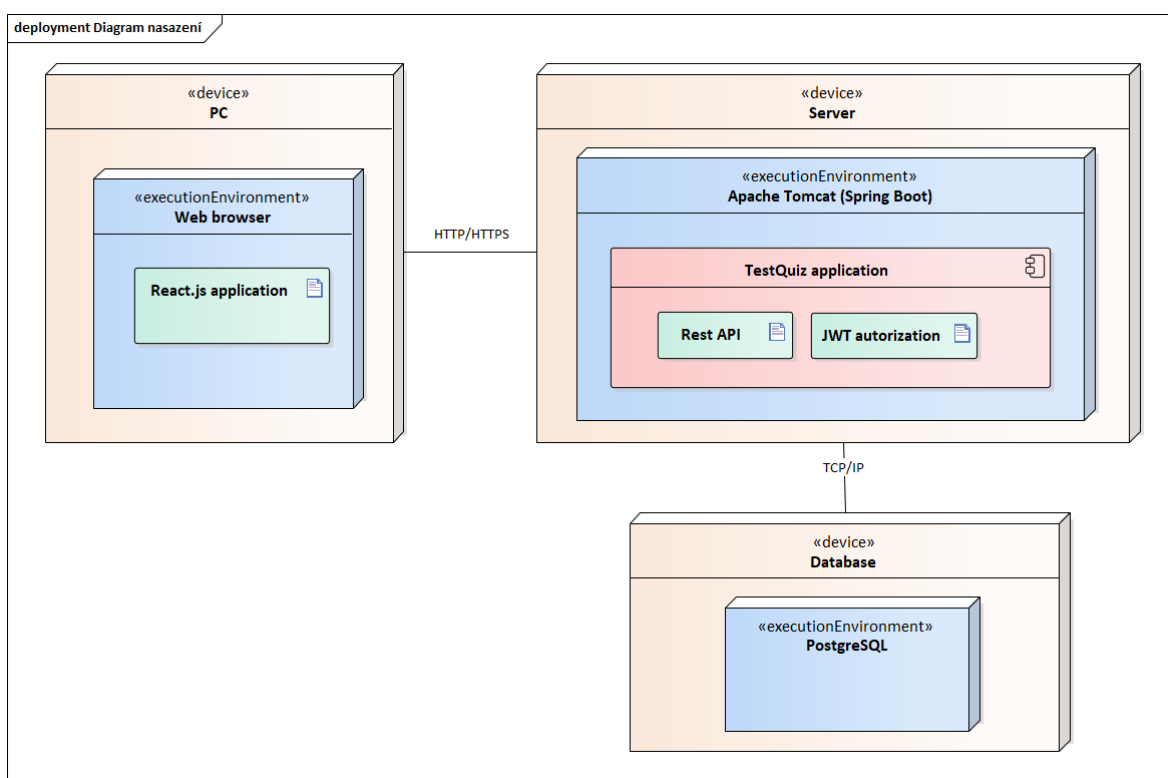
4. Implementace

Tato kapitola se zabývá popisem implementace webové aplikace.

Nejdříve je popsána fyzická architektura aplikace dále jsou uvedeny použité knihovny a nástroje pro vývoj. Následně je uveden způsob zabezpečení aplikace a způsob komunikace mezi klientem a serverem. Posléze je popsán postup nasazení aplikace na cloudovou platformu.

4.1. Fyzická architektura aplikace

Diagram nasazení popisuje hardwarovou architekturu systému a nasazení software na hardware [5]. Diagram nasazení jednotlivých částí aplikace je znázorněn na *obrázku 11. Diagram nasazení.*



Obr. 11. Diagram nasazení

Aplikace se skládá z databáze, klientské části a serverové části. Jak již bylo řečeno v kapitole 3. *Volba technologií*, serverová část je napsána v jazyce Java pomocí frameworku Spring Boot. S klientskou částí, která je v jazyce Javascript a frameworku React, komunikuje serverová část přes REST¹³ API ve formátu JSON¹⁴ (JavaScript Object Notation).

¹³ Representational state transfer je architektura, která umožňuje přistupovat k datům pomocí protokolu HTTP (Hypertext Transfer Protocol) [34].

¹⁴ JavaScript Object Notation je odlehčený formát pro výměnu dat. Je jednoduše čitelný i zapisovatelný člověkem a snadno

4.2. Použité knihovny

Spring Boot Starter Web

Startér pro vytváření webových aplikací, včetně RESTful, pomocí Spring MVC. Používá Tomcat jako výchozí vložený kontejner [6].

Spring Boot Starter Test

Starter pro testování Spring Boot aplikací s knihovnami včetně JUnit Jupiter, Hamcrest a Mockito.

Spring Boot Starter Data JPA

Spring Data JPA, součást větší rodiny Spring Data. Usnadňuje práci s DAO vrstvou, která obstarává manipulaci s daty.

Flyway

Flyway je open-source nástroj pro migraci databází. Upřednostňuje jednoduchost a konvenci před konfigurací. Flyway aktualizuje databázi z jedné verze na další pomocí migrací. Migrace jsou pásne buď v SQL se syntaxí specifickou pro databázi, nebo v Javě pro pokročilé databázové transformace [7].

SpringDoc OpenAPI

Knihovna, která pomáhá automatizovat generování API dokumentace. Tato knihovna funguje tak, že zkoumá aplikaci za běhu, aby odvodila sémantiku API na základě Spring konfigurací, struktury tříd a různých anotací [8].

Největší výhoda knihovny spočívá v tom, že automaticky generuje dokumentaci v rozhraních API formátu JSON/YAML a HTML. Tuto dokumentaci lze doplnit komentáři pomocí anotací swagger-api, což i bylo použito v této práci.

Swagger

Swagger je knihovna pro definici rozhraní podle specifikace OpenAPI. Slouží nejen k vytvoření popisu – pokrývá celý životní cyklus (návrh, dokumentace, testování, nasazení) [9]. Více o použití Swagger knihovny bude popsáno v podkapitole 4.5 *Rozhraní webové služby*.

Java JSON Web Token

Knihovna pro vytváření a ověřování webových tokenů JSON (JWT).

Lombok

Lombok je Java knihovna, která slouží k minimalizaci často použitelného kódu pomocí anotací, čímž snižuje nepřehlednost a zbytečnou rozsáhlost zdrojového kódu.

analyzovatelný i generovatelný strojově [35].

Axios

Axios je jednoduchý HTTP klient pro komunikaci se serverem.

Bootstrap

Bootstrap je knihovna užitečných, opakovaně použitelných základních kódů napsaných v HTML, CSS a JavaScriptu.

Material-UI

Material-UI je jednoduchá a přizpůsobitelná knihovna pro rychlejší vytváření aplikací.

Font Awesome

Font Awesome je sada fontů a ikon založených na CSS. Je to velká knihovna, která zahrnuje téměř šest set padesát ikony, které je možné jednoduše přidávat a upravovat.

i18next

i18next je Javascript knihovna, která slouží pro lokalizaci aplikací. Knihovna umožňuje vytvářet slovníky pro konkrétní jazyky v JSON formátu. Následně používat proměnné z těchto slovníků místo textu.

4.3. Další nástroje

4.3.1. Buildovací nástroj

Maven a Gradle se obecně řadí mezi nejpobulárnějšími nástroji pro správu, řízení a automatizaci buildů ale Gradle oproti Maven má kratší syntaxi, proto je víc čitelný a je přibližně 100krát rychlejší než Maven [23]. Proto pro tento projekt byl zvolen Gradle.

4.3.2. Verzovací nástroj

Pro verzování kódu byl použit nástroj Git a jeho nadstavba GitHub.

Git je distribuovaný systém pro správu verzí [23]. Funguje v podstatě tak, že detekuje změny, které se v projektu provedli. Všechny změny se posílají do repozitáře, jakmile je rozhodnuto udělat aktualizaci. Výhoda je v tom, že Git zachovává i předchozí verze, což dělá vývoji mnohem pohodlnější a bezpečnější.

Pro vývoj celé aplikace bylo založeno 2 Git repozitáře:

TQ-frontend pro klientskou část.

TQ-backend pro serverovou část.

Odkazy na tyto repozitáře lze najít v *příloze D. Seznam odkazů*.

4.3.3. Vývojové prostředí

Serverová část byla implementována ve vývojovém prostředí IntelliJ IDEA od

společností JetBrains. IntelliJ poskytuje bohatou sadu vývojářských nástrojů a vynikající podporu pro framework Spring. Při vývoje hodně pomohlo to, že prostředí detekuje chyby ještě, než se zkompileje projekt. Další pomůckou bylo, že prostředí poskytuje chytré dokončování kódu, což výrazně urychlilo proces vývoje.

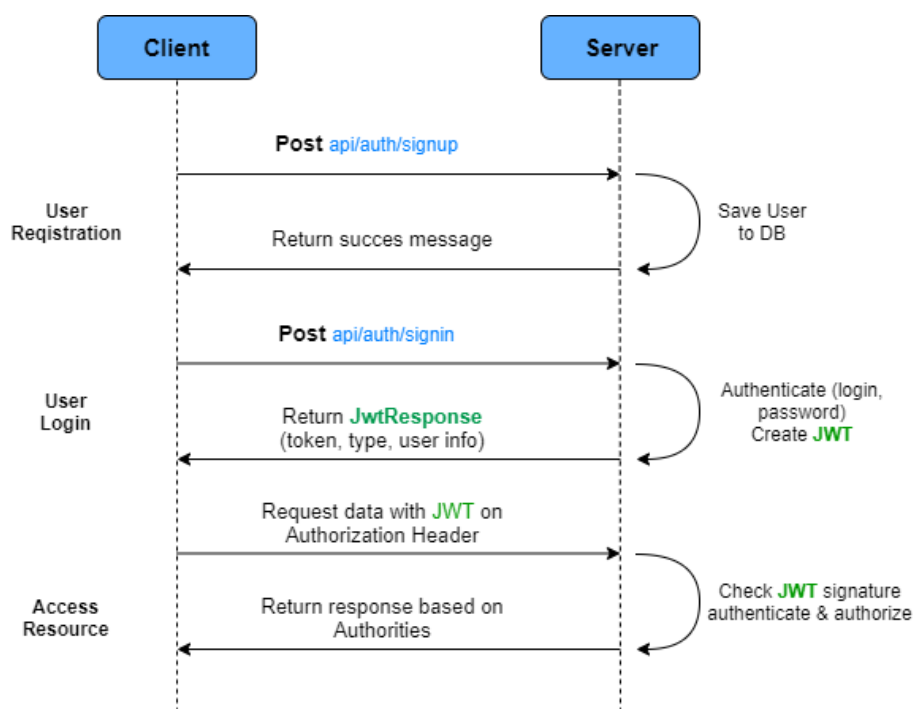
4.4. Zabezpečení

Mezi nejrozšířenější způsoby zabezpečení webových aplikací patří JSON Web Token (JWT), OAuth, a HTTP Basic.

Pro účely zabezpečení tohoto projektu bylo rozhodnuto použít JSON Web Token (JWT). Je jednodušší v implementaci než OAuth, více bezpečný než HTTP Basic.

JWT je otevřený standard, který definuje kompaktní a samostatný způsob pro bezpečný přenos informací mezi stranami ve formátu JSON. Tyto informace lze ověřit a důvěřovat jim, protože jsou digitálně podepsány. JWT lze podepsat pomocí tajného nebo pomocí páru veřejného/soukromého klíče pomocí RSA15 nebo ECDSA16 [10].

Následující diagram ukazuje postup, jak je implementován proces registrace, přihlášení a autorizace uživatele.



Obr. 12. Proces registrace, přihlášení a autorizace uživatele.

¹⁵ Rivest Shamir Adleman – asymetrický šifrovací algoritmus. Nejčastěji se používá k zajištění soukromí a zajištění autenticity digitálních dat [34].

¹⁶ Elliptic Curve Digital Signature Algorithm – je jedním ze složitějších kryptografických šifrovacích algoritmů s veřejným klíčem. [35].

4.5. Rozhraní webové služby

4.5.1. REST

Serverová část komunikuje s klientskou pomocí REST API. To znamená že umožňuje provádět CRUD operace (Create, Read, Update, Delete) nad aplikačními daty pomocí protokolu HTTP. V *tabulce 3. Vztah CRUD operací k HTTP metodám* je uveden vztah CRUD operací k HTTP metodám.

CRUD OPERACE	HTTP METODA
CREATE	POST
READ	GET
UPDATE	PUT
DELETE	DELETE

Tab. 3. Vztah CRUD operací k HTTP metodám

Vstupní a výstupní data v aplikaci se přenáší v JSON formátu. Tento formát byl zvolen kvůli jednoduchému zpracování dat na straně klienta a lepší čitelnosti než formát XML¹⁷.

4.5.2. Swagger

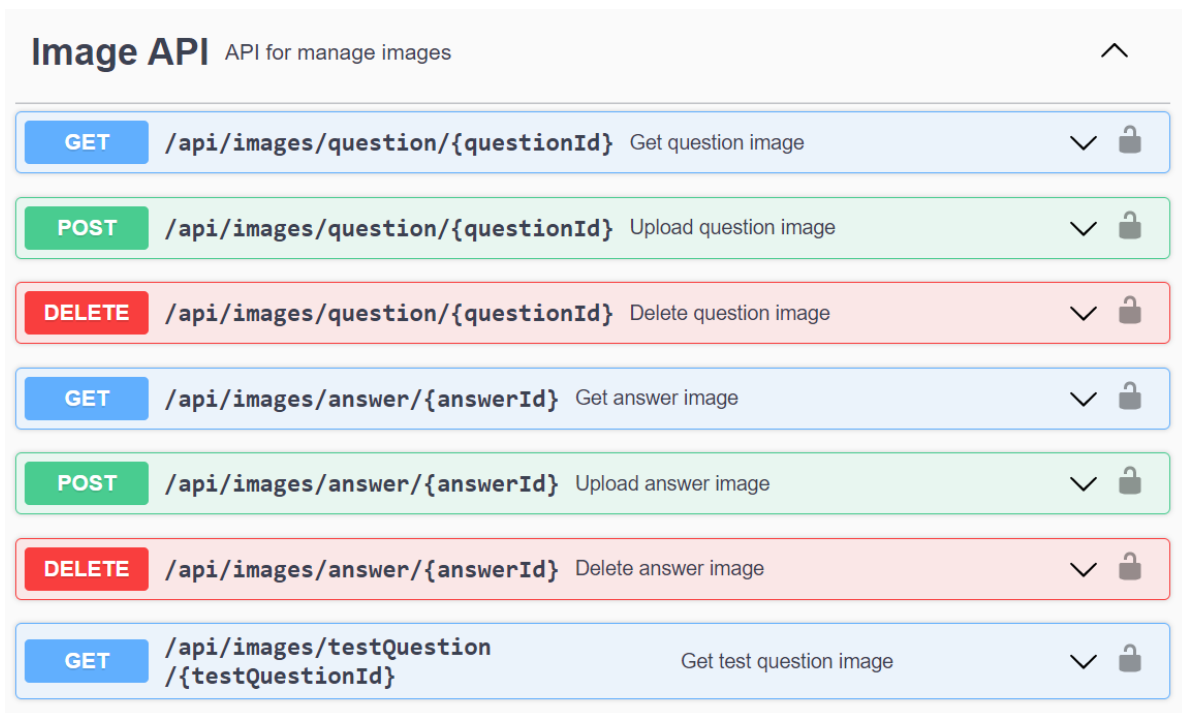
Pro detailnější popis REST API a dat, se kterými se pracuje, byl zvolen Swagger, který zároveň poskytuje možnost přímé interakce se serverem a umožňuje snadno vyzkoušet dané metody posláním HTTP dotazů v rozhraní. K uživatelskému rozhraní dokumentace lze přistupovat z webového prohlížeče.

Při generování Swagger dokumentace se dával důraz převážně na:

- Jednoduché popsání dotazů.
- HTTP stavové kódy.
- Typ HTTP metody.
- URL adresa akce.
- Návrátový typ metody.

Na *obrázku 13. Ukázka části uživatelského rozhraní projektu Swagger UI* znázorněna část REST rozhraní – Image API.

¹⁷ The Extensible Markup Language je jednoduchý textový formát pro reprezentaci strukturovaných informací [38].



Obr. 13. Ukázka části uživatelského rozhraní projektu Swagger UI

Celkově aplikace poskytuje 74 HTTP dotazů které jsou rozdělené na 8 balíčků podle Controllerů, kde se metody nachází:

- Test API
- Answer API
- Report API
- Topic API
- Image API
- Question API
- Auth API
- User API

a 12 vstupních a výstupních datových typů, které odpovídají DTO (Data Transfer Object) objektům:

- User
- Answer
- Image
- Question
- Topic
- UserTest
- Test
- TestCategory
- TestSearchCriteria
- Report

- SignInRequest
- ReportType

Kompletní dokumentace k rozhraní se nachází na adrese Swagger UI, kterou lze najít v příloze D. *Seznam odkazů*.

4.6. Nasazení

Nasazení na produkční server je konečná fáze implementaci aplikace.

4.6.1. Serverová část

Pro nasazení serverové části na Heroku byly provedené následující kroky:

1. Založení účtu.
2. Vytvoření projektu na Heroku.
3. Přípravení aplikace (nastavení Procfile¹⁸).
4. Propojení s databází.
5. Propojení s Git repositářem.
6. Samotné nasazení.
7. Otevření stránky a kontrola nasazení.

Po nasazení serverová část aplikace je dostupná na adrese, kterou lze najít v příloze D. *Seznam odkazů*.

4.6.2. Klientská část

Pro nasazení klientské části na Heroku byly provedené následující kroky:

1. Vytvoření projektu na Heroku.
2. Přípravení aplikace (startovací scripty).
3. Propojení s Git repositářem.
4. Samotné nasazení.
5. Otevření stránky a kontrola nasazení.

Po nasazení klientská část aplikace je dostupná na adrese, kterou lze najít v příloze D. *Seznam odkazů*.

V příloze E. *Snímky uživatelského rozhraní* lze najít náhledy webového rozhraní aplikace TestQuiz.

4.7. Shrnutí kapitoly

V této kapitole byla popsána a ilustrována fyzická architektura aplikace, dále byly

¹⁸ Procfile – textový soubor v kořenovém adresáři aplikace, který deklaruje, jaký příkaz by měl být proveden ke spuštění aplikace.

uvedeny použité knihovny a nástroje pro vývoj serverové a klientské části aplikace. Také byl popsán způsob zabezpečení pomocí JWT tokenu a způsob komunikace mezi dvěma částmi aplikace. Dále byl uveden popis poskytovaného webového rozhraní včetně nástrojů pro popis tohoto rozhraní. Nakonec byl uveden postup nasazení aplikace na cloudovou platformu. Teď je aplikace je připravena k testování.

5. Testování

Tato kapitola se zabývá testováním webové aplikace. Nejdříve je popsáno takzvané white box testování, které zahrnuje jednotkové a integrační testy. Pak je popsáno black box testování, které zahrnuje systémové a uživatelské testování.

Cíl testování – najít defekty systému, než se systém začne používat.

Všechny testy v aplikaci byly rozdělené na 2 kategorie podle znalosti vnitřní struktury:

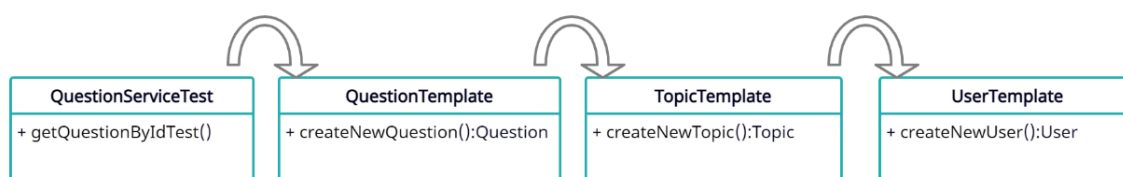
- White box. Testovací scénáře vytváříme na základě známé vnitřní struktury systému.
- Black box. Testovací scénáře vytváříme na základě analýzy specifikace nebo dokumentace funkcionality systému, neznáme vnitřní strukturu systému [11].

5.1. White-box testování

Testy využívají H2 in-memory databázi. In-memory technologie znamená, že se zpracování dat koná přímo v operační paměti. H2 umožňuje dostatečně rychle vytvořit čistou databázi a po vykonání testu rychle ji vymazat.

Pro minimalizaci často použitelného kódu, snadnější přípravu a organizaci testovacích dat byly vytvořené takzvané šablony. Šablony jsou Java třídy, které jsou příslušné entitám v databázi a připravují testovací data pro každý test.

Obrázek 14. Použití šablony ilustruje příklad použití šablon. Podle návrhu aplikace, pro vytvoření otázky, musí existovat téma. A pro vytvoření tématu, musí existovat uživatel. Proto při vytváření otázky se nejdřív zavolá metoda pro vytváření tématu, kde se zavolá metoda pro vytváření uživatele.



Obr. 14. Použití šablony

Všechny šablony mají jednotnou strukturu a díky tomu je vytváření testů mnohem jednodušší a rychlejší.

5.1.1. Jednotkové testy

Hlavní princip jednotkových testů – srovnání výsledku volání metody nebo procedury s očekávaným výsledkem [12].

V současné době serverová část aplikace obsahuje 107 jednotkových testů.

5.1.2. Integrační testy

Integrační testy jsou dalším krokem ke zjištění případných chyb systému.

Součástí integračních testů je testování koncových bodů. Cílem tohoto testování je ověření, zda poskytované rozhraní splňuje požadavky na funkcionalitu a bezpečnost.

Testy ověřují, zda koncový bod správně přijímá vstupní objekty a porovnávají odpovědi s očekávaným výstupem.

V současné době serverová část aplikace obsahuje 63 API testů.

5.2. Black-box testování

5.2.1. Systémové testování

Pro účely této práce byly použité funkční testovací scénáře vytvořené podle případů užití.

Samotné scénáře jsou detailně popsány v *Příloha B. Scénáře testování*. Pro přehlednost všechny scénáře byly rozdělené do pěti skupin: přihlášení, téma/podtéma, otázky/odpovědi, testy a reporty.

Scénáře mají následující atributy:

- Jednoznačný identifikátor/číslo
- Název
- Vstupní podmínky
- Testovací data
- Kroky testu
- Očekávaný výsledek

Během testování bylo nalezeno a upraveno 6 chyb.

5.2.2. Uživatelské testování

Účelem uživatelského testování je ověřit použitelnost uživatelského rozhraní. Vzhledem k tomu, že cílová skupina jsou primárně studenti, bylo vybráno 5 studentů ČVUT.

Testování probíhalo podle předpřipravených úkolů, které ilustrují běžné použití aplikace:

1. Přihlásit se do webové aplikace a změnit přihlašovací údaje.
2. Vytvořit téma a přidat k němu 2 otázky s odpověďmi. První otázka s odpověďmi je bez obrázků, druhá s obrázky.

3. Vytvořit test a přidat k němu existující otázky.
4. Spustit test, odpovědět na otázky, nahlédnout výsledky testu a spustit test znovu.
5. Zveřejnit test a následně vyhledat tento test v seznamu veřejných testů.
6. Sdílet test s uživatelem “bashkkse”.
7. Vyhledat veřejný test “Lineární Algebra“, vybrat libovolnou otázku a nahlásit příspěvek.
8. Odhlásit se z aplikace.

Po splnění všech úkolů každý uživatel vyplnil výstupní anketu, kde odpověděl na následující otázky:

1. Využili byste aplikaci TestQuiz pro přípravu na zkoušky a učení se?
2. Měli jste během splnění úkolů nějaké problémy? Pokud ano, uveďte úkol, problém a rekomendaci.

Vyplněné ankety jsou uvedené v *Příloha C. Výsledné ankety testování*.

Vyhodnocení výsledků ankety.

Všechny uživatele odpověděli že by použili aplikace TestQuiz pro přípravu na zkoušky a učení se.

Uživatel 1 uvedl že při zveřejnění test se mizí ze záložky s testy. Tato chyba byla opravena.

Uživatel 2 uvedl že na začátku nepochopil rozdíl mezi testem a tématem. Toto nebylo považováno za chybu, jelikož na přihlašovací stránce je úvodní informace o tom, co poskytuje aplikace.

Uživatel 3 uvedl že je potřeba vyplňovat ručně jméno uživatele při sdílení testu. Toto nebylo považováno za chybu, protože bez zadání jména aplikace nebude schopna najít uživatele.

Uživatel 4 uvedl že pro něj bylo matoucí najít veřejné testy. To bylo opraveno přidáním popisu při navedení myši na sekce v menu aplikace.

Uživatel 5 uvedl že by nebylo špatně vizuálně rozdělit otázky a odpovědi v testu. To bylo opravené.

6. Závěr

Cílem této bakalářské práce byla analýza, návrh, implementace a testování webové aplikace TestQuiz, která bude sloužit jako pomůcka studentům během studia.

V této práci byla provedena rešerše již existujících aplikací, které umožní vytvářet vlastní testy a cvičit je. Z analýzy vyplývá závěr, že žádná z existujících aplikací nesplňuje všechny požadavky, které by měla splňovat aplikace pro vytváření vlastních testů a následně jejich procvičení a editaci. To bylo důvodem k návrhu vlastní aplikace.

Dále byly definované požadavky na systém a na jejich základě byly detailně popsány případy užití a navržen analytický doménový model tříd.

Potom byla provedena analýza technologií a na základě této analýzy byly vybrány nejvhodnější technologie pro implementaci – Java, Spring Boot, Javascript, React, PostgreSQL a Heroku.

Potom co byly zvolené vhodné technologie byla popsána implementace aplikace včetně fyzické architektury, použitých knihoven, nástrojů, zabezpečení, poskytovaného webového rozhraní a nasazení na cloudovou platformu.

Na konci byly uvedeny způsoby, kterými se výsledná aplikace byla otestována včetně hodnocení potenciálních uživatelů.

Cíl této práce považuji za splněný. Všechny funkční a kvalitativní požadavky jsou splněné, aplikace je implementována, nasazena a připravena ke skutečnému provozu.

Dále budou uvedeny možnosti zlepšení a rozšíření aplikace.

6.1.1. Rozšíření systému z pohledu požadavků

Rozšíření typů testovacích otázek o:

- Doplnovací otázky, kde uživatel bude moct vyplnit odpověď buď slovem, souvětím nebo číslem.
- Uspořádací otázky, kde uživatel sestaví pořadí odpovědí.
- Přiřazovací otázky, kde uživatel k možnostem na levé straně přiřazuje z výběru odpovědí na pravé straně.

Přidání možnosti importu a exportu otázek.

Přidání možnosti vyzvat jiného uživatele na kvíz v reálném čase.

6.1.2. Rozšíření z pohledu implementace

První krok pro rozšíření je provést analýzu a implementovat automatizované testování (bez zásahu lidského faktoru), pro časovou úsporu testovacího procesu a snížení výskytu chyb při testování.

Druhý krok pro rozšíření je vytvoření placeného účtu na Heroku. To hlavně umožní použít vlastní doménu a aplikační server se po dobu nečinnosti 30 minut nebude vypínat.

Citovaná literatura

- [1] H. Hudák, „VŠE TESTER,“ [Online]. Available: <http://www.vsetester.wz.cz/>.
- [2] R. S. P. B. Maxim, „Software Engineering: A Practitioner's Approach,“ [Online]. Available: [http://ce.sharif.edu/courses/98-99/2/ce474-2/resources/root/Roger%20S.%20Pressman_%20Bruce%20R.%20Maxin%20-%20Software%20Engineering_%20A%20Practitioner%E2%80%99s%20Approach-McGraw-Hill%20Education%20\(2014\).pdf](http://ce.sharif.edu/courses/98-99/2/ce474-2/resources/root/Roger%20S.%20Pressman_%20Bruce%20R.%20Maxin%20-%20Software%20Engineering_%20A%20Practitioner%E2%80%99s%20Approach-McGraw-Hill%20Education%20(2014).pdf). [Přístup získán 15 květen 2021].
- [3] IBM, "Rational Software Architect," [Online]. Available: <https://www.ibm.com/docs/en/rational-soft-arch/9.7.0?topic=diagrams-use-case>. [Accessed 8 duben 2021].
- [4] M. KOMÁREK, „Případy užití [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, březen 2020,“ [Online]. Available: https://moodle.fel.cvut.cz/pluginfile.php/233850/mod_resource/content/5/P%C5%99edn%C3%A1%C5%A1ka4a5_PripadyUzitiKomplet.pdf. [Přístup získán 1 květen 2021].
- [5] M. Komárek, „UML Deployment Diagram [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2020,“ [Online]. Available: https://moodle.fel.cvut.cz/pluginfile.php/233856/mod_resource/content/1/UML%20-%20Diagram%20nasazeni.pdf. [Přístup získán 4 prosinec 2021].
- [6] „Spring Boot Starter Web,“ [Online]. Available: <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web/2.6.1>. [Přístup získán 30 listopad 2021].
- [7] „Database Migrations with Flyway,“ [Online]. Available: <https://www.baeldung.com/database-migrations-with-flyway>. [Přístup získán listopad 2021].
- [8] „springdoc-openapi v1.5.13,“ [Online]. Available: <https://springdoc.org/#Introduction>. [Přístup získán 2 prosinec 2021].
- [9] K. R. P. S. Tomáš Mayer, „Návrhy rozhraní [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2014,“ [Online]. Available: https://cw.fel.cvut.cz/b181/_media/courses/b6b36omo/12_specifikace_rozhrani_v_2.pdf. [Přístup získán 2 prosinec 2021].
- [10] „Introduction to JSON Web Tokens,“ [Online]. Available: <https://jwt.io/introduction>. [Přístup získán 30 listopad 2021].
- [11] M. Bureš, „Úvod, typy testů, V-model, W-model, black-box a white-box testing, testovací scénář, proces opravy chyb [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2016,“ [Online]. Available: https://moodle.fel.cvut.cz/pluginfile.php/235106/mod_resource/content/1/TS1_prednaska1.pdf. [Přístup získán 12 prosinec 2021].
- [12] M. Bureš, „Jendotkové testování: JUnit, TestNG a základy efektivního návrhu [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2016,“ [Online]. Available: https://moodle.fel.cvut.cz/pluginfile.php/235111/mod_resource/content/1/TS1_prednaska_6_7_8.pdf. [Přístup získán 12 prosinec 2021].

- [13] StackOverflow, „2021 Developer Survey,“ [Online]. Available: <https://insights.stackoverflow.com/survey/2021#technology-most-popular-technologies>. [Přístup získán 16 prosinec 2021].
- [14] J. P. Jakub Urban, „Vědecké programování v Pythonu,“ [Online]. Available: <https://coobas.gitlab.io/python-fjfi/posts/vyjimky.html>. [Přístup získán 6 červenec 2021].
- [15] J. Vokřínek, „Úvod. Programování v JAVA [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2018,“ [Online]. Available: https://cw.fel.cvut.cz/b202/_media/courses/b0b36pjv/prednasky/lecture01-slides.pdf. [Přístup získán 3 červenec 2021].
- [16] J. Máca, „Úvod do Node.js,“ [Online]. Available: <https://www.itnetwork.cz/javascript/nodejs/uvod-do-nodejs>. [Přístup získán 6 červenec 2021].
- [17] B. Wagner, „Prohlídka jazyka C#,“ [Online]. Available: <https://docs.microsoft.com/cs-cz/dotnet/csharp/tour-of-csharp/>. [Přístup získán 7 červenec 2021].
- [18] „W3Schools. PHP Tutorial,“ [Online]. Available: <https://www.w3schools.com/php/>. [Přístup získán 10 červenec 2021].
- [19] M. Klíma, „Základy jazyka PHP [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2018,“ [Online]. Available: https://cw.fel.cvut.cz/b181/_media/courses/b6b39zwa/lectures/04/php_jazyk.pdf. [Přístup získán 8 červenec 2021].
- [20] „Spring vs. Spring Boot vs. Spring MVC,“ [Online]. Available: <https://www.javatpoint.com/spring-vs-spring-boot-vs-spring-mvc>. [Přístup získán 20 listopad 2021].
- [21] J. Kvapil, „Lekce 1 - Úvod do TypeScriptu,“ [Online]. Available: <https://www.itnetwork.cz/javascript/typescript/uvod-do-typescriptu>. [Přístup získán 10 srpen 2021].
- [22] hotframeworks, „Find your new favorite web framework,“ [Online]. Available: <https://hotframeworks.com/>. [Přístup získán 10 srpen 2021].
- [23] React, „Getting Started,“ [Online]. Available: <https://reactjs.org/>. [Přístup získán 10 srpen 2021].
- [24] D. Kadleček, „MapReduce v NoSQL světě [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, březen 2018,“ [Online]. Available: https://cw.fel.cvut.cz/wiki/_media/courses/b6b36omo/8_-_map_reduce_patterns_2018.pdf. [Přístup získán 20 srpen 2021].
- [25] M. Ledvinka, „Spring Framework – Handout [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2015,“ [Online]. Available: https://cw.fel.cvut.cz/old/_media/courses/a7b39wpa/spring-handout.pdf. [Přístup získán 12 prosinec 2021].
- [26] „Gradle vs Maven: Performance Comparison,“ [Online]. Available: <https://gradle.org/gradle-vs-maven-performance/>. [Přístup získán 30 listopad 2021].
- [27] Microsoft, „What is Git?,“ [Online]. Available: <https://docs.microsoft.com/en-us/devops/develop/git/what-is-git>. [Přístup získán 9 prosinec 2021].
- [28] Y. SHAFRANOVICH, "Common Format and MIME Type for CSV Files," říjen 2005. [Online]. Available: <https://www.ietf.org/rfc/rfc4180.txt#page-1>.



- [29] L. Hrušková, „Databáze – úvod [přednáška], Střední škola informačních technologií a sociální péče, Brno, únor 2020,“ [Online]. Available: Dostupné z: https://moodle.sspbrno.cz/pluginfile.php/1950/mod_resource/content/1/database-uvod.pdf. [Přístup získán 10 květen 2021].
- [30] „Spring Framework,“ [Online]. Available: <https://projects.spring.io/spring-framework/>. [Přístup získán 8 duben 2021].
- [31] „TIOBE Index for November 2021,“ [Online]. Available: <https://www.tiobe.com/tiobe-index/>. [Přístup získán 30 listopad 2021].
- [32] „Spring framework,“ [Online]. Available: <https://spring.io/projects/spring-framework>. [Přístup získán 11 listopad 2021].
- [33] J. POTTER, „angular vs react vs vue,“ [Online]. Available: <https://www.npmtrends.com/@angular/core-vs-angular-vs-react-vs-vue>. [Přístup získán 30 listopad 2021].
- [34] D. Boneh, „Twenty Years of Attacks on the RSA Cryptosystem,“ [Online]. Available: <https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf>. [Přístup získán 30 listopad 2021].
- [35] „What is ECDSA Encryption? How does it work?,“ [Online]. Available: <https://www.encryptionconsulting.com/education-center/what-is-ecdsa/>. [Přístup získán 30 listopad 2021].
- [36] J. Šebek, „GRAPHQL VS REST API [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2019,“ [Online]. Available: https://cw.fel.cvut.cz/b191/_media/courses/b6b36omo/12_graphql.pdf. [Přístup získán 3 prosinec 2021].
- [37] „Úvod do JSON,“ [Online]. Available: <https://www.json.org/json-cz.html>. [Přístup získán 3 prosinec 2021].
- [38] L. R. E. Quin, „XML ESSENTIALS,“ [Online]. Available: <https://www.w3.org/standards/xml/core>. [Přístup získán 3 prosinec 2021].
- [39] M. Klíma, „Oddělení aplikační a prezentační logiky [přednáška]. Praha, ČVUT, Fakulta elektrotechnická, 2021,“ [Online]. Available: https://cw.fel.cvut.cz/wiki/_media/courses/b6b39zwa/lectures/11/mvc_mustache.pdf. [Přístup získán 4 prosinec 2021].
- [40] JetBrains, „The State of Developer Ecosystem 2021,“ [Online]. Available: <https://www.jetbrains.com/lp/devecosystem-2021/>. [Přístup získán 16 prosinec 2021].
- [41] D. Bureš, „Proč Python, jak Python, kde Python,“ [Online]. Available: <https://zdrojak.cz/clanky/proc-python-jak-python-kde-python/>. [Přístup získán 2 červenec 2021].
- [42] J. Ginchereau, „The V8 JavaScript Engine,“ [Online]. Available: <https://nodejs.dev/learn/the-v8-javascript-engine>. [Přístup získán 6 červenec 2021].



Seznam zkratek

API – Application Programming Interface
CSS – Cascading Style Sheets
DAO – Data Access Object
DOM – Document Object Model
DTO – Data Transfer Object
FRQ – Functional Requirements
HTML – Hypertext Markup Language
HTTP – Hypertext Transfer Protocol
JPA – Java Persistence API
JSON – JavaScript Object Notation
JWT – JSON Web Token
MVC – Model View Controller
NFR – Non-functional Requirement
OOP – Object-Oriented Programming
PaaS – Platform as a Service
REST – Representational State Transfer
SQL – Structured Query Language
SRQ – System Requirements
UI – User Interface
UML – Unified Modeling Language
XML – Extensible Markup Language
YAML – YAML Ain't Markup Language

Příloha A. Případy užití

Přihlášení

UC 101 - Zaregistrovat se v systému
<i>Popis: Systém umožní uživateli zaregistrovat se v systému.</i>
<i>Systémový požadavek: SRQ 101 – Zaregistrovat se v systému.</i>
<i>Možnosti spuštění: Kliknutím na zaregistrovat se na přihlašovací stránce aplikace.</i>
 Hlavní scénář. <ol style="list-style-type: none">1. Uživatel zvolí možnost vytvořit uživatelský účet.2. Systém zobrazí registrační formulář.3. Uživatel vyplní povinné údaje: login, heslo, jméno, příjmení a e-mail.4. Systém zobrazí hlášku, že uživatel byl úspěšně zaregistrován.
 Výjimečný scénář. Uživatel je již v systému. <ol style="list-style-type: none">1. Uživatel zvolí možnost vytvořit uživatelský účet.2. Systém zobrazí registrační formulář.3. Uživatel vyplní povinné údaje: login, heslo, jméno a příjmení a e-mail.4. Systém zobrazí hlášku, že uživatel je již zaregistrován.

UC 102 - Přihlásit se do systému
<i>Popis: Systém umožní uživateli přihlásit se do systému.</i>
<i>Systémový požadavek: SRQ 102 – Přihlásit se do systému.</i>
<i>Možnosti spuštění: Kliknutím na přihlásit se na hlavní stránce aplikace.</i>
 Hlavní scénář. <ol style="list-style-type: none">1. Uživatel zvolí možnost přihlásit se.2. Systém zobrazí okénko pro přihlášení.3. Uživatel vyplní přihlašovací údaje a potvrdí je.4. Systém přesměruje uživatele na hlavní stránku.
 Výjimečný scénář. Špatně zadané přihlašovací údaje. <ol style="list-style-type: none">1. Uživatel zvolí možnost přihlásit se.2. Systém zobrazí okénko pro přihlášení.3. Uživatel vyplní přihlašovací údaje a potvrdí je.4. Systém vypíše hlášku, že uživatel zadal špatný login nebo heslo.

UC 103 – Odhlásit se ze systému

Popis: Systém umožní uživateli odhlásit se ze systému.

Systémový požadavek: SRQ 103 – Odhlásit se ze systému.

Možnosti spuštění: Kliknutím na odhlásit se na hlavní stránce aplikace.

Hlavní scénář

1. Uživatel zvolí možnost odhlásit se.
2. Systém vypíše hlášku, že uživatel byl úspěšně odhlášen.

UC 104 – Změnit heslo

Popis: Systém umožní uživateli změnit heslo.

Systémový požadavek: SRQ 104 – Změnit heslo.

Možnosti spuštění: Kliknutím na změnit heslo v detailu účtu uživatele.

Hlavní scénář.

1. Uživatel zvolí možnost změnit heslo.
2. Systém zobrazí formulář pro změnu hesla, kde uživatel musí zadat staré a nové heslo.
3. Uživatel vyplní formulář a potvrdí jej.
4. Systém zobrazí hlášku, že heslo bylo úspěšně změněno.

UC 105 – Změnit e-mail

Popis: Systém umožní uživateli změnit e-mailovou adresu.

Systémový požadavek: SRQ 105 – Změnit e-mail.

Možnosti spuštění: Kliknutím na změnit e-mail v detailu účtu uživatele.

Hlavní scénář.


1. Uživatel zvolí možnost změnit e-mail.
2. Systém zobrazí formulář pro změnu e-mailu, kde uživatel musí zadat nový e-mail odlišný od starého.
3. Uživatel vyplní formulář potvrdí je.
4. Systém zobrazí hlášku, že e-mail byl úspěšně změněn.

UC 105 – Změnit login

Popis: Systém umožní uživateli změnit login.

Systémový požadavek: SRQ 106 – Změnit login.

Možnosti spuštění: Kliknutím na změnit login v detailu účtu uživatele.

 Hlavní scénář.

1. Uživatel zvolí možnost změnit login.
2. Systém zobrazí formulář pro změnu, kde uživatel musí zadat nový unikátní login odlišný od starého.
3. Uživatel vyplní formulář potvrdí jej.
4. Systém zobrazí hlášku, že login byl úspěšně změněn.


Téma/podtéma

UC 201 - Vytvořit téma/podtéma


Popis: Systém umožní uživateli vytvořit nové téma nebo podtéma v rámci vybraného tématu.

Systémový požadavek: SRQ 201 - Vytvořit téma/podtéma.

Možnosti spuštění: Kliknutím na vytvořit nové téma/podtéma.

 Hlavní scénář.

1. Uživatel vybere možnost vytvořit nové téma, nebo podtéma v rámci existujícího tématu.
2. Systém zobrazí okénko pro vytvoření.
3. Uživatel vybere, jestli chce vytvořit téma nebo podtéma, vyplní název a případně popis. Následně potvrdí formulář.
4. Systém zobrazí hlášku, že téma/podtéma bylo úspěšně vytvořeno a přidá je do seznamu témat.

 Výjimečný scénář. Téma s takovým názvem již existuje.

1. Uživatel vybere možnost vytvořit nové téma nebo podtéma v rámci existujícího tématu.
2. Systém zobrazí okénko pro vyplnění názvu tématu.
3. Uživatel vyplní název a potvrdí je.
4. Systém zobrazí hlášku, že téma s takovým názvem již existuje a nabídne možnost zadat jiný název nebo se vrátit zpátky.

UC 202 - Odstranit téma/podtéma

Popis: Systém umožní uživateli odstranit vytvořené téma a podtéma.

Systémový požadavek: SRQ 202 - Odstranit téma/podtéma.


Možnosti spuštění: Kliknutím na odstranit téma/podtéma.

UC 202 - Odstranit téma/podtéma

Popis: Systém umožní uživateli odstranit vytvořené téma a podtéma.

Systémový požadavek: SRQ 202 - Odstranit téma/podtéma.

Možnosti spuštění: Kliknutím na odstranit téma/podtéma.

 Hlavní scénář.

1. Uživatel vybere téma nebo podtéma a možnost odstranit.
2. Systém zobrazí potvrzovací okénko.
3. Uživatel potvrdí odstranění.
4. Systém zobrazí hlášku, že téma bylo úspěšně odstraněno ze systému.

UC 203 - Změnit název tématu/podtématu


Popis: Systém umožní uživateli změnit existující název tématu/podtématu.

Systémový požadavek: SRQ 203 - Změnit název tématu/podtématu.

Možnosti spuštění: Kliknutím na tlačítko změnit název u vybraného tématu.

 Hlavní scénář.

1. Uživatel vybere téma nebo podtéma a vybere možnost změnit název.
2. Systém zobrazí okénko, kde uživatel zadá nový název.
3. Uživatel zadá nový název.
4. Systém zobrazí hlášku, že název byl úspěšně změněn.

 Výjimečný scénář. Téma s takovým názvem již existuje.

1. Uživatel vybere téma nebo podtéma a vybere možnost změnit název.
2. Systém zobrazí okénko, kde uživatel zadá nový název.
3. Uživatel zadá nový název.
4. Systém zobrazí hlášku, že téma s takovým názvem již existuje. Následně nabídne možnost zadat jiný název nebo se vrátit zpět.

UC 204 - Zobrazit seznam témat/podtémat

Popis: Systém umožní uživateli zobrazit seznam vytvořených témat/podtémat.

Systémový požadavek: SRQ 204 – Zobrazit téma/podtéma.

Možnosti spuštění: Zobrazení všech témat – otevřením sekce téma. Zobrazení podtémat – rozkliknutím konkrétního tématu.

 Hlavní scénář.

1. Uživatel vybere možnost zobrazit seznam témat.
2. Systém zobrazí seznam.
3. Uživatel vybere nějaké téma.

UC 204 - Zobrazit seznam témat/podtémat

Popis: Systém umožní uživateli zobrazit seznam vytvořených témat/podtémat.

Systémový požadavek: SRQ 204 – Zobrazit téma/podtéma.

Možnosti spuštění: Zobrazení všech témat – otevřením sekce téma. Zobrazení podtémat – rozkliknutím konkrétního tématu.

4. Systém zobrazí seznam témat vytvořených v rámci daného tématu.

Otázky/odpovědi

UC 301- Vytvořit otázku


Popis: Systém umožní uživateli vytvořit otázku v rámci tématu/podtématu.

Systémový požadavek: SRQ 301 - Vytvořit otázku.

Možnosti spuštění: Kliknutím na tlačítko vytvořit otázku.

 Hlavní scénář.

1. Uživatel vybere možnost vytvořit novou otázku v rámci tématu nebo podtématu.
2. Systém zobrazí formulář pro vytvoření nové otázky.
3. Uživatel vyplní formulář, kde uvede text otázky a případně přidá obrázek (viz případ užití 302 - Přidat obrázek k otázce odpovědi nebo) a potvrdí jej.
4. Systém zobrazí hlášku, že otázka byla úspěšně vytvořena a přidá ji do seznamu otázek.

 Alternativní scénář. Přidat odpověď během vytvoření otázky.

1. Uživatel vybere možnost vytvořit novou otázku v rámci tématu nebo podtématu.
2. Systém zobrazí formulář pro vytvoření nové otázky.
3. Uživatel vyplní formulář, kde uvede text otázky a případně přidá obrázek.
4. Uživatel přidá odpověď (viz UC 307 – Vytvořit odpověď) a potvrdí formulář.
5. Systém zobrazí hlášku, že otázka byla úspěšně vytvořena a přidá je do seznamu otázek.

UC 302 - Přidat obrázek k otázce nebo odpovědi

Popis: Systém umožní uživateli přidat obrázek k otázce nebo odpovědi.

Systémový požadavek: SRQ 301 - Vytvořit otázku, SRQ 304 - Vytvořit odpověď.

Možnosti spuštění: Kliknutím na tlačítko přidat obrázek při vytvoření otázky nebo odpovědi.

 Hlavní scénář.


1. Uživatel vybere možnost přidat obrázek.
2. Systém zobrazí správce souborů.
3. Uživatel vybere obrázek.
4. Systém zobrazí obrázek v políčku otázky nebo odpovědi.

UC 303 – Odstranit otázku

Popis: Systém umožní uživateli odstranit vytvořenou otázku.

Systémový požadavek: SRQ 302 – Odstranit otázku.

Možnosti spuštění: Kliknutím na tlačítko odstranit otázku.

 Hlavní scénář.

1. Uživatel vybere otázku a vybere možnost odstranit ji.
2. Systém zobrazí potvrzovací okénko.
3. Uživatel potvrdí odstranění.
4. Systém zobrazí hlášku, že otázka byla úspěšně odstraněna ze systému.

UC 304 – Odstranit nahlášenou otázku

Popis: Systém umožní uživateli odstranit veřejně dostupnou otázku, která byla nahlášena jiným uživatelem.

Systémový požadavek: SRQ 303 – Odstranit nahlášenou otázku.

Možnosti spuštění: Kliknutím na tlačítko odstranit otázku.

 Hlavní scénář.

1. Systém zobrazí otázku, která byla nahlášena.
2. Uživatel vybere možnost odstranit otázku.
3. Systém zobrazí potvrzovací okénko.
4. Uživatel potvrdí odstranění.
5. Systém zobrazí hlášku, že otázka byla úspěšně odstraněna ze systému.

UC 305 – Zobrazit otázku

Popis: Systém umožní uživateli zobrazit vytvořenou otázku včetně všech odpovědí.

Systémový požadavek: SRQ 304 – Zobrazit otázku.

Možnosti spuštění: Rozkliknutím otázky.

 Hlavní scénář.

1. Uživatel vybere otázku.
2. Systém zobrazí otázku s odpověďmi.

UC 306 – Zobrazit seznam otázek

Popis: Systém umožní uživateli zobrazit vytvořené otázky včetně všech odpovědí.

Systémový požadavek: SRQ 305 – Zobrazit seznam otázek.

Možnosti spuštění: Rozkliknutím tématu.

 Hlavní scénář.

1. Uživatel vybere téma, u kterého chce zobrazit otázky.
2. Systém zobrazí seznam otázek.

UC 307 – Vytvořit odpověď

Popis: Systém umožní uživateli vytvořit odpověď k vybrané otázce.

Systémový požadavek: SRQ 306 – Vytvořit odpověď.

Možnosti spuštění: Kliknutím na přidat odpověď při vytvoření otázky nebo kliknutím přidat odpověď při rozkliknutí otázky v seznamu otázek.

 Hlavní scénář.


1. Uživatel vybere možnost přidat odpověď k otázce.
2. Systém zobrazí políčko pro přidání odpovědi.
3. Uživatel uvede text odpovědi a případně přidá obrázek (viz případ užití 302 - Přidat obrázek k otázce nebo odpovědi) a potvrdí přidání.
4. Systém zobrazí odpověď v seznamu odpovědí vybrané otázky.

UC 308 – Odstranit odpověď

Popis: Systém umožní uživateli odstranit vytvořenou odpověď.

Systémový požadavek: SRQ 307 – Odstranit odpověď.

Možnosti spuštění: Kliknutím na tlačítko odstranit odpověď u vybrané otázky.

 Hlavní scénář.

1. Uživatel vybere odpověď a vybere možnost odstranit ji.
2. Systém zobrazí potvrzovací okénko.
3. Uživatel potvrdí odstranění.
4. Systém odstraní odpověď ze seznamu odpovědí u vybrané otázky.

Testy

UC 401- Vytvořit test

Popis: Systém umožní uživateli vytvořit test.

Systémový požadavek: SRQ 401 – Vytvořit test.

Možnosti spuštění: Kliknutím na vytvořit test.

 Hlavní scénář.


1. Uživatel vybere možnost vytvořit nový test.
2. Systém zobrazí okénko pro zadání názvu testu.
3. Uživatel zadá název a potvrdí vytvoření.
4. Systém zobrazí hlášku, že test byl úspěšně vytvořen a přidá ho do seznamu testů.

UC 402 – Odstranit test

Popis: Systém umožní uživateli odstranit test se všemi otázkami.

Systémový požadavek: SRQ 402 – Odstranit test.

Možnosti spuštění: Kliknutím na tlačítko odstranit test.

 Hlavní scénář.


1. Uživatel vybere test ze seznamu testů a vybere možnost odstranit.
2. Systém zobrazí potvrzovací okénko.
3. Uživatel potvrdí odstranění.
4. Systém zobrazí hlášku, že test byl úspěšně odstraněn ze systému.

UC 403 – Zobrazit test

Popis: Systém umožní uživateli zobrazit test včetně seznamu otázek.

Systémový požadavek: SRQ 403 – Zobrazit test.

Možnosti spuštění: Rozkliknutím konkrétního testu.

 Hlavní scénář.


1. Uživatel vybere test, který chce prohlédnout.
2. Systém zobrazí test se seznamem otázek.

UC 404 – Zobrazit seznam vytvořených testů

Popis: Systém umožní uživateli zobrazit seznam všech vytvořených testů.

Systémový požadavek: SRQ 404 – Zobrazit seznam vytvořených testů.

Možnosti spuštění: Kliknutím na tlačítko testy.

 Hlavní scénář.


1. Uživatel vybere možnost zobrazit testy.
2. Systém zobrazí všechny testy, které vytvořil uživatel.

UC 405 - Přidat otázky k testu

Popis: Systém umožní uživateli přidat otázky k vytvořenému testu ze seznamu všech otázek.

Systémový požadavek: SRQ 405 – Přidat otázky.

Možnosti spuštění: Kliknutím na tlačítko přidat otázky v rámci vybraného testu.

 Hlavní scénář.


1. Uživatel vybere možnost přidat otázky.
2. Systém zobrazí seznam témat a podtémat s otázkami, které se tam nachází.
3. Uživatel vybere otázky a přidá je do testu.
4. Systém zobrazí vybrané otázky.

UC 406 - Odebrat otázky z testu

Popis: Systém umožní uživateli odstranit otázku z vybraného testu.

Systémový požadavek: SRQ 406 – Odebrat otázku.

Možnosti spuštění: Kliknutím na tlačítko odstranit otázku v rámci vybraného testu.

 Hlavní scénář.


1. Uživatel vybere otázku ze seznamu otázek v testu a vybere možnost odstranit ji.
2. Systém zobrazí potvrzovací okénko.
3. Uživatel potvrdí odstranění.
4. Systém zobrazí hlášku, že otázka byla úspěšně odstraněna ze systému.

UC 407 – Spustit test


Popis: Systém umožní uživateli spustit vytvořený test a procvičit jej. Při spuštění testu je nutné uvést čas za který test bude automaticky ukončen. Test se ukončí buď po vypršení času, nebo když uživatel vybere možnost ukončit test.

Systémový požadavek: SRQ 407 – Spustit test.

Možnosti spuštění: Kliknutím na spustit test.

 Hlavní scénář.

1. Uživatel vybere test ze seznamu testů a vybere možnost spustit test.
2. Uživatel vybere jeden ze způsobů kontroly odpovědí.
3. INCLUDE (UC 413 - Nastavit časovač na test)
4. Systém zobrazí popořadě všechny otázky, které byly přidány do testu.
5. Uživatel odpoví na otázky.
6. Po vypršení času systém ukončí test.

 Alternativní scénář. Uživatel chce ukončit test dřív, než vyprší čas.


1. Uživatel vybere test ze seznamu testů a vybere možnost spustit test.
2. Uživatel vybere jeden ze způsobů kontroly odpovědí.
3. INCLUDE (UC 413 - Nastavit časovač na test).
4. Systém zobrazí popořadě všechny otázky.
5. Uživatel odpoví na otázky a chce ukončit test.
6. INCLUDE (UC 408 – Ukončit test).

UC 408 - Ukončit test

Popis: Systém umožní uživateli ukončit spuštěný test dříve, než nevyprší čas, který byl nastavený uživatelem při spuštění testu.

Systémový požadavek: SRQ 408 – Ukončit test.

Možnosti spuštění: Kliknutím na ukončit test.

 Hlavní scénář.

1. Uživatel vybere možnost ukončit test.
2. Systém ukončí test a zobrazí výsledky.

UC 409 - Spustit test pouze špatně zodpovězených otázek

Popis: Systém umožní uživateli po dokončení testu spustit test špatně zodpovězených otázek. Tenhle krok se může opakovat, dokud seznam špatně zodpovězených otázek nebude prázdný.

Systémový požadavek: SRQ 409 - Spustit test pouze špatně zodpovězených otázek.

Možnosti spuštění: Kliknutím na tlačítko spustit test špatně zodpovězených otázek.

 Hlavní scénář.


1. Uživatel vybere možnost spustit test špatně zodpovězených otázek.
2. INCLUDE (UC 407 – Spustit test).
2. Systém spustí test.

UC 410 - Sdílet test

Popis: Systém umožní uživateli sdílet svůj test jinými uživateli, pokud jsou tito uživatelé také zaregistrováni v systému. Po sdílení se test zobrazí v seznamu sdílených testů uživatele, se kterými byl test sdílen.

Systémový požadavek: SRQ 410 – Sdílet test.

Možnosti spuštění: Kliknutím na tlačítko sdílet test.

 Hlavní scénář.

1. Uživatel vybere test, který chce sdílet a vybere možnost sdílet.
2. Systém zobrazí seznam uživatelů.
3. Uživatel vyhledá uživatele, kterému chce sdílet test a vybere ho.
4. Systém zobrazí potvrzovací okénko.
5. Uživatel potvrdí sdílení.
6. Systém zobrazí hlášku, že test byl úspěšně sdílen a zobrazí test v seznamu sdílených testů druhého uživatele.

UC 411 - Zveřejnit test

Popis: Systém umožní uživateli zveřejnit svůj test. Zveřejněný test se zobrazí ostatním uživatelům v seznamu veřejných testů.

Systémový požadavek: SRQ 411 – Zveřejnit test.

Možnosti spuštění: Kliknutím na sdílet test.

 Hlavní scénář.


1. Uživatel vybere test a vybere možnost zveřejnit.
2. Systém zobrazí formulář.
3. Uživatel vybere kategorii, ke které patří test, a potvrdí zveřejnění.
4. Systém zobrazí hlášku, že test byl úspěšně zveřejněn a zobrazí test v seznamu veřejných testů ostatních uživatelů.

UC 412 - Nahlednout vysledky testu

Popis: Systém po ukončení testu zobrazí informace o výsledcích testu, tj. počet špatně a správně zodpovězených otázek, celkovou úspěšnost a čas, za který byl test vyplněn.

Systémový požadavek: SRQ 412 – Nahlednout výsledky.

Možnosti spuštění: Spustí se automaticky po ukončení testu.

 Hlavní scénář.


1. Uživatel dokončí test.
2. Systém zobrazí výsledky testu.

UC 413 - Nastavit časovač na test

Popis: Systém umožní uživateli zadat čas po kterém se test automaticky ukončí.

Systémový požadavek: SRQ 413 – Nastavit časovač na test.

Možnosti spuštění: Kliknutím na tlačítko spustit test.

 Hlavní scénář.


1. Uživatel vybere možnost spustit test.
2. Systém zobrazí okénko pro zadání času.
3. Uživatel zadá čas.
4. Systém spustí test.

UC 414 - Ohodnotit test

Popis: Systém umožní uživateli ohodnotit veřejně dostupný test ve formě „líbí se“, „nelíbí se“. Test, který získá více než 100 špatných hodnocení bude odstraněn z veřejně dostupných testů.

Systémový požadavek: SRQ 414 – Ohodnotit test.

Možnosti spuštění: Kliknutím na tlačítko „líbí se“ nebo „nelíbí se“ v rámci testu ze seznamu veřejně dostupných testů.

 Hlavní scénář.

1. Uživatel vybere veřejně dostupný test a klikne na jednu ze dvou možností hodnocení.
2. Systém zvětší počet ohodnocení.

UC 415 – Přidat veřejný test do záložek

Popis: Systém umožní uživateli přidat veřejně dostupný test do záložek.

Systémový požadavek: SRQ 415 – Přidat veřejný test do záložek.

Možnosti spuštění: Kliknutím na přidání do záložek.

 Hlavní scénář.

1. Systém zobrazí veřejné testy.
1. Uživatel vybere test a vybere možnost přidat test do záložek.
2. Systém označí test jako přidání.

UC 416 – Zobrazit seznam testů přidanych do záložek

Popis: Systém umožní uživateli zobrazit testy přidané do záložek.

Systémový požadavek: SRQ 416 – Zobrazit seznam přidaných testů do záložek.

Možnosti spuštění: Kliknutím na uložené testy.

 Hlavní scénář.


1. Uživatel vybere možnost zobrazit uložené testy.
2. Systém zobrazí všechny testy, které uživatel přidal do záložek.

UC 417 – Zobrazit seznam testů sdílených s uživatelem

Popis: Systém umožní uživateli zobrazit testy, které mu sdíleli jiní uživatelé.

Systémový požadavek: SRQ 417 – Zobrazit seznam testů sdílených s uživatelem.

Možnosti spuštění: Kliknutím na sdílené testy.

 Hlavní scénář.

1. Uživatel vybere možnost zobrazit uložené testy.
2. Systém zobrazí všechny testy, které uživatel přidal do záložek.

UC 418 – Uložit test

Popis: Systém umožní uživateli uložit sdílený test, nebo test přidaný uživatelem, do záložek svých testů.

Systémový požadavek: SRQ 418 – Uložit test.

Možnosti spuštění: Kliknutím na uložit test.

 Hlavní scénář.

1. Uživatel vybere test, který chce uložit.
2. Systém zobrazí hlášku, že test byl úspěšně uložen.


Reporty

UC 501 - Nahlásit příspěvek

Popis: Systém umožní uživateli nahlásit příspěvek ohledně veřejně dostupné otázky. Pro nahlášení je nutné vybrat jeden z následujících důvodů: porušení práv k duševnímu vlastnictví, nahota nebo sexuální obsah, nenávistné slovní projevy nebo jiný důvod.

Systémový požadavek: SRQ 501 – Nahlásit příspěvek.

Možnosti spuštění: Kliknutím na tlačítko nahlásit u vybrané veřejné otázky.

 Hlavní scénář.


1. Uživatel vybere otázku z veřejně dostupného testu a klikne na nahlásit.
2. Systém zobrazí formulář pro nahlášení, ve kterém bude nutné vybrat důvod nahlášení a případně uvést další popis.
3. Uživatel vyplní formulář a odešle ho.
4. Systém zobrazí potvrzovací okénko.
5. Uživatel potvrdí odeslání.
6. Systém zobrazí nahlášený příspěvek v seznamu u administrátora.

UC 502 - Zobrazit reporty

Popis: Systém umožní administrátoru zobrazit nahlášené příspěvky.

Systémový požadavek: SRQ 502 – Zobrazit reporty.

Možnosti spuštění: Kliknutím na tlačítko reporty.

 Hlavní scénář.

1. Uživatel vybere možnost zobrazit reporty.
2. Systém zobrazí všechny reporty.

Příloha B. Scénáře testování

Přihlášení

TS 101 – Úspěšná registrace v systému	
Vstupní podmínky	<i>Uživatel není registrovaný</i>
Testovací data	<i>Uživatelské jméno: bashkkse Jméno: Kseniia Příjmení: Bashkova Email: bashkkse@fel.cvut.cz Heslo: bashkkse Potvrzení hesla: bashkkse</i>
Kroky testu	<i>1. Uživatel vstoupí do přihlašovací stránky a zvolí možnost Registrovat se 2. Uživatel vyplní povinné údaje a stiskne tlačítko Registrovat</i>
Očekávaný výsledek	<i>Uživatel se úspěšně zaregistruje a zobrazí se mu formulář pro přihlášení</i>

TS 102 – Neúspěšná registrace v systému	
Vstupní podmínky	<i>Uživatel s uvedeným uživatelským jménem nebo emailem již existuje</i>
Testovací data	<i>Uživatelské jméno: bashkkse Jméno: Kseniia Příjmení: Bashkova Email: bashkkse@fel.cvut.cz Heslo: bashkkse Potvrzovací heslo: bashkkse</i>
Kroky testu	<i>1. Uživatel vstoupí do přihlašovací stránky a zvolí možnost Registrovat se 2. Uživatel vyplní povinné údaje</i>
Očekávaný výsledek	<i>Tlačítko Registrovat nebude aktivní a uživatel se nebude moct zaregistrovat</i>

TS 103 – Úspěšné přihlášení do systému	
Vstupní podmínky	<i>Uživatel je registrovaný</i>
Testovací data	<i>Uživatelské jméno: bashkkse Heslo: bashkkse</i>
Kroky testu	<i>1. Uživatel vstoupí do přihlašovací stránky a zvolí možnost Přihlásit se 2. Uživatel vyplní povinné údaje</i>

<i>Očekávaný výsledek</i>	<i>Uživatel se úspěšně přihlásí se a zobrazí se mu domovská stránka</i>
---------------------------	---

TS 104 – Neúspěšné přihlášení do systému	
<i>Vstupní podmínky</i>	<i>Uživatel je registrovaný, zadá špatné uživatelské jméno nebo heslo</i>
<i>Testovací data</i>	<i>Uživatelské jméno: bashkkse Heslo: lala</i>
<i>Kroky testu</i>	<i>1. Uživatel vstoupí do přihlašovací stránky a zvolí možnost Přihlásit se 2. Uživatel vyplní povinné údaje</i>
<i>Očekávaný výsledek</i>	<i>Uživatel se nepřihlásí a zobrazí se mu chybová hláška</i>

TS 105 – Odhlášení ze systému	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému</i>
<i>Testovací data</i>	<i>-</i>
<i>Kroky testu</i>	<i>1. Uživatel stiskne tlačítko Odhlásit se v menu aplikace</i>
<i>Očekávaný výsledek</i>	<i>Uživatel není přihlášený a zobrazí se mu přihlašovací stránka</i>

TS 106 – Úspěšná změna hesla	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému</i>
<i>Testovací data</i>	<i>Nové heslo: noveheslo Potvrzovací heslo: noveheslo</i>
<i>Kroky testu</i>	<i>1. Uživatel stiskne tlačítko Nastavení účtu v menu aplikace 2. Uživatel vyplní povinné údaje pro změnu hesla</i>
<i>Očekávaný výsledek</i>	<i>Heslo je změněné</i>

TS 107 – Neúspěšná změna hesla	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; potvrzovací heslo se neshoduje s heslem</i>
<i>Testovací data</i>	<i>Nové heslo: noveheslo Potvrzovací heslo: noveheslooo</i>
<i>Kroky testu</i>	<i>1. Uživatel stiskne tlačítko Nastavení účtu v menu aplikace 2. Uživatel vyplní povinné údaje pro změnu hesla</i>

<i>Očekávaný výsledek</i>	<i>Zobrazí se hláška, že potvrzovací heslo musí být stejné jako heslo</i>
---------------------------	---

TS 108 – Úspěšná změna e-mailu nebo uživatelského jména	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému</i>
<i>Testovací data</i>	<i>Nové uživatelské jméno: ksebash Nový e-mail: ksebash@gmail.com</i>
<i>Kroky testu</i>	<i>1. Uživatel stiskne tlačítko Nastavení účtu v menu aplikace 2. Uživatel vyplní povinné údaje pro změnu uživatelského jména nebo e-mailu</i>
<i>Očekávaný výsledek</i>	<i>E-mail nebo uživatelské jméno je změněné</i>

TS 109 – Neúspěšná změna e-mailu nebo uživatelského jména	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; e-mail nebo uživatelské jméno již existuje</i>
<i>Testovací data</i>	<i>Nové uživatelské jméno: ksebash Nový e-mail: ksebash@gmail.com</i>
<i>Kroky testu</i>	<i>1. Uživatel stiskne tlačítko Nastavení účtu v menu aplikace 2. Uživatel vyplní povinné údaje pro změnu uživatelského jména nebo e-mailu</i>
<i>Očekávaný výsledek</i>	<i>E-mail nebo heslo je nezměněné, zobrazí se hláška že uživatelské jméno nebo e-mail již existuje</i>

Téma/podtéma

TS 201 – Vytvoření tématu	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému</i>
<i>Testovací data</i>	<i>Název tématu: LAG</i>
<i>Kroky testu</i>	<i>1. Uživatel vybere sekci Téma v menu aplikace a klikne na Vytvořit téma 2. Uživatel vyplní povinné údaje a potvrdí vytváření</i>
<i>Očekávaný výsledek</i>	<i>Téma se zobrazí v tabulce</i>

TS 202 – Vytvoření podtématu	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému</i>

<i>Testovací data</i>	<i>Téma: LAG</i> <i>Název podtématu: SPAN</i>
<i>Kroky testu</i>	<ol style="list-style-type: none"> 1. Uživatel vybere sekci <i>Témata</i> v menu aplikace a klikne na <i>Vytvořit téma</i> 2. Uživatel vybere možnost <i>vytvořit podtéma</i>, vybere <i>téma</i> a vyplní <i>povinné údaje</i> 3. Uživatel potvrdí <i>vytváření</i>
<i>Očekávaný výsledek</i>	<i>Podtéma se zobrazí v tabulce při rozkliknutí tématu</i>

TS 203 – Odstranění tématu/podtématu

<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; téma/podtéma je vytvořená</i>
<i>Testovací data</i>	<i>Téma: LAG</i>
<i>Kroky testu</i>	<ol style="list-style-type: none"> 1. Uživatel označí <i>téma</i> v <i>tabulce</i> 2. Uživatel klikne na <i>Odstranit vybrané téma</i> 3. Uživatel potvrdí <i>odstranění</i> kliknutím tlačítka <i>Odstranit</i> v <i>potvrzovacím formuláři</i>
<i>Očekávaný výsledek</i>	<i>Téma je odstraněné z tabulky</i>

TS 204 – Změna názvu tématu/podtématu

<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; téma/podtéma je vytvořené</i>
<i>Testovací data</i>	<i>Současný název: LAG</i> <i>Nový název: MATH</i>
<i>Kroky testu</i>	<ol style="list-style-type: none"> 1. Uživatel vybere sekci <i>Témata</i> v menu aplikace a klikne na <i>Upravit</i> 2. Uživatel zadá <i>nový název</i> místo <i>současného</i> 3. Uživatel potvrdí <i>upravení</i>
<i>Očekávaný výsledek</i>	<i>Téma se zobrazí v tabulce s novým názvem</i>

Otázky/odpovědi

TS 301 – Vytvoření otázky s odpověďmi a obrázky

<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; téma/podtéma je vytvořené</i>
<i>Testovací data</i>	<i>Text otázky: Jsou testy užitečné?</i> <i>Text odpovědi 1: Ano</i> <i>Text odpovědi 2: Ne</i> <i>Správná odpověď: Ano</i>

Kroky testu	<ol style="list-style-type: none"> 1. Uživatel vybere sekci <i>Témata</i> v menu aplikace a rozklikne <i>téma</i> 2. Uživatel vybere možnost vytvořit otázku 3. Uživatel vyplní text otázky a přidá obrázek 4. Uživatel vytvoří odpovědi a přidá k nim obrázky 5. Uživatel potvrdí formulář
Očekávaný výsledek	Otázka se zobrazí v tabulce s otázkami

TS 302 – Odstranění otázky	
Vstupní podmínky	Uživatel je přihlášený do systému; téma/podtéma je vytvořené, otázka je vytvořená
Testovací data	Otázka: Jsou testy užitečné?
Kroky testu	<ol style="list-style-type: none"> 1. Uživatel označí otázku v tabulce 2. Uživatel klikne na <i>Odstranit</i> vybranou otázku 3. Uživatel potvrdí odstranění kliknutím tlačítka <i>Odstranit</i> v potvrzovacím formuláři
Očekávaný výsledek	Otázka je odstraněná z tabulky

TS 303 – Editace otázky	
Vstupní podmínky	Uživatel je přihlášený do systému; téma/podtéma je vytvořené; otázka je vytvořená
Testovací data	<p>Otázka: Jsou testy užitečné?</p> <p>Nový text otázky: Testy jsou užitečné pro nalezení chyb v systému?</p>
Kroky testu	<ol style="list-style-type: none"> 1. Uživatel rozklikne otázku v tabulce 2. Uživatel uvede nový text otázky 3. Uživatel potvrdí editaci kliknutím tlačítka <i>Editovat</i> v editačním formuláři
Očekávaný výsledek	Otázka je editovaná a po rozkliknutí otázky se zobrazí nový text

Testy

TS 401 – Vytvoření testu	
Vstupní podmínky	Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořená
Testovací data	Název testu: LAG zápočtový test
Kroky testu	<ol style="list-style-type: none"> 1. Uživatel vybere sekci <i>Testy</i> v menu aplikace a klikne na <i>Vytvořit</i> 2. Uživatel napíše název testu 3. Uživatel potvrdí vytvoření

<i>Očekávaný výsledek</i>	<i>Otázka je vytvořená a zobrazí se v tabulce</i>
---------------------------	---

TS 402 – Přidání otázek k testu	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořená; test je vytvořený</i>
<i>Testovací data</i>	<i>Název testu: LAG zápočtový test</i> <i>Otázka: Jsou testy užitečné pro nalezení chyb v systému?</i>
<i>Kroky testu</i>	<ol style="list-style-type: none"> 1. Uživatel vybere sekci Testy v menu aplikace a rozklikne test 2. Klikne na Přidat otázku k testu 3. Vybere téma a otázku a klikne na tlačítko Přidat
<i>Očekávaný výsledek</i>	<i>Otázka je přidána k testu a zobrazí se v tabulce otázek</i>

TS 403 – Spuštění testu	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořená; test je vytvořený; alespoň jedna otázka je přidána k testu</i>
<i>Testovací data</i>	<i>Název testu: LAG zápočtový test</i>
<i>Kroky testu</i>	<ol style="list-style-type: none"> 1. Uživatel klikne na tlačítko Spustit test 2. Vybere počet minut pro vykonání testu 3. Klikne na Začít test
<i>Očekávaný výsledek</i>	<i>Test je spuštěný, zobrazí se první otázka, spustí se odpočet času</i>

TS 404 – Ukončení testu	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořená; test je vytvořený; alespoň jedna otázka je přidána k testu; test je spuštěný; časový limit nevypršel</i>
<i>Testovací data</i>	<i>Název testu: LAG zápočtový test</i>
<i>Kroky testu</i>	<ol style="list-style-type: none"> 1. Uživatel klikne na tlačítko Ukončit test
<i>Očekávaný výsledek</i>	<i>Test je ukončený a uživateli se zobrazí výsledky testu</i>

TS 405 – Spuštění testu špatně zodpovězených otázek	
<i>Vstupní podmínky</i>	<i>Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořená; test je vytvořený, alespoň jedna otázka je přidána k testu; test je ukončený; alespoň jedna otázka byla špatně zodpovězena</i>

<i>Testovací data</i>	Název testu: LAG zápočtový test Spatně zodpovězená otázka: Jsou testy užitečné pro nalezení chyb v systému?
<i>Kroky testu</i>	1. Uživatel klikne na tlačítko Spustit test spatně zodpovězených otázek
<i>Očekávaný výsledek</i>	Test je spuštěný

TS 406 – Sdílení testu

<i>Vstupní podmínky</i>	Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořena; test je vytvořený; alespoň jedna otázka je přidána k testu; uživatel, kterému test bude sdílený existuje
<i>Testovací data</i>	Název testu: LAG zápočtový test Uživatelské jméno: uzivatel2
<i>Kroky testu</i>	1. Uživatel vybere sekci Testy v menu aplikace a rozklikne test 2. Klikne na tlačítko Sdílet test 3. Vyhledá jiného uživatele 4. Klikne na tlačítko Sdílet
<i>Očekávaný výsledek</i>	Test je sdílený a zobrazí se jinému uživateli v seznamu sdílených testů

TS 407 – Zveřejnění testu

<i>Vstupní podmínky</i>	Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořena; test je vytvořený; alespoň jedna otázka je přidána k testu
<i>Testovací data</i>	Název testu: LAG zápočtový test Kategorie testu: Matematika
<i>Kroky testu</i>	1. Uživatel vybere sekci Testy v menu aplikace a rozklikne test 2. Klikne na tlačítko Zveřejnit test 3. Vybere kategorii 4. Klikne na tlačítko Zveřejnit
<i>Očekávaný výsledek</i>	Test je zveřejněný a je možné vyhledat tento test v zveřejněných testech

TS 408 – Hodnocení testu

<i>Vstupní podmínky</i>	Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořena; test je vytvořený; alespoň jedna otázka je přidána k testu; test je zveřejněný
<i>Testovací data</i>	Název testu: LAG zápočtový test Byl test ohodnocen uživatelem: Ne

Kroky testu	<ol style="list-style-type: none"> 1. Uživatel vybere veřejný test 2. Klikne na tlačítko <i>líbí se</i> nebo <i>nelíbí se</i>
Očekávaný výsledek	Test je ohodnocený a zvýší se počet kladných nebo záporných hodnocení

TS 409 – Přidání veřejného testu do záložek	
Vstupní podmínky	Uživatel je přihlášený do systému; téma/podtéma je vytvořené; alespoň jedna otázka je vytvořená; test je vytvořený; alespoň jedna otázka je přidána k testu; test je zveřejněný; test není v záložkách uživatele
Testovací data	Název testu: LAG zápočtový test
Kroky testu	<ol style="list-style-type: none"> 1. Uživatel vybere veřejný test 2. Klikne na tlačítko <i>Přidat test do záložek</i>
Očekávaný výsledek	Test je přidán do záložek a zobrazí se uživateli v uložených testech

Reporty

TS 501 – Nahlášení příspěvku	
Vstupní podmínky	Uživatel je přihlášený do systému; test je zveřejněný
Testovací data	<p>Název testu: LAG zápočtový test</p> <p>Důvod nahlášení: Porušení práva duševního vlastnictví</p> <p>Spatně zodpovězená otázka: Jsou testy užitečné pro nalezení chyb v systému?</p>
Kroky testu	<ol style="list-style-type: none"> 1. Uživatel vybere veřejný test 2. Vybere otázku 3. Klikne na tlačítko <i>Vytvořit report a vybere důvod</i> 2. Potvrdí vytvoření tlačítkem <i>Vytvořit</i>
Očekávaný výsledek	Otázka je nahlášená. Zobrazí se administrátorovi v seznamu nahlášených otázek

Příloha C. Výsledné ankety testování

Uživatel 1.

1. *Využili byste aplikaci TestQuiz pro přípravu na zkoušky a učení se?*

Ano, velmi přínosná aplikace.

2. *Měli jste během splnění úkolů nějaké problémy? Pokud ano, uveďte úkol, problém a rekomendaci.*

Při zveřejnění test se mizí ze záložky s testy, takže se nedá zveřejnit a sdílet jeden ten samý test, je potřeba ho vytvořit znovu.

Uživatel 2.

1. *Využili byste aplikaci TestQuiz pro přípravu na zkoušky a učení se?*

Ano.

2. *Měli jste během splnění úkolů nějaké problémy? Pokud ano, uveďte úkol, problém a rekomendaci.*

Není na začátku jasné rozdíl mezi “Test” a “Topic”.

Uživatel 3.

1. *Využili byste aplikaci TestQuiz pro přípravu na zkoušky a učení se?*

Ano.

2. *Měli jste během splnění úkolů nějaké problémy? Pokud ano, uveďte úkol, problém a rekomendaci.*

Při sdílení testu je potřeba vyplňovat jméno uživatele, není možnost vybrat z existujících.

Uživatel 4.

1. *Využili byste aplikaci TestQuiz pro přípravu na zkoušky a učení se?*

Ano, určité pro mě je mnohem lepší než Quizlet. Oceňuji možnost přidání obrázků. Je to užitečné.

2. *Měli jste během splnění úkolů nějaké problémy? Pokud ano, uveďte úkol, problém a rekomendaci.*

Na velké problémy jsem nenarazil. Bylo trochu matoucí najít veřejné testy, jelikož jsem očekával že budou ve složce testy, ale ve skutečnosti veřejný test se da najít ve složce “Discovery”.

Uživatel 5.

1. *Využili byste aplikaci TestQuiz pro přípravu na zkoušky a učení se?*

Ano.

2. *Měli jste během splnění úkolů nějaké problémy? Pokud ano, uveďte úkol, problém a rekomendaci.*

Nebylo by spatně v samotném testu vizuálně rozdělit otázky a odpovědi.

Příloha D. Seznam odkazů

Git repositář:

TQ-frontend – <https://github.com/KseniiaBashkova/TQ-frontend>

TQ-backend - <https://github.com/KseniiaBashkova/TQ-backend>

Swagger:

Swagger UI -

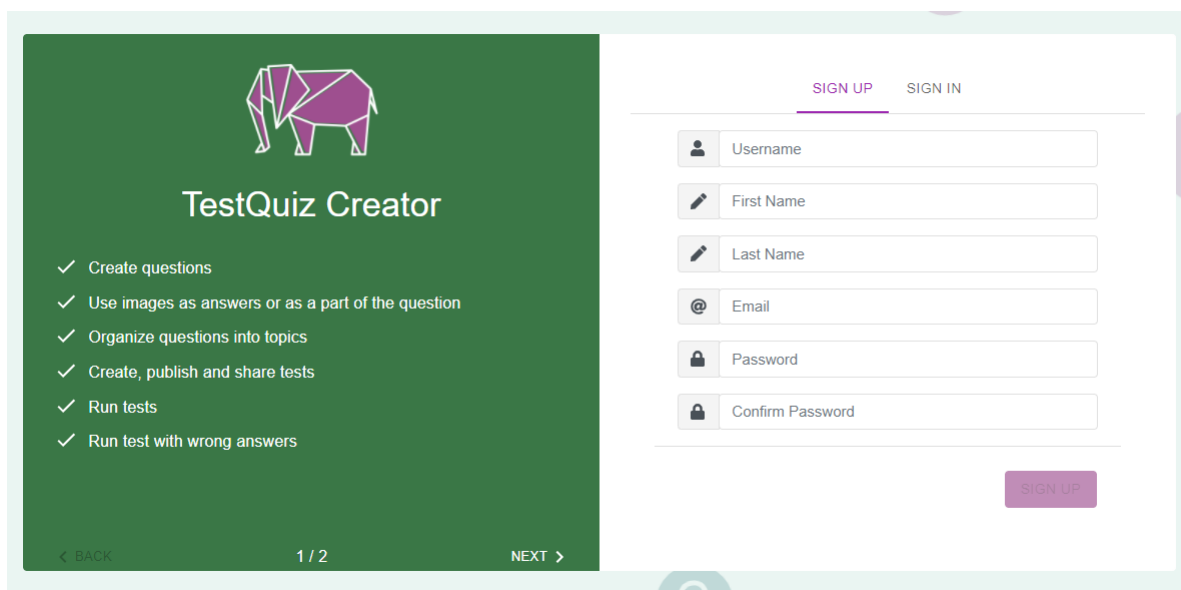
[https://thawing-shelf-20861.herokuapp.com/swagger-ui/index.html?configUrl=/api-docs/swagger-config#/.](https://thawing-shelf-20861.herokuapp.com/swagger-ui/index.html?configUrl=/api-docs/swagger-config#/)

Heroku:

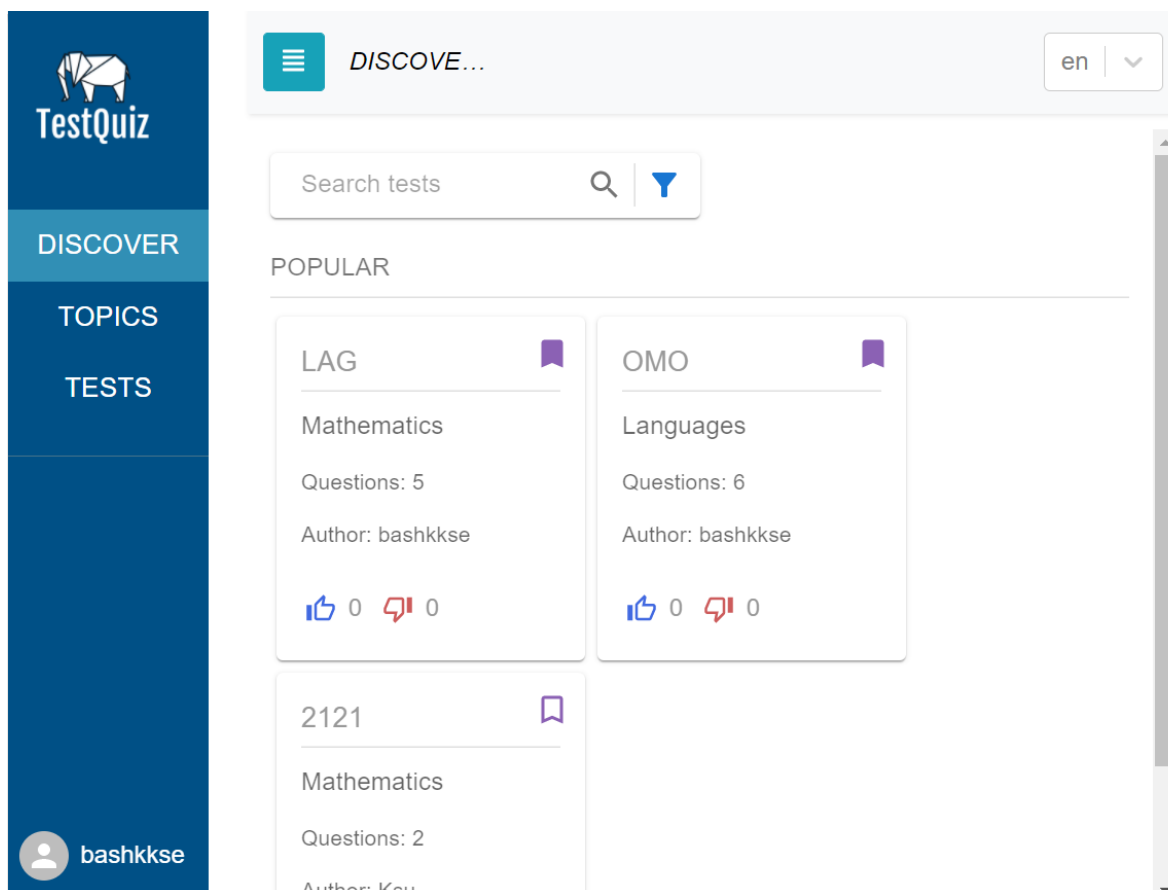
Serverová část - <https://test-quiz-application.herokuapp.com/sign>

Serverová část - <https://test-quiz-application.herokuapp.com/sign>

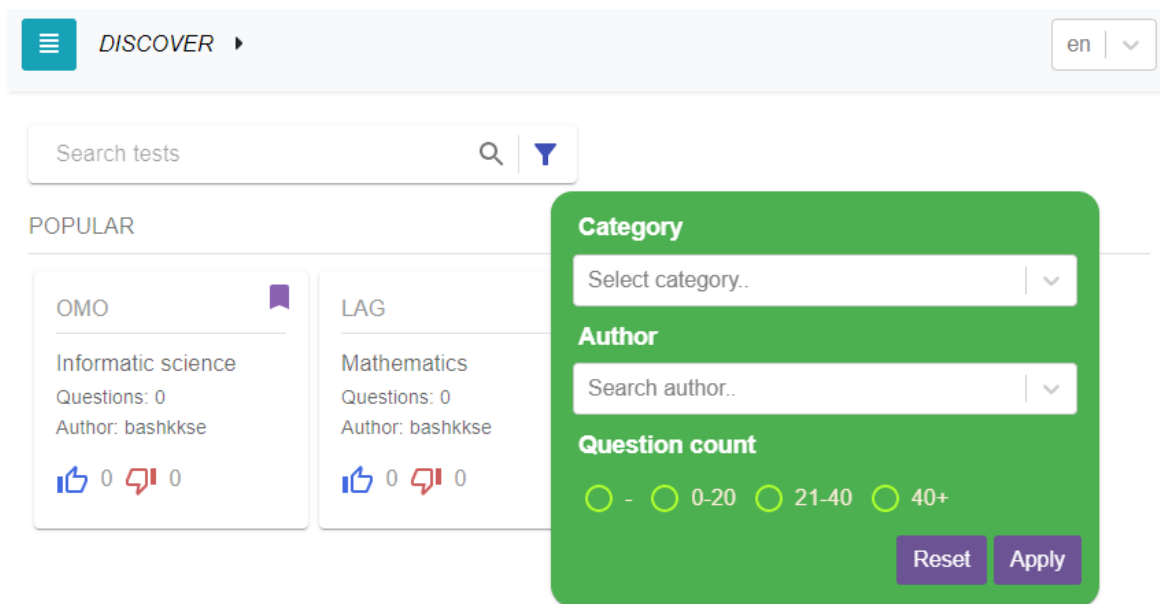
Příloha E. Snímky uživatelského rozhraní



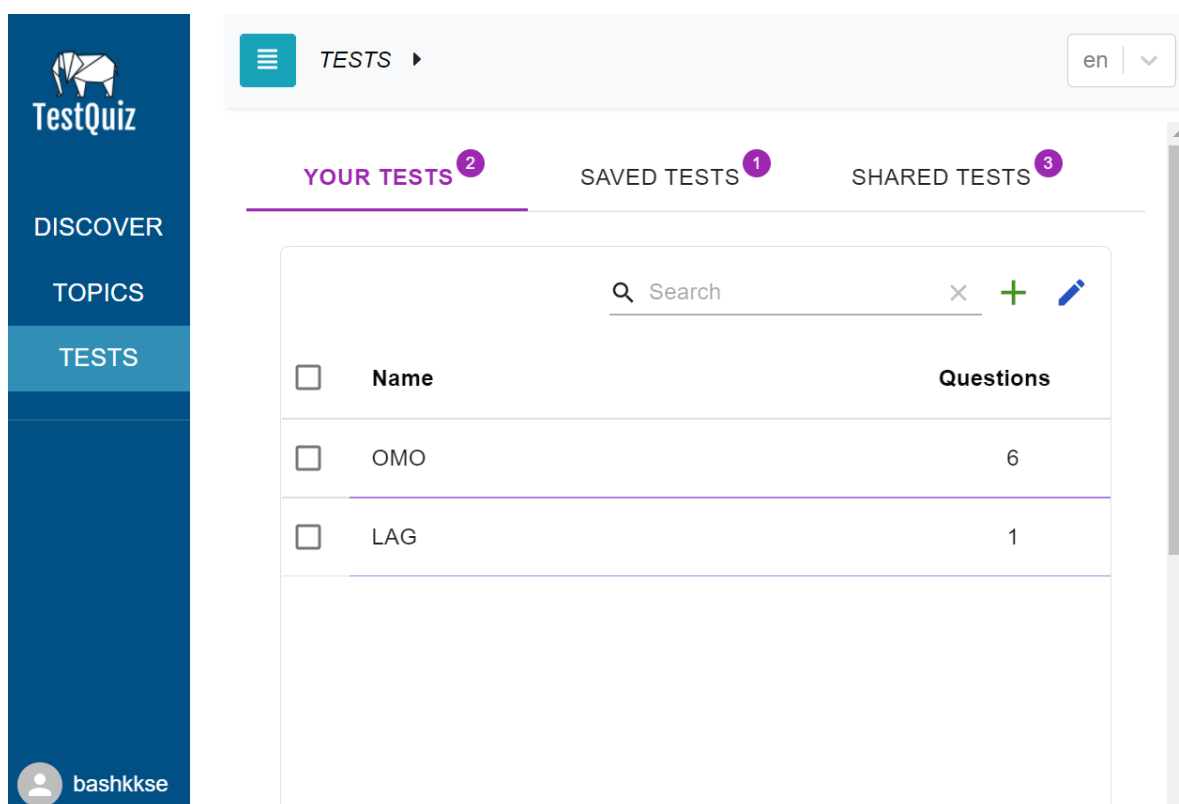
Obr. 15. Webová aplikace. Přihlašovací stránka



Obr. 16. Webová aplikace. Domovská stránka



Obr. 17. Webová aplikace. Filtrace veřejných testů



Obr. 18. Webová aplikace. Zobrazení vytvořených testů

Create question
✕

QUESTION

At' A je čtvercová reálná matice. Vyberte pravdivé tvrzení.

[UPLOAD](#)

ANSWERS

+

<input checked="" type="checkbox"/> ✕ Pokud je A regulární, pak je A <u>diagonalisovatelná</u>	<input checked="" type="checkbox"/> ✕ Pokud je A <u>diagonalisovatelná</u> , pak je A singulární
UPLOAD	UPLOAD
<input checked="" type="checkbox"/> ✕ Matice A má nulové vlastní číslo právě tehdy, když je A singulární	<input checked="" type="checkbox"/> ✕ Pokud má A mílové vlastní číslo, pak je A regulární

[CREATE](#)

Obr. 19. Webová aplikace. Vytvoření otázky

View question
✕

QUESTION

v R^3 je $\text{span}(\{a_1; a_2\})$, pro

ANSWERS

<input checked="" type="checkbox"/> ✕ přímka	<input checked="" type="checkbox"/> ✕ bod
<input checked="" type="checkbox"/> ✕ rovina procházející počátkem se směrem $(a_1; a_2)$	<input checked="" type="checkbox"/> ✕ Celá množina R^3

CLOSE

Obr. 20. Webová aplikace. Nahlédnutí otázky

TestQuiz

DISCOVER

TOPICS

TESTS

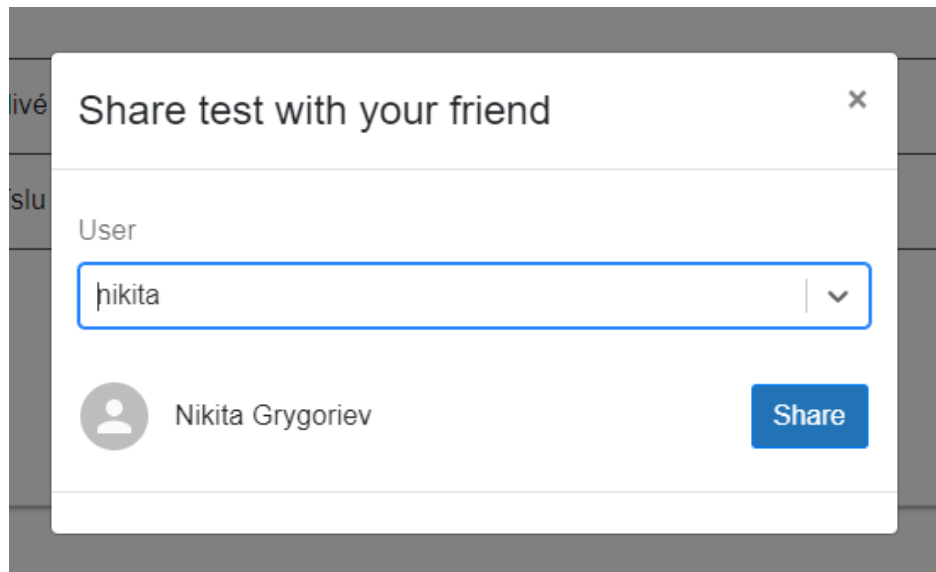
TESTS ▶ LAG ▶
en | ▼

▶ START
▶ SHARE
▶ PUBLISH

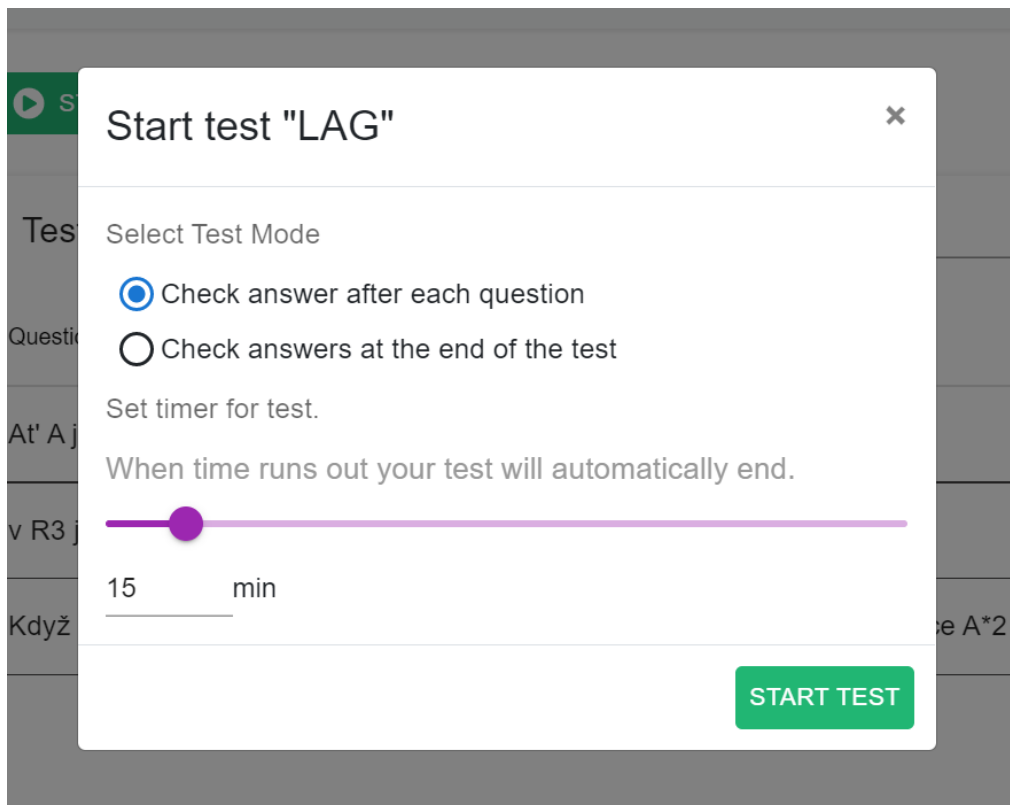
Test Question
🔍 Search ✕ +

Questions	Actions
At' A je čtvercová reálná matice. Vyberte pravdivé tvrzení.	🗑️
v R^3 je $\text{span}(\{a_1; a_2\})$, pro	🗑️
Když determinant matice A větší nebo roven číslu 5, pak determinant matice A^2	🗑️

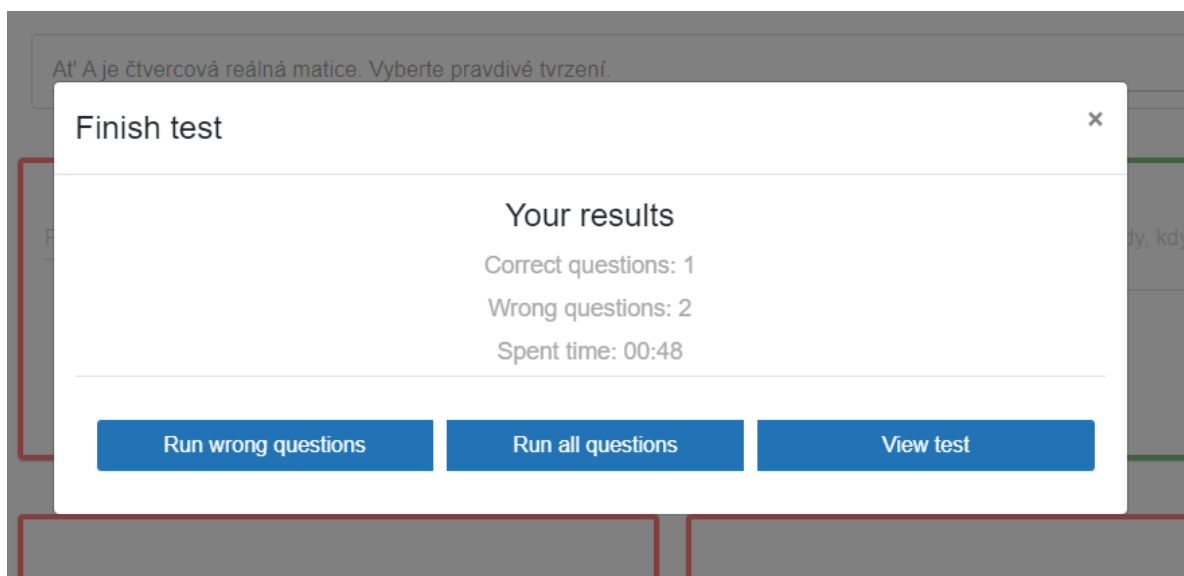
Obr. 21. Webová aplikace. Zobrazení testu



Obr. 22. Webová aplikace. Sdílení testu



Obr. 23. Webová aplikace. Spouštění testu



Obr. 24. Webová aplikace. Nahlédnutí výsledků testu