



TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií ■

Mobilní aplikace WhatToDo in Liberec

Bakalářská práce

Studijní program: B2646 – Informační technologie
Studijní obor: 1802R007 – Informační technologie

Autor práce: **Michaela Zámečnicková**
Vedoucí práce: Ing. Igor Kopetschke





TECHNICAL UNIVERSITY OF LIBEREC
Faculty of Mechatronics, Informatics
and Interdisciplinary Studies ■

Mobile app WhatToDo in Liberec

Bachelor thesis

Study programme: B2646 – Information Technology
Study branch: 1802R007 – Information Technology

Author: **Michaela Zámečnicková**
Supervisor: Ing. Igor Kopetschke





Zadání bakalářské práce

Mobilní aplikace WhatToDo in Liberec

Jméno a příjmení: **Michaela Zámečnicková**
Osobní číslo: M16000215
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávací katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: **2018/2019**

Zásady pro vypracování:

1. Proveďte rešerši existujících mobilních aplikací zaměřených na informovanost návštěvníků města Liberce.
2. Navrhněte vlastní aplikaci integrující informace, které by dle Vašeho názoru měl mít k dispozici každý turista a návštěvník města.
3. Podílejte se na návrhu datového modelu pro aplikaci na platformě Google Firebase.
4. Implementujte navrženou aplikaci včetně vícejazyčného rozhraní, otestujte a zkuste získat zpětnou vazbu od vybrané skupiny uživatelů.

Rozsah grafických prací: dle potřeby
Rozsah pracovní zprávy: 30 – 40 stran
Forma zpracování práce: tištěná/elektronická



Seznam odborné literatury:

- [1] The definitive guide to firebase. New York, NY: Springer Science+Business Media, 2017. ISBN 978-1-4842-2942-2.
- [2] LACKO, Ľuboslav. Mistrovství – Android. Přeložil Martin HERODEK. Brno: Computer Press, 2017. Mistrovství. ISBN 978-80-251-4875-4.
- [3] HERODEK, Martin. 333 tipů a triků pro Android: [sbírka nejužitečnějších postupů a řešení]. Brno: Computer Press, 2014. ISBN 978-80-251-4310-0.

Vedoucí práce: Ing. Igor Kopetschke
Ústav nových technologií a aplikované informatiky

Datum zadání práce: 18. října 2018

Předpokládaný termín odevzdání: 30. dubna 2019

L. S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci 18. října 2018

Prohlášení

Byla jsem seznámena s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracovala samostatně s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Současně čestně prohlašuji, že texty tištěné verze práce a elektronické verze práce vložené do IS STAG se shodují.

13. 3. 2019

Michaela Zámečníková

Poděkování

Ráda bych poděkovala vedoucímu své bakalářské práce panu Ing. Igoru Kopetschkovi za cenné rady při tvorbě tohoto projektu. Bez těchto rad by práce nedosahovala takové kvality a obsahu. Dále chci poděkovat firmě MITEL s.r.o. za poskytnutí serverů a pomoci pro tento projekt. Nakonec chci poděkovat také své rodině a přátelovi za podporu během studia.

Abstrakt

Tato práce je zaměřena na vývoj mobilní aplikace pro město Liberec. Cílem aplikace je sloužit jako turistický průvodce a usnadnit uživateli pobyt v Liberci. Obsahem práce je seznámení se s projektem, návrh projektu a poté postup jeho vytvoření. Práce dále představuje zajímavé knihovny a objekty, používané při vývoji Android aplikací. Čtenář by měl mít po přečtení této práce představu, co všechno tvorba aplikace zahrnuje a být schopen jednoduchou aplikaci vytvořit.

Klíčová slova

Android, aplikace, Java, Google Firebase, Google Maps

Abstract

This thesis is focused on mobile app for the Liberec city and its development. Main purpose of this app is to simplify users stay in Liberec and to be a tourist guide. It contains introduction of project, design of project and developing process. Thesis also contains introduction of interesting libraries and objects used for Android development. Reader should have idea about what you have to do while developing app and should be able to create a simple app.

Key words

Android, application, Java, Google Firebase, Google Maps

Obsah

1. Úvod	12
2. Podobné aplikace	14
2.1. <i>Google Maps</i>	14
2.2. <i>Mapy.cz</i>	14
2.3. <i>InCity</i>	14
2.4. <i>DarumaGo!</i>	15
3. Návrh aplikace	16
3.1. <i>Přehlednost, intuitivnost</i>	16
3.2. <i>Vše na jednom místě, na míru</i>	16
4. Datový model	17
4.1. <i>Komunikace databáze – aplikace</i>	17
4.2. <i>Návrh databáze</i>	17
5. Struktura aplikace	17
5.1. <i>Layout</i>	18
5.2. <i>Aktivita</i>	18
5.3. <i>Android Manifest</i>	19
5.4. <i>Build Gradle</i>	19
6. Zobrazované stránky aplikace WhatToDo	20
6.1. <i>Splash Screen</i>	20
6.2. <i>Úvodní stránka</i>	21
6.3. <i>Seznam událostí</i>	22
6.4. <i>Detail události</i>	23
6.5. <i>Seznam míst</i>	25
6.6. <i>Detail místa</i>	26
6.7. <i>Galerie</i>	27
6.8. <i>Mapa</i>	28
6.9. <i>Navigace</i>	29
6.10. <i>Menu a filtry</i>	31
7. Pomocné třídy	32
7.1. <i>Place, Event, Offer</i>	32
7.2. <i>PlaceListAdapter a EventListAdapter</i>	32
7.3. <i>OnSwipeTouchListener</i>	32
8. Použité technologie	33

8.1.	<i>Java vs Kotlin</i>	33
8.2.	<i>Firebase</i>	33
8.2.1.	<i>Realtime Database</i>	34
8.2.2.	<i>Cloud Firestore</i>	34
8.2.3.	<i>Cloud Storage</i>	34
8.3.	<i>Google Maps</i>	34
8.3.1.	<i>Maps SDK</i>	34
8.3.2.	<i>KML</i>	35
8.4.	<i>Mapbox</i>	36
8.4.1.	<i>Maps</i>	36
8.4.2.	<i>Navigation</i>	36
8.5.	<i>Boom Menu</i>	37
8.6.	<i>Zajímavé použité objekty a třídy</i>	37
8.6.1.	<i>CountDownTimer</i>	37
8.6.2.	<i>Bundle</i>	38
8.6.3.	<i>Intent</i>	38
8.6.4.	<i>Toast</i>	38
8.6.5.	<i>BitmapFactory</i>	39
8.6.6.	<i>AsyncTask</i>	39
8.6.7.	<i>Location</i>	40
8.6.8.	<i>LocationManager</i>	41
8.7.	<i>Formáty obrázků</i>	42
8.7.1.	<i>Vektor</i>	42
8.7.2.	<i>PNG</i>	42
8.8.	<i>Formát textových řetězců</i>	43
9.	Testování u uživatelů	44
10.	Úskalí práce	46
10.1.	<i>Android studio</i>	46
10.2.	<i>Problémy s verzemi OS Android</i>	46
10.3.	<i>Licencování GoogleMaps</i>	46
11.	Možná vylepšení	48
11.1.	<i>Vytvoření plánu dne</i>	48
11.2.	<i>Rozšíření pro platformu IOS</i>	48
11.3.	<i>Rozšíření do dalších měst</i>	48
12.	Závěr	49
	Seznam použité literatury	51

Seznam obrázků

Obrázek 1 – Barvy	16
Obrázek 2 - Komunikace databáze - aplikace	17
Obrázek 4 - Splash Screen	20
Obrázek 5 – Úvodní stránka	21
Obrázek 6 – Kód: dialog pro povolení GPS	22
Obrázek 7 - Seznam událostí	23
Obrázek 8 - Detail události	24
Obrázek 9 - Detail události: rozklik	24
Obrázek 10 - Seznam míst	25
Obrázek 11 - Získávání vzdálenosti: kód	26
Obrázek 12 - Detail místa	27
Obrázek 13 - Detail místa: rozklik	27
Obrázek 14 - Galerie	28
Obrázek 15 - Mapa	29
Obrázek 16 - Navigace	30
Obrázek 17 - Navigace: kód	30
Obrázek 18 - Menu	31
Obrázek 19 - Filtr míst	31
Obrázek 20 - KML: kód	35
Obrázek 21 - CountdownTimer: kód	37
Obrázek 22 – AsyncTask + Bitmap: kód	40
Obrázek 23 - Vektorový obrázek: kód	42
Obrázek 24 - Vektorový obrázek: výsledek	42
Obrázek 25 - Editor pro vícejazyčnost	43
Obrázek 26 -Graf 1	44
Obrázek 27 - Graf 2	44
Obrázek 28 - Graf 3	45

1. Úvod

V dnešní moderní době se telefony stávají stále více součástí našeho každodenního života. Jelikož jsou chytré mobilní telefony, popřípadě tablety cenově velmi dostupné, jeden z nich vlastní téměř každý člověk a počet vlastníků stále stoupá. Je tedy celkem přirozené, že mobilní aplikace pomalu nahrazují spoustu starých informačních zdrojů, jako například papírové mapy nebo cestovatelské příručky.

Pro uživatele je jednodušší a pohodlnější mít veškeré informace na jednom místě, ve svém telefonu. Turista s tím například ušetří místo v batohu a vzdá se přebytečného závaží. Je také jednodušší najít potřebnou informaci v takovéto aplikaci, než hledat a listovat v brožurce, nebo se ptát kolemjdoucích. To může být nejen pro ostýchavého člověka velký problém.

Myšlenka na tento projekt vznikla během cestování po zajímavých místech u nás i v zahraničí. Sama jsem podobné aplikace využívala. Nevyhovovalo mi však, že nenajdu všechny informace na jednom místě. Vyznat se poté v množství stažených aplikací je zdlouhavé a nepraktické. Liberec aplikaci, která by vyhovovala tomuto požadavku, neměl. Napadlo mě tedy ji v rámci bakalářské práce vytvořit.

Jelikož jsem věděla, že firma MITEL s.r.o. plánuje vytvořit podobnou aplikaci pro město Mladá Boleslav, domluvila jsem se s nimi, že pro ně aplikaci vytvořím. Koncipovat ji budu tak, aby se dala snadno použít pro další města. Firma na oplátku nabídla pro aplikaci své servery, bez kterých by funkčnost aplikace nebyla možná. Dále nabídli, že aplikaci budou pravidelně plnit novými daty. Také zařídí získávání výhod pro její uživatele, jako soutěže o vstupenky na události a podobně.

Jelikož projekt je svým rozsahem náročný, požádala jsem svého spolužáka Radovana Čapka o spolupráci. Práci jsme rozvrhli tak, že já vytvořím aplikaci, společně navrhne databázi a on naprogramuje server, vytvoří administrační rozhraní pro přidávání dat a propagační web.

Motivací pro tento projekt bylo v první řadě usnadnit návštěvníkům a obyvatelům města jejich pobyt v Liberci. Liberec je krásné město, avšak oproti jiným, stejně zajímavým městům, je turisty méně vyhledáván. Jedním z důvodů může být i nedostatečná informovanost návštěvníků o možných turistických cílech. Většina lidí si pod pojmem

Liberec představí pouze Ještěd a nic jiného nezná. Tato aplikace má za cíl lidem město lépe představit a jejich návštěvu co nejvíce zpříjemnit.

Další motivací bylo udělat snadno ovladatelnou aplikaci, se kterou si poradí i méně zkušený uživatel. Většina podobných aplikací totiž není zcela intuitivní, je psaná malým písmem a někteří uživatelé proto mohou mít problém se v takovéto aplikaci vyznat. Dále bylo cílem udělat aplikaci přímo na míru tomuto městu. Zařadit mezi zajímavá místa i ta, která znají pouze obyvatelé Liberce. Například dávat tipy, kam si zajít na svátek svatého Valentýna na romantický piknik, nebo kde na Prvního máje najít místa s rozkvetlými třešněmi.

2. Podobné aplikace

2.1. Google Maps

Aplikace je velice rozšířená a dlouhodobě profesionálně vyvíjená, obsahově je velice podobná, avšak je tvořena více globálně. Aplikace WhatToDo je oproti tomu tvořena lokálně přímo pro Liberec, kde uživatel nalezne i místa, které znají pouze obyvatelé Liberce. Aplikaci Google Maps se taktéž spousta uživatelů vyhýbá, kvůli zdánlivé složitosti. Google Maps má spoustu funkcí a informací navíc, kterými WhatToDo nedisponuje, které jsou pro běžného návštěvníka zbytečné a přidávají na komplikovanosti. WhatToDo oproti tomu je pro uživatele velice jednoduchá a jednoznačná. Velká skupina uživatelů také v dnešní době upřednostňuje aplikace nové, české a vyhýbá se aplikacím od velkých společností.

2.2. Mapy.cz

Aplikace mapy.cz je velice podobná aplikaci Google Maps, taktéž je uživatelsky méně intuitivní, než aplikace WhatToDo a více složitá. Je to aplikace česká, ale i přesto je tvořena více globálně, není tvořena na míru městu Liberec. Je to aplikace zaměřena pouze na mapy a navigaci, místa jsou pouze vypsána v mapě, není zde žádný seznam kde by se na ně uživatel mohl snadno podívat, taktéž tu nejsou události ani informace o městu Liberec.

2.3. InCity

Aplikace InCity je tvořena pro několik českých měst. Je více zaměřena na konkrétní města, čímž je aplikaci WhatToDo nejpodobnější z tohoto seznamu. Má však spoustu chyb, například po spuštění nabídne nejbližší města, kdy uživateli nacházejícímu se v Liberci nabídne na prvním místě Jičín, Liberec až na čtvrtém. Také mapa není tak intuitivní. Zatímco WhatToDo zobrazuje místa na mapě pomocí fotografií pro co největší přehlednost, aplikace InCity je zobrazuje formou špendlíků. Uživatel tedy musí klikat na každý špendlík zvlášť, aby se dozvěděl, co se na onom místě nachází. Také je zbytečně složitá, uživateli zabere poměrně dlouhou dobu, než se prokliká k místu, které chtěl najít.

2.4.DarumaGo!

Tato aplikace je velice hezká a uživatelsky přívětivá, avšak má jiný účel. Je to v podstatě průvodce městem, ke každému místu tu jsou fotografie a informace, ale není zde funkce navigace k místu. Je to v podstatě brožurka o městě v elektronické podobě.

3. Návrh aplikace

3.1. Přehlednost, intuitivnost

Cílem je vytvořit aplikaci, která bude pro uživatele co nejvíce intuitivní. Uživatel by měl být schopen již při prvním použití aplikace najít v rychlosti co potřebuje. U aplikace tohoto typu je to nezbytné, jelikož turista nechce ztrácet čas hledáním, chce vše vědět ihned a čas na dovolené trávit příjemnějšími činnostmi.

Pro přehlednost aplikace byla vybrána kontrastní paleta barev [Obrázek 1], díky které i při slunečním záření bude aplikace dobře čitelná. Vzhledem k tomu, že aplikace bude využívána především venku, během celého dne, je nezbytné počítat s těmito podmínkami. Aplikace bude díky těmto barvám také pohledově příjemná, díky čemuž zanechá na uživatele dobrý dojem. Barvy jsou také sladěny s barvami propagačního webu.



Obrázek 1 – Barvy

3.2. Vše na jednom místě, na míru

U aplikace tohoto typu je důležité, aby obsahovala všechny informace, které uživatel potřebuje znát. Ať už potřebuje najít hotel, památku, nebo toaletu, je dobré poskytnout mu tyto informace v rámci jedné aplikace. Není pohodlné, aby musel hledat v množství aplikací, každou informaci zvlášť. Cílem je, aby uživatel věděl, že cokoliv v rámci návštěvy města potřebuje zjistit, najde to na jednom místě, v rámci této aplikace.

Jelikož všechna města nejsou stejná, je dobré vytvořit aplikaci, která bude konkrétnímu městu na míru, bude tedy obsahovat pouze informace relevantní k tomuto městu. Vzhledem k výhodě znalosti města Liberec, je dobré využít této znalosti a nabídnout i místa, která nejsou všeobecně známá, přesto mohou být pro návštěvníka zajímavá. Taktéž může být pro uživatele příjemné získávat tipy na události, které se v tomto městě konají, nebo v budoucnosti konat budou. Tato informace by tedy v aplikaci taktéž neměla chybět.

4. Datový model

4.1. Komunikace databáze – aplikace



Obrázek 2 - Komunikace databáze - aplikace

V Cloud Storage jsou uloženy veškeré obrázky, které databáze obsahuje. Tyto obrázky jsou databázi Cloud Firestore předány jako String parametr, kterým je odkaz pro stažení obrázku. Z databáze se data nahrávají do aplikace pomocí vytváření příslušných objektů. Tyto objekty se následně pomocí jednotlivých stránek v aplikaci zobrazují uživateli. Uživatel má možnost přidávat do galerií k jednotlivým místům vlastní fotografie, které se automaticky nahrají do Cloud Storage a po schválení se přidají do Cloud Firestore databáze.

4.2. Návrh databáze

Téměř všechna pole uvedena v příloze A jsou typu String. Pole open a ListImage jsou pole řetězců, Location je typu GeoPoint, Date je ve formátu Timestamp a pole Rating a CountRating jsou typu Number. Bool pole jsou typu Boolean, vyjma Bool2 u události, který je String (z důvodu použití u typu kino, kde se jedná o informaci o věkové přístupnosti). Každá z uvedených kolekcí (Places, Event, Offer) má ještě svou kopii (PlacesToCheck, ...) se stejnými typy dokumentů, kam se dávají objekty, které jsou před schválením. Toto opatření je kvůli bezpečnosti. Jelikož každý uživatel smí přidávat objekty do databáze, mohly by se sem dostat i nevhodné či nepravdivé informace.

5. Struktura aplikace

5.1. Layout

Layout je soubor, pomocí něhož se dá definovat vzhled aktivity. Tento soubor má formát xml, který se propojí na straně aktivity metodou `setContentView()` a na straně Layoutu pomocí `tools:context=".Aktivita"`.

Tento soubor se obvykle vytváří pro každou třídu rozšířenou o `AppCompatActivity`. V tomto projektu layouty obsahují převážně prvky `TextView`, `ImageView` a `Button`. `TextView` je pole zobrazující text, má ale také možnost nastavení funkce `OnClick`, dá se s ním tedy pracovat jako s `Buttonem`. V opačném případě, `Button` může taktéž pracovat jako textové pole, avšak je to oproti `TextView` zbytečně komplikovaný prvek. `ImageView` je prvek zobrazující nějaký obrázek, musí mít vždy nastavený nějaký jako defaultní. V kódu se může podle potřeby měnit pomocí funkce `setImageResource()` nebo `setImageDrawable()`. Obrázek na pozadí se dá ale nastavovat i textovému poli, je-li to potřeba. Dá se říci, že celý layout může obsahovat pouze prvky typu `TextView`, jejichž chování je nastavené podle potřeby jako u prvků jiného typu.

5.2. Aktivita

Aktivita je třída v jazyce Java, která je zobrazována uživateli. Každá aktivita je propojena s příslušným xml souborem Layout, definujícím její vzhled. Každá tato třída musí být rozšířena o třídu `AppCompatActivity`, díky níž je možné používat metody jako `onCreate()`, nebo `onPause()`.

V metodě `onCreate` se zajišťuje v první řadě propojení s Layoutem, pomocí metody `setContentView()`. Dále se zde propojují jednotlivé součásti Layoutu s proměnnými v aktivitě, pomocí metody `findViewById()` a definují ostatní proměnné.

V metodě `onPause` lze definovat činnosti, které má aktivita vykonat po jejím opuštění uživatelem. Tedy co se má stát, když uživatel tuto stránku zavře, nebo otevře jinou. Tato metoda by se měla volat až poté, co jsou dokončeny všechny činnosti spuštěné uživatelem.

5.3. Android Manifest

Každý Android projekt musí mít v kořenovém adresáři zdrojového souboru projektu soubor s názvem `AndroidManifest.xml`. Program Android Studio tento soubor sám vygeneruje po vytvoření projektu. Tento soubor popisuje veškeré základní vlastnosti projektu, jako název aplikace, nebo jak má vypadat její ikona v telefonu uživatele.

Dále se zde nastavují vlastnosti pro všechny zobrazované aktivity, práva, která aplikace chce od uživatele povolit, nebo zálohování aplikace. Také se zde nastavují informace důležité pro Google Play, díky nimž povolí stažení aplikace jen uživatelům, jejichž operační systém tyto povoluje a podporuje všechny potřebné funkce a balíčky.

5.4. Build Gradle

Verze tohoto souboru v projektu existují dvě, jedna pro aplikaci a jedna pro projekt. Tyto soubory jsou stejně jako manifest v projektu povinné a Android Studio je samo generuje.

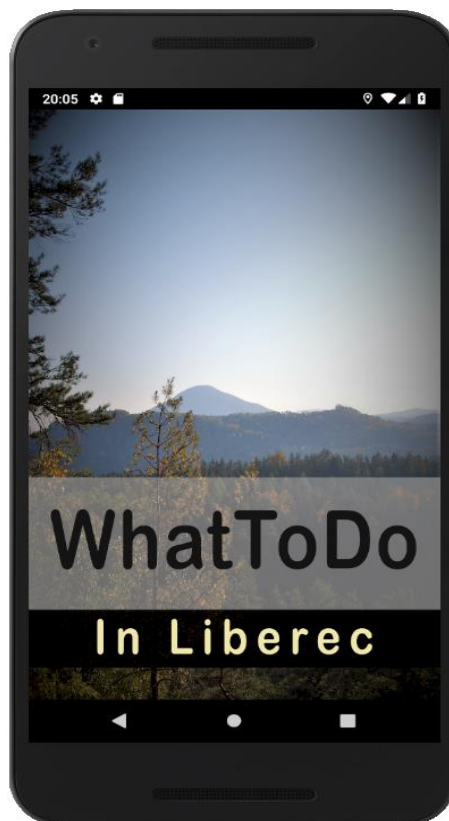
Ve verzi pro aplikaci se nastavuje především na jaké verzi operačního systému Android by ideálně měla aplikace běžet (`targetSdkVersion`), nebo minimální verze (`minSdkVersion`), na které bude aplikace fungovat. Je to důležité z důvodu vývoje verzí OS Android, kdy starší verze nepodporují nové knihovny a naopak. V tomto souboru se také nastavují závislosti. Veškeré balíčky, které chce aplikace používat, se vkládají ve formátu `implementation`.

Ve verzi pro projekt se nastavují například vzdálené zdroje, jako JCentre, nebo Maven Central. Stejně jako ve verzi pro aplikace, se zde nastavují závislosti. Zde jsou tvořeny jako `classpath`. Tyto závislosti určují, co je nezbytné použít pro sestavení projektu.

6. Zobrazované stránky aplikace WhatToDo

6.1.Splash Screen

Splash Screen je třída s vlastním layoutem, který zpravidla obsahuje obrázek pozadí a název aplikace. Může obsahovat taktéž i aktuální verzi softwaru. Obvykle se nastavuje na proces spouštění aplikace. Splash Screen má uživateli ukrátit čekání na načtení metody onCreate nastavené třídy.



Obrázek 3 - Splash Screen

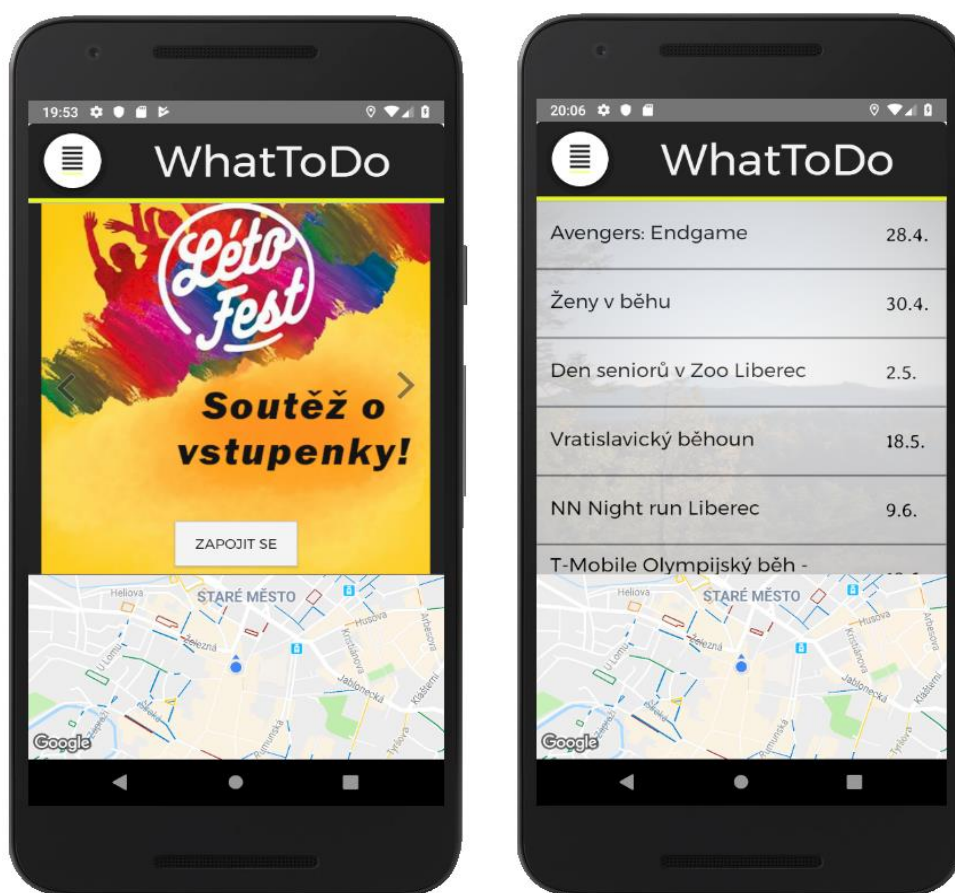
V aplikaci WhaToDo je tato třída nastavena jako launcher aplikace, funguje tedy během načítání třídy Main, zde konkrétně úvodní stránky. Metoda onCreate se v třídě Main spustí až poté, co je připravena komunikace s databází a mapa. Tento proces může zabrat v závislosti na výkonu zařízení a internetovém připojení i několik vteřin. Bez nastavení třídy Splash Screen by se automaticky zobrazila pouze bílá obrazovka, což může být pro některé uživatele nepříjemné. Zatímco s nastavenou Splash Screen se uživateli čekání opticky zkrátí a může se taktéž dozvědět důležité informace o aplikaci.

V aplikaci WhatToDo je stránka navržena minimalisticky [Obrázek 3], jako celý koncept aplikace. Obsahuje pouze pozadí, čímž je cestovatelská fotografie pořízena

v Libereckém kraji, dále je zde již jen název aplikace. Jelikož aplikace má zatím pouze svou první verzi, není zde nutné ji zobrazovat.

6.2. Úvodní stránka

Úvodní stránka je první stránka (po načtení), která se uživateli zobrazí. Musí tedy obsahovat informace, které uživatel bude chtít jako první vidět. Musí taktéž obsahovat menu, díky kterému uživatel bude moci zobrazit další informace, které aplikace obsahuje.



Obrázek 4 – Úvodní stránka

Úvodní stránka (Main Activity) aplikace WhatToDo [Obrázek 4] obsahuje v horní části menu a název aplikace. Jako první se zde zobrazí reklamní oznámení, kde bude aplikace uživatelům nabízet aktuální výhody, které pro ně má připravené. Jako například slevy na vstupenky na nějakou událost nebo soutěže o lístky. Uživatel se do soutěže zapojí kliknutím na tlačítko „Zapojit se“, aplikace ho poté přesměruje k pravidlům soutěže a k vyplnění potřebných údajů. Pravidla soutěže si bude sjednávat firma, pro kterou je aplikace tvořena, nyní tedy nejsou dostupná. Pokud uživatel v tomto místě posune

prstem nahoru, zobrazí se seznam doporučených událostí, které se dějí v aktuální den nebo blízké budoucnosti. Seznam všech událostí je rozebrán v kapitole [226.3]. Po kliknutí na jednotlivé události se uživateli zobrazí jejich detail [6.4].

Dále je zde zobrazena mapa, která je získávána od společnosti Google, více o ní v kapitole [6.8]. Zobrazení mapy je omezeno podle lokace uživatele. Ta je získávána jak z GPS v jeho telefonu („ACCESS_COARSE_LOCATION“), tak pro lepší přesnost i z internetu („ACCESS_FINE_LOCATION“). Pokud má uživatel některou z těchto funkcí vypnutou, aplikace ho vyzve k zapnutí [Obrázek 5]. Pokud uživatel odmítne, mapa se zobrazí okolo centra Liberce.

```
private void buildAlertMessageNoGps() {
    final AlertDialog.Builder builder = new AlertDialog.Builder(context, this);
    builder.setMessage(getString(R.string.gpsMessage))
        .setCancelable(false)
        .setPositiveButton(getString(R.string.yes), (dialog, id) -> startActivity(new Intent(
            android.provider.Settings.ACTION_LOCATION_SOURCE_SETTINGS)))
        .setNegativeButton(getString(R.string.no), (dialog, id) -> dialog.cancel());
    final AlertDialog alert = builder.create();
    alert.show();
}
```

Obrázek 5 – Kód: dialog pro povolení GPS

Dále je zde znemožněn pohyb v mapě. Mapa má sloužit pouze informativně a zobrazit místa nacházející se v největší blízkosti uživatele. Aplikace tím chce dosáhnout toho, že uživatel bude rychle vědět, co se kolem něj nachází a například tak nevynechá zajímavou památku, vedle které bude právě stát. Po kliknutí na tuto mapu se otevře stránka s mapou umožňující pohyb [6.8]. Zablokování pohybu v mapě je zajištěno příkazem „mMap.getUiSettings().setAllGesturesEnabled(false);“.

6.3. Seznam událostí

V této třídě je zobrazen celý seznam událostí z vlastní databáze. Seznam má na pozadí vodoznak úvodního obrázku aplikace. Po prvním otevření se zobrazí doporučené události, to znamená události s nejlepším hodnocením a nejbližším datem. Hodnocení se načítá od uživatelů a u událostí typu film z webu ČSFD. Když si uživatel otevře filtr, může si vybrat mezi typy: Film, Divadlo, Festival, Hudba, Sport a Výstava. Po označení se mu seznam vytrídí podle zvoleného typu, seřazen opět podle hodnocení. Uživatel taktéž může vyhledávat podle textu, který vloží do připraveného textového pole. Vyhledávání funguje automaticky již během toho, co uživatel píše. Pokud uživatel zadá

text, který se v databázi nenachází, zobrazí se mu v seznamu poslední načtená místa (například ta co obsahují předchozí napsané spojení písmen). Vyhledávač nevyhledává řetězec pouze na začátku slov, ale i uvnitř slova.



Obrázek 6 - Seznam událostí

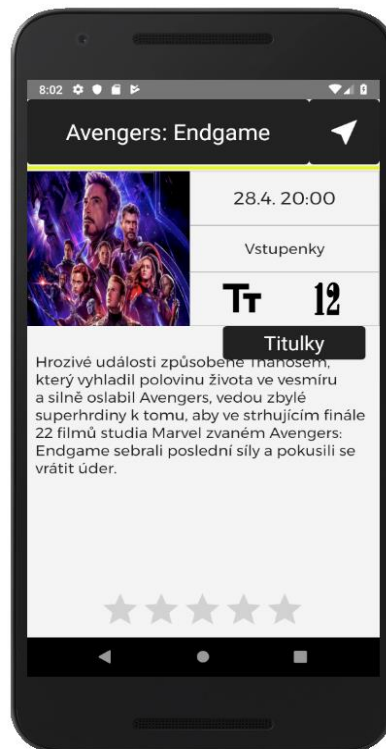
Třída seznam míst obsahuje v horní části opět hlavní menu a název aplikace. Dále v těle stránky zobrazuje seznam míst z databáze, konkrétně název a čas začátku události a ve spodní části filtr a vyhledávač, vše je popsáno v obrázku [Obrázek 6].

6.4.Detail události

Zde aplikace zobrazuje všechny důležité informace o událostech. V horní části layoutu se nachází název události a tlačítko pro spuštění navigace. Dále se zde nachází obrázek události, který je nastaven v kódu podle typu události. Obrázky jsou vlastní, ve formátu png, nebo svg, o tom více v kapitole formáty obrázků [418.7]. Uživatel samozřejmě může přidat obrázek vlastní, avšak aplikace počítá s tím, že to není důležitá informace, tudíž jeho přidání není povinné.



Obrázek 7 - Detail události



Obrázek 8 - Detail události: rozklik

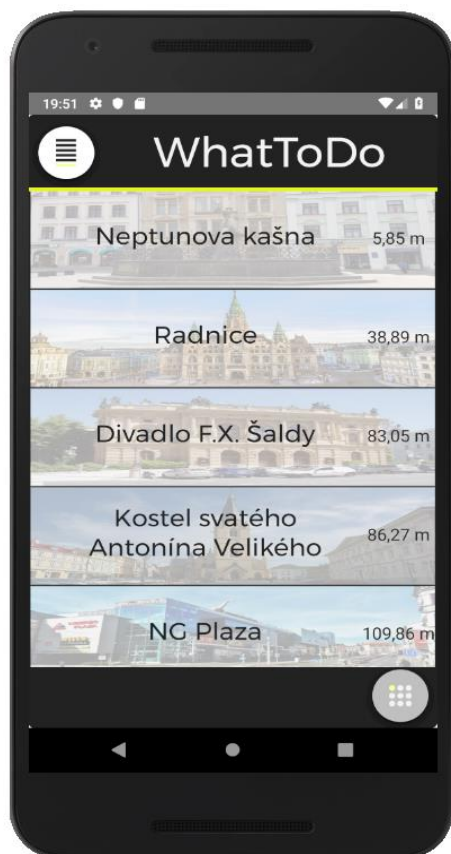
Dále se zde nachází informace o času události, konkrétně začátek. Pokud má aplikace stanovený i konec události, dá se zobrazit po kliknutí na čas začátku. Pod polem času je pole s informací o vstupném. Dále jsou zde dva obrázky, jejichž zobrazení je podmíněné pravdivou informací z databáze (hodnota True). U událostí typu kino je zde informace, zda je film s dabingem (ikona mikrofonu), nebo s titulky (ikona s písmeny) a dále informace o věkové přístupnosti (ikona s číslem). U událostí typu sport, událost, festival, divadlo, hudba a výstava je informace o vhodnosti pro děti a ikona pro koupi vstupenek, pokud je to možné. U ikon, které nepřesměřovávají na koupi vstupenek, se po kliknutí, zobrazí vysvětlivka, co ikona znamená, viz [Obrázek 8]. Čas zobrazení je nastaven pomocí objektu CountdownTimer [8.6.1].

Pod těmito poli je popis události, který je skrolovatelný. Dále je zde rating bar, kde může uživatel vidět, jak je událost hodnocena. Celý vzhled této stránky je vidět na obrázku [Obrázek 7]. Zobrazení hodnocení je nastaveno na hodnocení z databáze. V okamžik, kdy uživatel přidá své hodnocení, zobrazí se mu po dobu nastaveného časového intervalu počet hvězd, co zadal a po uplynutí intervalu opět aktualizované hodnocení z databáze. Po ohodnocení se taktéž zablokuje možnost dalšího hodnocení na dobu 24 hodin. Časové intervaly jsou nastavené pomocí objektu CountdownTimer [8.6.1].

6.5. Seznam míst

Tato stránka zobrazuje doporučená místa, seřazená podle vzdálenosti od uživatele. Vzdálenost je měřena vzdušnou čarou a je získávána výpočtem zohledňujícím pozici místa, uživatele a dále zakulacení země. Tento kód můžete vidět na obrázku [Obrázek 10]. Ideálnější by bylo zobrazovat trasu reálnou, pro pěší či pro automobily, nicméně tato funkce není součástí bezplatné verze, ale je součástí zpoplatněného balíčku Routes. Bylo tedy vyhodnoceno, že pro funkčnost aplikace postačí představa o vzdálenosti, není nutné znát přesnou vzdálenost.

Pod oblastí obsahující menu a název aplikace jsou zobrazena jednotlivá místa. Ke každému je zobrazen jako pozadí vodoznak jeho fotografie. Dále je zobrazeno vždy jméno místa a zmíněná vzdálenost vzdušnou čarou. Výpočet vzdálenosti můžete vidět na obrázku [Obrázek 10]. Ta se zobrazuje v ideální jednotce, pokud je místo vzdáleno několik metrů, je v metrech, pokud je to více než kilometr, zobrazuje se v kilometrech.



Obrázek 9 - Seznam míst


```

double distance(double fromLat, double fromLon, double toLat, double toLon) {
    double radius = 6378137;
    double deltaLat = toLat - fromLat;
    double deltaLon = toLon - fromLon;
    double angle = 2 * Math.asin( Math.sqrt(
        Math.pow(Math.sin(deltaLat/2), 2) +
        Math.cos(fromLat) * Math.cos(toLat) *
        Math.pow(Math.sin(deltaLon/2), 2) ) );
    return radius * angle;
}

```

Obrázek 10 - Získávání vzdálenosti: kód

6.6.Detail místa

Zde aplikace zobrazuje všechny důležité informace o zvoleném místě [Obrázek 11]. V horní části se zde nenachází menu, ale je zde vždy zobrazen název místa a tlačítko pro spuštění navigace. Pod tím se nachází fotografie místa, pokud databáze jeho fotografii neobsahuje, automaticky se zde nahraje vytvořený obrázek příslušný dle typu. Po kliknutí na obrázek se zobrazí rozšířená galerie [6.7] se všemi obrázky příslušnými k tomuto místu, které v databázi aktuálně jsou. Pod obrázkem události je zobrazena nejprve otevírací doba příslušná k aktuálnímu dni. Po rozkliknutí se uživateli zobrazí kompletní otevírací doba na celý týden, pakliže není vždy stejná nebo pokud je otevřeno nonstop/nikdy, viz [Obrázek 12].

Vedle otevírací doby je pole obsahující vstupné, jedná-li se o památku nebo jiný typ místa obsahující vstupné. Restaurace a bary zde mají tlačítko, které je přesměruje na web s menu. Toto textové pole nemění pouze svůj text, ale i vzhled, aby uživateli bylo jasné, zda je pole klikatelné, či ne. Jako další se zde nachází ikonky symbolizující informace k místům. U restaurací se zobrazuje, zda je zde dětský koutek, zahrádka a povolen vstup se psy. U barů to samé vyjma dětského koutku. U památek a kultury se zobrazuje, jestli jsou bezbariérové, vhodné pro děti nebo zda je povolen vstup psům. U toalet se zobrazuje, zda jsou bezbariérové, u parkoviště, jestli je hlídané. Ikony se zobrazují na základě informace z databáze, zda je hodnota pravda, nebo ne. Pokud u některého z míst ani jedna hodnota není pravdivá, nezobrazí se zde nic. Po kliknutí na jednotlivé ikony se uživateli zobrazí jejich význam slovně, pro úplnou přesnost, viz [Obrázek 12].



Obrázek 11 - Detail místa



Obrázek 12 - Detail místa: rozklik

V prostředku layoutu je zobrazen popis místa. V dolní části je rating bar, ukazující hodnocení místa přepočtené z databáze. Uživatel má možnost místo ohodnotit. Hodnocení je omezeno časovačem, za jeden den je povoleno jedno hodnocení pro každého uživatele, aby se zamezilo zneužití. Hodnocení je uloženo v databázi a při každém kliknutí na rating bar se jeho hodnota aktualizuje (přičte se hodnocení a zvedne se počet hodnocení). Aby uživatel věděl, že bylo jeho hodnocení přijato, změní se pomocí časovače na dvě vteřiny počet hvězd na hodnotu jím zvolenou, poté se zobrazí aktualizovaná hodnota z databáze.

6.7. Galerie

Galerie zobrazuje seznam obrázků, příslušný k otevřenému místu, nacházející se v databázi. Pokud k místu není vytvořen seznam, načte se sem pouze úvodní obrázek. Uživatel může obrázky posunout kliknutím na šipku, nebo posunutím prstem na stranu. Obrázky se točí dokola, takže po zobrazení posledního obrázku, se zobrazí opět první.

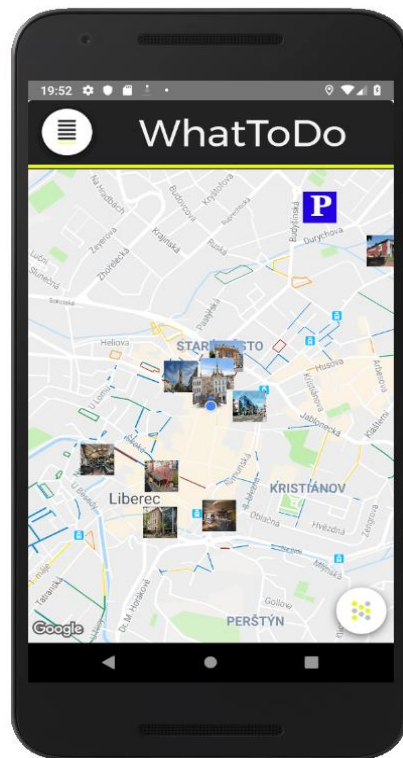


Obrázek 13 - Galerie

Uživatel zde má také možnost nahrát vlastní fotografii z telefonu. Fotografie se ale nepřidá ihned, uloží se nejprve do pomocné databáze. Tam je nutné ji schválit, aby se zobrazila. Právo schvalování má administrátor a správce. Po schválení se bude fotka zobrazovat v této galerii za dříve nahranými fotkami. Fotka se také před přidáním automaticky zmenší, kvůli rychlosti načítání a pro ušetření místa v databázi.

6.8. Mapa

Pro vytvoření této stránky aplikace bylo využito rozhraní Maps SDK [8.3.1] pro Android. Pomocí této API je vykreslována mapa. Dále je díky ní získávána lokace uživatele. Před zobrazením uživateli lokace aplikace nejprve ověří, zda jsou povolené funkce, které potřebuje. Nejprve aplikace zkontroluje, zda je spuštěna GPS a dále zda je dostupné připojení k internetu (wifi, nebo mobilní data). Pakliže má uživatel nějakou z těchto funkcí vypnutou, aplikace ho vyzve k jejímu zapnutí. Pokud uživatel odmítne nějakou z těchto funkcí zapnout, aplikace bude fungovat, ale ne tak, jak by měla. Při vypnutém internetu se uživateli nenačtou data z databáze. Na mapě neuvidí žádná místa. Pokud nepovolí získávání GPS, nezobrazí se mu jeho poloha a kamera mapy se vystředí na centrum Liberce.



Obrázek 14 - Mapa

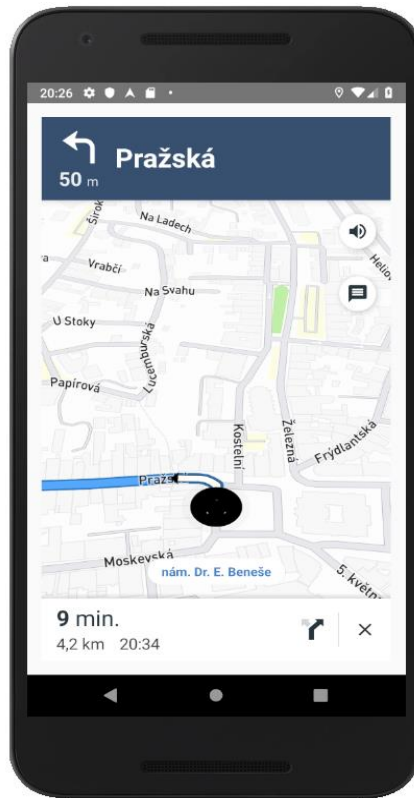
Mapa obsahuje miniaturní obrázky míst z databáze, které jsou vykresleny díky objektu bitmap. Jejich velikost je nastavena na 100*100. Jelikož se mapa kvůli rychlosti aplikace načte pouze na začátku, obrázky zůstanou ve stejné velikosti, jaká byla nastavena na počátku a při přiblížení mapy se nezvětšují. Tato funkce je možným budoucím vylepšením aplikace. Po kliknutí na obrázek místa se uživateli zobrazí stránka detail místa [6.6].

Mapa též obsahuje filtr [Obrázek 18], který umí třídit místa, která se zobrazují. Filtr obsahuje typy míst jako restaurace, toalety, parkoviště, památky nebo kultura. Při zvolení všech míst, nebo parkovišť, se uživateli na mapě zobrazí navíc vrstva s vykreslenými parkovacími zónami. Tato vrstva byla ručně vytvořena jako KML [8.3.2]. Informace o zónách jsou ze stránek Liberce [1].

6.9. Navigace

Navigace je tvořena za pomoci knihovny Mapbox [8.4], konkrétně její Navigation SDK. Poté, co uživatel na stránce Detail místa nebo Detail události klikne na symbol navigace (šipka), aplikace ho přesměruje na tuto stránku. Zároveň jako objekt Bundle předá informaci o tomto místě/události pro možnost získání její polohy.

Ze získané polohy a aktuální polohy uživatele je následně vytvořena trasa, ukázka kódu na obrázku [Obrázek 16]. Po vytvoření trasy se nastaví předvolby pro objekt NavigationView a následně je spuštěna navigace.



Obrázek 15 - Navigace

```
private void fetchRoute(Point origin, Point destination) {
    NavigationRoute.builder(getContext()).accessToken(getString(R.string.access_token))
        .origin(origin)
        .destination(destination)
        .build().getRoute(new Callback<DirectionsResponse>() {
            @Override
            public void onResponse(@NotNull Call<DirectionsResponse> call,
                                   @NotNull Response<DirectionsResponse> response) {
                directionsRoute = response.body().routes().get(0);
                startNavigation();
            }
            @Override
            public void onFailure(@NotNull Call<DirectionsResponse> call, Throwable t) {
            }
        });
}
```

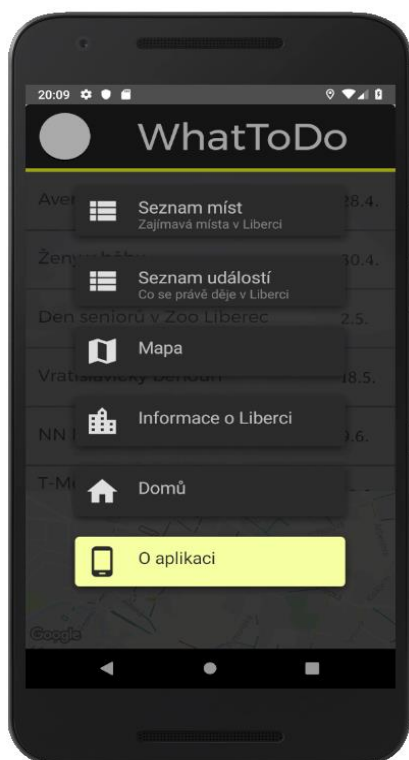
Obrázek 16 - Navigace: kód

6.10. Menu a filtry

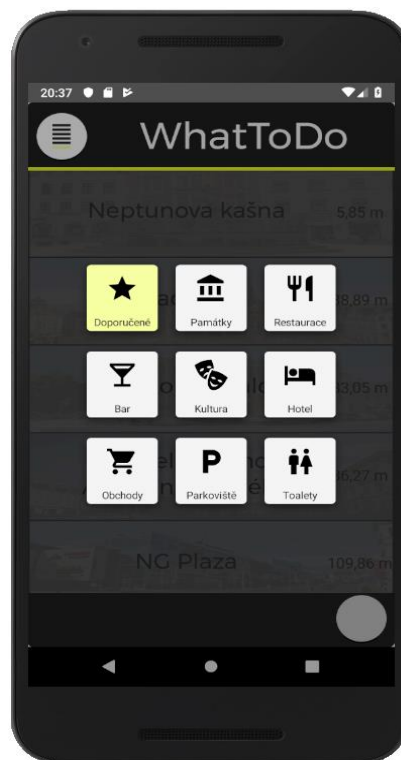
Menu i filtry jsou v tomto projektu vytvořeny za pomoci knihovny BoomMenu, o něm více v kapitole [8.5].

Menu zobrazuje všechny hlavní stránky aplikace, stránky Detail místa a Detail události se zobrazují po kliknutí na konkrétní položku v příslušném seznamu. Stránka Galerie se zobrazuje po kliknutí na úvodní fotku u některého z míst. Z jiného místa se kvůli přehlednosti k těmto stránkám nedá dostat. Všechny stránky nabízené v menu, obsahují totožné menu jako na obrázku [Obrázek 17]. Položky jsou na tmavě šedém pozadí se stříbrnou barvou textu, vyjma položky „O aplikaci“, jelikož svým účelem a obsahem nemá stejný význam jako zbylé položky. Pro běžného uživatele je vedlejší, je to pouze informativní stránka, je proto zobrazena jinou barvou. Položka je barevně výraznější, protože aplikace chce nalákat uživatele podívat se a zjistit více o tomto projektu.

Filtr je vzhledově stejný u seznamu míst i událostí, mění se jen jeho položky. Na obrázku [Obrázek 18] můžete vidět filtr míst. Všechny kategorie jsou zobrazovány na stříbrném pozadí s tmavě šedým textem, vyjma položky „Doporučené“, ta je zvýrazněná, aby bylo znát, že se liší.



Obrázek 17 - Menu



Obrázek 18 - Filtr míst

7. Pomocné třídy

7.1. Place, Event, Offer

Pomocí těchto tříd se vytváří příslušné objekty. Obsahují potřebné informace, které daný typ má. Jsou vytvářeny pomocí konstruktorů, kterým jsou předány parametry načtené z databáze.

Place: obsahuje proměnné id, jméno, typ, list obrázků, lokaci, popis, otevírací doba, hodnocení, počet hodnocení a tři boolovské proměnné, měnící se podle typu místa. Dále obsahuje konstruktor, gettery a settery ke všem proměnným, metody třídy Parcelable a metodu pro získání počtu hvězd pro ratingBar.

Event: je obsahem téměř stejný, jako Place, pouze neobsahuje list obrázků a boolovské proměnné má pouze dvě.

Offer: obsahuje pouze id, jméno a obrázek. V budoucnu bude přidána proměnná určující datum expirace nabídky. Stejně jako předchozí dvě třídy obsahuje konstruktor, metody třídy Parcelable a potřebné gettery a settery.

7.2. PlaceListAdapter a EventListAdapter

Tyto třídy slouží pro zobrazování položek v prvku ListView. Bez této třídy by nebylo možné zobrazovat více informací v jedné položce seznamu, ale pouze jednu. Také se zde upravuje chování a vzhled jednotlivých částí. Tato třída je rozšířena o třídu ArrayAdapter. V této třídě je také zajištěno filtrování položek, pomocí implementace Filterable. Každá z těchto tříd má vytvořené dva vlastní filtry, jeden pro vyhledávání a jeden pro filtrování, ve kterých přepisuje metody performFiltering() a publishResults().

7.3. OnSwipeTouchListener

Tato třída pomáhá k určení pohybu prstem po telefonu. Pomocí souřadnic zjišťuje, jak uživatel posunul prst po obrazovce (zprava doleva a naopak, shora dolů a naopak). Tato třída je využita například na Úvodní stránce, kde uživatel pohybem prstu po obrazovce může zavřít nabídky s akcemi a zobrazit seznam událostí. Dále je využita v Galerii pro posouvání mezi fotkami.

8. Použité technologie

8.1. Java vs. Kotlin

Java je objektově orientovaný jazyk vyvinutý společností Sun Microsystems v roce 1995. Java je dobře přenositelná, dá se použít nejen pro mobilní aplikace, ale i pro aplikace na osobní počítače, čipové karty, nebo jiná zařízení. To dělá z Javy jeden z nejpoužívanějších a nejoblíbenějších jazyků na světě.

Kotlin je objektově i funkcionálně orientovaný jazyk vyvinutý společností JetBrains v roce 2011. První oficiálně stabilní verze byla vydána až v roce 2016. Čerpá v mnoha věcech z Javy, na některých jejích knihovnách je dokonce závislý. Kompatibilní s Javou je pouze binárně, mohou tedy běžet na stejném virtuálním stroji, ale syntakticky jsou rozdílné. Syntaxe se dá přirovnat spíše k jazyku Python, ale je vyžadována deklarace typu proměnné a závorky okolo funkcí a cyklů. Kotlin je snadný pro naučení, jelikož vychází z již existujících jazyků jako Java, C, nebo Python.

Oproti Javě je u Kotlinu vývoj aplikací pro Android výrazně snazší a čistší. Vývojář udělá stejnou práci s mnohem méně řádků kódu. Je zde znát, že Kotlin byl vytvořen jako vylepšení Javy, odstraňuje tedy některé její nedostatky. Například Kotlin nenutí vývojáře ošetřovat tolik výjimek jako Java. Kotlin je ale nový a stále se pracuje na jeho stabilitě. Aktuálně je již hodně stabilní, proto některé velké společnosti jako Amazon, Uber, Netflix, nebo Pinterest již přešli na Kotlin. V budoucnosti je tedy možné, že Kotlin zcela vytlačí Javu z oblasti Android aplikací.

V tomto projektu byla použita pouze Java, vzhledem k větším zkušenostem, stabilitě a stále ještě širší dokumentaci. V budoucnosti ale není vyloučeno, že bude projekt převeden do Kotlinu, především kvůli úspoře kódu.

8.2. Firebase

Firebase je platforma společnosti Google (od roku 2014), určena pro vývoj Android, IOS, webových a C++ aplikací. Firebase má uživateli pomoci, vytvořit kvalitní aplikaci v krátkém čase. Uživatel má k dispozici statistiky využívání jeho aplikace. Má tak jasný přehled o tom, co je třeba vylepšit, jaký prvek databáze uživatele nejvíce zajímá a tak dále. Firebase nabízí hned několik služeb. [5]

8.2.1. Realtime Database

Jedná se o NoSQL databázi, která se zařízením komunikuje v řádu milisekund. Aplikace s touto databází není závislá na internetovém připojení, data mu zůstanou k dispozici i bez něj a po opětovném připojení se aktualizují. Tato databáze je vhodná pro aplikace, ve kterých probíhá komunikace mezi uživateli navzájem, protože přenos informace je velice rychlý a data v zařízení se automaticky aktualizují. Je zde také dobře a intuitivně vyřešen systém autentifikace. Nevýhoda této databáze je, že vývojář zde může mít pouze jednu databázi pro projekt, na rozdíl od Cloud Firestore.

8.2.2. Cloud Firestore

Firestore je nová a vylepšená Realtime Database, obsahuje tedy stejné funkce. Model je zde více intuitivní a dotazy jsou rychlejší. Vývojář může mít databází kolik potřebuje. Databáze se dělí vždy na dokumenty a kolekce.

8.2.3. Cloud Storage

Cloud Storage je internetové úložiště, které může vývojář do jisté míry využívat zdarma. Vývojář sem může nahrávat obrázky, nebo videa, která může odkazem propojit se svými dokumenty v databázi. Úložiště má zajištěna ošetření při ztrátě internetového připojení, jako například zastavení přenosu dat a při opětovném připojení automaticky přenos obnoví.

8.3. Google Maps

8.3.1. Maps SDK

Tato sada nástrojů pracuje na základě získávání dat z GoogleMaps. Automaticky získává přístup na servery společnosti Google, stahuje tamní data a zobrazuje získané informace. Vývojář může upravovat zobrazení mapy jako úhel kamery, její polohu na mapě, přidávat značky, nebo upravovat vzhled mapy. Také je možné vytvořit trasu pomocí přidávání linek a bodů, nebo vymezení území pomocí vykreslení obrazců. [0]

Sada map neobsahuje funkci pro výpočet trasy, nebo vzdálenosti, je zde možné pouze vykreslování map a vlastních informací. Pro trasy a vzdálenosti existuje balíček Routes. Pro získávání míst, uložených v databázi Google Map lze použít balíček Places.

8.3.2. KML

KML je vrstva mapy používající se pro zobrazení geografické informace. Vývojář pomocí ní může vykreslit informaci jako zónu, trasu, místo, lze zde měřit vzdálenosti, nebo přidávat přímky a obrazce. Lze zde také ovlivnit celkový vzhled mapy, změnit barvy a tak dále. Tyto vrstvy se dají skládat na sebe téměř v libovolném množství, omezen je pouze počet informací, nacházejících se ve vrstvách. Optimální maximální počet vrstev je 10–20.

Nejjednodušší způsob pro vytvoření KML vrstvy je využít prostředí Google a jejich službu Moje mapy. Po vytvoření mapy stačí vybrat export do KML a vrstva lze ihned použít. Příklad jednoduchého použití:

- Pro použití vrstvy v Android studiu, je nejprve nutné ji přidat do Raw složky projektu.
- Dále je třeba vytvořit import:
`import com.google.maps.android.kml.KmlLayer;`
- Ve třídě mapy vytvořit privátní proměnnou:
`private KmlLayer layer;`
- Ve metodě onCreate:

```
try {
    layer = new KmlLayer(mMap, R.raw.layer, context);
} catch (XmlPullParserException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}
}
try {
    layer.addLayerToMap();
} catch (IOException e) {
    e.printStackTrace();
} catch (XmlPullParserException e) {
    e.printStackTrace();
}
}
```

Obrázek 19 - KML: kód

8.4. Mapbox

8.4.1. Maps

Maps knihovna od společnosti Mapbox dokáže dynamicky vykreslovat v reálném čase mapu. Umožňuje mapu upravovat dle požadavků aplikace a využívat data, která jí jsou poskytnuta. Mapy Mapboxu jsou velice přesné, využívají pro to více než 130 zdrojů z celého světa.

Použití:

- Nejprve je nezbytné přidat náležitosti pro instalaci balíčků Mapbox

```
repositories {
    mavenCentral()
}
dependencies {
    implementation 'com.mapbox.mapboxsdk:mapbox-android-sdk:'version''
}
```
- Poté je nutné získat unikátní klíč, o který lze zažádat na webu společnosti, ten se poté používá při vytváření instance mapy.
- Dále je nutné nastavit oprávnění pro získávání lokace uživatele, stejně jako u map společnosti Google.
- Dále do příslušného Layoutu přidat prvek mapy a do aktivity jeho inicializaci.
- Poté se dá mapa upravovat podle potřeby co se týče vzhledu i funkčnosti.

8.4.2. Navigation

Mapbox oproti mapám Google umožňuje využívat přednastavenou metodu navigace. Dá se tedy s tímto balíčkem vytvořit snadná instance navigace, bez vlastnoručního vytváření bodů trasy a podobně. [7]

8.5.Boom Menu

Boom Menu je knihovna usnadňující úpravu vzhledu menu. Klasické menu, které nabízí Android Studio je nevzhledné a nelze ho téměř upravit. Boom Menu je možno upravovat podle potřeby. Například lze upravit velikost položek, velikost písma, barvy, pozadí, styl zobrazování a podobně. Je zde také spousta předdefinovaných stylů, ze kterých lze vybírat, všechny jsou ale velice barevné, a ne příliš elegantní. Další zajímavou funkcí jsou způsoby otevírání menu, lze vybrat z několika animací, jak se políčka menu nebo filtrů budou zobrazovat. V tomto projektu tato funkce nebyla využita, protože se nehodí ke stylu projektu. [80]

8.6.Zajímavé použité objekty a třídy

8.6.1. CountdownTimer

CountDownTimer, je objekt určený pro časování událostí v Android aplikaci. Má dvě metody, onTick a onFinish. Metoda onTick může nastavit událost, která se bude dít po dobu nastaveného časového intervalu. onFinish spustí nějakou událost, po uplynutí časového intervalu. [90]

Příklad použití viz [Obrázek 20]

```
CountDownTimer timer = new CountDownTimer( millisInFuture: 1500, countDownInterval: 1000) {
    @Override
    public void onTick(long millisUntilFinished) {
        ratingBar.setRating(rating);
        Toast.makeText
            (context, text: getString(R.string.thanks) + rating, Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onFinish() {
        ratingBar.setRating(place.getStarRating());
    }
}.start();
```

Obrázek 20 - CountdownTimer: kód

Zde je nastaven časový interval na vteřinu a půl. Tedy 1500 jednotek, kdy 1000 je jedna jednotka. Tento časovač nastavuje viditelnost a reakci na hodnocení místa od uživatele. Nejprve změní počet zobrazených hvězd na ukazateli na uživatelské hodnocení a vypíše poděkování. Po uplynutí intervalu se opět změní počet hvězd na aktualizovanou hodnotu z databáze.

8.6.2. Bundle

Bundle je objekt sloužící pro přenos informací mezi jednotlivými aktivitami. Při vytváření jakékoliv aktivity, se vždy jako objekt Bundle získává informace o stavu aplikace. Také se pomocí něho dá předávat Parcelable objekt. V aplikaci WhatToDo se mezi aktivitami předává vytvořené místo, nebo aktivita s příslušnými informacemi. [10]

```
Příklad použití:           Bundle data = getIntent().getExtras();
                           place = data.getParcelable("place");
                           name.setText(place.getName());
```

8.6.3. Intent

Intent je abstraktní popis budoucí operace. Pomocí této knihovny se dá vytvořit přechod z jedné aktivity do jiné. [11]

```
Příklad použití:   Intent intent = new Intent(context, EventDetail.class);
                   intent.putExtra("event", event);
                   startActivity(intent);
```

8.6.4. Toast

Je to objekt, umožňující vývojáři vypsát zprávu na obrazovku uživatele. Dá se pomocí něj například oznámit nějaká změna, upozornit uživatele o úspěchu nějaké operace a podobně. Toast se zobrazí obvykle v dolní části obrazovky, jeho pozice se nicméně dá nastavit pomocí funkce setGravity. Zobrazí se po nastavený časový interval a poté opět zmizí. Uživatel během tohoto výpisu není nijak omezen, zůstane stále na stejné stránce. [12]

Příklad použití:

```
Toast toast = Toast.makeText(context, „Váš text“, Toast.LENGTH_SHORT);
toast.setGravity(Gravity.CENTER, 0, 0);
toast.show();
```

Zde je nastaven časový interval na LENGTH_SHORT, dá se nastavit i na LENGTH_LONG. Je to přednastavená hodnota ve formátu int. Poloha výpisu je zde nastavena na střed obrazovky.

8.6.5. BitmapFactory

Objekty (obrázky) ve formátu bitmap se dají vykreslit pomocí BitmapFactory. Lze je vykreslit pomocí zdroje, souboru, nebo InputStreamu. V projektu byl použit právě InputStreamu. Příklad použití je vidět v obrázku [Obrázek 21].

8.6.6. AsyncTask

AsyncTask je abstraktní třída umožňující snadno provádět operace na pozadí a publikovat výsledky na vlákno uživatelského rozhraní bez nutnosti manipulace s vlákny a použití instance třídy Handler. AsyncTask je vhodné použít pouze pro krátké operace, trvající v řádu sekund. Pro operace trvající déle je vhodnější použít třídu FutureTask, nebo ThreadPoolExecutor. [13]

Po použití AsyncTasku se přiřazená operace provede na pozadí běhu aplikace, nezdržuje tedy načtení a může se načíst i po zavolání funkce onCreate(). Vývojář má možnost použít doInBackground(), onProgressUpdate(), onPreExecute() a onPostExecute() metody, které jsou popsány níže.

Funkce:

- doInBackground(): zavolá se ihned po provedení úlohy, tedy po dokončení metody onPreExecute(). Používá se k operacím, které mohou trvat déle. Obsahuje vstupní parametry a vrací výsledek této operace.
- onProgressUpdate(): Tato metoda se používá k zobrazení jakékoli formy pokroku v uživatelském rozhraní, zatímco operace je stále vykonávána na pozadí.
- onPreExecute(): provede se před provedením úlohy. Používá se například k nastavení úlohy, nebo vykreslení postupu průběhu nějaké operace.
- onPostExecute(): provede se po dokončení všech operací. Výsledek těchto operací mu je dán jako parametr.

V aplikaci WhatToDo byla tato třída použita například pro načítání obrázků na místo špendlíků v mapě. Jednotlivá místa se načtou ihned po spuštění aplikace, ale stažení obrázků chvíli trvá a při vykreslování mapy je nutné obrázek již znát, bylo nutné tento

problém vyřešit právě asynchronní operací. Na obrázku [Obrázek 21] je vidět ukázka kódu, kde je řešen právě tento problém.

```
@SuppressWarnings("StaticFieldLeak")
protected class downloadImagesTask extends AsyncTask<ArrayList<Places>, Void, Boolean> {
    ArrayList<MarkerOptions> markerOptionsAsync = new ArrayList<>();
    @SafeVarargs
    @Override
    protected final Boolean doInBackground(ArrayList<Places>... arrayLists) {
        Boolean ret = true;
        Bitmap bitmap;
        ArrayList<Places> placesList = arrayLists[0];
        for(Places p : placesList){
            try {
                MarkerOptions mOptions;
                URL imageUrl = new URL(p.getImage());
                bitmap = BitmapFactory.decodeStream(imageUrl.openConnection().getInputStream());
                mOptions = new MarkerOptions()
                    .position(new LatLng(p.getLocation().getLatitude(), p.getLocation().getLongitude()))
                    .title(p.getName())
                    .icon(BitmapDescriptorFactory.fromBitmap(resizeMapIcons(bitmap, width: 100, height: 100)));
                markerOptionsAsync.add(mOptions);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return ret;
    }
}
```

Obrázek 21 – AsyncTask + Bitmap: kód

8.6.7. Location

Location je třída dat, zobrazující zadané geografické umístění. Objekt Location generovaný pomocí třídy LocationManager, se vždy skládá ze zeměpisné šířky, délky a časového razítka, dalšími možnými parametry jsou například nadmořská výška, úhel, nebo rychlost. [14]

Zajímavé funkce:

- `convert(double souřadnice, int výstup)`: tato funkce převede zadanou souřadnici do textového formátu. Výstup může být ve formátu stupňů, minut, nebo sekund. Existuje taktéž funkce `Convert(String souřadnice)`, která převede hodnotu zpět na double.
- `distanceBetween()`: do této funkce se zadávají jako vstupní parametry dva geografické body, ve formátu zeměpisné šířky a délky. Funkce vrací vzdálenost v metrech mezi těmito body. Dá se pomocí ní zjistit například nejkratší cestu mezi dvěma místy.

- distanceTo(): tato funkce se dá zavolat na naši vytvořenou lokaci (například aktuální pozici uživatele), jako parametr zadat lokaci, ke které chceme zjistit vzdálenost a funkce tuto vzdálenost vrátí.
- getAccuracy(): vrací odchylku v metrech. Čím menší číslo, tím větší přesnost.
- getLatitude(): vrátí zeměpisnou šířku místa ve stupních.
- getLongitude(): vrátí zeměpisnou délku místa ve stupních.

8.6.8. LocationManager

Tato třída umožňuje aplikaci získat informaci o geografické poloze zařízení, nebo zjistit, zda se zařízení dorazilo k zadanému místu. [15]

Zajímavé funkce:

- addProximityAlert(): přidá vlastnost, že kdykoliv se zařízení dostane do blízkosti zadaného místa / opustí toto místo, upozorní to aplikaci. Informace může být nepřesná. Ve chvíli, kdy uživatel projde skrze toto místo rychle, nemusí se stihnout upozornění vyvolat. Pokud uživatel projde velice blízko zadaného místa, může to upozornění naopak vyvolat, aniž by uživatel na toto místo opravdu vstoupil.
- getGpsStatus(): získává informaci o aktuálním stavu GPS modulu. Tato metoda by se měla volat pouze po změně statusu GPS, aby fungovala korektně. K funkčnosti této metody je nutné mít povolenou přesnou lokaci (ACCESS_FINE_LOCATION).
- getLastKnownLocation(): metoda vrací poslední zaznamenanou lokaci uživatele, nikoliv jeho aktuální lokaci. Ty se mohou lišit například pokud uživatel vypnl přístup k GPS, nebo internetu. V takovém případě je zobrazována poslední uložená informace před vypnutím. K funkčnosti je nutné povolení přibližné i přesné lokace, to znamená povolení GPS v telefonu i povolení internetu (ACCESS_FINE_LOCATION a ACCESS_COARSE_LOCATION).

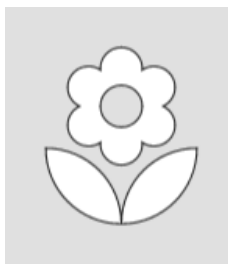
8.7. Formáty obrázků

8.7.1. Vektor

Prostředí Android Studia umožňuje snadné vytvoření obrázků ve formátu vektor. Po kliknutí pravým tlačítkem na složku Drawable v navigačním panelu na pravé straně Android Studia, vyberete možnost New a zde vyberete Vector Asset. V nabídce je k dispozici spousta klipartů, které lze libovolně upravovat. Je zde možnost přidání i vlastního obrázku s koncovkou svg. Po přidání a následném otevření obrázku se dá upravovat spousta parametrů, jako velikost, barva, nebo orámování, jak je vidět na obrázku [Obrázek 22]. Jak takto vytvořený obrázek vypadá, můžete vidět na obrázku [Obrázek 23], obrázek je zobrazen na šedém pozadí, to je upraveno jinde. [16]

```
<vector android:autoMirrored="true" android:height="24dp"
  android:viewportHeight="24.0"
  android:viewportWidth="24.0" android:width="24dp" xmlns:android="http://schemas.android.com/apk/res/android">
  <path android:fillColor="#FFFFFF"
    android:pathData="M12,22c4.97,0 9,-4.03 9,-9 -4.97,0 -9,-4.03 -9,9z"m5.6,10.25c0,
    1.38 1.12,2.5 2.5,2.5 0.53,0 1.01,-0.16 1.42,-0.441-0.02,0.19c0,1.38 1.12,2.5 2.5,
    2.5s2.5,-1.12 2.5,-2.51-0.02,-0.19c0.4,0.28 0.89,0.44 1.42,0.44 1.38,0 2.5,
    -1.12 2.5,-2.5 0,-1 -0.59,-1.85 -1.43,-2.25 0.84,-0.4 1.43,-1.25 1.43,-2.25 0,
    -1.38 -1.12,-2.5 -2.5,-2.5 -0.53,0 -1.01,0.16 -1.42,0.4410.02,-0.19c14.5,2.12 13.38,
    1 12,189.5,2.12 9.5,3.510.02,0.19c-0.4,-0.28 -0.89,-0.44 -1.42,-0.44 -1.38,0 -2.5,1.12 -2.5,
    2.5 0,1 0.59,1.85 1.43,2.25 -0.84,0.4 -1.43,1.25 -1.43,2.25z"m12,5.5c1.38,0 2.5,
    1.12 2.5,2.5s-1.12,2.5 -2.5,2.5s9.5,9.38 9.5,8s1.12,-2.5 2.5,-2.5z"m3,13c0,4.97 4.03,
    9 9,9 0,-4.97 -4.03,-9 -9,-9z"
    android:strokeColor="#212121"
    android:strokeWidth="0.2"
  />
</vector>
```

Obrázek 22 - Vektorový obrázek: kód



Obrázek 23 - Vektorový obrázek: výsledek

8.7.2. PNG

Obrázky ve formátu png se do projektu přidávají velice jednoduše. Stačí obrázek zkopírovat a vložit do složky Drawable ve Vašem Průzkumníku souborů. Obrázek se poté již bude zobrazovat v navigačním panelu a je možné s ním libovolně pracovat.

8.8. Formát textových řetězců

Z důvodu vícejazyčnosti, je dobré text do aplikace nepsat přímo, ale ukládat ho jako zdroj do složky Strings. Po vytvoření prvního souboru je možné otevřít Translation Editor, kde jde přidat libovolné množství jazyků a každému slovu/věťe přiřadit hodnoty v těchto jazycích, jak je vidět na obrázku [Obrázek 24]. Můžete také určit, že slovo nechcete překládat, to se hodí použít pro pomocná slova, která uživatel neuvidí. V tomto projektu byly vytvořeny 4 soubory, základní, čeština, angličtina a němčina. Po vytvoření těchto souborů, se bude nastavovat jazyk v aplikaci automaticky, na základě nastaveného jazyku na telefonu uživatele.

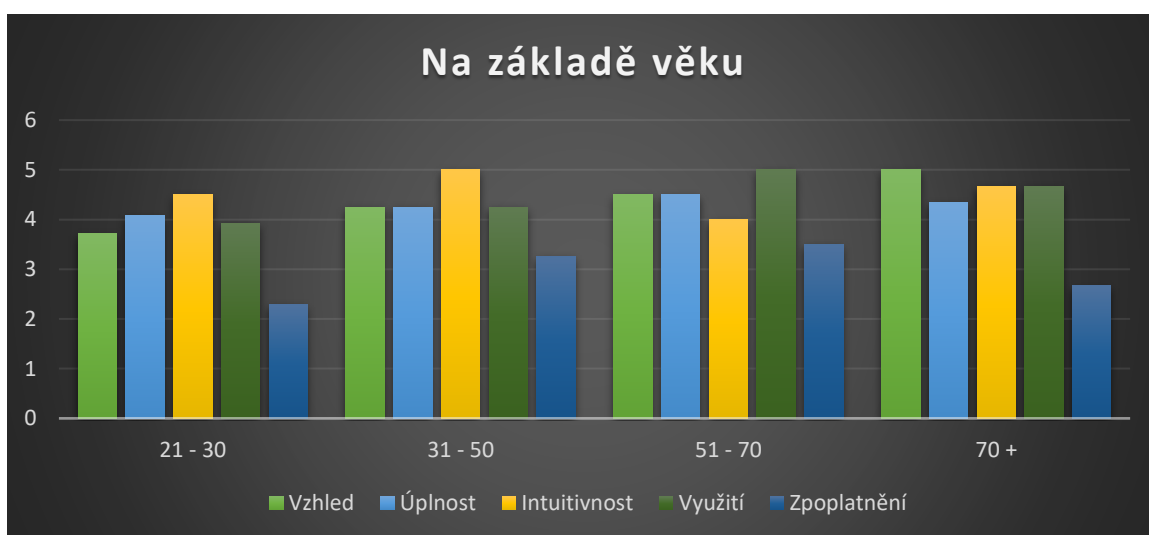
vše	app/src/main/res	<input type="checkbox"/>	all	vše	all	ales
památk	app/src/main/res	<input type="checkbox"/>	monument	památk	monument	denkmal
restaurace	app/src/main/res	<input type="checkbox"/>	restaurant	restaurace	restaurant	restaurant
kultura	app/src/main/res	<input type="checkbox"/>	culture	kultura	culture	kultur
obchod	app/src/main/res	<input type="checkbox"/>	shop	obchod	store	geschäft
ostatní	app/src/main/res	<input type="checkbox"/>	rest	ostatní	rest	andere
image	app/src/main/res	<input type="checkbox"/>	image	obrázek	image	der Bildchen
database	app/src/main/res	<input type="checkbox"/>	places	places	places_en	places_de
playground	app/src/main/res	<input type="checkbox"/>	playground	Dětský koutek	Kids corner	Kinderecke
description	app/src/main/res	<input type="checkbox"/>	description	description	description_en	description_de
entry	app/src/main/res	<input checked="" type="checkbox"/>	entry			

Obrázek 24 - Editor pro vícejazyčnost

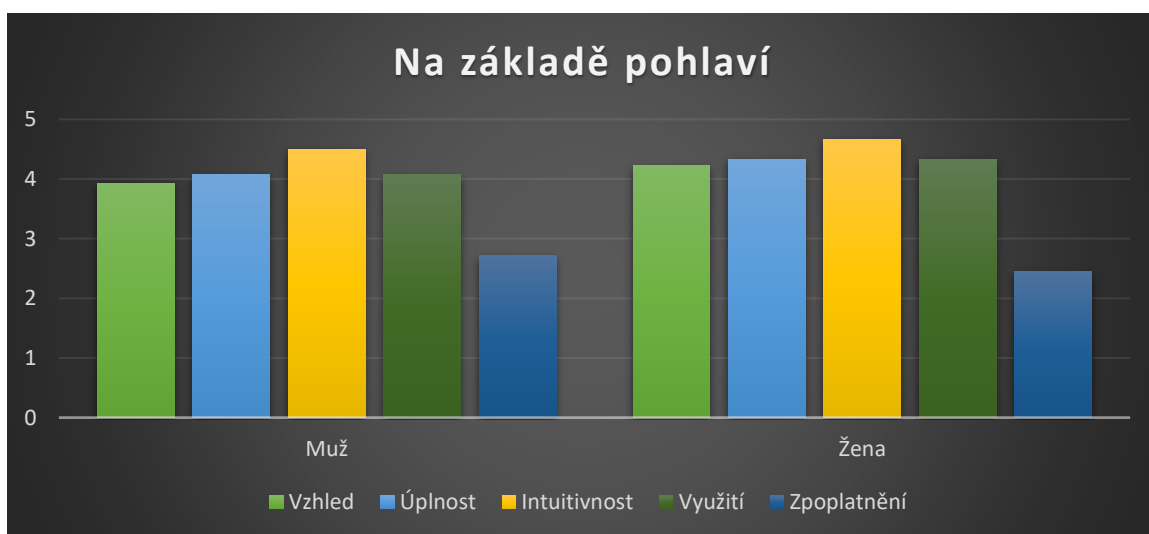
Řetězce získávané z databáze jsou primárně v češtině. Názvy a popisky míst má však uživatel možnost přidat i v angličtině a němčině, není to však povinností. Pokud zde existuje záznam o dalším jazyce, zobrazuje se automaticky informace podle jazyku nastaveného na telefonu uživatele.

9. Testování u uživatelů

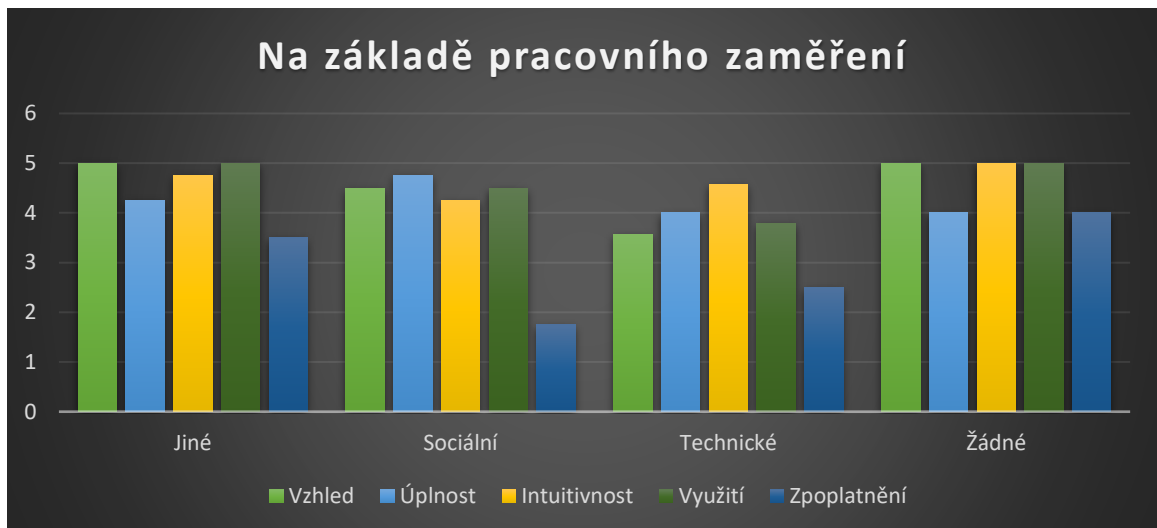
Testování probíhalo pouze na jednom zařízení, jelikož aplikace ještě není dostupná ke stažení. Uživatelé mohli posoudit především vzhled a vlastnosti aplikace. Kompatibilita s jednotlivými verzemi Androidu a různými typy zařízení byla testována pouze na virtuálních zařízeních. Zpětná vazba od uživatelů na toto téma tedy bohužel nebyla možná. Průzkum se dotazoval na uživatelův názor na vzhled, intuitivnost, obsahovou úplnost, zda by uživatel pro návštěvu Liberce aplikaci využil, případně jakou aplikaci obvykle využívá a zda by byl ochoten zaplatit za stažení aplikace, případně kolik korun. Získané informace byly seskupeny podle věku pohlaví a pracovního zaměření uživatele. Výsledek šetření viz [Obrázek 25] - [Obrázek 27].



Obrázek 25 - Graf 1



Obrázek 26 - Graf 2



Obrázek 27 - Graf 3

Uživatelům se nejvíce líbila myšlenka, jednoduchost a intuitivnost aplikace. Nelíbil se jim naopak vzhled aplikace, který vypadal staromódně a některé stránky působily nudným dojmem. Na základě této kritiky byla snaha tento nedostatek odstranit. Některým uživatelům se také nelíbilo hamburger menu, nahradili by ho gestem otevírané menu. To však vzhledem k dosažení intuitivnosti není vhodným řešením, některým uživatelům by to mohlo připadat nejasné. Aplikace, které respondenti uvedli že pro turistické účely běžně využívají jsou Trip Advisor, TimeOut a Mapy.cz. Cena, kterou by uživatelé byli ochotni za podobnou aplikaci utratit, se pohybuje okolo 30 Kč. Avšak cílem je aplikace zcela bezplatná. Nezkrácená zpětná vazba viz příloha B.

10. Úskalí práce

10.1. Android Studio

Android Studio je náročný program vyžadující výkonný počítač. Jelikož na počátku tvoření tohoto projektu takový počítač nebyl dostupný, program nefungoval tak, jak měl, nebylo možné ani vytvořit projekt. S koupí nového počítače se tento problém vyřešil.

10.2. Problémy s verzemi OS Android

Operační systém Android se neustále vyvíjí a zlepšuje svoje funkce, aby byl pro uživatele co nejpřívětivější. Vzniká tím ale problém s jeho verzemi. Jelikož nové verze systému přestávají podporovat staré knihovny. Tvůrci knihoven a balíčků pro tento operační systém s těmito pokroky často nestíhají držet krok. Stává se, že ve své dokumentaci doporučují použít balíčky, nebo funkce, které již nejsou dostupné.

Tvůrci systému pro vyřešení tohoto problému vyvinuli open-source projekt Android X. Tento projekt umožňuje využívání veškerých knihoven a funkcí, napříč verzemi. Usnadňuje tedy vývoj a testování aplikací.

Přenesení projektu je navíc velmi jednoduché, stačí zvolit v menu položku Refactor a vybrat Migrate to AndroidX. Tímto se ihned vyřeší problém s verzemi a lze používat všechny balíčky již od verze 1.0.0. systému Android.

10.3. Licencování GoogleMaps

Stejně jako se vyvíjí OS Android, vyvíjejí se i jeho knihovny. Google Maps se snaží být pro jeho uživatele čím dál tím lepší, tím pádem je logické, že svoji snahu také ocení. Tyto knihovny byly až do nedávné doby zcela zdarma a každý vývojář je mohl ve svém projektu použít. Dnes je ale bohužel velká část jeho funkcí již placená a spoustu projektů je nuceno migrovat na mapy jiné společnosti. Alternativ, na které lze projekt přenést je nespočet, zde jsou popsány tři, které byly v projektu vyzkoušeny. V projektu nakonec zůstali pro zobrazování mapy GoogleMaps, avšak vzhledem k možnosti, že se v budoucnu i tato funkce zpoplatní, bude nezbytná migrace na jednu z těchto map. Nyní však bylo vyhodnoceno, že je stále nejlepší volbou. Pro funkce navigace bylo nezbytné použít mapy Mapbox.

Alternativní knihovny pro mapy:

Mapbox: tyto mapy jsou zdarma 50 000 dotazů denně, jsou velice jednoduché a přívětivé pro vývojáře. Oproti mapám Google sice nemají tak rozsáhlou dokumentaci, avšak naprosto dostatečnou. Tyto mapy používá například společnost Facebook, nebo Snapchat. Oproti jiným API nabízí funkci přednastavené navigace, což je velice užitečné.

Here: Tyto mapy se vyznačují především přesností, jelikož data jsou téměř denně aktualizována. Poskytují také informace o dopravě a další, které ostatní API většinou neobsahují. Zdarma je až 250 000 požadavků denně.

TomTom: Mapy TomTom ihned využili zpoplatnění Google map a vytvořili jednoduchý návod, jak migrovat projekt na jejich API. To je velkou výhodou pro vývojáře, jelikož s tímto přechodem nemusí strávit moc času. Zdarma je zde 2500 dotazů denně, nebo 75 000 měsíčně, což není moc. Mezi jeho zajímavé funkce patří například informace o dopravě, nebo zjišťování ideální trasy mezi dvěma body.

OpenStreetMaps: Tyto mapy jsou kvalitní a jednoduché. Jejich data jsou velice přesná, jelikož čerpají z mnoha zdrojů. Použití těchto map je oproti jiným zdarma, bez žádných omezení. Nemají ale tolik rozšířenou dokumentaci a nejsou tak snadné pro pochopení a použití, jako například mapy Google, proto nebyly zvoleny. [17]

11. Možná vylepšení

11.1. Vytvoření plánu dne

V projektu bylo již na počátku v plánu vytvořit třídu, kde uživatel v seznamu míst bude moci zaškrtnout místa, která chce navštívit a aplikace mu nabídne nejkratší trasu. Seřadí místa podle toho, v jakém pořadí je ideální je navštívit. Tato funkce by nejlépe měla zohlednit i otevírací dobu památek a čas, který zde obvykle lidé stráví a v případě nepřiměřeného počtu míst je rozvrhnout do více dní. Tato funkce by uživatelům ještě více ulehčila jejich návštěvu Liberce, v budoucnu bude snaha ji dokončit.

11.2. Zvětšování textu

Vzhledem k faktu, že aplikace je navržena pro všechny věkové kategorie uživatelů, je dobré myslet na to, že malé texty může mít někdo problém přečíst. Do aplikace je tedy vhodné v budoucnu přidat tlačítko pro zvětšení textu, kdy by se hodnota všech velikostí zvětšila o určitý počet bodů, nebo procento.

11.3. Rozšíření pro platformu IOS

Počet uživatelů mobilních telefonů s operačním systémem IOS v České republice neustále roste. Ve světě je to dokonce nejvíce používaný operační systém pro mobilní zařízení. Pro aplikaci je do budoucna nezbytné její rozšíření i pro tuto platformu.

11.4. Rozšíření do dalších měst

Aplikace byla navržena tak, aby bylo snadné rozšířit ji i pro další města, stačí pouze přidat další databázi se stejnými prvky a vložit data. V budoucnu je plánováno rozšíření pro města Mladá Boleslav a Turnov.

12. Závěr

V rámci této bakalářské práce byl nejprve prozkoumán trh s podobnými turistickými aplikacemi, které jsou pro platformu Android k dispozici. Bylo shledáno, že žádná z doposud nabízených aplikací není zcela ideální a neposkytuje uživateli veškeré potřebné informace. Dále byla většina z těchto aplikací zbytečně složitá na ovládání a pro uživatele tím pádem nepohodlná. Téměř žádná z těchto aplikací také nebyla vytvořena přímo pro město Liberec, není tomu tedy uzpůsobena a obsahuje pouze informace všeobecně známé. Bylo tedy zhodnoceno, že je vhodné vytvořit aplikaci novou, která bude pro uživatele vhodnější a příjemnější na používání.

Byla navržena a vytvořena mobilní aplikace nad platformou Android. Aplikace plní účel turistického průvodce městem Liberec. Nabízí uživateli informace, které byly shledány potřebnými a zajímavými pro každého turistu i obyvatele. Například informuje o místech jako jsou památky, muzea, restaurace, hotely, obchody, parkoviště nebo toalety. Ke každému místu uživatel najde potřebné informace, například otevírací dobu nebo vstupné. Uživatel má také možnost přidávat k místům vlastní fotografie. Další nabízenou možností je zobrazení míst v mapě.

Aplikace dále informuje o událostech, které se v Liberci konají nebo v budoucnu konat budou. Ke každé události, stejně jako k místu, uživatel najde veškeré potřebné informace, jako datum konání nebo vstupné.

U každé události a místa má uživatel možnost využít navigace a snadno se pomocí toho dostat kam potřebuje. Navigace je aktuálně velice jednoduchá, nabízí pouze nejkratší možnou trasu.

Další možností, kterou zde uživatel má oproti jiným podobným aplikacím jsou aktuální výhodné nabídky. Například soutěže o vstupenky nebo procentuální slevy na události. Tyto výhody zadávající firma sjednává s organizátory událostí, výhradně pro uživatele této aplikace, výměnou za doporučení této události.

Při vytváření této aplikace byl kladen důraz především na její jednoduchost a intuitivnost. Cílem bylo vytvořit co nejjednodušší aplikaci, která ale zároveň obsahuje všechny potřebné informace. V neposlední řadě bylo dbáno na vzhled aplikace, aby byla pro uživatele lákavá a příjemná na pohled.

V rámci této práce byla navržena databáze, společně s panem Radovanem Čapkem. Tato databáze je zařízena v prostředí Google Firebase a je velice jednoduchá. Byla navržena tak, aby bylo možné aplikaci snadno rozšířit pro další města. Bezpečnost vkládání dat od uživatelů je řešena pomocnými kolekcemi. Každé místo, událost nebo fotku je nutné nejprve schválit a až poté se přesune z pomocné kolekce do zobrazované.

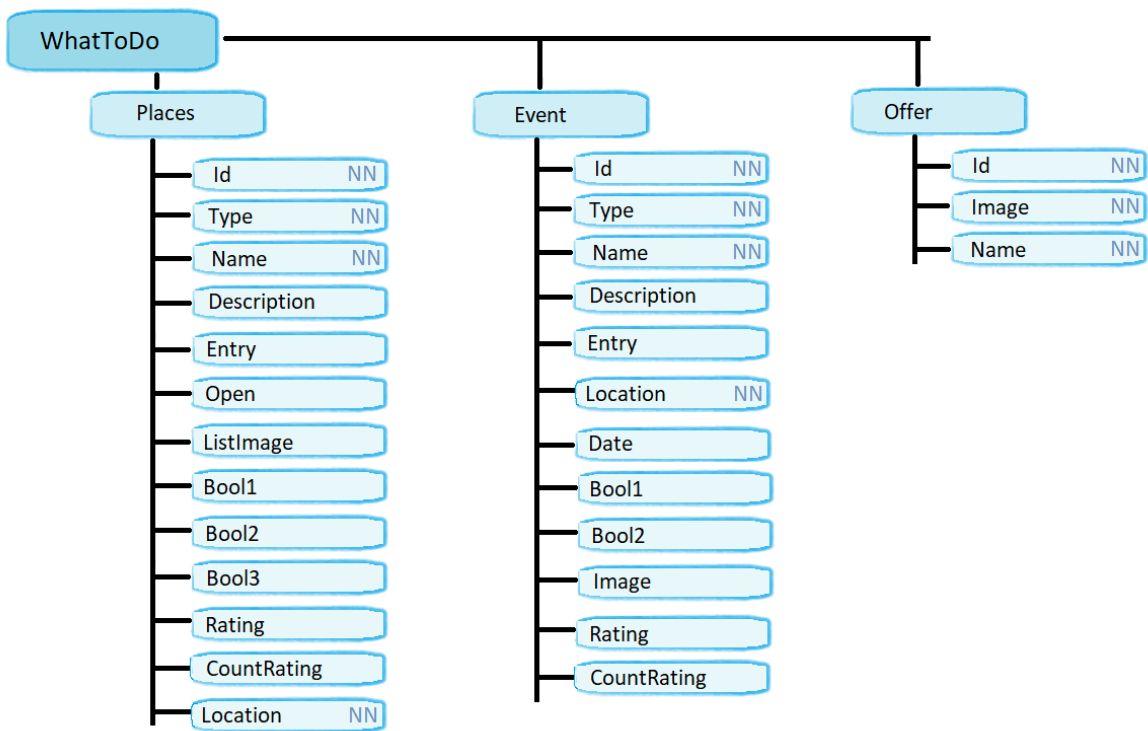
Aplikace byla po dokončení otestována u uživatelů. Uživatelé byli dotazováni především na vzhled, intuitivnost a obsáhlost aplikace. Některé jejich připomínky byly zohledněny. Veškerá zpětná vazba byla zpracována a zobrazena do grafů. Po schválení aplikace zadavatelem a následném přidání dat, bude aplikace připravena k nahrání na Google Play.

Seznam použité literatury

- [1] Parkovací místa. *Www.liberec.cz* [online]. Liberec, 2019 [cit. 2019-04-06]. Dostupné z: <https://www.liberec.cz/cz/prakticke-informace/doprava-udrzba-komunikaci/parkovaci-mista/parkovaci-mista.html>
- [2] The definitive guide to firebase. New York, NY: Springer Science+Business Media, 2017. ISBN 978-1-4842-2942-2.
- [3] LACKO, Ľuboslav. Mistrovství - Android. Přeložil Martin HERODEK. Brno: Computer Press, 2017. Mistrovství. ISBN 978-80-251-4875-4.
- [4] HERODEK, Martin. 333 tipů a triků pro Android: [sbírka nejužitečnějších postupů a řešení]. Brno: Computer Press, 2014. ISBN 978-80-251-4310-0.
- [5] Firebase. *Firestore* [online]. [cit. 2019-04-27]. Dostupné z: <https://firebase.google.com/docs/guides>
- [6] Google Maps Platform. *Developers.google* [online]. [cit. 2019-04-27]. Dostupné z: <https://developers.google.com/maps/documentation/>
- [7] Navigation SDK for Android. *Mapbox* [online]. [cit. 2019-04-27]. Dostupné z: <https://docs.mapbox.com/android/navigation/overview/>
- [8] BoomMenu. *Github* [online]. [cit. 2019-04-27]. Dostupné z: <https://github.com/Nightonke/BoomMenu>
- [9] CountdownTimer. *Developer.android* [online]. [cit. 2019-04-27]. Dostupné z: <https://developer.android.com/reference/android/os/CountDownTimer>
- [10] Bundle. *Developer.android* [online]. [cit. 2019-04-27]. Dostupné z: <https://developer.android.com/reference/android/os/Bundle>
- [11] Intent. *Developer.android* [online]. [cit. 2019-04-27]. Dostupné z: <https://developer.android.com/reference/android/content/Intent>
- [12] Toasts overview. *Developer.android* [online]. [cit. 2019-04-27]. Dostupné z: <https://developer.android.com/guide/topics/ui/notifiers/toasts>
- [13] AsyncTask. *Developer.android* [online]. [cit. 2019-04-27]. Dostupné z: <https://developer.android.com/reference/android/os/AsyncTask>
- [14] Location. *Developer.android* [online]. [cit. 2019-04-27]. Dostupné z: <https://developer.android.com/reference/android/location/Location>
- [15] LocationManager. *Developer.android* [online]. [cit. 2019-04-27]. Dostupné z: <https://developer.android.com/reference/android/location/LocationManager.html>

- [16] Vector drawables overview. *Developer.android* [online]. [cit. 2019-04-27].
Dostupné z: <https://developer.android.com/guide/topics/graphics/vector-drawable-resources>
- [17] Co je OpenStreetMap. Wiki.openstreetmap [online]. [cit. 2019-04-27]. Dostupné z: https://wiki.openstreetmap.org/wiki/Cs:Co_je_OpenStreetMap

A. Rozvržení databáze



B. Zpětná vazba od uživatelů

3.4.2019 19:50:50	31 - 50	Muž	Jiné	Určitě ano		Spíše ano		Určitě ano		Určitě ano	Trip Advisor	Nevím	21 - 50
4.4.2019 7:02:08	51 - 70	Muž	Jiné	Určitě ano		Určitě ano		Určitě ano		Určitě ano		Určitě ano	1 - 20
3.4.2019 21:31:59	21 - 30	Žena	Jiné	Určitě ano	***	Spíše ano		Určitě ano		Určitě ano		Nevím	21 - 50
12.4.2019 22:03:49	70 +	Žena	Jiné	Určitě ano	*****	Spíše ano	*////	Spíše ano	Zbytečné není nic.	Určitě ano	ne	Nevím	101 +
6.4.2019 9:18:48	21 - 30	Muž	Sociální	Určitě ano		Určitě ano		Spíše ano		Určitě ano	mapy.cz	Spíše ne	0
3.4.2019 19:48:24	21 - 30	Žena	Sociální	Spíše ano		Určitě ano		Určitě ano		Spíše ano		Spíše ne	21 - 50
11.4.2019 21:02:21	51 - 70	Žena	Sociální	Spíše ano	*/	Spíše ano		Nevím		Určitě ano	Ani ne	Spíše ne	0
3.4.2019 17:49:18	70 +	Žena	Sociální	Určitě ano		Určitě ano		Určitě ano		Spíše ano		Určitě ne	0
3.4.2019 17:28:01	21 - 30	Muž	Technické	Určitě ano		Spíše ano		Určitě ano		Spíše ano		Určitě ne	0
3.4.2019 17:35:42	21 - 30	Muž	Technické	Určitě ano		Určitě ano	Plánovač trasy	Určitě ano		Určitě ano		Určitě ne	0
3.4.2019 18:01:18	21 - 30	Muž	Technické	Určitě ano	*/	Spíše ano		Určitě ano		Spíše ano		Spíše ano	21 - 50
3.4.2019 18:04:16	21 - 30	Muž	Technické	Spíše ne	*/	Spíše ano		Spíše ano		Určitě ano		Nevím	21 - 50
5.4.2019 23:25:50	21 - 30	Muž	Technické	Spíše ano		Spíše ano		Spíše ano		Spíše ne	Ne	Spíše ne	0
6.4.2019 8:33:04	21 - 30	Muž	Technické	Spíše ano		Určitě ano		Určitě ano		Spíše ano		Určitě ne	1 - 20
6.4.2019 13:40:49	21 - 30	Muž	Technické	Spíše ne	****	Spíše ano	*****	Spíše ano	*****	Spíše ne	Ani ne	Spíše ne	1 - 20
6.4.2019 14:03:01	21 - 30	Muž	Technické	Určitě ne		Určitě ne		Nevím		Spíše ne		Určitě ne	0
6.4.2019 18:09:34	21 - 30	Muž	Technické	Určitě ano		Spíše ano		Určitě ano		Určitě ano		Spíše ano	1 - 20
7.4.2019 11:15:58	21 - 30	Muž	Technické	Spíše ne	dostupné funkce	Spíše ano		Spíše ano	*****	Určitě ano	nope	Spíše ano	1 - 20
5.4.2019 18:51:25	31 - 50	Muž	Technické	Určitě ano		Spíše ano		Určitě ano		Spíše ano		Určitě ano	21 - 50
3.4.2019 17:36:29	21 - 30	Žena	Technické	Nevím		Spíše ano		Určitě ano		Nevím		Spíše ne	1 - 20

3.4.2019 19:52:22	31 - 50	Žena	Technické	Spíše ne	*	Určitě ano		Určitě ano	**	Spíše ano	TimeOut	Určitě ne	0
5.4.2019 18:59:39	31 - 50	Žena	Technické	Určitě ano		Spíše ano		Určitě ano		Spíše ano		Spíše ano	1 - 20
12.4.2019 6:55:57	70 +	Žena	Žádné	Určitě ano		Srozumitelné	Spíše ano	Určitě ano	Nic	Určitě ano	Jinou neznám	Spíše ano	21 - 50

*Líbí se mi cíl aplikace, funkce, intuitivní pro konečného uživatele a informovanost. Moc se mi nelíbí design. Změnila bych font na menší, tenčí a nezakulacený. Nerotovala bych názvy, to je zastaralý způsob, zmenšením nadpisů se to tabulky vejde celý název. Vyhnout se dvojitým tabulkám či silnému stínu, preferuji "flat" design pro aplikace, případně pouze jemný stín. Nadpisy vedou příliš do okrajů (např. Velikonoční běh), měly by být menší. Také bych nekontrastovala bílou a černou, ale zjemnila bych to na tmavě šedou na pozadí a světle šedou pro nadpis. Žluto zelená barva v aplikaci se mi líbí, ale malinko bych ji ztmavila pro lepší kontrast. Náhledy fotek a log v seznamu míst bych roztáhla kompletně do stran tak, aby vyplňovali celý levý roh dané tabulky. Filtry bych asi seřadila vedle sebe do roviny. Menu tabulky by měly být v jedné barvě. Tabulku "O aplikaci" nepovažuji pro konečného uživatele relevantní, to bych odsunula asi někam jinam. Stránka při spuštění: zde je opět nadpis rotažený příliš do stran, zmenšila bych to. Změnila bych písmo pro in Liberec na jednoduché moderní písmo (např. Montserrat, Avenir...).

** Možná bych přidala k jednotlivým událostem proklik na koupi lístku, ať už oficiální web události či prodejce lístků... aby to lidi nemuseli hledat zvlášť. Jinak je vše tak akorát, není to přehlcené informacemi, což je ok.

***Vše se mi líbí, pěkné nadčasové logo, příjemná barevnost aplikace, blbuvzdorna aplikace

**** Vypadá staromódně. Uvítal bych něco moderního co dokáže zaujmout a nalákat očima, jelikož první dojem je nejdůležitější. velké obrázky, rolovací stránka, parallax efekt... prpště dynamika. ;-)

***** Možnosti komentářů jednotlivých míst/akcí. (pokud ta možnost je, tak jsem si ji nevšiml) Dále třeba sdílení očekávaných akcí s přáteli, nebo možnost přidání vlastních fotografií. Doplnit předpokládané počasí z nějakého zdroje na nadcházející akci.

***** Osobně jen nemám rád to hamburger menu, ale pokud to půjde nahradit nějakým gestem, tak s tím nemám problém a většině lidí to asi vadit nebude.

***** úvodní stránku, kde by mohli být samostatně doporučené události bez minimapy

***** Pokud nepřijedu autem, potřebuji se orientovat v městské dopravě, u detailu restaurace bych chtěla vědět, jestli tam mohu vzít psa, a pro cizince možná angličtina?

***** Líbí se mi nabídka aktuálních zajímavostí

*/ Nelíbí se mi moc barevnost

*// Líbí se mi menu

*/// líbí se mi vše, hlavně myšlenka

*//// Volba barev, velikost nápisů

*///// Cenovou nabídku např. parkoviště

*///// jednoduché ovádání, barevné provedení