

Gymnázium, Praha 6, Arabská 14

Programování

MATURITNÍ PRÁCE



Gymnázium, Praha 6, Arabská 14

Arabská 14, Praha 6, 160 00

Maturitní práce

Předmět: Programování

Jméno práce: Couponify

Autor: David Mikolášek

Třída: IV. E

Školní rok: 2024/2025

Vedoucí práce: Mgr. Jan Lána

Třídní učitel: Mgr. Blanka Hniličková

Prohlašuji, že jsem na této práci pracoval samostatně, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská 14 oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

V Praze dne

Anotace

Tato dokumentace popisuje ročníkový projekt s názvem Couponify, aplikace, která si pamatuje závazky. Cílem projektu bylo vytvořit aplikaci na android, která umožní uživatelům posílat si mezi sebou kupony, reprezentující určitý závazek nebo dluh. Neváže se však pouze na peněžní dluhy, může se jednat o jakýkoli slib, u kterého si chceme pojistit, že ho dotyčný vyplní. V aplikaci se dá vybízet k vyplnění obsahu kuponu pomocí oznámení. Dále se v aplikaci dají posílat herní kupony na ozvláštnění každodenního života. Uživatelé se také mohou inspirovat příklady herních kuponů přímo v aplikaci a dostávat doporučení podle typu kuponů, které se jim nejvíce líbí.

Abstract (English)

This documentation describes the year-end project titled Couponify, an application that keeps track of commitments. The goal of the project was to develop an Android app that allows users to send coupons to each other, representing a specific commitment or debt. However, it is not limited to financial debts—it can be any promise that users want to ensure is fulfilled. The app allows users to remind others to fulfill their coupons through notifications. Additionally, it enables users to send game coupons to make everyday life more interesting. Users can also get inspiration from pre-made game coupon examples within the app and receive recommendations based on the types of coupons they like the most.

Obsah

1. Úvod	2
1.1 Zadání práce	2
2. Použité technologie	3
2.1 Firebase	3
2.1.1 Firebase Authentication	3
2.1.2 Firebase Realtime Database	3
2.1.3 Firebase Cloud Messaging	4
2.2 OkHttp3	4
2.3 Canva	4
3. Struktura projektu	5
3.1 Registrace a přihlašování	6
3.1.1 Uvítací stránka	6
3.1.2 Registrace	7
3.1.3 Přihlášení	8
3.2 Hlavní aktivity projektu	9
3.2.1 Hlavní obrazovka	9
3.2.2 Přidávání přátel	9
3.2.3 Aktivita FriendDetail	10
3.2.4 Tvorba kuponů	10
3.2.5 Oznámení	11
3.2.6 Detail kuponu	11
3.2.7 Aktivní kupony	11
3.2.8 Herní kupony	12
3.2.9 Aktivita Inspiration Tab	12
4. Frontend	13
5. Závěr	13
6. Zdroje	14

1. Úvod

Všichni někdy děláme sliby, na které pak zapomeneme a nikdy je nesplníme. Couponify umožňuje si v tom, co slibujeme udělat pořádek pomocí kuponů. Jednoduše pokud někomu něco dlužím, napíšu mu na to kupon. V aplikaci pak mám přehled o tom, kdo mi poslal jaké kupony a nemusím si pamatovat, co mi kdo slíbil. Díky oznámením můžu také připomínat svým kamarádům, rodině, kolegům, nebo partnerovi, že je čas vyplnit to, co mi slíbili. Herní kupony dávají šanci udělat ze všedního dne něco zábavného, fungují jako balíček karet stolní hry, jejíž cílem je zpříjemnit si život. Pro ty co nenapadá, jaké herní kupony si na své kamarády vymyslet je v aplikaci zabudovaný doporučovací systém s příklady herních kuponů pro inspiraci.

1.1 Zadání práce

Mobilní aplikace je navržena pro přátele a páry a její hlavní funkcionalita je založena na systému recipročních kupónů, které slouží jako forma závazku mezi uživateli. Princip spočívá v tom, že když jeden uživatel něco dluží svému kamarádovi, nebo partnerovi, vytvoří pro tuto položku digitální kupón. Držitel kupónu pak může kdykoli kupón uplatnit a požadovat splnění příslušného závazku.

Další součástí aplikace je herní režim, který umožňuje interakci mezi dvěma uživatelskými účty. Oba uživatelé v tomto režimu vytvoří předem stanovený počet situačních kupónů, které se následně zamíchají a rozdělí tak, aby každý z nich obdržel polovinu, mohou si také nastavit časový limit k tomu, aby své kupóny využili.

Aplikace obsahuje také doporučovací systém, který nabízí inspiraci pro tvorbu nových kuponů. Uživatelé mohou procházet příklady kupónů a přidávat si je do oblíbených. Na základě typu kupónů, které si uživatel označí jako oblíbené, se jim zobrazují personalizované doporučení.

2. Použité technologie

Pro ulehčení práce jsem na některé části projektu používal knihovny, platformy, nebo servisy třetích stran. Nejvíce užitečná byla při zhotovování projektu platforma Firebase od společnosti Google. Celý projekt je v jazyce Java, na frontendové prvky používá XML. Projekt jsem naprogramoval ve vývojovém prostředí Androidstudio, umožňuje totiž jednoduché propojení s platformou Firebase.

2.1 Firebase

Firebase je platforma od společnosti Google pro vývoj mobilních a webových aplikací. Umožňuje rychlý vývoj a škálování aplikací bez nutnosti správy vlastního serverového zázemí. Poskytuje například nástroje pro autentizaci uživatelů, databáze a analytiku. Ve svém projektu jsem využil Firebase Authentication k jednoduchému registrování a přihlašování uživatelů, Firebase Realtime Database na ukládání dat a Firebase Cloud Messaging na posílání oznámení. [1]

2.1.1 Firebase Authentication

Firebase Authentication nabízí vícero způsobů, přes co přihlašovat a registrovat uživatele. Ve svém projektu jsem zvolil přihlašování pomocí e-mailu, vzhledem k tomu, že díky metodě `attemptautologin` předešle přihlášený uživatel nemusí znovu zadávat své přihlašovací údaje, ale je automaticky přihlášen, jsem neviděl potřebu zavádět vícero způsobů přihlášení. [2]

2.1.2 Firebase Realtime Database

Databázi Realtime Database jsem použil k uchování všech informací, tudíž kupóny, aktivní kupóny, herní kupóny, ale také tokeny zařízení u těch uživatelů, kteří povolili zaslání oznámení. Vybral jsem si Realtime Database, protože nepotřebuji skladovat žádné velké soubory, jako třeba obrázky, nebo videa, tak jsem zvolil možnost optimální pro rychlou práci s menšími daty. Realtime Database také pracuje s daty v reálném čase a reaguje na změny v databázi hned, jak nastanou. Díky tomu aplikace působí pohotově a responzivně. [3]

2.1.3 Firebase Cloud Messaging

Firebase Cloud Messaging je jednoduchý způsob, jak posílat oznámení na android zařízení. Implementací servisu `MyFirebaseMessagingService` mi Firebase umožnil posílat a zpracovávat oznámení vestavěnými metodami. Tento servis je automaticky white-listed, to znamená, že běží na pozadí, i když aplikace není spuštěná. Díky tomu se oznámení k uživateli dostanou i při vypnuté obrazovce, nebo třeba když běží jiná aplikace. [4]

2.2 OkHttp3

Knihovna `OkHttp3` se v projektu používá pro odesílání HTTP požadavků, konkrétně k odesílání push notifikací přes Firebase Cloud Messaging. `OkHttp3` umožňuje efektivní a jednoduchou komunikaci s FCM API, což aplikaci umožňuje informovat uživatele o sdílení kupónů nebo jiných důležitých událostech. [5]

2.3 Canva

Mobilní aplikaci Canva jsem využil na vytvoření loga Couponify. [6]



Obrázek 2.3 - Logo Couponify

3. Struktura projektu

Projekt je rozdělen na aktivity - obrazovky, mezi kterými prochází uživatel. Každá aktivita má svou třídu .java, ve které je logika dané aktivity a svůj .xml soubor, kde je definována její vizuální stránka a komponenty, jako třeba textová pole a tlačítka, které obsahuje. Všechny aktivity obsahují metodu `hideNavigationBar()`, která zajišťuje, že budou skryty navigační prvky systému.

```
private void hideNavigationBar() {  
    getWindow().getDecorView().setSystemUiVisibility(  
        View.SYSTEM_UI_FLAG_HIDE_NAVIGATION  
        | View.SYSTEM_UI_FLAG_LAYOUT_HIDE_NAVIGATION  
        | View.SYSTEM_UI_FLAG_IMMERSIVE_STICKY  
        | View.SYSTEM_UI_FLAG_LAYOUT_STABLE  
        | View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN  
        | View.SYSTEM_UI_FLAG_FULLSCREEN  
    );  
    getWindow().addFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN);  
}
```

zdrojový kód 3. - metoda `hideNavigationBar`

3.1 Registrace a přihlašování

3.1.1 Uvítací stránka

První aktivita, kterou uživatel uvidí, když poprvé spustí aplikaci je aktivita **Welcome**. Pokaždé, když se spustí, zavolá metodu `attemptAutoLogin()`, tato metoda se pokusí přihlásit uživatele automaticky, pokud už se jednou přihlásil a není odhlášen. Pokud se uživatel ještě nezaregistroval, nebo je odhlášen, má možnost si založit účet pomocí tlačítka Register, nebo se přihlásit tlačítkem Login.

```
private void attemptAutoLogin() {
    FirebaseUser currentUser = mAuth.getCurrentUser();
    if(currentUser != null){
        FirebaseUser authuser = mAuth.getCurrentUser();
        curuserid = authuser.getId();

        databaseReference = FirebaseDatabase.getInstance( url: "https://couponify1-636d2-default-rtdb.europe-west1.firebaseio.com").getReference( path: "users");
        valueEventListener = databaseReference.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot snapshot) {
                for (DataSnapshot itemSnapshot: snapshot.getChildren()) {
                    User userr = itemSnapshot.getValue(User.class);
                    if (Objects.equals(userr.getId(), curuserid)) {
                        curusername = userr.getUsername();
                        Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                        intent.putExtra( name: "curuserid", curuserid);
                        intent.putExtra( name: "curusername", curusername);
                        startActivity(intent);
                        finish();
                    }
                }
            }

            @Override
            public void onCancelled(@NonNull DatabaseError error) {
            }
        });
    }
}
```

zdrojový kód 3.1.1 - metoda `attemptAutoLogin`

3.1.2 Registrace

aktivita **Register** má tři textová pole na zadání uživatelského jména, e-mailu a hesla a tlačítko Enter. Po stisknutí tlačítka Enter zkontroluje, zda jsou vyplněna všechna textová pole, jestli ne, vyzve k jejich vyplnění pomocí Toastmessage. Pokud jsou všechna pole vyplněna, pokusí se vytvořit nového uživatele pomocí funkce `createUserWithEmailAndPassword`, která je vestavěna do `FirebaseAuth`. V případě, že je špatně zadaný e-mail, ukáže se zpráva, že se založení účtu nezdařilo. Když je všechno v pořádku, založí nového `FirebaseAuth` uživatele, přidá ho do databáze a přesune uživatele do aktivity login. [7]

```
mAuth.createUserWithEmailAndPassword(email, password)
    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            progressBar.setVisibility(View.GONE);
            if (task.isSuccessful()) {
                FirebaseUser user = mAuth.getCurrentUser();
                String curuserid = Objects.requireNonNull(user).getUid();
                addUserToDB(curuserid, username);
                Toast.makeText(context, Register.this, text: "Account created.",
                    Toast.LENGTH_SHORT).show();
                Intent intent = new Intent(getApplicationContext(), Login.class);
                startActivity(intent);
                finish();
            } else {
                Toast.makeText(context, Register.this, text: "Account creation failed.",
                    Toast.LENGTH_SHORT).show();
            }
        }
    });
```

zdrojový kód 3.1.2 - metoda `createUserWithEmailAndPassword`

3.1.3 Přihlášení

Aktivita **Login** slouží k přihlášení uživatele pomocí Firebase Authentication. Uživatel zadá své přihlašovací údaje a klikne na tlačítko přihlášení, čímž se spustí validace vstupu. Pokud jsou údaje správné, Firebase Authentication provede ověření pomocí metody `signInWithEmailAndPassword`. Po úspěšném přihlášení se získají uživatelská data z Firebase Realtime Database a aplikace přesměruje uživatele do hlavní aktivity `MainActivity`. Pokud přihlášení selže, zobrazí se chybová zpráva. Aktivita také umožňuje uživateli přejít do registrační obrazovky kliknutím na příslušný odkaz. [8]

```
mAuth.signInWithEmailAndPassword(email, password)
    .addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            progressBar.setVisibility(View.GONE);
            if (task.isSuccessful()) {
                FirebaseUser authuser = mAuth.getCurrentUser();
                curuserid = authuser.getId();
            }
        }
    });
```

Zdrojový kód 3.1.3 - metoda `signInWithEmailAndPassword`

3.2 Hlavní aktivity projektu

3.2.1 Hlavní obrazovka

Aktivita **MainActivity** představuje hlavní obrazovku aplikace Couponify¹, kde se uživatel po přihlášení dostane k funkcím správy přátel a kupónů. Po spuštění se skryjí navigační prvky a aplikace požádá o oprávnění k notifikacím. Pokud je oprávnění uděleno, aplikace získá a uloží token zařízení pro notifikace. Uživatel je v aplikaci identifikován na základě údajů z Firebase Authentication a Realtime Database. V případě, že aktuální uživatel není platný, je přesměrován zpět na přihlašovací obrazovku.

Po načtení hlavní aktivity se zobrazí uvítací zpráva se jménem uživatele a inicializuje se seznam přátel, který je získáván z Firebase Realtime Database. Přátelé jsou zobrazeni v RecyclerView s využitím adaptéru friendslistadapter. Uživatel může přidat nové přátele pomocí tlačítka, které jej přesměruje do aktivity addfriends.

Dalšími funkcemi jsou přechody do sekcí aktivních kupónů a inspirace, které jsou dostupné přes příslušná tlačítka. Pokud uživatel klikne na červené tlačítko odhlášení, jeho relace je ukončena pomocí `auth.signOut()`, a následně je přesměrován na úvodní obrazovku - aktivitu welcome.

3.2.2 Přidávání přátel

Aktivita **addfriends** umožňuje uživateli vyhledávat a posílat uživatelům žádosti o přátelství. Ty jsou zobrazeny v aktivitě **pendingrq**. Uživatel může zadat uživatelské jméno do vyhledávacího pole, na základě čehož se provede dotaz na Firebase Realtime Database, kde se hledají odpovídající uživatelé. Výsledky jsou zobrazeny v RecyclerView pomocí adaptéru, který umožňuje jejich interaktivní výběr. Pokud uživatel klikne na tlačítko přidání, jeho username se uloží do databáze v sekci „rq“ pod jménem daného uživatele.

3.2.3 Aktivita FriendDetail

Aktivita **FriendDetail** se spustí po kliknutí na prvek RecyclerView `friendslistrv` v aktivitě `MainActivity` a slouží k zobrazení detailů o přátelských kuponech, které byly od určitého přítele zaslány aktuálnímu uživateli. Každý kupon je reprezentován objektem `Coupon` a do seznamu jsou přidány pouze ty kupóny, které byly napsány vybraným přítelem.

Pokud není žádný kupon k dispozici, zobrazí se upozornění „No coupons available“. Aktivita rovněž umožňuje uživateli napsat nový kupon pro tohoto přítele prostřednictvím tlačítka `writecouponbtn`, které spustí aktivitu `writenewcoupon`.

3.2.4 Tvorba kuponů

Aktivita **writenewcoupon** umožňuje uživateli napsat nový kupon pro vybraného přítele zadáním jeho jména a popisu. Po stisknutí tlačítka `Send` se kromě zapsání nového kuponu do databáze také pošle oznámení pomocí metody `sendNotifWithID`. Metoda `sendNotifWithID` slouží pouze k tomu, aby z databáze dostala token zařízení daného uživatele a poslala ho jako argument metody `SendPushNotification` třídy `SendNotification`.

```
public void sendNotifWithID(String title, String desc, String friendid){
    FirebaseDatabase.getInstance().getReference().child("Tokens").child(friendid).get().addOnCompleteListener(new OnCompleteListener<DataSnapshot>() {
        @Override
        public void onComplete(@NonNull Task<DataSnapshot> task) {
            String token = task.getResult().getValue(String.class);
            final ExecutorService executor = Executors.newSingleThreadExecutor();
            executor.execute(new Runnable() {
                @Override
                public void run() {
                    SendNotification sn = new SendNotification();
                    sn.SendPushNotification(title, desc, token);
                }
            });
        }
    });
}
```

zdrojový kód 3.2.4 - metoda `sendNotificationWithID`

3.2.5 Oznámení

Třída **SendNotification** zajišťuje odesílání push notifikací pomocí Firebase Cloud Messaging. Hlavní metodou je `SendPushNotification`, která přijímá tři parametry: název notifikace (`notifTitle`), text notifikace (`notifBody`) a FCM token příjemce (`fcmtoken`), který je unikátní pro každé zařízení. V první fázi se připraví JSON objekt, který obsahuje informaci o cílovém tokenu, nadpisu a tělu notifikace. Tento JSON objekt je pak odeslán jako tělo HTTP požadavku. Požadavek je typu POST a je zaslán na Firebase Cloud Messaging endpoint (https://fcm.googleapis.com/v1/projects/{PROJECT_ID}/messages:send).

K požadavku je přidán Authorization header s Bearer tokenem získaným z metody `AccessToken.getAccessToken()`. Další hlavička, která je potřeba, je Content-Type, nastavená na `application/json`. Poté se pomocí knihovny `OkHttp` odešle požadavek na server FCM. Odesílání požadavku probíhá asynchronně, což znamená, že metoda `enqueue()` spustí požadavek na pozadí. S tímto aspektem aplikace (`AccessToken`, `SendNotification` a `MyFirebaseMessagingService`) mi pomohla umělá inteligence, protože jsem neuměl pracovat s FCM Http V1 a Firebase dokumentace není v tomto ohledu velkou pomocí. [9]

3.2.6 Detail kuponu

Aktivita **coupondetail** se spustí po kliknutí na kupón v seznamu kupónů v aktivitě `frienddetail`. Ukazuje detailní informace o kupónu a tlačítko `Redeem`, které kupón přesune do `activecoupons` aktuálního uživatele, ale i uživatele, který kupón napsal. V aktivitě `activecoupons` tedy vidí tento kupón oba uživatelé, a to proto, aby měl ten, kdo ho napsal přehled o tom, které požadavky má splnit a majitel kupónu, aby mohl odkliknout, že je požadavek splněn tlačítkem `Dismiss`, které se zobrazí pouze jemu.

3.2.7 Aktivní kupony

Aktivita **activecoupons**, zde je seznam všech aktivních kupónů uživatele, ať už ty, které dostal od kamarádů a sám je aktivoval, nebo ty, které napsal on a aktivoval ho některý z jeho kamarádů. Zobrazí se zde také herní kupóny, ty se liší červenou barvou.

3.2.8 Herní kupony

Aktivita **gamecoupons** ukazuje sdílené herní kupóny mezi aktuálním uživatelem a zvoleným kamarádem. Tlačítkem `writenewgamecoupon` se tvoří nové herní kupóny, tlačítko `startgamesession` zkontroluje, jestli je počet herních kupónů sudý a větší, než nula a pokud jsou všechny podmínky splněny, dovolí odstartovat hru. Ve chvíli, kdy se hra odstartuje, se sdílené kupóny náhodně rozdělí napůl pomocí metody `Collections.shuffle()` a každý dostane polovinu. Kupóny se aktivují, takže jdou využít. V případě, že aktuální uživatel už nemá žádné dostupné kupóny, ale zvolený kamarád stále ano, upozorní ho na to text na vršku obrazovky a tlačítka na tvorbu nových kupónů budou skryta.

3.2.9 Aktivita Inspiration Tab

Inspirační kupóny `InspoCoupon`, brané z databáze, se uživateli zobrazují podle typu kupónů, které si přidal do oblíbených. Nejdřív se vyfiltrují kupóny, které má uživatel v oblíbených, aby se mu neukazovaly znovu. Do nově vytvořeného listu přidá pouze ty `InspoCoupons`, které uživatelův seznam oblíbených neobsahuje. Dále se pošle seznam metodě `sortByliked`.

Metoda `sortByliked` nejdříve spočítá, kolik kupónů jakého typu má uživatel v oblíbených, vezme z databáze seznam kuponů, které si uživatel přidal do oblíbených. Dále zjistí, který typ je jeho oblíbený tak, že projde seznam oblíbených a přičte 1 k čítači k typu (páry, kamarádi, rodina, práce), který odpovídá typu kuponu ze seznamu oblíbených. Dále zjistí, který typ má největší četnost a dá kupóny tohoto typu na začátek listu, ostatní kupóny do seznamu přijdou až poté. Vrací nový seznam `InspoCoupon` kupónů.

Tlačítko s ikonou srdíčka na prvku `InspoCouponDetail` přidá kupon do seznamu oblíbených.

4. Frontend

Logo aplikace Couponify, které se používá jako jeho ikona, ikona oznámení i jako splash screen aplikace, bylo vytvořeno v aplikaci Canva. Stylem je minimalistické a má pouze dvě barvy, červenou a modrou. Použité odstíny modré a červené jsou také barevný motiv samotné aplikace a jsou uloženy jako `couponifyblue` a `couponifyred` v xml souboru `colours`, který se nachází v adresáři `values`. Barvu `couponifyblue` používají všechny prvky typu `Button`, `Textinputfield` a také normální kupóny. Herní kupóny mají pozadí barvy `couponifyred`, aby se od těch normálních odlišily. Výjimkou jsou navigační tlačítka na spodku a vršku aktivit.

Navigační tlačítka jsou prvky typu `ImageButton`. Mají podobu různých systémových ikon s černým pozadím, jejich xml soubory jsou uloženy v adresáři `drawable`. Každá aktivita má svou vizuální stránku uloženou v adresáři `layout` jako XML soubor. Všechny aktivity mají navigační tlačítka na spodní a horní liště, většina aktivit využívá třídu `RecyclerView` pro přehledného zobrazení seznamů. Každý prvek v takovém seznamu má definovaný vzhled s dovětkem `listitem`, také v adresáři `layout`.

5. Závěr

Podařilo se mi splnit má očekávání v rámci funkčnosti, všechno, co jsem chtěl aby moje aplikace uměla funguje tak jak by mělo. Místo, které by si zasloužilo zlepšení je ovšem doporučování v sekci inspirace, systém je totiž velmi jednoduchý. Projekt by také prospělo celkové refaktorování kódu, postupně jsem se učil, jak programovat aplikaci pro android a jak pracovat s Firebase, proto jsou aktivity, které jsem dělal mezi prvními místy nešikovně napsané. Na druhou stranu jsem rád, že jsem si zvolil aplikaci pro android, něco, co jsem nikdy nedělal a jsem vděčný za zkušenosti a vědomosti, které mi tento projekt přinesl.

6. Zdroje

- [1] - Platforma Firebase. Dostupné z : <https://firebase.google.com>
- [2] - Služba Firebase Authentication platformy Firebase. Dostupné z: <https://firebase.google.com/docs/auth/>
- [3] - Služba Firebase Realtime Database platformy Firebase. Dostupné z: <https://firebase.google.com/docs/database/>
- [4] - Služba Firebase Cloud Messaging platformy Firebase. Dostupné z: <https://firebase.google.com/docs/cloud-messaging/>
- [5] - okhttp3 - HTTP klient pro android a java aplikace. Dostupné z: <https://square.github.io/okhttp/3.x/okhttp/>
- [6] - Canva - nástroj pro grafický design. Dostupné z: https://www.canva.com/cs_cz/
- [7] - Metoda createUserWithEmailAndPassword z dokumentace Firebase. Dostupné z: <https://firebase.google.com/docs/auth/android/password-auth>
- [8] - Metoda signInWithEmailAndPassword z dokumentace Firebase. Dostupné z: https://firebase.google.com/docs/auth/android/password-auth#java_5
- [9] - ChatGPT, Dotaz: „how do i send notifications in my androidstudio java project using the new fcm api v1?” Dostupné z: <https://chatgpt.com/>