

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

APLIKACE PRO HODNOCENÍ KVALITY POKRMŮ NA ANDROID

BAKALÁŘSKÁ PRÁCE

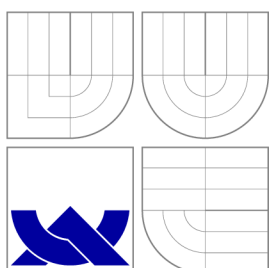
BACHELOR'S THESIS

AUTOR PRÁCE

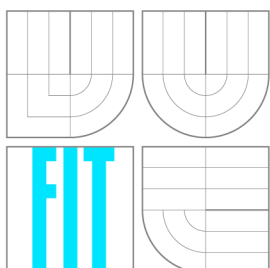
AUTHOR

JAN TOMEŠEK

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

APLIKACE PRO HODNOCENÍ KVALITY POKRMŮ NA ANDROID

APPLICATION FOR FOOD QUALITY RATING FOR ANDROID

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JAN TOMEŠEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2015

Abstrakt

Tato práce se zabývá návrhem, implementací a testováním mobilní aplikace, která představuje součást vznikající služby pro hodnocení kvality pokrmů ve stravovacích zařízeních, přičemž hodnocení pocházejí přímo od strávníků. Uživatelé se mohou nechat inspirovat hodnoceními jiných uživatelů, mohou hodnocení sami přidávat, případně si pokrmy a stravovací zařízení vyhledávat. Primárně se práce soustředí na návrh uživatelského rozhraní. Služba je uživatelům zprostředkována pomocí aplikace pro platformu Android. V rámci práce byla vytvořena aplikace, která je kompatibilní s 94 % Android zařízení na současném trhu a kterou lze po odladění v budoucnu uvolnit pro reálné použití. Čtenáři tato práce poskytuje náhled na aktuální trendy ve vývoji mobilních aplikací, předkládá úvahy související s návrhem reálně uplatnitelné aplikace a nabízí výběr nástrojů a prostředků, které lze pro vývoj využít či do aplikace integrovat.

Abstract

This thesis deals with design, implementation and testing of a mobile application which represents a part of a newly emerging service for food quality rating in catering facilities. The ratings come directly from consumers. Users can get inspired by ratings from other users, they can add ratings on their own or search for dishes and facilities. The thesis primarily focuses on the user interface design. The service is brought to users through an application for the Android platform. Application, which is compatible with 94 % of Android devices on the market and which could be released for real usage in the future, was created in this thesis. To a reader the thesis provides a look at the current trends in mobile application development, it presents considerations related to the design of a real world application and offers a selection of tools and means which can be used for development or be integrated into the application.

Klíčová slova

Android, aplikace, mobilní, návrh uživatelských rozhraní, hodnocení kvality pokrmů, trendy uživatelských rozhraní, návrhové vzory uživatelských rozhraní, Material Design, programování pro platformu Android.

Keywords

Android, application, mobile, user interface design, food quality rating, trends in user interfaces, user interface design patterns, Material Design, programming for the Android platform.

Citace

Jan Tomešek: Aplikace pro hodnocení kvality pokrmů na Android, bakalářská práce, Brno, FIT VUT v Brně, 2015

Aplikace pro hodnocení kvality pokrmů na Android

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jan Tomešek
19. května 2015

Poděkování

Děkuji svému vedoucímu, panu Ing. Vítězslavu Beranovi, Ph.D., za motivaci, odborný dohled a především podnětné a nápádné rady pro vylepšení výsledku mé práce. Dále děkuji týmu svých kolegů, Petru Bobákovi a Jakubu Špačkovi, za spolupráci na tvorbě služby, jejíž součástí má aplikace tvořit.

© Jan Tomešek, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	2
2 Znalosti potřebné pro vytvoření Android aplikace	3
2.1 Existující řešení v oblasti hodnocení stravovacích zařízení	3
2.2 Designové principy platformy Android	6
2.3 Trendy a návrhové vzory mobilních uživatelských rozhraní	9
2.4 Material Design	11
2.5 Programování pro platformu Android	13
3 Návrh aplikace pro hodnocení kvality pokrmů	16
3.1 Koncept aplikace	16
3.2 Možné alternativy v návrhu rozhraní	22
3.3 Rozšiřující funkce	23
4 Realizace aplikace pro hodnocení kvality pokrmů	25
4.1 Základní nástroje použité pro vytvoření aplikace	25
4.2 Foursquare API	26
4.3 Úvodní dotazování a prvotní reakce uživatelů	26
4.4 Implementace aplikace a integrace existujících služeb	27
4.5 Zajištění podpory co nejvíce zařízení	33
4.6 Návrh testování uživatelského rozhraní	35
4.7 Interpretace výsledků testování uživatelského rozhraní	36
5 Závěr	38
A Struktura základních obrazovek výsledné aplikace	41
B Výňatek z odpovědi na požadavek na Foursquare API	43
C Výňatek z odpovědi na požadavek na vlastní API	44
D Scénář testování uživatelského rozhraní	45
E Použité knihovny	47

Kapitola 1

Úvod

V této práci se zabývám zrodem mobilní aplikace pro novou službu se zaměřením na hodnocení pokrmů. Procházím celým procesem nutným pro vytvoření nového produktu – aplikace pro platformu Android. Jedná se o aplikaci, ve které se budou moci běžní strávníci nechat inspirovat hodnoceními pokrmů, které vloží jiní uživatelé. Naleznou zde inspiraci pro nejbližší večeři, případně mohou světu sami předat své poselství v podobě hodnocení. Aplikace jim umožní vyhledat si stravovací zařízení ve vybrané lokalitě stejně jako vyhledat pokrm dle jejich libosti. Celá služba si v konečném důsledku klade za cíl vytvořit virtuální svět pro gurmány, kritiky i běžné strávníky, do kterého se uživatelé díky sociálním a gamifikačním prvkům budou mít chuť denně vracet.

Služba jako celek vzniká ve spolupráci s dalšími dvěma kolegy, Petrem Bobákem a Jakubem Špačkem, kteří po řadě zajišťují verzi aplikace pro systém iOS a aplikační backend.

Práce mapuje budování související aplikace od samotného studia základů tvorby uživatelského rozhraní až po testování výsledného produktu. Primární náplní je snaha o tvorbu kvalitního uživatelského rozhraní, což akcentuje i následující text.

V následujících kapitolách nejdříve proberu znalosti, které jsou k vytvoření aplikace a efektivního rozhraní nutné. Každá platforma a zařízení, která ji používají, si vynucují specifické atributy uživatelského rozhraní a nejinak je tomu u platformy Android. Z tohoto důvodu se budu zabývat studiem designových principů přímo této platformy (2.2), jakož i obecnými trendy a návrhovými vzory ve světě mobilních zařízení (2.3). Neopomenu ani studium jiných existujících řešení/aplikací (2.1), u nichž vyzvednu klady a zdůrazním zápory, a pokusím se odnést si z nich ty nejlepší aspekty, které uplatním u vlastního rozhraní.

Tyto znalosti následně využiji k navržení smysluplného konceptu aplikace (3.1), kde každá obrazovka a prvky v rámci ní budou mít své opodstatněné místo. V rámci celého návrhu se budu snažit vhodně zkombinovat obecné znalosti návrhu uživatelských rozhraní a zaměření aplikace na oblast stravování.

V souvislosti s převodem těchto znalostí a myšlenek z papíru na interaktivní prostředí se budu v práci věnovat i programování pro platformu Android (2.5) a souvisejícím nástrojům (4.1).

V implementační části (4.4) se zaměřím především na představení a popis různých nástrojů, knihoven a API, které mohou obohatit výsledný uživatelský zážitek.

Finální a důležitou část pak tvoří testování rozhraní na potenciálních uživateli, ve které popíši návrh celého testování (4.6) a také využití získané zpětné vazby pro zdokonalení řešení (4.7).

Kapitola 2

Znalosti potřebné pro vytvoření Android aplikace

2.1 Existující řešení v oblasti hodnocení stravovacích zařízení

V současném světě mobilních aplikací je těžké přijít s nápadem, který by, alespoň v mírně pozměněné podobě, ještě nebyl zpracován. Existuje tedy i řada aplikací, které se v určitém ohledu zabývají stravovacími zařízeními či pokrmy. Nejen u takto zaměřených aplikací se pokusím najít nedostatky, ze kterých se poučím pro tvorbu vlastního rozhraní, stejně jako užitečnou inspiraci.

Všechny následující aplikace a především jejich obrazovky jsou zmíněny proto, že se jejich pozitivní aspekty určitým způsobem projeví v mém uživatelském rozhraní a nebo naopak budou tyto obrazovky podnětem k neopakování stejné chyby.

2.1.1 DameJidlo.cz

Aplikace zprostředkovává českou službu známou také jako DámeJídlo.cz. Jejím primárním cílem je poskytnutí seznamu stravovacích podniků, které nabízejí dovoz jídla až k zákazníkovi domů. Přímou přes aplikaci lze vytvořit a odeslat objednávku, kterou lze často rovnou i zaplatit pomocí mobilních plateb.

Pozitiva uživatelského rozhraní

- jediná úvodní obrazovka (uživatel není zahlcen, obr. 2.1)
- využití výsuvného panelu pro pokročilé vyhledávání (nabízí dostatek místa pro změnu parametrů vyhledávání)

Negativa uživatelského rozhraní

- jednotlivé pokrmy nejsou doplněny fotografiemi, které by uživatelům poskytl lepší představu

2.1.2 Restu

Za službou Restu stojí společnost MITON CZ, která dříve pracovala na projektech jako jsou Hotel.cz, Heuréka.cz a výše zmíněné DámeJídlo.cz. Stejnoujmennou aplikaci vytvořil vývojářský tým The FUNTASTY. Primárním snahou této aplikace je realizace rezervací v podnicích skrze chytrý telefon.

Pozitiva uživatelského rozhraní

- časté využívání fotografií podniků či ilustračních fotografií potravin (oživují aplikaci)

Negativa uživatelského rozhraní

- horní navigační panel s příliš mnoho položkami (mohou uživatele zahltit, obr. 2.2)
- jednotlivé pokrmy nejsou doplněny fotografiemi
- nevhodné ovládání v detailu podniku, kde má tažení prstem různé výsledné chování v závislosti na místě dotyku (matoucí ovládání)

2.1.3 Foursquare

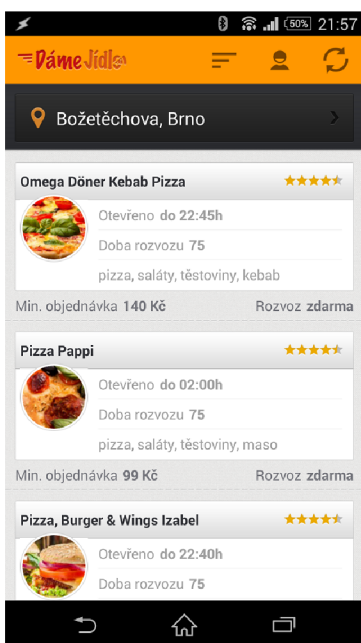
Foursquare je asi nejznámější službou a zároveň největší databází podniků z oblastí všeho druhu. S postupem času se čím dál více soustředí na oblast stravovací, ze které uživatelům doporučuje kvalitní podniky podle jejich chutí a umožňuje jim vyhledávat v již zmíněné rozsáhlé databázi.

Pozitiva uživatelského rozhraní

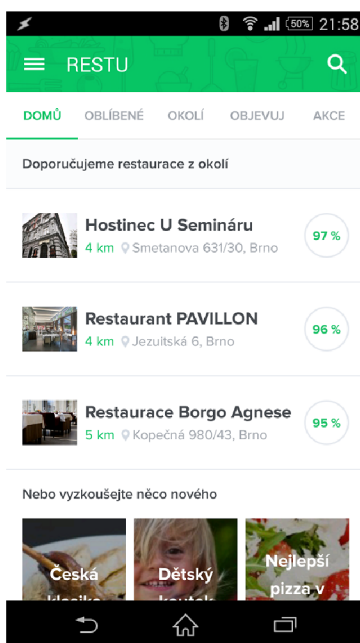
- hojné využívání fotografií podniků

Negativa uživatelského rozhraní

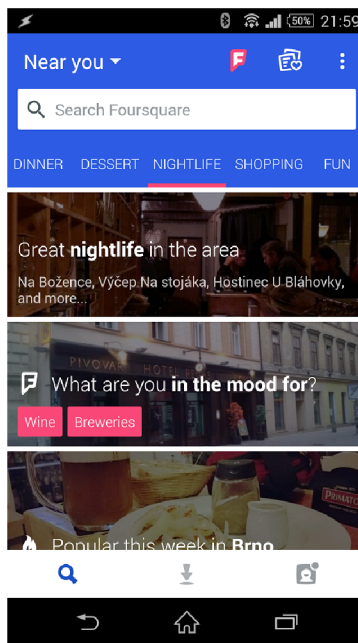
- chaotické prostředí, ve kterém se uživatel musí navigovat s využitím hned tří prepínacích panelů (obr. 2.3)
- nejsou k dispozici žádné informace o nabízených pokrmech



Obrázek 2.1: DameJidlo.cz



Obrázek 2.2: Restu



Obrázek 2.3: Foursquare

2.1.4 Meníčka/iMeníčka

Tato aplikace se soustředí na jednoduchý cíl – zobrazování denního menu. Nabízí podniky včetně jejich aktuálních denních nabídek, a to buď podle uživatelské polohy, případně podle vybraného města. Požadované informace ovšem nepodává příliš přívětivě či stravitelně. Hodilo by se využít fotografií měst / stravovacích zařízení / pokrmů či jakékoli jiné grafické reprezentace pro vyvolání konkrétnější představy v mysli uživatele.

Pozitiva uživatelského rozhraní

- obrazovka s podniky ve vybrané oblasti vhodně umísťuje informace o podniku i jeho denním menu na jednu obrazovku (uživatel není nucen vždy proklikávat na detail podniku, obr. 2.4)
- možnost jednoduchého přepnutí barevného schématu aplikace

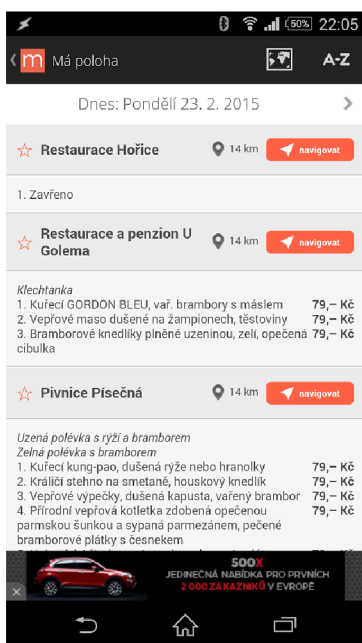
Negativa uživatelského rozhraní

- zdržující úvodní obrazovka s rozcestníkem
- strohé, přehuštěné a nezájímavé podání informací formou pouhého textu (obr. 2.5)
- příliš obtěžující reklama

2.1.5 Swarm

Swarm je aplikace fungující ruku v ruce s Foursquare. Přestože nemá se stravovacím průmyslem příliš společného, některé obrazovky mě zaujaly, byť je jejich zaměření trochu odlišné. Inspirativní je pro mě především obrazovka související se sociálním jádrem aplikace.

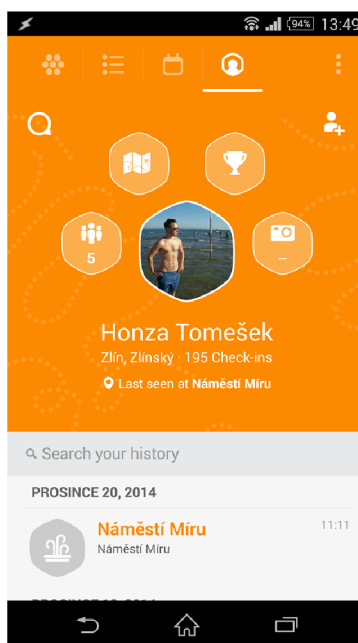
Profilová obrazovka na obrázku 2.6 smysluplně spojuje vše, co s osobou uživatele aplikace souvisí. V centru se nachází profilová fotografie včetně základních údajů, jíž intuitivně obklopují související funkce, které aplikace nabízí. Dole je pak rolovací seznam navštívených míst a při pohybu v rámci tohoto seznamu jsou profilové informace odsunuty z obrazovky, aby nepřekážely a uživateli zůstal zobrazen pouze obsah, který v daném okamžiku potřebuje.



Obrázek 2.4: Meníčka



Obrázek 2.5: Meníčka



Obrázek 2.6: Swarm

2.1.6 Závěr o existujících řešeních

Analýza vybraných existujících řešení ukázala, že zaměření mé aplikace na jednotlivé pokrmy a jejich hodnocení je stále originálním a nerealizovaným nápadem, který by při nasazení aplikace do reálného provozu mohl přinést konkurenční výhodu. Všechna zmíněná řešení nabízejí přinejlepším hodnocení podniků jako celků, přičemž uvedené hodnocení nemusí vůbec odpovídat kvalitě konkrétního pokrmu, který strávnick obdrží.

V souvislosti s uživatelským rozhraním je z mého pohledu nejčastějším neduhem nadměrný počet obrazovek, a to především v případech, kdy všechny tyto obrazovky čekají na uživatele hned po spuštění aplikace. Vzhledem k tomu, že i já osobně jsem v těchto aplikacích občas ztrácel přehled a schopnost orientace, rozhodl jsem se co nejvíce zjednodušit uživatelskou zkušenost po spuštění mé aplikace, tedy omezit počet obrazovek ve vstupní/hlavní části aplikace a především zvolit pouze takové obrazovky, které jsou v aplikaci klíčové.

Dále jsem poměrně často vnímal jakýsi stereotyp a nezajímavost dlouhých seznamů pokrmů, které obsahovaly pouze názvy jídel a jejich cenu. Takové obrazovky mne nedokázaly ujistit, že zvolený pokrm dostanu v takovém stavu, v jakém bych si jej dostat přál, a především těmto obrazovkám chyběla živost a schopnost vyvolat chuť k jídlu. Použití fotografií pokrmů všude, kde se to bude nabízet, tedy považují za nutnost.

Naopak způsob zobrazování dat u aplikace Menička mi připadal velmi užitečný. Přítomnost denního menu přímo u položky restaurace velmi zrychlovala získávání informací z aplikace, když nebylo nutné vždy proklikávat na detail podniku pro zobrazení denního menu. V upravené a odlehčené formě by tento princip mohl velmi zefektivnit uživatelský zážitek.

2.2 Designové principy platformy Android

Pro většinu operačních systémů lze nalézt specifikaci doporučených návrhových vzorů pro tvorbu rozhraní. V tomto ohledu nezaostává ani systém Android, navrch ovšem přidává tzv. *designové principy*, tedy jakési myšlenky či obecně filozofii, která by měla být do rozhraní otisknuta.

Designové principy si kladou za cíl ukázat směr a navzájem přiblížit uživatelská prostředí všech aplikací (především třetích stran), které uživatel na této platformě může nalézt. Uživatelská rozhraní na systému Android by se měla řídit třemi základními principy. Ve volném překladu lze tyto principy nazvat jako „Okouzli mě“ (ang. *Enchant me*), „Zjednoduš mi život“ (ang. *Simplify my life*) a „Udělej ze mě hvězdu“ (ang. *Make me amazing*). Každý z těchto principů se kromě své základní filozofie skládá z jakýchsi pod-pravidel. Zde uvádím úzký výběr těch, na které se nejvíce zaměřím.

Kompletní a detailní popis těchto principů společně s doplňujícími ilustracemi lze nalézt na stránce *Android Design Principles* [1].

2.2.1 Princip „Okouzli mě“ (*Enchant me*)

Pro splnění tohoto principu je potřeba soustředit se na vizuální zpracování rozhraní hned na několika úrovních. Rozhraní by mělo obsahovat intuitivní a rychlé přechody či animace. Rozvržení prvků či volba fontu textu musí být smysluplná a dobře vstřebatelná. Velký důraz je kladen i na tvorbu ikony aplikace, kterou je potřeba vyladit do nejmenších detailů jako by šlo téměř o umělecké dílo, které si uživatel na první pohled zamiluje.

„Potěš mě neočekávaným způsobem“ (*Delight me in surprising ways*)

Pravidlo vyzývá návrháře rozhraní, aby využíval vhodně umístěné animace nebo přesně načasované zvukové efekty, které v uživateli vzbudí dojem, že se vše děje s lehkostí a že je aplikace jakousi vyšší všudypřítomnou silou.

V aplikaci proto budu hojně využívat drobné animace, aktivované především při dotyku obrazovky a představující vizuální zpětnou vazbu pro potvrzení určité uživatelské akce.

„Dovol mi si vše přizpůsobit“ (*Let me make it mine*)

Lidé mají ze své podstaty rádi drobná přizpůsobení, jelikož jim pak prostředí připadá více přátelské a cítí nad ním kontrolu. Je vhodné zvolit šetrné počáteční nastavení aplikace, ale následně dovolit volitelné úpravy, které ovšem nesmějí nijak narušit provádění hlavních úkonů v aplikaci.

Jako důsledek tohoto pravidla plánuji dát uživateli možnost nastavit si, která obrazovka aplikace pro něj bude vstupní. Dále uvažuji nad možností barevně si upravit některé prvky v rozhraní. Zde je ovšem nutné být obezřetný, protože ve výsledku by měla aplikace vždy primárně používat barvy reprezentující celou službu.

2.2.2 Princip „Zjednoduš mi život“ (*Simplify my life*)

Tento princip klade důraz na intuitivnost a efektivitu práce s rozhraním. To zahrnuje například okamžité pochopení hlavních možností, které aplikace nabízí, hned při prvním použití. Tato jednoduchost a pochopitelnost ovšem musí sahat i za první použití a vždy by měla v uživateli vytvářet pocit jednoduchosti tím, že mu skryje méně líbivé činnosti jako jsou správa souborů nebo synchronizace dat. Provádění jednoduchých úkolů by mělo vyžadovat téměř nulovou interakci a splnění komplexnějších úkolů by mělo být ušito na míru lidskému uvažování a způsobu ovládání zařízení. Uživatel by měl vždy cítit, že má aplikaci pod kontrolou, nezávislé na jeho věku či zkušenosti. Rozhraní by jej nikdy nemělo zahltnout přílišným rozhodováním a volbami či zbytečnými upozorněními.

„Obrázky vypoví více než slova“ (*Pictures are faster than words*)

Na patřičných místech by měly být pro vyjádření myšlenek vždy upřednostněny obrázky a další grafické formy oproti textu. Obrázky na sebe stahují pozornost, jejich vyjadřovací schopnosti bývají oproti slovům výrazně lepší a jsou rychleji vstřebatelné.

Z tohoto důvodu budu v aplikaci používat především v oblasti navigace různé obrázky a ikony, kterými nahradím či doplním textové popisky.

„Zobrazuj jen to, co potřebuji, a když to potřebuji“ (*Only show what I need when I need it*)

Je snadné zahltnout mysl uživatele, pokud je mu předloženo mnoho možností a voleb najednou. Je vhodné rozdělit úkony a informace na stravitelnější části. Rozhraní by mělo skrývat možnosti, které v daném kontextu nejsou uplatnitelné či nezbytné, a zbylé možnosti uživatele učít s postupem času.

Jak je patrné i z konceptu aplikace v kapitole 3.1, v aplikaci se budu snažit na hlavních obrazovkách zobrazit minimum informací. Pouze pokud bude mít uživatel zájem o podrobnosti, může volbou daného prvku přejít na samostatnou obrazovku s detailem, kde mu bude poskytnuto více informací, a to přehledněji a na větší ploše.

„Vždy bych měl vědět, kde se nacházím“ (*I should always know where I am*)

Při pohybu aplikací se uživatel musí vždy orientovat. Musí si být vědom, jak se na danou obrazovku dostal. Je vhodné jednotlivé části aplikace od sebe mírně a smysluplně odlišit a používat animované přechody mezi obrazovkami. Pokud nějaká činnost stále ještě probíhá, měl by o jejím stavu uživatel získávat zpětnou vazbu.

Pro usnadnění orientace budu v aplikaci používat panely pro přepínání obrazovek, na kterých bude vždy zvýrazněna aktuální obrazovka. Stejně tak u šipek umožňujících návrat na předcházející obrazovku použiji textové popisky vysvětlující, na jaké obrazovce se uživatel aktuálně nachází.

„Nesmíš přijít o mé věci“ (*Never lose my stuff*)

Aplikace by si měla pamatovat uživatelské výtvořby či volby a umožnit k nim přístup odkudkoliv. Uživateli to navíc nezpůsobí komplikace při změně hardwaru a uchování jeho činnosti zvýší šance, že aplikaci bude opět používat.

Tento princip naše služba, a tedy i moje aplikace, dodržuje z důvodu použití vlastní databáze a serveru, na kterém budou všechny hodnocení, komentáře, mezilidské vazby či fotky uchovány. Následně je potřeba zajistit pravidelnou a včasnou synchronizaci dat.

2.2.3 Princip „Udělej ze mě hvězdu“ (*Make me amazing*)

Aplikace by měla dát svému uživateli příležitost zkusit s ní nové věci, používat ji lehce inovativním způsobem. Je důležité vzít v úvahu možnosti multitaskingu¹ či sdílení dat mezi aplikacemi navzájem na platformě Android. Právě tyto technické možnosti společně s dobře promyšleným rozhraním umožní uživateli provádět neočekávané toky práce a skutečně posunout jeho produktivitu o kousek dál. Stejně tak by se měl uživatel cítit v aplikaci pánem a mít možnost si ji personalizovat pro svůj specifický případ používání. Aplikace by jednoduše měla poskytovat přístup k náročným technickým činnostem s elegancí.

„Umožni mi triky, které fungují všude“ (*Give me tricks that work everywhere*)

Pro lidi je pocit z toho, že na něco přijdou sami, velmi uspokojující. Dosažení tohoto pocitu jim lze usnadnit, pokud rozhraní používá podobné vizuální vzory (rozmístění prvků) či ovládání jako jiné aplikace platformy Android. Mezi tyto prvky patří například ovládání gesty / tažením prstu.

Nejen kvůli tomuto pravidlu chci použít pro navigaci v aplikaci tzv. *tabs* (panel přepínatelných záložek), které jsou v Android aplikacích velmi časté.

„Drobná povzbuzení“ (*Sprinkle encouragement*)

Rozhraní by mělo uživatele stále ujišťovat v tom, že jeho počínání vede ke kýženému výsledku. Komplexní úkoly je vhodné roztříštit na jednodušší kroky, které lze lehce splnit a na které může aplikace reagovat drobnou zpětnou vazbou.

Především nenásilně ale všudypřítomné zpětné vazby uživatele ujišťují, že na něj aplikace reaguje, a uživatel v závislosti na této zpětné vazbě upraví své následující kroky. Do této problematiky patří i upozornění, že některé akce nelze provést. Například při pokusu o posunutí se gestem (*swipe*) za poslední obrazovku na okraji zobrazím drobnou animaci naznačující překážku.

¹možnost v průběhu práce přepínat tam a zpět mezi různými činnostmi (zde aplikacemi)

„Zrychli a usnadni důležité akce“ (*Make important things fast*)

Ne všechny akce/funkce aplikace jsou stejně důležité. Je potřeba zvážit, které ovládací prvky jsou v rozhraní nejdůležitější, a umožnit k nim okamžitý přístup.

Jelikož bude častou činností v aplikaci vyhledávání, jedná se například o ikonu pro vyhledávání přístupnou na horním panelu vždy, když v daném kontextu bude dávat smysl.

2.3 Trendy a návrhové vzory mobilních uživatelských rozhraní

Přestože na trhu s operačními systémy figuruje současně hned několik odlišných mobilních systémů, trendy rozhraní, které se v aplikacích pro tyto systémy objevují, jsou si ve svém základu velmi podobné napříč všemi těmito systémy. Jednou z hlavních příčin této podobnosti jsou malé rozměry displejů, kde není příliš mnoho prostoru pro okázalé prvky rozhraní, ale naopak je potřeba vždy přijít s co nejefektivnějším způsobem podání informací. Ať už se jedná o přehlednou a úspornou navigaci nebo obyčejný seznam položek vyplňující co nejvíce prostoru užitečnými daty.

V souvislosti se zaměřením této práce se dále budu věnovat vybraným trendům a návrhovým vzorům přímo pro platformu Android. Nebude-li však uvedeno jinak, lze základní principy těchto vzorů úspěšně aplikovat/nalézt i u dalších hlavních mobilních platform, například iOS společnosti Apple.

Velmi aktuální pohled na současné trendy a návrhové vzory mi nabídla kniha *Mobile UI Design Patterns* [2], která vysvětluje použití nejvýznamnějších návrhových vzorů na příkladu těch nejpobulárnějších aplikací současnosti. Hlavním zdrojem informací pro tuto podkapitulu pro mě však byla oficiální vývojářská stránka *Patterns* [3].

Níže vybírám návrhové vzory, které budu v práci nejvíce uplatňovat.

2.3.1 Struktura aplikace

Mobilní aplikace se typicky skládají ze dvou typů obrazovek – „top level“ obrazovek na nejvyšší úrovni hierarchie, které zpřístupňují odlišnou funkcionalitu, a „detail“ obrazovek níže v hierarchii obrazovek, které slouží ke konzumaci či vytváření obsahu. Mezivrstvou pak mohou tvořit tzv. „category“ obrazovky vhodné pro rozsáhlé a členité aplikace, kde usnadňují přechod mezi „top level“ a „detail“ obrazovkami.

Na platformě Android by „top level“ obrazovky měly dodržovat dvě základní pravidla. Za prvé by měly přímo nabízet to, co bude s největší pravděpodobností typický uživatel dané aplikace chtít po zapnutí provést. Neměly by se zde objevit obrazovky pouze s navigací dále do aplikace. Za druhé by každá obrazovka měla používat tzv. *action bar*, tedy horizontální panel v horní části, viditelný minimálně na všech hlavních obrazovkách aplikace. Je to specifická komponenta rozhraní na Androidu a hraje klíčovou roli. Slouží k propagaci značky díky zobrazení loga, jsou do ní umístěny prvky pro navigaci, a pokud aplikace nabízí vyhledávací funkcionalitu, její použití je také zpřístupněno zde.

Pohyb mezi „top level“ obrazovkami může být umožněn třemi základními způsoby. Použitím dříve zmíněných *tabs*, což je samostatný, fixně umístěný panel členěný na několik záložek, mezi kterými se lze přepínat pomocí gest do stran. Dále použitím otevíracího menu (*spinner*) přímo v *action baru*, vhodného pro různé pohledy na stejná data (například dny kalendáře v týdenním/měsíčním/ročním náhledu). Třetím způsobem je *navigation drawer*, což je menu výsuvné z levého okraje obrazovky, které lze vyvolat kdekoli v aplikaci a může obsahovat odkazy na velké množství cílových obrazovek, včetně hlouběji zanořených.

2.3.2 Navigace

Android má poměrně důmyslný systém navigace, především co se týče navigace zpětné, tedy k dříve navštíveným obrazovkám. Jeho složitost vychází především z použití dvou prvků (šipek) pro návrat. První šipka se nachází na pevném systémovém panelu a její funkcionalita je klasický návrat v historii, ve zpětném chronologickém pořadí obrazovek, dokonce však i před použití aktuální aplikace, pokud tato byla vyvolána z aplikace jiné. Podobné chování lze nalézt i u ostatních mobilních systémů. Navíc je zde však v *action baru* přítomna druhá šipka, která vždy zajišťuje návrat v hierarchii obrazovek o jednu úroveň výše. Tato kombinace ovládání poskytuje bohaté možnosti, bohužel pro mnoho běžných uživatelů je spíše matoucí. Největší využití pak nabízí v situaci, kdy se uživatel během realizace určité činnosti najednou ocitne zanořený hluboko v jiné aplikaci. Následně má možnost navrácení buď v rámci prováděné činnosti, nebo se může v aktuální aplikaci dostat na její „povrch“.

2.3.3 Gesta

Gesta se v mobilním světě za poslední roky stala standardem pro interakci s rozhraním. Z lidského pohledu se jedná o mnohem přirozenější způsob ovládání, kdy rozhraní (nebo konkrétní vyobrazené objekty) ovládáme napřímo dotykem namísto využití nějakého mezi-prostředku (myši). S gesty se lze setkat snad naprosto ve všech moderních mobilních operačních systémech. Android jich v současnosti podporuje 8 a nezanedbatelná část se shoduje s gesty ostatních systémů. Slouží k výběru položek, zvětšení/zmenšení obsahu nebo k reorganizaci objektů na obrazovce.

Pro uživatele jsou gesta opravdu intuitivní především proto, že jejich použití často kopíruje reálné činnosti (odstranění objektu proběhne přidržením prstu, přesunem nad virtuální koš a následným uvolněním objektu). Aby však tyto činnosti gesta co nejvíce připomínala, musí na ně rozhraní poskytovat patřičnou zpětnou vazbu. Touto zpětnou vazbou jsou animace. Myšlenky a koncepty stojící za tvorbou profesionálních animací představují samostatnou oblast ke studiu. K nahlédnutí pod pokličku tohoto umění jsem využil především článku *A New Mobile UX Design Material* [4], který vysvětluje 12 základních principů uplatněných v animacích. Doplnující informace pak poskytuje například článek *Gestures & Animations* [5].

2.3.4 Více-panelové obrazovky

Android nabízí modulární přístup k tvorbě obrazovek, což následně umožňuje ideální využití prostoru různě velkých displejů. Pokud vývojář na tyto možnosti pamatuje, nic mu nebrání ve vystavění rozhraní složeného z pod-obrazovek, se kterými lze libovolně manipulovat. Účel tohoto přístupu je pak patrný při porovnání rozhraní na malém smartphonu a velkém tabletu. Například v případě adresáře smartphonu klasicky zobrazuje pouze seznam kontaktů, pro detail určité osoby je potřeba se přepnout na zcela jinou obrazovku. Za to tablet může seznam vždy zobrazovat v levé části a v pravé pouze ukazovat aktuálně zvolený kontakt, což urychluje pohyb v aplikaci.

2.3.5 Notifikace

Systém upozornění platformy Android je ve svém základu opět totožný s jinými mobilními systémy. Při vybraných událostech je upozornění uživateli zobrazeno na horním stavovém

řádku, který lze vysunout a na kterém si lze prohlédnout podrobnosti. Notifikace se typicky skládá z ikony aplikace, která ji vytvořila, z titulku, časového razítka a dodatečných informací. Na Androidu lze pomocí gesta notifikaci roztáhnout a zjistit si výrazně více informací, včetně např. fotografií. Interaktivita je prohloubena i přítomností tlačítek pro rychlé akce přímo v notifikaci. Od Androidu 5.0 lze navíc notifikace zobrazovat i na uzamknuté obrazovce, kde s nimi lze přímo interagovat (např. zahazovat pomocí gesta).

2.4 Material Design

Material Design představuje nový designový směr, případně vizuální jazyk, rozhraní, který společnost Google začala propagovat v červnu roku 2014 na vývojářské konferenci Google I/O. O zhmotnění této designové filozofie usiluje především u operačního systému Android v jeho nejnovější verzi 5.x Lollipop.

Díky důkladné práci vývojářských týmů na knihovnách pro zpětnou kompatibilitu je ale možné vytvářet shodná rozhraní i pro starší verze tohoto systému a přesně této možnosti při své práci využiji i já.

Cílem celého přístupu je především snaha o sjednocení uživatelského zážitku napříč platformami a obrazovkami různých rozměrů. Material design je metaforou pro přístup k tvorbě rozhraní, který respektuje zásady fyzikálního světa, reálného prostoru a chování objektů při pohybu v tomto prostoru.

Obsah této kapitoly staví především na oficiální *specifikaci Material Designu* [6].

2.4.1 Filozofie Material Designu

Material design se inspirovuje v použití papíru a inkoustu. Inspirace v takto primitivních prostředcích má za následek velmi jednoduché a snadno vstřebatelné rozhraní. Znázornění povrchů a hran objektů kopíruje realitu a umožňuje uživateli jednoduše pochopit, co rozhraní reprezentuje. Vše **respektuje základy fyziky**. Použití (nebo spíše simulování) zdrojů světla, různých typů povrchů a pohybu prvků rozhraní je klíčové pro správné vyjádření toho, kde se objekty nacházejí, a jak se pohybují v prostoru ve vztahu k ostatním objektům. Realistické použití světla vytváří stíny, což umožňuje přirozené vizuální oddělení jednotlivých elementů rozhraní.

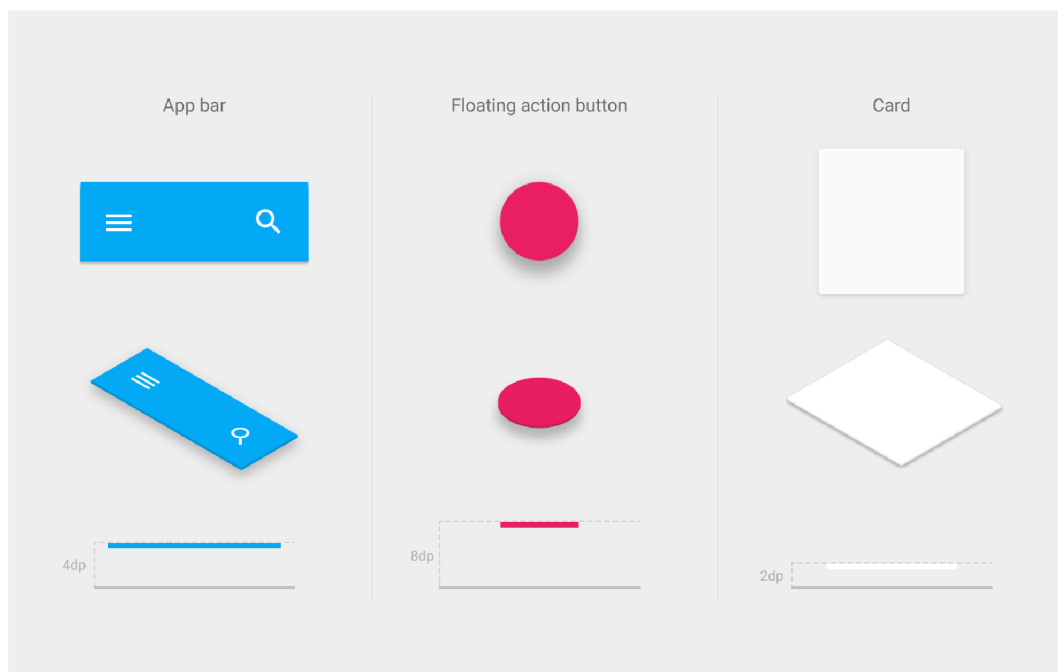
Material design je směšlý v použití **pestrých barev**, obrázků, které zabírají co nejvíce prostoru, nebo **výrazné typografie**.

Veškeré změny v rozhraní jsou doprovázeny pohybem a **animacemi**, přičemž iniciátorem pohybu je vždy uživatel a epicentrem je místo dotyku. Všechny objekty se tak pohybují, transformují a reorganizují bez narušení plynulosti, díky čemuž uživatel neztrácí orientaci v prostředí.

Při umísťování objektů do prostoru se nově využívá i třetí, **z-souřadnice**, tedy vyvýšení objektu nad ostatními. V kombinaci se zdroji světla pak objekty **vrhají stíny**, odpovídající jejich vyvýšení, v určité směru či stíny měkké.

Velmi často se jako základní element rozhraní volí **karty**. Karty jsou ploché, rovinné, s ostrými hranami, a na základě svého vyvýšení také vrhají stín, čímž jsou vizuálně odděleny od dalších karet (reprezentujících jiné objekty).

Google přímo doporučuje vybrat barvy pro rozhraní (konkrétně tři) z jeho předpřipravených palet. V Material designu by se měla využít jedna základní barva pro hlavní prvky rozhraní, její tmavší odstín pro vedlejší prvky (např. systémové panely) a jedna doplňující barva (např. pro ovládací prvky). V mém případě se jedná o odstíny oranžové a zelené.



Obrázek 2.7: Komponenty UI vyvedené v Material Designu, jejich stíny a vyvýšení v klidovém stavu [6]



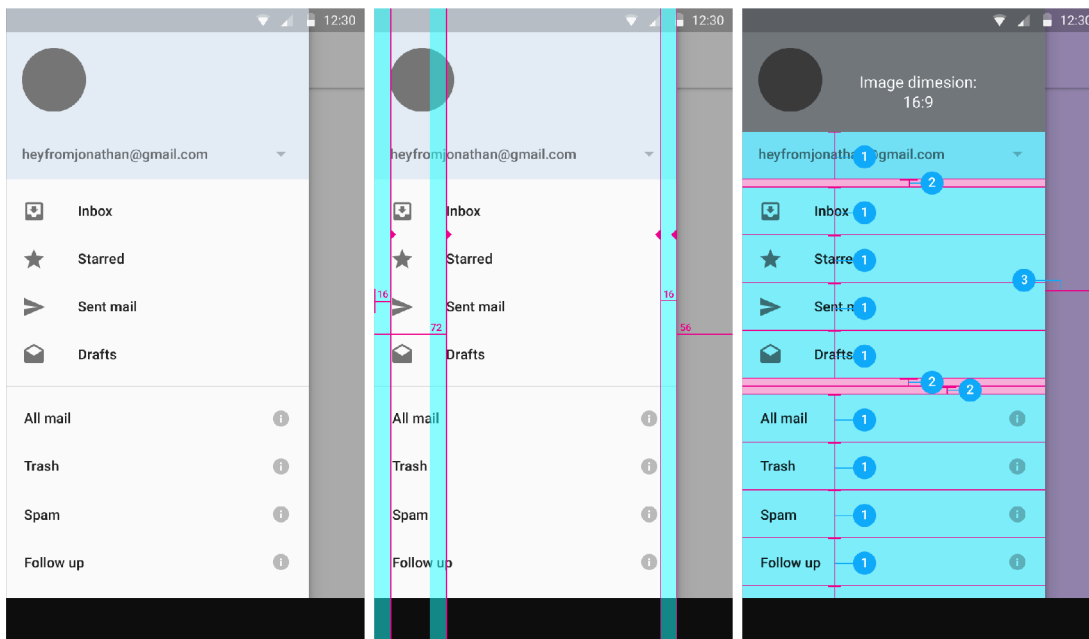
Obrázek 2.8: Výběr barev pro aplikaci

2.4.2 Metriky a mřížky pro Material Design

Snaha Googlu o sjednocení vzhledu aplikací, které jsou pro Android dostupné, zachází až do nejmenších detailů, jak dokazují přesně definované metriky a mřížky pro rozmístění a rozměry jednotlivých elementů uživatelského rozhraní. Lze v nich najít definice fontů a velikostí písma, odsazení elementů, poměr stran fotografií, rozměry ikon a mnohé další. Jako příklad uvádím na obrázku 2.9 specifikaci pro *navigation drawer*², který v aplikaci budu pravděpodobně využívat, a tedy se těmito metrikami řídit. Stejně tak jsou ale předpřipraveny metriky pro další typy obrazovek, jako jsou seznam položek (např. seznam restaurací) či detail určité entity (např. detail pokrmu). V neposlední řadě existují specifikace pro jednotlivé elementy rozhraní jako jsou karty, tlačítka nebo modální okna.

Stoprocentní respektování těchto zásad ovšem velmi potlačují originalitu aplikace a je potřeba hledat určitý kompromis mezi zapadáním do světa Androidu a realizací vlastního nápadu. Proto i já některé části aplikace vytvořím zcela dle vlastních představ.

²navigační panel, který lze vysunout gestem od levého okraje obrazovky



Obrázek 2.9: Metriky pro *navigation drawer* [6]

2.5 Programování pro platformu Android

2.5.1 Použitelná vývojová prostředí

Mezi nejrozšířenější způsoby programování aplikací pro operační systém Android patří využití **vývojového prostředí Eclipse** či **prostředí Android Studio**. Obě tato prostředí jsem v rámci studia dostupných nástrojů osobně vyzkoušel.

Jako první jsem pracoval ve vývojovém prostředí **Eclipse** od The Eclipse Foundation ve verzi Luna Service Release 1 (4.4.1). V rámci něj jsem využil stažitelné pluginy pro efektivnější práci v jazyce Java (*Java Development Tools – JDT*). Jelikož Eclipse jako takové je aplikace založená na Javě, vyžaduje použití nějakého Java virtuálního stroje (*Java Virtual Machine – JVM*). Konkrétně alespoň *Java Runtime Environment (JRE)*, aby mohlo fungovat, případně *Java Development Kit (JDK)* pro vývoj. V mém konkrétním případě, z důvodu použití Eclipse verze Luna, se jednalo o Java 7 JRE/JDK. Dalším velmi potřebným nástrojem je plugin *Android Development Tools (ADT)*, který rozšiřuje schopnosti Eclipse o programování pro Android. Umožňuje pohodlné založení nového Android projektu, vytvoření uživatelského rozhraní, debugování (ladění) aplikace a exportování aplikace (APK souboru) pro distribuci. Posledním zásadním dílem skládačky je pak sada *Android Software Development Kit (Android SDK)*, která obsahuje vývojářské nástroje a knihovny potřebné pro tvorbu Android aplikací.

V počátcích práce se Eclipse jevilo jako nejvhodnější volba. Bylo zavedeným a velmi rozšířeným nástrojem v poměrně odladěném stavu a internet byl plný rad a tipů pro práci v tomto prostředí. Naproti tomu úhlavní konkurent Android Studio byl stále ještě v beta verzi a nepříliš rozšířený. Na slabinu tohoto prostředí jsem však narazil při práci s externími knihovnami, kde Eclipse vyžadovalo přímé importování knihoven do pracovního adresáře a celá automatická správa těchto knihoven a jejich aktualizací na novější verze mi často působila potíže, když jsem nemohl využívat funkcionalitu dostupnou v nejnovější verzi určité knihovny a Eclipse neuznalo za vhodné tuto verzi použít.

Právě z tohoto důvodu jsem později začal využívat **Android Studio**, které v té době již opustilo beta verzi a dostalo se do finální podoby. Jeho sestavovací nástroj **Gradle**, který sám dohledává potřebné knihovny na základě závislostí definovaných ve skriptu, vyřešil předchozí potíže. Tento sestavovací nástroj je přímo integrován do vývojového prostředí a je vůbec největším rozdílem oproti Eclipse. Android studio je vývojovým prostředím z dílen samotného Googlu a aktuálně je oficiálním prostředím pro vývoj Android aplikací. Založeno je na **IntelliJ IDEA** a přidává mnoho další funkcionality. V době mého přechodu bylo ve verzi 1.0. Stejně jako Eclipse vyžaduje i Android Studio sadu *Android SDK*.

Definitivním impulzem pro setrvání u tohoto prostředí pak bylo ukončení aktivního vývoje *ADT pluginu* pro Eclipse, aby se mohl Google plně soustředit právě na Android Studio. V současné době je z mého pohledu Android Studio vývojovým prostředím přesně šitým na míru vývoji Android aplikací, které vývojáři usnadňuje práci na každém kroku. Od možnosti definovat uživatelské rozhraní ve vylepšeném grafickém editoru s náhledem, přes možnost namíchat barvy pro prvky rozhraní namísto pouhé definice v kódu, až po předpřipravené šablony pro jednotlivé typy zdrojových souborů, pokročilé doplňování kódu a refactoring³.

2.5.2 Programování a vlastnosti platformy Android

I když jsem ve svém konkrétním případě nakonec zvolil prostředí Android Studio, jak dále popisují v kapitole 4.1, programování a organizace souborů se řídí obecnými zvyklostmi práce s **jazykem Java**, ve kterém valná většina implementace probíhá.

Z toho vyplývá, že pro tvorbu Android aplikací je potřeba osvojit si dvě základní dovednosti. Obecně programování v jazyce Java a specifitěji programování s využitím Android frameworku. Literatura zaměřená na programování pro Android si často vyhrazuje část prostoru právě pro vysvětlení základů jazyka Java, jak je tomu v případě *Programming Android* [7]. Kromě této knihy jsem pro načerpání základních znalostí o implementaci pro Android a o vlastnostech systému samotného využil knih jako jsou *Android 4* [8] nebo *Beginning Android 2* [9]. Tuto tištěnou literaturu lze rozhodně doporučit pro studium základních principů a technik programování, jelikož uvedené příklady implementace jsou prověřeny časem. Zároveň však tato četba velmi rychle zastarává, jelikož vývoj mobilních operačních systémů je opravdu rapidní, a často tak mohou být některé pasáže neaktuální již v době, kdy se kniha dostává na pulty prodejen. Užitečnost této literatury utrpěla velký zásah především nyní, kdy se oficiálním vývojovým prostředím stalo Android Studio, přičemž tyto knihy se často soustředí na práci v prostředí Eclipse. Z těchto důvodů je pro řešení konkrétnějších problémů a seznámení se s novinkami vhodnější čerpat informace např. z *oficiálních vývojářských stránek pro Android*, viz [10], ze kterých jsem čerpal i já.

V jedné ze složek projektu, v Android Studiu označené `java/`, se udržují veškeré zdrojové kódy napsané v Javě, sdružované do logických celků – balíčků (*packages*). Tento princip usnadňuje orientaci například při využívání balíčků třetích stran po jejich pouhém nakopírování do tohoto adresáře. Další velmi významnou složkou je složka `res/` (*resources*) obsahující veškerá data, která nejsou přímo kódem a která jsou označována jako zdroje/suroviny (řetězce, styly, obrázky, ...).

Architektura operačního systému Android je rozdělena do **5 vrstev**. Nejnižší leží **jádro operačního systému**, založené na Linuxovém jádře, vytvářející mezivrstvu mezi hardwarem a softwarem vyšší úrovně. Další vrstvou jsou **knihovny**, jejichž funkcionality je vývojáři poskytnuta prostřednictvím Android aplikačního frameworku. **Android Runtime**

³proces úpravy zdrojového kódu s cílem zlepšit jeho čitelnost a obecně kvalitu, bez dopadu na chování

je vrstva, která se liší v závislosti na verzi systému. Do verze 4.3 zde byl používán virtuální stroj *Dalvik*, od verze 4.4 se pak používá *ART* s dopřednou kompilací. V této vrstvě se nacházejí také základní knihovny jazyka Java. Z pohledu vývojáře je nejpodstatnější vrstvou tzv. **aplikační framework**. Jedná se o vrstvu, která pro aplikaci zpřístupňuje mnoho služeb. Nalezneme zde přístup k prvkům uživatelského rozhraní, umožnění přístupu k datům dalších aplikací, přístup ke zdrojům (řetězce, bitmapy, ...), použití stavového řádku či řízení životního cyklu aplikací. Nejvyšší vrstvou systému jsou pak **běžné aplikace**.

Určení návrhového vzoru, který by naprosto přesně odrážel filozofii programování aplikací pro Android, je poněkud obtížné. Názory vývojářů se často liší, nicméně v hrubé podstatě se jedná o klasický koncept *Model-View-Controller (MVC)*. Abych byl konkrétnější, uživatelské rozhraní je definováno v **XML souborech** (*View*), stejně tak jsou v těchto souborech popsány různé zdroje (řetězce, ...). Tzv. *Activity*, napsané v Javě, zde fungují jako *Controller* a využívají právě zmíněných XML souborů.

Obecné informace k projektu jako jsou vyžadovaná oprávnění v telefonu, informace o vzájemném vztahu obrazovek, verze aplikace nebo třeba využívaná API se zapisují do souboru **AndroidManifest.xml**. Další informace jako verze systému Android, na kterou vývojář cílí nebo potřebné knihovny třetích stran se zapisují do sestavovacího skriptu **build.gradle**.

Běžná, plně funkční obrazovka aplikace se z pohledu zdrojových kódů, které musí vývojář napsat, skládá nejméně ze dvou souborů. Tím prvním je třída odvozená od *Activity*, která reprezentuje logiku obrazovky a nachází se libovolně ve složce `java/`. Ta během uživatelského používání aplikace prochází jednotlivými fázemi svého životního cyklu. Ty se mění na základě toho, jestli je obrazovka spuštěná, viditelná, minimalizovaná nebo třeba plně ukončená. V závislosti na těchto fázích je potřeba patřičně hospodařit se zdroji fyzického zařízení (např. zajistit, aby aplikace zbytečně nepoužívala modul GPS, když je minimalizována). Druhý nutný soubor je nejčastěji obsahem složky `res/layout/` a definuje rozložení a vzhled grafických prvků v rámci obrazovky.

Doplnění k *Activity* se objevilo ve verzi Android 3.0 a nazývá se *Fragment*. Přichází s novou filozofií ve znovupoužitelnosti částí aplikace/kódu. *Fragmenty* se principiálně chovají stejně jako *Activity*, představují logiku obrazovky a procházejí fázemi životního cyklu. Rozdíl je v tom, že jedna *Activity* se může skládat z více *Fragmentů*, přičemž tyto *Fragmenty* lze dynamicky za běhu přidávat, vyměňovat či odstraňovat. Tento princip je velmi přínosný při zohledňování různě velkých displejů během programování aplikace. Pokud jsou obrazovky naimplementovány jako *Fragmenty*, lze je následně opakovaně využívat v různých layoutových souborech⁴ příslušejících různě velkým displejům. Na menším mobilním zařízení pak může být například zobrazen v jeden časový okamžik vždy jen jeden *Fragment*, zatímco na tabletu mohou být zobrazeny dva *Fragmenty* vedle sebe.

⁴soubory definující rozložení a rozmístění prvků rozhraní

Kapitola 3

Návrh aplikace pro hodnocení kvality pokrmů

Náplní této kapitoly je především návrh klíčových obrazovek aplikace, jejichž průchodem nakonec uživatel nalezne to, *co opravdu hledá*, případně zjistí, *co vlastně chce*. Tedy **jaký pokrm** si opravdu chce objednat, a to na základě podrobných hodnocení, fotografií či ceny pokrmu. Případně také zjistí, **kde** si jej má objednat.

Primárním zdrojem informací a podkladem pro vybudování dále popisovaného rozhraní mi byla kniha *Designing the user interface* [11]. Mezi klíčová témata v souvislosti s návrhem rozhraní, která tato kniha rozebírá, bych zařadil vymezení vlastností, které má výsledné rozhraní splňovat, uvažování rozličné základny uživatelů či zařízení a snaha zajistit univerzální použitelnost. Kniha zmiňuje například 8 zlatých pravidel návrhu rozhraní nebo 4 základní pilíře takového návrhu. Více jsem z této knihy čerpal také v oblasti různých stylů interakce s rozhraním (přímá manipulace s objekty, menu, dialogy, ...).

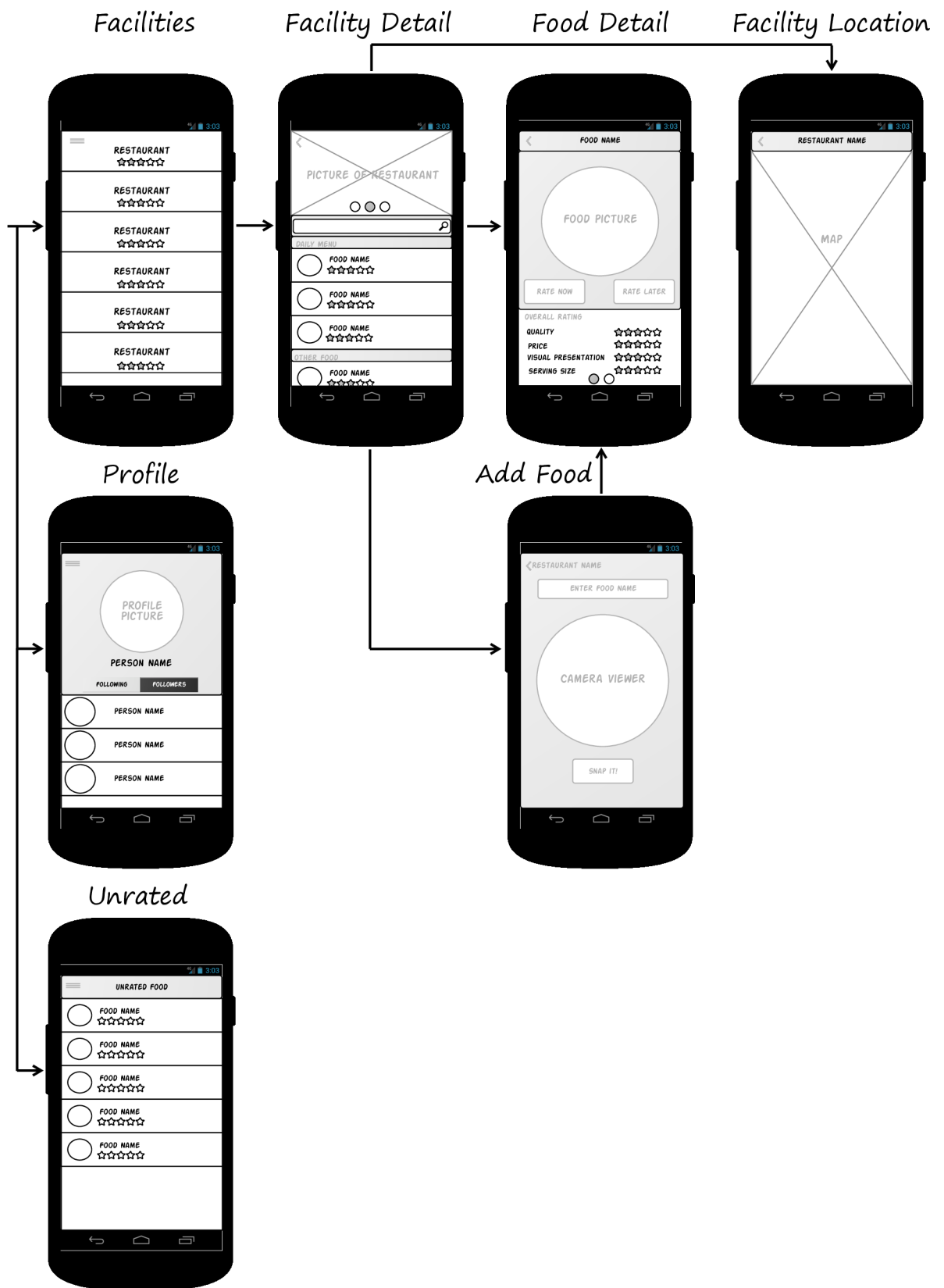
Další poznatky z praxe a tipy pro dobrou použitelnost rozhraní mi poskytla kniha *Pixel Perfect Precision* [12], která se mimo jiné věnuje i návrhu s ohledem na uživatele trpící poruchou zraku, sluchu či jinými omezeními.

3.1 Koncept aplikace

Schéma na obrázku 3.1 zachycuje rozvržení základních obrazovek aplikace. Pro obrazovky budu v této práci používat jejich anglická označení, která jsou většinou úspornější a jasně odlišitelná od okolního textu. Obrazovka vlevo nahoře představuje úvodní obrazovku celé aplikace. Obrazovky v levém sloupci dohromady tvoří obrazovky, mezi kterými se dá v aplikaci na nejvyšší úrovni jednoduše přepínat. Směrem doprava jsou vyobrazeny obrazovky na nižší hierarchické úrovni, většinou zobrazují detaily různých objektů.

3.1.1 Vstupní obrazovka (*Facilities*)

To, že chce uživatel v široké škále nabízených pokrmů nalézt ten aktuálně vytoužený, ještě neznamená, že může vybírat ze všech pokrmů světa. Tou nejvýraznější překážkou je uživatelská poloha. Jestliže chce vyrazit na rychlé obědové menu, jistě pro něj nebude relevantní výběr jídel z odlehlých koutů země. Svou cestu k hledanému pokrmu (o kterém uživatel ještě **dost možná neví, který to vlastně je**) proto uživatel začíná na obrazovce *Facilities*, která mu zobrazuje nejbližší stravovací podniky seřazené podle jejich celkového hodnocení, přičemž kromě názvu restaurace zobrazuje vždy ještě její fotografii a vzdálenost od uživatele.



Obrázek 3.1: Koncept aplikace

Volba úvodní obrazovky ovšem natolik ovlivňuje charakter aplikace, že je zcela možné, že tato obrazovka bude v průběhu práce ještě předmětem změny.

3.1.2 Detail restaurace (*Facility Detail*)

Výběr restaurace je prvním důležitým krokem. Dle vzdálenosti, hodnocení podniku nebo jeho kategorie uživatel poprvé zužuje výběr pokrmů, a specifikuje tak své chutě. I když jde uživateli především o samotné jídlo, **prostředí podniku** je pro něj jistě důležitým faktorem. V horní části obrazovky jsou mu proto k dispozici fotografie vybraného podniku, mezi nimiž se může pohybovat tažením prstu do stran. A když už v této fázi zkoumá spíše podnik, nabízím mu v této sekci i informace jako jsou otevírací doba či adresa.

To jsou sice informace užitečné, ovšem ne jedinečné (a tedy cenné). Graf, který se v této sekci také nachází, ovšem jedinečný (a tedy cenný) je. Odráží vývoj spokojenosti zákazníků s tímto podnikem za poslední dobu. Graf zobrazuje hodnoty celkového hodnocení restaurace vzorkované například v týdenních intervalech. Znázornění tohoto vývoje může být užitečné jak pro provozovatele podniku jakožto podnět pro změny, tak především pro strávnicka, který vidí, co se v poslední době s úrovní podniku děje. Na základě těchto dat přichází v úvahu také vytvoření obrazovky, která bude uživatelům doporučovat podniky, které se v poslední době zásadně zlepšily a (opět) stojí za návštěvu.

Hlavní (spodní) část obrazovky pak tvoří rolovací seznam jídel, členěný na jednotlivé kategorie, který si lze hezky celý projít. Je však možné, že uživatel předem ví, co chce. Proto je mu pod horní sekci k dispozici *search bar* (vyhledávací panel), s jehož využitím může filtrovat/vyhledávat jídla ve spodní části obrazovky. Ta nabízí v patřičnou denní dobu i meníčka a dále běžná jídla, která podnik nabízí.

Obsah vyplňující tuto obrazovku musí ovšem nejdříve vzniknout. Případně musí být průběžně aktualizován. Někdo (uživatelé) jej musí tvořit. Pro přidání záznamu o pokrmu slouží tlačítko na samém konci rolovacího seznamu. Toto umístění jsem zvolil, jelikož by mělo částečně zamezit vytváření duplikátů již existujících jídel. Uživatel buď nejdříve musí projít celým seznamem jídel a ke tlačítku se dostane pouze, pokud se dříve nezastavil u hledaného jídla, případně se během vyhledávání jídla na této obrazovce podle zadávané hodnoty filtruje seznam jídel, až na konec zůstane pouze prvek pro přidání nového jídla.

Proces přidávání nového jídla, včetně jeho hodnocení, vysvětluji detailněji v podkapitole 3.1.7. Co se souvisejících obrazovek týče, po kliknutí na výše zmíněné tlačítko pro přidání pokrmu je uživatel přenesen na obrazovku s hledáčkem fotoaparátu (*Add Food*), kde navíc musí vyplnit název jídla a může ihned pořídit i fotografii. Následně má možnost vytvořené jídlo ohodnotit okamžitě či hodnocení odložit na později.

3.1.3 Poloha restaurace (*Facility Location*)

Až se bude později chtít uživatel dopravit k vybrané restauraci, bude nejprve potřebovat zjistit, kam že to vlastně míří. Proto se z obrazovky detailu restaurace může přenést na mapu nejbližšího okolí dané restaurace včetně vyznačení restaurace samotné, případně k ní aktivovat navigaci.

3.1.4 Detail jídla (*Food Detail*)

Uživatel shledal pokrm na tolik vyhovujícím, že se rozhodl prohlédnout si jeho detail. Podstatný je první dojem, a tak zde nejvýrazněji vystupuje fotografie, kterou lze dotykem zvětšit přes celou obrazovku, a následně si prohlížet i případné další dostupné fotografie.

To, co však uživateli definitivně pomůže s výběrem pokrmu, se nachází ve spodní části obrazovky. Zde má uživatel k dispozici přepínatelnou sekci zobrazující různá hodnocení daného jídla. Její první část, označená jako *Overall* (celkové hodnocení), zobrazuje zprůměrovaná hodnocení od ostatních uživatelů pro příslušné kategorie. Těmi jsou **kvalita, přiměřenost ceny, vizuální prezentace a velikost porce**. Stejně tak si pomocí rolování níže může procházet komentáře hodnotitelů. Druhá část této sekce, dostupná po tažení prstu doleva, zobrazuje dřívější hodnocení daného pokrmu přihlášeným uživatelem.

Po jídle by uživatel mohl chtít tento pokrm i ohodnotit. Pro uživatelský vstup jsou k dispozici tlačítka *Rate Now* a *Rate Later*, jejichž volbou uživatel udává, kdy bude chtít hodnocení provést. Pokud jej chce ohodnotit ihned (tlačítko *Rate Now*), je mu předloženo modální okno, ve kterém tažením prstu vyplní hodnocení v podobě hvězdiček pro čtyři výše jmenované kategorie.

Nebo by uživatel mohl ještě před jídlem využít alternativní volby ohodnocení později (tlačítko *Rate Later*). Ta je vhodná právě v situacích, kdy uživatel jídlo teprve dostane a jeho ohodnocení chce odložit až na okamžik, kdy jídlo zkonsumuje. Při této volbě je ušetřeno modálního okna pro hodnocení. Toto jídlo je však přidáno do jeho osobního seznamu zatím neohodnocených jídel (viz. podkapitola 3.1.5) a je naplánována notifikace (viz. podkapitola 4.4.5), která mu v patřičné chvíli hodnocení připomene.

3.1.5 Obrazovka s nehodnocenými jídly (*Unrated*)

Jelikož aplikace staví zejména na datech od uživatelů, rozhodl jsem se do aplikace zařadit obrazovku speciálně pro seznam ještě neohodnocených jídel. To jsou taková, která si uživatel nejdříve vyhledal a následně u nich zvolil ohodnocení později, například až bude po jídle. Cílem obrazovky je pak nenásilně upozorňovat, že je potřeba ještě dokončit hodnocení, což je pro aplikaci velmi kritické.

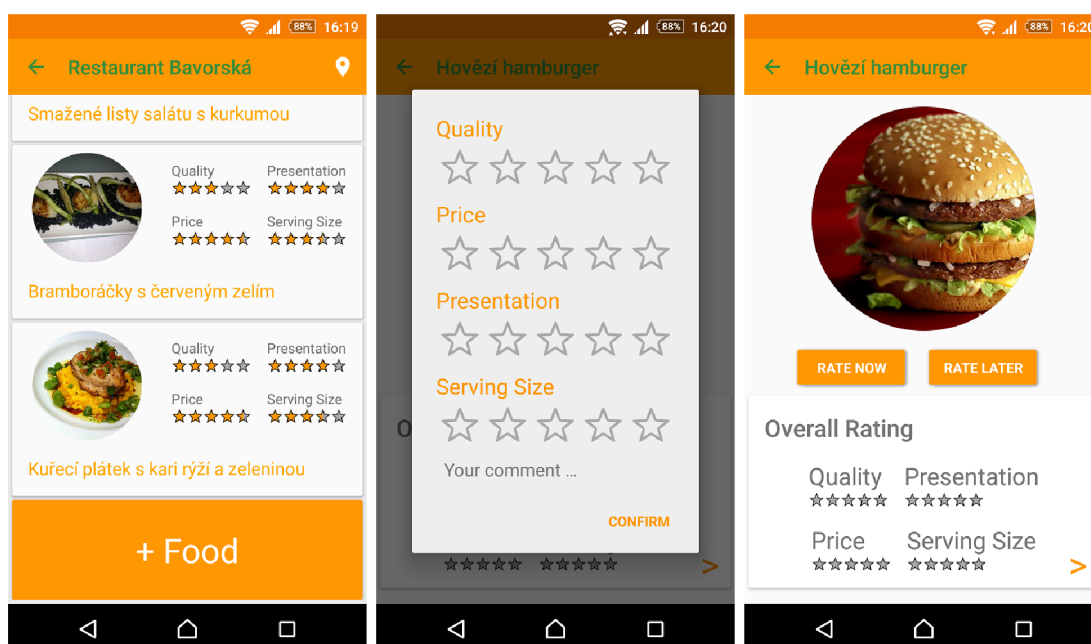
3.1.6 Profilová obrazovka (*Profile*)

Aby celá aplikace mohla řádně fungovat a mohla být někomu prospěšná, je nejdůležitější vybudovat kolem ní základnu uživatelů, která bude záznamy o pokrmech vkládat a přidávat k nim hodnocení. Proto v aplikaci neopomím ani sociální funkce a gamifikační prvky, jejichž centrem je právě profilová obrazovka. Ta poskytuje náhled na profilovou fotografii, základní údaje o uživateli a v dolní části tři seznamy, mezi kterými se lze přepínat. První seznam nabízí všechna jídla, která uživatel již hodnotil, a to chronologicky seřazená. Druhý zobrazuje další uživatele, které přihlášený uživatel sleduje, a naopak třetí seznam poskytuje ty uživatele, kteří sledují uživatele přihlášeného. Sledovaným a sledujícím uživatelům se dále věnuji v podkapitole 3.3.1.

3.1.7 Proces vytváření a hodnocení jídla

Naše služba stojí a padá na databázi jídel a jejich hodnocení, kterou je nutné naplnit. To má za úkol proces vytváření jídla, který může být obecně velmi komplexní. Mým cílem je jej co nejvíce zefektivnit a ušít na míru uživatelům, například snížit počet potřebných obrazovek, a neodradit tak uživatele od jeho provedení. S tímto úmyslem jsem tomuto procesu během práce věnoval zvýšenou pozornost a rozhodl se navrhnout několik koncepčně odlišných způsobů pro vytvoření záznamu o jídle. Výsledkem jsou tři různé způsoby. Z pohledu požadovaných informací jsou si všechny způsoby rovny, jedná se pouze o jiný způsob prezentace.

Přechod na obrazovky pro vytváření nového jídla je pro všechny způsoby stejný a probíral jsem jej v podkapitole 3.1.2. Obecně na těchto obrazovkách požadují od uživatele dva základní údaje. Prvním je název jídla, pokud možno shodný s názvem uváděným na jídelním lístku. Tento údaj je povinný. Dalším „údajem“ je fotografie, kterou zde může uživatel pořídit, jelikož je dostupný i hledáček fotoaparátu. K fotografování ovšem není nucen, protože je taková činnost v restauraci stále pro mnoho lidí nepříjemná. Po pořízení fotografie má vždy uživatel možnost focení opakovat pro dosažení kýženého výsledku. V okamžiku, kdy je s výsledkem spokojen, se může přesunout k samotnému hodnocení pokrmu. To je opět vizualizováno ve formě hvězdiček a probíhá tažením prstu, dokud není vyplněn požadovaný počet hvězd.



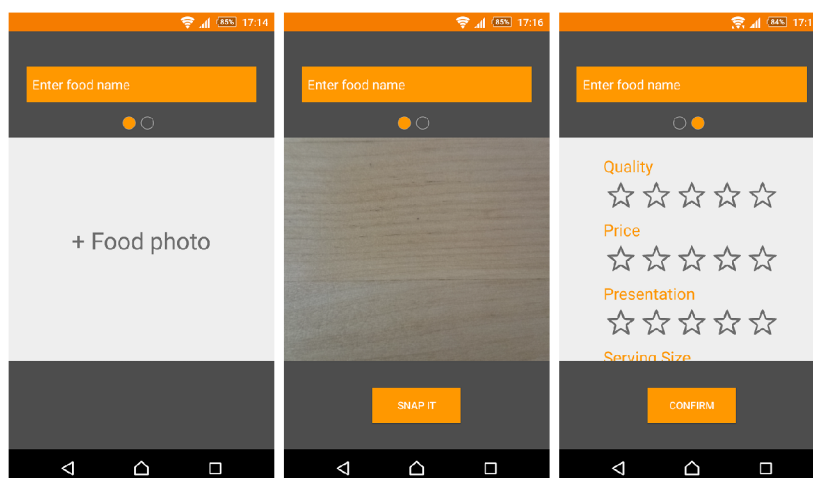
Obrázek 3.2: Společná výchozí obrazovka pro vytvoření pokrmu, standardní dialog pro hodnocení a společná konečná obrazovka s detailem vytvořeného pokrmu

Dále má uživatel možnost vyplnit krátký slovní komentář k hodnocení. K této možnosti jsem se uchýlil proto, že často nelze vyjádřit své názory pouze pevně přednastavenými kategoriemi. Uživatel by také mohl chtít zanechat vzkaz nebo podnět ke zlepšení přímo podniku. Komentáře jsou také jedním z kandidátů na monetizaci aplikace, když by na ně majitelé podniků mohli reagovat po uhrazení drobného poplatku.

Po jednoduchém potvrzení vyplněných údajů je uživatel přenesen na detail nově vytvořeného jídla.

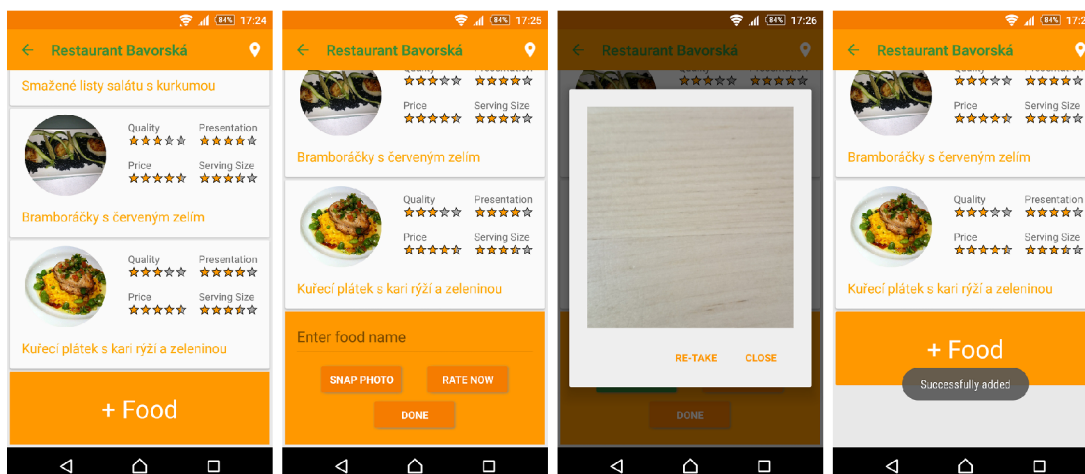
Jednotlivé navržené způsoby přidání nového pokrmu se pak liší ve zpracování části mezi přechodem na tvorbu jídla a koncovým potvrzením vyplněných údajů. Pokud bych měl jednotlivé způsoby nějak charakterizovat, pak bych tyto způsoby označil jako *moderní*, *rychlý* a *pomalý*.

Moderní způsob využívá jediné obrazovky a zaměřuje se na ovládání pomocí gest do stran, s jejichž použitím mění jednotlivé komponenty rozhraní za jiné (například panel s hledáčkem fotoaparátu za panel s hodnocením).



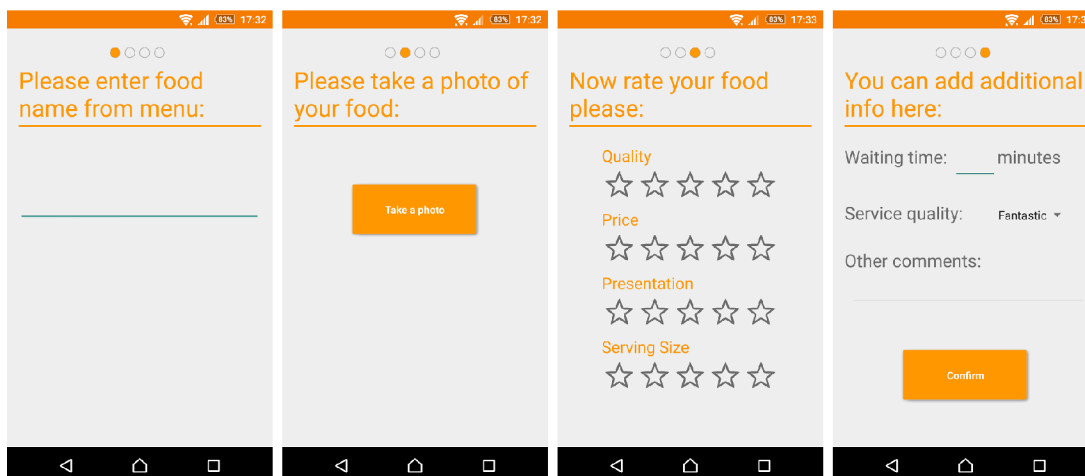
Obrázek 3.3: *Moderní způsob* přidání nového pokrmu

Rychlý způsob dokonce žádnou další obrazovku nevyužívá. Po aktivaci tlačítka pro přidání jídla na obrazovce detailu podniku je toto tlačítko s využitím animací transformováno na textové pole pro vyplnění názvu pokrmu a sadu tlačítek. Celkově se tento způsob spoléhá na využití modálních/dialogových oken, ve kterých probíhá hodnocení, případně zobrazení pořízených snímků. Po potvrzení údajů uživatel zůstává na stávající obrazovce a může pokračovat v činnosti, aniž by byl přenesen na detail nově vytvořeného pokrmu.



Obrázek 3.4: *Rychlý způsob* přidání nového pokrmu

Pomalý způsob, jak už název napovídá, nikam nespěchá a na uživatele netlačí. Skládá se ze čtyř po sobě jdoucích obrazovek, které s klidem a dostatkem prostoru uživateli vysvětlí, co se po něm aktuálně požaduje. Celkově je tak tento způsob složen z několika samostatných kroků, díky kterým by uživatel při hodnocení neměl být ztracen.



Obrázek 3.5: *Pomalý* způsob přidání nového pokrmu

3.2 Možné alternativy v návrhu rozhraní

3.2.1 Vstupní obrazovka – přesná lokalizace vs. seznam restaurací

Volba vstupní/úvodní obrazovky je naprosto klíčová, jelikož utváří první dojem v uživateli, a proto se snažím o její pečlivý výběr. Alternativou k výše zmíněnému seznamu restaurací by mohla být obrazovka s detailem restaurace, ve které se uživatel nejpravděpodobněji nachází. Aplikace by tak mohla na uživatele zapůsobit jakýmsi inteligentním dojmem, když by sama poskytla detail zařízení, do kterého se host vydal.

Celá tato myšlenka má ovšem několik zásadních úskalí. Zejména v husté městské zástavbě, kde se může nacházet na malém prostoru mnoho stravovacích zařízení, by mohla přesnost lokalizace kolísat, a často by se tak uživateli vnutila nesprávná restaurace, z jejíhož detailu by se musel nějak dostat pryč. To by si vyžadovalo vyvinout a vhodně zakomponovat pohodlný systém pro rychlé přepínání podniku v situacích, kdy dojde k nepřesné lokalizaci uživatele. Stejně tak chování v situaci, kdy se uživatel nachází na kilometry daleko od jakékoli restaurace, je rozporuplné. V takové situaci by bylo nejvhodnější zobrazit pouze seznam restaurací, byť vzdálených. Tím ovšem vzniká zásadní nekonzistence v chování aplikace po jejím spuštění. V městském prostředí by se otevíral detail restaurace zatímco na kraji města pouze seznam restaurací.

Z tohoto důvodu nebudou snaha o přesnou lokalizaci a následné automatické zvolení restaurace zahrnuty do výsledné aplikace.

3.2.2 Úvodní obrazovka – seznam restaurací vs. seznam jídel

Přestože jsem v předcházející podkapitole 3.2.1 zahrnul variantu přesné lokalizace uživatele, samotná obrazovka se seznamem restaurací nemusí být finální volbou. Alternativou by se mohla stát obrazovka se seznamem dobře hodnocených pokrmů z blízkého okolí. Jednalo by se o obrazovku, jejímž obsahem by bylo zhruba dvacet nejlépe hodnocených pokrmů napříč okolními restauracemi, například pod označením *Top Around*, a celkově by tak obrazovka přinášela jiný přístup k poskytování informací uživateli.

Seznam restaurací by zůstal zachován a byl přístupný přímo z této úvodní obrazovky, pouze by již nehrál hlavní roli.

3.2.3 Volba hlavních obrazovek a jejich přepínání

Za hlavní obrazovky zde považuji ty, mezi kterými se dá přímo přepínat na nejvyšší úrovni hierarchie obrazovek. Podle počtu těchto hlavních obrazovek je nutné zvolit i mechanismus jejich přepínání (navigaci). V mém případě zde řadím **vstupní obrazovku** (se seznamem restaurací v okolí), **profilovou obrazovku** a **obrazovku s neohodnocenými jídly**. Pro tento relativně nízký počet hlavních obrazovek jsou ideální volbou pro jejich přepínání dříve zmiňované *tabs*. Jedná se o fixní lištu, na které jsou umístěny záložky reprezentující odkazované obrazovky. Stiskem takovéto záložky dojde k přechodu na danou obrazovku. Alternativou pro přepínání by mohl být ze strany vysouvací *navigation drawer*, který je ovšem pro takto nízký počet hlavních obrazovek zbytečně komplikovaný.

Zdůvodnění volby hlavních obrazovek je nasnadě. O přítomnosti vstupní obrazovky, pointy celé aplikace, mezi těmito hlavními obrazovkami, nelze pochybovat. Profilovou obrazovku zde řadím proto, že na ní uživatel nalezne jak své přátele, tak své dosavadní úspěchy v rámci aplikace. Tyto prvky považuji za klíčové, jelikož by měly zajistit, že uživatelé budou aplikaci používat dlouhodobě a pravidelně. Poslední hlavní obrazovkou jsou neohodnocená jídla. Jak doufám, přímý přístup k této obrazovce bude mít za následek intenzivnější plnění naší databáze hodnoceními, jelikož neprovedená hodnocení budou mít uživatelé stále na očích.

Odlišná situace by nastala po zařazení obrazovky *Top Around* zmiňované v předcházející podkapitole 3.2.2. Čtyři hlavní obrazovky jsou již poměrně vysokým počtem a navigace pomocí *tabs* pro takový případ již nevyhovuje. Nabízí se tak alternativní strukturování aplikace. Na nejvyšší úrovni by se pomocí *tabs* dalo přepínat pouze mezi seznamem okolních restaurací (*Facilities*) a seznamem kvalitních pokrmů z okolí (*Top Around*), čímž by byla uživateli po spuštění předložena pouze klíčová funkcionalita aplikace. Vše ostatní, včetně profilové obrazovky, seznamu neohodnocených jídel i třeba nastavení, by bylo přístupné pomocí výsuvného *navigation draweru*, který by tedy nabízel vedlejší funkcionalitu.

3.3 Rozšiřující funkce

Aplikace se primárně snaží být zdrojem informací o kvalitě pokrmů a stravovacích zařízeních. Toto zaměření si nutně nevynucuje vytvářet kolem svého jádra sociální (či jiné) prvky. V naší službě to ovšem vidím jako velmi užitečnou až nutnou záležitost. Naše služba je nadstandardně závislá na svých uživateli, jelikož ti jsou zdrojem informací, které aplikace následně nabízí.

3.3.1 Sledující/sledování

Umožnění vazeb a interakce mezi jednotlivými uživatelskými účty by mělo zajistit určitou motivaci pro uživatele, aby aplikaci používali častěji (jelikož budou vnímat, že i jejich známí tak činí), což by mělo vést k rychlejšímu plnění databáze. Neuchýlili jsme se ovšem s kolegy k mechanismu klasického „přátelství“ ve smyslu oboustranně potvrzené vazby mezi uživateli. Jako vhodnější se zde jeví „jednostranné vztahy“. Tedy, že uživatelé mohou sledovat jiné uživatele, aniž by tomu tak muselo být i obráceně. Tento princip velmi pomáhá v situacích, kdy určitou aplikaci používají i významnější osobnosti a velké množství jiných uživatelů se zajímá o jejich činnost (v tomto případě o jejich názory na pokrmy).

3.3.2 Gamifikace

Gamifikace může nesporně přispět k dlouhodobějšímu užívání aplikace, když se uživatel bude snažit dosáhnout určitého cíle či překonat přátele. V žádném případě ovšem nechci vytvořit samoučelný bodový systém bez myšlenky, a k tomu všemu jen proto, aby vůbec nějaký existoval.

Přestože je vytvoření celého gamifikačního modelu a jeho následné provázání s aplikací prací přesahující rámec této bakalářské práce, velmi rád bych jeho základy položil již v počátcích aplikace. Z mého pohledu je pro uživatele poměrně frustrující jedna chyba, kterou stále a stále opakují nově začínající projekty – s gamifikací přijdou až po určitém čase, například po roce. To znamená jediné, věrní uživatelé ze samého počátku služby jsou po roce ve stejné pozici jako nováčci, jelikož jejich aktivitu nikdo nezaznamenával. Z tohoto důvodu považuji za důležité přijít s gamifikací již na počátku, případně alespoň dokázat interně zaznamenávat činnosti, které by v budoucnu mohly být bodovány, a po spuštění gamifikace tyto záznamy již pouze přetavit do potřebné bodové podoby.

Jako „bodovatelné“ činnosti jsem v aplikaci identifikoval hodnocení pokrmu, přidání fotografie k pokrmu a vytvoření nového záznamu o pokrmu.

Kapitola 4

Realizace aplikace pro hodnocení kvality pokrmů

V této části práce vycházím z návrhu rozhraní, který jsem provedl v minulé kapitole. Jeho podobu ovšem upravuji na základě zpětné vazby uživatelů a testování. V samotné implementaci popisují spíše užitečné API a knihovny třetích stran než realizaci rozhraní.

4.1 Základní nástroje použité pro vytvoření aplikace

Pro wireframy/schéma rozhraní jsem zvolil webovou aplikaci **LucidChart**¹. Její výhoda spočívá především ve velké základně předpřipravených grafických prvků souvisejících s prostředím systému Android. Jedná se například o stylizovaná tlačítka, textová pole či dialogová okna. Veškeré výtvořky jsou vektorové a dají se exportovat i do formátu pdf.

Jako integrované vývojové prostředí (IDE) používám **Android Studio**² od společnosti Google ve verzi 1.1.0. Pro jeho běh je nutná instalace *Java Development Kit (JDK)* verze 7 a vyšší. Zároveň zde využívám několik pluginů, o kterých píší dále.

Aby aplikace mohla využívat některé z mnohých služeb poskytovaných Googlem (např. zjišťování polohy telefonu), musí využívat různé **Google Play services API**. Jejich použití na oplátku vyžaduje mít založený projekt aplikace s *Android SDK*. *SDK* rozděluje nástroje a další komponenty do balíčků, které lze dodatečně stáhnout s využitím **Android SDK Manager**. Právě pro výše zmiňované zjišťování polohy telefonu využívám balíček *Google Play services*.

Pro zajištění kompatibility s co nejvíce Android zařízeními (různými verzemi systému Android) pracuji s **Android Support Library**, která obsahuje rozšířenou množinu API, které jsou kompatibilní s většinou verzí systému Android.

Přestože lze funkcionality aplikace a vzhled rozhraní nejlépe ověřit na fyzických zařízeních různých parametrů, z důvodu omezených prostředků to v mém případě není možné, a proto využívám nástroj pro tvorbu virtuálních zařízení běžících na systému Android. Konkrétně se jedná o program **Genymotion**³, který umožňuje tvorbu virtuálních zařízení různých úhlopříček displeje, rozlišení, poměrů stran obrazovky a především různých verzí systému. Genymotion zároveň nabízí plugin pro pohodlné použití přímo v Android Studiu.

¹<https://www.lucidchart.com/personahomepage>

²<http://developer.android.com/tools/studio/index.html>

³<https://www.genymotion.com/#/>

4.2 Foursquare API

V počátcích práce ke svému fungování aplikace napřímo využívala veřejně dostupné API služby Foursquare. Jejich stejnojmennou aplikaci jsem analyzoval v podkapitole 2.1.3. K tomuto účelu je na jejich webových stránkách⁴ potřeba založit jediný účet, který se následně vztahuje na celou aplikaci či celý projekt, nikoliv pouze na jediné uživatelské zařízení. Ke každému účtu je přiřazeno jedinečné **ID** (*Client id*) a **heslo** (*Client secret*), které se v aplikaci používají pro autentizaci při některých podrobnějších dotazech na databázi společnosti Foursquare. Pro použití na Androidu dokonce Foursquare doporučuje i knihovnu *easyFoursquare4Android*⁵, která práci s API zjednodušuje. V souladu s původním návrhem jsem tuto databázi (i zmíněnou knihovnu) využíval jako zdroj míst a podniků, ze kterých jsem uvažoval pouze ty, které nějak souvisí se stravovacím odvětvím. Vytvářet a plnit vlastní databázi samotnými podniky by bylo extrémně náročné a především by se přinejlepším jednalo pouze o duplikát těchto volně přístupných dat. Původní koncept načítání dat do aplikace tedy spočíval v prvotním **zjištění okolních podniků s využitím Foursquare API**. V druhé řadě pak **následoval dotaz na naši vlastní databázi** požadující pokrmy nabízené v daném podniku.

Komunikace s Foursquare probíhá zasíláním **HTTP dotazů**, pomocí metody *GET*, na předem definovaná url (*API Endpoints*) doplněná o upřesňující parametry. Příkladem budiž endpoint pro nalezení podniků v blízkém okolí FIT VUT. Skládá se z adresy API `https://api.foursquare.com/v2/`, specifikace endpointu a související akce `venues/search?` a povinného parametru se zeměpisnými souřadnicemi `ll=49.226995,16.596797`. Pro testovací účely se dále doplňuje dočasný autentifikační klíč v parametru `oauth_token`, v produkční verzi aplikace pak výše zmiňované *Client id* a *Client secret*.

Odpovědi na požadavek jsou data serializovaná do **formátu JSON**. Jako příklad uvádím v příloze B výňatek z odpovědi na výše uvedený dotaz.

Komunikace se dvěma různými API ovšem byla na poměry koncové aplikace poměrně komplexní. Proto jsme se zhruba v polovině práce po dohodě s mými kolegy rozhodli přesunout komunikaci s Foursquare na náš server, a tuto práci tak dostal na starosti kolega zabývající se tvorbou databáze. Důsledkem tohoto kroku je logické vyčlenění procesu shromažďování dat na server. Komunikaci s vlastním serverem dále popisují v podkapitole 4.4.2.

4.3 Úvodní dotazování a prvotní reakce uživatelů

Úvodní dotazování a průzkum si nekladli za cíl získat konkrétní vyčíslitelné výsledky, které bych následně mohl zanést do grafu. Šlo spíše o zdravou diskuzi s cílem získat pohled na celou aplikaci i od potenciálních uživatelů. Po krátkém a svobodném pohybování v naimplementovaném rozhraní aplikace mi uživatelé předali několik zajímavých postřehů.

Tím nejvýznamnějším byla připomínka, že současná úvodní obrazovka (seznam restaurací) možná není tím pravým, co by v takto zaměřené aplikaci ocenili. V souladu s podkapitolou 3.2.2 tedy do aplikace začlením obrazovku *Top Around* s vybranými pokrmy napříč okolními podniky. S tím souvisí i změna struktury aplikace a navigace v ní, především přidáním *navigation draweru*, do podoby, kterou jsem navrhl v podkapitole 3.2.3.

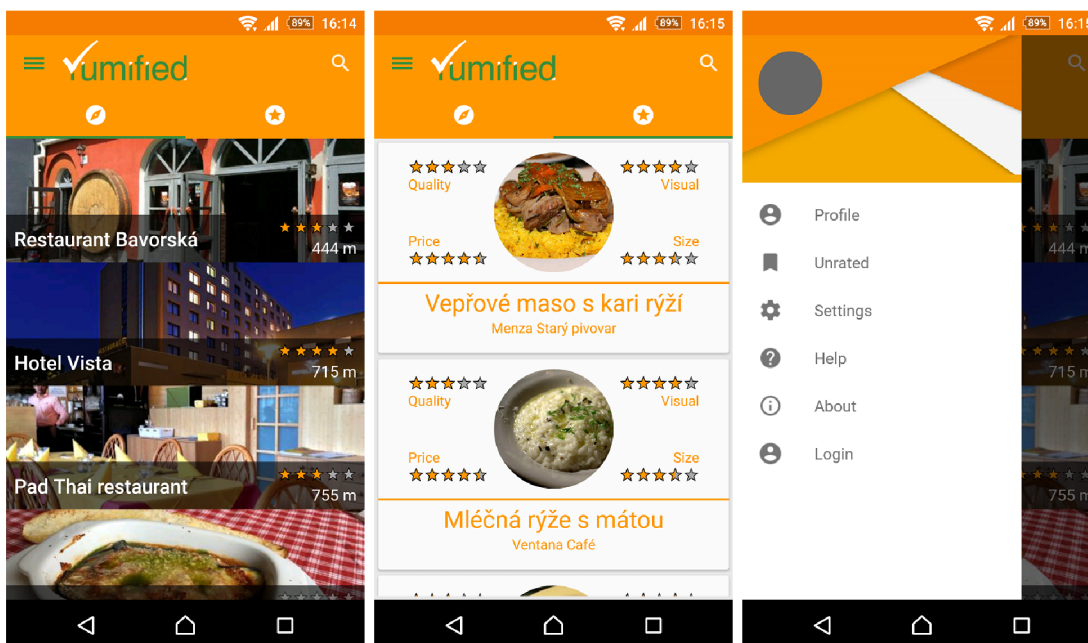
Dále jsem narazil na nedostatky u některých ovládacích prvků (například tlačítek *Rate Now* a *Rate Later*), u kterých někteří testující nejenže nepochopili, k čemu přesně tyto

⁴<https://developer.foursquare.com>

⁵<https://github.com/condesales/easyFoursquare4Android>

prvky slouží, ale jiní ani nerozpoznali, že s nimi lze interagovat. Bude tedy zapotřebí je lépe vizuálně zvýraznit, případně i lépe vyjádřit jejich účel.

Poslední postřeh je v zásadě samozřejmý, nicméně bylo užitečné si uvědomit, jak velké množství uživatelů stále nedisponuje dostatečnými znalostmi anglického jazyka. Z tohoto důvodu bude s ohledem na rozšíření aplikace nutností i česky „komunikující“ rozhraní.



Obrázek 4.1: Nová struktura aplikace (hlavní obrazovky *Facilities* a *Top Around*, použití *navigation draweru* pro navigaci)

4.4 Implementace aplikace a integrace existujících služeb

4.4.1 Přihlašování pomocí Facebooku

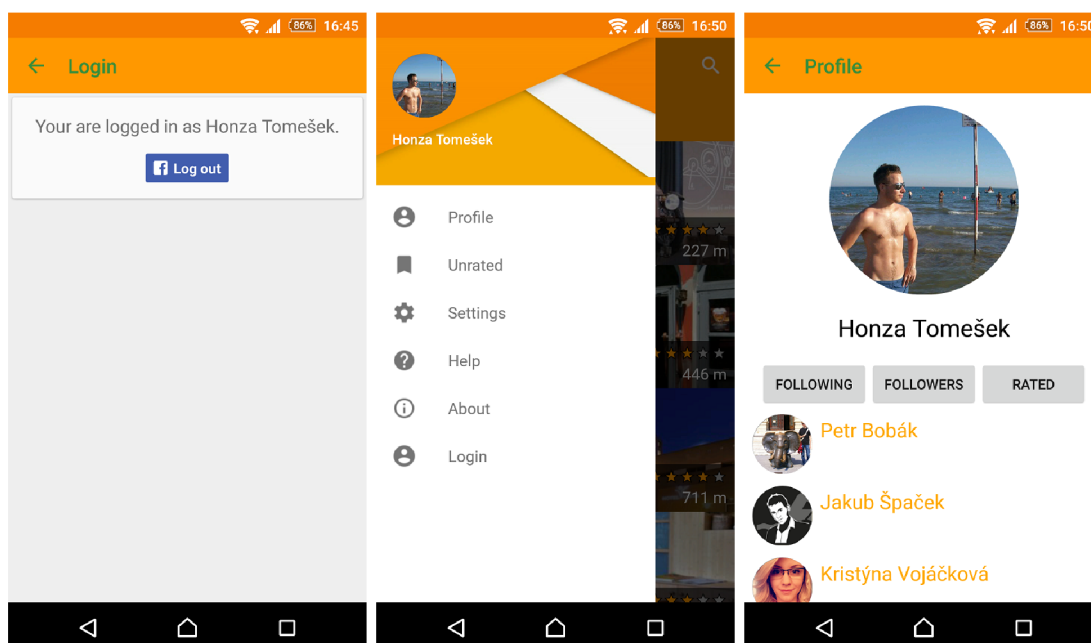
I když je tvorba nových uživatelských účtů nutností, kterou aplikace musí nabízet, uživatelský komfort zajistí teprve jednoduché přihlášení pomocí existujících účtů třetích stran. A to především těch extrémně rozšířených, mezi které lze zařadit Facebook nebo například Gmail. Koncept takového přihlašování do aplikace redukuje uživatelskou námahu na minimum – jeden dotek. Většinou je k tomuto účelu vyhrazeno tlačítko vzhledu typického pro danou službu třetí strany. Pokud je uživatel v této službě přihlášen, může nová aplikace využít informací, které již uživatel do služby dříve zadal. Uživatel tedy nemusí při přihlašování do nové aplikace vyplňovat všechny údaje nanovo.

V současnosti pravděpodobně nejrozšířenějšími účty jsou ty od společnosti Facebook. Ta nyní čerstvě uveřejnila novou verzi svého API pro Android aplikace označenou *Facebook SDK for Android v4.0*. API se soustředí především na přihlašování pomocí Facebook účtu a možnost sdílení informací do sociální sítě Facebook. Pro využití tohoto API je potřeba provést registraci aplikace na příslušných vývojářských stránkách⁶ a také zadat unikátní hashovací klíč příslušející k aplikaci pro následné ověřování aplikace při využívání API.

⁶<https://developers.facebook.com/>

API poskytuje možnost plné programové kontroly přihlašování, udržování sezení a odhlášení, stejně tak ovšem nabízí celou tuto funkcionalitu zapouzdřenou do jediného tlačítka, které lze přidat do grafického rozhraní. Stisk tlačítka pak pouze přihlašuje či odhláší. Pokud má uživatel na svém zařízení nainstalovanou aplikaci Facebook a je do ní přihlášen, po stisknutí tlačítka uživatel nové aplikaci pouze potvrdí přístup k požadovaným informacím a proces je dokončen. Pokud na zařízení aplikace Facebook chybí, je uživateli nabídnuta možnost přihlášení skrze Facebook účet pomocí modálního okna. Následně již aplikace může přistupovat k příslušným údajům.

Osobně jsem se rozhodl pro využití tlačítka zapouzdřujícího přihlašovací funkcionalitu, a to na obrazovce speciálně vyhrazené pro přihlašování, kde mohou v budoucnu přibýt možnosti přihlášení pomocí dalších účtů. Ochota uživatelů sdílet své údaje má samozřejmě své meze, a proto je vhodné vždy požadovat přístup jen k těm nejnужnějším údajům. V mém případě se jedná uživatelské jméno, e-mail a fotografii, které využiji na profilové obrazovce i dalších místech, díky čemuž bude jednoduše rozpoznatelné, kdo je do aplikace přihlášen. Sociální život aplikace se pak pokusím nastartovat s využitím seznamu přátel přihlášeného uživatele a možností zobrazovat jejich uživatelské účty.



Obrázek 4.2: Přihlašovací obrazovka, zobrazení přihlášeného uživatele a profilová obrazovka

4.4.2 Komunikace s vlastním serverem

Pro komunikaci s aplikací je na straně serveru využíván *proces routování*, nebo-li překlad mezi url a odpovídajícím aplikačním požadavkem. Všechny takové dotazovací url jsme s kolegy společně definovali a každé url odpovídá požadavku na jeden typ informací (vytváří jeden *API endpoint*). Tabulka 4.1 shrnuje podporované endpointy pro získávání dat.

Ke každému z endpointů samozřejmě přísluší dodatečné povinné i nepovinné parametry. Například i parametry pro stránkování většího počtu výsledků. Dotaz a připojení na url probíhá pomocí **HTTP protokolu** a jeho metody *GET*, kdy hlavička dotazu obsahuje hodnotu `accept: application-json`.

Endpoint	Popis
facilities	podniky v okolí zadaných zeměpisných souřadnic
toparound	kvalitní pokrmy napříč podniky v okolí zadaných zeměpisných souřadnic
facility	informace o konkrétním podniku
menu	pokrmy nabízené v konkrétním podniku
dish	informace o konkrétním pokrmu
comment	komentáře ke konkrétnímu pokrmu

Tabulka 4.1: Podporované endpointy pro získávání dat

Zde uvádím možný tvar jednoho z endpointů na našem dočasném testovacím serveru:

`http://apitest.g6.cz/web/facilities.json?lat=49&lng=16&page_limit=3&page=2`

Navrácená odpověď je ve **formátu JSON**. Námi definovaný tvar odpovědi na předcházející url je k vidění v příloze C.

Na straně aplikace používám pro komunikaci knihovnu *Volley*, kterou pro jednodušší účely doporučuje i Google na svých vývojářských stránkách. *Volley* nabízí jednoduchý způsob odesílání požadavků a získávání odpovědi v samostatném vlákně. Jelikož vždy požaduji odpověď ve formátu JSON, ve *Volley* vytvářím dotaz pomocí třídy `JSONObjectRequest`, která je pro tento účel ideální. Při konstrukci objektu této třídy předávám cílové url, typ metody HTTP protokolu, doplňuji výše zmíněnou hlavičku požadavku a specifikuji metody, které mají být zpětně zavolány při vyřízení požadavku. Navíc specifikuji i chování jako je *timeout* nebo počet opakování požadavku pro případ potíží s komunikací. Následně již stačí objekt přidat do fronty požadavků a knihovna *Volley* se postará o jejich vyřízení. Pro stahování obrázků používám v kombinaci s *Volley* i cache paměť, ze které se *Volley* pokouší obrázky načíst dříve, než provede síťový požadavek.

4.4.3 Využití Google Maps

Ve svém návrhu, konkrétně ve podkapitole 3.1.3, jsem popisoval možnost zobrazit si podnik na mapě, včetně možnosti nechat se k němu navigovat. K tomuto účelu se dokonale nabízí využít světově známé a populární *Google Maps* společnosti Google. API, které mi umožnilo tyto mapy integrovat do aplikace, se nazývá *Google Maps Android API v2*⁷. Pro jeho použití je však nejprve potřeba provést několik kroků. Především je potřeba mít k dispozici knihovny se samotným API. Ty lze získat v balíku *Google Play services* pomocí **Android SDK Manager**.

Dále je nezbytné vyříditi si **privátní API klíč** (příslušející k aplikaci), který se následně vloží do zdrojových kódů aplikace. Za tímto účelem je potřeba si zjistit digitální certifikát aplikace a také registrovat svůj projekt/aplikaci na stránkách spravujících přístup ke všem Google API. Konkrétně je nutné zjistit zkrácenou formu tohoto digitálního certifikátu, které se říká **SHA-1** (je generována hashovacím algoritmem SHA-1). To lze provést nástrojem **keytool**. Pro registraci projektu je potřeba přes webový prohlížeč navštívit tzv. *Google Developers Console*⁸. Jedná se o stránky pro správu všemožných Google API. Zde je nutno vytvořit nový projekt a pro tento projekt povolit právě *Google Maps Android API v2*. Po vložení SHA-1 již stačí o API klíč zažádat. Jeden API klíč odpovídá jednomu

⁷<https://developers.google.com/maps/documentation/android/>

⁸<https://console.developers.google.com>

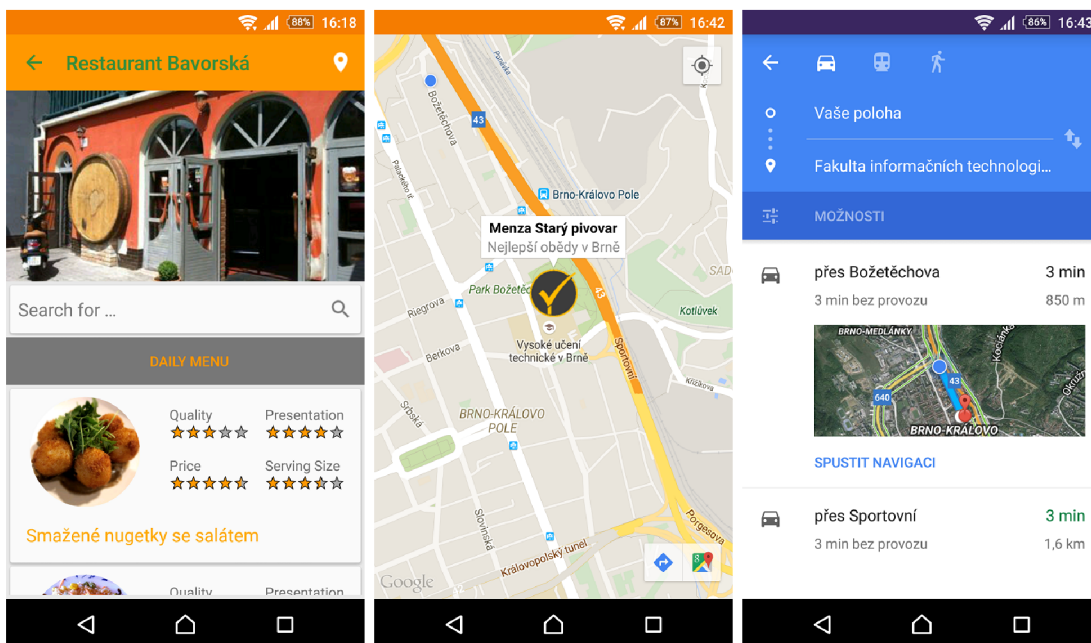
páru certifikát–název balíku s aplikací. Tento klíč se vloží jako speciální značka do souboru **AndroidManifest.xml**.

Integrace obrazovky s mapou do aplikace se významně neliší od přidávání obrazovek vlastních. Rozdíl je pouze v tom, že obrazovka s mapou je již předpřipravena a stačí ji použít.

Mapa je však vysoce upravitelná. Lze zvolit režim zobrazení (klasická/satelitní/3D mapa), úroveň přiblížení, animovat přesuny po mapě, změnit vzhled ukazatelů na místa na mapě či upravit textové popisky.

Celý proces potřebný pro využívání tohoto API lze nalézt na příslušné vývojářské stránce⁹.

Osobně využívám vlastní ukazatel na podniky, který graficky vychází z ikony aplikace a celou aplikaci tak reprezentuje. Pro praktické účely zobrazuji mapu v klasickém režimu a umožňuji návrat na aktuální polohu uživatele pomocí tlačítka, stejně jako spuštění navigace.



Obrázek 4.3: Přechod z detailu podniku na mapu a spuštění navigace

4.4.4 Vlastní prostředí fotoaparátu.

Poněkud odvážnou myšlenku v sobě nese celý koncept přidávání nového záznamu o pokrmu do aplikace, popisovaný například v podkapitole 3.1.7. Jednalo se o možnost zadávání názvu pokrmu i pořizování fotografie na téže obrazovce, což implikuje nutnost tvorby vlastního uživatelského rozhraní pro fotoaparát.

Implementace takového rozhraní je však v ekosystému Android aplikací spíše výjimečná a váže se k aplikacím hlouběji zaměřeným na pořizování snímků. Pro aplikace zaměřené jinak, tedy u kterých fotografie hrají vedlejší roli, Google doporučuje jiný postup. Ten čerpá z otevřenosti systému Android a možnosti vzájemné spolupráce různých aplikací. Jelikož v současnosti již prakticky každý přístroj obsahuje vlastní/vestavěnou aplikaci fotoaparátu,

⁹<https://developers.google.com/maps/documentation/android/start>

lze jednoduše pomocí mechanismu *intentů*¹⁰ přejít pro pořízení fotografie do vestavěné aplikace a po vyfocení snímku lze fotografii použít v původní aplikaci.

Přesto jsem se pro dodržení návrhu rozhodl pro vytvoření vlastního rozhraní. Tento přístup zahrnuje relativně přímou manipulaci se samotným hardwarem zařízení a může přinášet problémy s funkčností na zařízeních různých výrobců. Přestože Google zveřejnil novou verzi API pro manipulaci se snímačem zařízení pojmenovanou *android.hardware.camera2*, tato verze zatím není natolik prověřená časem, abych ji do aplikace zakomponoval. Proto využívám původní verzi *android.hardware.Camera*.

Základy takto zaměřené obrazovky jsou totožné jako u obrazovek jiných. Je tedy potřeba vytvořit *Activity* (obsahující logiku) a xml soubor s rozložením rozhraní. Důležitým elementem rozhraní je pak *SurfaceView*, které umožňuje přenášet náhled ze snímače na obrazovku. Než ovšem v aplikaci náhled spustím, analyzuji nejprve možné hodnoty parametrů pro snímání obrazu a vyberu z nich ty nejvhodnější. Zde patří dostupná rozlišení snímače a také dostupná rozlišení náhledu (displeje zařízení). Stejně tak lze zároveň nastavit třeba automatické ostření či použití blesku. Náhled je také nutno v patřičné okamžiky pozastavovat, například během ukládání pořízené fotografie. Právě ukládání pořízených dat mi přineslo další komplikace. Spadá sem například různé pootočení výsledného snímku na různých zařízeních.

Na druhou stranu mi však vlastní rozhraní poskytlo téměř neomezené možnosti. Mohu vytvářet vlastní tlačítka, poskytovat náhledy s různým poměrem stran, přidávat vlastní animace či efekty nebo takovouto obrazovku zakomponovat do jiné. Hlavním bonusem je pak sladění vzhledu se zbytkem aplikace.

4.4.5 Geolokace a notifikace

Jak jsem již vícekrát zmínil dříve, vyplňování uživatelských hodnocení na jednotlivé pokrmy je pro naši službu kritické. S prováděním hodnocení však souvisí jistý problém. V době, kdy uživatel s aplikací pracuje, ještě není schopen hodnocení provést. Naopak v době, kdy již hodnotit může, nemá potřebu aplikaci znovu spouštět. Konkrétněji řečeno, když si uživatel v aplikaci vyhledá pokrm, který si následně objedná, neměl ještě možnost jídlo ochutnat, a tedy nemůže hodnotit. Následně smartphone odloží. Po konzumaci pokrmu již potřebné „znalosti“ pro ohodnocení má, aplikaci však již pravděpodobně zavřel a nemá důvod se k ní vracet.

Tuto situaci řeším s využitím patřičně načasované, nebo spíše naplánované, notifikace (upozornění). Bezpochyby neuškodí, když aplikace svému uživateli předvede trochu vlastní inteligence a při odchodu uživatele z podniku se jej dotáže, jak mu vlastně chutnalo.

K takovému účelu lze využít tzv. *geolokace*, anglicky *geofences*, což jsou pomyslné kruhové oblasti umístěné do reálného světa, přičemž lze sledovat, jaký je vzájemný vztah mezi polohou uživatele a samotné geolokace. Práce s nimi probíhá přes *location API*. Nejprve je nutno vytvořit samotnou geolokaci (objekt třídy *Geofence*) pomocí objektu *Geofence.Builder*. U ní lze nastavit především zeměpisné souřadnice a taky průměr samotné oblasti. Dále je podstatné určit, na jakou událost má být aplikace upozorněna. Tedy vstup do oblasti, opuštění oblasti nebo setrvání v oblasti po určitou dobu. Dodatečně lze specifikovat i dobu, po jakou má být geolokace monitorována. Vytvořené geolokace se pak jednoduše předají samotnému API, přičemž je zároveň nutno specifikovat, jaká akce (typicky kontaktování určité komponenty aplikace) má být provedena po výskytu sledované události. K tomu slouží

¹⁰Mechanismus předávání objektů typu *Intent* (= *záměr*), které reprezentují operaci, která má být provedena, typicky přechod na jinou obrazovku.

objekt třídy `PendingIntent`, jež je v požadovaný okamžik aktivován.

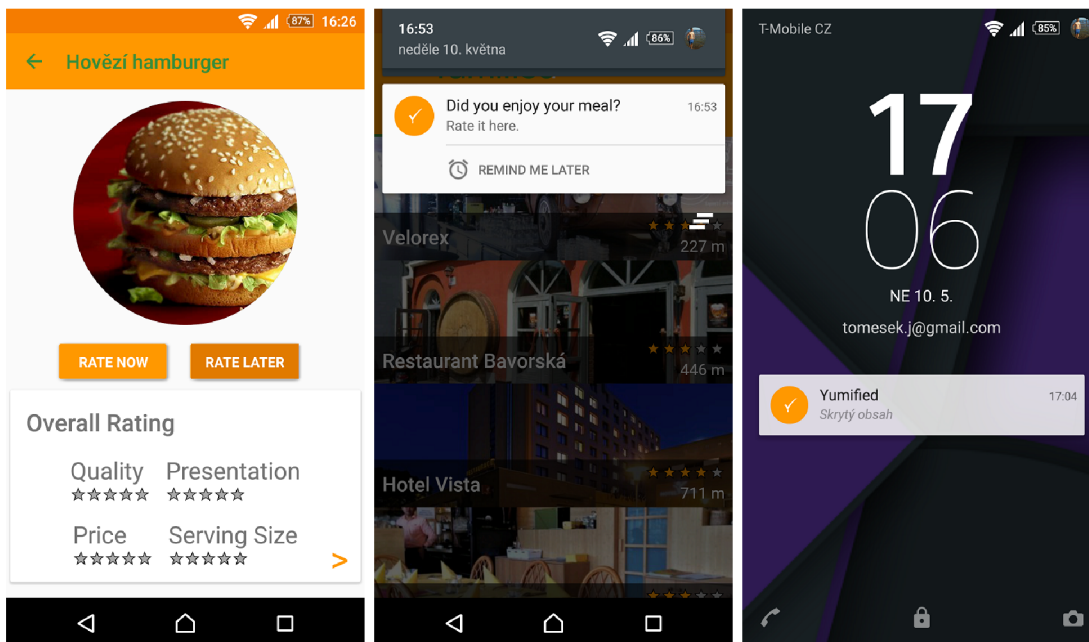
Od okamžiku předání vytvořených geolokací do API se již veškeré monitorování provádí na pozadí systému. Na výskyt události (např. vstup uživatele do zvolené oblasti) lze později reagovat např. s využitím vlastního `BroadcastReceiver`, tedy objektu, který může být kdykoli spuštěn i z procesu běžícího na pozadí. V tomto případě jej spustí `location API` na základě dříve definovaného `PendingIntentu`. Chování `BroadcastReceiveru` je kompletně v rukou programátora.

Celý tento mechanismus je v mé aplikaci využit v okamžiku, kdy si uživatel vybere pokrm a zvolí možnost ohodnocení později (se záměrem ohodnotit jej po konzumaci). V tento okamžik může smartphone odložit. Aplikace ovšem na tuto událost reaguje zjištěním aktuální polohy uživatele, vytvořením nové geolokace na této pozici s průměrem 1 kilometru a nastavením sledování na opuštění oblasti. Jakmile k takové události dojde, ve vlastním `BroadcastReceiveru` vytvořím notifikaci, která se uživatele dotáže na spokojenost s pokrmem. Po stisknutí této notifikace je uživatel přenesen na detail hodnoceného jídla a je mu přímo předloženo modální okno pro hodnocení.

Samotné notifikace jsou jedním z mechanismů operačního systému Android, který byl za léta své existence mnohokrát upravován a vylepšován. S Androidem 4.1 přibyla možnost interakce s notifikací přímo v panelu, který se vysouvá z horního stavového řádku, a to pomocí tlačítek umístěných přímo v notifikaci. Notifikace lze od této verze navíc roztahovat a zmenšovat pro získání dodatečných informací. Další významné posílení notifikací přinesl Android 5.0, který přidal možnost zobrazovat notifikace přímo na zamykací obrazovce.

Všechny tyto možnosti dělají z notifikací velmi mocnou prodlouženou ruku mé aplikace, jelikož si uživatel může zjistit podrobné informace (pomocí roztažení notifikace) nebo provést základní úkony (pomocí tlačítek v notifikaci) aniž by musel do aplikace vůbec vstoupit.

Notifikace vytvářím s využitím objektu třídy `NotificationCompat.Builder`. Pomocí metod tohoto objektu lze mimo jiné nastavit titulek a obsah notifikace, její ikonu, barvu i akci (opět pomocí `PendingIntentu`), která se provede po stisknutí notifikace. Stejně tak lze přidat zmiňovaná interaktivní tlačítka a obsáhlejší text dostupný po roztažení notifikace.



Obrázek 4.4: Volba ohodnocení později, interaktivní notifikace a skrytí citlivého obsahu

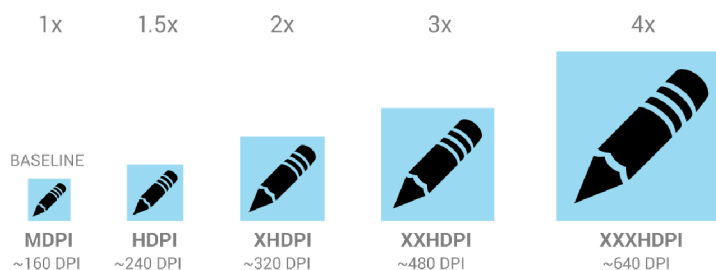
4.5 Zajištění podpory co nejvíce zařízení

V současnosti je Android bezkonkurenčně nejrozšířenějším mobilním operačním systémem. Zapříčinila se o to především otevřenost tohoto systému, kdy prakticky kterýkoli výrobce může zkusit své štěstí v oblasti mobilní elektroniky. Z pohled vývojáře ovšem není ani tak zajímavá příčina tohoto úspěchu, jako důsledek. Tím je extrémní roztržitost systému mezi parametrově odlišná zařízení, se kterou se musí vývojář při implementaci potýkat, a snažit se zajistit bezproblémový běh své aplikace na jakémkoli hardwaru. Android je všude a v různých podobách dle lokálních potřeb. Vývojáře pak nejvíce ovlivňují tři parametry každého zařízení. Těmi jsou **úhlopříčka displeje, rozlišení displeje a verze systému**.

Zatímco u přizpůsobení aplikace různým úhlopříčkám a rozlišením není velký prostor pro kompromisy a je potřeba pokrýt všechny varianty, nad podporovanými verzemi systému lze uvažovat.

Odlišné úhlopříčky a rozlišení ovlivňují reálnou velikost grafiky/obrázků na konkrétním zařízení. Při vysokém rozlišení (a tedy drobnější grafice) může utrpět nejen čitelnost, ale i ovladatelnost (existuje jistá minimální velikost grafických elementů pro pohodlné ovládání zařízení). V novějších verzích systému lze s grafikou pracovat ve vektorové podobě a docílit tak automatického škálování, většinou se však používá rastrový formát **PNG**. S tím pracuji i já. Zvětšení obrázku v tomto formátu má samozřejmě za následek snížení kvality, a tak pro každou takovou grafickou surovinu musím dodat více jejích verzí pro různé (Googlem definované) kategorie hustot rozlišení (uváděné v **DPI** či **PPI**). Tyto kategorie s vývojem hardwaru postupně přibývají, v současnosti jich pak existuje 6. LDPI, MDPI (~160 DPI), HDPI (~240 DPI), XHDPI (~320 DPI), XXHDPI (~480 DPI), XXXHDPI (~640 DPI). V praxi je potřeba zaměřit se na grafiku pro MDPI a výše, jelikož LDPI variantu si systému umí vytvořit sám z variant pro vyšší rozlišení. MDPI zároveň tvoří jakousi startovní pozici a ostatní k ní mají relativní vztah.

V různých specifikacích pak lze nalézt rozměry jednotlivých komponent rozhraní v tzv. **dp** či **dip** (*density independent unit*), tedy jakýchsi na hustotě nezávislých jednotkách, které se následně přepočítávají na konkrétní počet pixelů pro jednotlivé kategorie (MDPI, HDPI, ...) zobrazovačů, a tím je zajištěna podobná reálná velikost zobrazované grafiky. Pokud bylo například zadáno, že velikost ikony má být 48 dp, pak jsem musel připravit verzi této ikony pro MDPI zařízení (zde 48 dp = 48 px), HDPI zařízení (zde 48 dp = 72 px), XHDPI (48 dp = 96 px), atd. Jednotlivé verze ikony se pak umístí do příslušných adresářů projektu a systém následně již sám určí, kdy má být která verze použita. Podrobnější specifikaci včetně popisu následujících technik si lze přečíst na stránce *Devices and Displays* [13].



Obrázek 4.5: Velikosti grafiky pro jednotlivé hustoty rozlišení [13]

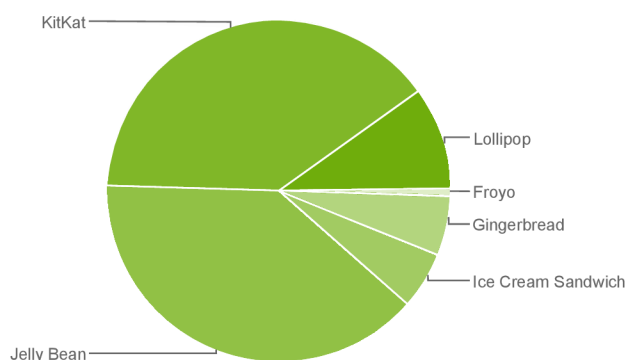
Dalším přístupem, který souběžně užívám, je využití automatického škálování jednotlivých elementů rozhraní. Zde je potřeba být si vědom toho, které elementy lze bez problému

zvětšit/zmenšit, a u kterých je velikost více méně pevně daná. Například pokud mám u položky jídla v restauraci vlevo fotografii a vpravo hvězdy zobrazující hodnocení, pak si musím být vědom, že zatímco velikost obrázku lze libovolně měnit, o hvězdách to říci nelze. Dobrým postupem pak může být poskytnout hvězdám přesně tolik prostoru, o kolik si sami požádají, a obrázek nechat zaplnit zbývajícím prostorem.

Pro opravdu velké úhlopříčky displejů (6" a výše) pak lze vytvořit úplně jiné rozložení rozhraní, případně využít modularitu fragmentů (zmiňované v podkapitole 2.5.2) pro zobrazení více „obrazovek“ na jedné obrazovce, a zajistit tak optimální využití prostoru.

Jak jsem zmiňoval výše, nad **minimální podporovanou verzí systému** je vhodné se nejdříve trochu zamyslet. Čím níže bych s minimální verzí cílil, o to náročnější by implementace mohla být, ať už kvůli absenci určité funkcionality nebo prostých metod pro zjednodušení běžné práce. Proti této absenci poměrně zdatně bojuje **Android Support Library**. Rozsáhlá knihovna, která umožňuje využívat funkcionality z velkého množství nově vznikajících API i na starších verzích systému. Tato knihovna bývá nedílnou součástí každého projektu, který necílí pouze na nejnovější verzi systému Android, a i já ji v aplikaci intenzivně využívám.

Pokud v cestě vývoji stojí určitá klíčová vlastnost systému, která přišla až s vyšší verzí a nebyla zpětně zpřístupněna, je jednoduše nutno číslo minimální podporované verze zvýšit. Někdy je však volba verze pouze osobním kompromisem mezi prací nutnou pro zajištění zpětné kompatibility a velikostí uživatelské základny, které bude aplikace zpřístupněna. Tak je tomu i v mém případě. V počátcích práce jsem se snažil cílit co nejnižší, na nejnižší ještě používanou verzi, přičemž jsem vycházel z oficiálních dat Googlu [14], viz obrázek 4.6. Tou je ve statistikách dlouhodobě uváděná verze 2.2 Froyo, její méně než půlprocentní podíl je však tak zanedbatelný, že o ní nemá smysl uvažovat. Na řadu tak přišla verze 2.3 Gingerbread, která byla dříve extrémně populární, a proto ji i dnes (po více než 4 letech) užívá okolo 6 % uživatelů. Spousta metod a funkcí, které se při implementaci používají na denním pořádku, však tuhle verzi ještě těsně minula, a tak jsem se ve výsledku ustálil na verzi 3.0 Honeycomb (API level 11). Tato verze však byla určena pouze pro tablety a dnes již téměř kompletně vymizela, a tak lze za **nejnižší podporovanou verzi považovat 4.0 Ice Cream Sandwich**, která je již přes 3 roky stará, což stále považuji za dostatečnou zpětnou kompatibilitu. Slovy čísel by má aplikace měla být **spustitelná na 94 % zařízení se systémem Android**, která jsou aktuálně v oběhu, přičemž tato hodnota se již může jedinečně zvyšovat.



Obrázek 4.6: Rozšířenost verzí systému Android k 4.5.2015 [14]

4.6 Návrh testování uživatelského rozhraní

Pro podtržení orientace celé práce na uživatelské rozhraní jsem naplánoval a vytvořil test, ve kterém budou testující uživatelé podle předem připraveného scénáře realizovat různé úkony v aplikaci a podávat zpětnou vazbu, ať už spontánní nebo na základě předpřipravených otázek. Pro vytvoření testů a celkově naplánování testování jsem využil poznatků načerpaných z knihy *Designing the User Interface* [11], která se testování poměrně intenzivně věnuje, ať už se jedná o testování použitelnosti, průzkumy nebo testy splnění stanovených požadavků. Především mě zaujala technika *think aloud*, tedy jakési hlasité přemýšlení, kdy uživatel během práce s aplikací nahlas komentuje, co, jak a proč právě dělá. Tuto techniku se pokusím do svého testování zakomponovat. Jejím nedostatkem může být zpomalení úkonů uživatele kvůli nutnosti současně celý postup komentovat, což lze vyřešit zpětným okomentováním až po splnění úkolu. Z mého pohledu ovšem rozhraní aplikace není natolik složité, aby takový problém opravdu nastal.

Přestože se pro otestování rozhraní jako takového dají použít obyčejné *mock-upy* na papíře nebo pouze naimplementované *prototypy*, testování provedu **přímo na výsledné aplikaci**, jelikož se s ohledem na plánované budoucí uvolnění aplikace na *GooglePlay Store* v testování zaměřím i na některé aspekty z oblasti funkcionality, případně připojím dotazy, které se nevztahují čistě na rozhraní. Navíc se jedná o první větší testování aplikace a rozhodně si nedovolím predikovat, které všechny podněty, a z jakých oblastí, mi účastníci během testování sdělí. Proto bude nejrozzumnější dát jim do rukou reálný produkt, který by jednou opravdu mohli používat na svém zařízení.

Testovací scénář se skládá z **jednotlivých úkolů**, přičemž každý začíná instruktorovým (mým) povelům k provedení určitého úkonu a obsahuje několik **doplňujících otázek** po provedení daného úkonu, často navíc i prostor pro **spontánní reakce** uživatele. Zároveň jsem si u jednotlivých úkolů předem ujasnil, které atributy (např. čas provedení úkonu) budu sledovat a dělat si o nich **poznámky**. Některé dotazy nabízejí uživateli dvě slovní alternativy pro vyjádření názoru a požadují po něm, aby se s využitím číselné hodnoty přiklonil k jedné z možností.

Součástí testování je pak samozřejmě i uživatelské zhodnocení jednotlivých způsobů vytváření záznamů o pokrmech z podkapitoly 3.1.7.

Cílem testování je určit celkovou **přívětivost** celého rozhraní, především jestli je dostatečně smysluplné, nepředkládá informace, které ve skutečnosti nejsou užitečné, a jestli se v něm uživatel dokáže dostatečně **rychle zorientovat** a provést úkony, které jsou podstatou aplikace. Neméně důležitým výsledkem by mělo být **identifikování ideálního způsobu přidávání záznamu o pokrmu**, ať už jednoho z navržených, či nového, vycházejícího z navržených možností, doplněného o uživatelské návrhy ke zlepšení.

Vedlejším produktem pak může být uživatelský pohled na **užitečnost celé aplikace**. Zda-li by takovou aplikaci zvolili pro každodenní používání a jestli jim opravdu dokáže pomoci s výběrem pokrmu.

Pro testování je stejně tak důležité stanovit **podmínky** a předpoklady, které zajistí získání hledaných výsledků. Před testem účastníkovi především vysvětlím podstatu a účel aplikace, zejména proto, že v reálném světě by si před stažením aplikace nejdříve alespoň přečetl její popis. Na druhou stranu s aplikací a rozhraním žádné předchozí zkušenosti mít nebude. Mým cílem je zaujmout aplikací hned napoprvé. Pro tento účel je potřeba dosáhnout okamžitého pochopení aplikace, což si takto mohu snadněji ověřit. Testovat budu na jediném zařízení pro dosažení shodných podmínek při všech testech. Bude se jednat o zařízení průměrných parametrů, ať už z pohledu úhlopříčky displeje nebo vůbec způsobu

ovládání. Cílem testování totiž není vyhodnotit ovládání aplikace v extrémních případech či na specifických zařízeních, proto tento faktor eliminuji.

Počet **účastníků** testování jsem stanovil na rozsah **5 až 10 osob**. Drtivá většina z nich bude ve věkovém rozsahu **15–30 let**, jelikož je to skupina, na kterou cílím nejvíce. Testu se zúčastní minimálně **2 ženy**. Ač je tento předpoklad možná mylný, očekávám od žen větší nároky především v oblasti navigace z důvodu nižší schopnosti orientace. Naopak jejich estetické cítění může být na lepší úrovni, což může pomoci v oblasti designu rozhraní. Všichni účastníci budou **standardními (denními) uživateli** smartphonu a (až na jedinou výjimku) budou **znalí platformy Android**. Tato volba mi umožní odhalit, na kolik správně jsem postupoval podle designových principů, návrhových vzorů a celkově zvyklostí u aplikací na platformě Android. Výše zmíněnou výjimku bude představovat **1 uživatel konkurenční platformy**, jehož názor by mohl přinést nový pohled na věc. Navíc se díky němu může dále utvrdit názor na intuitivnost aplikace v závislosti na tom, jak si s aplikací poradí.

Pro detailnější představu o testování lze nahlédnout do **scénáře testování**, který je umístěn v příloze **D**.

4.7 Interpretace výsledků testování uživatelského rozhraní

Namísto prezentování konkrétních odpovědí jednotlivých uživatelů na jednotlivé otázky se v této podkapitole pokusím shrnout nejčastější a nejvíce hodnotné poznatky z testování, úspěšnost realizace plánovaných vlastností rozhraní a důsledek testování na změny v rozhraní.

Z uživatelských reakcí po spuštění aplikace usuzuji, že **realizace úvodních obrazovek a celkově prvního dojmu byla úspěšná**. Všichni uživatelé bez výjimky pochopili, co ztělesňují obrazovky se seznamem jídel a výběrem jídel z okolí. Drobné výhrady se objevily pouze u druhé jmenované, kde bude potřeba lépe zdůraznit, že se jednotlivé pokrmy nacházejí v odlišných podnicích. Také doplním vzdálenost k podniku s daným pokrmem. Přiklonění účastníků k jedné z těchto obrazovek pro denní využití však jednoznačné nebylo a názory se zde rozcházely, proto **nepochybuji o užitečnosti (a zachování) obou těchto hlavních obrazovek**.

Při svém studiu i návrhu jsem se poměrně intenzivně věnoval **uplatnění designových principů i návrhových vzorů platformy Android** tak, aby má aplikace do tohoto ekosystému bez problému zapadala a uživatelé ji stejným způsobem i přijali. Testování v tomto ohledu prokázalo, že **se snaha vyplatila**, jelikož všechny ovládací prvky, mezi které patří třeba mechanismus pro obnovení dat nebo výsuvný navigační panel, byly vždy **nalezeny s časem pod 5 sekund** a účastníci vždy zmiňovali, že s hledáním neměli nejmenší problém, jelikož tyto koncepty znají ze svých dalších aplikací.

Velká část testování se věnovala jednotlivým způsobům vytváření záznamu o pokrmu. Na **pomalém způsobu** vytváření jídla účastníci oceňovali především **jednoduchou orientaci**, a že vždy věděli, co se od nich čeká. Tedy přesně podle plánu. Zalíbení v tomto způsobu pak našly především ženy. Mnozí by však po vyplnění údajů ocenili automatické přenesení na další krok, aby se tímto úkonem nemuseli zdržovat.

Trochu k mému překvapení byl **obecně nejlépe hodnocen rychlý způsob**. Ten se soustředil na kompaktní rozměry a setrvávání na jediné obrazovce. Přestože jsem měl obavu, že uživatelům bude méně prostoru činit potíže, nikdo z účastníků si nestěžoval. Naopak **ocenili zpracování s využitím animací** nebo zvýraznění kroků hodnocení, které již splnili (příslušná tlačítka zezelenala). Překvapivě pak způsobu vytýkali nemožnost vyplnění

dodatečných informací k hodnocení, což je v zásadě pozitivní zpráva ukazující, že **uživatelé jsou ochotní sdílet i větší množství informací**.

Naopak největší kritiku si odnesl *způsob moderní*, který spoléhal na gesta pro záměnu komponent rozhraní, a účastníci se v něm obecně **špatně orientovali**. Často však oceňovali, že vyplněný název pokrmu měli následně celou dobu na očích, což indikuje potřebu stále zobrazovat již vyplněné údaje.

Zatímco některé způsoby byly založeny na mém vlastním uživatelském rozhraní fotoaparátu, jiné využívaly vestavěnou aplikaci fotoaparátu. Výhodou vestavěné aplikace je velké množství funkcí a nastavení, jejichž napodobení by stálo velké úsilí. Její využití je v mé aplikaci ovšem komplikováno potřebou pořizovat čtvercové fotografie, což vestavěná aplikace sama o sobě neumožňuje. Situace, kdy takto účastník pořídil obdélníkovou fotku, která se mu posléze zobrazila ořezána na čtvercovou, byla často oprávněně kritizována. Proto se budu v aplikaci opravdu držet vlastního rozhraní fotoaparátu zobrazujícího přímo čtvercový náhled. Bude však nutné intenzivně zapracovat na funkcionalitě a lepší zpětné vazbě, která zde účastníkům dramaticky chyběla.

Jak jsem již zmínil, **nejlepší hodnocení si nakonec odnesl rychlý způsob**. Ideální řešení by pak podle účastníků bylo založeno právě na tomto způsobu, bylo co nejrychlejší, poskytovalo by čtvercový náhled fotoaparátu a umožňovalo vyplnit dodatečné informace. Vyplněné informace by pak měly zůstat stále na očích.

Na dotaz na ideální účet pro přihlašování většina účastníků uvedla buď přímo oddělený účet aplikace, případně použití účtu služby Facebook. Výsledek tedy odpovídal mému očekávání a aplikace na něj byla předem připravena.

S výsledkem testování jsem celkově spokojen. Nejdůležitější pro mě bylo, aby uživatelé z úvodních obrazovek bez problému pochopili účel aplikace a taky abych dokázal určit vhodný způsob pro vytváření záznamů o jídle. Obojí se povedlo. Podobně důležité je pak pro mě do budoucna i velké množství drobných poznámek k rozhraní, které sice nestojí za zmínění v této práci, mně osobně ovšem výrazně napovědělo, co je potřeba udělat pro zdokonalení celého zážitku.

Kapitola 5

Závěr

V práci se podařilo vybudovat jádro aplikace, jejíž potenciál dle mého názoru sahá za hranice běžných studentských projektů, které málokdy spatří světlo světa. Byl položen základ pro reálně uplatnitelný produkt, jež se po konečném odladění může distribuovat například na největším současném obchodu s aplikacemi *GooglePlay Store*. Již v tento okamžik je aplikace kompatibilní s 94 % zařízení se systémem Android na trhu. Aplikace zároveň není osamoceným kusem softwaru, ale je součástí celé služby skládající se i z aplikačního backendu v podobě databáze a API. Nejen návrh a zprovoznění komunikace s tímto backendem, ale také týmová spolupráce, byly součástí této práce. Tato spolupráce spočívala například v definování formátu dat přenášených po síti nebo průběžném diskutování směru, kterým se má celá služba ubírat a který by má i kolegova aplikace měly odrážet.

Z mého pohledu jsou možnosti dalšího vývoje poměrně široké. Nejvýznamnější posun tak vidím ve výše zmíněném odladění a uvolnění aplikace mezi běžné uživatele. Modulární přístup, kterým jsem aplikaci implementoval, navíc umožňuje pohodlně navrhnout alternativní rozhraní pro zařízení s větší úhlopříčkou (tablety), a dosáhnout tak na další uživatele. Z dlouhodobějšího pohledu by se pro úspěšné zavedení aplikace měl zprovoznit i gamifikační systém, pro který jsem ve své práci také položil základy.

Práce byla realizována v plném souladu se zadáním. V oblasti studia jsem se věnoval nejen obecným trendům uživatelských rozhraní, ale také návrhovým vzorům a designovým principům, které se uplatňují u aplikací běžících na systému Android. Rozebral jsem technické aspekty platformy Android i související základy programování. Uživatelské rozhraní bylo detailně navrženo a naimplementováno s využitím rozličných knihoven a byla realizována i komunikace s backendem. Znalosti načerpané při studiu testování uživatelské zkušenosti jsem využil v závěru práce, kde jsem se důkladně věnoval testování výsledné aplikace, přičemž jsem také diskutoval získané výsledky.

V důsledku všech těchto skutečností jsem přesvědčen, že se vytvořená aplikace může stát každodenním pomocníkem a rádcem běžných lidí se zájmem v oblasti stravování.

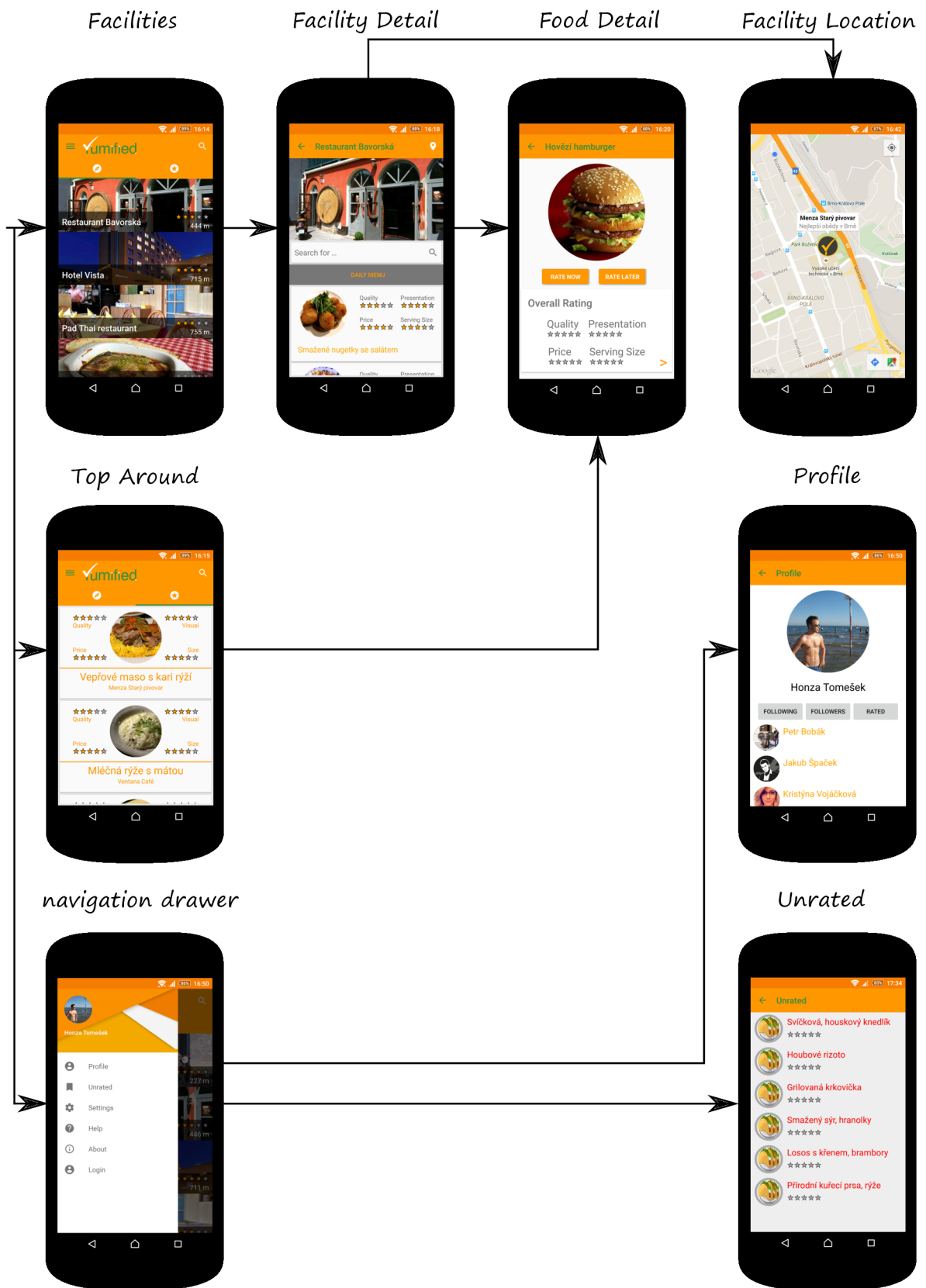
Literatura

- [1] *Android Design Principles* [online]. 2014 [cit. 2015-02-10]. Dostupné na: <<http://developer.android.com/design/get-started/principles.html>>.
- [2] BANK, C. a ZUBERI, W. *Mobile UI Design Patterns: A Deeper Look At The Hottest Apps Today* [online]. 2014 [cit. 2015-02-05]. Dostupné na: <https://s3.amazonaws.com/uxpin/uxpin_mobile_ui_design_patterns_2014.pdf>.
- [3] *Patterns* [online]. 2014 [cit. 2015-02-15]. Dostupné na: <<http://developer.android.com/design/patterns/index.html>>.
- [4] HINMAN, R. A New Mobile UX Design Material. *Smashing Magazine* [online]. 2012 [cit. 2015-01-20]. Dostupné na: <<http://www.smashingmagazine.com/2012/10/30/motion-and-animation-a-new-mobile-ux-design-material/>>.
- [5] BANK, C. Gestures & Animations: The Pillars of Mobile Design. *UX Magazine* [online]. 2014, č. 1351 [cit. 2015-01-25]. Dostupné na: <<http://uxmag.com/articles/gestures-animations-the-pillars-of-mobile-design>>.
- [6] *Material design* [online]. 2014 [cit. 2015-02-20]. Dostupné na: <<http://www.google.com/design/spec/material-design/introduction.html>>.
- [7] MEDNIEKS, Z., DORNIN, L., MEIKE, G. B. et al. *Programming Android*. Sebastopol: O'Reilly, 2011. 482 s. ISBN 978-1-449-38969-7.
- [8] ALLEN, G. *Android 4*. 1. vyd. Brno: Computer Press, 2013. 656 s. ISBN 978-80-251-3782-6.
- [9] MURPHY, M. L. *Beginning Android 2*. New York: Apress, 2010. 397 s. ISBN 978-1-4302-2629-1.
- [10] *Android Developers* [online]. 2014 [cit. 2015-02-25]. Dostupné na: <<http://developer.android.com/index.html>>.
- [11] SHNEIDERMAN, B. a PLAISANT, C. *Designing the user interface: strategies for effective human-computer interaction*. 5. vyd. Boston: Addison-Wesley, 2010. 606 s. ISBN 978-0-321-53735-5.
- [12] USTWO. *Pixel Perfect Precision* [online]. 3. vyd. 2014 [cit. 2015-02-10]. Dostupné na: <<http://cdn.ustwo.com/PPP/PP3.pdf>>.
- [13] *Devices and Displays* [online]. 2015 [cit. 2015-03-01]. Dostupné na: <<http://developer.android.com/design/style/devices-displays.html>>.

- [14] *Dashboards* [online]. 2015 [cit. 2015-05-08]. Dostupné na:
<<http://developer.android.com/about/dashboards/index.html>>.

Příloha A

Struktura základních obrazovek výsledné aplikace



Příloha B

Výňatek z odpovědi na požadavek na Foursquare API

```
{
  ...
  response: {
    venues: [
      {
        id: "4f60816e4fc6f460d32e48fa", // první záznam o podniku
        name: "Knihovna FIT VUT",
        contact: {
          phone: "541141204",
          formattedPhone: "541 141 204"
        },
        location: {
          address: "Božetěchova 2",
          lat: 49.22687475718783,
          lng: 16.59685492515564,
          ...
        },
        categories: [
          ...
        ],
        ...
      },
      {
        id: "4b6fc471f964a520fdfb2ce3", // druhý záznam o podniku
        name: "FIT VUT v Brně",
        ...
      },
      ... // další záznamy o podnicích
    ],
  }
}
```

Příloha C

Výňatek z odpovědi na požadavek na vlastní API

```
{
  count: 10,
  facilities: [
    {
      id: "4c778e61923ba143195a67e6",
      category: "Diner",
      category_id: "4bf58dd8d48988d147941735",
      name: "Restaurace Racek",
      distance: 623,
      fsrating: 3.5,
      photo_prefix: "https://irs0.4sqi.net/img/general/",
      photo_suffix: "/HiJuvyrcTMHjNYD-bQjoQUOsZc6alnJQPdIQ6S1Gafc.jpg"
    },
    {
      id: "4bdd69df645e0f477a6f6b19",
      category: "African Restaurant",
      category_id: "4bf58dd8d48988d1c8941735",
      name: "Klub cestovatelů",
      distance: 651,
      fsrating: 3.5,
      photo_prefix: "https://irs2.4sqi.net/img/general/",
      photo_suffix: "/NBamI9gcL1MixlXfp0tLaDAoOicp0m3gjhrINqz4-i8.jpg"
    },
    ...
  ],
  next_page: "http://apitest.g6.cz/web/facilities?lat=49.22634225027&lng=16.59581422806&page_limit=10&page=3&time_zone=0"
}
```


Příloha D

Scénář testování uživatelského rozhraní

1. Spusťte aplikaci a prozkoumejte úvodní obrazovky.

Otázka: Jaký je význam úvodních obrazovek?

1. obrazovka:

.....

2. obrazovka:

.....

Vlastní poznámka: uživatel našel navigační panel
ano / ne

Otázka: Považujete obrazovku s nejlepšími jídly z okolních restaurací za užitečnou
(použil byste ji spíše než seznam restaurací)?

.....

2. Vyhledejte podnik podle jména.

Otázka: Je podle vás vyhledávání dostatečně při ruce? Zdůvodněte.

.....

3. Obnovte načtená data.

Vlastní poznámka: doba nalezení aktualizacího mechanismu:
..... sekund

4. Vyberte si pokrm z okolních restaurací (projděte nejméně 3 restaurace).

Otázka: Jak byste seřadil kategorie hodnocení od nejužitečnější k méně
užitečným.

1. kategorie (nejužitečnější)

.....

2. kategorie

.....

3. kategorie

.....

4. kategorie

.....

Otázka: Která další kategorie by pro vás byla užitečná?

.....

Otázka: Označil byste některou kategorii hodnocení za zbytečnou? Kterou?

.....

Otázka: Jaké další informace by vám usnadnily výběr?

.....

Vlastní poznámka: uživatel našel možnost procházení fotografií podniku.
ano / ne

5. Zobrazte podnik na mapě.

Vlastní poznámka: doba do spuštění mapy:
..... sekund

6. Aktivujte navigaci k podniku.

Vlastní poznámka: doba do spuštění navigace:
..... sekund

7. Přidejte nový pokrm "pomalým" způsobem.

Vlastní poznámka: doba nalezení tlačítka pro přidání pokrmu:
..... sekund

Otázka: Je pro vás umístění tlačítka smysluplné?

Otázka: Jak byste ohodnotil pohodlnost vyplňování informací?
1 (negativní) 2 3 4 5 (pozitivní)

Otázka: Uveďte, co se vám na tomto způsobu líbilo.

Otázka: Uveďte, co se vám na tomto způsobu nelíbilo.

Otázka: Jak byste celkově ohodnotil tento způsob?
1 (negativní) 2 3 4 5 (pozitivní)

8. Přidejte nový pokrm "rychlým" způsobem.

Otázka: Jak byste ohodnotil pohodlnost vyplňování informací?
1 (negativní) 2 3 4 5 (pozitivní)

Otázka: Uveďte, co se vám na tomto způsobu líbilo.

Otázka: Uveďte, co se vám na tomto způsobu nelíbilo.

Otázka: Jak byste celkově ohodnotil tento způsob?
1 (negativní) 2 3 4 5 (pozitivní)

9. Přidejte nový pokrm "moderním" způsobem.

Otázka: Jak byste ohodnotil pohodlnost vyplňování informací?
1 (negativní) 2 3 4 5 (pozitivní)

Otázka: Uveďte, co se vám na tomto způsobu líbilo.

Otázka: Uveďte, co se vám na tomto způsobu nelíbilo.

Otázka: Jak byste celkově ohodnotil tento způsob?
1 (negativní) 2 3 4 5 (pozitivní)

10. Seřadíte všechny 3 způsoby podle vašich preferencí.

1. pozice (nejužitečnější)

2. pozice

3. pozice

Otázka: Jak by podle vás vypadal ideální způsob?

11. Prohlédněte si obrazovku s detailem pokrmu.

Vlastní poznámka: uživatel objevil uživatelské komentáře
ano / ne

Otázka: Rozumíte účelu tlačítek "Rate Now" a "Rate Later"?

Tlačítko "Rate Now":

Tlačítko "Rate Later":

12. Přihlaste se do aplikace pomocí Facebook účtu.

Otázka: Kdo je nyní přihlášen do aplikace?

Otázka: Jaký z vašich účtů byste nejdříve použil pro přihlášení (případně účet nový)?

Příloha E

Použité knihovny

- ANDROID VOLLEY
 - *Apache License, Version 2.0*
 - síťová komunikace
 - <https://github.com/mcxiaoke/android-volley>
- MATERIAL DESIGN ANDROID LIBRARY
 - *Apache License, Version 2.0*
 - Material Design tlačítka
 - <https://github.com/navasmdc/MaterialDesignLibrary>
- ANDROID VIEWPAGERINDICATOR
 - *Apache License, Version 2.0*
 - indikátor aktuální obrazovky
 - <https://github.com/JakeWharton/ViewPagerIndicator>
- FACEBOOK SDK FOR ANDROID
 - přihlašování pomocí Facebook účtu
 - <https://developers.facebook.com/docs/android>
- CRASHLYTICS KIT
 - reportování chyb a analýza používání aplikace
 - <https://get.fabric.io/crashlytics>