



**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

|                          |                                     |
|--------------------------|-------------------------------------|
| <b>Název:</b>            | Android aplikace pro hledání přátel |
| <b>Student:</b>          | Tomáš Lederer                       |
| <b>Vedoucí:</b>          | Ing. Miroslav Balík, Ph.D.          |
| <b>Studijní program:</b> | Informatika                         |
| <b>Studijní obor:</b>    | Webové a softwarové inženýrství     |
| <b>Katedra:</b>          | Katedra softwarového inženýrství    |
| <b>Platnost zadání:</b>  | Do konce letního semestru 2018/19   |

### Pokyny pro vypracování

Vytvořte seznamovací aplikaci pro Android nabízející uživateli potenciální přátele, které inteligentně vybírá pomocí dat uvedených ve veřejných profilech. Po registraci bude uživateli k dispozici chat s již nalezenými přáteli a vyhledávač přátel. Ve vyhledávači budou postupně zobrazovány jednotlivé profily, u nichž budou uvedeny důvody, proč byl tento návrh zobrazen.

- 1) Popište možnosti podobných aplikací.
- 2) Proveďte analýzu dostupných technologií a API a vyberte vhodné.
- 3) Definujte funkční a nefunkční požadavky.
- 4) Aplikaci implementujte a otestujte.
- 5) Nasaďte betaverzi aplikace.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 2. února 2018





**FAKULTA  
INFORMAČNÍCH  
TECHNOLGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## **Android aplikace pro hledání přátel**

*Tomáš Lederer*

Katedra softwarového inženýrství

Vedoucí práce: Ing. Miroslav Balík, Ph.D.

9. května 2018



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické 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. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 9. května 2018

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2018 Tomáš Lederer. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Lederer, Tomáš. *Android aplikace pro hledání přátel*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

---

## Abstrakt

Cílem této práce je vytvořit mobilní aplikaci, která umožňuje uživateli najít a kontaktovat nové lidi s podobnými zájmy pomocí analýzy jeho aktivit na sociálních sítích. Pro získání relevantních informací jsou využity sociální sítě jako je Facebook, Youtube nebo třeba Spotify, které umožňují vývojářům aplikací po souhlasu uživatele přístup k jeho profilu a aktivitě. Mobilní aplikace je vytvořena pro platformu Android, pro hladký chod využívá cloudových řešení různých společností.

**Klíčová slova** Android, Microsoft Azure, hledání přátel, sociální sítě

---

## Abstract

Android application, which allows user to meet and contact new people with same interests was created as part of this project. It is going to be connected with social networks like Facebook, Youtube or Spotify for accessing user information and providing relevant data. Activity on those networks gives the app an opportunity to match individuals with same interests. Variuos cloud solutions are used for smooth experience.

**Keywords** Android, Microsoft Azure, finding friends, social networks





---

# Obsah

|  |           |
|--|-----------|
| Úvod   | 1         |
| <b>1 Cíl práce</b>                                 | <b>3</b>  |
| <b>2 Existující řešení - Tinder</b>                | <b>5</b>  |
| <b>3 Sociální sítě</b>                             | <b>7</b>  |
| 3.1 Facebook . . . . .                             | 7         |
| 3.2 Spotify . . . . .                              | 8         |
| 3.3 Google . . . . .                               | 9         |
| <b>4 Cloudové služby</b>                           | <b>11</b> |
| 4.1 Microsoft Azure . . . . .                      | 11        |
| 4.2 Cloudinary . . . . .                           | 11        |
| <b>5 Architektura</b>                              | <b>13</b> |
| 5.1 Mobilní aplikace . . . . .                     | 13        |
| 5.2 Server . . . . .                               | 14        |
| 5.3 Datový model . . . . .                         | 15        |
| 5.4 Integrace jednotlivých částí . . . . .         | 16        |
| <b>6 Implementace serveru</b>                      | <b>19</b> |
| 6.1 Přihlašování . . . . .                         | 19        |
| 6.2 Správa profilu a shromažďovaných dat . . . . . | 20        |
| 6.3 Vytváření návrhů přátel . . . . .              | 20        |
| 6.4 Správa návrhů . . . . .                        | 22        |
| 6.5 Chat . . . . .                                 | 23        |
| <b>7 Implementace klienta</b>                      | <b>25</b> |
| 7.1 Podporované verze systému Android . . . . .    | 25        |

|          |   |           |
|----------|---|-----------|
| 7.2      | Navigace v aplikaci, její ovládání a vzhled . . . . . | 26        |
| 7.3      | Přihlašování . . . . .                                | 27        |
| 7.4      | Nastavení . . . . .                                   | 28        |
| 7.5      | Zobrazování návrhů přátel . . . . .                   | 29        |
| 7.6      | Chat . . . . .  | 30        |
| 7.7      | Zpracování obrázků . . . . .                          | 31        |
| <b>8</b> | <b>Testování</b>                                      | <b>33</b> |
| 8.1      | Průběžné testování při vývoji . . . . .               | 33        |
| 8.2      | Uživatelské testování . . . . .                       | 33        |
|          | <b>Závěr</b>  | <b>35</b> |
|          | <b>Literatura</b>                                     | <b>37</b> |
| A        | Seznam použitých zkratk                               | 41        |
| B        | Obsah příloženého flash disku                         | 43        |

---

## Seznam obrázků

|     |  |    |
|-----|--|----|
| 5.1 | MVP diagram . . . . .                            | 14 |
| 5.2 | Datový model . . . . .                           | 16 |
| 5.3 | Integrace . . . . .                              | 17 |
| 6.1 | Algoritmus filtrování společných zájmů . . . . . | 22 |
| 7.1 | Nastavení profilu, Facebooku a Spotify . . . . . | 28 |
| 7.2 | Návrhy přátel . . . . .                          | 29 |
| 7.3 | Chat mezi dvěma uživateli . . . . .              | 30 |



---

# Seznam tabulek

|  |   |
|--|---|
| 2.1 Srovnání tvořené aplikace s Tinderem . . . . . | 6 |
|--|---|



---

# Úvod

Sociální sítě jsou stále rozsáhlejší, lidé na nich tráví více času a především na nich sdílejí více a více informací o svém osobním životě. Tyto veřejně přístupné informace mohou dotyčné osobě značně zkomplikovat život. Mohou ho ale také usnadnit. Pokud uživatel ví, jak a proč se s jeho daty nakládá, a jejich sdílení mu v něčem pomůže, udělá to. Toho bude v práci využito.

Sociální sítě vystavují rozhraní pro vývojáře, kteří pod určitými pravidly a s explicitním souhlasem uživatele mohou nakládat s jeho daty. Je tak možné sestavit list sledovaných celebrit, zájmových skupin, informačních zdrojů, oblíbené hudby a mnoha dalších aktivit na sociálních sítích a vytvořit tak přibližnou představu o zájmech konkrétního uživatele.

Bude vytvořena aplikace pro mobilní platformu Android, která umožní uživatelům vyhledávat nové přátele. Díky znalosti atributů a zájmů konkrétního uživatele v různých sociálních sítích aplikace zobrazí návrhy lidí, se kterými by si mohl rozumět. On se pak rozhodne, zda je chce kontaktovat.

Shromažďování dat tohoto typu může u některých uživatelů vyvolat obavy o to, jak jsou využita a kde všude jsou zobrazena - proto je považováno za důležité uživateli jasně ukázat, jaká data se ukládají, na co se využívají a především, kde a komu jsou zobrazována. Dále bude uživateli poskytnuta možnost nastavení ukládání pouze takových dat, u kterých si to přeje.





---

## Cíl práce

Cílem této práce je vytvořit mobilní aplikaci, s jejíž pomocí bude možné vyhledávat nové přátele. Aplikace bude zobrazovat vhodné návrhy potencionálních nových přátel. Důraz je kladen na jednoduchost aplikace, aby jí mohl používat opravdu každý. Zároveň je ale důležité poskytnout uživateli přehled zpracovávaných dat a možnost jejich úpravy či nepoužití.

Aplikace bude propojena se sociálními sítěmi Facebook, Spotify nebo Youtube a pomocí získaných dat vytvoří jednoduchý přehled o zájmech uživatele. Díky bude moci aplikace navrhnout nové přátele, se kterými by si uživatel mohl rozumět (mají podobné online zájmy).



---

## Existující řešení - Tinder

Tinder je již existující seznamovací aplikace s celosvětovým rozšířením. Byl a je průkopníkem seznamovacích mobilních aplikací, neboť přinesl nový, jednodušší přístup - zobrazuje velké množství návrhů. Tyto návrhy jsou tvořeny podle pohlaví, polohy jednotlivých uživatelů (údaj získaný pomocí GPS) a věku. Uživatelé pak jednotlivé návrhy přijímají či odmítají. Tinder nabízí i několik desítek návrhů, uchyluje se tedy ke kvantitativnímu nabízení potencionálních přátel, z nichž několik může být relevantních.

Kromě základních informací Tinder k tvorbě návrhů využíval data ze sociální sítě Facebook. Po změnách v opatřeních pro ochranu soukromí, stručně popsanych v kapitole 3.1, se Tinder dostal do problémů, kdy mnoho uživatelů nemohlo pokračovat dále do aplikace přes nekonečnou přihlašovací smyčku, kde se aplikace snažila žádat o oprávnění pro přístup k datům, která nově nejsou přes API dostupná [1].

Po prozkoumání aplikace se lze domnívat, že Tinder dále Facebook užívá jen pro přihlašování a na načítání dat rezignoval. Je tedy možné tvrdit, že návrhy jsou nyní zobrazovány pouze na základě lokace, pohlaví, věku a možná historie přijmutí/odmítnutí jednotlivých návrhů [2].

Aplikace vytvořená v rámci této práce má za cíl využít uživatelem poskytnutá data pro zlepšení nabízených návrhů. Kromě zmíněného Facebooku lze využít řady dalších sociálních sítí (popis všech využitých sítí se nachází v kapitole 3). Poloha uživatele není využívána k zobrazení návrhu, neboť není primárním cílem pomoci najít partnera pro milostný vztah, ale „pouze kamaráda“, se kterým typicky není problém být v kontaktu na větší vzdálenost.

Srovnání hlavních funkcí jednotlivých aplikací je uvedeno v tabulce 2.1.

## 2. EXISTUJÍCÍ ŘEŠENÍ - TINDER

---

Tabulka 2.1: Srovnání tvořené aplikace s Tinderem

| Funkce                                | Tinder    | Tvořená aplikace |
|---------------------------------------|-----------|------------------|
| Přihlašování přes Facebook            | ano       | ano              |
| Přihlašování přes Google              | ne        | ano              |
| Přihlašování pomocí telefonního čísla | ano       | ne               |
| Chat s ostatními uživateli            | ano       | ano              |
| Využití polohy uživatele              | ano       | ne               |
| Využití věku uživatele                | ano       | ano              |
| Využití informací z Facebooku         | částečně* | ano              |
| Využití informací ze Spotify          | ne        | ano              |
| Využití informací z YouTube           | ne        | ano              |

\*Pouze jméno, profilový obrázek a věk

## Sociální sítě

### 3.1 Facebook

Facebook poskytuje vývojářům rozsáhlé nástroje pro načítání uživatelských dat, analýzu návštěvnosti, či obsluhu stránek společností (vkládání obsahu, příjem a zasílání soukromých zpráv atp.). V aplikaci je využita možnost uživatele přihlásit se ke svému Facebook účtu a pomocí získaného tokenu se autentizovat v aplikaci třetí strany [3]. Rozhraní umožňuje i načítání veřejně přístupných dat uživatele (např. označení stránek a skupin jako „to se mi líbí“, zúčastnění se akcí apod.).

Přístup k datům dostává pouze klientská aplikace, která je aktivní - uživatel ji právě používá a udělil explicitní oprávnění pro čtení (Facebook při přihlašování zobrazí seznam informací, ke kterým aplikace požaduje přístup - dostane možnost některé nebo všechny odmítnout, případně se vůbec nepřihlásit). O přidání či odebrání může být aplikací požádáno i později po přihlášení, čímž je umožněno uživateli dříve schválená oprávnění měnit [4].

Výhodou je existence knihovny pro Android, což značně zjednodušuje především přihlašování a poskytuje vysokoúrovňové rozhraní pro přístup k uživatelským datům a správě oprávnění [5].

Přístup k některým oprávněním vyžaduje manuální schválení od zaměstnanců Facebooku, kteří především zkoumají jejich přínos pro kvalitu užívání aplikace a mají možnost oprávnění neudělit. Do takové aplikace není uživateli povoleno se přihlašovat [6]. Experimentálně bylo zjištěno, že tento proces není potřeba podstupovat již ve fázi testování, vývojář totiž může ručně povolit jednotlivým Facebook účtům přístup. Takové účty jsou evidovány jako účty testovací.

Od 4. dubna 2018 Facebook v zájmu ochrany osobních údajů odpírá přístup k některým osobním datům uživatelů. Z pohledu aplikace se jedná o důležitá data jako politické a náboženské názory, vzdělání, zaměstnání a krátký text, který uživatel o sobě napsal, aby se představil ostatním [7]. Tyto změny jsou reakcí na neoprávněné užití osobních údajů více než 50 milionů uživatelů

společností Cambridge Analytica [8].

#### **Získané informace:**

- jméno uživatele,
- profilový obrázek,
- datum narození (věk),
- místo bydliště,
- rodné město,
- stránky, které uživatel označil jako „to se mi líbí“.

**Právní záležitosti** – v rámci práce není žádáno o schválení Login Review, neboť aplikace je vyvíjena do fáze beta - nikoliv produkční verze. Login Review tedy není požadováno [6]. Přistupovat do aplikace mohou pouze vybrané účty.

V mobilní aplikaci je využíváno loga Facebook pro označení obsahu, který pochází z Facebooku (při zobrazování návrhů přátel, viz obrázek 7.2), což je v souladu s podmínkami užití [9].

## 3.2 Spotify

Spotify je sice platforma pro legální poslech digitální hudby, nikoliv sociální síť, ale aby mohla nabízet personalizované návrhy hudby, analyzuje aktivitu jednotlivých uživatelů. Zároveň k těmto datům dává po souhlasu uživatele přístup externím vývojářům. Lze tak zjišťovat sledované seznamy skladeb nebo interprety, nejposlouchanější skladby či nedávno přehranou hudbu [10].

Stejně jako Facebook Android vývojářům nabízí knihovnu, která je ale v době psaní práce ve fázi beta verze s upozorněním, že funkce a jejich využití se může bez varování a radikálně měnit [11]. Samotné REST rozhraní, které knihovna obaluje, se pravděpodobně měnit nebude (je využíváno velkým množstvím aplikací a knihoven, které v beta verzi nejsou), zřejmě tedy bude možné dále používat starší verzi knihovny, pokud by nová přinášela nepříznivé změny pro chod aplikace.

#### **Získané informace:**

- oblíbení interpreti,
- oblíbené skladby,
- sledované veřejné seznamy skladeb.

**Právní záležitosti** – načítání uživatelských dat postačuje povolení uživatele a transparentní zpracování jeho dat. Není třeba žádného schvalování aplikace jako v případě Facebooku

Opět je využito loga Spotify k označení obsahu pocházejícího z této služby (viz obrázek 7.2). Toto využití je v souladu se stanovenými podmínkami [12].

### 3.3 Google

Google je technologickým gigantem, který mimo jiné zaštiťuje mobilní platformu Android, využitou v této práci. Díky tomu pro tuto platformu udržuje velké množství knihoven pro jednoduché propojení se zbytkem svých služeb.

Jednou z těchto služeb je například využití již existujícího účtu Google pro autentizaci v serveru třetí strany (podobně jako Facebook) [13]. Velkou výhodou však je, že u většiny uživatelů platformy Android lze předpokládat vlastnictví tohoto účtu (pro hojně využívané Google Play Services je nezbytný).

Vzhledem k velkému množství poskytovaných služeb by bylo krajně nepraktické distribuovat společně s aplikací jednu knihovnu podporující všechny služby. Celý ekosystém je tak rozdělen na mnoho menších knihoven, z nichž každá podporuje jednu ze služeb. V aplikaci je využito již zmíněného přihlašování, dále Google Places API (pro načítání míst a bodů zájmu – využití viz kapitola 7.4) a YouTube API, jež poskytuje přístup k aktivitě uživatele v rámci této služby [14].

#### Získané informace:

- jméno uživatele,
- profilový obrázek,
- seznam všech měst (manuální nastavení bydliště uživatelem),
- seznam vzdělávacích institucí (manuální nastavení vzdělání uživatele),
- sledované kanály videí na YouTube,
- oblíbená videa na YouTube.

**Právní záležitosti** – ochrana osobních údajů funguje podobně jako u Spotify: vývojář musí souhlasit s podmínkami využití služby, které primárně zahrnují transparentní nakládání s daty a jejich nezneužívání. V aplikaci je využita ikona Youtube pro označení obsahu pocházejícího z této sítě, což je v souladu s podmínkami využití [15].





---

# Cloudové služby

## 4.1 Microsoft Azure

Microsoft Azure je rozsáhlá cloudová platforma, která poskytuje velké množství služeb od provozu webových stránek přes databáze a úložiště po virtualizované systémy a umělou inteligenci [16].

Azure je placená služba, funguje na principu platby za využití prostředky (např. využití úložiště, přenesená data, minuty výpočetního výkonu atd.) [17]. Studenty vysokých škol může být využita nabídka bonusového kreditu a provozu vybraných služeb zcela zdarma [18].

V práci je využito služby SQL databáze pro data a Webové aplikace pro komunikaci s klientem přes REST rozhraní. Výhodou oproti čistě mobilnímu backendu (případně podobné službě Firebase) je možnost následného vytvoření webové stránky nebo nativní desktopové aplikace bez nutnosti úpravy serveru.

Cloudová služba Webová aplikace uvnitř skrývá IIS - webový server běžící v prostředí .NET. Na serveru je možné spustit ASP.NET program napsaný v jazyce C#. Je možné využít framework pro tvorbu REST rozhraní i Entity Framework, což je ORM framework navržený přímo pro .NET, podporující LINQ (funkcionální paradigma pro tvorbu dotazů na datamodel).

V Azure je možno použít několik typů SQL databází, zvolena byla SQL databáze společnosti Microsoft.

## 4.2 Cloudinary

Cloudinary je cloudová služba sloužící pro ukládání a správu multimédií. V práci je využito služeb pro obrázky. Kromě samotného úložiště (v bezplatné verzi 10GB [19]) jsou k dispozici rozsáhlé nástroje pro správu - převedení obrázku na jiný formát, oříznutí, vycentrování na obličej, zakulacení rohu a spoustu dalších, které v práci nejsou využity [20]. Každá takováto transfor-

#### 4. CLOUDOVÉ SLUŽBY

---

mace probíhá na serveru Cloudinary a je uložena, může tak být opětovně použita, čím se výrazně snižuje doba odezvy.

K dispozici je knihovna pro Android, která umožňuje jednoduše přistupovat k API [21]. Veškerou manipulaci s obrázky tedy zadává klient serverům Cloudinary a na server následně ukládá pouze URL, čímž se docílí snížení zátěže tohoto serveru.

Cloudinary lze využít i pro velké projekty, kromě neplacené služby existuje možnost škálovat i do velikosti mnoha terabajtů úložiště a vysokého uživatelského provozu [19].

---

# Architektura

## 5.1 Mobilní aplikace

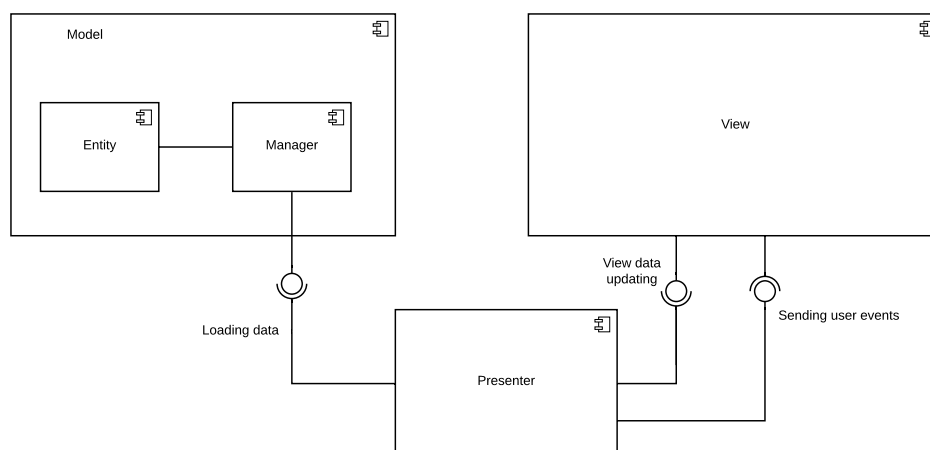
Pro mobilní (Android) aplikaci byla zvolena architektura MVP. Oproti standardní MVC architektuře, která se při vývoji pro Android využívá, nabízí jasnější oddělení business logiky od logiky uživatelského rozhraní. Snižuje tím jejich provázanost a minimalizuje tak dopad změn v jedné do druhé.

V MVC architektuře roli controlleru zastává Aktivita či Fragment. Ten načítá data z modelu a zobrazuje je v uživatelském rozhraní definovaném v XML souboru. Tento jednoduchý přístup se v mnoha případech (především pro jednoduché obrazovky) hodí. Slabinou ale je, že složitější manipulace s rozhraním nelze provádět přímo v XML. Ve výsledku pak v Aktivitách vznikají metody, které načítají data z modelu a zároveň upravují uživatelské rozhraní. Takový kód je hůře čitelný a špatně udržovatelný [22].

Přidání Presenteru umožňuje zcela oddělit business logiku od zobrazovacího tým, že se Aktivita či Fragment „degraduje“ na pasivní View. V praxi to znamená, že View vystaví rozhraní, pomocí kterého může Presenter přidávat či upravovat zobrazovaná data (vedlejším efektem je, že správnost zobrazovaných lze testovat pomocí jednotkových testů, což je v případě Aktivit jakožto controlleru téměř nemožné). View předává Presenteru všechny události vyvolané akcemi uživatele (například zmáčknutí tlačítka). Nutno podotknout, že odeslání této události je veškerá aktivita, kterou View provádí. Jakoukoliv možnou změnu dat neočekává ani od Presenteru nepožaduje. Presenter reaguje na zmíněné akce uživatele a v případě potřeby upravuje View, které pro to poskytuje rozhraní.

Poslední část, Model, zaštiťuje veškerou business logiku. Model typicky neinicuje komunikaci s Presenterem, pouze vystavuje veřejné rozhraní, čímž umožňuje přístup k potřebným datům, ať už jsou umístěny kdekoliv. V obecném MVP není přímo řečeno, jak má Model přesně vypadat, pro účely této práce však bylo zvoleno rozdělení na entities (datové objekty), managers (správce) a utilities (nástroje). Presenter se dotazuje manageru na konkrétní entity či

Obrázek 5.1: MVP diagram



jejich kolekce. Zároveň může požádat o vytvoření, úpravu, či smazání entity. Manager tak může například spravovat lokální databázi, kterou v pozdější fázi vývoje začne synchronizovat se vzdáleným úložištěm - tato změna by při správné implementaci neměla ovlivnit Presenter ani View.

Základní struktura aplikace je vyjádřeno diagramem na obrázku 5.1.

## 5.2 Server

Na serveru je využito vzoru architektury MVC. View zde není implementováno - controlerem je vystavováno REST rozhraní, pomocí kterého je možné komunikovat s view. Tím může být například webová stránka nebo nativní aplikace. V případě této práce lze z pohledu serveru za View považovat celou mobilní aplikaci.

Uvnitř aplikace je využito více controlerů (konkrétně zvláště pro správu uživatelských účtů, profilů, chatu a návrhů přátel). Ty obsluhují jednotlivé přístupové body REST rozhraní.

Data jsou databáze ukládány pomocí databázových kontextů rozšiřujících základní databázový kontext poskytnutý ORM frameworkem. Kontexty mohou být volány přímo z controleru (vhodné pro jednoduché CRUD operace),

případně je vytvořena mezivrstva nazvaná manager, která přesune složitější kód mimo controler. Jednotlivé kontexty a managery tak tvoří model aplikace.

### 5.3 Datový model

Mobilní aplikace i server využívají velké množství datových tříd jak pro vnitřní potřebu, tak pro komunikaci mezi sebou. V této kapitole je popsána pouze hlavní, nejdůležitější část.

**Profile** slouží k uložení všech užitečných informací o uživateli. Obsahuje základní informace jako jméno, datum narození, krátký textový popis osoby, odkaz na profilový obrázek v nízkém a vysokém rozlišení. Dále i kolekci zájmů (*Interest*), které jsou získány z připojených sociálních sítí.

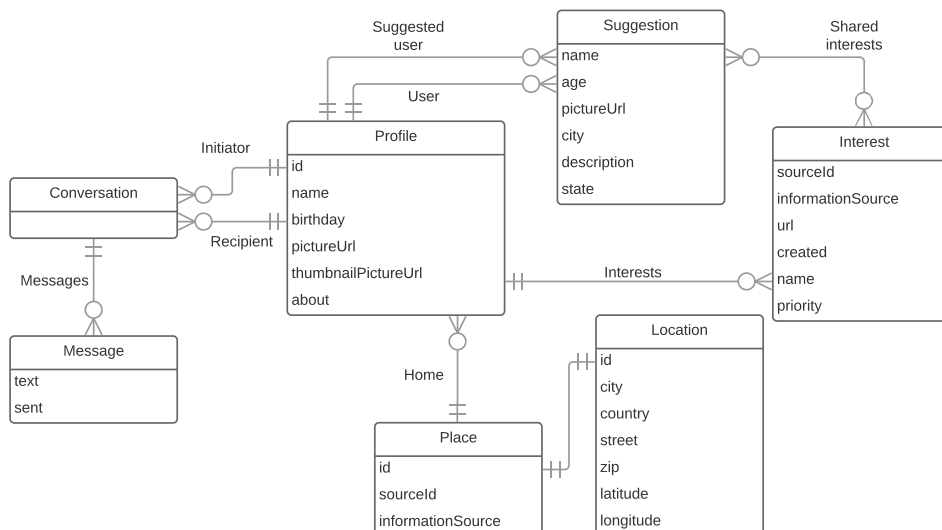
**Interest** obsahuje informace o zájmu uživatele. Ukládán je název zájmu (pro zobrazení dalším uživatelům), ale i odkaz pro zjištění dalších informací. Pro identifikaci je uchován zdroj informace a unikátní ID, které zájem má přidělené v rámci zdroje (například ID skladby ve službě Spotify).

**Suggestion** je návrh potencionálního přátelství pro konkrétního uživatele. Tento objekt je vytvářen serverem na základě shodných zájmů jednotlivých uživatelů. Kromě základních textových informací o uživateli obsahuje položku *state*, která určuje stav návrhu - nový, přijatý či odmítnutý. Návrhy pak mohou být uchovány a nezobrazovány opakovaně.

**Conversation** reprezentuje konverzaci mezi uživateli automaticky vytvořenou ve chvíli, kdy uživatel přijme návrh (*Suggestion*). Obsahuje kolekci zpráv zaslaných mezi uživatelem co návrh přijal a uživatelem, co byl navržen.

Hlavní část modelu je popsána i v diagramu na obrázku 5.2.

Obrázek 5.2: Datový model



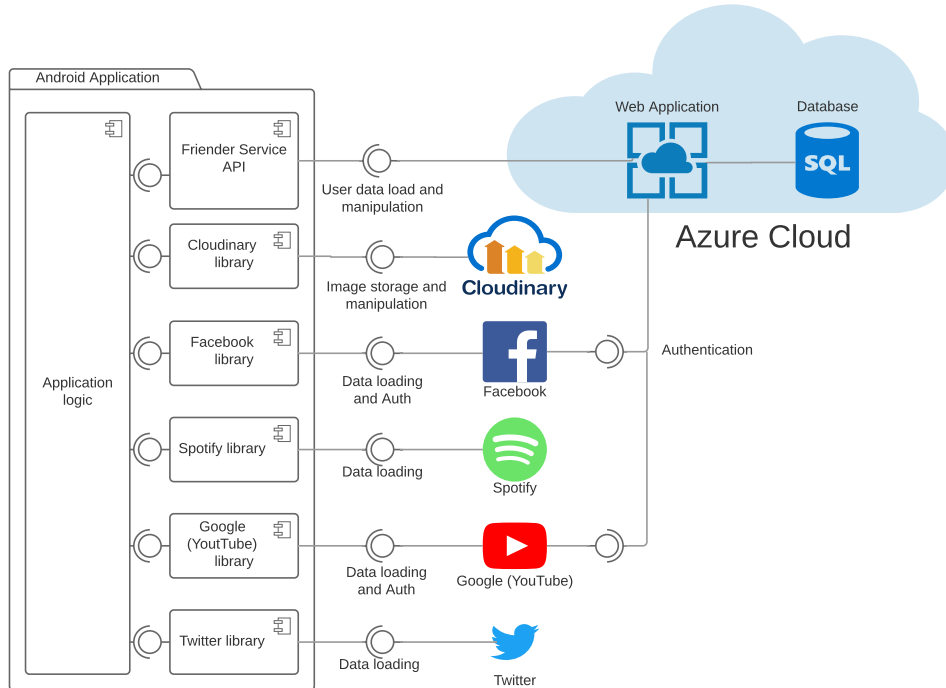
## 5.4 Integrace jednotlivých částí

Základem integrace je propojení klientské a serverové části aplikace. Komunikují spolu pomocí REST API. Pro podepisování požadavků na server je využíván token, který je klientem získán přihlášením – detailně popsáno v kapitole 6.1. Na server je ukládán uživatelský profil obsahující základní údaje o dané osobě a aktivitě v propojených sociálních sítích.

Tyto informace jsou načítány klientem z jednotlivých sítí – jsou zpracovány tak, aby byly kompatibilní s datovým modelem použitým na serveru a následně nahrány na server jakožto úprava profilu. Sociální síť tedy nejsou ze strany serveru dotazovány na uživatelská data, s výjimkou autentizace, kdy je serverem ověřována identita uživatele (kontrola přístupového tokenu).

Pro ukládání obrázků je využito služby Cloudinary (viz. kapitola 4.2). Obrázky jsou klientem ukládány na Cloudinary, kde proběhne konverze obrázku na vhodnou velikost a případné transformace. Na server se následně místo obrázku ukládá pouze URL vygenerované službou Cloudinary. Výhodou je značné snížení zátěže serveru (krátké textové URL oproti potenciálně velkému obrázku) a přesunutí procesorově náročných transformací na externí službu. Současná implementace může být jednoduše upravena na použití s jiným cloudovým řešením nebo případným vlastním serverem.

Obrázek 5.3: Integrace



Komunikace mezi všemi využitými službami a aplikacemi je zjednodušeně popsána v diagramu na obrázku 5.3.





---

# Implementace serveru

Server je implementován jakožto ASP.NET služba nasazená v aplikačním serveru IIS. Pro potřeby aplikace je využito cloudové platformy Azure (viz kapitola 4.1). Pro komunikaci s klientem jsou poskytovány REST služby, které využívají standardního protokolu HTTP (případně lépe HTTPS).

Pro uložení dat je využito databáze SQL, ke které je přistupováno nepřímo přes ORM framework.

## 6.1 Přihlašování

Přihlašování probíhá způsobem podobným OAuth2. Díky tomu lze v aplikaci využít autentizaci pomocí poskytovatele identity, například Facebook nebo Google. Přímá autentizace pomocí jména a hesla není možná, proto uživatel musí být zaregistrován u jednoho z poskytovatelů.

Pro využití těchto služeb je nutné aplikaci vývojářem zaregistrovat v jednotlivých systémech. Server poté může přímo komunikovat s poskytovatelem identity. Přihlašování uživatele funguje následovně:

1. Uživatel se na svém zařízení přihlásí k poskytovateli identity (prostřednictvím nativní aplikace nebo přes webový prohlížeč).
2. Poskytovatelem identity je klientské aplikaci vystaven přístupový token.
3. Token je odeslán na server.
4. Serverem je ověřena platnost tokenu u poskytovatele identity.
5. Pokud je token validní, server získá od poskytovatele ID přihlášeného uživatele, pomocí kterého může být nalezen příslušný již existující účet v databázi, případně vytvořen nový.
6. Serverem je vystaven jiný přístupový token, kterým budou podepisovány požadavky klienta na server. Tento proces tak nemusí být opakován při

každém požadavku z klientské aplikace (token je uložen ve vlastní databázi).

REST API je bezstavové, neukládá data o komunikujících uživateli. Token je tedy vyžadován v hlavičce každého dotazu, neboť slouží i k rozlišení jednotlivých uživatelů, kteří mohou být připojeni ve stejný čas.

### 6.2 Správa profilu a shromažďovaných dat

Jednotlivé údaje o uživateli jsou na server nahrávány v modelové třídě `Profile` (viz diagram na obrázku 5.2). Pokud uživatelský profil ještě neexistuje, je vytvořen nový. V opačném případě je starý upraven. Úprava profilu probíhá:

- V případě jednoduchého textového pole nebo čísla (jméno, datum narození...), jehož hodnota v nahrané entitě nenabývá hodnoty `null`, je hodnota přepsána. Pole s hodnotou `null` označuje přání hodnotu neměnit.
- V případě kolekce hodnot (zájmy) proběhne sloučení s již existující kolekcí. Hlavním řídicím údajem je `informationSource`, určující zdroj získané informace. Upravovaná kolekce je tak zúžena na zájmy, které mají stejný zdroj informací, jako zájmy v nově nahrané kolekci kolekce (může jich být více než 1). Při sloučení jsou odstraněny záznamy, které se v nově nahrané kolekci nenacházejí, aktualizovány záznamy, které již existují a přidány záznamy nové.

Dále je možné smazat všechny zájmy pocházející z vybraného informačního zdroje (seznam informačních zdrojů je uveden v kapitole 5.3). Toho je využito především, když se uživatel rozhodne dále nepoužívat některé informace, které byly dříve na server nahrány. Zároveň umožní zcela smazat veškerá data o službě, kterou uživatel od aplikace odpojí. Z implementačních důvodů není možné takto smazat základní údaje jako jméno a datum narození, uživatel ale má možnost je, na rozdíl od zájmů, manuálně změnit či odstranit.

### 6.3 Vytváření návrhů přátel

Ze zájmů uložených v databázi jsou vytvářeny návrhy potencionálních přátel. Při změně zájmů uživatele jsou jeho staré návrhy, které jsou ve stavu „nový“ – uživatel se k nim zatím nevyjádřil – smazány a vygenerovány nové. Návrhy, jejichž cílem je uživatel měněny nejsou. Tím je dosaženo kompromisu mezi zobrazováním nejnovějších informací a výpočetním výkonem nutným k provedení operace změny profilu a vygenerování návrhů. Drobnou nevýhodou je časová mezera z pohledu ostatních uživatelů při smazání či změně některých údajů, ta by ovšem neměla být delší než 24 hodin (interval pro aktualizaci

údajů z každého informačního zdroje v implementaci klienta), kdy každému uživateli budou postupně vygenerovány aktuální návrhy.

V první fázi vytváření návrhu jsou postupně procházeny jednotlivé zájmy uživatele. Pro každý zájem jsou vyhledáni všichni uživatelé, kteří daný zájem sdílejí. V operační paměti jsou vytvořeny návrhy spojení s těmito uživateli.

Podle společných zájmů je vytvářena priorita návrhu – číslo (čím větší, tím je návrh silnější). Ta závisí na počtu společných zájmů a jejich vlastní prioritě. Konkrétně – zájem s nízkou prioritou má váhu 1, se střední 2 a s vysokou 3\*. Jednotlivé hodnoty jsou sečteny a je vytvořena priorita návrhu.

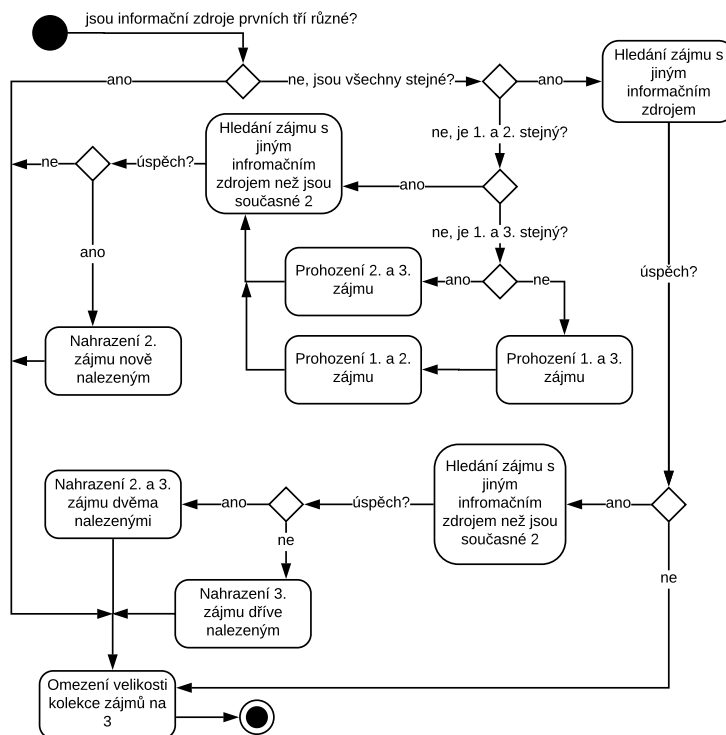
Následně probíhá filtrování společných zájmů u každého z návrhů. Cílem je uložit a zobrazovat pouze 3 nejpodstatnější společné zájmy, aby uživatel nebyl zahlcen informacemi a zároveň nebylo zobrazováno zbytečně mnoho informací o cíli návrhu. Algoritmus se také snaží použít návrhy z co nejvíce různých informačních zdrojů. Filtrování probíhá následovně:

1. Zájmy v návrhu jsou seřazeny podle priority z pohledu uživatele, pro kterého je návrh tvořen.
2. Probíhá optimalizace seřazení tak, aby první 3 zájmy byly z co největšího počtu informačních zdrojů. Zde může nastat více situací.
  - a) Každý zájem má jiný informační zdroj: ideální situace – není nutné pořadí návrhů upravovat.
  - b) Všechny zájmy mají stejný informační zdroj: v kolekci zájmů je hledán zájem s jiným informačním zdrojem. Pokud je nalezen, je dále hledán zájem se zdrojem jiným než původní zájmy a jiným než má nově nalezený. Podle úspěchu těchto operací nejsou nahrazeny žádné z původních společných zájmů (neexistuje společný zájem z jiného zdroje informací), případně je nahrazen poslední z původních zájmů nebo prostřední a poslední.
  - c) Třetí společný zájem má jiný informační zdroj než první dva: Je hledán společný zájem s jiným informačním zdrojem, než mají tyto tři zájmy. V případě nalezení je jím nahrazen druhý z těchto zájmů.
  - d) Druhý společný zájem má jiný informační zdroj než první s třetím: Prohozením třetího zájmu s druhým je dosaženo situace shodné s bodem c) a je pokračováno stejným způsobem.
  - e) První zájem má jiný informační zdroj než druhý a třetí: Prohozením prvního a třetího zájmu je dosaženo situace shodné s bodem c), aby ale byl nahrazen zájem s nižší prioritou, jsou navíc prohozeny zájmy, které jsou nově na 1. a 2. pozici.

---

\*tyto hodnoty jsou generovány relativně k celkovému počtu zájmů z jednoho informačního zdroje. Zdroje poskytující větší množství informací tak nemají nutné vyšší prioritu než zdroje s méně daty

Obrázek 6.1: Algoritmus filtrování společných zájmů



### 3. Počet společných zájmů je omezen na 3.

Následně mohou být jednotlivé návrhy a jejich vztahy k zájmům uživatele nahrány do databáze, odkud mohou být načteny a zobrazeny uživateli. Výše popsaný algoritmus je zobrazen diagramem na obrázku 6.1

## 6.4 Správa návrhů

Server poskytuje rozhraní pro načtení všech návrhů ve stavu „nový“. Ty následně mohou být v klientské aplikaci zobrazeny a uživatel má možnost se k nim vyjádřit.

V případě, že je návrh přijat, mění se jeho stav na „přijatý“ a je automaticky vytvořena konverzace mezi uživatelem a cílem návrhu. Je tak poskytnuta možnost navrženého uživatele kontaktovat. V případě odmítnutí je pouze změněn stav na „odmítnutý“.

Přijaté a odmítnuté návrhy nejsou mazány, jsou dále využívány ke kontrole duplicit návrhů (uživateli tak nejsou zobrazovány stále stejné návrhy).

## 6.5 Chat

Server obsahuje rozhraní pro jednoduchý chat. Nutno podotknout, že není cílem této práce vytvořit konkurenceschopnou chatovací platformu. Důvod existence této funkce je možnost kontaktování navrženého potencionálního přítele, krátký rozhovor, případně výměna kontaktních informací a přesunutí dalších interakcí na platformu k tomu určenou.

Není však tvořeno žádné technické omezení, které by uživatelům bránilo v komunikaci. Není tedy nutné aplikaci opouštět, pokud si to uživatelé nepřejí. Postrádány však mohou být některé funkce, které moderní chatovací nástroje běžně obsahují.

Klientské aplikaci je umožněno načíst seznam konverzací, které uživatel vede. V těchto konverzacích, z důvodu snížení velikosti přenášených dat, nejsou obsaženy jednotlivé zprávy (pouze poslední pro zobrazení náhledu). Zprávy obsažené v jednotlivých konverzacích jsou načítány zvlášť. Je možné využít načítání pouze těch zpráv, které byly odeslány až po určité zprávě (vhodné pro optimalizaci množství přenesených dat, kdy klientská aplikace lokálně ukládá již načtené zprávy).

Nejsou implementovány push notifikace pro mobilní telefony ani websockets pro internetové prohlížeče. Server tak nemá možnost přímo kontaktovat klienta. Nové zprávy tak klient může získat jen aktivním požadováním nových zpráv od serveru (ideálně za použití výše zmíněného rozhraní, kdy se načítají pouze nové zprávy).

Nejsou implementovány moderní oblíbené funkce jako samolepky (vtipné sady obrázků, které se dají poslat jako zpráva), posílání vlastních obrázků nebo videí, GIF klávesnice (automatické vyhledávání GIF animací na internetu a jejich odeslání).

Uživatelé si mezi sebou mohou posílat jen čistě textové zprávy, které však díky kódování UTF-8 mohou doplnit smajlíky či jinými symboly obsaženými v tomto kódování, jejichž vzhled bude záviset na platformě konkrétního uživatele (symboly jsou předávány jako kód symbolu, samotný případný obrázek je pak součástí fontu v konkrétním zařízení).



---

## Implementace klienta

Klientská aplikace je vytvořena pro mobilní platformu Android. Využívá se nativní framework určený pro tvorbu aplikací této platformy vyvíjený společností Google, majitelem Androidu. Aplikace se řídí sadou doporučení o vzhledu grafických prvků Material Design [23], tvořenou zmíněnou společností. Díky tomu aplikace vzhledově zapadá do ekosystému aplikací v této platformě a zkušenější uživatelé tak vědí, kde hledat běžné ovládací prvky. Je tak pro ně jednodušší aplikaci začít používat.

Kromě popisu jednotlivých funkcí jsou součástí práce jednotlivé obrazovky zachycené z reálného zařízení. Pro přehlednější oddělení od textu a zdůraznění původu obrázku je využito nástroje pro vytvoření rámečku zařízení okolo zachycené obrazovky [24]. Data zobrazená na těchto snímcích jsou vždy data smyšlená, případně týkající se výhradně autora práce.

### 7.1 Podporované verze systému Android

Verze systému označené 6 (Marshmallow) a 7 (Nougat) dohromady podle statistik Google Play [25] tvoří téměř 58% všech aktivních zařízení platformy Google Play, lze je tedy považovat za nejvíce rozšířené verze a jsou hlavním cílem vývoje aplikace. Stále ještě hojně využívaná verze 5 (Lollipop) s necelými 23 procenty zařízení, dosluhující verze 4.4 (Kitkat) s podílem 10% nejsou testovány ani podporovány. Nejnovější verze 8 (Oreo) je testována a podporována i se svým podílem ve velikosti čtyř procent, neboť lze v nejbližších letech očekávat její rozšíření.

Verze Lollipop i všechny starší neobsahují některé pro tvořenou aplikaci podstatné části Android frameworku. Pro jejich podporu by bylo nutné provádět jisté kompromisy v návrhu aplikace i jejím vizuálním vzhledu pro koncového uživatele, autor se proto rozhodl tyto verze netestovat ani nepodporovat.

Minimální požadovanou verzí systému pro běh aplikace je Android 6 (Marshmallow), podporovány tak jsou necelé dvě třetiny všech aktivních zařízení na Google Play.

## 7.2 Navigace v aplikaci, její ovládání a vzhled

Při prvním spuštění aplikace je zobrazena obrazovka s jednotlivými možnostmi autentizace. Po přihlášení je otevřena hlavní aktivita aplikace, která obsahuje 3 záložky, mezi kterými lze přepínat gestem „přetažení do strany“ nebo zvolením příslušné záložky na panelu akcí v horní části obrazovky. Jedná se o jeden z doporučených návrhových vzorů navigace pro malé množství stejně důležitých obrazovek podle Material Design [26]. Každá z těchto záložek obsahuje jednu z hlavních funkcí aplikace (počínaje z levé strany – nastavení profilu, zobrazení návrhů, chat).

Jednotlivé záložky mohou pro své potřeby otevírat nové aktivity (například pro vyšší úroveň detailu nastavení v profilu). Tyto aktivity mají jednotný design - v horní části obrazovky obsahují jednoduchý panel akcí s názvem konkrétní aktivity a šipku směřující vlevo, indikující její podřazenost a možnost přejít výše v hierarchii aplikace [27].

Každá z hlavních záložek obsahuje `SwipeRefreshLayout`, Android prvek implementující návrhový vzor „Swipe to Refresh“ popsany v Material Design [28]. Tím je umožněno manuálně, stažením obrazovky směrem dolů, obnovit zobrazený obsah.

Zmíněný `SwipeRefreshLayout` je často pro efektivní zobrazení velkého množství vizuálně shodných prvků (návrhy přátel, konverzace, zprávy v konverzaci) doplněn `RecyclerView`. Každý z prvků je umístěn na jedné kartě, která obsahuje jednotlivá `view` s daty. V případě klasického seznamu by bylo nutné pro každý prvek vytvořit v paměti kartu a případně jí zobrazit. Jak již název napovídá, `RecyclerView` jednotlivé karty recykluje, vytvořeno je jen nezbytně nutné množství (tolik, kolik se vejde na obrazovku včetně drobného přesahu) a při odsunutí karty z obrazovky ji znovu použije, pouze se změní údaji.



## 7.3 Přihlašování

Pro využití aplikace je uživatel povinen se zaregistrovat a dále přihlašovat svým účtem. Při prvním spuštění aplikace je mu tak zobrazena přihlašovací obrazovka.

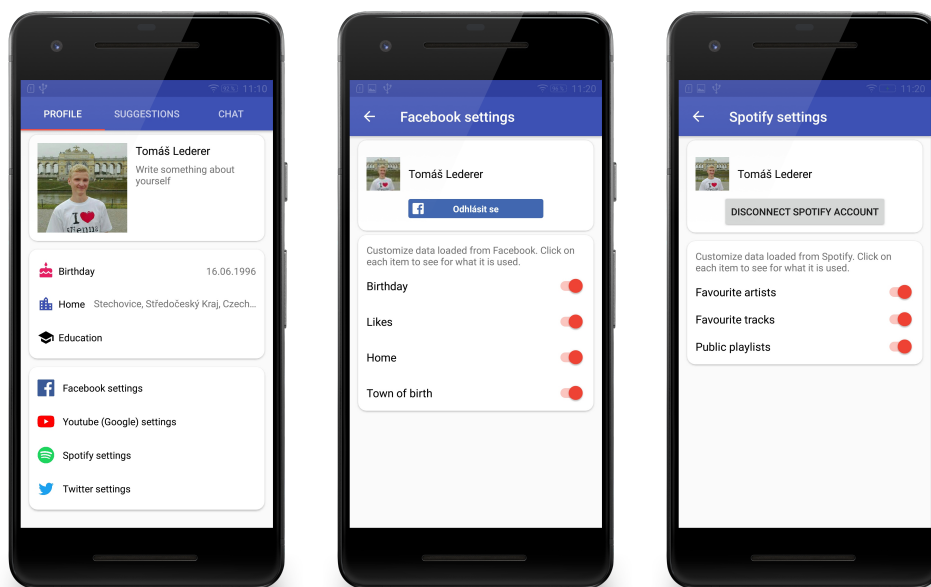
Autentizace uživatelé probíhá pomocí externích poskytovatelů identity, Facebook nebo Google. Vzhledem k platformě aplikace lze u uživatele předpokládat vlastnictví alespoň účtu Google (především při distribuci přes platformu Google Play, kde je tento účet pro stažení aplikace vyžadován). Přihlašování jen pomocí jména (případně e-mailu) a hesla možné není.

Při prvním spuštění aplikace je uživatel vyzván, aby se přihlásil pomocí jednoho z těchto poskytovatelů a zrevidoval oprávnění přístupů, které aplikace požaduje. Ty lze později měnit – více v kapitole 7.4. Schválení oprávnění při přihlášení ovlivní pouze výchozí nastavení aplikace.

Pokud předchozí kroky proběhly v pořádku, je zahájena autentizace uživatele vůči serveru aplikace. Celý proces je detailně popsán v kapitole 6.1. V případě úspěšného přihlášení a vytvoření nového uživatele je zahájeno navíc počáteční načítání základních informací o uživateli od poskytovatele identity (jméno, věk, profilový obrázek) a až poté přeměrováno na obrazovku s návrhy přátel (viz obrázek 7.2). Je tak zabráněno uživateli vyplnit základní informace, které mohou být získány automaticky a je mu tím ušetřena případná zbytečná práce. Pokud si uživatel přeje údaje vyplnit ručně, může načtené údaje přepsat.

Přístupový token vydaný serverem po úspěšném přihlášení je aplikací uložen v perzistentní paměti zařízení, aby mohl být znovu použit při příštím startu aplikace a proces přihlašování nemusel být pokaždé opakován a nezpomaloval spuštění aplikace. Přístupové tokeny externích služeb mají omezenou platnost, volby uživatele jsou ale uloženy (v případě nativních aplikací v datech dané aplikace, v případě přihlášení přes webový prohlížeč v cookies tohoto prohlížeče), v případě opakované žádosti o přihlášení tak je využito předchozí volby a nový přístupový token je bez prodlení a zásahu uživatele aplikací vydán.

Obrázek 7.1: Nastavení profilu, Facebooku a Spotify

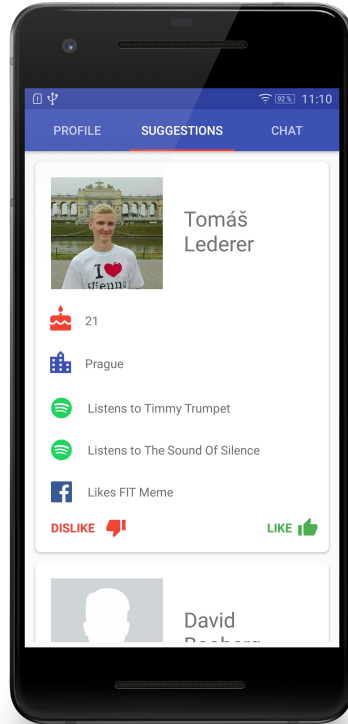


### 7.4 Nastavení

V záložce Profil (viz obrázek 7.1) může uživatel prohlížet základní data o své osobě (jméno, datum narození, popis). Všechny údaje lze klepnutím na příslušnou položku změnit. To platí i pro profilový obrázek (klepnutím je otevřena galerie, kde je možné zvolit například dříve pořízenou fotografii). Místo bydliště a vzdělání se mění ve speciální aktivitu, která nabízí automatické doplňování místa bydliště (město, obec...) a vzdělávacích institucí za pomoci Google Places API.

Kromě úpravy profilu může uživatel měnit nastavení jednotlivých sociálních sítí, případně je ke svému účtu připojit nebo naopak odpojit. Každá sociální síť poskytuje určitá data, která lze o uživateli získat (viz kapitola 3). Každý typ informace je možné v nastavení pomocí přepínače vypnout či zapnout a mít tak plnou kontrolu nad zpracovávanými daty. Pokud je to technicky možné, při vypnutí načítání určité skupiny dat jsou dříve načtená data nevratně smazána z úložiště serveru (toto konkrétně není možné u atomických polí typu jméno nebo datum narození, kde z důvodu zvoleného způsobu uložení není jasné, odkud je tento údaj získán). Nastavení pro síť Facebook a Spotify jsou zobrazeny na obrázku 7.1

Obrázek 7.2: Návrhy přátel



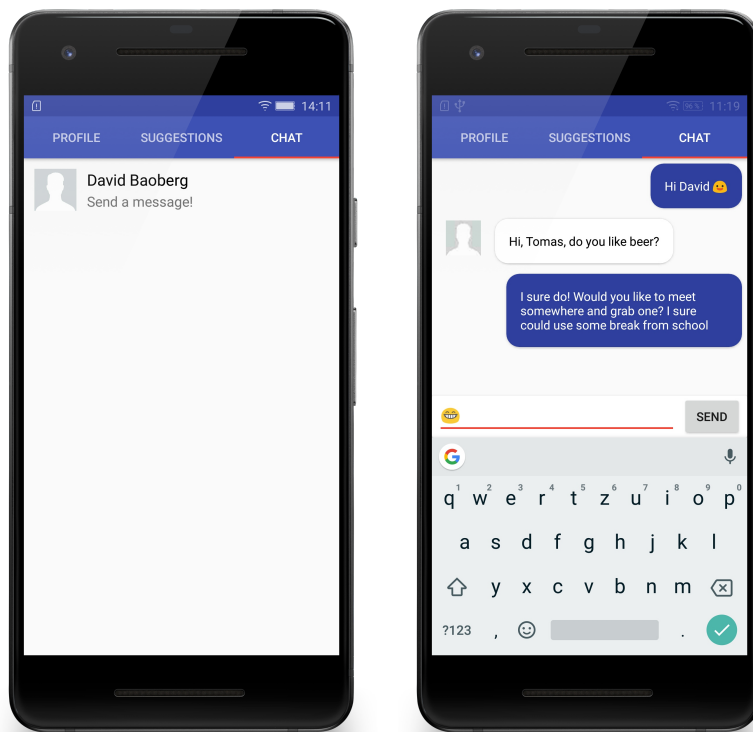
## 7.5 Zobrazování návrhů přátel

V záložce návrhů zachycené na obrázku 7.2 jsou uživateli zobrazovány jednotlivé návrhy potencionálních přátel. Kromě profilového obrázku a jména jsou na kartě návrhu uvedeny informace o věku, bydlišti uživatele a až 3 zájmy, které navrhovaná osoba sdílí s uživatelem. Tyto zájmy jsou označeny jako nejdůležitější ze všech společných za využití algoritmu popsáno v kapitole 6.3.

Uživatel má možnost daný návrh odmítnout - tlačítkem „dislike“. Tento návrh tak bude vyřazen se seznamu návrhů a již se uživateli nezobrazí.

V opačném případě má možnost návrh přijmout tlačítkem „like“ a následně danou osobu přímo v aplikaci pomocí chatu kontaktovat.

Obrázek 7.3: Chat mezi dvěma uživateli



### 7.6 Chat

V případě přijetí návrhu přátelství jedním z uživatelů je automaticky vytvořena konverzace. Seznam těchto konverzací je umístěn v záložce chat. Klepnutím na danou konverzaci se je možné ji otevřít a prohlížet zprávy, které si mezi sebou uživatelé poslali, případně zaslat zprávu novou, viz obrázek 7.3.

Jednotlivé zprávy jsou zobrazeny na kartách. Pro vizuální oddělení přijatých a odeslaných zpráv jsou odeslané zprávy na modrém pozadí a jsou zarovnané vpravo. Přijaté zprávy jsou zarovnané vlevo a je vedle nich umístěn profilový obrázek uživatele, se kterým probíhá konverzace.

Ve spodní části obrazovky se nachází textové pole, umožňující napsat zprávu pro odeslání. Pokud si to uživatel přeje, může ve zprávě využít UTF-8 smajlíky - většina softwarových klávesnic umožňuje jejich přímé zadání do textového pole. Pro ty nejběžnější je však vytvořen parser pro jejich ASCII zápis.

## 7.7 Zpracování obrázků

Aplikace byla navržena pro spolupráci se službou Clodianary (viz kapitola 4.2). Bylo využito knihovny pro Android a pohodlného zpracování obrázků ve službě, bez výpočetního zatížení klienta. Využívalo se snížení rozlišení pro rychlejší přenos, oříznutí na čtverec (vycentrovaný na obličej) a zakulacení rohů obrázku. Spolupráce se službou fungovala hladce v případě využití již zkomprimovaných obrázků v nízkém rozlišení (méně než 1Mpx) načtených z jednotlivých sociálních sítí. Služba provedla zmíněně transformace – výstupními soubory byly dva obrázky: jeden v rozlišení 500x500px, druhý 64x64px. Větší byl využit při návrzích nových přátel a v nastavení profilu, menší jako ikona v zobrazení konverzace a chatu.

V pozdní fázi vývoje, při implementaci manuální úpravy profilového obrázku – například změnou na fotografii z uživatelova zařízení – vyvstaly jisté komplikace. Samotné nahrávání fotografie o velikosti i několika megabajtů na server trvalo příliš dlouho (především při využití mobilní sítě) a samotné zpracování cloudovou službou též nebylo okamžité – transformovaný obrázek byl k dispozici v extrémním případě až za několik hodin. Průměrně se však jednalo o minuty. Nebylo zjištěno, zda autor provedl chybu v implementaci, nebo zda byla služba nešťastnou náhodou příliš vytížena, případně zda limitovala výpočetní výkon pro neplatící uživatele.

Z časových důvodů, jelikož šlo o jednu z posledních funkcí, které byly implementovány, bylo zvoleno implementačně nejjednodušší řešení – zpracování obrázku pomocí mobilního zařízení a následné nahrání na server ve formátu **Base64** (řetězec) v textovém poli určeném pro odkaz na obrázek v Clodianary. Nebylo tak nutné měnit implementaci serveru, pouze asynchronní úlohu klientské aplikace zpracovávající jednotlivé obrázky.

Android framework obsahuje třídu **Bitmap**, která umožňuje základní manipulace s obrázky. Pomocí v ní obsažené **BitmapFactory** lze vytvořit bitmapu ze souboru v zařízení. Ta je následně pomocí metody **createBitmap** zmenšena na největší možný vycentrovaný čtverec (v případě obrázku na šířku oříznutí shodných svislých pruhů po stranách, v případě obrázku na šířku oříznutí horizontálních pruhů). Následně je bitmapa zmenšena na 500x500px, zduplikována, duplikát zmenšen na 64x64px. Obě bitmapy jsou zkomprimovány pomocí JPEG, uloženy do bitového pole, které je následně převedeno do Base64 a uloženo jako řetězec.

Ačkoliv se jedná o nepříliš elegantní řešení, je značně výkonnější než řešení předchozí. I na relativně starém zařízení (více než 3 roky) zpracování obrázků trvá několik desetin sekundy. Výsledné obrázky mají velikost v řádu desítek kilobajtů, následné nahrání obrázků na server tak zabere minimální čas a i při využití mobilní sítě je celá operace provedena téměř okamžitě a nenarušuje tak pohodlnou interakci s aplikací.



---

# Testování

## 8.1 Průběžné testování při vývoji

Testovací prostředí pro vývoj bylo navrženo tak, aby testování probíhalo co nejrychleji, nejpohodlněji a příliš nezpomalovalo vývoj.

Server byl spouštěn lokálně na aplikačním serveru IIS verze Express umožňující ladění (krokování, sledování hodnot proměnných atp.) běžícího serveru. Jako databáze bylo využito LocalDB, speciální odlehčené verze SQL Serveru (který je v plné verzi použit při nasazení v Azure) určené pro vývoj aplikací.

Mobilní aplikace byla testována na emulátoru zařízení Google Pixel 2 s nejaktuálnější verzí Android 8.1 (Oreo). Dále bylo využito autorova osobního zařízení Huawei P8 Lite s Android verze 6.0 (Marshmallow), zapůjčeného Lenovo Vibe S1 též s Android verze 6.0 a zapůjčeného Xiaomi Redmi 4X s Android verze 7.1 (Nougat). Tím jsou pokryty všechny podporované verze systému a mělo by tak být zajištěno fungování aplikace na většině zařízení obsahující některou z nich.

## 8.2 Uživatelské testování

Již v průběhu vývoje byly prováděny drobné uživatelské testy. Testovacímu subjektu vždy byla zapůjčena aplikace a subjekt byl požádán, aby provedl nějakou operaci (např. přihlásit se, změnit si jméno v profilu, přijmout návrh přátelství a kontaktovat danou osobu). Tyto testy nebyly dokumentovány, protože jich bylo velké množství a neměly by vysokou vypovídající hodnotu – podněty a postřehy testovacích subjektů byly zaregistrovány a případné nepřehledné či nevýrazné ovládací prvky byly upraveny.





---

## Závěr

V rámci práce byla vytvořena aplikace, která pomocí dat ze sociálních sítí vytváří návrhy osob, se kterými by si uživatel mohl rozumět. Je zobrazen krátký profil uživatele (jméno, profilový obrázek, věk, bydliště, společné zájmy), pomocí kterého se uživatel rozhodne, zda chce danou osobu kontaktovat. Pokud ano, je mu to umožněno pomocí vestavěného základního chatovacího nástroje.

Ačkoli při plánování práce bylo považováno za nejsložitější část aplikace vytváření návrhů přátel, tak při vhodném vytvoření datového modelu nepředstavovalo výraznou překážku. Nejvíce času při vývoji zabralo získávání dat ze sociálních sítí (především co se týče výzkumu dostupných technologií a jejich integrace) a také optimalizace mobilní aplikace (snaha o využití návrhových vzorů Material Design – vzhledové a funkční začlenění aplikace do Android ekosystému) a optimalizace procesů připojování a odpojování sociálních sítí od účtu uživatele (především zajištění smazání dříve načtených dat).

Značnou překážku v pozdní fázi vývoje představovalo Cloudinary, cloudové úložiště obrázků, které nedokázalo dostatečně rychle zpracovávat nově nahrané obrázky ve vysoké kvalitě – fotografie z mobilního telefonu. Vzhledem k tomu, že tato nevýhoda se projevila až v pozdní fázi implementace (vysoce komprimované obrázky v nízkém rozlišení ze sociálních sítí nepředstavovaly problém), bylo využito jiného řešení, tedy zpracování obrázku pomocí výpočetních zdrojů mobilního zařízení a následné uložení v databázi serveru ve formátu Base64 s ostatními daty profilu. Více v kapitole 7.7.

V implementaci je částečně připravena podpora sociální sítě Twitter, která však nebyla v dostatečné míře prozkoumána ani implementována.

Aplikace se nachází ve fázi beta, která implikuje fungování primárních funkcí aplikace, nevylučuje ale projevení některých nedokonalostí v optimalizaci nebo kontrole špatných vstupů, které nebyly odhaleny během testování. Aplikace však za běžných situací běží stabilně a poskytuje plynulý průběh jejího užívání.

Díky nutnosti projít manuálním schválením aplikace pro vhodnost fungování se sociální sítí Facebook a případně potřebě rozšíření fyzických zdrojů

## ZÁVĚR

---

využitého serveru (a neexistujícího obchodního modelu pro pokrytí těchto nákladů) není aplikace veřejně publikována v obchodě Google Play, ani žádném obdobném; není přístupná veřejnosti.

---

## Literatura

- [1] Liao, S.: All of Facebook's privacy fixes have broken Tinder. *The Verge [online]*, duben 2018, [cit. 2018-04-15]. Dostupné z: <https://www.theverge.com/2018/4/4/17200034/facebook-broke-tinder-down-privacy-api-fixes>
- [2] Tinder: Tinder. [software]. Dostupné z: <https://play.google.com/store/apps/details?id=com.tinder>
- [3] Manually Build a Login Flow. *Facebook for Developers Docs [online]*, [cit. 2018-03-06]. Dostupné z: <https://developers.facebook.com/docs/facebook-login/manually-build-a-login-flow>
- [4] Permissions with Facebook Login. *Facebook for Developers Docs [online]*, [cit. 2018-03-06]. Dostupné z: <https://developers.facebook.com/docs/facebook-login/permissions/overview>
- [5] Facebook SDK for Android. *Facebook for Developers Docs [online]*, [cit. 2018-03-04]. Dostupné z: <https://developers.facebook.com/docs/android>
- [6] What is Login Review? *Facebook for Developers Docs [online]*, [cit. 2018-03-04]. Dostupné z: <https://developers.facebook.com/docs/facebook-login/review/what-is-login-review>
- [7] Archibong, I.: API and Other Platform Product Changes. *Facebook for Developers Blog [online]*, duben 2018, [cit. 2018-04-07]. Dostupné z: <https://developers.facebook.com/blog/post/2018/04/04/facebook-api-platform-product-changes/>
- [8] Cambridge Analytica: all the news about Facebook's data privacy scandal. *The Verge [online]*, [cit. 2018-04-15]. Dostupné z: <https://www.theverge.com/2018/3/19/17141266/facebook-cambridge-analytica-user-data-donald-trump-campaign-2016-election>

- [9] f logo. *Brand Resource Center [online]*, [cit. 2018-04-14]. Dostupné z: <https://en.facebookbrand.com/assets/f-logo>
- [10] Web API Endpoint Reference. *Spotify Developer [online]*, [cit. 2018-04-05]. Dostupné z: <https://developer.spotify.com/web-api/endpoint-reference/>
- [11] Spotify Android SDK. *Spotify Developer [online]*, [cit. 2018-04-05]. Dostupné z: <https://developer.spotify.com/technologies/spotify-android-sdk/>
- [12] Design Resources. *Spotify Developer [online]*, [cit. 2018-04-15]. Dostupné z: <https://developer.spotify.com/design/>
- [13] Google Sign-In for Android. *Google Developers [online]*, [cit. 2018-05-07]. Dostupné z: <https://developers.google.com/identity/sign-in/android/>
- [14] YouTube API Reference. *Google Developers [online]*, [cit. 2018-05-07]. Dostupné z: <https://developers.google.com/youtube/v3/docs/>
- [15] YouTube API Services - Branding Guidelines. *Google Developers [online]*, [cit. 2018-05-07]. Dostupné z: <https://developers.google.com/youtube/terms/branding-guidelines>
- [16] Produkty Azure. *Microsoft Azure [online]*, [cit. 2018-04-11]. Dostupné z: <https://azure.microsoft.com/cs-cz/services/>
- [17] Ceny Azure. *Microsoft Azure [online]*, [cit. 2018-04-11]. Dostupné z: <https://azure.microsoft.com/cs-cz/pricing/>
- [18] Azure for Students. *Microsoft Azure [online]*, [cit. 2018-04-11]. Dostupné z: <https://azure.microsoft.com/cs-cz/offers/ms-azr-0170p/>
- [19] Pricing & Plans. *Cloudinary [online]*, [cit. 2018-04-9]. Dostupné z: <https://cloudinary.com/pricing>
- [20] Image Manipulation. *Cloudinary [online]*, [cit. 2018-04-9]. Dostupné z: [https://cloudinary.com/features/image\\_manipulation](https://cloudinary.com/features/image_manipulation)
- [21] Android integration. *Cloudinary [online]*, [cit. 2018-04-09]. Dostupné z: [https://cloudinary.com/documentation/android\\_integration](https://cloudinary.com/documentation/android_integration)
- [22] Leiva, A.: MVP for Android: how to organize the presentation layer. *Antonio Leiva Blog [online]*, [cit. 2017-12-29]. Dostupné z: <https://antoniroleiva.com/mvp-android/>
- [23] Material Design Guidelines. *Material Design [online]*, [cit. 2018-04-28]. Dostupné z: <https://material.io/guidelines/>

- 
- [24] Device art generator. *Android Developers [online]*, [cit. 2018-04-28]. Dostupné z: <https://developer.android.com/distribute/marketing-tools/device-art-generator.html>
- [25] Distribution dashboard. *Android Developers [online]*, [cit. 2018-04-30]. Dostupné z: <https://developer.android.com/about/dashboards/>
- [26] Patterns – Navigation. *Material Design [online]*, [cit. 2018-05-06]. Dostupné z: <https://material.io/guidelines/patterns/navigation.html#navigation-patterns>
- [27] Patterns - Navigation, Up and Back buttons. *Material Design [online]*, [cit. 2018-05-06]. Dostupné z: <https://material.io/guidelines/patterns/navigation.html#navigation-up-back-buttons>
- [28] Patterns - Swipe to refresh. *Material Design [online]*, [cit. 2018-05-06]. Dostupné z: <https://material.io/guidelines/patterns/swipe-to-refresh.html>



## Seznam použitých zkratk

**.NET** dotnet - prostředí pro běh programů, podobné Java Virtual Machine

**API** Application Programming Interface

**ASP** Active Server Pages

**CRUD** Create Read Update Delete

**GIF** Graphics Interchange Format

**GPS** Global Positioning System

**GUI** Graphical User Interface

**HTTP** HyperText Transfer Protocol

**HTTPS** HyperText Transfer Protocol Secure

**ID** Identifier

**IDE** Integrated Development Environment

**IIS** Internet Information Services

**JPEG** Joint Photographic Experts Group (podle tvůrců formátu)

**LINQ** Language Integrated Query

**MVC** Model-View-Controller

**MVP** Model-View-Presenter

**OAuth2** Open Authentication 2

**ORM** Object-Relational Mapping

**REST** Representational State Transfer

## A. SEZNAM POUŽITÝCH ZKRATEK

---

**SQL** Structured query language

**URL** Uniform Resource Locator

**UTF-8** Unicode Transformation Format

**XML** Extensible Markup Language



---

## Obsah přiloženého flash disku

|  |                  |  |
|--|------------------|--|
|  | readme.txt.....  | stručný popis obsahu CD  |
|  | bin .....        | adresář se spustitelnou formou implementace                                      |
|  | mobile.apk.....  | instalační soubor mobilní aplikace   |
|  | server .....     | výstupní soubory backend serveru   |
|  | src .....        | zdrojové kódy implementace   |
|  | mobile.....      | Android studio projekt – mobilní aplikace  |
|  | server .....     | Visual Studio projekt – server   |
|  | thesis .....     | zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ |
|  | doc.....         | dokumentace  |
|  | thesis.pdf ..... | text práce ve formátu PDF  |