

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Geolokační aplikace pro Android s využitím gamifikace
Diplomová práce

Autor: Bc. Michael Adamec
Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Ing. Filip Malý, Ph.D.

Hradec Králové

září 2016

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 8.8.2016

Bc. Michael Adamec

Poděkování:

Děkuji vedoucímu diplomové práce doc. Ing. Filipu Malému, Ph.D. za metodické vedení práce, cenné rady a odborný dohled.

Anotace

Diplomová práce se zabývá analýzou, návrhem řešení a implementací geolokační aplikace pro operační systém Android. V úvodu seznamuje s vývojem navigace od jejího počátku po současnost. Další část detailně popisuje způsoby určování geografické polohy mobilního zařízení. Je zde popsán princip, jak funguje lokalizace telefonu pomocí GSM sítě, bezdrátových Wi-Fi sítí a GPS. Následně je představena platforma Android, pro kterou byla mobilní aplikace vyvíjena. Hlavní část práce popisuje návrh a realizaci aplikace včetně problémů, které byly během implementace řešeny. Závěrečná část je věnována důkladnému testování, jehož závěry byly promítnuty do výsledné podoby aplikace. Výstupem této práce je plně funkční geolokační aplikace a webové aplikační rozhraní, které zajišťuje její funkcionalitu.

Annotation

Title: Geolocation application for Android using gamification

This diploma thesis deals with the analysis, designing the solution and implementation of geolocation application for the Android operating system. The development of navigation from its beginning to the present is described in the introduction of the paper. The next section explains the methods for determining a geographic position of a mobile device in detail. The principles of the phone localization using GSM network, wireless Wi-Fi networks and GPS are described there. The following chapter introduces the Android platform which the mobile application was developed for. The main part of this thesis describes the design and realization of application, including the problems solved during the implementation. The final section is dedicated to thorough testing whose conclusions were incorporated into the final version of application. The outcome of this work is completely functional geolocation application and web application interface that ensures its functionality.

Obsah

1	Úvod.....	1
2	Historie navigace.....	3
3	Způsoby určení polohy mobilního zařízení.....	7
3.1	Určení polohy pomocí GSM sítě.....	7
3.1.1	Struktura GSM sítě	7
3.1.2	Způsoby určení polohy	8
3.1.2.1	Cell of Origin (COO).....	9
3.1.2.2	Timing Advance (TA)	10
3.1.2.3	Angle of Arrival (AoA).....	11
3.1.2.4	Trilaterace	11
3.1.2.5	Radio Fingerprinting (RF)	13
3.1.3	Shrnutí určování polohy pomocí sítě GSM.....	13
3.2	Určení polohy pomocí Wi-Fi sítí.....	14
3.3	GPS	17
3.3.1	Struktura systému	17
3.3.1.1	Kosmický segment.....	18
3.3.1.2	Řídící a kontrolní segment	18
3.3.1.3	Uživatelský segment	20
3.3.2	Princip určení polohy.....	20
3.3.3	Přesnost systému GPS.....	23
3.3.4	Shrnutí určování polohy pomocí GPS	23
4	OS Android	25
4.1	Stručná historie.....	25
4.2	Fragmentace OS Android.....	26
4.3	Architektura	29

4.3.1	Jádro (Kernel)	29
4.3.2	Knihovny (Libraries).....	30
4.3.3	Běhové prostředí (Android Runtime)	30
4.3.4	Aplikační framework (Application Framework).....	31
4.3.5	Aplikace (Applications).....	31
5	Požadavky na aplikaci	32
6	Návrh a implementace	34
6.1	Webové aplikační rozhraní (API).....	34
6.1.1	Popis funkcionality	34
6.1.2	Návrh aplikace	34
6.1.2.1	Databáze	35
6.1.2.2	REST API.....	37
6.1.3	Implementace	39
6.2	Mobilní aplikace pro Android.....	41
6.2.1	Popis funkcionality	41
6.2.2	Návrh aplikace	43
6.2.3	Implementace	47
6.2.3.1	Použité technologie.....	48
6.2.3.2	Struktura aplikace	48
6.2.3.3	GPS poloha.....	50
6.2.3.4	Geofencing	52
6.2.3.5	Podvržení GPS polohy	53
6.2.3.6	Proces aktivace památky	54
6.2.3.7	Synchronizace profilu uživatele	57
6.2.3.8	Šetření mobilních dat	58
7	Testování aplikace a výsledky.....	60

8	Závěr.....	65
9	Seznam použité literatury.....	68
10	Přílohy.....	72

Seznam obrázků

Obr. 1: Evropa z Katalánského atlasu	4
Obr. 2: Buňková síť GSM	8
Obr. 3: Směrové antény v síti GSM	8
Obr. 4: Metoda Cell of Origin	9
Obr. 5: Metoda Cell of Origin + Timing Advance.....	10
Obr. 6: Metoda Angle of Arrival.....	11
Obr. 7: Metoda trilaterace.....	12
Obr. 8: Rozmístění stanic řídicího segmentu GPS	19
Obr. 9: Generování C/A kódu družic a GPS přijímačem	21
Obr. 10: Princip určení polohy pomocí GPS	22
Obr. 11: Prodej mobilních telefonů v Q1 2016.....	25
Obr. 12: Fragmentace verzí OS Android, květen 2016.....	27
Obr. 13: Diagram architektury OS Android	29
Obr. 14: Databázový model serverové části aplikace	35
Obr. 15: Use Case diagram mobilní aplikace	43
Obr. 16: Volání aktivit - přihlášení do aplikace.....	44
Obr. 17: Volání aktivit - hlavní část aplikace.....	46
Obr. 18: Postranní nabídka	47
Obr. 19: Geofence události	52
Obr. 20: Proces aktivace památky	55
Obr. 21: Herní mapa.....	64
Obr. 22: Detail památky.....	64

Seznam tabulek

Tabulka 1: Přehled IEEE 802.11 standardu	15
Tabulka 2: Počet a síla signálu Wi-Fi v okolí vybraných památek.....	61

1 Úvod

Mobilní telefon již dávno není pouhým nástrojem na volání a posílání SMS zpráv. Podobně jako člověk, jsou dnešní mobilní telefony schopny přijímat a zpracovávat informace z okolního prostředí. Člověk pro vnímání okolních informací využívá pět základních smyslů, kterými jsou čich, chuť, hmat, sluch a zrak. Do telefonu jsou pro tyto účely přidávány různé druhy senzorů.

Jedním ze základních senzorů naprosté většiny současných mobilních telefonů je akcelerometr. Ten slouží k určení relativní polohy v prostoru. Aplikace díky němu například poznají, zda je telefon orientován na šířku či na výšku a mohou upravit rozložení grafických prvků na obrazovce. Dalším často používaným senzorem je elektronický kompas. Tento snímač je schopný detekovat magnetické pole Země a dokáže tak určit světové strany. Aplikace tento senzor využívají nejen k určení severního pólu, ale například i k detekci kovu. Mezi další, ne už tak časté senzory patří senzor přiblížení a senzor intenzity okolního osvětlení. První z nich je schopný po přiblížení telefonu k uchu zhasnout displej a vypnout dotykovou vrstvu, aby nedošlo k nechtěné interakci s telefonem. Druhý ze senzorů umožňuje na základě intenzity okolního osvětlení upravit jas displeje tak, aby informace na displeji byly pro uživatele neustále pohodlně čitelné a zároveň, aby se neplýtvalo elektrickou energií, které je u mobilních telefonů stále nedostatek.

Hlavním senzorem u dnešních mobilních telefonů je však senzor pro určení geografické polohy. Většina zařízení obsahuje čip amerického systému GPS či ruského GLONASS nebo jejich kombinaci. Informace o geografické poloze jsou jedny z nejcennějších dat, které může programátor mobilní aplikace o uživateli získat. Lze je využít například pro navigaci uživatele po mapě nebo pro zaslání GPS souřadnic záchranné službě při dopravní nehodě.

Cílem této práce je nahlédnout do problematiky určování geografické polohy na mobilním zařízení. V úvodu práce je stručně popsán vývoj navigace od jejího počátku po současnost. Dále jsou detailně představeny aktuální způsoby určování

polohy na mobilních telefonech. Součástí diplomové práce je i plně funkční geolokační aplikace s herními prvky pro operační systém Android, která využívá popsané způsoby lokalizace mobilního zařízení. Hlavní část práce se tak soustředí na popis návrhu, implementace a testování naprogramované mobilní aplikace. Ta si klade za cíl zábavným způsobem provést uživatele po historických památkách a dalších zajímavých místech ve městě.

2 Historie navigace

Při zpracovávání této kapitoly bylo čerpáno ze zdrojů [1], [2] a [3].

Historie navigace se začala psát v době, kdy člověk začal cestovat mezi dvěma navzájem vzdálenými místy. Lidé potřebovali umět určit svou aktuální polohu, aby mohli zkontrolovat, zda se drží předem vytyčené trasy, která je měla rychle a bezpečně dovést k cíli. Na souši toto většinou nebyl problém. Ve zmapovaném terénu bylo zpravidla možné určit dostatečný počet orientačních bodů (kameny, stromy, hory, řeky, atd.) a pohybovat se podle nich.

Lidstvo však brzy zjistilo, že pohyb po vodě je mnohem snazší a rychlejší než pohyb po souši. Lodní doprava významně usnadnila přepravu osob a zboží na velké vzdálenosti a později i mezi kontinenty. První lodě, které byly schopné přepravovat náklad, jsou známy z doby přibližně 3500 let př.n.l. Mapy v té době ještě neexistovaly. První plavby tak byly vedeny při pobřeží, aby se mořeplavci mohli orientovat podle známých orientačních bodů na pevnině (útesy, zálivy, baje, atd.). Plavbu bylo možné provádět pouze ve dne a za dobré viditelnosti.

Jakmile se však námořníci se svými loděmi odpoutali od pevniny a vydali se na otevřenou moře, potřebovali najít jiný způsob, jak určit polohu plavidla a směr, kterým mají plout. Posádka byla schopna určit zeměpisnou šířku na základě výšky Slunce, resp. Polárky nad obzorem. Zeměpisnou délku pouze odhadovali z rychlosti a doby plavby. Čas měřili pomocí přesýpacích hodin a rychlost odvodili z pozorování vzájemného pohybu lodi a různých předmětů plovoucích na hladině. Přesto se Vikingové okolo roku 1000 n.l. odvažovali plout na Island či dokonce až do Ameriky.

Významného pokroku v navigaci bylo dosaženo ve 13. století zavedením magnetického kompasu. Z počátku byl používán pouze pro stanovení směru větru nebo při špatném počasí, kdy nebylo přes mraky vidět na oblohu. Tehdejší námořníci ale nechovali ke kompasu moc velkou důvěru. Nedokázali si vysvětlit odchylky, které se při měření kompasem objevovaly. Nebyla jim totiž známa

skutečnost, že kompas ukazuje k magnetickému a ne k zeměpisnému severu. Mnohem hodnotnějším nástrojem byla v té době obyčejná olovnice, která se používala k měření hloubky mořského dna. Instrukce pro plavbu po známých trasách pak obsahovaly také informace o hloubce pobřežních vod. To vedlo k dalšímu zpřesnění určování polohy lodi. Společně se sílící námořní dopravou začaly na přelomu 13. a 14. století postupně vznikat námořní nákresy, nazývané portolánové mapy. Na mapě byla zakreslena především mořská pobřeží, přístavy, zálivy a zátoky. Portolánové mapy neobsahovaly ze začátku jejich tvorby geografickou síť. Ta se na mapách objevila až později. Místo toho byly na mapě nakresleny větrné růžice, které znázorňovaly směr plavby od jednoho přístavu k druhému. Na obrázku č. 1 je vidět Evropa z Katalánského atlasu, který je považován za nejvýznamnější portolánovou mapu 14. století.



Obr. 1: Evropa z Katalánského atlasu

Zdroj: https://upload.wikimedia.org/wikipedia/commons/b/b1/Europe_Mediterranean_Catalan_Atlas.jpeg

Období od začátku 15. století do poloviny 16. století se označuje za věk zámořských objevů. Během těchto 150 let byl uskutečněn velký počet zámořských výprav, které přinesly mnoho nových zeměpisných objevů. Nové poznatky přispěly mimo jiné k přesnějšimu vypracování map. V roce 1415 se podařilo Portugalcům dobýt město Ceuta na pobřeží severní Afriky, čímž získali první kolonii mimo evropské území. Portugalci velmi rychle pochopili, že Afrika je kontinent bohatý na slonovinu, zlato i jiné obchodní artikly. Bylo tedy vypraveno několik objevitelských plaveb, které napomohly k poznání dalších částí Afriky. V roce 1488 dokázali Portugalci obeplout mys Dobré naděje, který Evropě otevřel cestu na východ do Indie. V této době se o zámořské plavby pokoušely i jiné přímořské státy v čele se Španělskem. Vše vyvrcholilo 12. října 1492, kdy Krištof Kolumbus objevil nový světadíl. Objevení Ameriky se často považuje za symbolický počátek novověku [4].

V 15. století se také začaly objevovat první astronomické přístroje pro měření úhlu Slunce a hvězd nad obzorem. Patří mezi ně Jakobova hůl, Davisův kvadrant a námořní astroláb, který byl později nahrazen sextantem. Ačkoliv metody pro určování zeměpisné šířky byly neustále zdokonalovány, přesně určit zeměpisnou délku bylo pro námořníky nadále obtížné. K významnému pokroku došlo až v 16. století se zavedením zařízení zvaného chip log, pomocí kterého byla měřena rychlost lodi. Jednalo se o dřevěnou desku navázanou na provaz, na kterém byly rozmístěny uzlíky v pravidelných vzdálenostech. Po vhození do moře zůstal chip log odporem vody na místě a navázaný provaz s uzlíky se začal rychle odvíjet. Rychlost lodi se pak určila podle toho, kolik uzlíků se odvinulo za daný časový interval. Od té doby se rychlost plavby na moři udává v uzlech.

Počátkem 18. století byly vytvořeny mapy magnetických deklinací. Tím byla výrazně zlepšena navigace pomocí magnetického kompasu. K dalšímu zpřesnění určování zeměpisné délky došlo společně s vynálezem námořního chronometru, který udržoval čas výchozího bodu plavby a spolu s astronomicky měřeným místním časem pomáhal určit rozdíl zeměpisných délek obou míst. James Cook používal námořní chronometr při svých plavbách kolem světa a i díky němu

vytvořil velmi detailní a přesné mapy, které významným způsobem ovlivnily další vývoj navigace.

Moderní metody navigace se začaly rozvíjet ve 20. století v souvislosti s rozvojem elektrotechniky a elektroniky. Velkou zásluhu na tom má Guglielmo Marconi, který položil základy přenosu informací na velkou vzdálenost pomocí rádiových vln. Princip je založen na měření času, za který dorazí rádiový signál od vysílače k přijímači. Pokud se vynásobí čas, za který signál tuto vzdálenost urazí, rychlostí vysílaného signálu, dostaneme vzdálenost mezi vysílačem a přijímačem. Rychlost šíření rádiových vln v prostoru je přibližně rovna rychlosti světla ve vakuu, tj. 300 000 km/s.

Na počátku 40. let 20. století byl vyvinut ve Spojených Státech pozemní rádiový navigační systém se zkratkou LORAN (Long Range Navigation). Vysílací stanice s dosahem až 2 400 km byly stavěny v párech při pobřeží. Systém byl založen na principu vysílání pulzního rádiového signálu, kde na základě naměřeného časového rozdílu mezi přijetím signálu ze dvou různých vysílačů bylo možné určit polohu přijímače. LORAN byl využíván především námořnictvem a tomu odpovídalo i malé pokrytí signálem ve vnitrozemí. Vznikla tedy myšlenka umístit vysílače na oběžnou dráhu Země, aby bylo možné určit geografickou polohu kdekoliv na světě.

Začátkem 60. let uvedlo ministerstvo obrany Spojených států do provozu první družicový polohový systém s názvem Transit. Systém byl primárně určen pro navigaci letadlových lodí a jaderných ponorek. Později byl však využíván i v civilní námořní dopravě a v oboru geodézie. Družice systému Transit vysílala navigační signál na dvou frekvencích (150 MHz a 400 MHz). Poloha se určila na základě měření Dopplerova posunu rádiového signálu. Zkušenosti se systémem Transit vedly k vyvinutí současných globálních družicových polohových systémů.

3 Způsoby určení polohy mobilního zařízení

Jednou z unikátních vlastností dnešních mobilních aplikací je možnost zjistit geografickou polohu uživatele. Jedná se o jedny z nejcennějších dat, která může programátor o uživateli získat. Tato data pak mohou být využita například k navigaci uživatele po mapě, napovídání nejbližších zastávek při vyhledávání dopravního spojení, sdílení polohy s přáteli na sociálních sítích nebo k předpovědi počasí pro město, ve kterém se uživatel právě nachází. Existují různé způsoby, jak určit polohu mobilního zařízení. Liší se především v přesnosti, rychlosti určení polohy, množství spotřebované energie a prostředí, ve kterém je daný způsob možné použít. V této kapitole budou představeny tři metody, které se používají nejčastěji při určování polohy mobilního zařízení ve venkovním prostředí.

3.1 Určení polohy pomocí GSM sítě

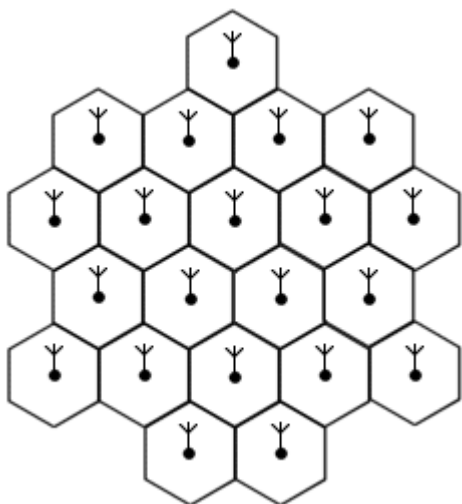
GSM (Global System for Mobile Communications) je, jak už zkratka napovídá, globální systém pro mobilní komunikaci, který je v současnosti využíván ve více než 200 zemích po celém světě [5].

Je dobré si uvědomit, že kromě programátorů mobilních aplikací se snaží určit polohu uživatele pomocí GSM sítě i samotní mobilní operátoři. Dobrým příkladem, proč potřebuje mobilní operátor mít možnost určit polohu koncového zařízení, je volání na tísňovou linku. Ačkoliv technologie lokalizace volání z mobilních telefonů neumožňuje zcela přesné určení místa, odkud volání přichází, v kombinaci s dalšími informacemi může významně přispět k úspěšné lokalizaci volajícího a následnému vyslání záchranného týmu na správné místo [6].

3.1.1 Struktura GSM sítě

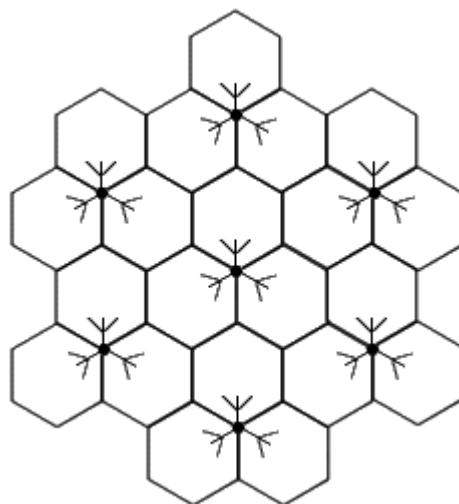
Struktura GSM sítě je značně složitá, aby mohla poskytovat veškeré vyžadované služby. Pro určení polohy mobilního telefonu je však nejdůležitější subsystém základnových stanic (BSS – Base Station Subsystem) a samotné koncové mobilní zařízení, které se pomocí základnových převodních stanic (BTS – Base Transceiver Station) připojuje do sítě mobilního operátora. Základnové stanice jsou rozmístěny

po celé ploše, kterou chce operátor pokrýt mobilním signálem. Tato plocha je rozdělena na buňky. Jednu buňku obsluhuje právě jedna BTS, která je umístěna v jejím středu (obr. 2). Aby se však nemuselo stavět velké množství základnových stanic, umísťují se většinou BTS do společných bodů tří sousedních buněk, do kterých je následně vysílán mobilní signál pomocí směrových antén (obr. 3). Poté se mluví o tzv. sektorech základnové stanice. Velikost jednotlivých buněk je závislá na počtu koncových zařízení v dané oblasti, hustotě provozu a okolním terénu. Podle toho se následně volí výška, výkon a typ antény. Vysílače jsou schopné pokrýt území od několika stovek metrů až po desítky kilometrů. Největší vzdálenosti, které lze teoreticky dosáhnout dle specifikace GSM, je 35 km [7, 8].



Obr. 2: Buňková síť GSM

Zdroj: Převzato z [8]



Obr. 3: Směrové antény v síti GSM

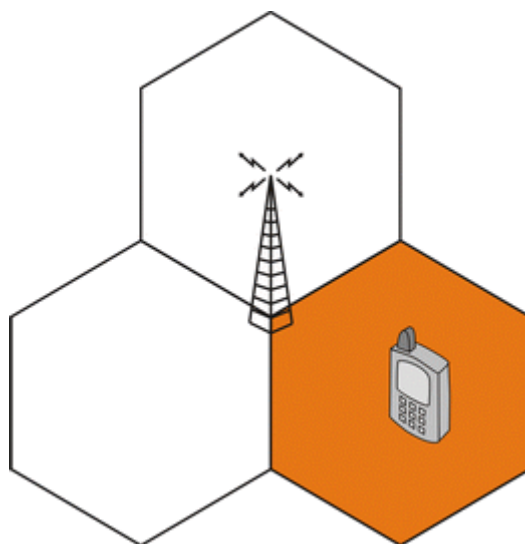
Zdroj: Převzato z [8]

3.1.2 Způsoby určení polohy

V této podkapitole budou představeny základní způsoby určení polohy mobilního telefonu v GSM síti. Všechny metody vycházejí v principu z informací, které lze získat z komunikace mobilního telefonu se základnovými stanicemi. Liší se především v přesnosti výsledné polohy v závislosti na složitosti použitého systému [7, 9, 10, 11, 12, 13].

3.1.2.1 Cell of Origin (COO)

Základním principem této metody je odvození polohy mobilního telefonu od polohy BTS. Přitom je využito buňkové architektury sítě GSM. Mobilní telefon komunikuje vždy právě s jednou základnovou stanicí. Tu je možné identifikovat na základě tzv. Cell ID, které je pro všechny stanice v dané oblasti unikátní. Pokud je známa přesná geografická poloha BTS, se kterou telefon aktuálně komunikuje, je možné určit buňku (příp. sektor), ve které se koncové zařízení nachází. Přesnost této metody je závislá na velikosti dané buňky. Může se pohybovat od 200 metrů, v hustě obydlených oblastech, až do 35 kilometrů, což je maximální teoretický dosah vysílače dle specifikace GSM. Jedná se o nejstarší a nejjednodušší metodu pro určení pozice mobilního zařízení.



Obr. 4: Metoda Cell of Origin

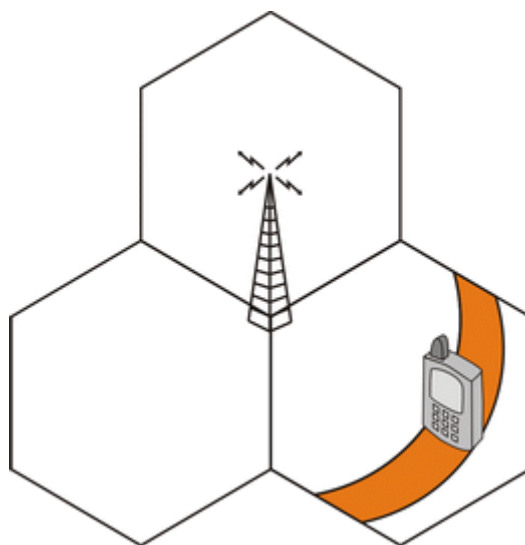
Zdroj: http://access.feld.cvut.cz/image/200602281325_fig1.gif

Je jasné, že mobilní operátor zná přesnou polohu všech svých základnových stanic s jejich vysílacím výkonem a předpokládaným dosahem. Pomocí této metody tak dokáže velmi jednoduše určit přibližnou polohu hledaného koncového zařízení. Programátor mobilních aplikací ale potřebuje k určení polohy mobilního telefonu rozsáhlou databázi základnových stanic, která obsahuje jejich geografické souřadnice. Největším katalogem zaměřených BTS je komerční služba LocationAPI,

kteřá mapuje vysíláče ve většině zemí světa. Její bezplatnou alternativou je například služba OpenCellId.

3.1.2.2 Timing Advance (TA)

Tato metoda vychází z předešlého způsobu určení polohy, který dále zpřesňuje. Na jednotlivých kanálech sítě GSM se ve vysílání periodicky střídá více mobilních telefonů. Jednotlivé telefony musejí vysílat tak, aby jejich signál dospěl k základnové stanici ve správném čase a nekolidoval s vysíláním dalšího telefonu v řadě. Závažnost tohoto problému narůstá v okamžiku, kdy je mobilní telefon vzdálen od BTS natolik, že nelze zanedbat dobu šíření rádiových vln. Proto se pro každý telefon v síti GSM kalibruje hodnota TA (Timing Advance – časový předstih). Základnová stanice změří zpoždění příchozího signálu a spočítá hodnotu TA, kterou odešle mobilnímu telefonu. Na základě této hodnoty pak mobilní telefon vysílá signál v takovém předstihu, aby dorazil k BTS ve správný okamžik. Parametr TA nabývá hodnot v rozmezí od 0 do 63 a je zvyšován o 1 každých 550 metrů. Odtud plyne maximální možná vzdálenost mobilního telefonu od BTS, která činí $63 \times 550m \cong 35km$. Oproti předešlé metodě je tedy možné, na základě hodnoty TA, určit i přibližnou vzdálenost mobilního telefonu od základnové stanice v rámci GSM buňky.

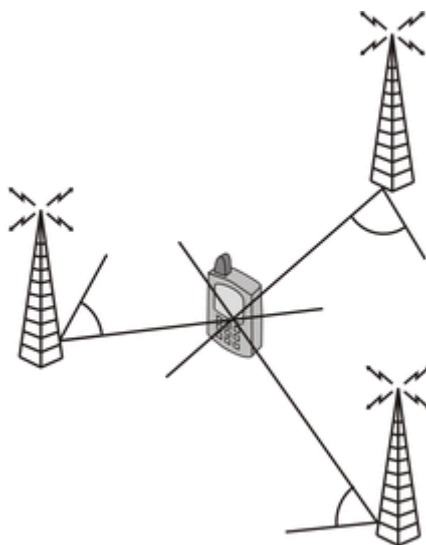


Obr. 5: Metoda Cell of Origin + Timing Advance

Zdroj: http://access.feld.cvut.cz/image/200602281326_fig2.gif

3.1.2.3 Angle of Arrival (AoA)

Pro použití této metody musí být základnové stanice dodatečně upraveny. Je totiž potřeba, aby byla BTS schopna změřit úhel, pod kterým je přijímán signál od mobilního telefonu. Pro určení 2D polohy teoreticky stačí změřit úhel přijímaného signálu pouze na 2 základnových stanicích. Přesnost takto vypočítané polohy by však byla příliš malá. Proto se reálně provádí měření na 3 a více BTS.



Obr. 6: Metoda Angle of Arrival

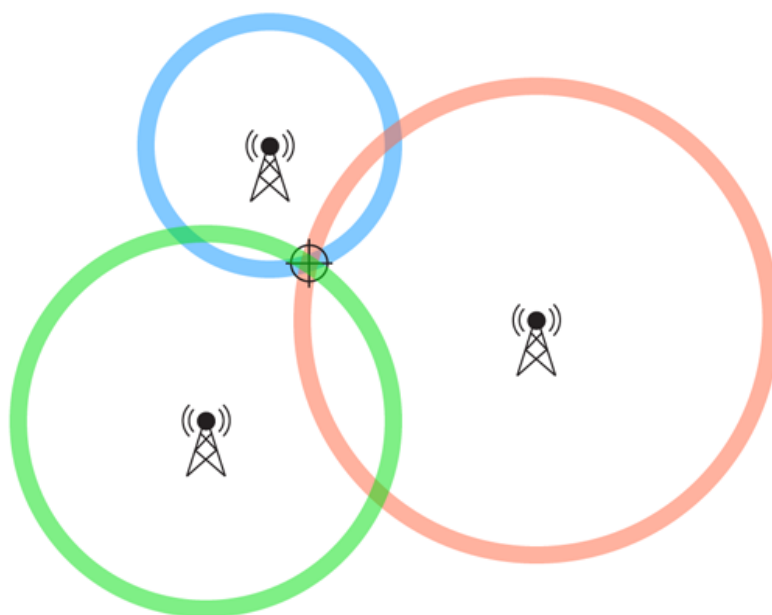
Zdroj: http://access.feld.cvut.cz/image/200602281326_fig4.gif

Princip výpočtu polohy je vidět na obr. číslo 6. Od každé BTS je vedena přímka ve směru, odkud dorazil signál mobilního telefonu. Takto vedené přímky vytvoří průsečík (v ideálním případě jeden), který udává polohu koncového zařízení. Ve městě lze touto metodou určit polohu s přesností na 100 - 200 metrů. Mimo město se pak přesnost sníží na několik kilometrů. Tato metoda lokalizace byla navržena primárně pro mobilní operátory. Měřit úhly signálu na mobilním telefonu by bylo značně složité a vyžadovalo by další vybavení.

3.1.2.4 Trilaterace

V současnosti jedním z nejpřesnějších způsobů určení polohy mobilního telefonu pomocí sítě GSM je tzv. trilaterace. Jedná se o geometrickou metodu, která umožňuje vypočítat polohu mobilního telefonu na základě vzdálenosti k více

základnovým stanicím. Pomocí trilaterace se tedy hledá průsečík kružnic, kde středem kružnice jsou souřadnice BTS a poloměr je vzdálenost základnové stanice od mobilního telefonu. Při výpočtu pouze s dvěma BTS dostaneme dva možné body, kde se může mobilní telefon nacházet (průsečíky dvou kružnic). Proto je třeba, pro jednoznačné určení polohy, přidat do výpočtu minimálně ještě jednu základnovou stanici, abychom dostali právě jeden průsečík (viz obr. 7).



Obr. 7: Metoda trilaterace

Zdroj: <http://www.neilson.co.za/wp-content/uploads/2013/10/triangulation.png>

Určit přesně vzdálenost mobilního telefonu od základnové stanice je však velmi obtížné. Lze k tomu využít hodnotu TA nebo sílu signálu. Tyto údaje jsou ale velmi nepřesné. Další možností je vzdálenost odvodit na základě znalosti rychlosti šíření rádiových vln. K tomu je potřeba změřit dobu cesty signálu od vysílače k přijímači. Přesnost měření je velmi důležitá, protože chyba 1 mikrosekundy v měřeném čase způsobí chybu 300 metrů ve vzdálenosti. Rádiové signály se navíc mohou v hustě zastavěných oblastech snadno odrážet a lámat. Díky tomu jsou pak naměřené vzdálenosti delší než ty, které ve skutečnosti signál urazí.

Existuje několik způsobů implementace trilaterace, které mohou operátoři nasadit do svých mobilních sítí. Liší se především tím, kde probíhá měření vzdáleností

a jaké prvky sítě musí mít navzájem synchronizovaný přesný čas. U metody zvané Time of Arrival (ToA) probíhá výpočet na straně operátora. Sběrem informací o vzdálenosti koncového zařízení jsou pověřeny základnové stanice. Tento princip ovšem vyžaduje synchronizaci přesného času u všech BTS, které provádějí měření. To by bylo pro mobilního operátora velmi nákladné za cenu jen malého zvýšení přesnosti polohy ve městě oproti metodě COO + TA. Proto se tato technologie téměř nepoužívá. Další možností je přesunout měření vzdáleností na koncové zařízení. Tento způsob implementace se nazývá E-OTD (Enhanced-Observed Time Difference). Ve městě se přesnost určení polohy touto metodou pohybuje mezi 50 a 125 metry.

3.1.2.5 Radio Fingerprinting (RF)

Metoda „otisku signálu“ spočívá ve vytvoření databáze otisků určité charakteristiky signálu, která je závislá na vzájemné poloze mobilního telefonu a BTS. Touto charakteristikou může být například síla mobilního signálu. Během lokalizace odešle mobilní telefon síly signálu všech okolních BTS. Tento otisk se porovná s databází a telefonu se vrátí odpovídající geografické souřadnice, kde byly podobné charakteristiky naměřeny. Základním problémem této metody je vytvoření databáze otisků. V zásadě existují dvě možnosti. První z nich je vytvořit databázi měření. Je však nutné v pravidelných intervalech fyzicky naměřit sílu mobilního signálu jednotlivých základnových stanic a následně tato měření společně s aktuální polohou uložit. Druhou možností je pak získání databáze pomocí empirického modelu šíření. Toto řešení však přináší horší výsledky. Velkou výhodou metody otisků je, že její přesnost není snížena odražením a lámáním rádiových vln o budovy ve městě. Problém však nastane například při nové výstavbě, díky které se změní otisky v dané oblasti a je tak nutné tuto oblast znovu fyzicky přeměřit. Ve městě je možné pomocí této metody určit polohu mobilního telefonu s přesností okolo 40 metrů.

3.1.3 Shrnutí určování polohy pomocí sítě GSM

Díky svému globálnímu rozšíření je GSM síť ideální prostředek pro určení geografické polohy mobilního telefonu. Polohu může určovat jak programátor,

s pomocí databáze základnových stanic, tak i samotný mobilní operátor, který tuto funkčnost využívá primárně k lokalizaci volajícího na tísňovou linku. Hlavní výhodou oproti globálním družicovým polohovým systémům je možnost určit polohu uvnitř budov a v hustě zastavěných oblastech. Zároveň je možné získat souřadnice telefonu do několika sekund za cenu jen malé spotřeby elektrické energie. Na druhou stranu tento způsob funguje pouze tam, kde je mobilní signál. Další nevýhodou je proměnlivá přesnost, která je závislá na počtu okolních základnových stanic a velikosti GSM buňky, ve které se mobilní telefon nachází. Ve městě je možné v nejlepším případě určit polohu s přesností 50 metrů. Mimo město přesnost klesne až na několik kilometrů. Tento způsob tak není vhodný pro aplikace, které potřebují určit polohu uživatele s přesností na metry (např. navigace pro auto nebo aplikace pro zaznamenávání délky výletu na kole). Dobrým příkladem, kde lze GSM lokalizaci využít, je aplikace pro předpověď počasí, která stahuje data na základě oblasti, ve které se uživatel aktuálně nachází. Pro tento případ je přesnost určené polohy více než dostatečná a není tak nutné plýtvat elektrickou energií na lokalizaci pomocí GPS, která by navíc mohla být časově náročnější.

3.2 Určení polohy pomocí Wi-Fi sítí

V roce 1997 byl mezinárodní neziskovou organizací IEEE (Institute of Electrical and Electronics Engineers) vyvinut průmyslový standard IEEE 802.11 pro přenos dat v lokálních bezdrátových sítích, který je v dnešní době známý především pod pojmem Wi-Fi. Tento standard je od té doby postupně rozšiřován a modernizován pomocí tzv. dodatků. Nejdůležitější z nich jsou uvedeny v tabulce č. 1. Liší se především v dosažitelné maximální rychlosti a pásmu, ve kterém se bezdrátová síť vysílá. Nejčastěji se používají frekvence okolo 2,4 GHz a 5 GHz. Jak je možné vidět v tabulce, první generace bezdrátových sítí pracovala na frekvenci 2,4 GHz s propustností pouze 2 Mbit/s. K hromadnému rozšíření Wi-Fi sítí došlo až v roce 2003 s příchodem standardu IEEE 802.11g, který zvýšil přenosovou rychlost na 54 Mbit/s. Další velký skok v rychlosti přišel v roce 2009 společně se standardem IEEE 802.11n, který zvýšil maximální rychlost na 600 Mbit/s. Aktuálně nejnovější standard IEEE 802.11ac z roku 2013 pracuje na frekvenci 5 GHz a dosahuje

rychlosti 3,6 Gbit/s. Za zmínku ještě stojí standard 802.11ad, vysílající na frekvenci 60 GHz s maximální rychlostí 7 Gbit/s. Tato bezdrátová síť je však určena pouze pro krátké vzdálenosti (zhruba do 10 m v rámci místnosti), protože rádiový signál vysílaný na takto vysoké frekvenci většinou nedokáže překonat zeď [14].

Standard	Rok vydání	Frekvence [GHz]	Maximální rychlost [Mbit/s]
původní IEEE 802.11	1997	2,4	2
IEEE 802.11a	1999	5	54
IEEE 802.11b	1999	2,4	11
IEEE 802.11g	2003	2,4	54
IEEE 802.11n	2009	2,4 nebo 5	600
IEEE 802.11ac	2013	5	3600
IEEE 802.11ad	2012	60	7000

Tabulka 1: Přehled IEEE 802.11 standardu

Zdroj: <http://www.zive.cz/clanky/novy-standard-wi-fi-gigabit-vzduchem/sc-3-a-165687>

Bezdrátová síť Wi-Fi je v dnešní době prakticky v každé domácnosti. Většina poskytovatelů internetového připojení ji nakonfiguruje společně s instalací routeru, aby se mohli zákazníci připojit na internet z notebooku, tabletu nebo chytrého telefonu. Dle projektu Wifilekas.cz je v České republice přes 2 milióny bezdrátových Wi-Fi sítí [15]. Ačkoliv tato technologie nebyla navržena přímo pro lokalizaci, díky jejímu globálnímu rozšíření je možné ji k tomuto účelu využít. Metody pro určení polohy na základě Wi-Fi jsou v principu stejné s těmi, které byly popsány v předchozí kapitole o lokalizaci pomocí GSM sítě.

Metody lokalizace pomocí Wi-Fi sítí:

- **Cell of Origin** – Zatímco BTS vysílač používá jako jedinečný identifikátor tzv. Cell ID, každý přístupový bod Wi-Fi, jakožto síťové zařízení, má svoji MAC adresu (hodnota BSSID – Basic Service Set Identification). Mobilní telefon změří sílu signálu okolních bezdrátových sítí a vybere tu nejsilnější. K vybrané síti je pak nutné v databázi dohledat odpovídající zeměpisné souřadnice, které jsou zároveň považovány za polohu, kde se nachází

mobilní telefon. Wi-Fi sítě mají typicky dosah od 25 do 250 metrů, což je zároveň přesnost určené pozice touto metodou.

- **Angle of Arrival** – Ačkoliv tato metoda přináší dobré výsledky, moc se nepoužívá, protože vyžaduje zásah do hardwarového vybavení mobilního telefonu. Se standardní anténou není možné změřit úhel, pod kterým je přijímán Wi-Fi signál, na jehož základě dochází k výpočtu polohy telefonu.
- **Trilaterace** – Princip metody je stejný jako u GSM sítě. Největším problémem je, jak co nejpřesněji změřit vzdálenost mezi mobilním telefonem a přístupovým bodem. Na rozdíl od GSM sítě není možné určit dobu, za kterou tuto vzdálenost signál urazí, protože uvnitř Wi-Fi sítě nedochází k synchronizaci přesného času. Metody ToA nebo TDoA tak nelze pro určení polohy použít. Z příchozího signálu je možné zjistit jeho sílu (RSS – Received Signal Strength) a odstup od šumu (SNR – Signal-to-noise Ratio). Pro výpočet vzdálenosti se tedy většinou používá síla signálu, jelikož poskytuje zpravidla přesnější výsledky. Podobně jako u mobilního signálu dochází k nepřesnostem v měření vzdálenosti kvůli překážkám, které způsobují útlum a lámání rádiových vln.
- **Radio Fingerprinting** – Jedná se o metodu, pomocí které lze dosáhnout nejpřesnějších výsledků. Problém je však naměření a aktualizace otisků se silou signálu.

K určení polohy je potřeba mít k dispozici databázi přístupových bodů s geografickými souřadnicemi. Nabízí se tedy otázka, kde například Google, který umožňuje ve svých aplikacích lokalizaci pomocí Wi-Fi, tyto údaje získává. V roce 2007 spustila tato společnost projekt Street View, který nabízí panoramatické snímky z měst po celém světě. Tyto fotografie jsou snímány pomocí flotily automobilů, která musí projet všechny ulice ve městě. Během jízdy jsou kromě jiného sbírána data právě o Wi-Fi a mobilních sítích. Mnohem efektivnější je ale do sběru dat zapojit komunitu. Uživatelé operačního systému Android, kteří mají zapnutou GPS a zároveň jsou připojeni k mobilnímu internetu, odesílají data

o bezdrátových sítích na servery Googlu a pomáhají tak vytvořit a zpřesnit celý lokalizační systém společnosti.

Ve městech, kde se nachází velké množství vzájemně se překrývajících Wi-Fi sítí, je určení polohy ve většině případů přesnější a jen o něco málo energeticky náročnější než lokalizace na základě mobilní sítě. Nejčastěji se tento způsob využívá uvnitř budov, ve kterých je možné určit polohu s relativně velkou přesností za pomoci metody otisků síly signálu. Toho lze využít například při nakupování ve velkém obchodním centru, kdy je možné na mobilním telefonu zobrazit plánec s rozmístěním jednotlivých obchodů a vykreslit na něm aktuální pozici uživatele. Zároveň může být uživateli při vstupu do určitého obchodu nabídnuta sleva na zboží. Dalším příkladem může být aplikace pro navigování po rozlehlých letištních terminálech, která dokáže pomoci cestujícím nalézt bránu, odkud odlétá jejich letadlo. V listopadu 2011 společnost Google spustila ve svých mapách navigaci uvnitř budov, která je právě založena na Wi-Fi technologii. Ve venkovním prostředí mimo města je tento způsob prakticky nepoužitelný nebo velmi nepřesný. V těchto případech je většinou výhodnější použít GPS. Ta je však energeticky náročnější a vyhledání tzv. první pozice trvá mnohem déle [13, 16, 17, 18].

3.3 GPS

Global Positioning System, zkráceně GPS, je vojenský globální družicový polohový systém provozovaný Ministerstvem obrany USA, s jehož pomocí lze určit polohu a přesný čas kdekoliv na Zemi. GPS navazuje na již dříve zmiňovaný projekt Transit, který dále rozšiřuje o lepší dostupnost, kvalitu a přesnost určené polohy [1, 3, 18, 19, 20, 21].

3.3.1 Struktura systému

Systém GPS je tvořen třemi základními segmenty: kosmický segment, řídicí segment a uživatelský segment. Pro správné určení polohy je důležitá dostupnost všech tří částí.

3.3.1.1 Kosmický segment

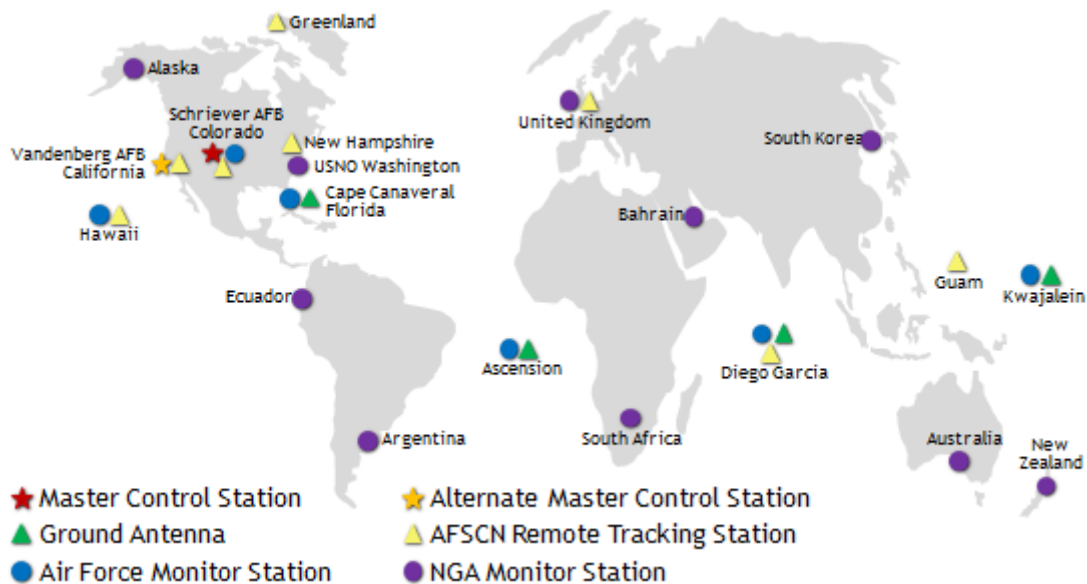
Kosmický segment byl projektován na 24 družic. V současné době kolem Země obíhá 31 družic rychlostí 3,8 km/s ve výšce 20 200 km nad povrchem Země po 6 kruhových drahách se sklonem 55 stupňů. Ačkoliv 7 funkčních satelitů navíc může zlepšit přesnost a dostupnost systému, jsou ve vesmíru umístěny primárně jako záloha v případě poruchy nebo údržby ostatních satelitů. Doba oběhu jedné družice kolem Země je 11 hodin 58 minut, což je polovina siderického dne. Zřizovatel zaručuje, že minimálně 4 družice jsou pozorovatelné 24 hodin denně z každého místa na Zemi (4 družice jsou minimem, pro určení přesné polohy). Například v České republice bylo v roce 2008 nejčastěji viditelných 8 družic (minimum 6 družic a maximum 12 družic) při elevaci větší než 10 stupňů [22]. Každá družice nese atomové hodiny pro určení přesného času, RHCP antény pro vysílání rádiového signálu v pásmu 1000 - 2000 MHz, antény pro komunikaci s pozemními kontrolními stanicemi, antény pro vzájemnou komunikaci družic, solární panely jako zdroj elektrické energie a různé senzory pro detekci startů balistických raket a jaderných výbuchů.

3.3.1.2 Řídící a kontrolní segment

Řídící a kontrolní segment se skládá z několika částí a je zodpovědný za funkčnost celého globálního polohovacího systému. Rozmístění jednotlivých stanic ve světě je vidět na obr. číslo 8.

- **řídící středisko** (Master Control Station) – Hlavní řídící středisko je umístěno ve státu Colorado. Záložní řídící středisko se nachází ve státu California a je připraveno v případě nouze převzít řízení systému do 24 hodin. Řídící středisko sbírá data od monitorovacích stanic a na jejich základě je vypočítána přesná poloha GPS satelitu ve vesmíru včetně jeho budoucí dráhy letu. Tyto údaje jsou společně s korekcí atomových hodin a dalšími informacemi odeslány ve formě tzv. navigační zprávy do příslušné družice pomocí povelových stanic. Řídící středisko se zároveň stará o správné rozmístění satelitů na orbitě.

- **4 povelové stanice** (Ground Antenna) – Povelové stanice jsou používány pro komunikaci a zasílání příkazů GPS satelitům.
- **16 monitorovacích stanic** (Monitor Stations) – Bezobslužné pozemní monitorovací stanice jsou ovládány dálkově z hlavního řídicího střediska. V podstatě se jedná o velice přesné GPS přijímače, které jsou doplněné o vlastní atomové hodiny. Stanice prakticky neprovádí žádné zpracování přijatých dat, pouze určují vzdálenosti ke družicím, které spolu s přijatými navigačními zprávami odesílají do řídicího střediska.



Obr. 8: Rozmístění stanic řídicího segmentu GPS

Zdroj: <http://www.gps.gov/systems/gps/control/map.png>

Výsledek monitoringu, který provádí řídicí segment, je zveřejňován v navigační zprávě každé družice. Platnost této zprávy je řádově několik hodin a obsahuje:

- korekce pro atomové hodiny
- zdravotní stav dané družice (zda lze použít pro výpočet polohy)
- predikce dráhy družice, tzv. efemeridy
- přibližné pozice ostatních družic a jejich zdravotní stav

- data pro ionosférický model (přijímač je využívá pro přibližný odhad vlivu ionosféry na GPS signál pro kterékoliv místo a čas)

3.3.1.3 Uživatelský segment

Poslední částí globálního polohovacího systému je uživatelský segment. Uživatelé pomocí pasivního GPS přijímače zachytávají signály z jednotlivých družic, které jsou v danou chvíli nad obzorem. Na základě přijatých dat je možné vypočítat polohu zařízení včetně nadmořské výšky a určit přesný čas. Komunikace probíhá pouze ze strany satelitů k uživateli, proto není možné GPS přijímač zaměřit.

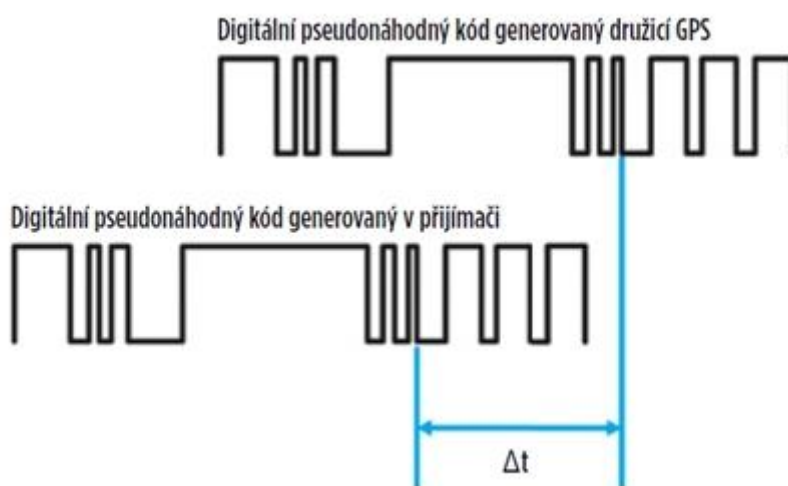
Uživatele GPS lze rozdělit do dvou skupin. Autorizovaní uživatelé využívající službu PPS (Precise Positioning Service) mají k dispozici dekódovací klíč k Y-kódu na frekvencích L1 a L2 (viz další podkapitola). Díky tomu mohou měřit zpoždění signálu při průchodu ionosférou a tím zvýšit přesnost určené pozice. Jedná se především o armádu. Ostatní uživatelé mohou využívat službu SPS (Standard Positioning Service), přičemž mají k dispozici pouze C/A kód na frekvenci L1. Přijímače pro civilní sektor musejí mít navíc zabudované omezení výšky (do 18 kilometrů) a rychlosti (do 515 m/s). Jde o preventivní opatření, aby nebylo možné přijímač použít například pro navádění balistických střel.

3.3.2 Princip určení polohy

GPS signál vysílaný družicí je kombinací nosné vlny, dálkoměrného kódu a navigační zprávy. Družice vysílají signál na pěti nosných frekvencích. Frekvence L1 (1575,42 MHz) je modulována dvěma dálkoměrnými kódy, které jsou reprezentovány tzv. pseudonáhodnými šumy. Jedná se o P-kód (Precision code), který může být pro vojenské účely zašifrován (poté je označován jako Y-kód). Druhým dálkoměrným kódem je veřejně dostupný a nešifrovaný C/A kód (Coarse/Acquisition code). Druhá frekvence L2 (1227,62 MHz) je modulována pouze P-kódem, případně jeho zašifrovanou variantou. Kromě C/A kódu a P-kódu je oběma nosnými frekvencemi přenášena také navigační zpráva, která je kódována pomocí fázových posunů nosných vln.

Frekvence L3 (1381,05 MHz) je používána pro vysílání signálu, který obsahuje data o startu balistických raket a detekci jaderných výbuchů. Frekvence L4 (1379,913 MHz) se využívá pro měření ionosférické refrakce (změna rychlosti šíření rádiových vln při průchodu ionosférou). Novější družice, 19 ze současných 31 družic, vysílají druhý civilní signál na frekvenci L2. Při kombinaci s klasickým C/A kódem je možné vypočítat korekci pro průchod signálu ionosférou a dosáhnout tak přesnějších výsledků, podobně jako to dělá armáda s využitím zašifrovaného P-kódu. Je však potřeba vlastnit dvoufrekvenční přijímač. Poslední frekvence L5 (1176,45 MHz) se plánuje pro vysílání dalšího civilního signálu, který bude využíván primárně v letectví a dopravě (aktuálně ho vysílá 12 nejnovějších družic).

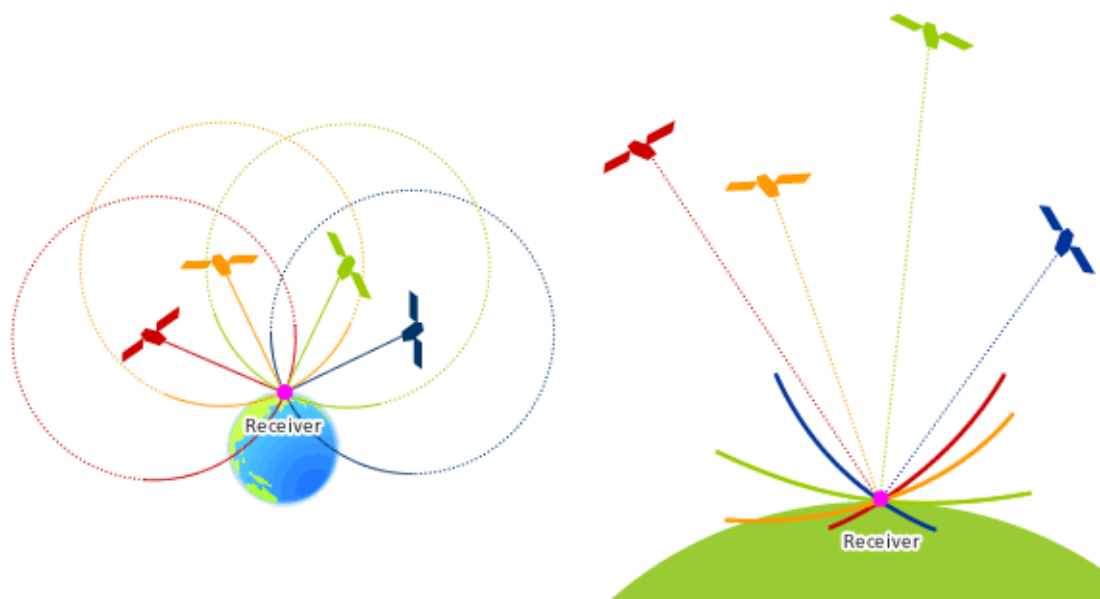
Pro samotnou lokalizaci je nejdůležitější digitální kód C/A. Jedná se v podstatě o pseudonáhodnou posloupnost nul a jedniček, kterou každý satelit vysílá v nepatrně pozměněné podobě, aby nedošlo k jejich záměně. GPS přijímač obsahuje vlastní generátor kódu C/A, který je přesně synchronizovaný s každou družicí. Než signál s aktuálním C/A kódem dorazí z družice do mobilního zařízení, čip v přijímači už generuje poněkud jinou hodnotu (viz obr. č. 9).



Obr. 9: Generování C/A kódu družicí a GPS přijímačem

Zdroj: http://zive.v.mfstatic.cz/GetThumbNail.aspx?id_file=934356578&width=479&height=268&q=100

GPS čip umožňuje zpětně zjistit, o jak velký časový rozdíl se jedná. V obrázku č. 9 je označen jako Δt . Tento časový rozdíl odpovídá době, po kterou letí signál od družice do mobilního telefonu. Díky tomu lze pak už relativně snadno vypočítat vzdálenost mezi družicí a GPS přijímačem za použití vztahu $d = \Delta t \cdot c$, kde c je rychlost šíření rádiových vln.



Obr. 10: Princip určení polohy pomocí GPS

Zdroj: http://global.jaxa.jp/countdown/f18/overview/img/pict_gps1e.gif

Vzdálenost mezi družicí a GPS přijímačem je možné vyjádřit koulí, kde se družice nachází v jejím středu a mobilní telefon někde na povrchu Země (viz obr. č. 10). Jedná se vlastně o již dříve popisovanou trilateraci, tentokrát ale v trojrozměrném prostoru. Aby bylo možné určit polohu, je potřeba zachytit signál minimálně ze 3 satelitů. Až poté je možné vypočítat dva různé body, kde by se mohl GPS přijímač nacházet. Jeden z bodů se bude vždy nacházet mimo zemský povrch a může být tedy vyloučen. Takto určená poloha však bude velmi nepřesná, a proto je zapotřebí získat signál ze čtvrté družice pro korekční výpočty.

Pro určení polohy GPS přijímače je nutné znát přesnou pozici satelitů ve vesmíru. Tu je možné zjistit z navigační zprávy, kterou každý satelit vysílá. Navigační zpráva

je nahrávána z pozemního střediska do satelitů každé 2 hodiny a má platnost několik málo hodin. Stažení zprávy přímo ze satelitů po delší době nečinnosti přijímače může trvat i několik minut. Proto byla zavedena technologie A-GPS (Assisted GPS), která si klade za cíl urychlit prvotní zjištění pozice. Mobilní telefon s GPS přijímačem si může příslušné navigační zprávy stáhnout z internetu. Díky tomu je možné v ideálním případě vypočítat první polohu v řádu několika sekund.

3.3.3 Přesnost systému GPS

Přesnost horizontální pozice určené pomocí GPS je v dnešní době zpravidla lepší než 3,5 metru. Záleží však na mnoha faktorech, které mohou způsobit větší či menší nepřesnost ve výpočtu. Patří mezi ně například chyba v predikování polohy družice na oběžné dráze, nepřesný čas GPS přijímače, ionosférická a troposférická refrakce (lom rádiového signálu při průchodu atmosférou), vícecestné šíření (odraz signálu od budov), počet viditelných družic, metodika výpočtu nebo geometrické rozmístění družic. V 90. letech byla dokonce do signálu určeného pro civilní použití vnášena záměrná chyba, která snižovala přesnost až o několik desítek metrů. Jednalo se o tzv. selektivní dostupnost (Selective Availability). Toto opatření mělo zabránit zneužití systému pro navádění balistických raket. Protože USA vyvinuly způsob, jak lokálně rušit GPS signál pomocí pozemního vysílače, byla v roce 2000 selektivní dostupnost vypnuta.

Pro dosažení ještě vyšší přesnosti je možné využít tzv. Diferenciální GPS (zkr. DGPS). DGPS zařízení přijímá signál GPS společně s pozemním rádiovým signálem ze stanice, o které je známa její přesná pozice. Tato referenční stanice vyhodnocuje odchylky měření v systému od skutečného stavu své polohy a vypočítává korekce. Přesnost tohoto systému je maximálně několik desítek centimetrů a je využíván hlavně v oboru geodézie.

3.3.4 Shrnutí určování polohy pomocí GPS

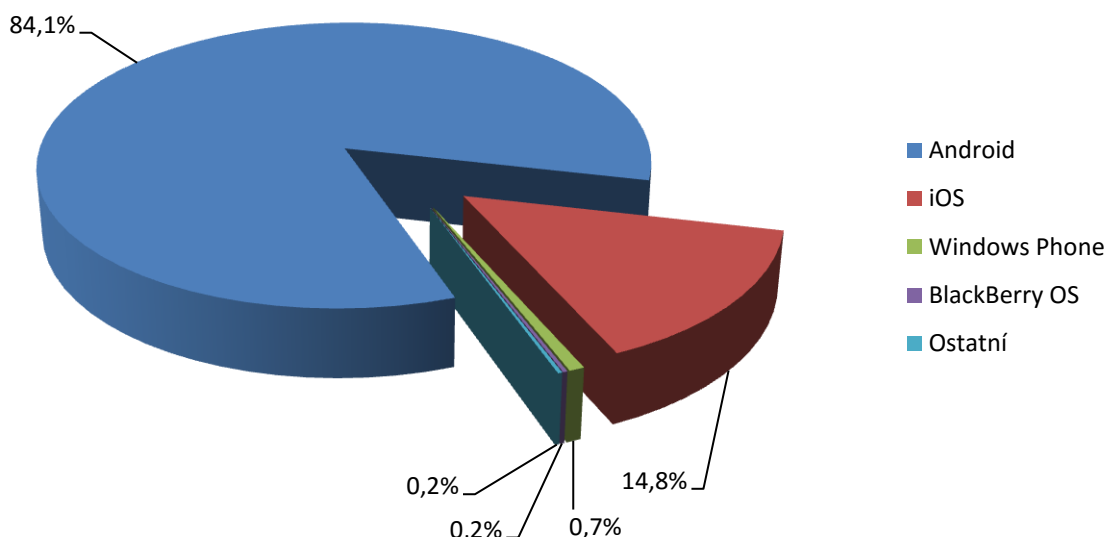
Jedná se o nejpřesnější z popisovaných způsobů určení geografické polohy ve venkovním prostředí. Mezi hlavní výhody se řadí celosvětová dostupnost, přesnost a možnost získat trojrozměrné souřadnice. Mezi nevýhody patří vysoká spotřeba

elektrické energie a nefunkčnost uvnitř budov nebo v hustě zastavěných oblastech, kde je omezený výhled na oblohu. Další nevýhodou může být doba, která je nutná pro určení první polohy. To však neplatí, pokud si zařízení umí stáhnout pozice jednotlivých satelitů z internetu. GPS není jediným globálním družicovým polohovým systémem. Za zmínku jistě stojí ruský GLONASS nebo plánovaný evropský satelitní navigační systém Galileo.

4 OS Android

Android je v současnosti nejrozšířenějším mobilním operačním systémem na trhu. Z grafu na obrázku č. 11 je vidět, že v prvním čtvrtletí roku 2016 bylo prodáno více jak 84% mobilních telefonů koncovým zákazníkům právě s tímto operačním systémem. Vývoj systému zaštiťuje konsorcium Open Handset Alliance, jehož nejdůležitějším členem je společnost Google.

Prodej mobilních telefonů v Q1 2016



Obr. 11: Prodej mobilních telefonů v Q1 2016

Zdroj: <http://www.gartner.com/newsroom/id/3323017>

4.1 Stručná historie

Společnost Android Inc. založili v roce 2003 Andy Rubin, Rich Miner, Nick Sears a Chris White. Jejich cílem bylo vytvářet chytřejší mobilní zařízení. Činnost společnosti z počátku nezbuzovala mnoho pozornosti. To se změnilo v roce 2005, kdy Android Inc. odkoupila americká firma Google, která z ní udělala svoji dceřinou společnost.

Po akvizici vyvinul tým Googlu pod vedením Andyho Rubina platformu založenou na linuxovém jádře. V září roku 2007 získala společnost Google několik patentů v oblasti mobilních technologií a začaly se objevovat první spekulace, že chce Google tímto krokem vstoupit na trh chytrých mobilních zařízení.

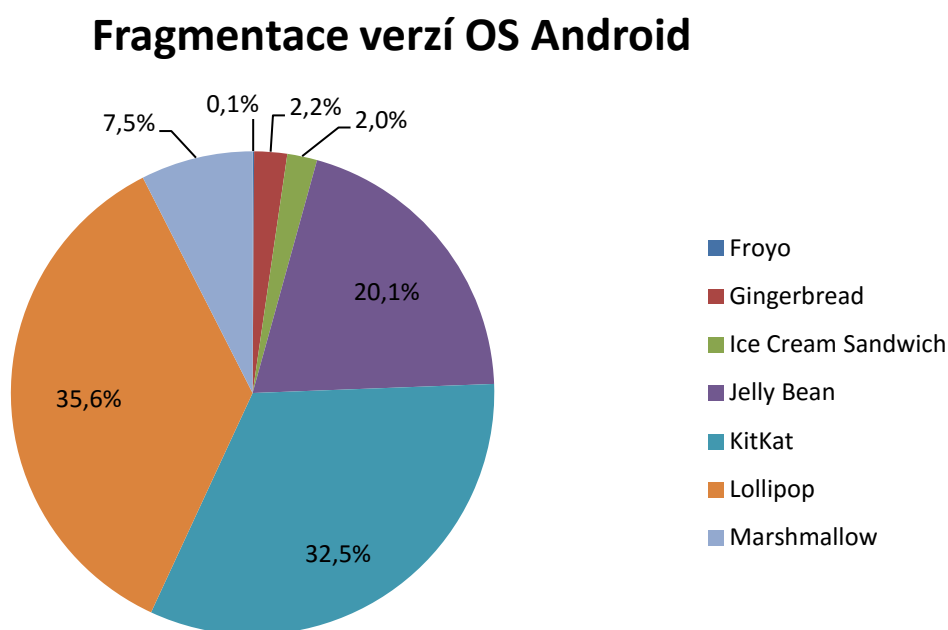
Dne 5. listopadu roku 2007, jen pár měsíců poté, kdy byl na trh uveden v mnoha ohledech revoluční iPhone od společnosti Apple, bylo vytvořeno uskupení Open Handset Alliance. Mezi zakládající členy konsorcia patřily kromě společnosti Google například NVIDIA, Samsung, LG, HTC, Motorola, Intel a mnozí další. Cílem tohoto uskupení bylo vyvinout otevřený standart pro mobilní zařízení. V ten samý den Open Handset Alliance představilo nový mobilní operační systém s názvem Android, který bude možné použít na tisících rozdílných telefonních modelech.

Prvním mobilním telefonem s OS Android ve verzi 1.0, který se dostal koncem roku 2008 do prodeje, se stal T-Mobile G1, známý také pod kódovým označením HTC Dream. Toto zařízení odstartovalo éru obrovských úspěchů platformy Android na trhu s chytrými mobilními telefony. V roce 2009 vzrostl počet zařízení využívajících OS Android na více jak dvacet. Na konci roku 2010 se Android stal vedoucí platformou na poli mobilních operačních systémů. Počátkem roku 2012 činil podíl na trhu 59% a v roce 2013 už Android jednoznačně dominoval trhu s podílem okolo 80% [23, 24].

4.2 Fragmentace OS Android

Google se při vývoji operačního systému vydal cestou otevřenosti. Velká část zdrojových kódů platformy Android je šířena pod licencí Apache 2.0. Výrobci mobilních telefonů platí licenční poplatky pouze za využití proprietárních služeb, kterou je například obchod s aplikacemi Google Play. Díky tomu, že jsou zdrojové kódy otevřené, může výrobce operační systém jakkoliv upravit a optimalizovat pro dané mobilní zařízení. Zároveň je ale také zodpovědný za jeho aktualizaci na novější verzi. Google navíc nikterak neomezuje hardwarové vybavení telefonu, na kterém je Android nainstalován. Programátoři tak musí při vývoji aplikace řešit podporu různých velikostí a rozlišení zobrazovacích panelů nebo kontrolovat

přítomnost senzorů, které jsou pro běh aplikace potřebné. Toto všechno společně s rychlým vývojem nového operačního systému přináší značnou fragmentaci platformy. Na obrázku č. 12 je znázorněn podíl jednotlivých verzí operačního systému Android na zařízeních, která přistupují do obchodu Google Play (květen 2016). Ačkoliv verze operačního systému Android 2.2 (Froyo) a 2.3 (Gingerbread) byly vydány v roce 2010, stále běží na 2,3% zařízeních. Android s kódovým označení Jelly Bean, který byl představen v roce 2012 je nainstalován dokonce na 20,1% zařízeních. V následujících odstavcích budou představeny vybrané verze OS Android společně s nejdůležitějšími novinkami, které přinesly [24, 26, 27].



Obr. 12: Fragmentace verzí OS Android, květen 2016

Zdroj: <https://developer.android.com/about/dashboards/index.html>

Android 2.2 – Froyo (2010)

- možnost instalace aplikací na paměťovou kartu
- zvýšena rychlost systému díky novému kompilátoru Dalvik JIT
- podpora push notifikací
- podpora animovaných obrázků ve formátu GIF

Android 2.3 – Gingerbread (2010)

- podpora NFC (Near Field Communication)
- souborový systém ext4 místo dosavadního YAFFS
- nativní podpora pro více senzorů (např. gyroskop nebo barometr)

Android 3.0 – Honeycomb (2011)

- speciální verze optimalizovaná pro velké obrazovky tabletů
- poprvé představeny fragmenty
- nový systémový panel (softwarová tlačítka – zpět, domů, menu)

Android 4.0 – Ice Cream Sandwich (2011)

- sjednocení tabletové a mobilní verze OS

Android 4.1, 4.2, 4.3 – Jelly Bean (2012)

- „Project Butter“ (plynulejší uživatelské rozhraní)
- nativní podpora pro jazyky psané zprava doleva
- podpora Bluetooth Low Energy

Android 4.4 – KitKat (2013)

- Android RunTime (ART)
- aplikace přes celou obrazovku (možnost skrýt navigační a stavovou lištu)
- veřejné API pro textové zprávy

Android 5.0, 5.1 – Lollipop (2014)

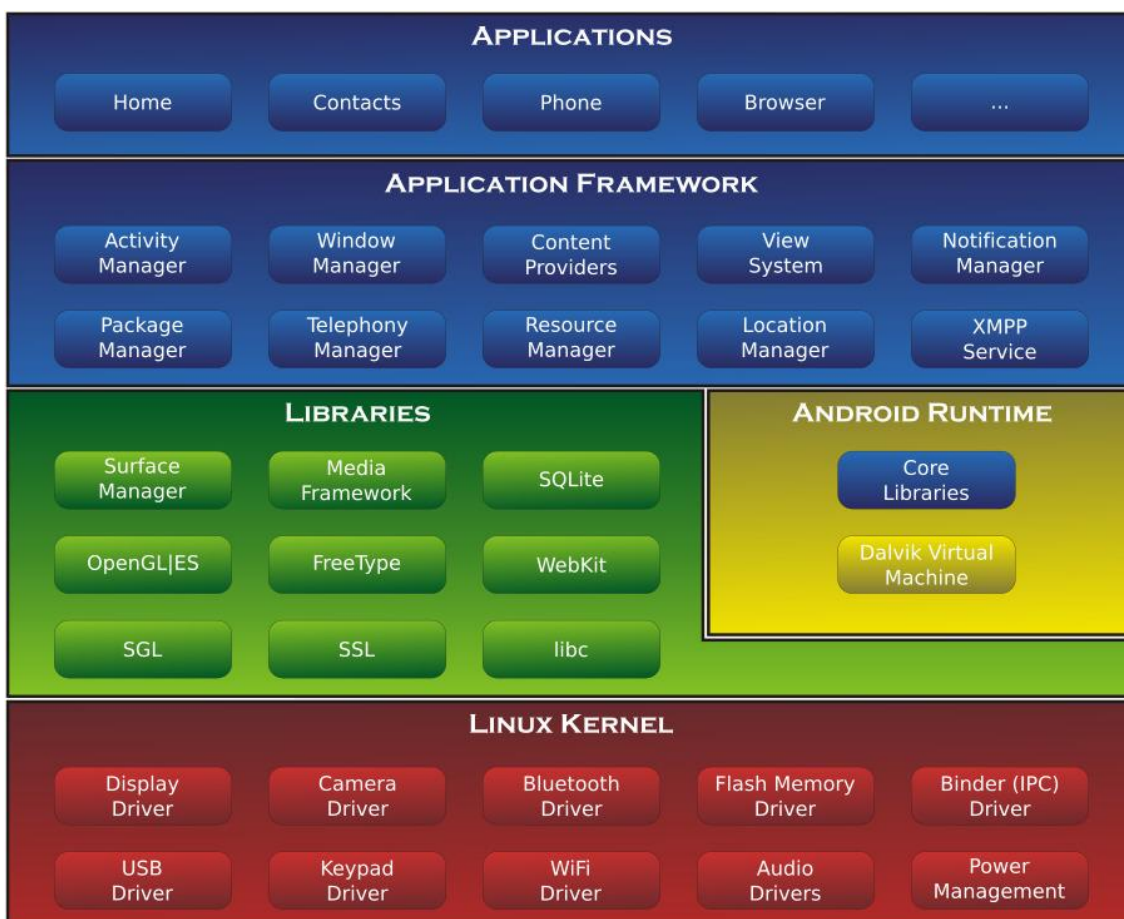
- nový vzhled „Material design“
- podpora 64-bitových procesorů
- „Project Volta“ (vylepšená spotřeba baterie)

Android 6.0 – Marshmallow (2015)

- nový způsob přidělování oprávnění pro aplikace
- nový režim úspory baterie
- podpora snímače otisků prstů

4.3 Architektura

Architektura operačního systému Android se skládá ze softwarových komponent, které jsou rozděleny celkem do 5 částí tvořící 4 hlavní vrstvy. Členění architektury je přehledně vyobrazeno na následujícím obrázku č. 13 [23, 24, 25].



Obr. 13: Diagram architektury OS Android

Zdroj: <https://upload.wikimedia.org/wikipedia/commons/a/af/Android-System-Architecture.svg>

4.3.1 Jádro (Kernel)

Nejnižší vrstva architektury je jádro, které tvoří abstraktní vrstvu mezi hardwarem a softwarem ve vyšších vrstvách. Mezi jeho hlavní části patří ovladače, které umožňují komunikaci s jednotlivými hardwarovými moduly (displej, kamera, Bluetooth, atd.). Dále také jádro zajišťuje správu procesů a paměti. Android je operační systém postaven na linuxovém jádře, které je upraveno pro specifické požadavky mobilních zařízení převážně s procesory ARM.

4.3.2 Knihovny (Libraries)

Nativní knihovny OS Android jsou napsány v jazyce C/C++ a jejich funkce jsou vývojářům poskytnuty prostřednictvím aplikačního frameworku. Vrstva obsahuje tyto základní typy knihoven:

- Surface Manager – zobrazování aplikací a jejich vrstvení
- OpenGL a SGL – knihovny pro práci s 3D a 2D grafikou
- Media Framework – práce s multimediálními soubory
- FreeType – knihovna pro vykreslování písma
- SSL – šifrování a zabezpečení přenosu dat
- SQLite – relační databázová knihovna
- WebKit – vykreslovací jádro pro webový prohlížeč
- libc – standardní knihovna jazyka C

4.3.3 Běhové prostředí (Android Runtime)

Část architektury systému, která se stará o běh samotných aplikací. Vzhledem k tomu, že jsou aplikace pro Android psány v programovacím jazyce Java, nachází se zde Dalvik Virtual Machine. Jedná se o virtuální stroj, který je speciálně upraven a optimalizován pro běh na mobilních zařízeních a zajišťuje překlad zdrojového kódu aplikace do nativního kódu procesoru. Virtuální stroj Java Virtual Machine nebyl v operačním systému Android použit především z licenčních důvodů. Od verze Androidu 4.4 se používá běhové prostředí ART (Android Runtime) s tzv. dopřednou kompilací, kdy je zdrojový kód zkompilován při instalaci aplikace do telefonu. Výsledkem je vyšší rychlost při spouštění aplikace a zvýšení výdrže mobilního zařízení. Nevýhodou je delší čas potřebný pro instalaci aplikace. V této části architektury se také nachází standardní knihovny jazyku Java.

4.3.4 Aplikační framework (Application Framework)

Tato vrstva je pro vývojáře nejdůležitější. Poskytuje knihovny napsané v jazyce Java, které tvoří systémové aplikační rozhraní. Programátor pomocí tohoto API může využívat jednotlivé služby nabízené operačním systémem. Jedná se například o vytváření grafického uživatelského rozhraní aplikace, přístup k obsahu jiných aplikací, zjištění polohy mobilního zařízení nebo přístup k hardwarovému vybavení telefonu.

4.3.5 Aplikace (Applications)

Nejvyšší vrstvu systému tvoří samotné aplikace, které mohou být předinstalovány výrobcem telefonu nebo dodatečně staženy uživatelem z obchodu Google Play. Příkladem těchto aplikací může být kalendář, poštovní klient, webový prohlížeč, aplikace s kontakty nebo hry.

5 Požadavky na aplikaci

V dnešní době je běžně dostupné velké množství aplikací a her pro mobilní zařízení. Jen z obchodu Google Play, kde se nachází aplikace pro operační systém Android, je jich možné stáhnout a nainstalovat přes 2 milióny [28]. Asi nejznámější geolokační aplikací využívající prvky rozšířené reality je online hra s názvem Ingress [29], která byla vydána přímo společností Google. Po registraci a přihlášení do hry si hráči vyberou ze dvojice frakcí tu, za kterou budou po zbytek hry hrát. Hráči mají za úkol obsazovat virtuální portály, které jsou však umístěny v reálné mapě na souřadnicích zajímavých staveb a památek. Frakce mezi sebou soupeří o maximální počet obsazených portálů a o vytvoření co největšího pole, které obsazené portály vytvářejí. Hra je graficky na velmi vysoké úrovni a je dostupná globálně po celém světě. Z českého prostředí je možné uvést například aplikaci GEOFUN [30], ve které mohou uživatelé plnit různé úkoly na zajímavých místech v rámci České republiky. Obě zmíněné aplikace jsou dostupné pro platformy Android a iOS.

Cílem diplomové práce není konkurovat těmto vyspělým aplikacím, ale spíše se jimi inspirovat a vytvořit plně funkční geolokační aplikaci s herními prvky, na které bude možné demonstrovat v práci popsané způsoby určování polohy mobilního zařízení.

Základním principem aplikace bude provést uživatele po historických památkách a dalších zajímavých místech ve městě. Zároveň by aplikace mohla přispět ke zvýšení návštěvnosti těchto památek. Ke každému místu bude dostupná fotografie, popis dané památky a webový odkaz pro získání dalších informací. Jedná se tak o poměrně velké množství dat, které bude muset mít uživatel dostupné v telefonu. Jedním z cílů práce bylo vymyslet, jak co nejvíce šetřit data, která jsou přenášena přes mnohdy pomalé mobilní připojení, na které jsou aplikované různé FUP limity a omezení.

Uživatel bude mít tedy za úkol dojít k vyznačenému místu na mapě, kde se nachází zajímavá historická památka. Pomocí GPS se ověří, zda se uživatel skutečně

vyskytuje v její blízkosti. Jako vylepšení ověření polohy aplikace také zkontroluje MAC adresy přilehlých přístupových bodů k bezdrátovým Wi-Fi sítím oproti databázi. Za navštívení památky a její úspěšnou aktivaci dostane uživatel přidělen určitý počet bodů. Hráči pak spolu budou soupeřit v žebříčku, kdo má nejvíce bodů. Zároveň bude aplikace umožňovat synchronizovat herní profil. Uživatel se tedy bude moci přihlásit na jakémkoli zařízení a bude mít k dispozici vždy aktuální data o navštívených památkách a přiděleném počtu bodů.

6 Návrh a implementace

Aby bylo možné stanovenou funkcionalitu implementovat, musela být aplikace rozdělena do dvou částí. První částí je jednoduché webové aplikační rozhraní. Druhou částí je pak samotná aplikace pro Android. V následujících podkapitolách bude detailně popsán návrh a implementace obou částí.

6.1 Webové aplikační rozhraní (API)

Tato podkapitola se bude zabývat návrhem a popisem implementace serverové části aplikace, kterou je webové aplikační rozhraní.

6.1.1 Popis funkcionality

Webové aplikační rozhraní zajišťuje následující funkčnost pro vytvářenou mobilní aplikaci:

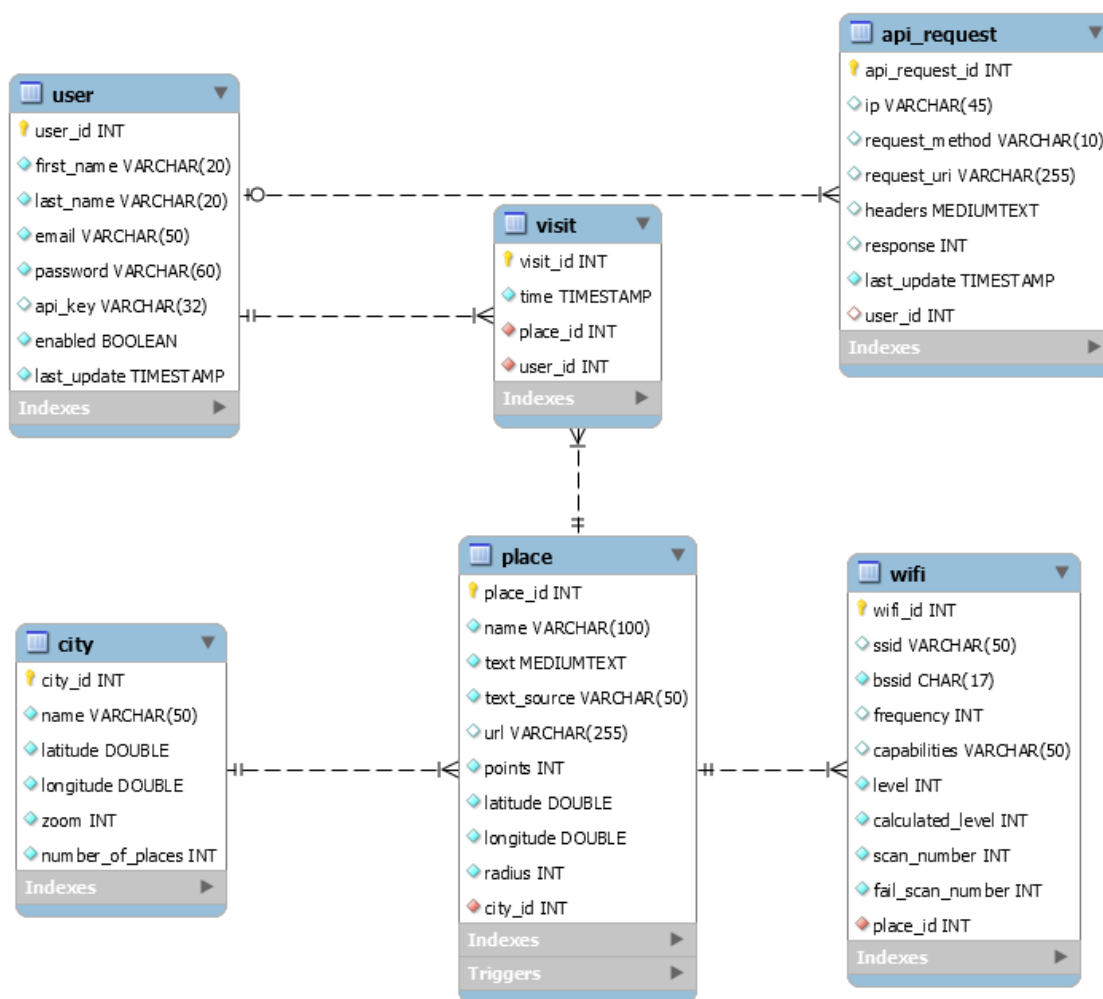
- Registrace a přihlášení do aplikace
- Validace platnosti přiděleného API klíče
- Seznam měst a jejich detail
- Seznam památek ve městě a jejich detail
- Aktivace (navštívení) památky
- Seznam okolních Wi-Fi sítí pro danou památku
- Odeslání seznamu naskenovaných Wi-Fi sítí při aktivaci památky
- Globální a městský žebříček uživatelů
- Synchronizace profilu

6.1.2 Návrh aplikace

Serverová část je napsána ve skriptovacím programovacím jazyce PHP. Tento jazyk je určený především pro programování dynamických webových aplikací a je také vhodný pro vytvoření jednoduchého webového aplikačního rozhraní. Data, se kterými aplikace pracuje, jsou uložena v relační databázi MySQL od společnosti Oracle.

6.1.2.1 Databáze

Při vytváření databázového návrhu byl využit vizuální nástroj MySQL Workbench, který integruje vývoj SQL, návrh databáze a její správu. Schéma databáze je vidět na obr. č. 14.



Obr. 14: Databázový model serverové části aplikace

Zdroj: vlastní, MySQL Workbench

user – Tabulka *user* obsahuje údaje o uživateli. Kromě jména, příjmení a emailové adresy je zde uloženo také v zašifrované podobě heslo, které uživatel vkládá při přihlášení do aplikace. Po registraci a úspěšném přihlášení je uživateli vygenerován unikátní API klíč, pomocí kterého jsou prováděny všechny následující požadavky na webové aplikační rozhraní. Poslední ukládanou informací o uživateli

je hodnota ve sloupci *enabled*. Nastavením tohoto sloupce na hodnotu *false* je možné zablokovat konkrétní uživatelův účet a znemožnit mu tak přihlášení do aplikace.

city – V této tabulce se nachází jednotlivá města, která je možné stáhnout do zařízení. Kromě názvu je zde uložena zeměpisná šířka, zeměpisná délka a velikost přiblížení. Tyto údaje využívá mobilní aplikace pro zobrazení počátečního pohledu na herní mapu. Do posledního sloupce s názvem *number_of_places* je databázovým triggerem vypočítáván počet památek, které jsou ve městě k dispozici.

place – Tabulka reprezentující herní místo na mapě, v níž jsou uloženy obecné informace o památce. Mezi ně patří název, popis památky včetně zajímavostí, zdroj textu (odkud byl popis převzat), zeměpisná šířka a zeměpisná délka. Pokud má památka webová stránka, je ve volitelném sloupci *url* uložen webový odkaz, kde může uživatel získat další informace. Ke každé památce jsou navíc uložena herní data. První uloženou informací je počet bodů, které uživatel dostane za navštívení a úspěšnou aktivaci památky. Tabulka také obsahuje vzdálenost udávající poloměr, ve kterém je možné památku aktivovat.

wifi – Tato tabulka obsahuje bezdrátové sítě v okolí památky, které jsou použity pro dodatečné ověření polohy uživatele. Mobilní aplikaci jsou předávány pouze sloupce *bssid* a *calculated_level*. Sloupec *bssid* obsahuje MAC adresu přístupového bodu k Wi-Fi síti. Hodnota tohoto sloupce musí být pro každou památku unikátní. Sloupec *calculated_level* obsahuje sílu signálu přepočítanou na hodnotu od 0 do 100. Za pomoci sloupců *scan_number* a *fail_scan_number* dochází k výpočtu, zda se příslušná Wi-Fi síť stále nachází v okolí památky. Pokud není síť po určitou dobu uživateli nahrávána na server, dojde k jejímu odstranění ze seznamu sítí, které je nutné zachytit, aby došlo k úspěšné aktivaci památky.

visit – Do tabulky *visit* se zaznamenává úspěšná aktivace památky pro daného uživatele včetně času návštěvy.

api_request – Veškeré požadavky na webové API se ukládají do této tabulky. Je v ní uložen typ HTTP požadavku, hlavička požadavku, odpověď serveru a IP adresa zařízení, odkud požadavek na API přišel. Pokud to je možné, dojde také ke spárování požadavku s registrovaným uživatelem v databázi. Logování požadavků do databáze umožňuje vytvoření statistik o používání aplikace a rychlejší odhalení případných chyb.

6.1.2.2 REST API

Pro API byla zvolena architektura REST (Representational State Transfer). Jedná se o architekturu rozhraní, která umožňuje vytvořit, číst, editovat nebo smazat informace ze serveru pomocí jednoduchých HTTP volání. REST definuje čtyři základní metody pro přístup k datům. GET požadavek na server slouží ke čtení dat. Pomocí tohoto požadavku je možné například získat seznam památek v daném městě. Metoda PUT slouží k aktualizaci dat na serveru. Pomocí metody POST je možné přidat nového uživatele do databáze. DELETE je poslední HTTP metodou, díky které je možné smazat záznamy z databáze [32].

Součástí odpovědi serveru na klientský požadavek je kromě samotných dat i stavový kód HTTP, který se nachází v hlavičce. Stavový kód upřesňuje, zda byl požadavek na server vyřízen kladně, záporně nebo zda došlo během zpracování k chybě. Nyní zde budou uvedeny stavové kódy, které webové API používá, společně s příkladem, kdy mohou nastat.

- **200 OK** – požadavek proběhl v pořádku (*úspěšný dotaz na detail památky*)
- **201 Created** – při požadavku POST, pokud byl vytvořen nový obsah (*úspěšná registrace nového uživatele*)
- **400 Bad Request** – špatný požadavek ze strany klienta (*chyba v parametru, při registraci nemá zadané heslo délku mezi 5 a 20 znaky nebo emailová adresa není validní*)
- **401 Unauthorized** – klient není ověřen (*server vrací tento stavový kód, pokud se zadaný API klíč neshoduje se záznamem v databázi*)

- **404 Not Found** – zdroj nenalezen (*chyba v parametru, id památky není obsaženo v databázi*)
- **500 Internal Server Error** – obecná chyba na straně serveru (*tato chyba může nastat například při výpadku spojení s databází*)

K webovému API byla vytvořena HTML dokumentace všech HTTP požadavků pomocí programu apiDoc [31]. Program generuje dokumentaci na základě anotací přímo ze souborů se zdrojovým kódem aplikace. V dokumentaci je možné dohledat k jednotlivým požadavkům URL adresy, parametry dotazu, ukázkou úspěšné odpovědi s popisem všech hodnot a příklady chyb, které mohou během komunikace nastat.

Pro přenos dat mezi serverem a mobilní aplikací byl zvolen datový formát JSON kvůli jednoduchosti implementace na obou stranách a také vzhledem k menšímu objemu dat, který tento formát oproti XML při přenosu spotřebuje. To může být značnou výhodou u mnohdy pomalého mobilního internetového připojení.

Příklad samotné komunikace klienta se serverem může vypadat následovně. Klient odešle GET požadavek na URL `/v1/places/1`. Chce tedy dostat detail památky, která má v databázi uloženo číslo 1 ve sloupci `place_id`. Řetězec `v1` na začátku URL adresy označuje verzi API. Oddělení jednotlivých verzí aplikačního rozhraní umožňuje vydat novou verzi bez ohrožení funkčnosti stávajících klientů. Podmínkou pro úspěšné zpracování tohoto dotazu je odeslání validního API klíče v hlavičce požadavku v poli `Authorization`.

Ukázka odpovědi serveru:

```
HTTP/1.1 200 OK
{
  "id": 1,
  "name": "Bílá věž",
  "text": "Text s popisem památky.",
  "textSource": "Wikipedie",
  "url": "http://www.bilavez.cz",
  "points": 20,
  "latitude": 50.208974,
  "longitude": 15.830888,
  "radius": 20,
  "imageUrl": "http://madamec.cz/images/places/image_1.jpg"
}
```

6.1.3 Implementace

Pro zjednodušení a zrychlení vývoje webového aplikačního rozhraní byl použit Slim Framework [33] ve verzi 2.6.1. Jedná se o micro framework, který umožňuje velmi jednoduše mapovat jednotlivé HTTP požadavky na příslušné URL adresy a zajistit jejich obsluhu. Zároveň poskytuje způsob, jak globálně upravit všechny příchozí a odchozí HTTP požadavky a odpovědi. To je využíváno například při validaci API klíče, který je nutné zkontrolovat v každém příchozím požadavku na server. Jde tedy o vhodný framework pro tvorbu jednoduchých webových aplikací a právě API rozhraní, kde by použití klasického plnohodnotného frameworku bylo zbytečné.

Serverová část aplikace byla napsána pomocí vývojového prostředí PhpStorm. Hlavní výhodou tohoto programu je, že obsahuje vestavěného REST klienta. Díky němu bylo možné pohodlně otestovat naprogramované webové API bez nutnosti paralelního vývoje mobilní aplikace.

Struktura aplikace je velmi jednoduchá. Ve složce *documentation* se nachází vygenerovaná HTML dokumentace pomocí programu *apiDoc*. Ve složce *images*

jsou obrázky měst a jednotlivých památek. Složka *include* obsahuje několik souborů používaných napříč celou aplikací. Nachází se zde konfigurace aplikace, metody pro přístup k databázi, třída *Logger*, která slouží pro logování dotazů na API a třída *Helper*, která obsahuje pomocné metody, jako je například metoda pro vygenerování unikátního API klíče. Ve složce *libs* se nachází Slim framework. Složka *v1* pak obsahuje zdrojové soubory samotné aplikace. Soubor *index.php* zajišťuje inicializaci a zavedení frameworku. Ve složce *v1/controllers* jsou jednotlivé kontrolery, které se starají o mapování požadavků na URL adresy a jejich obsluhu. V těchto souborech se také nacházejí komentáře sloužící pro vygenerování dokumentace. Poslední složka *v1/db* obsahuje třídy, které umožňují práci s databází. Nachází se v ní například třída *UserDatabase*, která obsahuje metody týkající se uživatele (*createUser()*, *checkLogin()*, *getUserByEmail()*, *isValidApiKey()*, atp.).

Ukázka kódu ze souboru *cityController.php*:

```
$app->get('/cities/:id', 'authenticate', function ($id) use ($app) {
    $cityDatabase = new CityDatabase();
    $result = $cityDatabase->getCity($id);

    if ($result->num_rows == 0) {
        Helper::echoErrorResponse(404, "Požadované město nebylo nalezeno.");
        $app->stop();
    }

    $response = $result->fetch_assoc();
    $response["logoUrl"] =
        BASE_URL . "/images/cities/logo_" . $response["id"] . ".png";
    $response["imageUrl"] =
        BASE_URL . "/images/cities/image_" . $response["id"] . ".jpg";

    $result->close();

    Helper::echoResponse(200, $response);
});
```

V ukázce kódu je vidět zpracování GET požadavku zasláného na URL *v1/cities/:id*. Odpověď by měla obsahovat data ve formátu JSON s detailem památky. Na prvním řádku dochází k namapování požadavku na příslušnou URL adresu. Ta je předána v prvním parametru metodě *get()*, kterou poskytuje Slim framework. Druhým parametrem je název funkce, která ověří platnost zasláného API klíče a případně

ukončí provádění celé aplikace. Třetím parametrem je anonymní funkce zajišťující samotnou obsluhu požadavku. V ní se vytvoří instance třídy *CityDatabase*, která slouží pro připojení k databázi a získání dat. Řádek z databáze obsahující detail města se převede na asociativní pole pomocí metody *fetch_assoc()*. Následně jsou do pole přidány odkazy na obrázek a logo města. Tyto URL nejsou ukládány v databázi, ale skládají se dynamicky na základě ID. Výsledné asociativní pole se převede pomocí standardní PHP funkce *json_encode()* do formátu JSON a společně s návratovým HTTP kódem se odešle klientovi. Na stejném principu funguje zpracování všech ostatních požadavků.

6.2 Mobilní aplikace pro Android

Tato podkapitola se zabývá návrhem a popisem implementace mobilní aplikace pro operační systém Android. Při zpracování bylo čerpáno ze zdrojů [34] a [35].

6.2.1 Popis funkcionality

Uživatel se nejprve musí přihlásit do aplikace. Po úspěšném přihlášení se zobrazí seznam měst, ze kterého je možné jedno město vybrat a stáhnout do zařízení. Po dokončení stahování je načten mapový podklad, ve kterém jsou vyznačeny jednotlivé památky. Navštívené památky reprezentuje zelený bod na mapě a nenavštívené červený. Po kliknutí na památku je zobrazena informační bublina s názvem dané památky, počtem bodů, které uživatel za navštívení může získat, a statusem, zda již byla památka navštívena. Pokud se jedná o navštívenou památku (zelený bod), je možné kliknutím na bublinu zobrazit přímo detail památky. Detail nenavštívených památek je možné zobrazit pouze tehdy, pokud se uživatel nachází v aktivačním kruhu okolo památky. Ten je na mapě vykreslen modrou barvou. Jakmile hráč do tohoto kruhu vstoupí, zobrazí se v dolní části obrazovky tlačítko s názvem památky. Kliknutím na toto tlačítko je možné vyvolat detail památky.

Detail obsahuje fotografii, webový odkaz, vzdálenost od památky, počet bodů za aktivaci a podrobné informace o památce. U nenavštívených památek je zde také tlačítko, které vyvolá proces aktivace. Během aktivace památky dochází k ověření

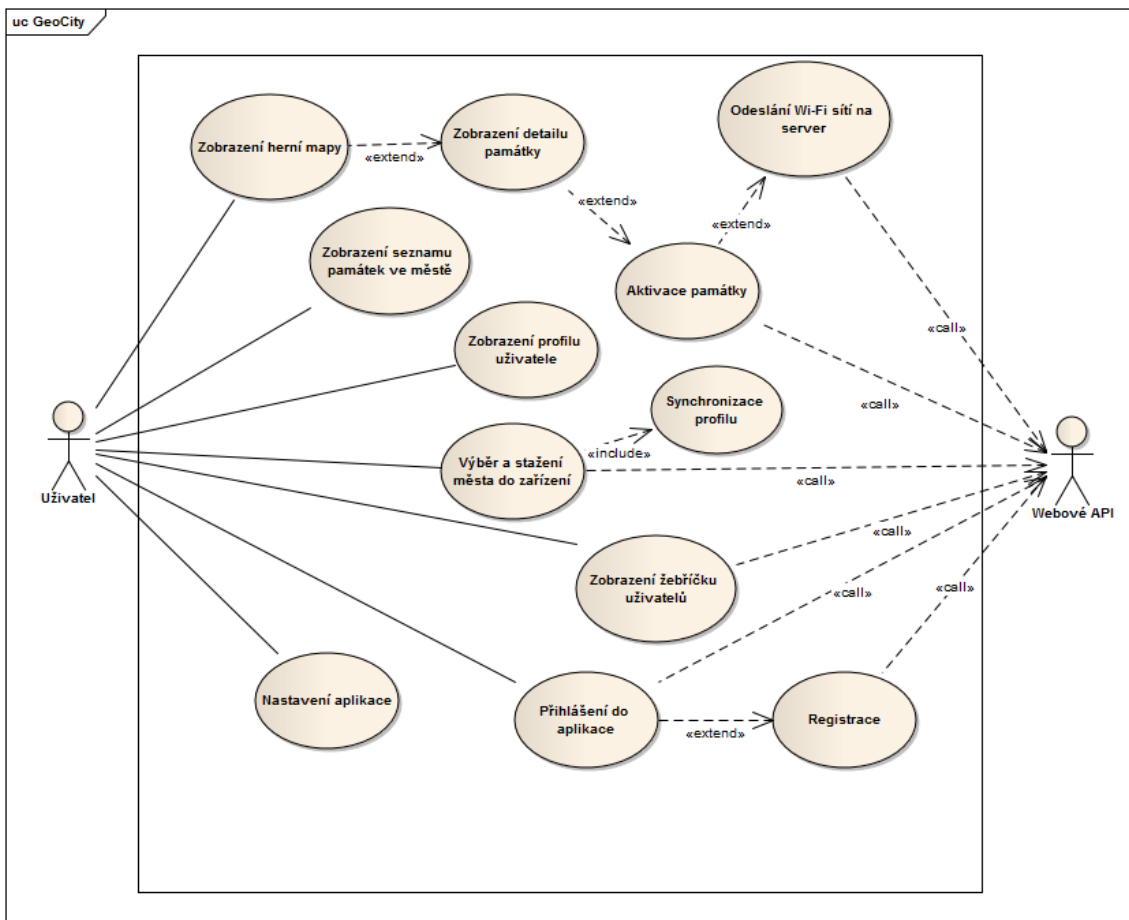
GPS polohy a je porovnán seznam naskenovaných okolních Wi-Fi sítí oproti databázi. Po úspěšné aktivaci je hráči připsán příslušný počet bodů. Pokud to má uživatel v nastavení povolené, je také odeslán naskenovaný seznam okolních bezdrátových sítí na server. Tím je udržována aktuální databáze přístupových bodů v okolí jednotlivých památek.

Dále je možné zobrazit seznam památek ve městě. Z něho je vidět, zda je daná památka navštívena, kolik bodů je za její aktivaci a jak je daleko od současné polohy uživatele. Památky lze řadit podle názvu nebo podle vzdálenosti. Kliknutím na památku ze seznamu se uživatel dostane na herní mapu, kde je příslušná památka označena.

V profilu uživatele jsou vidět osobní údaje zadané při registraci a aktuální postup pro stažené město. Je tak možné zjistit, kolik hráč získal bodů, kolik památek ve městě navštívil a kolik jich ještě zbývá navštívit. Součástí obrazovky s profilem uživatele je také historie návštěv, kde je vidět datum a čas, kdy byla památka navštívena.

Poslední funkcí je zobrazení aktuálního žebříčku uživatelů. V aplikaci jsou 2 typy žebříčků, mezi kterými je možné v nastavení přepínat. Jedná se o městský žebříček, ve kterém jsou zobrazeny body a pořadí uživatelů pouze pro aktuálně stažené město. Druhým žebříčkem je globální žebříček, ve kterém je součet bodů za návštěvy památek ve všech městech.

Pokud se uživatel odhlásí z aplikace, musí při dalším přihlášení znovu vybrat a stáhnout město do zařízení.



Obr. 15: Use Case diagram mobilní aplikace

Zdroj: vlastní, Enterprise Architect

Na obrázku č. 15 je zachyceno chování aplikace z pohledu uživatele pomocí Use Case diagramu, ve kterém je také naznačeno, v jakých případech dochází ke komunikaci s webovým API.

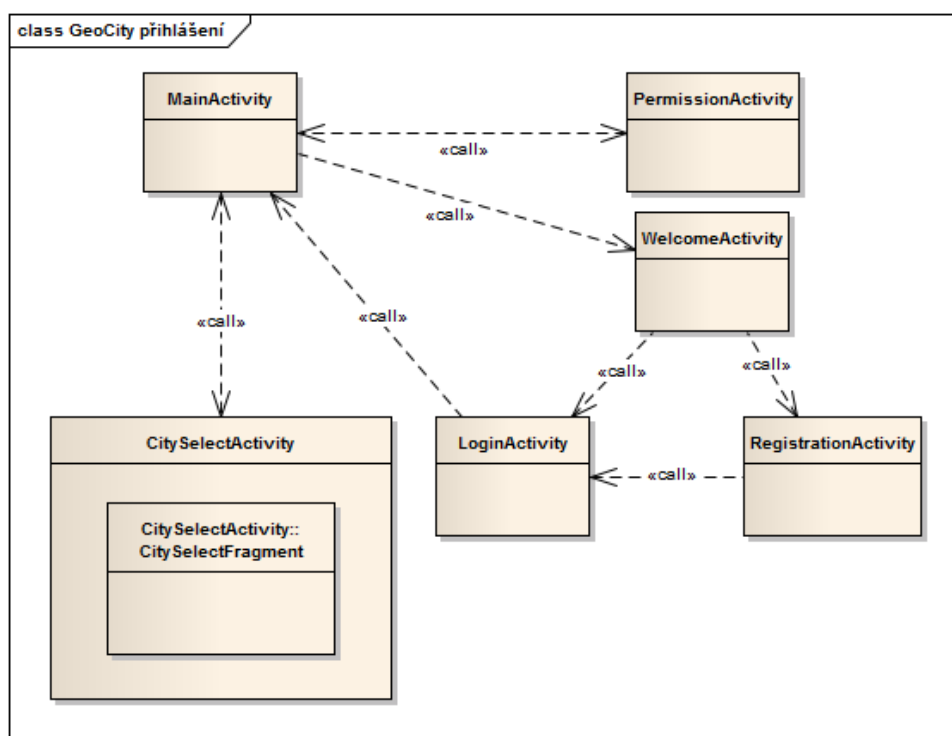
6.2.2 Návrh aplikace

Android se skládá z aplikačních komponent, které musejí být definovány v souboru *AndroidManifest.xml*. Komponenta *Activity* odpovídá jedné obrazovce. Obsahuje grafické uživatelské rozhraní, definované pomocí značkovacího jazyka XML, jehož prostřednictvím probíhá komunikace s uživatelem. Aplikace obsahuje obvykle více aktivit, mezi kterými je možné přepínat. Komponenta *Service* neposkytuje uživatelské rozhraní, ale představuje proces běžící na pozadí. Většinou se používá k provedení dlouhotrvajících úkonů nebo pro přístup ke vzdáleným zdrojům, kde

není známá doba odezvy. *Content provider* je rozhraní pro sdílení dat mezi aplikacemi. Poslední aplikační komponentou je *Broadcast receiver*, který reaguje na oznámení zaslané aplikací nebo operačním systémem.

Společně s uvedením operačního systému Android 3.0 (Honeycomb) byly představeny fragmenty. Každá aktivita v sobě může mít jeden či více fragmentů, který má definované vlastní grafické uživatelské rozhraní. Díky fragmentům je možné stavět sofistikovanější GUI pro větší obrazovky. Zároveň pomáhají škálovat zobrazení aplikace mezi malými a velkými displeji.

Před započítím implementace bylo nutné rozdělit aplikaci do jednotlivých obrazovek, navrhnout jejich rozložení a zvolit způsob, jak se mezi nimi bude uživatel navigovat.



Obr. 16: Volání aktivit - přihlášení do aplikace

Zdroj: vlastní, Enterprise Architect

Na obrázku č. 16 je znázorněno, jakým způsobem jsou provázány aktivity, které jsou použity při přihlášení uživatele. Na začátku je spuštěna hlavní aktivita s názvem *MainActivity*. Ta nejprve ověří, zda jsou aplikaci přidělena veškerá potřebná oprávnění.

Do verze Androidu 6.0 (Marshmallow) jsou oprávnění přidělována při instalaci programu. Od verze 6.0 včetně došlo k velké změně a vývojáři musejí kontrolovat oprávnění za běhu aplikace. Uživatel si může přesně vybrat, které oprávnění mobilní aplikaci přidělí a které zamítne.

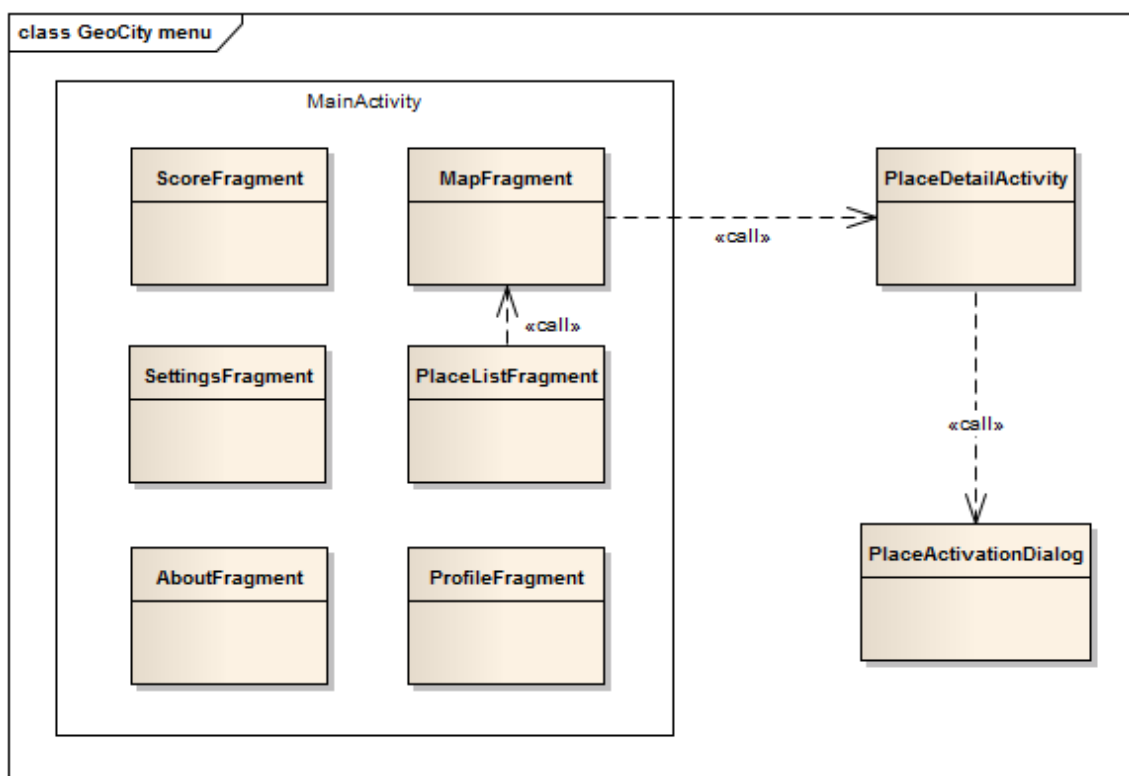
Android navíc rozděluje oprávnění do dvou kategorií. První z nich jsou tzv. normální oprávnění, která stačí uvést do manifestu. Oprávnění z této kategorie jsou považována za bezpečná a jsou aplikaci přidělena automaticky. Druhou kategorií jsou tzv. nebezpečná oprávnění, která musejí být také uvedena v manifestu. Navíc je ale o jejich přidělení potřeba od verze Androidu 6.0 explicitně požádat uživatele.

Vyvíjená aplikace vyžaduje z nebezpečných oprávnění přístup k přesné poloze mobilního zařízení (*android.permission.ACCESS_FINE_LOCATION*) a právo na zápis do externího úložiště (*android.permission.WRITE_EXTERNAL_STORAGE*). Tato dvě oprávnění jsou klíčová pro správný běh aplikace. Pokud je v *MainActivity* zjištěno, že aplikace těmito oprávněními nedisponuje, je spuštěna aktivita *PermissionActivity*. Jediným cílem této aktivity je získat potřebná oprávnění. Pokud se jí to nezdaří, nemá cenu dále pokračovat. Na obrazovce se zobrazí text o nedostatečném oprávnění společně s informací, jak tuto situaci napravit. V opačném případě se znovu zavolá *MainActivity*.

Dále se v *MainActivity* ověřuje, zda je uživatel přihlášen. Pokud není, zobrazí se uvítací obrazovka, která je reprezentována třídou *WelcomeActivity*. Na této obrazovce se může uživatel rozhodnout, zda se chce do aplikace registrovat nebo přihlásit. Na základě jeho rozhodnutí se volá buď přímo *LoginActivity* nebo *RegistrationActivity*, ze které se uživatel po vyplnění registračního formuláře

dostane na přihlašovací obrazovku. Po úspěšném přihlášení se zavolá znovu *MainActivity*.

Před spuštěním hlavní části aplikace je nutné stáhnout město do zařízení, aby se později co nejvíce šetřila mobilní data. Pokud není město staženo, volá se aktivita *CitySelectActivity*, ve které je vložen fragment *CitySelectFragment* se seznamem dostupných měst. Uživatelem vybrané město se předá službě *CityDownloadService*, která má na starosti jeho stažení na pozadí. Fragment je informován o průběhu stahování pomocí broadcastu. Jakmile je stahování dokončeno, zavolá se opět *MainActivity* a dojde ke spuštění hlavní části aplikace.

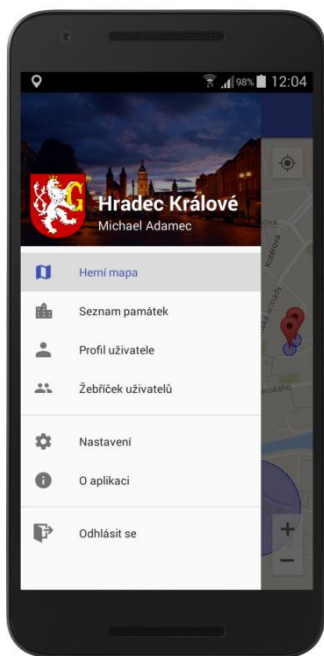


Obr. 17: Volání aktivit - hlavní část aplikace

Zdroj: vlastní, Enterprise Architect

Jak bylo popsáno v předchozích odstavcích, *MainActivity* je hlavní spouštěcí aktivita, ze které jsou v případě potřeby volány ostatní aktivity. Dochází k tomu, pokud aplikace nemá potřebná oprávnění, pokud není uživatel přihlášen nebo pokud v zařízení není staženo město.

Pro navigaci mezi jednotlivými obrazovkami v hlavní části aplikace byla zvolena pro Android typická vysouvací postranní nabídka. Ta je implementována právě ve třídě *MainActivity*. Přesun mezi jednotlivými obrazovkami je realizován výměnou fragmentů. Fragmenty, které se vkládají do *MainActivity* jsou přehledně zobrazeny na obrázku č. 17. Z fragmentu obsahující herní mapu je volána aktivita *PlaceDetailActivity*, která reprezentuje obrazovku s detailem památky. Po kliknutí na tlačítko „Aktivovat památku“ se vkládá do této obrazovky fragment *PlaceActivationDialog*, který zobrazuje průběh aktivace. Na obrázku č. 18 je vidět postranní nabídka aplikace.



Obr. 18: Postranní nabídka

Zdroj: vlastní, obrázek z mobilní aplikace

6.2.3 Implementace

Na počátku implementace mobilní aplikace pro Android bylo nutné určit, na jaké verze tohoto OS bude možné aplikaci nainstalovat. Čím starší verze Androidu je podporována, o to pracnější je pak vývoj celé aplikace. Pokud se zvolí minimální verze API příliš vysoko, aplikaci půjde nainstalovat jen na malé procento zařízení. Jde tedy o to najít vyváženou kombinaci mezi počtem podporovaných zařízení a pracností vývoje. Jako *targetSdkVersion* bylo zvoleno nejnovější API 23,

tj. Android 6.0 Marshmallow. Minimální verze API nutná pro spuštění programu je nastavena na číslo 15, tedy Android 4.0.3 s kódovým označením Ice Cream Sandwich. Dle statistiky obchodu Google Play z května 2016 je možné aplikaci s takto zvolenou minimální verzí API nainstalovat na 97,7% všech zaregistrovaných zařízení [35].

6.2.3.1 Použité technologie

Pro vývoj mobilní aplikace bylo použito oficiální vývojové prostředí Android Studio, které je založeno na platformě IntelliJ od společnosti JetBrains. Navíc aplikace využívá funkčnost několika knihoven. První z nich je Support Library přímo od společnosti Google. Tato knihovna zajišťuje zpětnou kompatibilitu se staršími verzemi operačního systému Android a zároveň umožňuje používat funkce představené společně s vyššími verzemi API. Knihovna Google Play Services zprostředkovává přístup ke geografické poloze mobilního zařízení a zároveň umožňuje použít mapové podklady od této společnosti. Knihovna Picasso [36] je v aplikaci použita pro práci s obrázky. Pro komunikaci s webovým API je využíváno knihovny Retrofit [36] s HTTP klientem OkHttp [36]. Tato knihovna umožňuje jak synchronní tak asynchronní volání a přijatou JSON odpověď dokáže velmi jednoduše převést na modelovou třídu. Aplikace využívá i dvě grafické knihovny. Jsou jimi Material-ish Progress [37] a Round Corner Progress Bar [38], díky kterým je možné přidat na obrazovku různé typy ukazatelů průběhu s konzistentním vzhledem napříč všemi verzemi Androidu.

6.2.3.2 Struktura aplikace

Aplikace využívá návrhový vzor Model-View-Controller (MVC), který vyplývá z frameworku poskytnutého platformou Android. Modelové třídy jsou umístěny v balíčku *model*. Jedná se o reprezentaci informací v podobě POJO (Plain old Java object), s nimiž aplikace pracuje. Vrstva view je v Androidu reprezentována pomocí XML souborů, které definují rozložení jednotlivých grafických komponent na obrazovce. Tyto XML soubory se nacházejí ve zdrojové složce */res/layout*. Controller v aplikaci slouží jako řídicí vrstva. V Androidu je reprezentována aktivitou nebo fragmentem.

Projekt se skládá celkem ze 7 aktivit, 12 fragmentů, 4 služeb, 6 rozhraní a dalších 34 tříd, které jsou rozděleny do následujících 8 hlavních balíčků. Každý balíček představuje určitou část aplikace.

activities – tento balíček obsahuje veškeré aktivity. Každá aktivita reprezentuje jednu obrazovku na mobilním zařízení. V tomto balíčku je obsažena například spouštěcí aktivita s názvem *MainActivity*.

adapters – balíček obsahuje adaptéry pro všechny seznamy, které jsou v aplikaci použity. Tedy například adaptér pro seznam památek nebo žebříček uživatelů. Aby bylo listování seznamem co nejplynulejší, je použit návrhový vzor „view holder“ [39].

api – zde jsou obsaženy třídy používané pro komunikaci s webovým API. Hlavní třídou tohoto balíčku je *RestClient*, kde je vytvořen adaptér pro komunikaci s REST službou. Za zmínku stojí i třída *ApiKeyRequestInterceptor* s metodou *intercept()*. Pokud je uživatel přihlášen do aplikace, tato metoda automaticky vkládá unikátní API klíč do každého požadavku. Balíček je dále rozdělen na *model* a *services*. V podbalíčku *model* se nachází modelové třídy všech možných API odpovědí a v podbalíčku *services* jsou definovány jednotlivé API požadavky.

database – veškeré památky a informace o nich jsou po stažení z webového API uloženy do databáze. Tento balíček obsahuje třídy a metody potřebné pro komunikaci s SQLite databází. Pro přístup k datům byl zvolen návrhový vzor „Table Data Gateway“, který zapouzdřuje operace nad příslušnou databázovou tabulkou.

fragments – v balíčku jsou obsaženy všechny fragmenty a dialogy, které se nacházejí v aplikaci. Je zde například *MapFragment* obsahující obrazovku s herní mapou a *PlaceActivationDialog*, který má na starost proces aktivace památky.

model – tento balíček obsahuje modelové třídy aplikace. Je zde například třída *Place*, která reprezentuje památku.

others – zde jsou umístěny pomocné třídy, které jsou používány na různých místech v aplikaci.

services – poslední balíček obsahuje čtyři služby, které provádějí dlouhotrvající práci na pozadí a nemají grafické uživatelské rozhraní. Jsou jimi *LocationService*, *GeofenceTransitionsService*, *CityDownloadService* a *WifiUploadService*. Služba *CityDownloadService* má za úkol stáhnout na pozadí vybrané město. Uživatel tak může během stahování města provádět na zařízení jinou činnost. *WifiUploadService* vyhledá v okolí památky dostupné Wi-Fi sítě a pokud to uživatel v nastavení aplikace nezakázal, odešle je na server. Zbylé dvě služby budou detailněji popsány v následujících odstavcích.

6.2.3.3 GPS poloha

Existují v zásadě dva způsoby jak určit GPS polohu v operačním systému Android. Prvním způsobem je přímý přístup k GPS modulu pomocí *GPS Location Provider*. Novější a v dokumentaci doporučovaný způsob je však využít *Fused Location Provider*. Na rozdíl od přímého přístupu k GPS toto API kombinuje více způsobů určení polohy a to pomocí GSM sítě, Wi-Fi sítí a GPS. V aplikaci je tedy implementován tento druhý způsob. Je však nutné, aby měl uživatel nainstalovanou službu *Google Play Services*. Ta je automaticky dostupná v každém telefonu, který má přístup do obchodu Google Play.

Zjišťování polohy zařízení probíhá v již dříve zmiňované službě *LocationService*. Před použitím *Fused Location Provider API* je nejprve nutné do manifestu přidat oprávnění pro určení přesné polohy. Dále se musí aplikace při vytváření služby v metodě *onCreate()* připojit k *LocationServices API*.

Jakmile dojde k úspěšnému připojení, je zavolána metoda *onConnected()*. V této metodě dochází k zaregistrování služby *LocationService* implementující rozhraní *LocationListener* k přijímání aktualizací o poloze. Během vytváření požadavku se nastavují celkem tři parametry. Prvním parametrem je interval aktualizací. Ten je v aplikaci nastaven na 2 sekundy, aby byl pohyb po mapě dostatečně plynulý. Dále

je nutné nastavit nejkratší časový interval, ve kterém je aplikace schopna aktualizace o poloze zpracovávat. Tento parametr je nastaven na 1 sekundu. Poslední parametr určuje prioritu požadavku, na jejímž základě API určí, jaký má použít zdroj lokalizačních dat. Tato priorita je v aplikaci nastavena na nejlepší dostupnou polohu.

Upravená ukázka kódu ze souboru *LocationService.java*:

```
@Override
public void onConnected(Bundle bundle) {
    Log.d(TAG, "Připojeno ke GoogleAPI");
    // vytvoření požadavku na polohu
    LocationRequest locationRequest = new LocationRequest();
    locationRequest.setInterval(2000); // 2 sekundy
    locationRequest.setFastestInterval(1000); // 1 sekunda
    locationRequest.setPriority(LocationRequest.PRIORITY_HIGH_ACCURACY);

    // požadavek o zasílání aktualizací polohy
    try {
        LocationServices.FusedLocationApi.requestLocationUpdates(
            googleApiClient, locationRequest, this);
    } catch (SecurityException e) {
        Log.e(TAG, "Aplikace nemá dostatečná oprávnění.");
    }
}
```

Po vytvoření požadavku na přijímání aktualizací o poloze dochází k periodickému volání metody *onLocationChanged()*, ve které je předávána aktuální poloha mobilního zařízení.

Upravená ukázka kódu ze souboru *LocationService.java*:

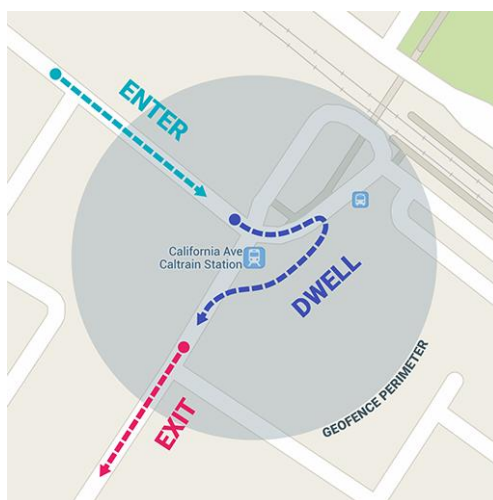
```
@Override
public void onLocationChanged(Location location) {
    if (location.getAccuracy() > MINIMUM_GPS_ACCURACY) {
        Log.d(TAG, "Presnost je prilis nizka. Poloha ignorovana.");
        return;
    }
    geofenceMonitor.start(); // start geofencingu, pokud již neběží
    lastLocation = location; // uložení polohy
    // vyslat broadcast
    Intent intent = new Intent(LocationService.BROADCAST_LOCATION_UPDATE);
    LocalBroadcastManager.getInstance(this).sendBroadcast(intent);

    // předání nových souřadnic mapám
    if (onLocationChangedListener != null) {
        onLocationChangedListener.onLocationChanged(location);
    }
}
```

Na začátku metody je zkontrolována přesnost předané polohy. Pokud je nižší jak 30 metrů, poloha je ignorována. Následně je nově získaná poloha uložena a je odeslán broadcast informující o její změně. Na konci metody dochází také k předání polohy mapám, které ji vykreslí. Služba *LocationService* běží na pozadí po celou dobu běhu aplikace. Jednotlivé aktivity a fragmenty se k této službě připojují, pokud potřebují znát aktuální informace o geografické poloze. Jakmile aktivita nebo fragment zachytí službou vyslaný broadcast, zavolá se metoda *getLastLocation()*, která vrátí naposledy získanou polohu zařízení.

6.2.3.4 Geofencing

Kromě určení polohy mobilního telefonu byl dalším řešeným problémem tzv. geofencing. Jedná se o událost, kdy uživatel vstoupí do nějaké vymezené oblasti na mapě nebo ji naopak opustí. Android poskytuje pro detekci těchto událostí *Geofencing API*. API dokáže upozornit na vstup do oblasti, opuštění oblasti a pobyt v oblasti po určitou dobu (viz obr. č. 19).



Obr. 19: Geofence události

Zdroj: <https://developer.android.com/images/training/geofence.png>

Každá památka má přiřazenou zeměpisnou šířku, zeměpisnou délku a poloměr sledované oblasti. Jakmile služba *LocationService* obdrží první polohu s dostatečnou přesností, jsou všechny nenavštívené památky ve městě předány *Geofencing API*. V případě, že uživatel vstoupí nebo vystoupí z hlídané oblasti,

dojde k upozornění služby *GeofenceTransitionsService*, která má za úkol geofence události zpracovávat. Služba následně vyšle broadcast obsahující id památky a typ události. Tento broadcast zachytí fragment obsahující herní mapu. Pokud uživatel vstoupil do modrého aktivačního kruhu okolo nenavštívené památky, je na mapu vykresleno tlačítko, pomocí kterého je možné zobrazit detail příslušné památky a zahájit proces aktivace. V případě, že broadcast obsahuje informaci o opuštění hlídané oblasti, je tlačítko památky skryto. Nevýhodou poskytnutého *Geofencing API* je nemožnost přidat více jak 100 hlídaných oblastí. S tím již ale bylo v návrhu aplikace počítáno a památky jsou rozděleny podle měst, ve kterých se nachází.

6.2.3.5 Podvržení GPS polohy

Na platformě Android je velmi jednoduché podvrhnout svoji GPS polohu. Stačí provést několik následujících kroků.

1. Odkrytí položky „Vývojářské možnosti“ v nastavení telefonu.
2. V této nabídce aktivovat zaškrtačací políčko „Povolit fiktivní polohy“.
3. Stáhnout aplikaci z obchodu Google Play, která dokáže nasimulovat fiktivní GPS polohu. Například se může jednat o aplikaci s názvem „Fake Location Spoofer Free“ [40].

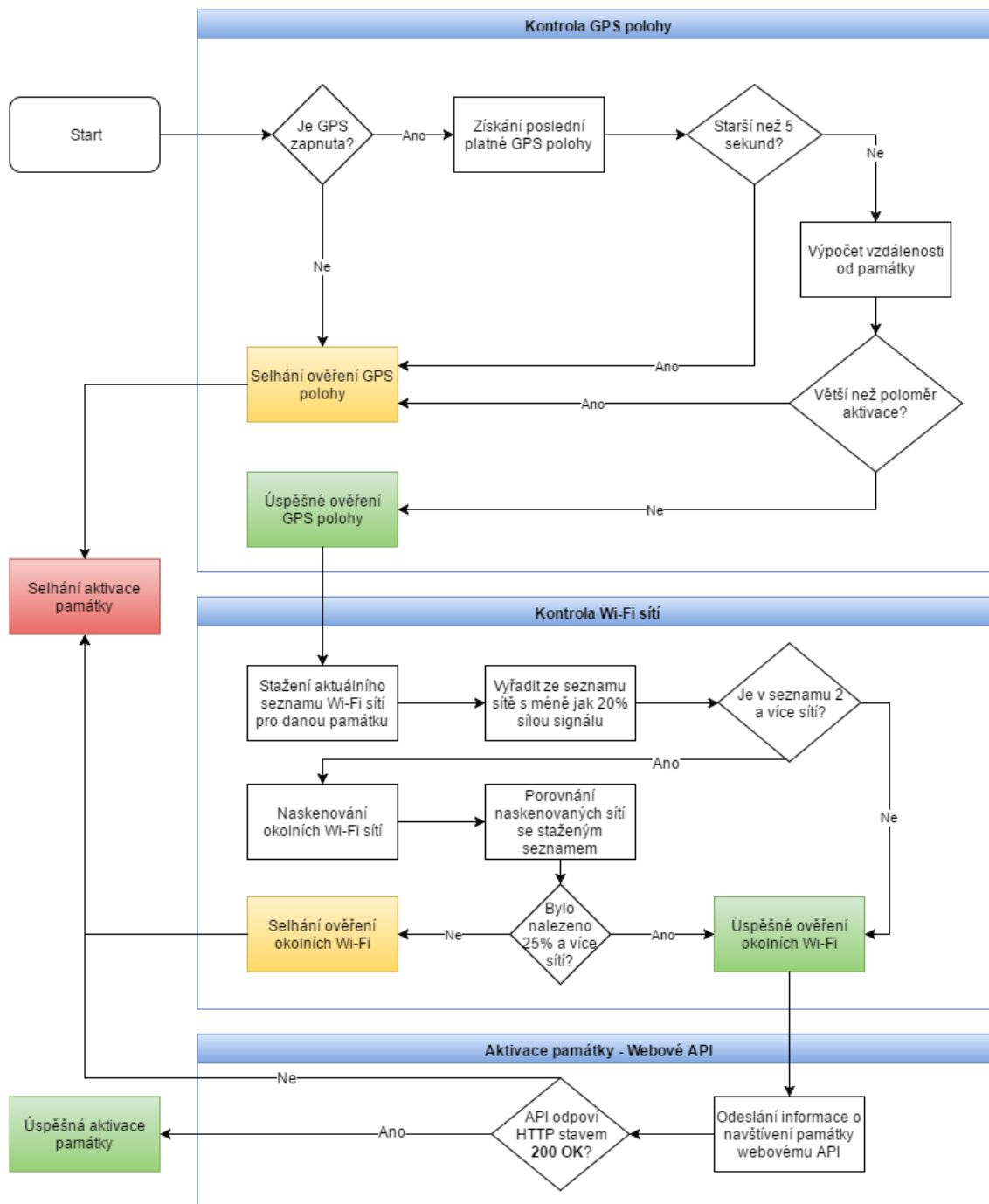
V nainstalované aplikaci si pak stačí na mapě zvolit svoji fiktivní polohu, která je předávána ostatním aplikacím místo skutečné polohy získané z GPS modulu. Uživatel by tak mohl aktivovat veškeré památky na mapě bez nutnosti se nacházet fyzicky v jejich blízkosti. Za každou aktivovanou památku by mu byly neoprávněně přiděleny body a posunul by se v žebříčku uživatelů na přední místa. To by nebylo spravedlivé vůči ostatním uživatelům, kteří poctivě obcházelí a navštěvovali jednotlivé památky.

Obrana proti podvrženým GPS souřadnicím bohužel není jednoduchá. Od verze API 18 je k dispozici na objektu třídy *Location* metoda *isFromMockProvider()*, která vrací hodnoty *true* nebo *false* v závislosti na tom, zda je předaná pozice

nasimulována. Pro projekt však bylo zvoleno minimální API ve verzi 15. Tuto metodu tedy není možné použít. Další možností je zkontrolovat, zda je v nastavení zapnuta položka „Povolit fiktivní polohy“ a případně vůbec nedovolit zahájit proces aktivace památky. Některé aplikace ale tuto funkci používají pro legitimní účely a nutit uživatele tuto položku v nastavení neustále vypínat by nebylo ideální. Byl tedy navržen vlastní způsob obrany, který bude detailněji popsán v následující podkapitole.

6.2.3.6 Proces aktivace památky

Jak je vidět z vývojového diagramu na obrázku č. 20, proces aktivace památky je značně komplikovaný.



Obr. 20: Proces aktivace památky

Zdroj: vlastní, <https://www.draw.io>

Aktivaci je možné vyvolat z obrazovky detailu nenavštívené památky. Je však nutné být připojen k internetu, jinak se proces aktivace vůbec nespustí. Nejprve je kontrolována GPS poloha. Ta nesmí být starší 5 sekund a zároveň přesnost polohy nesmí být nižší než 30 metrů. Poloha s nižší přesností je považována za neplatnou.

Následně je vypočítána vzdálenost aktuální pozice uživatele od zeměpisných souřadnic památky a to pomocí metody *distanceBetween()* ze třídy *Location*. Pokud je vzdálenost vyšší než aktivační poloměr památky, proces aktivace selže. V opačném případě je považováno ověření GPS polohy za úspěšné.

Jak již bylo zmíněno v předcházející podkapitole, na platformě Android je relativně jednoduché nasimulovat svoji aktuální GPS polohu. Aby bylo zamezeno případnému podvádění, jsou kromě GPS polohy zároveň kontrolovány i přilehlé Wi-Fi přístupové body oproti databázi. Nejprve je ze serveru stažen aktuální seznam Wi-Fi sítí pro danou památku. Pokud stažení aktuálního seznamu selže, je použit seznam sítí získaný při přihlášení uživatele a stažení města do zařízení. Poté jsou ze seznamu vyřazeny sítě, které mají méně jak 20% síly signálu. Tento krok je prováděn z důvodu různé citlivosti Wi-Fi přijímačů na jednotlivých mobilních zařízeních. Ověření polohy tak dále probíhá pouze s dostatečně silnými Wi-Fi sítěmi. Následně je zkontrolován počet sítí v seznamu. Pokud pro památku nejsou zaregistrovány alespoň 2 dostatečně silné sítě, je ověřování pozice pomocí bezdrátových sítí přeskočeno a považováno za úspěšně dokončené. V opačném případě mobilní zařízení naskenuje Wi-Fi přístupové body v okolí. Tyto sítě jsou následně porovnány se seznamem získaným z databáze. Jestliže se shoduje alespoň 25% sítí, je ověření polohy považováno za úspěšné. Pokud je v okolí památky nalezena méně jak čtvrtina sítí, ověření polohy a tím i proces aktivace památky selže.

V případě, že se podaří úspěšně ověřit polohu, je odeslána informace o navštívení konkrétní památky na server. Jakmile se povede požadavek mobilního klienta zpracovat a server odpoví HTTP stavem 200, je proces aktivace dokončen a památka je považována za navštívenou.

Ověření polohy pomocí Wi-Fi sítí řeší zmiňovaný problém s podvržením GPS souřadnic. Aby bylo možné vytvořit ke každé památce seznam bezdrátových sítí v okolí a zároveň ho udržovat aktuální, uživatelé odesílají po úspěšné aktivaci

nalezené bezdrátové sítě na server. Tato možnost je volitelná a lze ji v nastavení aplikace vypnout.

Odeslaný seznam sítí webové API zpracovává následujícím způsobem. Jestliže je u konkrétní památky zaměřena stejná Wi-Fi síť třikrát po sobě různými uživateli, je přidána do seznamu sítí dané památky. Síla signálu, která určuje, zda je síť dostatečně silná pro ověření polohy, se průměruje na základě všech měření odeslaných z různých mobilních zařízení. Ostatní uživatelé pak tuto síť musejí při ověřování polohy zaměřit, aby bylo možné památku aktivovat. Samozřejmě se může stát, že dříve zaměřená síť přestane vysílat. Je tedy nutné neaktivní Wi-Fi sítě ze seznamů odebírat. To se stane za předpokladu, že se pěti po sobě jdoucím uživatelům nepodaří danou síť u příslušné památky zaměřit.

Bezdrátové Wi-Fi sítě jsou tedy k jednotlivým památkám přidávány, pokud je uživatelé třikrát po sobě zaměří. Zároveň jsou ale i odebírány, pokud se je pětkrát po sobě zaměřit nepovede. Díky tomu jsou seznamy sítí pro jednotlivé památky neustále aktuální a je možné je spolehlivě využít při ověřování polohy uživatele.

6.2.3.7 Synchronizace profilu uživatele

Dalším řešeným problémem byla synchronizace profilu uživatele. Uživatel by měl mít možnost používat aplikaci na jakémkoliv svém zařízení a měl by na něm mít vždy aktuální data. Tedy hlavně označené památky, které již navštívil, a počet bodů, který za jejich navštívení získal. Existují v zásadě dva způsoby, jak synchronizaci profilu řešit.

Prvním z nich je, že při jakékoliv změně v profilu uživatele by byla zaslána zpráva ze serveru na mobilní telefon. Zpráva by mohla například obsahovat příslušné změny v profilu nebo informaci, že si má telefon stáhnout nové údaje. Jedná se o tzv. Push notifikaci ze strany serveru. To například umožňuje služba Google Cloud Messaging. Nejedná se však o zcela jednoduchou funkčnost na implementaci.

Druhým způsobem jsou periodické dotazy na server, zda nedošlo ke změně v profilu od posledního požadavku. Tento způsob je sice jednodušší na implementaci, ale má určité nevýhody. Zbytečně zatěžuje mobilní internetové připojení a server dotazy, i když nedošlo k žádné změně. Zároveň je také obtížné nastavit, v jakém intervalu mají být změny kontrolovány. Ve výsledku by se tak mohlo stát, že stejně údaje v telefonu nebudou aktuální.

Nakonec byl vytvořen do určité míry vlastní způsob synchronizace. Během přihlášení do aplikace je uživateli vygenerován unikátní API klíč, s pomocí kterého jsou prováděny všechny dotazy na webové aplikační rozhraní. Během stahování města dojde zároveň ke stažení dat o tom, která z památek již byla navštívena a kolik za její navštívení dostal uživatel bodů. Pokud se uživatel přihlásí do aplikace z dalšího zařízení, bude mu vygenerován nový API klíč. API klíč získaný pro první zařízení se stane v daný okamžik neplatný a dojde k automatickému odhlášení z aplikace. Hru je tedy možné hrát v jeden okamžik pouze na jediném zařízení. Tím je zajištěno, že stačí stáhnout profilová data uživatele pouze při přihlášení do aplikace a není nutné je v průběhu hry dále synchronizovat.

Zároveň zvolený způsob synchronizace profilu zajišťuje ochranu před podváděním, kdyby se na stejný účet mohlo přihlásit současně více uživatelů a dosáhnout tak rychleji lepšího bodového hodnocení.

6.2.3.8 Šetření mobilních dat

Jedním z cílů práce bylo vymyslet způsob, jak co nejvíce šetřit mobilní data. Návrh aplikace počítá s tím, že se do ní uživatel přihlásí a stáhne si město pomocí webového API přes Wi-Fi připojení. Do zařízení se stáhnou fotografie památek a veškeré informace, které aplikace potřebuje ke svému chodu. Zároveň dochází během stahování města k synchronizaci profilu. Tímto se ušetří spousta dat, která by se jinak musela stahovat přes mnohdy pomalé mobilní připojení, na které jsou většinou aplikovány různé FUP limity a omezení. Mobilní data jsou tedy potřeba pouze v následujících situacích.

- Aktivace památky
 - Stažení aktuálního seznamu Wi-Fi sítí
 - Odeslání informace webovému API o navštívení památky
 - Odeslání okolních bezdrátových sítí na server (*pouze pokud to má uživatel v nastavení aplikace povoleno*)
- Stažení aktuálního žebříčku uživatelů
- Mapové podklady

Pro aplikaci byly zvoleny mapové podklady od společnosti Google. Jejich výhodou je relativně jednoduché použití v rámci operačního systému Android. Zároveň jsou uživatelé tohoto systému důvěrně seznámeni s funkcemi a vizuální podobou Google map. Nevýhodou tohoto řešení je nutnost být připojen k internetu. Nejvíce dat je tak přenášeno právě kvůli mapovým podkladům. Pomocí cachování byl však tento objem přenášených dat snížen na minimum.

7 Testování aplikace a výsledky

Testování aplikace bylo rozděleno do několika částí. Před začátkem vývoje mobilní aplikace bylo nutné otestovat, zda je naprogramované webové API plně funkční a na všechny dotazy vrací očekávané JSON odpovědi. Webová část byla vyvíjena v aplikaci PhpStorm. Toto vývojové prostředí obsahuje nástroj REST Client, který dovoluje důkladné testování jednotlivých HTTP požadavků bez nutnosti mít hotovou klientskou část aplikace. Nástroj umožňuje přidat jednotlivé parametry dotazu, a to včetně hlavičky obsahující přidělený API klíč. Odpověď je pak naformátována a přehledně zobrazena spolu s návratovým stavovým kódem, celkovým časem komunikace a velikostí přijaté zprávy.

Po otestování webového API bylo přistoupeno k implementaci mobilní aplikace pro operační systém Android. Ta byla vyvíjena na telefonu Samsung Galaxy S4 Mini obsahující Android ve verzi 4.4.2 s kódovým označením KitKat. Aplikace je však cílena na nejnovější Android 6 s kódovým označením Marshmallow. Bohužel, zařízení s touto verzí Androidu nebylo během vývoje fyzicky k dispozici. Veškerá funkčnost na nejaktuálnější verzi systému tak musela být ověřena pomocí emulátoru.

V první fázi testování mobilní aplikace bylo zjišťováno, zda se u jednotlivých památek nachází dostatečné množství bezdrátových sítí s přijatelnou kvalitou vysílaného signálu. Bylo vybráno sedm nejvýznamnějších památek v Hradci Králové dle zdroje [41]. U těchto památek bylo provedeno důkladné měření okolních bezdrátových sítí s pomocí upravené mobilní aplikace, která výsledky nahrávala přímo do databáze. Měření probíhalo u každé památky celkem pětkrát v různých vzdálenostech a na více přístrojích, aby byla zajištěna co nejvyšší přesnost. Konkrétně se jednalo o zařízení Samsung Galaxy S4 Mini a Samsung Galaxy S3.

Výsledky měření jsou uvedeny v tabulce číslo 2. Ve druhém sloupci je počet Wi-Fi sítí, které se povedlo u památky naměřit. Tento počet zahrnuje i síť s velmi slabým signálem, které se povedlo zachytit pouze při jednom z pěti pokusů na konkrétním

zařízení. Třetí sloupec obsahuje počet Wi-Fi sítí, které se povedlo zaměřit minimálně třikrát z pěti pokusů. Ve druhé části tabulky se nachází průměrná síla signálu a medián síly signálu. Tato statistická data byla vypočítána pouze ze sítí, které se povedlo zaměřit minimálně třikrát. Výkon je udáván v jednotkách decibel nad miliwattem [dBm]. Navíc mobilní aplikace přepočítává naměřenou sílu signálu na procenta, která jsou v tabulce uvedena v závorkách.

Název památky	Počet Wi-Fi v okolí památky	Počet Wi-Fi v okolí - 3x zaměřeno	Průměrná síla signálu [dBm]	Medián síly signálu [dBm]
Bílá věž	29	10	-82,70 (38,8%)	-83 (38%)
Katedrála sv. Ducha	34	12	-78,08 (49,5%)	-79 (48%)
Mariánský sloup	70	33	-82,39 (39,5%)	-82 (39%)
Kostel Nanebevzetí Panny Marie	69	22	-82,55 (38,5%)	-83 (37%)
Muzeum východních Čech	91	38	-80,21 (44,1%)	-80 (44%)
Vodní elektrárna Hučák	31	6	-81,83 (40,0%)	-83 (39%)
Kostel sv. Mikuláše	6	0	-	-

Tabulka 2: Počet a síla signálu Wi-Fi v okolí vybraných památek

Zdroj: vlastní, naměřeno pomocí aplikace

Z tabulky je patrné, že ve městě je u většiny památek dostatečný počet bezdrátových sítí. Navrhovaný způsob dodatečného ověření polohy zařízení tak bylo možné implementovat. Problém však nastal u kostela sv. Mikuláše, který se nachází v parku. V okolí této památky bylo naměřeno pouze šest bezdrátových sítí s velmi slabým vysílaným signálem. Ani jednu síť se nepovedlo opakovaně zaměřit.

Aplikace při aktivaci památky ověřuje nejprve GPS souřadnice a následně porovnává okolní Wi-Fi sítě oproti databázi. Na základě měření byly nastaveny již dříve popisované podmínky. U památky musí být zaznamenány minimálně dvě bezdrátové sítě a síla signálu u těchto sítí musí přesahovat 20%. Pokud jsou tyto podmínky porušeny, aplikace během aktivace památky přeskočí kontrolu bezdrátových sítí. V opačném případě musí být nalezeno 25% bezdrátových sítí se sílou signálu více jak 20% pro úspěšné provedení aktivace a následné přidělení bodů.

Mobilní aplikace odesílá na server mimo dat, která se používají pro dodatečné ověření polohy uživatele, i informaci o použitém zabezpečení bezdrátové sítě. Pro zajímavost je zde možné uvést výsledky získané během testování. Celkově bylo provedeno 35 měření u 7 památek ve městě. Během tohoto měření bylo zaznamenáno 261 unikátních bezdrátových sítí. Unikátnost je určena na základě odlišných MAC adres. Z těchto sítí jich 199 (76,2%) používalo šifrování WPA2, 19 (7,3%) šifrování WPA, 11 (4,2%) šifrování WEP a 32 (12,3%) sítí bylo nezabezpečeno. Většina sítí tedy používá zabezpečení WPA (Wi-Fi Protected Access), které je při zvolení dostatečně dlouhého hesla jen velmi těžko prolomitelné. Na druhou stranu, u 11 sítí bylo zaznamenáno zastaralé zabezpečení WEP (Wired Equivalent Privacy) z roku 1997. Prolomení takto zabezpečené sítě je otázkou několika málo minut.

V druhé fázi testování byla mobilní aplikace nainstalována pěti různě pokročilým uživatelům k podrobnému otestování. Testování probíhalo na těchto zařízeních.

- Samsung Galaxy S4 mini, Android 4.4.2 (API 19), 4,3 palce, 540x960 pixelů
- Samsung Galaxy S3, Android 4.3 (API 18), 4,8 palce, 720x1280 pixelů
- Samsung Galaxy S3 mini, Android 4.4.2 (API 19), 4 palce, 480x800 pixelů
- Samsung Galaxy S2, Android 4.1.2 (API 16), 4,3 palce, 480x800 pixelů
- LG Nexus 5X, Android 6.0.1 (API 23), 5,2 palce, 1080x1920 pixelů

Všichni uživatelé měli za úkol se přihlásit do aplikace, stáhnout město Hradec Králové a navštívit minimálně dvě památky. Během tohoto testování bylo odhaleno hned několik menších problémů, které bylo potřeba vyřešit.

První možné vylepšení bylo objeveno hned při úvodním spuštění aplikace. Uživatelé si stěžovali, že při přihlášení musí zdlouhavě zadávat do přihlašovacího formuláře tytéž údaje, které vyplňovali při registraci o krok dříve. Nyní tedy dochází k automatickému přihlášení ihned po registraci, bez nutnosti zadávat

přihlašovací údaje. Navíc byl registrační a přihlašovací formulář vylepšen o napovídání emailových adres, které má uživatel uloženy v zařízení.

Po přihlášení do aplikace se zobrazí seznam měst dostupných ke stažení. Zde byl na základě uživatelské odezvy přidán dialog, který varuje před stažením velkého množství dat přes mobilní internetové připojení. Nemělo by se tak stát, že uživatel nevědomky vyčerpá značnou část svého FUP limitu.

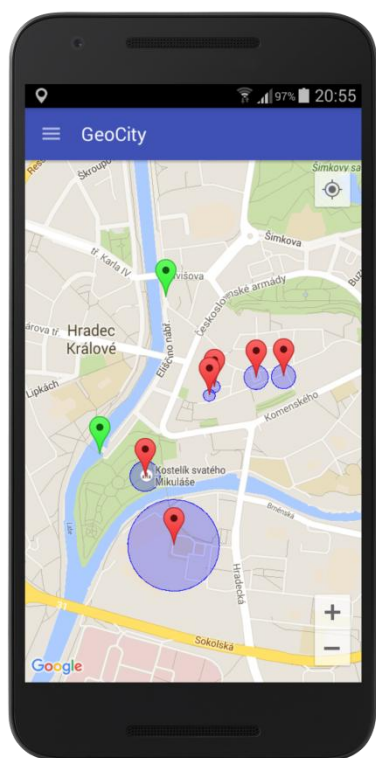
Dále uživatelé hlásili, že se jim v určitých případech objevuje dialog s následující zprávou: „Došlo k chybě při komunikaci se serverem“. Po bližším prozkoumání byl objeven nedostatek v nastavení času, jak dlouho mobilní aplikace čeká na odpověď serveru, než tuto chybu zahlásí. Během testování se ukázalo, že v základu nastavených deset sekund je nedostatečně dlouhá doba. Tento čas byl tedy navýšen na dvojnásobek.

Dalším problémem bylo špatné informování uživatele o cachování žebříčku s pořadím. Aby byla šetřena mobilní data, je žebříček načítán z interní paměti mobilního zařízení. Tato data se obnovují nejdříve jednou za 30 minut, pokud je k dispozici internetové připojení. Uživatelé o tomto faktu však nebyli v GUI nijak informováni. Mohlo se tak stát, že uživatel po navštívení památky viděl v žebříčku zastaralá data s menším počtem bodů, než ve skutečnosti v danou chvíli měl. Tento problém byl vyřešen následujícím způsobem. Pokud se žebříček načítá z interní paměti, je zobrazena lišta, která uživatele informuje o stáří právě zobrazovaných dat. Uživatel má také možnost si kdykoliv stáhnout aktuální žebříček z internetu pomocí tlačítka umístěného v horní liště aplikace.

Poslední problém byl odhalen pouze na zařízení LG Nexus 5X s poslední vydanou verzí operačního systému Android. Na tomto telefonu došlo k tomu, že na mapovém podkladu nebyla vůbec vykreslena uživatelova poloha a v horní části obrazovky byla zobrazena lišta s oznámením o probíhající lokalizaci. K určení polohy však za celou dobu spuštění aplikace nedošlo. Po pečlivém prozkoumání logů z aplikace byl odhalen problém ve službě *LocationService*, která má za úkol

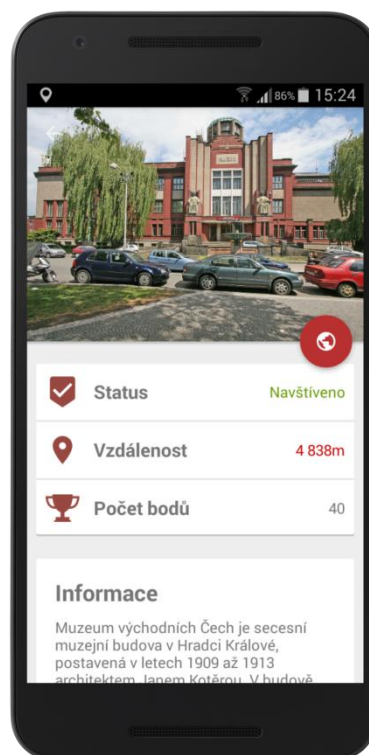
předávat získanou polohu mapám. Zde se aplikace spoléhala na *GpsStatus.Listener*, který aplikaci předával informace o stavu GPS modulu. Bohužel v nové verzi operačního systému nebyl tento listener vůbec volán. Aplikace měla za to, že nedošlo k zafixování GPS pozice. Proto nebyly předávány aktuální GPS souřadnice mapovému podkladu. Problém byl vyřešen přidáním anonymního *BroadcastReceiveru* pro akci *LocationManager.PROVIDERS_CHANGED_ACTION*, který tuto funkčnost na novém Androidu zastal.

Po odstranění výše zmíněných problémů byli testovaní uživatelé s aplikací spokojeni. Chválili její grafické zpracování, nízkou spotřebu mobilních dat a celkovou rychlost odezvy. Výsledkem této práce je plně funkční geolokační aplikace pro operační systém Android. Pro ilustraci jsou zde uvedeny následující dva obrázky pořízené přímo z naprogramované aplikace. Náhledy ostatních obrazovek si je možné prohlédnout v příloze číslo 1.



Obr. 21: Herní mapa

Zdroj: vlastní, obrázek z mobilní aplikace



Obr. 22: Detail památky

Zdroj: vlastní, obrázek z mobilní aplikace

8 Závěr

Úvodní část diplomové práce seznamuje s vývojem navigace od jejího počátku po současnost. V další kapitole jsou detailně popsány a navzájem srovnány způsoby určování geografické polohy na mobilním zařízení. Jsou zde vysvětleny principy lokalizace telefonu pomocí GSM sítě, bezdrátových Wi-Fi sítí a GPS. Liší se především v přesnosti, rychlosti určení polohy, množství spotřebované energie a prostředí, ve kterém je daný způsob možné použít.

V dnešní době je běžně dostupné velké množství aplikací a her pro mobilní zařízení. Asi nejnámější geolokační aplikací využívající prvky rozšířené reality je online hra s názvem Ingress. Cílem diplomové práce nebylo konkurovat takto vyspělým aplikacím, ale spíše se jimi inspirovat a vytvořit plně funkční mobilní aplikaci pro operační systém Android, která bude využívat popsané způsoby lokalizace.

Základní myšlenkou vytvořené aplikace je provést uživatele po historických památkách ve městě. Uživatel má za úkol dojít k vyznačenému místu na mapě a provést aktivaci příslušné památky, za kterou dostane přidělen určitý počet bodů. Hráči pak spolu soupeří o umístění v žebříčku. Zároveň aplikace umožňuje synchronizovat herní profil, kde se mimo jiné nachází historie navštívených památek.

Aby bylo možné stanovenou funkcionalitu implementovat, musela být aplikace rozdělena do dvou částí. První částí je jednoduché REST API, které bylo napsáno ve skriptovacím jazyce PHP. Pro zjednodušení a zrychlení vývoje byl použit Slim Framework. Přenos dat mezi serverem a mobilní aplikací probíhá v datovém formátu JSON, který byl zvolen kvůli jednoduchosti implementace a také vzhledem k menšímu objemu dat, který tento formát oproti XML při přenosu spotřebuje.

Druhou částí je mobilní aplikace pro operační systém Android. Existují v zásadě dva způsoby, jak určit přesnou polohu na tomto operačním systému. Prvním způsobem je přímý přístup k GPS modulu pomocí *GPS Location Provider*. Novější

a v dokumentaci doporučovaný způsob je však využít *Fused Location Provider*. Na rozdíl od přímého přístupu k GPS toto API kombinuje více způsobů určení polohy, a to pomocí GSM sítě, Wi-Fi sítí a GPS. V aplikaci je tedy implementován tento druhý způsob.

Kromě určení polohy mobilního telefonu byl dalším řešeným problémem tzv. geofencing. Android poskytuje pro detekci těchto událostí *Geofencing API*, které dokáže upozornit na vstup do oblasti, opuštění oblasti a pobyt v oblasti po určitou dobu. Nevýhodou poskytnutého aplikačního rozhraní je nemožnost přidat více jak 100 hlídaných oblastí. S tímto již bylo ale v návrhu aplikace počítáno a památky jsou rozděleny podle měst, ve kterých se nachází.

Během analýzy problematiky určování polohy bylo zjištěno, že v operačním systému Android je relativně jednoduché nasimulovat a podvrhnout svoji aktuální GPS polohu. Bylo proto navrženo a implementováno dodatečné ověření polohy na základě otisků bezdrátových sítí v okolí památky.

Naprogramovaná aplikace se snaží co nejméně využívat mobilní datové připojení. Návrh aplikace počítá s tím, že se do ní uživatel přihlásí a stáhne si město pomocí webového API přes Wi-Fi připojení. Do zařízení se stáhnou fotografie památek a většina dalších informací, které aplikace potřebuje ke svému chodu. Tím se ušetří značný objem dat, která by se jinak musela stahovat přes mnohdy pomalé mobilní připojení, na které jsou většinou aplikovány různé FUP limity a omezení.

Při návrhu aplikace byl kladen důraz na její budoucí rozšiřitelnost. Prvním vylepšením by mohlo být přidání několika testových otázek ke každé památce. Za jejich správné zodpovězení by uživatelé získávali bonusové body, které by jim mohli přinést lepší umístění v celkovém žebříčku. Pro účely diplomové práce byly zpracovány významné památky ve městě Hradec Králové. Pokud by však měla být aplikace v budoucnu reálně nasazena v obchodu Google Play, bylo by potřeba získat přístup k vhodné databázi památek pro celou Českou republiku.

Výsledkem této práce je plně funkční geolokační aplikace. Ve srovnání s jinými tématicky podobnými aplikacemi je k určení polohy kromě GPS využíváno navíc i otisků bezdrátových sítí. Tím je uživateli zamezena možnost zmanipulovat svoje souřadnice na mapě a tím podvádět ve hře. Mobilní aplikace byla důkladně otestována na různých verzích operačního systému Android a veškeré nalezené chyby se podařilo úspěšně odstranit.

9 Seznam použité literatury

- [1] RAPANT, Petr. Družicové polohové systémy [online]. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita, 2002 [cit. 2016-02-24]. ISBN 80-248-0124-8. Dostupné z: http://gis.vsb.cz/vojtek/content/gnps/files/_source/RAP02.pdf
- [2] VONDRÁK, Jan. Historie navigace – od kvadrantu k GNSS. Pokroky matematiky, fyziky a astronomie [online]. 2013, 58(1), 11-20 [cit. 2016-02-24]. Dostupné z: http://dml.cz/bitstream/handle/10338.dmlcz/143254/PokrokyMFA_58-2013-1_2.pdf
- [3] ČÁBELKA, Miroslav. Úvod do GPS [online]. [cit. 2016-02-24]. Dostupné z: https://www.natur.cuni.cz/geografie/geoinformatika-kartografie/ke-stazeni/vyuka/gps/skriptum-uvod-do-gps?student_welcome=1. Skripta. Přírodovědecká fakulta UK v Praze.
- [4] History of Portugal. HistoryWorld [online]. [cit. 2016-02-24]. Dostupné z: <http://www.historyworld.net/wrldhis/PlainTextHistories.asp?ParagraphID=has>
- [5] GSM World Coverage Map. WorldTimeZone [online]. 2016 [cit. 2016-03-01]. Dostupné z: <http://www.worldtimezone.com/gsm.html>
- [6] FRANĚK, Ondřej a Miloš SMEJKAL. Lokalizace volání z mobilních telefonů u příchozích tísňových volání v podmínkách hl.m. Prahy. Urgentní medicína [online]. 2011(4) [cit. 2016-03-01]. Dostupné z: http://www.zachrannasluzba.cz/zajimavosti/2012_lokalizace_mt.pdf
- [7] KOPP, Michal. Metody určení polohy mobilního zařízení v síti GSM [online]. [cit. 2016-03-01]. Dostupné z: http://radio.feld.cvut.cz/personal/mikulak/MK/MK06_semestralky/UrceniPolohyGSM_KoppM.pdf. Semestrální práce. ČVUT - Fakulta elektrotechnická.
- [8] Princip buňkového systému. Technologie pro mobilní komunikaci [online]. [cit. 2016-03-01]. Dostupné z: <http://tomas.richtr.cz/mobil/bunk-princip.htm>
- [9] SINGH, Balaram, Soumya PALLAI a Susil Kumar RATH. A Survey of Cellular Positioning Techniques in GSM Networks. NATIONAL CONFERENCE IN MOBILE COMPUTING-NCMC'12 [online]. [cit. 2016-03-01]. Dostupné z: https://www.researchgate.net/publication/256736858_A_Survey_of_Cellular_Positioning_Techniques_in_GSM_Networks
- [10] Mobile Positioning Techniques in GSM Cellular Networks: A Comparative Performance Analysis. International Journal of Computer Technology and Electronics Engineering (IJCTEE) [online]. 2012, 6(2), 21-29 [cit. 2016-03-01]. Dostupné z: http://www.ijctee.org/files/VOLUME2ISSUE6/IJCTEE_1212_05.pdf

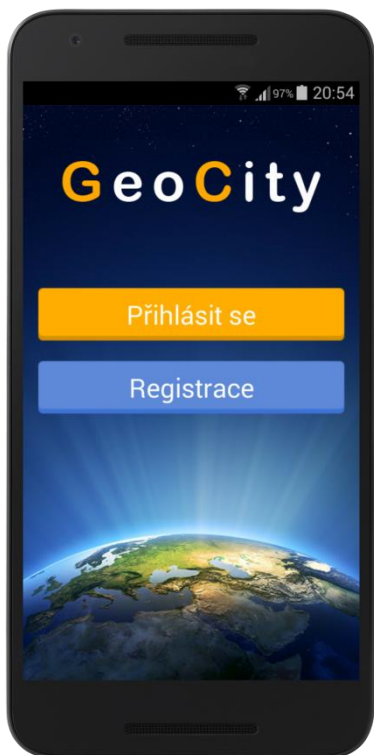
- [11] Jak určit polohu mobilního telefonu. MobilMania.cz [online]. 2004 [cit. 2016-03-01]. Dostupné z: <http://www.mobilmania.cz/jak-urcit-polohu-mobilniho-telefonu/a-1107567/>
- [12] SZTEFEK, Lukáš. GSM lokalizace [online]. 2013 [cit. 2016-03-01]. Dostupné z: http://www.lukassztefek.cz/files/fit/projekt_sen.pdf. Textová rešerše.
- [13] JEČNÝ, Tomáš. Lokalizace v sítích WLAN [online]. [cit. 2016-03-01]. Dostupné z: http://radio.feld.cvut.cz/personal/mikulak/MK/MK05_semestralky/Lokalizace_v_s%EDt%EDch_WLANlong.pdf. Semestrální práce. ČVUT - Fakulta elektrotechnická.
- [14] Nový standard Wi-Fi: Gigabit vzduchem. Živě.cz [online]. 2012 [cit. 2016-03-11]. Dostupné z: <http://www.zive.cz/clanky/novy-standard-wi-fi-gigabit-vzduchem/sc-3-a-165687/>
- [15] Wifileaks [online]. [cit. 2016-03-13]. Dostupné z: <http://www.wifileaks.cz/>
- [16] HENNIGES, Robin. Current approaches of Wifi Positioning [online]. cit. 2016-03-15]. Dostupné z: https://www.snet.tu-berlin.de/fileadmin/fg220/courses/WS1112/snet-project/wifi-positioning_henniges.pdf
- [17] On outdoor positioning with Wi-Fi. Journal of Global Positioning Systems [online]. 2008, 7(1), 18-26 [cit. 2016-03-15]. Dostupné z: http://www.gnss.com.au/JoGPS/v7n1/JoGPS_v7n1p18-26.pdf
- [18] Jak je možné, že mobil ví, kde zrovna jsme? Živě.cz [online]. 2014 [cit. 2016-03-15]. Dostupné z: <http://www.zive.cz/clanky/jak-je-mozne-ze-mobil-vi-kde-zrovna-jsme/lokalizace-pomoci-gps/sc-3-a-174857-ch-93694/default.aspx#articleStart>
- [19] GPS.gov: Official U.S. Government information about the Global Positioning System (GPS) and related topics [online]. [cit. 2016-03-19]. Dostupné z: <http://www.gps.gov/>
- [20] Global Positioning System. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation [cit. 2016-03-19]. Dostupné z: https://cs.wikipedia.org/wiki/Global_Positioning_System
- [21] EL-RABBANY, Ahmed. Introduction to GPS: the Global Positioning System. Boston, MA: Artech House, c2002. ISBN 15-805-3183-0.
- [22] Analýza četnosti výskytu družic GPS a GLONASS [online]. [cit. 2016-03-18]. Dostupné z: <http://gis.templ.net/gps/Prague-20081704/>
- [23] Mobilní operační systém Android. Diit.cz: Deep in IT [online]. [cit. 2016-05-28]. Dostupné z: <http://diit.cz/clanek/mobilni-operacni-system-android>

- [24] Android (operační systém). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-28]. Dostupné z: [https://cs.wikipedia.org/wiki/Android_\(opera%C4%8Dn%C3%AD_syst%C3%A9m\)](https://cs.wikipedia.org/wiki/Android_(opera%C4%8Dn%C3%AD_syst%C3%A9m))
- [25] Android Architecture. TutorialsPoint [online]. [cit. 2016-05-29]. Dostupné z: http://www.tutorialspoint.com/android/android_architecture.htm
- [26] Android history. Android [online]. [cit. 2016-05-30]. Dostupné z: <https://www.android.com/history/>
- [27] Historie Androidu v kostce aneb Od verze 1.0 až po Android M. Svět Androida [online]. [cit. 2016-05-30]. Dostupné z: <https://www.svetandroida.cz/historie-androidu-201506>
- [28] Number of Google Play Store apps 2016. Statistic [online]. [cit. 2016-06-11]. Dostupné z: <http://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>
- [29] Ingress [online]. [cit. 2016-06-11]. Dostupné z: <https://www.ingress.com/>
- [30] GEOFUN [online]. [cit. 2016-06-11]. Dostupné z: <http://www.geofun.cz/>
- [31] ApiDoc: Inline Documentation for RESTful web APIs [online]. [cit. 2016-06-20]. Dostupné z: <http://apidocjs.com>
- [32] Create REST applications with the Slim micro-framework: Use the Slim PHP micro-framework to efficiently prototype and deploy a REST API. IBM developerWorks [online]. 2012 [cit. 2016-06-20]. Dostupné z: <https://www.ibm.com/developerworks/library/x-slim-rest/>
- [33] Slim Framework [online]. [cit. 2016-06-22]. Dostupné z: <http://www.slimframework.com>
- [34] ALLEN, Grant. Android 4: průvodce programováním mobilních aplikací. Brno: Computer Press, 2013. ISBN 978-80-251-3782-6.
- [35] Android Developers [online]. [cit. 2016-07-03]. Dostupné z: <https://developer.android.com>
- [36] Square Open Source [online]. [cit. 2016-07-03]. Dostupné z: <http://square.github.io/>
- [37] Material-ish Progress [online]. [cit. 2016-07-03]. Dostupné z: <https://github.com/pnikosis/materialish-progress>
- [38] Android-RoundCornerProgressBar [online]. [cit. 2016-07-03]. Dostupné z: <https://github.com/akexorcist/Android-RoundCornerProgressBar>

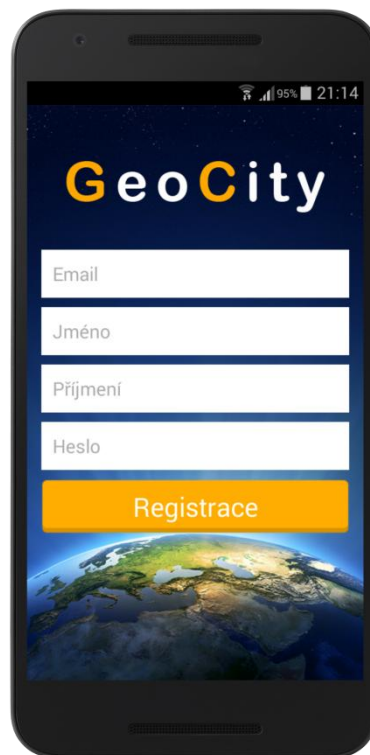
- [39] Making ListView Scrolling Smooth. Android Developers [online]. [cit. 2016-07-05]. Dostupné z:
<https://developer.android.com/training/improving-layouts/smooth-scrolling.html>
- [40] Fake Location Spoofer Free. Google Play [online]. [cit. 2016-07-06]. Dostupné z:
<https://play.google.com/store/apps/details?id=com.incorporateapps.fakegps.fre>
- [41] Nejvýznamnější památky v Hradci Králové. Hradec Králové: oficiální stránky statutárního města [online]. [cit. 2016-07-15]. Dostupné z:
<http://www.hrdeckralove.org/urad/nejvyznamnejsi-pamatky-v-hradci-kralove>

10 Přílohy

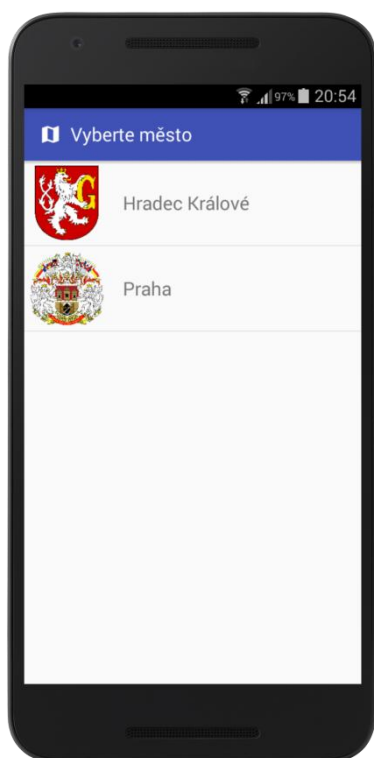
- 1) Náhledy obrazovek z naprogramované mobilní aplikace
- 2) Datový nosič s vytvořenou aplikací



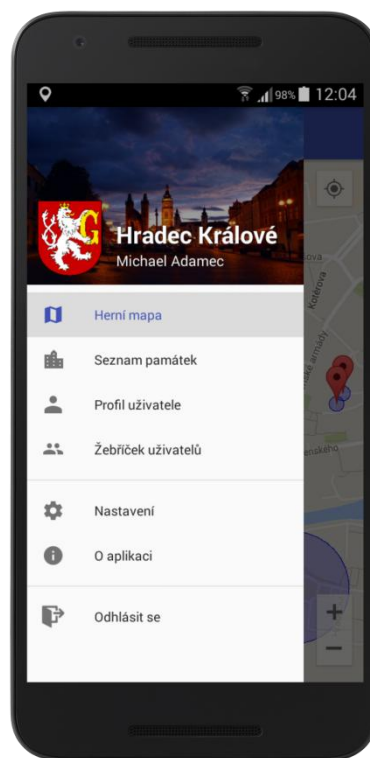
Obr. 1: Úvodní obrazovka



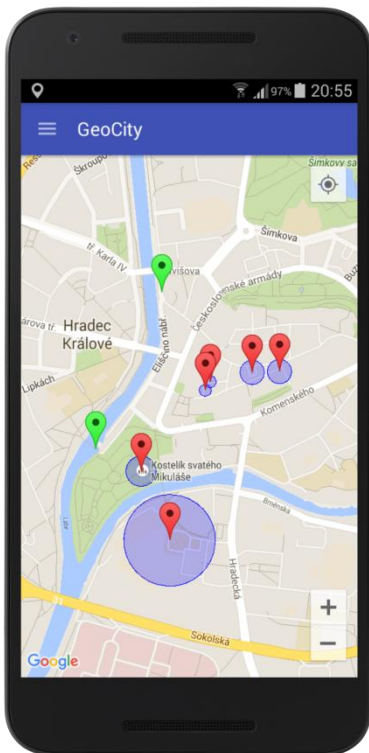
Obr. 2: Registrace do aplikace



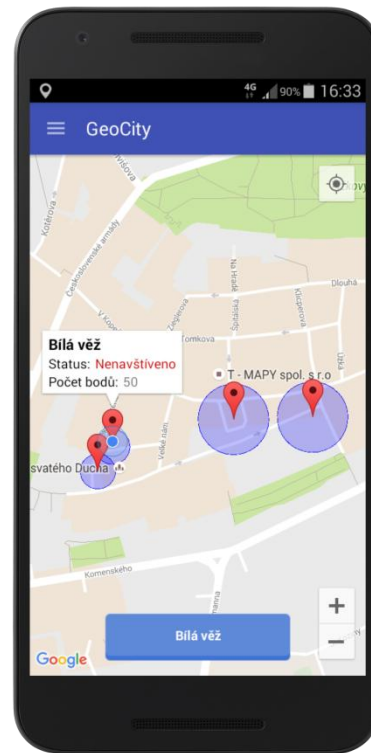
Obr. 3: Výběr města



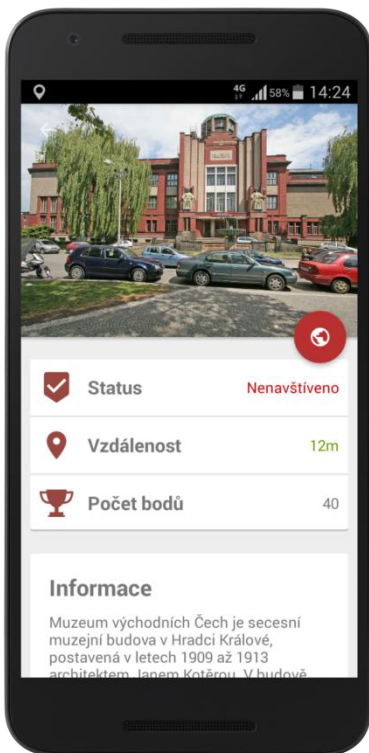
Obr. 4: Postranní nabídka



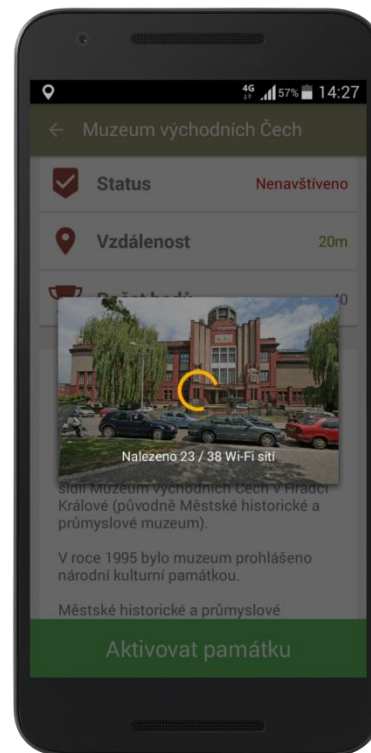
Obr. 5: Herní mapa



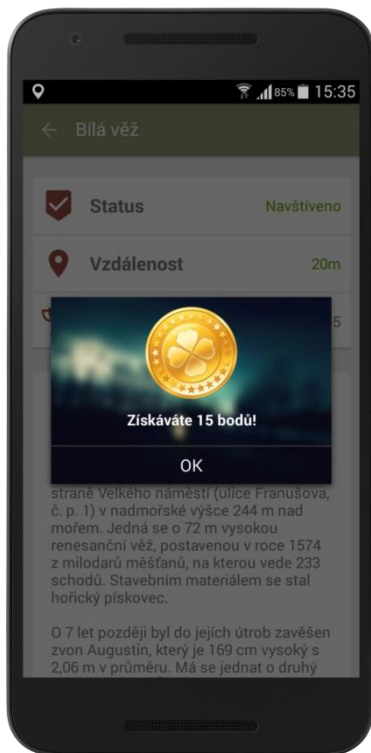
Obr. 6: Herní mapa - nenaštvívěná památka



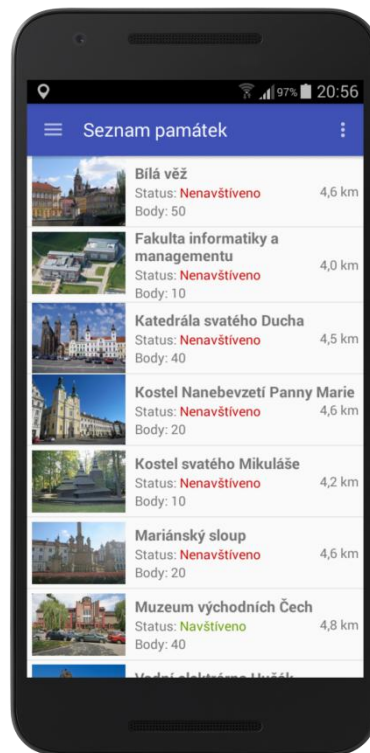
Obr. 7: Detail památky



Obr. 8: Proces aktivace památky



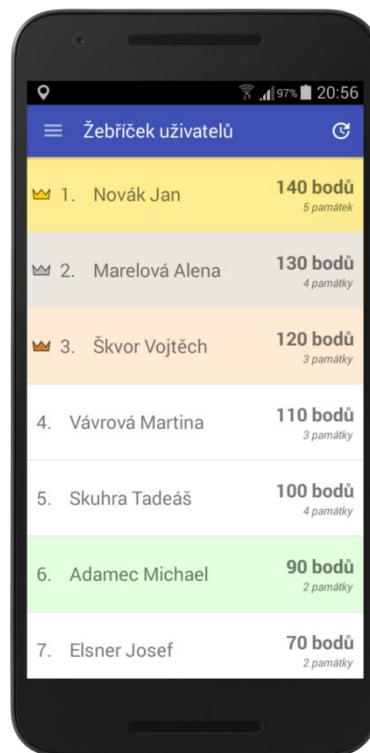
Obr. 9: Úspěšná aktivace památky



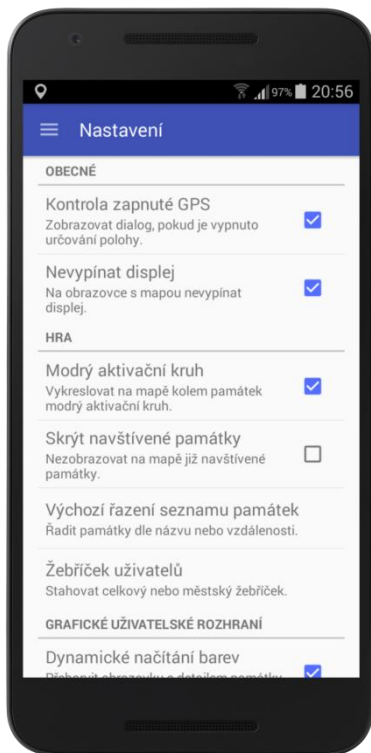
Obr. 10: Seznam památek



Obr. 11: Profil uživatele



Obr. 12: Žebříček uživatelů



Obr. 13: Nastavení aplikace



Obr. 14: O aplikaci

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Adamec Michael	Dvorská 494, Hradec Králové - Svobodné Dvory	11456

TÉMA ČESKY:

Geolokační aplikace pro Android s využitím gamifikace

TÉMA ANGLICKY:

Geolocation application for Android using gamification

VEDOUCÍ PRÁCE:

doc. Ing. Filip Malý, Ph.D. - KIKM

ZÁSADY PRO VYPRACOVÁNÍ:

Cíl práce:

Cílem této práce je seznámit se s různými možnostmi určování polohy mobilního zařízení a dále navrhnout a implementovat aplikaci pro operační systém Android, která těchto principů bude využívat. Výstupem práce bude plnohodnotná a funkční aplikace.

Osnova práce:

1. Úvod
2. Možnosti určování geografické polohy
3. OS Android
4. Požadavky na aplikaci
5. Návrh a implementace
6. Výsledky
7. Závěr
8. Seznam použité literatury

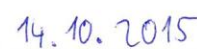
SEZNAM DOPORUČENÉ LITERATURY:

1. GENTILE, Camillo. Geolocation techniques: principles and applications. New York: Springer, c2013, xii, 287 p. ISBN 978-146-1418-351.
2. EL-RABBANY, Ahmed. Introduction to GPS: the Global Positioning System. 2nd ed. Boston, MA: Artech House, c2006, xiv, 210 p. ISBN 978-59-6930-162.
3. FRIESEN, Jeff. Learn Java for Android development. Third edition. xxx, 1167 pages.
4. ANNUZZI, Joseph, Lauren DARCEY a Shane CONDER. Advanced android application development. Fourth edition. xxxi, 570 pages. ISBN 978-013-3892-383.

Podpis studenta:


.....

Datum:


.....

Podpis vedoucího práce:


.....

Datum:


.....