



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**MOBILNÍ APLIKACE PRO PREZENTACI INFORMACÍ
ZE SVĚTA ANIME**

MOBILE APPLICATION FOR PRESENTATION OF THE ANIME WORLD INFORMATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VEDOUCÍ PRÁCE

SUPERVISOR

JIŘÍ ODEHNAL

Ing. JIŘÍ HYNEK

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Odehnal Jiří**

Obor: Informační technologie

Téma: **Mobilní aplikace pro prezentaci informací ze světa anime**
Mobile Application for Presentation of Anime World Information

Kategorie: Uživatelská rozhraní

Pokyny:

1. Provedte průzkum dostupných aplikací a služeb určených pro komunitu lidí se zájmem o anime. Analyzujte jejich výhody a nedostatky.
2. Prostudujte algoritmy určené k doporučování produktů.
3. Seznamte se s principem tvorby mobilních aplikací pro operační systém Android.
4. Na základě pokynů vedoucího a poznatků získaných z bodů 1 až 3 navrhnete mobilní aplikaci určenou pro prezentaci a doporučování informací ze světa anime (prezentace titulů; doporučení titulů dle historie a preferencí uživatele; vyhledávání blízkých uživatelů se stejnými zájmy na základě GPS apod.). Práci implementujte jako rozšíření existující služby happo.cz.
5. Navrženou aplikaci implementujte.
6. Otestujte přívětivost implementovaného uživatelského rozhraní, navrhnete možná rozšíření a nastiňte další vývoj projektu.

Literatura:

- Johnson, J.: *Designing with the Mind in Mind: Simple Guide to Understanding User Interface Design Guidelines*. Burlington: Morgan Kaufmann Publishers/Elsevier, 2010, ISBN 978-0-12-375030-3.
- Machine Learning. *Coursera: Online Courses From Top Universities* [online]. 2016 [cit. 2016-11-08]. Dostupné z: <https://www.coursera.org/learn/machine-learning>
- Introduction to Android. *Android Developers* [online]. [cit. 2015-10-23]. Dostupné z: <https://developer.android.com/guide/index.html>.

Pro udělení zápočtu za první semestr je požadováno:

- Body 1 až 4.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Hynek Jiří, Ing.**, UIFS FIT VUT

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta Informačních technologií
Ústav informačních systémů
612 66 Brno, Božetěchova 2



doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá návrhem a vývojem mobilní aplikace pro platformu Android jakožto rozšíření stávající webové služby Happo.cz. Cílem práce je implementovat jednoduché a přívětivé uživatelské rozhraní, které by poskytovalo základní informace o animovaných filmech. Aplikace by měla zároveň podporovat přátelské vztahy mezi uživateli a podněcovat jejich vzájemnou komunikaci. Součástí práce je také realizace serverové části a systému pro doporučování obsahu.

Abstract

This bachelor thesis deals with the design and development of a mobile application for Android platform as an extension of the existing Happo.cz web service. The aim of the work is to implement a simple and user-friendly user interface that provides basic information about animated films. The application should also support friendly relationships between users and encourage their mutual communication. Part of the thesis is also implementation of the server part and the content recommendation system.

Klíčová slova

mobilní aplikace, Android, uživatelské rozhraní, design, doporučování obsahu, anime

Keywords

mobile application, Android, user interface, design, recommender system, anime

Citace

ODEHNAL, Jiří. *Mobilní aplikace pro prezentaci informací ze světa Anime*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Hynek Jiří.

Mobilní aplikace pro prezentaci informací ze světa Anime

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Hynka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Jiří Odehnal
11. května 2017

Poděkování

Chtěl bych poděkovat svému vedoucímu bakalářské práce Ing. Jiřímu Hynkovi za odborné vedení, rady a připomínky při vypracovávání této práce. Dále bych chtěl poděkovat všem, kteří mi pomáhali s testováním aplikace a za jejich zpětnou vazbu.

Obsah

1 Úvod	3
2 Analýza a současná situace	4
2.1 Anime a komunita v ČR	4
2.1.1 Anime	4
2.1.2 Japonská jména a názvy děl	4
2.1.3 Česká komunita	5
2.2 Současná situace	5
2.3 Nativní aplikace a nová funkcionalita	7
2.4 Analýza uživatelských požadavků	8
2.5 Podobné a existující aplikace	9
2.6 Výběr mobilní platformy	10
2.6.1 Verze API	12
3 Metody doporučování obsahu	13
3.1 Systém doporučování obsahu	13
3.1.1 Definice problému doporučování obsahu	13
3.1.2 Dělení systémů doporučování	14
3.2 Doporučení na základě obsahu	14
3.2.1 Základní princip	14
3.2.2 Reprezentace položky	14
3.2.3 Uživatelský profil a jeho učení	15
3.2.4 Metody a algoritmy	15
3.2.5 Výhody a nevýhody	16
3.3 Kolaborativní filtrování	17
3.3.1 Základní princip	17
3.3.2 k-Nearest Neighbor	18
3.3.3 Association rules base prediction	18
3.3.4 Matrix factorization	19
3.3.5 Výhody a nevýhody CF	20
3.4 Hybridní systémy	21
4 Návrh mobilní aplikace	22
4.1 Uživatelské rozhraní	22
4.1.1 Návrh matice objektů a akcí	22
4.1.2 Návrh obrazovek a barevné rozvržení	23
4.1.3 Dialogy a notifikace	27
4.2 Serverová část	28

4.2.1	Databáze	28
4.2.2	Systém doporučování obsahu	28
4.3	Komunikace se serverem	30
4.3.1	Formát dat	30
5	Implementace	31
5.1	Programovací jazyky a vývojové prostředí	31
5.2	Klientská část	31
5.2.1	Přenos a ukládání multimédií	31
5.2.2	Seznamy	32
5.2.3	Mapa a GPS souřadnice	32
5.2.4	Notifikační zprávy a upozornění	33
5.2.5	Chat	34
5.2.6	Statistiky	34
5.3	Použité externí knihovny	34
5.3.1	Http Request	34
5.3.2	MPAndroidChart	34
5.3.3	Google Maps API	35
5.3.4	ViewPagerIndicator	35
5.3.5	CircleImageView	36
5.4	Serverová část	36
5.4.1	API rozhraní	36
5.4.2	Systém doporučování obsahu	37
6	Testování	38
6.1	Testování aplikace na různých verzích API	38
6.2	Testování grafického uživatelského rozhraní	38
6.2.1	Testování použitelnosti aplikace	38
6.2.2	Vyhodnocení testování	39
6.2.3	Shrnutí	43
7	Závěr	44
	Literatura	45
	Přílohy	48
	Seznam příloh	49
A	Dotazník	50
B	Screeener pro aplikaci Happo.cz	52
C	Formulář pro ověření uživatelského rozhraní aplikace Happo.cz	53
C.1	Sada úloh	53

Kapitola 1

Úvod

Moderní technologie jsou všude kolem nás a jen stěží se jim dokážeme vyhnout. Díky rozšíření a všude přítomné dostupnosti internetu se mladistvým více dostává do podvědomí fenomén kreslených animovaných filmů z asijsky mluvících zemí nazývaný anime. V současné době se stále více rozrůstá česká a slovenská komunita zabývající se touto tematikou, avšak doposud neexistuje jediná aplikace, která by českým a slovenským uživatelům poskytovala být jen ty nejzákladnější informace o jednotlivých titulech v jejich rodném jazyce. Mobilní aplikace by tak mohla být řešením, pro stále se rozrůstající komunitu nadšenců animovaných filmů a seriálů. Mohla by jim poskytnout rychlé a jednoduché ovládání a značnou úroveň mobility. Aplikace by mohla uživatelům nabízet nejen samotné vyhledávání, ale i prohlížení titulů nebo vytváření seznamů s oblíbenými položkami. I když aktuálně existuje řešení v podobě webové aplikace, z pohledu mobilního uživatele se nejedná o nejlepší řešení, neboť není pro mobilní zařízení uzpůsobeno ani po grafické a ani po funkční stránce. Díky vzrůstajícímu zájmu o toto téma však nastává hned několik problémů. Hodně diváků často neví, jaký titul sledovat, anebo hledají nové tituly na základě svých oblíbených žánrů. Druhým problémem je komunikace. Sociální sítě a diskusní fóra se jen hemží dotazy nadšenců, zda se v jejich okolí nevyskytuje někdo se stejným zájmem. Často tak nastává problém s kým diskutovat, když ve svém okolí nikoho s podobným zájmem neznají. Tyto problémy by mohla vyřešit mobilní aplikace v podobě doporučování obsahu, podněcování komunikace a vytváření přátelských vztahů mezi uživateli. Uživatelé by například měli možnost sdílet svoji pozici na mapě, být upozornění na přátele v okolí a v neposlední řadě dát ostatním vědět, že se v jejich okolí také nachází někdo se stejným zájmem.

Cílem práce je tak navrhnout a implementovat mobilní aplikaci jako rozšíření stávající webové služby Happs.cz, doplněnou o funkcionalitu sdílení polohy a doporučování obsahu. Součástí práce je analýza existujících řešení a výběr vhodného postupu při řešení. Přičemž patřičný důraz je kladen na návrh uživatelského rozhraní, se zaměřením na jednoduchost a přívětivost.

Kapitola 2

Analýza a současná situace

V této kapitole jsou představeny důležité pojmy a je analyzován současný stav aplikací pro získávání informací o anime titulech. Taktéž jsou rozebrány požadavky na aplikaci a následně je uvedena vize aplikace.

2.1 Anime a komunita v ČR

2.1.1 Anime

Termín *anime* byl původně převzat z anglického slova *animation*. Poprvé jej použil Imamura Taihei v roce 1948 ve své knize, jako náhradu za těžkopádné slovo *mangaeiga*. V Japonsku se termínem *anime* označuje jakákoliv animovaná tvorba, neohledě na zemi původu nebo styl. Mimo Japonsko se místo *anime* používá spíše termín japonská animovaná tvorba. V západních zemích lze za japonské animované dílo (anime) považovat animaci, která buď vznikla přímo v Japonsku nebo většina hlavních tvůrců (režisér, scénárista, návrháři postav či animátor) jsou Japonci. [19]

O to, v jakém kontextu by se mělo slovo anime používat, se mezi fanoušky vedou již delší dobu spory. Mezi anime je totiž zařazena spousta titulů, které jsou vykreslovány pomocí počítačů a snaží se tak spíše o realistickou vizualizaci, než o animovaný vzhled. [21] Na druhou stranu se mezi fanoušky objevují názory, že za anime lze označit i díla animovaná japonským stylem, ve kterých se objevují typické rysy pro japonská díla. Mezi anime jsou tak zařazovány i tituly, které sice svým stylem i kresbou anime připomínají, ovšem jedná se o tvorbu mimo Japonko. Neexistuje tak hranice, která by jednoznačně určila, co anime je a co není.

Pojem anime zahrnuje velké odvětví animované tvorby, která je stejně jako filmy, členěna do různých žánrů. Proto veřejností často zaujímaný názor „Anime je pro děti.“ je zcela milný. Díla pro děti tvoří pouze část celkové tvorby. Naopak větší část je věnována spíše dorostu a dospělému publiku. Samostatné odvětví pak tvoří pornografická díla tzv. *hentai*.

2.1.2 Japonská jména a názvy děl

Pro lepší orientaci v japonských jménech a názvech je vhodné si v krátkosti uvést způsoby přepisů. Japonský písemný systém používá převzaté čínské znaky tzv. kanji a dvě odvozené slabikové abecedy, hiraganu a katakanu. Japonská jména se tak zpravidla skládají z několika kanji znaků, přičemž každý znak má sám o sobě určitý význam. Japonský písemný systém

je ovšem značně komplexní a pro zbytek světa i složitý na naučení. Proto se přistupuje k používání různých způsobů přepisu japonských názvů do latinky. Jedním z nejpoužívanějších přepisů je Hepburnový přepis. Ten slouží hlavně pro přepis mezi japonštinou a angličtinou. U jmen však zůstává zachováno japonské pořadí, tedy napřed příjmení a poté jméno. [7]

V případě jmen je většinou používán jen Hepburnový přepis. U názvů filmů a seriálů je situace poněkud komplikovanější, neboť vedle Hepburnova přepisu se lze často setkat i s oficiálním anglickým názvem, neoficiálním anglickým překladem, Hepburnovým přepisem s českou transkripcí a v neposlední řadě i s různými zkomoleninami překladů. Obecně jsou ale nejvíce zažity tři verze názvu díla. Oficiální název psaný japonskými znaky, Hepburnův přepis s anglickou transkripcí a v případě, že bylo dílo vydáno i v anglicky mluvících zemích, tak i oficiální anglický název. [7]

2.1.3 Česká komunita

Vznik české komunity fanoušků a příznivců anime lze datovat do období roků 1999 a 2000, kdy začaly vznikat první malé skupiny nadšenců. Tyto skupiny a jednotlivci mezi sebou komunikovali převážně pomocí chatu. V té době bylo poměrně složité sehnat jakékoliv anime a možnosti internetu ještě nebyly tak široké jako dnes. Skupiny se tak začaly scházet převážně za účelem výměny filmů a seriálů. V pozdějších letech se podobné komorní srazy proměnily v pravidelné setkávání. V průběhu roku 2003 a 2004 začaly vznikat i první internetové portály, které se zabývaly anime a manga tematikou, čímž položily základ pro budoucí moderní českou komunitu. [9] V následujících letech na internetu začaly vznikat i novinkové portály a přístup k anime filmům a seriálům byl značně jednodušší. Ve velkých městech se začaly pořádat srazy, které se postupně přeměnily na festivaly. Díky lepší dostupnosti anime titulů vznikaly první překladatelské skupiny tvořené převážně nadšenci.

I když v posledních letech zaznamenala česká komunita fanoušků značný nárůst, stále je tvořena spíše jednotlivci. Značná část fanoušků je spíše pasivní, kdy se sice o anime zajímají, ale nedávají to veřejně najevo. Tento stav je z části způsoben faktem, že v očích společnosti jsou manga komiksy a sledování anime označovány za projev dětinskosti a často bývají terčem posměchu.

2.2 Současná situace

Současná webová aplikace Happs.cz¹ slouží jako databáze anime, postav, soundtracků a vizuálních novel. Hlavním účelem aplikace je poskytnout uživateli informace o jednotlivých titulech či postavách. Uživatel má možnost vytvoření vlastního účtu, pod kterým může spravovat své seznamy oblíbených položek, hodnotit a komentovat tituly nebo se podílet na tvorbě obsahu. Tato služba je sice přístupná na mobilních zařízeních přes běžný webový prohlížeč, nicméně častému používání nevyhovuje hned z několika důvodů.

Prvním problémem je neoptimalizované uživatelské rozhraní. Aplikace nebyla navržena pro mobilní zařízení a případná úprava by byla značně náročná na časové zdroje. Z toho důvodu webová aplikace není připravena pro současné rozlišení displejů mobilních zařízení. Je nevhodná pro ovládání pomocí dotyků a práce s ní je tak značně zdlouhavá. Navíc, fragmenty stránek poskytující navigaci se v rámci aplikace nezobrazují korektně a působí tak nepřijemným a matoucím dojmem.

¹<http://anime.happs.cz>

Profile

Level 34



Stitch

- Profil
- Zprávy
- Přátelé
- Animelist
- Characterlist
- Soundtracklist
- Novellist
- Administrace

Anime menu

- Vyhledávání
- Anime
- Licencovaná anime v ...
- Video
- Uživatelé
- Top otaku
- Statistiky
- Informace a pravidla
- Happo tým
- Slovníček

Překladačské skupiny

- Happo tým
- Pozvání skupin
- Přehled všech skupin
- Založit skupinu

Inou Battle wa Nichijou-kei no Naka de

Základní informace Shmutí Autoři Odkazy Sdílej

Originální název: Inou Battle wa Nichijou-kei no Naka de Kategorie: Harém

Název ve znacích: 異能バトルは日常系のなかで Komedie

Název anglicky: Inou Battle Within Everyday Life Romantika

Rok vydání: 2014 Supersila

Vysíláno: 07.10.2014 - 23.12.2014 Školní život

Typ zpracování: TV Série

Předloha: Light novela

Počet epizod: 12

Země původu: Japonsko

Studio: Trigger

Čas trvání anime: 5 hodin 0 minut

Čas na epizodu: 0 hodin 25 minut

Počet zobrazení: 230 x

47.5% (4)

Epizody/titulky Hodnocení Komentáře a recenze Animelist Překladky Anime vztahy Postavy Soundtracky Novely Galerie a videa

Pořadí	Originální název		Český název			Doba
	Přidáno	Skupina	Přeložil	Release	Verze	
1.	Change (Alpha Episode)					25 min
2.	Misconception					25 min
3.	Rendezvous Point					25 min
4.	Capricious Lady					25 min
5.	Sensitive Age					25 min
6.	Vice Penalty					25 min
7.	Juggernaut On					25 min
8.	Holmgang Battle					25 min
9.	Girls Approach					25 min
10.	Fool's Labyrinth					25 min
11.	Cupid Error					25 min
12.	Usual Days					25 min

Obrázek 2.1: Ukázka webové aplikace.

S tím souvisí i těžkopádné ovládání chatu. Uživatel je neustále nucen stránku přibližovat, neboť text je bez přiblížení prakticky nečitelný. Zároveň dochází pomocí technologie AJAX k periodické aktualizaci okna chatu, což způsobuje nepříjemné problikávání stránky.

Dalším problémem je datová náročnost. Při načtení stránky zobrazující jediný titul, dojde ke stáhnutí dat o objemu přibližně 1,5 až 3 MB. V případě zobrazení na osobním počítači s běžným internetovým připojením jde o zanedbatelný objem dat. Dokonce i v případě mobilních zařízení připojených do Wi-Fi sítě by se nejednalo o problém. Ve většině případů jde totiž o internetové připojení, kde není omezena velikost přenesených dat. To ovšem neplatí při použití mobilního internetového připojení, kdy většina operátorů nastavuje Fair Use Policy (FUP). S objemem stažených dat také souvisí rychlost připojení. I když v současnosti je značně rozšířena technologie 4G LTE, stále je dostupnost a rychlost ovlivňována různými

faktory jako je vzdálenost od vysílače, zastávka, materiál budov, profil terénu nebo počet uživatelů v okolí. To vše může působit na rychlost načítání jednotlivých stránek aplikace.

2.3 Nativní aplikace a nová funkcionalita

V dnešní době vlastní tzv. chytrý telefon, tedy mobilní telefon s operačním systémem, dotykovou obrazovkou a výkonným hardwarem, více než 55% všech domácností. [6] Mobilní telefon má oproti počítači obrovskou výhodu, která spočívá v jeho vysoké mobilitě. To je také jeden z důvodů, proč jsou v dnešní době nativní aplikace tak populární.

Nespornými výhodami nativní aplikace tak jsou:

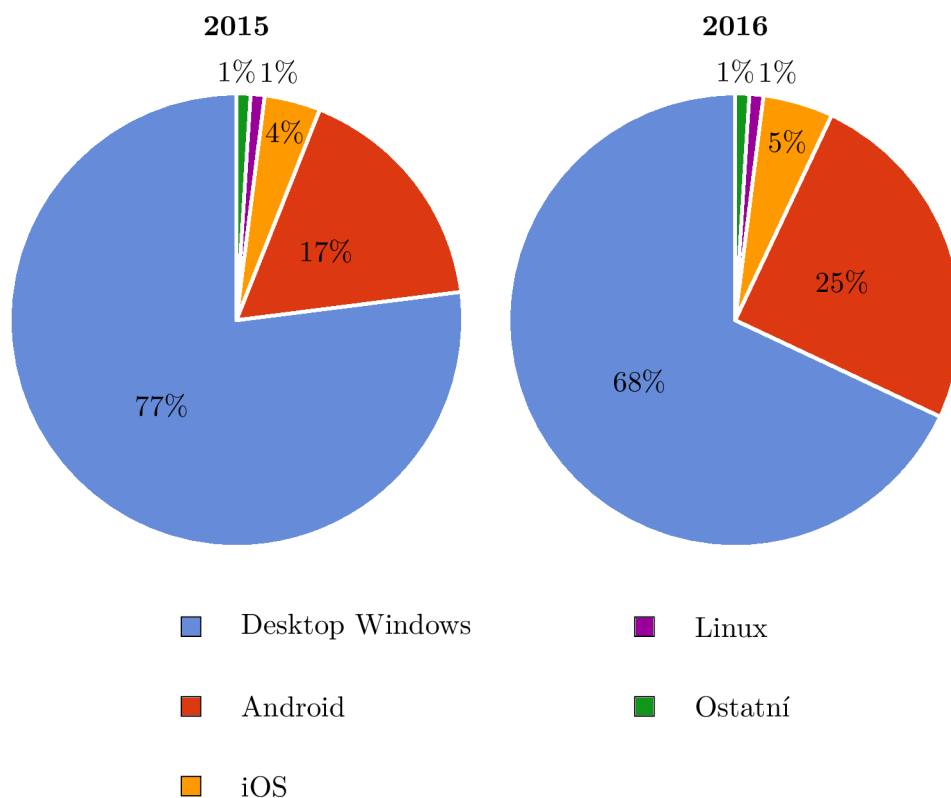
- Garance stejného vzhledu aplikace na všech zařízeních napříč různými rozlišeními.
- Lepší uživatelská interakce pomocí dotykového displeje.
- Lepší integrace v rámci mobilního systému, což vede k možnosti využít funkce, které by ve webové aplikaci nemohly být využívány.
- Uzpůsobení aplikace přímo na konkrétní typ zařízení, což vede k rychlejší práci a lepší optimalizaci aplikace.
- Efektivnější distribuce, kterou lze provést pomocí integrovaných obchodů s aplikacemi.

Z výše uvedených důvodů vyplývá, že nativní aplikace přináší uživateli více komfortu a umožňuje rychlejší práci, než v případě webové aplikace. Další z důvodů pro vývoj nativní aplikace pro web Happo.cz je stále se zvyšující počet uživatelů přistupujících na web z mobilního zařízení. Na grafech 2.2 je patrné, že oproti roku 2015 se přístup z mobilních zařízení značně zvýšil. V roce 2016 tak celých 31% návštěvníků přistupovalo na web právě z mobilního zařízení. Z grafů tedy plyne, že nezanedbatelná část návštěvníků využívá k přístupu na web právě chytrý telefon. Proto je vhodné přinést i této skupině uživatelů patřičný komfort v práci s aplikací.

Vedle potřeb samotného webu je tu také potřeba rozšíření stávající webové aplikace o prvky, které by v rámci webového rozhraní byly těžko proveditelné. Prvním rozšířením, je využít systém přátel, chat a GPS modul k možnosti zobrazit své aktivní přátele na mapě. Cílem rozšíření je podnítit lepší komunikaci mezi přáteli a zároveň pobídnout uživatele, aby spolu vytvářeli přátelské vztahy. Pro nové uživatele by rozšíření umožňovalo přepnout se do módu hledání přátel, kde by daný uživatel byl viditelný pro všechny ostatní uživatele v okolí. Mohl by jej tak kontaktovat libovolný uživatel a navázat s ním přátelský vztah. Navíc, díky propojení s chatem je tu možnost okamžité interakce mezi uživateli, přičemž není potřeba využívat aplikace třetích stran.

Dalším rozšířením je systém doporučování obsahu. Na základě určitých aktivit, které uživatel v minulosti provedl, by byl systém schopen doporučovat vhodná anime díla. Popis metod pro doporučování obsahu a výběr vhodné metody je popsán v kapitole 3.

Z výše uvedených důvodů a z potřeb dalšího rozvoje je patrné, že je vhodné pro web Happo.cz vyvinout nativní mobilní aplikaci, jejímž cílem by bylo dosáhnout větší mobility a dostupnosti.



Obrázek 2.2: Zařízení, ze kterých bylo přistupováno na web Happo.cz v roce 2015 a 2016

2.4 Analýza uživatelských požadavků

Webová aplikace (dále jen web) je značně rozsáhlá a nabízí spoustu funkcí, avšak jen část z nich je opravdu dennodenně využívána. Mobilní aplikace by uživatelům měla poskytnout jen ty nejoblíbenější a nejčastěji využívané služby a přitom si zachovat jednoduchost a přehlednost. Proto byla potřeba stanovit funkce, které mají být do mobilní aplikace zahrnuty. Web je navíc tematicky rozdělen na čtyři části, kterými jsou anime, postavy, soundtracky a vizuální novely. Proto prvotním úkolem bylo zjistit, jaké tematické části má aplikace obsahovat.

K získání potřebných informací ohledně zájmu o jednotlivé části webu, jsem využil zaznamenávání aktivit uživatelů. V rámci webu jsou ukládány všechny důležité aktivity, které uživatel v minulosti provedl. Tím pádem bylo poměrně jednoduché zjistit, které části webu jsou nejvíce využívány, a tudíž které by měly být do aplikace zahrnuty.

Z tabulky 2.1 vyplývá, že nejvíce navštěvovanými částmi jsou anime a postavy, které dohromady tvoří více jak 75% celkové návštěvnosti webu. Spolu s tematickými částmi tak byla zahrnuta i základní funkcionalita, která má být v rámci mobilní aplikace implementována. Především se jedná o vyhledávání v rámci tematických okruhů, zobrazení děl a jejich vzájemné vztahy.

Další etapou analýzy byl průzkum nejčastěji používaných funkcí. Jako základ mi opět posloužily záznamy aktivit uživatelů. Ty ovšem poskytují pouze početní zastoupení jednotlivých funkcí, což jsou nedostatečné informace. Například se z nich nedočteme, proč jsou určité funkce používány více než jiné nebo konkrétní účel, ke kterému uživatelé web používají. Z těchto důvodů vyplynulo, že je potřeba provést průzkum na vzorku uživatelů. Proto

Tématická část	Oblíbenost
Anime	62%
Postavy	23,7%
Soundtracky	12,4%
Vizuální novely	1,8%

Tabulka 2.1: Oblíbenost tématických částí webové aplikace.

pro získání detailnějších informací jsem oslovil 18 respondentů, s nimiž bylo provedeno krátké interview. Cílem bylo získat informace o nejčastěji prováděných aktivitách v rámci webu, preferencích uživatele a účelu, k jakému web používají. Mimo jiné měli respondenti během rozhovoru zvolit tři nejčastěji používané funkce a vysvětlit důvod svého výběru. Vedlejší částí interview pak bylo pozorování uživatelů při běžné práci s webem. Tato část byla zařazena hlavně z toho důvodu, abych získal představu o tom, jak uživatelé jednotlivé akce provádějí, případně jak se k nim dostávají. Vzor dotazníku použitého během interview je k nahlédnutí v příloze A.

Z průzkumu vyplynulo, že zhruba polovina dotázaných respondentů navštěvuje web jen několikrát týdně, převážně za účelem získávání informací o anime a udržování seznamů s oblíbenými položkami. Druhá polovina dotázaných uvedla, že web navštěvuje denně. Tato skupina se navíc zajímá o aktivitu svých přátel o vyjadřování názorů pomocí hodnocení a důkladně si udržuje seznamy s oblíbenými položkami a zhlédnutými epizodami. Z výpovědí respondentů bylo tedy zjištěno, že web převážně využívají jako zdroj informací, přičemž oceňují možnost vytváření a spravování seznamů oblíbených položek. Naopak se vyskytly výhrady na přehlednost a jednoduchost prováděných akcí, mezi něž patří označování epizod jako zhlédnuté a operace se seznamy oblíbených položek. Výhrady se týkaly převážně složitosti akcí, kdy je třeba přecházet mezi několika stránkami s čímž mnoho uživatelů má problémy. Tento fakt potvrdilo i následné pozorování uživatelů při běžné práci s webem.

Na základě vyhodnocení dotazníků, rozhovorů a pozorování byl sestaven seznam nejčastějších důvodů, proč uživatelé web navštěvují. Taktéž jsem získal představu o tom, co by v mobilní aplikaci uvítali, případně k jakým aktivitám by ji chtěli využívat. Po detailnější analýze výpovědí respondentů a záznamů aktivit uživatelů byl tak zvolen soubor nejčastěji používaných funkcí. Ten zahrnoval především vyhledávání, hodnocení, komentování, prohlížení anime i postav, chat, uživatelský profil se statistikami a správu oblíbených položek.

2.5 Podobné a existující aplikace

Před vznikem samotné aplikace bylo nutno zjistit, zda na trhu neexistují aplikace se stejným, či podobným zaměřením. Jako zdroj aplikací posloužil obchod Google Play a Apple Store. V době analýzy se v obchodech vyskytovalo pouze pět aplikací, které bylo možno označit za databázi anime.

- aniDBchan – je neoficiální aplikací známé webové databáze AniDB. Ovšem po nové aktualizaci webu a API je tato aplikace zcela nepoužitelná, neboť se nedokáže spojit s API.
- PocketMAL – je nativní aplikace pro další známou webovou anime databázi MyAnimeList. Aplikace disponuje především slušným vzhledem a možností vytvářet a spravovat své seznamy oblíbených titulů. Ovšem aplikace neposkytuje žádné informace o postavách ani autorech.

- Crunchyroll – slouží jako doplňková aplikace k americké webové stránce zabývající se anime, mangou, filmy a asijskou kulturou. Nespornou výhodou je možnost sledování jednotlivých epizod přímo v mobilním zařízení, avšak v japonském nebo anglickém znění. Naopak všudypřítomná reklama působí značně rušivě a snižuje přehlednost aplikace.
- Tenpencii – databáze poskytující základní informace o anime a manze. Aplikace přehledně zobrazuje informace jak u jednotlivých titulů, tak i v uživatelském profilu. Velkým mínusem je ovšem vyhledávání, které se provádí ve všech kategoriích zároveň a výsledky nejsou uspořádány. Taktéž použití velkých obrázků působí značně rušivě.
- Fukuro – malá anime databáze, která poskytuje pouze několik málo titulů. Na druhou stranu disponuje intuitivním ovládáním a přehledností. Předností je také možnost vytváření seznamů s oblíbenými položkami.

Problém je v tom, že ani jedna z databází neobsahuje českou lokalizaci a ve většině případů se jedná pouze o databázi anime titulů nebo mangy. Další informace, jako jsou postavy, autoři, nebo příbuzné tituly již aplikace nenabízí a je nutno je dohledávat jiným způsobem.

V důsledku nízkého počtu anime databází jsem následně provedl analýzu i filmových databází, které mají co do obsahu k anime databázím dost blízko. I když je na poli filmových databází situace značně lepší, stále jen zlomek analyzovaných aplikací je opravdu použitelných.

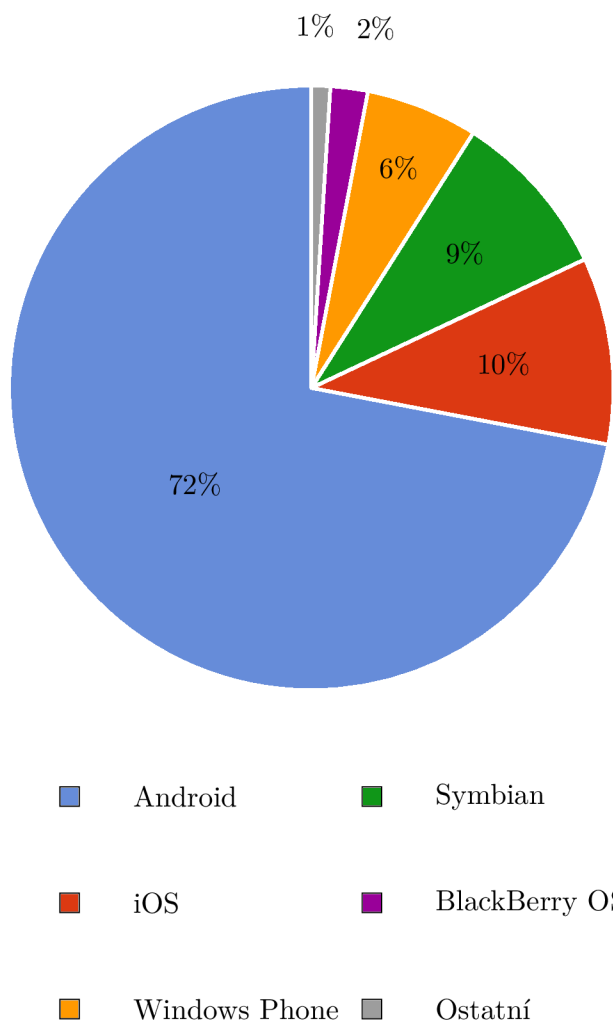
- FDb.cz – mobilní aplikace známé filmové databáze poskytuje sice dostatek informací o filmech a postavách, ovšem design aplikace je již značně zastaralý.
- ČSFD – je mobilní formou webové stránky CSFD.cz, která nabízí dostatečné množství informací o filmech a autorech zasazených do moderního designu. Ovšem vložená reklama v jistých místech znepřehledňuje aplikaci.

Z analýzy podobných aplikací vyplynulo, že na českém trhu neexistuje žádná anime databáze s českou lokalizací. I v případě, že by se uživatel přiklonil k používání zahraničních aplikací, výběr by byl velice omezený. Navíc pro náročnějšího uživatele by aplikace nejspíš neposkytovaly dostatek informací. Proto už jen samotné zobrazení anime titulů, postav a tvůrců by bylo značným přínosem pro české fanoušky anime.

2.6 Výběr mobilní platformy

Po prozkoumání všech nejrozšířenějších platform pro chytré telefony bylo nutno rozhodnout, kterou z nich zvolit pro vytvoření aplikace pro prezentaci informací ze světa anime. Pro začátek bylo třeba stanovit kritéria, která by měla platforma splňovat:

- Platforma musí být značně rozšířena.
- Musí být cenově dostupná pro mladší publikum.
- Platforma nesmí být zastaralá a její vývoj či existence v budoucnu nesmí být nejistá.
- Možnost publikovat výslednou aplikaci v online obchodech přičemž publikování nesmí být drahé.



Obrázek 2.3: Srovnání podílu platformech chytrých telefonů u tuzemských mobilních operátorů.

Pokud si projdeme jednotlivé platformy a aplikujeme na ně výše uvedená kritéria, zjistíme, že nejvhodnější platformou je Android. V případě iOS narazíme u druhého kritéria. Platforma iOS nepatří mezi nejlevnější a vezmeme-li v úvahu, že cílovou skupinou je mladší publikum a děti, iOS se tak jeví jako nevyhovující. V případě Windows Phone není splněno dokonce už prvním kritériem. Ač byla platforma v minulosti docela rozšířená, aktuálně upadá a její budoucnost je nejistá. Navíc, pokud se podíváme na graf 2.3 zjistíme, že opět převažující platformou mezi mobilními zařízeními je Android. Z těchto důvodů jsem tedy jako cílovou platformu pro vytvoření aplikace zvolil Android.

2.6.1 Verze API

Než se však přistoupí k návrhu aplikace, je nutné zvolit, pro jakou minimální verzi Android API bude aplikace vyvíjena. Od volby API se především odvíjí grafický návrh a následná implementace. Ve starších verzích totiž chybí spousta moderních grafických prvků, čímž by bylo nutné větvit zobrazení pro různé verze systému. Dle tabulky 2.2 je patrné, že vývoj pro Android 2.x.x se již nevyplatí, protože je používá jen zanedbatelná část uživatelů.

Verze	Název	Verze API	Rozšířenost
2.3.3 - 2.3.7	Gingerbread	10	0.9%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.9%
4.1.x	Jelly Bean	16	3.5%
4.2.x	Jelly Bean	17	5.1%
4.3	Jelly Bean	18	1.5%
4.4	KitKat	19	20.0%
5.0	Lollipop	21	9.0%
5.1	Lollipop	22	23.0%
6.0	Marshmallow	23	31.2%
7.0	Nougat	24	4.5%
7.1	Nougat	25	0.4%

Tabulka 2.2: Tabulka zobrazující nejrozšířenější verze Android API a jejich zastoupení na trhu. Tabulka nezahrnuje ty verze API, které mají menší zastoupení jak 0.1%.[\[1\]](#)

Já se rozhodl podporovat co možná nejširší možné rozpětí API verzí, a proto jako minimální verzi jsem zvolil Android 4.0 Ice Cream Sandwich verze API 14. Podpora od této verze tak pokryje téměř 99% všech zařízení.

Nicméně chceme-li aplikaci vyvíjet pod nejmodernější verzí API a přesto ji spouštět i na zařízeních se starší verzí API, je nutné do projektu zahrnout knihovny Android Support Library, které zajišťují tzv. dopřednou kompatibilitu. V opačném případě by nové prvky na starších zařízeních nefungovaly.[\[17\]](#)

Kapitola 3

Metody doporučování obsahu

3.1 Systém doporučování obsahu

Systém doporučování obsahu neboli *Recommender System* (dále jen RS), je technologie, která dokáže automatizovat proces doporučování položek (jako jsou filmy, články, obrázky, zboží v internetových obchodech apod.), které mohou být pro konkrétního uživatele zajímavé. [18]

RS jsou hojně rozšířené především na internetu, kde jsou využívány pro doporučování produktů a služeb uživatelům. Jednou z nejrozšířenějších funkcí RS je pomáhat uživatelům najít přesně ty informace, o které by mohli mít zájem, a tudíž které vyhovují jejich preferencím. Jako příklad může posloužit třeba systém filmové databáze, které uživatel předá informaci o zhlédnutých filmech. RS mu na základě těchto informací a informací od ostatních uživatelů doporučí filmy, které odpovídají jeho oblíbeným žánrům.

Dalším důležitým přínosem RS je napomáhání zvyšování zisků. To se týká především internetových obchodů, kde RS jsou nedílnou součástí marketingu a cílené reklamy. [18]

3.1.1 Definice problému doporučování obsahu

Mějme množinu uživatelů U a množinu doporučovaných položek S . Každý uživatel $u \in U$ je definován uživatelským profilem, který obsahuje charakteristické informace (např. věk, pohlaví, zájmy apod.) o daném uživateli. V nejjednodušším případě obsahuje pouze identifikátor uživatele. Obdobně je definována i množina S , kde každá položka obsahuje charakteristické vlastnosti, do kterých musí být minimálně zahrnut identifikátor položky. Pro představu množina S může reprezentovat například množinu filmů, kde každá položka $s \in S$ obsahuje charakteristické rysy v podobě identifikátoru filmu, názvu filmu, roku natočení, žánru apod. V mnoha aplikacích jsou často množiny U a S značně rozsáhlé.

RS lze pak definovat jako funkci p , která měří prospěšnost položky $s \in S$ pro uživatele $u \in U$. Vznikne tak množina $p : U \times S \rightarrow R$, kde R je úplně uspořádaná množina ohodnocení (nezáporná celá či reálná čísla s určitým rozsahem.) [18]

Výstup RS je generován na základě vstupu poskytnutého systémem. Výstup může být vyjádřen jako předpověď (číslo na číselné stupnici reprezentující zájem uživatele u o položku s), nebo jako doporučení (seznam N položek seřazených dle relevance, váhy či jiného kritéria). [15]

3.1.2 Dělení systémů doporučování

RS systémy je možno obecně rozdělit do několika kategorií, z nichž každá je postavena na jiných vstupních datech a jiném způsobu jejich zpracování. V současné době se běžně používají RS z následujících kategorií:

- **Doporučení na základě obsahu (Content-based recommender systems)** bere v úvahu vlastnosti položek ve vztahu k preferencím uživatele. Algoritmy jsou zaměřeny především na vyhodnocování preferencí uživatele a filtrování položek, které těmto preferencím vyhovují. Blíže popsáno v podkapitole 3.2.
- **Kolaborativní filtrování (Collaborative filtering recommender systems)** doporučuje uživateli položky na základě historie ohodnocených položek ostatními uživateli se stejným zájmem. Jednotlivé metody jsou uvedeny v podkapitole 3.3.
- **Hybridní systémy (Hybrid recommender systems)** obvykle kombinují metody z výše popsaných kategorií. Kombinací metod se tak kompenzují jejich nevýhody. Blíže rozvedeno v podkapitole 3.4.

3.2 Doporučení na základě obsahu

3.2.1 Základní princip

Doporučení na základě obsahu (*Content-based-recommendations*) provádí doporučení položky na základě předpovídání užitečnosti položky pro konkrétního uživatele. Při tom se vychází z podobných položek, které uživatele v minulosti zaujaly. Z tohoto hlediska je položka obvykle reprezentována sadou vlastností. Například u doporučování filmů může být film zastoupen vlastnostmi, jakými je konkrétní herec, režisér, žánr atd. Zájmy a preference uživatele jsou zastoupeny podobnou sadou vlastností v podobě uživatelského profilu. Vlastnosti profilu mohou být získány explicitně (např. prostřednictvím dotazníku) nebo implicitně z uživatelova chování v průběhu času. Doporučení jsou tak utvářeny porovnáváním stejné sady vlastností uživatelova profilu s kandidátními položkami. Uživateli se následně doporučí nejvhodnější nebo nejvíce podobné položky.[18]

3.2.2 Reprezentace položky

Položky, které lze doporučit uživateli, jsou často uloženy v databázové tabulce. Tabulka 3.1 ukazuje jednoduchou databázi s několika záznamy, které popisují tři restaurace.

ID	Jméno	Kuchyně	Obsluha	Cena
10001	Mike's Pizza	Italská	Bar	Nízká
10002	Chris's Cafe	Francouzská	Stůl	Střední
10003	Jacques Bistro	Francouzská	Stůl	Vysoká

Tabulka 3.1: Jednoduchá databáze restaurace.

Názvy sloupců jsou vlastnosti nebo také atributy restaurace. Jedinečný identifikátor (ID v tabulce 3.1) umožňuje od sebe odlišit jednotlivé záznamy a zároveň slouží jako klíč k získání dalších atributů záznamu. Tabulka 3.1 je příkladem strukturovaných dat s malým počtem atributů, kde každá položka je reprezentována stejnou množinou vlastností. Díky tomu je známa množina hodnot, kterých jednotlivé atributy nabývají.

Volný nestrukturovaný text vytváří celou řadu komplikací při učení uživatelského profilu. Profil uživatele může například naznačovat, že se uživateli s 80% pravděpodobností líbí francouzská restaurace. To vyplývá z uživatelova ohodnocení dřívějších restaurací. Nicméně nestrukturované texty jsou obvykle jedinečné. Tudiž kdyby byly restaurace popsány textem jako např. „Okouzlující kavárna s ochotným personálem a výhledem na řeku“, neexistovala by žádná zpětná vazba.

Z toho důvodu se používají techniky, které dovedou převést nestrukturovaný text do strukturované podoby. Tento proces se nazývá *vyplývání*. Jeho cílem je nalézt společný termín pro syntakticky podobná slova (např. *compute*, *computation*, *computer*). Hodnota proměnné asociovaná se získaným termínem je reprezentována reálným číslem, které představuje relevanci nebo důležitost. Termíny jsou následně seřazeny podle váhy. Problém tohoto přístupu je však ten, že nezachycuje kontext, ve kterém se slovo používá. Ztrácí se tak vazby mezi slovy v rámci textu. [20]

3.2.3 Uživatelský profil a jeho učení

Uživatelský profil je používán ve většině RS. Může se sestávat z několika různých typů informací:

- **Preference uživatele** – popisují typy položek, které uživatele zajímají. Existuje mnoho způsobů této reprezentace, avšak jednou z nejčastějších je funkce, která pro každou položku určí pravděpodobnost, s jakou se uživateli zamlouvala.
- **Historie aktivit uživatele s RS** – jedná se o ukládání položek a aktivit, které nad nimi uživatel provedl (např. zda danou položku koupil nebo ohodnotil).

Existuje několik způsobů, jak využít historie aktivit uživatele. Systém může buď jednoduše zobrazit naposledy navštívené položky a usnadnit tak uživateli návrat k těmto položkám. V opačném případě může systém vyfiltrovat položky, se kterými se uživatel již setkal a tím pádem mu je znovu nenabízet. Dalším důležitým způsobem, jak využít historie je využití těchto položek jako vstup pro RS. Takové položky pak slouží jako trénovací data, z nichž je vytvořen uživatelský vzor k naučení.

Vytvoření vzoru preferencí uživatele z jeho historie je forma klasifikačního učení. Trénovací data jsou rozdělena do kategorií, např. binární kategorie (položky, které se uživateli líbí a položky, které se mu nelíbí). Tohoto určování se dosahuje buď pomocí explicitní zpětné vazby (kde uživatel přímo zadá své hodnocení) nebo implicitní zpětné vazby (sledování interakcí uživatele s položkami). Například pokud uživatel koupí nějakou položku, je to znamení, že se mu líbí. Naopak, pokud položku vrátí, je to znamení, že se mu daná položka nezamlouvá. Obecně platí, že implicitní metoda může sbírat velké množství dat s určitou nejistotou, zda se položka uživateli skutečně líbí. Naproti tomu, pokud uživatel explicitně ohodnotí položku, dochází k přesnějšímu učení. Problém je však v tom, že uživatelé mají tendenci poskytovat explicitní zpětnou vazbu jen u malého počtu položek, se kterými interagují. [20]

3.2.4 Metody a algoritmy

Decision tree

Vytváří rozhodovací stromy na základě rekurzivního dělení trénovacích dat do podskupin. Dělení provádí do té doby, dokud podskupiny neobsahují pouze instance jednotlivých tříd.

Dělení je realizováno testováním na některou vlastnost – v souvislosti s textovou klasifikací jde typicky o testování přítomnosti či absence jednotlivých slov či slovních spojení. Každý uzel stromu tak představuje rozhodování podle jedné vlastnosti objektu. [20]

Nearest Neighbor Methods

Algoritmus nejbližších sousedů jednoduše ukládá všechna trénovací data, ať už explicitně či implicitně označených položek. Aby bylo možné klasifikovat novou neoznačenou položku, musí ji algoritmus nejprve srovnat se všemi doposud uloženými položkami pomocí funkce podobnosti, a určit tak nejbližší sousedy. Výsledná kategorie nebo numerické ohodnocení pro doposud neoznačené položky mohou být odvozovány ze vzorů již označených položek. [20]

Rocchio's Algorithm

Obecnou zásadou tohoto algoritmu je umožnit uživatelům ohodnotit navrhované výsledky doporučujícího systému s ohledem na jejich informační potřebu. Tato forma zpětné vazby může být následně použita k postupnému upřesnění uživatelského profilu nebo k zdokonalení procesu doporučování. [23]

Naive Bayes

Naive Bayes je založen na pravděpodobnostním přístupu k induktivnímu učení¹. Vytváří pravděpodobnostní model založený na dřívějších datech. [23]

3.2.5 Výhody a nevýhody

Hlavní výhodou doporučování na základě obsahu je, že může začít doporučovat, jakmile dostane k dispozici informace o položkách. To znamená, že systém doporučování si dokáže vystačit pouze z historií aktivit uživatele.

Další výhodou je: [23]

- Průhlednost – Může být poskytnuto průhledné vysvětlení, na základě čeho byla doporučena konkrétní položka. Pomocí toho se může uživatel rozhodnout, jestli má doporučení věřit nebo ne. Oproti tomu kolaborativní systémy fungují spíše jako černá skříňka, protože mohou poskytnout pouze vysvětlení, že neznámému uživateli s podobným vkusem se položka líbila.
- Doporučování nových položek – Systémy na bázi doporučování dle obsahu jsou schopny doporučit i položky, které ještě nikdo neohodnotil či nenavštívil. Je to důsledek toho, že tyto systémy netrpí na problém prvního hodnotitele, který postihuje kolaborativní filtrování. Kolaborativní systémy tak nedovedou doporučit novou položku, kterou neohodnotil dostatečný počet uživatelů.

Současně však tyto výhody vedou k několika nevýhodám: [12]

- Položky a její vlastnosti musí být strojově rozpoznatelné.

¹Indukce jde opačnou cestou než dedukce – od jednotlivého k obecnému. Stanovujeme obecná pravidla na základě konkrétních případů. Petr má auto – Karel má auto – všichni muži mají auto... (to ovšem platí, dokud nezjistíme, že Standa auto nemá). Induktivní úsudky platí vždy pouze s určitou pravděpodobností, nikdy ne na 100%. [3]

- Nelze filtrovat položky na základě ohodnocení kvality, stylu nebo názoru. Vzhledem k tomu, že nejsou brány v úvahu zkušenosti ostatních uživatelů, systém nemůže provádět žádné posuzování kvality pro jednotlivé položky.
- Absence osobních doporučení. Vzhledem k tomu, že systém nebere v úvahu hodnocení či názory ostatních uživatelů, doporučení jsou tak založena jen na vlastnostech položky.
- Žádné serendipita² položky.
- Synonyma³. Pokud existují slova se stejným významem, ale každé z nich se jinak píše, systém je rozpozná jako dvě nezávislá slova a nebude schopen mezi nimi najít podobnost.

3.3 Kolaborativní filtrování

3.3.1 Základní princip

Kolaborativní filtrování (*Collaborative filtering* dále jen CF) je jedna z nejstudovanějších a nejrozšířenějších metod používaných v oblasti doporučování obsahu v praxi. Klíčovou charakteristikou CF je, že předpovídá užitečnost položek pro konkrétního uživatele na základě dříve ohodnocených či zakoupených položek podobně smýšlejícími uživateli. Často se využívá znalosti interakce dat uživatel-položka, kdy se ignorují atributy obou množin a soustředí se pouze vztah.[18]

V běžném životě lidé v mnoha případech volí různé možnosti, aniž by jim plně rozuměli, či je znali. V těchto případech tak spoléhají na názory a doporučení jiných (známých) lidí nebo lidí, jejichž názor respektují. CF systémy této myšlenky plně využívají a snaží se konkrétnímu uživateli nabídnout názor ostatních uživatelů.

Existují různé přístupy CF:[8]

- **Memory-based** – používají heuristiky, které předpovídají hodnocení na základě dříve ohodnocených položek ostatními uživateli.
- **Model-based** – cílem je za použití dolování dat nalézt takový vzor trénovacích dat, na základě kterého bude model schopný předpovídat hodnocení.

Oba tyto přístupy následně procházejí třemi etapami doporučování:[8]

1. Analýza a sběr dat – Jednotlivá hodnocení je nejprve třeba shromáždit a optimalizovat pro další zpracování.
2. Seskupení uživatelů – Uživatelé se stejným či podobným vkusem jsou seskupeni a na jejich základě je vypočteno doporučení za pomoci míry podobnosti. K určení míry podobnosti může sloužit například Pearsonův korelační koeficient.
3. Generování předpovědi – Jakmile byli uživatelé seskupeni dle podobnosti, systém je použije k výpočtu předpovědi pro cílového uživatele.

²Serendipita označuje „šťastnou náhodu“ či „příjemné překvapení“. Jedná se o schopnost systému doporučit uživateli překvapivě zajímavou, ale nepředpokládanou položku. [12]

³Synonyma jsou slova nebo slovní spojení se vzájemně stejným významem. Můžeme tak slovo jeho synonymem nahradit, aniž bychom výrazněji změnili význam věty. [4]

3.3.2 k-Nearest Neighbor

Metoda *k-Nearest Neighbor* (dále jen k-NN) umožňuje na základě všech vztahů uživatel-položka generovat předpovědi přímo. Algoritmus funguje na principu vyhledání podobných uživatelů nebo položek (tzv. sousedé). Pokud takové položky/uživatele nalezne, využije jejich hodnocení k předvídání preferencí cílového uživatele. Přitom se používá klasifikace k-nejbližších sousedů pro předpověď uživatelova ohodnocení či sklonu k nákupu, dle výpočtu korelace mezi profilem cílového uživatele a profilem ostatních uživatelů.[18]

User-based k-NN

Typický k-NN algoritmus se skládá ze dvou fází: fáze formování sousedů a fáze doporučování. V první fázi se porovnávají záznamy aktivit cílového uživatele (též návštěvník) s historickými záznamy T ostatních uživatelů s cílem najít top k uživatelů, kteří mají podobný vkus. Mapování záznamů návštěvníka na sousedy může být založeno na podobnosti hodnocených položek, zájmu o podobný obsah nebo nákupu podobných položek. Necht záznam (nebo profil) cílového uživatele je u (reprezentován jako vektor) a záznam jiného uživatele je v , kde $v \in T$. Nejvíce podobných záznamů k uživateli u je považováno za sousedy u . Podobnost mezi cílovým uživatelem u a sousedním uživatelem v lze vypočítat pomocí Pearsonova korelačního koeficientu:

$$sim(u, v) = \frac{\sum_{i \in C} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in C} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in C} (r_{v,i} - \bar{r}_v)^2}}, \quad (3.1)$$

kde C je množina položek, které jsou ohodnoceny oběma uživateli u a v , $r_{u,i}$ a $r_{v,i}$ jsou ohodnocení položky i cílovým uživatelem u a potencionálním sousedem v a \bar{r}_u a \bar{r}_v je průměrné hodnocení uživatele u a v . Na základě vypočtené podobnosti jsou vybráni nejvíce podobní uživatelé (sousedé). Taktéž je běžné odfiltrovat sousedy s podobností menší než stanovený práh, aby se zabránilo předpovědi na základě příliš vzdálených sousedů od uživatele u . [18]

Item-based k-NN

Problém *user-based varianty* je nedostatek škálovatelnosti. Algoritmus vyžaduje v reálném čase porovnání cílového uživatele se všemi záznamy všech ostatních uživatelů za účelem vytvoření předpovědi. Tento problém však řeší varianta *item-based*, která předem vypočítá všechny možné páry položek na základě jejich podobnosti napříč všemi uživateli. Pro výpočet míry podobnosti se typicky používá upravená kosinová podobnost:

$$sim(i, j) = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u)(r_{u,j} - \bar{r}_u)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_u)^2}}, \quad (3.2)$$

kde U je množina všech uživatelů, i a j jsou položky, $r_{u,i}$ je ohodnocení uživatele $u \in U$ položky i a \bar{r}_u je průměrné hodnocení uživatele. [18]

3.3.3 Association rules base prediction

Algoritmus ze získaných asociačních pravidel sdružuje vztahy mezi položkami. Jinak řečeno, na základě obsahu uživatelova košíku, mu jsou doporučovány další položky, které by tam mohl ještě přidat. Tento přístup se stal velice populární a zároveň se ukázala jeho užitečnost v mnoha oblastech. [22]

Použití asociačních pravidel pro doporučování je zcela přirozené. Zakoupené položky každého z uživatelů lze jednoduše považovat za transakce. Jednotlivá asociační pravidla pak mohou být získána z transakcí všech uživatelů a použita pro předpověď či klasifikaci. Zejména levá strana pravidla může být použita k předpovědi pravé strany pravidla.[18]

TID	Položky
1	{chleba, mléko}
2	{chleba, pleny, pivo, vejíčka}
3	{mléko, pleny, pivo, cola}
4	{chleba, mléko, pleny, pivo}
5	{chleba, mléko, pleny, cola}

Tabulka 3.2: Příklad transakcí spotřebního koše.

Příkladem může být velké množství údajů o nakupujících, které jsou každý den shromažďovány. Tabulka 3.2 ilustruje příklad takových údajů, běžně známých jako pod pojmem spotřební koš. Každý řádek v této tabulce odpovídá transakci, která obsahuje jedinečný identifikátor TID a sadu zákaznickem zakoupených položek. Obchodníci se často zabývají analýzou těchto dat, protože se z nich mohou dozvědět chování svých zákazníků. Získané informace mohou následně využít v propagačních akcích, řízení zásob a přizpůsobování vztahů k zákazníkům.

Vztahy položek mohou být reprezentovány ve formě asociačních pravidel nebo v sadě častých položek. Např. následující pravidlo může být extrahováno z dat uvedených v tabulce 3.2:

$$\{pleny\} \longrightarrow \{pivo\}$$

Pravidlo ukazuje, že existuje silný vztah mezi prodejem plenek a piva, protože mnoho zákazníků, kteří kupují pleny, koupili také pivo. Na základě tohoto pravidla lze následně uživateli, který má v košíku pleny doporučit pivo, protože je velká pravděpodobnost, že o něj bude mít zájem.[26]

Ovšem problém systému na bázi asociačních pravidel je ten, že systém má s doporučováním potíže, pokud je datová množina příliš řídká, což je často se vyskytující jev v CF systémech. Důvodem řídkosti může být stav, kdy uživatel navštívil/ohodnotil jen velmi malý počet dostupných položek. Proto je pro algoritmus často obtížné nalézt množinu společných položek ve více uživatelských aplikacích.[18]

3.3.4 Matrix factorization

Myšlenka faktorizace matice (*Matrix Factorization*, dále jen MF) je mnohem mladší než výše popsané metody. Hlavní myšlenkou MF je rozložit matici M na několik matic $M = F_1 F_2 \dots F_n$, kde n může být libovolné číslo, obvykle 2 nebo 3.[18]

Jako příklad si můžeme uvést situaci, kterou řešila firma Netflix. Mějme doporučovací systém, který má 490 000 uživatelů a 17 700 položek reprezentovaných v matici R . Taková matice má přibližně 8,5 miliardy vstupů (hodnocení zanechaných uživateli u položek). Přitom existuje pouze 100 milionů skutečně zanechaných hodnocení. Zbýlých 8,4 miliardy jsou nuly. Takto reprezentovaná matice má tedy 99% řídkost. Každá položka či uživatelský profil může být navíc reprezentován dalšími informacemi. Stejně tak uživatelé mohou mít své preference a zajímat se o položky spadající do určitého okruhu informací.[13]

V praxi to znamená vytvoření modelu, který popisuje data s určitým počtem parametrů, a jejich zpětnou kombinací se vracíme k původní matici. Tento model se nazývá *Singular Value Decomposition* (SVD). Předpokládejme, že uživatelské ohodnocení konkrétní položky je výsledkem součtu zájmů jednotlivých parametrů položky. Počet takových parametrů omezíme například na 20 položek, kdy každá položka bude popsána z 20 různých hledisek. Stejně tak každý uživatel bude mít 20 různých zájmů o tyto parametry. Vynásobením každého parametru položky na každé úrovni zájmu uživatele získáme celkový zájem uživatele o tuto položku.

Součet čísel použitých v tomto příkladě je $20 * (490000 + 17700)$, což je asi 10 milionů, neboli 800 krát menší počet, než původních 8,5 miliardy. To je hlavní myšlenkou MF.[13]

3.3.5 Výhody a nevýhody CF

Většina významných výhod CF plyne z nevýhod doporučování na základě obsahu:[11]

- CF nevyžaduje žádné kontextuální informace ani o položce, ani o uživatelském profilu, které jsou strojově rozpoznatelné. Metody CF využívají pouze hodnocení a dodatečné informace o položkách či profilech neberou v úvahu.
- Systémy dovedou provádět posouzení kvality, stylu nebo názoru dle zkušeností ostatních uživatelů.
- Významnou výhodou CF systémů je, že mohou vytvářet osobní doporučení, protože berou v úvahu zkušenosti jiných uživatelů, přičemž doporučení jsou založena právě na těchto zkušenostech.
- Další výhodou je schopnost systému z pozorování chování podobně smýšlejících uživatelů navrhnout serendipita položky.

CF systémy mají ovšem i své nedostatky:[25]

- Studený start (cold start). Problém nastává, pokud do systému vstoupí nový uživatel nebo položka. Pro takového položku je pak obtížné najít podobné položky, protože není k dispozici dostatek informací. Nové položky tak nelze doporučit, dokud ji nějaký uživatel neohodnotí. A naopak, pro tvorbu spolehlivých doporučení systém potřebuje, aby uživatel ohodnotil dostatečný počet položek.
- Problém řídkosti (sparsity problem). V praxi se mnoho komerčních RS používá pro vyhodnocení velké množiny produktů. Pokud však uživatelé hodnotí pouze malou podmnožinu všech položek, dochází k tomu, že matice použita pro CF bude velmi řídká. Výsledkem je nepřesná identifikace sousedů a omezení kvality doporučení.
- Problém škálovatelnosti (scalability problem). Pokud v systému rychle roste počet uživatelů či položek, nastává u tradičních CF algoritmů problém s výpočetními zdroji. Systém pro velký počet položek potřebuje velký výpočetní výkon. Stejně tak mnohé systémy musí reagovat na on-line požadavky od všech uživatelů, což vyžaduje vysokou škálovatelnost systému.
- Šedé a černé ovce (gray and black sheep). Tzv. šedá ovce je uživatel, jehož názory se výrazně liší od skupiny ostatních uživatelů. Opakem jsou tzv. „černé ovce“, což je skupina lidí, jejichž výstřední vkus dělá doporučování prakticky nemožné.

- Šilinkové útoky (shilling attacks). Tento problém nastává, pokud mnoho uživatelů dá položce spoustu pozitivních doporučení z důvodu vlastního prospěchu a naopak negativní doporučení konkurenci.

3.4 Hybridní systémy

Doporučovací systémy založené na obsahu jsou poměrně přesné v poskytování předpovědí, když jsou k dispozici dodatečné informace o položkách, avšak nedaří se jim generovat serendipita a personalizovaná doporučení. Naopak CF systémy berou v úvahu chování lidí a dle podobně smýšlejících sousedů dovedou navrhnout doporučení, nicméně zase neberou v úvahu jakékoliv dodatečné informace o položkách. Proto z důvodu překonání těchto nevýhod byly vyvinuty hybridní systémy. Klíčovou myšlenkou je poskytnout doporučení na základě informací u položek i chování uživatelů. Hybridní přístup tak umožňuje kombinovat ty nejlepší vlastnosti z obou systémů.[14]

Hybridní systémy lze rozdělit do sedmi kategorií:

- Weighted hybrid – Kombinuje výsledky jednotlivých komponent (část systému, ze kterých je hybridní systém složen) pomocí lineární formulace.
- Switched hybrid – Vybírá nejvíce přesvědčivé komponenty.
- Mixed hybrid – Kombinuje hodnocení z různých algoritmů, z nichž sestavuje jedinou hodnotu.
- Feature combination hybrid – Jedna komponenta přidává funkčnost ke zdroji druhé.
- Feature augmentation hybrid – Více přizpůsobivá varianta Feature combination hybrid.
- Cascade hybrid – Jedna komponenta vytváří ohodnocení, které druhá komponenta využije.
- Meta-level hybrid – Výstup z jedné komponenty tvoří vstup pro druhou komponentu.

Weighted, switched a mixed systémy jsou nejsnadněji realizovatelné, protože mohou použít algoritmy pro generování doporučení, přičemž výsledky jsou spojeny pomocí jednoduchého vzorce. Ostatní systémy vyžadují hlubší porozumění algoritmům, neboť jsou vzájemně propojené v jeden celek (podobně jako firma Netflix kombinuje více než 200 algoritmů).

Hlavní výhodou hybridních systémů je možnost doporučovat jak na základě obsahu, tak i uživatelských zkušeností, čímž lze do doporučení zahrnout 100% položek. Na druhou stranu, vývoj těchto systémů je značně komplikovaný a vyžaduje dodatečné ladění, aby se dosáhlo dobrého vybalancování výsledků.[16]

Kapitola 4

Návrh mobilní aplikace

V této kapitole se budu podrobněji zabývat návrhem grafického rozhraní mobilní aplikace. Popíši způsob komunikace mezi serverem a mobilní aplikací a zaměřím se na výběr algoritmu pro doporučování obsahu a návrh jeho zakomponování do systému.

4.1 Uživatelské rozhraní

4.1.1 Návrh matice objektů a akcí

Před samotným grafickým návrhem jednotlivých obrazovek aplikace, bylo nutné nejprve stanovit, jaké akce by měly jednotlivé obrazovky poskytovat.

	Vyhledávání	Prohlížení seznamů s tituly, postavami a uživateli	Prohlížení informací	Hodnocení	Komentování	Operace se seznamy oblíbených položek	Chatování
Profil anime	×	×	×	×	×	×	
Profil postavy	×	×	×	×	×	×	
Profil autora	×	×	×	×	×		
Profil uživatele	×	×	×				
Seznamy oblíbených titulů	×	×				×	
Vyhledávání	×	×					
Chat	×	×					×

Tabulka 4.1: Matice zobrazuje akce pro každou obrazovku. Z tabulky vyplývá, že se v aplikaci budou vyskytovat objekty (Vyhledávání nebo Prohlížení seznamů s tituly, postavami a uživateli), které sdílí mnoho společných akcí a tudíž se uživatel nebude muset učit nové akce, ale bude moci použít již osvojené z jiných objektů. Naproti tomu se tu vyskytují i specifické akce, které jsou vázány na specifické objekty. Příkladem takové specifické akce je chatování, které se váže pouze na objekt chat.

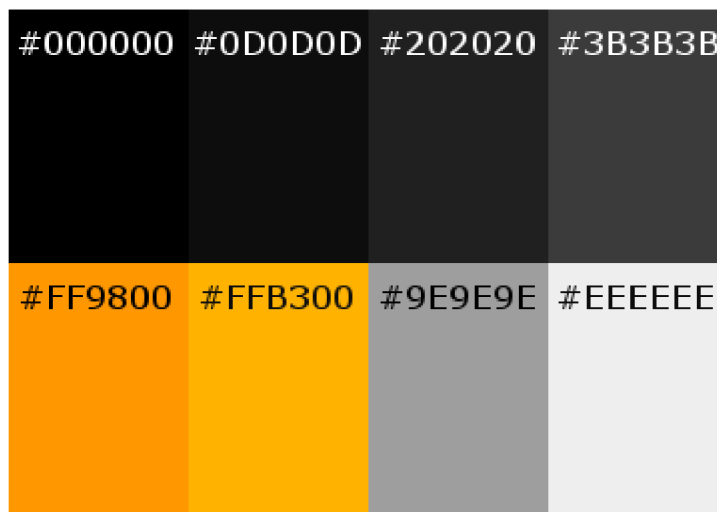
Pro tento účel byl využit model matice, která zobrazuje vztah mezi objekty a akcemi. Jinak řečeno, matice zobrazuje hlavní obrazovky aplikace a určuje, jaké akce se mohou v rámci obrazovky provádět. Na základě tohoto modelu lze určit, zda výsledná aplikace bude pro uživatele těžká či lehká na naučení.

Vytvoření matice objektů/akcí nám umožní vizualizovat, jak jednoduchá či složitá je naše aplikace. Přičemž v našem případě objekt reprezentuje obrazovku aplikace. Velká matice udává, že existuje více konceptů k naučení. Vysoká matice poukazuje na mnoho objektů v aplikaci a široká matice na mnoho akcí k naučení. Matice nám také ukazuje, jak konzistentní či nekonzistentní je náš model. Neboli, jak snadné je pro uživatele přenést to, co se naučili v jedné části aplikace do druhé. Čím má matice větší hustotu, tím snadnější bude aplikace pro uživatele k naučení. [10]

4.1.2 Návrh obrazovek a barevné rozvržení

Během rozvržení komponent bylo třeba dbát na přívětivost a intuitivnost rozhraní. Komponenty proto byly do obrazovky umístěny tak, aby uživatele jasně vyzývaly a naváděly k akci, kterou poskytují. Aby byla aplikace pro uživatele co možná nejvíce intuitivní, bylo třeba její ovládání co nejvíce přiblížit k samotné webové službě Happo.cz.

S tím souvisí i barevné rozložení. Web je situován do kontrastních barev, kde prvky jako jsou okna, pozadí, seznamy a nadpisy jsou situovány do barev odstínů modré, tmavě modré a odstínů šedé. Oproti tomu písmo, nadpisy a zvýraznění je situováno do bílé a oranžové barvy. Podobné barevné schéma jsem proto zvolil i pro mobilní aplikaci. Jako výchozí barvu (pro pozadí a velké plochy) jsem vybral tmavé odstíny šedé. Barva běžného textu nebyla vybrána přímo bílá, neboť bílá je značně výrazná barva a ve spojení s tmavým pozadím by byl kontrast příliš vysoký. V důsledku toho by došlo k přílišnému záření textu, což by působilo rušivě a nepříjemně. Proto jako barva pro běžný text byla zvolena světle šedá, která je ve spojení s tmavým podkladem dobře čitelná a přitom nepůsobí rušivým dojmem. Pro zvýraznění prvků a panelu nástrojů jsem zvolil odstíny oranžové s černou barvou písma.



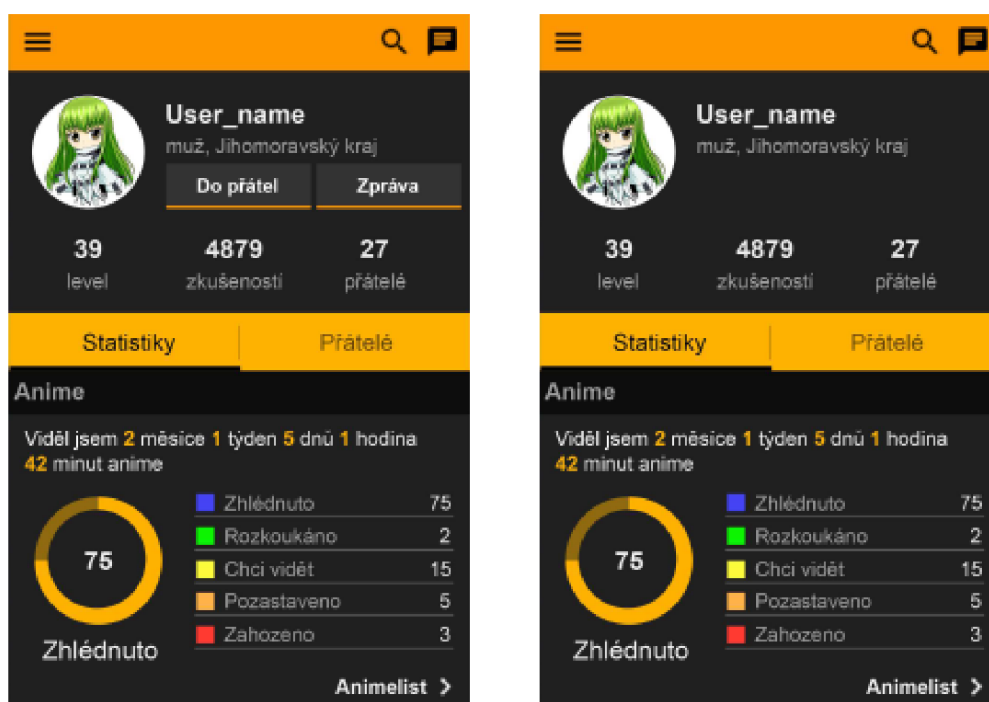
Obrázek 4.1: Vzorčky barev použitých v aplikaci.

Pro rozložení prvků jsem využil postupy tzv. *Material Designu*. Jedná se o postupy, jak navrhovat jednotný design uživatelského rozhraní. Myšlenka je taková, že všechny prvky

uživatelského rozhraní reagují na dotek velmi intuitivně a za pomoci animací a jasných barev tak vynikají důležité informace. Tento přístup vede k mnohem intuitivnějšímu a přehlednějšímu návrhu, díky kterému se uživatel dokáže v aplikaci mnohem lépe a rychleji orientovat. [2]

Zobrazení profilu uživatele

Aplikace bude dovolovat přístup i pro nepřihlášené uživatele, kteří pouze hledají informace. Pro takové uživatele bude ovšem funkčnost omezena pouze na vyhledávání a prohlížení titulů, postav a autorů. V případě, že se uživatel do aplikace přihlásí, bude propojen se svým účtem na Happo.cz, díky čemuž bude mít přístup ke svému profilu, seznamu oblíbených položek, chatu atd. Obrazovka zobrazující profil uživatele bude navržena tak, aby přehledně zobrazovala ty nejdůležitější informace. K dosažení těchto cílů bude využito dashboardů a grafů, které dovolují zobrazit mnoho podstatných informací v přehledné formě.

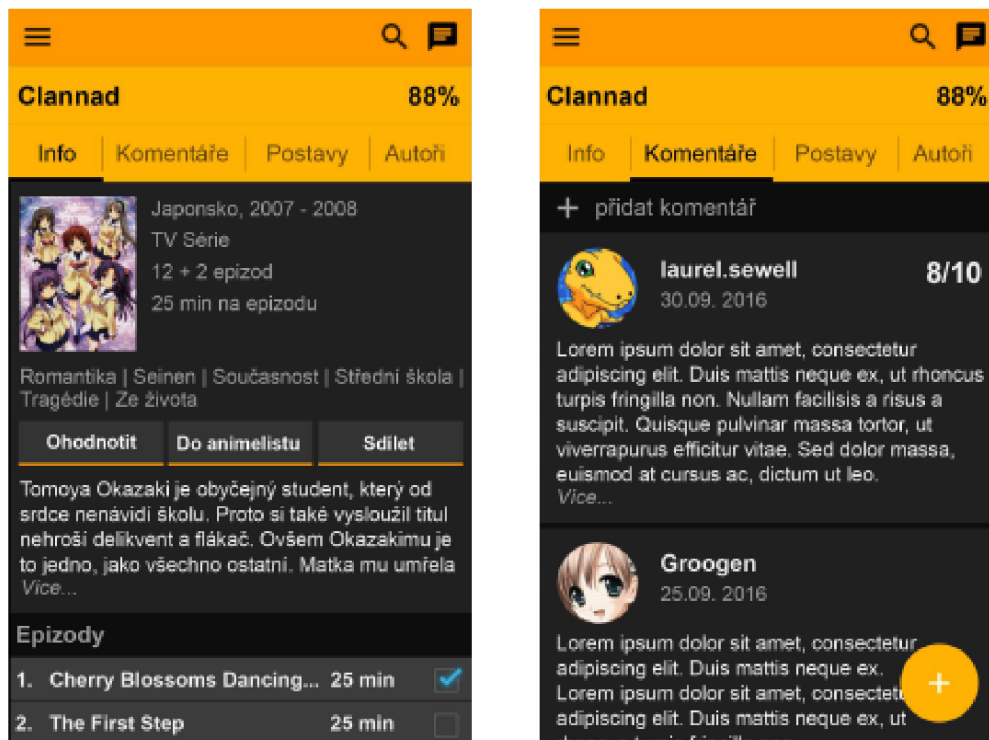


Obrázek 4.2: Návrh rozložení uživatelského profilu. Cizí profil (vlevo) a vlastní profil (vpravo).

Na obrázku 4.2 je možné vidět pod základními informacemi záložky, které budou dovolovat uživateli přepínat mezi statistikami vlastního profilu a seznamem přátel, se kterými je v kontaktu. Obrazovka uživatelského profilu bude téměř stejná jak pro vlastní profil, tak i pro profil cizího uživatele. Jedinou odlišností cizího profilu budou dvě tlačítka umístěna pod uživatelským jménem. Tlačítko **Do přátel** bude umožňovat přidat uživatele do seznamu svých přátel a tlačítko **Zpráva** bude umožňovat zaslání textové zprávy dotyčnému uživateli.

Zobrazení titulů a postav

V případě zobrazení informací o anime, postavách či autorech se jedná o jeden stejný grafický návrh, který se liší pouze v detailech pro každou kategorii. Proto návrh těchto obrazovek popíši dohromady.



Obrázek 4.3: Návrh rozložení uživatelského rozhraní pro zobrazení anime titulu.

Základním prvek pro orientaci a identifikaci titulu bude oblast obsahující název a hodnocení titulu. Tato oblast bude napříč záložkami statická, tudíž při přecházení mezi záložkami by měl mít uživatel stále na očích název titulu, na kterém se nachází. Následují záložky, by měly umožňovat přepínání mezi pohledy, čímž by se uživateli umožnilo rychle přecházet mezi okruhy s informacemi k danému titulu. Záložky u titulu by pak měly poskytovat následující informace viz 4.3:

- Info: obecné informace o titulu, náhledový obrázek a seznam epizod. Možnost vložení titulu do listu oblíbených, ohodnocení či sdílení.
- Statistika: statistická uživatelských hodnocení, zastoupení v listech oblíbených položek.
- Komentáře: přehled komentářů k titulu, možnost vložit vlastní komentář.
- Postavy: seznam postav účinkujících v titulu.
- Autoři: seznam autorů podílejících se na tvorbě titulu.
- Podobná a příbuzná anime: seznamy podobných a příbuzných titulů vztahujících se k aktuálnímu titulu.

Záložky u postavy budou poskytovat následující informace:

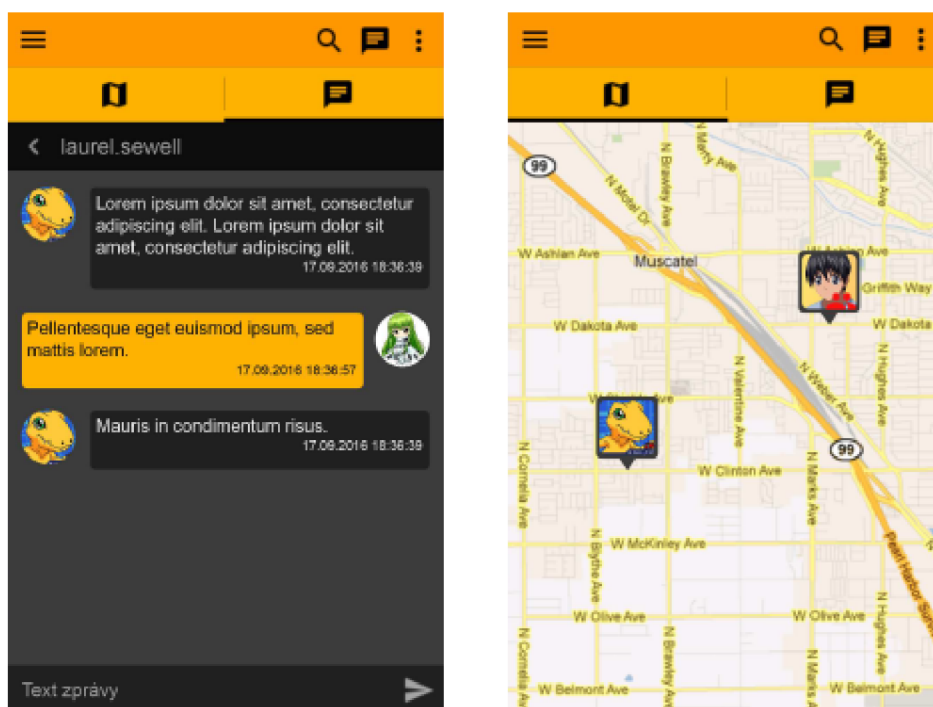
- Info: obecné informace o postavě a seznam dabérů, kteří postavě propůjčili svůj hlas. Možnost vložení postavy do listu oblíbených, ohodnocení či sdílení.
- Statistika: statistická uživatelských hodnocení, zastoupení v listech oblíbených položek.
- Komentáře: přehled komentářů k postavě, možnost vložit vlastní komentář.
- Příbuzná anime: seznam titulů, ve kterých postava účinkuje.

Záložky u autora bude nabízet tyto informace:

- Info: obecné informace o autorovi a seznam postav, které autor nadaboval. Možnost ohodnocení či sdílení autora.
- Komentáře: přehled komentářů k autorovi, možnost vložit vlastní komentář.
- Tvorba: seznamy titulů a způsob, jakým se autor na titulech podílel.

Chat a mapa

V rámci podněcování komunikace mezi uživateli bude do aplikace zařazen i chat s rozšířením v podobě mapy. Uživatel tak bude mít možnost přepínat mezi oknem chatu, ve kterém může komunikovat s jinými uživateli a oknem mapy, na které se bude zobrazovat pozice uživatelů v přátelském vztahu. 4.4 (vlevo).



Obrázek 4.4: Návrh rozložení uživatelského rozhraní pro okno chatu (vlevo) a navigační mapu zobrazující přátele v okolí (vpravo).

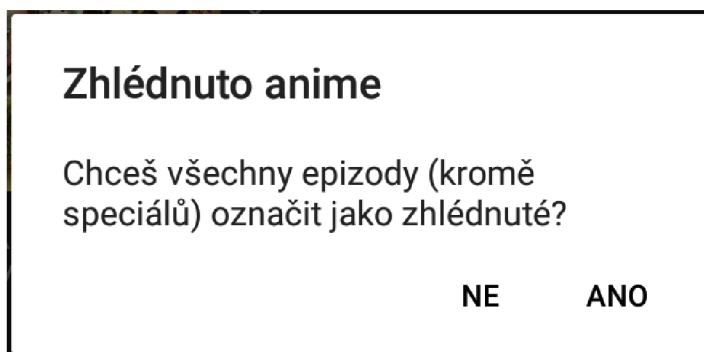
Okno chatu bude disponovat přehledem uživatelů, se kterými bylo v minulosti komunikováno. Při výběru se zobrazí standardní okno chatu zobrazující konverzaci. Okno mapy bude zobrazovat mapu vycentrovanou na aktuální pozici uživatele. Ukazatel pozice uživatele bude na mapě obsahovat ikonu uživatele a případný příznak, že uživatel hledá nové přátele. Při výběru ukazatele se zobrazí standardní okno chatu zobrazující konverzaci. Přepínání mezi mapou a chatem bude řešeno pomocí záložek s ikonou mapy a bubliny chatu.

Obrazovka vyhledávání

Jednou z nejdůležitějších obrazovek aplikace bude obrazovka s vyhledáváním. Tu by měl tvořit panel nástrojů (*toolbar*), ve kterém bude vložen vyhledávací formulář. Vyhledávací formulář bude v toolbaru k dispozici na všech obrazovkách aplikace, díky čemuž může uživatel vyhledávat z jakékoliv obrazovky. Obrazovka s vyhledáváním následně bude obsahovat záložky, které budou přepínat mezi okruhy (anime, postavy, autoři, uživatelé) vyhledávání. Uživatel tak může vyhledávat ve všech okruzích zároveň a jednoduše se mezi nimi přepínat.

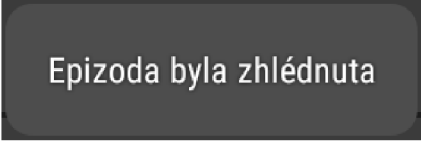
4.1.3 Dialogy a notifikace

Zobrazení zprávy v dialogovém okně bude mít za snahu upoutat uživatelskou pozornost a poskytnout zpětnou vazbu k akci, kterou uživatel provedl. Dialogové okno, pokud se nebude jednat pouze o oznámení chyby, bude od uživatele vyžadovat jistou interakci. To je důvod, proč je třeba dialogová okna používat střídavě, neboť přerušují uživatelskou předchozí činnost. Nadměrné nebo nevhodné používání dialogových oken může uživatele obtěžovat.^[10]



Obrázek 4.5: Ukázka dialogového okna dotazující se na označení všech epizod při zhlédnutí anime.

Dialogová okna budou v aplikaci použita pouze pro oznámení chyby, která uživateli znemožňuje pokračovat v jeho předešlé činnosti, či v případě nutné interakce s uživatelem. Při běhu aplikace však mohou nastat situace, kdy je potřeba uživateli oznámit, že nastala určitá událost. Často je třeba uživatele informovat o události, která byla dokončena, či zareagovat na některou z činností uživatele. Notifikace tak budou mít za úkol informovat uživatele, že se daná akce skutečně provedla. Stanou se tak důležitým prvkem zpětné vazby, která uživateli dodává jistotu, že akce, kterou provedl, se skutečně vykonala.



Epizoda byla zhlédnuta

Obrázek 4.6: Ukázka notificačního sdělení o zhlédnutí epizody.

4.2 Serverová část

Hlavní úlohou serverové části bude ve spolupráci s databází získávat a připravovat data, která požaduje mobilní aplikace. Mimo operace s databází a formátování dat bude zahrnovat i systém pro doporučování obsahu.

4.2.1 Databáze

V rámci mobilní aplikace se bude pracovat převážně s daty z původní databáze, která slouží i jako zdroj pro webovou službu. Avšak rozšíření služby na mobilní zařízení si žádá rozšířit databázi o několik nových tabulek.

- Tabulka `uzivatele_prihlaseni_app` bude obsahovat data uživatelů přihlášených přes mobilní zařízení. Konkrétně se bude uchovávat identifikátor uživatele, token sloužící k autentizaci uživatele, IP adresa, datum posledního přihlášení, poslední známé GPS souřadnice, příznak povolující sdílení polohy, příznak povolující hledání přátel a text pro hledané přátele.
- Tabulka `uzivatele_poloha_zakaz` bude sloužit k zákazu sdílení polohy s konkrétními uživateli. Uživatelé tak budou moci sdílet svou polohu pouze s vybranými uživateli. Atributy tabulky tak budou `id_uzivatel` a `id_pritel`. Atribut `id_uzivatel` bude reprezentovat identifikátor zakazujícího uživatele a atribut `id_pritel` identifikátor zakázaného uživatele. Pro zachování soukromí, pokud uživatel A zakáže sdílení polohy pro uživatele B, tak nejen že uživatel B neuvidí polohu uživatele A, ale ani uživatel A neuvidí polohu uživatele B.

4.2.2 Systém doporučování obsahu

Pro návrh a následnou implementaci systému, který by uživatelům doporučoval podobné položky, jsem zvolil přístup doporučování dle obsahu (*Content-based-recommendations*). Důvodem této volby je fakt, že systém by měl umět doporučovat i neregistrovaným uživatelům, přičemž jako referenční množina by měla sloužit množina navštívených položek. Naopak v případě registrovaných uživatelů by měl jako referenční množinu brát množinu ohodnocených položek. Doporučování dle obsahu tak dokáže zahrnout obě tyto možnosti.

Návrh algoritmu

Naše položka reprezentuje anime titul, přičemž každá položka je reprezentována množinou vlastností. Pod vlastností si lze představit štítek, který udává žánr či prvky, které se v titulu vyskytují. Proto každá položka bude v systému reprezentována vektorem přidělených vlastností. Uživatel bude v systému reprezentován vektorem oblíbených vlastností.

Porovnávání podobnosti vektoru vlastností položky s vektorem oblíbených vlastností uživatele bude prováděno pomocí *Vector Cosine Angle Distance* (dále jen VCAD) vzorce.

$$VCAD(X, Y) = \cos(\theta) = \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}} \quad (4.1)$$

Pokud X a Y jsou vektory $X \equiv \{x_1, x_2, \dots, x_n\}$ a $Y \equiv \{y_1, y_2, \dots, y_n\}$, pak $\cos \theta$ lze považovat za cosinus úhlu mezi vektory X a Y v n dimenzích. Jednou z důležitých vlastností VCAD je, že poskytuje metriku podobnosti mezi dvěma vektory narozdíl od Manhattanské metriky a Euklidovské metriky, obě totiž poskytují metriku odlišnosti. [5]

Celý systém doporučování bude navržen jako modulární systém, který umožňuje vkládat a odebírat jednotlivé moduly. Do budoucna je tak umožněno snadné rozšíření systému o moduly poskytující jiné přístupy doporučování.

Popis algoritmu

1. Získání všech položek z databáze a sestavení vektoru vlastností pro každou položku. Pokud určitá položka obsahuje určitou vlastnost, je ve vektoru reprezentována váhou 1. V opačném případě je vlastnost reprezentována 0. Množinu všech položek označíme jako N .
2. Množina U reprezentuje množinu všech uživatelů v systému. Získáme množinu položek M_u , které uživatel $u \in U$ ohodnotil nebo navštívil.
 - (a) **Registrovaný uživatel:** Množina M_u je sestavena dle dříve ohodnocených položek. Pokud uživatel žádné položky neohodnotil, je s ním zacházeno jako s neregistrovaným uživatelem.
 - (b) **Neregistrovaný uživatel:** Množina M_u je sestavena na základě dříve navštívených položek. Pokud uživatel v minulosti žádnou položku nenavštívil, je s ním zacházeno jako se zcela novým uživatelem 4.2.2.
3. Pro každou položku z množiny M_u je získán vektor vlastností, který je vynásoben hodnocením, které uživatel položce udělil. Pokud se jedná o neregistrovaného uživatele, bere se, jakoby položku ohodnotil nejvyšší známkou.
4. Následně je proveden součet vektorů všech položek z množiny M_u a jejich následný podíl počtem položek v množině M_u . Tím vznikne nový vektor F reprezentující oblíbené vlastnosti uživatele u .
5. Zavedeme novou množinu Q , která je rovna rozdílu množin N a M_u , čímž v množině Q zůstanou pouze položky, které uživatel u neohodnotil nebo nenavštívil.
6. Pro každou položku z množiny Q je provedeno porovnání podobnosti s vektorem F . Porovnání se provádí pomocí VCAD vzorce 4.1. Z výsledků porovnání je sestavena výsledná množina L , ve které jsou položky seřazeny dle nejlepší shody s vektorem F .
7. Z množiny L je vybráno x položek s nejlepší podobností, které jsou následně doporučeny uživateli.

Zcela nový uživatel — popis algoritmu

Při výběru uživatele, který nenavštívil žádné položky, musí umět algoritmus doporučit položky i tomuto uživateli.

1. Vybere se n položek z databáze s jejich největším počtem a součtem ohodnocení.
2. Z vybraných n položek se vytvoří seznam S , ve kterém jsou položky reprezentovány ve tvaru $\{id, \sum_{rating}, count_{rating}\}$, kde id udává identifikátor položky, $count_{rating}$ udává počet ohodnocení, které byly položce uděleny a \sum_{rating} udává součet všech ohodnocení, které položka obdržela. Následně byly zvoleny koeficienty, které ovlivňují součty tak, aby se doporučovalo jak na základě ohodnocení položky, tak na počtu ohodnocení. Tímto krokem je výsledek doporučení přesnější, protože nezávisí pouze na jedné či druhé hodnotě.
3. Množina S je sestupně seřazena dle vypočtené výsledné váhy, která se získá vynásobením jednotlivých součtů s koeficienty.
4. Z množiny S jsou pomocí normálního rozložení vybrány položky k doporučení.

4.3 Komunikace se serverem

Základem pro fungování aplikace je komunikace se serverem služby Happs.cz. V aplikaci bude využívána komunikace pomocí protokolu HTTP. Komunikace pak bude probíhat tak, že aplikace sestaví POST dotaz, který následně zašle na server a to na jednu konkrétní adresu. Server po přijetí dotazu načte obdržené hodnoty a na jejich základě vyvolá příslušnou funkci, která se postará o provedení požadované akce. Jako odpověď budou aplikaci zasílána strukturovaná data ve formátu JSON. Poslední fází je příjem dat aplikací a jejich následné zpracování a interpretování uživateli.

4.3.1 Formát dat

Vzhledem k tomu, že data budou stahována do mobilního telefonu, je hlavním kritériem velikost přenášených dat. Cílem je tedy minimalizovat objem dat, který je potřeba přenést. Z toho důvodu jsem vybral formát JSON, neboť oproti formátu XML poskytuje větší kompaktnost při zachování dostatečného strukturování přenášených dat. V mém případě jsou přenášená data jen málo strukturovaná, neboť se jedná převážně o tabulkové hodnoty. Použití formátu XML by tak nepřineslo žádnou výhodu.

Kapitola 5

Implementace

V této kapitole bude popsána implementace zajímavých částí aplikace, použitých externích knihoven a důvod jejich použití. Dále bude popsána implementace serverové části a systému doporučování obsahu.

5.1 Programovací jazyky a vývojové prostředí

K naprogramování aplikační logiky nativní aplikace byl použit jazyk Java. K vývoji aplikace pro operační systém Android jsem zvolil Android Studio verzi 2.1.3. Android Studio je vývojové prostředí vyvíjené ve spolupráci firmy Google a JetBrains. Přestože většina produktů od JetBrains jsou placené komerční nástroje, Android Studio je zdarma. Je postaveno nad Community verzí prostředí IntelliJ IDEA. Díky tomu získává mnoho možností práce s kódem jako je navigace v kódu, našeptávání, refaktoring, analýza kódu atd. Android Studio navíc poskytuje i grafický návrhář, ve kterém je možné navrhovat vzhled buď přímo v XML nebo v tzv. *Design módu*. V případě psaní v XML návrhář automaticky zobrazuje náhled ve vybraném rozlišení či orientaci obrazovky.

Serverová část je implementována v jazyce PHP ve spolupráci s dotazovacím jazykem MySQL, který slouží k získávání dat z databáze. Pro serverovou část je možné zvolit jako vývojové prostředí libovolný textový editor. V mém případě jsem využil volně šiřitelný univerzální editor PSPad.

5.2 Klientská část

V následujících podkapitolách je popsána implementace těch nejdůležitějších částí mobilní aplikace.

5.2.1 Přenos a ukládání multimédií

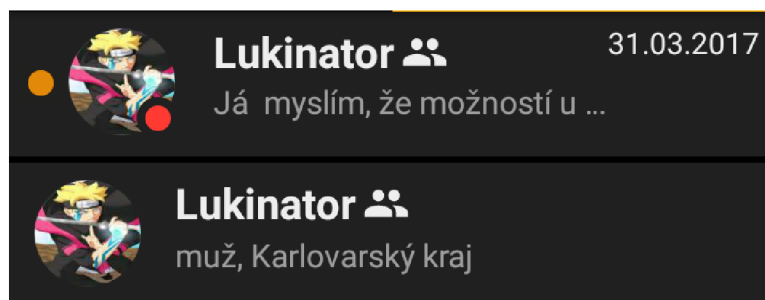
Aplikace nepracuje pouze s textovými daty, ale ve velké míře používá i multimediální soubory (konkrétně bitmapy ve formátech PNG a JPG). Formát JSON však nebylo možno k účelu přenosu dat použít, neboť podporuje pouze přenos textových informací. Řešením bylo vytvořit spojení se serverem, které vrací objekt typu *URLConnection* reprezentující spojení na zdroj. Spojení se vytváří přímo na URL adresu dané bitmapy. Následně se pomocí metody `getInputStream()` z balíku `java.io.InputStream` ze zdroje získal vstupní proud, ze kterého se načetla data. Ta byla následně převedena zpět na bitmapu.

Jelikož je přenos multimediálních souborů poměrně náročný na objem přenesených dat, mobilní aplikace si stažené bitmapy ukládá lokálně. Ke stahování tak dochází pouze v případě, že bitmapa nebyla již dříve stažena.

O stažení i ukládání se stará pomocná třída *ImageStorage*. Pokud některá část aplikace chce zobrazit bitmapu, zavolá metodu `setImg` případně `setImgInAsynch` pro neblokující operace. Obě metody nejprve provedou kontrolu, zda je bitmapa k dispozici v externím úložišti. Pokud k dispozici není, spustí se asynchronní úloha, která zajistí její stažení. Bitmapa je následně uložena do externího úložiště, čímž je k dispozici pro pozdější použití. V případě, že uživatel aplikaci zakáže přístup do externího úložiště, budou bitmapy při každém požadavku opět znova stahovány. Tím se však rapidně zvýší objem přenesených dat.

5.2.2 Seznamy

Jedním z nejdůležitějších prvků aplikace jsou seznamy. Aplikace disponuje mnoha seznamy s různorodými položkami. Základní kontejner seznamu však zůstává stejný. Seznamy tak obsahují pouze odlišné položky, které pro jednotlivé kategorie a obsahy byly stylovány do samostatných souborů. Položky byly navrženy dynamicky pro různé varianty použití v rámci aplikace.



Obrázek 5.1: Ukázka přizpůsobení položky uživatele dle účelu vypisování. Položka v chatu (nahore) a položka profilu uživatele (dole)

Příkladem může být položka pro zobrazení uživatelského profilu (viz obrázek 5.1). Stejná položka se však používá i v chatu pro výpis posledních zpráv. Důvodem tohoto řešení je, že obě položky používají stejné rozložení a tudíž by bylo zbytečné implementovat dvě velice podobné verze položky. Prvky, které jsou v rámci daného zobrazení nežádoucí, jsou jednoduše skryty. Jelikož bylo všem položkám seznamu vytvořeno vlastní rozložení, bylo třeba implementovat i vlastní adaptéry. Adaptér zajišťuje mapování dat na konkrétní prvky, ve kterých se mají data zobrazit. I v případě adaptérů byly velice podobné položky 5.1 zpracovávány jedním adaptérem. Při vytváření adaptéru se předal pouze parametr, který udával, pro jaký účel budou položky vypisovány. Vnitřní implementace adaptéru pak zajistila vše potřebné.

5.2.3 Mapa a GPS souřadnice

Implementace mapy se skládá ze dvou částí. První je zobrazení samotné mapy a druhou je získání své vlastní pozice. Vyobrazení mapy využívá Google Maps API (viz sekce 5.3.3), které uživateli poskytuje mapu, na níž jsou vyobrazeny polohy uživatelů v okolí. V tomto bodě bylo třeba zajistit, aby byla mapa zobrazena pouze v případě, že má uživatel povoleno sdílení své vlastní polohy. V opačném případě by mohlo dojít k narušení soukromí, kdy by

uživatel polohu nesdílel, ale zároveň by měl přístup k poloze ostatních uživatelů. Z toho důvodu se přistoupilo k opatření. Pokud uživatel nemá povoleno sdílení polohy, mapa mu není zobrazena. Na mapě jsou uživatelé reprezentováni pomocí tzv. *markerů*, které zaměřují aktuální polohu uživatele. Jelikož základní *marker*, který poskytuje Google Maps API je nedostačující, vznikla potřeba implementovat svůj vlastní. Požadavek byl takový, aby marker uměl zobrazit obrázek uživatelského profilu a zároveň jeho rám reprezentoval aktivitu módu hledání přátel. Nejprve byla vytvořena bitmapová šablona 5.2. Následně byl dynamicky do šablony vložen uživatelský obrázek. Nakonec byla vytvořena bitmapa, která byla zobrazena jako výsledný *marker* na mapě.



Obrázek 5.2: Šablony markeru.

Dále bylo třeba zajistit získávání polohy uživatele. Získávání polohy je řešeno pomocí rozhraní *LocationListener*, díky kterému je možné implementovat metody, které umožňují získat aktuální polohu uživatele. Jednou z klíčových metod je metoda `onLocationChanged()`, která je automaticky volána vždy, když se změní poloha uživatele. Před samotným získáním polohy je však nutné zkontrolovat práva na získání polohy a následně detekovat a vybrat vhodný zdroj, ze kterého bude poloha načítána. Jako zdroj polohy může posloužit mobilní či wifi síť, anebo získání souřadnic z GPS modulu zařízení. Získávání polohy z GPS je výrazně přesnější, avšak je i náročnější na spotřebu energie. Navíc ne všechna zařízení obsahují GPS modul. Proto byly implementovány obě varianty s tím, že pokud uživatel povolil získávání dat z GPS, je tato varianta upřednostňována.

Na pozadí probíhají pravidelné asynchronní dotazy na server ohledně aktualizace polohy, přičemž dotaz v sobě obsahuje aktuální polohu uživatele. Naopak odpověď ze serveru obsahuje aktuální polohy uživatelů.

5.2.4 Notifikační zprávy a upozornění

Aplikace disponuje událostmi, na které by měl být uživatel upozorňován. Mezi takové události patří přijetí nové zprávy v chatu, upozornění na přátele v okolí, žádost o přátelství atd. Bylo tak potřeba zajistit pravidelné stahování a upozorňování na vznik nové události. V intervalech (desítky sekund) tak na pozadí aplikace probíhají asynchronní dotazy na server žádající o nová upozornění. V případě aktivní aplikace se všechny zprávy s upozorněním zobrazují v seznamu na hlavní obrazovce aplikace. Pokud se jedná o událost upozornění na novou zprávu v chatu či výskyt přítele v okolí je navíc aktivován *Snackbar*¹ s upozorněním. V případě neaktivní aplikace je periodicky v určitých intervalech spouštěn tzv. *Service*², který zajišťuje získávání nových událostí ze serveru. Nově získané události však není možné promítnout do samotné aplikace, protože v době aktivity *Service* není aktivní aplikace. Z toho důvodu je pro upozornění uživatele využito notifikací přímo mobilního zařízení.

¹<https://developer.android.com/reference/android/support/design/widget/Snackbar.html>

²<https://developer.android.com/guide/components/services.html>

5.2.5 Chat

Jedná se o aktivitu zobrazující seznam zpráv, které proběhly s konkrétním uživatelem (viz obrázek 4.4 vlevo). Důležité bylo zajistit, aby seznam byl rozšiřitelný jak o nové zprávy, tak i o zprávy staré. Proto byly zvoleny proměnné, které uchovávají nejnovější a nejstarší zobrazenou zprávu. V případě příchodu nové zprávy je identifikátor nově příchozí zprávy nastaven jako nejnovější. Obdobně je tomu i při načítání historie zpráv, kdy vždy ta nejstarší zpráva je uložena do příznaku jako nejstarší. Seznam zpráv je následně rozšiřován od spodu o nové zprávy a z vrchu o zprávy staré. Bylo také zapotřebí řešit rozdílné grafické zobrazení pro vlastní odeslané zprávy a přijaté zprávy. Jak už bylo popsáno v kapitole 5.2.2, položky chatu byly stejně jako jiné podobné položky stylovány jedním jediným souborem. Samotné grafické rozvržení tak poskytuje všechny grafické prvky. Adaptér chatu zajišťuje zobrazení pouze těch grafických prvků, které mají být pro daný typ zprávy zobrazeny.

5.2.6 Statistiky

V rámci aplikace bylo třeba uživatele informovat o statistických údajích. Mezi takové údaje patří např. počet uživatelem zhlédnutých epizod nebo počet udělených ohodnocení a škála ohodnocení. Nejvhodnějším způsobem, jak tyto informace reprezentovat, bylo použití grafů a diagramů. Ty totiž umožňují názornou a grafickou reprezentaci dat oproti tabulkám. Uživatelé jim tak lépe porozumí. Pro implementaci grafů a diagramů byla využita knihovna MPAndroidChart (viz kapitola 5.3.2). Knihovna umožňuje jak jednoduchou implementaci různých typů grafů, tak i jejich detailní nastavení. Z knihovny jsem využil koláčový a spojnicový graf pro znázornění počtu zhlédnutých anime titulů a počtu zhlédnutých epizod za den. Sloupcový graf (viz obrázek 5.3) posloužil pro interpretaci rozložení škály hodnocení u anime titulů a postav.

5.3 Použité externí knihovny

5.3.1 Http Request

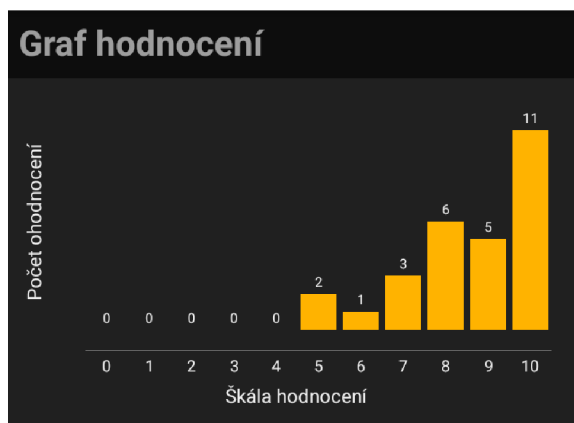
Jedná se o knihovnu umožňující jednoduché použití *HttpURLConnection* pro vytváření a zasílání různých druhů žádostí na server. Krom dotazů zajišťuje i získávání odpovědí ve formě HTTP stavových kódů, webových stránek nebo i souborů. Více o knihovně Http Request na GitHub³.

5.3.2 MPAndroidChart

MPAndroidChart je jednoduchá a velice efektivní knihovna umožňující implementovat přehledné grafy, které lze přizpůsobit přesnému obrazu aplikace. Důvodem zahrnutí této knihovny do projektu je ten, že knihovna nabízí širokou škálu grafů, jako jsou např. lineární grafy, spojnicové grafy, sloupcové grafy, bodové grafy anebo i koláčové a výsečové grafy. Dokonce lze některé z těchto typů grafů i kombinovat. Každý graf lze plně graficky i funkčně přizpůsobit potřebám aplikace, přičemž umožňuje uživateli i jednoduchou práci s grafy v podobě přiblížení, zvýraznění bodů apod. Knihovna je podporovaná u všech verzí Android API od verze 8. Více o knihovně na oficiální GitHub stránce⁴.

³<https://github.com/kevinsawicki/http-request>

⁴<https://github.com/PhilJay/MPAndroidChart>



Obrázek 5.3: Ukázka použití grafů z knihovny MPAndroidChart v aplikaci.

5.3.3 Google Maps API

Jedná se o knihovnu⁵ poskytující rozhraní pro mapovou aplikaci Google Maps. Umožňuje tak do aplikace integrovat základní mapy, trojrozměrné zobrazení budov, význačná místa, *Street View*, satelitní snímky i *markery*. Zároveň poskytuje prostředky pro zobrazení map a polohy na mapě dle zadaných souřadnic. Navíc umožňuje graficky přizpůsobit vyobrazení prvků, úroveň detailů či *markerů*, čímž lze mapu přizpůsobit konkrétním potřebám aplikace.

5.3.4 ViewPagerIndicator

Knihovna poskytující grafický prvek v podobě indikátoru stránkování. Prvek je kompatibilní s prvkem `ViewPager` a umožňuje vytvářet grafickou indikaci stránkování. V aplikaci je prvek stránkování použit na úvodní stránce aplikace u zobrazení novinek. Více o knihovně `ViewPagerIndicator` na [GitHub](https://github.com)⁶.



Obrázek 5.4: Ukázka použití `ViewPagerIndicator` ve spojení s `ViewPager` pro zobrazování novinek. Prvek indikátoru stránkování je umístěn v pravém horním rohu.

⁵<https://developers.google.com/maps/android/>

⁶<https://github.com/JakeWharton/ViewPagerIndicator>

5.3.5 CircleImageView

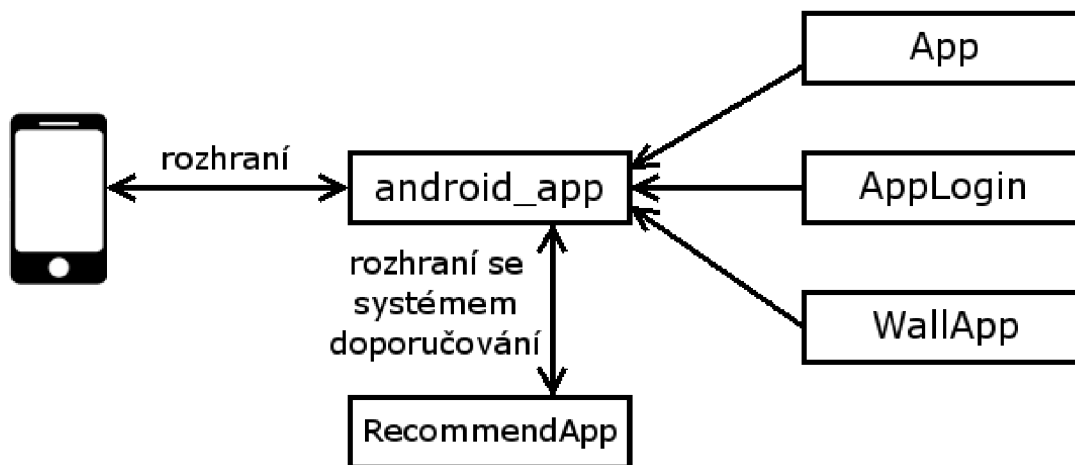
Jde o knihovnu poskytující prvek uživatelského rozhraní pro zobrazení obrázku v kruhu. Prvek této knihovny je v aplikaci využit pro zobrazení profilových fotek v uživatelském profilu a v seznamech uživatelů. Více o knihovně CircleImageView na GitHub⁷.

5.4 Serverová část

Serverová část je složena z části rozhraní, ke které se připojuje mobilní aplikace a z části systému pro doporučování obsahu.

5.4.1 API rozhraní

Část s rozhráním se skládá z hlavního souboru `android_app.php`, který zajišťuje příjem požadavků. Soubor je tvořen přepínačem, který dle přijaté hodnoty `action` v POST dotazu rozhoduje, která akce má být provedena. Na základě rozhodnutí je vyvolána příslušná funkce, která zajistí vykonání požadované operace. Převážná většina funkcí je složena pouze z dotazu do databáze a převedení dat do formátu JSON. U složitějších aktivit jako je např. vložení komentáře či ohodnocení je předáno řízení původním třídám, které slouží již pro akce spojené s webovou aplikací. Z důvodu odstranění opakování kódu byly vytvořeny dvě pomocné třídy `App` a `AppLogin`. Statická třída `App` obsahuje především metody pro převod dat do formátu JSON. Tyto metody jsou v rámci souboru `android_app.php` často používané a proto bylo vhodné jejich centrální umístění do zvláštní třídy. Obdobou je statická třída `AppLogin` jejíž metody zajišťují kontrolu přihlášení uživatele, případně provádí jeho přihlášení. V aplikaci je využívána takzvaná `zeď`, která slouží ke sledování aktivity a to jak vlastní, tak i uživatelů v přátelském vztahu. Získání dat aktivity zajišťuje třída `WallApp`, která se stará o výpis jednotlivých příspěvků a akcí jak pro vlastní, tak i pro cizí profil uživatele.



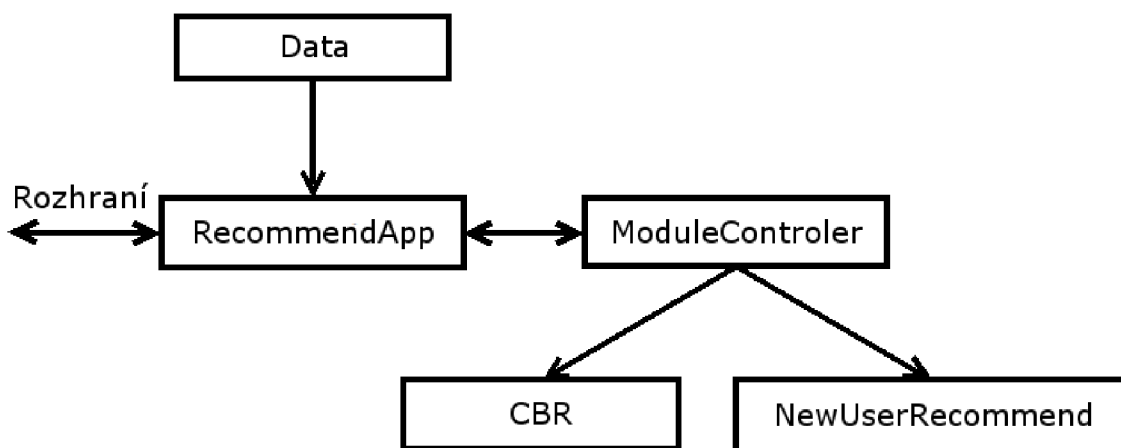
Obrázek 5.5: Schéma zobrazuje vztahy mezi význačnými třídami a soubory aplikačního rozhraní.

⁷ <https://github.com/hdodenhof/CircleImageView>

5.4.2 Systém doporučování obsahu

Systém pro doporučování obsahu je implementován jako systém modulů, které je možno jednoduše vkládat a odebírat. Součástí této práce je implementace dvou základních modulů, z nichž jeden zajišťuje doporučení na základě dříve navštívených/ohodnocených položek a druhý zajišťuje doporučení pro nově registrované nebo nepřihlášené uživatele.

- Třída *ModuleControler*: Jedná se o řídicí třídu, která zajišťuje předání řízení příslušnému modulu a předání potřebných dat. Samotné moduly jsou reprezentovány v adresáři `modules`. V rámci této práce byly implementovány dva moduly.
- Modul *CBR* (*Content Base Recommend*): *CBR* implementuje logiku dle algoritmu popsaného v kapitole 4.2.2
- Modul *NewUserRecommend*: *NewUserRecommend* implementuje algoritmus popsaný v kapitole 4.2.2, který zajišťuje doporučení položek pro nového uživatele.
- Třída *Data*: Moduly zajišťují pouze samotnou výpočetní logiku. Zajištění referenčních dat pro moduly se provádí mimo moduly ve třídě *Data*. Třída zajišťuje jak samotné dotazy na databázi na získání dat, tak i převod dat do formátu, který vyžadují jednotlivé moduly. Tím pádem moduly implementují pouze samotnou výpočetní logiku doporučování a nemusí se starat o způsob získávání dat.
- Třída *RecommendApp*: Představuje rozhraní mezi souborem `android_app.php` a systémem doporučování. Hlavním účelem této třídy je ve spolupráci s třídou *Data* připravit data do vhodného formátu pro jednotlivé moduly doporučování. Třída také zajišťuje výběr vhodného modulu na základě stavu uživatele (přihlášení, ohodnocených titulech atd.).



Obrázek 5.6: Schéma názorně zobrazující vztahy mezi třídami a moduly systému doporučování obsahu.

Kapitola 6

Testování

V této kapitole je shrnut postup testování implementované aplikace. Jsou vyhodnoceny zjištěné výsledky a na jejich základě stanoveny závěry a případné úpravy aplikace.

6.1 Testování aplikace na různých verzích API

Jak již bylo popsáno v kapitole 2.6.1, aplikace by měla podporovat všechny verze Android API vyšších než verze Android 4.0 (API 14). Samotný vývoj aplikace byl prováděn na fyzickém zařízení s verzí 6.0 a 4.1. Nicméně i když na obou verzích aplikace nevykazovala žádné problémy, bylo nutno ověřit všechny ostatní verze API. Testování aplikace probíhalo na emulátoru v rámci vývojového programu Android Studio. Postupně byla aplikace spouštěna a na jednotlivých verzích API, přičemž hlavní pozornost byla směřována na vykreslování grafického uživatelského rozhraní. Během testování nebyly zaznamenány žádné větší problémy. Objevila se pouze drobná chyba při vykreslování textu uvnitř tlačítek v uživatelském profilu na zařízeních s malou velikostí obrazovky. Chyba byla odstraněna pomocí detekce velikosti obrazovky, na jejímž základě došlo ke zmenšení velikosti písma u tlačítka.

6.2 Testování grafického uživatelského rozhraní

Během vývoje je vhodné otestovat již samotný návrh či prototyp aplikace, neboť úpravy v návrhu jsou značně jednodušší oproti úpravám ve finální aplikaci. Proto již v průběhu vývoje byl testujícím uživatelům předložen návrh jednotlivých obrazovek aplikace, přičemž nad každou obrazovkou proběhla krátká diskuze týkající se rozložení prvků, volby barev či velikosti písma.

Diskuze se zúčastnilo celkem 11 uživatelů. Na základě poznatků získaných od testujících uživatelů došlo k drobným změnám v grafickém návrhu aplikace. Na obrazovce chatu a mapy (viz obrázek 4.4) došlo v záložkách k výměně ikon za texty, čímž se eliminovaly stavy, kdy jednou byly v záložkách texty a podruhé ikony. Na obrazovce zobrazující informace o titulu (viz obrázek 4.3) došlo k prohození anotace s tlačítka, což vede k lepší soudržnosti textu a tlačítka zároveň oddělují základní informace od seznamu epizod.

6.2.1 Testování použitelnosti aplikace

Testování použitelnosti se skládalo z následujících tří etap:

1. Vytvoření sady úloh – byla vytvořena sada scénářů, na kterých se ověřovalo fungování aplikace.
2. Provedení testování úloh na skupině uživatelů – skupině uživatelů byla předložena sestavená sada úloh. Uživatelé měli za úkol splnit jednotlivé úlohy, přičemž během provádění úloh byl uživatel pozorován a byly zaznamenávány jeho reakce.
3. Vyhodnocení testů – během testování byly zaznamenány jak úspěchy, tak i problémy se kterými se uživatelé potýkali. Ze získaných údajů byly vyvozeny závěry a navrženy opravy nedostatků.

Screener

Než bylo zahájeno samotné testování, bylo třeba zvolit skupinu uživatelů, kteří budou aplikaci používat. K tomuto účelu byl sestaven screener.

Jedná se o druh dotazníku, který slouží k rozhodnutí, zda potenciální účastník testu spadá do zástupců uživatelů. Screener by měl vyřadit takové uživatele, kteří mají znalosti, které by mohly ovlivnit výsledky testů.[24]

Screener, který byl poskytnut potenciálním testovaným uživatelům je zahrnut v příloze **B**. Každá z otázek má předem definovanou předpokládanou odpověď. Mezi takové odpovědi patří hlavně aktivní využívání chytrého mobilního telefonu s platformou Android. Uživatelé, jejichž odpovědi se shodovaly s předpokládanými odpověďmi, byli vhodnými kandidáty pro následné testování samotné aplikace, neboť spadali do skupiny potenciálních uživatelů aplikace. Vhodnou skupinu uživatelů tak tvořili uživatelé aktivně využívající chytrý telefon s operačním systémem Android.

Dotazník a seznam úloh

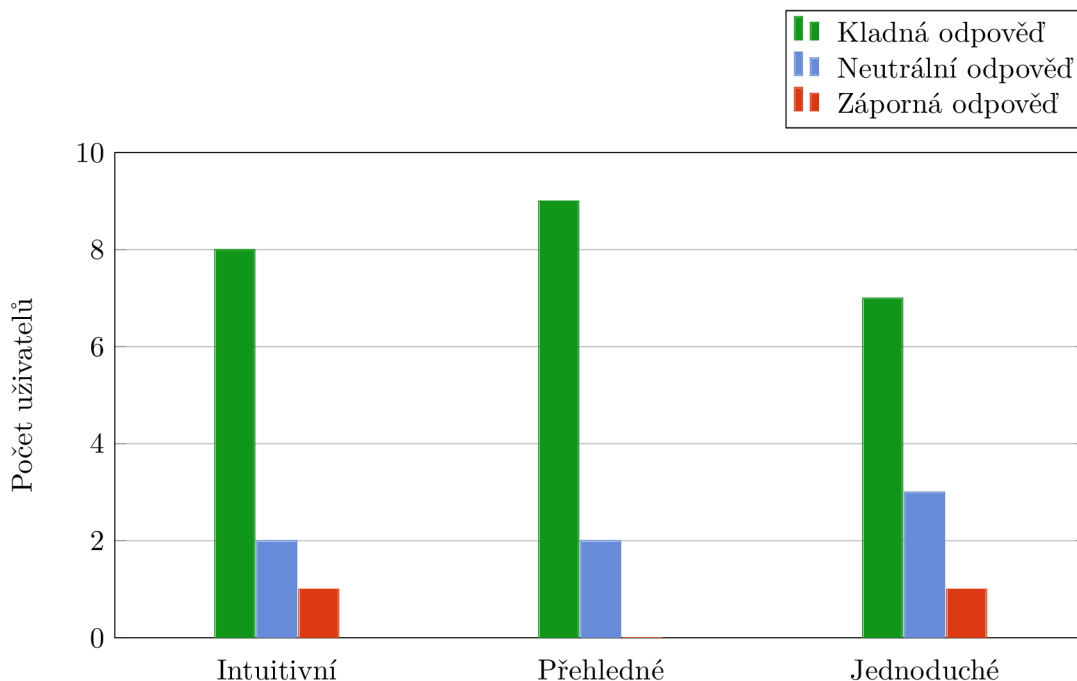
Po vybrání vhodné skupiny uživatelů pro testování se přistoupilo k samotnému testování aplikace. Testování probíhalo pomocí dotazníku, kde uživatel nejprve vyplnil obecné informace, které sloužily k lepšímu vyhodnocení testu. Následně se přistoupilo k seznamu úloh, které měl uživatel vykonat. Během provádění úloh byly pozorovány a zaznamenávány reakce uživatele. Pro lepší pochopení pocitů, měl uživatel za úkol verbálně popisovat své pocity a akci, nad kterou přemýšlí, což ve spojení s mými poznámkami utvořilo kompletní obraz, jak uživatel s aplikací pracuje. Na závěr, po vykonání všech úloh, měl uživatel vyplnit několik málo otázek, ve kterých sdělil své osobní dojmy jak ze samotného testování, tak z aplikace. Vzor sestaveného dotazníku se seznamem úloh je zahrnut v příloze **C**.

Během celého testu jsem byl jakožto moderátor přítomen, nicméně do samotného testování jsem nijak nezasahoval, až na výjimky, kdy si uživatel nevěděl rady. Testování probíhalo v tichém a klidném prostředí, kde uživatele nic nerozptylovalo od provádění úkolů.

6.2.2 Vyhodnocení testování

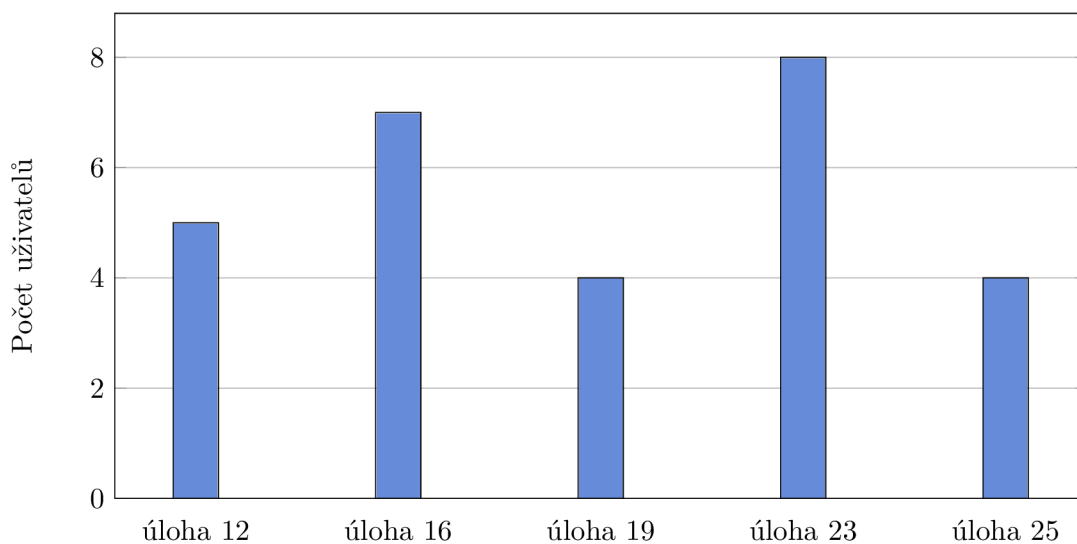
Testování se zúčastnilo celkem 11 uživatelů. Graf **6.1** zobrazuje odpovědi na otázky ohledně intuitivnosti a přívětivosti aplikace. Z odpovědí uživatelů plyne, že uživatelské rozhraní aplikace lze prohlásit za přehledné a srozumitelné. Tím byl splněn jeden z požadavků, který si dávala aplikace za cíl. Taktéž na otázku, zda by uživatelé uvažovali nad dalším využíváním aplikace, téměř 75% dotazovaných odpovědělo, že by nad touto myšlenkou vážně uvažovalo.

Během diskuze nad grafickým rozhraním však téměř polovina uživatelů uvedla, že jim přijde poměrně velká velikost písma. Zvláště pak v položkách seznamů. Na základě těchto



Obrázek 6.1: Volba odpovědí uživatelů na otázku jednoduchosti, přehlednosti a intuitivnosti aplikace.

poznatků došlo k mírnému zmenšení velikosti písma v celé aplikaci. Další častá připomínka se týkala seznamu zpráv v chatu, kdy při aktivaci klávesnice došlo k překrytí posledních zpráv. Tento nedostatek byl v zápětí odstraněn.



Obrázek 6.2: Úlohy, se kterými měli uživatelé největší problémy.

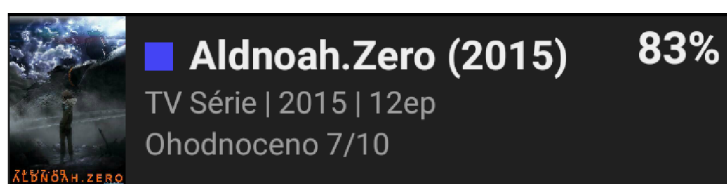
Pomocí moderovaného testování bylo důkladněji otestována funkčnost a použití aplikace. Uživatelé měli možnost vyzkoušet si práci s aplikací a zároveň napomohli k odhalení problémů, které by mohly mít vliv na celkovou plynulost práce nebo intuitivnost. V grafu

6.2 jsou uvedeny úlohy, které dělaly uživatelům největší potíže, či u kterých byly odhaleny nedostatky, které nebyly během návrhu a implementace vzaty v úvahu.

Úloha 12 – zobrazit animelist a zkontrolovat, zda se v něm nachází vložené anime

Provedení samotné úlohy nebyl ve většině případů problém. Avšak ne jeden uživatel poukázal na fakt, že mu u položek v animelistu chybí informace, zda danou položku ohodnotil a jakou známku mu dal.

Opatření Zde se naskýtalo poměrně jednoduché řešení. Položka animelistu obsahovala skryté textové pole pro doplňující obsah. Tudíž stačilo jen na straně serveru zajistit posílání údajů o ohodnocení anime (viz obrázek 6.3).



Obrázek 6.3: Ukázka doplnění informace o hodnocení uživatele do položky v animelistu.

Úloha 16 – přejít do characterlistu a vyfiltrovat postavy pouze nejoblíbenější a muže

Přejítí na obrazovku characterlistu nebyl problém ani v jediném případě. Avšak poměrně hodně uživatelů se potýkalo s problémem filtrování. Problém tkvěl v tom, že uživatelé hledali způsob, jak filtrování docílit. Přitom si však nevšimli, že v panelu nástrojů přibyla ikona aktivující dialogové okno s nastavením filtru. Uživatel, který si toho všiml, již neměl problém úlohu dokončit. Nicméně osm uživatelů z jedenácti hledalo právě tuto ikonu.

Opatření Dospěl jsem k závěru, že umístění ikony pro filtrování do panelu nástrojů není ideální řešení. V rámci celé aplikace se panel nástrojů mění jen minimálně, tudíž mu uživatel nevěnuje dostatečnou pozornost. Je tak jednoduché přehlédnout, že se v něm objevila další ikona. Z toho důvodu jsem ikonu umístil do tzv. *Floating Action Buttons* připnutého k pravému dolnímu okraji obrazovky.

Úloha 19 – vrátit se na hlavní stránku a otevřít obrazovku s mapou a chatem

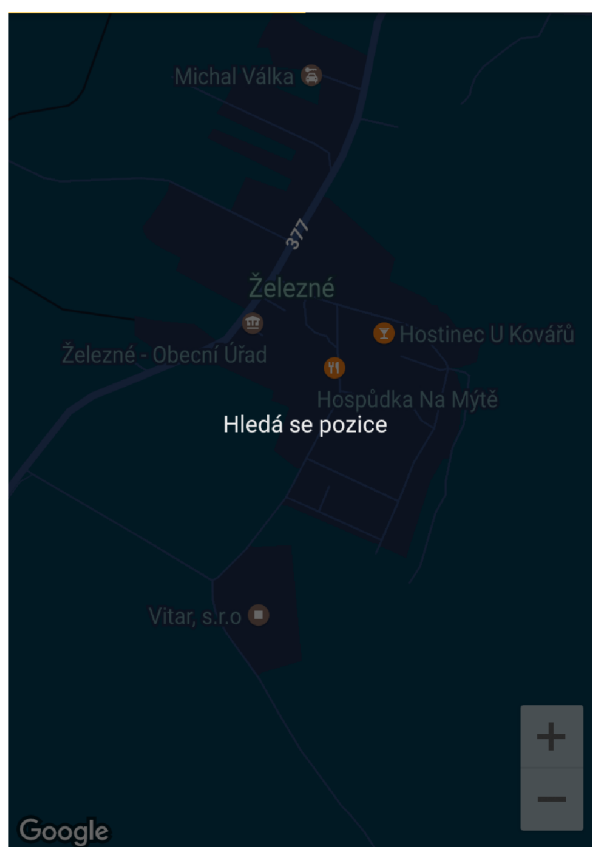
V případě této úlohy se jedná spíše o poznatek uživatelů ohledně položky v menu. V menu se vyskytuje položka chat, která zobrazí obrazovku, na které je možné se přepínat mezi chatem a mapou. V zadání úlohy stojí otevřít obrazovku s mapou a chatem. Dle vyjádření uživatelů tak v menu nejprve hledali položku mapa a chat. Položku chat sice zaregistrovali, ale nebyli si jisti, zda se jedná o onu obrazovku. Až když prohledali celé menu, vybrali tuto položku v domnění, že to bude nejspíš ta správná.

Opatření Řešení tohoto poznatku může mít dvě varianty. Buďto v menu vytvořit ještě jednu položku, která by odkazovala přímo na obrazovku s mapou nebo jen přejmenovat položku chat na položku chat a mapa. V mém případě jsem zvolil variantu druhou, tedy přejmenování položky.

Úloha 23 – zkontrolovat svoji polohu na mapě a poté přejít na svůj profil

Tato úloha tvořila největší potíž, neboť se zde objevily hned dva problémy. Prvním byla kontrola pozice na mapě. Po načtení mapy je jistá prodleva, mezi zobrazením mapy a načtením aktuální pozice uživatele. To se projevuje tak, že se na mapě hned nezobrazí *marker* uživatele. Uživatelé tak byli zmateni a nebyli si jisti, zda opravdu povolili sdílení své polohy nebo jestli aplikace funguje tak jak má. Další problém se vyskytl hned v zápětí, kdy uživatel měl zkontrolovat svoji pozici a přejít na svůj vlastní profil. Téměř polovina uživatelů se domnívala, že po vybrání svého vlastního *markeru* přejdou na svůj profil. Avšak aplikace je implementována tak, že kliknutí na svůj vlastní *marker* nevyvolá žádnou akci. Opět jsem se setkával se zmatením uživatelů a opakovaným výběrem svého vlastního *markeru*.

Opatření Řešením prvního problému bylo jednoduché. Přes mapu v průběhu načítání je vypsána hláška o načítání pozice, čímž informuji uživatele o probíhající činnosti (viz obrázek 6.4). Druhý problém byl vyřešen přidáním akce přesměrování na svůj profil v případě, že dojde ke kliknutí na vlastní *marker*.

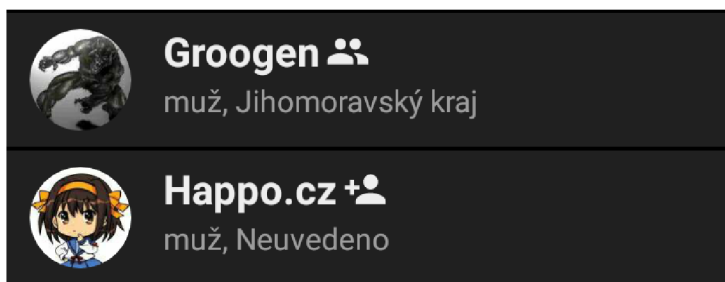


Obrázek 6.4: Ukázka hlášky informující uživatele o načítání aktuální polohy.

Úloha 25 – zobrazit seznam svých přátel

S touto úlohou neměli uživatelé ve většině případů problémy. Spíše se jednalo o poznámky všímavých uživatelů. V seznamu přátel se vypisovali všichni přátelé a to včetně těch, kterým byla zaslána žádost o přátelství. Jenže chyběla možnost rozlišení uživatelů v přátelském vztahu od uživatelů, kterým byla teprve zaslána žádost o přátelství.

Opatření Řešení bylo poměrně jednoduché a spočívalo v zavedení ikon, které by jednoznačně odlišily uživatele v přátelském vztahu od uživatele se zaslanou žádostí o přátelství. Ikony byly v položce seznamu umístěny za uživatelské jméno (viz obrázek 6.5).



Obrázek 6.5: Ukázka zavedení ikon k odlišení uživatele v přátelském vztahu (nahore) od uživatele, jemuž byla zaslána žádost o přátelství (dole).

6.2.3 Shrnutí

Testování dopadlo ve výsledku pozitivně, přičemž napomohlo k odhalení drobných nedostatků, které byly následně odstraněny. Uživatelům se úkoly jevily jako jednoduché a pouze v několika málo případech jim zabralo jejich vykonávání delší čas, neboť se jednalo o první seznámení s aplikací. Obecně byla práce s aplikací ohodnocena jako jednoduchá a intuitivní, čímž bylo potvrzeno splnění cíle aplikace. Testování probíhalo jak s uživateli, kteří se pohybují v prostředí techniky a moderních technologií, tak i s průměrnými uživateli chytrých mobilních telefonů.

Kapitola 7

Závěr

Cílem této práce bylo navrhnout a implementovat mobilní aplikaci, která umožní rychlý a jednoduchý přístup k informacím ze světa animovaných filmů a seriálů a zároveň bude podněcovat komunikaci mezi uživateli zabývající se touto tematikou. Aplikace navíc dokáže doporučovat vhodné položky na základě uživatelských preferencí a nabízí mu tak tituly, které by jej mohly zajímat. Dle výsledků návrhu i testování byl tento cíl úspěšně splněn.

Pro dosažení výsledků bylo nutné provést průzkum existujících řešení a analyzovat nej-používanější funkce webové aplikace Happo.cz. Taktéž bylo zapotřebí nastudovat jednotlivé přístupy a algoritmy pro doporučování obsahu a následně vybrat ten nejvhodnější pro implementaci. Pro následné zhotovení mobilní aplikace bylo třeba nastudovat pravidla správného vývoje uživatelského rozhraní pro platformu Android. To vedlo k navržení uživatelského rozhraní splňujícího tato pravidla. Během testování se správnost návrhu a implementace promítla do výsledků hodnocení aplikace. Aplikace je v době vydání této práce ojedinělou databází lokalizovanou pro české a slovenské uživatele.

V závislosti na oblíbenosti aplikace a zpětné vazby od uživatelů by mohla být aplikace v budoucnu rozšířena o další dva okruhy (soundtracky a vizuální novely) a doplněna o rozšiřující funkce v oblasti sdílení vzájemného obsahu mezi uživateli. Taktéž by bylo vhodné věnovat větší pozornost doporučování obsahu a navrhnout sofistikovanější moduly, které by se neomezovaly pouze na doporučování anime titulů. Budoucí plány budou samozřejmě úzce souviset s hodnocením a zpětnou vazbou uživatelů z obchodu Google Play Store.

Literatura

- [1] *Dashboards*. Android Developers.
URL <https://developer.android.com/about/dashboards/index.html>
- [2] *Introduction - Material design*. Material design.
URL <https://material.io/guidelines/#introduction-principles>
- [3] *Myšlení, myšlenkové operace, řešení problémů*. Studium psychologie.
URL <http://www.studium-psychologie.cz/obecna-psychologie/11-mysleni-myslenkove-operace.html>
- [4] *Pravidla – Synonyma, antonyma, homonyma*. Pravopisně.cz.
URL <http://pravopisne.cz/2014/10/synonyma-homonyma-antonyma/>
- [5] A. Vadivel, S. S., A. K. Majumdar: *Performance comparison of distance metrics in content-based image retrieval applications*. [Online; navštíveno 02.03.2017].
URL https://www.researchgate.net/publication/308953711_Performance_comparison_of_distance_metrics_in_content-based_image_retrieval_applications
- [6] Český statistický úřad: *Chytré telefony zvyšují počet uživatelů internetu*. [Online; navštíveno 07.11.2016].
URL <https://www.czso.cz/csu/czso/chytre-telefony-zvysuji-pocet-uzivatelu-internetu>
- [7] Bartůšek Josef, V. F.: Jak v češtině zacházet se jmény z východoasijských jazyků. *Naše řeč*, ročník 42, 1959: s. 204–214, ISSN 0027-8203.
- [8] Emilio J. Castellano, L. M., Manuel J. Barranco: *Academic Orientation Supported by Hybrid Intelligent Decision Support System*. [Online; navštíveno 29.11.2016].
URL <http://www.intechopen.com/books/efficient-decision-support-systems-practice-and-challenges-from-current-to-future/academic-orientation-supported-by-hybrid-intelligent-decision-support-system>
- [9] Horgoš, J.: *Otaking po česku: Komunity tehdy a dnes*. [Online; navštíveno 06.11.2016].
URL <http://hintzu.otaku.cz/otaku/otaking-po-cesku-komunity>
- [10] Johnson, J.: *Designing with the Mind in Mind*. Morgan Kaufmann Publishers, 2010, ISBN 978-0-12-375030-3.
- [11] Krasnoshchok, O.: *Collaborative Filtering Recommender Systems – Benefits and Disadvantages*. Recommender Systems, [Online; navštíveno 30.11.2016].
URL <http://recommender.no/info/collaborative-filtering-approach/>

- [12] Krasnoshchok, O.: *Content-Based Filtering Recommender Systems – Benefits and Disadvantages*. Recommender Systems, [Online; navštíveno 29.11.2016].
URL <http://recommender.no/info/content-based-filtering-recommender-systems/>
- [13] Krasnoshchok, O.: *How the Matrix Factorization algorithm works?* Recommender Systems, [Online; navštíveno 30.11.2016].
URL <http://recommender.no/algorithms/matrix-factorization-algorithm-works/>
- [14] Krasnoshchok, O.: *Hybrid Recommender Systems*. Recommender Systems, [Online; navštíveno 01.12.2016].
URL <http://recommender.no/info/hybrid-recommender-systems/>
- [15] Krasnoshchok, O.: *The Input, the Output, the Goal and Tasks of Recommender Systems*. Recommender Systems, [Online; navštíveno 29.11.2016].
URL <http://recommender.no/info/recommender-systems-tasks-goal-input-output/>
- [16] Krasnoshchok, O.: *What recommender systems algorithm is the best for my objectives?* Recommender Systems, [Online; navštíveno 01.12.2016].
URL <http://recommender.no/algorithms/recommender-systems-algorithm-best-objectives/>
- [17] Lacko, L.: *Vývoj aplikací pro Android*. Computer Press, 2015, ISBN 978-80-251-4347-6.
- [18] Liu, B.: *Web data mining: exploring hyperlinks, contents, and usage data. 2nd ed.* New York: Springer, 2011, ISBN 978-3-642-19459-7.
- [19] McCarthy, H., Jonathan Clements: *The Anime Encyclopedia: A Guide to Japanese Animation Since 1917*. STONE BRIDGE PRESS, 2006, ISBN 1-933330-10-4.
- [20] Michael J. Pazzani, D. B.: *Content-Based Recommendation Systems*. The Adaptive web, [Online; navštíveno 27.11.2016].
URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.8327&rep=rep1&type=pdf>
- [21] Odine, D.: *Velmi stručná historie anime: Od Shimokawy po současnost*. [Online; navštíveno 06.11.2016].
URL <http://www.wakasa.eu/anime.pdf>
- [22] Palanisamy, S. K.: *Association Rule Based Classification*. Disertační práce, WORCESTER POLYTECHNIC INSTITUTE, Worcester, 2006.
- [23] Pasquale Lops, G. S., Marco de Gemmis: *Content-based Recommender Systems: State of the Art and Trends*. Springer US, 2011, ISBN 978-0-387-85820-3_3.
- [24] Russ Unger, C. C.: *A project guide to UX design: for user experience designers in the field or in the making*. New Riders, 2009, ISBN 0-321-60737-6.
- [25] Xiaoyuan Su, T. M. K.: A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, ročník 2009, 2009: str. 19, ISSN 1687-7489.

- [26] Xiong, H.: *Association Analysis: Basic Concepts and Algorithm*. [Online; navštíveno 30.11.2016].
URL <http://www.columbia.edu/~jwp2128/Teaching/W4721/papers/ch6.pdf>

Přílohy

Seznam příloh

A	Dotazník	50
B	Screeener pro aplikaci Happo.cz	52
C	Formulář pro ověření uživatelského rozhraní aplikace Happo.cz	53
C.1	Sada úloh	53

Příloha A

Dotazník

1. Jak často navštěvujete webovou aplikaci Happo.cz?
 - Zřídka, několikrát do měsíce
 - Několikrát do týdne
 - Denně
2. Máte vytvořen na Happo.cz účet?
 - Ano
 - Ne
3. Za jakým účelem navštěvujete web? (možnost výběru více položek)
 - Získávání informací o titulech a postavách
 - Udržování seznamů s oblíbenými tituly a postavami
 - Kontakt s přáteli
 - Vyjádření svého názoru na jednotlivá díla (hodnocení, komentování)
 - Podílím se na tvorbě obsahu (přidávání postav, vytváření vztahů mezi tituly, tvorba anotací,...)
 - Sledování aktivit přátel
4. Jaké akce provádíte nejčastěji? (vyberte maximálně 3 položky)
 - Prohlížení anime
 - Prohlížení postav
 - Vyhledávání nových titulů a postav
 - Udržování listů s oblíbenými položkami
 - Vytváření vztahů mezi tituly a postavami
 - Tvorba anotací k titulům a postavám
 - Hledání příbuzných titulů k určitému titulu
 - Hledání příbuzných postav k určitému titulu
 - Hodnocení
 - Komentování

- Tvorba recenzí
- Procházení obrázků u titulů či postav
- Procházení videí u titulů či postav
- Procházení svých statistik v profilu
- Procházení globálních statistik webu
- Procházení příspěvků a aktivit přátel
- Komentování příspěvků a aktivit přátel
- Procházení profilů přátel
- Hledání nových přátel
- Chatování
- Jiné: _____

5. Připadá Vám aktuální web přehledný?

- Ano
- Občas nevím, kde co hledat
- Často se ztrácím
- Rozhodně ne

6. Uvítali byste nějakou novou funkcionalitu?

- Ano
Jakou? _____
- Ne
Proč? _____

Příloha B

Screeener pro aplikaci Happs.cz

1. Používáte chytrý mobilní telefon?

- Ano
- Ne

2. Zvolte operační systém Vašeho telefonu.

- Android
- iOS
- Windows Phone
- Jiný

3. Používáte mobilní telefon i na jiné úkony, než jen volání a zasílání textových zpráv?

- Ano
- Ne

4. Jak často se z mobilního telefonu připojujete k internetu (mobilní síť i WIFI)?

- Nikdy (internet v mobilu nevyužívám)
- Několikrát do měsíce
- Alespoň jednou týdně
- Několikrát do týdne
- Denně

Příloha C

Formulář pro ověření uživatelského rozhraní aplikace Happo.cz

1. Používáte webovou aplikaci Happo.cz?

- Ano
- Ne

2. Máte předchozí zkušenosti s testováním?

- Ano
- Ne

Poznámka:

C.1 Sada úloh

1. Spusťte aplikaci Happo.cz
2. Seznamte se s uživatelským rozhraním
3. Přihlaste se pod účtem testuser (heslo: drf381)
4. Vyhledejte anime „Claymore“ a zobrazte detailní informací o tomto anime
5. Prohlédněte si záložky a zjistěte, kolik hlavních postav v anime účinkuje
6. Přejděte na záložku Komentáře a vložte nový komentář
7. Zobrazte si svůj animelist a zkontrolujte, zda se v něm nenachází žádné anime
8. Vraťte se na Hlavní stránku a zobrazte si detailní informace o libovolném anime ze seznamu Doporučujeme
9. Ohodnoťte anime libovolnou známkou
10. Přidejte zobrazené anime do animelistu pod statutem Rozkoukáno
11. Označte první tři epizody u anime za zhlédnuté
12. Zobrazte animelist a zkontrolujte, zda se v něm nachází Vámi vložené anime

13. Odstraňte anime z animelistu
14. Vyhledejte postavu „Alicia Luu“ a zobrazte si detailní informace o této postavě
15. Přidejte postavu do characterlistu pod libovolným statusem
16. Přejděte do characterlistu a vyfiltrujte postavy pouze nejoblíbenější a muže
17. Vraťte se na Hlavní stránku, vyhledejte uživatele „Stitch“ a zobrazte si jeho profil
18. Přidejte si uživatele do přátel a zašlete mu libovolnou zprávu
19. Vraťte se na Hlavní stránku a otevřete obrazovku s mapou a chatem
20. V chatu vyberte uživatele Happo.cz a zašlete mu libovolnou zprávu
21. Vraťte se zpět na chat (seznam uživatelů) a v záložkách se přepněte na mapu
22. Klikněte na tlačítko NASTAVENÍ, povolte Sdílet polohu a vraťte se zpět na mapu
23. Zkontrolujte svoji polohu na mapě a poté přejděte na svůj profil
24. Zkontrolujte, zda jste přihlášení pod uživatelem testuser
25. Zobrazte si seznam svých přátel
26. Odhlaste se
27. Ukončete aplikaci

3. Jak byste ohodnotil/a obtížnost vykonání úloh?

- Jednoduché
- Občas jsem si nebyl/a jistý/á jak daný úkol provést
- Úlohy byly obtížné

Poznámka:

4. Připadalo Vám uživatelské rozhraní intuitivní?

- Ano
- Informace či ovládací prvky jsem musel/a hledat
- Rozhraní není intuitivní

Poznámka:

5. Připadalo Vám uživatelské rozhraní přehledné?

- Ano
- Ztrácel/a jsem se
- Rozhraní není přehledné

Poznámka:

6. Jak se vám s aplikací pracovalo?

7. Přemýšlíte, že byste aplikaci využíval/a i nadále?

- Ano
- Ne

Poznámka: