

GYMNÁZIUM JANA KEPLERA

Parléřova 2/118, 169 00 Praha 6



Mobilní aplikace na podporu čtení

Maturitní práce

Autor: Ondřej Kolačný

Třída: 4.B

Školní rok: 2020/2021

Předmět: Informatika

Vedoucí práce: Šimon Schierreich

Praha, 2021



GYMNASIUM JANA KEPLERA
Kabinet informatiky

ZADÁNÍ MATURITNÍ PRÁCE

Student: Ondřej Kolačný

Třída: 4.B

Školní rok: 2020/2021

Platnost zadání: 30. 9. 2021

Vedoucí práce: Šimon Schierreich

Název práce: Mobilní aplikace na podporu čtení

Pokyny pro vypracování:

Cílem práce je vytvořit mobilní aplikaci, která by měla sloužit jako nástroj pro podporu pravidelnosti čtení (nejen) pro děti z prvního stupně základních škol (nejen) v době distanční výuky. Aplikace uživateli na základě předem vytvořeného plánu pravidelně připomíná, že má číst a nabízí sledování postupu v dané knize. Za pravidelnou činnost a plnění plánu má uživatel možnost následně získat různé odměny.

Doporučená literatura:

[1] DIMARZIO, Jerome J. F. *Android: A Programmers Guide*. New York: McGraw-Hill, 2008. ISBN 9780071599887.

[2] EVANS, Eric. *Domain-Driven Design: Tackling Complexity In the Heart of Software*. Boston: Addison-Wesley Longman Publishing, 2003. ISBN 978-0-321-12521-7.

[3] MARTIN, Robert C. *Design Principles and Design Patterns*. www.objectmentor.com, 2000. Dostupné z:

https://fi.ort.edu.uy/innovaportal/file/2032/1/design_principles.pdf.

[4] FOWLER, Martin. *Patterns of enterprise application architecture*. Boston: Addison-Wesley Professional, 2003. ISBN 978-0-321-12742-6.

URL repozitáře:

<https://gitlab.com/gjk/maturitni-projekt>

vedoucí práce

student

V Praze dne 30. 10. 2020

Prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů. Nemám žádné námitky proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Praze dne 7. 4. 2021

Ondřej Kolačný

Poděkování

Poděkování patří panu Šimonu Schierreichovi, který mi pomohl cennými radami.

Abstrakt

Tato práce je dokumentací k maturitnímu projektu z informatiky, jehož cílem bylo vytvořit komunikační kanál mezi učitelem a žákem základní školy prostřednictvím mobilní aplikace pro operační systém Android. Tento komunikační prostředek má sloužit k zprostředkování kontroly plnění zadaných úkolů z předmětu český jazyk / literatura spočívajících v četbě určitého počtu stran knihy. V první části práce je zdůrazněna problematika zhoršující se čtenářské aktivity. Druhá část práce popisuje postup, představuje řešené problémy, rozebírá, jakých cílů bylo a nebylo dosaženo, jakým způsobem je aplikace implementována. Poslední, třetí část dává instrukce jak postupovat při zprovoznění.

Klíčová slova

Android, aplikace, čtení, vzdělávání, motivace, MySQL databáze

Abstract

This work is a documentation for the graduation project in informatics, the aim of which was to create a communication channel between a teacher and a primary school student through a mobile application for the Android operating system. This means of communication is intended to mediate the control of the fulfillment of the assigned tasks from the subject Czech language / literature consisting in reading a certain number of pages of the book. The first part of the thesis emphasizes the issue of deteriorating reading activity. The second part of the work describes the procedure, presents the solved problems, discusses what goals were and were not achieved, how the application is implemented. The last, third part gives instructions on how to proceed during commissioning.

Keywords

Android, applications, reading, education, motivation, MySQL database

Obsah

1. Teoretická část	11
1.1. Problém klesající čtenářské aktivity	11
1.1.1. Pokles zapříčiněný rozvojem informačního odvětví	11
1.1.2. Negativní vliv izolace v době „covidové“	11
1.2. Návrh prostředku pro motivaci ke čtení	12
1.2.1. Můj přístup k věci – východiska, definování stavu a nedostatků	12
1.2.2. Můj záměr – definování cíle - nástroj pro kontrolu a motivaci ke čtení	12
2. Implementace	13
2.1 Logika aplikace	13
2.1.1 Vytvoření účtu, přihlašování	13
2.1.2 Učitelská verze	13
2.1.3 Studentská verze	14
2.1.4 Systém odměn	14
2.1.5 Obecná charakteristika REST API skriptů	14
2.2. Výběr technologií	15
2.2.1 Vývojové prostředí	15
2.2.2 Programovací jazyk	15
2.2.3 Minimální API	15
2.2.4 Správa serveru a databáze	16
2.2.5 Použité knihovny	16
2.2.5.1 Android Volley	16
2.2.5.2 RecyclerView	16
2.3 Nedostatky aplikace	17
2.3.1 Grafická stránka a interakce s uživatelem	17
2.3.2 Oznámení	17
3. Technická dokumentace	18
Závěr	19
Použité zdroje	20

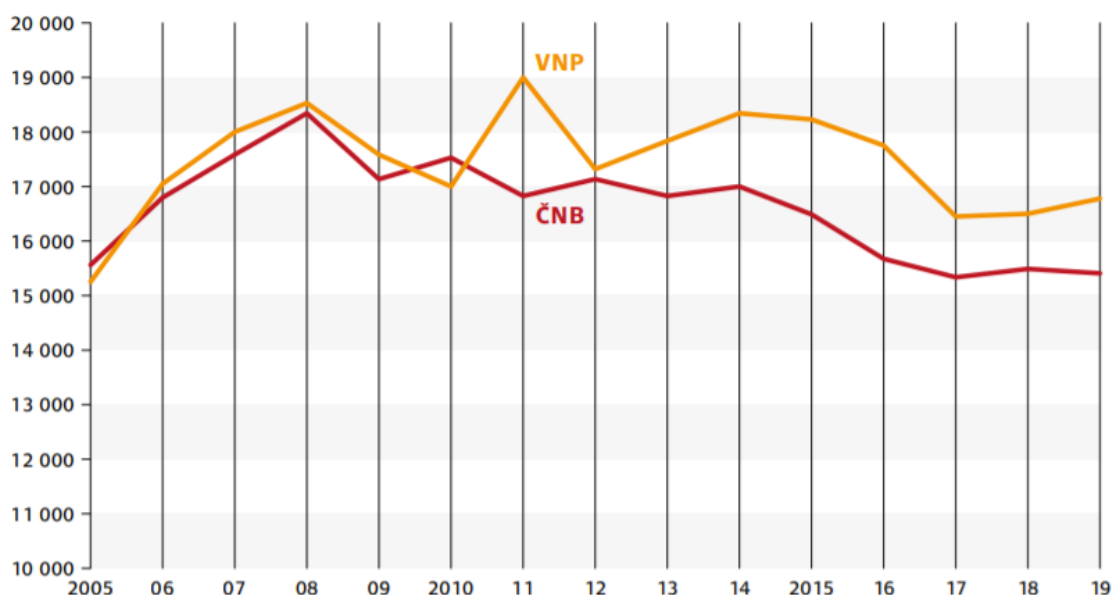
1. Teoretická část

1.1. Problém klesající čtenářské aktivity

1.1.1. Pokles zapříčiněný rozvojem informačního odvětví

S prudkým rozvojem informačního oboru a informačních technologií v posledních desetiletích se stal realitou nových generací pokles zájmu o čtení ať už knih v papírové podobě nebo i elektronických knih. Souvisí to samozřejmě s nabídkou široké škály nových atraktivních aktivit pro uživatele informačních technologií a sociálních sítí. O poklesu čtenářského zájmu svědčí i údaje ze Zprávy o českém knižním trhu 2019/2020, konkrétně v grafu (viz obrázek 1) „Počet vydaných titulů“ sledujeme v čase (od roku 2008) klesající trend počtu v ČR vydávaných titulů.

Obrázek 1. Počty titulů vydaných v ČR v jednotlivých letech



VNP – Výkaz o neperiodických publikacích Národní knihovny ČR

ČNB – Česká národní bibliografie Národní knihovny ČR

Zdroj obrázku: https://www.sckn.cz/file/wysiwyg/files/Zprava_o_kniznim_trhu_2019def.pdf

1.1.2. Negativní vliv izolace v době „covidové“

V dnešní době „covidové“, kdy se ve školách musí řešit komunikace žáků s učiteli na dálku prostřednictvím různých informačních technologií, nutně dochází k absenci vzájemně prospěšné interakce žáků a vyučujícího. I tato skutečnost má negativní vliv na systematický a zodpovědný přístup některých žáků ke studiu a podporuje inklinaci k vynechávání školních povinností a úkolů.

1.2. Návrh prostředku pro motivaci ke čtení

1.2.1. Můj přístup k věci – východiska, definování stavu a nedostatků

Vycházím z počátečního stavu, kdy jediným prostředkem, jak učitel může komunikovat se žákem, je e-mail. Pro tak specifický předmět, jako je český jazyk / literatura, však tento způsob komunikace – spolupráce nestačí. Například zde není obsažena v dostatečné míře motivace – jediné jako motivace negativní, kdy za nesplnění úkolu hrozí špatná známka nebo jiné sankce ze strany vyučujícího. Studentovi zde chybí nějaká forma pozitivní motivace. Chybí rovněž srovnání s ostatními žáky, což by se fyzickou přítomností ve škole určitě dělo.

1.2.2. Můj záměr – definování cíle - nástroj pro kontrolu a motivaci ke čtení

Chtěl bych přispět k ulehčení situace obou, jak vyučujícího, tak i žáka, a proto jsem vytvořil mobilní aplikaci, která by mohla v tomto směru pomoci. Cílem mé práce bylo poskytnout učiteli nástroj pro kontrolu spojenou s motivací žáků v předmětu český jazyk / literatura.

Určitou inspirací mi mohou být četné rozšířené a oblíbené již existující aplikace sledující čtenářskou aktivitu. Na rozdíl od nich v té mé je navíc jednak prvek účasti učitele, který může snahy svých žáků kontrolovat a koordinovat, a jednak prvek vzájemného srovnání všech žáků ve třídě. Obojí považuji za kladnou motivaci.

2. Implementace

2.1 Logika aplikace

Na úvod je potřeba říci, že aplikace, vzhledem ke svému cíli, v sobě skrývá vlastně dvě aplikace. Jedna slouží pro učitele, jako nástroj k zadávání a kontrole úkolů, a druhá je určena pro žáky, k zadávání svého postupu v daném čtenářském plánu.

2.1.1 Vytvoření účtu, přihlašování

Logika registrace a přihlašování je řešena v obou verzích aplikace víceméně stejně, rozdíl je pouze v tabulce v databázi, do které jsou profily vkládány. Při prvním spuštění je uživatel požádán, aby zadal uživatelské jméno, e-mail a své heslo do formuláře. Pokud již účet má, může se pomocí e-mailu a hesla přihlásit. S registrací účtů souvisí také jeden z problémů, které jsem musel při vývoji řešit. Pro uživatelské pohodlí jsem chtěl zajistit funkčnost, že nebude potřeba se znovu přihlašovat při každém spuštění aplikace. V několika projektech/ návodech jsem se setkal s realizací pomocí uložení dat do třídy *SharedPreferences*, tedy do paměti zařízení ve formátu hodnota - klíč. Později jsem ale narazil na problém, že data uložená tímto způsobem, se dají za určitých podmínek modifikovat. Zde je velký problém, protože upravením ID uživatele by se snadno dala získat data někoho jiného. Toto riziko jsem se snažil minimalizovat pomocí vytvoření dodatečné tabulky v databázi *validloginsteacher*, respektive *validloginsstudent*. Pokaždé, kdy proběhne úspěšné přihlášení uživatele do aplikace, vygeneruje se na serveru náhodné číslo. Toto číslo se uloží spolu s uživatelským ID do zmíněné tabulky. Zároveň se tato data uloží u uživatele na zařízení a při každém spuštění aplikace dochází ke srovnání dat. V případě, že se neshodují, je uživatel znovu vyzván k přihlášení. Šance, že někdo uhodne shodující se ID a náhodné číslo, je minimální. Toto řešení mi přijde jako vhodné, i když cítím, že v praxi je realizace možná jiná.

2.1.2 Učitelská verze

Po úspěšném vstupu uživatele do aplikace je spuštěna hlavní aktivita, která zobrazí všechny skupiny, v nichž je evidován jako učitel. Seznam je, stejně jako všechny ostatní, realizován pomocí prvku *RecyclerView*. S ním se pojí poměrně specifická syntaxe a v mém kódu se vyskytuje velmi často, proto mu věnuji samostatnou sekci. Učitel má možnost přidat novou skupinu – po kliknutí na tlačítko se spustí nová aktivita. V přidávání uživatelů do skupiny se mi, myslím, povedlo vytvořit zajímavý a dynamický systém, kdy je po zadání e-mailu pomocí skriptu *UserToGroup* ověřena existence uživatele a v případě úspěchupřidání do seznamu. Následně je možné záznam odeslat a uložit do tabulky *skupiny*. Uživatelé jsou k dané skupině přiřazeni tím, že je vloženo *user_id* a *skupina_id* do tabulky *skupinauzivatel*.

Dále, po kliknutí na jednotlivé skupiny se při spuštění aktivity zobrazí seznam s plány pro danou skupinu. V horní části obrazovky je znovu tlačítko sloužící k zadání nového plánu. Po zapsání informací do formuláře jsou data poslána do databáze, do tabulky *plany*.

Učitel má dále možnost po kliknutí na daný plán zobrazit postup všech studentů ve skupině v daném plánu

2.1.3 Studentská verze

Verze pro studenta byla původně primární zájmem mé maturitní práce, v průběhu se však ukázalo, že složitější – i co se délky kódu týče, bude ta učitelská.

Po úspěšném přihlášení či ověření již přihlášeného uživatele se spustí hlavní aktivita, v níž jsou vypsaný všechny nesplněné plány shodující se s ID uživatele. Po kliknutí na plán se otevře aktivita s detailními informacemi o něm, obsahující rovněž žebříček srovnávající postup všech uživatelů, k jehož realizaci je použit stejný skript jako v učitelské verzi.

Dále je zde pak tlačítko, které otevře prvek DialogFragment, pomocí něhož je možné přidat záznam plnění daného plánu. Po odeslání jsou data vložena do databáze, do tabulky progress.

Z hlavní aktivity je možné se ještě pomocí tlačítka přepnout do aktivity UserProfile, kde, jak již název napovídá, jsou k nalezení informace o uživateli. Při spuštění této aktivity se pomocí uloženého user_id načtou z databáze kromě základních informací o uživateli také statistiky a odměny, které získá. Systém kontroly odměn – a jestli mají být přidány do seznamu, je poněkud zvláštní, ale mně osobně se velmi líbí. Ze serveru dostaneme pole obsahující odměny (čísla), které konkrétní uživatel získal. Při vytvoření aktivity se pak spustí cyklus, který zkontroluje, zda je v poli číslo, a pokud ano přidá do seznamu název, popis a obrázek, které mají id shodné s prvkem pole.

2.1.4 Systém odměn

Implementace odměn a celkově snaha o motivaci žáků k plnění zadaných úkolů pro mě byla klíčovým cílem. Myslím, že tento cíl se mi podařilo naplnit tak, že jsem s ním spokojený. Uživatel je motivován několika způsoby. První je, že u svého profilu může vidět statistiky – tedy celkový počet stran a celkovou dobu, kterou po čas používání aplikace nashromáždil. Pomocí celkového času čtení se také vypočítává úroveň uživatele, a to pomocí následující rovnice
$$\frac{-30 + \sqrt{900 + 20 * \text{minuty} * \text{celkem}}}{10}$$
 Maximální úroveň je zatím stanovena na 16, což odpovídá přibližně 1800 minutám. Dalším prvkem snažícím se o motivaci žáka je systém odznaků, které lze získat za speciální události – jako příklad uvádím splnění prvního nebo přečtení více stran než bylo zadáno. Jestli byl odznak získán, je kontrolováno při každém vložení záznamu. Zde opět vidím prostor pro zlepšení – kontrola by mohla být implementována pomocí nějakého objektového návrhu, ne pouze pomocí podmínek. Odměny jsou pak uloženy v databázi ve formátu user_id: odmena:id. Poslední věc, která může uživateli dodat chuť do práce, je žebříček u každého plánu – tedy možnost srovnání s ostatními. V žebříčku jsou povoleny i hodnoty větší než 100%, nejedná se o chybu. Je také možné, že uživatelé nebudou vždy přesně seřazeni podle procent, což je způsobeno tím, že někdo může splnit například za stejnou dobu větší počet stran. To jsou ale spíše výjimky.

2.1.5 Obecná charakteristika REST API skriptů

Skripty používají pro komunikaci s databází MySQL rozhraní PDO, které nabízí přípravu dotazů, díky čemuž nemůže dojít k napadení databáze pomocí SQL injection.

2.2. Výběr technologií

2.2.1 Vývojové prostředí

Co se týče nabídky vývojových prostředí určených pro vývoj mobilních aplikací pro platformu Android běžících na systému Windows, který preferuji, je v současnosti bezpochyby nejrozšířenější program *Android Studio*. Navíc se jedná o oficiální vývojové prostředí pro platformu Android, doporučované Googlem. Mezi jeho největší výhody patří zejména příjemné uživatelské rozhraní, zejména možnost editovat XML soubory definující rozhraní aplikace v grafickém režimu, tedy že je okamžitě vidět výsledek, a dokonce s prvky přímo pohybovat. Velmi důležité je zmínit také emulátor AVD (Android Virtual Device), který je součástí zmiňovaného IDE a je tedy možné ho spouštět přímo při psaní kódu.

2.2.2 Programovací jazyk

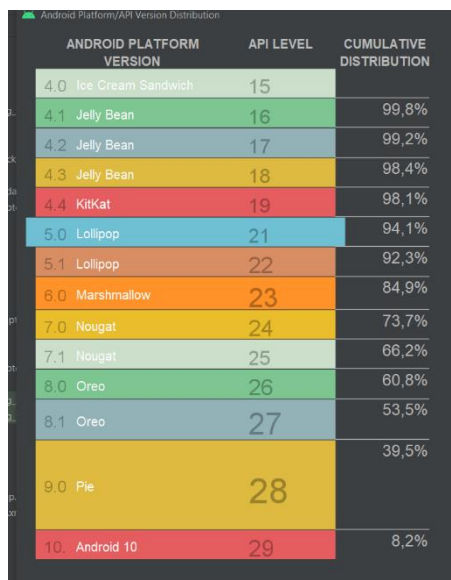
Při výběru programovacího jazyka jsem měl volbu Java, nebo Kotlin. Nakonec jsem se rozhodl pro Javu, protože se jedná o jazyk, který je více rozšířený a není spojen pouze s aplikacemi pro Android. Využil jsem tedy příležitost k rozvoji širších znalostí.

2.2.3 Minimální API

Při zohlednění, na jaké minimální verzi bude možné aplikaci spustit, jsem zohlednil zejména fakt, že žáci základní školy, na které je cílena, často používají starší telefony, například zděděné apod. Proto jsem jako minimum zvolil API 21, tedy Android 5.0 Lollipop. Bude tedy možné ji používat na 94% zařízení, jak je patrné z obrázku níže.

Obrázek 2: Verze Androidu v současných mobilních telefonech a jejich využitelnost.

Zdroj: Android Studio



2.2.4 Správa serveru a databáze

Nedílnou součástí mé aplikace je skutečnost, že velká většina dat potřebných pro správné fungování musí být přístupná pro více uživatelů, tedy je potřeba vytvořit databázi a server, který ji bude spravovat. Pro tyto účely jsem využil softwarový balíček XAMPP, který obsahuje jednak odlehčenou distribuci Apache, která vytváří lokální testovací server, a dále pak databázi MariaDB. Tyto technologie byly pro mou aplikaci ideální. Vzhledem k tomu, že XAMPP server využívá distribuci Apache serveru, nebyl by případně, pokud bych chtěl aplikaci publikovat, problém s přechodem na reálný server. Při výběru této technologie opět také sehrála roli jednoduchost a uživatelská přívětivost.

2.2.5 Použité knihovny

V první fázi projektu, jak jsem již výše zmiňoval, jsem počítal s tím, že aplikace bude komunikovat se serverem a databází přímo. Pracoval jsem tedy s knihovnou **Connector/J**, jejíž funkce spočívá ve spojení aplikace psané v jazyce Java s databází. Původně jsem takto chtěl předejít nutnosti psát kód v několika programovacích jazycích a v několika různých vývojových prostředích. Jak jsem ale při studiu zjistil, použití tohoto způsobu pro mobilní aplikace podle všeho není vhodné. Connector je závislý na stabilním internetovém připojení, a tím pádem, jak jsem pochopil, je vhodné spíše pro aplikace pro počítače. Místo toho se doporučuje použít REST API – vložit mezi server a samotné zařízení mezipřímek, kterým je v tomto případě PHP skript připravující data z databáze.

2.2.5.1 Android Volley

Tato knihovna slouží k zjednodušení správy HTTP požadavků, umí přijímat data v různých formátech – například JSON objekty, JSON pole nebo obyčejné textové řetězce. Sloužila mi jako prostředek, díky kterému jsem získal data ze serveru připravená PHP skriptem. Mezi nesporné výhody této knihovny oproti vestavěným metodám patří také rychlost, a tím pádem větší efektivita. Je vhodná pouze pro přenosy menších objemů dat, čili v mém případě šlo o ideální řešení. V kódu se nachází jedna třída sloužící plně pro účely Volley – VolleySingleton, respektující softwarový princip singleton, která má za úkol spravovat RequestQueue (frontu požadavků na databázi) a zároveň zajišťuje její jedinečnost. Opět musím přiznat, že jedním z důvodů pro výběr této knihovny byla skutečnost, že bylo k dispozici velké množství různých návodů, a tím pádem jednodušší pochopení systému.

2.2.5.2 RecyclerView

Všechny seznamy, na kterých aplikace v podstatě stojí, jsou realizovány jako prvek RecyclerView. Oproti klasickým seznamům má tu výhodu, že při větším množství řádků má v paměti pouze ty, které jsou aktuálně v obrazu. Tím tedy šetří paměť. Jinou výhodou, které jsem rovněž využil, je více způsobů, jak jednotlivé záznamy rozložit – kromě klasického lineárního rozložení také například dlaždicové (Grid layout).

S prvkem RecyclerView a jeho inicializací se pojí trojice tříd, které bylo potřeba implementovat pro každý seznam. K přiřazení dat do seznamu a převedení na správný datový typ slouží třída, splňující softwarový návrh Adapter. Druhou třídou, kterou jsem použil ke všem seznamem typu RecyclerView, je tzv. model. Jedná se o takovou třídu, jejíž objekty následně přebírá adapter. Objekty jsou zpravidla vytvářeny v cyklu poté, co je získáme z databáze. Navíc se s RecyclerView pojí ještě třída ViewHolder, která zajišťuje právě propojení konkrétního řádku se správnými daty. U některých seznamů je také potřeba přidat rozhraní, které obsluhuje, co se má stát po kliknutí na daný prvek – což je zpravidla otevření nové aktivity.

Uznávám, že implementace seznamů mi možná zabrala až příliš mnoho času, a je možné, že by šlo použít i jiný způsob zobrazování dat – nebo alespoň ho lépe optimalizovat, čímž bych mohl získat čas na něco jiného.

2.3 Nedostatky aplikace

V průběhu prací na svém projektu jsem musel řešit určité problémy, jednalo se o pro mě zcela novou zkušenost, takže i přes nedostatky, které v této části představím a kterých je jistě hodně, musím konstatovat, že i na to, co jsem zvládl, jsem velmi pyšný.

2.3.1 Grafická stránka a interakce s uživatelem

Aplikací jsem cílil na žáky prvního stupně základních škol, tedy by bylo záhodno, aby zde byla určitá míra interakce či alespoň aby zájem vzbuzovala vizuální stránka. V některých částech se mi to, myslím, povedlo – mám na mysli zejména uživatelský profil, který obsahuje hodně barev a obrázků. Dále spatřuji nedostatek v tom, že v některých případech nemám dostatečně ošetřený vstup – například při zadávání e-mailu nebo přidávání časových záznamů. Zároveň nezbyl čas ani na úpravu záznamů v databázi – například změna hesla, přidání dalšího uživatele do existující skupiny apod. Nejedná se sice o nezbytné prvky, ale jejich přítomnost by jistě zvýšila atraktivitu aplikace.

2.3.2 Oznámení

V zadání jsem chtěl, aby aplikace dětem také připomínala, že by měly číst. Bohužel, kvůli špatnému časovému harmonogramu a nedostatku znalostí, se na implementaci oznámení nedostalo.

3. Technická dokumentace

Ke spuštění aplikace je potřeba mít nainstalovaný lokální Apache server a MySQL databázi. Jak jsem zmiňoval v kapitole výběr technologií, já jsem používal softwarový balíček XAMPP, který lze získat z tohoto odkazu <https://www.apachefriends.org/index.html> Návod na instalaci je zde: <https://www.ionos.com/digitalguide/server/tools/xampp-tutorial-create-your-own-local-test-server/>

Strukturu databáze, včetně testovacích dat, je k dispozici v souboru `readingapp.sql` ve složce Database. Následně je třeba provést import tohoto souboru. Spustíte XAMPP Control Panel a spustíte Apache a Mysql. U Mysql poté vyberte klikněte na admin. V phpmyadmin, který se spustí vytvořte databázi: <https://www.homeandlearn.co.uk/php/php12p2.html>. Následně uskutečňte import: <https://help.one.com/hc/en-us/articles/115005588189-How-do-I-import-a-database-to-phpMyAdmin->

Dále je potřeba na server uložit PHP Rest API skripty ve složce RAScripts, které zprostředkovávají data z databáze. Při použití XAMPP by se měly uložit do složky `xampp/htdocs`. Také bude možná potřeba změnit údaje ve skriptu `DatabaseConfig.php`, který se nachází v samotné složce RAScripts a dále ve složkách Student a Teacher. Zadané údaje jsou výchozí. V repozitáři jsou všechny věci související s databází a serverem uloženy ve složce Database.

Jednotlivé verze aplikace – studentskou i učitelskou lze nalézt v repozitáři pod názvy `Reading_teacher` a `Reading_student`

Doporučuji rovněž spouštět aplikace v programu Android Studio a integrovaném emulátoru AVD (ke stažení zde: <https://developer.android.com/studio>, návod k instalaci zde: <https://developer.android.com/studio/install>). V případě, že by se používal jiný emulátor, možná bude potřeba změnit v aplikaci adresy (na veřejnou IP adresu Vašeho počítače) u všech PHP skriptů. AVD totiž používá výchozí IP adresu pro zařízení, na kterém je emulátor spuštěn, 10.0.2.2. Po otevření aplikace v IDE je pak již snadné pomocí ikony zelené šipky vpravo v horní liště spustit emulátor a na něm samotnou aplikaci.

Závěr

Při tvorbě aplikace jsem se blíže seznámil s programovacím jazykem JAVA, stejně jako s prací s daty na serveru. Sestavování některých částí programu vyžadovalo důkladné prostudování konkrétní problematiky v literatuře, která je dostupná pro různé úrovně odbornosti. Bylo tudíž důležité věnovat dostatek času ke studiu jednotlivých publikací, abych zde našel informace podstatné pro konkrétní situaci. Jedná se v mém životě o první projekt takového rozsahu, mohu s klidným svědomím prohlásit, že jsem na něm pracoval sám a věnoval mu hodně času a úsilí. Kromě studia programovacího jazyka jsem si procvičil a naučil se mnoho dalších souvisejících činností jako je vyhledávání a citace zdrojů, úprava dokumentu, práce s aplikacemi. Zamýšlel jsem se nad situací, požadavky a potřebami uživatelů a také jsem chtěl alespoň základním způsobem ošetřit bezpečnost celého projektu.

Sám na programu vidím určitá místa, která by se dala ještě zdokonalovat, především kvůli uživatelskému pohodlí, z grafického hlediska atd., avšak práce by vyžadovala delší časový úsek, možná i práci více lidí. Spatřuji zde potenciál pro své další studium, respektive pro další studenty, kteří by můj projekt mohli dále rozpracovat.

Použité zdroje:

LACKO, Luboslav. *Vývoj aplikací pro Android*. 1. vyd. Brno: Computer Press, 2015. 472 s. ISBN 978-80-251-4347-6.

LACKO, Luboslav. *Mistrovství - Android*. 1. vydání. Brno: Computer Press, 2017. 647 stran. Mistrovství. ISBN 978-80-251-4875-4.

HERODEK, Martin. *333 tipů a triků pro Android*. 1. vyd. Brno: Computer Press, 2014. 205 s. ISBN 978-80-251-4310-0.

HERODEK, Martin. *Android: jednoduše*. 2., aktualiz. vyd. Brno: Computer Press, 2014. 128 s. Naučte se za víkend. Jednoduše. ISBN 978-80-251-4298-1.

Císař, Jaroslav. *Zpráva o českém knižním trhu pro rok 2019/2020* [e-kniha]. Praha: SČKN, 2020 [vid. 2020-03-30]. ISBN 978-80-907020-6-6. Dostupné z: https://www.sckn.cz/file/wysiwyg/files/Zprava_o_kniznim_trhu_2019def.pdf

Android Developers [online]. [cit. 2021-04-07]. Dostupné z: <https://developer.android.com/>

W3 schools [online]. [cit. 2021-04-07]. Dostupné z: <https://www.w3schools.com/>

Geeks for Geeks [online]. [cit. 2021-04-07]. Dostupné z: <https://www.geeksforgeeks.org/volley-library-in-android/>

Medium [online]. [cit. 2021-04-07]. Dostupné z: <https://medium.com/swlh/android-recyclerview-part-1-creating-a-simple-list-d84392c268b0>

JECAS [online]. [cit. 2021-04-07]. Dostupné z: <https://jecas.cz/pdo>