



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	System pro testování znalostí matematiky
Student:	Jan Fara
Vedoucí:	Ing. Filip Glazar
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

Cílem práce je realizace softwarového řešení na podporu testování znalostí matematiky žáků základních a středních škol.

Při realizaci postupujte podle těchto kroků:

1. Analyzujte potřeby cílové skupiny uživatelů.
2. Na základě analýzy proveďte vhodný softwarový návrh.
3. Při návrhu řádně diskutujte budoucí funkčnost aplikace s potenciálními uživateli aplikace.
4. Před vlastní implementací realizujte prototyp uživatelského rozhraní v rozsahu vyplňování testu, který následně otestujete na použitelnost.
5. Implementujte za použití vhodných technologií první verzi aplikace.
6. Implementaci podrobte vhodným testům. Vhodnost testů řádně podložte.
7. Pokuste se zajistit nasazení aplikace pro budoucí použití.
8. Zhodnoťte možnosti rozšíření aplikace.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 25. prosince 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

System pro testování znalostí matematiky

Jan Fara

Katedra softwarového inženýrství

Vedoucí práce: Ing. Filip Glazar

15. května 2019

Poděkování

Tímto bych chtěl poděkovat Ing. Filipu Glazarovi, vedoucímu mé práce, za cenné rady, konzultace, trpělivost a čas, který mi v průběhu vypracování bakalářské práce věnoval. V další řadě bych chtěl poděkovat Mgr. Marii Zálešákové, která mi věnovala svůj čas a byla ochotná pomoci zorganizovat testování aplikace ve škole. Dále také Mgr. Markétě Drhové, která se podílela na testování aplikace ve škole. Děkuji také všem učitelkám, které mi poskytly informace do této práce, jmenovitě Mgr. Ladislava Srpová, Mgr. Monika Průchová, Mgr. Marie Zálešáková. Mé díky také patří cvičící předmětu BI-DPR Ing. Elišce Šestákové, která mi svými připomínkami hodně pomohla. Závěrem bych rád poděkovat všem, kteří mě při tvorbě této práce podporovali, především své rodině a všem blízkým.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 15. května 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Jan Fara. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Fara, Jan. *Systém pro testování znalostí matematiky*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019. Dostupný také z WWW: (<http://www.prijimackysnadno.cz>).

Abstrakt

Tato bakalářské práce usnadňuje studentům základních škol přípravu na přijímací řízení z matematiky. Aplikace studentům umožňuje zkoušet si přijímací řízení ze sedmé a deváté třídy nanečisto. Studenti si mohou spustit test, který obsahuje stejná témata a velice podobné příklady, jaké je čekají u oficiální zkoušky. Program je ve formě responzivní webové aplikace, která běží na většině webových prohlížečů.

Klíčová slova matematika, základní škola, přijímací řízení, online test, Java, Spring, React, Redux, PostgreSQL

Abstract

This bachelor thesis makes the preparation for the Math Entrance Examinations easier for students. The application enables the students to practise Entrance Admissions in the seventh and ninth class. Students can generate the test which contains the same topics and similar arithmetical problems that are a part of the Entrance Examination. The program is in a responsive form of the web application which runs on most web browsers.

Keywords math, primary school, entrance examination, test online, Java, Spring, React, Redux, PostgreSQL

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza	5
2.1 Existující řešení	5
2.2 Průzkum ve školách	11
2.3 Anketa	13
2.4 Shrnutí	14
3 Softwarový návrh	15
3.1 Požadavky na systém	15
3.2 Výběr platformy	23
3.3 Prototyp uživatelského rozhraní	23
3.4 Databázový model	28
3.5 Použité technologie	28
4 Implementace	37
4.1 Server	37
4.2 Klient	40
4.3 Testování	45
4.4 Nasazení	46
5 Uživatelské testování	49
5.1 Průběh testování	49
5.2 Anketa	50
5.3 Zpětná vazba	50
Závěr	53
Literatura	55
A Seznam použitých zkratk	61

B Instalace	63
B.1 Spuštění	63
B.2 Vývoj	64
C Obrázky	65
D Ukázka výsledné aplikace	69
E Obsah přiloženého CD	79

Seznam obrázků

2.1	Webová stránka – <code>matematika.cz</code>	7
2.2	Webová stránka – <code>statniprijimackz.cz</code>	8
2.3	Webová stránka – <code>zkousky-nanecisto.cz</code>	9
2.4	Webová stránka – <code>to-das.cz</code>	10
3.1	Diagram případů užití – přihlášený a nepřihlášený uživatel	21
3.2	Prototyp uživatelského rozhraní – domovská stránka	26
3.3	Prototyp uživatelského rozhraní – vyplňování testů – začátek	26
3.4	Prototyp uživatelského rozhraní – vyplňování testů – průběh	27
3.5	Prototyp uživatelského rozhraní – vyplňování testů – vyhodnocení	27
3.6	Angular vs. React vs. Vue	30
4.1	Uživatelské rozhraní – popis hlavních komponent	42
4.2	Ukázka změny v menu u přihlášeného a nepřihlášeného uživatele	43
4.3	Uživatelské rozhraní – administrátor – příklady	44
4.4	Architektura systému	47
5.1	Výsledky ankety – hodnocení aplikace	51
5.2	Ukázka změny v zápisu znaku krát	51
C.1	Případy užití – administrátor aplikace	66
C.2	Databázový model	67
D.1	Úvodní obrazovka	69
D.2	Pokyny k testu	70
D.3	Ukázka vyplňování testu	70
D.4	Ukázka vyhodnocení testu	71
D.5	Reportování příkladu	71
D.6	Přihlášení – špatné jméno nebo heslo	72
D.7	Uživatel – statistika testů	72
D.8	Uživatel – přehled testů	73
D.9	Admin – nastavení testu	73
D.10	Admin – seznam kategorií	74
D.11	Admin – seznam kategorií – sdílení s jinými třídami	74
D.12	Admin – seznam příkladů	75

D.13 Admin – zobrazení reportovaného příkladu	75
D.14 Admin – přidávání příkladů – 1. část	76
D.15 Admin – přidávání příkladů – 2. část	76
D.16 Admin – editování LaTeX zkratk	77
D.17 Admin – výpis všech uživatelů	77

Seznam tabulek

2.1	Shrnutí nabízených služeb na přijímací řízení	14
3.1	Pokrytí případu užití	22
3.2	HTTP metody	32
3.3	REST úrovně	34

Seznam zdrojových kódů

3.1	Ukázka HTTP požadavku na server	32
3.2	Ukázka HTTP odpovědi od server	33
3.3	Ukázka formátu JSON	35
4.1	Server – konfigurace databáze	39

Úvod

Přijímací řízení na střední školy je jeden z klíčových milníků za cestou ke vzdělání a pro mnoho studentů je to vůbec první velká zkouška. Proto v rámci přípravy, dle mého názoru, mnoho studentů ocení možnost vyzkoušet si test nanečisto a třeba i sledovat svoje průběžné zlepšení. Konkrétně matematika je jeden z předmětů, který by studenti měli zvládnout, pokud se chtějí dostat na svou preferovanou střední školu. Forma přijímacího testu z matematiky je vždy podobná — test obsahuje přibližně šestnáct příkladů z několika kategorií. Typy příkladů se často opakují.

V dnešní době je mnoho služeb, které lidem pomáhají, zpoplatněné a je tomu i tak u přijímacího řízení. Některé finančně slabší rodiny nemohou dítěti tyto služby dopřát a proto přijímací řízení nemusí být v tomto směru spravedlivé. Na internetu lze nalézt velké množství aplikací, které slouží k procvičování matematiky, pokrývající látku od základní po vysokou školu. Žádná z nich se ale nezaměřuje právě na procvičení testů na přijímací řízení pro základní školy.

Z těchto důvodů jsem si vybral toto téma pro bakalářskou práci. Hlavní přínos aplikace je umožnit všem studentům připravit se na přijímací řízení rovnoprávně.

Práce je rozdělena na pět kapitol. Na úvod v kapitole 1 jsou rozebrány cíle práce. Kapitola 2 se zabývá analýzou uživatelských potřeb, průzkumem dostupných aplikací na internetu a rozebráním možností příprav, které studenti mají. Dále je v této kapitole popsán průzkum ohledně přípravy na přijímací řízení se studenty a průzkum s učiteli ohledně aplikací, které používají v rámci výuky nebo které se studentům doporučují na přípravu.

Kapitola 3 se zabývá návrhem aplikace. V první řadě jsou zde definovány požadavky na funkčnost aplikace. Také je zde uveden výběr platformy a první návrh uživatelského rozhraní. Dále je v této kapitole zmíněn databázový model a jsou zde rozebrány použité technologie, které jsou při implementaci použity.

ÚVOD

Poté je v kapitole 4 rozebrán vývoj aplikace od programování klientské a serverové části, následného otestování až po nasazení aplikace do provozu.

Závěrečná kapitola 5 je věnována vyzkoušení aplikace na studentech základních škol, včetně zpětné vazby ohledně funkčnosti a používání aplikace.



Cíl práce

Cílem této práce je vytvořit aplikaci, která studentům základních škol prostřednictvím internetu umožní opakovaně si vyzkoušet obdobný test, který je bude čekat u oficiálního přijímacího řízení. Uživatelé si budou moct test vygenerovat, vyplňovat a následně vyhodnotit. Po vyhodnocení testu uživatelé uvidí, na jaké příklady odpověděli správně a na jaké chybně.

Test se bude skládat z šestnácti příkladů, stejně jako oficiální test. Příklady budou vybírány z jednotlivých kategorií, které se v oficiálním testu nachází.

Aplikaci mohou používat přihlášení i nepřihlášení uživatelé. Uživatelům, kteří jsou přihlášení, se test po vyhodnocení uloží pod jejich osobní profil a bude si tak moct prohlédnout historii testů a sledovat své zlepšení, které bude zobrazeno grafem.

O obsah aplikace se bude starat administrátor. Administrátorem je uživatel s potřebným oprávněním. Toto oprávnění mám aktuálně pouze já. Pokud by se chtěl někdo o aplikaci starat, mohu ho také pověřit rolí administrátora. Pro snadnou správu materiálů bude administrátor používat grafické rozhraní aplikace, které mu lehce umožní přidat, upravit či smazat příklad nebo kategorii.

Příklady do systému bude administrátor zadávat v jazyce \LaTeX . Administrátor aplikace může být ale i osoba, která tento jazyk neovládá. Z tohoto důvodu bude mít aplikace vlastní \LaTeX editor, díky kterému vytváření příkladů bude snadné i pro ty, co \LaTeX neumí.

Analýza

V této kapitole budou zmíněny možnosti, které studenti mají pokud se chtějí připravovat na přijímací řízení na střední školy. Dále je zde rozebráno jak se studenti připravují a jaký software učitelé studentům doporučují na procvičování a nebo využívají v rámci výuky.

2.1 Existující řešení

Pro zmapování konkurenčních aplikací byl použit vyhledávač Google, do kterého byla zadána následující slovní spojení:

- přijímací zkoušky na střední školu,
- přijímací zkoušky na gymnázium,
- přijímací zkoušky na SŠ test online,
- přijímací zkoušky na SŠ nanečisto.

V této kapitole jsou rozebrány některé z webových stránek, které se zabývají matematikou a přípravou na přijímací zkoušky z matematiky. V této části kapitoly jsou zmíněny ty, které byly nejlépe dohledatelné a nabízely služby zabývající se přípravou studentů na přijímací řízení ať v podobě online přípravy či doučování s lektorem.

2.1.1 Scio

Scio je nejznámější společnost, která se specializuje na přípravu studentů k přijímacímu řízení nejen na střední školy, ale také na vysoké školy. Scio nabízí hned několik variant přípravy. [1]

— Zakoupení tištěných materiálů

1. *Sada testů* – tato varianta obsahuje 10 tištěných testů a videorozbor s podrobným řešením u dvou testů.
2. *Cvičebnice* – kniha příkladů, kterou si student může vypracovat. Tato varianta není určena pro sedmé třídy.
3. *Praktický průvodce přijímačkovým rokem* – souhrn informací o středních školách, průběhu testu, rad ohledně přípravy a souhrn všech termínů přijímacích zkoušek.

— Vyzkoušení si testu nanečisto

Tato varianta studentům umožňuje vyzkoušet si test nanečisto. Test má stejnou formu jako oficiální test u přijímacího řízení. Jedná se o akci, která se koná hromadně v předem daný termín na konkrétním místě.

— Přípravné kurzy

Kurzy, které jsou vedené profesionálními lektory v daný čas a v konkrétní učebně. Kurz není určen pro studenty sedmé třídy.

— Online přípravné kurzy

Kombinace dvou následujících balíčků je určena pro efektivní přípravu prostřednictvím internetu.

1. *Testuje mě to* – deset online testů s automatickým vyhodnocením. Řešení u jednoho z testů je multimediálně rozebráno.
2. *Učí mě to* – sto lekcí pro pátou třídu a sto třicet lekcí pro devátou třídu. Každá lekce je zakončena krátkým testem.

Všechny materiály a příklady, které jsou studentovi poskytnuty, jakoukoliv výše uvedenou službou, jsou kvalitně zpracovány. Na druhou stranu veškeré tyto služby jsou zpoplatněné a ne každý si je může dovolit. Například test nanečisto studenta vyjde přibližně na 1 000 Kč. Pokud si student chce vyzkoušet test nanečisto bez finančních prostředků má zde dostupný pouze jeden ukázkový test. Webové stránky jsou přehledné a uživatelsky přívětivé.

2.1.2 Cermat

Cermat je společnost, která pořádá oficiální přijímací zkoušky na střední školy. Na svých stránkách každý rok zveřejňuje aktuální ilustrační test. U každého testu je k dispozici i klíč se správným řešením. Celkově si tedy studenti mohou vyzkoušet přibližně šest testů. [2]

2.1.3 matematika.cz

Tento web obsahuje řadu matematických témat pokrývající učivo od základní školy až po vysokou školu. Každé téma obsahuje několik příkladů, na kterém je problematika srozumitelně vysvětlena. Tato stránka je vhodná spíše pro studenty, kteří chtějí látku pochopit, ale už se příliš nehodí na procvičování příkladů. Webové stránky jsou spíše přehledné, ale nachází se na nich reklamy, viz obrázek 2.1.[3]

The screenshot shows the homepage of matematika.cz. At the top, there are navigation tabs for different sections: STROJIRENSTVI.CZ, KONSTRUKTER.CZ, 3D-TISK.CZ, MATEMATIKA.CZ (highlighted), and NOVAMEDIA.CZ. Below the tabs is the main header with the logo 'Matematika.cz tady to pochopíš :-)' and the text 'zš sš vš fórum'. A search bar is located on the right side of the header.

The main content area is titled 'Matematika pro základní školy'. It features a list of categories and sub-topics:

- Kategorie: Základní škola**
 - Sčítání a odečítání
 - Násobení
 - Násobení
 - Násobení na papíře
 - Násobení záporných čísel
 - Dělení čísel
 - Mnohočleny
 - Dělení mnohočlenů
 - Lomené výrazy
 - Zlomky
 - Vytýkáání
 - Procenta
 - Zaokrouhlování
 - Trojčlenka
 - Společná práce
- Kategorie: Funkce**
 - Funkce
 - Co je to funkce
 - Funkce více parametrů
 - Skládání funkcí
 - Graf funkce
 - Vlastnosti funkce
 - Definiční obor
 - Inverzní funkce
 - Sudá a lichá funkce
 - Lineární funkce
 - Mocniny

On the right side, there is an advertisement for 'STROJIRENSTVI.CZ' magazine, titled 'ODEBÍREJTE EXKLUZIVNÍ TIŠTENÝ MAGAZIN PRO VŠECHNY STROJÁŘE VÍCE ZDE'. Below the ad is a vertical menu with dropdown arrows for various topics: Základní škola, Funkce, Rovnice a nerovnice, Čísla, Geometrie, Kombinatorika, Analytická geometrie, Statistika, Důkazy, Ostatní, Maturita, Analýza, Lineární algebra, Informatika, and Kryptografie a šifrování.

At the bottom right, there is a small social media widget for 'Matematika.cz' with 5 likes and a 'Poslat zprávu' button.

Obrázek 2.1: Webová stránka – matematika.cz

2. ANALÝZA

2.1.4 statniprijimacky.cz

Další možnosti, které mohou studenti k přípravě na přijímací zkoušky využít, nabízí stránka statniprijimacky.cz. Stránka obsahuje archiv oficiálních testů z minulých let. Mimo jiné se studenti mohou podívat, jak vypadalo zadání a záznamový arch testů. Webové stránky jsou jednoduše a přehledně zpracovány, ale obsahují velké množství reklam, jak lze vidět na obrázku 2.2. [4]

The screenshot shows the website statniprijimacky.cz. At the top, there is a navigation menu with links: STÁTNÍ PŘIJÍMAČKY, PŘIJÍMAČKY Z ČEŠTINY, PŘIJÍMAČKY Z MATEMATIKY, and PŘÍPRAVNÉ KURZY. Below the navigation is a large banner for "Státní přijímačky" with a background image of a city and a portrait of a man. The main content area is titled "Přijímačky z matematiky 2018" and features a large blue box with the text "STÁTNÍ PŘIJÍMAČKY Z MATEMATIKY 2018". To the right of this box is a text block titled "Zadání a klíče správných odpovědí pro přijímací testy z matematiky na střední školy 2018" with a description of the content. Below this is a list of links for various math tests, such as "Přijímací zkoušky - test z matematiky pro čtyřleté obory 2018 1. termín". The right side of the page contains several advertisements, including one for "B.i. Sensor" and "Pečujte o své oči" by "alza.cz". At the bottom right, there is a Facebook link and a "VYZKOUŠEJ" button with a "PŘIJÍMAČKY NA SŠ NANEČISTO" logo.

Obrázek 2.2: Webová stránka – statniprijimacky.cz

2.1.5 zkousky-nanecisto.cz

Tato společnost se zabývá přípravou studentů k přijímacím zkouškám. Studentům jsou zde nabídnuty dvě možnosti přípravy. Lekce pro menší skupinu studentů nebo test nanečisto. Nevýhoda obou variant je, že se akce nedají domluvit individuálně, ale vždy se konají na konkrétním místě a v konkrétním čase. Navíc jsou tyto akce dostupné pouze v Praze. Webová stránka má starší design a přehledná zrovna také není, stránka neobsahuje žádné reklamy, viz. obrázek 2.3. [5]

Ve srovnání se společností Scio, tato společnost nepůsobí profesionálním dojmem a to především kvůli nepřehledným a designově zastaralým webovým stránkám. Na druhou stranu nabízí cenově dostupnější služby. Například za test nanečisto zaplatí student při registraci na místě 300 Kč, což je přesně jedna čtvrtina z ceny za stejné služby u společnosti Scio.

The screenshot shows the website 'ZKOUSKY NANEČISTO'. The navigation bar includes links for 'Jak si vybrat', 'Časté dotazy', 'ROZVRH', 'Kontakt', 'Přihlásit se', and 'Košík 0 Kč'. Below the navigation are circular buttons for grade levels: 3. třída, 4. třída, 5. třída, 6. třída, 7. třída, 8. třída, 9. třída, maturanti, soutěže, and prázdniny. The main heading is 'Příprava na přijímací zkoušky pro žáky 7. tříd' with a sub-heading 'Výuka Praha:'. The content is organized into several sections:

- Zkoušky nanečisto v Praze:** Vyzkoušejte si, jak byste na zkouškách dopadli ve srovnání s jinými žáky. Matematika, český jazyk, angličtina a všeobecné znalosti. (Srovnání, Otevřené úlohy, Výklad)
- Dlouhodobý kurz matematika, český jazyk, angličtina:** Intenzivní pravidelná příprava na zkoušky od října do dubna. Veškerá potřebná látka. (Dlouhodobá příprava, Výklad)
- Angličtina pro žáky 7. třídy:** Zábavné a hravé formy výuk angličtiny. (Dlouhodobá příprava, Výklad)
- Matematická pohotovost:** Doučování matematiky v Praze. Probíráme co Vás zrovna trápí. Pomůžeme s úkoly. Připravíme na písemku... (Výklad)
- Anglický kemp pro mladé studenty:** Příměstský tábor s výukou angličtiny nadíjným mluvčím a českou lektorkou pro mladé studenty ve věku 12-15 let. (Letní prázdniny)
- Letní kreativní kempy:** V létě pořádáme příměstské tábory plné her a soutěží. Poďte s námi v létě rozvíjet svou kreativitu zábavnou formou! (Letní prázdniny)
- Učebnice a elektronické materiály:**
 - Naše učebnice:** Nabídka našich učebnic a sbírek testů pro 7. třídu. Matematika, český jazyk a testy všeobecných znalostí. (Tiskné materiály, Dlouhodobá příprava, Otevřené úlohy, Uzavřené úlohy)
 - Balíčky se studijními materiály:** 24 balíčků elektronických materiálů, které Vás postupně provedou veškerou potřebnou látkou z matematiky, českého jazyka, všeobecného přehledu a angličtiny. (Dlouhodobá příprava, Elektronické materiály)
- Co musíte umět ke zkouškám:** Matematický seriál tvoří 38 kapitol. Každá kapitola je jedno základní téma vysvětlené jasně a přehledně na několika příkladech. (Dlouhodobá příprava, Elektronické materiály, Tiskné materiály)
- Testy z obecných studijních předpokladů "OSP":** Pro úspěšnost není ani tak důležitý věk, důležitější jsou logika a cit pro čísla, slova, text. Vyhodnocení pomocí příloženého klíče. (Uzavřené úlohy, Elektronické materiály)
- Přijímací zkoušky z různých škol - matematika - vzorové vypracování:** Nabízíme řešené písemné práce z matematiky z osmiletých gymnázií, originální práce z různých škol a z různých let přímo z přijímacích zkoušek. (Uzavřené úlohy, Elektronické materiály)
- E-learning, online testy a soutěže:**
 - Online testy:** Online "klikací" testy pro rychlé procvičení a okamžitě získání výsledků. Matematika, český jazyk a všeobecné znalosti. (Srovnání, Uzavřené úlohy)
 - Zkoušky nanečisto přes internet:** Pro ty co se nemohou zúčastnit našich výuk v Praze nabízíme alternativu "přes internet". Testy vyplníte doma a pošlete našemu lektorovi k opravě. (Otevřené úlohy)
 - LOM Letní olympiáda matematiky:** Soutěž je určena všem žákům 7. tříd, kteří si rádi započítají a mají v oblíbě složitější úlohy. (Srovnání)

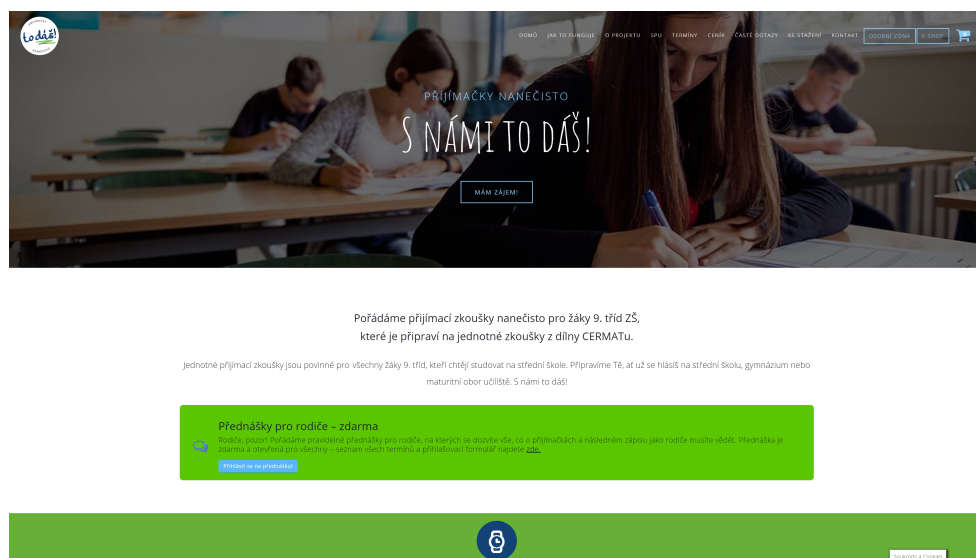
The footer contains contact information: 'Kontakty: Neváhejte nám zavolat 702 147 931', 'Naše další projekty: Jazyková škola Březinka, Obchod Zkoušek nanečisto - online prodej našich učebnic', and 'O nás: V médiích, Kaniera'.

Obrázek 2.3: Webová stránka – zkousky-nanecisto.cz

2. ANALÝZA

2.1.6 to-das.cz

Společnost, která se zabývá přípravou studentů devátých tříd na přijímací řízení. Akce organizují téměř každou sobotu v době před přijímacím řízením. Forma přípravy je rozdělena na dvě části psaní testu a následná oprava doplněná konzultací jednotlivých příkladů. Služba je zpoplatněna 580 Kč při včasné registraci a platbě on-line. Webové stránky působí přehledně a profesionálně zpracované, viz obrázek 2.4. Nevýhoda je, že se veškeré akce organizují pouze v Praze. [6]



Obrázek 2.4: Webová stránka – to-das.cz

2.2 Průzkum ve školách

Kromě analýzy aplikací, které byly vyhledány na internetu, je také zajímavé, co používají sami studenti k přípravě na přijímací řízení. Dále je užitečné zjistit, zda nějaký program používají učitelé. Program, který by sloužil jako podpora výuky při hodinách a nebo k zadávání domácích úloh či pouze doporučený k procvičování matematiky.

2.2.1 Průzkum s učiteli

Informace do této části kapitoly poskytly tři paní učitelky, Mgr. Marie Zálešáková, Mgr. Ladislava Srpová a Mgr. Monika Průchová.

2.2.1.1 Mgr. Marie Zálešáková

Paní učitelka Zálešáková učí matematiku dvanáct let na základní škole, z toho jedenáct let na ZŠ Černošice. Pro zpestření výuky matematiky používá webovou aplikaci **matematika.in**. [7]

Aplikaci vyvíjí neziskové sdružení **Matematika.in z.s.** Aplikace umožňuje online procvičování matematiky z každé třídy a pokrývá téměř celou látku, kterou si mohou studenti procvičit i pomocí matematických her. Každé téma obsahuje 5 příkladů. Příklady jsou jednoduššího typu a slouží spíše pro naučení základních principů v daném tématu. [8]

2.2.1.2 Mgr. Ladislava Srpová

Paní učitelka Srpová učí matematiku na základní škole od roku 2000. Aktuálně vyučuje na ZŠ Dobřichovice. Žákům pro procvičování matematiky doporučuje dvě webové aplikace. První z nich je **onlinecviceni.cz**. Tato aplikace umožňuje studentům si vyzkoušet počty z různých kapitol, všech tříd formou online testů které lze vyhodnotit [9]. Každé téma má však vždy stejné příklady, kterých není mnoho.

Další je **prikladyzmatematiky.cz** což je také webová aplikace. Na této aplikaci si studenti mohou najít zadání příkladů z různých kapitol včetně kapitoly, která je věnována přijímacímu řízení [10].

Paní učitelka si také vytvářela vlastní DUM (Digitální učebnicové materiály) prostřednictvím SMART Notebook od společnosti **AV Media**.

Jeden čas paní učitelka pro zpestření výuky využívala program **DRILL & SKILL**. Jedná se o placenou webovou aplikaci, která umožňuje učitelům vytvářet různé cvičení, testy a domácí úkoly pro žáky. [11, 12]

2.2.1.3 Mgr. Monika Průchová

Paní učitelka Průchová učí matematiku na základní škole 31 let, aktuálně vyučuje na Základní škole Příbram VIII. Pro přípravu studentů na střední školy dříve využívala učebnici: *Sbírka úloh z matematiky pro základní školu – František Běloun*, nyní využívá řadu publikací:

- Didaktis: *Testy 2019 z matematiky pro žáky 9. tříd ZŠ*
- Computer Media: *Přijímací zkoušky nanečisto – Matematika (9. ročník ZŠ)*
- Fortuna: *Testy pro deváťáky k jednotným přijímacím zkouškám – Matematika*
- Gaudetop: *Tvoje přijímací zkoušky 2019 na střední školy a gymnázia: MATEMATIKA*
- Fraus: *Matematika 9. ročník – K přijímačkám s nadhledem*

Paní učitelka Průchová pro vytváření testů používá aplikaci **Alf**. Tento program slouží pro vytváření online testů a lze používat po zakoupení jeho licence. Cena licence se odvíjí podle funkcí, který test bude umět. Například nejlevnější licence je zdarma a nejrozsáhlejší stojí 15 590 Kč. [13]

Program, který paní učitelka dětem doporučuje na online procvičování je **onlinecviceni.cz**, který je již zmíněný v kapitole 2.2.1.2. Pro zpestření výuky využívá programy od **Terasoft, a.s.**, **SILCOM Multimedia, s.r.o.**. Tyto programy umožňují procvičování matematiky prostřednictvím počítačů ve formě různých her a testů. Programy jsou zpoplatněny pořizovací cenou, která se pohybuje řádově kolem 1 000 Kč. [14, 15, 16]

2.2.2 Průzkum s žáky

Pro získání nezávislých dat byl průzkum proveden na několika odlišných školách. Konkrétně ZŠ Černošice, ZŠ Všenory, ZŠ Dobřichovice, ZŠ a MŠ na Dlouhém Lánu, ZŠ Praha-Radotín. Pro splnění GDPR jsou žáci anonymizováni.

Všichni z tázaných studentů odpověděli, že nevyužívají žádnou aplikaci pro přípravu na přijímací řízení. Většina studentů se připravuje pomocí ukázkových testů či starých testů z minulých let. Všichni studenti by aplikaci, která jim generuje různé testy a následně je vyhodnotí v rámci přípravy využili. Půlka z nich si pro lepší přípravu a vyzkoušení nových příkladů platí služby od společnosti Scio.

2.3 Anketa

Jako další část analýzy byl zvolen dotazník ve formě Google Forms, který byl určen pro získání více dat z jedné cílové skupiny uživatelů. Dotazník vyplnilo 37 dětí na Základní Škole Černošice.

2.3.1 Otázky ankety

Dotazník je složen ze tří otázek. Jako první bylo důležité zjistit, zda se studenti vůbec na střední školu hlásí. Další dvě otázky dotazníku jsou určeny pouze pro studenty, kteří se hlásí na střední školu. Jejich cílem bylo zjistit, jak se studenti připravují na přijímací řízení na střední školy.

1. Hlásíte se na střední školu?
 - Ano
 - Ne

2. Připravujete se na přijímací řízení?
 - Ano
 - Ne
 - Ještě ne, ale budu

3. Jaké materiály používáte na přípravu na přijímací řízení?
 - Učebnici
 - Oficiální zveřejněné testy
 - Hledám si příklady na internetu
 - Prostřednictvím Scio
 - Otevřená odpověď

2.3.2 Výsledek ankety

Na střední školu se z 37 studentů hlásí 15, což je přibližně 40 %.

Ze studentů, kteří se hlásí na střední školu se na přijímací řízení hodlá připravit 93 % studentů, z toho 71 % se připravuje už teď 10. 4. 2019, což je přibližně 14 dní do přijímacího řízení. Většina (71 %) z nich se připravuje pomocí oficiálních zveřejněných testů, které jsou buď ukázkové nebo zkouškové z předchozích let. Přesně polovina studentů si hledá příklady na internetu a nebo k přípravě používá učebnice. Přibližně 42 % studentů využívá služby od společnosti Scio. Někteří studenti se také připravují pomocí přípravných kurzů, které pořádá škola, na kterou se hlásí.

2.4 Shrnutí

Studenti mají k dispozici široký výběr služeb. Mohou si najít vysvětlené témata, vyzkoušet si test nanečisto jak online, tak i písemně. Dále mají k dispozici staré testy, které byly u přijímacích řízeních v minulých letech a nebo se mohou připravit pomocí zakoupených materiálů.

Kdyby se ale tyto služby shrnuly, tak se buď jedná o placené služby nebo si studenti mohou přečíst vysvětlení určitého matematického tématu. Ukázkových testů zdarma je na internetu poměrně málo. Aplikace, která by studentům umožnila online test, který by si mohli vyzkoušet vícekrát s různými příklady, však žádná dohledatelná není. Služby jsou shrnuty v tabulce 2.1.

Podle získaných informací se tedy žádný program, který by sloužil k přípravě studentů na střední školy nepoužívá ani v rámci výuky. Ale jak studenti, tak učitelé by ocenili, kdyby měli nějakou možnost, která by studentům umožnila online, zdarma a s elektronickým vyhodnocením vyzkoušet si opakovaně přijímací řízení nanečisto.

Společnost	Web	Služby	Placené
Scio	Moderní Přehledný	Materiály Zkouška nanečisto Lekce s lektorem	Ano
Cermat	Přehledný	Přehled starých testů	Ne
matematika.cz	Plný reklam	Postupy řešení příklad	Ne
statniprijimacky.cz	Plný reklam	Přehled starých testů	Ne
zkousky-nanecisto.cz	Starší Nepřehledné	Online test Materiály Zkouška nanečisto Lekce s lektorem	Ano
to-das.cz	Moderní Přehledný	Zkouška nanečisto	Ano

Tabulka 2.1: Shrnutí nabízených služeb na přijímací řízení

Softwarový návrh

V této kapitole jsou prvně specifikovány požadavky, které má systém zvládat. Druhá část je zaměřena na vybrání vhodné podoby aplikace. V další části je navržená část uživatelského rozhraní jako wireframe. V neposlední řadě je zmíněn návrh databáze. Závěrečná část se zabývá výběrem technologií, které jsou při vývoji aplikace využity.

3.1 Požadavky na systém

Funkce systému mohou být rozsáhlé, a proto je důležité je předem vymezit. Níže jsou vypsány požadavky na systém, které jsou rozděleny do dvou kategorií – **funkční** a **nefunkční**. Funkční požadavky definují, co všechno bude aplikace uživateli poskytovat za služby. Zatímco nefunkční definují kvalitu software, například požadavky jako je výkon, bezpečnost, dostupnost, spolehlivost atd. [17]

3.1.1 Funkční požadavky

— FP1 Zkoušení testů

Primární funkcí této aplikace je, aby si uživatelé mohli zkoušet testy nanečisto. Uživatel si bude moct:

1. vybrat si z jaké třídy se hlásí,
2. přečíst si pokyny k průběhu testu,
3. vygenerovat nový test,

4. vyplňovat test,
5. nechat si test vyhodnotit.

Celou tuto akci bude moct uživatel neomezeně opakovat.

— **FP2 Prohlížení materiálů**

Uživatelé si budou moct procházet jednotlivé třídy, jejich kategorie a příklady, které dané kategorie obsahují.

— **FP3 Prohlížení svých hotových testů**

Každý přihlášený uživatel bude mít k dispozici přehled všech svých hotových testů, které kdy vyplnil.

— **FP4 Správa uživatelského profilu**

Přihlášený uživatel si bude moct prohlížet svůj profil a měnit své osobní údaje: email, jméno, příjmení a heslo.

— **FP5 Správa materiálů**

Administrátor aplikace bude mít k dispozici následující funkce:

1. vytvářet nové kategorie pod libovolnou třídu,
2. sdílet jednotlivé kategorie s jinými třídami,
3. přejmenovat libovolnou kategorii,
4. smazat libovolnou kategorii,
5. vytvářet nové příklady pod jednotlivé kategorie,
6. upravovat jednotlivé příklady,
7. smazat libovolný příklad.

— **FP6 Správa uživatelů**

Administrátor se bude moct podívat na ostatní registrované uživatele a prohlížet si jejich výsledky. Bude mít práva ke změně hesla a role libovolného uživatele. Tedy bude mít možnost libovolného uživatele pověřit administrátorskou rolí, ale také mu ji odebrat.

— **FP7 Nastavení obsahu testů**

Každý test pro jednotlivou třídu bude obsahovat příklady z různých kategorií. Administrátor aplikace bude moct nastavit, kolik bude test obsahovat příkladů z daných kategorií.

— **FP8 Reportování příkladů**

V příkladu se může objevit samozřejmě chyba, ať z důvodu špatné odpovědi nebo překlepu v zadání, či neodpovídajícímu příkladu na dané téma. Proto uživatelé budou mít možnost nahlásit libovolný příklad s patřičným odůvodněním.

— **FP9 Správa reportovaných příkladů**

Administrátor má za úkol spravovat obsah, proto by měl kontrolovat, zda uživatelé nemají s něčím problém. Konkrétně zda nenastala chyba v některém z příkladů. Proto bude mít k dispozici seznam všech reportovaných příkladů. Ke každému novému reportovanému příkladu se zobrazí notifikace, která administrátora upozorní. Jelikož administrátor chybu vyřeší nebo se uživatel při reportu mylí, bude moci odstranit jednotlivé reporty u příkladů, aby byl seznam přehlednější a nezabýval se něčím, co je již vyřešené.

— **FP10 Správa L^AT_EX zkratek**

Příklady se zadávají v jazyce L^AT_EX. Protože administrátor aplikace může být někdo, kdo tento jazyk neumí, musí mít i tak možnost vytvářet příklady. Proto administrátor bude mít při vytváření příkladu k dispozici L^AT_EX editor, obsahující základní L^AT_EX příkazy, které pro vytvoření příkladu potřebuje. Administrátor má také možnost zobrazit si tento editor, upravovat jednotlivé zkratky, přidávat nové a mazat je.

3.1.2 Nefunkční požadavky

Nefunkční požadavky jsou doplněk k funkčním požadavkům, definují spíše pasivní chování aplikace.

— **NP1 Výkon**

Jedna z častých věcí, která uživatele odrazuje od používání aplikace je dlouhá odezva. Proto byla stanovena, doba odezvy u žádostí. 90 % žádostí musí mít maximální možnou odezvu 3 sekundy a žádný z žádostí nesmí trvat déle než 10 sekund.

— **NP2 Bezpečnost**

Zabezpečení aplikace je velice důležité kvůli možnému úniku citlivých informací. Bohužel 100% bezpečnost nelze zaručit nikdy. Typů útoků na aplikace je mnoho a postupem času vznikají nové. Proto ani nejde odhadnout, jaký typ útoku útočník zvolí.

Jsou však možnosti, jak bezpečnost vylepšit. Jedna z nich je použití HTTPS komunikace mezi klientem a serverem, viz kapitola 3.5.7 Tato komunikace je šifrovaná pomocí protokolu SSL a TLS. Tyto šifry jsou asymetrické a jsou založené na výměně klíčů. Odposlouchávat tuto šifrovanou komunikaci je velice obtížné.

Další možnost je hashovat hesla uživatelů před vložením do databáze. Hash je funkce, která převádí vstupní posloupnost bitů na posloupnost bitů pevné délky. Hashování je založeno na problému faktorizace. Pokud

by data z databáze unikla, hesla nejsou jasně čitelná a pro jejich přesný znění je třeba je nejdříve dešifrovat, což může být problém. [18]

Kvůli různým uživatelským oprávněním je třeba řešit autentizaci a autorizaci. Ne každý uživatel má k dispozici všechny obsah a funkce, které systém nabízí.

— NP3 Multiplatformita

Aplikace není určena pro konkrétní uživatele, ani zařízení a proto by měla být spustitelná na většině zařízení a operačních systémech.

— NP4 Responsibilita

Aby byla aplikace použitelná na většině zařízení, je třeba aby se uměla přizpůsobit různým velikostem obrazovek.

3.1.3 Účastníci

Aplikaci používají uživatelé s různými rolemi, konkrétně nepřihlášený uživatel, přihlášený uživatel a administrátor aplikace. Účastníci představují všechny možné role uživatelů.

3.1.3.1 Nepřihlášený uživatel

Nepřihlášený uživatel je takový uživatel, který navštíví a používá aplikaci, aniž by měl svůj uživatelský účet, pod kterým by byl přihlášený. Tito uživatelé mohou procházet jednotlivé třídy, kategorie a příklady. Mohou si také vygenerovat, vyplnit a vyhodnotit test. Tuto roli bude podle autorových předpokladů představovat většina uživatelů aplikace.

3.1.3.2 Přihlášený uživatel

Uživatel se stane přihlášený ve chvíli, kdy se přihlásí pod svým uživatelským účtem a nebo si ho založí. Přihlášený uživatel má stejné funkce jako nepřihlášený uživatel a navíc se mu ukládají vyhodnocené testy pod jeho profil, který si může spravovat.

3.1.3.3 Administrátor

Administrátor aplikace je uživatel s administrátorským pověřením. Tedy s rolí, která ho dělá administrátorem. Tato role je určena pouze pro správce aplikace, kteří budou spravovat její obsah. Administrátor může prohlížet, vytvářet, upravovat a mazat téměř jakýkoliv obsah, tedy kategorie, příklady, uživatele atd.

3.1.4 Případy užití

Případy užití rozšiřují funkční požadavky z hlediska účastníka a popisují možnou interakci mezi jednotlivými účastníky a aplikací. [19] Případy užití pro účastníky jako přihlášený a nepřihlášený uživatel lze vidět na obrázku 3.1.

— **PU1 Zobrazení tříd**

Uživatel si bude moct zobrazit všechny třídy.

— **PU2 Zobrazení kategorií**

Uživatel si bude moct zobrazit všechny kategorie, které jsou pod jednotlivými třídami.

— **PU3 Vytváření kategorií**

Uživatel s administrátorskou rolí bude moct vytvářet nové kategorie.

— **PU4 Upravování kategorií**

Uživatel s administrátorskou rolí bude moct upravovat kategorie.

— **PU5 Mazání kategorií**

Uživatel s administrátorskou rolí bude moct mazat kategorie.

— **PU6 Sdílení kategorií**

Uživatel s administrátorskou rolí bude moct sdílet obsah tříd, tedy kategorie s jinými třídami.

— **PU7 Zobrazení příkladů**

Uživatel si bude moct zobrazit všechny příklady, pod každou kategorií.

— **PU8 Vytváření příkladů**

Uživatel s administrátorskou rolí bude moct vytvářet nové příklady.

— **PU9 Upravování příkladů**

Uživatel s administrátorskou rolí bude moct upravovat příklady.

— **PU10 Mazání příkladů**

Uživatel s administrátorskou rolí bude moct mazat příklady.

— **PU11 Zobrazení testu**

Uživatel si bude moct zobrazit test, přečíst si pokyny k testu a kliknout na tlačítko spustit test.

— **PU12 Nastavení obsahu testu**

Uživatel s administrátorskou rolí bude moct nastavovat obsah testů aplikace, podle jednotlivých kategorií.

— **PU13 Vygenerování nového testu**

Uživatel si bude moct vygenerovat nový test.

— **PU14 Vyplňování testu**

Uživatel bude moct odpovídat na otázky ve vygenerovaném testu.

— **PU15 Vyhodnocení testu**

Uživatel si bude moct nechat vyhodnotit vygenerovaný test.

— **PU16 Zobrazení vyplněných testů**

Přihlášený uživatel si bude moct zobrazit všechny jeho v minulosti vyplněné testy.

— **PU17 Zobrazení profilu**

Přihlášený uživatel se bude moct podívat na svůj profil.

— **PU18 Editace profilu**

Přihlášený uživatel bude moct editovat svůj profil.

— **PU19 Reportování příkladů**

Uživatel bude moct reportovat libovolný příklad a zdůvodnit proč.

— **PU20 Zobrazení reportovaných příkladů**

Uživatel s administrátorskou rolí si bude moct zobrazit všechny reportované příklady.

— **PU21 Mazání reportovaných příkladů**

Uživatel s administrátorskou rolí bude moct mazat jednotlivé reportované příklady.

— **PU22 Mazání uživatelů**

Uživatel s administrátorskou rolí bude moct smazat registrovaného uživatele.

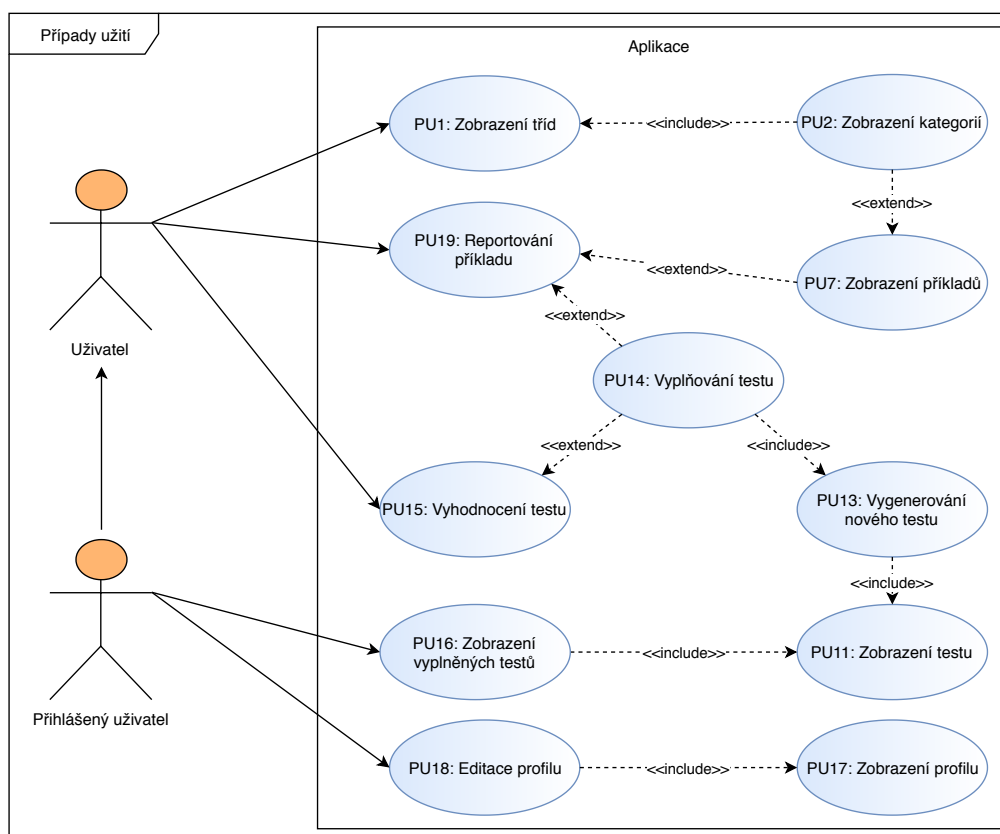
— **PU23 Upravování rolí uživatelů**

Uživatel s administrátorskou rolí bude moct měnit pověřit rolí administrátora jiného registrovaného uživatele a nebo mu tuto roli odebrat.

— **PU24 Obnovení hesla uživatelů**

Uživatel s administrátorskou rolí bude moct ostatním uživatelům změnit heslo.

- **PU25 Zobrazení uživatelů**
Uživatel s administrátorskou rolí si bude moct zobrazit všechny registrované uživatele.
- **PU26 Zobrazení \LaTeX zkratk**
Uživatel s administrátorskou rolí si bude moct zobrazit všechny \LaTeX zkratky.
- **PU27 Vytváření \LaTeX zkratk**
Uživatel s administrátorskou rolí bude moct vytvářet nové \LaTeX zkratky.
- **PU28 Upravování \LaTeX zkratk**
Uživatel s administrátorskou rolí bude moct upravovat \LaTeX zkratky.
- **PU29 Mazání \LaTeX zkratk**
Uživatel s administrátorskou rolí bude moct mazat \LaTeX zkratky.



Obrázek 3.1: Diagram případů užití – přihlášený a nepřihlášený uživatel

3.1.5 Pokrytí případu užití

V tabulce 3.1 lze vidět úplné pokrytí funkčních požadavků a případů užití.

	FP1	FP2	FP3	FP4	FP5	FP6	FP7	FP8	FP9	FP10
PU1		✓								
PU2		✓								
PU3		✓			✓					
PU4					✓					
PU5					✓					
PU6					✓					
PU7		✓								
PU8					✓					
PU9					✓					
PU10					✓					
PU11	✓									
PU12							✓			
PU13	✓									
PU14	✓									
PU15	✓									
PU16			✓							
PU17				✓						
PU18				✓						
PU19								✓		
PU20									✓	
PU21									✓	
PU22						✓				
PU23						✓				
PU24						✓				
PU25						✓				
PU26										✓
PU27										✓
PU28										✓
PU29										✓

Tabulka 3.1: Pokrytí případu užití

3.2 Výběr platformy

Jeden z nejdůležitějších softwarových rozhodnutí je výběr správné platformy na které aplikace poběží. Bylo zvažováno mezi provedení jako mobilní, webové a desktopové aplikace.

Aplikace má sloužit ke zkoušení testů a jednotlivé příklady mohou mít delší zadání. Proto i přes velkou popularitu mobilních aplikací je řešení jako mobilní aplikace pro tento problém nevhodné.

Jelikož aplikace nemá být paměťově náročná a je třeba aby běžela online, pak řešení jako desktopová aplikace by pouze odradila z důvodu potřebné instalace. Další nevýhodou tohoto řešení je omezení použití aplikace na jiných operačních systémech, než pro které byla vyvinuta.

Proto bylo zvoleno aplikaci implementovat jako webovou. Řešení v podobě webové aplikace je použitelné na mnoha operačních systémech a při patřičné responsivitě je toto řešení použitelné téměř na všech typech zařízení, včetně mobilních telefonů.

3.3 Prototyp uživatelského rozhraní

Pro snažší implementaci je vhodné graficky navrhnout část aplikace, kterou budou uživatelé nejčastěji využívat. V případě této aplikace se jedná o vyplňování testů.

Proto, aby uživatelé využili služby, které aplikace nabízí, je důležité, aby je měli dostupné na co nejméně kliknutí. Na každé stránce také musí být jasné, co se po uživateli chce a co konkrétní stránka nabízí. Jednotlivé stránky musí být přehledné a nesmí obsahovat příliš informací, které uživatel aktuálně nepotřebuje.

Při vyplňování testu je důležité, aby uživatelé věděli všechny potřebné informace které k vyplnění testu potřebují. Seznam příkladů musí být přehledný. Uživatelé musí být zřejmé, k jakým otázkám patří jaké odpovědi a jak na ně odpovídat.

Návrh byl realizován v aplikaci `draw.io` v podobě čtyř fází, konkrétně úvodní obrazovka, začátek testu, průběh testu a vyhodnocení testu.

3.3.1 Domovská stránka

Domovská stránka je první stránka, kterou uživatel uvidí, když vyhledá patřičnou doménu, viz obrázek 3.2. Jelikož je aplikace určena ke zkoušení online testů, je důležité aby uživatelé, kteří stránku navštíví hned věděli, kde si test mohou vyzkoušet.

Uživatelé zde mají dokonce dvě možnosti, jak se na test dostat. Jedna z nich je jít přes menu vlevo, kde si rozkliknou konkrétní třídu a v její nabídce zvolí možnost *Zkusit test*. Tedy jim k tomu stačí dvojitým kliknutím.

Druhá možnost je ještě jednodušší, uživatelé se na test dostanou pouze jedním kliknutím. Na domovské stránce každé třídy, která má možnost vyzkoušet test, je umístěn rámeček s popisem a tlačítkem *Zkusit test*. Toto tlačítko uživatele přesměruje na test pro danou třídu.

3.3.2 Začátek testu

Tato stránka je uživateli zobrazena před začátkem testu. Na stránce jsou důležité dvě věci, pokyny a tlačítko *Spustit test*, viz obrázek 3.3. Pokyny obsahují veškeré informace, které student potřebuje pro vyplnění testu:

1. Hodnocení příkladů

- Test obsahuje šestnáct příkladů.
- Každý správně zodpovězený příklad je ohodnocen 1 bodem.
- Špatná či nevyplněná odpověď je ohodnocena 0 body.

2. Čas

- Čas máte stejný jako na oficiální test, tedy 2 hodiny.
- Po zahájení testu se vám začne čas odpočítávat.
- Protože tento test slouží k přípravě, tak i po vypršení času můžete dále odpovídat na otázky a nechat si test vyhodnotit.

3. Průběh testu

- V testu jsou tři typy příkladů: příklady s otevřenou odpovědí a příklady s více možnostmi, buď s právě jednou správnou odpovědí a nebo s různým počtem správných odpovědí.
- Na otázky můžete odpovídat v libovolném pořadí.
- Svou odpověď můžete v průběhu testu kdykoliv změnit.
- Pokud vyplňujete příklad s otevřenou odpovědí, ujistěte se, že odpověď neobsahuje žádné jiné znaky včetně mezer.
- Pokud chcete u příkladu s otevřenou odpovědí zapsat zlomek například "tři pětiny" zapište ho jako "3/5".

4. Průběh testu

- Test vyhodnotíte kliknutím na tlačítko *Vyhodnotit*, které se nachází na stránce dole vpravo.
- Po vyhodnocení testu se vám zobrazí tabulka s úspěšností.
- Každý příklad, na který jste neodpověděli, a nebo zodpověděli špatně, bude zvýrazněn červenou barvou.
- Každý příklad, který jste zodpověděli správně, bude zvýrazněn zelenou barvou.

Kliknutím na tlačítko *Spustit test* se uživateli vygeneruje nový test.

3.3.3 Průběh testu

Po spuštění testu se uživateli vygenerují příklady, spustí se odpočet času a zobrazí se tlačítko *vyhodnotit*, viz obrázek 3.4. Příklady jsou tří typů:

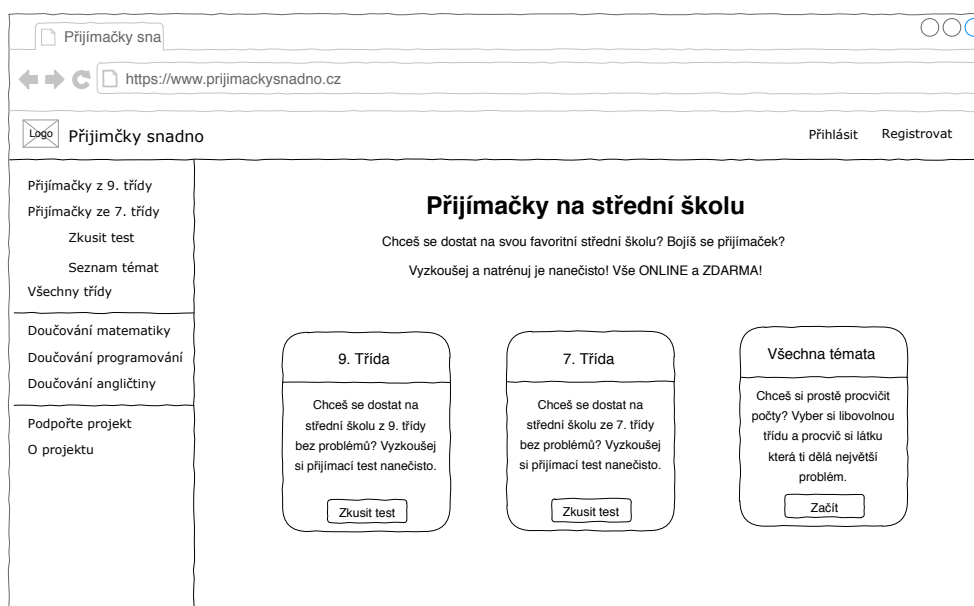
- příklady s otevřenou odpovědí,
- příklady s právě jednou správnou odpovědí,
- příklady s různým počtem správných odpovědí.

Uživatel může na příklady odpovídat v libovolném pořadí. Pokud uživatel vybere odpověď z nabídky, jeho řešení se zvýrazní vyplněním *checkboxu*. V případě příkladů s otevřenou odpovědí musí svou odpověď zapsat do *inputu* k danému příkladu. Svou odpověď uživatel v průběhu testu může libovolně krát změnit. Pokud uživatel klikne na tlačítko *vyhodnotit*, test se ukončí a vyhodnotí.

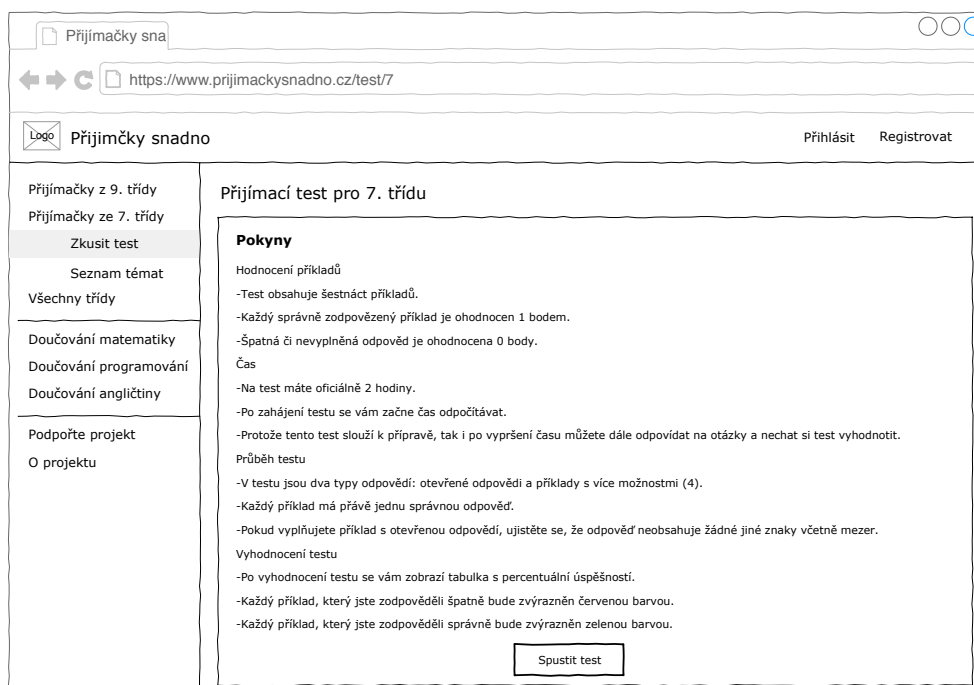
3.3.4 Vyhodnocení testu

V této fázi uživatel již nemůže odpovídat na otázky ani měnit své odpovědi. Příklady, které odpověděl špatně nebo na ně neodpověděl vůbec, se vybarví červeně a příklady, na které zodpověděl správně, se vybarví zeleně. Uživateli se také zobrazí tabulka s počtem zodpovězených otázek, správných odpovědí, procentuální úspěšností, která je zobrazena i graficky a procentuální úspěšností vůči všem vyplněným testům, viz obrázek 3.5.

3. SOFTWAREOVÝ NÁVRH

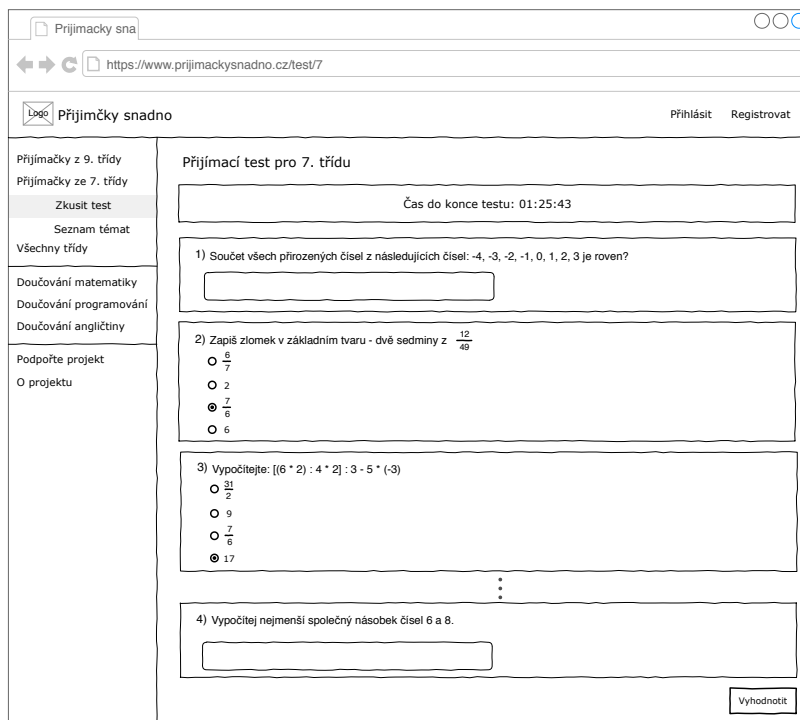


Obrázek 3.2: Prototyp uživatelského rozhraní – domovská stránka

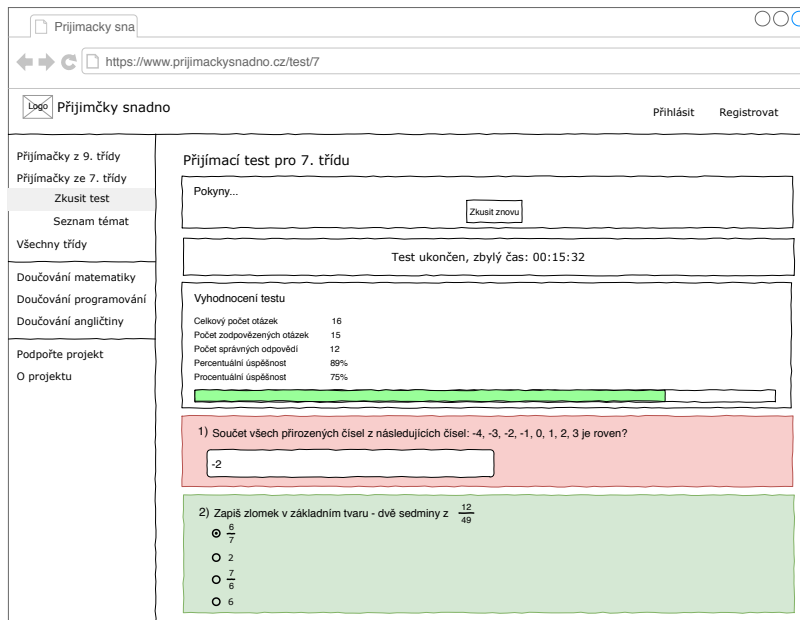


Obrázek 3.3: Prototyp uživatelského rozhraní – vyplňování testů – začátek

3.3. Prototyp uživatelského rozhraní



Obrázek 3.4: Prototyp uživatelského rozhraní – vyplňování testů – průběh



Obrázek 3.5: Prototyp uživatelského rozhraní – vyplňování testů – vyhodnocení

3.4 Databázový model

Aplikace umožňuje uživatelům zkusit testy, které se skládají z příkladů. Každý příklad má několik různých odpovědí a nachází se v určité kategorii. Jednotlivá kategorie patří do několika tříd, minimálně však do jedné. Všechny tyto data jsou uloženy v databázi.

Každý test je složen z několika náhodných příkladů, počet příkladů z každé kategorie, které budou v testu je nastavitelný a proto je nutné toto nastavení ukládat do databáze.

Vyhodnocený test se ukládá z důvodu statistiky, kterou mají uživatelé k dispozici. Navíc pokud se jedná o uživatele, který je přihlášen, test se mu uloží pod jeho profil aby měl přehled o všech svých vyplněných testech.

Dále je třeba si ukládat uživatele a jejich role, které určují, jaké má uživatel oprávnění. Pokud se jedná uživatele s administrátorskou rolí, má k dispozici přehled reportovaných příkladů, které ostatní uživatelé nahlásili a jsou také uloženy v databázi. Model celé databáze lze vidět na obrázku C.2 v příloze.

3.5 Použité technologie

V následujících částech této kapitole jsou popsány technologie, které byly zvoleny pro implementaci aplikace.

3.5.1 Java

Java je jeden z nejpoužívanějších programovacích jazyků na světě [20]. Jedná se o objektově orientovaný jazyk. Jeho velkou výhodou je multiplatformní využití a to, že je spustitelná kdekoliv, kde je nainstalovaný JDK (Java Development Kit). Java umožňuje vývoj různých aplikací: desktopové, mobilní, webové. [21]

3.5.2 Spring

Spring je open-source framework založený na Java pro snadný vývoj enterprise edition aplikací. První verze vznikla v roce 2003. [22] Spring je složen přibližně z 20 modulů. Tyto moduly jsou podle [23] rozděleny do 6 vrstev:

- **Core Container** – vrstva poskytuje základní funkcionalitu Springu, například Dependency injection, tedy anotaci `@Autowired`. Také nabízí Inversion of Control, což je nástroj, který slouží pro správu objektů od jejich vytvoření až po jejich zničení.
- **Data access/integration** – vrstva poskytuje operace s databázovým serverem a integrační vrstvy pro API (Application Programming Interface) objektově relačního mapování, jako například Hibernate, iBatis, JDO (Java Data Objects). Také umožňuje mapování Object/XML

(eXtensible Markup Language) pro JAXB (Java Architecture for XML Bindings), Castor, Xstream a XMLBeans.

- **Web** – vrstva se skládá ze 4 modulů, které umožňují základní webově orientované funkce a implementaci MVC (Model-view-controller) pro webové aplikace.
- **Aspect Oriented Programming and Instrumentation** – Jedna z nejdůležitějších vlastností Springu. Tato vrstva umožňuje oddělovat části kódů do jednotlivých aspektů. Tyto aspekty se pak dají snadno použít na jakýkoliv POJO (Plain Old Java Object) objekt. Dalo by se také říct, že AOP (Aspect Oriented Programming) říká, že konkrétní kód má být vykonán před nebo po nebo kolem daných míst v programu. Příklad AOP může být třeba anotace `@Transactional` která před zavoláním metody spustí `BEGIN` a po dokončení funkce `COMMIT / ROLLBACK`.
- **Test** – vrstva podporuje testování komponent pomocí JUnit a TestNG.

3.5.2.1 Spring Boot

Spring Boot umožňuje z aplikace vytvořit soubor typu `.jar`, což bylo kvůli spuštění aplikace na VPS (Virtual Private Server) využito. Spring Boot usnadňuje uživatelskou konfiguraci aplikace. [24]

3.5.3 JavaScript

Jedná se o multiplatformní jazyk, který lze spustit na mnoha různých operačních systémech jako je OSX, Windows, Unix a Android. Pro spuštění stačí téměř jakýkoliv webový prohlížeč. Jazyk byl navrhnut k automatizaci úloh na stávajícím systému a proto se mu říká skriptovací. JavaScript je také interpretovaný, to znamená, že pro jeho spuštění nemusí být kompilován. [25]

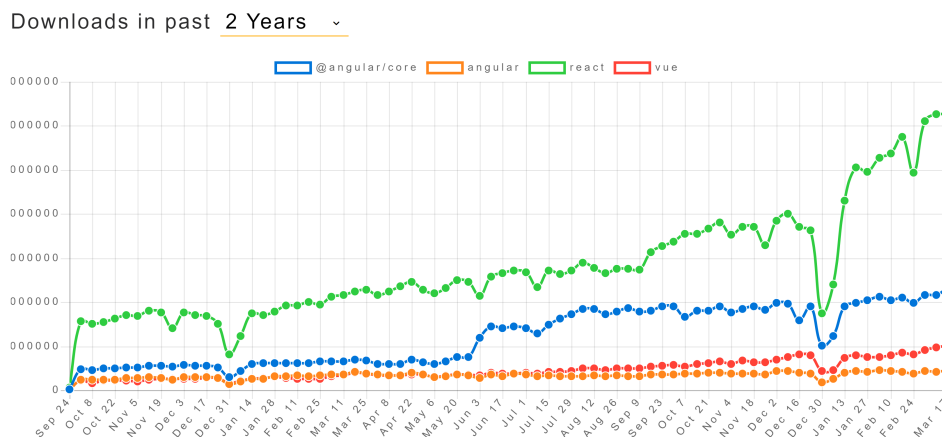
3.5.4 React

React, také známý jako `React.js`, je JavaScript knihovna pro vytváření uživatelského rozhraní. React byl představen v roce 2013 Facebook komunitou, která ho vyvíjí. Z pohledu MVC React reprezentuje View, tedy řeší pouze vzhledovou vrstvu. [26]

Toto rozhraní je rozděleno do jednotlivých komponent. Každá komponenta má vlastní funkce, vzhled a stav. Povinná je však jen jedna metoda `render()`, která vykresluje vzhled komponenty. Tato metoda se zavolá z počátku jednou a pak pokaždé, když se změní stav komponenty. To znamená, že po změně stavu se komponenta vykresluje znova.

Mohlo by se tedy zdát, že je tento přístup velice neefektivní. React využívá takzvaného stínování DOM (Document Object Model). DOM je objekově orientované aplikační rozhraní. Toto rozhraní díky své stromové struktuře

3. SOFTWAREOVÝ NÁVRH



Obrázek 3.6: Angular vs. React vs. Vue

za pomocí JavaScript umožňuje přístup k jednotlivým prvkům v HTML dokumentu, které tak lze klidně měnit, mazat nebo přidávat. Prvky jsou například tagy, elementy nebo jen obyčejný text. React si při první načtení stránky sestaví přesný obraz DOM, který si uloží do paměti. Po každém dalším vykreslení si React vytvoří nový DOM, který následně porovná s původním a změní pouze nutné části k vykreslení stránky. Ve finále se tedy změní třeba jen jeden již zmiňovaný prvek z celého dokumentu. díky tomu React umožňuje téměř úplnou abstrakci od DOM. [26, 27]

Při výběru jazyka na frontend bylo zvažováno mezi použitím Angular, Vue a právě již zmiňovaného React. Všechny tyto knihovny jsou velice populární. Jelikož všechny tyto technologie byly pro autora nové, rozhodlo se podle komunity, kterou měl výrazně největší React, viz obrázek 3.6 [28].

3.5.5 Redux

React je složen z mnoha komponent a každá komponenta může obsahovat stavy. Pokud se ale bude k jednomu stavu přistupovat z více komponent může to znamenat problém. Například, když se v jedné komponentě stav přejmenuje, už si nikdo nemusí pamatovat, kde všude je tento stav použit.

Proto byl v roce 2015 představen Redux, který je přímo pro React navrhnut. Redux je open-source JavaScript knihovna, která vychází ze starší architektury Flux. Hlavní úkol této knihovny je shromáždit stavy na jedno místo. Stavy jsou uloženy v jednom objektu ve stromové struktuře. Na tyto stavy jsou napojeny jednotlivé komponenty, které se překreslují, pokud se stav změní. [29]

Redux se nejčastěji používá v rozložení na tři různé konstrukce: store, reducer a actions. Každá tato konstrukce má na starosti konkrétní práci s daty.

Ve store jsou data uloženy. Actions představují funkce, které uživatel může volat. Každá tato funkce po zavolání provede patřičnou operaci a pošle data s typem akce reduceru. Reducer podle typu akce provede danou změnu v store [30, 31].

Pro vývoj webových aplikací za použitím Reduxu existuje rozšíření Redux DevTools [32] do vyhledávače Google Chrome. Toto rozšíření umožňuje vývojáři přehled nad všemi stavy. Umožňuje také možnost krokování, díky kterému lze vrátit akce, které vývojář provedl a provést je v jednotlivých krocích.

3.5.6 PostgreSQL

PostgreSQL je objektově relační databázový systém, který je licencován pod licencí BSD (Berkeley Software Distribution) nebo MIT (Massachusetts Institute of Technology), řadí se tedy mezi open source aplikace. [33] Databáze je podporovaná na mnoha operačních systémech macOS, Microsoft Windows a primárně Linux, pro který tato databáze původně vznikla. [34]

3.5.7 HTTP

HTTP (Hypertext Transfer Protocol) je nejpoužívanější internetový protokol. Tento protokol se používá ke komunikaci klient-server. Kde klient je například webový prohlížeč. Komunikace funguje tak, že klient odešle požadavek na VPS pomocí TCP/TLS protokolu, kde HTTP server žádost přepošle patřičnému aplikačnímu serveru. Aplikační server žádost zpracuje a odešle odpověď rovnou klientovi. Jako HTTP server se nejčastěji používá program Apache [35]. Komunikace může být pro větší bezpečnost šifrovaná, pak se jedná o HTTPS.

3.5.7.1 HTTP metody

HTTP umožňuje více typů metod a ty jsou znázorněny v tabulce 3.2. [36] V aplikaci byly použity metody GET, POST, DELETE.

3.5.7.2 Verze HTTP

První verze protokolu vznikla v roce 1991. z počátku tato verze nebyla nijak označována, později byla označena jako HTTP/0.9. Tato verze uměla jenom metodu GET a odpověď obsahovala pouze HTML stránku.

O pět let později, tedy v roce 1996, vyšla verze HTTP/1.0. Verze byla rozšířená o nové metody a umožňovala posílat jiný formát než jen HTML stránku. Zpráva měla hlavičku, která obsahovala metadata a status kód.

V roce 1997 vznikla verze HTTP/1.1, která je používána až do dnes. Verze umožňuje znovu použití připojení. Při komunikaci používala pipelining, má cache kontrolovací mechanismus a mnoho dalších rozšíření.

Nová verze HTTP/2.0 vznikla až v roce 2015. Hlavní přínos této verze je rychlost přenosu dat. Rychlost je zajištěna kompresí hlavičky. Soubor je

3. SOFTWAREOVÝ NÁVRH

GET	Metoda se využívá pro získání dat. Dotaz neobsahuje v těle žádné data
POST	Tato metoda se používá pro vytváření. Její tělo obsahuje data které mají být vytvořené
DELETE	Metoda určená pro mazání dat
PUT	Nahrává data na server.
HEAD	Tato metoda je podobná metodě GET s tím rozdílem, že předává pouze metadata, nikoli data v těle metody.
TRACE	Odesílá kopii získaného požadavku
OPTIONS	Tato metoda zjistí, jaké server poskytuje metody.
CONNECT	Spojení přes konkrétní port.

Tabulka 3.2: HTTP metody

pak v binárním formátu. Po jednom TCP spojení se může přenášet více požadavků. Server také může některá data posílat dřív, než o ně klient požádá, jedná se o data, u kterých server ví, že by je za nedlouho stejně poslal. Tato verze však funguje pouze na HTTPS.

Poslední verze vznikla v roce 2018. Pro komunikaci je namísto TCP/TLS protokolů v této verzi použit QUIC protokol, který je založen na protokolu UDP. [37]

3.5.7.3 Komunikace

Požadavek na server může vypadat jako v zdrojovém kódu 3.1

```
1 GET /grade/findAll HTTP/1.1
2 HOST: www.prijimackysnadno.cz
3
```

Zdrojový kód 3.1: Ukázka HTTP požadavku na server

První řádek obsahuje typ metody, cestu a verzi HTTP, viz tabulka 3.2. Druhý řádek obsahuje hlavičku. V ukázce je to doména, na kterou je dotazováno. Na serveru může běžet více domén, proto musí být zřejmé, se kterou má být komunikace uskutečněna. Další řádek je prázdný. Pokud by bylo potřeba posílat nějaké data, následovala by po prázdném řádku.

Klient si pomocí DNS (Domain Name Server) zjistí IP (Internet Protocol) adresu serveru, na kterou odešle dotaz přes TCP protokol. Typy odpovědí serveru jsou trojčíferná čísla a jsou rozdělena celkem do pěti kategorií podle začínající číslice.

1. **Odpovědi začínající jedničkou** jsou informační zprávy. Například odpověď 100 Continue říká, že je zatím vše v pořádku a že má klient pokračovat s dotazy.
2. **Odpovědi začínající dvojkou** symbolizují úspěch. Například odpověď 200 Ok říká, že bylo vše v pořádku.
3. **Odpovědi, které začínají na trojku** symbolizují přesměrování. Například kód 301 Moved Permanently znamená trvalé přesunutí. Server s tímto kódem posílá nové URL, na které se má být pro úspěch zeptáno.
4. **Odpovědi začínající čtyřkou** znamenají, že je chyba v požadavku. Nejznámější takový kód je asi 404 Not Found. Tento kód říká, že server nenašel dotazovanou URL adresu.
5. **Odpovědi, které začínají pětikou** říkají, že nastala chyba na straně serveru. Jedna z častých chyb je například 500 Internal Server Error. Tento kód značí že se server dostal do situace, kterou neumí řešit. [38]

Odpověď pak vypadá například jako v zdrojovém kódu 3.2.

```

1 HTTP/1.1 200 OK
2 Access-Control-Allow-Credentials: true
3 Access-Control-Allow-Headers: Origin, X-Requested-With, Content-
  ↪ Type, Accept
4 Access-Control-Allow-Methods: POST, GET, PUT, OPTIONS, DELETE,
  ↪ PATCH
5 Access-Control-Allow-Origin: http://www.prijimackysnadno.cz
6 Access-Control-Max-Age: 3600
7 Cache-Control: no-cache, no-store, max-age=0, must-revalidate
8 Content-Type: application/json;charset=UTF-8
9 Date: Mon, 11 Mar 2019 19:52:14 GMT
10 Expires: 0
11 Pragma: no-cache
12 Transfer-Encoding: chunked
13 X-Content-Type-Options: nosniff
14 X-Frame-Options: DENY
15 X-XSS-Protection: 1; mode=block

```

Zdrojový kód 3.2: Ukázka HTTP odpovědi od server

3.5.8 REST

REST (Representational state transfer) je architektura rozhraní, která byla navrhována a vyvíjena Roy Fieldingem, což je jeden ze spoluautorů HTTP. Představena byla v roce 2000. K realizaci se používá také HTTP protokol,

3. SOFTWAREOVÝ NÁVRH

Úroveň	Název vrstvy	Popis
0	The Swamp of POX	Přenos dat – nejčastěji HTTP
1	Resources	Každý zdroj má právě jeden koncový bod
2	HTTP Verbs	Název akce kterou provádíme (GET, POST, PUT, DELETE, ...) a stavový kód
3	Hypermedia Controls	Známe pouze koncový bod, který vrací veškeré data s patřičnými odkazy

Tabulka 3.3: REST úrovně

čemuž se pak říká RESTful. Leonard Richardson rozdělil REST do 4 úrovní, viz tabulka 3.3. [39]

REST API musí být bezstavové, tedy pro vykonání každého dotazu musí stačit data obsažená v tomto dotazu a nesmí záležet na stavech v programu či cookies. REST je orientován datově, nikoli procedurálně jako třeba SOAP. To znamená, že si klient a server předávají pouze data, nikoli akce či příkazy. Data mohou být reprezentována třeba ve formátech JSON nebo XML. [40]

3.5.9 MVC

MVC je softwarová architektura, která rozděluje program na tři základní nezávislé části. Tyto části spolu pouze komunikují. [41]

- **Model** v sobě uchovává veškeré data, která aplikace obsahuje, bussines logiku a pomocí API poskytuje data controller.
- **View** vrstva prezentuje data uživateli, tvoří tedy uživatelské rozhraní.
- **Controller** je vrstva mezi modelem a view, stará se tedy o komunikaci mezi těmito dvěma vrstvami. Na základě uživatelských vstupů rozhoduje o tom, jaký data z modelu budou zobrazeny a za pomoci jakého view.

3.5.10 JSON

JSON (JavaScript Object Notation) je formát, který umí přehledně reprezentovat data. Data lze reprezentovat jako proměnné, kterým je přiřazena hodnota, z těchto proměnných lze pak tvořit objekty a pole. [42]

Java i JavaScript umí s tímto formátem pracovat. Java ho umí převést na objekt a JavaScript běžně reprezentuje data v tomto formátu. JSON byl využit pro výměnu dat mezi klientem a serverem. Ukázka formátu JSON jako pole objektů, které reprezentují jednotlivé stupně na základní škole, viz zdrojový kód 3.3.

```
1  [
2  {
3      "id_grade": 3,
4      "name": "3. Trida"
5  },
6  {
7      "id_grade": 2,
8      "name": "2. Trida"
9  },
10 {
11     "id_grade": 1,
12     "name": "1. Trida"
13 }
14 ]
```

Zdrojový kód 3.3: Ukázka formátu JSON

3.5.11 Apache

Apache HTTP server je nejpoužívanější softwarový webový server [35], který funguje již od roku 1995. Tento server funguje na většině platformách, mimo jiné i na Linuxu. Jeho licence se řadí mezi permisivní. Tyto licence umožňují svobodné užívání software, požadují jen zachování autorství. Z těchto důvodů byl Apache použit v této práci.

Úkol tohoto webového serveru je vyřizovat HTTP požadavky od klienta. Server tedy dostane požadavek a podle URL ho přešle příslušnému aplikačnímu serveru, pro který je tento požadavek určený. [43]

3.5.12 Gitlab

Pro verzování a zálohování práce byl zvolen verzovací systém Gitlab. Tento systém se používá hlavně při práci ve více lidech na jednom projektu. v průběhu práce na bakalářské práci byl problém s notebookem, který přestal fungovat a proto byla tato volba oceněna. Díky tomuto systému nebyl problém stáhnout projekt na jiném počítači a pokračovat na něm. kromě záloh, byly také využity starší verze programu. Po větších úpravách přestal projekt několikrát fungovat nebo byl vymyšlen jiný způsob řešení daného problému a tak stačilo pouze stáhnout starší verzi.

Implementace

Cílem této kapitoly je stručně rozebrat samotný vývoj aplikace, jak jsou jednotlivé části aplikace navrženy a popsat, jak spolu komunikují. Dále jsou zmíněny důležité knihovny, které byly při implementaci využity.

4.1 Server

Serverová část má za úkol vyřizovat dotazy od klienta a obsahuje veškerou business logiku. Server dostává požadavky, na které následně odpovídá. Pro zpracování některých požadavků se také může dotazovat na databázi.

4.1.1 Vývoj

Pro vývoj enterprise aplikací je téměř samozřejmé použití vývojového prostředí tedy IDE, které práci velice usnadní. Pro implementaci serveru bylo zvoleno intellij IDEA od společnosti JetBrains. Toto vývojové prostředí také nabízí možnost navázat spojení s databází, což bylo využito k spouštění SQL skriptů.

Server vystavuje REST metody, která pak klient volá. Zvolená architektura klient-server nabízí možnost vyvíjet serverovou a klientskou část nezávisle, a proto může nastat situace, že server bude poskytovat metody, které klient ještě nemá naimplementované. Proto se hodí vyzkoušet správnou funkcionalitu metod, nezávisle na klientovi. Za tímto účelem byl využit program Postman, což je dektopová aplikace, který umí provolávat metody serveru. Při vývoji aplikace byl Postman využit na otestování téměř všech metod, které server poskytuje.

4.1.2 Architektura

Tato část kapitoly se bude věnovat architektuře serveru. Bude zde zmíněno, na jaké části je server navržen, jak ukládá data a komunikuje s databází. Důraz je tu také kladen na zabezpečení aplikace.

4.1.2.1 Entitní třídy

Entity jsou třídy, které reprezentují tabulky v databázi. Aby se třída stala entitní třídou používá se anotace `@Entity`. Pokud se třída jmenuje stejně jako tabulka v databázi, JPA namapuje tuto třídu na tabulku a není třeba přidávat další anotaci. Pokud se třída jmenuje jinak, je třeba přidat anotaci `@Table(name = "jmeno-tabulky")`, která podle jména namapuje třídu na tabulku v databázi. [44]

4.1.2.2 Repository

Repository je interface, který je rozšířen o `JpaRepository`. Tento interface obsahuje základní funkce pro komunikaci s databází. Například metody jako:

- `findAll()` nepřijímá parametr a vrací `List<E>` všech záznamů konkrétní entity v databázi.
- `save()` přijímá v parametru entitní třídu, pokud tato třída neobsahuje `id`, vytvoří nový záznam v databázi. Pokud by však třída obsahovala `id`, pak přepíše v databázi záznam s tímto `id`, funkce tedy slouží i pro úpravu záznamů v databázi. Funkce vrací vytvořenou, či upravenou entitní třídu.
- `delete()` smaže konkrétní záznam v databázi podle parametru `id`, který přijímá. Funkce nemá žádnou návratovou hodnotu.
- `findOne()` vrací konkrétní záznam v databázi, její parametr je `id` záznamu, který má být nalezen. [45]

4.1.2.3 Service

Service obsahuje veškerou business logiku. Funkce těchto tříd jsou volány ve většině případů controller a nebo jiné funkce v service. Tyto funkce také komunikují s repository. Service je třída s anotací `@Service`. [46]

4.1.2.4 Controller

Funkcí Controller je vystavit REST služby třetí straně. Tato třída je anotována `@RestController` a lze jí přidat anotaci `@RequestMapping("/cesta1")`, kde `cesta1` je adresa, pod kterou jsou tyto služby dostupné [47]. Každá funkce ve třídě controller je anotována `@RequestMapping(value="/cesta2",`

`methos=typ-metody`), kde `cesta2` udává adresu, pod kterou se tato konkrétní metoda nachází a typ metody je GET, POST, PUT, DELETE, atd. Jednotlivou metodu pak lze volat například jako `adresa-serveru/cesta1/cesta2`. [48]

4.1.2.5 Spring security

Spring Security je užitečný a uživatelsky přizpůsobitelný framework, který poskytuje autentizaci a autorizaci Java aplikací. Velice silnou stránkou tohoto framerowk je jeho snadné použití a přizpůsobení vlastním potřebám. [49]

Framework dokáže chránit aplikaci proti útokům jako je třeba session fixation, clickjacking, cross site request forgery, a další. [50] V aplikaci byl tento framework použit pro autentizaci uživatele a autorizaci požadavků.

4.1.3 Databáze

Pro uložení dat byla použita databáze PostgreSQL. Jedná se o relační databázi, jejíž entity jsou vymodelovány v serverové části jako Entitní třídy. Komunikace s databází je díky knihovně Postgresql a JPA velice jednoduchá. Konfigurace je umístěna v souboru `application-dev.properties`, viz zdrojový kód 4.1.

```

1 spring.datasource.url=jdbc:postgresql://localhost:5432/
   ↪ math_trainer
2 spring.datasource.username=app
3 spring.datasource.password=app
4 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.
   ↪ PostgreSQLDialect

```

Zdrojový kód 4.1: Server – konfigurace databáze

4.1.3.1 pgAdmin

pgAdmin je databázový klient, který se používá při vytváření PostgreSQL databází. Poskytuje grafické rozhraní pro kontrolu dat v tabulkách. Dále nabízí SQL konzoli, díky ní bylo možné si snadno navrhnout jednotlivé tabulky a vkládat testovací data.

Tento pomocník není vhodný nejen na vývoj databáze, ale i na její průběžnou údržbu. Nejčastěji byly využity funkce:

- **Backup** umožňuje zálohování dat či celé databáze.
- **Restore** slouží k nahrání již existující zálohy.

4.1.4 Použité knihovny

Spring pro rozšíření služeb používá dependency. Tyto knihovny se nastavují v konfiguračním souboru `pom.xml`. Pro aplikaci byly využity následující knihovny.

- **Postgresql** – Umožňuje konfiguraci a komunikaci s databází PostgreSQL.
- **spring-boot-starter-thymeleaf** – Pro vývoj MVC aplikací za pomoci Thymeleaf, což je Java XML/XHTML/HTML template engine. [51]
- **spring-boot-starter-web** – Pro vývoj webů, zahrnuje RESTful a použití MVC. [52]
- **spring-boot-starter-data-jpa** – Pro ukládání a získání dat z relační databáze. [53]
- **spring-boot-starter-test** – Umožňuje používat JUnit testy. [54]
- **spring-boot-starter-security** – Umožňuje konfiguraci Spring Security. [55]

4.1.5 Dokumentace

U rozsáhlejších programů, nastává problém s přehledností a orientací v kódu. Mnohdy i programátor, který sám aplikaci vyvíjel zapomíná k čemu jaké funkce slouží. Navíc může nastat situace, že se na vývoji aplikace začne podílet i někdo jiný. Proto je důležité si po celou dobu vývoje aplikace vést řádnou dokumentaci. Z těchto důvodů byl server řádně zdokumentován pomocí softwarové pomůcky JavaDoc. Dokumentace je dostupná v příloze bakalářské práce.

4.2 Klient

Klient zobrazuje uživateli veškerá data a reaguje na chování uživatele. Pokud klient potřebuje uživateli zobrazit nějaké data, které nemá k dispozici, pošle požadavek na server, který mu je poskytne.

4.2.1 Vývoj

Pro vývoj klienta byl použit programovacím editor Visual Studio Code. Tento editor byl využit pro psaní kódu. Kód byl pak kompilován pomocí `npm`. pro zobrazení přeloženého kódu byl použit Google Chrome s rozšířením Redux DevTools, který umožňuje přehledný vizuální přehled nad stavy v aplikaci.

4.2.2 Architektura

Tato část kapitoly se bude věnovat architektuře klienta. Bude zde zmíněno, jak je klient implementován, na jaké části se dělí a jak tyto části spolu komunikují.

4.2.2.1 Komponenty

Klient se skládá z jednotlivých komponent, tyto komponenty vykreslují obsah webové stránky. K vykreslení obsahu slouží funkce `render()`, což je jediná povinná metoda, kterou každá komponenta musí obsahovat. Jedna komponenta může obsahovat více komponent, což usnadní navržení jednotlivých stránek a přehlednost v kódu.

Komponenta obsahuje stavy buď lokální nebo stavy ze store, který je popsán níže, v kapitole 4.2.2.3. Pokud se stav změní, funkce `render()` se zavolá znovu a stránka se překreslí. [56]

4.2.2.2 Reducer

Jedinou funkcionalitou reducer je upravovat stavy v úložišti. Reducer přijme akci a popřípadě nepovinný parametr `data` a podle dané akce se rozhodne co v úložišti provede. Reducer se většinou implementuje jako `switch`, který je vázán na akci. [31]

4.2.2.3 Store

Hlavní myšlenkou Redux je jednotné úložiště v aplikaci a tím je Store. Store je to objekt se stromovitou strukturou. V tomto úložišti je většina stavů z aplikace. Tyto stavy může upravovat pouze Reducer, což zajišťuje určitou přehlednost v kódu a kontrolu nad stavy. [31]

4.2.2.4 Actions

Store tedy představuje úložiště, které spravuje reducer. Actions slouží pokud uživatel potřebuje dát vědět reducer, co se bude v úložišti měnit. Jsou to funkce, které v komponentách volá uživatel a následně předávají data s danou akcí reducer. V mnoha případech tyto funkce komunikují se serverem, například za použití funkcí z knihovny `axios`. [31]

4.2.2.5 Shrnutí

Pro lepší pochopení jak spolu jednotlivé části klienta komunikují bude v této části kapitoly popsána ukázka na konkrétním příkladě. Například, jak může vypadat, když se chce uživatel přihlásit do aplikace.

Uživatel se nachází na stránce s formulářem pro přihlášení, který vyplní svými přihlašovacími údaji a klikne na tlačítko *Přihlásit se*. Po kliknutí na toto tlačítko se zavolá funkce `login()` v actions, která přijímá přihlašovací údaje

4. IMPLEMENTACE

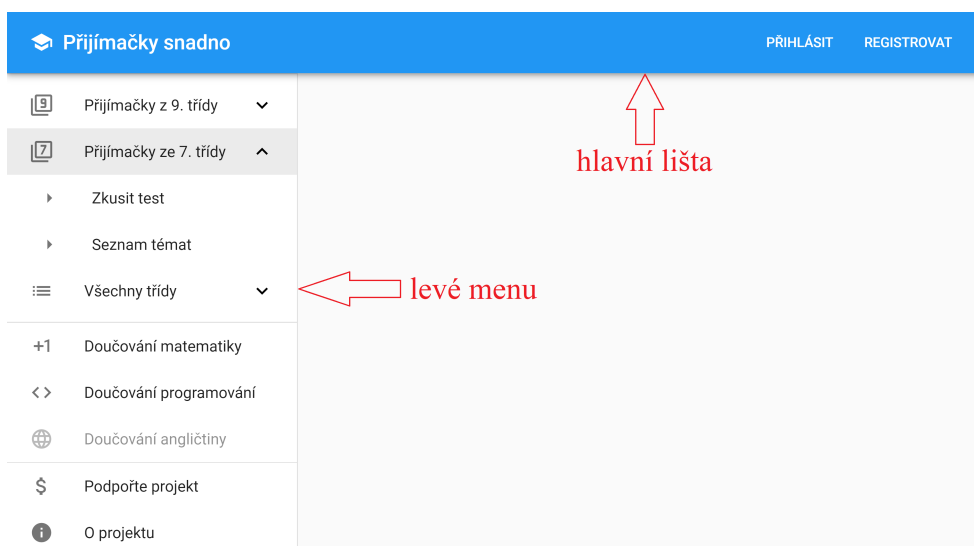
uživatele. Funkce `login()` pošle přihlašovací údaje ve formátu JSON serveru přes HTTP dotaz a server jí vrátí uživatele z databáze. Funkce ho s akcí `SAVE_USER` předá reducer. Reducer je switch, který ví, že když dostane akci `SAVE_USER`, tak má uložit obdržená data, pod objekt `user` ve store. Uživatel je následovně přesměrován na novou stránku, která vykresluje informace o přihlášeném uživateli ze stavu `user` ze store. Tato stránka se vykreslí dřív, než se uživatel do store uloží, ale protože tato stránka vykresluje stav `user`, po změně tohoto stavu se znova zavolá funkce `render()`, která stránku překreslí.

4.2.3 Uživatelské rozhraní

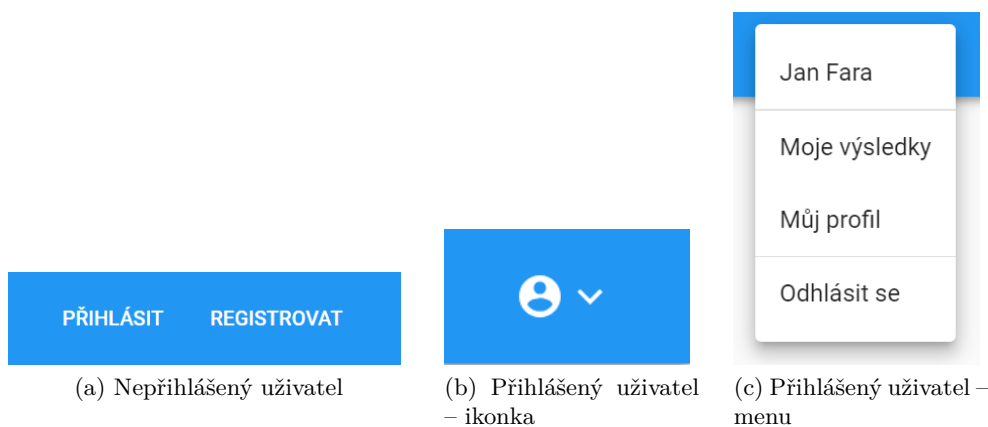
Uživatelské rozhraní bylo navrženo celkem pro 3 typy uživatelů:

- nepřihlášený uživatel,
- přihlášený uživatel,
- administrátor.

Každý z těchto uživatelů má jiné rozhraní. Hlavní komponenty uživatelského rozhraní, které dále budou popisovány jsou znázorněny na obrázku 4.1. Pro nepřihlášeného a přihlášeného uživatele se uživatelské rozhraní liší hlavně v hlavní liště aplikace. Nepřihlášený uživatel má možnost se zaregistrovat a nebo přihlásit. Zatímco přihlášený uživatel má zde k dispozici ikonku, která po rozkliknutí rozbalí uživatelské menu, viz obrázek 4.2.



Obrázek 4.1: Uživatelské rozhraní – popis hlavních komponent



Obrázek 4.2: Ukázka změny v menu u přihlášeného a nepřihlášeného uživatele

Menu nabízí přihlášenému uživateli další tři obrazovky, na které se nepřihlášený uživatel nedostane. A to jsou výsledky se seznamem svých testů, konkrétní starý test se všemi odpověďmi a nastavení, ve kterém může upravovat svůj profil.

U administrátora jsou změny v rozhraní už mnohem širší. Administrátor má hned v hlavní liště k dispozici odkaz na reportované příklady, které si může jednotlivě procházet. V uživatelském menu má na rozdíl od přihlášeného uživatele navíc seznam všech registrovaných uživatelů, kde si může každého jednotlivě otevřít a popřípadě editovat profil. Další změnou v menu je odkaz na seznam \LaTeX zkratk, které zde administrátor může libovolně editovat. Ukázku rozhraní administrátora lze vidět na obrázku 4.3, konkrétně na seznamu příkladů.

U administrátora se však liší i většina uživatelského rozhraní. Na každé stránce s materiály má administrátor navíc ikonky nebo tlačítka které mu umožní libovolně materiál upravovat.

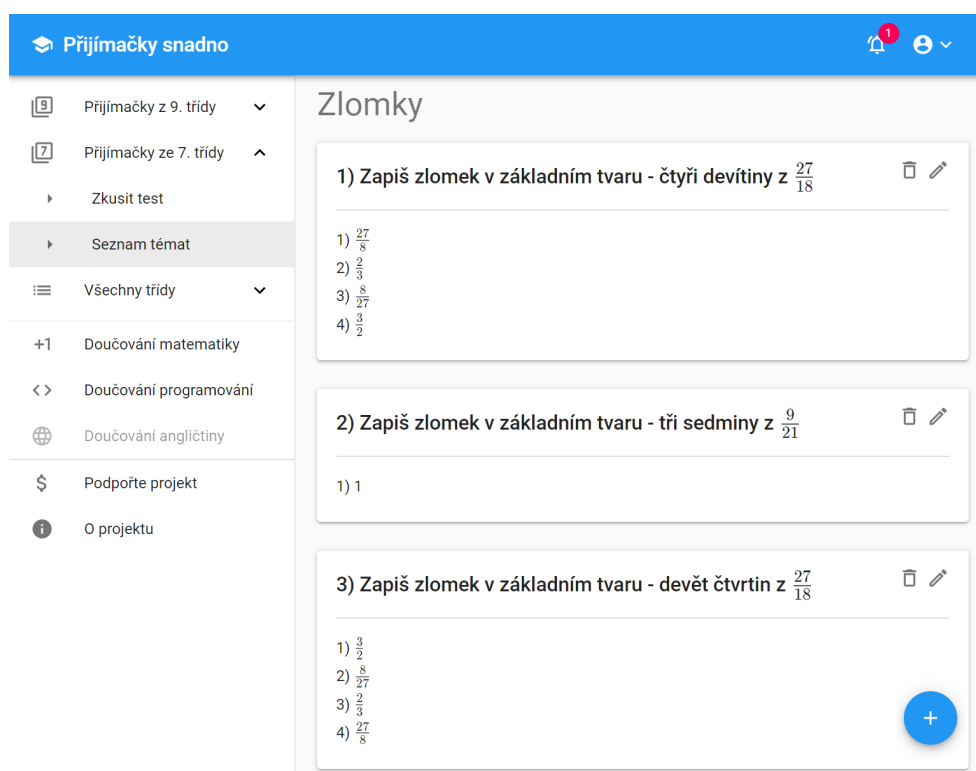
4.2.4 Responsibilita

Webová aplikace může být spustitelná na různých zařízeních:

- notebook,
- stolní počítač,
- tablet,
- mobil.

Každé z těchto zařízení má jinak velkou obrazovku. Aby byla aplikace přehledná na všech těchto zařízeních, bylo třeba některé komponenty navrhnout zvlášť, pro různé typy obrazovek.

4. IMPLEMENTACE



Obrázek 4.3: Uživatelské rozhraní – administrátor – příklady

Největší takovou úpravou bylo levé navigační menu pro mobilní verzi. Menu zabíralo většinu obrazovky, a tak nezbylo místo na obsah stránky. Tento problém byl vyřešen přesunutím menu do podoby tlačítka, které je umístěno v horní liště. Menu si tak lze zobrazit pouze tehdy, kdy je to nutné.

Další nutnou úpravou pro mobilní verzi byla horní lišta, do které se nevešlo uživatelské menu, které lze vidět na obrázku 4.2. Toto uživatelské menu bylo přidáno pod levé navigační menu.

4.2.5 Použité knihovny

V části této kapitoly jsou zmíněny důležité knihovny, které byly při vývoji aplikace využity.

- **Material Design** je CSS framework, který usnadňuje tvorbu uživatelského rozhraní. Pro tvorbu UI se používají komponenty, které tento framework poskytuje. Jednotlivá komponenta je například tlačítko nebo input. Velkou výhodou tohoto frameworku je snadné použití a přizpůsobení jednotlivých komponent. Za použití tohoto frameworku je také velice jednoduché udělat web responzivní.

- **Lodash** tato knihovna obsahuje veliké množství funkcí pro práci s objekty a poli v JavaScript.
- **Axios** je knihovna pro HTTP dotazy na server.
- **react-tex** tato knihovna byla použita pro vypsání zadání a řešení příkladu v jazyce \LaTeX .
- **Recharts** poskytuje volné využití různých grafů které lze upravit podle svých potřeb. Knihovna byla použita protože každý uživatel může sledovat svůj postup ve vyplňování testů na grafu.
- **react-router-dom** slouží pro linkování mezi jednotlivými komponenty. Za použití `stránka` by se komponenty načetly znovu, to by zabralo mnoho času a také by se resetovaly stavy komponent.
- **react-redux** je knihovna pro jednotné ukládání stavů, viz kapitola 3.5.5.
- **react-thunk** je knihovna která poskytuje Thunk middleware pro Redux. Middleware rozšiřuje schopnosti úložiště a umožňuje asynchronní logiku kódu při práci s úložištěm.

4.3 Testování

Aby se v aplikaci minimalizovali chyby provádí se takzvané testování. Způsobu testování je mnoho, například:

- unit testy,
- integration testy,
- regression testy,
- smoke testy.

Každý typ testů se zaměřuje na otestování různé části programu, ať už jako celku nebo jen jedné části.

4.3.1 JUnit

JUnit testy se zaměřují na testování konkrétních jednotek kódu. Každý test by měl být izolovaný, to znamená že je nezávislý na ostatních testech. Každý test se zaměřuje na konkrétní funkčnost, nemělo by se stát tedy, že jeden test bude testovat najednou více funkcí v programu. Po spuštění testu musí být jasné, zda uspěl, či nastala chyba. [57] Tyto testy byly použity pro otestování serverové části.

4.4 Nasazení

Proto, aby byla aplikace dostupná na internetu a uživatelé si ji mohli vyhledat v prohlížeči, je třeba udělat proces, který je popsán v následujících kapitolách.

4.4.1 Webhosting

Webhosting je pronajmutí si prostoru na serveru, který vlastní poskytovatel. Na tomto serveru se můžou soubory pouze nahrávat a mazat, nikoli je instalovat či spouštět. Proto je tato služba určena k uložení webových stránek například JavaScript, který si prohlížeč při navštívení adresy stáhne.

4.4.2 Doména

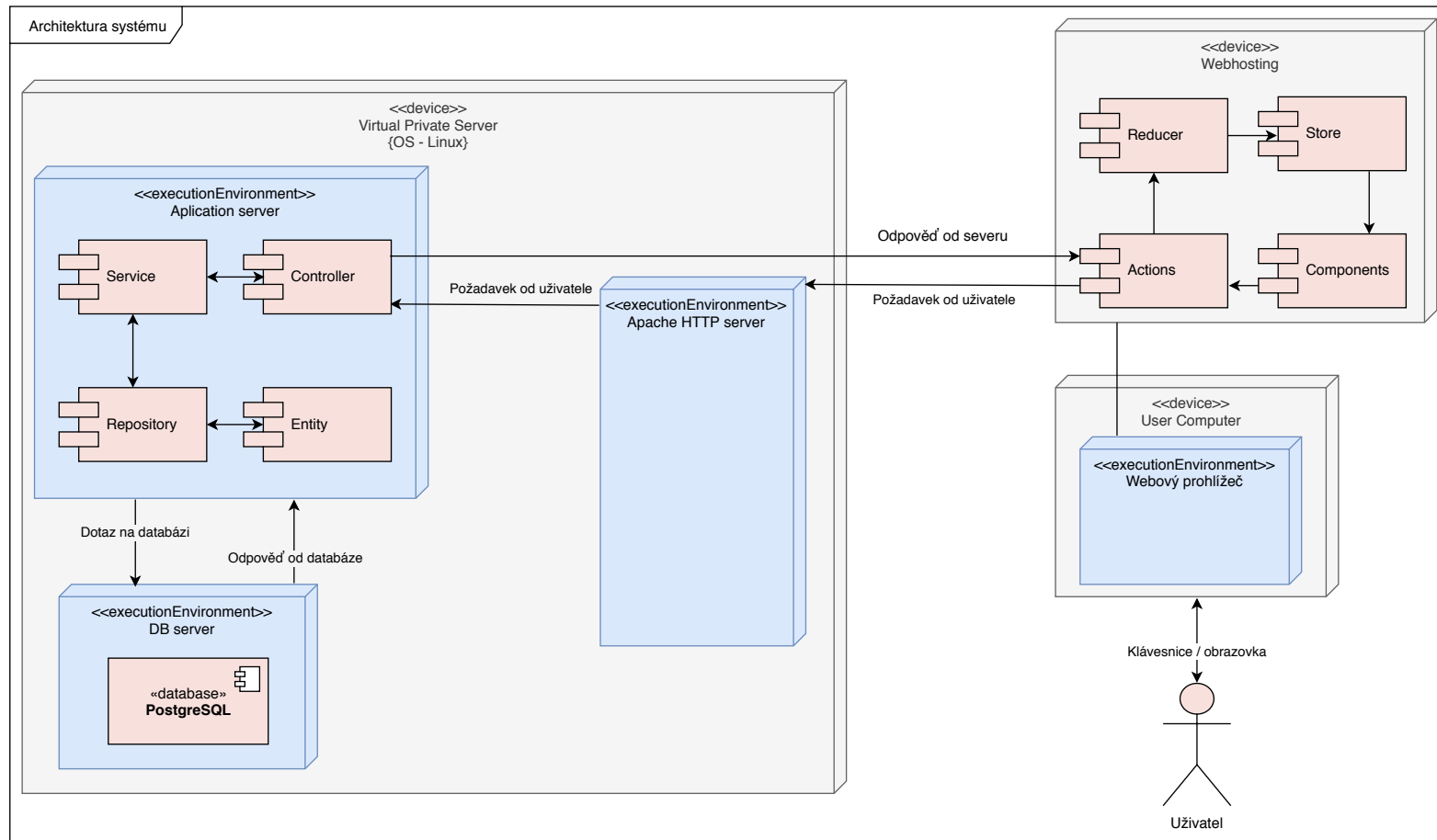
Doména slouží jako „přezdívka“ pro IP adresu webhostingového serveru na kterém je web uložený, přesněji se jí říká doménové jméno. Překlad doménového jména na IP adresu má na starosti DNS což je síťová služba, která překládá doménové jména na IP adresy. Bez doménového jména by si uživatelé webovou stránku nemohli vyhledat. Jako doménové jméno bylo zvoleno `www.prijimackysnadno.cz`, pod touto adresou lze aplikaci dohledat.

4.4.3 Virtual private server

Protože je žádoucí, aby byl server dostupný pořád, je třeba, aby byl na nějakém zařízení stále spuštěný. Jedna z možností, která byla zvolena je pořízení si VPS. VPS je počítač vlastněn společností, která tuto službu poskytuje. Konkrétně byly zvoleny služby od společnosti Wedos.

Na tomto počítači běží operační systém Linux. Počítač má vlastní IP adresu, pomocí ní se lze k počítači připojit například pomocí příkazu `ssh` nebo pomocí aplikace PuTTY. Na tomto počítači je vlastněn uživatel `root`, což je na Linux systému uživatel s nejvyšším oprávněním, proto se zde na rozdíl od webhostingu můžou aplikace libovolně instalovat a spouštět.

Na tomto počítači jsou spuštěny HTTP server, aplikační server a databáze.



Obrázek 4.4: Architektura systému

4.4.4 Shrnutí

Celý proces nasazení aplikace byl poměrně náročný, jak finančně, tak časově. Roční provoz této aplikace vyjde přibližně na 2 000 Kč. Velkým oříškem byla správná konfigurace VPS serveru, kde kromě nainstalování všech patřičných aplikací bylo třeba provést několik konfigurací, například PostgreSQL, Apache a IPv4 adresy, která musela být dodatečně zakoupena protože usnadňuje připojení k VPS.

Nasazení se neprovádí pouze jednou, aplikace se často upravuje nebo se opravují dříve přehlédnuté chyby. Aplikace v produkci má jinou konfiguraci než tu, která slouží pro vývoj. Například na serveru bylo nutné rozlišit adresu pro ukládání obrázků, origin adresu a adresu k databázi. Z tohoto důvodu byly na serveru vytvořeny dva konfigurační soubory, `application-dev.properties` je konfigurace pro vývoj a `application-prod.properties` pro produkční verzi. Díky této konfiguraci stačí program spouštět s přepínači `-Dspring.profiles.active=dev/prod`. Podobné opatření bylo provedeno i u klienta, u kterého se liší adresy na které klient posílá dotazy.

Po nahrání klienta na webhosting byl problém, že po načtení konkrétní URL adresy se adresa přepsala na domovskou. Proto dalším nutným opatřením bylo upravení souboru `.htaccess`, což je konfigurační soubor, který je umístěn s klientem na webhostingu.

Po všech dodatečných úpravách se úspěšně aplikaci podařilo zprovoznit online. Aplikace je nyní dostupná na adrese <http://www.prijimackysnadno.cz>. A studenti si zde již můžou zkusit testy nanečisto.

Uživatelské testování

Jeden z nejdůležitějších testů aplikace je nechat vyzkoušet aplikaci na skupině uživatelů, pro kterou je přímo určena. Proto aplikaci 10.04.2019 vyzkoušeli studenti na Základní škole Černošice v rámci hodiny matematiky v počítačové učebně. Z důvodu čtyřiceti-pěti minutové hodiny byl pro studenty připraven test, který se skládal ze sedmi témat. V testu každé téma zastupoval jeden příklad.

5.1 Průběh testování

Pro testování aplikace byly k dispozici dvě třídy dohromady přibližně o 40 žácích. Každá třída měla na testování 45 minut.

Na začátku hodiny si každý student zapnul počítač, přihlásil se a v prohlížeči vyhledal danou webovou stránku, která jim byla řečena. Při hledání webové stránky mělo pár studentů problém se slovem „prijimacky“, které psali jako „prijmacky, primacky“. Další problém byl že adresu zadávali do vyhledávače a ne do URL adresy, protože aplikace ještě nemá dobrou SEO konfiguraci, nebyla prostřednictvím vyhledávače dohledatelná.

Když všichni studenti vyhledali webovou stránku a byli na hlavní stránce, byly jim dány pokyny, aby se dostali do sekce testy pro sedmé třídy. s tímto neměl žádný student problém. Poté byly studentům řečeny instrukce pro vyplňování testů, aby se nezdržovali čtením pokynů, pár studentů si i přesto pokyny přečetli.

Při vyplňování testů studenti nenarazili na žádný vážný problém, pouze na drobnosti, které jsou zmíněny v kapitole níže 5.3. Někteří studenti si test dokonce stihli vyplnit vícekrát. Po dokončení testu každý student vyplnil anketu, která mu byla poskytnuta v elektronické podobě.

5.2 Anketa

Pro zjištění, zda aplikace bude žádaná a pro zpětnou vazbu ohledně aplikace byla vytvořena anketa. Anketa proběhla na Základní Škole Černošice s žáky sedmé třídy hned po tom, co si studenti vyzkoušeli aplikaci na krátkém testu. Anketa v elektronické podobě formou Google Forms.

5.2.1 Otázky ankety

Anketa proběhla za účelem zjistit, kolik studentů by mělo zájem v rámci přípravy používat prezentovanou aplikaci, zda jim přišla přehledná a zda se při používání nevyskytl nějaký problém.

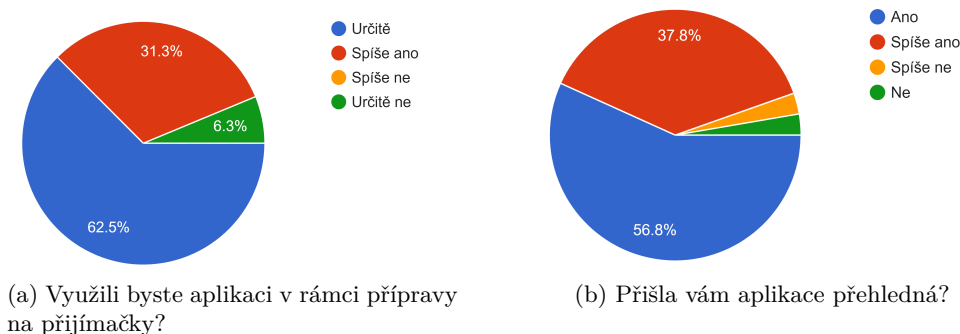
1. Využili byste aplikaci v rámci přípravy na přijímací řízení?
 - Určitě
 - Spíše ano
 - Spíše ne
 - Určitě ne
2. Přišla vám aplikace přehledná?
 - Ano
 - Spíše ano
 - Spíše ne
 - Ne
3. Měli jste během používání aplikace nějaké problémy? Pokud ano napište jaké.
 - Otevřená odpověď

5.2.2 Výsledek ankety

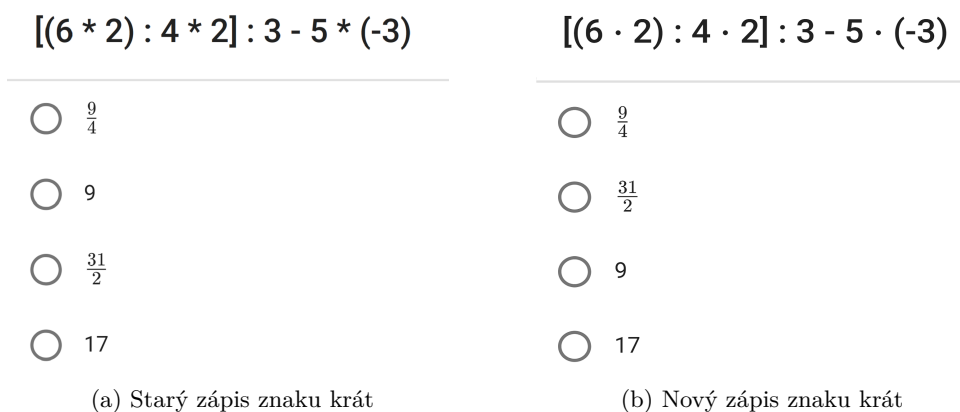
Anketu vyplnilo 37 studentů. Aplikaci by 94 % studentů pravděpodobně v rámci své přípravy využilo, viz graf na obrázku 5.1a. Studenti aplikaci hodnotili spíše jako přehlednou, konkrétně 95 % studentů, viz graf na obrázku 5.1b.

5.3 Zpětná vazba

Studenti v průběhu tetu neměli žádný zásadní problém s používáním aplikace. Vyskytli se jen menší problémy, které jsou rozepsány v následujících kapitolách.



Obrázek 5.1: Výsledky ankety – hodnocení aplikace



Obrázek 5.2: Ukázka změny v zápisu znaku krát

5.3.1 Čitelnost znaku krát

Největším problémem, který se při vyplňování testů vyskytl byl čitelnost znamínka krát v zadání příkladů. Původně byl pro toto znamínko použit znak „*“, viz obrázek 5.2a.

Přibližně 5 studentů z 40 se v průběhu testu přihlásilo, protože si nebyli jistí, co znamená znak „*“. Tři studenti do ankety tento problém s špatně čitelným znakem napsali. Proto byl tento znak nahrazen za \LaTeX znak \cdot , viz obrázek 5.2b.

5.3.2 Zápis zlomku

Dalším problémem, kvůli kterému se děti často hlásili, byl zápis zlomku u příkladů s otevřenou odpovědí. Před začátkem testu bylo dětem řečeno, že příklady s otevřenou odpovědí vychází většinou v celých číslech. I přes to, velká část dětí zlomek do odpovědi zapsala.

5. UŽIVATELSKÉ TESTOVÁNÍ

Problém zápisu zlomků měl být vyřešen tak, že všechny příklady s otevřenou odpovědí vycházely v celých číslech. Kdyby však tuto informaci studenti dostali, měli by velkou nápovědu u řešení těchto příkladů.

Tento problém byl vyřešen připsáním informace do pokynů, které si každý student před začátkem testu přečte. Zlomek se bude nadále zadávat pomocí znaku „/“, například tři pětiny: $3/5$.

Závěr

Cílem této bakalářské práce bylo vytvořit aplikaci, která by studentům základních škol usnadnila přípravu na přijímací řízení. Cíl této bakalářské práce byl splněn. Po analýze existujícího řešení a po konzultacích se studenty byl navržen prototyp uživatelského rozhraní na vyplňování testů. Dále byla aplikace naimplementována tak, aby pokryla veškeré funkční a nefunkční požadavky, které byly stanoveny v analýze. Aplikace byla otestována a nasazena na produkční server a webhosting. Po nasazení se povedlo na základní škole aplikaci v rámci hodiny vyzkoušet ve dvou třídách.

Výsledkem této bakalářské práce je plně funkční webová aplikace, která je přístupná přes internet, kde slouží studentům k přípravě na přijímací řízení. Aplikace má nyní řádově desítky reálných uživatelů, kteří využívají její služby. Pevně věřím, že těmto uživatelům aplikace pomůže dostat se na jimi zvolenou střední školu, čímž se také naplní hlavní myšlenka této aplikace.

Vývoj této aplikace byl velice přínosný i pro mě. Nejen, že jsem si vyzkoušel práci na větším projektu, ale také jsem se naučil nové a populární technologie, ve kterých jsem práci vyvíjel.

Možnosti rozšíření

Na vývoji aplikace jsem strávil opravdu mnoho času, ať už učením se technologií nebo samotnou implementací. Proto bych rád v budoucnu aplikaci udržoval v provozu a poskytoval k ní podporu.

V budoucnu bych rád přidal testy pro studenty, kteří se hlásí na osmiletá gymnázia, tedy testy pro páté třídy. Jako další možné zlepšení je rozšíření databáze příkladů, aby studenti měli důvod aplikaci dlouhodobě využívat a mohli si vyzkoušet co nejvíce různých příkladů. Také bude třeba zakoupení HTTPS certifikátu, který jsem z důvodu nákladů zatím nepořizoval.

Aplikace řeší konkrétní problematikou, touto problematikou se žádná jiná aplikace nezabývá. Navíc je její používání velice intuitivní. Proto za předpokladu splnění předchozích podmínek není důvod, proč by studenti aplikaci v rámci své přípravy na střední školy nepoužívali. Dovoluji si říct, že aplikace má dobrý potenciál pro to, aby ji používalo velké množství uživatelů.

Nakonec bude třeba zlepšit SEO a pár marketingových opatření. Mám v plánu investovat do reklamy na Google a Facebook. Tyto reklamy se bu-

ZÁVĚR

dou zobrazovat pouze cílové skupině, která bude vyhledávat klíčová slova jako *přijímačky na střední školu, přijímačky online* atd. Jako další marketingový tah plánuji vytisknout letáčky, které osobně rozdám v několika základních školách. Pokud aplikace bude opravdu studentům pomáhat, určitě si ji mezi sebou doporučí sami.

Literatura

1. SCIO. *Příprava na státní přijímací zkoušky*. [online]. 2019. Dostupné také z: <https://www.scio.cz/prijimaci-zkousky-na-ss/priprava>. [vid. 2. 4. 2019].
2. CERMAT. *Základní informace*. [online]. 2019. Dostupné také z: <https://www.cermat.cz/zakladni-informace-1404035544.html>. [vid. 2. 4. 2019].
3. NOVÁ MÉDIA, S. R. O. *Matematika pro základní školy*. [online]. 2019. Dostupné také z: <https://matematika.cz/matematika-pro-zakladni-skoly>. [vid. 2. 4. 2019].
4. *Státní přijímací zkoušky na střední školy 2019*. [online]. 2019. Dostupné také z: <https://www.statniprijimacky.cz/>. [vid. 2. 4. 2019].
5. *Zkoušky nanečisto pořádané v Praze pro žáky 7. tříd*. [online]. 2019. Dostupné také z: <https://zkousky-nanecisto.cz/7trida/zkousky-nanecisto-poradane-v-praze>. [vid. 9. 3. 2019].
6. TO-DAS. *Přijímačky nanečisto*. [online]. 2019. Dostupné také z: <https://www.to-das.cz/jak-to-funguje/#prubehkurzu>. [vid. 10. 3. 2019].
7. ZÁLEŠÁKOVÁ, Mgr. Marie [osobní sdělení].
Základní škola Černošice, Pod Školou 447, 252 28 Černošice.
2019-04-10.
8. MATIKA.IN Z.S.
Úlohy z matematiky pro děti na základních školách. [online]. 2019. Dostupné také z: <https://www.matika.in/cs/>. [vid. 10. 3. 2019].
9. FRANTIŠEK KUDLÁČEK. *Online cvičení*. [online]. 2019. Dostupné také z: <https://www.onlinecviceni.cz>. [vid. 17. 4. 2019].
10. *Příklady z matematiky*. [online]. 2019. Dostupné také z: <http://www.prikladyzmatematiky.cz>. [vid. 15. 4. 2019].

11. EDUFOR SERVICES S. R. O.
Online procvičování DRILL and SKILL. [online]. 2019.
Dostupné také z: <https://www.drillandskill.com>. [vid. 20. 4. 2019].
12. SRPOVÁ, Mgr. Ladislava.
Bakalářská práce - průzkum [elektronická pošta].
Příjemce zprávy: srpova@zsdobrichovice.cz. 2019-04-15.
13. *Chcete zaujmout pozornost žáků?. [online].* 2019.
Dostupné také z: <https://programalf.com/alf/cz/index.html>.
[vid. 04. 05. 2019].
14. SILCOM MULTIMEDIA.
Výukové programy pro základní školy a jejich žáky. [online]. 2019.
Dostupné také z: <http://www.silcom-multimedia.cz>.
[vid. 04. 05. 2019].
15. TERASOFT, A. S. *Výukové programy. [online].* 2019.
Dostupné také z: <https://www.terasoft.cz>. [vid. 04. 05. 2019].
16. PRŮCHOVÁ, Mgr. Monika.
Bakalářská práce - průzkum [elektronická pošta].
Facebook – messenger. 2019-05-03.
17. *Functional and Nonfunctional Requirements: Specification and Types. [online].* 2019. Dostupné také z:
<https://www.altexsoft.com/blog/business/functional-and-non-functional-requirements-specification-and-types/>.
[vid. 19. 4. 2019].
18. *The In's and Out's of Cryptographic Hash Functions. [online].* 2019.
Dostupné také z:
<https://blockgeeks.com/guides/cryptographic-hash-functions/>.
[vid. 07. 05. 2019].
19. LAURA BRANDENBURG. *What is a Use Case?. [online].* 2019.
Dostupné také z:
<https://www.bridging-the-gap.com/what-is-a-use-case/>.
[vid. 20. 4. 2019].
20. *The most popular programming language. [online].* 2019.
Dostupné také z: <https://www.amarinfotech.com/top-6-popular-programming-languages-in-2019.html>. [vid. 7. 4. 2019].
21. GOSLING, James; JOY, Bill; STEELE, Guy; BRACHA, Gilad.
The Java language specification. Addison-Wesley Professional, 2000.
22. JOHNSON, Rod et al. *The spring framework–reference documentation. interface.* 2004, roč. 21, s. 27.

23. *Introduction to Spring Framework*. [online]. 2019. Dostupné také z: <https://docs.spring.io/spring/docs/3.0.x/spring-framework-reference/html/overview.html#d0e276>. [vid. 8. 4. 2019].
24. *Introducing Spring Boot*. [online]. 2019. Dostupné také z: <https://docs.spring.io/spring-boot/docs/2.1.3.RELEASE/reference/htmlsingle/#boot-documentation>. [vid. 4. 4. 2019].
25. *JavaScript (JS)*. [online]. 2019. Dostupné také z: <https://www.techopedia.com/definition/3929/javascript-js>. [vid. 1. 4. 2019].
26. VOJTECH MIKSU. *React - Úvod*. [online]. 2019. Dostupné také z: <https://www.dzejes.cz/react-uvod.html>. [vid. 21. 3. 2019].
27. ANDRE KHONG. *Winning the war of CSS conflicts through the Shadow DOM*. [online]. 2019. Dostupné také z: <https://medium.com/rate-engineering/winning-the-war-of-css-conflicts-through-the-shadow-dom-de6c797b5cba>. [vid. 6. 4. 2019].
28. JOHN POTTER. *@angular/core vs angular vs react vs vue*. [online]. 2019. Dostupné také z: <https://www.npmtrends.com/@angular/core-vs-angular-vs-react-vs-vue>. [vid. 5. 3. 2019].
29. WIKIPEDIA. *Redux (JavaScript library)*. [online]. 2019. Dostupné také z: [https://en.wikipedia.org/wiki/Redux_\(JavaScript_library\)](https://en.wikipedia.org/wiki/Redux_(JavaScript_library)). [vid. 22. 3. 2019].
30. BANKS, Alex; PORCELLO, Eve. *Learning React: functional web development with React and Redux*. "O'Reilly Media, Inc.", 2017.
31. SEBASTIAN ESCHWEILER. *Learn Redux—Introduction To State Management With React*. [online]. 2019. Dostupné také z: <https://medium.com/codingthesmartway-com-blog/learn-redux-introduction-to-state-management-with-react-b87bc570b12a>. [vid. 4. 3. 2019].
32. *Redux DevTools*. [online]. 2019. Dostupné také z: <https://chrome.google.com/webstore/detail/redux-devtools/lmhkpbekcpmknklioieibfkpmmfijld>. [vid. 8. 3. 2019].
33. THE POSTGRESQL GLOBAL DEVELOPMENT GROUP. *License*. [online]. 2019. Dostupné také z: <https://www.postgresql.org/about/licence/>. [vid. 4. 3. 2019].

34. MOMJIAN, Bruce. *PostgreSQL: introduction and concepts*. Addison-Wesley New York, 2001.
35. *Usage of web servers*. [online]. 2019. Dostupné také z: https://w3techs.com/technologies/overview/web_server/all. [vid. 9. 4. 2019].
36. *HTTP request methods*. [online]. 2019. Dostupné také z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>. [vid. 17. 4. 2019].
37. *Evolution of HTTP*. [online]. 2019. Dostupné také z: https://developer.mozilla.org/en-US/docs/Web/HTTP/Basics_of_HTTP/Evolution_of_HTTP. [vid. 9. 4. 2019].
38. *HTTP response status codes*. [online]. 2019. Dostupné také z: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Status>. [vid. 13. 4. 2019].
39. MARTIN FOWLER. *Richardson Maturity Model*. [online]. 2019. Dostupné také z: <https://martinfowler.com/articles/richardsonMaturityModel.html>. [vid. 13. 4. 2019].
40. MARGARET ROUSE. *REST (REpresentational State Transfer)*. [online]. 2019. Dostupné také z: <https://searchmicroservices.techtarget.com/definition/REST-representational-state-transfer>. [vid. 11. 4. 2019].
41. *MVC Framework - Introduction*. [online]. 2019. Dostupné také z: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm. [vid. 25. 4. 2019].
42. BASSETT, Lindsay. *Introduction to JavaScript Object Notation: A to-the-point Guide to JSON*. "O'Reilly Media, Inc.", 2015.
43. *Introduction to Apache Web Server (2.X.XX)*. [online]. 2019. Dostupné také z: <https://www.opensourcevarsity.com/intro-to-apache-webserver/>. [vid. 9. 4. 2019].
44. *Accessing Data with JPA*. [online]. 2019. Dostupné také z: <https://spring.io/guides/gs/accessing-data-jpa/>. [vid. 25. 3. 2019].
45. *Working with Spring Data Repositories*. [online]. 2019. Dostupné také z: <https://docs.spring.io/spring-data/commons/docs/1.6.1.RELEASE/reference/html/repositories.html>. [vid. 25. 3. 2019].

46. *@Component vs @Repository and @Service in Spring. [online].* 2019. Dostupné také z: <https://www.baeldung.com/spring-component-repository-service>. [vid. 25. 3. 2019].
47. *Spring RequestMapping. [online].* 2019. Dostupné také z: <https://www.baeldung.com/spring-requestmapping>. [vid. 25. 3. 2019].
48. *The Spring @Controller and @RestController Annotations. [online].* 2019. Dostupné také z: <https://www.baeldung.com/spring-controller-vs-restcontroller>. [vid. 25. 3. 2019].
49. *What is Spring Security?. [online].* 2019. Dostupné také z: <https://docs.spring.io/spring-security/site/docs/3.0.x/reference/introduction.html>. [vid. 20. 3. 2019].
50. *Spring Security. [online].* 2019. Dostupné také z: <https://spring.io/projects/spring-security>. [vid. 20. 3. 2019].
51. *Spring Boot Thymeleaf Starter. [online].* 2019. Dostupné také z: <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-thymeleaf/1.2.5.RELEASE>. [vid. 5. 4. 2019].
52. *Spring Boot Web Starter. [online].* 2019. Dostupné také z: <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-web/1.2.5.RELEASE>. [vid. 3. 4. 2019].
53. *Accessing Data with JPA. [online].* 2019. Dostupné také z: <https://spring.io/guides/gs/accessing-data-jpa/>. [vid. 18. 3. 2019].
54. *Spring Boot Test Starter. [online].* 2019. Dostupné také z: <https://mvnrepository.com/artifact/org.springframework.boot/spring-boot-starter-test>. [vid. 1. 4. 2019].
55. *Spring Boot Security Auto-Configuration. [online].* 2019. Dostupné také z: <https://www.baeldung.com/spring-boot-security-autoconfiguration>. [vid. 20. 3. 2019].
56. JOHN KAGGA. *Understanding React Components. [online].* 2019. Dostupné také z: <https://medium.com/the-andela-way/understanding-react-components-37f841c1f3bb>. [vid. 20. 3. 2019].
57. VIRENDER SINGH. *Junit Introduction. [online].* 2019. Dostupné také z: <https://www.toolsqa.com/java/junit-framework/junit-introduction/>!. [vid. 11. 3. 2019].

Seznam použitých zkratk

- AOP** Aspect Oriented Programming
- API** Application Programming Interface
- CSS** Cascading Style Sheets
- DNS** Domain Name Server
- DOM** Document Object Model
- DUM** Digitální učebnicové materiály
- GDPR** General Data Protection Regulation
- HTML** Hypertext Markup Language
- HTTP** Hypertext Transfer Protocol
- ID** identity
- IDE** Integrated Development Environment
- IP** Internet Protocol
- IPv4** Internet Protocol version 4
- JAXB** Java Architecture for XML Binding
- JDK** Java Development Kit
- JDO** Java Data Objects
- JPA** Java Persistence API

A. SEZNAM POUŽITÝCH ZKRATEK

JSON JavaScript Object Notation

MVC Model View Controller

NPM Node.js package manager

POJO Plain Old Java Object

POX Plain Old XML

QUIC Quick UDP Internet Connections

REST Representational State Transfer

SEO Search Engine Optimization

SSH Secure Shell

SQL Structured Query Language

SOAP Simple Object Access Protocol

TCP Transmission Control Protocol

TLS Protokol Transport Layer Security

UI User Interface

UML Unified Modeling Language

URL Uniform Resource Locator

VPS Virtual Private Server

XHTML Extensible Hypertext Markup Language

XML eXtensible Markup Language

Instalace

V první části kapitoly je popsáno spuštění aplikace a v druhé části návod pro vývoj aplikace. Aplikace využívá databázi PostgreSQL, a proto je třeba ji nainstalovat, viz stránka <https://www.postgresql.org/download/>. Pro vytvoření tabulek a jejich inicializaci je v příloze k dispozici záloha databáze `pg_dump.backup`. Tato záloha obsahuje administrátora s přihlašovacími údaji:

Uživatelské jméno:	admin
Heslo:	admin

B.1 Spuštění

V příloze ve složce `/src` je spustitelná verze aplikace `app.jar`. Pro jeho spuštění je třeba mít nainstalované JDK, webový prohlížeč a v příkazové řádce zadat kód:

```
java -jar -Dspring.profiles.active=dev app.jar &
```

Aplikace se poté spustí na pozadí na portu 8090 ve vývojové verzi a lze ji otevřít v prohlížeči na adrese `http://localhost:8090/`. Aplikace očekává databázi s následnou konfigurací:

B. INSTALACE

Host:	localhost
Port:	5432
Database:	math_trainer
User:	jan
Password:	jan

B.2 Vývoj

B.2.1 Server

Server je umístěn ve složce `/src/app/server/` a lze vyvíjet v libovolném IDE, který podporuje vývoj enterprise aplikací pomocí Spring boot. Například IntelliJ IDEA, Eclipse IDE, NetBeans, atd. Databázi, port a origin adresu lze nakonfigurovat ve souboru `application-dev.properties`. Server je třeba spouštět s aktivním profilem `dev`.

B.2.2 Klient

Pro vývoj klienta je třeba mít nainstalovaný webový prohlížeč, `npm`, viz stránka <https://www.npmjs.com/get-npm> a libovolný textový editor, v lepším případě takový, který umí číst JavaScript. Například Visual Studio Code, Nuclide, Sublime Text, atd. Dále je třeba nainstalovat moduly pomocí příkazu:

```
npm install
```

v příkazové řádce ve složce `/src/app/klient`. Po nainstalování modulů lze aplikaci spustit příkazem:

```
npm start
```

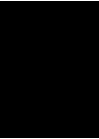
Pokud by bylo třeba vytvořit novou spustitelnou verzi aplikace, lze zadat kód:

```
npm run-script build
```

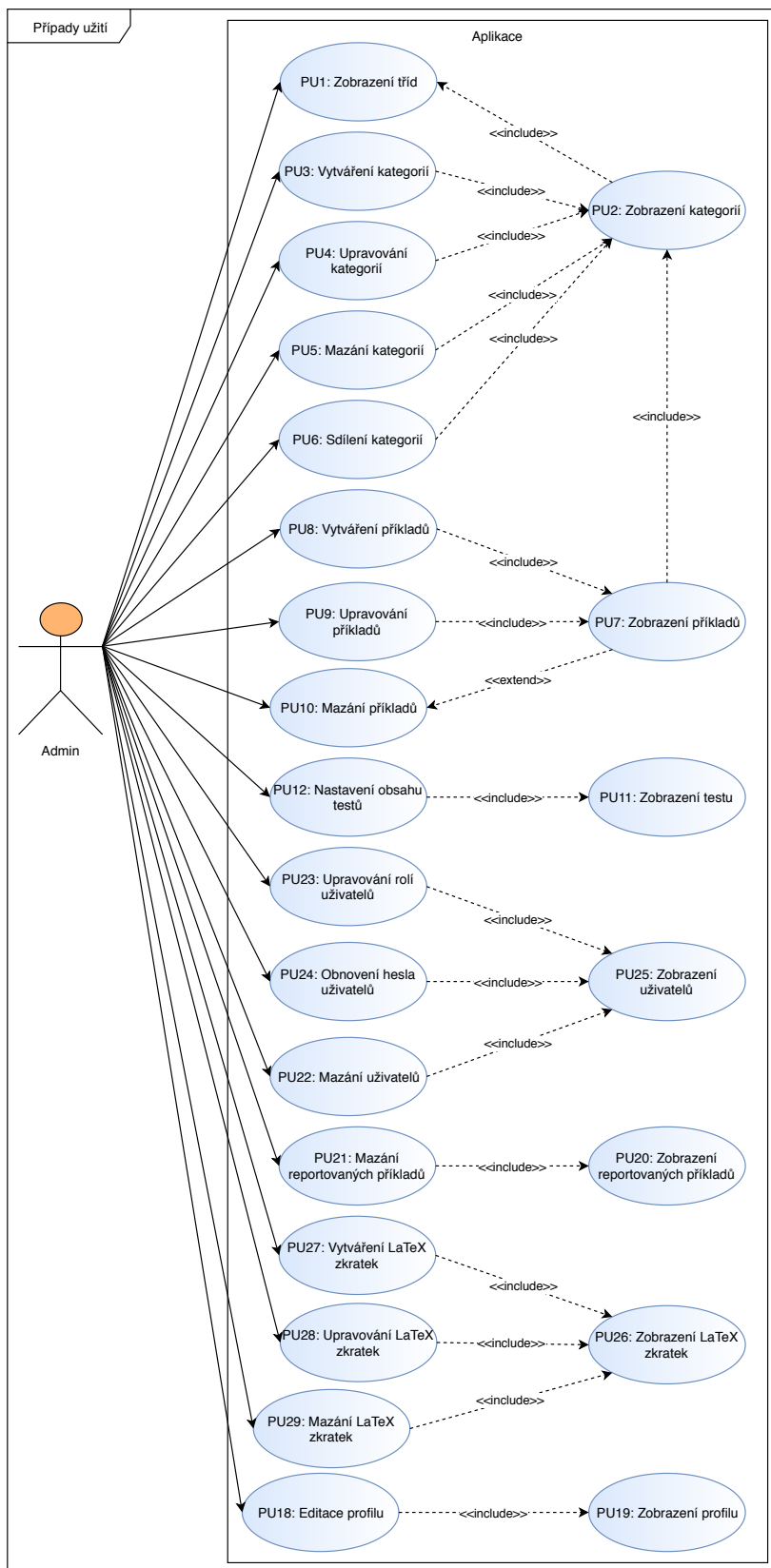
Spustitelnou verzi je pak možné najít ve složce `/src/app/klient/build`.

PŘÍLOHA

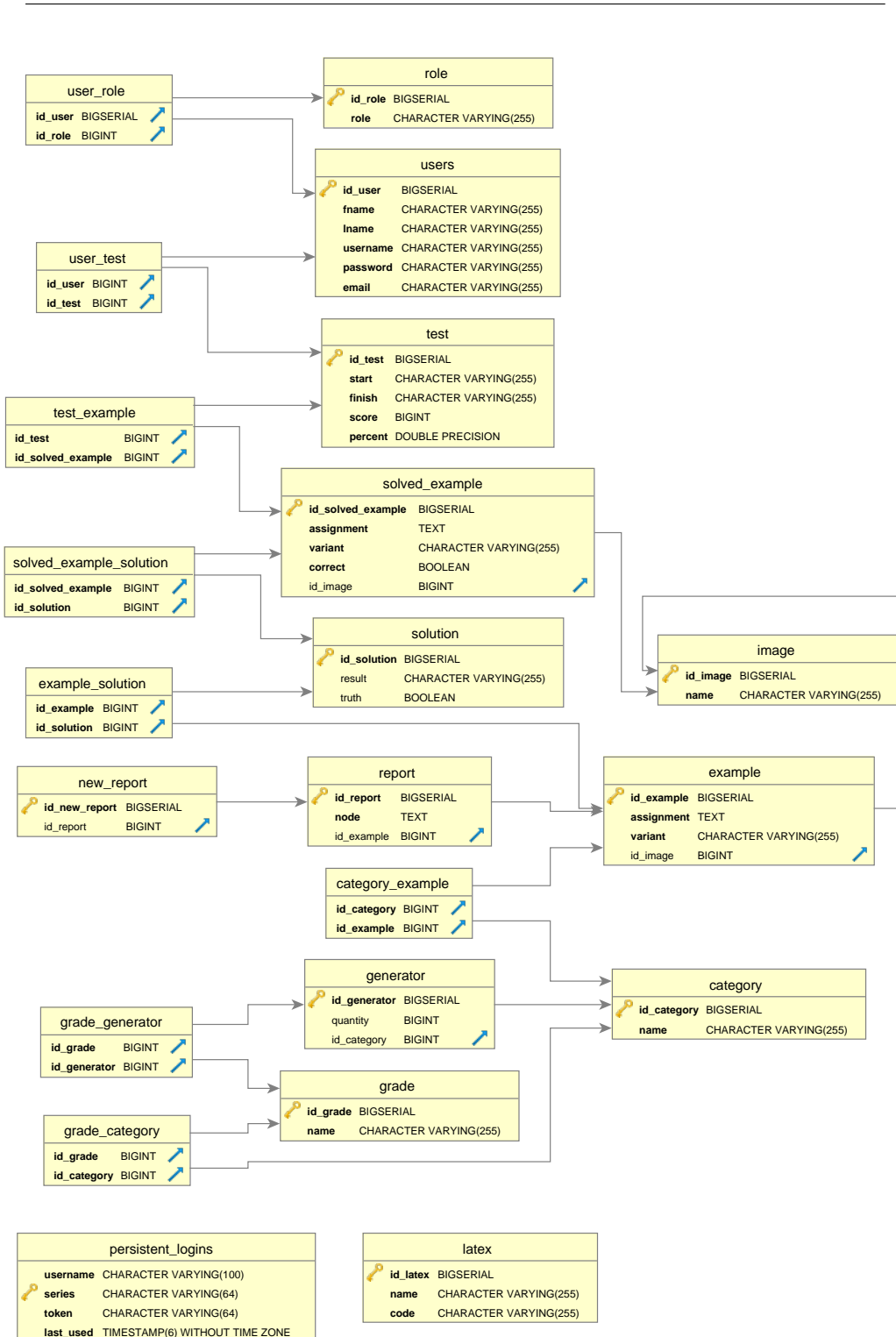
C



Obrázky

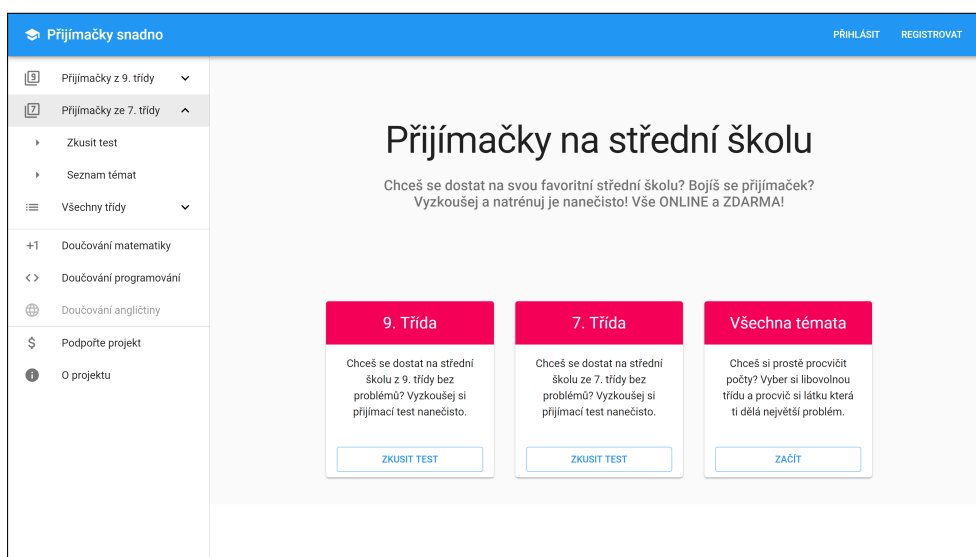


Obrázek C.1: Případy užítí – administrátor aplikace



Obrázek C.2: Databázový model

Ukázka výsledné aplikace



Obrázek D.1: Úvodní obrazovka

D. UKÁZKA VÝSLEDNÉ APLIKACE

The screenshot shows the 'Příjmačky snadno' application interface. The top navigation bar is blue with the text 'Příjmačky snadno' on the left and 'PŘIHLÁSIT' and 'REGISTROVAT' on the right. A left sidebar contains a menu with items like 'Příjmačky z 9. třídy', 'Příjmačky ze 7. třídy', 'Zkusit test', 'Seznam témat', 'Všechny třídy', 'Doučování matematiky', 'Doučování programování', 'Doučování angličtiny', 'Podpořte projekt', and 'O projektu'. The main content area is titled 'Přípravný test pro 7. třídy' and contains a 'Pokyny' section with the following text:

Hodnocení příkladů

- Test obsahuje sedm příkladů.
- Každý správně zodpovězený příklad je ohodnocen 1 bodem.
- Špatná či nevyplněná odpověď je ohodnocena 0 body.

Čas

- Čas máte stejný jako na oficiální test, tedy 2 hodiny.
- Po zahájení testu se vám začne čas odpočítávat.
- Protože tento test slouží k přípravě, tak i po vypršení času můžete dále odpovídat na otázky a nechat si test vyhodnotit.

Průběh testu

- V testu jsou tři typy příkladů: příklady s otevřenou odpovědí a příklady s více možnostmi, buď s právě jednou správnou odpovědí a nebo s různým počtem správných odpovědí.
- Na otázky můžete odpovídat v libovolném pořadí.
- Svou odpověď můžete v průběhu testu kdykoliv změnit.
- Pokud vyplňujete příklad s otevřenou odpovědí, ujistěte se, že odpověď neobsahuje žádné jiné znaky včetně mezer.
- Pokud chcete u příkladu s otevřenou odpovědí zapsat zlomek například "tři pětiny" zapíšete ho jako "3/5".

Vyhodnocení testu

- Test vyhodnotíte kliknutím na tlačítko vyhodnotit, které se nachází na stránce dole vpravo.
- Po vyhodnocení testu se vám zobrazí tabulka s výsledky.
- Každý příklad, který jste odpověděli špatně, bude zvýrazněn červenou barvou.
- Každý příklad, který jste zodpověděli správně, bude zvýrazněn zelenou barvou.

At the bottom of the instructions is a blue button labeled 'SPUSTIT TEST'.

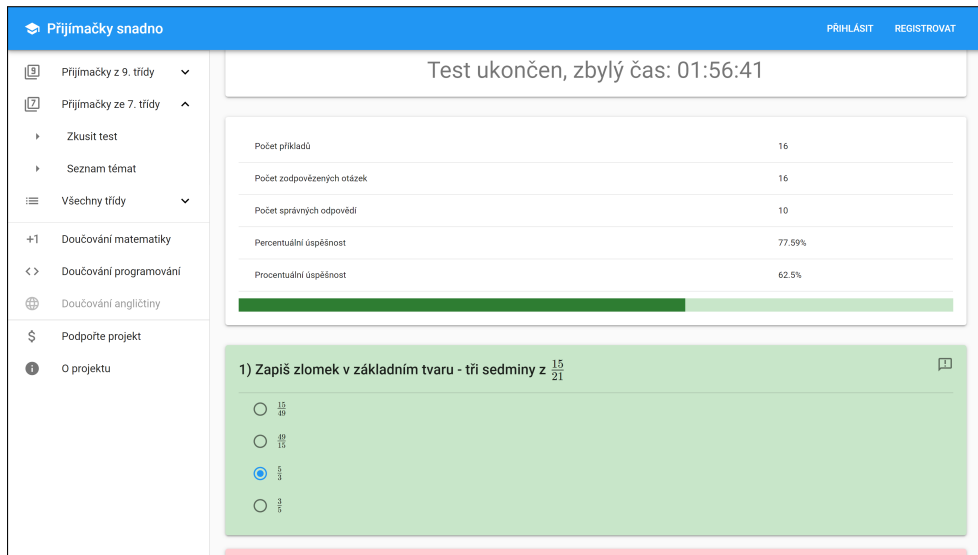
Obrázek D.2: Pokyny k testu

The screenshot shows the 'Příjmačky snadno' application interface with a test question. The top navigation bar is blue with the text 'Příjmačky snadno' on the left and 'PŘIHLÁSIT' and 'REGISTROVAT' on the right. The left sidebar is the same as in the previous screenshot. The main content area shows a question: '2) Vypočítej největší společný dělitel čísel 12 a 21.' Below the question are four radio button options: $\frac{7}{6}$, $\frac{6}{7}$, 6, and 2. The $\frac{6}{7}$ option is selected. Below the question is a text input field for the answer, containing the number '0'. The question is followed by a mathematical expression for question 3:

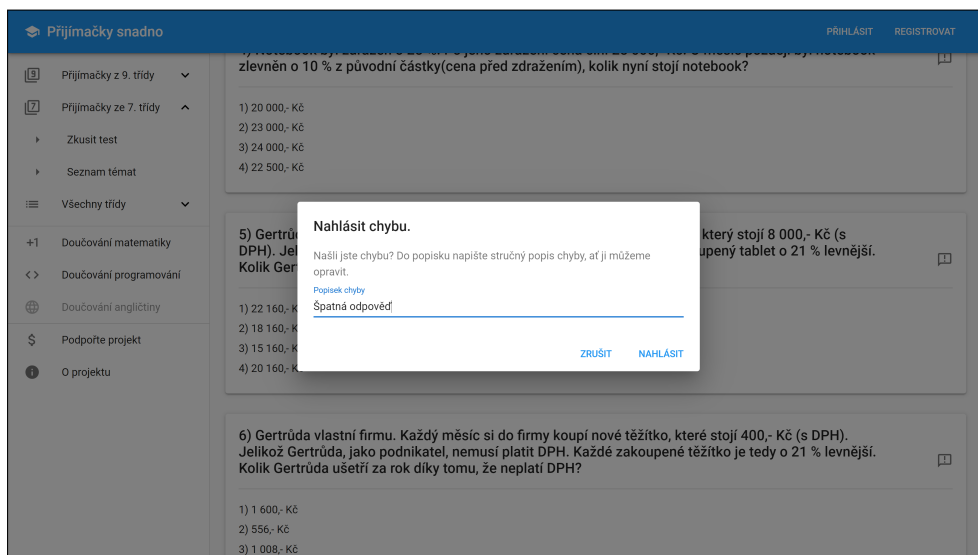
$$3) \frac{\frac{12 \cdot 2 - 2 \cdot 6}{2 \cdot 2} - \frac{6 \cdot 6}{6 \cdot 2}}{\frac{4 \cdot 3 - 2}{(-2+6) \cdot 2} - \frac{6 \cdot (2 \cdot 3)}{2}}$$

Below the expression is a text input field for the answer, containing the number '0'.

Obrázek D.3: Ukázka vyplňování testu

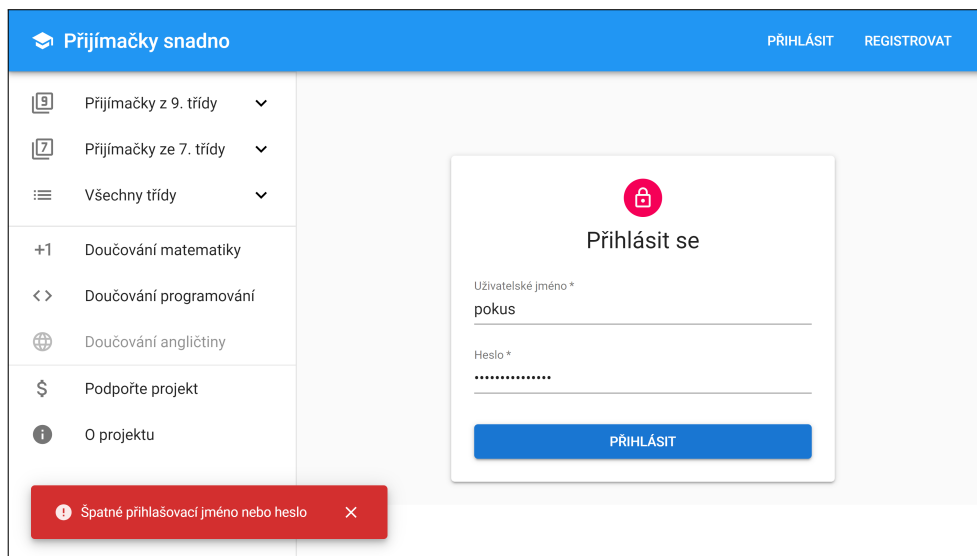


Obrázek D.4: Ukázka vyhodnocení testu



Obrázek D.5: Reportování příkladu

D. UKÁZKA VÝSLEDNÉ APLIKACE



Obrázek D.6: Přihlášení – špatné jméno nebo heslo



Obrázek D.7: Uživatel – statistika testů

Přijímačky snadno

Přijímačky z 9. třídy
Přijímačky ze 7. třídy
Všechny třídy

+1 Doučování matematiky
<> Doučování programování
Doučování angličtiny
\$ Podpořte projekt
i O projektu

Seznam absolvovaných testů

Datum zahájení testu	Doba trvání testu	Procentuální úspěšnost (%)	Počet správných odpovědí
09.4.2019 11:6:37	21 sekund	28.57	2/7
09.4.2019 11:7:5	30 sekund	42.86	3/7
09.4.2019 11:7:40	31 sekund	14.29	1/7
09.4.2019 11:8:18	57 sekund	42.86	3/7
09.4.2019 11:9:20	1 minut a 38 sekund	71.43	5/7
09.4.2019 11:11:5	26 sekund	71.43	5/7
09.4.2019 11:11:37	1 minut a 6 sekund	85.71	6/7
09.4.2019 11:12:50	3 minut a 8 sekund	100	7/7
30.4.2019 21:2:41	2 minut a 25 sekund	18.18	2/11
30.4.2019 21:23:39	44 sekund	45.45	5/11

Rows per page: 10 1-10 of 11

Obrázek D.8: Uživatel – přehled testů

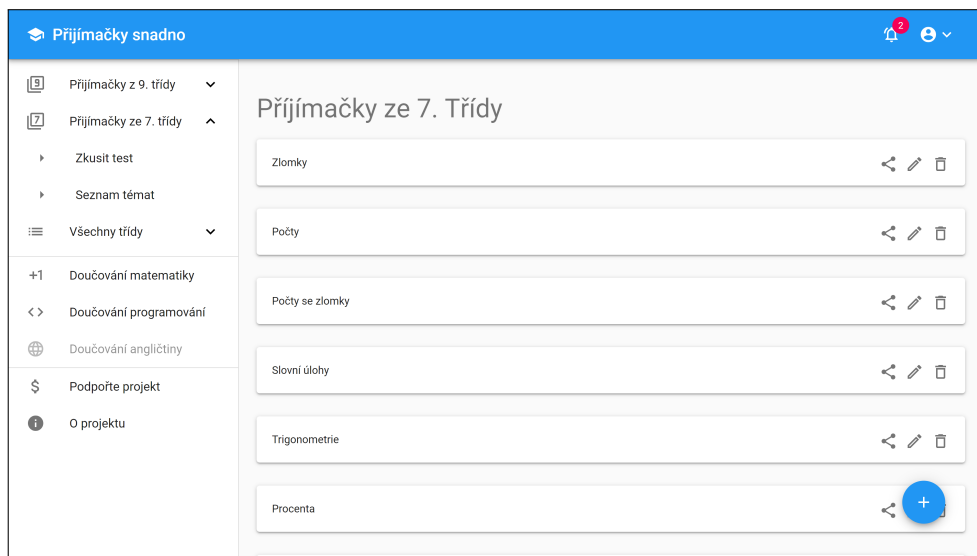
Nastavení testu ULOŽIT

Počet příkladů v testu z jednotlivých kategorií

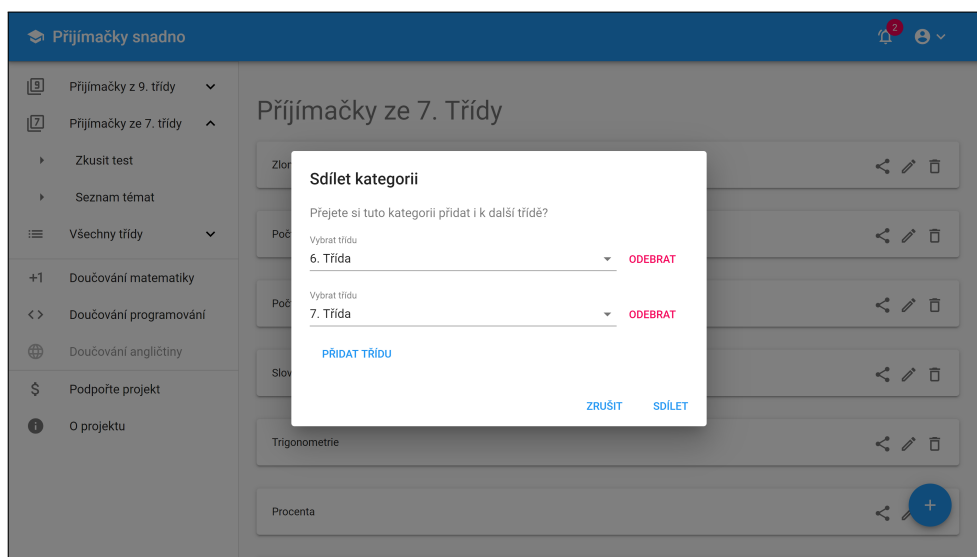
Kategorie	Počet příkladů v kategorii	Počet příkladů v testu
Geometrie	0	0
Počty	7	1
Poměry	6	1
Přímá a nepřímá úměra	4	1
Procenta	7	4
Trigonometrie	0	0
Čísla (reálná, přirozená, celá, racionální)	11	0

Obrázek D.9: Admin – nastavení testu

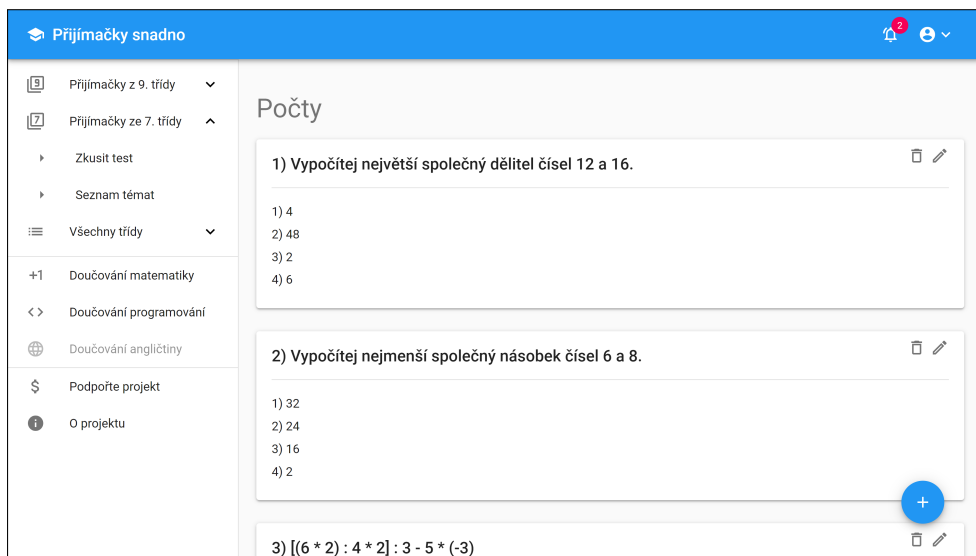
D. UKÁZKA VÝSLEDNÉ APLIKACE



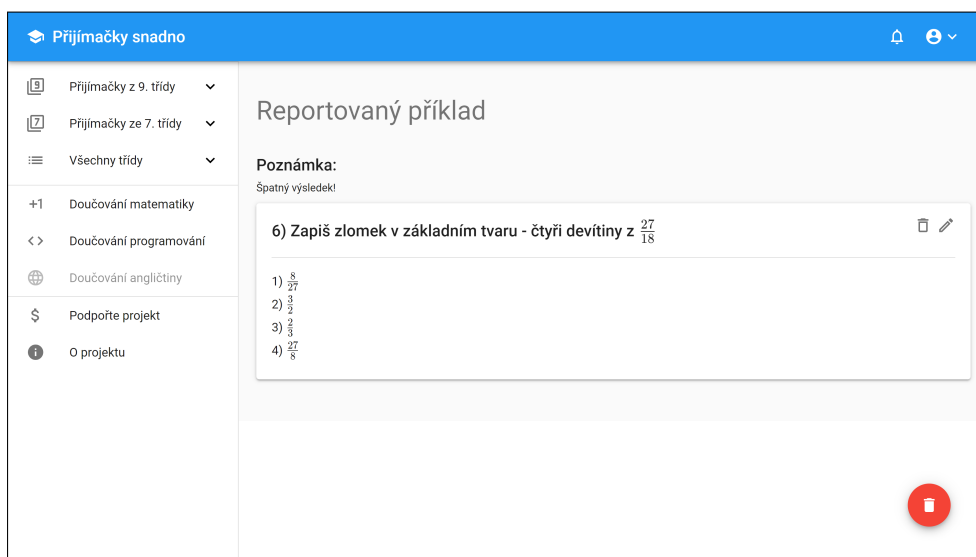
Obrázek D.10: Admin – seznam kategorií



Obrázek D.11: Admin – seznam kategorií – sdílení s jinými třídami



Obrázek D.12: Admin – seznam příkladů



Obrázek D.13: Admin – zobrazení reportovaného příkladu

D. UKÁZKA VÝSLEDNÉ APLIKACE

Vytvořit nový příklad

LaTeX zkratky

x^y $\frac{a}{b}$ \sqrt{x} A_{12}

Znak

Zadání

Zadání *

Vypočítej největší společný dělitel čísel 12 a 21.

NAHRÁT OBRÁZEK

Řešení

PRÁVĚ JEDNA SPRÁVNÁ ODPOVĚĎ N SPRÁVNÝCH ODPOVĚĎÍCH OTEVŘENÁ ODPOVĚĎ

Řešení *

3 ODSTRANIT ODPOVĚĎ

Řešení *

84 ODSTRANIT ODPOVĚĎ

Řešení *

7 ODSTRANIT ODPOVĚĎ

Řešení *

4 ODSTRANIT ODPOVĚĎ

PŘIDAT ODPOVĚĎ

Obrázek D.14: Admin – přidávání příkladů – 1. část

Zadání

Zadání *

Vypočítej největší společný dělitel čísel 12 a 21.

NAHRÁT OBRÁZEK

Řešení

PRÁVĚ JEDNA SPRÁVNÁ ODPOVĚĎ N SPRÁVNÝCH ODPOVĚĎÍCH OTEVŘENÁ ODPOVĚĎ

Řešení *

3 ODSTRANIT ODPOVĚĎ

Řešení *

84 ODSTRANIT ODPOVĚĎ

Řešení *

7 ODSTRANIT ODPOVĚĎ

Řešení *

4 ODSTRANIT ODPOVĚĎ

PŘIDAT ODPOVĚĎ

Náhled příkladu

1) Vypočítej největší společný dělitel čísel 12 a 21.

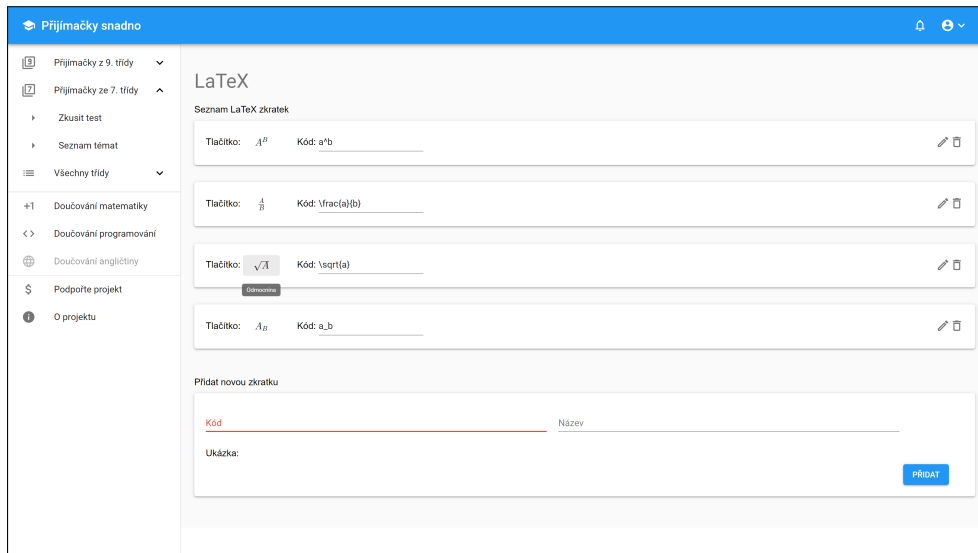
1) 3

2) 84

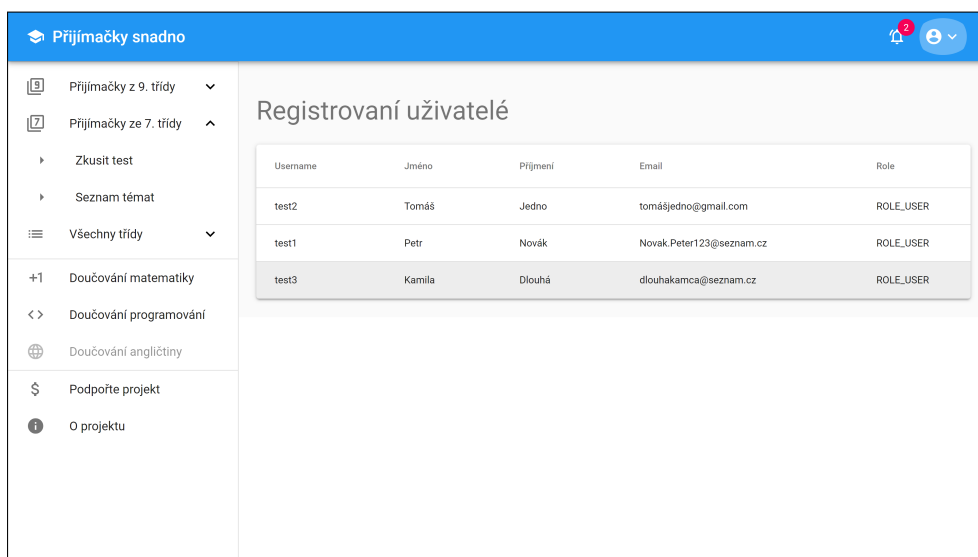
3) 7

4) 4

Obrázek D.15: Admin – přidávání příkladů – 2. část



Obrázek D.16: Admin – editování LaTeX zkratek



Obrázek D.17: Admin – výpis všech uživatelů

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
prirucka.pdf	instalační příručka
pg_dump.backup...	Skript pro vytvoření tabulek v databázi a inicializace
javadoc.....	adresář s HTML dokumentace
src	
├─ app	zdrojové kódy implementace
│ └─ server	zdrojové kódy serveru
│ └─ klient	zdrojové kódy klienta
└─ thesis	zdrojová forma práce ve formátu L ^A T _E X
exe	adresář se spustitelnou formou implementace
text	text práce
└─ thesis.pdf	text práce ve formátu PDF