

**Praha, 2022**

**Jiří Fryjauf, Jan Jindrák, Adam Lisner**

**Praha**

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

**Obor č. číslo oboru: Název oboru**

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

**Obor č. číslo oboru: Název oboru**

**MatKap – mobilní aplikace pro výuky literatury**

**MatKap – literature learning mobile application**

**Autoři:** Jiří Fryjauf, Jan Jindrák, Adam Lisner

**Škola:** Gymnázium, Praha 6, Arabská 14

**Kraj:** Praha

**Konzultant:** Ing. Daniel Kahoun, Mgr. Ivana Vondráčková

Praha 2022

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval/a samostatně a použil/a jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod 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 2. března 2022 ………………………………………………

Poděkování

Rádi bychom chtěli na tomto místě poděkovali našemu vedoucímu projektu, panu Ing. Danielu Kahounovi, se kterým jsme naši práci konzultovali a který nám dal mnohé věcné rady a tipy. Dále bychom také rádi poděkovali paní profesorce Mgr. Ivaně Vondráčkové za konzultace ohledně literárních děl.

Anotace

V naší práci SOČ jsme se zabývali vytvořením mobilní aplikace pro výuku literatury. Uživatel si může v aplikace procvičit své znalosti literatury a také se získat mnoho nových znalostí z této oblasti. Aplikace je určená pro mobilní zařízení s operačním systémem Android. Naprogramována je v jazyce Java s použitím mnohých moderních technologií.

Klíčová slova

Android, literatura, mobilní aplikace, Java, databáze

Abstract

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Pellentesque sapien. Nullam at arcu a est sollicitudin euismod. Duis pulvinar. Aliquam erat volutpat. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Fusce aliquam vestibulum ipsum. Pellentesque ipsum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et.

Keywords

Umpalumpa, Willy Wonka, Karlíku, neboj se

Annotation

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Pellentesque sapien. Nullam at arcu a est sollicitudin euismod. Duis pulvinar. Aliquam erat volutpat. Duis sapien nunc, commodo et, interdum suscipit, sollicitudin et, dolor. Fusce aliquam vestibulum ipsum. Pellentesque ipsum. Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et.

Stichwörter

Ahoj tati, Martin Chmelař, Petr Mára, Nový Video

Obsah

[1.Úvod 7](#_Toc96804538)

[2. Použité technologie 8](#_Toc96804539)

[3. Frontend 9](#_Toc96804540)

[3.1. Activity Main a Register 9](#_Toc96804541)

[3.2. MenuActivity 9](#_Toc96804542)

[3.3. ProfileActivity 10](#_Toc96804543)

[3.4. QuestActivity 10](#_Toc96804544)

[3.5. CheatSheet 10](#_Toc96804545)

[3.6. Kvíz 11](#_Toc96804546)

[3.6.1. QuizSettings 11](#_Toc96804547)

[3.6.2. QuestionActivity 11](#_Toc96804548)

[3.6.3. QuizSummary 11](#_Toc96804549)

[4. Závěr 12](#_Toc96804550)

[Příloha 13](#_Toc96804551)

1. Úvod

Tato práce se zabývá naprogramováním mobilní aplikace MatKap (zkrácenina „*Maturita v kapse*“), jejíž účelem je pomoci studentům maturitních ročníků se lépe připravit na ústní zkoušku z českého jazyka a literatury. Cílem je udělat plně funkční mobilní aplikaci pro operační systém Android, která bude uživatelsky přívětivá a intuitivní. Dalším cílem bylo vytvořit testovací systém, ve kterém si uživatel může ověřit a prohloubit své znalosti, které bude potřebovat u zkoušky. Kromě testovací fáze si může uživatel také pročíst krátké zápisky o jednotlivých autorech, knihách a literárních epochách.

Práce byla rozdělena do tří hlavních směrů: front-end, databáze a back-end.

1. Použité technologie

V této práci je zkombinováno mnoho moderních a užitečných technologií. Zdrojový kód aplikace je napsán v programovacím jazyce Java ve verzi IDK DOPLNTE PLSKY. Program byl napsán ve vývojovém prostředí Android Studio ve verzi Artic Fox / 2020.3.1. Toto prostředí je založené na jádře IntelliJ od české společnosti JetBrains. Data aplikace jsou uložena v SQLite databázi. SQLite je odnoží SQL, která umožňuje lokálně uložit databázi. Pro práci s databází byl použit program DataGrip ve verzi 2021.3.4. Tento program je velmi užitečný pro práci s více tabulkami najednou a umožňuje práci s mnoha databázovými systémy včetně právě SQLite. Stojí za zmínku, že i DataGrip je českým produktem od společnosti JetBrains.

Pro autentikaci, správu účtů, sdílené cloudové úložiště některých souborů byl použit open-source nástroj FireBase od Googlu. Firebase je výborný nástroj díky své dostupnosti a funkčnosti. Udržuje zabezpečení účtů, nabízí mnoho možností komunikace s uživateli (např. no-reply emaily). FireBase je využíván mnohými velkými projekty světově známých firem, což je také známkou kvality. Dobrou vlastností FireBase je také skvělé propojení s Android Studiem a dobře popsaná dokumentace, která obsahuje názorné video tutoriály.

Během vývoje bylo použito také mnoho grafických a designových nástrojů. Velmi nápomocný byl program Figma. Jedná se o vektorový grafický editor a nástroj pro navrhování webových, desktopových a mobilních aplikací, který používají mnozí profesionální UI/UX designéři. Další grafické programy použité v práci jsou programy z dílny Adobe, Adobe Photoshop a Adobe Illustrator. Tyto programy jsou určeny pro práci s rastrovou a vektorovou grafikou.

Pro optimalizaci a usnadnění týmové práce byl použit Git a GitHub. Git je nástroj pro správu verzí zdrojového kódu. Byl vyvinut počátkem tohoto století a jedná se o open-source systém. GitHub je naopak služba, která nabízí zdarma webhosting pro open-source projekty a jejich verzování pomocí Gitu. Zdrojový kód je uložen v repozitáři, který je uložen na [GitHubu](https://github.com/MetrKolben/MatKap).

1. Frontend

Front-endová část aplikace je taková část aplikace, kterou vidí a kterou používá uživatel. Proto je potřeba mu věnovat čas a úsilí, stejně jako back-endu. Front-end v Android aplikacích je psán pomocí značkovacího jazyka XML (*Extensible Markup Language*) a tzv. aktivit. O jednotlivých aktivitách a jejich struktuře se dozvíte v následujících kapitolách.

* 1. Activity Main a Register

Activity Main je první obrazovkou, se kterou se uživatel setká. Na obrazovce se nachází *Button* s nápisem přihlášení, který naslouchá uživatelovým příkazům pomocí Listeneru. V případě, že uživatel má již provedenou registraci, tak po stisknutí tlačítka proběhne přihlášení. V druhém případě, kdy ještě nedošlo k jakémukoliv přihlášení do aplikace skrze uživatelovo zařízení, je uživatel přesměrován do další aktivity, do Activity Register. Přesměrování mezi aktivitami probíhá pomocí Intentů, které ve svých argumentech přebírají informace o tom, z jaké aktivity dojde ke přesměrování a do které se má přesměrovat. Ukázku Intentu naleznete na obrázku č. . Ukázku této aktivity si můžete prohlédnout v příloze pod číslem.

V aktivitě Register se uživatel registruje do aplikace pomocí svého Google účtu (více o registraci a přihlašování se dozvíte v kapitole xXx). Po stisknutí tlačítka s nápisem “*Registrovat”* se vytvoří uživatelův účet.

* 1. MenuActivity

Aktivita Menu je základním kamenem orientace uživatele v aplikaci, jelikož právě menu aplikace umožňuje uživateli se dostat do všech aktivit, které aplikace obsahuje. Vpravo nahoře je *ImageButton*, jehož stisknutím se uživatel dostane na svůj profil. Uprostřed stránky je tlačítko, kterým se zapíná kvíz. Ve spodku okna se nachází ovládací panel. Tímto panelem se uživatel pohybuje mezi jednotlivými aktivitami a je stejný ve většině aktivit. Tento panel je reprezentován *FlexBoxLayoutem*, který má velkou řadu výhod. Jak již z názvu napovídá *FlexBoxLayout* je velmi podobný tomu, který většina webových inženýrů zná z CSS. A tato androidová implementace je mu velmi podobá, co se týče jeho použití. Obsahuje většinu funkcí, které jeho webový příbuzný, jako například: alignItems, flexDirection, alignContent nebo justifyContent. Poslední z jmenovaných příkladů funkcí byl použit pro tento panel. Díky justifyContent se všechny prvky daného layoutu rovnoměrně uspořádají, což je dobré pro správné vykreslení na různých telefonech, jelikož rozlišení není na mobilních zařízeních sjednoceno. FlexBoxLayout ale není v základním Android Studio balíčky, tudíž jej člověk musí implementovat v souboru *build.gradle*.

* 1. ProfileActivity

V profilové aktivitě nalezne uživatel informace o svém účtu. V horní části obrazovky nalezne uživatel svůj profilový obrázek, který se mění s jeho úrovní, a jeho doposud dosaženou úroveň. Pod touto dvojicí ukazatelů se nachází *ProgressBar*, jenž slouží jako ukazatel počtu XP do nové úrovně. Dodatečnými informacemi jsou jméno a email uživatele. Všechny dosud jmenované položky jsou silně spjaty s back-endovou částí programu o níž se dozvíte v kapitole xXx. Kromě osobních informací obsahuje *ProfileActivity* tlačítka odhlásit a smazat účet. Po stisknutí druhého jmenovaného se objeví *AlertDialogBox* (vyskakovací okno), který se uživatele zeptá, jestli opravdu chce zrušit svůj účet. Ukázku této obrazovky si můžete prohlédnout na obrázku č. 4542112. Posledním prvkem je ovládací panel na spodku obrazovky, kterým se přepíná mezi jednotlivými aktivitami aplikace.

* 1. QuestActivity

V této aktivitě nalezne uživatel úkoly na daný den. Na každý den jsou vždy 3 úkoly. U každého úkolu je vždy jeho zadání, ukazatel pokroku v jeho dokončení, počet XP za dokončení a ukazatel, který indikuje dokončení úkolu. Tyto ukazatele jsou zabaleny v LinearLayoutu, který dále obsahuje ConstraintLayout se zadáním a ukazatelem dokončení, *ProgressBar* a množství XP, které získá. Na obrázku č. si můžete prohlédnout, jak tato aktivita vypadá. O back-endovém principu úkolů se dozvíte v kapitole xXx.

* 1. CheatSheet

Aktivita CheatSheet slouží k učení jednotlivých pojmů. Po otevření této aktivity objeví uživatel seznam jednotlivých autorů, knih a směrů, kterými může scrollovat a rozkliknout je pro další informace. Seznam je reprezentován RecycleView, které je nutné implementovat v souboru build.gradle. RecycleView umožňuje vývojářům dynamicky přidávat jednotlivé položky na obrazovku z ArrayListu. Jedna z jeho velkých výhod je také dobrá optimalizace, jelikož pracuje pouze s daty potřebnými pro vykreslení, nikoliv s celým seznamem, což je velmi nápomocné zejména při práci s velkými daty.  Pro nastavení jednotlivých záznamů v RecycleView je nutné vytvořit třídu dědící z RecycleView.Adapter<ViewHolder>. Po implementování všech potřebných metod a vytvoření třídy dědící z RecyclerView.ViewHolder se může nastavit vše potřebné. Ve funkci onCreateViewHolder je vytvořen nový View, který bere mnoho vlastností z XML souboru *book\_card.xml.* (jeho strukturu najdete v příloze pod číslem xXx) V metodě onBindViewHolder jsou kartám nastaveny jednotlivé texty a informace. Informace, které jednotlivé info-karty obsahují jsou brány z databáze, o jejímž fungování se dozvíte více v kapitole XxX.

Pro každou variantu seznamu (autoři, knihy, směry) je vytvořen vlastní Adapter i třída, která obsahuje přístupové metody k jednotlivým proměnným. Tyto soubory najdete v package *cheat\_sheet*.

* 1. Kvíz

Jednou z hlavních částí této práce je literární kvíz. V následujících kapitolách se dozvíte více informací o jednotlivých částech tohoto kvízu.

* + 1. QuizSettings

Po stisknutí tlačítka “Spustit kvíz” v menu je uživatel přesměrován do aktivity nastavení kvízu. V této aktivitě si nastaví, jaké otázky bude kvíz obsahovat a jak se bude kvíz chovat. Pro výběr pouze některých okruhů otázek (okruhem je myšleno dané umělecké období / směr) je nutné stisknout čtyřúhelník, který je vytvořen pomocí *TextView*. Po této události se uživateli objeví AlertDialogBox. Toto vyskakovací okno obsahuje všechny směry z databáze a kliknutím na ně si může uživatel zvolit jednotlivá témata kvízu. Je také možné si nastavit variantu tohoto kvízu. První varianta je kvíz, při kterém je hned po zodpovězení zobrazena správná / špatná odpověď. Druhá varianta je test, při této variantě se uživatel dozví svůj výkon až po testu. Tyto varianty se nastavují pomocí *CheckBoxu*. Po nastavení lze kvíz zapnout stisknutím tlačítka.

* + 1. QuestionActivity

Tato aktivita je už samotný kvíz. Kvíz má defaultně deset otázek, pouze v případě, že se v daných okruzích nenalézá dostatek otázek, je tento počet snížen. Toto je však velmi okrajová situace, ke které by nemělo docházet. Ovládání kvízu je velmi jednoduché. V horní části obrazovky se nachází text otázky. Pod zadáním se nachází čtyři možné odpovědi, kde vždy právě jedna je správná. Políčka na odpověď jsou jednotlivé *RadioButtony,* které dohromady tvoří jednu skupinu neboli *RadioGroup*. Po zvolení odpovědi ji může uživatel potvrdit pomocí tlačítka v pravém dolním rohu. Když není zvolena odpověď, není možné odpovědět. U varianty kvízu (viz předchozí kapitola) je také po odpovězení na 1,5 vteřiny zobrazena správná a špatná odpověď. Zároveň je v kvízu uživateli zobrazeno aktuální skóre. V průběhu testu se ukládají jednotlivé odpovědi (reprezentované objektem AnsweredQuestion dědící z Sql.Question) do ArrayListu<AnsweredQuestion>. Po dokončení kvízu je uživatel přesměrován do souhrnu kvízu (QuizSummary). Ukázku kvízu najdete na obrázku č. xXx

* + 1. QuizSummary

Po dokončení kvízu se zobrazí vyhodnocení kvízu. Uživateli se zobrazí skóre v procentech, zpráva o tom, jak si vedl. Dále se v případě více špatných odpovědí na jedno téma zobrazí zpráva: “*Měl by sis procvičit směr:” +* směry k procvičení. Také si může uživatel zpětně přečíst otázky, na které odpovídal, společně s jejich správnou odpovědí, případně jeho špatnou odpověď. Zpětné čtení otázek je vytvořeno pomocí RecyclerView, stejně jako *CheatSheet*. Více o jeho fungování se dozvíte v kapitole CheatSheet.

1. Závěr

Příloha



