FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání bakalářské práce

Název:	Educhild - dětská část android aplikace
Student:	Tomáš Hanko
Vedoucí:	Ing. Jiří Hunka
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2021/2022

Pokyny pro vypracování

Cílem této práce je softwarový návrh a následná implementace dětského módu android aplikace Educhild, která slouží jako pomůcka pro rodiče k motivaci plnění edukačních úkolů a limitaci jiných aktivit dítěte na mobilním zařízení s OS android.

Postupujte v těchto krocích:
• Analyzujte požadavky potřebné pro cílové uživatele a stávající konkurenční řešení. Spolupracujte s Terezou Langovou, která řeší samotné uživatelské rozhraní a grafickou podobu.
• Na základě analýzy navrhněte vhodné funkcionality dětské části mobilní aplikace (Kvízy a jejich tvorba, odměny, úspěchy, měnu aplikace apod.)
• Proveďte vhodný a kompletní softwarový návrh dětské části Android aplikace.
• Implementujte navrženou část aplikace, neopomeňte řádně testovat.
• Proveďte vlastní hodnocení funkčnosti a použitelnosti implementované části aplikace včetně návrhu možných vylepšení.

Elektronicky schválil/a Ing. Michal Valenta, Ph.D. dne 26. ledna 2021 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Bakalářská práce

Educhild - dětská část Android aplikace

Tomáš Hanko

Katedra softwarového inženýrství
Vedoucí práce: Ing. Jiří Hunka

13. května 2021

Poděkování

Rád bych poděkoval Ing. Jiřímu Hunkovi, za možnost vzniku a vedení mé bakalářské práce. Dále bych chtěl poděkovat svým kolegům z týmu. Jmenovitě chci poděkovat Tereze Langové, za skvěle návrhy uživatelského rozhraní a grafické podoby aplikace. Dále Petru Šímovi, za rady a trpělivost při odpovídaní na mé dotazy. V neposlední řadě chci poděkovat své rodině, přítelkyni a svým přátelům za psychickou a morální podporu při studiu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při 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 citovaného zákona.

V Praze dne 13. května 2021 . . . . . .

Odkaz na tuto práci

Hanko, Tomáš. *Educhild - dětská část Android aplikace.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Tato práce se zabývá analýzou, návrhem a implementací dětské části mobilní aplikace na platformě Android a je psaná v programovacím jazyce Kotlin. Tato část poskytuje vzdělávaní pro děti ve formě kvízu. Dále jim poskytuje motivaci k plnění těchto kvízu. Jedna z těchto motivací, je nutnost splnit kvíz, protože je dítěti omezen limit k používaní jiných aplikací. Další forma motivace je získávaní měny a její následní používaní k nákupu odměn od rodiče. Součástí této práce je i analýza dosavadních řešení a zhodnocení

Klíčová slova mobilní aplikace, Android, edukace dětí, motivace ke vzdělaní, Kotlin

Abstract

This thesis is focused on analysis, design and implementation of children mode, which is part of mobile application developed for Android platform written in programming language Kotlin. This mode provides education for children in the form of quizzes. It also provides motivation needed to complete these quizzes. As first part of motivation, it is necessity to complete quiz in order to be allowed using another mobile apps by children. The next thing that motivates the child is that it can earn currency which is used to purchase

rewards given by parents. Analysis of existing solutions and evaluation is part of this thesis.

Keywords mobile application, Android, children education, motivation for learning, Kotlin

Obsah

Úvod **1**

1 Analýza **3**
1.1 Koncept aplikace . . . 3
1.2 Analýza konkurence . . . 3
1.2.1 Quizlet . . . 4
1.2.2 Duolingo . . . 4
1.2.3 Quizizz . . . 5
1.2.4 Khan Academy . . . 6
1.2.5 Shrnutí . . . 6
1.3 Analýza požadavků . . . 7
1.3.1 Vhodný úkol . . . 7
1.3.2 Motivace . . . 7
1.4 Specifikace požadavků . . . 8
1.4.1 Funkční požadavky . . . 8
1.4.2 Nefunkční požadavky . . . 9
1.5 Případy užití . . . 9

2 Návrh **11**
2.1 Platforma . . . 11
2.2 Programovací jazyk . . . 11
2.3 Vývojové prostředí . . . 12
2.4 Knihovny třetích stran . . . 12
2.4.1 Teanity . . . 12
2.4.2 Firebase . . . 12
2.4.2.1 Firebase Crashlytics . . . 12
2.4.3 Koin . . . 12
2.4.4 Room . . . 12
2.4.5 MockK . . . 13

2.4.6 Timber . . . 13
2.4.7 Glide . . . 13
2.4.8 MPAndroidChart . . . 13
2.5 Architektura . . . 13
2.5.1 MVC . . . 13
2.5.2 MVP . . . 14
2.5.3 MVVM . . . 14
2.5.4 Shrnutí . . . 15

3 Implementace **17**
3.1 Použité komponenty . . . 17
3.1.1 Základní komponenty . . . 17
3.1.2 Komponenty navigace . . . 18
3.1.3 Vazební komponenty . . . 18
3.1.4 Jiné komponenty . . . 18
3.2 Tvorba aplikace . . . 20
3.2.1 MainActivity . . . 20
3.2.1.1 KidModeFragment . . . 20
3.2.1.2 KidRewardsFragment . . . 21
3.2.1.3 KidRewardsDetailFragment . . . 21
3.2.2 QuizActivity . . . 21
3.2.2.1 QuizStartFragment . . . 21
3.2.2.2 QuizFragment . . . 21
3.2.2.3 MultichoiceQuestionFragment . . . 22
3.2.2.4 OpenQuestionFragment . . . 22
3.2.2.5 PairsQuestionFragment . . . 22
3.2.2.6 QuestionExplanationFragment . . . 22
3.2.2.7 QuizFinishFragment . . . 22
3.3 Zdrojové kódy . . . 22

4 Testovaní **25**
4.1 Testovaní autorem . . . 25
4.2 Unit testy . . . 25

5 Zhodnocení a výhled do budoucna **27**

Závěr **29**

Literatura **31**

A Seznam použitých zkratek **35**

B Snímky z aplikace **37**

C Obsah přiloženého média **43**

Seznam obrázků

1.1 Ukázka aplikace Quizzlet 4
1.2 Ukázka aplikace Duolingo 5
1.3 Ukázka aplikace Quizziz 6
1.4 Ukázka aplikace Khan Academy 7
1.5 Diagram případů užití 9

2.1 MVC schéma 14
2.2 MVP schéma 15
2.3 MVVM schéma 15

3.1 Ukázka ViewEvent 19
3.2 Ukázka CompoundUseCase 19
3.3 Ukázka Notifyable ve ViewModelu 20

4.1 Ukázka pokrytí odměn testy 26

B.1 Dětské menu 37
B.2 Dětské odměny a detail odměn 38
B.3 Startovací obrazovka kvízu 38
B.4 Otázka s výběrem odpovědí 39
B.5 Otázka s otevřenou odpovědí 39
B.6 Otázka s párováním odpovědí 40
B.7 Vysvětlení otázky 40
B.8 Konec kvízu 41

Seznam tabulek

1.1 Tabulka pokrytí požadavků 10

Úvod

Lidé si od nepaměti snaží ulehčit svou práci i své povinnosti, což umožnilo vzniku různých technologií, které mají tento účel plnit. Jednou z těchto technologií jsou i ty mobilní.
V současné době lze pozorovat stále rostoucí trend popularity v koupi mobilních telefonů. Je běžné mít osobní mobilní telefon a stejně tak nabývají na popularitě i služební telefony, které lidé čím dál častěji dostávají v rámci firemních benefitů.
Jeho primární funkci je ulehčit lidem práci či oprostit je od různých povinností. Této funkce se však dle mého názoru často zneužívá. Mezi povinnosti patří i výchova a edukace dětí. Mnoho rodičů však tuto povinnost bere na lehkou váhu nebo ji dokonce kompletně ignorují. Často tak svým ratolestem půjčí či rovnou zakoupí mobilní telefon za účelem zábavy. Děti pak tráví celé dny s mobilním telefonem, hrají na něm různé hry či sledují svá oblíbená videa bez omezení, pokud jim tato omezení nevymezí rodič. Samotné omezení se však špatně hlídá a rodič se může uchýlit k instalaci různých omezovacích aplikací na mobilní zařízení, aby mohl dítě uhlídat, přestože se právě vyskytuje mimo jeho dosah, Takový přístup zaručí, že dítě bude méně používat mobilní telefon, ale již nezaručí, že se půjde učit do školy či případně jinak vzdělávat. Nemá k tomu důvod ani potřebnou motivaci, a raději si najde jinou zábavu.
Tato myšlenka zapříčinila vznik této práce. Vytvoření aplikace, která umožní rodičům omezit svým dětem zábavu na mobilním zařízení, ale zároveň mu dát přístup ke vzdělaní a motivovat jej k němu.
Aplikace je výsledkem práce týmu a práce má návaznost na další bakalářské práce.

Kapitola 1

Analýza

Tato kapitola je zaměřená na analýzu požadavků kladených na dětskou část tvořené aplikace. Nejprve představím koncept aplikace. Poté provedu analýzu stávajících konkurenčních řešení aplikací. V závěru kapitoly použiji nasbírané informace a provedu specifikaci požadavku a jejich případy užití.

Tato práce je zaměřena pouze na jednu část mobilní aplikace, a proto funkcionality, které řeší jiná část, budou vynechány. Grafickou podobu a samotný návrh uživatelského rozhraní řeší má kolegyně, Tereza Langová, se kterou na tvorbě dětské části aplikace spolupracuji.

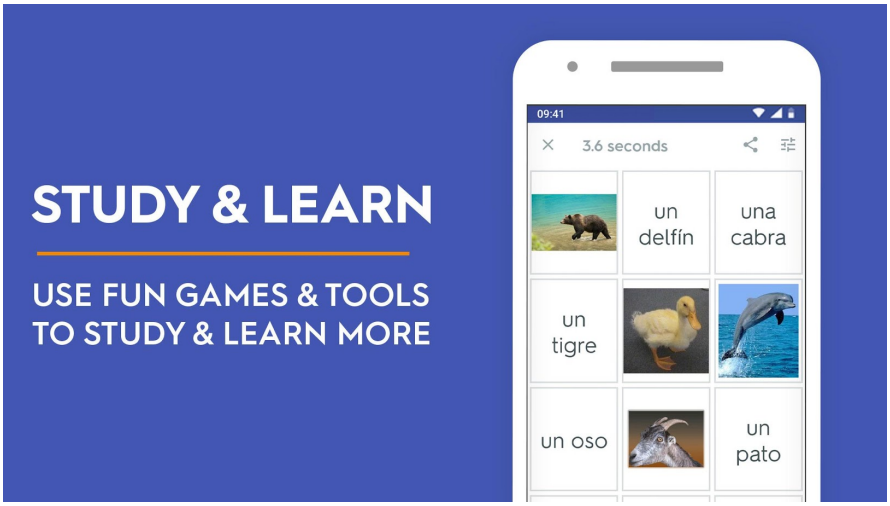
1.1 Koncept aplikace

Základním konceptem mnou tvořené části aplikace je poskytnutí možnosti plnit edukační úkoly. Cílovou skupinou jsou děti ve věku od 6 do 11 let, což odpovídá věku žáků prvního stupně základních škol v České republice. Aplikace má za úkol limitovat dítě v užívaní jiných aplikací na mobilním zařízení. Po vypršení limitu daného rodičem, bude dítěti vyvolán edukační úkol. Pokud chce pokračovat v používaní blokovaných aplikací, musí tento úkol splnit. Je třeba zajistit vhodnou formu úkolu a připadnou motivaci k jejich plnění.

Koncept aplikace vznikl v předmětu BI-SP1. V předmětu BI-SP2.1 následně vznikl základ aplikace a začala prvotní implementace. Má práce na tento základ navazuje.

1.2 Analýza konkurence

V současné době existuje již řada mobilních aplikací, zaměřující se na vzdělaní. S kolegyní jsme si z obchodu Google Play [1] vybrali ty nejoblíbenější a ty které nám přišli zajímavé. Z mého hlediska jsem posuzoval jakou obsahují funkcionalitu, tedy jak zajišťuji vzdělávací i motivační část.



Obrázek 1.1: Ukázka aplikace Quizzlet

1.2.1 Quizlet

Aplikace Quizlet [2] je vzdělávací aplikace, určená především k učení a procvičovaní jazyk. Uživatelé zde mají možnost vytvořit si vlastní kartičky k procvičovaní slovní zásoby či vytvoření vlastních testu. Pokud se uživateli nechce tvořit své testy, může si stáhnout testy vytvořené jinými uživateli. Test obsahuje různé typy úkolů. Tyto úkoly se zde řadí na:

- výběr správného pojmu ze čtyř možností
- otázka s otevřenou odpovědí
- uzavřená otázka s výběrem ano/ne

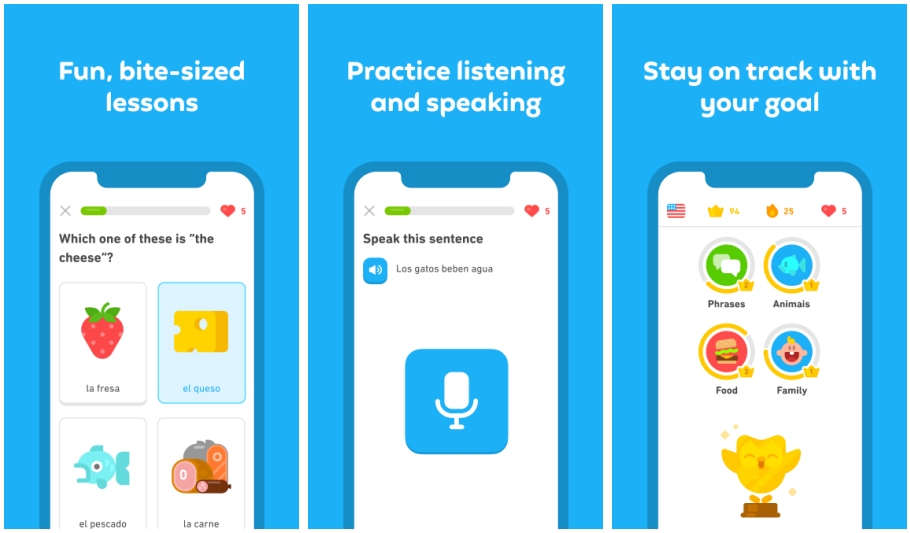
Po každém testu je vyhodnoceno, jak si uživatel vedl.

Dále v aplikaci nalezneme 2 hry. V první hře uživatel přesouvá související pojmy na sebe. Ve druhé hře má za cíl ochránit planetu před padajícími asteroidy tím, že napíše správnou odpověď na pojem vepsaný do daného asteroidu.

Základní aplikace je zdarma podmíněna neustálým vyskakováním reklam. Pro více funkcionalit, jako je například noční režim aplikace, možnost používat aplikaci bez přístupu k internetu či nahráváním obrázku do pojmů, je třeba si zakoupit prémiový balíček.

1.2.2 Duolingo

Duolingo [3] je také aplikace určena k učení jazyků. Obsahuje spousty lekcí přidaný autory aplikace. Každá lekce obsahuje několik kvízů skládajících se ze souboru otázek. Otázky jsou ve formě textu, obrázků nebo dokonce poslechu mluveného slova. Řešení otázek jsou ve formě:



Obrázek 1.2: Ukázka aplikace Duolingo

- výběr správné odpovědi ze dvou až čtyř možností
- uzavřená otázka s výběrem ano/ne
- seřazením textu do správného pořadí

Po vyplnění všech otázek uživatel získá zkušenosti, které se promítnou do jeho úrovně znalostí daného jazyka. Po získaní dostatku zkušeností splní lekci a může se přesunout k další.

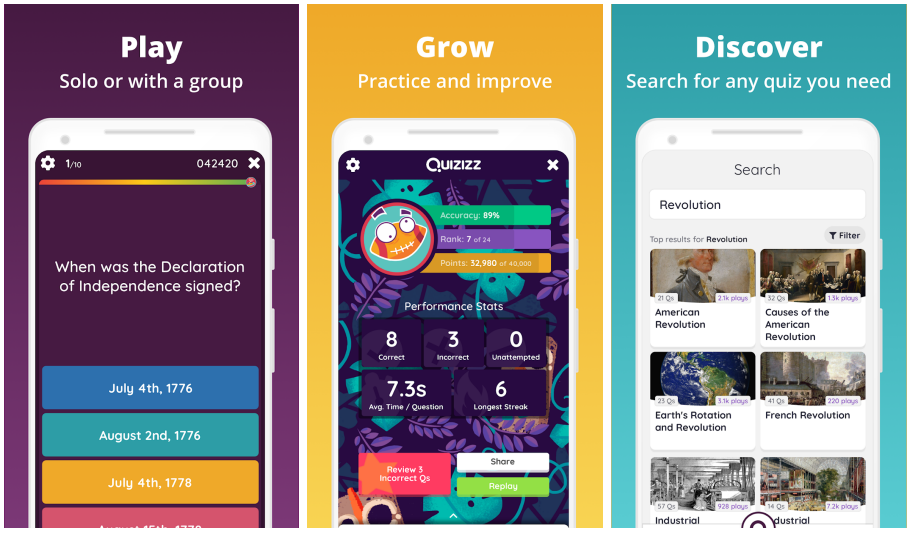
Jako forma motivace je zde odemykaní nejen lekcí, ale také získávaní úspěchů a korun. Koruna je měna aplikace, která uživateli umožní třeba přeskočit daný kvíz. Úspěch odemkne líbivou ikonku poté je splněna jeho podmínka, jako je například získaní dostatečného počtu korun nebo splnění určitého počtu kvízů.

Aplikace je zdarma s použitím vyskakovacích reklam a omezením absolvovaní několika lekcí na den. Je zde také možnost nákupu v aplikaci. Zakoupit zde lze měna aplikace, nebo také zrušení reklam a navýšení počtu lekcí.

1.2.3 Quizizz

Quzizz [4] je aplikace pro obecné vzdělávaní. Zajímavosti teto aplikace je, že zde uživatel nemusí být samotný jedinec, ale může být součásti větší skupiny. Je tedy vhodná i pro školy, kdy učitel může vytvořit třídu a zadat jí různé úkoly, které poté může okomentovat nebo dokonce oznámkovat.

Edukační část je ve formě kvízu, které se uživatel stáhne či vytvoří. Každý kvíz obsahuje různý počet otázek. Na otázku lze odpovídat výběrem jedné správné odpovědi ze čtyř možností nebo otevřenou textovou odpovědí.



Obrázek 1.3: Ukázka aplikace Quizziz

Aplikace ukládá různé statistiky, které si lze kdykoliv zobrazit a sdílet mimo aplikaci. Je zde také možnost vyzvat své přátele k souboji vědomostí.

Mnoho funkcionalit je v základní verzi omezeno. Je zde možnost měsíčního předplatné pro vylepšenou verzi nebo možnost zakoupení školních licencí.

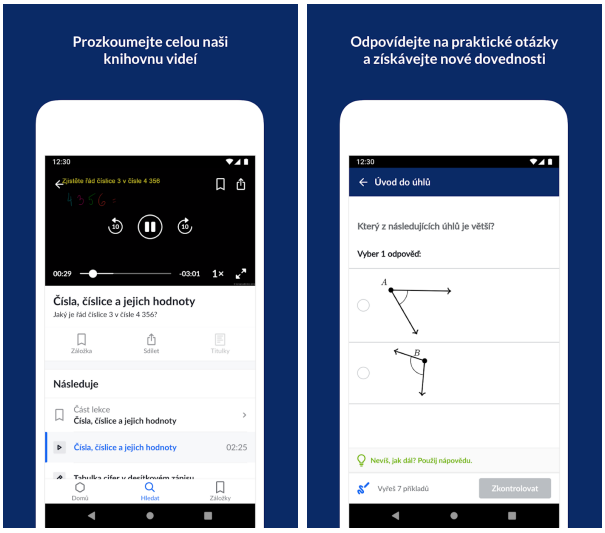
1.2.4 Khan Academy

Aplikace Khan Academy [5] poskytuje vzdělaní ve formě kurzů. Uživatel má možnost stáhnout si velké množství učiva rozdělených dle kategorií i podkategorií. Kurzy obsahují naučná videa, rozsáhle texty a kvízy. V kvízech nalezneme otázky, na které lze odpovídat pouze vybráním ze 4 možností. Jedinou výjimkou jsou matematické otázky, kdy lze odpovědět vepsáním číselné odpovědi. Při každém spuštění kvízu se zobrazí odhadovaná doba vyplnění a počet otázek k vyplnění. Po každé správně zodpovězené otázce vyskočí pochvala, což je zde jediná forma motivace.

Nespornou výhodou této aplikace je, že obsahuje obrovské množství učiva, přičemž je vše zcela zdarma, jelikož je spravována neziskovou organizací .

1.2.5 Shrnutí

Z provedené analýzy bylo zjištěno, že je velice populárni vzdělávaní pomocí kvízů. Forma motivace je ve všech aplikacích buď rozdílná, nebo žádnou motivaci neobsahují.



Obrázek 1.4: Ukázka aplikace Khan Academy

1.3 Analýza požadavků

Z popisu základního konceptu aplikace a z výsledků analýzy konkurenčních řešení jsem vyvodil následující:

1.3.1 Vhodný úkol

Začal jsem zjišťovat, proč se kvízy čím dál častěji užívají ke vzdělávacím účelům. Dle sesbíraných informací [6] je většina dětí považují za zábavné. Soustavné čtení rozsáhlých textů často selhává v udržení pozornosti čtenáře. Takové učení nebývá moc efektivní. Kvízy vzhledem k neustále interakci naopak pozornost dítěte udrží, jelikož zde musí dbát na to, co a proč právě dělají a musí se u nich dostatečně koncentrovat. [7] Dítě si zároveň procvičí co se učí a lépe si to zapamatuje, čímž se kvízy stávají mnohem efektivnějším nástrojem k získávaní vědomostí.
Co se týče tvorby kvízu, není vhodné, aby si dítě vytvářelo své vlastní kvízy. Tvoření kvízu patří do spíše do části aplikace, která je určená pro rodiče.

1.3.2 Motivace

Dle konceptu je o jedné motivaci již zpočátku rozhodnuto. Tou je obnova limitu v používaní aplikací. Díky této motivaci dítě zaručeně splní úkol, to však nezaručí, aby se dítě vzdělávalo i bez vynuceného impulzu.

Podle získaných informací [8] je důležité dítě nenutit k splnění úkolu, ale naležíte jej motivovat, aby je splnil sám. Pokud je dítě dostatečně motivované, je to dobrý předpoklad k tomu, aby své chovaní a prováděné činnosti opako-

valo samo.
[9] Jedna z forem takové motivace jsou odměny. Rodič může dítě odměnit mnoha způsoby. Mezi takové odměny patří například společně strávený čas, společné aktivity, popřípadě zakoupení nějakého dárku. Trofeje nejsou dostatečným způsobem motivace a zároveň nejsou ani způsobem komunikace mezi rodičem a dítětem. Mohou být však podpůrným nástrojem motivace v případě, že rodič nevyužije možnosti odměn.

Primární motivací se tedy stane odblokovaní aplikací. Sekundární motivací se stanou odměny zadané rodičem.

1.4 Specifikace požadavků

Tato sekce popisuje specifikaci funkčních a nefunkčních požadavků odvozené z předešlých sekcí.

1.4.1 Funkční požadavky

- **F1: Zobrazení zbývajícího času do blokace** Uživatel si bude moci zobrazit čas zbývající do blokace aplikací a vyvolání vzdělávací části.

- **F2: Kvízy** Vzdělávací část bude probíhat formou kvízů. Kvíz bude obsahovat soubor otázek. Otázky budou typu: Výběr z více možností, textová odpověď, spojovaní pojmů. Uživatel bude moci tyto kvízy spouštět a plnit.

- **F3: Měna** Uživatel získá za splnění vzdělávací části měnu

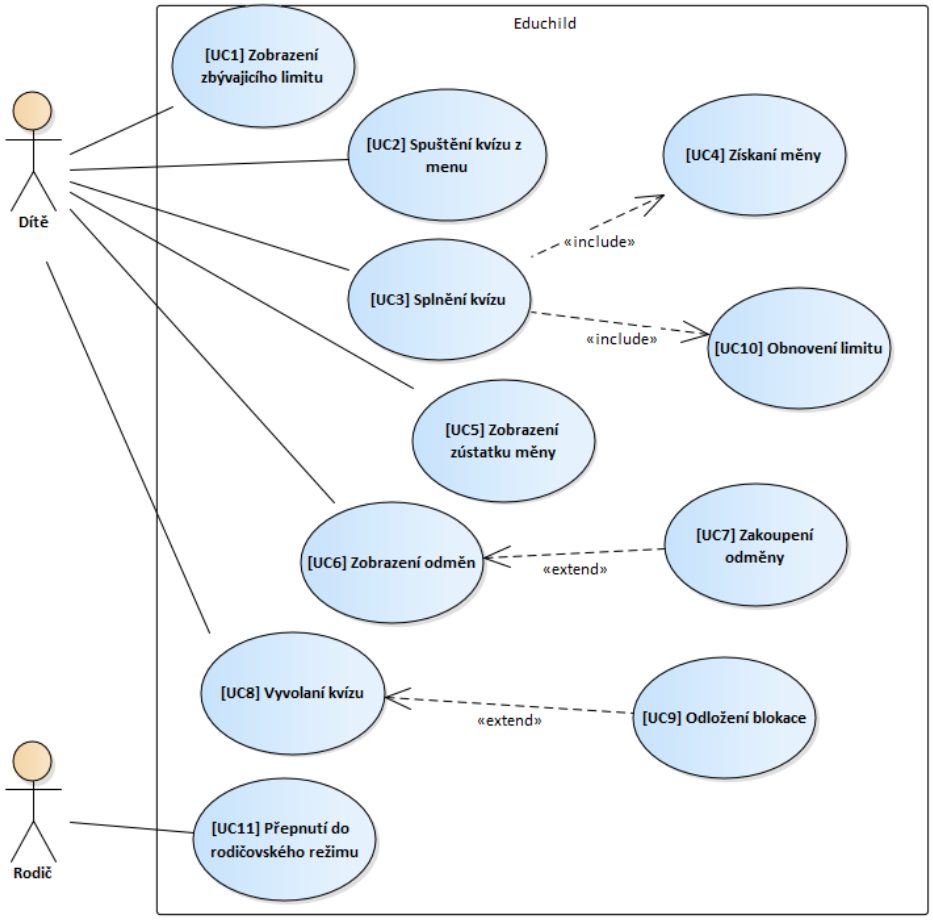
- **F4: Odměny** Uživatel si bude moci zobrazit a zakoupit odměny za získanou měnu

- **F5: Vyvolání kvízu po vypršení limitu** Uživateli bude po vypršení limitu vyvolán kvíz

- **F6: Odložení blokace** Uživatel si bude moci při prvním vyvolání kvízu prodloužit limit a oddálit tak blokaci. Tato možnost bude opět přístupná až po splnění kvízu.

- **F7: Obnovení limitu** Uživatel si obnoví limit pro používaní blokovaných aplikací splněním kvízů

- **F8: Oddělený režim rodiče a dítěte** Uživatel bude moct přepnout z dětského režimu do režimu rodiče .



Obrázek 1.5: Diagram případů užití

1.4.2 Nefunkční požadavky

- **N1: Aplikace na platformě Android** Aplikace bude vytvořena pro operační systém Android
- **N2: Vysoká míra podpory zařízení** Aplikace bude fungovat na velkém množství mobilních zařízení
- **N3: Použitelnost offline** Aplikace bude fungovat i bez internetového připojení

1.5 Případy užití

Tabulka 1.1: Pokrytí požadavků případy užití

	UC1	UC2	UC3	UC4	UC5	UC6	UC7	UC8	UC9	UC10	UC11
F1	x										
F2		x	x								
F3				x	x						
F4						x	x				
F5								x			
F6									x		
F7										x	
F8											x

Kapitola 2

Návrh

Tato kapitola se zaměřuje na návrh první verze dětské části aplikace. Byly zde využity a zpracovaný výsledky z analytické části práce.

2.1 Platforma

Jako platforma na kterou bude aplikace vypuštěna, je operační systém Android. Podíl systému Android na trhu mobilních zařízení v současné chvíli činí 71.12 % což z něj dělá nejrozšířenější mobilní operační systém na světě. [10]

V současné chvílí již bylo vydáno mnoho verzí OS Android. S každým vydáním nové verze jsou představeny nové funkcionality systému. Vždy však uběhne mnoho času, než se nová verze dostane na dostatek zařízení. Proto je důležité správně vybrat verzi systému, na kterou se má aplikace implementovat. Nově vyvíjenou verzí systému (ke dni 10. 5. 2021) je Android 12, ta se však ještě nevyskytuje na trhu a poslední vydanou verzí je tak Android 11 (API 30), která je v současnosti na pouhých 2,4 % podílu používaných verzí OS Android na trhu. [11]

Při vývoji je důležité zvolit správnou minimální verzi systému, aby aplikace fungovala na co největším počtu zařízení, ale zároveň aby nebyla rychlé zastaralá a měla všechny požadované funkcionality. Jako minimální verze systému byla vybraná verze Android 5.0 (API 21), která má dle samotných tvůrců systému podporu na přibližné 94,1 % mobilních zařízení bežících platformě Android.

2.2 Programovací jazyk

Jako programovací jazyk byl vybrán Kotlin, který je podporován jako hlavní a oficiální jazyk určený pro vývoj aplikaci pro Android. Je to moderní jazyk, který nahradil svého předchůdce. [12] Tímto předchůdcem byl programovací jazyk Java. Kotlin je však nadále plně kompatibilní s kódem psaným

v Javě. Oproti Javě však nabízí mnohem přehlednější syntaxi, především díky kratšímu zápisu. Dále umožňuje jednodušší práci s asynchronními operacemi pomocí knihovny Kotlin Coroutines. V neposlední řadě snižuje šanci neočekávaným pádům aplikace díky bezpečnějšímu kódu s představením datového typu *nullable*

2.3 Vývojové prostředí

Jako vývojové prostředí jsem si vybral Android Studio. Oficiální a doporučené vývojové prostředí pro platformu Android, založené na prostředí IntelliJ IDEA od společnosti JetBrains s.r.o.

2.4 Knihovny třetích stran

2.4.1 Teanity

Knihovna Teanity [13] je pateří této aplikace. Přidává přes 500 metod určených k redukci zbytečně opakujícího se kódu a zvýšení celkové přehlednosti. Úzce spolupracuje s knihovnou Kotlin Coroutines a knihovnami řešící životní cykly komponent.

2.4.2 Firebase

Firebase [14] je rozšířenou platformou na poli poskytovali služeb pro vývoj webových a mobilních aplikací. Jednou s funkcionalit, která je v aplikaci používaná je i autentizace uživatelů, tu však má na starost kolega, Daniel Matoušek, ve své bakalářské práci zaměřující se na režim pro rodiče.

2.4.2.1 Firebase Crashlytics

Firebase Crashlytics [15] je nástroj k zaznamenávaní neočekávaných chyb a pádu v reálném čase. V aplikaci je použit jako podpůrná nástroj pro testovaní.

2.4.3 Koin

Koin [16] je knihovna pro funkci automatického vkládaní závislosti. Je to výborný nástroj který vede k lepší přehlednosti kódu a zjednodušení testovaní.

2.4.4 Room

Knihovna Room [17] zjednodušuje práci s lokálním relačním databázovým systémem SQLite, který je využíván v aplikaci.

Architektura knihovny se skládá z jednotlivých Entit, které představují jednu tabulku v databázi, Room Database, což je abstraktní třída definující

entity databáze a Data Access Object, zkráceně DAO, obsahující metody pro čtení a ukládaní entit databáze.

2.4.5 MockK

MockK [18] je knihovna určená k mockovaní objektů, tedy k vytváření testovacích imitací původního objektu. Imitace objektu se chová jako původní objekt, volání jeho metod nám však vrací naše přededem připravená data. Toho se hojně využívá v testovaní.

2.4.6 Timber

Timber [19] je knihovna sloužící k jednoduchému zaznamenávaní výpisu při debugovaní aplikace a získávaní tak potřebných informaci o stavu a hlášeni chyb v testovací fázi.

2.4.7 Glide

Knihovna Glide [20] je určena pro správu medii a rychlé načítaní obrázku v aplikaci. Nabízí spoustu využiti, přičemž její používaní je velice jednoduché.

2.4.8 MPAndroidChart

MPAndroidChart [21] je knihovna používaná ke tvorbě komplexnějších grafů.

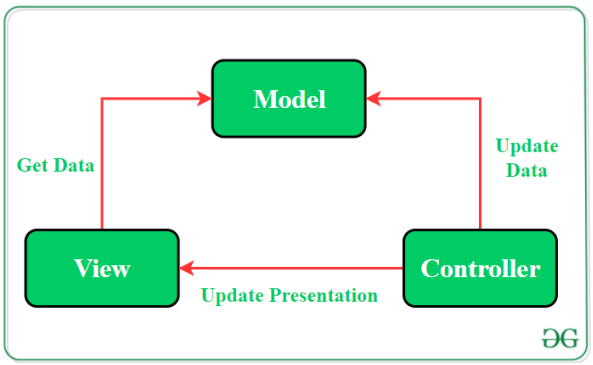
2.5 Architektura

Při vývoji aplikací je doporučováno použít architektonické vzory. Architektonický přidává aplikaci na modularitě. To přináší možnost důkladnějšího testovaní, udržitelnosti kódu a lepší škálovatelnosti aplikace v případe rozšiřovaní funkcionalit. Mezi nejpopulárnější a preferované architektonické vzory ve vývoji aplikací patří třívrstvé vzory. Třívrstvé se nazývají proto, že se kód snaží rozdělit na 3 komponenty nebo také vrstvy, kde každá komponenta zodpovídá za jinou funkcionalitu. Mezi takové vzory patří architektura MVC, MVP a MVVM. [22]

2.5.1 MVC

Vzor MVC, neboli Model-View-Controller, je jeden z nejznámějších a nejpoužívanějších architektonických vzorů. Aplikaci rozděluje na tyto vrstvy:

- **Model** Datová vrstva aplikace. Zodpovídá za komunikaci s databázemi a síťovými prvky .
- **View** Vrstva uživatelského rozhraní. Zobrazuje data uchované v modelu a zodpovída za interakci s uživatelem.



Obrázek 2.1: MVC schéma

- **Controller** Logická vrstva. Tato komponenta navazuje spojení mezi View a Modelem. Získáva odezvu od akcí uživatele z View a aktualizuje data uchovaná v Modelu.

2.5.2 MVP

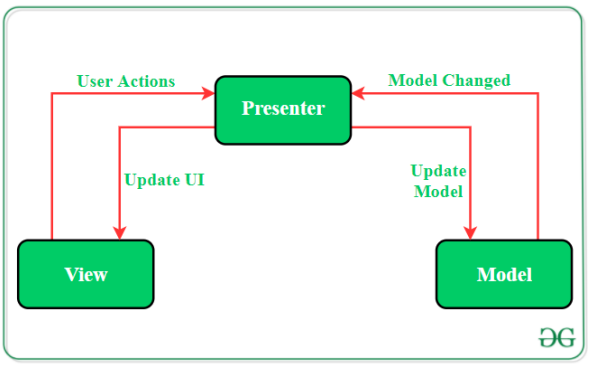
Vzor MVP, neboli Model-View-Presenter, patří k dalším popularním architektonických vzoru. Aplikaci rozděluje takto:

- **Model** Datová vrstva aplikace. Zodpovídá za komunikaci s databázemi a síťovými prvky .
- **View** Vrstva uživatelského rozhraní. Zobrazuje data uchované v modelu a hlída akce uživatele a upozorňuje na tyto provedené akce vrstvu Presenter.
- **Presenter** Logická vrstva. Bere data z Modelu a zodpovídá za to, co se zobrazí ve View. Reaguje na upozornění získané z View,

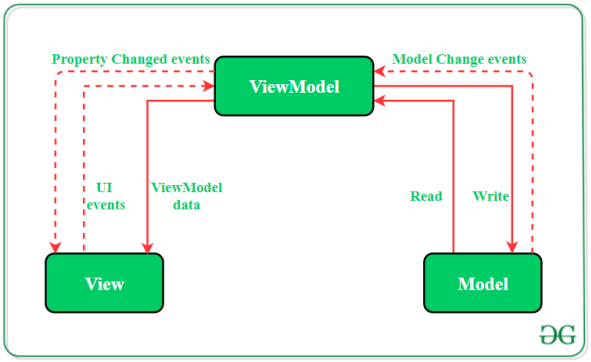
2.5.3 MVVM

Vzor MVVM, neboli Model-View-Viewmodel je poslední z představovaných vzorů a momentálně se mu dostává nejvyšší popularity. Rozděluje aplikaci na tyto komponenty

- **Model** Datová vrstva aplikace. Zodpovídá za abstrakci datových zdrojů. Model i ViewModel spolupracují na získávaní a ukládaní dat
- **View** Tato vrstva informuje ViewModel o akcích uživatele. Neobsahuje žádnou aplikační logiku a pozoruje změny ve ViewModelu



Obrázek 2.2: MVP schéma



Obrázek 2.3: MVVM schéma

- **ViewModel** Zprostředkovává komunikaci mezi Modelem a View. Zpracovává získaná data.

2.5.4 Shrnutí

MVC je z těchto architektur nejstarší a jeho nedostatky daly za vznik MVP. Největší nevýhodou je, že je obsahuje těsnou vazbu mezi View a Modelem, což zabraňuje dobré škálovatelnosti a testovatelnosti aplikace. vzory MVVM i MVP tento nedostatek řeší, MVVM však jde ještě více do hloubky a zbavuje se veškeré závislosti dat (komponenty Model) na View. To znamená, že v architektuře MVP muže být komponenta Presenter vztahu k pouze jedné komponentě View, zatímco v architektuře MVVM muže být jedna komponenta ViewModel namapovaná k více komponentám View najednou. Architektonický vzor MVVM je doporučován pro tvorbu aplikací na platformě Android a i při implementaci této práce je využit.

Kapitola 3

Implementace

Tato kapitola je zaměřená na vlastní implementaci aplikace. Na úvod představím dané komponenty a poté přejdu na tvorbu funkcionalit a uživatelského rozhraní.

3.1 Použité komponenty

3.1.1 Základní komponenty

Mezi základní komponenty patří:

- ***Activity***
  Základ každé Android aplikace. Jedná se o prvek UI, zodpovědný za vytvoření jedné obrazovky. Nejčastěji se používá jako kontejner pro zobrazovaní fragmenty. V MVVM architektuře se jedná o komponentu View. [23]

- ***Fragment***
  Prvek UI zobrazený v aktivitě. Jeden fragment je jedna část uživatelského rozhraní. Jedná se o komponentu View.[24]

- ***Service***
  Služba bežící na pozadí, múže zajistit funkcionalitu ikdyž uživatel výjde z aplikace. Muže vyvolat různé akce. V aplikaci je implementován pro kontrolu limitu dítěte a vyvolaní kvízu. [25]

- ***ViewModel***
  Jak již název napovídá, ViewModel je i stejnojmenná komponenta architektonického vzoru MVVM. Zpracovává data a řeší co má ukázat komponenta View, tedy *fragment* či *activity*. [26]

- ***Layout***
  Definuje vizuální strukturu uživatelského rozhraní. Ve formátu XML. [27]

3.1.2 Komponenty navigace

Tato komponenta obsluhuje navigaci v aplikaci [28]. Mezi tyto komponenty patří:

- ***Navigation graph***
  Zdrojový soubor ve formátu XML. Obsahuje všechny s navigací spjaté informace.

- ***NavHost***
  Prázdný kontejner který zobrazuje cílové komponenty definované v Navigation graph. Implementace tohoto kontejneru, *NavHostFragment* zobrazuje cílové fragmenty.

- ***NavController***
  Objekt který spravuje navigaci v kontejneru *NavHost* a zaměňuje jejich obsah.

3.1.3 Vazební komponenty

Binding komponenta, neboli vazební komponenta, funguje jako vazba mezi komponentou a UI prvkem definovaným v *layoutu*. Mezi binding komponenty patří:

- ***View Binding***
  Slouží k vazbě *Activity* nebo *Fragment* na UI prvky v *layoutu*. Nahrazuje volání metody *findViewById*. Vazba je vytvořena automaticky. [29]

- ***Data Binding***
  Slouží k vazbě *ViewModel* na UI prvky v *layoutu*. Lze díky němu použít data přímo v *layoutu*. K tomu se používa specialní syntaxe.[30] Narozdíl od *View Binding* lze vazbu použít obousměrně,tedy například při vepsaní textu uživatelem. [31]

3.1.4 Jiné komponenty

Do této sekce bych zařadil 3 komponenty vytvořené v používané knihovně Teanity. V aplikaci jsou velmi často používaný.

- ***ViewEvent***
  Tato komponenta je abstraktní třída používaná pro předávaní událostí z *ViewModel* do *Activity/Frament*.

```kotlin
/**
 * @author Tomáš Hanko
 *
 * ViewEvent used to load question fragments for [QuizFragmentStateAdapter] in [QuizFragment]
 */
class QuizLoadViewEvent(private val fragments: ArrayList<Fragment>) : ViewEvent(), FragmentExecutor {
    override fun invoke(fragment: Fragment) {
        if(fragment !is QuizFragment) return
        fragment.binding.questions.adapter = QuizFragmentStateAdapter(
                fragment.childFragmentManager,
                fragment.lifecycle,fragments)
        fragment.binding.questions.offscreenPageLimit = 2
        fragment.binding.questions.isUserInputEnabled = false
    }
}
```

Obrázek 3.1: Ukázka ViewEvent

```kotlin
/**
 * UseCase used for getting questions and their answers in quiz
 *
 * Returns List of [QuizQuestion] using string as uuid of quiz
 *
 * @author Tomáš Hanko
 */
class GetQuizQuestionsUseCase(
        private val questionsDao: QuizQuestionDao
) : CompoundUseCase<String,List<QuizQuestion>>(){
    override suspend fun execute(input: String): List<QuizQuestion> {
        return questionsDao.getQuestionsWithAnswersByQuizId(input)
    }
}
```

Obrázek 3.2: Ukázka CompoundUseCase

- ***CompoundUseCase***
  Abstraktní třída využívaná pro testovatelné a izolované užití business logiky. Lze volat asynchronně díky Kotlin Coroutines. Přes třídy implementující tuto třídu se získávají data z lokální databáze pomocí *DAO* metod

- ***Notifyable***
  Interface, který implementuje veřejný interface *Observable*.[32] Využívá se pro pozorovaní změn přes *Databinding* nebo manuálně přiřazených posluchačů. Velmi často používaný ve *ViewModelu*

```
var name: String by observable( initialValue: "", BR.name)
    @Bindable get
    private set

var category: String by observable( initialValue: "", BR.category)
    @Bindable get
    private set

var text: Text? by observable( initialValue: null, BR.text)
    @Bindable get
    private set
```

Obrázek 3.3: Ukázka Notifyable ve ViewModelu

3.2 Tvorba aplikace

Zde v rychlosti popíšu jak jsem tvořil jednotlivé obrazovky a jejich funkce.

Aplikace se v současné chvílí skládá ze dvou *Activity*.

MainActivity Ve které jsou obrazovky s dětským navigačním menu a odměnami

QuizActivity Reprezentuje jeden spuštěný kvíz

Ve všech obrazovkách je *Data binding* užitý *ViewModely* a *layouty*, případně je použit *View binding* z *fragmentů* pomocí volaní tříd *ViewEventů* také z *ViewModelu*.

3.2.1 MainActivity

MainActivity je hlavní obrazovka aplikace. Má implementace je pouze část této aktivity.

3.2.1.1 KidModeFragment

KidModeFragment zobrazuje dětské menu se všemi základními informacemi a potřebnou navigací.
Mezi informace patří zobrazení zbývajícího času do blokace aplikací a vyvolaní kvízu. Toto zobrazení času vyžadovalo vytvoření vlastního UI prvku za pomoci knihovny MPAndroidChart. Dále je zde vidět současný zůstatek získávané měny. V současné chvíli jsou zde k dispozici tři tlačítka. První nás dostane na obrazovku spuštění kvízu, další nás dostane do přehledu odměn a poslední nám vyvolá požadavek na přechod do rodičovského módu. B.1

3.2.1.2 KidRewardsFragment

KidModeFragment představuje část aplikace, kde má dítě možnost vidět možné odměny od rodičů. Je zde zobrazen seznam v získaný z lokální databáze pomocí vyvolaní tříd dědicích z *CompoundUseCase* a předaní do UI prvku *RecyclerView* [33] pomocí *Databinding*. Nad seznamem se nachazí tlačítko, které dítě navede zpět do menu a vidí svůj zůstatek. Na každé položce seznamu je zobrazen název odměny, dále je vidět možný obrázek a cena. Pokud má dítě dostatečný zůstatek měny, může si odměnu tlačítkem zakoupit. Jestliže však nemá, je mu zobrazeno, kolik ještě potřebuje získat měny, aby si danou odměnu mohl zakoupit. Dítě si může také na danou odměnu kliknout a přejde tak do jejího detailu. B.2

3.2.1.3 KidRewardsDetailFragment

KidRewardsDetailFragment zobrazí detail celé odměny, skládající se z názvu, volitelného obrázku, popisu odměny a jeho ceny. V Případě že má dítě dostatek měny, bude zde možnost zakoupení a vypočtení nového zůstatku předem, ve druhém případě zde bude výpočet, kolik pro danou odměnu je potřeba získat měny.

3.2.2 QuizActivity

QuizActivity představuje obrazovku kvízu, od jeho spuštění, vyplnění i ukončení. Tato *activity* je vyvolána ze *service* po vypršení limitu nebo spuštěna dítětem z dětského menu.

3.2.2.1 QuizStartFragment

QuizStartFragment zobrazí název kvízu, kategorii ve které se kvíz nachází a na kolik otázek je potřeba správně odpovědět, aby se dal kvíz považovat za splněný. Je zde tlačítko spouštějící kvíz. Pokud je kvíz vyvolán po vypršení limitu, je zde také možnost kvíz na 5 minut odložit. Toto lze však pouze jedno. Při dalších vyvolaní kvízu je tato možnost zakázaná, dokud není splněn kvíz. B.3

3.2.2.2 QuizFragment

V *QuizFragment* je umístěn prvek *ViewPager2*, ve kterém jsou umístěny všechny otázky kvízu a jejích vysvětlení. [34] Zvolil jsem tento prvek, protože je to prvek určený k uchovaní více *fragmentů* najednou, jejich zobrazovaní a jednoduchému přesouvaní mezi nimi přejetím prstem s příslušnou animací. K uchovaní *fragmentu* využívá *FragmentStateAdapter* [35]. Funkce přejetí prstem mezi nimi je nahrazena stiskem tlačítka. *ViewPager* je umístěn aby zobrazoval jednotlivé *fragmenty* otázek a vysvětlení a nezabírá celou plochu obrazovky.

Nastavení jak moc velkou plochu chceme jednotlivými *fragmenty* zabírat je další užitečná vlastnost, která umožní zobrazit i jiné prvky, například ty, které jsou tyto obrazovky společné. V mém případě je zde vidět prvek *HorizontalProgressBar* který zobrazuje počet již správně zodpovězených otázek z počtu potřebných otázek k splnění kvízu.

3.2.2.3 MultichoiceQuestionFragment

MultichoiceQuestionFragment Zobrazí otázku a možnosti, ze kterých lze vybírat odpovědi. Počet možností je minimálně 2 a maximálně 4, přičemž správně může být i více odpovědi najednou. B.4

3.2.2.4 OpenQuestionFragment

OpenQuestionFragment Zobrazí otázku a textové pole, při jehož nakliknutí se zobrazí softwarová klávesnice, pomocí které dítě zapíše svou odpověď. B.5

3.2.2.5 PairsQuestionFragment

PairsQuestionFragment Zobrazí otázku a odpovědi. Odpovědi musí správně spárovat přetažením. Na tuto funkcionalitu je použit prvek *RecyclerView* a na zprostředkovaní metody přetažení je použita knihovní třída *ItemTouchHelper*. [36] B.6

3.2.2.6 QuestionExplanationFragment

QuestionExplanationFragment po každém zodpovězení otázky zobrazí její vysvětlení.B.7

3.2.2.7 QuizFinishFragment

QuizFinishFragment je poslední obrazovka v *QuizActivity*. Zobrazí se pouze po splnění kvízu. Dítě zde uvidí, že splnil všechny otázky a kolik za to dostal měny. Dále uvidí kolik dostal času na blokované aplikace, a to jak textově, tak také pomocí prvku *HorizontalProgressBar*. Tlačítko jej pak dostane zpět do *MainActivity* a zobrazí mu navigační menu. B.8

3.3 Zdrojové kódy

K ukládání zdrojových kódu a jejich historie je použit školní webový repozitář GitLab. Funkcionality jsou členěny do větví *feature/názevFunkcionality*, které byly průběžně mergovany do hlavní větve *master*. Případné opravení chyb probíhalo ve větvích *fix/coOpravujeme*.
Odkaz na zdrojové kódy práce: `https://gitlab.fit.cvut.cz/simapet4/educhild-android/tree/feature/rewards`

V tomto repozitáři je nastavená automatická CI/CD pipeline, která při každém požadavku na merge do větve *master* zkontroluje, zda se aplikace správně sestaví, zkontroluje chybný kód nebo redundantní importy či jiné vady, a spustí automatické testy.

Kapitola 4

Testovaní

V této kapitole popíšu, jak probíhalo testovaní mé práce. Má implementovaná část nebyla testována uživatelskými testy, jelikož nestojím za návrhem grafické podoby a uživatelského rozhraní, toto testovaní v pozdějších verzích provede má kolegyně, Tereza Langová.

4.1 Testovaní autorem

Aplikaci jsem průběžně testoval během vývoje. Testoval jsem zejména funkčnost aplikace a řešil její pády. Z počátku jsem využíval virtuální mobilní zařízení, které nabízí vývojové prostředí. Možnosti více zařízení s libovolnou velikostí či rozdílnou verzí OS Android jsem ocenil. Tohle testovaní bylo důležité zejména kvůli responzivitě uživatelského rozhraní a zjištění, zda se na různých verzích systému aplikace chová stále stejně.

Po skončení implementace funkcionalit jsem si aplikaci nahrál na svůj starý mobilní telefon, a stále připojený k počítači jsem ve svém vývojovém prostředí vychytával chyby.

Díky knihovně *Timber*, která umožňuje jednoduché kromě logovaní zpráv a chyb, jsem mohl zkoumat při zkoušení aplikace jak se co chová.

Prostřednictvím služeb *Firebase Crashlytics* mi také byly zasílané zprávy o chybách a pádech aplikace, kde k nim došlo a která část kódu daný problém způsobila.

Dále jsme zkoumal lokálně uloženou databázi pomocí databázového inspektora, které Android studio také obsahuje. Pro některé testy jsem si musel často něco vymyslet a přidávat záznamy.

4.2 Unit testy

Dalším typem testovaní mé práce byla implementace automatických jednotkových testů. V aplikaci je používaná lokální databáze s použitím knihovny

Coverage Summary for Package: cz.fit.cvut.persistence.usecase.rewards

Package	Class, %	Method, %	Line, %
cz.fit.cvut.persistence.usecase.rewards	100% (4/ 4)	100% (8/ 8)	100% (16/ 16)

Class	Class, %	Method, %	Line, %
PurchaseRewardUseCase	100% (1/ 1)	100% (2/ 2)	100% (7/ 7)
GetRewardsByChildUseCase	100% (1/ 1)	100% (2/ 2)	100% (3/ 3)
GetRewardById UseCase	100% (1/ 1)	100% (2/ 2)	100% (3/ 3)
GetNotPurchasedRewardByChildUseCase	100% (1/ 1)	100% (2/ 2)	100% (3/ 3)

Obrázek 4.1: Ukázka pokrytí odměn testy

Room. Samotnou databázi jsem až na malé změny netvořil, o to se totiž postarali moji kolegové, přesto jí má implementovaná část velmi často využívá. Například na odměny či získávaní kvízu jsem si musel vytvořit své metody. Získávaní těchto dat bylo zprostředkováno třídami, které implementovaly *CompoundUseCase*. Především kvůli této třídě, bylo vytvoření testů poměrně jednoduché. Na automatické testy a imitaci objektu s daty byla využita knihovna *MockK*.

Kapitola 5

Zhodnocení a výhled do budoucna

S analýzou, návrhem a implementací mého řešení jsem poměrně spokojený. Všechny funkcionality, které jsem si vytyčil fungují tak jak jsem si představoval. Co se týče přidávaní dalších funkcionalit a tedy škálovatelnosti, je na tom má část aplikace dobře. Stejně tak i přehlednost a pochopitelnost kódu.

Při analýze konkurenčních řešení mě nepřekvapilo, že jsou kvízy tak populárni vzdělávací formou. Co mě naopak zarazilo, je rozdílnost v přístupu z hlediska motivace. Některé aplikace dokonce motivaci, pokud opomenu získané vědomosti, ani nemají. Mě osobně žádná z nalezených možností nevyhovovala. Mé řešení, odměny zadávané rodičem, jsou ne jen dle mého názoru skvělou motivací pro děti. Nevidím důvod, proč by tato forma motivace nemohla fungovat i pro staršího studenta či dospělou osobu, a to s jedinou změnou, že by si tyto odměny vytyčil sám. Umím si představit, že by si dotyčný uživatel chtěl průběžně procvičovat různé otázky na test či zkoušku do školy a po splnění by si za odměnu třeba něco koupil. Aplikace by hlídala, plnění těchto kvízu a při splnění dostatečného počtu otázek by uživatele upozornila, že si svou odměnu zasloužil. V případě mé práce toto řeší měna a odměny od rodiče. Pro některé rodiče se možná může jevit jako přítěž a tuto funkci nevyužívat, dle analýzy je to však dobrá možnost jak motivovat a podpořit své děti ve vzdělávaní.

Co se týče implementace, byla pro mě zpočátku těžká. V jazyku Kotlin a na tvorbě aplikace v Android jsem pracoval poprvé a musel jsem se učit vše od začátku, co a jak funguje. Myslím si, že se má práce s kódem výrazně zlepšovala každou přidanou funkcí, přesto zde vidím ještě poměrně velký možný posun v učení a ve zkušenostech. Zmíním převážně část s kvízy, kde i přes to, že vše funguje tak jak má a jejich funkcionalita se dá se stále dobře rozšiřovat, je zde dle mého názoru, možnost zlepšení mé implementace kódu. Je to jeden z mých hlavních cílů do budoucna.

V testovaní vidím menší zádrhel v neprovedení uživatelského testování. Toto však není v popisu mé práce a mého řešení. Po jejich provedení je však

má implementovat případně změny.

Z mého pohledu je aplikace přínosem na poli mobilních aplikací určených ke vzdělání. Jedním z důvodu je určitě omezovaní dítěte v používaní různých aplikací a ztrácení svého času. Dalším důvodem je, že aplikaci vidím jako další prostředek v komunikaci mezi rodičem a dítětem.

Na této aplikaci se bude pracovat i nadále a já, stejně jako v této práci, budu pokračovat v implementaci dětské části. V budoucnu plánuji implementaci:

- Ukládaní statistik kvízu a jejich předávaní
- Ukládaní odpovědí dítěte, aby si je poté mohl rodič zobrazit
- Trofeje a jejich získávaní
  Bude to přidaná funkce a jako menší alternativa místo odměn od rodičů, jelikož se rodič může rozhodnout odměny nevyužívat.
- Kontrola a refactoring kódu

Závěr

Dle zadaní práce jsem provedl analýzu stávajících řešení a požadavků, následně jsem na ní navázal návrhem a implementací mého řešení. Výsledkem je funkční dětská část mobilní aplikace. Tato část aplikace zprostředkovává vzdělávaní pro děti formou kvízu. K plnění kvízu jej motivuje především blokace jiných aplikací. Po splnění kvízu dostane obnovený čas k používaní blokovaných aplikací. Dále jej motivuje získávaná měna, za kterou si od bude moct zakoupit odměny, jenž mu nabídne rodič. Nakonec jsem své řešení zhodnotil a nastínil budoucí pokračovaní ve vývoji aplikace a jejího dětského módu.

Literatura

[1] Google: *Google Play [online]*. [cit. 2021-05-12]. Dostupné z: https://play.google.com/store

[2] Quizlet Inc: *Quizlet[online]*. [cit. 2021-05-12]. Dostupné z: https://play.google.com/store/apps/details?id=com.quizlet.quizletandroid

[3] Duolingo: *Duolingo [online]*. [cit. 2021-05-12]. Dostupné z: https://play.google.com/store/apps/details?id=com.duolingo

[4] Quizziz Inc: *Quizziz[online]*. [cit. 2021-05-12]. Dostupné z: https://play.google.com/store/apps/details?id=com.quizizz_mobile

[5] Khan Academy: *Khan Academy[online]*. [cit. 2021-05-12]. Dostupné z: https://play.google.com/store/apps/details?id=org.khanacademy.android

[6] *Why are Quizzes Valuable in Education? [online]*. [cit. 2021-05-12]. Dostupné z: https://www.educationquizzes.com/knowledge-bank/why-are-quizzes-valuable-in-education/

[7] *Kvízy udrží pozornost [online]*. [cit. 2021-05-12]. Dostupné z: https://perpetuum.cz/2016/04/kvizy-udrzi-pozornost/

[8] *How To Stay Motivated To Study [online]*. [cit. 2021-05-12]. Dostupné z: https://www.oxfordlearning.com/how-to-stay-motivated-to-study/

[9] *How to Use Rewards [online]*. [cit. 2021-05-12]. Dostupné z: https://www.cdc.gov/parents/essentials/consequences/rewards.html

[10] *Mobile & Tablet Operating System Market Share Worldwide - April 2021 [online]*. [cit. 2021-05-12]. Dostupné z: https://gs.statcounter.com/

os-market-share/mobile-tablet/worldwide/#monthly-202004-202104-bar

[11] *Mobile & Tablet Android Version Market Share Worldwide - April 2021 [online].* [cit. 2021-05-12]. Dostupné z: https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide/#monthly-202004-202104-bar

[12] *Kotlin [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/kotlin

[13] *Teanity[online].* [cit. 2021-05-12]. Dostupné z: https://github.com/skoumalcz/teanity

[14] Google: *Firebase [online].* [cit. 2021-05-12]. Dostupné z: https://firebase.google.com/

[15] Google: *Firebase [online].* [cit. 2021-05-12]. Dostupné z: https://firebase.google.com/products/crashlytics

[16] *Koin [online].* [cit. 2021-05-12]. Dostupné z: https://insert-koin.io/

[17] *Room [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/training/data-storage/room

[18] *MockK [online].* [cit. 2021-05-12]. Dostupné z: https://mockk.io/

[19] *Timber [online].* [cit. 2021-05-12]. Dostupné z: https://github.com/JakeWharton/timber

[20] *Glide [online].* [cit. 2021-05-12]. Dostupné z: https://github.com/bumptech/glide

[21] *MPAndroidChart [online].* [cit. 2021-05-12]. Dostupné z: https://github.com/PhilJay/MPAndroidChart

[22] *Difference Between MVC, MVP and MVVM Architecture Pattern in Android [online].* [cit. 2021-05-12]. Dostupné z: https://www.geeksforgeeks.org/difference-between-mvc-mvp-and-mvvm-architecture-pattern-in-android/

[23] *Activity [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/reference/android/app/Activity

[24] *Fragments [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/guide/fragments

[25] *Services [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/guide/components/services

[26] *ViewModel [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/topic/libraries/architecture/viewmodel

[27] *Layouts [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/guide/topics/ui/declaring-layout

[28] *Navigation [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/guide/navigation

[29] *View Binding [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/topic/libraries/view-binding

[30] *Data Binding [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/topic/libraries/data-binding

[31] *Android Two-Way Data Binding [online].* [cit. 2021-05-12]. Dostupné z: https://bignerdranch.com/blog/two-way-data-binding-on-android-observing-your-view-with-xml/

[32] *Observable [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/reference/kotlin/java/util/Observable

[33] *RecyclerView [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/jetpack/androidx/releases/recyclerview

[34] *Create swipe views with tabs using ViewPager2 [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/guide/navigation/navigation-swipe-view-2

[35] *FragmentStateAdapter [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/reference/androidx/viewpager2/adapter/FragmentStateAdapter

[36] *ItemTouchHelper [online].* [cit. 2021-05-12]. Dostupné z: https://developer.android.com/reference/androidx/recyclerview/widget/ItemTouchHelper

Příloha A

Seznam použitých zkratek

UI User Interface

API Application Programming Interface

XML Extensible markup language

MVC Model-View-Controller

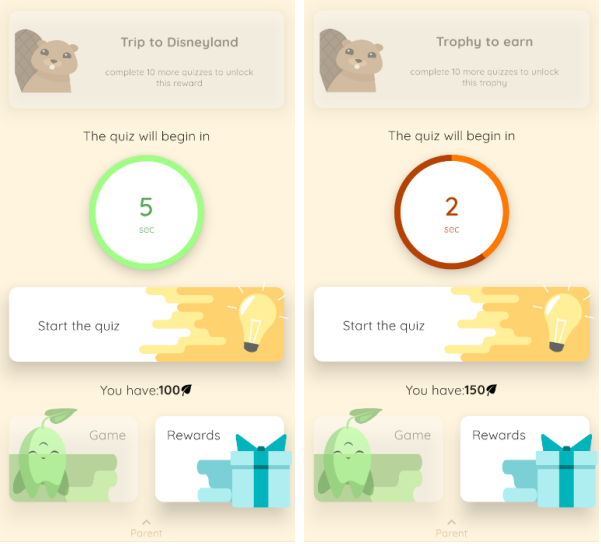
MVP Model-View-Presenter

MVVM Model-View-ViewModel

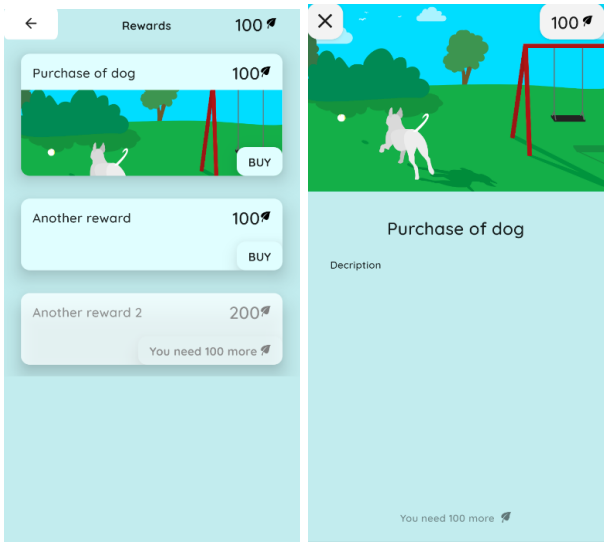
DAO Data access object

Příloha B

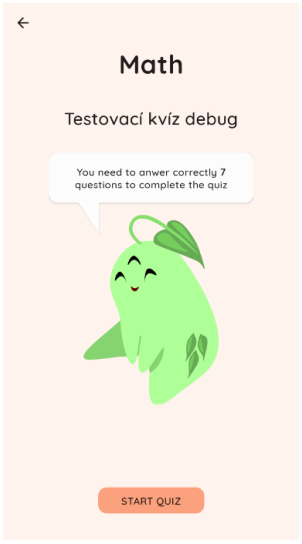
Snímky z aplikace



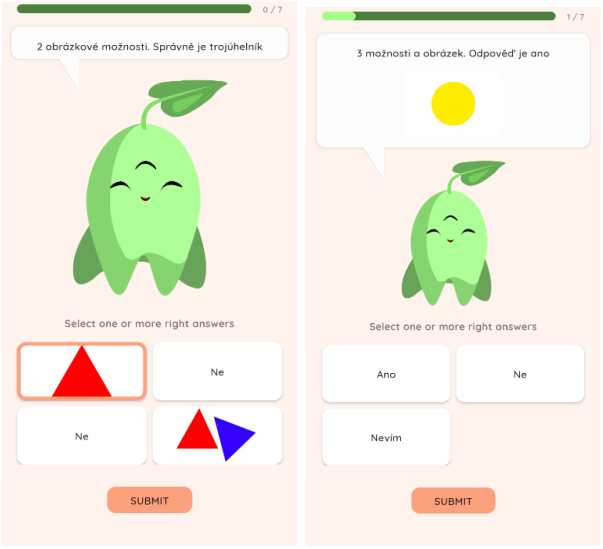
Obrázek B.1: Dětské menu



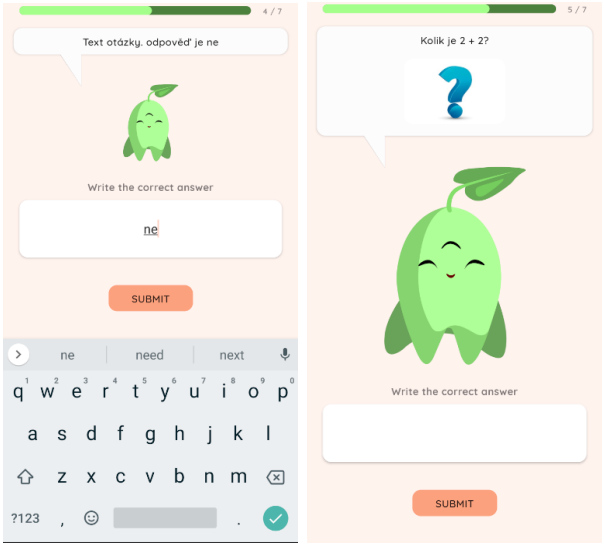
Obrázek B.2: Dětské odměny a detail odměn



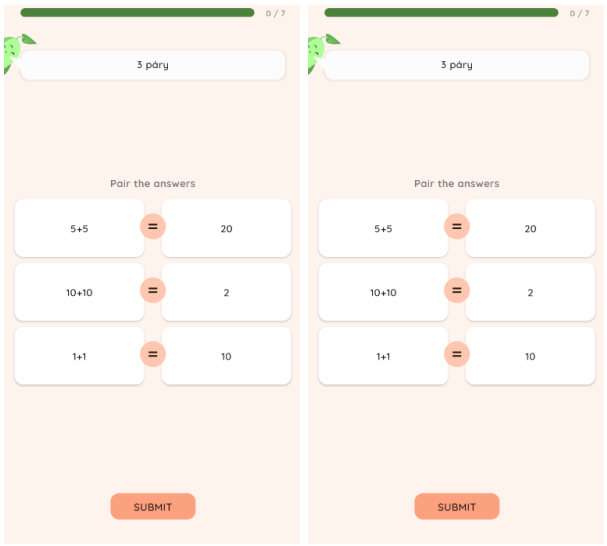
Obrázek B.3: Startovací obrazovka kvízu



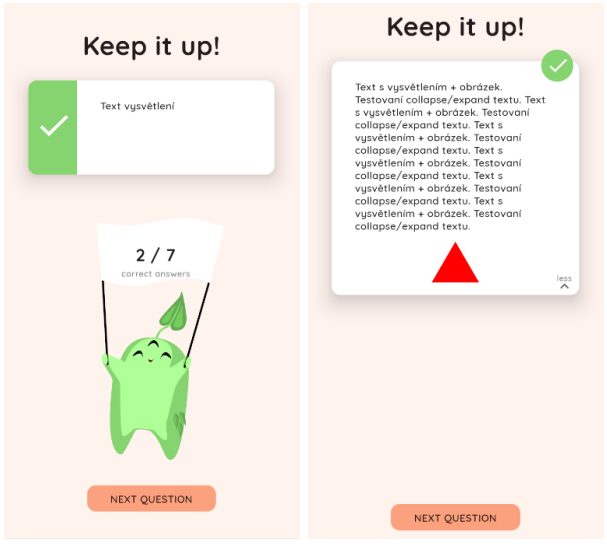
Obrázek B.4: Otázka s výběrem odpovědí



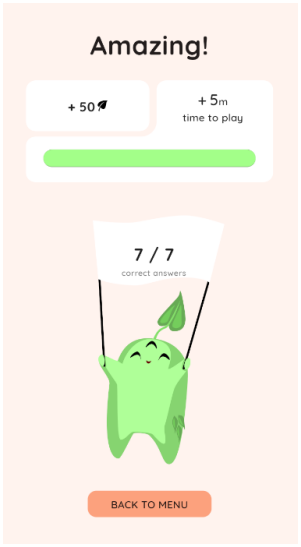
Obrázek B.5: Otázka s otevřenou odpovědí



Obrázek B.6: Otázka s párováním odpovědí



Obrázek B.7: Vysvětlení otázky



Obrázek B.8: Konec kvízu

Příloha C

Obsah přiloženého média

- `thesis` ..... zdrojová forma práce ve formátu LaTeX
  - `img` ..... použité obrázky
    - `app` ..... snímky z aplikace
- `text` ..... text práce
  - `thesis.pdf` ..... text práce ve formátu PDF