

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

**Mobilní aplikace pro uživatelské hodnocení míst pro
OS Android**

Bc. Pavel MARTINEC



Česká zemědělská univerzita v Praze
Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Autor práce: Bc. Pavel Martinec
Studijní program: Systémové inženýrství a informatika
Obor: Informatika
Vedoucí práce: Ing. Dana Vynikarová, Ph.D.
Garantující pracoviště: Katedra informačního inženýrství
Jazyk práce: Čeština

Název práce: **Mobilní aplikace pro uživatelské hodnocení míst pro OS Android**

Název anglicky: **Android OS mobile application for user rating of places**

Cíle práce: Cílem teoretické části diplomové práce je získání podkladů pro implementaci mobilní aplikace. Tato teoretická část se bude skládat ze dvou dílčích cílů. Prvním dílčím cílem je analýza funkcí existujících aplikací zpracovávajících mapy nebo hodnocení, na jejichž základě bude definován seznam funkcí, případně dílčích řešení budoucí aplikace. Následně druhým dílčím cílem bude analýza nástrojů pro tvorbu mobilních aplikací, jejímž výsledkem bude zpracování porovnání těchto nástrojů a výsledný výběr nástroje pro tvorbu budoucí aplikace.

Cílem praktické části je samotné vytvoření aplikace, prvním dílčím cílem je vytvoření vhodného prototypu aplikace. Ten se bude skládat z popisu logického fungování aplikace a dále z prototypu UI aplikace. Druhým dílčím cílem je naprogramování aplikace s využitím analýz z teoretické části a prototypem aplikace, součástí je také naprogramování vhodné serverové části. Serverová část aplikace bude zodpovědná za sběr dat od uživatelů a následně poskytování těchto dat ostatním uživatelům aplikace.

Metodika: V teoretické části nejdříve proběhne studium existujících řešení a shromáždění informací o funkcích a jejich řešení, rozdílech a nedostatcích aplikací a projektů, zabývajících se zobrazováním mapových podkladů nebo hodnocením míst (např. společností, restauračních zařízení apod.). Tyto informace budou následně použity k sestavení seznamu funkcionalit budoucí aplikace.

Následně proběhne analýza nástrojů pro vývoj mobilní aplikace, na výsledném hodnocení a výběru nástroje pro vývoj aplikace se bude podílet

zejména: možnost kompilace aplikace na více typů mobilních OS, jednoduchost použití a efektivita těchto nástrojů při vývoji, výkon kódu na zařízení, technické možnosti nástrojů (potřebné knihovny a funkce pro vývoj řešení aplikace) a cena těchto nástrojů.

V praktické části nejdříve vznikne popis fungování aplikace, slovně a/nebo vhodnou grafickou formou. Následně vznikne prototyp UI složený z drátěného modelu obrazovek aplikace a jejich jednoduchá vizualizace. Následně bude s použitím výsledků analýz z teoretické části a prototypu naprogramována samotná aplikace společně se serverovou částí, která bude s aplikací komunikovat a zajišťovat přenosy dat od a k uživatelům.

Doporučený rozsah práce: 60 - 80 stran

Klíčová slova: Mobilní aplikace, Android, mapa, hodnocení míst

Doporučené zdroje informací:

1. ANDERSON, Nathanael J. Getting Started with NativeScript. Packt Publishing Ltd., 2016. ISBN 978-1785888656
2. PETZOLD, Charles. Creating Mobile Apps with Xamarin.Forms. Microsoft Press, 2016. ISBN 978-1-5093-0297-0
3. SMYTH, Neil. Android Studio 2.3 Development Essentials - Android 7 Edition. CreateSpace Independent Publishing Platform, 2017. ISBN 978-1544275437

Předběžný termín obhajoby: 2017/18 LS - PEF

Elektronicky schváleno: 11. 1. 2018
Ing. Martin Pelikán, Ph.D.
Vedoucí katedry

Elektronicky schváleno: 11. 1. 2018
Ing. Martin Pelikán, Ph.D.
Děkan

Čestné prohlášení

Prohlašuji, že jsem svou diplomovou práci "Mobilní aplikace pro uživatelské hodnocení míst pro OS Android" vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30. března 2018

Poděkování

Rád bych touto cestou poděkoval Ing. Daně Vynikarové, Ph.D za vedení mé diplomové práce. Dále bych rád poděkoval své rodině a přátelům za podporu, kterou mi během psaní této práce poskytli.

Anotace

Abstrakt

Tato práce vznikla jako reakce na problematiku turismu. Základem jsou dvě myšlenky. Za první, turista je v cizím prostředí a i když se před cestou připraví (nastuduje si materiály o lokalitě, kterou navštívuje), nikdy nepokryje vše, co daná lokalita vyžaduje nebo nabízí. Druhá myšlenka pak navazuje na tu první, kdy místní obyvatelé mohou zneužít právě neznalosti a nepřipravenosti turistů a vznikají turistické pasti, kdy se turista stane obětí nejrůznějších podvodů.

Cílem této práce je tedy vytvořit koncept mobilní aplikace, která by v budoucnu turistům usnadnila orientaci v prostředí cizí země, a to jak nabídkou zajímavých (leckdy pro turisty nezdokumentovaných) míst, či jiných reálií dané oblasti, tak varováním před podvody a turistickými pastmi.

Tato práce řeší jak samotný návrh aplikace, tak technické zpracování první demonstrativní verze aplikace.

Klíčová slova

Turismus, Mobilní aplikace, Mapa, Hodnocení míst, Android

Annotation

Abstract

This work is dedicated to tourism and problems coming with it. It is based on two thoughts. First one is a potential unpreparedness of tourist visiting foreign country, where he/she cannot cover in his/her preparations every aspect of what the country requires or offers. The second thought follows the first one, where some local people takes advantage of the unpreparedness of tourists and creates various tourist traps to cheat the tourist.

The goal of this work is to create a concept of mobile application, which can make the tourists life easier in foreign environment. The application should provide information of local places and specialities, same as to provide warnings about the mentioned tourist traps.

This work covers the concept and technical solution of the application.

Key words

Tourism, Mobile application, Place rating, Map, Android

1 Obsah

Abstrakt.....	5
Klíčová slova.....	5
Abstract.....	6
Key words.....	6
1 Obsah.....	7
1.1 Seznam obrázků.....	8
1.2 Seznam tabulek.....	9
1.3 Seznam příloh.....	9
2 Úvod.....	10
3 Cíle práce a metodika.....	12
3.1 Cíle práce.....	12
3.2 Metodika.....	12
4 Teoretická východiska.....	14
4.1 Předpokládané součásti aplikace.....	14
4.1.1 Mobilní aplikace.....	14
4.1.2 Serverová aplikace.....	15
4.2 Rozbor existujících aplikací.....	16
4.2.1 Google Maps.....	16
4.2.2 Mapy.cz.....	19
4.2.3 Maps.Me.....	21
4.2.4 WC Kompas.....	24
4.2.5 Map Marker.....	26
4.2.6 Shrnutí.....	27
4.3 Výběr frameworku pro vývoj aplikace.....	28
4.3.1 Android studio.....	28
4.3.2 Flutter.....	28
4.3.3 Apache Cordova.....	29
4.3.4 Native Script.....	30
4.3.5 Embarcadero RAD studio - Delphi.....	30
4.3.6 Xamarin.....	31
4.3.7 Shrnutí.....	31
5 Vlastní řešení.....	32
5.1 Výběr funkcí.....	32
5.1.1 Mapy.....	32
5.1.2 Seznam míst v okolí.....	33
5.1.3 Přidání bodu.....	33
5.1.4 Hodnocení místa.....	33
5.1.5 Detailní zobrazení bodu.....	33
5.1.6 Náhled zobrazení bodu v mapě.....	33
5.1.7 Systém hlasování.....	34

5.1.8	Dlouhodobé a krátkodobé body	34
5.1.9	Fotografie	35
5.2	Vizualizace aplikace	35
5.2.1	Mapa	35
5.2.2	Rychlé zobrazení místa	36
5.2.3	Detailní zobrazení místa	37
5.2.4	Přiřazené recenze k záznamu	39
5.2.5	Fotografie	40
5.2.6	Přidání záznamu	42
5.3	Vývoj aplikace	45
5.3.1	Cena provozu aplikace	45
5.3.2	Serverová aplikace	46
5.3.3	Mobilní aplikace	61
6	Výsledky a diskuse	69
7	Závěr	70
8	Seznam užitých zdrojů	71
9	Přílohy	73

1.1 Seznam obrázků

Obrázek 1	- Ilustrační obrázky - Google Maps	19
Obrázek 2	- Ilustrační obrázky - Mapy.cz	21
Obrázek 3	- Ilustrační obrázky - Maps.me	23
Obrázek 4	- Ilustrační obrázky - WC Kompas	25
Obrázek 5	- Ilustrační obrázky - Map Marker	27
Obrázek 6	- Vizualizace - Mapa	36
Obrázek 7	- Vizualizace - Rychlé zobrazení místa	37
Obrázek 8	- Vizualizace - Detailní zobrazení bodu	38
Obrázek 9	- Vizualizace - Podřazené recenze bodu	39
Obrázek 10	- Vizualizace - Seznam fotografií	40
Obrázek 11	- Vizualizace - Detail fotografie	41
Obrázek 12	- Vizualizace - Přidání záznamu	42
Obrázek 13	- Vizualizace - seznam typů bodu	43
Obrázek 14	- Vizualizace - přidání bodu	44
Obrázek 15	- Proces přihlášení	58
Obrázek 16	- Proces ověřování pomocí klíčů	59

Obrázek 17 - Aplikace – mapa.....	75
Obrázek 18 - Aplikace - rychlé zobrazení bodu	76
Obrázek 19 - Aplikace - Detailní zobrazení bodu.....	77
Obrázek 20 - Aplikace - podřazené recenze - seznam	78
Obrázek 21 - Aplikace - zobrazení podřazené recenze.....	79
Obrázek 22 - Aplikace - hlasování.....	80
Obrázek 23 - Aplikace - přidání recenze	81
Obrázek 24 - Aplikace - obrázky recenze	82

1.2 Seznam tabulek

Tabulka 1 - Databáze - Tabulka "members"	48
Tabulka 2 - Databáze - Tabulka "devices"	49
Tabulka 3 - Databáze - Tabulka "writeAccess"	49
Tabulka 4 - Databáze - Předpis tabulky „POINTGRID_YxZ“	50
Tabulka 5 - Databáze - Předpis tabulky „POINTGRID_RECORDS_YxZ“	51
Tabulka 6 - Databáze - Předpis tabulky hlasů.....	53
Tabulka 7 - Databáze - Nevhodné ukládání hlasů	55

1.3 Seznam příloh

Příloha 1 - Příklad úpravy ListView - Android studio.....	73
Příloha 2 - Flutter – příklad programování objektů UI	74
Příloha 3 - Snímky obrazovky aplikace	75

2 Úvod

Turismus, jak ho známe dnes, je velmi rozšířený, a pravděpodobně každý ve svém životě již do ciziny cestoval, nebo někdy cestovat bude. A pravděpodobně každý na takovoto zahraniční cestě řeší spoustu věcí, které se týkají nejen poznání, ale také věci každodenní potřeby, jako je například jídlo, doprava a podobně. Dobrý dojem z takovoto cesty má člověk jen tehdy, pokud skutečně byla naplněna jeho očekávání, z pohledu této práce zejména v podání bezproblémovosti řešení výše zmíněných problémů.

Klíčovým aspektem je příprava. Pokud cestujeme do cizí země, je nutné si načíst materiály ohledně jejich zvyklostí, každodenního života i zajímavostech. A to je první klíčový problém. Lidé leckdy nemají chuť nebo čas se řádně na zahraniční cestu připravit. A pokud ano, příprava na takovouto cestu může být velmi náročná vzhledem k informačním zdrojům. Cestovní průvodci často ukazují jen to hezké nebo populární a dostat se k užitečným praktickým informacím ohledně každodenního života nemusí být lehké. Dalším aspektem, který může turistu odradit od delší přípravy, je vysoká fragmentace těchto materiálů, kdy kompletní obraz dostane až po poskládání dílčích částí z mnoha informačních zdrojů.

Dalším problémem se zdroji je jejich relativní zastaralost. Tištěné materiály, jako jsou knižní průvodci, nelze aktualizovat pravidelně. Internetové zdroje jsou pak aktuálnější, i přes to jsou ale aktualizovány v delších časových intervalech, protože není možné, aby autoři internetových portálů neustále monitorovali oblast, které se věnují a stále data aktualizovali.

Třetím problémem, se kterým se turista může setkat, je nečisté jednání od obyvatel jeho destinace. V tomto smyslu se jedná o podvody, zejména různé cesty, jak turistu (nepřipraveného, neznalého prostředí) připravit o co nejvíce peněz (ne nutně ve všech případech však nelegální cestou).

Na tomto základě vzniká má diplomová práce. Na internetovém portálu www.stream.cz běží pořad „Honest Guide“, který moderuje pan Rubeš, který ve svých videích upozorňuje nejen na turistické pasti (nepoctivé směnárny, taxi, obchody), zároveň ale také ukazuje zajímavá místa, která stojí za to navštívit, na která ale běžný turista nikdy nepříjde, protože se o takových místech obvykle v žádném průvodci nepíše.

Po osobním rozhovoru právě s panem Rubešem mám potvrzeno, že žádná internetová či mobilní aplikace, která by se věnovala právě podvodným místům nebo i jiným zajímavostem, neexistuje, a mohl by to být velmi hodnotný doplněk zejména pro turisty z celého světa. Zároveň v rámci této aplikace chci řešit aktuálnost dat. Výsledkem by tedy měla být aplikace, která turistovi na cestách prostřednictvím mobilního telefonu nabídne co nejaktuálnější data o zajímavých, či naopak potenciálně nebezpečných místech v okolí. To samozřejmě do jisté míry umožňují i některé jiné aplikace (Google Maps, Mapy.cz aj.), nicméně tyto aplikace jsou vesměs komplexní a není to jejich hlavní cíl, tudíž efektivita práce s nimi nemusí být největší (nehledě na výše zmíněnou aktuálnost dat, některé recenze jsou staré řadu měsíců a mohou snižovat hodnocení místa i přes to, že by se jednalo o problém dávno odstraněný, nebo naopak hodnocení neadekvátně zvyšovat). Tyto aplikace nicméně ale některá místa, jako například směnárny, nepokrývají vůbec.

Cílem teoretické části aplikace je definice předpokládaných vlastností nové aplikace, a sběr dat potřebných pro rozhodnutí, jak aplikace bude dále vyvíjena (tedy jak mohou být některé prvky řešeny, a pomocí jakých nástrojů bude aplikace realizována).

Cílem praktické části této práce je vyvinout koncept aplikace, která uživatelům poskytne aktuální data o zajímavostech či hrozbách v jejich okolí.

Práce zahrnuje analýzu některých existujících mapových aplikací (zejména z hlediska jejich fungování a rozsahu funkcí), která poslouží jako inspirace pro tvorbu mé aplikace. Dále práce zahrnuje výběr frameworku pro realizaci aplikace a poté i samotný vývoj konceptu aplikace, tedy definování předpokládaných funkcí, návrh některých obrazovek aplikace a nakonec popis některých konkrétních problémů a funkcionalit vyvíjené aplikace.

Motivací pro výběr tohoto tématu je pro mě samotný pořad Honest Guide, který sleduji již od jeho vzniku, ale také i osobní zkušenost ze zahraniční cesty. Díky velmi aktivní paní průvodkyni znalé dané oblasti jsme viděli i to, co většina ostatních na stejném zájezdu s jinými průvodci neviděla (jednalo se například o různé vyhlídky, o kterých ví jen místní nebo právě lidé, kteří oblast navštěvují častěji). S touto novou aplikací by bylo možné i tato místa doporučit uživatelům bez průvodce a obohatit je o nové zážitky z jejich cest.

3 Cíle práce a metodika

3.1 Cíle práce

Cílem teoretické části diplomové práce je získání podkladů pro implementaci mobilní aplikace. Tato teoretická část se bude skládat ze dvou dílčích cílů. Prvním dílčím cílem je analýza funkcí existujících aplikací zpracovávajících mapy nebo hodnocení, na jejichž základě bude definován seznam funkcí, případně dílčích řešení budoucí aplikace. Druhým dílčím cílem bude následně analýza nástrojů pro tvorbu mobilních aplikací, jejímž výsledkem bude výsledný výběr nástroje pro tvorbu budoucí aplikace. Cílem praktické části je samotné vytvoření konceptu aplikace, prvním dílčím cílem je vytvoření vhodného prototypu aplikace. Ten se bude skládat z popisu logického fungování aplikace a dále z prototypu UI aplikace. Druhým dílčím cílem je naprogramování první verze aplikace s využitím analýz z teoretické části a prototypem aplikace, součástí je také naprogramování vhodné serverové části. Serverová část aplikace bude zodpovědná za sběr dat od uživatelů a následně poskytování těchto dat ostatním uživatelům aplikace.

3.2 Metodika

V teoretické části nejdříve proběhne studium existujících řešení a shromáždění informací o funkcích a jejich řešení, rozdílech a nedostacích aplikací a projektů, zabývajících se zobrazováním mapových podkladů nebo hodnocením míst (např. firem, restauračních zařízení apod.). Tyto informace budou následně použity k sestavení seznamu funkcionalit budoucí aplikace.

Následně proběhne analýza nástrojů pro vývoj mobilní aplikace, na výsledném hodnocení a výběru nástroje pro vývoj aplikace se bude podílet zejména: možnost kompilace aplikace na více typů mobilních OS, jednoduchost použití a efektivita těchto nástrojů při vývoji, výkon kódu na zařízení, technické možnosti nástrojů (potřebné knihovny a funkce pro vývoj řešení aplikace) a cena těchto nástrojů.

V praktické části nejdříve vznikne popis fungování aplikace, slovně a/nebo vhodnou grafickou formou. Následně vznikne prototyp UI složený z drátěného modelu obrazovek aplikace nebo jejich jednoduchá vizualizace.

Poté bude s použitím výsledků analýz z teoretické části a prototypu naprogramována samotná aplikace společně se serverovou částí, která bude s aplikací komunikovat a zajišťovat přenosy dat od a k uživatelům.

4 Teoretická východiska

4.1 Předpokládané součásti aplikace

Nejprve je třeba vytvořit úvodní představu, co všechno bude aplikace poskytovat uživateli, respektive definovat souhrn předpokládaných funkcí.

4.1.1 Mobilní aplikace

Jádro aplikace musí tvořit tři základní stavební kameny.

Prvním základním stavebním kamenem je mapa. Uživatel se pohybuje v cizím prostředí, a aktuálně řeší nějaký problém, například, chce se najíst a hledá podnik, který vaří dobrou a levnou lokální kuchyni. Aplikace tedy zprostředkuje zobrazení relevantních míst v podobě nadefinovaných bodů v mapě („markers“ – jedná se o přidání bodů do mapy, které nejsou součástí databáze poskytovatele mapových podkladů). Pro tento typ užití aplikace je nejpřehlednější základní mapa (grafická mapa s popisy ulic/domů/čtvrtí apod.). Dále je pro dobrou orientaci vhodné zobrazení aktuální polohy uživatele.

Druhým stavebním kamenem je spolupráce na datech v aplikaci. Cílem je poskytnout co nejnovější data a zároveň filtrovat i kvalitu dat. Uživatel bude mít možnost přidávat nová zajímavá místa, přidávat recenze k již existujícím místům nebo hodnotit recenze ostatních uživatelů. Na základě těchto funkcí bude zajištěno zobrazování dat, které nejlépe odráží skutečnost. Součástí hodnocení pak je také schopnost aplikace zpracovávat multimediální obsah, zejména fotografie.

Třetím stavebním kamenem důležitým pro užívání aplikace je rozumná míra použitelnosti v prostředí bez internetového připojení. Tato aplikace je silně závislá na jejím pravidelném užívání a aktualizacích dat od uživatelů, nicméně při používání se uživatel může dostat do situace, kdy bude muset aplikaci využít i bez internetového připojení.

4.1.2 Serverová aplikace

Serverová aplikace slouží jako centrální úložiště dat a jejich zprostředkování uživatelům. Zároveň slouží jako autentizační autorita pro uživatele.

Požadavky na serverovou část aplikace jsou následující:

První částí je samotná databáze s daty o uživateli a jednotlivých místech. Server musí disponovat rozumnou rychlostí odezvy při poskytování těchto dat, do budoucna se předpokládá vysoké množství poskytovaných záznamů na celosvětové úrovni. V rámci přístupu k datům bude třeba vymyslet katalog dat takovým způsobem, aby byla data zprostředkována na co nejkratší vzdálenost a server musel vykonat algoritmicky co nejméně operací při poskytování těchto dat.

Druhou částí je samotný program, který na serveru poběží a bude zajišťovat komunikaci s koncovým zařízením a zprostředkovávat přenosy požadavků od klienta a vyžádaných dat zpět klientovi.

Webové rozhraní projektu je třetí součástí, kde musí existovat minimálně internetová stránka informující o existenci projektu. Dodatečně na této stránce mohou být i další funkce, jako třeba administrace profilu (primárně však většina funkcí bude zabudována do aplikace), a v budoucnu i webová verze aplikace.

Kritickým faktorem v rámci tohoto projektu je i cena. Zatímco mobilní aplikaci lze vyrobit bez jakýchkoliv finančních nákladů, serverová část se s největší pravděpodobností realizovat bezplatně nedá (tak, aby server měl dostatečnou kapacitu pro poskytování služby). Při hledání řešení bude taktéž nutné hledat i nejlevnější řešení.

4.2 Rozbor existujících aplikací

V této kapitole jsem popsal některé z aplikací, které slouží pro zobrazování map a míst v nich. Samozřejmě není možné postihnout každou aplikaci a každou eventuální funkcionalitu vzhledem k množství aplikací, proto jsem vybral pět různých aplikací, které se držely vysoko v žebříčku Google Play obchodu. Tři z těchto aplikací jsou velké a komplexní, tedy zobrazují nejen místa a umožňují recenzovat obsah, ale zároveň slouží například i jako navigace, informují o dopravě, nebo mají jiné dodatečné funkce. Dvě další aplikace jsou naopak menší projekty, které byly vyvinuty jako úzce zaměřené.

V tomto rozboru se zaměřuji zejména na komponenty uživatelského rozhraní a celkový dojem z používání aplikace, samozřejmě se ale jedná také o samotný popis aplikace a její hodnocení jako celek, protože se jedná v některých případech o potenciální konkurenty. Komentovány však jsou jen části, které jsou relevantní k předpokládanému rozsahu mé aplikace.

Tento rozbor aplikace poslouží jako inspirace pro volbu jednotlivých funkcí a jejich podobu zpracování v praktické části diplomové práce.

4.2.1 Google Maps

Jedná se asi o nejkompexnější aplikaci v testování. Google Maps disponují širokou škálou funkcionalit, stejně tak jako složitějším UI.

4.2.1.1 Mapy

V mapových podkladech se nachází různé typy bodů, při oddálení se zobrazují zejména historické památky nebo některé důležitější body, jako například nákupní centra. Při přiblížení se pak postupně objevují různé další značky pro hotely, restaurace apod., stejné značky se však objevují na různých úrovních a není jasné, podle jakého klíče jsou tyto značky přednostně zobrazovány (rychlý test ukázal, že se nejedná o zobrazování na základě četnosti recenzí nebo hodnocení, spíše je možné, že se jedná o dřívější zobrazení podniků, které si přednostní zobrazení zaplatily – třeba jako zobrazení na prvních místech v seznamu vyhledávání na Google). Aplikace umí zobrazit různé typy map (satelitní, terén), chybí však ovladače pro změnu přiblížení, tudíž s aplikací nelze při prohlížení pohodlně pracovat jednou rukou. Navíc mapa umožňuje otáčení mapy pohybem dvou prstů, občas

se tedy při neopatrné manipulaci mapa mírně otočí současně s přiblížením/oddálením, což ve výsledku také snižuje přehlednost mapy, respektive orientaci v mapě.

4.2.1.2 Rychlé zobrazení bodu

Při kliknutí na bod se ve spodní části zobrazí přehled o daném bodě. V tomto přehledu však není příliš informací, pouze hodnocení, počet hodnotících a poté prvky týkající se dopravy k tomuto bodu. Dále je přítomen element, který otevírá Street view v daném místě.

4.2.1.3 Detailní zobrazení bodu

Karta s detailními informacemi se dělí do tří sekcí.

Sekce „Přehled“ zobrazuje popis místa a dále jednotlivé informační položky, jako je adresa, telefon, otevírací doba, webové stránky a další. Mezi další funkce této sekce patří navigace k místu, uložení mezi oblíbené položky, sdílení záznamu a stažení lokální mapy.

Sekce „Recenze“ zobrazuje celkové hodnocení a počty hodnocení dle přidělených hvězd. Dále obsahuje hlasovací hvězdy, které umožní uživateli přidat recenzi a následně seznam recenzí (počet hvězd, text, případně fotografie) jednotlivých uživatelů. Recenze lze podpořit ikonou „To se mi líbí“, recenzi jde nahlásit jako nevhodnou. Recenze jsou seřazeny dle času od nejnovějších, do hodnocení se ale promítají i recenze staré i několik let. Zajímavý problém nastává při hlasování. Uživatel zvolí počet hvězd a je přesměrován na vyplnění formuláře s dodatečným textem a přidáním fotografie (text i fotografie jsou nevyžadované položky). Takovouto recenzi lze na horní liště buďto odsouhlasit a zveřejnit, nebo křížkem recenzi zrušit. Naneštěstí však i při stisknutí křížku se recenze zapíše do systému a to v podobě přidělených hvězd, tudíž pokud si akci uživatel při vyplňování recenze rozmyslí, stejně musí zpětně svojí recenzi smazat.

Třetí sekcí jsou Fotografie, kde se v mřížce zobrazí seznam všech fotografií, které byly nahrány všemi uživateli. Zde lze také přidat novou fotografii.

V kontextové nabídce karty místa je také možné navrhnout úpravu místa, kde má uživatel možnost navrhnout nový název, adresu a další doplňující informace, nebo dále označit místo jako neaktuální (uzavřené, nebo přesunuté).

4.2.1.4 Přidání nového místa

Přidání nového místa je možné z menu, z mého pohledu by tato položka mohla být více zvýrazněna, volba se nachází až ve spodní, hůře viditelné části menu, a bez ikony. Samotná karta přidání je poměrně jednoduchá, uživatel zadá název místa, adresu (textově, nebo vybere na mapě), kategorie, respektive typ bodu, fotografie a případně další položky, jako je telefon, otevírací hodiny a podobně. Nové místo je pak odesláno ke schválení a Google poté informuje o schválení nebo zamítnutí.

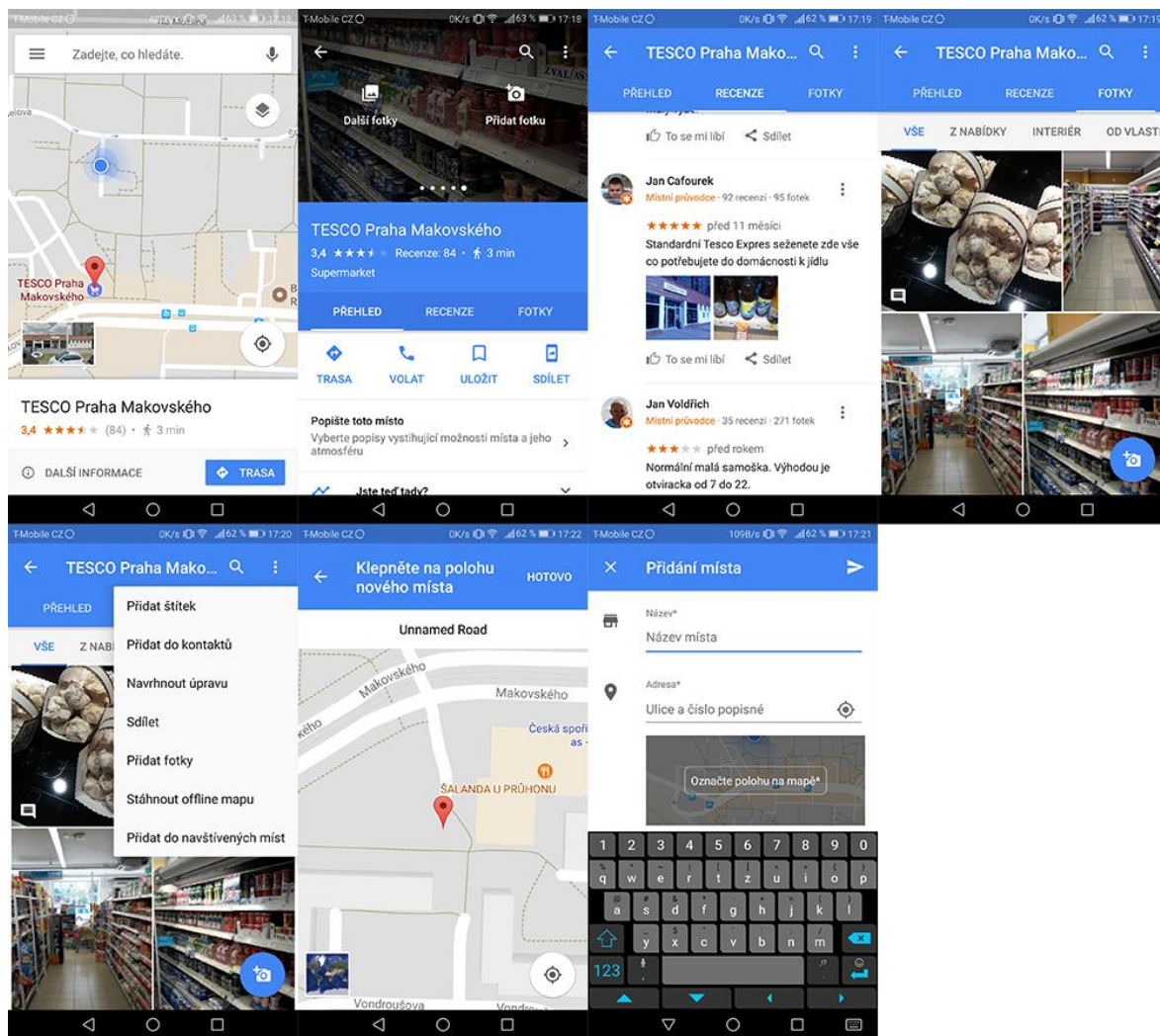
4.2.1.5 Další zajímavé funkce

Aplikace umožňuje omezit připojení pouze na WiFi, což může některým uživatelům, zejména pokud navštěvují město, pomoci snížit spotřebu mobilních dat.

Aplikace umožňuje stažení mapových podkladů, což opět ušetří přenášená data při používání aplikace.

Aplikace umožňuje uživateli se zapojit do vylepšování informací o daném bodu. Tato volba je však schovaná pod volbou „Vaše příspěvky“. Zde je možné se zapojit dvěma způsoby, na mapě se objeví ikony bodů a uživatel je vyzván buď k doplnění chybějících informací, nebo verifikaci již existujících informací. Verifikace se však omezuje pouze na otázku „Je tento název správný pro jazyk Angličtina?“ a nikoliv i na další položky. Také body, které mapa zobrazuje, jsou pouze s univerzální ikonou „+“, což je nevhodné pro celkovou přehlednost. Zde existuje alespoň sada filtrů, které omezí zobrazované body, ovšem za cenu zabránění přibližně třetiny obrazovky displeje blokem s filtry.

4.2.1.6 Ilustrační obrázky



Obrázek 1 - Ilustrační obrázky - Google Maps

4.2.2 Mapy.cz

Druhá velká aplikace společnosti Seznam, která v ČR konkuruje Google mapám. Mapy se specializují na vysoký detail map a navigaci (zejména pro turisty). V porovnání s Google mapami se však jedná o aplikaci s mnohem menším rozsahem funkcí.

4.2.2.1 Mapy

Oproti Google mapám se dá říci, že mapové podklady jsou o dost propracovanější. Nejen že se mně osobně zdají mnohem přívětivější po designové stránce, zobrazují i výrazně více objektů a zajímavostí (například pamětní desky na budovách), stejně tak jako mapa celkově je lépe popsána. To však platí jen na území České republiky, za hranicemi je

aplikace prakticky nepoužitelná, v zahraničí není zanesena většina bodů, zejména bodů jako jsou hotely, restaurace a další.

Body se zobrazují různě na různých úrovních, některé podle typu, některé však jednoho typu na různých úrovních, opět pravděpodobně na základě předplacené priority ve výsledcích vyhledávání u Seznamu.

S mapou se příjemně pracuje i jednou rukou, aplikace zobrazuje jezdec na pravé straně obrazovky, kterým lze mapu plynule přibližovat a oddalovat. Aplikace dále umí zobrazit několik dalších typů map (zimní a leteckou).

4.2.2.2 Rychlé zobrazení bodu

Při kliknutí na bod se zobrazí pouze malý, s bodem plovoucí banner, který obsahuje název, adresu, případně jaké má podnik otevírací hodiny. Součástí je i ikona pro pěší navigaci k bodu. V tomto případě bych si dokázal na plovoucím banneru představit více informací. V případě kliknutí na místo, kde se nachází více bodů, otevře se vyskakovací dialog s těmito informacemi, kde jeho položky jsou právě jednotlivé bannery

4.2.2.3 Detailní zobrazení bodu

Karta detailu bodu obsahuje zobrazení fotografií v bloku záhlaví karty, název, ikonu pro navigaci, volby pro zobrazení v panoramatu nebo ve 3D pohledu, textový popis, blok pro hodnocení a poté další informativní položky bodu, jako je telefon, email, webové stránky, otevírací hodiny nebo geografickou polohu.

Hodnocení je zobrazováno pomocí hvězd, naneštěstí pro zobrazení jednotlivých hodnocení aplikace přesměruje v integrovaném prohlížeči na záznam na stránkách www.firmy.cz. Hodnotit pak lze přímo v aplikaci zadáním počtu hvězd a textovým příspěvkem. Recenze je pak odeslána ke schválení. Fotografie u hodnocení nelze přidávat, lze však přidat fotografie přímo k místu v kontextovém menu karty.

4.2.2.4 Přidání nového místa

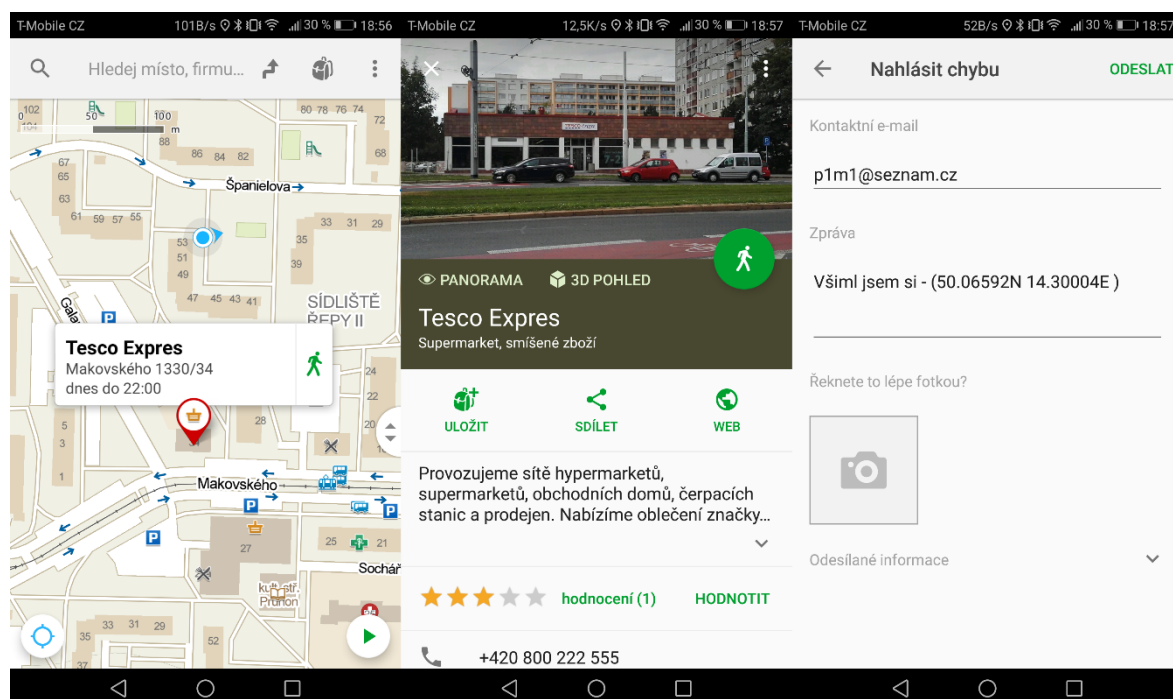
Přidání místa je velmi skryté. Je nutné dlouze stisknout v mapě, zobrazí se banner s geografickou polohou. Pokud jej rozklikneme, dostaneme se na stejnou kartu záznamu, jako u bodu, tato karta bude obsahovat fotografii z 3D pohledu a navigaci. Zde v kontextovém menu můžeme přidat nové místo, ovšem uživatel je odkázán na formulář

nahlášení problému (formulář má tento nadpis). Zde je předvyplněn email uživatele a do zprávy je předvyplněna geografická pozice. Dále lze přidat fotografii. Bod je takto odeslán ke schválení, respektive k jeho vytvoření administrátory, protože nelze zadat žádné přesné doplňující informace, ani typ bodu

4.2.2.5 Další zajímavé funkce

Zajímavou funkcí je obrazovka s první pomocí, kde se zobrazují nejen tipy pro první pomoc, aplikace tak rovnou umožní vyhledat nejbližší zařízení první pomoci – tedy označí nejbližší lékařské zařízení.

4.2.2.6 Ilustrační obrázky



Obrázek 2 - Ilustrační obrázky - Mapy.cz

4.2.3 Maps.Me

Opět se jedná o velkou aplikaci, která umožňuje zobrazení různých dat na mapovém podkladu, aplikace se dále specializuje na dopravu a dopravní informace

4.2.3.1 Mapy

Mapy samotné jsou dostatečně detailní, jejich grafika však je výrazně méně přehledná. Ve výchozím stavu se zobrazují 3D modely budov, ty se dají v menu vypnout. I tak ale zvolená škála barev se mi pro vyobrazení nezdá příliš vhodná, dominují velmi silné

barvy pro travnaté/přírodní povrchy v kontrastu se stupni šedi pro zástavbu, pro menší cesty jsou pak užity různé čárkované trasy.

Na mapě dominují body zejména kulturní, ostatní body, jako restaurace, obchody a jiná zařízení, se zobrazují až po přiblížení (z mého pohledu až při přílišném přiblížení). Opět je zde nastavena nějaká priorita zobrazování bodů stejného typu dle úrovně přiblížení.

S mapou se pracuje pohodlně, pro přiblížení a oddálení jsou zde tlačítka + a -, ovšem mapa umožňuje otáčení pomocí gest a stejně jako tomu je u aplikace Google Maps, pokud uživatel přibližuje/oddaluje gestem, může mapu nechtěně otočit.

Nevýhodou aplikace je nutnost stahování mapových podkladů a při přiblížení konkrétního regionu aplikace vyzve ke stažení.

4.2.3.2 Rychlé zobrazení bodu

Při kliknutí na bod se otevře ve spodní části obrazovky panel, který zobrazuje název, typ bodu a případně nějaké zajímavé informace (například „Restaurace; regionální kuchyně“), adresu, hodnocení, ikony pro uložení, bodu do oblíbených, navigaci a sdílení. Součástí je i reklamní banner.

4.2.3.3 Detailní zobrazení bodu

Detailní zobrazení se aktivuje vytáhnutím panelu rychlého zobrazení, tedy horní část detailního zobrazení se nemění. Na panelu se dále zobrazují recenze, další informace o bodu, jako telefon, web, u restaurací například typ kuchyně apod. Hodnocení však není konzistentní napříč body. V základu uživatel může hodnotit pěti různými „smajlíky“, které v následující obrazovce předvyplní rovnoměrně tři kategorie, ve kterých lze hlasovat hvězdami. Dále v recenzi lze uvést text, nikoliv však fotografii. Nicméně u některých bodů takto hlasovat nelze, například hotely zobrazují jiný typ recenzí (slovní „plusy a mínusy“), které jsou zdrojovány z portálu třetí strany. Z karty lze také záznam upravit pomocí formuláře s danými položkami.

4.2.3.4 Přidání místa

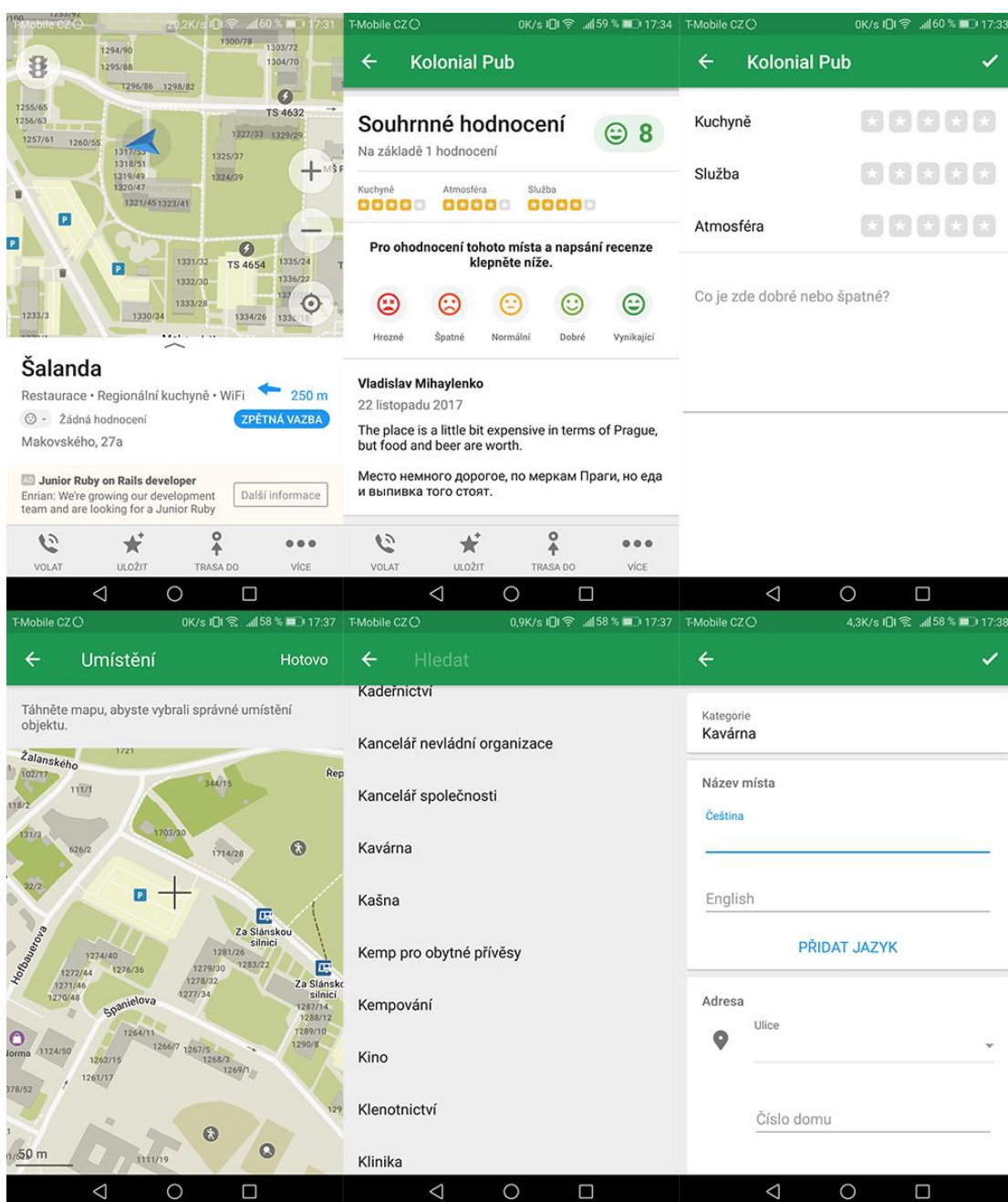
Přidání místa je v hlavním menu aplikace. Uživatel v módu přidání posouvá mapu s křížem uprostřed, kříž označuje místo. Poté volbu potvrdí a je vyzván k výběru ze seznamu typů bodu (atrakce, autoopravna, bankomat, bar apod. (celkově se jedná o několik desítek

položek). Následně se otevře formulář pro vyplnění potřebných položek, formulář je na míru postaven pro každý typ bodu. Následně je bod zaslán ke schválení.

4.2.3.5 Další zajímavé funkce

Zajímavá věc je spolupráce právě s třetími stranami, kde aplikace nabízí například zamluvení hotelového pokoje přes internetový portál, nebo objednání přepravní služby Uber.

4.2.3.6 Ilustrační obrázky



Obrázek 3 - Ilustrační obrázky - Maps.me

4.2.4 WC Kompas

Jedná se o úzce zaměřenou aplikaci, která se zaměřuje, jak již název napovídá, na vyhledávání toalet v okolí uživatele.

4.2.4.1 Mapy

Aplikace používá mapy od společnosti Google. Typ mapy je základní. Mapy také neukazují většinu bodů/míst, které mapy normálně poskytují (respektive místa se ukazují při vyšších úrovních přiblížení, ale nejsou aktivní pro další interakci), místo toho aplikace implementuje vrstvu vlastních bodů (při dostatečném přiblížení) nebo „shluky“, tj. číselný indikátor umístěný v mapě při oddálení, který určuje počet bodů v dané oblasti. Vzhledem k užívání Google map aplikace při práci s mapami trpí stejným neduhem jako aplikace Google Maps, kdy chybí prvky pro přiblížení/oddálení mapy a manipulace s aplikací pouze jednou rukou je obtížnější.

4.2.4.2 Rychlé zobrazení bodu

Rychlé zobrazení je v případě této aplikace velmi jednoduché, při kliknutí na bod se v daném místě objeví malý, s bodem plovoucí banner, který zobrazuje název a jednu relevantní položku popisu. Kliknutím na tento banner se otevře detailní zobrazení záznamu.

4.2.4.3 Detailní zobrazení bodu

Detailní zobrazení bodu se skládá z názvu zařízení, typu, vzdálenosti, fotografie (s možností přidat další fotografii), ikon pro zobrazení na mapě a navigaci, dále z dalších informačních polí a uživatelských komentářů. Součástí je také jednoduché hodnocení pomocí hvězd. Nebylo však možné získat bližší náhledy plně vyplněných míst, protože jsem nenašel jedinou ohodnocenou položku.

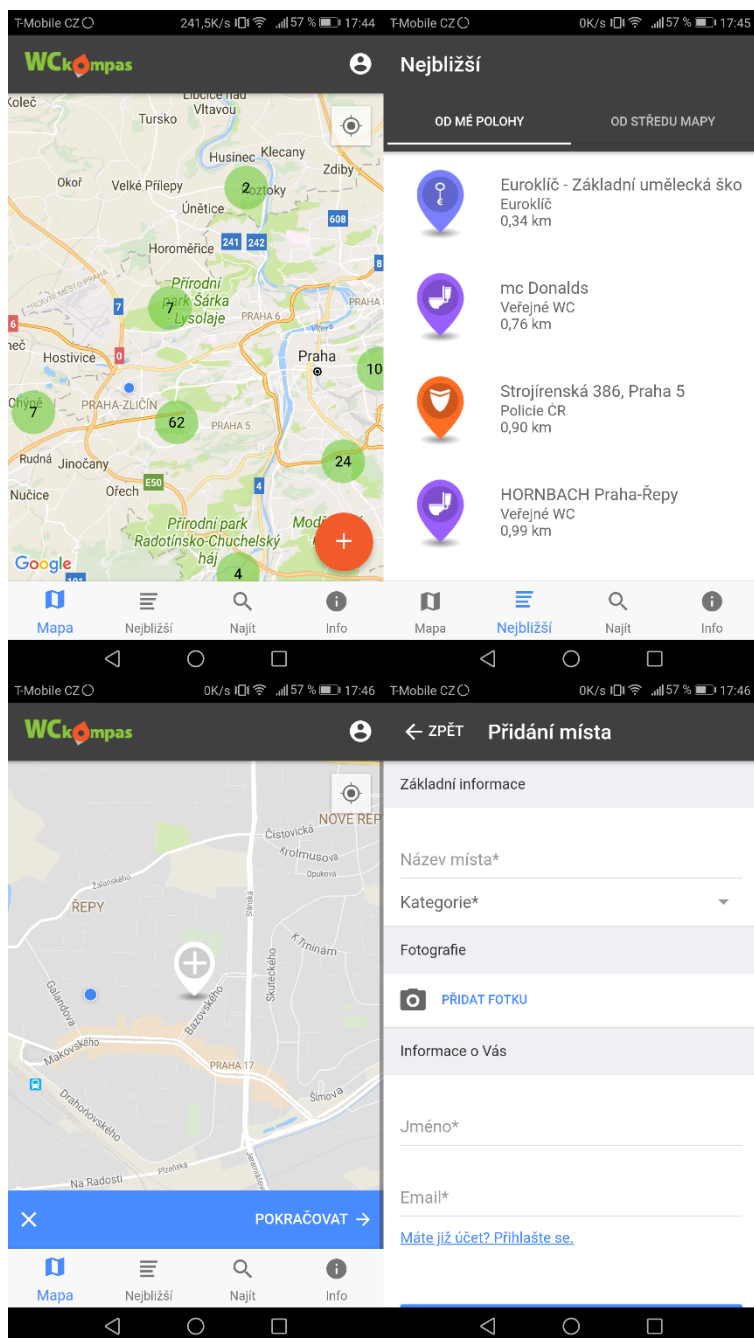
4.2.4.4 Přidání místa

Přidání místa je dostupné ihned z mapy pomocí tlačítka v rohu obrazovky. Po aktivaci přidání uživatel klikne do mapy, kam si přeje bod přidat, následně se otevře formulář s vyplněním názvu, kategorie, přidáním fotografie, na základě vybrané kategorie se pak následně ještě zpřístupní relevantní pole pro další popisy místa. Následuje ještě vyplnění osobních údajů a poté lze místo odeslat ke schválení.

4.2.4.5 Další zajímavé funkce

Aplikace má možnost zobrazení jmenného seznamu míst v okolí bez použití mapy. K dispozici jsou dva seznamy, jeden seznam zobrazuje místa v okolí uživatele (pomocí GPS), druhý seznam ukazuje body v okolí středu mapy (tedy v oblasti, na kterou se uživatel právě kouká v mapě).

4.2.4.6 Ilustrační obrázky



Obrázek 4 - Ilustrační obrázky - WC Kompas

4.2.5 Map Marker

Tato aplikace není spojena s jakoukoliv službou pro poskytování dat, slouží jako personální záznamník míst v mapě a souvisejících poznámek.

4.2.5.1 Mapy

Aplikace opět využívá mapové podklady od společnosti Google. Stejně jako u předchozích aplikací, které užívají tyto podklady, chybí prvky pro manipulaci s přiblížením. Aplikace zobrazuje originální body/místa mapových pokladů, ta jsou ale opět ponechána bez další možné interakce (v místech jde však hledat, přesunout se na jejich pozici v mapě a například spustit navigaci k místu, ovšem pomocí externí aplikace). Interaktivní jsou pouze vlastní body.

4.2.5.2 Rychlé zobrazení bodu

Při kliknutí na bod se ve spodní části obrazovky zobrazí panel s názvem bodu, popisem (pokud je vyplněn), telefonem (pokud je vyplněn) a datem vytvoření. Následují ikony pro editaci bodu, navigaci, kopírování dat bodu do schránky a volání přiřazeného telefonního čísla.

4.2.5.3 Detailní zobrazení bodu

Detailní zobrazení bodu je v tomto případě formulář pro jeho editaci. Tento formulář je relativně rozsáhlý, umožňuje grafickou editaci bodu, vyplnění názvu, popisku, telefonního čísla, přidání fotografií, a dále přidání velké škály vlastních polí, mezi které patří například email, textové pole, „dropdown“ menu nebo webový odkaz.

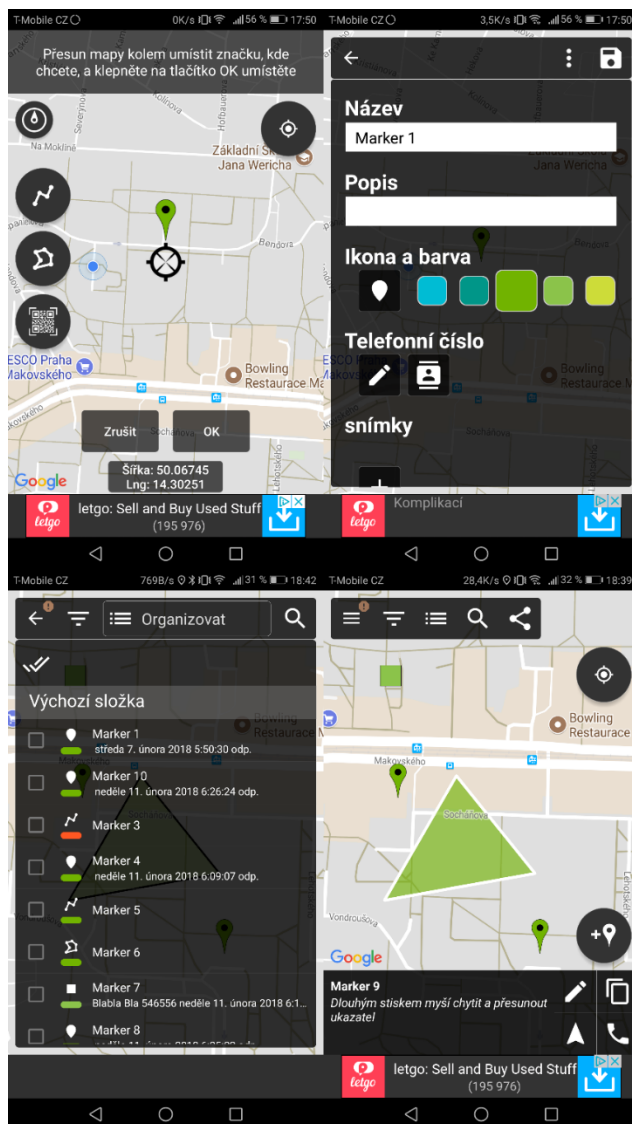
4.2.5.4 Přidání místa

Přidání místa je realizováno tlačítkem v mapě v dolním rohu obrazovky. Uprostřed se zobrazí zaměřovací kříž, uživatel pak posouvá mapu na dané místo. Potvrzením polohy se bod vytvoří. Uživatel pak následně musí bod editovat pro vyplnění informací.

4.2.5.5 Zajímavé funkce

Aplikace umožňuje vytvoření nejen bodu, ale také trasy a oblasti. Uživatel při přidání bodu pouze změní mód a pak postupně přidává jednotlivé body, které tvoří trasu nebo oblast. Po potvrzení se objekt chová pro editaci stejně jako bod.

4.2.5.6 Ilustrační obrázky



Obrázek 5 - Ilustrační obrázky - Map Marker

4.2.6 Shrnutí

V této kapitole jsem popsal některé existující aplikace, které řeší problematiku zobrazení míst v mapě. Obecně shledávám aplikace pro můj účel příliš komplexní, výjimkou je aplikace WC Kompas, která řeší zobrazování specifického typu míst a díky tomu disponuje i jednodušším uživatelským prostředím. Jako inspirace poslouží tedy spíše jednotlivé funkcionality, nikoliv řešení z komplexnějšího pohledu. Konkrétní zpracování jednotlivých částí aplikace na základě této analýzy řeším v praktické části práce.

4.3 Výběr frameworku pro vývoj aplikace

Cílem této kapitoly je popsat alespoň některá existující vývojová prostředí a vybrat prostředí pro vývoj této aplikace.

Vybrány k popisu byly spíše známější frameworky, podmínkou pak je podpora map, v tomto konkrétním případě Google map. Frameworky jsem vybíral tak, aby se zde objevil zástupce pro vývoj nativní aplikace pro zařízení, hybridní aplikace, nativní multiplatformní aplikace i licencovaný enterprise framework .

4.3.1 Android studio

Jedná se o nativní prostředí pro telefony s operačním systémem Android. Aplikace se programuje v jazyce Java. Výhodou je výkon aplikace a přístup ke všem zdrojům zařízení.

Jako nevýhodu vidím neintuitivní práci s UI. Studio pro návrh UI je nepoužitelné, velmi často dochází ke špatné detekci záměru vývojáře a rozložení prvků se rozpadne. Jediná bezpečná a vlastně nutná varianta je pak využití XML se specifickými Android značkami a vlastnostmi. I následná práce s UI komponentami v kódu je obtížná, v zásadě platí, že pokud nechceme využít předdeklarovanou komponentu, ale upravit její vzhled nebo výchozí chování, leckdy to znamená nepřiměřené úsilí oproti výsledku. Jedním takovým příkladem za všechny je nastavení položky seznamu tak, aby navíc zobrazovala i ikonu. V praxi toto znamená složitou adaptaci existující komponenty (Příloha 1 - Příklad úpravy ListView - Android studio).

Druhou nevýhodou je omezenost platformy. Ačkoliv práce se má věnovat aplikaci pro operační systém Android, je důležité vědět, že jedna platforma pro úspěšné rozšíření nestačí a bude potřeba do řešení zahrnout minimálně operační systém iOS.

4.3.2 Flutter

Jedná se o relativní novinku, první verze Flutteru byla vydána v květnu 2017 (1). Framework využívá jazyka Dart. Výrobce se chlubí bezkonkurenčním výkonem (aplikace se má kompilovat přímo do instrukcí) (2) (3) a použitelností v rámci UI (veškeré komponenty jsou na míru a nezávislé na platformě, tedy na všech zařízeních bude aplikace vypadat stejně a tvorba vlastních komponent není problém) (4). Další výhodou je úprava aplikace při vývoji v reálném čase. Flutter umožňuje provádět změny tak, že se ihned

projevují v zařízení, není nutné s každou změnou aplikaci sestavovat a odesílat na zařízení (5).

Tento framework na mě udělal dojem, i když z hlediska UI aplikace považuji vývoj opět neintuitivní (UI se programuje přímo v kódu do vnořujících se bloků kódu, složitější struktura se velmi rychle stává nepřehlednou (Příloha 2 - Flutter – příklad programování objektů UI). Důvodem zamítnutí pro vývoj mojí aplikace byla ale absence plnohodnotného mapového pluginu pro tento framework v době výběru.

4.3.3 Apache Cordova

Cordova je framework pro tvorbu hybridních aplikací. Hybridní aplikace je taková aplikace, která sice využívá nativních API zařízení, její uživatelská rozhraní a většina logiky aplikace je však řešena v prohlížeči, respektive ve WebView (6), což umožňuje v rámci HTML maximální flexibilitu zpracování grafického rozhraní aplikace. Pro uživatele se pak aplikace jeví jako každá jiná, reálně je mu však zobrazována HTML stránka.

Cordova tedy umožňuje programovat aplikaci jako jakýkoliv jiný běžný web, zároveň implementuje možnost přístupu k funkcím zařízení skrz pluginy (7). Tyto pluginy však mají javascriptové API a vývojář pak celou logiku aplikace plně programuje právě v javascriptu.

Cordova umožňuje kompilaci na řadu platform, kromě Androidu a iOS například i na Windows Phone, Windows, Linux, pro běžný prohlížeč a ve starších verzích Cordovy i na další platformy (Tizen, Amazon FireOS) (8) (9). Z tohoto hlediska je tento framework optimální.

Nevýhodou je snížený výkon, kdy je vlastě celý kód aplikace interpretován prohlížečem, v případě přístupu k nativnímu API musí být převáděna JS volání na volání API a odpověď převáděna zpět do JS. Problém je i hardwarová akcelerace vykreslování, která se neaplikuje na každý element, ale je nutné použít triky, které hardwarovou podporu vynutí (10). Většina současných telefonů má již dnes však dostatečný výkon na to, aby i takto vytvořená hybridní aplikace fungovala plynule.

4.3.4 Native Script

NativeScript je Framework pro vývoj multiplatformních aplikací. Dává vývojáři k dispozici jazyky Javascript, TypeScript nebo Angular (11). Uživatelské rozhraní je pak definováno pomocí XML (12).

Uživatelské rozhraní vytvářené tímto stylem však osobně považuji v jistém smyslu omezující, NativeScript je v tomto případě v základu vázán na nativní komponenty, jako je například ListView apod., Lze si sice naprogramovat vlastní UI komponenty, ovšem tento proces je již o něco složitější (13).

Na rozdíl od Cordovy však aplikace neběží uvnitř prohlížeče. Při kompilaci jsou objekty převáděny na nativní komponenty dané platformy. To může mít například vliv na vizuální stránku aplikace napříč platformami (14).

NativeScript podporuje platformy Android a iOS. Stejně jako Cordova je pak pro NativeScript k dispozici řada pluginů s různými pokročilými funkcionalitami.

4.3.5 Embarcadero RAD studio - Delphi

RAD Studio - Delphi je zástupcem enterprise řešení vývoje aplikací. Jedná se o komplexní vývojové studio, které je založeno na vývoji ve vizuálním prostředí (15). Vývojář na pracovní plátno vkládá široké spektrum komponent, a to jak komponent grafického rozhraní, tak komponent čistě funkčních, jako je například komponenta konektivity přes http. Většina těchto komponent má definice široké škály parametrů a akcí (events), na které reagují a většina těchto parametrů a akcí se dá nastavit v UI vývojového prostředí. Takto vzniká na pozadí zdrojový kód programu, prostředí vývoje je však navrženo tak, aby minimalizovalo potřebu kód upravovat.

RAD studio – Delphi poskytuje vývojáři jazyk Object Pascal, existují ale i varianty RAD studia s podporou jazyka C++.

Problémem v tomto případě je licence. Volně dostupná verze RAD studia umí kompilovat aplikace jen pro OS Windows, a to ještě jen jako 32-bit aplikace. Placené licence podporující kompilaci na více platformách začíná na částce přes 1700 dolarů (cena aktuální k 26. 3. 2017) (16).

4.3.6 Xamarin

Xamarin vznikl jako nezávislý projekt, který umožňoval při vývoji využít vlastnosti a schopnosti jazyka C#. Vznikl ovšem jako projekt komerční a dokud ho společnost Microsoft neodkoupila, řada jeho funkcí byla placená (například knihovna pro práci se sítí). Nyní je tento Framework zdarma a plně integrován do vývojového prostředí MS Visual Studio.

Již od začátku však i tento Framework kopíroval princip vývoje, známý z Android Studia. Tedy UI pracovalo výhradně s XML a nativními komponentami a ani práce s nimi v kódu nebyla snazší (při vývoji desktopové aplikace například ve Windows forms se ovládací prvek přetáhl na plátno, a tento prvek se adekvátně sám zapsal do příslušných míst v kódu a byl vývojáři ihned k užití. Jakmile se však vytvořil projekt pro mobilní aplikaci, musel se prvek umístit na plátno – ručně, nebo pomocí XML, a následně ještě deklarovat v kódu, úplně stejně jako v Android studiu, dokonce ani syntakticky se zápis prakticky nelišil). Tento stav přetrvává do současné verze (17).

Jako další velkou nevýhodu zde vidím nemožnost pracovat s jedním kódem pro obě podporované platformy (Android, iOS), i když se jedná o projekt multiplatformní aplikace, je nutné psát specifický kód pro každou platformu zvlášť (18).

4.3.7 Shrnutí

V této kapitole jsem popsal některé z existujících frameworků pro vývoj multiplatformních aplikací. Volba frameworku je věcí individuální, každý Framework má své zastánce i odpůrce. Já jsem si vybral pro vývoj Framework Apache Cordova kvůli možnosti kompilace na více typů zařízení a kvůli návrhu uživatelského prostředí, kde k vývoji webových stránek mám nejbliže.

5 Vlastní řešení

Tato kapitola pojednává o výběru funkcí, které bude finální aplikace obsahovat, návrhu uživatelského prostředí a dále popis technického zpracování samotné aplikace a serverové části.

5.1 Výběr funkcí

V této části následuje seznam a popis funkcí mobilní aplikace. Jednotlivé funkce vychází z dříve analyzovaných aplikací a z potencionálních potřeb uživatele vysledovaných z videomateriálů zmíněných v úvodu této práce. Jedná se však o obecné stanovení jednotlivých funkcionalit bez návaznosti na konkrétní obrazovku. Rozvržení funkcionalit a obrazovek bude dále věnována část kapitoly návrhu a tvorby aplikace.

5.1.1 Mapy

Klíčovou část aplikace tvoří mapové podklady a s nimi související API, které musí funkčně vyhovovat požadavkům. V aplikaci budou použity mapové podklady společnosti Google. Jedná se o bezesporu o nejrozšířenější mapové podklady a z globálního hlediska se dá předpokládat, že na Google mapy bude zvyklá většina uživatelů. Nejedná se však jen o uživatele na straně koncového zařízení, ale i o vývojáře, kde se dá předpokládat vyšší míra podpory komunity při řešení specifických problémů. S rozšířeností také souvisí i podpora map v různých prostředích pro vývoj, takže i když by se v budoucnu aplikace předělávala pomocí jiného vývojového software, podpora právě Google map by měla být ve většině případů zajištěna.

Součástí map budou, pokud to API dovolí, prvky pro manipulaci s přiblížením. Bude se jednat o tlačítka + a -.

Pokud to API dovolí, bude zrušeno načítání originálních bodů, které jsou mapami poskytovány za účelem snížení přenášených dat ze serveru. Aplikace bude řešit veškerý obsah svými vlastními body.

Při dalším vývoji aplikace se počítá i s řešením „shluků“ bodů podle aplikace WC Kompas, tato funkcionalita však pravděpodobně nebude součástí první verze aplikace, která vzniká v rámci této práce.

Součástí práce s mapou je i označení polohy uživatele pomocí GPS. V budoucnu by bylo zajímavé vyvinout i jednoduchou navigaci k bodu.

5.1.2 Seznam míst v okolí

Po vzoru aplikace WC Kompas bude realizován seznam míst v okolí uživatele a v okolí souřadnic, na kterých se právě nachází zobrazení mapy.

5.1.3 Přidání bodu

Uživatel bude mít možnost přidat nový bod na mapu. Prvek pro přidání nového bodu musí být jasně viditelný, jedná se o jednu z hlavních funkcionalit aplikace.

Pokud to API dovolí, výběr pozice se bude řešit kliknutím do mapy, nikoliv posunem mapy pod zaměřovacím křížem.

Přidání bodu bude řešeno jako průvodce, tedy prvním krokem bude výběr místa, druhým krokem výběr kategorie bodu a následně vyplněním příslušného formuláře.

5.1.4 Hodnocení místa

Hodnocení místa bude řešeno pomocí tří stupňů – výborné, dobré a špatné. Protože aplikace má primárně sloužit k zobrazení skutečně doporučených míst (tedy výborné, stojí za to je navštívit), nebo naopak k zobrazení míst, kterým se má uživatel vyhnout (podvodné podniky apod.), není třeba jemnějšího rozdělení doporučení.

Celkové hodnocení se pak bude skládat jak z hodnocení tvůrce bodu, tak z hodnocení ostatních recenzentů, a to rovnou měrou (hodnocení tvůrce nebude zvýhodněno nad hodnocením dalšího recenzenta).

5.1.5 Detailní zobrazení bodu

Detailní zobrazení bodu se bude skládat z hodnocení bodu, slovní recenze (vytváří zakladatel bodu, případně editor – editor viz. dále), fotografií s popisky a dalších doplňujících polí, jako jsou otevírací hodiny, jídelní lístky a další pole dle potřeby (řada polí bude přidána až v dalším vývoji aplikace). Součástí je i odkaz na další recenze uživatelů.

5.1.6 Náhled zobrazení bodu v mapě

V mapě se po stisku bodu zobrazí panel s hlavní fotografií a základními informacemi, tedy popiskem a hodnotou hlasování.

5.1.7 Systém hlasování

Všechny zkoumané aplikace mají problém s potenciální neaktuálností dat. Toto se ve velké míře nová aplikace pokusí eliminovat pomocí uživatelského hlasování.

5.1.7.1 Aktuálnost hlavních informací bodu

První uživatel zakládá bod včetně svého textu recenze, fotky a dalších informačních polí. U každého existujícího bodu bude k dispozici hlasování o správnosti informací, editaci nebo smazání bodu. V případě, že bod nashromáždí příslušný počet hlasů, bude buďto smazán, nebo otevřen pro editaci prvním uživateli, který editaci provede.

Pro další vývoj se počítá s hlídáním aktivity u bodu respektive recenze bodu, kdy pokud u bodu dlouho nepřibude podřazená recenze, hlas hlavní recenze nebo hlas podřazené recenze, systém označí bod jako potenciálně zastaralý a bude žádat potvrzení o aktuálnosti uživatele v okolí.

5.1.7.2 Pravdivost a aktuálnost recenzí

K bodu mohou uživatelé přidávat další recenze, i u nich je však potřeba hlídat aktuálnost a pravdivost. U každé recenze bude možnost odhlasovat správnost/nesprávnost. Pokud recenze získá příliš negativních hlasů, bude odstraněna.

5.1.8 Dlouhodobé a krátkodobé body

Toto dělení vychází z podstaty některých kategorií bodu. Může nastat například situace, kdy uživatel vytvoří negativní bod jako varování před nepoctivým řidičem taxi zaměřeným na okrádání cizinců, nebo na stánek s nevhodným občerstvením. Tento bod by měl být rozhodně v mapě zaznamenán, nicméně se jedná o bod, který nemusí být v řádu hodin či dnů aktuální (řidič taxi změní stanoviště, stánek již nebude na místě, protože skončily trhy, kterých byl součástí, apod.). Pro takové případy bude implementován bod s krátkou životností.

Krátkodobý bod bude aktivní po pevně stanovenou dobu, kdy se bude jevit jako jakýkoliv jiný dlouhodobý bod (s výjimkou jeho označení krátkodobého bodu v detailu a náhledu bodu). Po uplynutí této doby, nehledě na přidávané recenze, bude označen jako potenciálně neaktuální a bude vyžadovat hlasování o aktuálnosti. Pokud dostatek uživatelů potvrdí jeho aktuálnost, bod bude obnoven s původním hodnocením a znovu

se nastaví odpočítávání stanovené doby do opětovného hlasování. Pokud uživatelé budou hlasovat o neaktuálnosti, nebo po další stanovený čas nepřibude žádný hlas pro ani proti, bude daný bod automaticky odstraněn.

5.1.9 Fotografie

Jak již bylo zmíněno, k bodům se budou dát přidat fotografie. Fotografie budou moci obsahovat hlavní recenze (popis bodu) i další přiřazené uživatelské recenze. Fotografie budou vždy schovány pod vlastní volbou zobrazení, aby se omezil obsah přenášených dat. Fotografie, na rozdíl od ostatních aplikací, budou moci mít přiřazený popis.

5.2 Vizualizace aplikace

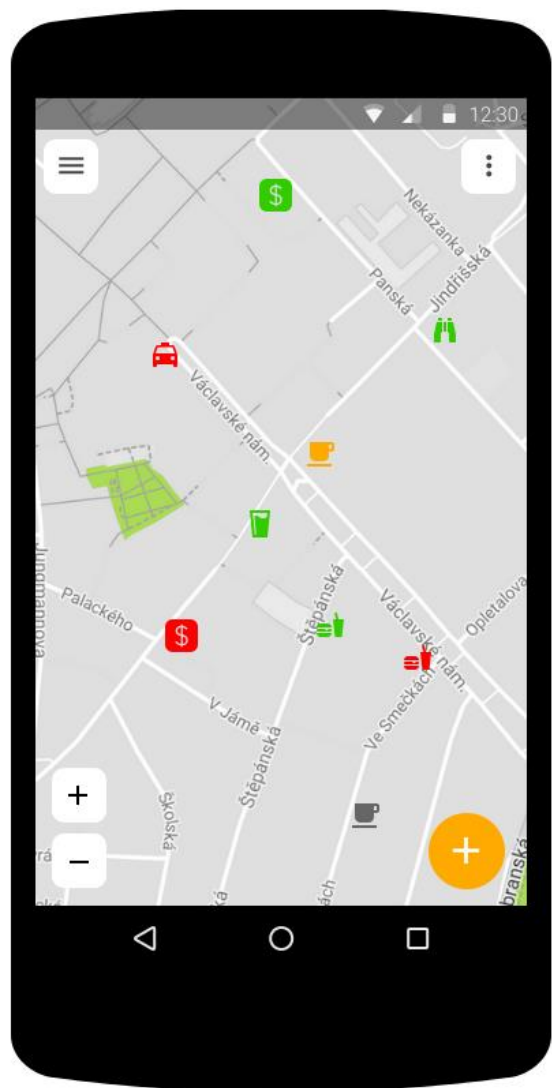
Tato kapitola má za úkol přiblížit koncept aplikace formou vizuálního náhledu. Jedná se pouze o koncept, spíše obecný předpis toho, jak bude finální aplikace vypadat. Finální podoba aplikace bude závislá zejména na technických možnostech vývojového nástroje při řešení jednotlivých úkolů a funkcionalit a také na zvolených grafických materiálech. Vizualizace byla tvořena v nástroji Evolus Pencil za pomoci dodaných grafických knihoven (stencils). Vizualizace řeší nejen rozmístění výše zmíněných funkcionalit a prvků na obrazovky, ale v některých případech i konkrétní workflow.

5.2.1 Mapa

Uživatel vidí mapu s ikonami typů míst (restaurace, směnárna, vyhlídka, atd.). Jednotlivé ikony jsou podbarveny podle hodnocení, tedy je na první pohled vidět, kterým místům se vyhnout, nebo které naopak navštívit. Mimo barvy doporučení (zelená, žlutá a červená) mohou být místa označena i šedivou barvou, a to v případě neaktuálnosti – ve stavu vyčkávání na ověření dat bodu.

Mimo to jsou na obrazovce dostupná tato tlačítka:

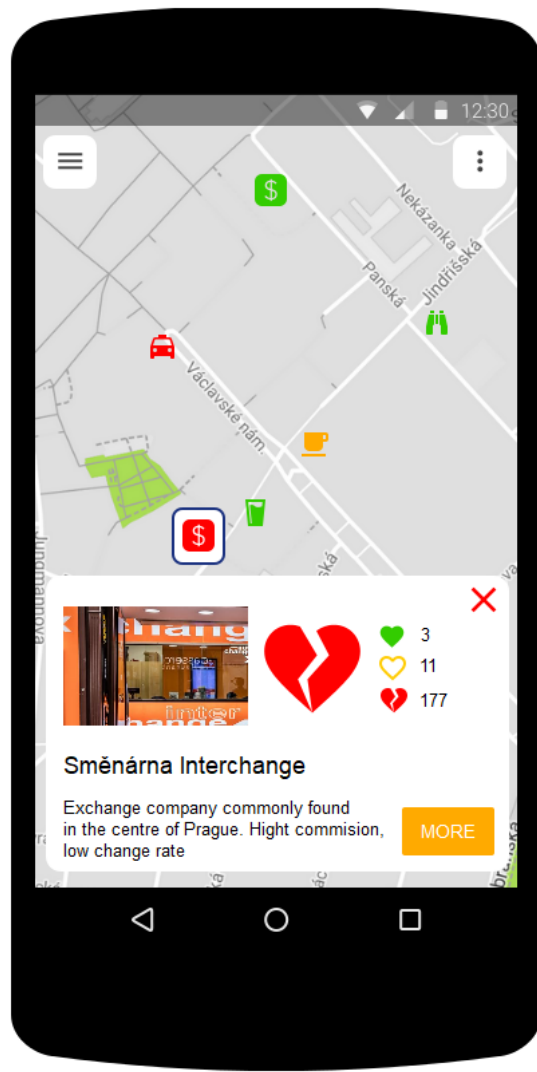
- Tlačítko přidání nového bodu
- Tlačítka pro přiblížení/oddálení mapy
- Tlačítko menu – vysouvá hlavní nabídku aplikace
- Tlačítko kontextového menu – přidáno v případě potřeby nebo dalších funkcionalit vážící se na mapu



Obrázek 6 - Vizualizace - Mapa

5.2.2 Rychlé zobrazení místa

Při kliknutí na ikonu místa se ve spodní části displeje otevře panel s fotografií místa, názvem místa, částí hlavního popisku/recenze a stavem hodnocení. Součástí jsou i tlačítka pro zavření panelu nebo otevření detailního popisku místa. Panel dočasně překryje tlačítka pro přidání bodu a přiblížení mapy.

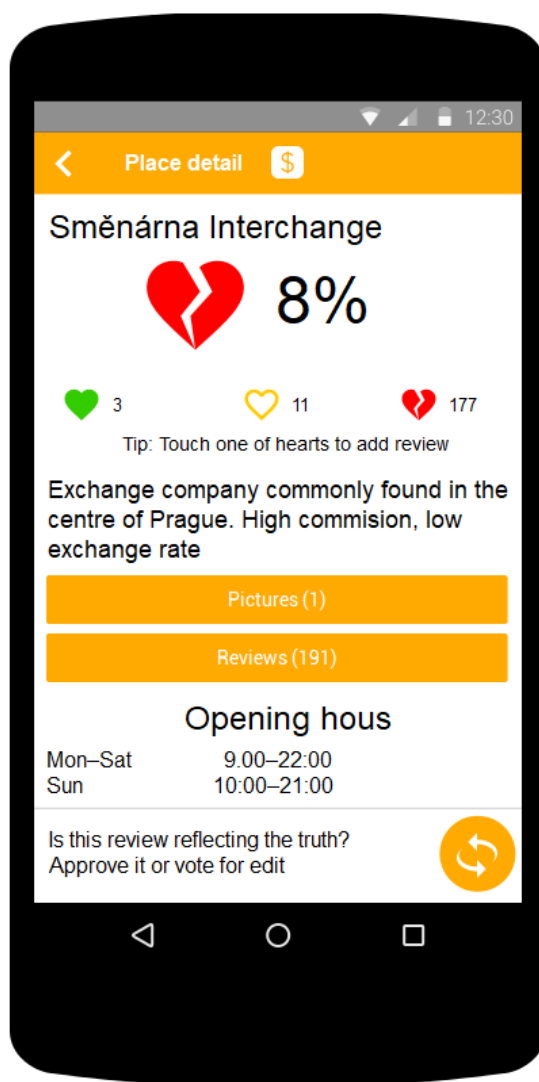


Obrázek 7 - Vizualizace - Rychlé zobrazení místa

5.2.3 Detailní zobrazení místa

V záhlaví je popisek s názvem obrazovky, na které se uživatel nachází společně s identifikátorem typu místa. Na stránce poté je již samotný obsah, tedy název místa, jeho hodnocení (celkové i dílčí). Tyto ikony jsou aktivní, s jejich pomocí lze přidat novou recenzi. Dále se na obrazovce nachází popis místa (úvodní recenze) a odkazy na obrázky recenze a další uživatelské recenze. Jak již bylo řečeno, tyto dva prvky nejsou integrovány do jedné stránky z důvodu potenciálně většího objemu stahovaných dat při zobrazování jednotlivých bodů. Poté se na obrazovce nachází další popisová pole, jejichž design i řešení bude záviset na jejich typu a jejich přítomnost na jejich vyplnění recenzentem.

Nakonec je součástí obrazovky mechanismus pro hlasování. Tlačítko společně s popiskem se nachází ve spodní části obrazovky. Po kliknutí na tlačítko se otevře vyskakovací okno s volbami hlasů (data jsou aktuální, data jsou zastaralá a měla by se upravit, nebo bod by měl být smazán). V případě otevření bodu pro editaci bude hlasování nahrazeno výzvou k editaci.



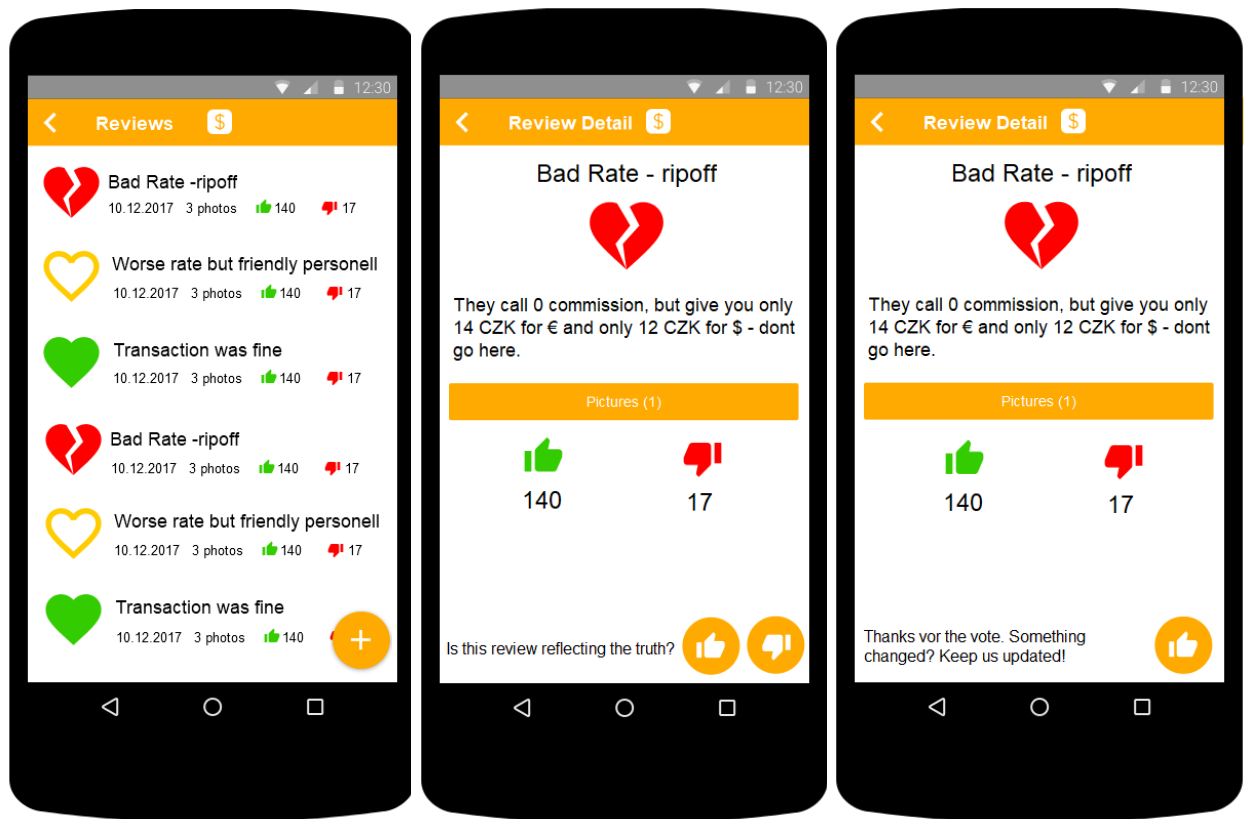
Obrázek 8 - Vizualizace - Detailní zobrazení bodu

5.2.4 Přiřazené recenze k záznamu

V záhlaví je opět název obrazovky, na které se uživatel nachází a identifikátor typu místa. Dále je zobrazen jednoduchý seznam přiřazených recenzí, kdy každá recenze zobrazuje hodnocení, název, datum, případně jiná metadata a hodnocení recenze (pravdivá/nepravdivá).

Po rozkliknutí recenze se na obrazovce zobrazí detail recenze, tedy název recenze, hodnocení, textový popis, a tlačítko s odkazem na fotografie. Dále se zobrazí také stav hodnocení samotné recenze.

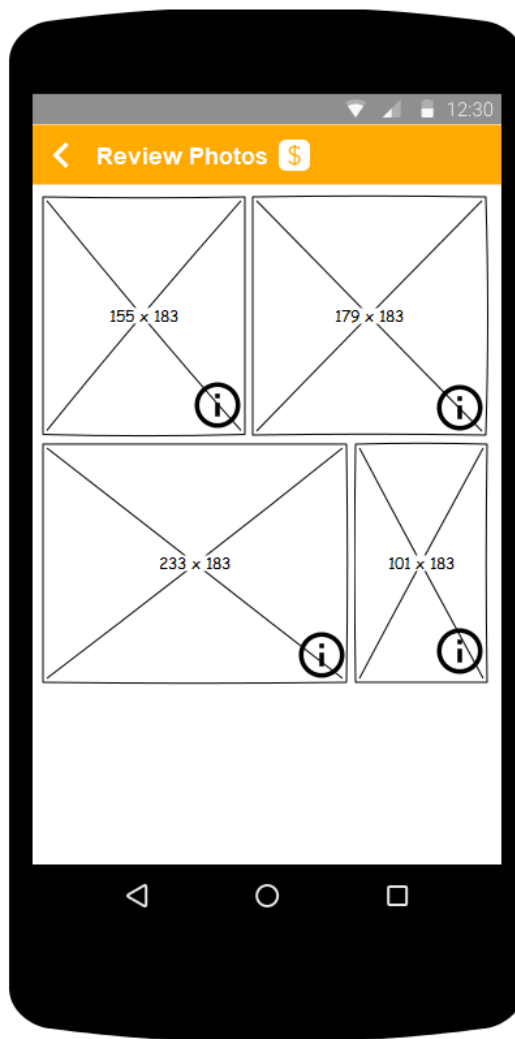
Ve spodní části obrazovky jsou k dispozici opět tlačítka pro hlasování, o samotné recenzi.



Obrázek 9 - Vizualizace - Podřazené recenze bodu

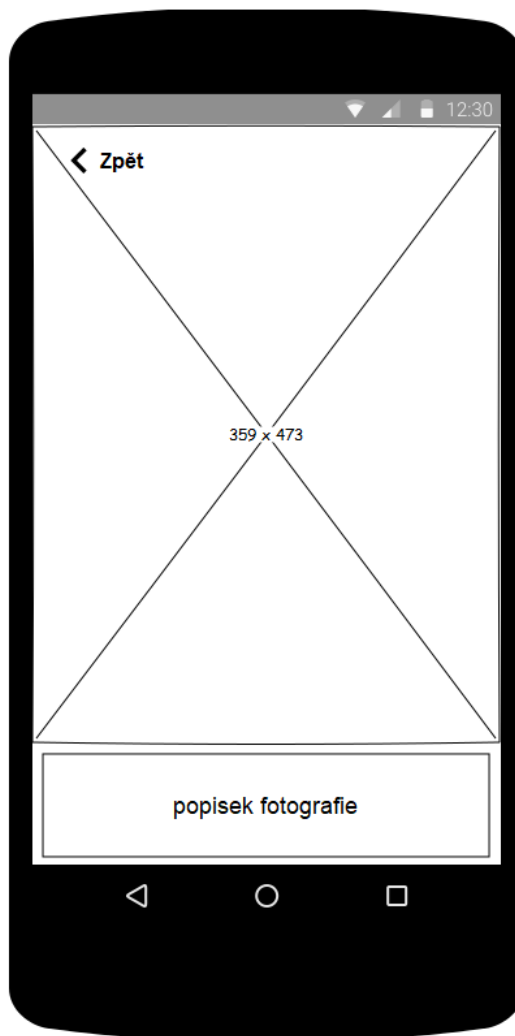
5.2.5 Fotografie

Tato obrazovka je stejná jak u fotografií hlavní recenze, tak u fotografií dalších přiřazených recenzí. V záhlaví je opět popisek s identifikátorem typu záznamu. Obsah obrazovky pak vyplňují náhledy obrázků (přesné rozvržení náhledů bude záviset na možnostech vývojového nástroje). Každý obrázek má přiřazenou informační ikonu, která ve vyskakovacím okně zobrazí popis obrázku.



Obrázek 10 - Vizualizace – Seznam fotografií

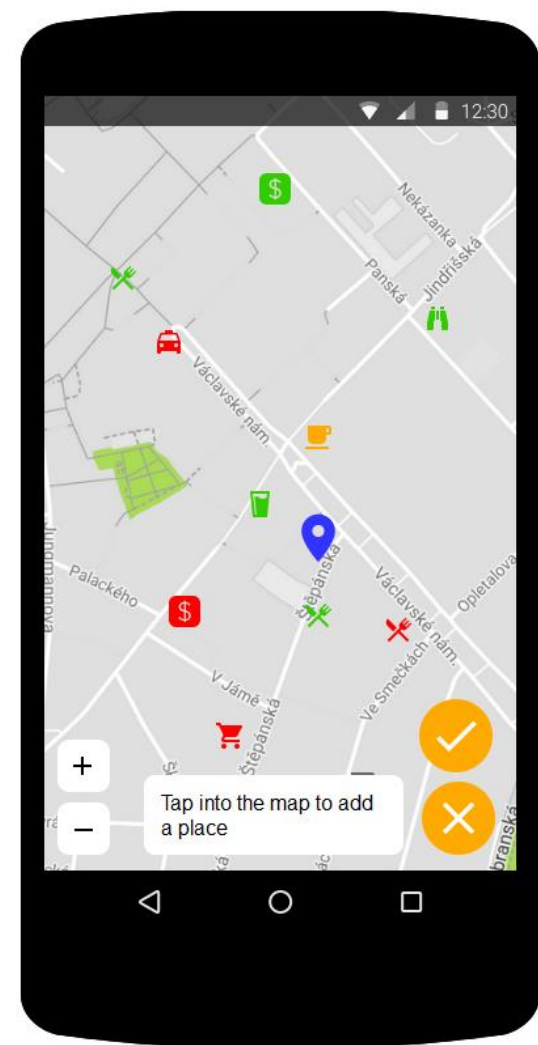
Po rozkliknutí fotografie se zobrazí celoobrazovkový náhled, kdy se zobrazí vždy fotografie a její popis (pokud popis chybí, fotografie vyplní celou obrazovku). Součástí je i navigační ikona pro přechod zpět k seznamu obrázků



Obrázek 11 - Vizualizace - Detail fotografie

5.2.6 Přidání záznamu

Vytvořit nový bod lze stiskem příslušného tlačítka v mapě. Mapa přejde do módu vytvoření bodu, ikony pro menu a kontextovou nabídku jsou schovány a tlačítko pro přidání bodu je nahrazeno popiskem a tlačítka pro potvrzení výběru pozice nebo zrušení akce.

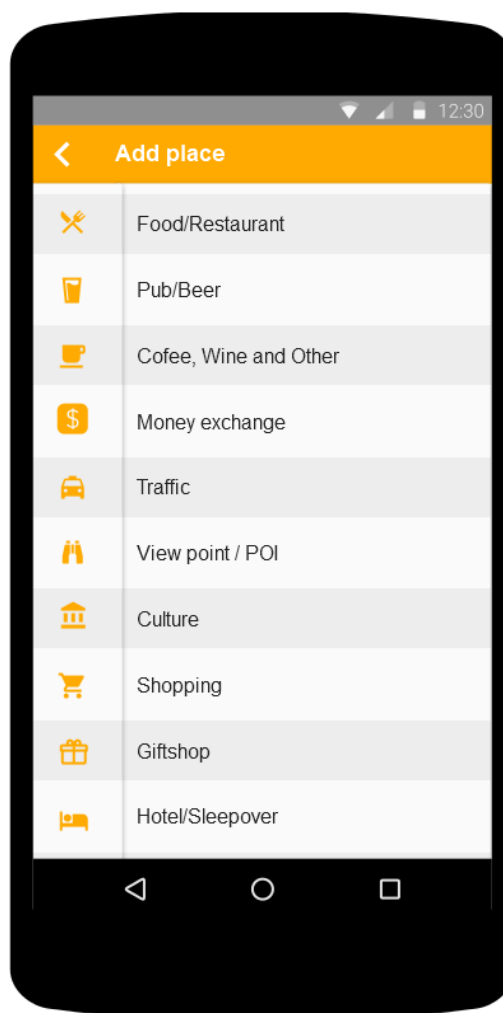


Obrázek 12 - Vizualizace - Přidání záznamu

Po potvrzení pozice nového bodu uživatel vybere typ bodu. Seznam těchto typů vychází ze sledování průvodce na stream.cz, na jehož základě je tato aplikace realizována, jedná se o tyto typy:

Restaurace/jídlo, Hospoda/Pivo (ve smyslu spíše podniku bez kuchyně, zaměřeného na občerstvení), Kavárny/vinárny/ostatní restaurační zařízení, Směnárný/Směna peněz (směnárný, ATM, pouliční směna), Doprava (zejména stanoviště Taxi apod.), Vyhlídka (nebo nějaký zajímavý bod v okolí obecně), Kulturní místo, Obchody/nakupování (běžné nákupy jako jídlo, oblečení atd.), Dárkové obchody (zejména pro doporučení nákupu lokálních dárků/specialit), Hotely a jiné noclehy, Ostatní bod/nezařazeno.

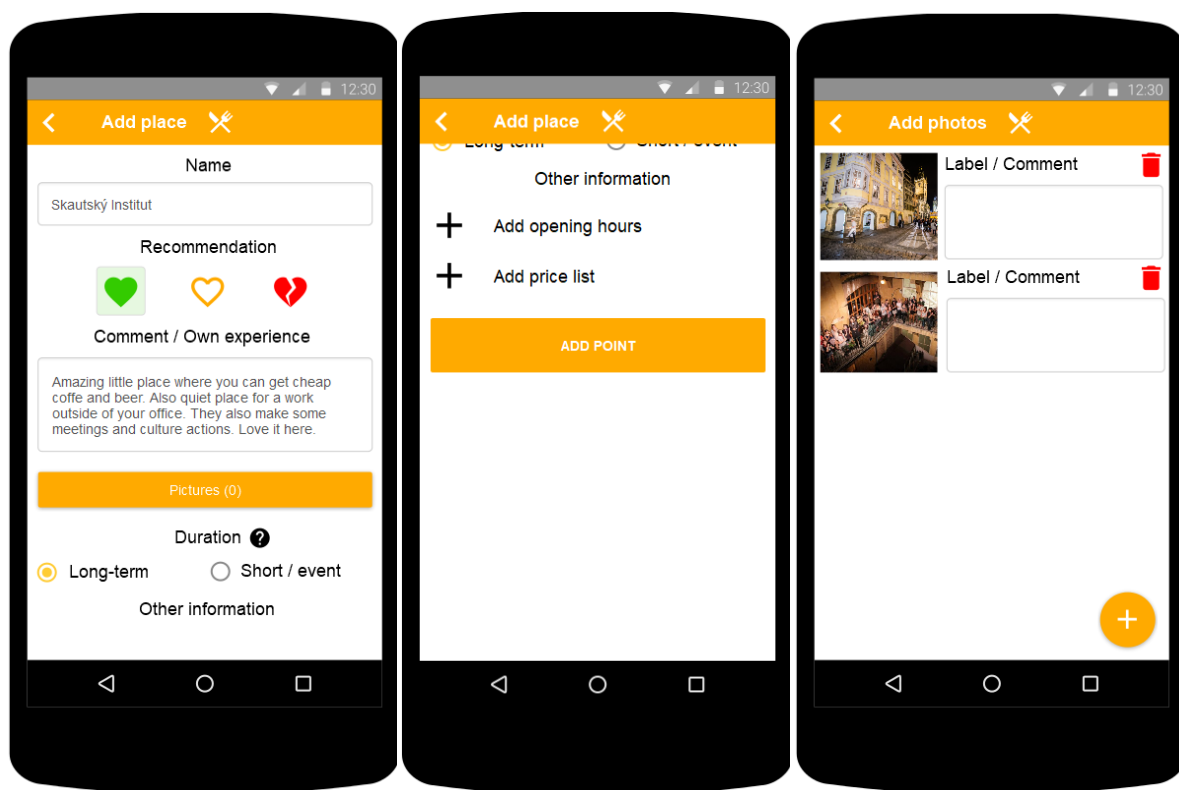
Jednotlivé kategorie mohou být upraveny dle aktuálních potřeb.



Obrázek 13 - Vizualizace - seznam typů bodu

Následně je uživateli zobrazena obrazovka s vyplněním formuláře. Zadává název, slovní hodnocení, celkové hodnocení (velmi dobré, dobré, špatné), dále uživatel volí, jestli se jedná o krátkodobý nebo dlouhodobý bod, a nakonec může přidat další volitelná pole, jako například otevírací hodiny, URL adresu a další. Fotografie jsou řešeny na samostatné obrazovce, kde uživatel nahraje fotografii a přidá popisek. Součástí je samozřejmě i fotografii odstranit. Tyto fotografie jsou pak zobrazeny v seznamu.

Pokud je uživatel s vyplněním spokojen, pomocí příslušného tlačítka bod akceptuje a je přesunut zpět na mapu.



Obrázek 14 - Vizualizace - přidání bodu

5.3 Vývoj aplikace

Tato kapitola je věnována dílčím problémům při vývoji aplikace. Postihuje architekturu aplikace, jak z pohledu serveru, tak samotné mobilní aplikace a dále se věnuje jednotlivým problémům a zajímavostem při vývoji.

5.3.1 Cena provozu aplikace

Tato aplikace je a s největší pravděpodobností i dále bude vyvíjena jako neziskový projekt. To s sebou samozřejmě přináší řadu omezení, která se projeví při vývoji.

Prvním omezením je cena za vývojové prostředí. Ačkoliv volba v tomto bodě pro mě byla jasná – tedy hybridní aplikace založená na webových technologiích, pokud bych měl k dispozici finanční prostředky, zvolil bych spíše PhoneGap, který je nadstavbou pro Cordovu a umožňuje například výrazně snazší kompilaci pro více platforem (není třeba mít knihovny pro danou platformu, kompilaci řeší cloud PhoneGapu) (19).

Druhým, a vlastně největším omezením je cena za hosting řešení na serveru. Free hostings mají velmi často výrazná omezení, ať již velikost databáze, velikost diskového prostoru, nebo třeba maximální počet denních přístupů. Tuto problematiku řeším návrhem datové struktury celé databáze, která umožňuje rozdělit serverovou část na téměř libovolný počet serverů, kde aplikace bude vědět dle geografické polohy, k jakému serveru se má dotázat na požadovaná data.

Součástí problematiky ceny hostingu je i cena za doménu. Pokud má aplikace „prorazit do světa“, musí mít řádnou identitu, jejíž součástí je i vhodně zpracovaný web, ale především doména webu (jakákoliv aplikace, nebo spíše můžeme říct jakýkoliv produkt obecně, pokud je inzerován na doméně třetího řádu, nebude u uživatelů vzbuzovat takovou důvěru). Náklady na malý hosting s doménou jsou však minimální, a to v řádu pár set korun ročně, v závislosti na poskytovateli.

Pokud však aplikace poroste, bude nutné zvážit získání financí na adekvátní serverové prostředí, tedy výkonnější hosting s menšími limity. Možností je několik, například:

- 1) Vložení reklamy – jednoduše lze vložit reklamu Google AdSense. Druhou možností je individuálně kontaktovat společnosti s možností inzerce v aplikaci.

Osobně však nejsem příznivcem reklam, které jsou dnes všudypřítomné a ve velké řadě případů obtěžující. Bylo by nutné reklamy vkládat tak, aby jimi uživatel nebyl konfrontován příliš často. Problém však je, že tyto reklamy generují zisk zejména při jejím zobrazení nebo prokliku (PPV, PPC). Minimalizace zobrazování reklamy uživateli je tedy poněkud v přímém rozporu s podstatou reklam. V každém případě se reklamy obvykle vyplatí až od vyššího počtu uživatelů (20), tedy tato otázka prozatím není příliš relevantní a její řešení bude záviset na aktuální situaci okolo reklam až ve chvíli, kdy aplikace již bude mít stabilní uživatelskou základnu.

- 2) Dobrovolné příspěvky – druhou variantou je dát uživateli možnost přispět libovolnou částkou na provoz projektu. Zde je možností více, ať již přímé zaslání peněz přes PayPal účet, možnost přispět kryptoměnou, nebo například využít služeb příspěvkových portálů, jako je například portál buymeacoffee.com. Zde je však dobré uživatelům dát možnost vidět, na co se dané peníze skutečně používají. Myslím, že pokud žádám od svých uživatelů peníze na neziskový projekt, uživatel by měl mít jistotu, že jeho peníze skutečně jsou použity na správný účet. Inspiraci zde čerpám z některých komunitních herních serverů, které také často pracují s dobrovolnými příspěvky a některé dokonce zpřístupňují vyúčtování za svůj provoz (obvykle faktura od poskytovatele serveru apod.). Pokud by se aplikace vydala tímto směrem, zobrazovala by aktuální stav nákladů na síť serverů, výši vybraných příspěvků a podobně.
- 3) Nákupy v aplikaci – třetí možností je vytvořit nějakou funkcionalitu (nebo nějakou funkcionalitu omezit), která se zpřístupní až po zaplacení nějakého balíčku. Dle mého názoru ovšem toto není typ aplikace, ve kterém by se nákupy v aplikaci daly efektivně použít. Pokud v aplikaci navíc dojde k nějakému omezení, je pravděpodobnější, že jí uživatel přestane používat, což jde ovšem proti logice aplikace, která je naopak závislá na co největším počtu uživatelů.

5.3.2 Serverová aplikace

U serverové aplikace považuji za nejdůležitější 3 oblasti. Technologické zpracování, tedy jakým způsobem server pracuje a jaké technologie využívá, datovou strukturu,

tedy s jakými daty pracujeme a dle jaké logiky s nimi pracujeme, a třetí částí je zabezpečení, tedy jakým způsobem jsou data chráněna proti zneužití. Tyto oblasti tedy budou pokryty v následujících podkapitolách.

5.3.2.1 Technologické zpracování

Serverová část mé aplikace je navržena tak, aby mohla pracovat téměř na libovolném hostingu. Předpoklad je, že server nebude jen jeden, ale bude rozdělen do několika uzlů.

Pokud se aplikaci podaří rozšířit do světa, bude třeba zajistit maximální přístupnost dat v aplikaci. Touto přístupností je myšlena plynulost poskytování dat. Pokud se například aplikace rozšíří po Evropě a Spojených státech amerických, bylo by vhodné, aby data relevantní pro danou oblast byla dostupná ze serveru nejbližší dané oblasti (tedy aby data relevantní pro Spojené státy všichni uživatelé nenačítali z Evropy a naopak), a to zejména kvůli odezvě mezi aplikací a serverem, ale také kvůli nižší rychlosti spojení při připojení přes mobilní data.

Této myšlence je třeba přizpůsobit nejen datovou strukturu, která je diskutována dále, ale i volba technologií, která umožní přenositelnost částí serverové aplikace na různé servery v různých zemích.

Serverová aplikace bude tedy napsána plně ve skriptovacím jazyce PHP, který je dostupný prakticky u všech běžných poskytovatelů webového hostingu. Samotný kód pak nevyužívá žádné specializované funkce, které by vyžadovaly aktivaci některých dodatečných knihoven, serverová aplikace by pak měla být schopna běžet na libovolném PHP serveru s výchozím nastavením.

Stejně je to i s databází, využita bude databáze MySQL. Ačkoliv by se našla jistě řada jiných, výkonnějších databází, klíčovou vlastností je právě rozšířenost MySQL, která v budoucnu zajistí bezproblémovou migraci dat na nový server nebo na další rozšiřující servery.

5.3.2.2 Datová struktura

V této podkapitole je znázorněna a vysvětlena struktura dat a její dělení mezi jednotlivé uzly. Jedná se však o první návrh struktury a v průběhu vývoje se tato struktura může změnit.

Datová struktura je rozdělena do tří částí, respektive do tří databází a to zejména kvůli přehlednosti ukládaných dat. Využití tří databází sice ve výsledku může omezit výběr poskytovatele hostingu (některý poskytovatel, zejména pokud je služba zdarma, neumožní vytvoření více databází pod jedním účtem), na druhou stranu kromě přehlednosti získáme i výkon, kdy zvolená struktura umožní procházení menšího objemu dat při zadání požadavku do databáze.

Pracovně jsem databáze pojmenoval MAIN, POINTS a VOTES. Databáze MAIN je databáze určená pro autentifikaci uživatelů a jednotlivých zařízení. Databáze POINTS drží informace o jednotlivých bodech a všech příslušných recenzích k daným bodům a databáze VOTES drží informace o všech hlasováních ke všem jednotlivým recenzím. V budoucnu nejspíše přibude ještě databáze PICTURES pro obrázky.

(Pozn.: v následujících tabulkách se v popisu nachází definice cizích klíčů. Vzhledem k tomu, že celá datová struktura bude fragmentována na více serverů, je tento popis čistě logický a propojení přes tato pole se řeší čistě na straně skriptu serveru. Definice cizích klíčů nejsou přímo v databázi.)

5.3.2.2.1 Databáze MAIN

Tabulka 1 - Databáze - Tabulka "members"

Název	Typ	Vlastnosti
memberID	int(11)	PK, AUTO_INCREMENT
username	varchar(255)	
password	varchar(255)	
email	varchar(255)	
active	varchar(255)	
resetToken	varchar(255)	
resetComplete	varchar(3)	
isAdmin	tinyint(1)	
accClosed	tinyint(1)	

Běžná tabulka pro uchovávání dat o užitelích. Kromě přihlašovacích údajů obsahuje i pole pro stavy účtu a pole pro realizaci změny hesla. Tato tabulka je relevantní pro uzel poskytující autentifikační služby.

Tabulka 2 - Databáze - Tabulka "devices"

Název	Typ	Vlastnosti
id	bigint(50)	PK, AUTO_INCREMENT
userID	bigint(50)	CK:members.memberID
deviceUUID	text	
privateKey	text	
nextPublicKey	text	
nextRequestKey	text	
active	tinyint(1)	

Tabulka ukládající informace o jednotlivých zařízeních každého uživatele. Kromě identifikátoru zařízení, jeho stavu a příslušnosti k uživateli se ukládá i sada klíčů pro zabezpečení požadavků. Více o problematice ověřování identity bude vysvětleno v samostatné části. Tato tabulka je relevantní pro uzel poskytující autentifikační služby.

Tabulka 3 - Databáze - Tabulka "writeAccess"

Název	Typ	Vlastnosti
id	bigint(50)	PK, AUTO_INCREMENT
userID	bigint(50)	CK:members.memberID
deviceUUID	text	CK: devices.deviceUUID
writeKey	text	

Tabulka ukládající klíč pro zápis. Tato tabulka je relevantní pro uzel držící data. Protože datový uzel nemá vlastní autentifikaci, dostane od autentifikačního serveru po zabezpečeném kanále klíč, vůči kterému se následně ověří zařízení.

Toto řešení však není ideální, protože se při každém zápisu do databáze kontaktuje autentifikační server a mohlo by dojít k jeho zahlcení. Do budoucna tak bude potřeba vymyslet nový postup při ověřování vůči jednotlivým uzlům, které omezí počet přístupů na hlavní autentifikační server.

5.3.2.2 Databáze POINTS

Databáze POINTS obsahuje větší množství tabulek, všechny ovšem pomocí identického předpisu. Struktura této databáze vychází z problematiky samotné mapy. Pozice na mapě se udává v zeměpisné šířce a zeměpisné délce. Zeměpisná šířka nabývá hodnot od

-90 do 90 a zeměpisná délka od -180 do 180 (21). Časem v aplikaci může být velké množství bodů a ukládání bodů do jedné tabulky je nevhodné minimálně ze dvou důvodů:

- 1) Vznikne velká tabulka, kterou bude stále obtížnější číst s rozumnou měrou délky běhu dotazu. Při specifickém dotazu, ať již výběr dat nebo třeba úprava specifického záznamu, by server musel vyhodnotit podmínku oproti velmi vysokému počtu řádků
- 2) Ztratíme možnost rozdělení dat na více uzlů

Datovou strukturu jsem rozdělil na dva předpisy tabulek. V databázi jsou tabulky pro samotné body a pro záznamy k daným bodům. Tato data jsou pak rozdělena do kvadrantů 10x10, celkem je tedy v této databázi 1406 tabulek, 703 je vyhrazených pro body na mapě a 703 pro recenze k těmto bodům. Tabulky jsou pak ve tvaru například POINTGRID_10x50 pro body v kvadrantu a k ní tabulka se záznamy POINTGRID_RECORDS_10x50

Tabulka 4 - Databáze - Předpis tabulky „POINTGRID_YxZ“

Název	Typ	Vlastnosti
id	bigint(50)	PK, AUTO_INCREMENT
pointID	text	
lat	decimal(23,18)	
lng	decimal(23,18)	
type	text	
shortTerm	tinyint(1)	
expire	bigint(20)	

Tabulka ukládá pozici bodu, jeho typ (restaurace, obchod, vyhlídka atd.), metadata o časových vlastnostech bodu, se kterými pak server pracuje při rozhodování o aktuálnosti bodu a dále ukládáme druhý identifikátor bodu pointID. Standardní číselný sloupec id s vlastností AUTO INCREMENT nám zajistí unikátnost v rámci jedné tabulky bez nutnosti starat se o integritu hodnoty primárního klíče. Pole pointID je pak globálním unikátním identifikátorem bodu napříč celou datovou strukturou. Pokud by v budoucnu nastala potřeba datovou strukturu nějak upravovat (například zjemnit mřížku zeměpisné délky nebo šířky, nebo reindexovat záznamy v rámci primárního klíče v důsledku vysokého počtu smazaných bodů a tudíž velkých mezer mezi indexy jednotlivých bodů), zůstane zachována integrita

bodů samotného, tedy k bodu zůstanou navázány relevantní záznamy hodnocení, protože zde existuje vazba přes právě globální unikátní hodnotu pointID.

Tabulka 5 - Databáze - Předpis tabulky „POINTGRID_RECORDS_YxZ“

Název	Typ	Vlastnosti
id	bigint(50)	PK, AUTO_INCREMENT
pointID	text	CK, POINTGRID_YxX.pointID
recordID	text	
parentRecordID	text	CK: this.recordID
userID	bigint(50)	CK, MAIN.members.memberID
editOpened	tinyint(1)	
name	text	
text	text	
photosJSON	longtext	
created	bigint(50)	
recommend	smallint(2)	

Zde ukládáme jednotlivé recenze bodů. Ukládáme identifikátor bodu, ke kterému recenze patří, ukládáme globální identifikátor samotné recenze, ukládáme rodičovskou recenzi (hlavní recenze bodu a další uživatelské recenze mají stejnou datovou strukturu, tudíž není potřeba vytvářet jinou tabulku, stačí hlídat, která je skutečně hlavní – nemá parentRecordID pro daný bod, a které jsou podřazené), dále pak také kdo recenzi vložil, jestli je záznam otevřen pro editaci, kdy záznam vznikl a poté samotná data recenze, tedy název recenze, text recenze, fotografie ukládané v base64 a samotné doporučení. V této tabulce dále v průběhu vývoje vzniknou sloupce s daty pro jednotlivé „widgety“ recenze (otevírací hodiny apod.).

Pozn.: V průběhu vývoje se ukázalo, že takto uložené fotografie nejspíše nebudou nejhodnějším řešením a s největší pravděpodobností vznikne ještě čtvrtá databáze vyčleněná pouze fotografiím. Důvodem je nevhodná délka textového řetězce ukládaná do jednoho pole a také manipulace s konkrétní fotografií. Ačkoliv uživatelům nebude zpětně umožněno k existující recenzi přidávat další fotografie, problematická by mohla být například akce smazání jedné nevhodné fotografie administrátorem, což by znamenalo načtení jednoho několika desítek MB dlouhého textového řetězce, jeho rozdělení na objekty, vyhledání v objektu, smazání části objektu, jeho převedení zpět na textový řetězec

a jeho opětovné vložení do databáze. Další akcí, která se současným návrhem není možná, je například omezení načtení fotografií, například, pokud by bylo implementováno stránkování fotografií nebo tzv. lazyLoad, který fotografie načítá postupně, aby uživatel měl přístup k již staženým, zatímco se na pozadí stahují další fotografie. Zde je nutné nahrát všechny fotografie do zařízení ještě předtím, než budou uživateli zobrazeny.

5.3.2.2.3 Databáze VOTES

V celém systému předpokládám nejvíce hlasů. Hlasy v tomto případě nejsou hlasy hodnocení místa, ale hlasy relevantnosti recenze místa (tedy pro hlavní recenzi je to hlasování, zdali je bod stále aktuální, zdali je třeba ho upravit, nebo zdali by bod měl být smazán, pro podřazené recenze je to pak hlasování o relevantnosti dané recenze).

MySQL databáze dle dokumentace neomezuje počet tabulek v databázi (22), jediným omezením je systém, na kterém databáze běží, respektive přidělený diskový prostor. Tato vlastnost je klíčová, protože umožní ukládat data pro každou recenzi zvlášť. Pro každý bod tedy bude vytvořena nová tabulka, která v jednoduchém formátu ukládá všechny hlasy k danému záznamu. To při vyšším počtu existujících bodů v databázi umožní přímý přístup k relevantním datům přes název tabulky (SQL dotaz bude poskládán tak, aby se dotázal na konkrétní tabulku recenze).

Tento koncept jsem vytvořil na následující myšlence: Mějme v jednom kvadrantu například 50 000 bodů (toto číslo je reálné, a pravděpodobně i tak je tento odhad nízký - ačkoliv velikost kvadrantu se mění s jeho pozicí směrem od rovníků k pólům, pro mírný podnebný pás je velikost kvadrantu přibližně 1111km * 700km, což zejména v hustě osídlených oblastech znamená vysoký počet bodů na kvadrant). Každý bod v kvadrantu bude mít v průměru například 10 recenzí (včetně recenze hlavní) a každá recenze 20 hlasů od náhodných kolemjdoucích. Pokud bychom drželi záznamy o hlasech stejně jako v databázi POINTS, měla by jen tabulka pro tento kvadrant 10 000 000 záznamů.

A v tomto okamžiku je nutné si uvědomit, jak často se tato data budou procházet. Pokud nevezmeme v úvahu cache na straně PHP serveru, tabulka se bude procházet velmi často, a to při zobrazení konkrétního záznamu, nebo dokonce mnohonásobně při zobrazení seznamu podřazených recenzí k recenzi hlavní. A tento fakt je problém, protože procházení takového počtu záznamů trvá relativně dlouho, nehledě na to, že tabulku by neprocházel pouze jen jeden uživatel v jeden časový okamžik a velmi rychle by mohlo dojít k zahlcení

databázového serveru. Kromě uživatelů ale tabulku využívají také úlohy v cronu, které periodicky kontrolují stav hlasování bodů a označují body ke smazání nebo editaci.

Naproti tomu model „co recenze, to tabulka hlasů“ umožní procházet jen specifickou oblast dat. Klientská aplikace ví, z jakého kvadrantu hledá jaký bod nebo seznam bodů. A protože tabulky budou pojmenovávány ve formátu „KVADRANT_recordID“, bude velmi jednoduché na straně serveru poskládat sadu dotazů do databáze na konkrétní tabulky. Například příkaz pro výpis deseti podřazených recenzí a jejich hlasování bude v části získání součtu hlasování sice znamenat deset přístupů do databáze, ale každý dotaz projde jen několik desítek nebo stovek záznamů naproti jednomu přístupu, který by ale procházel oněch 10 000 000 záznamů, navíc s výrazně složitější podmínkou v dotazu, pokud by se data hlasování ukládala po kvadrantech.

Tabulka 6 - Databáze - předpis tabulky hlasů

Název	Typ	Vlastnosti
voteID	bigint(50)	PK, AUTO_INCREMENT
userID	bigint(50)	CK, MAIN.members.memberID
voteType	smallint(3)	
voteValue	smallint(2)	

V této tabulce kromě identifikátoru hlasu držíme informaci o tom, kdo hlas udělil, sloupce voteType a voteValue poté určují typ hlasu (v tomto případě typ určuje, jestli se jedná o hlasování aktuálnosti hlavní recenze a tím celého bodu, nebo jestli se jedná o relevantnost podřazené recenze) a jeho hodnotu. Tyto typy hlasu a hodnoty jsou numerické a vyhodnocují se skriptem. Výhodou je univerzálnost předpisu, který se může v budoucnu rozšířit například na hlasování o aktuálnosti fotografií u recenze.

Vhodnost tohoto řešení si můžeme ukázat na serverovém skriptu, který zprostředkovává data o jedné recenzi (ať již hlavní, nebo podřazené):

```
1. $selectTable = "POINTGRID_RECORDS_" . $_GET["lat"] . "x" . $_GET["lng"];
2. // Získáme data o recenzi
3. $stmt=$GLOBALS['dbPoints'] ->
  prepare("SELECT `pointID`, `recordID`, `parentRecordID`, `editOpened`, `name`,
  `text`, `created` FROM `" . $selectTable . "` WHERE `recordID`='" .
  $_GET["recordID"] . "'");
4. $stmt->execute();
5. $row = $stmt->fetch(PDO::FETCH_ASSOC);
6. // Získáme informace o svém hlasu (slouží k nastavení ovládacích prvků na obrazovce aplikace)
7. $stmtv = $GLOBALS['dbVotes'] ->
  prepare("SELECT `voteID`, `voteType`, `voteValue` FROM `".$_GET["lat"] . "x" .
  $_GET["lng"] . "_` . $_GET["recordID"] . "` WHERE `userID`='" . $_GET["userID"] .
  "'");
8. $stmtv->execute();
9. $rowv = $stmtv->fetch(PDO::FETCH_ASSOC);
10. $row["myVote"] = $rowv;
11. $recommend = array();
12. $voteValue = array();
13. // pokud žádáme o hlavní recenzi, musíme získat hodnoty jednotlivých recenzí (výborný - dobrý - špatný)
14. if (isset($_GET["main"]))
15. {
16.   $stmtc = $GLOBALS['dbPoints'] ->
  prepare("SELECT `recommend` FROM `" . $selectTable . "` WHERE `recordID`='" .
  $_GET["recordID"] . "' OR `parentRecordID`='" . $_GET["recordID"] . "'");
17.   $stmtc->execute();
18.   $rowc = $stmtc->fetchAll(PDO::FETCH_ASSOC);
19.   $recommend[0] = 0;
20.   $recommend[1] = 0;
21.   $recommend[2] = 0;
22.   for ($i = 0; $i < count($rowc); $i++)
23.   {
24.     $recommend[intval($rowc[$i]["recommend"])]++;
25.   }
26.   $row["recommend"] = $recommend;
27. }
28. // pokud recenze není hlavní ale podřazená, získáváme místo toho hlasy relevantnosti dané recenze
29. else
30. {
31.   $stmtc = $GLOBALS['dbVotes'] ->
  prepare("SELECT `voteValue` FROM `".$_GET["lat"] . "x" . $_GET["lng"] . "_` .
  $_GET["recordID"] . "` WHERE `voteType`=1");
32.   $stmtc->execute();
33.   $rowc = $stmtc->fetchAll(PDO::FETCH_ASSOC);
34.   $voteValue[0] = 0;
35.   $voteValue[1] = 0;
36.   for ($i = 0; $i < count($rowc); $i++)
37.   {
38.     $voteValue[intval($rowc[$i]["voteValue"])]++;
39.   }
40.   $row["voteValue"] = $voteValue;
41. }
42. // výsledek pošleme na výstup
43. echo json_encode($row);
```

Na řádce č. 1 získáme data o kvadrantu. Tato data zašle klient, který si je pamatuje z dat bodu, na obrazovce mapy. Následně je poskládán název tabulky, ke které budeme přistupovat.

Na řádcích 2 až 5 provedeme dotaz na samotná data recenze. Tato data jsou na klientovi přímo zobrazována, nebo slouží klientovi pro nastavení dalších ovládacích prvků nebo pro další komunikaci se serverem.

Na řádcích 7 až 10 se dotazujeme do tabulky hlasování dané recenze na svůj hlas. A zde se ukazuje výhoda řešení „co recenze, to tabulka“. Takto na základě vstupních parametrů skriptu můžeme zacílit na malou oblast dat, v tomto případě třeba jen pár desítek záznamů. Pokud bychom měli všechny hlasy všech recenzí v kvadrantu v jedné tabulce, bylo by nutné procházet výrazně větší množství dat.

Z tohoto hlediska je ale zajímavější část kódu na řádcích 29 až 41. Zde chceme vrátit hodnoty, jak který uživatel hlasoval o relevantnosti recenze. Výsledkem jsou dvě čísla, a to kolik uživatelů hlasovalo pro a kolik uživatelů hlasovalo proti. V tomto případě opět zacílíme na konkrétní tabulku, která reprezentuje přímo hlasování o recenzi, tedy rovnou vybíráme relevantní data. Pokud by se všechny hlasy recenzí ukládaly v rámci kvadrantů do jedné tabulky, vypadalo by to přibližně takto:

Tabulka 7 - Databáze - nevhodné ukládání hlasů

Název	Typ	Vlastnosti
id	bigint(50)	PK, AUTO_INCREMENT
pointID	text	CK, POINT_kvadrant.recordID
recordID	text	CK, POINT_RECORDS_kvadrant.recordID
userID	text	CK, MAIN.members.memberID
voteType	smallint(3)	
voteValue	smallint(2)	

Dotaz pro získání dat z takovéto tabulky by pak vypadal takto:

```
1. SELECT * FROM POINTGRID_YxZ_VOTES WHERE recordID = $_GET["recordID"];
```

Takto bychom procházeli nejen výrazně více záznamů, každý záznam navíc ještě porovnáváme s parametrem, což celý dotaz výrazně zpomalí.

5.3.2.3 Zabezpečení

Z hlediska zabezpečení jsem řešil dva problémy. Jedním je zabezpečení dat uživatelů, druhým problémem pak je zabezpečení samotných přenosů, tedy v tomto případě zabránit zápisu nevhodných dat na server.

5.3.2.3.1 Zabezpečení dat uživatelů

Problém ochrany osobních údajů je v této aplikaci minimální. Jediným skutečně choulostivým záznamem je heslo uživatele, které je zašifrováno šifrou SHA-256. Druhým, potenciálně choulostivým údajem je e-mailová adresa. E-mailová adresa uživatele však za žádných okolností není promítána do cizích zařízení, veškeré recenze v aplikaci jsou pro ostatní uživatele anonymní. Jediná stránka, ze které by mohl za určitých okolností hrozit únik dat je tak administrace účtů, což je jediné místo, kde se adresy uživatelů zobrazí najednou.

5.3.2.3.2 Zabezpečení komunikace

Zabezpečení komunikace představuje v tomto projektu větší výzvu. Problém je, že většina hostingů, které jsou zdarma, neposkytují možnost HTTPS a příslušného SSL certifikátu. Veškerá komunikace se serverem je tak v nešifrované podobě a hrozí například odcizení přihlašovacích údajů, pokud se přihlásíme na nezabezpečené síti, kterou někdo odposlouchává.

Existuje sice možnost, jak SSL certifikát získat i bez podpory poskytovatele (23), mně se však zatím nepodařilo tuto část realizovat (služby byly ve skutečnosti buď jen „Trial“, tedy časově omezené k vyzkoušení, nebo nějakým způsobem selhalo ověření dané domény).

Tento problém se mi podařilo částečně vyřešit, kdy jediným zranitelným bodem je registrace, první přihlášení a první autorizace zařízení. Veškerá následující komunikace je již zabezpečena pomocí sady klíčů pro zápis dat do systému.

Zde je nutné říci, které akce jsou skutečně zabezpečené, a které nikoliv:

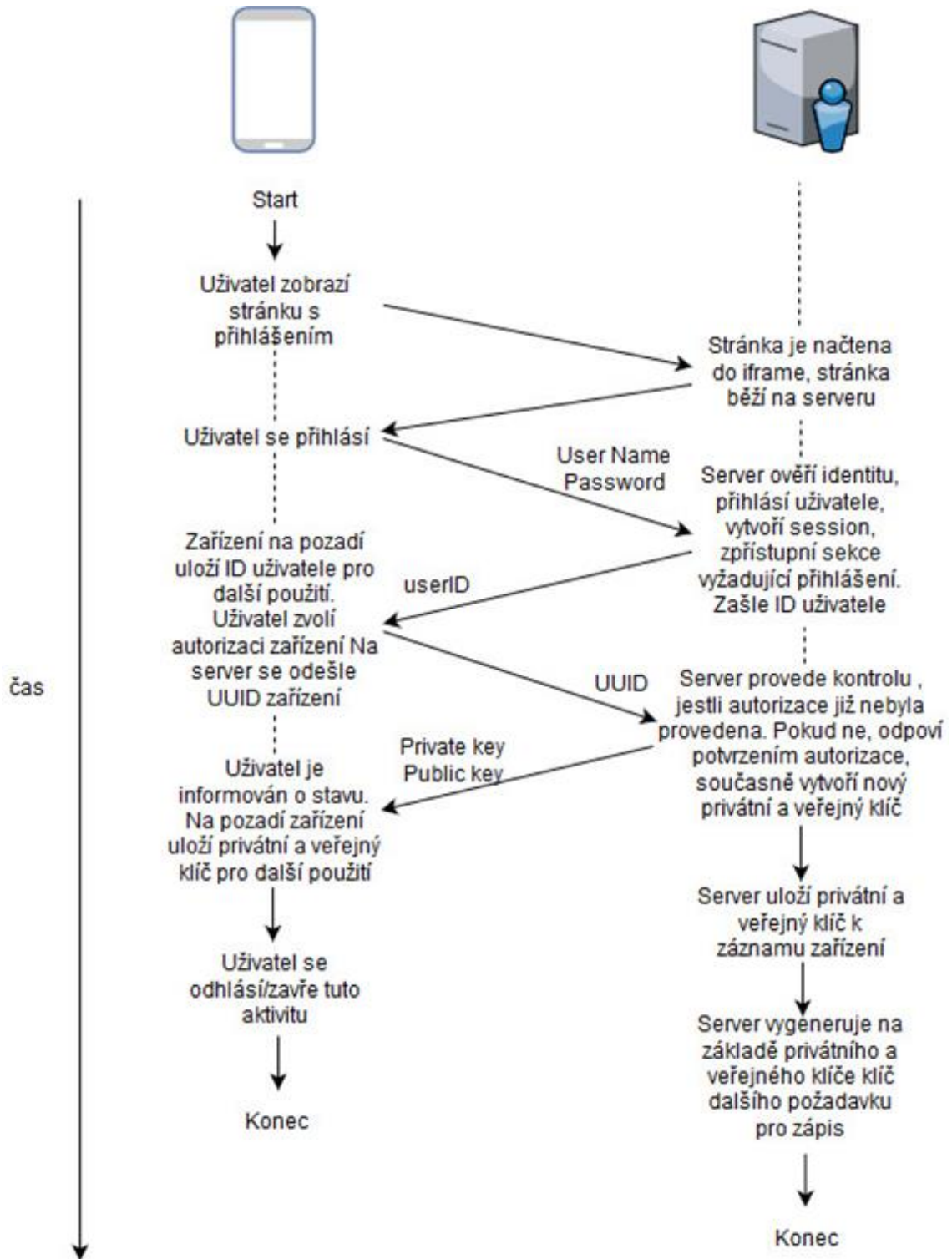
- 1) Přihlášení do aplikace a autorizace zařízení – tato akce využívá webový přístup (i v zařízení pomocí „iframe“, pro uživatele se však obrazovka jeví jako jakákoliv jiná část aplikace). Ověření probíhá pomocí jména a hesla, následně je vytvořena session pro práci s částmi aplikace pro ověřené uživatele – jedná se o nastavení

účtu a správu zařízení uživatele. I přes to, že je uživatel ověřován, tato komunikace zatím není šifrována (předpoklad ale je, že autentifikační část aplikace bude provozována na placeném hostingu, který bude podporovat HTTPS). Prozatím se uživateli bude zobrazovat při přihlášení nebo registraci hlášení, že by tuto akci měl provádět jen na důvěryhodných připojeních k internetu.

- 2) Požadavky na čtení dat – tyto požadavky nejsou zabezpečeny vůbec. Není třeba jakkoliv ověřovat uživatele, pokud chce data zobrazit, nejedná se o jakkoliv nebezpečnou akci. Jediným potencionálním rizikem je zahlcení serveru skrze čtení záznamu (jedná se o řadu dotazů do databáze, příliš rychlých dotazů by teoreticky mohlo zahltit skript nebo databázový server). To ovšem jde řešit na straně serveru, kdy skript bude kontrolovat, jak často se záznam volá a pokud proběhne příliš volání v určitém časovém intervalu, čtení záznamu se na daný časový interval vypne.
- 3) Zápis dat – tato část je velmi důležitá. Je třeba zajistit, aby zápis do systému mohli provádět jen registrovaní uživatelé a aby se do systému nedostal nikdo zvenčí (například voláním PHP souborů mimo aplikaci – třeba ručně z webového prohlížeče).

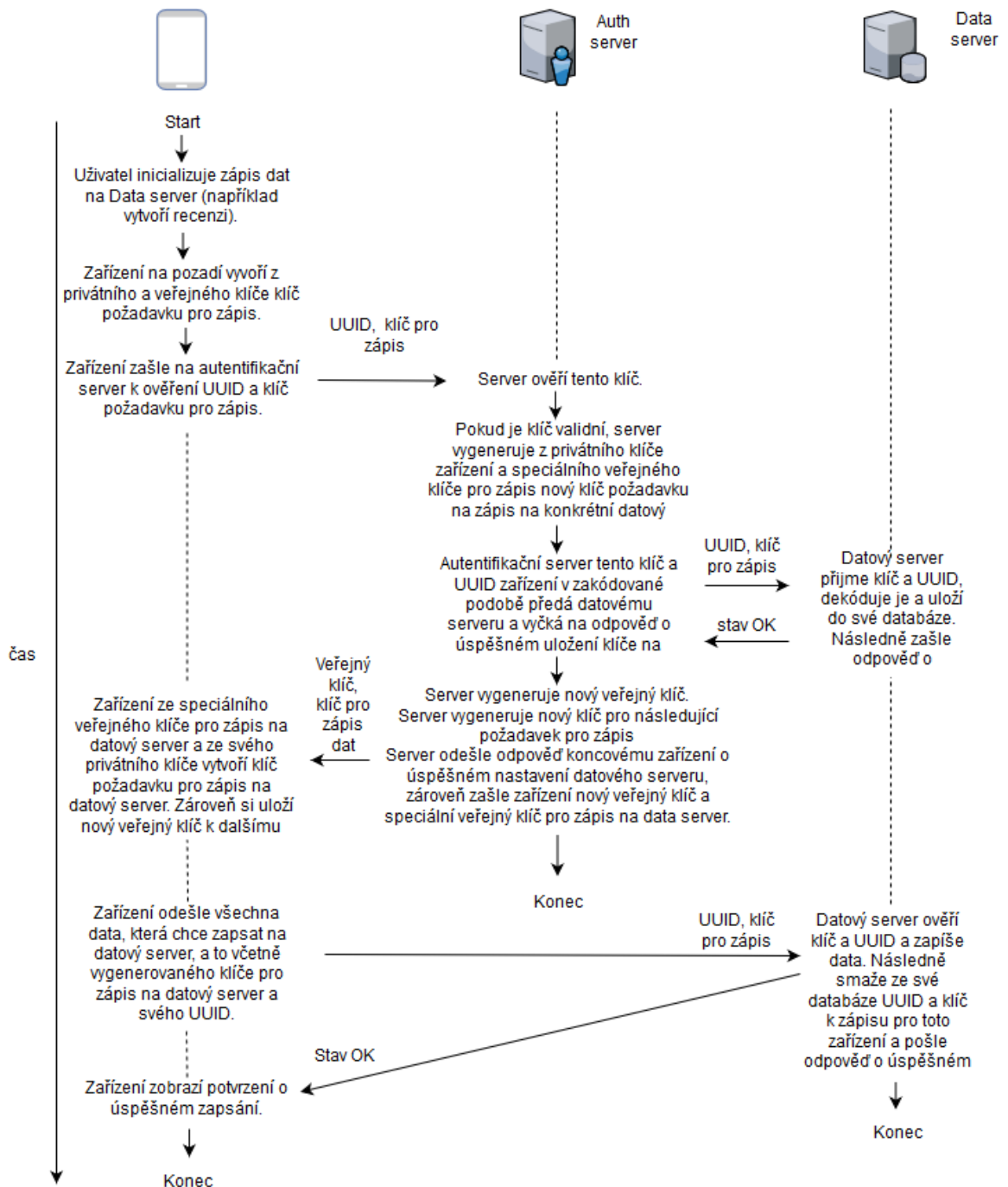
Pro zápis dat jsem vytvořil systém privátních a veřejných klíčů. Celý systém ověření uživatele a následné ověřování jednotlivých požadavků na zápis dat ukazují následující diagramy:

První diagram znázorňuje postup při prvotním přihlášení do aplikace a autorizace zařízení. Na konci tohoto procesu má server UUID zařízení a zařízení má privátní a veřejný klíč. Privátní a veřejný klíč si k danému zařízení uloží také server.



Obrázek 15 - Proces přihlášení

Druhý diagram znázorňuje řešení ověření konkrétního zázpisu dat na server:



Obrázek 16 - Proces ověřování pomocí klíčů

Proč je tato metoda bezpečná, tedy za předpokladu, že nedojde k úniku dat při procesu autorizace zařízení?

Jednoduše proto, že se po celou další dobu používání aplikace používá výhradně model výměny klíčů. Nikde se neposílají žádná hesla, a to ani v zašifrované podobě, které by mohl útočník (ten co může komunikaci odposlouchávat) zachytit a následně použít k připojení k cizímu účtu.

Nyní k samotným klíčům. Vždy platí toto:

Veřejný klíč + Privátní klíč -> SHA256 -> autorizační klíč pro zápis.

Privátní klíč zná pouze zařízení a server, kromě prvního zaslání privátního klíče při autorizaci se nikdy neposílá.

Při zápisu na server zařízení vygeneruje výsledný klíč a pošle ho autentifikačnímu serveru. Ten v rámci tohoto jednoho požadavku, pakliže je klíč validní, požadavek schválí a okamžitě vygeneruje nový veřejný klíč, který zašle v odpovědi zařízení. Zároveň přepočítá nový výsledný klíč pro další požadavek zápisu, který mu v budoucnu zařízení na základě právě vygenerovaného veřejného klíče zašle.

Tím pádem, i když by útočník zachytil požadavek s výsledným klíčem a pokusí se klíč použít pro svůj požadavek, klíč již nebude v systému existovat (mezitím byl přepočítán) a server tento útočníkův požadavek zamítne. Stejně tak nelze zneužít zachycený veřejný klíč, útočník nemá k dispozici privátní klíč zařízení a tím pádem nemůže dopočítat výsledný klíč pro požadavek zápisu.

Na stejném principu funguje klíč pro zápis na datový server, místo veřejného klíče se pro výpočet klíče pro zápis použije veřejný klíč speciálně zasláný k tomuto účelu.

Zbývá vysvětlit poslední část schématu, a tou je přenos klíče z autentifikačního serveru na datový server. Zde se využije kódování pomocí algoritmu, který zakóduje klíč do číselné sekvence a na straně serveru se pomocí obdobného algoritmu tato sekvence opět dekóduje. Tento kódovací algoritmus je taktéž můj výtvar, avšak z bezpečnostních důvodů ho nelze dále komentovat vzhledem k veřejnosti práce.

Nyní může nastat otázka: Proč se tento kódovací algoritmus nedá použít i pro kódování veškeré ostatní komunikace?

Odpovědi jsou hned dvě:

Za první, můj algoritmus je neefektivní co do délky zakódovaného řetězce. Tedy ač je kód sám o sobě bezpečný, délka kódu může za jistých okolností být až několikrát delší než původní řetězec. Což je v praxi nevhodné, turista využívá roaming nebo nějaký předplacený balíček a potřebuje přenášet co nejméně dat.

A za druhé je to přístupnost algoritmu. Instalační balíček aplikace lze pomocí různých technologií otevřít. V něm se pak nacházejí v případě hybridní aplikace přímo soubory zpracovávané prohlížečem, který v aplikaci běží, tedy HTML, JS soubory a vše ostatní. Takto by se algoritmus dostal na veřejnost a v tu chvíli by opět útočnickovi stačilo odchytit komunikaci mezi dvěma servery a použít tento algoritmus a tím získat klíč pro zápis a eventuálně ho zneužít ještě před tím, než dojde k zápisu zařízením, které má být skutečně podle tohoto klíče autorizováno k zápisu. Naopak uvnitř PHP souborů tento kódovací algoritmus využít lze, protože samotné PHP soubory jsou uloženy na serveru a nikdo k jejich obsahu nemá přístup.

5.3.3 Mobilní aplikace

K vývoji mobilní aplikace byl využit Framework Cordova. Tento Framework umožňuje vytvářet hybridní aplikace, což znamená, že ačkoliv uživatelské prostředí je provozováno v integrovaném browseru (WebView), aplikace má přístup i k nativnímu API zařízení a tudíž má na rozdíl od čisté webové aplikace přístup například k notifikacím, souborovému systému apod.

Samotný vývoj aplikace probíhá prakticky stejně jako vývoj jakéhokoliv jiného webu. Vývojář pracuje se standardním HTML, Javascriptem a kaskádovými styly. Samozřejmostí je možnost přidat libovolné javascriptové knihovny nebo jiné komponenty, které lze přidat na jakýkoliv jiný web.

Celá aplikace je pak vyvíjena právě pomocí HTML, JS a CSS, kdy je na této úrovni skriptována jak podoba aplikace, tak její logika.

Ještě před vývojem samotné aplikace je však nutné importovat pluginy samotné Cordovy. Tyto pluginy se importují obvykle pomocí příkazového řádku ve složce projektu. Instalační proces tohoto pluginu plugin nejen stáhne, ale automaticky vloží do projektu k jeho okamžitému užití. Vývojář má poté k dispozici javascriptové API, skrze které daný

plugin používá, vývojář se tak nedostane ke zdrojovým kódům pluginu, který může být napsán v jiných programovacích jazycích, obvykle v jazyce nativním pro všechna podporovaná zařízení.

5.3.3.1 Zavedení projektu

Ačkoliv v dokumentaci Cordovy (24) je kompletní návod, pro lepší představu výše uvedeného popisu uvedu zkrácenou verzi instalace Cordovy a spuštění aplikace:

- 1) Vývojář musí stáhnout a nainstalovat aplikaci Node.JS, která mu umožní instalovat další aplikace z repozitářů pomocí příkazu „npm“.
- 2) Vývojář následně nainstaluje Framework Apache Cordova příkazem „npm install -g cordova“ na příkazovém řádku.
- 3) Následně již lze použít příkaz „cordova“, který pracuje přímo s projektem:
 - a. Příkaz `cordova create myApp com.example.myApp MyApp` vytvoří novou adresářovou strukturu s novým projektem.
 - b. Příkaz `cordova platform add/remove platforma` přidá do projektu platformu, pro kterou se budou moci zdrojové kódy kompilovat a pro které se budou nastavovat cordova pluginy.
 - c. Příkaz `cordova plugin add/remove pluginName` přidá nebo odebere Cordova plugin do/z projektu. Obvykle je pak nutné projekt nejen znovu sestavit (příkazem `cordova build`), v některých případech se setkáme i s potřebou odebrat a znova přidat určitou platformu.
 - d. Následně již můžeme upravovat soubory webu a aplikaci spouštět příkazy `cordova emulate platformName` pro spuštění v emulátoru, nebo `cordova run platformName` pro spuštění aplikace na připojeném fyzickém zařízení.

5.3.3.2 Omezení kompilace

V průběhu vývoje jsem se setkal s problémem ohledně kompilace. Vyšlo najevo, že společnost APPLE nemá k dispozici knihovny (SDK) pro jinou platformu než svojí. V praxi to tedy znamená, že nelze aplikaci zkompilovat na iOS, pokud vývojář nemá k dispozici zařízení s macOS. Jediná možnost, jak projekt zkompilovat na iOS bez vlastnictví dalšího zařízení APPLE je využití virtualizace, ovšem macOS není svobodný software a takovéto řešení je v rozporu s licencí operačního systému. Tuto problematiku řeší

PhoneGap coby nadstavba Cordovy, který umožňuje kompilaci aplikace v jeho cloudu, ovšem tato služba je zpoplatněna (19).

5.3.3.3 Omezení pluginů

Opatrný musí být vývojář při výběru pluginů pro Cordovu. Většina pluginů je Open Source a dají se najít nejen na stránkách Cordovy, ale i v různých GIT repositářích. Zde je důležité ohlídat aktuálnost pluginu a jeho údržbu. Operační systémy zařízení se mění a tyto pluginy komunikují přímo s API operačního systému. S příchodem novějších verzí pak aplikace může přestat fungovat, protože daný plugin jeho autor neaktualizoval.

Druhou věcí, která se musí ohlídat při zavádění pluginu, je podpora napříč platformami. Některé pluginy nemusí podporovat platformy, pro které vývojář potřebuje aplikaci zkompilovat.

5.3.3.4 Problémy při vývoji

Jak již bylo zmíněno, aplikace se vyvíjí jako běžná webová stránka a tudíž rámci samotného vývoje logiky aplikace jsem nenarazil téměř na nic neobvyklého. Problémy s vývojem mobilní aplikace se vyskytly především na úrovni pluginů nebo samotné Cordovy.

5.3.3.4.1 Podpora některých CSS vlastností

Při vývoji jsem se setkal s absencí podpory vlastnosti kaskádových stylů. V tomto případě se jedná o výpočet v rámci CSS, například „width: calc(100vw – 20px);“ pro Android verze 4.4.4 a starší. V zásadě absence tohoto příkazu znamená pouze manuální dopočítávání některých velikostí, příliš neomezuje funkcionalitu, spíše pohodlnost vývoje UI. Otázkou nicméně je, jaké další vlastnosti se ukáží jako nepodporované na některé z platforem.

5.3.3.4.2 Podpora gest

Tato část je v okamžiku odevzdání práce ještě ve fázi experimentů. Cordova sice má plugin zvaný „cordova-native-gestures“, nicméně tento plugin se zdá být neudržovaný, navíc se zdá, že funguje jen pro platformu Android (25).

Gesta se nyní snažím řešit pomocí javascriptové knihovny Hammer.js, která umožňuje aplikovat gesta jako je třeba „pinch“ (užívaný pro zoom – roztáhneme dva

prsty od sebe nebo k sobě). Problém je neplynulost obrazu tohoto řešení. Řešení gest tedy zůstává zatím otevřené a stále hledám vhodnější implementaci.

5.3.3.4.3 Povolení komunikace přes AJAX

Tento problém ukazuje, jak je důležité psát dokumentaci k software, který mají následně užívat další lidé.

Při běžném vývoji pro Android je nutné při potřebě připojení k internetu ošetřit dvě věci:

- 1) Config.xml – zde musí být zavedeny definice, které zařízení řeknou, že může používat například připojení k Wi-Fi, připojení k mobilním datům, přístup do souborového systému, atp.
- 2) Od určité verze Android SDK je taktéž navíc nutné přímo v kódu o toto povolení požádat přesně v okamžiku, když je potřeba (tedy dříve stačilo odsouhlasit přístupy při instalaci aplikace, tyto přístupy se braly z Config.xml, v současných aplikacích se uživatel setká s požadavkem pro specifické povolení daného přístupu až za běhu aplikace).

V Cordově je potřeba tyto přístupy přidat do souboru Config.xml pro danou platformu (Android). Zatím jsem se neseťkal s potřebou žádat o povolení přístupu v kódu, to za vývojáře řeší Cordova.

Ovšem Cordova místo toho implementuje další vrstvu oprávnění, které musí vývojář nadefinovat:

- 1) Explicitně povolit přístupy k webu, lze omezit různé platformy, různé protokoly, nebo přímo různé příkazy. Část konfiguračního souboru Cordova aplikace může vypadat například takto:

```

1. <access origin="*" />
2. <allow-intent href="http://*/*" />
3. <allow-intent href="https://*/*" />
4. <allow-intent href="tel:*" />
5. <allow-intent href="sms:*" />
6. <allow-intent href="mailto:*" />
7. <allow-intent href="geo:*" />
8. <allow-intent href="gap:*" />
9. <allow-navigation href="https://*/*" />
10. <allow-navigation href="data:*" />
11. <platform name="android">
12.     <allow-intent href="market:*" />
13. </platform>

```

- 2) Dále je potřeba ještě povolit samotné spuštění některých příkazů, mezi něž patří i AJAX. Do každého HTML souboru, kde chceme AJAX použít, musíme vložit speciální META tag, který to umožní.

Problém je, že tento META tag má mnoho parametrů a zároveň řeší jak skripty, tak i „cizí“ CSS soubory. V dokumentaci Cordovy je vše řádně vysvětleno (26), k dispozici jsou i vzorové META tagy, které jsou okomentované, co všechno umožňují. Problém ovšem je, že se s novou verzí Cordovy (já používám verzi 8.0.0) něco změnilo a žádný tento META tag nefungoval. Ani řešení, která radila komunita (převážně na stackoverflow.com), nefungovala. Aplikace s různými META tagy vracela vždy jednu z těchto tří chyb:

- Nelze najít cílový soubor (myšleno PHP soubor, který voláme)
- Nelze se spojit s cílovým souborem, protože je to v rozporu s oprávněním (zadaným META tagem)
- Nelze najít file:///aes.js – jedná se o nějaký systémový javascript, který vývojář pravděpodobně neřeší na úrovni webové aplikace

Toto jsou některé doporučené META tagy z dokumentace Cordovy:

```

1. <!-- This policy allows everything (eg CSS, AJAX, object, frame, media, etc) except that
2.     * CSS only from the same origin and inline styles,
3.     * scripts only from the same origin and inline styles, and eval()
4. -->
5. <meta http-equiv="Content-Security-Policy" content="default-src *; style-src 'self' 'unsafe-inline'; script-
   src 'self' 'unsafe-inline' 'unsafe-eval'">
6.
7. <!-- Allows XHRs only over HTTPS on the same domain. -->
8. <meta http-equiv="Content-Security-Policy" content="default-src 'self' https:">
9.
10. <!-- Allow iframe to https://cordova.apache.org/ -->
11. <meta http-equiv="Content-Security-Policy" content="default-src 'self'; frame-
   src 'self' https://cordova.apache.org">

```

Problém jsem vyřešil náhodou při zkoušení velkého množství dalších variant META tagů. Klíčem je odstranit parametr “self” z požadovaných míst v META tagu. Zpětně jsem procházel dokumentaci Cordovy, zdali jsem tuto poznámku nepřehlédl a skutečně toto řešení k datu 23.3.2018 chybí.

V mém případě funkční META tag je tedy:

```
<meta http-equiv="Content-Security-Policy" content="default-src 'unsafe-inline' 'unsafe-eval' *">
```

5.3.3.4.4 Omezení a řešení mapového pluginu

Pro tuto aplikaci využívám plugin „cordova-plugin-googlemaps“. Jedná se o plugin, který pokryje všechny potřeby aplikace.

Výhodou je i jeho rychlost, využívá totiž hardwarové akcelerace. Zajímavá je jeho funkčnost na obrazovce, kdy, ač je v kódu plugin definován jako blok (div), dle autora (27) fyzicky běží pod prohlížečem jako nativní komponenta zařízení. Browser pak kontroluje, jestli uživatel kliká na nějaký prvek HTML (například tlačítko), nebo na prázdné místo. Pokud je dotek/klik detekován na prázdném místě, informace o doteku se převedou na komponentu map. Obecně mohu říci, že běh komponenty map je v zařízení naprosto plynulý a vyhovující.

I zde se však vyskytly určité problémy.

Za prvé, tento plugin naneštěstí nepodporuje běh v prohlížeči, ani na dalších platformách kromě Androidu a iOS (pokud se aplikace „nakompiluje“ pro běžný prohlížeč v počítači, mapový plugin nebude fungovat – nezobrazí se). Mojí myšlenkou bylo časem vytvořit i verzi pro běžný prohlížeč, tato disfunkce pluginu však znemožní plné znovuvyužití kódu z mobilních zařízení.

Druhým problémem je disfunkce jedné akce (eventu) mapy. Mapa by měla detekovat svůj vlastní posun, a pokud ho detekuje, měla by spustit kód, který je k této akci přiřazen. Tento problém se řeší na GIT repositáři pluginu, avšak zatím tato akce nebyla opravena. Musel jsem tedy zvolit náhradní řešení, kdy se po načtení mapy spouští časovač s předem stanoveným intervalem, který kontroluje, zda byla mapa posunuta a pokud ano, spustí další kód. Řešení je funkční, přesto není ideální. Ačkoliv sledování zatížení procesoru při běhu

tohoto časovače neprokázalo žádnou viditelnou zátěž, stále se jedná o neustále běžící kód, který by potenciálně mohl mít vliv na výdrž baterie telefonu při delším použití aplikace.

Třetím problémem je disfunkce GPS. Ačkoliv API tohoto pluginu umožňuje získat aktuální polohu uživatele, zatím se mi nepovedlo tuto funkcionalitu aktivovat. V tomto případě bude nutné najít jiný plugin, který aktuální polohu uživatele získá a následně již nebude problém vytvořit „marker“ v mapě s touto polohou.

Posledním, velmi zajímavým problémem, se kterým jsem se setkal, je přístup k objektu „markeru“. Problém nastal, když jsem potřeboval kódově změnit pozici markeru v mapě. Když se marker do mapy vytvoří, není dále přímo přístupný dalším voláním, pro další volání je nutné ho zároveň uložit do proměnné. Poté by měl být přístupný jako jakýkoliv jiný objekt, jeho vlastnosti by pak měly být přístupné přes tečkovou notaci. Na první pohled zajímavý je ovšem podobjekt, který drží data markeru, tímto podobjektem je `marker.Symbol(vars)`. Překvapivě do tohoto objektu nejde zapisovat (přiřadit hodnotu konkrétnímu parametru uvnitř `Symbol(vars)`), pokus skončí chybou a skript spadne. Původně jsem myslel, že problém je v názvu, kde užití závorek je nevhodné, při tečkové notaci se závorky interpretují jako funkce s parametrem. Ovšem ani pokus o přístup přes `marker[“Symbol(vars)“].parametr` (tento přístup je ekvivalentní tečkové notaci) nebyl úspěšný. Při testování mimo projekt Cordovy a objektu markeru se ukázalo, že samotný zápis problém není, je to pravděpodobně problém samotného objektu markeru. K zápisu dat, nebo jejich čtení je tedy nutné využít API markeru, nelze k tomuto objektu přistupovat přímo.

5.3.3.4.5 Debugging

Debuggování, pokud ho porovnáme s debuggováním například nativní aplikace napsanou v Javě v Android Studiu, mi přijde celkově pohodlnější. I zde však vidím prostor pro vylepšení.

Debuggování probíhá přes konzoli prohlížeče Google Chrome v počítači. Ve volbách konzole lze najít položku „Remote devices“, otevře se nové okno konzole, které se napojí na WebView spuštěné aplikace a lze pracovat s konzolí stejně jako při debuggování jakéhokoliv webu.

Výhodou je, že se pracuje přímo s HTML, CSS a JS soubory, takže lze v reálném čase měnit vizuální elementy, nebo i JS kód, změny se projevují jak na obrazovce zařízení, tak na virtuální obrazovce v konzoli. Další výhodou je volání funkcí z konzole, tedy lze v různých okamžicích zavolat funkci například pro výpis nějaké proměnné, aniž bychom proměnnou museli složitě volat na výstup konzole v kódu na různých místech, nebo nastavovat break points.

Naproti tomu zásadní nevýhodu vidím v plynulosti práce mezi jednotlivými sestaveními aplikace. Pokud upravíme kód aplikace a necháme aplikaci znovu sestavit a pošleme jí na zařízení, aplikace se restartuje a Google Chrome ztratí s aplikací kontakt. Okno konzole sice nabídne možnost se znovu připojit, avšak tato funkce nefunguje, načte se jen prázdné konzolové okno. Je třeba po každém tomto kroku zavřít okno debugovací konzole a znovu vybrat zařízení k debugování. I když tento problém neomezuje vývojáře co do funkčnosti, je práce s tímto nástrojem v tomto pohledu značně nepohodlná.

Druhý, minoritní problém se projevuje při práci z konzolí, kdy se občas stane, že se část obrazu na zařízení a na virtuální obrazovce konzole rozostří. Toto je problém čistě debugování a chyba se neprojevuje při běžném provozu aplikace. Jediná nepříjemnost je, že dochází k nečitelnosti obrazovky a je tedy nutné konzoli a aplikaci restartovat.

6 Výsledky a diskuse

Cílem této práce bylo navrhnout koncept aplikace, která umožní uživatelům sdílet doporučení míst v okolí. Aplikace byla realizována do fáze testovací verze.

Původně jsem předpokládal vyšší míru dokončení samotné aplikace co do škály funkcionalit, které uživatelům poskytne. V průběhu vývoje se však ukázalo, že některé body bude třeba řešit komplexněji, a to zejména v oblasti serverové části, tedy návrhu struktury databáze a zabezpečení přenosů dat. Vývoj také velmi zpomalilo řešení některých omezení platformy Cordova, nejvíce času však zabralo hledání řešení ohledně bezpečnostních META tagů, kdy jsem správné řešení objevil až po delší době, kdy jsem byl uprostřed vývoje náhradního řešení, které by taktéž (i když s jistými omezeními) umožnilo komunikaci se serverem bez použití technologie AJAX. Aplikace je tedy stále ve fázi vývoje a její dokončení a spuštění si vyžádá další čas (můj osobní optimistický odhad je podzim 2018).

Důležité otázky nastávají také ohledně ceny, kde je téměř nevyhnutelné aplikaci spustit kompletně bez nákladů. Další vývoj ukáže, zdali budu schopen aplikaci vydat bez cizí pomoci, tzn. do aplikace zpočátku investovat a doufat v její úspěch a návratnost investic, nebo jestli budu muset najít sponzora, který by částečně aplikaci převzal do správy (takto se nabízí právě společnost Seznam, která má dostatečné zázemí a která zařídí projekt, jenž byl pro tuto práci inspirací). Poslední eventualitou (ne však příliš pravděpodobnou) by bylo aplikaci dokončit do stavu prvního vydání a pokusit se celý projekt prodat společnosti, která bude schopná dalšího profesionálního vývoje aplikace.

V rámci této diplomové práce tedy byl stanoven koncept, je však nutný další vývoj pro nasazení aplikace do provozu a její úspěšné rozšíření do světa.

7 Závěr

Cílem této práce bylo vytvořit návrh a realizovat aplikaci pro hodnocení míst, konkrétně aplikace pro užití turisty. Aplikace tohoto druhu – tedy zaměřená právě na turisty je unikátní a do budoucna turistům pomůže při orientaci v cizím prostředí, navede turistu na zajímavá místa nebo doporučí místní věci k vyzkoušení. Zároveň by také měla omezit šanci turisty, že spadne do nějaké turistické pasti, stane se obětí podvodů zaměřených právě na turisty.

Byl realizován grafický návrh fungování aplikace, který využívá prvky ostatních mapových aplikací tak, aby použití bylo intuitivní. Ačkoliv je takto návrh použitelný, do budoucna budu zvažovat změnu grafiky aplikace, aby byla pro uživatele nejen použitelná, ale také hezčí na pohled.

Následně byl vybrán Framework pro realizaci, a to Framework Apache Cordova, který umožňuje tvorbu hybridních aplikací, které jdou kompilovat na různé platformy (kromě Androidu i na iOS, Windows Phone nebo jako běžná webová aplikace pro PC). Při práci s tímto frameworkem jsem narazil na řadu problémů, i tak bych ale Framework doporučil vývojářům, kteří se orientují na vývoj webu a nejsou tak dobře seznámeni s dalšími programovacími jazyky. Framework bych doporučil na tvorbu spíše jednodušších aplikací, které nebudou využívat příliš specializovaných funkcionalit vzhledem k potencionálním problémům s Cordova pluginy.

Realizace aplikace se ukázala jako problém na delší časové období. V tuto chvíli existuje koncept serverové části, který řeší provoz aplikace na více serverech kvůli dostupnosti a implementuje vlastní systém zabezpečení komunikace mezi zařízeními a serverovou částí.

Samotná aplikace je taktéž stále ve fázi vývoje, kdy prozatím existuje testovací verze, která neimplementuje všechny funkcionality. Moje vize uvedení do provozu je podzim 2018. V příloze jsou přidány vybrané obrazovky realizované aplikace. Tyto obrazovky jsou skutečnými snímky obrazovky z fyzického zařízení (Příloha 3 - Snímky obrazovky aplikace).

Dalším krokem bude pokus o sehnání sponzora/sponzorů a pokus o spuštění aplikace v profesionálním prostředí.

8 Seznam užitečných zdrojů

1. **Flutter GIT repository.** *GitHub*. [Online] [Citace: 25. 3. 2018.]
<https://github.com/flutter/flutter/releases?after=v0.0.10>.
2. **Flutter Technology FAQ.** *Flutter.io*. [Online] [Citace: 25. 3. 2018.]
<https://flutter.io/faq/#technology>.
3. **Flutter Performance profiling.** *Flutter.io*. [Online] [Citace: 25. 3. 2018.]
<https://flutter.io/ui-performance/>.
4. **Why Flutter doesn't use OEM widgets.** *Medium.com*. [Online] [Citace: 25. 3. 2018.]
<https://medium.com/flutter-io/why-flutter-doesnt-use-oem-widgets-94746e812510>.
5. **Flutter - Using Hot reload.** *Flutter.io*. [Online] [Citace: 25. 3. 2018.]
<https://flutter.io/hot-reload/>.
6. **What is Hybrid Mobile app.** *developer.telerik.com*. [Online] 18. 12. 2017.
[Citace: 25. 3. 2018.] <https://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>.
7. **Cordova Plugins.** *cordova.apache.com*. [Online] 7.x. [Citace: 25. 3. 2018.]
<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>.
8. **Cordova Platform Support.** *cordova.apache.com*. [Online] 7.x. [Citace: 25. 3. 2018.]
<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>.
9. **Cordova Platform Support.** *cordova.apache.com*. [Online] 5.4.0. [Citace: 25. 3. 2018.]
<https://cordova.apache.org/docs/en/5.4.0/guide/support/index.html>.
10. **Increase Your Site's Performance with Hardware-Accelerated CSS.**
blog.teamtreehouse.com. [Online] [Citace: 25. 3. 2018.]
<http://blog.teamtreehouse.com/increase-your-sites-performance-with-hardware-accelerated-css>.
11. **NativeScript FAQ.** *nativescript.org*. [Online] [Citace: 25. 3. 2018.]
<https://www.nativescript.org/faq>.
12. **NativeScript documentation: Chapter 2—Creating User Interfaces.**
docs.nativescript.org. [Online] [Citace: 25. 3. 2018.]
<https://docs.nativescript.org/tutorial/chapter-2>.
13. **NativeScript documentation: Custom components.** *docs.nativescript.org*. [Online]
[Citace: 25. 3. 2018.] <https://docs.nativescript.org/ui/basics#custom-components>.
14. **NativeScript documentation: User Interface Widgets.** *docs.nativescript.org*.
[Online] [Citace: 25. 3. 2018.] <https://docs.nativescript.org/ui/components>.
15. **Embarcadero Features – Design.** *embarcadero.com*. [Online] [Citace: 25. 3. 2018.]
<https://www.embarcadero.com/products/rad-studio/features/design>.
16. **Embarcadero product store – Delphi.** *embarcadero.com*. [Online]
[Citace: 26. 3. 2018.] <https://www.embarcadero.com/app-development-tools-store/delphi>.

17. **Xamarin documentation: Hello Android – Part 1: Quickstart.** *docs.microsoft.com*. [Online] [Citace: 25. 3. 2018.] <https://docs.microsoft.com/en-us/xamarin/android/get-started/hello-android/hello-android-quickstart?tabs=vswin>.
18. **Xamarin.Forms Quickstart.** *docs.microsoft.com*. [Online] [Citace: 25. 3. 2018.] <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/get-started/hello-xamarin-forms/quickstart?tabs=vswin>.
19. **Build PhoneGap.** *build.phonegap.com*. [Online] [Citace: 25. 3. 2018.] <https://build.phonegap.com/>.
20. **Are ads within average Android applications profitable?** *softwareengineering.stackexchange.com*. [Online] 29. 4. 2012. [Citace: 25. 3. 2018.] <https://softwareengineering.stackexchange.com/questions/96911/are-ads-within-average-android-applications-profitable?answertab=active#tab-top>.
21. **Latitude and Longitude formats.** *geomidpoint.com*. [Online] [Citace: 25. 3. 2018.] <http://www.geomidpoint.com/latlon.html>.
22. **MySQL Database count limit.** *dev.mysql.com*. [Online] [Citace: 25. 3. 2018.] <https://dev.mysql.com/doc/refman/5.5/en/database-count-limit.html>.
23. **Free SSL Certificate.** *geekflare.com*. [Online] 9. 8. 2017. [Citace: 25. 3. 2018.] <https://geekflare.com/free-ssl-tls-certificate/>.
24. **Cordova documentation: Create your first app.** *cordova.apache.org*. [Online] 7.x. [Citace: 25. 3. 2018.] <https://cordova.apache.org/docs/en/latest/guide/cli/index.html>.
25. **NPM plugin repository: cordova-native-gestures.** *npmjs.com*. [Online] [Citace: 25. 3. 2018.] <https://www.npmjs.com/package/cordova-native-gestures>.
26. **Cordova documentation: Whitelisting Guide.** *cordova.apache.org*. [Online] 7.x. [Citace: 25. 3. 2018.] <https://cordova.apache.org/docs/en/latest/guide/appdev/whitelist/index.html>.
27. **Cordova-plugin-googlemaps GIT repository.** *github.com*. [Online] [Citace: 26. 3. 2018.] <https://github.com/mapsplugin/cordova-plugin-googlemaps#how-does-this-plugin-work>.

9 Přílohy

Příloha 1 - Příklad úpravy *ListView* - *Android studio*

```
1. ListAdapter adapter = new ArrayAdapter<Item.DialogItem>(this,android.R.  
   layout.select_dialog_item,android.R.id.text1,items){  
2.     public View getView(int position, View convertView, ViewGroup parent) {  
3.         //Use super class to create the View  
4.         View v = super.getView(position, convertView, parent);  
5.         TextView tv = (TextView)v.findViewById(android.R.id.text1);  
6.  
7.         //Put the image on the TextView  
8.         tv.setCompoundDrawablesWithIntrinsicBounds(items[position].icon, null,  
           null, null);  
9.  
10.  
11.         //Add margin between image and text (support various screen densities)  
12.         int dp5 = (int) (5 * getResources().getDisplayMetrics().density + 0.5f);  
  
13.         tv.setCompoundDrawablePadding(dp5);  
14.  
15.         if(position==0){tv.setTextColor(Color.rgb(142,83,0));}  
16.         if(position==1){tv.setTextColor(Color.rgb(255,0,0));}  
17.         if(position==2){tv.setTextColor(Color.rgb(0, 0, 220));}  
18.  
19.         return v;  
20.     }  
21. };
```

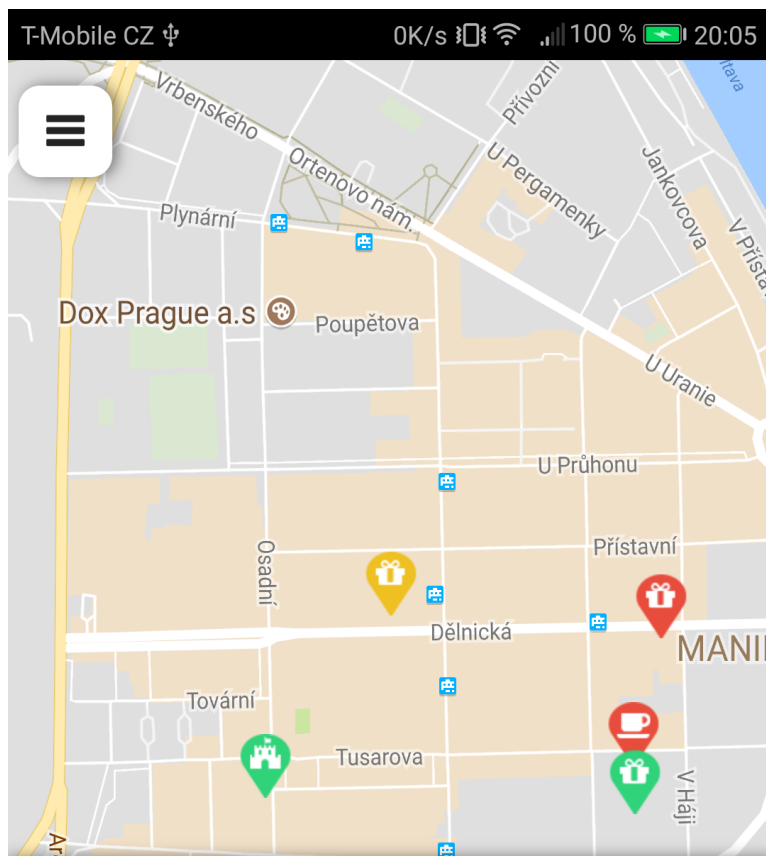
Příloha 2 - Flutter – příklad programování objektů UI

```
1. Widget build(BuildContext context) {
2.   return new DefaultTabController(
3.     length: demos.length,
4.     child: new Scaffold(
5.       appBar: new AppBar(
6.         title: new Text(title),
7.         actions: <Widget>[
8.           new Builder(
9.             builder: (BuildContext context) {
10.              return new IconButton(
11.                icon: const Icon(Icons.description),
12.                tooltip: 'Show example code',
13.                onPressed: () {
14.                  _showExampleCode(context);
15.                },
16.              );
17.            },
18.          ),
19.        ],
20.        bottom: new TabBar(
21.          isScrollable: true,
22.          tabs: demos.map((ComponentDemoTabData data) => new Tab(text:
23.            data.tabName)).toList(),
24.        ),
25.        body: new TabBarView(
26.          children: demos.map((ComponentDemoTabData demo) {
27.            return new Column(
28.              children: <Widget>[
29.                new Padding(
30.                  padding: const EdgeInsets.all(16.0),
31.                  child: new Text(demo.description,
32.                    style: Theme.of(context).textTheme.subhead
33.                  )
34.                ),
35.                new Expanded(child: demo.demoWidget)
36.              ],
37.            );
38.          }).toList(),
39.        ),
40.      ),
41.    );
42. }
```

Příloha 3 - Snímky obrazovky aplikace



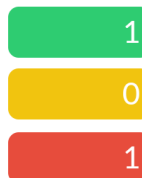
Obrázek 17 - Aplikace – mapa



Trap giftshop

Does not sell czech items at all.

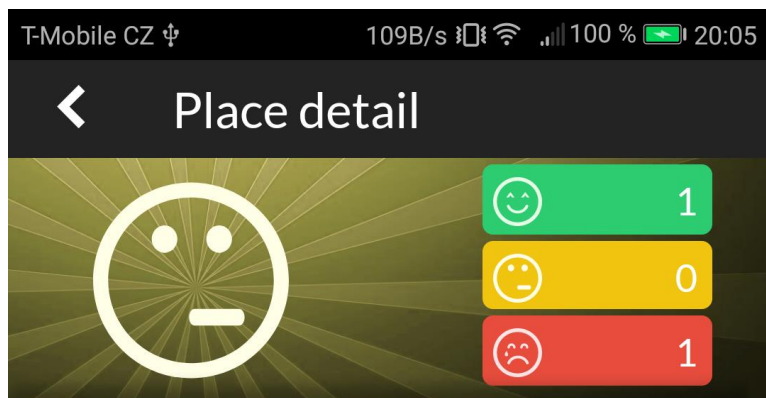
Everything is made in china. Also overprised.



[Detail](#)



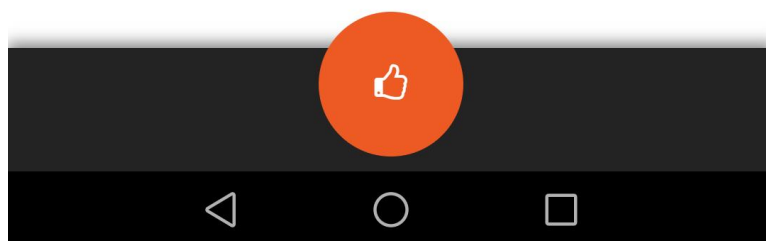
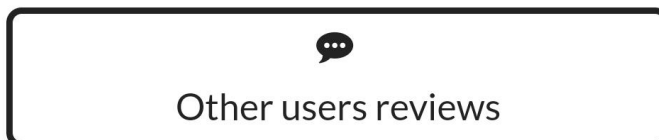
Obrázek 18 - Aplikace - rychlé zobrazení bodu



Trap giftshop

Does not sell czech items at all. Everything is made in china. Also overprised.

29 Mar 2018 17:33:32



Obrázek 19 - Aplikace - Detailní zobrazení bodu



Other reviews



Thu Mar 29 2018



1



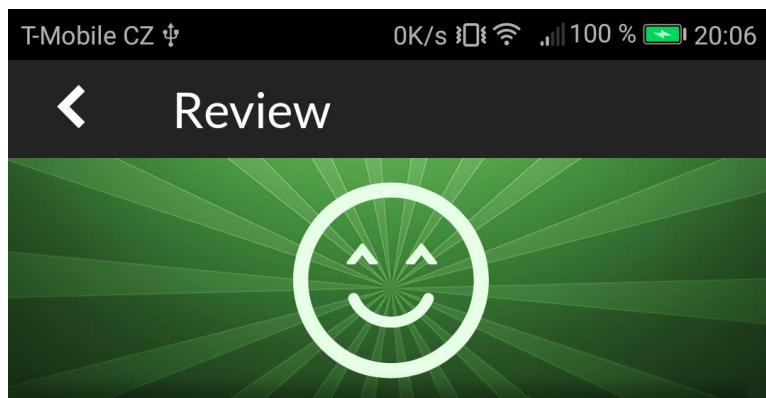
1

Good price, various stuff

Small giftshop, great atmosphere, also
the seller (...)



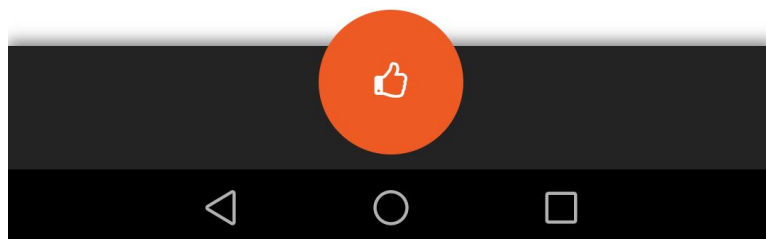
Obrázek 20 - Aplikace - podřazené recenze - seznam



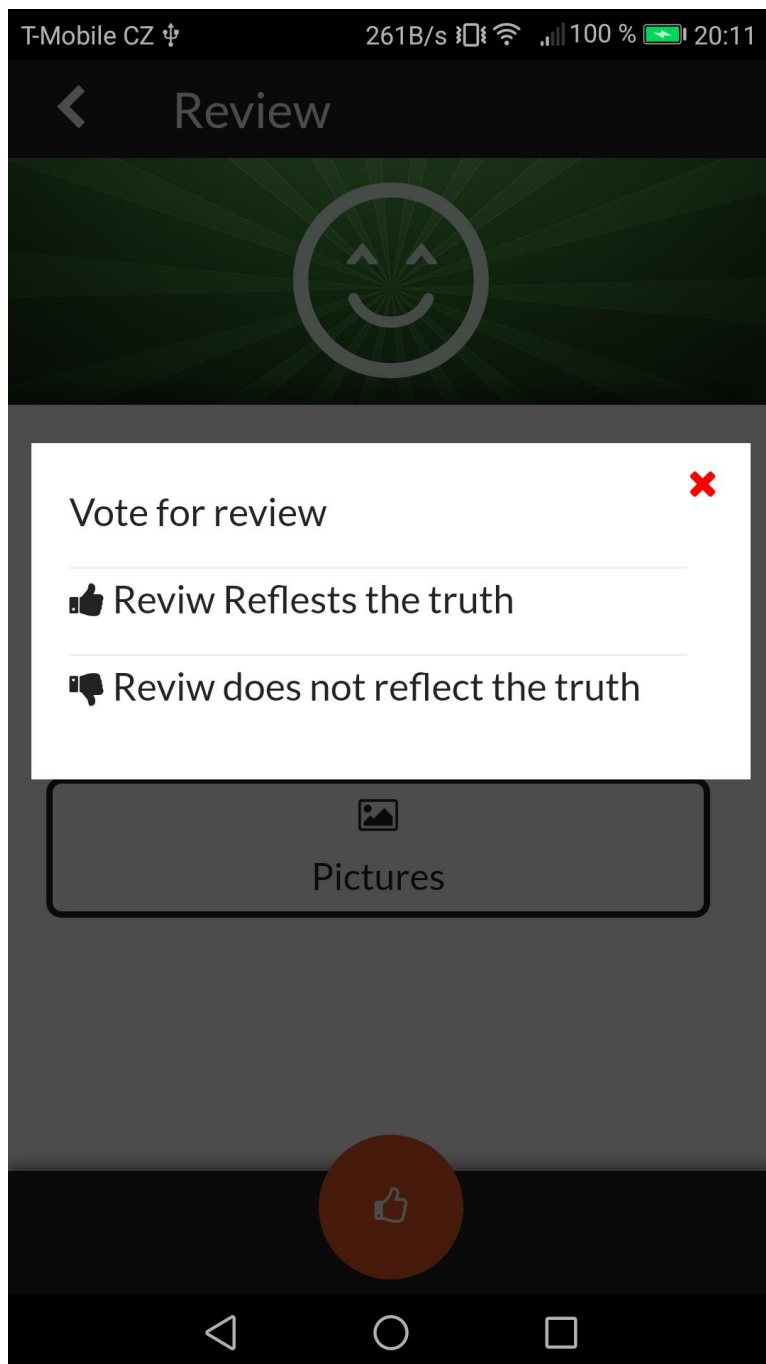
Good price, various stuff

Small giftshop, great atmosphere, also the seller was really kind and helped with selection of gift.

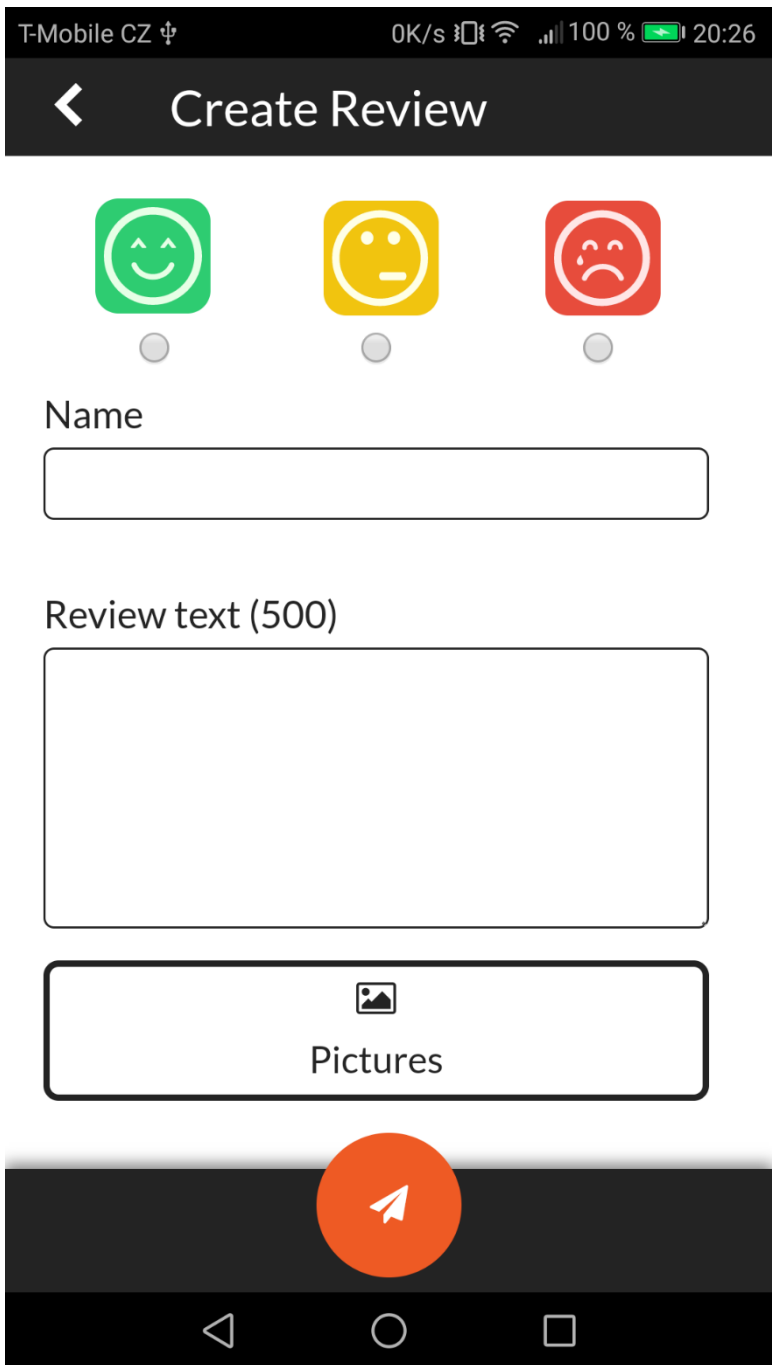
29 Mar 2018 17:34:13



Obrázek 21 - Aplikace - zobrazení podřazené recenze



Obrázek 22 - Aplikace - hlasování



Obrázek 23 - Aplikace - přidání recenze

← Images



Obrázek 24 - Aplikace - obrázky recenzí