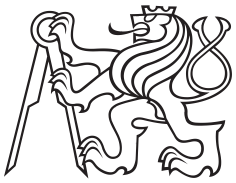
Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra Počítačů

# Formativní hodnocení ve výuce programování

Lukáš Vala

Školitel: RNDr. Ingrid Nagyová, Ph.D.
Obor: Softwarové inženýrství a technologie
Květen 2022

# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vala** Jméno: **Lukáš** Osobní číslo: **495547**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra počítačů**

Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Formativní hodnocení ve výuce programování**

Název bakalářské práce anglicky:

**Formative assessment in teaching programming**

Pokyny pro vypracování:

Cílem práce je analyzovat stávající možnosti dostupných online nástrojů formativního hodnocení ve výuce programování a navrhnout a implementovat systém umožňující tvorbu skupinových kvízů s automatickou kontrolou.
1. Analyzujte dostupné online nástroje formativního hodnocení (Kahoot, Socrative…). Zaměřte se na různé typy otázek, které nástroje podporují, a na formu zpětné vazby, kterou nabízí. Zvažte možnosti jejich praktického využití, specifikujte přednosti a nedostatky jejich využití při výuce programování.
2. Analyzujte funkční požadavky systému, který umožní vytvářet, archivovat a prezentovat testy. Zaměřte se na jednotlivé uživatele systému a na jejich potřeby v rámci systému.
3. Navrhněte a implementujte online systém formativního hodnocení ve výuce programování. Zvažte také možnost využití systému při vysvětlování učiva.
4. Definujte sadu testů ve vytvořeném nástroji a systém otestujte prezentací testů před skupinou studentů. Zhodnoťte náročnost tvorby testů a jejich prezentaci.

Seznam doporučené literatury:

1. Oficiální stránky prostředí Kahhot. Dostupné na: https://kahoot.com/
2. Oficiální stránky prostředí Socrative. Dostupné na: https://www.socrative.com/
3. Hetesi, S. (2021) "A comparative review of Kahoot and Socrative" in Teaching English as a Second Language Electronic Journal (TESL-EJ), 24(4). https://tesl-ej.org/pdf/ej96/m2.pdf
4. Sun, J. Wu, W. Rong and W. Liu. (2019) "Formative assessment of programming language learning based on peer code review: Implementation and experience report," in Tsinghua Science and Technology, vol. 24, no. 4, pp. 423-434, Aug. 2019, doi: 10.26599/TST.2018.9010109.
5. L. Benotti, M. C. Martnez and F. Schapachnik (2018) "A Tool for Introducing Computer Science with Automatic Formative Assessment," in IEEE Transactions on Learning Technologies, vol. 11, no. 2, pp. 179-192, 1 April-June 2018, doi: 10.1109/TLT.2017.2682084.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**RNDr. Ingrid Nagyová, Ph.D. kabinet výuky informatiky FEL**

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

Datum zadání bakalářské práce: **07.02.2022** Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

| RNDr. Ingrid Nagyová, Ph.D.<br>podpis vedoucí(ho) práce | podpis vedoucí(ho) ústavu/katedry | prof. Mgr. Petr Páta, Ph.D.<br>podpis děkana(ky) |
|---|---|---|

 © ČVUT v Praze, Design: ČVUT v Praze, VIC

## III. PŘEVZETÍ ZADÁNÍ

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

. _____
Datum převzetí zadání

_____
Podpis studenta

 © ČVUT v Praze, Design: ČVUT v Praze, VIC

# Poděkování

Děkuji mé vedoucí RNDr. Ingrid Nagyové, Ph.D. za to, že si na mě udělala čas vždy, když jsem to potřeboval a za její vždy podnětné a přínosné návrhy na zlepšení mé práce.


# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 18. května 2022

# Abstrakt

Práce se zabývá pozitivními dopady formativního hodnocení na studenty a jeho zanesení do výuky programování skrze kvízovou aplikaci. Tato aplikace umožňuje vyučujícím nenásilnou formou v reálném čase ověřit znalosti svých studentů a studentům dát zpětnou vazbu o jejich porozumění probírané látky v rámci vyučovaného předmětu, aniž by to jakkoli ovlivnilo jejich prospěch.
Vyučující v aplikaci nejprve vytvoří kvíz sestávající z několika otázek zaměřených zejména, ale ne výhradně na programování. Vyučující tedy může do otázky napsat programovací kód v několika programovacích jazycích. Poté na hodině se svými studenty tento kvíz promítá a studenti na jednotlivé otázky odpovídají pomocí svých zařízení (laptop, mobilní telefon). Po ukončení každé otázky je ihned viditelné její vyhodnocení, podle čehož může vyučující v případě potřeby otázku vysvětlit a studenti mohou k otázce pokládat dotazy.



**Klíčová slova:** vzdělávání, kvíz, zpětná vazba, formativní hodnocení

**Školitel:** RNDr. Ingrid Nagyová, Ph.D.



# Abstract

The focus of this bachelor thesis is to study positive effects of formative assessment on students and implementing it in teaching programming through a quiz application. This application enables teachers to forcelessly verify students' knowledge in real time and students to test their understanding of current topic in the studied subject without affecting their grades. In the application, the teacher first of all creates a new quiz containing multiple questions oriented mainly, but not exclusively on programming. The teacher is enabled to create programming questions in multiple programming languages. In their class the teacher then starts the quiz and students answer to the questions using their own devices (notebook, mobile phone). After each question, its evaluation is visible, which enables the teacher to further explain the topic if needed and students can ask their teacher if they do not understand anything about the question.



**Keywords:** education, quiz, feedback, formative assessment

**Title translation:** Formative assessment in teaching programming

# Obsah

# Obrázky



# Tabulky

# Kapitola 1
# Úvod

Hodnocení studentů hraje důležitou úlohu ve vzdělávacím procesu a formativní hodnocení je odborníky považováno za důležitou součást vyučovacího a učebního procesu. Formativní hodnocení je nástrojem pedagogické komunikace, informuje studenta o možnostech zlepšení a formuje tak jeho vývoj v rámci učebního procesu. [1] Momentálně existuje několik online aplikací pro formativní hodnocení studentů, například Kahoot nebo Socrative. Při používání těchto aplikací specificky ve výuce programování však narážíme na jistá omezení (více v kapitole 2 - Současný stav).
Cílem tohoto projektu je tedy navrhnout a vytvořit funkční nástroj pro podporu formativního hodnocení studentů ve výuce, který se mimo jiné zaměří i na oblast programování. Pro dosažení tohoto cíle bude nutné zabývat se v bakalářské práci následujícími body:

- Současný stav - Analýza dostupných online nástrojů formativního hodnocení se zaměřením na typy otázek, které nástroje podporují, možnosti jejich praktického využití a jejich přednosti a nedostatky při výuce programování
- Analýza požadavků systému, který umožní tvorbu a archivaci testů a také jejich prezentování ve skupině přibližně 20 studentů
- Kompletní návrh a implementace systému pro realizaci formativního hodnocení ve výuce programování. Tento systém by měl umožňovat tvorbu kvízů s automatickou kontrolou odpovědí a motivující zpětnou vazbou
- Otestování vytvořeného nástroje před skupinou studentů a zhodnocení náročnosti tvorby testů a jejich prezentace pro učitele a žáky

# Kapitola 2
# Analýza problému

## 2.1 Definice základních pojmů

Tématem této bakalářské práce je Formativní hodnocení ve výuce programování. Nejdříve se tedy podíváme na hodnocení studentů obecně, na to, jak jsou studenti typicky na školách hodnoceni a v čem může právě formativní hodnocení pomoci.

### 2.1.1 Hodnocení studentů

Hodnocení studentů je sdělení učitelů, které je určené žákům a vypovídá o míře jejich úspěšnosti ve vzdělávacím procesu. Vzdělávací výsledky žáků jsou hodnoceny oficiálně prostřednictvím klasifikace (známky) nebo prostřednictvím písemných zpráv (slovní hodnocení)[2]. Máme několik způsobů hodnocení studentů, konkrétně je dělíme na bezděčné / záměrné, normativní / kriteriální a sumativní / formativní[3]. Vzhledem k tomu, že tématem této práce je formativní hodnocení, budeme se dělení na sumativní / formativní hodnocení dále věnovat detailněji.

### 2.1.2 Sumativní hodnocení

Sumativní hodnocení je celkovým a souhrnným zhodnocením dosahovaných výkonů v dané oblasti a jejich přiřazení k určité úrovni v rámci nějaké škály.[1]
Jedním z hlavních cílů sumativního hodnocení je žáka nějakým způsobem zařadit, buď se záměrem diagnostikovat žáka anebo informovat ho o jeho úspěšnosti po delším úseku vykonané práce.[7]
Sumativní hodnocení může mít formu jak známek (na vysvědčení, v žákovské knížce), tak i slovní formu (např.: Jiřina po celý rok patřila k nejlepším žákům třídy). K sumativnímu hodnocení patří například zkoušení, písemné práce, závěrečné vysvědčení, ale i přijímací zkoušky na střední či vysokou školu nebo testy při pracovních konkurzech.[8]
Oddíl Londýnské univerzity (Evidence for Policy and Practice Information and Co-ordinating Centre at the University of London) našel přímou korelaci mezi výkony na státních standardizovaných testech a sebevědomím studentů. Studenti, kteří zaznamenali špatné výsledky na těchto testech, prožili snížení

sebevědomí, což způsobilo, že tito studenti měli ještě menší motivaci zaznamenat jakýkoli akademický úspěch. [9]
Sumativní hodnocení tedy v některých případech působí na studenty negativně. S tím ovšem může pomoci formativní hodnocení, které lze použít jako podporu sumativního hodnocení a může zamezit ztrátě motivace u studentů.

### 2.1.3 Formativní hodnocení

Formativní hodnocení je nástrojem pedagogické komunikace mezi žákem a učitelem [1]. Podává učiteli během učebního procesu informace o tom, jak se žák vypořádává s látkou, zda látce porozuměl, případně jaký druh pomoci potřebuje k zvládnutí učiva [4]. Formativní hodnocení je orientováno na dosažení společného cíle žáka a učitele – podporuje další efektivní učení a kompletní porozumění probírané látce. Zpětná vazba v procesu učení je poskytována učiteli i žákovi [5]. Zpětnou vazbu nepodává žákovi pouze učitel, ale i žák sám sobě, případně žáci sobě navzájem. Formativní hodnocení je průběžné a nabízí žákovi a učiteli postupy, jak výkon zlepšit ještě v průběhu samotného učení [1]. Učitelé používající formativní hodnocení nesrovnávají žáky mezi sebou, ale zaměřují se na dosahování výukových cílů každého z nich [6].
Poskytování pravidelné zpětné vazby je nezbytný element formativního hodnocení a výzkumy ukazují, že jakmile studenti vidí pokrok v jejich učení a cítí se podporováni svými učiteli, často prožívají pozitivní změnu v motivaci se učit a vzdělávat. [4]
Formativní hodnocení je tedy založeno zejména na těchto bodech: [6]

- Důrazu na komunikaci mezi žákem a učitelem i žáky navzájem
- Pravidelném a četném vyhodnocování žákovské práce (poskytnutí zpětné vazby – informace o tom, co se podařilo, na co je třeba se zaměřit v budoucnu a jak konkrétně je třeba postupovat). Toto může být realizováno například formou kvízů, čímž se z velké části zabývá tato bakalářská práce.
- Stanovení výukových cílů a sledování pokroku každého žáka na cestě k těmto cílům.

#### Jak formativní hodnocení ovlivňuje studenty a vyučující

Vyučující, kteří byli součástí zavádění formativního hodnocení do jejich výuky typicky jako hlavní změny, kterých si po zavedení změny všimli, uvádějí: [4]

- Lepší zaměření se na priority ve výuce, typicky na látku, které studenti nerozumí
- Umožnění úpravy vyučování podle toho, co studenti vyžadují
- Podpora společné práce studentů a vyučujících ke společnému cíli
- Zvýšení zapojení a motivace studentů

### 2.1.4 Souhrn

Sumativní hodnocení je tedy hodnocení konečné. Používá se k vyjádření toho, do jaké míry si student látku osvojil. [10]
Formativní hodnocení je průběžné, nehodnotí výsledek učení, ale jeho samotný proces. Užívá se ve chvíli, kdy se žák ještě učí a má prostor ke zlepšení. Způsobů poskytování zpětné vazby je mnoho, důležité ale je, aby objekt hodnocení vždy obdržel informaci o tom, do jaké míry jeho práce odpovídá požadavkům a co může udělat pro to, aby byl příště úspěšnější. [10]
Je možné využívat obě metody hodnocení zároveň, kde formativní hodnocení slouží jako podpora studentů při studiu a jako forma zamezení negativních dopadů sumativního hodnocení, jako je například pokles sebevědomí studentů. Studenti skrze formativní hodnocení, konkrétně předávání zpětné vazby vyučujícímu, mohou také ovlivnit, jakou formu sumativního hodnocení preferují a mohou jejich vyučujícímu navrhovat různé alternativy, například ve známkování, typu testů a jiné.

Kvízová aplikace, která bude v rámci této práce vyvíjena, má pomoci vyučujícím programování na vysokých, středních i základních školách se začleněním formativního hodnocení do jejich výuky. Měla by podporovat zejména:

- **Důraz na komunikaci mezi žákem a učitelem i žáky navzájem**
  Aplikace by měla umožnit diskuzi nad probíranou látkou. Vyučující by měl mít možnost reagovat na výsledky studentů v otázkách a dovysvětlovat látku tam, kde jsou výsledky špatné. Studenti by měli mít možnost ptát se kdykoli během kvízu svého vyučujícího na doplňující otázky.
- **Pravidelné a četné vyhodnocování žákovské práce: poskytnutí zpětné vazby o tom, co se podařilo, na co je třeba se zaměřit v budoucnu, a jak konkrétně je třeba postupovat**
  Pravidelné vyplňování kvízů o probírané látce předá studentům zpětnou vazbu o jejich znalostech a vyučujícímu o tom, jak studenti danou látku chápou a na co je třeba více se zaměřit.

## 2.2 Rozbor existujících řešení

V rámci této podkapitoly se budeme zabývat některými aplikacemi, které umožňují testování studentů skrz formativní či sumativní hodnocení. Na konci se nachází shrnutí této analýzy a co to bude znamenat pro naší vyvíjenou aplikaci.

### 2.2.1 Kahoot

Kahoot[1] je herně založená výuková platforma, která se používá pro kontrolu znalostí studentů, formativní hodnocení studentů, nebo pouze pro oživení výuky. S návštěvností 70 milionů unikátních uživatelů za měsíc se Kahoot řadí mezi nejpopulárnější herně založené výukové platformy. [11]
Existují 2 režimy aplikace, každý se nachází na jiné adrese. Kahoot.it slouží uživatelům pro připojení se ke spuštěnému kvízu pomocí PINu, který mu poskytuje organizátor kvízu. Na stránce kahoot.com se nachází informace o aplikaci a možnost registrace. Při registraci uživatel volí, zda má zájem o typ účtu basic, který je zdarma. Tato analýza se dále bude zabývat primárně tímto typem. Dále Kahoot nabízí účty premium, nebo premium+, které jsou placené. Premium obsahuje všechny funkcionality, co obsahuje účet typu basic a několik dalších navíc. Premium+ obsahuje vše, co obsahuje premium a opět ještě několik dalších funkcionalit navíc, ale za vyšší cenu.
Uživatelům, kteří volí verzi basic, jsou v tvoření kvízů poskytnuty 2 typy otázek a 1 typ poskytnutí informací mimo otázky (více v sekci *Tvoření otázek*). V basic verzi mohou uživatelé oproti premium verzím hostit menší počet lidí, kteří se mohou připojit do jejich kvízů. Aplikace také uživatele neustále upozorňuje na vše, o co používáním basic typu účtu přichází, což může znehodnocovat uživatelskou zkušenost.
Po registraci má uživatel možnost tvořit, ukládat, upravovat, mazat a spouštět kvízy. Má také přístup k vyhodnocením již spuštěných kvízů, kde je zobrazena celková úspěšnost, průměrná úspěšnost u jednotlivých otázek, obtížnost otázek a další podrobnosti.

#### Tvoření otázek

Kahoot nabízí několik typů otázek, ze kterých si lze při tvoření kvízů vybrat:

- Kvízová - Student má na výběr dvě až čtyři odpovědi, z nichž jedna až čtyři můžou být správné. Pokud je právě jedna správná odpověď, je po prvním kliknutí na jednu z odpovědí studentovo volba ihned zaznamenána a student již nemá možnost svou volbu změnit. Pokud je na výběr odpovědí víc, student má možnost nejprve vybrat odpovědi a pak svoji volbu potvrdit tlačítkem.
  Při vyhodnocení této otázky je viditelné, kolikrát byly jednotlivé odpovědi zvoleny studenty a které z nich jsou správné.

[1] Adresa aplikace: https://kahoot.com/

- True/False - Student má na výběr vždy pouze 2 odpovědi - true a false, z nichž jedna je správná. Po kliknutí na jednu z odpovědí je studentovo volba ihned zaznamenána a student ji již nemůže změnit.
  Vyhodnocení této otázky je stejné jako u té předchozí, což znamená, že je zobrazeno, kolikrát studenti zvolili možné odpovědi, a která z nich je správná

- Slide - Tento typ otázky je klasifikován jako prezentační, slouží tedy vyučujícímu k poskytnutí vysvětlivek nebo zpětné vazby k právě zkoušené látce. Lze do něj vložit název, popis a obrázek nebo video.

Při tvoření otázek typu Kvízová a True/False je také uživatelům poskytnuto několik možností, jak otázku upravit:

- Nastavení časového limitu

- Nastavení bodového ohodnocení otázky

Uživatelům, kteří jsou ochotni za Kahoot platit, je poskytnuto více typů otázek, těmi se ale v rámci této analýzy již nebudu zabývat.

#### Programovací kód v otázkách

Pokud chce vyučující do otázky zanést programovací kód, nemá jinou možnost, než vložit výstřižek bloku kódu jako obrázek. Tento obrázek pak ale není vidět u vyhodnocení otázky. Jestliže pak chce u vyhodnocení vyučující vysvětlovat nejasnosti okolo daného kódu, je nucen přepínat mezi záložkami na jeho prohlížeči.

#### Výhody

- Responsivita - v drtivé většině případů se změny na stránce dějí instantně a uživatelé nemusí na nic zbytečně čekat.

- Možnost upravovat velké množství detailů u každé otázky, jako například časový limit na odpověď nebo bodové ohodnocení otázky.

- Podrobné vyhodnocení celých kvízů pro každé jejich spuštění. Je k dispozici spoustu informací, jako například průměrná procentuální úspěšnost studentů nebo nejobtížnější otázka v kvízu. Lze také najít informace ke každé otázce - kolikrát byly zvoleny různé odpovědi, procentuální úspěšnost nebo kolik studentů se zdrželo odpovědi.

#### Nevýhody

- Chybí podpora programovacího kódu v otázkách. Pokud je kód vložen formou obrázku, není vidět u vyhodnocení, což je důležité pro vyučujícího, pokud chce kód jakkoliv komentovat.

- Neustálé upozorňování uživatele, pokud nemá placenou verzi Kahootu. Základní verze je v mnoha ohledech dostačující, tato upozornění však velmi znepříjemňují uživatelskou zkušenost.
- Není na první pohled zřejmé, zda je u kvízové otázky možné volit více odpovědí. Pokud není a student si myslí opak, klikne na jednu z možností a jeho odpověď je ihned odeslána, aniž by ji mohl dál měnit, což může být matoucí.

### 2.2.2 Socrative

Socrative[2] v porovnání s Kahootem nabízí čistší a přehlednější vzhled. Úvodní stránka prezentuje uživateli přehledně v bodech základní informace, které je třeba při manipulaci s aplikací znát. Také zde lze najít video, které uživatele uvádí do možnosti placené verze aplikace. Jedním z největších rozdílů mezi Kahootem a Socrative je, že Socrative neumožňuje okamžité vyhodnocení otázek pro vyučujícího. Hlavní stránka se nachází na adrese Socrative.com, kde najdou vše potřebné všichni uživatelé. Při registraci si uživatel pouze vybírá, zda je student, nebo učitel a podle toho je mu přizpůsobena stránka po přihlášení. Pokud uživatel zvolí, že je student, stačí mu pouze zadat ID místnosti, které je mu poskytnuto jeho organizátorem kvízu. Následně se účastní kvízů, které organizátor spouští v různých režimech. Vyučující má možnost zaregistrovat se pomocí svých údajů a následně ve svém účtu vytvářet kvízy a manipulovat s nimi.
Po vytvoření kvízu jej může vyučující spustit ve třech režimech:

- Teacher paced - Učitel řídí, na které otázky můžou v danou chvíli studenti odpovídat. To znamená, že učitel otevře první otázku, studenti na ni odpoví, a poté čekají, než jim učitel umožní odpovídat na další otázky. Učitel ale bohužel není schopný vědět, zda již na otázku odpověděli všichni studenti. K vyhodnocení otázek má vyučující přístup až po ukončení kvízu.
- Instant feedback - Vyučující v tomto případě pouze spustí kvíz a studenti odpovídají na otázky svým tempem, nemohou však mezi otázkami volně přepínat, musí nejdříve potvrdit odpověď na danou otázku, a až poté se můžou posunout k další otázce. Učitel může neustále sledovat jejich rychlost a úspěšnost odpovídání na otázky.
- Open navigation - Stejný systém jako u Instant feedback, studenti však mohou volně přepínat mezi otázkami a potvrzují jen dokončení kvízu, pokud mají odpovězeno na všechny otázky.

Vyučující mohou při spuštění kvízu také nastavit několik parametrů kvízu, jako povinné zadávání jmen, náhodné míchání odpovědí a otázek nebo viditelný feedback u každé otázky.

[2] Adresa aplikace: https://www.socrative.com/

### Tvoření otázek

Bezplatná verze Socrative nabízí vyučujícím tvoření těchto typů otázek:

- Kvízová - Student má na výběr dvě a více odpovědí. Na rozdíl od Kahootu v tomto typu otázky student musí vždy svou volbu po svém rozhodnutí potvrdit tlačítkem. Pokud je pouze jedna správná odpověď, student jich nemůže vybrat více, po vybrání jiné odpovědi se zruší výběr původní odpovědi a dále je zaznamenána jen nová volba. Po potvrzení výběru je studentovi zobrazeno, zda zvolil správné možnosti.
- True/False - Stejný systém jako u kvízové otázky, student má ale vždy na výběr pouze odpovědi true a false.
- Krátká odpověď - Zde je studentovi položena otázka a on odpovídá formou krátkého textu, většinou ne více než pár slov. Po potvrzení odpovědi je studentovi zobrazeno, zda zvolil správné slovo nebo sousloví.

Socrative nenabízí typ otázky Slide jako Kahoot, který slouží pro přidání vysvětlivek k otázkám, tuto funkcionalitu však nabízí přímo u každé otázky. Učitel do nich může vysvětlivku napsat a ta je následně studentům po potvrzení jednotlivých odpovědí zobrazena, pokud tak zvolí vyučující při spuštění kvízu.

### Programovací kód v otázkách

Socrative podobně jako Kahoot explicitně neposkytuje možnost zadávat programovací kód do otázek, placená verze umožňuje pouze tvoření matematických rovnic. Pokud tedy chce vyučující do otázky přidat blok programovacího kódu, je to nucen udělat stejně jako u Kahootu formou obrázku, který se ale nezobrazuje u vyhodnocení otázek.

### Výhody

- Příjemný vzhled stránek a manipulace s aplikací
- Intuitivní tvoření otázek4
- Jednoznačné odpovědi na kvízové otázky - student musí ve všech případech svůj výběr potvrdit tlačítkem.
- Responsivita - Instantní změny na stránkách
- Několik režimů kvízů, vyučující si může režim vybrat podle potřeby

### Nevýhody

- Chybí podpora programovacího kódu v otázkách. Pokud je kód vložen formou obrázku, není vidět u vyhodnocení, což je důležitého pro vyučujícího, pokud chce kód jakkoliv komentovat

- Vyhodnocení otázek - chybí celkové vyhodnocení každé otázky po jejím skončení. Pokud má k průběžnému skóre studentů vyučující přístup, nemůže řídit postup studentů kvízem a zastavovat se u jednotlivých otázek

### 2.2.3 Hot Potatoes

Aplikace Hot Potatoes[3] byla poprvé zveřejněna již v roce 1998 [12]. Aplikace již v delší dobu kromě minoritních aktualizací není vyvíjena, proto jí zůstal velmi zastaralý vzhled. Nabízí však stále velmi zajímavé možnosti pro vyučující a jejich studenty a celá aplikace je přístupná zcela zdarma.
Na úvodní stránce se nachází informace o aplikaci, často pokládané otázky a možnost stáhnout aplikaci do svého počítače. V aplikaci následně uživatelé mohou vytvářet kvízy a exportovat je do html souboru, který se dá otevřít ve webovém prohlížeči.

#### Tvoření otázek

Hot Potatoes i přes svou velmi dlouhou existenci nabízí co se týče typů otázek o dost více možností než Socrative a Kahoot v bezplatných verzích:

- JMatch - Přiřazování. Vyučující má při tvorbě této otázky k dispozici několik textových polí ve dvou sloupcích. Do řádků píše výroky, které patří k sobě. Když pak spustí tuto otázku student, pravý sloupec je náhodně zamíchán a jeho úkolem je tento sloupec pomocí technologie drag'n drop přeskládat tak, aby k sobě pasovaly pole v levém a pravém sloupci. Po kliknutí na tlačítko Check se studentovi ukáže procentuální úspěšnost jeho přiřazení.

- JQuiz - Kvízová otázka. Vyučující má na výběr nastavit několik specifikací otázky pro uživatele, který na otázku odpovídá - multi-select, short-answer, hybrid a multiple-choice. Vyučující navíc může ke každé odpovědi poskytnout vysvětlivky, které mohou objasňovat, proč jsou dané odpovědi správné či chybné. Dříve zmíněné specifikace tohoto typu otázky přináší spoustu možností pro vyučujícího, pokud například volí multiple-choice možnost, stačí studentovi, aby klikl na jednu ze správných odpovědí a tím dosáhne stoprocentního úspěchu v otázce. Pokud ale volí multi-select možnost, student musí pro stoprocentní úspěch správně zvolit všechny odpovědi.

- JCloze - Vyučující vytvoří text a některá slova vymaže, studentův úkol je pak tyto slova do textu doplnit. Vyučující mu může poskytnout nápovědy.

- JCross - Jedná se o klasickou křížovku, ve které student hledá schovaná slova.

- JMix - Vyučující rozdělí větu nebo více vět do několika částí a studentův úkol je pak tyto části poskládat ve správném pořadí.

[3] Adresa aplikace: https://hotpot.uvic.ca/

- Masher - Umožňuje kombinovat předchozí typy otázek do kvízů. Student pak v předvoleném pořadí postupně odpovídá na všechny otázky a následně je vyhodnoceno jeho skóre. Předchozí typy otázek totiž vyučující může testovat pouze jednotlivě.

### Programovací kód v otázkách

V této aplikaci se také nenachází podpora pro otázky obsahující programovací kód. Překvapivě ale i tato aplikace poskytuje možnost vložit do otázky obrázek, proto je možné vložit programovací kód do něj.

### Výhody

- Aplikace je zcela zdarma
- Velké množství typů otázek
- Spoustu zajímavých možností u kvízové otázky
- Automatická a okamžitá kontrola kvízů pro jednotlivé studenty. Studenti toto hodnocení vidí sami na svých přístrojích

### Nevýhody

- Vzhled aplikace, což je ale vzhledem k jejímu stáří očekávatelné
- Není možné spouštět kvízy v reálném čase. Lze je pouze vygenerovat a přeposlat studentům. Vyučující nemá možnost kvíz jakkoliv ovládat, studenti si jej pouze otevřou na svých zařízeních a podle sebe kvíz vyplňují. Vyučující také nemá možnost vidět hodnocení studentů.

### 2.2.4 Shrnutí existujících řešení

Z předchozí analýzy vyplývají informace, které uvádí tabulka 2.1:

| | **Kahoot** | **Socrative** | **Hot potatoes** |
|---|---|---|---|
| Vzhled | Dětský vzhled, mnoho barev a zbytečných efektů | Příjemný a čistý vzhled | Velmi zastaralý vzhled |
| Přehlednost | Příliš mnoho informací, ve kterých se uživatel lehce ztratí | Občas je obtížné najít jednotlivé sekce, kromě toho ale celkově dobrá přehlednost | Přiměřená přehlednost, u tvoření otázek jsou ale nejasné některé funkcionality |
| Podpora programovacího kódu | Žádná | Žádná | Žádná |
| Množství typů otázek | 2 typy otázek a 1 prezentační otázka - Slide | 3 typy otázek | 5 typů otázek |
| Formativní hodnocení | Velmi dobré, po každé otázce vidí vyučující vyhodnocení se všemi důležitými detaily | Téměř žádné, vyučující nemá možnost pozastavovat se u každé otázky a nemá ani možnost vidět vyhodnocení jednotlivých otázek | Žádné, vyučující může studentům pouze poslat vygenerovaný test, ti si ho pak sami vyplní |
| Formativní samohodnocení | Dobré, student po každé otázce vidí, zda byl při výběru odpovědí úspěšný a které odpovědi byly správné | Velmi dobré, student může po každé otázce vidět, zda zvolil správné odpovědi, které všechny odpovědi byly správné a navíc si může přečíst vysvětlivky k otázce, pokud je vyučující poskytl | Velmi dobré, student po každé otázce vidí, zda volil správně a které odpovědi byly správné |
| Použitelnost na mobilním telefonu pro studenty | S odpovídáním na otázky není problém | S odpovídáním na otázky není problém | Užívání na mobilním telefonu není možné |

**Tabulka 2.1:** Shrnutí existujících řešení

I. U čeho se inspirovat:

1. Kahoot - Průběžné vyhodnocení otázek. Nutno ale přidat možnost vidět u vyhodnocení celé zadání včetně programovacího kódu nebo obrázků
2. Typy otázek - Kvízové a true/false otázky společně s prezentační otázkou - slide
3. Unikátní PIN vytvořený pro studenty, pomocí kterého se můžou připojit do kvízů

II. Co dělat jinak, než tyto aplikace:

1. Přidat podporu programovacího kódu v otázkách
2. Zkombinovat profesionální a přehledný vzhled Socrativu s možnostmi formativního hodnocení studentů Kahootu

# Kapitola 3

# Požadavky aplikace

V této kapitole se nachází zpracované požadavky aplikace. Konkrétně se jedná následující požadavky:

- Cílová skupina - pro které uživatele je aplikace tvořena
- Business požadavky - požadavky od cílových skupin na funkcionalitu systému - co jim má aplikace umožňovat
- Funkční požadavky - specifikace konkrétních funkcionalit sytému
- Kvalitativní požadavky - definují systémové atributy jako bezpečnost, spolehlivost, výkon nebo použitelnost [13]

## 3.1 Cílová skupina

Aplikace je určena pro tyto cílové skupiny:

- **Vyučující na vysokých, středních a základních školách zaměřených na výuku programování** - Mohou aplikaci využít pro získání zpětné vazby o znalostech jejich studentů. Také mohou chtít svou výuku aplikací obohatit a přinést do výuky něco netradičního, co upoutá pozornost studentů během vyučovacích hodin.
- **Studenty** - Mohou aplikaci využívat společně se svým vyučujícím k zadávání odpovědí na jednotlivé otázky a k ověření svých znalostí probírané látky.
- **Kdokoli, kdo má nějaké publikum, které chce zaujmout, nebo od kterého chce získat zpětnou vazbu** - Příklady: Doučující na školách, řečníci na workshopech

## 3.2 Business požadavky

Tato kapitola definuje požadavky na software podle cílové skupiny uživatel. Jedná se o nejvyšší úroveň požadavků na software. Požadavky byly definovány na základě rozhovorů se zadavatelem projektu – vyučujícím na ČVUT FEL. Pro definici jednotlivých požadavků je použito tzv. User Story, které je populární v agilních metodikách a řídí se následující šablonou:

Jako *[kdo]* potřebuji *[co]*, protože *[proč].*[14]

**BR 1** Jako vyučující na vysoké, střední, nebo základní škole zaměřené na výuku programování potřebuji od mých studentů vědět, jaké jsou jejich znalosti o probírané látce, protože mi to dává informace o tom, jak úspěšná je má výuka dané látky, podle kterých pak mohu upravit svůj způsob výuky.

**BR 2** Jako vyučující na vysoké, střední, nebo základní škole zaměřené na výuku programování potřebuji svým studentům dát možnost ověřit si své znalosti na pravidelné bázi, protože chci, aby studenti o svých znalostech probírané látky věděli a tomu přizpůsobili své studium.

**BR 3** Jako vyučující na vysoké, střední, nebo základní škole zaměřené na výuku programování potřebuji na začátku hodiny/přednášky zaujmout pozornost svých studentů, protože studenti jsou na začátku hodin/přednášek často nesoustředění.

**BR 4** Jako student programování na vysoké, střední, nebo základní škole potřebuji získat zpětnou vazbu o svých znalostech, protože se mi často stává, že svoje znalosti považuji za dostatečné a poté se nestíhám připravit na písemné testy nebo zkoušky.

**BR 5** Jako student programování na vysoké, střední, nebo základní škole potřebuji být upozorněn na detaily v probírané látce, protože mi často při výuce unikají.

## 3.3 Funkční požadavky

V některých požadavcích je nepřihlášený uživatel pro zpřehlednění specifikován jako student. V systému však student jako samostatná entita evidován nebude. Některé z těchto požadavků vychází z kapitoly 2 - shrnutí současného stavu. Za takovými požadavkami je odkaz specifikován v hranatých závorkách. Funkční požadavky se zde řídí se následující šablonou:

Systém bude umožňovat *[komu] [co]*

**FR 1** Systém bude umožňovat nepřihlášenému uživateli zaregistrovat se pomocí e-mailové adresy a hesla

**FR 2** Systém bude umožňovat nepřihlášenému uživateli přihlásit se k existujícímu účtu pomocí e-mailové adresy a hesla

**FR 3** Systém bude umožňovat nepřihlášenému uživateli vyžádat zaslání nového hesla na jeho e-mailový účet v případě jeho zapomenutí

**FR 4** Systém bude umožňovat přihlášenému uživateli změnit své heslo

**FR 5** Systém bude umožňovat učiteli vytvářet kvízy

**FR 5.1** Systém bude umožňovat učiteli v kvízech vytvářet různé typy otázek podle potřeby. Konkrétně se bude jednat o tyto typy otázek:

- Kvízová [I. 2.]
- True/False [I. 2.]
- Puzzle
- Přiřazování
- Word cloud

**FR 5.2** Systém bude umožňovat učiteli v kvízech mezi otázkami poskytovat dodatečné informace k tématu pomocí otázky typu slide [I. 2.]

**FR 5.3** Systém bude umožňovat učiteli do otázek a jednotlivých odpovědí psát programovací kód v jednotlivých programovacích jazycích [II. 1.]. Podporovány budou tyto jazyky:

- Java
- C
- C++
- Python

**FR 5.4** Systém bude umožňovat učiteli do otázek vkládat obrázky

**FR 5.5** Systém bude umožňovat učiteli nastavit časový limit na zodpovězení jedné otázky

**FR 5.6** Systém bude umožňovat učiteli v kvízu nastavit náhodné míchání otázek a odpovědí na ně

**FR 6** Systém bude umožňovat učiteli uložit vytvořený kvíz pro pozdější využití

**FR 7** Systém bude umožňovat učiteli uložený kvíz smazat

**FR 8** Systém bude umožňovat učiteli vytvořit kopii uloženého kvízu

**FR 9** Systém bude umožňovat učiteli vygenerovat unikátní pin, díky kterému se budou studenti moci připojit do kvízu [I. 3.]

**FR 10** Systém bude umožňovat studentovi zúčastnit se kvízu pomocí unikátního pinu poskytnutého jeho učitelem [I. 3.]

**FR 11** Systém bude umožňovat učiteli opakovaně spustit uložený kvíz

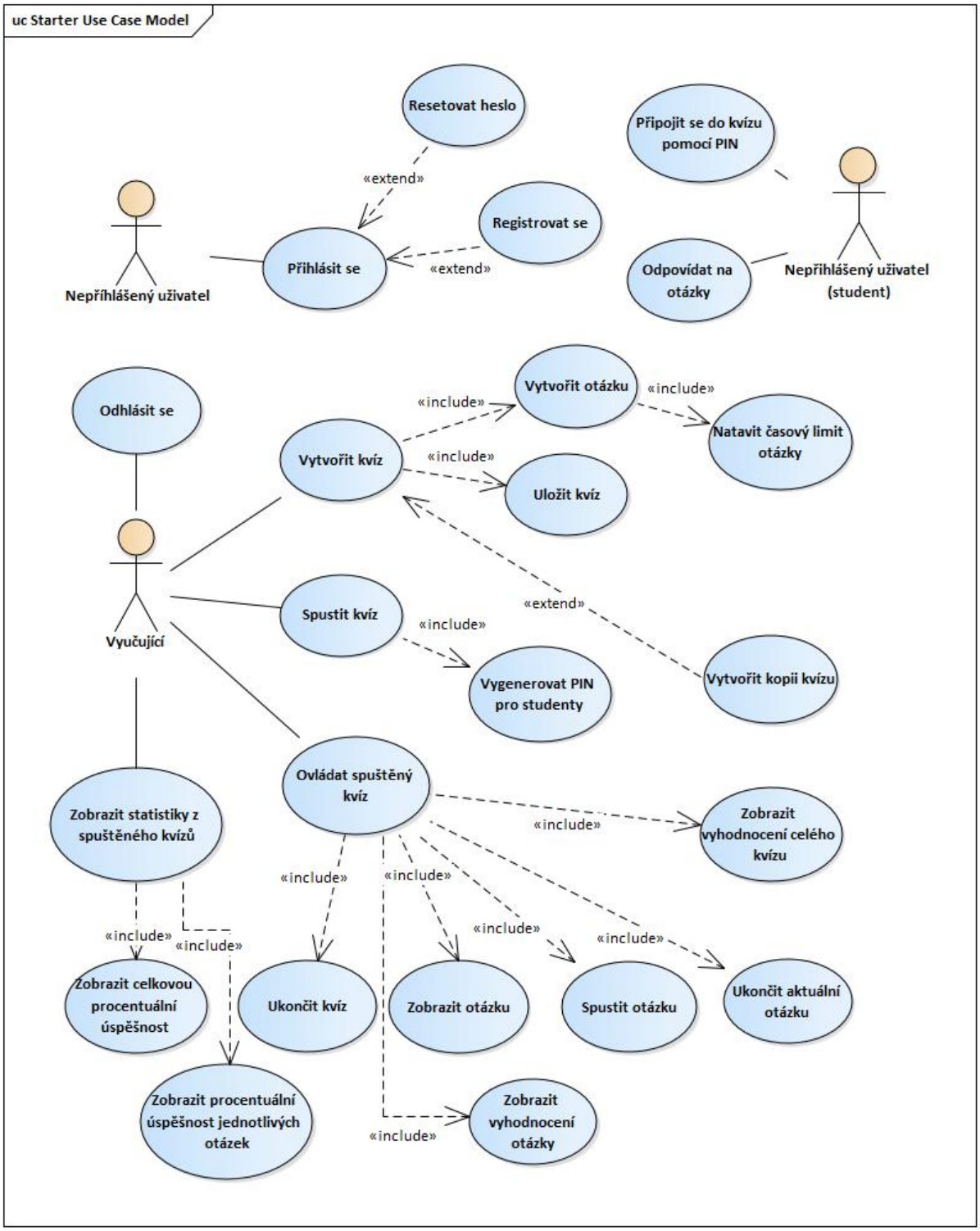
**FR 12** Systém bude umožňovat učiteli ovládat jím spuštěný kvíz. Ovládáním kvízu se myslí tyto činnosti:

- Zobrazit otázku
- Spustit otázku
- Zobrazit vyhodnocení otázky včetně programovacího kódu, obrázků a jiných médií. V tomto vyhodnocení by mělo být viditelné, kolik studentů odpovědělo na otázku správně. - [I. 1.]
- Ukončení kvízu

**FR 13** Systém bude umožňovat studentovi odpovídat na jednotlivé otázky na svých mobilních zařízeních

**FR 14** Systém bude umožňovat učiteli zobrazit statistiky z již spuštěného kvízu

S funkčními požadavky souvisí následující obrázek 3.1 obsahující diagram užití, který zpracovává většinu požadavků vizuálně.




**Obrázek 3.1:** Diagram užití

## 3.4 Kvalitativní požadavky

**QR 1** Webové rozhraní funkční na prohlížečích Microsoft Edge, Google Chrome a Mozilla Firefox - Webová aplikace musí být optimalizována pro internetové prohlížeče Google Chrome od společnosti Google, Microsoft Edge od společnosti Microsoft a Mozilla Firefox od společnosti Mozilla Corporation. Očekává se, že aplikace bude plně funkční i na jiných prohlížečích, avšak tato skutečnost již není zaručena.

**QR 2** Bezpečnost uživatelů a jejich dat - Musí být zaručena bezpečnost dat uživatelů, zejména jejich přihlašovací údaje. Heslo nesmí být ukládáno v otevřené podobě.

**QR 3** Integrita dat - Data, která uživatelé ukládají do databáze, musí být uložena jako celek a nesmí dojít ke ztrátě části dat.

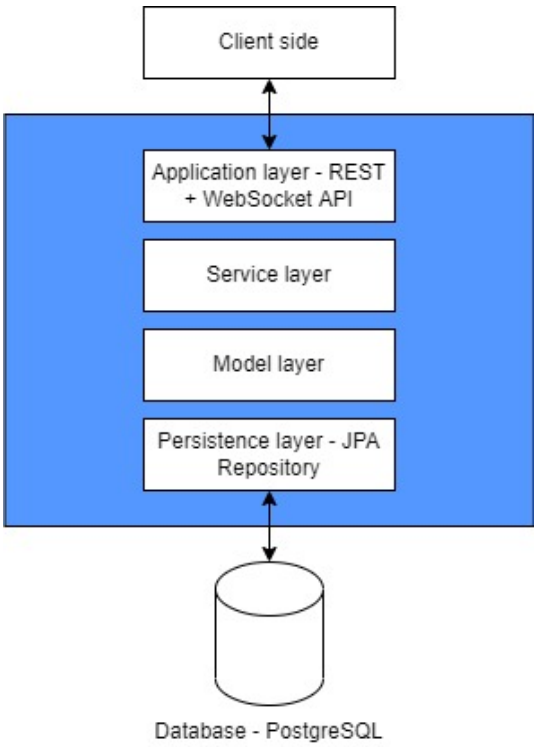
**QR 4** Uživatelská přívětivost a přehlednost GUI - Uživatel by se měl sám v aplikaci bez vnější pomoci orientovat. Uživatelské rozhraní tedy musí poskytovat předvídatelný a konzistentní vzhled.

**QR 5** Optimalizace rozhraní pro mobilní telefony - Studenti, kteří se připojí ke spuštěnému kvízu, musí mít možnost na otázky odpovídat na svých mobilních telefonech. Tato část uživatelského rozhraní tedy musí být mobilním telefonům uzpůsobena.

# Kapitola 4

# Architektura systému

Ze zpracovaných analýzy a požadavků aplikace se nabízí kvízovou aplikaci implementovat ve formě webové aplikace v modelu klient-server. Klient a server společně budou komunikovat pomocí technologií WebSocket a REST API. Klient odešle JSON data serveru. Ten data podle potřeby zpracuje a odešle požadavek do databáze pro získání potřebných dat. Tyto data pak pošle opět ve formátu JSON zpátky klientovi. V případě komunikace pomocí WebSocket bude také server posílat zprávy klientům, které má označené jako své posluchače. Architekturu abstraktně znázorňuje obrázek 4.1.




**Obrázek 4.1:** Architektura systému

## 4.1 Serverová část

Při návrhu serverové části se vzalo v potaz několik různých softwarových architektur.

1. **Pipes and filters**
   Tato architektura má 2 základní komponenty - **pipes** a **filters**. Filters reprezentují komponenty v aplikaci, které jsou na sobě navzájem nezávislé a pipes slouží jako konektory mezi těmito entitami. Hlavním účelem této architektury je vzít data na vstupu a skrz několik různých filtrů je přetvořit na požadovaný výstup.[15] Toto znázorňuje následující obrázek 4.2.


**Obrázek 4.2:** Transformace dat v architektuře pipes and filters[15]

   Pipes and filters architektura má několik výhod, mezi nimi možnost změny jednotlivých komponent (filters), aniž bychom ovlivňovali ostatní komponenty, nebo fakt, že uživatel nemusí vědět nic o konkrétní implementaci, stačí mu pouze zadefinovat vstupní data a formát výstupních dat. Tato architektura má ale velmi specifické možnosti využití, jako například implementace jazykových kompilerů nebo UNIX shellů. Tato architektura je také vhodná použít, pokud se dá řešený problém rozdělit do sady nezávislých kroků[16]. Vyvíjená aplikace se ale vždy nedá rozdělit zcela nezávisle a také ne vždy je třeba přetvořit typ dat na vstupu na jiný typ dat na výstupu. Proto se nejspíše dá najít jiná architektura, která by byla pro implementaci webové aplikace vhodnější.

2. **N-tier architektura**
   N-tier architektura je koncept klient-server architektury v softwarovém inženýrství, kde jednotlivé části systému jako prezentace, zpracování nebo správa dat jsou logicky i fyzicky odděleny. Tyto části jsou spuštěny na oddělených zařízeních, aby byly schopny poskytovat funkcionalitu na nejvyšší úrovni díky tomu, že nesdílí zdroje s jinými funkcionalitami. Toto oddělení zjednodušuje správu jednotlivých částí, protože změna v jedné části neovlivňuje jiné části. Pokud se v aplikaci vyskytne nějaký problém, typicky se týká pouze jedné z částí a můžeme ho izolovat.[17] Tato architektura tedy má několik výhod, mezi které patří:

   - Škálovatelnost - Systém je dobře rozšiřitelný, protože jednolitvé části spolu napřímo nijak nesouvisí
   - Individuální správa - Brání vzniku problému ve více částech systému najednou, možnost izolace problému pouze v jedné části.
   - Flexibilita - Možnost expanze systému dle nových požadavků
   - Bezpečnost - Každá část může být zabezepečena podle jejích potřeb

N-tier architektura se také využívá i pro tvorbu webových aplikací, proto o ní lze uvažovat jako o jedné z možností pro implementaci aplikace.

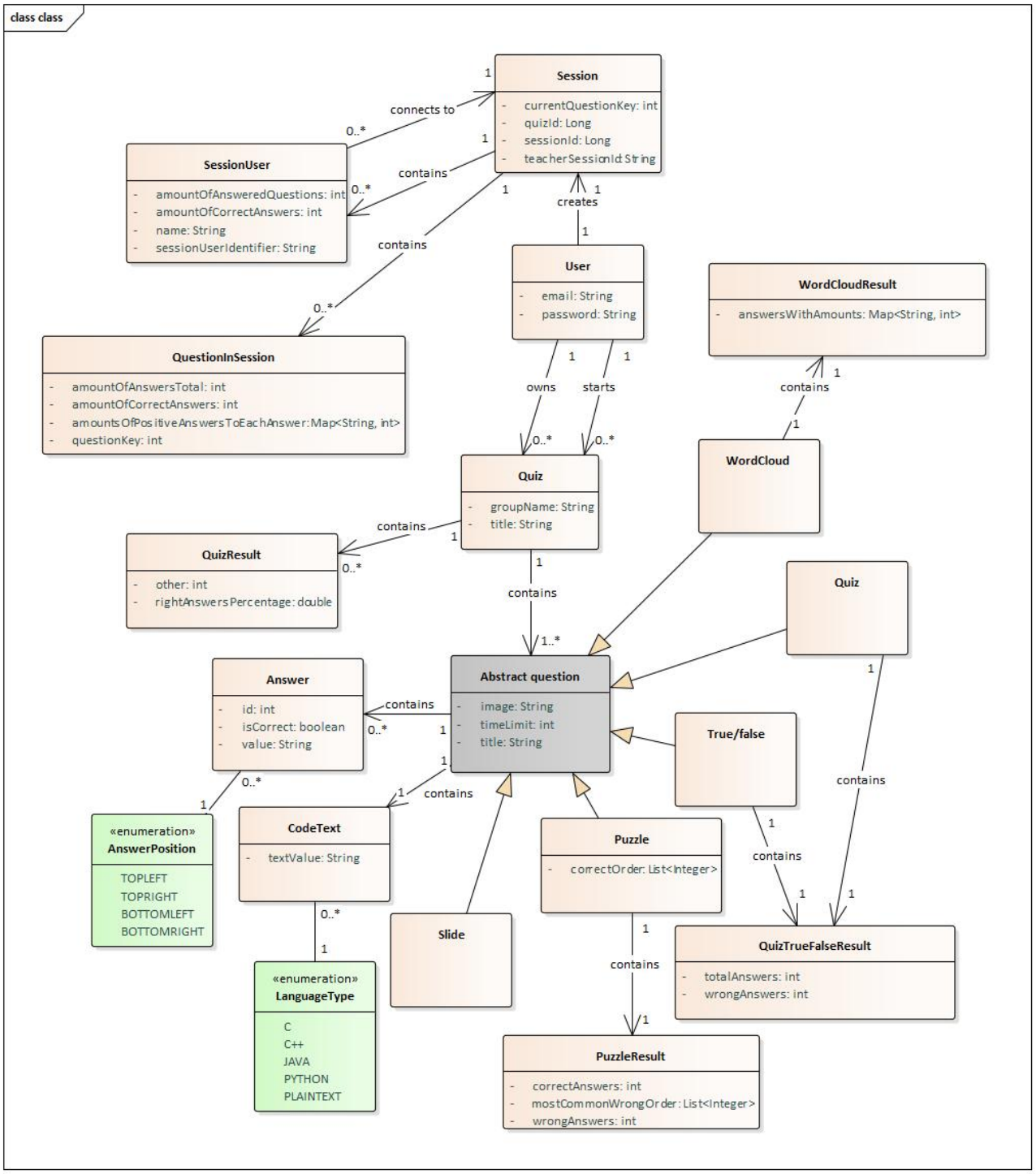
3. **Vrstevnatá architektura** Vrstevnatá architektura je velmi podobná N-tier architektuře. Jedná se o jednu z nejběžnějších a nejvíce rozšířených architektur v softwarovém vývoji. Je složena z několika horizontálně oddělených vrstev které spolu fungují jako jeden prvek.[18] Narozdíl od N-tier architektury jsou tedy vrstvy rozděleny pouze logicky, ale ne fyzicky. Jednotlivé vrstvy tedy nemusí běžet na různých zařízeních nebo virtualizacích, mohou běžet na jednom serveru. Vrstvy tedy dohromady tvoří jeden proces.
   Vrstevnatá architektura je jednoduchá a má pouze nízkou cenu na realizaci[19]. Je vyučována v několika předmětech na ČVUT. Narozdíl od N-tier architektury také není k realizaci této architektury třeba tolik zdrojů, všechny vrstvy mohou běžet pouze na jednom serveru. Vzhledem k tomu, že jedním z hlavních účelů této architektury je realizování CRUD operací, čímž se zabývá i vyvíjená aplikace, pro implementaci byla zvolena právě tato architektura.

Implementace tedy bude realizována ve formě vícevrstvé architektury. Každá vrstva může komunikovat pouze se svou úrovní nebo s úrovněmi pod ní. Tato architektura využívá několik klíčových konceptů při tvorbě aplikací:

- **Separation of Concerns (SoC) - Oddělení zodpovědností**
  Rozdělení počítačového programu na různé části tak, aby se z hlediska funkcionality tyto části co možná nejméně překrývaly. To znamená, aby určitou funkcionalitu vykonávala pouze část programu k tomu určená. Další část programu by pak neměla kopírovat v sobě stejnou funkcionalitu. Z toho plyne několik výhod, které nám SoC v aplikaci přináší:
  - Zamezuje duplikacím kódu - zajišťuje singularitu
  - Zajišťuje stabilnost a udržovatelnost aplikace - změna v jedné vrstvě neovlivní jiné vrstvy
- **Encapsulation - Zapouzdření**
  Zapouzdření je jeden ze základních konceptů objektově orientovaného programování. Popisuje myšlenku zabalit data a metody, které s těmito daty operují do jednoho celku. [20] Metody zapouzdření jsou dostupné skrze třídy ve většině objektově orientovaných programovacích jazyků. Zapouzdření rovněž umožňuje ukrytí atributů a metod v objektu pomocí stavby nepropustné zdi, která brání kód proti nechtěným změnám. Jednotlivé vrstvy si tedy mezi sebou předávájí pouze data a metody, které nezbytně potřebují.
- **Low Coupling High Cohesion**
  Jednotlivé vsrtvy aplikace by spolu měly být propojeny jen tak moc, jak je to nezbytně nutné a zároveň části konkrétních vrstev by měly být propojené co nejvíce [21]

Konkrétně půjde v pořadí o tyto vrstvy:

- Prezentační vrstva - Prezentuje obsah uživateli a odesílá jeho požadavky aplikační vrstvě.
- Aplikační vrstva - Na této vrstvě aplikace poslouchá a přijímá požadavky klientské části skrz technologie REST + WebSocket API. Tyto požadavky pouze přijímá, může zkontrolovat správnost požadavku a dále je předává do servisní vrstvy, kde jsou požadavky zpracovány.
- Servisní vrstva - Obsahuje většinu logiky aplikace. Od aplikační vrstvy přijímá data ke zpracování. Po zpracování předává data persistentní vrstvě, pokud je třeba komunikace s databází, nebo zpět aplikační vrstvě, která data odesílá zpět na klienta.
- Modelová vrstva - Reprezentuje třídy, pomocí kterých vytváříme objekty a vztahy mezi nimi. Struktura této vrstvy je znázorněna na obrázku 4.3, který obsahuje diagram tříd.
- Persistentní vrstva - Řídí přístup k databázi.




**Obrázek 4.3:** Diagram tříd

# Programové nástroje v serverové části

### 4.1.1 Java

Java je lehce přenositelná - zkompilovaný Java kód známý jako bytecode běží na většině operačních systémů – konkrétně například na Linuxu, Windows nebo Mac OS. V projektu bude využita Java verze 11 [22] Pro využití Javy jsme se rozhodli zejména kvůli:

- Přenositelnosti
- Bezpečnosti
- Škálovatelnosti
- Stabilnosti

### 4.1.2 PostgreSQL

Pro ukládání dat bylo zvoleno PostgreSQL verze 12. Jedná se o svobodný a open source objektově-relační databázový systém. Na jeho vývoji se podílí globální komunita vývojářů a firem, čímž je zaručena jeho kvalita.
Pro PostgreSQL jsme se rozhodli kvůli těmto bodům:

- Objektově orientované vlastnosti - Navíc od standardní databáze je PostgreSQL obohaceno o objektově orientované vlastnosti, což nám umožňuje definování komplexních datových typů nebo využívání dědičnosti. PostgreSQL se díky tomuto bodu nabízí použít s Javou, která je objektově orientovaný jazyk
- Dodržování standardu ACID: Atomicity, Consistency, Isolation, Durability. Tento standard zaručuje, že datové transakce proběhnou vždy spolehlivě[23]
- Je vyvíjen skrz open-source, což znamená, že PostgreSQL je vyvíjeno bez korporátního dozoru a jeho četná komunita se stará o jeho kvalitu

### 4.1.3 Spring

Pro Spring jsme se rozhodli zejména kvůli ulehčení práce s objekty v projektu. Spring značně pomáhá v redukci takzvaného boiler plate kódu, který je považován za zbytečný. Využívá Dependency Injection, který je jednou z implementací návrhového vzoru Inversion of Control

- **Inversion of Control** předává kontrolu nad objekty nebo nad částmi kódu frameworku nebo kontejneru. Mezi výhody tohoto vzoru patří oddělení exekuce od implementace, ulehčení změny mezi více implementecami nebo větší modularita programu[24].

Dependency Injection realizuje Inversion of Control pomocí injektování závislostí do jednotlivých objektů. Pro zadefinování injekce stačí pouze přidat anotaci například v konstruktoru a při spuštění programu se pak Spring sám postará o injektování a vývojáři se nemusí starat o inicializaci objektů a jejich parametrů.

### 4.1.4 Spring Data

Pro zjednodušení ukládání do databáze bude také využita technologie Spring Data JPA. DAO[1] vrstva většinou obsahuje hodně nadbytečného kódu. Jeden z hlavních účelů Spring Data je eliminace tohoto nadbytečného kódu, což značně zjednodušuje napsaný kód. V tomto zjednodušení spočívá několik výhod, jako například snížení počtu artefaktů, které musíme v projektech definovat a udržovat nebo konzistence konfigurace a objektů, které zprostředkovávají přístup k datům. Spring Data toto zjednodušení posouvá ještě o krok dál a umožňuje kompletní odstranění DAO vrstvy [26]. V projektu bude využit primárně interface JpaRepository, který automaticky umí převést objekt v projektu do JPA[2] formátu a ukládat do databáze. Umí také načíst entity z databáze a převést je zpátky do objektového formátu. Vývojář tedy nemusí v kódu definovat přechod mezi objekty v modelové vrstvě a datovými objekty, toto chování je zadefinováno v již zmíněném interface.

### 4.1.5 WebSocket

Komunikační protokol pro klient-server aplikace. Je součástí specifikace Java EE7. Jedná se o obousměrnou, plně duplexní a persistentní komunikaci mezi serverem a webovým prohlížečem (klientem). Pokud je navázáno WebSocket spojení, zůstává otevřeno, dokud se ho server nebo klient nerozhodne uzavřít [25]. WebSockets mají také nízkou odezvu zprostředkovanou skrz protokol TCP. Tato technologie bude využita pro zprostředkování komunikace zejména při spuštěném kvízu. Příklad funkcionality je následující: Při spuštěném kvízu vyučující spustí první otázku, tato informace se pomocí WebSocket odešle na server, který zprávu pošle všem zaregistrovaným klientským zařízením v tomto spuštěném kvízu. Klienti tuto zprávu zpracují a studentům se následně zobrazí na jejich zařízeních otázka, na kterou mohou odpovídat.

### 4.1.6 REST API

Server bude poskytovat REST API zprostředkující komunikaci mezi serverem a klientem tam, kde není potřeba obousměrná komunikace. Jedná se o situace, kdy komunikaci vždy začíná klient a server pouze zpracovává jeho požadavky. Tato technologie bude využita například pro registraci a přihlášení vyučujícího do aplikace.

[1] Data access objects - objekty zprostředkující přechody mezi databází a objekty v projektu

[2] Java Persistence API - umožňuje objektově relační mapování objektů

Při zvažování, které API pro jednosměrnou komunikaci využít, připadala v úvahu také technologie SOAP. REST má oproti této technologii několik výhod, mezi které patří:[27]

- Jednoduchost tvorby a adaptibilita
- Menší požadavky na zdroje
- Není nutnost používat různé porty na různé typy notifikací

Mezi další důvody zvolení REST API patří také tyto faktory:

- Škálovatelnost
- Flexibilita
- Nezávislost

## 4.2 Klientská část

### 4.2.1 React

React je JavaScriptová knihovna pro tvorbu uživatelského rozhraní. Je vyvíjený Facebookem a komunitou samostatných vývojářů a společností. React může být využit jako základ pro tvorbu single-page nebo mobilních aplikací, protože je optimální pro práci s rychle se měnícími daty. Kvízová aplikace, která bude implementována v rámci této práce, bude obsahovat velké množství rychle měnících se dat, například při tvoření kvízu, nebo při spouštění kvízu před studenty. React je dle průzkumu StackOverflow za rok 2021[3] také nejvyužívanější webový framework za rok 2021.
Dalšími důvody, proč zvolit pro tvoření klientské části React, jsou:

- Flexibilita - kód v Reactu je lehce spravovatelný a flexibilní díky jeho modulární struktuře
- Výkon - Aplikace napsané v Reactu jsou velmi responzivní
- Přepoužitelné komponenty - Pro stejné nebo podobné funkcionality je možné přepoužít již napsané komponenty. Není tedy nutné psát podobný kód několikrát

### 4.2.2 TypeScript

TypeScript je nadstavba nad jazykem JavaScript, který se používá i pro tvorbu webových aplikací v Reactu. TypeScript přidává k JavaScriptu syntax navíc, aby podpořil silnější vazbu s kódovým editorem [28]. Díky tomu je možné zachytit chyby již v editoru a ne až při spuštění aplikace. Typescript

[3]Průzkum StackOverflow za rok 2021:
https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-web-frameworks

je typovaný superset jazyka JavaScript[4], což znamená, že přidává pravidla, jak mohou být různé typy hodnot použity. TypeScript například nepřipouští připsání proměnné hodnotu typu Integer, pokud předtím měla hodnotu typu String. Typový revizor jazyka TypeScript je navržen tak, aby zachytil co nejvíce chyb již při psaní kódu.
Mezi hlavní výhody jazyka TypeScript tedy patří:[29]

- Statická typová kontrola - umožňuje zachytit chyby již v editoru a zabraňuje vzniku bugů
- Přehledná definice API - Pokud chce vývojář umožnit ostatním používat jeho knihovnu napsanou v jazyce TypeScript, musí nejdříve publikovat API, které obsahuje instrukce pro použití knihovny
- Vestavěná podpora jazyka JavaScript

### 4.2.3 Material UI

Material UI poskytuje robustní, upravitelnou a přístupnou knihovnu základních i pokročilých komponent, což umožňuje tvorbu vlastního návrhového systému a rychlejší tvorbu React aplikací.
Material UI je nejpopulárnější React komponentová knihovna dle aktivity na GitHubu. Je jednoduchá, nezatěžující a postavena podle Google Material Design specifikací. Obsahuje komponenty pro správu rozložení, formuláře, zobrazování dat a mnoho dalšího [30].

Je potřebné zmínit, že u velké části vybraných technologií hrála při výběru svou roli předchozí znalost s těmito technologiemi z výuky na ČVUT, nebo z praxe.

[4]To znamená, že jazyk napsaný v jazyce JavaScript je validní v jazyce TypeScript

# Kapitola 5

# Implementace

## 5.1 Rozsah implementace

Vzhledem k velkému rozsahu práce a množství požadovaných funkcionalit aplikace bylo rozhodnuto, že v rámci této práce neproběhne její kompletní implementace. Výstupem ovšem je fungující aplikace, která umožní vytvoření kvízu a jeho zkoušení se studenty. Celkový rozsah implementace je definován následujícími funkčními požadavky (ze sekce 3.3):

**FR 1** Systém umožňuje nepřihlášenému uživateli zaregistrovat se pomocí e-mailové adresy a hesla

**FR 2** Systém umožňuje nepřihlášenému uživateli přihlásit se k existujícímu účtu pomocí e-mailové adresy a hesla

**FR 5** Systém umožňuje učiteli vytvářet kvízy

**FR 5.1** Systém umožňuje učiteli v kvízech vytvářet různé typy otázek podle potřeby. Konkrétně se bude jednat o tyto typy otázek:

- Kvízová

**FR 5.3** Systém umožňuje učiteli do otázek a jednotlivých odpovědí psát programovací kód v jednotlivých programovacích jazycích
. Podporovány budou primárně tyto jazyky:

- Java
- C
- C++
- Python
- Zde je také navíc možnost nepsat otázku v žádném programovacím jazyce, pouze v prostém textu

**FR 6** Systém umožňuje učiteli uložit vytvořený kvíz pro pozdější využití

**FR 7** Systém umožňuje učiteli uložený kvíz smazat

**FR 9** Systém umožňuje učiteli vygenerovat unikátní pin, díky kterému se budou studenti moci připojit do kvízu

**FR 10** Systém umožňuje studentovi zúčastnit se kvízu pomocí unikátního pinu poskytnutého jeho učitelem

**FR 11** Systém umožňuje učiteli opakovaně spustit uložený kvíz

**FR 12** Systém umožňuje učiteli ovládat jím spuštěný kvíz. Ovládáním kvízu se myslí tyto činnosti:

- Spustit otázku
- Zobrazit vyhodnocení otázky včetně programovacího kódu
- Ukončení kvízu

**FR 13** Systém umožňuje studentovi odpovídat na jednotlivé otázky na svých mobilních zařízeních

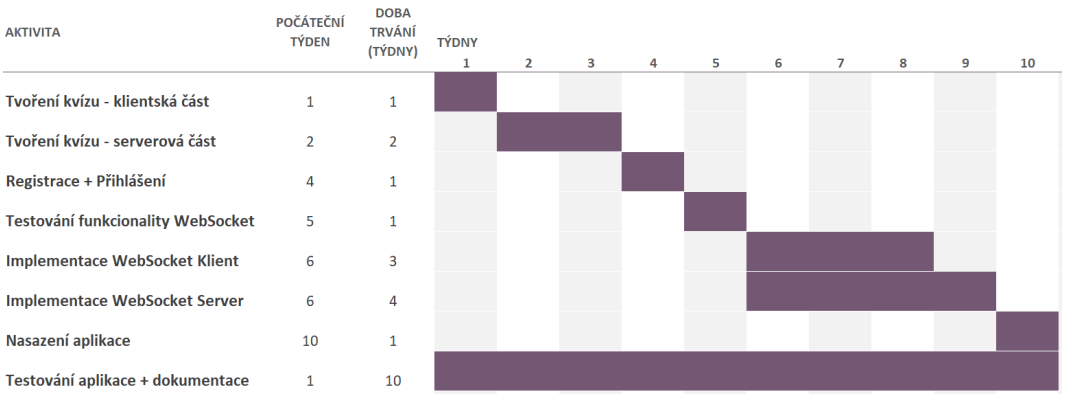
**FR 14** Systém umožňuje učiteli zobrazit statistiky z již spuštěného kvízu

## 5.2 Postup a popis implementace

Vzhledem k tomu, že bylo stanoveno velké množství požadavků a pro splnění těchto požadavků bylo třeba realizovat klientskou a serverovou část aplikace, databázi a nasazení kvůli zpřístupnění webové aplikace z URL, bylo již před začátkem implementace stanoveno, jaký bude její rozsah. Konkrétně se jednalo o umožnění vytvoření kvízu s jedním typem otázek, jeho uložení a následné spuštění tohoto kvízu před studenty, pro což bylo nutné zprovoznit obousměrnou komunikaci mezi serverem a klientem.
Plánování jednotlivých úkolů, které bylo třeba splnit pro realizaci větších částí, probíhalo v programu Jira Software[1] od společnosti Atlassian. Byla využita její bezplatná verze. Jednotlivé úkoly byly rozděleny do několika kategorií:

- **Epic** - Velká část implementace, kterou je třeba rozdělit do několika menších úkolů typu Story
- **Story** - Větší úkol, který je součástí "Epiců". Může být rozdělen do několika menších úkolů typu Task.
- **Task** - Menší typ úkolu, typicky se dál nedělí a jeho realizace by neměla přesahovat dobu několika dní.
- **Bug** - Chyba v programu

Postup implementace se řídil Ganttovým diagramem z obrázku 5.1:




**Obrázek 5.1:** Ganttův diagram

Celá implementace byla verzována v systému Git, konkrétně ve školním GitLabu na adrese `https://gitlab.fel.cvut.cz/valaluk2/thesisquizapp`. Každý úkol stanovený v programu Jira byl implementován ve vedlejší větvi a po dokončení úkolu byla tato větev sloučena s hlavní větví master.

[1] https://www.atlassian.com/software/jira

Dále se budeme zabývat stěžejními částmi implementace a postupem jejich realizace.

### 5.2.1 Tvoření kvízu

Implementace aplikace začala realizací stránky, kde může uživatel vytvořit nový kvíz, nebo později upravovat již vytvořený kvíz. Prvním krokem bylo vytvoření klientské části této stránky, tedy té části, kterou vidí uživatel. Stránka byla rozdělena na 2 části, levý boční panel, kde uživatel vidí seznam otázek a na zbytku stránky může upravovat aktuálně zvolenou otázku. Po implementaci klientské části této stránky mohl uživatel kvíz vytvořit se všemi daty, které tvoření kvízu umožňuje, kvůli absenci serveru ale tato data ještě nebyla persistentní - nedala se uložit.
Proto začala implementace serverové části. Dle návrhu byla realizována modelová vrstva, kde byly vytvořeny třídy pro reprezentaci kvízu. Všechny třídy v modelové vrstvě jsou tvořeny v JPA formátu, což znamená, že jsou připraveny na uložení do databáze. Tyto třídy však plně nekorespondovaly s daty odesílanými z klientské části, proto byly vytvořeny některé DTO třídy pro reprezentaci objektů tak, jak přichází od klientské části. Pro převod z DTO formátu do modelového formátu byly také vytvořeny pomocné mapovací třídy, které přetváří data od klientské části (tedy data v DTO formátu) na data v modelovém formátu a obráceně. Následně byl vytvořen REST koncový bod v aplikační vrstvě definující URL cestu, na které je pro klienty dostupný. Aplikační vrstva však pouze požadavky od klienta přijímá, ke zpracování těchto požadavků byla vytvořena business vrstva. Ta v sobě obsahuje kontrolu požadavků od klienta, již vytvořené mapovací třídy, pomocí kterých přetváří data od klienta na data v modelovém tvaru, a také již zmíněný interface JpaRepository, který ukládá objekty v modelovém tvaru do databáze.
Po spuštění serverové části se pomocí JPA podle modelové vrstvy automaticky tvoří tabulky v databázi, některé z nich však bylo pro zajištění správné funkcionality nutné ručně upravit.

### 5.2.2 Registrace a přihlášení

V tuto chvíli tedy uživatel mohl vytvořit a uložit kvíz. Tento kvíz ale pak byl přístupný globálně všem uživatelům, kteří si otevřeli domovskou stránku. Proto byla vytvořena registrace, při které si může uživatel vytvořit nový účet. Pokud poté vytvoří kvíz, uživatel je označen jako jeho vlastník a přísup k němu má pouze on.
Realizace systému registrace a přihlášení je však realizována velmi jednoduše. Hlavním cílem aplikace bylo umožnit vytvoření a zkoušení kvízu a v době implementace ještě nebylo jasné, jak komplexní problém bude vytvořit zkoušení kvízu se studenty. Registrace a přihlášení jsou tedy vytvořené pouze proto, aby nemohl někdo cizí uživateli upravovat nebo mazat jeho kvíz. Přihlášení však plně persistentní, to například znamená, že pokud uživatel aktualizuje po přihlášení stránku, je odhlášen a musí se přihlásit znovu. Tato část je jedna z těch, které bude v budoucnu třeba upravit a plně dokončit.

### 5.2.3 Zkoušení kvízu

Tato část začala analýzou technologie WebSocket, protože bylo třeba zjistit, zda tato technologie umožní realizaci komunikace mezi vyučujícím a studenty při spuštěném kvízu. Byl proto vytvořen testovací projekt, kde proběhlo testování této technologie.
Vytvoření komunikace mezi více připojenými uživateli v testovacím projektu proběhlo rychle a bez problémů, větší problém bylo zjistit, jak posílat v rámci jednoho spojení zprávy jen konkrétním uživatelům. Díky několika návodům na internetu [2] se toto však povedlo realizovat, což znamenalo, že technologie WebSockets se dá využít i do vyvíjené aplikace.

V samotné aplikaci poté tato implementace začala vytvořením relace (session). Pokud tedy vyučující klikne u vytvořeného kvízu na jeho spuštění, je odeslán požadavek na server, kde se relace vytvoří, uloží do databáze a klientovi se zpět odešle její identifikátor. Ten je poté zobrazen na obrazovce vyučujícího. Každý student, který se poté připojí, odesílá požadavek se svým jménem a identifikátorem od vyučujícího. Po úspěšné kontrole údajů je student přidán do seznamu připojených uživatelů v kvízu a vyučujícímu je odeslána zpráva o připojení nového studenta. Klient u vyučujícího tuto zprávu zpracuje a zobrazí nového studenta na obrazovce.
Po začátku kvízu s připojenými studenty začíná cyklus, který se opakuje až do poslední otázky:

1. Vyučující kliknutím na tlačítko začít kvíz / další otázka odesílá požadavek na posunutí aktuální otázky v kvízu. Tato informace se následně odesílá všem studentům ze seznamu v relaci. Klienti u studentů tuto zprávu zpracují a zobrazí odpovědi, které mohou studenti následně vybírat a odesílat svoji volbu.

2. Každá odpověď se odesílá na server, který ji ohodnotí jako správnou/chybnou a ukládá. Také kontroluje, zda již odpověděli na otázku všichni, pokud ano, odesílá tuto informaci vyučujícímu. Alternativně může vyučující kdykoli odeslat požadavek na odeslání aktuálních statistik právě zkoušené otázky, čímž otázka končí. Všem studentům je pak odeslána informace o konci otázky, čímž je jim zamezeno odesílání odpovědí, pokud tak již neučinili a vyučujícímu je statistika odeslána

3. Po tom, co klient u vyučujícího přijme statistiku otázky, je tato statistika zpracována a zobrazena na obrazovce vyučujícího. Zobrazeno je také zadání otázky, aby dle něj mohl vyučující vysvětlovat správné odpovědi.

Po vyhodnocení poslední otázky jsou vyučujícímu dány 2 možnosti pomocí tlačítek - zobrazit výsledky, nebo ukončit kvíz.
Pokud chce zobrazit výsledky, od serveru mu přijde seznam nejlepších studentů. Klient pak tento seznam seřadí podle počtu získaných bodů (správných

[2] `https://www.youtube.com/watch?v=LdQY-OUM2mk&ab_channel=LiliumCode`
`https://www.baeldung.com/websockets-spring`

odpovědí) a zobrazí 3 nejlepší studenty v tabulce. Každému studentovi ze seznamu je také odesláno jeho osobní skóre.
V případě ukončení kvízu je tato informace odeslána všem studentům, čímž je klient přesměruje zpět na formulář k připojení ke kvízu. Ukončit kvíz může vyučující kdykoli v jeho průběhu opuštěním stránky. Během testování tohoto rozhraní se ale několikrát stalo, že uživatel, který aplikaci testoval, stránku opustil omylem, proto bylo navíc implementováno výstražné okno, kde je vyučující otázán, zda chce opravdu opustit stránku.

### 5.2.4 Testování a dokumentace

Testování i dokumentace probíhala v celém průběhu implementace. Funkcionalita serverové části je kompletně pokryta jednotkovými testy a bylo také vytvořeno několik integračních testů pro otestování funkcionality více komponent najednou. Každá funkcionalita serverové i klientské části je popsána komentáři, server standardní formou JavaDoc, klient formou klasických komentářů.

# Kapitola 6

# Testování funkcionality aplikace

## 6.1 Testování uživatelského rozhraní

Součástí této práce byla i tvorba uživatelského rozhraní aplikace. Bylo tedy nutné ověřit, že uživatelské rozhraní je přehledné a intuitivní. Za tímto účelem byl vytvořen seznam kroků, které by uživatelé měli být v aplikaci sami schopni provést.
Pro toto testování bylo osloveno 5 respondentů technického i netechnického založení v rozmezí 20 - 50 let pro zajištění objektivity testování.
Testování probíhalo z pohledu vyučujícího a studenta. U vyučujícího se jedná o následující části:

1. Registrace + Přihlášení
2. Tvoření kvízu
3. Spuštění kvízu se studenty

U studenta je pouze jedna část a to připojení ke kvízu a odpovídání na otázky. Tyto části se navíc skládají z jednotlivých kroků, které má uživatel v aplikaci provést. Tyto instrukce mu byly podány a následně bylo pouze sledováno, jestli se uživatel v aplikaci správně orientuje a ví, jak daný úkon vykonat. Všechny kroky a reakce uživatelů na instrukce jsou detailně zapsány v jedné z příloh.

### 6.1.1 Celková zpětná vazba

Po průchodu aplikací bylo každému respondentovi položeno několik otázek, které reflektují jeho celkový dojem z aplikace:

- Jak jednoduché bylo používat a zorientovat se v uživatelském rozhraní? [1 - 10]
  - 7,5 (1x)
  - 8,5 (1x)
  - 9 (2x)

  - 10 (1x)

- Jak jste byli spokojeni s komunikací programu (výstrahy, oznámení a chybové hlášky) s uživatelem? [1 - 10]
  - 9 (2x)
  - 8 (1x)
  - 10 (2x)
- Našli jste v programu nějaké neošetřené chyby?
  - Ano
  - Ne 5x
- Rozuměli jste všem chybovým hláškám / výstrahám?
  - Ano, bez problémů (5x)
  - Ano, s menšími problémy
  - Ne
- Jak byste ohodnotili vzhled a vizuální stránku aplikace? [1 - 10]
  - 7 (2x)
  - 8 (1x)
  - 9 (1x)
  - 10 (1x)
- Stalo se někdy, že jste nevěděli, jak provést některý z úkonů?
  - Ano 2x - v obou případech respondenti nevěděli pouze, jak se připojit ke kvízu z přihlášeného účtu, momentálně je totiž třeba, aby se nejprve odhlásili. Po přesměrování na domovskou stránku bylo již vše jasné. Lze tedy předpokládat, že pokud by se chtěli připojit ke kvízu z pozice studenta, který právě otevřel aplikaci, nebyl by s tím žádný problém.
  - Ne 3x

### 6.1.2 Shrnutí

Orientace v aplikaci tedy kromě jednoho kroku nedělala respondentům žádný problém a nikdo nenašel v aplikaci neošetřené chyby. Také vzhled byl hodnocen velmi pozitivně. Několikrát byla oceněna jednoduchost aplikace a fakt, že uživatelé nejsou zahlcování zbytečnými informacemi.
Opakovalo se však také několik negativních komentářů. Aplikace má dle respondentů v některých případech problémy s navigací. Bylo doporučeno přidat více navigačních tlačítek, aby se eliminovala potřeba pohybovat se mezi stránkami pomocí historie v prohlížeči. Respondentům také nebylo vždy jasné, že některé elementy na stránkách obsahují vysvětlivky nebo doplňující

informace, které se zobrazují po najetí na dané elementy myší. Pro zamezení tohoto problému bylo doporučeno přidat do aplikace více ikon, které mohou v uživatelích podněcovat chtěnou akci. Tyto problémy a doporučení budou v budoucnu zanalyzovány a poté adresovány.
Celkově tedy testování uživatelského rozhraní skončilo úspěšně, zejména kvůli orientaci respondentů v aplikaci. Pouze dva z nich si nevěděli rady s jednou instrukcí, se zbylými sedmnácti kroky si úspěšně poradil každý z nich. Navíc bylo ověřeno, že aplikace funguje správně na všech stanovených webových prohlížečích a navíc i na prohlížeči Safari od společnosti Apple [1].

## 6.2 Zkoušky aplikace při výuce programování

Po implementaci bylo také třeba ověřit, zda aplikace plní stanovený účel a zda zvládá i více uživatelů najednou. Proto proběhla zkouška aplikace při čtyřech cvičeních zaměřených na výuku programování.
Aplikace byla vyzkoušena při výuce na ČVUT FEL na čtyřech cvičeních z programovacího jazyka C. Vyučujícím, který kvíz spouštěl v roli vyučujícího, byla vedoucí této práce RNDr. Ingrid Nagyová, Ph.D. při výuce jednoho z jejích předmětů. Jí vytvořený kvíz obsahoval 7 otázek a jeho téma byly vlákna v jazyce C. Spuštění kvízu proběhlo ve webovém prohlížeči Mozilla Firefox. U studentů byly zaznamenány prohlížeče Safari, Microsoft Edge, Google Chrome a Mozilla Firefox.

### 6.2.1 První zkouška

Prvního cvičení, na kterém byla implementovaná aplikace vyzkoušena, se zúčastnilo 13 studentů. Kvíz byl spuštěn na začátku cvičení a studentům byl poskytnut vygenerovaný kód k připojení se k aplikaci. Po přibližně dvou minutách se ke kvízu připojilo 10 studentů, nikdo z nepřipojených studentů však nevznášel žádné námitky o tom, že by jim připojení ke kvízu nefungovalo, proto lze předpokládat, že tito studenti pouze neměli o opakovací kvíz zájem.
Po spuštění kvízu začali studenti odpovídat na jednotlivé otázky. U několika otázek je bylo od vyučujícího třeba ručně ukončit, protože neodpověděli všichni studenti, důvod nezodpovězení je neznámý.
U vyhodnocení kvízu byla využita většina času celého zkoušení kvízu. Bylo položeno několik dotazů od studentů k danné otázce, vyučující se těmto dotazům i s pomocí programovacího kódu ve vyhodnocení otázky věnoval. Na některé odpovědi nezodpověděl správně žádný student, zejména těmto otázkám se pak vyučující věnoval z vlastní iniciativy více dopodrobna. Programovací kód otázky byl vyučujícím z otázky vícekrát zkopírován a exekuován ve vedlejším editoru, na čemž byly vysvětleny správné odpovědi na otázku. V některých případech vyučující takto vysvětloval daný programovací kód i více než 5 minut, když se pak vrátil zpět ke kvízu, jeho kvízové spojení se studenty stále fungovalo v pořádku.

[1] `https://www.apple.com/safari/`

Na konci kvízu byla zobrazena tabulka nejlepších studentů, kde se nacházeli 3 studenti se 3 body.
Celková doba zkoušení kvízu byla více než 30 minut. Během této doby se nevyskytly žádné chyby nebo problémy a aplikace po celou dobu fungovala správně. Bylo tedy ověřeno, že aplikace skutečně podporuje formativní hodnocení ve výuce programování, konkrétně:

- **Komunikaci mezi vyučujícím a studenty**
  Studenti se ptali na doplňující dotazy k otázkám a vyučující otázky vysvětloval
- **Předávání zpětné vazby studentům a vyučujícím**
  Studenti dostali zpětnou vazbu o jejich znalostech. Díky kvízu měli možnost uvědomit si, čemu rozumí a do jaké míry se v probírané problematice orientují. Vyučující také dostal zpětnou vazbu o znalostech jeho studentů a na co se má ve výuce více zaměřit.

Vyučující navíc silně ocenil i možnost kopírovat programovací kód přímo z otázky a poté ho přenést do vedlejšího editoru. Tato funkcionalita nebyla předem stanovena, i přesto je ale nyní velmi přínosná.
Vyšlo ovšem také najevo pár věcí, které je možné v průběhu kvízu zlepšit:

- Fialová barva na projektoru splývá s textem, proto bylo pro studenty v případě potřeby téměř nemožné přečíst odpověď s fialovým pozadím z projektoru
- U některých studentů byl zřejmý problém přečíst vygenerovaný kód pro připojení ke kvízu, písmo je příliš malé.
- Vyučující se po kvízu zmínil, že by při čekání na odpovědi od studentů ocenil mít informaci o tom, kolik studentů již zodpovědělo na otázku. Takto může samozřejmě čekání kdykoli ukončit, ale neví, kolika studentům tím zamezí na otázku zodpovídat

Všechny tyto body byly adresovány a jejich oprava již byla do aplikace implementována. Vyučující tedy nyní při čekání na odpovědi od studentů ví, kolik studentů již na otázku odpovědělo a byla vylepšena viditelnost některých částí aplikace pro studenty.
Kvíz byl otestován se studenty ještě na třech dalších cvičeních. Jejich průběh byl velmi podobný jako první zkouška s výjimkou počtu studentů, proto zde nejsou takto podrobně sepsány. Důležité je ale zdůraznit, že kvíz na všech cvičeních proběhl úspěšně bez chyb a problémů i přes to, že zkoušení několikrát přesáhlo dobu 30 minut a u některých otázek bylo zkoušení přerušeno na několik minut. Aplikace tedy plní účel stanovený dříve v dokumentu a skutečně podporuje formativní hodnocení ve výuce programování.

# Kapitola 7

# Závěr

Cílem této práce bylo zanalyzovat, navrhnout, implementovat a otestovat online aplikaci podporující formativní vzdělávání ve výuce programování.
Nejprve byla realizována analýza problému zabývající se primárně formativním hodnocením ve výuce studentů ve školách a výhodami, které tento typ hodnocení může do výuky přinést. Také proběhla analýza požadavků, kde bylo definováno, co a proč by měla aplikace umožňovat jejím uživatelům.
Poté byla navržena softwarová architektura aplikace a byly vybrány programové nástroje umožňující její implementaci. Po srovnání několika možností byla pro implementaci zvolena vícevrstvá architektura. Následovala implementace celé aplikace od úplného počátku, která se skládala z tvorby klientské i serverové části, databáze a testování. Aplikace byla také nasazena pro umožnění testování její funkcionality s více uživateli.
Implementovaná aplikace byla otestována na čtyřech cvičeních při výuce programování na ČVUT FEL. Vyučující nejdříve v aplikaci vytvořil nový kvíz týkající se právě probírané tématiky a poté jej spustil se svými studenty, kteří se k němu pomocí svých zařízení připojili. Kvíz proběhl na všech cvičeních i přes dlouhé trvání a dlouhé přestávky mezi jednotlivými otázkami bez chyb a problémů. Bylo ověřeno, že vyvinutá aplikace skutečně podporuje formativní hodnocení, zejména pak komunikaci mezi vyučujícím a studenty a předávání zpětné vazby studentům i vyučujícímu o znalostech studentů. Vyučující navíc ocenil možnost kopírovat programovací kód přímo z vyhodnocení otázky a jeho přenos do vedlejšího editoru, což využil pro podrobnější vysvětlení problematiky v případě dotazů od studentů.
Proběhlo také testování uživatelského rozhraní, bylo verifikováno, že uživatelé technického i netechnického typu jsou se v aplikaci schopni bez pomoci zaregistrovat, vytvořit kvíz a poté ho spouštět s jinými uživateli.

## 7.1 Budoucnost projektu

Další práce na projektu se bude skládat zejména z rozšiřování funkcionality aplikace dle definovaných funkčních požadavků, které ještě nebyly realizovány vzhledem k velkému rozsahu implementace aplikace. Velmi důležitou částí je přidání dalších typů otázek, které mohou výuku obohatit a udělat jí

zajímavější. V případě zájmu od vyučujících o používání této aplikace v jejich výuce bude aplikace nasazena na výkonější server pro zajištění stabilnosti a lepší rychlosti vyřizování požadavků od aplikace. V případě rozšíření aplikace do výuky na ČVUT také bude možné od vyučujících i studentů získat zpětnou vazbu, na základě které se bude aplikace dále rozšiřovat a vylepšovat.

# Literatura


[1] KRABSOVÁ, Veronika a Kateřina, NOVOTNÁ. *Formativní hodnocení: Případová studie.* 2013, s. 355 - 371.

[2] PRŮCHA, Jan, Eliška WALTEROVÁ a Jiří MAREŠ. *Pedagogický slovník.* 4. aktualizované vydání. Praha: Portál, 2009, s. 322. ISBN 978-80-7367-416-8.

[3] SLAVÍK, Jan. *Hodnocení v současné škole - východiska a nové metody pro praxi.* Praha: Portál, 1999, s. 33-41. ISBN 80-7178-262-9.

[4] GREENSTEIN, Laura. *What Teachers Really need to know about Formative Assessment.* ASCD, 2010. ISBN 978-1-4166-0996-4.

[5] PRŮCHA, Jan. *Pedagogická encyklopedie.* Praha: Portál, 2009. ISBN 978-80-7367-546-2.

[6] Web Inkluzivní škola, *Formativní hodnocení.* 2022. URL: `https://inkluzivniskola.cz/content/formativni-hodnoceni`

[7] SLAVÍK, Jan. *Hodnocení v současné škole - východiska a nové metody pro praxi.* Praha: Portál, 1999, s. 38. ISBN 80-7178-262-9.

[8] STARÝ, Karel. *Sumativní a formativní hodnocení.* Metodický portál, 2006.

[9] SPIRA, Matthew. *Disadvantages of Summative Assessments.* 2018. URL: `https://www.theclassroom.com/effects-standardized-tests-teachers-students-10379.html`

[10] Web Zapojme Všechny. *Sumativní hodnocení vs. Formativní hodnocení.* 2021. URL: `https://zapojmevsechny.cz/landing-pages/detail/sumativni-hodnoceni-vs-formativni-hodnoceni`

[11] WANG, Alf Inge a Rabail TAHIR. *The effect of using Kahoot! for learning – A literature review.* Creative Commnons, 2020.

[12] ARNEIL, Stewart a Martin, HOLMES. *Hot Potatoes: Free Tools for Creating Interactive Language Exercises for the World Wide Web.* 1998.

[13] Scaled Agile Framework. *Nonfunctional Requirements*. Scaled Agile, 2021. URL: https://www.scaledagileframework.com/nonfunctional-requirements/.

[14] Atlassian. *Atlassian Agile Coach*. 2018.

[15] SYED, Hasan. *Pipe and filter architecture*. 2019. URL: https://syedhasan010.medium.com/pipe-and-filter-architecture-bd7babdb908

[16] Microsoft Azure docs. *Model kanálů a filtrů*. Microsoft, 2022. URL: https://docs.microsoft.com/cs-cz/azure/architecture/patterns/pipes-and-filters

[17] Techopedia. *N-tier architecture*. Janalta Interactive, 2022. URL: https://www.techopedia.com/definition/17185/n-tier-architecture

[18] Baeldung. *Layered Architecture*. Baeldung, 2021. URL: https://www.baeldung.com/cs/layered-architecture

[19] RICHARDS, Mark a Neal FORD. *Fundamentals of Software Architecture: An Engineering Approach*. O'Reilly, 2020. ISBN 978-1492043454.

[20] JANNSEN, Thorben. *OOP Concept for Beginners: What is Encapsulation*. Stackify, 2022. URL: https://stackify.com/oop-concept-for-beginners-what-is-encapsulation/

[21] MONTIEL, Ivan. *Low Coupling, High Cohesion*. Medium, 2021. URL: https://medium.com/clarityhub/low-coupling-high-cohesion-3610e35ac4a6

[22] Oracle. *Java documentation*. Oracle, 2022. URL: https://docs.oracle.com/en/java/

[23] WATTS, Stephen. *ACID Explained: Atomic, Consistent, Isolated and Durable*. BMC, 2020. URL: https://www.bmc.com/blogs/acid-atomic-consistent-isolated-durable/

[24] CRUSOVEANU, Loredana. *Intro to Inversion of Control and Dependency Injection with Spring*. Baeldung, 2022. URL: https://www.baeldung.com/inversion-control-and-dependency-injection-in-spring

[25] Baeldung. *Intro to WebSockets with Spring*. Baeldung, 2021. URL: https://www.baeldung.com/websockets-spring

[26] PARASCHIV, Eugen. *Introduction to Spring Data JPA*. Baeldung, 2021. URL: https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa

[27] Web Chakray. *What are the advantages of Rest API*. Chakray, 2022. URL: https://www.chakray.com/advantages-of-rest-api/

[28] TypeScript. *What is Typescript?*. Microsoft, 2022. URL: `https://www.typescriptlang.org/`

[29] FENG, Raphael. *The Benefits of Migrating from JavaScript to TypeScript*. AppDynamics, 2015. URL: `https://www.appdynamics.com/blog/engineering/the-benefits-of-migrating-from-javascript-to-typescript/`

[30] DE MOOR, Thomas. *11 best React component libraries in 2021.* X-Team, 2021. URL: `https://x-team.com/blog/best-react-component-libraries/`

# Příloha A

## Seznam použitých zkratek

- API - Application programming interface
- ČVUT - České Vysoké Učení Technické
- FEL - Fakulta elektrotechnická
- JPA - Java Persistence API
- TCP - Transmission Control Protocol
- CRUD - Create, Read, Update, Delete
- SoC - Separation of Concerns
- URL - Uniform resource locator
- DTO - Data transfer object
- DAO - Data access object

# Příloha B

# Manuál k aplikaci

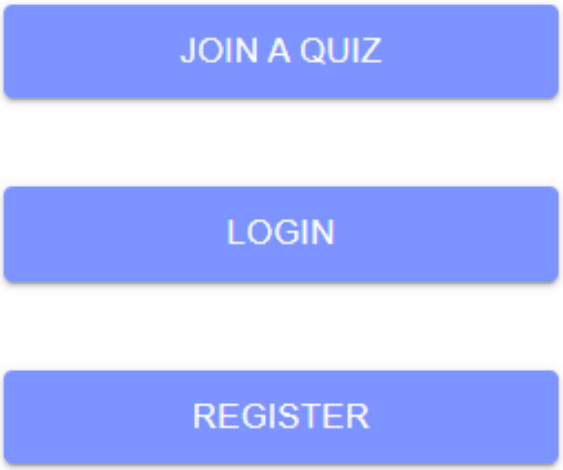
V této příloze se nachází popis jednotlivých stránek a činností, které na nich mohou uživatelé vykonávat.

## B.1 Domovská stránka pro nepřihlášeného uživatele

Toto je první stránka, kterou uživatel vidí po otevření webové aplikace. Výchozí rozhraní této domovské stránky je pro nepřihlášeného uživatele, kterému umožňuje pomocí tlačítek přejít na jiné stránky, které jsou:

- Připojení ke spuštěnému kvízu
- Registrace - vytvoření nového účtu
- Přihlášení k existujícímu účtu

Tato stránka je znázorněna na obrázku B.1:



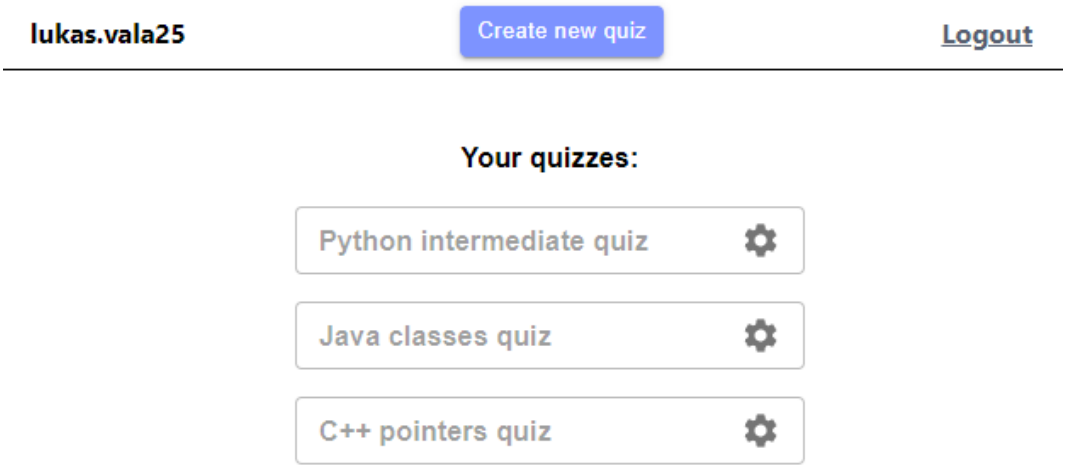

**Obrázek B.1:** Domovská stránka aplikace pro nepřihlášeného uživatele

## B.2 Domovská stránka pro přihlášeného uživatele

Po registraci, která probíhá klasickou formou vyplňování textového formuláře pomocí e-mailu a hesla, nebo přihlášení pomocí existujícího účtu, je uživatel přesměrován na domovskou stránku pro přihlášeného uživatele. Zde má možnost vytvořit nový kvíz, odhlásit se, nebo interagovat s již vytvořenými kvízy. Tato interakce probíhá pomocí kliknutí na název kvízu, po čemž se uživateli zobrazí následující možnosti:

- Upravit kvíz - uživatel je přesměrován na tvoření kvízu, kde jsou již obsaženy všechny otázky z kvízu, který si přeje upravovat
- Smazat kvíz
- Začít zkoušení kvízu se studenty - uživatel je přesměrován na start kvízu v roli organizátora/vyučujícího

Vytvořit nový kvíz může uživatel pomocí tlačítka nacházejícího se v hlavičce stránky. Po kliknutí na toto tlačítko uživatel napíše do textového pole název kvízu a poté je přesměrován na tvoření nového kvízu s jednou prázdnou otázkou, kterou může začít upravovat.
Na následujícím obrázku B.2 je znázorněno, jak vypadá domovská stránka přihlášeného uživatele s e-mailem lukas.vala25@seznam.cz a s třemi vytvořenými kvízy.




**Obrázek B.2:** Domovská stránka aplikace pro přihlášeného uživatele

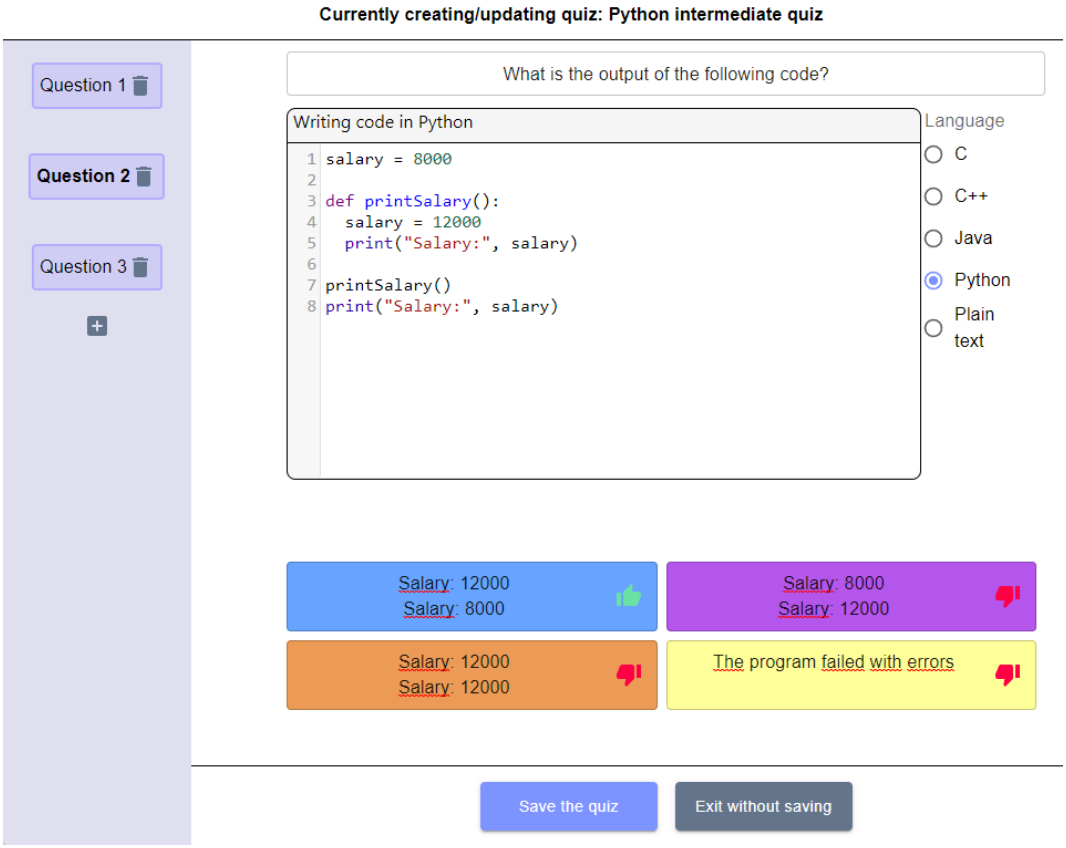
## B.3 Tvoření/Úprava kvízu

Z domovské stránky tedy uživatel může začít tvořit nový kvíz, nebo upravovat existující kvíz, zvolením obou z těchto možností je přesměrován na tuto stránku. Tvoření kvízu je rozděleno na dvě hlavní části:

- Boční panel obsahující již vytvořené otázky. Po kliknutí na jednu z těchto otázek je uživateli tato otázka zobrazena na zbytku stránky a může jí začít dále upravovat. Kliknutím na ikonu koše v jednotlivých otázkách také může uživatel tyto otázky mazat
- Parametry otázky vykrývající zbytek stránky. Zde uživatel upravuje právě zvolenou otázku z bočního panelu. Uživatel zde volí následující parametry otázky:
  - Název otázky
  - Programovací kód otázky, nebo specifikování otázky v prostém textu. Mezi jednotlivými jazyky a prostým textem lze přepínat pomocí klikáním na možnosti vedle programovacího kódu
  - Odpovědi - u každé z těchto odpovědí uživatel volí jejich textový obsah a zda jsou odpovědi správně

  Ve spodní části se nachází tlačítka "Save the quiz", které pošle požadavek na uložení právě tvořeného kvízu na server skrz technologii REST a v případě úspěchu přesměruje uživatele zpět na domovskou stránku a "Exit without saving", které uživatele přesměruje bez uložení upravování

Na následujícím obrázku B.3 je znázorněno upravování druhé otázky ve kvízu, ve které je programovací kód v jazyce Python a jedna otázka ze čtyř jsou správně.




**Obrázek B.3:** Domovská stránka aplikace pro přihlášeného uživatele

## B.4 Zkoušení kvízu

Z domovské stránky může vyučující také začít zkoušení vytvořeného kvízu. Tato část aplikace společně s připojením ke kvízu u studenta je realizována pomocí již zmíněné technologie WebSocket. Pokud chce tedy například vyučující komunikovat se studenty připojenými ve kvízu, pošle požadavek na server, ten požadavek zpracuje a poté ho přeposílá daným studentům.
Ovládání kvízu z pohledu vyučujícího má několik různých rozhraní:

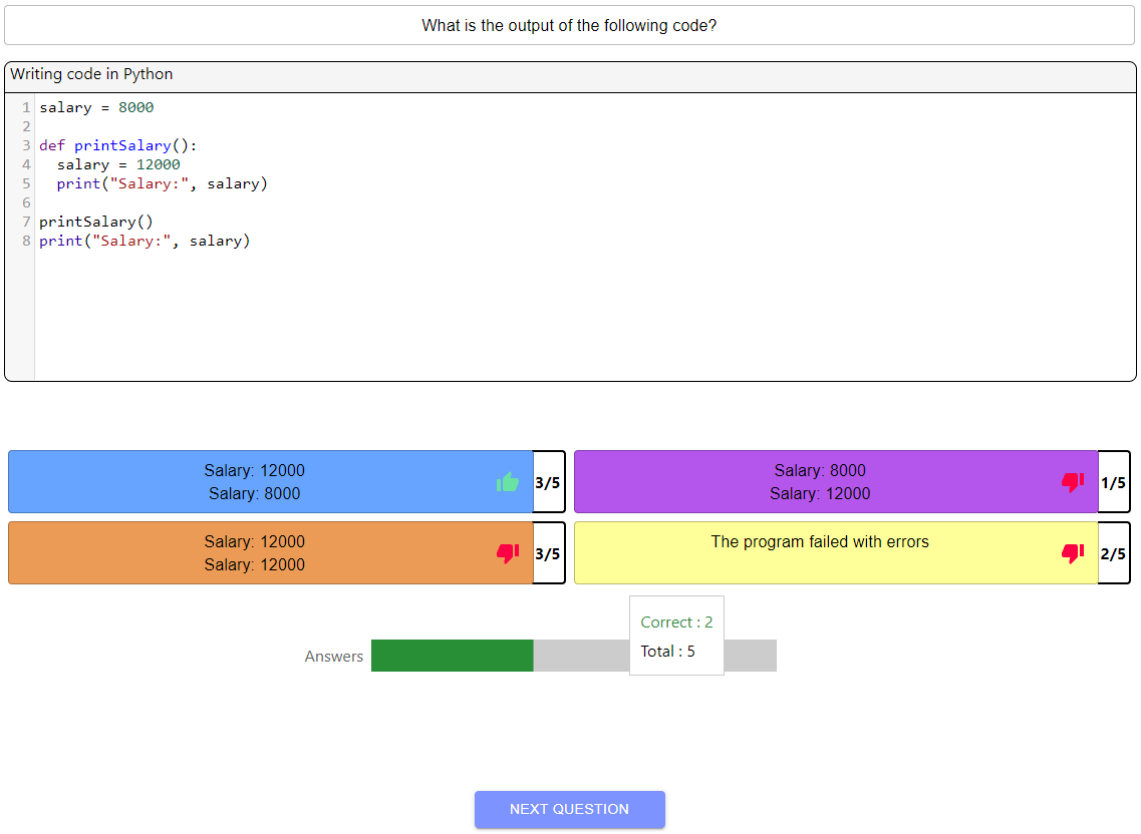
1. Nejdříve je na obrazovce vyučujícího zobrazen unikátní identifikátor právě spuštěného kvízu, který skrze projektor vyučující poskytne svým studentům a studenti se pomocí něj a svého jména/přezdívky poté mohou ke kvízu připojit. Každé studentovo jméno je pak po připojení ke kvízu zobrazeno na projektoru, čímž si mohou studenti ověřit, že se ke kvízu opravdu připojili a vyučující sleduje počet připojených studentů. Vyučující může kvíz spustit s jedním nebo více studenty, bez studentů však tuto možnost nemá.

2. Po spuštění kvízu začíná první otázka. Na obrazovce vyučujícího je nyní velmi podobné rozhraní jako při tvoření kvízu, samozřejmě kromě informace o tom, které odpovědi jsou správné. Studenti v tuto chvíli

mají možnost označit odpovědi jako chybné, nebo správné, a tuto volbu odeslat. Vyučující může během čekání otázku kdykoli ukončit, nebo může čekat, dokud neodpoví všichni studenti. Ve spodní části stránky vyučující také vidí, kolik studentů již na otázku odpovědělo, podle toho se může rozhodnout, zda již může otázku ukončit.

3. Po ukončení otázky se vyučujícími zobrazí na jeho zařízení vyhodnocení otázky. Je opět velmi podobné jako tvoření otázky, má ale několik rozšíření. U každé odpovědi je nyní vidět, kolik studentů ji označilo jako správnou a kolik studentů na otázku odpovídalo celkově. Pod odpověďmi je navíc pomocí jednoduchého grafu viditelné, kolik studentů zvolilo správnou a chybnou kombinaci odpovědí. Vyhodnocení otázky je znázorněno na obrázku B.4. V tomto případě je připojeno ke kvízu 5 studentů a každý z nich na otázku odpověděl. Vidíme, že tři z pěti studentů zvolili správnou odpověď, po najetí myší na graf odpovědí ale vyučující zjistí, že pouze dva studenti zvolili kombinaci zcela správně, jeden ze studentů tedy musel označit jako správnou i jinou chybnou odpověď. Aby se studentovi připsal v kvízu bod, musí mít otázku totiž zcela správně - všechny chybné otázky musí označit jako chybné a správné otázky jako správné.
   Následující průběh kvízu se skládá ze střídání spouštění otázky a následném vyhodnocení. Změna nastává při vyhodnocení poslední otázky v kvízu.

4. Při vyhodnocení poslední otázky má vyučující místo přepnutí na další otázky 2 jiné možnosti - Ukončit kvíz, nebo zobrazit skóre studentů. V případě ukončení kvízu se vyučující vrací zpátky na domovskou stránku a studentům je odeslána zpráva o ukončení kvízu, což způsobí jejich přesměrování na formulář, kde se mohou připojit k novému kvízu. V případě zobrazení skóre studentů se na obrazovce vyučujícího objeví tabulka obsahující tři nejlepší studenty v kvízu s jejich bodovým ohodnocením. Zbylí studenti v tabulce nejsou obsaženi. Všem studentům je ale na jejich zařízení jejich osobní skóre odesláno a zobrazeno.




**Obrázek B.4:** Vyhodnocení otázky, na kterou odpovědělo 5 studentů

# Příloha C

# Struktura aplikace + lokální vývoj

Tato příloha obsahuje návod jak lokálně pracovat na vývoji aplikaci a jak ji lokálně spouštět.

## C.1 Verzování aplikace

Celá implementace aplikace je zaznamenána ve veřejném školním repozitáři verzovacího systému GIT na adrese `https://gitlab.fel.cvut.cz/valaluk2/thesisquizapp`

## C.2 Serverová část

### C.2.1 Struktura

Serverová část se nachází v projektové složce 'api'. Tato složka obsahuje konfigurační soubor 'pom.xml', který obsahuje informace o využívané verzi jazyka Java a jednotlivé závislosti aplikace. Dále je obsahuje jednotlivé složky rozdělené podle funkcionality aplikace. Důležité jsou z hlediska architektury zejména tyto:

- Controller - Reprezentuje aplikační vrstvu. Obsahuje aplikační logiku aplikace, zajišťuje tedy komunikaci s klientskou částí. Tato složka je dále dělena na třídy podle entit, okolo kterých má klient požadavky, konkrétně:
  - UserController - Požadavky na vytvoření, úpravu, mazání a čtení uživatelů - implementován pomocí REST API
  - QuizController - Požadavky na vytvoření, úpravu, mazání a čtení kvízů od uživatelů - implementován pomocí REST API
  - SessionController - Správa spojení mezi studenty a vyučujícími při spouštění kvízu - implementován pomocí WebSocket
- Service - Reprezentuje business vrstvu. Zde se nachází většina logiky aplikace, tato vrstva zejména zpracovává požadavky od aplikační vrstvy.

Strukturálně je rozdělena stejně jako Controller vrstva, třída UserService tedy zpracovává požadavky od třídy UserController

- Model - Reprezentuje modelovou vrstvu. Zde se nachází jednotlivé entity aplikace a vztahy mezi nimi. V této formě jsou entity dále ukládány a získávány z databáze. Po spuštění aplikace je tedy databáze automaticky nakonfigurována dle této vrstvy
- Repository - Reprezentuje persistentní vrstvu. Obsahuje JpaRepository rozhraní, pomocí kterých jsou entity jednoduše ukládány a získávány z databáze

### C.2.2 Lokální vývoj

Serverová část je psána v jazyce Java verze 11. Pro vývoj a správné fungování této části je tedy nutné mít v zařízení nainstalované JDK 11 - Java Development Kit 11. Stáhnout ho lze přímo ze stránek Oracle - `https://www.oracle.com/cz/java/technologies/javase/jdk11-archive-downloads.html`. Po instalaci JDK by aplikaci mělo být možné aplikaci zkompilovat a spustit. Aplikace po spuštění poslouchá na adrese http://localhost:8080, s čímž počítá klientská část. Databáze by se měla pomocí JPA sama nakonfigurovat a inicializovat, není ji proto třeba ručně sestavovat.

## C.3 Klientská část

### C.3.1 Struktura

Klientská část se nachází ve složce 'ui'. Zde je důležitý konfigurační soubor 'package.json', pomocí kterého aplikace před spuštěním stáhne všechny potřebné balíky pro její spuštění. Samotná implementace je poté rozdělena na několik složek rozdělených podle jednotlivých stránek aplikace - ve složce 'pages':

- Login, Registration - Obsahuje podobné formuláře pro přihlášení a registraci, jejich zpracování a odesílání dat na server
- Home - Domovská stránka, kde se nachází seznam kvízů přihlášeného uživatele a možnost začít vytvářet nový kvíz
- Quiz - Stránka pro vytváření a upravování kvízů
- QuizSession - Stránka pro spojení studujících a vyučujících. Tato složka je dělena na pohled vyučujícího a studentů

### C.3.2 Lokální vývoj

Klientská část je implementována v knihovně React programovacího jazyka JavaScript. Pro její spuštění je tedy nutném mít na zařízení fungující verzi npm. - node package manager. Stáhnout lze ze stránky `https://nodejs.org/en/download/`. Po stažení by v příkazové řádce měl být rozeznán příkaz 'npm'. V projektové složce 'ui' je nyní třeba vykonat příkaz 'npm install', který do této složky stáhne všechny potřebné balíky zajišťující správný běh aplikace. Po stáhnutí již stačí pouze vykonat příkaz 'npm start' a aplikace by se měla spustit v prohlížeči na adrese http://localhost:3000. Pokud je nyní spuštěna i serverová část, aplikace by měla fungovat tak, jak má.

# Příloha D

# Externí přílohy

## D.1 Testování uživatelského rozhraní

**test_ui.pdf**: Popis jednotlivých kroků testování uživatelského rozhraní a popis toho, jak se jednotlivým respondentům dařilo kroky plnit.

## D.2 Zdrojový kód

**thesisquizapp.zip**: Kompletní zdrojový kód včetně dokumentace vyvinuté aplikace v rámci této práce