

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Mobilní aplikace pro cyklistické výlety

Plzeň, 2016

Petr Šrámek

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 23. června 2016

Petr Šrámek

Abstract

The main goal of this work is to create a system supporting cycling tours for individuals and groups with client part on Android platform. The first part of the document describes ways to locate users of mobile devices and the possibilities to work with maps and map sources on Android. These information are then used for the implementation of a system, which contains a client-side application on Android platform and server part servicing the mutual communication of users in group mode. The application allows to record the user trace, show the trace in real time, add points of interest and save the final trace. Also contains monitoring function, which can send SMS in case of emergency. Group mode of the application allows the user to show the position of all users of the same group on a map and send/receive messages to all members of a group.

Abstrakt

Cílem této práce je vytvoření systému pro podporu cyklistických výletů jednotlivců i skupin, s klientskou částí na platformě Android. První část dokumentu se zabývá popisem různých způsobů lokalizace uživatele mobilního zařízení a možnostmi práce s mapou a mapovými podklady na platformě Android. Následně jsou tyto znalosti využity k návrhu a realizaci samotného systému, který se sestává z aplikace na platformě Android a serverové části pro obsluhu vzájemné komunikace uživatelů v rámci skupinového módu. Aplikace umožňuje záznam trasy uživatele, její zobrazení na zařízení v reálném čase, možnost přidání bodů zájmu a uložení celé trasy. Také obsahuje hlídací funkci, která v případě potřeby automaticky odešle nouzovou SMS. Skupinový mód umožňuje zobrazení aktuálních pozic všech členů skupiny na mapě a posílání/přijímání zpráv v rámci skupiny.

Poděkování

Chtěl bych poděkovat vedoucímu práce Ing. Ladislavu Pešíčkovi za cenné rady a ochotu v celém průběhu vytváření této práce.

Také bych rád poděkoval své rodině za veškerou poskytnutou podporu při studiu.

Obsah

| | | |
|-------|--|----|
| 1 | Úvod..... | 1 |
| 2 | Lokalizace uživatele | 2 |
| 2.1 | Globální satelitní navigační systém..... | 2 |
| 2.1.1 | Global Positioning System (GPS)..... | 3 |
| 2.1.2 | Asistovaná GPS (A-GPS) | 3 |
| 2.1.3 | GLONASS | 3 |
| 2.1.4 | Další globální satelitní navigační systémy..... | 4 |
| 2.1.5 | Výhody a nevýhody lokalizace pomocí GPS přijímačů..... | 4 |
| 2.2 | Lokalizace pomocí sítě | 4 |
| 2.2.1 | Lokalizace dle identifikačního čísla vysílače (CGI)..... | 5 |
| 2.2.2 | CGI-TA | 6 |
| 2.2.3 | Lokalizace pomocí více BTS..... | 6 |
| 2.2.4 | WiFi | 7 |
| 2.2.5 | Výhody a nevýhody lokalizace pomocí sítě | 7 |
| 2.3 | Hlavní cíle lokalizace..... | 7 |
| 2.4 | Shrnutí..... | 9 |
| 3 | Práce s mapou na platformě Android | 10 |
| 3.1 | Google Maps Android API | 10 |
| 3.1.1 | Vložení Google map do aplikace..... | 10 |
| 3.1.2 | Počáteční nastavení mapy | 11 |
| 3.1.3 | Nevýhody Google Maps Android API..... | 12 |
| 3.2 | OpenStreetMap..... | 12 |
| 3.2.1 | Práce s Osmdroid API..... | 13 |
| 3.2.2 | Nevýhody Osmdroid API | 13 |
| 3.3 | Shrnutí..... | 14 |
| 4 | Aplikace Cyclo | 15 |
| 4.1 | Minimální požadavky aplikace | 15 |
| 4.2 | Povolení, která aplikace vyžaduje | 16 |
| 4.2.1 | Povolení nutná pro lokalizaci uživatele | 16 |
| 4.2.2 | Povolení potřebná pro hlídací funkci..... | 16 |

| | | |
|-------|---|----|
| 4.2.3 | Povolení potřebná pro Google Maps..... | 16 |
| 4.3 | Celková struktura aplikace | 17 |
| 4.4 | Úvodní obrazovka aplikace | 18 |
| 4.4.1 | Nápověda | 18 |
| 4.4.2 | Nastavení | 18 |
| 4.5 | Uchovávání uložených tras | 19 |
| 4.5.1 | Návrh databázové struktury | 19 |
| 4.5.2 | Práce s databází SQLite | 20 |
| 4.6 | Záznam trasy uživatele | 21 |
| 4.6.1 | Získání aktuální pozice uživatele | 21 |
| 4.6.2 | Vykreslení trasy | 22 |
| 4.6.3 | Návrat do aktivity se záznamem trasy po přerušení | 23 |
| 4.6.4 | Kontrola, zda je uživatel v pořádku | 24 |
| 4.6.5 | Informace o trase | 24 |
| 4.7 | Mód jednotlivce | 26 |
| 4.7.1 | Výpis tras uložených uživatelem | 26 |
| 4.7.2 | Zobrazení uložené trasy | 27 |
| 4.8 | Skupinový mód | 27 |
| 4.8.1 | Návrh databázové struktury skupinového módu | 28 |
| 4.8.2 | Komunikace se serverem | 29 |
| 4.8.3 | Obsluha požadavků na serveru | 30 |
| 4.8.4 | Založení nové skupiny | 31 |
| 4.8.5 | Ukončení skupiny | 31 |
| 4.8.6 | Připojení se k existující skupině | 32 |
| 4.8.7 | Aktualizace skupinových informací | 32 |
| 4.8.7 | Skupinové zprávy | 33 |
| 5 | Testování | 35 |
| 5.1 | Zobrazení a ovládání aplikace | 35 |
| 5.2 | Běh aplikace | 35 |
| 5.3 | Správnost a přesnost dat | 36 |
| 5.3.1 | Přesnost lokalizace | 36 |
| 5.3.2 | Záznam trasy | 36 |

| | | |
|-------|-----------------------------------|----|
| 5.3.3 | Ukládání informací o trase..... | 36 |
| 5.3.4 | Skupinové funkce..... | 38 |
| 5.3.5 | Hlídací funkce..... | 38 |
| 5.3.6 | Další testované scénáře | 38 |
| 6 | Možnosti rozšíření aplikace | 40 |
| 7 | Závěr | 41 |

1 Úvod

Cílem této práce je vytvoření systému pro podporu cyklistických výletů jednotlivců i skupin, s klientskou částí na platformě Android a serverovou částí pro obsluhu skupinových funkcí. V první části práce se nachází teoretický rozbor, díky kterému čtenář získá lepší představu, jaké možnosti lze využít pro lokalizaci uživatele mobilního zařízení a jaké lze použít mapy a mapové podklady v rámci vytváření aplikací pro zařízení s operačním systémem Android. Následně jsou tyto znalosti využity k návrhu a realizaci systému pro podporu cyklistických výletů.

Úvodní část práce je věnována teoretickému rozboru a prozkoumání jednotlivých stěžejních částí navrhované aplikace. Nejprve se práce věnuje rozboru způsobů, jak lze uživatele mobilního zařízení lokalizovat, popisu principů, jak samotná lokalizace probíhá a jaké jsou výhody a nevýhody jednotlivých variant. Druhým důležitým prvkem je práce s mapou a zobrazování informací v dané mapě. V další kapitole jsou tedy popsána vybraná rozhraní pro práci s mapou na platformě Android, dostupné typy map, jaké zdroje map lze pro navrhovanou aplikaci využít a jaké možnosti jednotlivá rozhraní nabízí.

S využitím znalostí, získaných v úvodní části práce, je snazší pochopení principů, jak jednotlivé technologie fungují, které z nich bude nejlepší pro navrhovanou aplikaci použít, jak je lze implementovat na platformě Android a jaké možnosti a funkce poskytnou navrhované aplikaci. Tyto znalosti jsou následně aplikovány při návrhu a realizaci systému pro cyklistické výlety na platformě Android, který podporuje výlety jednotlivců i skupin.

Aplikace bude podporovat kromě záznamu a vizualizace aktuální trasy uživatele také skupinové funkce, jako například vizualizaci jednotlivých členů skupiny na mapě na základě informací o jejich aktuální poloze a zasílání hromadných zpráv všem uživatelům v rámci stejné skupiny. Testování bude probíhat v průběhu celého vývoje systému na netriviálních datech a na závěr bude součástí i testování systému v reálném provozu při cyklistických výletech jednotlivců i skupin o minimálně třech účastnících.

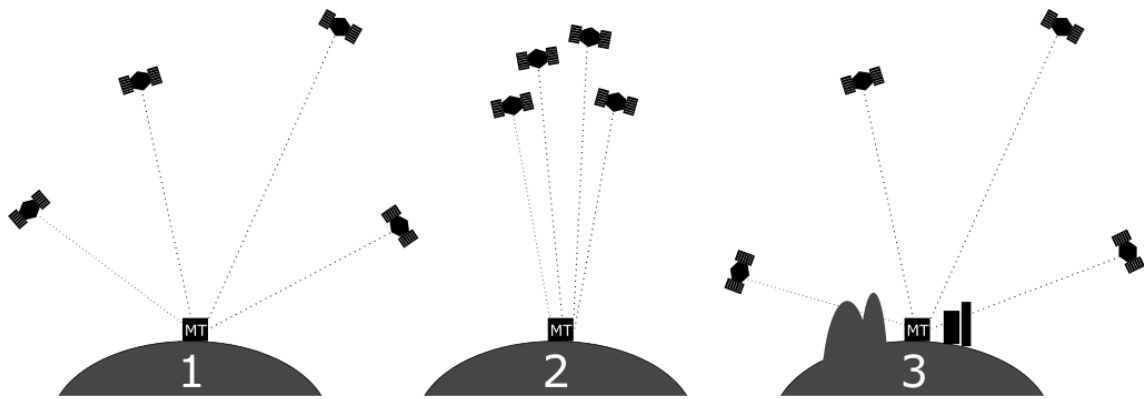
2 Lokalizace uživatele

Znalost polohy uživatele umožňuje aplikacím, aby poskytovaly informace a výsledky optimalizované přímo pro lokalitu, kde se daný uživatel právě nachází, případně tuto informaci předaly k dalšímu zpracování. Existují dvě základní možnosti, jak lze získat polohu uživatele: využití GPS přijímače nebo lokalizace na základě dat z mobilní sítě, případně v kombinaci s WiFi. Vlastnosti těchto možností spolu s jejich výhodami a nevýhodami budou popsány v následujících kapitolách.

2.1 Globální satelitní navigační systém

Globální satelitní navigační systém umožňuje lokalizaci, včetně určení nadmořské výšky a přesného času kdekoli na Zemi s přesností v řádu metrů [1], pokud má zařízení s čipem pro příjem GPS přímý výhled na oblohu a nachází se nad ním (na horizontu) čtyři nebo více satelitů GPS [2]. Pokud by se jednalo pouze o zjištění horizontální polohy (zeměpisné šířky a délky), stačily by pro její určení s menší přesností i jen tři satelity [3]. Informace ze čtvrtého satelitu pomáhá lépe určit časovou odchylku mezi časem zařízení a satelitním časem a je nutná k výpočtu nadmořské výšky, díky čemuž poskytuje minimální počet 4 satelitů plnohodnotnou prostorovou lokalizaci (zeměpisnou šířku a délku, nadmořskou výšku a přesný čas) a větší přesnost. Samozřejmě čím více satelitů je dostupných, tím přesnější lokalizace je.

Čip pro příjem GPS signálu je pasivní zařízení, které pouze přijímá radiové signály vysílané satelity z daného satelitního navigačního systému. Vysílaný signál obsahuje informace, jaký satelit jej vyslal a přesný čas jejich vyslání. Po přijetí signálu si zařízení s přijímačem GPS vypočte rozdíl mezi časem vyslání a časem přijetí signálu. Díky tomu, že je u každého systému předem známa přesná poloha, kde daný satelit v konkrétní čas byl, si zařízení může vypočítat svou vzdálenost od daného satelitu. A zná-li zařízení svou vzdálenost od nejméně 4 satelitů, může si na základě těchto dat vypočítat svou aktuální polohu, tedy zeměpisnou šířku a délku, včetně určení nadmořské výšky a přesného času. Na obrázku 2.1 je znázorněno, že pro přesnou lokalizaci uživatele nestačí jen mít k dispozici dostatečný počet satelitů, ale je také velmi důležité jejich rozmístění.



Obrázek 2.1 - Ukázka různých možností rozmístění satelitů

1: ideální rozmístění satelitů

2: zhoršené podmínky, pokud jsou satelity rozmístěné blízko u sebe, může to zapříčinit menší přesnost lokalizace

3: velmi špatná situace, kdy je sice rozmístění satelitů dobré, ale dochází k rušení výhledu okolím, kvůli kterému je lokalizace nemožná, nebo nepřesná

2.1.1 Global Positioning System (GPS)

Správu sítě satelitů globálního polohovacího systému (GPS) [11] zajišťují Spojené státy americké, které se zavázaly, že bude 95% času dostupných minimálně 24 satelitů, vysílajících radiové signály sloužící k určení přesné polohy. Tyto satelity se pohybují v přibližné výšce 20 200 kilometrů nad povrchem Země a jsou rovnoměrně rozdělené do 6 oběžných drah kolem Země, kde na každé dráze jsou umístěné minimálně čtyři satelity. Každý ze satelitů stihne za jeden den oběhnout Zemi dvakrát. V červnu roku 2011 byla síť GPS satelitů rozšířena z původních 24 na 27, díky čemuž se ještě více zlepšilo pokrytí a dostupnost GPS.

2.1.2 Asistovaná GPS (A-GPS)

Jedním z problémů GPS je pomalý start, kdy uživatel musí po spuštění čekat, než dostane první informaci o své aktuální poloze. To je zapříčiněné tím, že přijímač pro určení polohy musí nejdříve zjistit, jaké satelity se nad ním nacházejí a jaké je jejich rozmístění. U zařízení s připojením k internetu lze ale díky A-GPS [4] start urychlit tím, že si zařízení stáhne ze vzdáleného serveru informace o aktuální poloze satelitů GPS a přijímač díky tomu ihned ví, které satelity by pro něj měly být teoreticky viditelné a jaké je jejich rozmístění. Přijímač tedy v případě A-GPS využívá pro zrychlení počátečního startu asistenci vzdáleného serveru.

2.1.3 GLONASS

Kromě GPS provozovaného USA existuje také globální satelitní navigační systém GLONASS [5], který je provozován Ruskem a jeho dostupnost i přesnost je srovnatelná s GPS. GLONASS využívá 24 satelitů, které na rozdíl od GPS obíhají jen na 3 oběžných drahách kolem Země a na každé této dráze je rovnoměrně rozmístěných 8 satelitů.

Výrobci GPS přijímačů mohou kombinovat signály ze satelitů obou těchto systémů a mohou tudíž mít k dispozici přibližně až 24 satelitů najednou (celkový počet satelitů by byl 27 + 24, ale polovina jich je nad druhou polokoulí a tím pádem mimo viditelnost).

Díky vyššímu počtu satelitů, které je možné využít, lze dosáhnout pokrytí i na místech, kde by tyto satelitní navigační systémy neměly dostatečný počet svých vlastních satelitů, a vyšší přesnosti lokalizace i při částečně omezeném výhledu na oblohu.

2.1.4 Další globální satelitní navigační systémy

Jelikož by se ve výjimečných případech mohlo stát, že USA nebo Rusko omezí dostupnost svých satelitních navigačních systémů, plánují i další země spuštění vlastních systémů, aby byly nezávislé na ostatních státech. Evropská unie má v plánu spustit svůj vlastní globální satelitní navigační systém nazvaný Galileo [6], který by se měl sestávat z 27 satelitů, jež budou obíhat kolem Země rovnoměrně rozdělené na třech drahách. Dle původních plánů měl být systém Galileo provozuschopný již roku 2010, ale podle nových plánů byl rok spuštění posunutý nejdříve na rok 2018.

Čína plánuje vybudování vlastní globální sítě satelitů nazvanou COMPASS, Indie má vlastní regionální navigační systém GAGAN a další země také plánují vybudování svých regionálních navigačních systémů.

2.1.5 Výhody a nevýhody lokalizace pomocí GPS přijímačů

Zjišťování polohy uživatele pomocí GPS přijímačů je v drtivé většině případů velmi přesné, s odchylkou maximálně 10 metrů a funguje téměř kdekoliv na Zemi. Má ale některé nevýhody a je limitované řadou omezení. Správná funkčnost GPS závisí na přímém výhledu na oblohu, uvnitř budov nebo zastavěných oblastí nemusí být funkční vůbec, nebo s malou přesností. Další důležitou věcí je kvalita samotného GPS přijímače, použitého v konkrétním mobilním zařízení. U levnějších přístrojů se můžeme setkat s méně kvalitními GPS čipy, kterým nalezení přesné polohy trvá delší dobu. Velkou nevýhodou představuje také spotřeba baterie, vzhledem k tomu, že vyhledávání vhodných satelitů spolu s výpočtem polohy uživatele jsou velmi náročné na výkon. A v neposlední řadě je nevýhodou i rychlost určení polohy, které může trvat především při prvotním spuštění delší dobu.

2.2 Lokalizace pomocí sítě

Další možností jak lokalizovat mobilní zařízení, je využití informací z GSM sítě nebo pomocí okolních vysílačů WiFi. Pro lokalizaci pomocí GSM sítě [7] jsou potřeba dva základní prvky - základnové stanice (BTS) a mobilní telefon (MT). Základnová stanice je osazena jedním nebo více vysílači, které se starají mimo jiné o vysílání a příjem rádiových signálů. V dnešní době je běžné osazení BTS více vysílači s použitím směrových antén, díky čemuž se oblast kolem BTS rozdělí do několika sektorů. Například při osazení třemi směrovými anténami se oblast rozdělí na 3 kuželové sektory s rozpětím 120°, kde má každý sektor své identifikační číslo (CGI). Dle umístění BTS je také různý dosah jejich vysílačů. V hustěji zabydlených oblastech je potřeba více

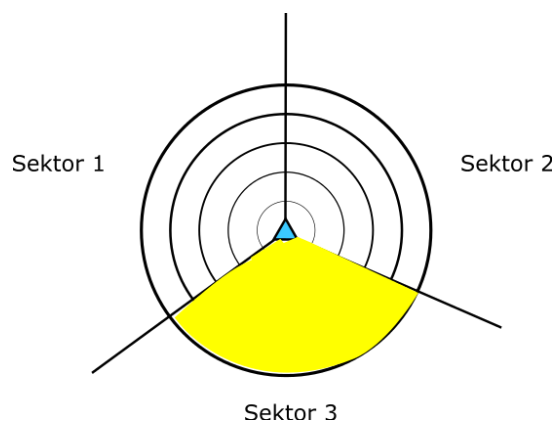
vysílačů, takže se jejich dosah pohybuje jen v okruhu několika kilometrů, kdežto v řídké osídlených oblastech může být dosah jedné BTS až cca 35 kilometrů.

Aby BTS mohla obsloužit více MT najednou, je každá vysílací frekvence rozdělena na časové úseky, takzvané time sloty. Díky rozdělení na 8 time slotů může BTS v jedné periodě obsloužit až 8 zařízení. Aby nedocházelo k vzájemnému rušení signálů jednotlivých zařízení, je každému zařízení přiřazený konkrétní time slot, který představuje osminu času z každé periody. MT má tedy přesně určený časový úsek, kdy je přenosový kanál jen jeho a může vysílat. Zbývajících čas během periody zařízení mlčí, aby nerušilo vysílání v rámci ostatních time slotů. Jelikož by při několikakilometrové vzdálenosti MT od základnové stanice mohl mít vysílaný signál od zařízení časové zpoždění, kvůli kterému by nedorazil k BTS ve svém přiděleném časovém úseku, je nutné si před začátkem komunikace vypočítat vzájemnou vzdálenost (TA), která se určí s přesností 550m. MT na základě této vzdálenosti vyšle signál případně o chvíli dřív, aby po přičtení časového zpoždění dorazil k BTS ve správný čas, tedy v jemu přiděleném časovém úseku.

2.2.1 Lokalizace dle identifikačního čísla vysílače (CGI)

Nejméně přesným způsobem lokalizace je ten, pokud máme data o poloze zařízení jen z jedné BTS, která je umístěná mimo obydlenu oblast (a tím pádem má velký dosah v okruhu až 35km) a navíc má jen jeden vysílač. V tomto případě by byla přesnost lokalizace okruh s poloměrem 35km, což je velmi nepřesné. Dosah BTS může být samozřejmě mnohem menší, poloměr okruhu záleží na hustotě rozmístění BTS v dané oblasti.

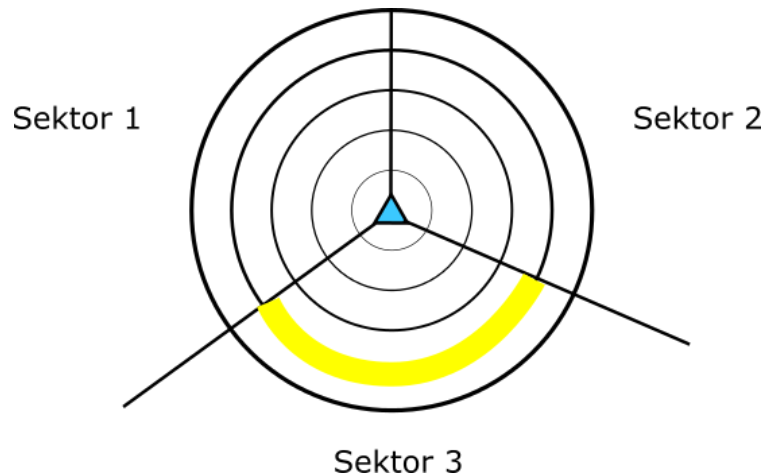
Pokud by daná BTS měla více vysílačů (směrových antén), došlo by k výraznému zlepšení, jelikož by se daný okruh rozdělil na několik kuželových sektorů, dle počtu směrových antén. MT by byl tedy lokalizovaný v konkrétním sektoru, což je například u tří směrových antén zpřesnění na jednu třetinu původního okruhu, viz obrázek 2.2, na kterém je vyznačen výsledný sektor lokalizace.



Obrázek 2.2 - BTS uprostřed, se třemi směrovými anténami. Antény rozdělují prostor kolem BTS na 3 sektory. Uživatel by byl lokalizován tedy v jednom z těchto sektorů, viz zvýrazněný sektor.

2.2.2 CGI-TA

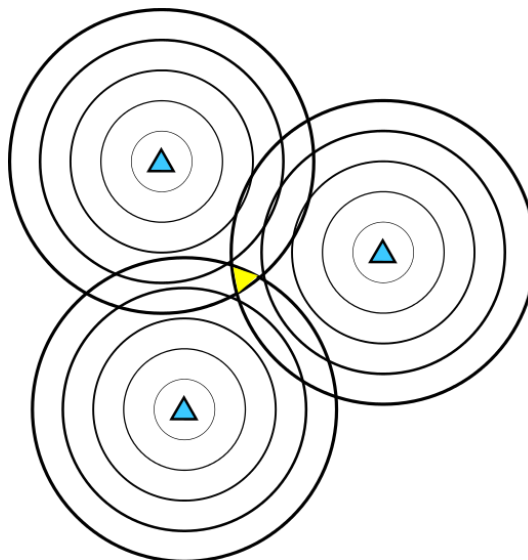
Možnost zlepšení přesnosti lokalizace představuje kombinace výše uvedené metody s údajem o vzdálenosti zařízení od BTS (TA). Díky TA se zúží oblast lokalizace pouze na výřez uvnitř daného sektoru, jako je vyznačeno v obrázku 2.3. Daná oblast bude mít šířku v závislosti na přesnosti TA, která se pohybuje kolem 550m.



Obrázek 2.3 - Oblast lokalizace při zpřesnění pomocí TA

2.2.3 Lokalizace pomocí více BTS

Pokud má MT ve svém dosahu více BTS, lze polohu určit na základě protnutí sektorů lokalizovaných jednotlivými BTS. Čím více BTS bude mít MT v dosahu, tím přesnější lokalizace docílíme. Oproti předchozím obrázkům je u obrázku 2.4 na první pohled zřetelné zmenšení vyznačené oblasti lokalizace jen na malou oblast průniku tří sousedních BTS.



Obrázek 2.4 - Lokalizace pomocí 3 sousedních BTS

2.2.4 WiFi

Lokalizovat uživatele je možné, i pokud má zapnutý jen WiFi přijímač [10]. Každý vysílač (přístupový bod) WiFi poskytuje zařízením ve svém dosahu základní údaje o sobě, jako například název sítě (SSID) nebo svou fyzickou adresu (BSSID). Existují databáze WiFi vysílačů s uloženou polohou, kde se vysílač s daným BSSID nachází.

Například společnost Google má vlastní databázi WiFi vysílačů, kterou neustále rozšiřuje a aktualizuje díky uživatelům používajícím zařízení se systémem Android. Pokud totiž uživatel povolí společnosti Google anonymně zaznamenávat informace o své poloze, tak dochází v určitých intervalech ke zjištění aktuální polohy s využitím GPS, BTS a WiFi a ukládá si informace o poloze WiFi vysílačů (pokud nějaké v okolí zaznamená) pod jejich BSSID do databáze.

Takže pokud mobilní zařízení v okolí zaznamená nějaký WiFi vysílač, který je již v databázi, lze na základě informací o poloze vysílače z databáze získat přibližnou polohu uživatele. Tyto informace lze kombinovat s informacemi z BTS a využít je ke zpřesnění lokalizace, nebo pokud je zaznamenáno více WiFi vysílačů, lze stejně jako v případě více BTS lokalizovat uživatele uvnitř průniku daných WiFi vysílačů.

2.2.5 Výhody a nevýhody lokalizace pomocí sítě

Lokalizace uživatele s využitím informací z GSM sítě, případně s využitím WiFi, je velmi rychlá, funguje i uvnitř budov a není náročná na výkon, díky čemuž nedochází k velké spotřebě baterie mobilního zařízení. Při využití ve městech může být tato metoda i přesnější než lokalizace pomocí GPS, pokud je v okolí uživatele více BTS a případně i nějaké přístupové body WiFi. Mimo města v místech s malou hustotou osídlení je však téměř nepoužitelná, jelikož je možné, že v okolí bude jen jedna BTS a tím pádem bude lokalizace velmi nepřesná.

2.3 Hlavní cíle lokalizace

V této kapitole jsou zmíněny zásadní body, nad kterými je důležité se při návrhu aplikace využívat data o poloze uživatele zamyslet. Jednotlivé body jsou také doplněny o konkrétní příklady metod a funkcí, se kterými se lze setkat při implementaci lokalizace uživatele na platformě Android.

Rychlost - Je velmi důležité, aby měla aplikace po svém spuštění co nejdříve k dispozici informaci o pozici uživatele. Při startu aplikace je možné zjistit pomocí *Location Manageru* poslední známou pozici uživatele, pokud je dostupná. Tato pozice nemusí být úplně nejpřesnější, vzhledem k tomu, že od jejího zaznamenání mohl uplynout určitý čas a zařízení mohlo být v pohybu, ale pomůže při startu aplikace zobrazit alespoň přibližnou pozici, než se vypočítá aktuální přesnější pozice. Pokud je k dispozici více posledních známých pozic, můžeme si vybrat, jakou z nich použijeme. Každá

z posledních známých pozic uchovává kromě souřadnic uživatele také údaje o času, kdy byla poloha zaznamenána a s jakou přesností. Díky tomu si lze z dostupných posledních známých poloh vybrat dle potřeby buď tu s nejnovější (s nejbližším časem), nebo tu s největší přesností.

Četnost aktualizací polohy - Pokud aplikace zaznamenává trasu, po které se uživatel pohybuje, je nutné si promyslet, jak často se budou aktualizovat souřadnice uživatele. Interval aktualizací závisí především na rychlosti, jakou předpokládáme, že se bude uživatel pohybovat. V metodě *LocationRequest* nastavíme tedy preferovaný interval aktualizací polohy pomocí *setInterval()* a nejrychlejší povolený interval přes *setFastestInterval()*. Velikosti intervalů jsou zadávané v milisekundách.

Přizpůsobivost - Vzhledem k tomu, že se může stát, že uživatel nebude mít ve svém zařízení povolenou GPS, musí se aplikace umět vypořádat i s takovou situací. Možným řešením je buďto vypsání hlášky, že aplikace pro svůj chod vyžaduje povolení GPS, nebo zvolit jiný zdroj pro zjištění polohy, například pomocí mobilních sítí.

Přesnost - Při návrhu aplikace je dobré si promyslet, jak přesnou pozici uživatele bude aplikace pro správnou funkčnost vyžadovat. Například pokud by se jednalo o aplikaci zobrazující restaurace v okolí, není příliš důležité, jestli pozice uživatele bude určená s přesností na 10 metrů, nebo 100 metrů. Aplikaci v takovém případě stačí jakákoliv přibližná poloha. Ale pokud se jedná například o aplikaci zaznamenávající ujetou trasu, je pro správnou funkčnost nezbytné, aby pozice byla určená co nejpřesněji.

Výdrž baterie - Všechny výše zmíněné body mají vliv na spotřebu baterie, která je u mobilních zařízení velmi důležitým faktorem. Při nastavování jednotlivých parametrů je tedy nutné brát ohledy na spotřebu energie, aby se nestalo, že se uživateli v polovině cesty vybijí jeho mobilní zařízení. Proto například v případě navrhované cyklistické aplikace zvolíme interval mezi jednotlivými aktualizacemi polohy raději 5 sekund (což při průměrné rychlosti 20km/h znamená aktualizaci polohy každých přibližně 25 metrů), než abychom každou půl sekundu vyžadovali nové souřadnice pro co nejpřesnější záznam trasy (po každých přibližně třech ujetých metrech).

Významnou roli má také volba zdroje aktuální polohy, který lze určit metodou *setPriority()* třídy *LocationRequest* [12]. Tato volba v podstatě udává, co je pro aplikaci nejdůležitější, jestli minimální zátěž baterie (na úkor malé přesnosti, která je kolem 10km), nebo vysoká přesnost v rámci metrů (ale za cenu vysoké spotřeby baterie). Podle tohoto nastavení si aplikace zvolí, jestli jako zdroj dat o poloze využije primárně GPS přijímač, mobilních sítí, nebo jejich kombinaci.

2.4 Shrnutí

Při rozhodování, jakou možnost pro lokalizování uživatele použít, je důležité si promyslet, v jakém prostředí a za jakých podmínek bude aplikace především používána. Z tohoto prvotního rozboru pak vychází rozhodnutí, jaký zdroj lokalizace bude vhodné pro aplikaci použít, jak přesná by měla být a v jakém časovém intervalu by se měla aktualizovat.

Z hlediska přesnosti je využití GPS ve venkovním prostředí mimo velká města jasnou volbou. Při vhodných podmínkách dosahuje přesnosti kolem několika metrů [1]. Kdežto při návrhu aplikací, které například budou vyhledávat „body zájmu“ jako restaurace nebo obchody může být použití síťových informací rychlejší, ohleduplnější ke spotřebě baterie a dokonce i přesnější. Jelikož většina uživatelů tohoto druhu aplikací bude ve větších městech, kde je výhled na oblohu značně omezený zástavbou, nebo budou přímo uvnitř nějaké budovy, tak by lokalizování pomocí GPS bylo spíše až sekundárním zdrojem informací o poloze.

3 Práce s mapou na platformě Android

U navrhované cyklistické aplikace se předpokládá vykreslení aktuální polohy a ujeté trasy uživatele do mapy, pomocí jednotlivých bodů zaznamenaných při lokalizaci uživatele. Je tedy nutné využít nějaké mapové podklady, do kterých budou tyto informace vykreslovány.

3.1 Google Maps Android API

Při vývoji aplikací na platformě Android se jako první možnost nabízí využití *Google Maps Android API* [13], které je založené na mapových podkladech *Google Maps*. Rozhraní (API) se automaticky stará o komunikaci se servery, na kterých jsou Google mapy uloženy a o stahování mapových podkladů z těchto serverů. Také zajišťuje zobrazení mapy a interakci uživatele s mapou, jako například přibližování a oddalování, posouvání mapy a další. Pomocí volání různých metod API lze také do mapy vykreslovat body zájmu, trasy pomocí křivek, bitmapové obrázky, měnit mód zobrazení mapy a jiné. Aby mohla aplikace přistupovat k mapovým serverům Googlu, je nutné vygenerovat si vlastní identifikační klíč, takzvaný *Api key*. Pro používání Google map musí mít každý uživatel ve svém zařízení aktuální verzi *Google Play services* a přístup k internetu, aby bylo možné stahovat ze serverů aktuálně potřebné mapové podklady.

3.1.1 Vložení Google map do aplikace

Nejrychlejší způsob vložení Google map do aplikace je pomocí *Android Studio*. Při zakládání nového projektu stačí zvolit z předpřipravených šablon aktivitu vytvoření *Google Maps Activity*. Při sestavování projektu se vytvoří v adresáři *res/values* soubor nazvaný *google_maps_api.xml*, ve kterém se nachází URL adresa a pokyny, jak lze získat *Google Maps API key*, tedy klíč pro přístup aplikace k mapovým serverům společnosti Google. Po zadání URL adresy do prohlížeče je uživatel přesměrován na web *Google Developer Console*, kde je nutné zaregistrovat danou aplikaci, aby mohla používat *Google Maps Android API*. Stačí zde stisknout tlačítko „vytvořit nový projekt“ a k danému projektu se vygeneruje *google maps API key*, který začíná znaky „Alza“. Pro vygenerování klíče je nutné být přihlášen na google účtu. Jedná se o velmi rychlou a díky dobrému popisu i nenáročnou akci. Získání klíče je zdarma a po zadání klíče do souboru *google_maps_api.xml* je možné mapu v aplikaci využívat bez ohledu na počet uživatelů používajících danou aplikaci.

3.1.2 Počáteční nastavení mapy

Po spuštění aktivity s mapou je nastaven počáteční stav mapy, který se má zobrazit. Jedná se o nastavení parametrů mapy jako například přiblížení mapy, souřadnice, na kterých se zobrazí, nebo o jaký typ mapy se bude jednat. Na výběr je pět základních typů map:

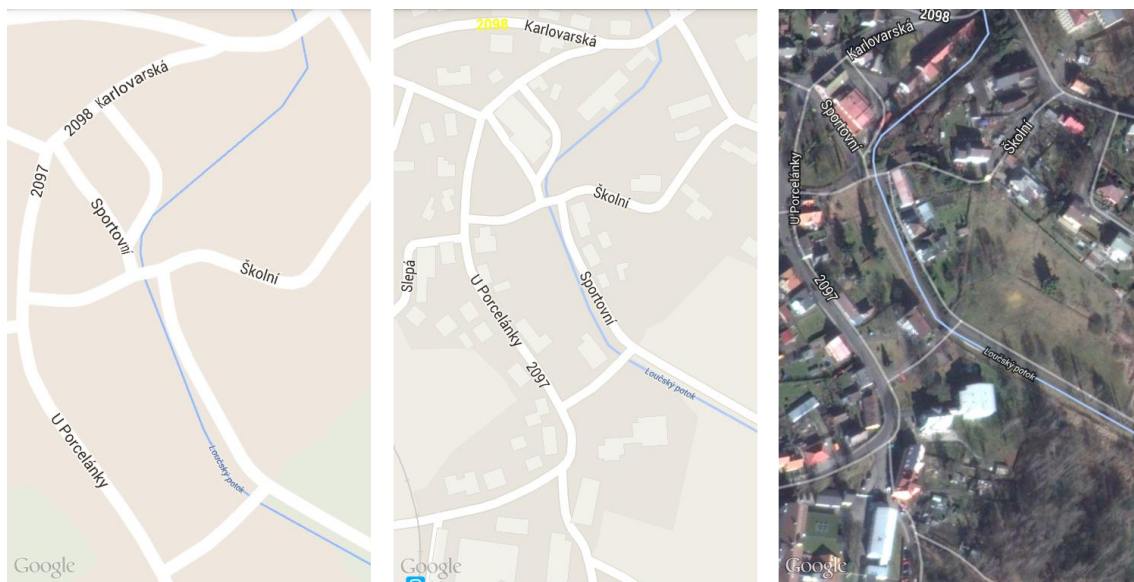
Mapa terénu - zobrazuje model terénu pomocí barevných přechodů, vrstevnic a stínování vyvýšených bodů. Obsahuje také část popisků a zvýraznění cest. Ukázka zobrazení mapy je znázorněna na obrázku 3.1 jako první vlevo.

Normální mapa - jedná se o mapu zobrazující cesty, obrysy staveb a přírodní prvky, jako jsou například vodní plochy. Ukázka vykreslení tohoto typu mapy je znázorněná na obrázku 3.1 uprostřed.

Satelitní mapa - mapa je tvořena pouze ze satelitních snímků, které neobsahují žádné zvýraznění cest nebo přírodních prvků, ani žádné popisky.

Hybridní mapa - hybridní mapa je v podstatě satelitní mapa, která je jen rozšířena o zvýraznění cest, přírodních prvků a popisky. Příklad této mapy si lze prohlédnout na obrázku 3.1 vpravo.

Prázdná mapa - mapa neobsahuje žádná data a zobrazí se pouze prázdná (bílá) plocha, která se ale stále řídí souřadnicovým systémem. To znamená, že i v tomto případě zobrazení jakýchkoliv informací, například bodů, bude přesně odpovídat své zeměpisné šířce a délce.



Obrázek 3.1 - Ukázka zobrazení terénní, normální a hybridní mapy vykreslené s využitím *Google Maps Android API*

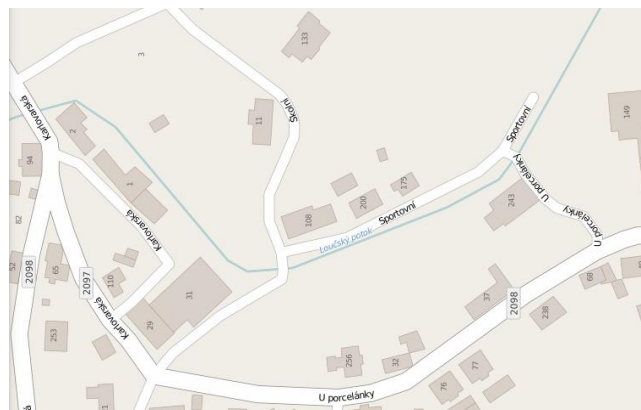
3.1.3 Nevýhody Google Maps Android API

Jedná se o online mapy, kde se všechna data a mapové podklady stahují přímo ze serverů Google v okamžik, kdy se mají vykreslit. Proto je první nevýhodou pro bezproblémovou funkčnost nutnost připojení mobilního zařízení k internetu. Ale není problém, pokud připojení k internetu občas na několik minut vypadne, například pokud uživatel projíždí oblastí se špatným signálem. Pokud je zařízení připojené k internetu, je kromě aktuálně vykreslované mapy stáhnuto také její okolí, takže i při výpadku je mapa vykreslena, ale například s menšími detaily. Další nevýhodou je nutnost mít v mobilním zařízení nainstalovanou nejnovější verzi *Google Play services*, bez kterých Google mapy nelze spustit, ale to mohou ostatně požadovat i jiné aplikace.

3.2 OpenStreetMap

To, že je platforma Android propojená s řadou služeb společnosti Google, včetně Google Maps nemusí nutně znamenat, že je při vývoji aplikací zobrazujících informace o poloze uživatele na mapě jedinou možností použití map od společnosti Google.

Jednou z neznámějších alternativních možností je například projekt *OpenStreetMap* (OSM) [14]. Cílem tohoto projektu je vytváření volně dostupných geografických podkladů, o jejichž správu, editaci a aktualizaci se stará otevřená komunita uživatelů. Pro práci s mapovými daty z *OpenStreetMap* lze vytvořit vlastní rozhraní, nebo použít některé z již hotových řešení. V případě vývoje vlastního řešení spočívá oproti Google mapám velká výhoda především ve volně dostupných mapových podkladech a neomezených možnostech vytvářet vlastní metody a funkce, což umožňuje vývojáři optimalizovat funkce přesně dle potřeb konkrétní aplikace. Pokud by bylo preferováno využití nějakého již hotového rozhraní, existuje celá řada API založených na datech z OSM. Jedním z těchto rozhraní je například *Osmdroid*, které se svou funkcí snaží konkurovat právě výše zmíněnému *Google Maps Android API*. Na obrázku 3.2 je ukázka zobrazení základní mapy *OpenStreetMaps* vykreslené v aplikaci pomocí *Osmdroid API*. Existuje celá škála typů map a vrstev, které lze na mapě vykreslit, jako například cyklistické trasy, různé body zájmu a další.



Obrázek 3.2 - Zobrazení OpenStreetMaps pomocí Osmdroid API

3.2.1 Práce s Osmdroid API

Jako první krok je nutné do projektu aplikace nejdříve importovat knihovnu *Osmdroid*, aby bylo možné v aplikaci využívat možnosti, které rozhraní *Osmdroid* nabízí. Pomocí Android studia je import velmi jednoduchý a stačí pouze přidat do souboru *build.gradle* řádek `compile 'org.osmdroid:osmdroid-android:5.0.1@aar'`, pro import verze 5.0.1, nebo případně zadat jinou verzi knihovny, dle potřeby.

Další důležitou částí je volba zdroje, odkud se budou brát mapové podklady a jakým způsobem budou vykreslovány. Existuje velké množství formátů, ve kterých jsou mapové podklady založené na datech z *OpenStreetMaps* distribuovány. V případě *Osmdroid API* je mapový podklad uložený v databázi malých obrázků (dlaždiček) ve formátu *.png.tile*, ze kterých se vždy poskládá aktuálně vykreslovaná část mapy. O vykreslování mapy z databáze jednotlivých dlaždiček se v tomto případě stará framework *Mapnik* [15]. Databáze s kousky mapy může být dostupná buď online nebo offline. V online případě se při používání aplikace uživateli stahují do zařízení jednotlivé aktuálně zobrazované kousky mapy, aby bylo možné je v budoucnu použít přímo z paměti telefonu, bez nutnosti jejich opětovného stahování ze serverů. V případě vývoje aplikace s offline mapou je potřeba si tato data ve správném formátu nejdříve připravit pro offline použití. Pro stažení těchto dat jsou k dispozici některé programy, ale všechny, které jsem objevil, mají bohužel blokován přístup k serverům *OpenStreetMaps*. Důvodem jejich blokování je fakt, že servery *OpenStreetMaps* fungují pouze díky příspěvkům dobrovolníků a mají omezený výkon a programy pro stahování mapových podkladů je velmi vytěžovaly [16].

Příprava offline dat probíhá v podstatě tak, že je pomocí souřadnic vybrána plocha, jejíž mapové podklady se mají uložit a dále se zvolí stupně přiblížení, které má offline databáze mapových dlaždiček obsahovat. Pro představu při výběru oblasti celé České republiky a pouze čtyř úrovní přiblížení by výsledný balík obsahoval necelé čtyři milióny mapových dlaždiček.

3.2.2 Nevýhody Osmdroid API

Oproti *Google Maps Android API* je implementace *Osmdroid API* o něco složitější a je těžší shánění podrobných informací a návodů. Jedná se o rozhraní naprogramované a popisované komunitou dobrovolníků, tudíž kvalita ani rozsah dokumentace není takový, jako v případě Google dokumentace pro vývojáře. Jako velký problém vidím také složité získávání offline mapových podkladů ve formátu *.png.tile*, což je bez nastavení vlastního serveru pro generování dlaždiček mapy téměř nemožné, kvůli blokování přístupu k serverům *OpenStreetMaps* známým aplikacím, které k tomuto účelu slouží.

3.3 Shrnutí

Pro praktické porovnání zobrazení mapy a vytváření aplikace s využitím obou rozhraní jsem si vytvořil jednoduchou aplikaci zobrazující mapu pomocí *Osmdroid API* a pomocí *Google Maps Android API*. *Osmdroid API* nabízí při vývoji aplikace oproti řešení od Googlu větší volnost a není závislá na společnosti Google a Google Play services. Ale lépe se mi pracovalo s *Google Maps Android API*, vzhledem k lepší dokumentaci a množství návodů, které vývoj značně urychlují. Výhodou OpenStreetMap by tedy mohlo být jen použití offline mapových podkladů. Proto jsem se rozhodl v navrhované aplikaci využít pro práci s mapou *Google Maps Android API*, a pro správné zobrazení mapy bude tedy nutné připojení k internetu.

Jako typ mapy jsem se rozhodnul použít mapu terénu, která názorně zobrazuje profil terénu díky stínování vyvýšených bodů a uživatel by mohl ocenit zobrazení vrstevnic pro lepší představu o výškovém profilu okolí.

4 Aplikace Cyclo

V této kapitole je popisována mnou vytvořená aplikace s názvem Cyclo, která slouží pro podporu cyklistických výletů jednotlivců i skupin na platformě Android.

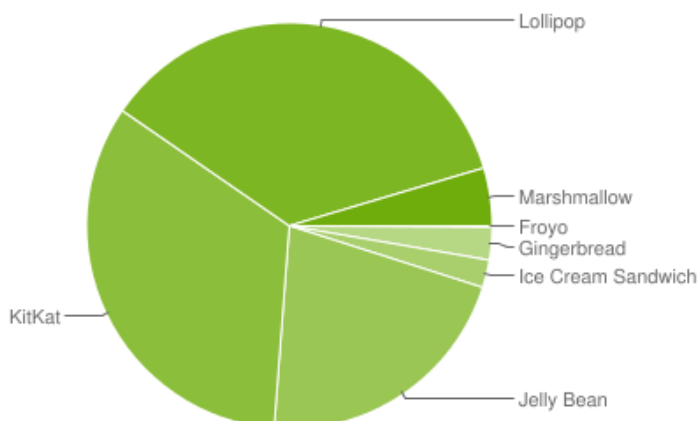
Základními funkcemi aplikace jsou:

- Vykreslování aktuální trasy uživatele do mapy a možnost vypsání informací o dané trase (průměrná rychlost, ujetá vzdálenost a další)
- Možnost uložení ujeté trasy do databáze, včetně automatického průběžného ukládání trasy, aby byla trasa uložena i v případě náhlého ukončení aplikace (například z důvodu vybití baterie)
- Možnost přidat do mapy „body zájmu“ při spuštěném záznamu trasy
- Hlídací funkce (odeslání nouzové SMS s aktuálními souřadnicemi na uživatelem zvolené číslo, pokud se pozice uživatele delší dobu nezmění a mohlo například dojít k jeho pádu a ztrátě vědomí)
- Možnost vytvoření nové skupiny, případně připojení se k existující skupině
- Vykreslení aktuálních pozic jednotlivých členů skupiny na mapě
- Zasílání skupinových zpráv všem členům stejné skupiny

Uživatelský manuál, popisující podrobněji funkce aplikace a práci s ní se nachází v příloze A1. Instalační manuál je umístěn v příloze A2.

4.1 Minimální požadavky aplikace

Minimální požadovanou verzí systému Android je u této aplikace 4.1.x - Jelly Bean s API 16. Tato verze byla vybrána s ohledem na to, aby aplikaci mohlo používat co nejširší spektrum uživatelů, aniž by museli mít nejnovější mobilní zařízení. Dle statistik společnosti Google [17] se odhaduje, že verzi systému Android 4.1 a vyšší má 95,1% všech zařízení se systémem Android. Obrázek 4.1 zobrazuje rozšířenost jednotlivých verzí systému Android, kde čím je barevný odstín tmavší, o to novější verzi se jedná.



Obrázek 4.1 - počet uživatelů jednotlivých verzí systému Android, zdroj [17]

Aplikace pro svou správnou funkčnost také vyžaduje, aby dané zařízení mělo povolené určování polohy pomocí systému GPS a přístup k internetu, aby bylo možné bezproblémové vykreslování mapy, sdílení informací o aktuální poloze s ostatními členy skupiny a zasílání skupinových zpráv u skupinového módu. Uživatel také musí mít ve svém mobilním zařízení nainstalované Google Play services [18] verzi 9.0.83 a vyšší, jejichž součástí jsou Google Maps.

4.2 Povolení, která aplikace vyžaduje

Aplikace při instalaci vyžaduje několik povolení, která musí uživatel odsouhlasit. Všechna tato povolení jsou deklarována v souboru *AndroidManifest.xml* [28] a jsou nezbytná pro jednotlivé funkce aplikace. V souboru *AndroidManifest.xml* jsou z toho důvodu, že při instalaci systém nejdříve zpracovává právě tento soubor, ve kterém se nacházejí základní informace o aplikaci a její požadavky a jen v případě odsouhlasení požadavků se přejde k samotné instalaci a k ostatním souborům aplikace.

4.2.1 Povolení nutná pro lokalizaci uživatele [8]

android.permission.ACCESS_COARSE_LOCATION - povolí aplikaci využití síťových dat pro lokalizaci (data z WiFi a mobilních vysílačů)

android.permission.ACCESS_FINE_LOCATION - povolí aplikaci výpočet aktuální polohy s maximální přesností, díky kombinaci systémů GPS a dat ze sítě

4.2.2 Povolení potřebná pro hlídací funkci

android.permission.SEND_SMS - umožní odesílání SMS přímo samotnou aplikací

android.permission.VIBRATE – povoluje aplikaci vyvolat vibrace telefonu

4.2.3 Povolení potřebná pro Google Maps [21]

com.google.android.providers.gsf.permission.READ_GSERVICES - pro přístup ke Google Play services

android.permission.INTERNET - povolení přístupu přes internet k serverům Google Maps

android.permission.ACCESS_NETWORK_STATE – slouží pro kontrolu stavu připojení, zda je možné data ze serverů stahovat

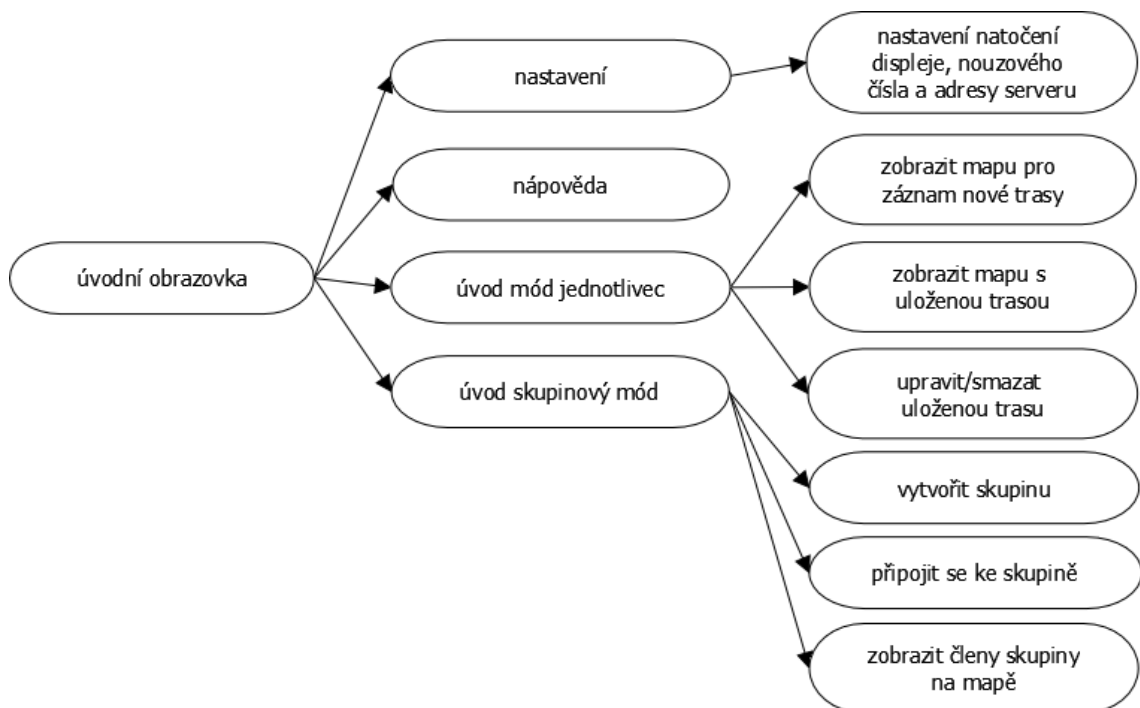
android.permission.WRITE_EXTERNAL_STORAGE - povoluje případný zápis některých mapových dat na externí paměť mobilního zařízení, například pro dočasné uložení mapových podkladů

4.3 Celková struktura aplikace

Aplikace je rozdělena na 15 tříd a 7 aktivit. Pro snadnou správu a případné změny vlastností celé aplikace a slouží třída *Konstanty*. V této třídě jsou umístěny konstanty definující například velikosti intervalů aktualizací informací, typ vykreslované mapy, vlastnosti prvků vykreslovaných na mapě a další.

Po spuštění aplikace je zobrazena úvodní obrazovka, která slouží jako „rozcestník“ pro zvolení požadované akce uživatelem. Uživatel si může vybrat mezi zobrazením módu jednotlivce a skupinového módu, nebo případně zobrazit nastavení a nápovědu.

Módy jednotlivce i skupiny mají svou vlastní úvodní stránku, která slouží pro zobrazení informací a volbu akce. V případě jednotlivce se na úvodní stránce nachází tlačítko pro záznam nové trasy a pod ním jsou vypsány jednotlivé trasy uložené uživatelem. Uživatel si tedy může v módu jednotlivce zvolit mezi záznamem nové trasy, zobrazením uložené trasy na mapě, případně upravit/smazat některou z uložených tras. Ve skupinovém módu jsou zobrazeny tlačítka pro založení nové skupiny a připojení se ke skupině. Pokud je uživatel připojený k nějaké skupině, zobrazí se pod danými tlačítky text s názvem (případně identifikačním číslem) dané skupiny a tlačítko pro zobrazení mapy s jednotlivými členy dané skupiny. Na obrázku 4.2 je znázorněno, jaké konkrétní akce lze vyvolat ze které aktivity.



Obrázek 4.2 – Znázornění akcí, které lze vyvolat z jednotlivých aktivit

4.4 Úvodní obrazovka aplikace

Hlavní třídou aplikace je *UvodniObrazovka*, která je zavolána ihned po spuštění aplikace. Základní funkcí této třídy je zobrazení úvodní obrazovky aplikace. Úvodní obrazovka byla navržena s důrazem na minimalistické provedení a rychlý intuitivní výběr akce, o kterou má uživatel zájem. Obrazovce tedy dominují pouze dvě ikony, a to ikona jednotlivce a skupiny, které slouží jako tlačítka pro přechod do daného módu. Při vývoji aplikace byla využita zdarma dostupná sada ikon [19].

Kromě těchto dvou ikon se v pravém horním rohu úvodního okna nachází tlačítka menu, které po kliknutí nabídne možnost zobrazení nápovědy, případně přechod do nastavení aplikace.

4.4.1 Nápověda

Nápověda je určená pro uživatele, kteří si po spuštění aplikace nejsou jistí, jaký režim zvolit, případně jaké funkce aplikace nabízí a jaké jsou požadavky aplikace nutné pro správnou funkčnost. K zobrazení nápovědy slouží *AlertDialog*, který se sestává z titulku, hlavního textu a tlačítka ve spodní části. U hlavního textu nápovědy je použita funkce *Html.fromHtml()*, sloužící pro převod HTML znaků, což umožnilo snadné formátování textu uvnitř dialogu. Ve spodní části dialogu je umístěné tlačítka „OK“, které slouží ke skrytí dialogu s nápovědou.

4.4.2 Nastavení

Pokud uživatel v menu zvolí položku Nastavení, je zavolána nová aktivita, která umožňuje změnu nastavení vlastností aplikace.

Pro ukládání uživatelských nastavení slouží třída *SharedPreferences* [23]. Data jsou ukládána jako dvojice klíč-hodnota (u nouzového čísla například tedy: ("*nouzoveCislo*", "*123456789*")) a lze takto ukládat primitivní datové typy (*int*, *float*, *boolean*, *long*, *String* a *Set<String>*). Výhodou užití *SharedPreferences* je jak jednoduchý přístup k uloženému nastavení a jeho snadná správa, tak také vlastnost, že nastavení zůstává uložené permanentně, i po ukončení aplikace, nebo vypnutí mobilního zařízení.

Aby nedocházelo k samovolnému otáčení zobrazení displeje, má aplikace zakázané otáčení zobrazení. Ale každému uživateli samozřejmě vyhovuje jiný režim natočení, a proto je v nastavení možnost volby, jestli má být aplikace zobrazována na šířku displeje, nebo na výšku. Nastavení zobrazení je automaticky uloženo po kliknutí na příslušný *RadioButton* a platí pro celou aplikaci. Další možností, kterou nastavení nabízí, je zadání nouzového čísla, na které má být odeslána SMS v případě, pokud aplikace vyhodnotí, že by uživatel mohl být v nesnázích. Nouzové číslo je zadáváno do textového pole, které má omezené zadávání znaků pouze na celá čísla a čísel může být zadáno maximálně devět. Poslední pole v nastavení slouží pro případnou změnu adresy serveru, pokud uživatel bude chtít použít vlastní server pro obsluhu skupinového módu. Pro následné uložení slouží tlačítka pod daným polem.

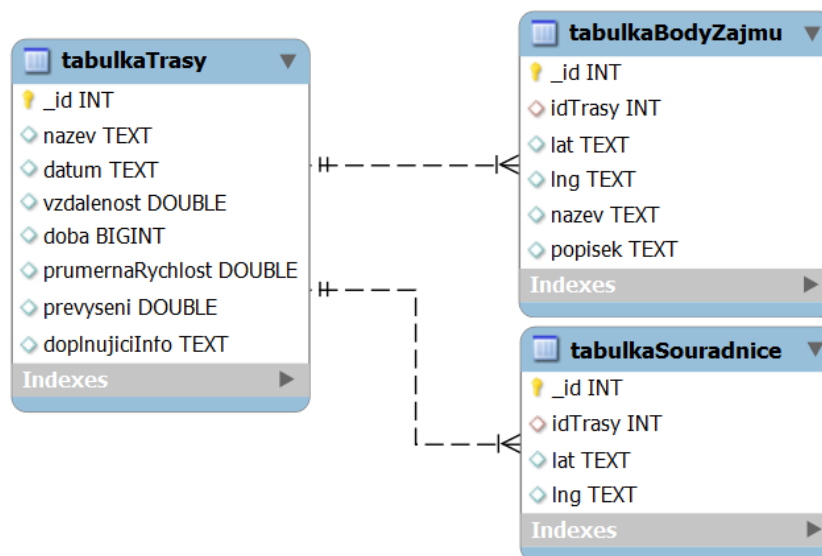
4.5 Uchovávání uložených tras

Během návrhu aplikace jsem při výběru možnosti uchovávání dat k jednotlivým trasám uloženým uživatelem uvažoval mezi interní SQLite databází a XML soubory. Výhodou XML souborů je, že se jedná o standardizovaný formát, u kterého by byla snadná možnost exportu dané trasy z aplikace a případný import dat do jiné aplikace. Použití tohoto řešení by ale bylo náročnější na implementaci, ukládání dat do souboru a jejich čtení by mohlo být při velkém objemu dat časově a paměťově náročné a dle mého názoru by složitější implementace také vedla k nepřehlednému kódu.

Z těchto důvodů jsem se rozhodnul pro využití SQLite databáze [25]. Databáze je ukládána do interní paměti zařízení a všechna data se nacházejí v soukromém adresáři aplikace, do kterého je povolen přístup jen dané aplikaci. Výhodou tohoto řešení je snadná rozšiřitelnost, jednoduchý přístup pro aplikaci k jakýmkoliv datům uloženým v databázi, rychlost a malá paměťová náročnost. Soubory XML by ale mohly být například využity jako formát pro export dat z databáze.

4.5.1 Návrh databázové struktury

Databáze je pojmenována *trasy.db* a skládá se ze tří vzájemně propojených tabulek. První tabulka s názvem *tabulkaTrasy* je hlavní tabulkou a obsahuje základní informace o dané trase, jako například ID trasy, název, popis, celkovou délku trasy, průměrnou rychlost uživatele a další. Druhá tabulka pojmenovaná *tabulkaSouradnice* slouží k ukládání jednotlivých zaznamenaných souřadnic (ty jsou reprezentovány zeměpisnou šířkou a délkou) a každý řádek tabulky obsahuje jako cizí klíč ID trasy, ke které dané souřadnice patří. Poslední tabulkou je tabulka *tabulkaBodyZajmu*, do které jsou ukládány jednotlivé body zájmu, které uživatel vložil do mapy při záznamu trasy. Tabulka s body zájmu má podobnou strukturu jako tabulka se souřadnicemi. Má pouze navíc sloupec pro název bodu zájmu a pro popis daného bodu. Na obrázku 4.3 je znázorněn ERA model této databáze.



Obrázek 4.3 - ERA model SQLite databáze pro ukládání tras uživatele

4.5.2 Práce s databází SQLite

Pro vytvoření databáze, její správu a přístup k datům lze využít rozhraní třídy *SQLiteOpenHelper*. Navrhovaná aplikace obsahuje třídu *DBTrasy*, která třídu *SQLiteOpenHelper* rozšiřuje a dědí její metody. Ve třídě *DBTrasy* je nejdříve definována struktura jednotlivých tabulek dané databáze, které jsou přepsáním metody *onCreate()* v databázi vytvořeny a poté následují metody pro práci s jednotlivými záznamy.

Všechny tabulky mají některé společné metody a některé své specifické. Společnými metodami jsou:

- *pocetRadku(String nazevTabulky)* - metoda vrací celkový počet řádků dané tabulky
- *vyberTrasu(int id, String nazevTabulky, String sloupecID)* - vrátí všechny údaje k trase s odpovídajícím *id* ve sloupci *sloupecID*, z tabulky s daným názvem
- *vyberVsechnyZaznamy(String nazevTabulky)* - vybere všechny záznamy z dané tabulky
- *smazZaznamy(int id, String nazevTabulky, String sloupecID)* - smaže všechny záznamy ze zadané tabulky, které mají ve sloupci *sloupecID* odpovídající *id*

Dále jsou uvedené metody specifické pro potřeby konkrétních tabulek:

- *aktualizujCelouTrasu(int id, String vzdalenost, String prumernaRychlost, long doba, String prevyseni)* - metoda pro aktualizaci statistických informací trasy s daným *id*
- *aktualizujTrasu(int id, String nazev, String doplnujiciInfo)* - aktualizuje název a doplňující informace u trasy s daným *id*
- *lastID()* - slouží pro získání ID poslední vytvořené trasy. Tato metoda je volána ihned po vytvoření nové trasy, aby aplikace měla informaci, k jakému ID trasy má přiřadit souřadnice a body zájmu.
- *vlozBodZajmu(int idTrasy, String lat, String lng, String nazev, String popis)* - vkládá do tabulky s body zájmu nový bod s danými parametry, popis nemusí být vyplněný
- *vlozSouradnici(int idTrasy, String lat, String lng)* - metoda pro vložení nové souřadnice do tabulky se souřadnicemi
- *vlozTrasu(String nazev, String datum)* - vložení nové trasy s daným názvem a datem do tabulky s trasami

4.6 Záznam trasy uživatele

Po kliknutí na tlačítko pro spuštění záznamu nové trasy, na úvodní obrazovce módu jednotlivce, je uživateli zobrazena mapa, na které se nacházejí tři ikonky sloužící jako tlačítka (viz obrázek 4.5). Tato tlačítka umožňují spuštění/ukončení záznamu trasy, vypsání statistických informací o aktuálně zaznamenané trase a přidání bodu zájmu. Aktivitě je nastavena vlastnost, aby nedocházelo při zobrazení mapy k samovolnému vypnutí displeje, pomocí příkazu:

```
getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON)
```

Pro spuštění záznamu nové trasy slouží tlačítko „play“, které se nachází na pravé straně obrazovky. Záznam nové trasy spustí vykreslování ujeté trasy do mapy, uživatel si může během záznamu zobrazit informace o zaznamenané trase a také lze k dané trase vkládat body zájmu. Během záznamu nové trasy se jednotlivé informace automaticky ukládají přímo do databáze, ve které je pro ně vytvořena nová trasa s názvem „Neuložená trasa“. Díky tomu není nutné si udržovat informace o všech souřadnicích trasy v paměti RAM, ale tyto informace jsou přímo ukládány do databáze. Také se tím docílí průběžného ukládání dat, takže nehrozí jejich ztráta, v případě nečekaného ukončení aplikace, například z důvodu vybití baterie nebo nedostatku RAM. V případě, že by k nečekanému ukončení aplikace došlo, zobrazí se po opětovném spuštění aplikace mezi trasami uživatele trasa s názvem „Neuložená trasa“, jelikož nebyla uložena přímo uživatelem. V této trase budou uloženy veškeré informace, souřadnice a body zájmu, které byly zaznamenány před neočekávaným ukončením aplikace.

4.6.1 Získání aktuální pozice uživatele

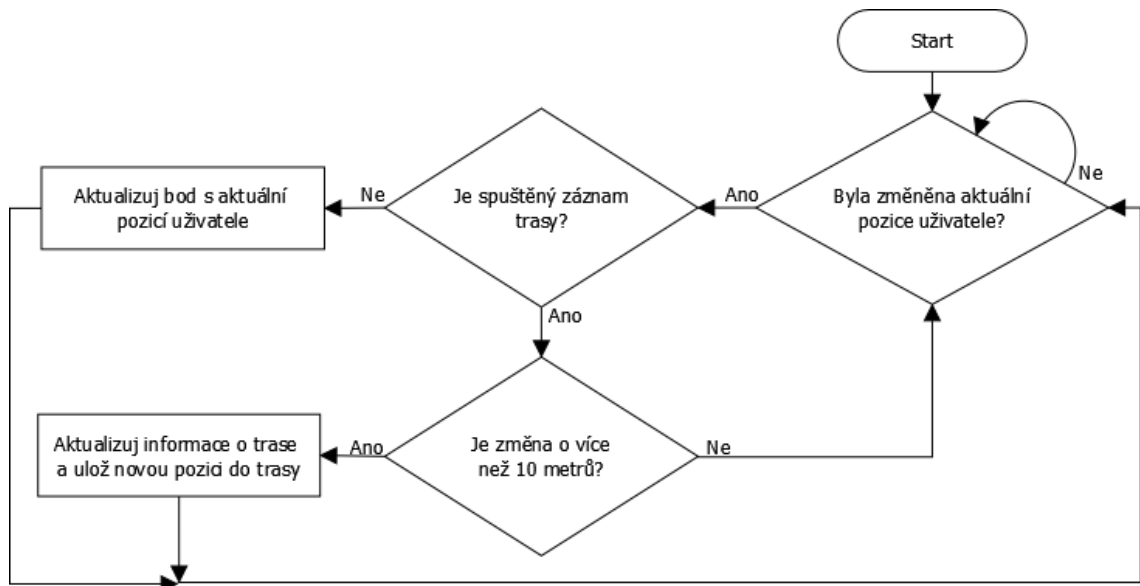
Pro vykreslení trasy je nutné znát aktuální pozici uživatele a být informován, pokud byla pozice změněna. K těmto účelům je nutné nejdříve vytvořit *GoogleApiClient*, který slouží jako prostředník pro komunikaci s rozhraním *LocationServices API* [20]. *LocationServices API* je součástí *Google Play services*, poskytuje aplikaci informace o poloze uživatele a umožňuje aplikaci využívat své lokalizační služby.

Poté je zavolána metoda *buildLocationRequest()*, která vytvoří požadavek na zasílání informací o aktuální pozici uživatele a nastaví parametry, jakou přesnost lokalizačních dat aplikace vyžaduje a v jak častém intervalu. Pro potřeby aplikace Cyclo je nastavena nejvyšší možná přesnost a interval aktualizace přibližně každých 5 sekund (viz kapitola 2.3). Pro start zasílání informací dle vytvořeného požadavku je zavolána metoda *startLocationUpdates()*, která požadavek odešle prostřednictvím *GoogleApiClient*.

Jelikož může několik desítek sekund trvat, než bude k dispozici informace o aktuální poloze uživatele, aplikace využije pro vykreslení mapy nejdříve poslední známou pozici uživatele, pokud je k dispozici. Pokud není, je vykreslení mapy nastaveno na celou Českou republiku. Ihned v okamžiku, kdy je aktuální pozice k dispozici, je na mapě vykreslen bod označující aktuální pozici uživatele.

Kontrolu, jestli došlo ke změně aktuální pozice uživatele, má na starost metoda *onLocationChanged(Location location)*, která má jako parametr novou aktuální pozici. Průběh algoritmu, který je spuštěn vždy po získání nové pozice, je znázorněn ve vývojovém diagramu na obrázku 4.4. Pokud dojde ke změně pozice uživatele a je zapnutý záznam trasy, je zavolána metoda *aktualizuj()*, která novou pozici zpracuje a rozhodne, zda jí aplikace použije či nikoliv. Pokud záznam trasy není spuštěn, pouze se aktualizuje bod s aktuální pozicí uživatele na mapě.

Aby aplikace kvůli drobným odchylkám mezi zjištěnými pozicemi nevyhodnotila, že se uživatel pohybuje, i když by jen stál na místě, je v metodě *aktualizuj()* nastavena podmínka, že se nová poloha musí změnit oproti předchozí o více než 10 metrů, což postačuje pro výrazné omezení ovlivnění z důvodu drobných odchylek lokalizace [1]. Pokud je změna větší než 10 metrů, nová souřadnice je přidána do databáze, zobrazení mapy je vystředěno dle nové pozice a jsou zavolány metody pro vykreslení trasy na mapě a pro přepočítání statistických údajů o dané trase.



Obrázek 4.4 - Vývojový diagram aktualizace polohy uživatele

4.6.2 Vykreslení trasy

Při návrhu aplikace byly zvažovány dva možné přístupy, jak by bylo možné ujetou trasu vykreslit.

První možností bylo využít *Google Directions API* [22]. Toto rozhraní umí vyhledat trasu mezi dvěma nebo více body na základě zadaného módu, kde je na výběr mezi pěší chůzí, jízdou na kole, automobilem a dalšími. Tato funkce má na první pohled řadu výhod. Například při zadaném módu „automobil“ by se hledala mezi body jen taková trasa, která vede po silnicích, případně vedlejších cestách a odpovídá dopravním předpisům. Díky tomu by nedocházelo k vykreslování trasy mimo silnici, k čemuž by

mohlo docházet z důvodu drobných odchylek v lokalizaci. Také by stačilo ukládat méně bodů, ze kterých by se následně cesta vytvářela. Tato možnost se mi ale zdála příliš složitá, pro potřeby vyvíjené aplikace zbytečně velmi rozsáhlá a také by nemusela vždy zobrazit správnou cestu, pokud by cyklista jel například po nevyznačených cestách. Proto jsem se rozhodnul pro druhou možnost, která je jednodušší a plně postačující pro potřeby navrhované aplikace.

Druhou možností je vykreslení křivky na mapě, která bude procházet všemi uloženými body. O vykreslení trasy se stará metoda `vykresliTrasu(LatLng newCoordinate)`, které je jako parametr předána nová souřadnice uživatele. Uvnitř metody je nadefinována `Polyline` pomocí `PolylineOptions()`, které jsou přiřazeny všechny body (souřadnice), jimiž má procházet a také je jí nastavena barva a šířka. Pro vykreslovanou křivku je nastavena funkce `geodesic`, která slouží k vykreslování zaoblených čar mezi jednotlivými body, což působí přirozeněji, než kdyby byly jednotlivé body propojovány rovnými přímkami. Celá křivka je přidána do mapy a postupně jsou jí přidávány další body dle aktuálního pohybu uživatele. Oproti první možnosti je ale nutné ukládat více bodů, aby vykreslení křivky bylo co nejpřesnější. Jak vypadá následné vykreslení trasy v mapě, je znázorněno na obrázku 4.5. Záznam této vykreslené trasy probíhal v terénu s omezenou viditelností (a tím pádem zhoršenými podmínkami pro přesnost lokalizace), a přesto je relativně členitá trasa zaznamenána jen s drobnými odchylkami.



Obrázek 4.5 – Náhled vykreslení zaznamenané trasy aplikací Cyclo

4.6.3 Návrat do aktivity se záznamem trasy po přerušení

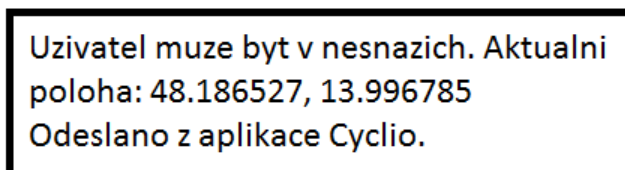
Pokud je aktivita přerušena spuštěním jiné aplikace, případně zamčením obrazovky, je zavolána metoda `onPause()`. Tato metoda měla původně na starost zrušení požadavku na zaslání informací o aktuální pozici. Díky tomu měla být prodloužena výdrž baterie daného mobilního zařízení, jelikož by zařízení nebylo neustále zatěžováno zjišťováním a výpočtem nové pozice. Ihned po návratu zpět do aplikace je zavolána metoda

onResume(), která zajišťovala opětovné vykreslení mapy v případě potřeby, obnovila požadavek na zasílání aktuální pozice uživatele a aplikace plynule pokračovala v záznamu trasy. Po důkladném zvážení jsem ale došel k názoru tuto funkci odstranit a nechat záznam trasy pokračovat i při zamčení telefonu nebo spuštění jiné aktivity. Funkce *onPause()* tedy nevykonává žádnou akci a funkce *onResume()* pouze obnoví v případě potřeby vykreslení zobrazované mapy. Hlavním důvodem bylo dát uživatelům možnost se rozhodnout, jak chtějí aplikaci používat, protože pro uživatele, kteří se budou chtít například podívat na mapu jen zřídka, by bylo zbytečné mít zapnutý displej se spuštěnou aplikací v popředí celou dobu a navíc vypnutí displeje telefonu také přispěje k prodloužení výdrže baterie.

4.6.4 Kontrola, zda je uživatel v pořádku

Pokud uživatel prostřednictvím nastavení aplikace zadá číslo pro případ nouze, je automaticky součástí spuštěného záznamu trasy také „hlídací funkce“. Tato funkce slouží pro kontrolu, jestli je uživatel v pořádku a nedostal se do nějakých nesnází, jako například ztráta vědomí po pádu z kola.

Při každé změně polohy uživatele je automaticky spuštěn časovač, který je nastavený na 6 minut. Hlídací funkce je spuštěna až poté, co uživatel opravdu vyjede (po změně startovní pozice). Pokud v průběhu spuštěného časovače nedojde k žádné změně polohy uživatele, je uživatel po 4 minutách dotázán prostřednictvím *AlertDialogu*, jestli je vše v pořádku a neudělal si například jen během cesty někde pauzu. Zobrazení *AlertDialogu* je navíc doprovázeno vibrační zařízením, aby se snížila pravděpodobnost odeslání poplašné zprávy z důvodu, že by uživatel samotný dialog přehlédnul. Pokud je *AlertDialog* potvrzený stisknutím tlačítka, že je vše v pořádku, časovač se zruší a nic se neděje. Dokud ale dialog není uživatelem skrytý, časovač běží dál a pokud doběhne do konce (pozice uživatele se tedy celých 6 minut nezmění o více než 10 metrů a zároveň uživatel nepotvrdí prostřednictvím dialogu, že je vše v pořádku), odešle se na nouzové číslo zpráva informující o tom, že uživatel může být v nesnázích. Součástí zprávy je také poslední známá pozice uživatele, jak lze vidět na obrázku 4.6.



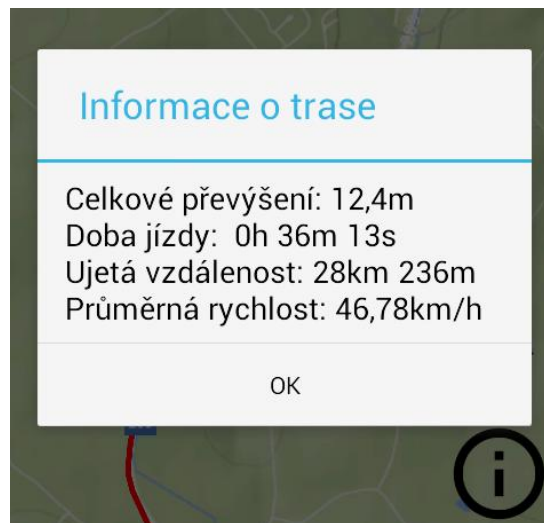
Uživatel muze byt v nesnazich. Aktualni poloha: 48.186527, 13.996785
Odeslano z aplikace Cyclio.

Obrázek 4.6 – Náhled nouzové SMS odeslané aplikací Cyclio

4.6.5 Informace o trase

V průběhu záznamu trasy jsou při každé změně pozice uživatele vypočítávány statistické informace, které má uživatel možnost si zobrazit, kliknutím na ikonku „i“. Jedná se o celkovou ujetou vzdálenost (zobrazovanou v kilometrech a metrech), dobu jízdy od spuštění záznamu trasy (zobrazovanou v hodinách, minutách a sekundách),

průměrnou rychlost v jednotkách km/h a celkové převýšení v metrech. Náhled dialogu s informacemi o trase je zobrazen na obrázku 4.7.



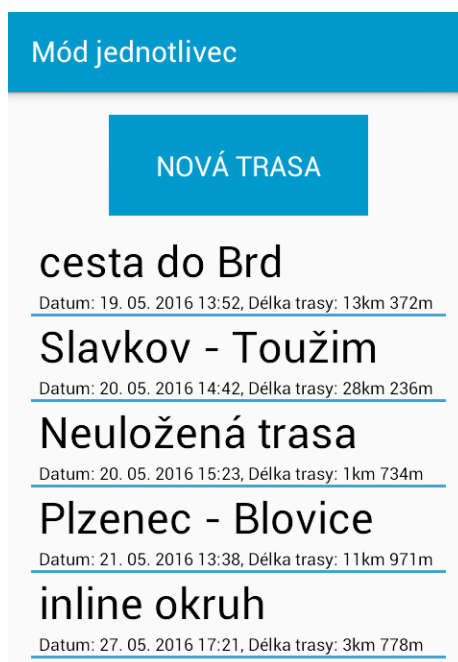
Obrázek 4.7 - Dialog s aktuálními informacemi o trase

Výpočet statistických údajů provádí metoda *statistika(double nadmorskaVyska, double vzdalenostMeziBody)*. Prvním parametrem je aktuální nadmořská výška uživatele a druhým je ujetá vzdálenost od poslední uložené pozice. Celková vzdálenost je vypočtena přičtením druhého parametru k aktuální celkové ujeté vzdálenosti a je ukládána v metrech. Celkový čas aplikace získá odečtením času spuštění od aktuálního času. Získaný údaj je ukládán v milisekundách a pro přehledný výpis je převeden na hodiny, minuty a sekundy. Pro získání průměrné rychlosti v jednotkách km/h pak stačí jen vydělit celkovou vzdálenost v kilometrech celkovým časem v hodinách.

Posledním údajem je celkové převýšení, které udává rozdíl mezi nejvýše a nejniže položeným bodem ujeté trasy (v metrech). Součástí polohy, kterou aplikace získá ze systému GPS je kromě souřadnic také údaj o výšce, ve které se uživatel nachází. Tento údaj je získán z dané polohy pomocí metody *getAltitude()* [27]. Nejedná se ale o nadmořskou výšku, ale o výšku dle referenčního elipsoidu WGS84, což je matematická aproximace tvaru Země. Daná hodnota se od skutečné nadmořské výšky tedy liší. To ale v případě této aplikace nevádí, jelikož aplikace potřebuje pouze zjistit rozdíl mezi nejvýše a nejniže položeným bodem trasy. Bohužel je ale získaná hodnota velmi nepřesná. Jako základní kontrola, pro eliminaci velkých nepřesností, je kontrolována výška mezi po sobě jdoucími body a pokud se liší o více než 3 metry, je daný údaj ignorován a považován za chybný. K hodnotě 3 metry jsem došel na základě předpokladu, že stoupání mezi dvěma po sobě jdoucími body nebude více než 10% (při předpokládané vzdálenosti mezi jednotlivými body 30 metrů, což odpovídá rychlosti 6m/s) [31]. Jedná se ale jen o jednoduchou kontrolu, tudíž údaj o převýšení může být i přesto nepřesný.

4.7 Mód jednotlivce

Mód jednotlivce je určený pro individuální vyjížďky a dělí se na dvě základní části, záznam trasy a zobrazení uložené trasy. Při výběru módu jednotlivce je zavolána třída *JednotlivecUvod*, která zobrazí tlačítko pro spuštění záznamu nové trasy a pod ním seznam tras, které si uživatel již uložil. Náhled úvodní obrazovky módu jednotlivce je zobrazen na obrázku 4.8.



Obrázek 4.8 – Úvodní obrazovka módu jednotlivce

4.7.1 Výpis tras uložených uživatelem

Jednotlivé trasy jsou vypsány prostřednictvím komponenty *ListView*. V základní verzi umožňuje *ListView* zobrazit na každém řádku pouze jeden textový záznam. Aby bylo možné zobrazit nejen název trasy, ale také například druhý řádek s podrobnostmi o trase, případně nastavit i jakýkoliv jiný formát zobrazení trasy, je nutné formát zobrazení jednotlivých řádků upravit vlastním layoutem, který bude definovat rozložení položek v každém řádku *ListViewu* [24]. Layout, definující formát výpisu tras, má název *list_view_layout.xml* a nachází se ve složce *res/layout*. Tento layout obsahuje celkem tři textové pole, kde první pole slouží pro zobrazení názvu trasy, druhé pro výpis informací o trase a ve třetím se nachází ID dané trasy. ID trasy ale slouží pouze pro interní potřeby aplikace a uživateli se nezobrazuje.

Klíčovým krokem při vytváření vlastního formátu zobrazení řádků uvnitř komponenty *ListView* je vytvořit vlastní *ArrayAdapter*, který slouží jako rozhraní mezi *ListView* a daty, které má zobrazit. Přepsáním jeho metody *getView()* lze nastavit každému řádku formát dle vytvořeného layoutu. Tento layout je následně naplněn daty z *ArrayListu seznamTras*, který obsahuje seznam všech uložených tras daného uživatele.

Po kliknutí na jakoukoliv trasu ze seznamu uložených tras je zobrazen *AlertDialog* umožňující výběr mezi akcemi smazat trasu, upravit trasu a zobrazit trasu na mapě. Při zvolení smazání trasy je uživatel vyzván k potvrzení akce, a pokud akci potvrdí, je celá trasa, včetně všech souvisejících údajů v ostatních tabulkách, odstraněna z databáze. Úprava trasy umožňuje změnu názvu trasy a doplňujících informací o trase. Pokud je zvolena možnost zobrazit trasu na mapě, je zavolána třída *MapaUlozenaTrasa* a nové aktivitě je jako parametr předáno ID vybrané trasy.

4.7.2 Zobrazení uložené trasy

Pokud uživatel v hlavním okně módu jednotlivec vybere nějakou trasu ze seznamu uložených tras a zvolí možnost „zobrazit na mapě“, otevře se nová aktivita, ve které je daná mapa vykreslena. Trasa je zde zobrazena ve stejném formátu, v jakém se zobrazovala při záznamu. Po kliknutí na ikonku „i“ si uživatel může vypsat detailnější informace o zobrazené uložené trase, viz obrázek 4.7.

Třída *MapaUlozenaTrasa*, sloužící pro vykreslení uložené trasy, je z velké části shodná s třídou pro záznam trasy nové. Přibyla zde jen metoda *vykresliUlozenouTrasu(int id)*, která jako parametr dostane id trasy, kterou má vykreslit. Na základě id trasy jsou z databáze získány základní informace o trase a ve dvou oddělených cyklech pak také všechny souřadnice dané trasy a případně všechny body zájmu. Body zájmu jsou ihned zobrazeny na mapě a souřadnice trasy jsou ukládány do *PolylineOptions*. Po ukončení obou cyklů je následně vykreslena na mapě křivka s uloženou cestou.

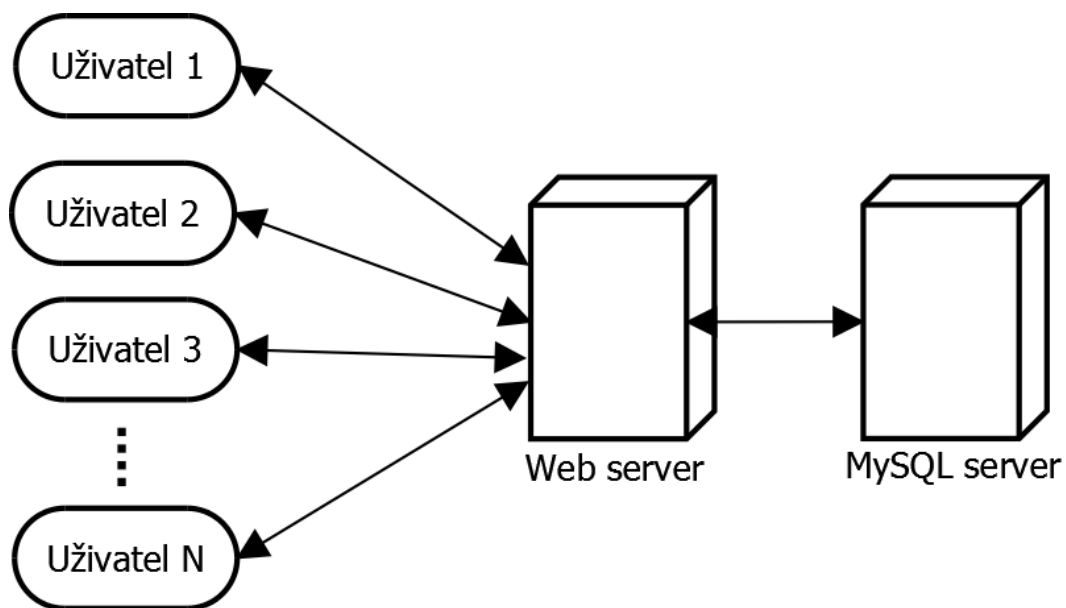
Pokud uživatel klikne na ikonku pro spuštění záznamu trasy, spustí se totožná metoda, jako je u aktivity pro záznam nové trasy a průběh je tedy stejný, jako by se jednalo o záznam trasy nové. Jediným rozdílem je to, že zobrazená uložená trasa zůstává stále vykreslena na mapě a uživatel tedy může danou trasu následovat a zároveň sledovat záznam své aktuální trasy, takže ihned uvidí, pokud by se od uložené trasy odchýlil.

4.8 Skupinový mód

Skupinový mód je určený pro skupinové cyklistické výlety. V rámci tohoto módu je vytvořena skupina o n členech a tito členové spolu vzájemně sdílejí informace o své aktuální poloze a mohou si vzájemně zasílat skupinové zprávy. Konstanta n značí maximální počet členů skupiny, který je určen při zakládání každé skupiny. Pro obsluhu jednotlivých uživatelských požadavků, odeslaných z klientské aplikace v rámci skupinového módu, slouží webový server. Ten komunikuje s databázovým serverem s MySQL databází, ve které jsou uloženy jednotlivé skupinové informace. Na základě informací z databáze následně webový server odesílá uživateli zpět odpověď na daný požadavek, jak lze vidět na obrázku 4.9.

Při výběru skupinového módu je zavolána třída *SkupinaUvod*, která zobrazí tlačítka pro výběr mezi založením nové skupiny a připojením se k již existující skupině. Při

skupinovém výletu tedy jeden člen skupiny na začátku založí novou skupinu, sdělí ostatním údaje potřebné k připojení se k dané skupině (ID skupiny a heslo) a ostatní se k dané skupině připojí. Po úspěšném připojení do skupiny se uživatel stane součástí dané skupiny a může si zobrazit aktivitu se skupinovou mapou. V této aktivitě jsou na mapě vyznačené pozice jednotlivých členů skupiny a je možné zde přijímat a odesílat skupinové zprávy. Zakladatel skupiny je po úspěšném založení automaticky členem dané skupiny. Součástí skupinové mapy jsou kromě skupinových funkcí také stejné funkce, jako v případě módu jednotlivce. Uživatel tedy může i ve skupinovém módu zaznamenávat svou aktuální trasu a tu si následně uložit ke svým ostatním trasám. Vzájemná komunikace uživatelů probíhá zasíláním HTTP POST požadavků s různými parametry na web server prostřednictvím *URLConnection* [29] spojení.

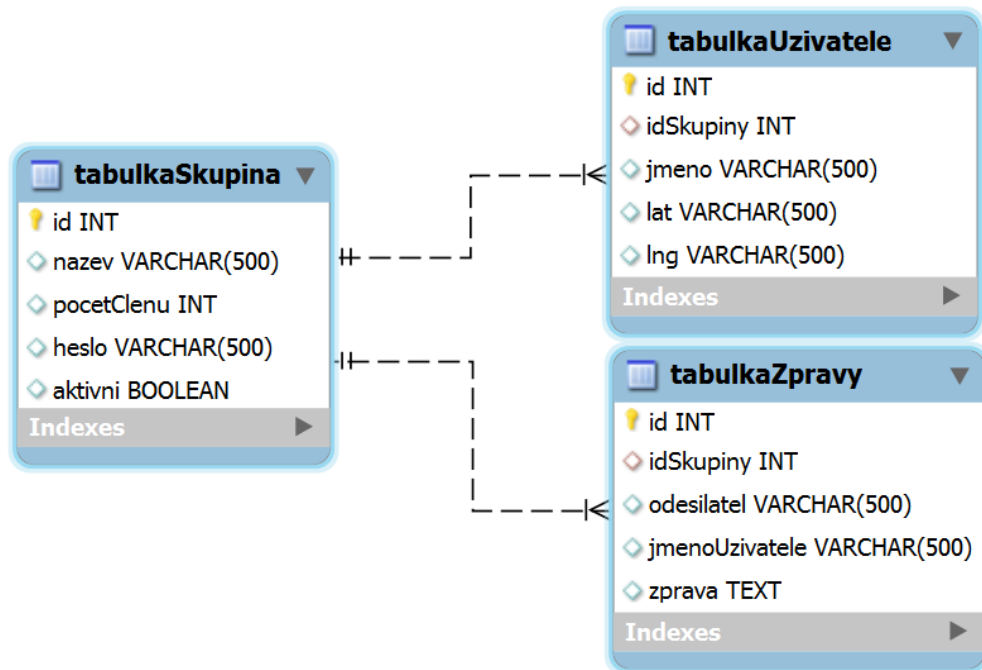


Obrázek 4.9 – Princip fungování skupinového módu a obsluhy jednotlivých požadavků

4.8.1 Návrh databázové struktury skupinového módu

Informace k jednotlivým skupinám jsou ukládány do MySQL databáze na vzdáleném serveru. Databáze obsahuje tabulku s jednotlivými skupinami nazvanou *tabulkaSkupina*, tabulku *tabulkaUzivatele* s aktuálně připojenými uživateli a tabulku s nepřepočtenými skupinovými zprávami *tabulkaZpravy*. V tabulce se skupinami jsou uloženy základní údaje ke každé skupině, tedy název dané skupiny, MD5 hash hesla skupiny, maximální počet členů, kteří mohou být připojeni k dané skupině a informace, jestli je daná skupina aktivní, nebo již byla ukončena. Tabulka s uživateli obsahuje informace o aktuálně připojených uživateli k jednotlivým skupinám. U každého uživatele je uloženo jeho jméno, které je unikátní v rámci skupiny, ID skupiny, ke které je připojen a souřadnice značící aktuální polohu daného uživatele. Do tabulky se zprávami jsou ukládány skupinové zprávy obsahující ID skupiny, v rámci které byly

zaslány, jméno odesílatele zprávy, jméno příjemce a text zprávy. ERA model databáze pro ukládání skupinových informací je zobrazen na obrázku 4.10.



Obrázek 4.10 - ERA model MySQL databáze pro ukládání informací o jednotlivých skupinách

4.8.2 Komunikace se serverem

Pro komunikaci aplikace se vzdáleným webovým serverem a přístup k databázi s informacemi k jednotlivým skupinám slouží třída *OdesliPozadavek*, která rozšiřuje třídu *AsyncTask* [30] a dědí její metody. Třída *AsyncTask* umožňuje vykonávání operací na pozadí v jiném vlákně, než ve kterém běží hlavní aktivita. Díky tomu je komunikace se serverem zpracovávána na pozadí a daná aktivita obdrží od třídy *OdesliPozadavek* až výslednou odpověď od serveru a není tedy v průběhu komunikace nijak omezená její činnost. Průběh zpracování odeslání požadavku na server je takový, že je nejdříve spuštěna metoda *onPreExecute()*, která je zavolána před začátkem vykonávání operací na pozadí. V případě odesílání dat na server u aplikace Cyclo je v této metodě zobrazen *ProgressDialog* informující uživatele o tom, že probíhá odesílání jeho požadavku na server. Poté je zavolána metoda *doInBackground()*, ve které probíhá samotná komunikace se serverem a zpracování odpovědí přijatých ze serveru.

Spojení se serverem je vytvořeno prostřednictvím třídy *HttpURLConnection* [32]. Aby se zamezilo dlouhému čekání při komunikaci se serverem, je nastaven maximální časový limit pro navázání spojení a také pro čtení odpovědi ze serveru na 10 sekund. Jak název třídy napovídá, pro komunikaci klienta se serverem je využíván protokol HTTP (HyperText Transfer Protocol) [33]. Komunikace probíhá na výchozím portu 80 a spolehlivost spojení se serverem zajišťuje protokol TCP (Transmission Control Protocol). HTTP metoda pro předávání parametrů z aplikace Cyclo je nastavena na POST. Aplikace Cyclo tedy naváže přes protokol TCP spojení se serverem dle zadané

URL adresy. V rámci tohoto spojení pošle HTTP požadavek na server, který daný požadavek zpracuje a odešle aplikaci zpět odpověď, která je také formátovaná dle HTTP protokolu. Na prvním řádku odpovědi je dle HTTP protokolu umístěný kód odpovědi [34], který určuje, jestli komunikace se serverem proběhla v pořádku, či nikoliv. Pokud je kód odpovědi v pořádku, tedy *HTTP/1.1 200 OK* u HTTP verze 1.1, metoda pokračuje dál zpracováním odpovědi od serveru. V opačném případě došlo k nějaké chybě spojení, metoda je ukončena a uživateli je vypsána hláška s danou chybou spojení.

Pro zpracování odpovědi od serveru slouží *BufferedReader*, do kterého je uložena celá odpověď a k jejímu postupnému zpracování dochází v cyklu *while*, který čte postupně jednotlivé informace řádek po řádku. Jako první je uložena informace o jakou akci se jedná a na jejím základě je odpověď dále zpracovávána. Po dokončení zpracování odpovědi od serveru na pozadí je zavolána metoda *onPostExecute()*, která skryje *ProgressDialog* a předá odpověď od serveru aktivitě, která daný požadavek odeslala.

4.8.3 Obsluha požadavků na serveru

Zpracování jednotlivých požadavků na serveru zajišťuje PHP soubor *answer.php*, který slouží jako rozhraní pro příjem HTTP požadavků od aplikace, komunikaci s MySQL databází a odesílání odpovědí zpět aplikaci, která daný požadavek odeslala.

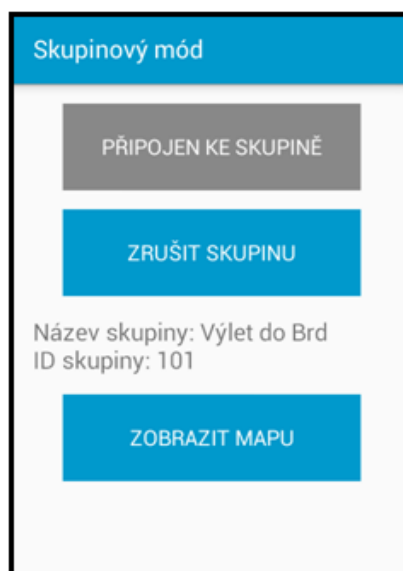
Pro připojení se k databázi slouží funkce *pripojeniKdatabazi()*, umístěná v souboru *answer.php*, která nejdříve načte informace nutné pro připojení se k databázi ze souboru *config.ini*. Informace v souboru *config.ini* obsahují přihlašovací údaje k celé databázi, tudíž je lepší z hlediska bezpečnosti uchovávat tyto informace v jiném souboru, který je dostupný pouze lokálně na serveru, než přímo v souboru ze kterého je voláno připojení se k databázi a který je dostupný online. Po načtení těchto informací se metoda pokusí vytvořit spojení s danou databází a v případě chyby vypíše uživateli chybovou hlášku, že se připojení k databázi nezdařilo.

Jelikož se u většiny z odesílaných parametrů na server jedná o vstupy zadávané přímo uživatelem, je nutné tyto parametry v rámci zabezpečení ihned při přijetí na serveru řádně ošetřit a escapovat případné nežádoucí znaky. K tomuto účelu je použita funkce *mysql_real_escape_string*, která je navržena přímo pro escapování všech znaků, které by mohly být potenciálně nežádoucí v následně volaném MySQL dotazu, čehož například využívá technika SQL Injection [35].

Jednotlivé požadavky, zasílané z aplikace Cyclo na server, vždy začínají parametrem akce, na jehož základě se určí, co uživatel od serveru požaduje a jaká metoda má být zavolána. Po zpracování požadavku je odeslána zpráva s výsledkem zpět na zařízení uživatele, který daný požadavek zaslal.

4.8.4 Založení nové skupiny

Pro založení nové skupiny slouží tlačítko „Založit novou skupinu“, které se nachází na úvodní obrazovce skupinového módu. Po kliknutí na toto tlačítko je zobrazen dialog sloužící pro zadání jednotlivých údajů o nové skupině a všechny údaje jsou povinné. Jedná se o jméno uživatele, pod kterým bude do dané skupiny připojen, název skupiny, heslo skupiny a maximální povolený počet členů, kteří se ke skupině mohou připojit. Zadané údaje jsou následně odeslány na server prostřednictvím třídy *OdesliPozadavek* a pokud je vše v pořádku, je na serveru zavolána funkce *zalozSkupinu()*. Tato funkce zajišťuje vložení nové skupiny do tabulky se skupinami. Zároveň je uživatel, který skupinu vytvořil, automaticky přidán do dané skupiny, takže je do tabulky s uživateli, vložen nový záznam se jménem uživatele a ID dané skupiny. Po úspěšném vytvoření nové skupiny v databázi server vrátí klientovi ID vytvořené skupiny, které je spolu s názvem skupiny zobrazené na úvodní obrazovce skupinového módu, jak lze vidět na obrázku 4.11. Také se uživateli zobrazí nové tlačítko, sloužící pro spuštění aktivity zobrazující mapu s jednotlivými členy dané skupiny a pro příjem a odesílání skupinových zpráv.



Obrázek 4.11 – Náhled úvodní obrazovky skupinového módu se založenou skupinou

4.8.5 Ukončení skupiny

Ukončit skupinu může pouze uživatel, který danou skupinu založil. Po kliknutí na tlačítko „zrušit skupinu“ je vyzván k potvrzení akce a v případě potvrzení je na server odeslána žádost na ukončení skupiny, obsahující ID dané skupiny. Server po přijetí požadavku zavolá funkci *ukonciSkupinu()*, která zajistí v databázi u dané skupiny změnu pole aktivní na 0, což značí, že skupina již byla ukončena. Po ukončení skupiny se k ní již nelze připojit a všem členům skupiny je zobrazena hláška, že došlo k jejímu ukončení a skupina již není aktivní. Při ukončení jsou také z tabulky s uživateli odstraněny záznamy všech členů dané skupiny a z tabulky se zprávami všechny nepřečtené skupinové zprávy, aby byla databáze udržovaná a nedocházelo k zahlcení „zbytkovými“ údaji, které již nemají žádný význam.

4.8.6 Připojení se k existující skupině

Pokud se chce uživatel připojit k nějaké existující skupině, musí od zakladatele dané skupiny znát identifikační číslo dané skupiny (ID skupiny) a heslo. Důvod vyžadování zadání hesla je prostý, a to aby se k informacím o poloze jednotlivých členů skupiny nedostal nikdo, komu nejsou určeny. Ke zvýšení zabezpečení informací v rámci skupiny, slouží také údaj s maximálním počtem členů skupiny, který zadává zakladatel skupiny. Pokud se bude chtít ke skupině připojit někdo, kdo nemá, ale například zaslechl údaje pro připojení se ke skupině, je mu vypsána hláška, že skupina je již plná a není možné se k ní připojit. Kromě ID skupiny a hesla uživatel musí zadat také své jméno, které musí být v rámci skupiny jedinečné. Jedinečnost je vyžadována z toho důvodu, že prostřednictvím daného jména je uživatel v rámci celé skupiny označován na mapě a kdyby byla povolena stejná jména, mohlo by docházet k záměně jednotlivých členů.

Všechny tyto údaje jsou následně zaslány prostřednictvím třídy *OdesliPozadavek* na server, kde se o jejich zpracování stará funkce *pridejDoSkupiny()*. Tato funkce postupně ověří, jestli jsou zadané ID skupiny a heslo správné, jestli je daná skupina stále aktivní, jestli není naplněn maximální počet členů a jestli je zadané jméno v rámci skupiny unikátní. Pokud jsou všechny tyto podmínky splněny, provede se přidání uživatele s ID dané skupiny do tabulky s uživateli v databázi a uživateli je zaslána odpověď o úspěšném přidání do skupiny. Po úspěšném připojení se do skupiny je daný uživatel ihned zobrazen ostatním členům skupiny na skupinové mapě, ale zatím bez aktuální pozice. Pro zasílání informací o své aktuální pozici musí daný uživatel spustit aktivitu se skupinovou mapou, která lze spustit kliknutím na nově zobrazené tlačítko „zobrazit mapu“. Po spuštění aktivity se skupinovou mapou je zahájeno odesílání informací o aktuální pozici uživatele na server, danému uživateli jsou na mapě vykresleny aktuální pozice jednotlivých členů skupiny a také může přijímat a odesílat skupinové zprávy.

Odpojení uživatele od skupiny zajišťuje funkce *odeberZeSkupiny()*, která po přijetí požadavku na odebrání uživatele smaže z databáze záznam o daném uživateli. Tím dojde k jeho odebrání ze skupiny a již nebude vykreslován na skupinové mapě jednotlivých členů skupiny.

4.8.7 Aktualizace skupinových informací

Každý člen skupiny může kliknutím na tlačítko „zobrazit mapu“ spustit aktivitu zobrazující jednotlivé členy skupiny na mapě a vypisující skupinové zprávy. Tato aktivita je definována třídou *MapaSkupina()* a podporuje mimo skupinových funkcí také stejné funkce, které jsou dostupné v módu jednotlivce. Je tedy také možný záznam aktuální trasy, zobrazení statistických informací o zaznamenané trase (ve kterých navíc přibyl údaj o čase poslední aktualizace skupinových informací), uložení trasy mezi trasy uživatele a hlídací funkce.

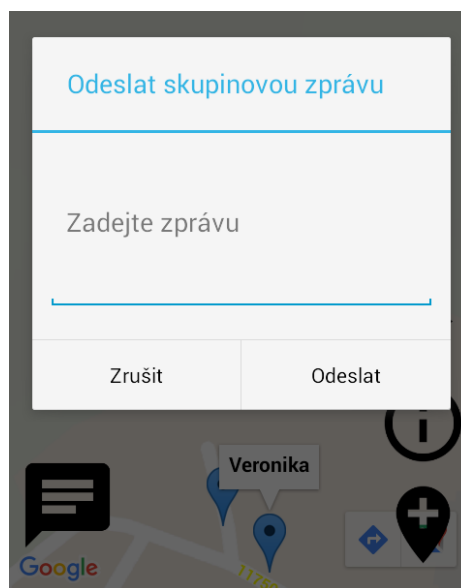
Pravidelnou aktualizaci skupinových informací zajišťuje service, který je zahájen třídou *MapaSkupina()* ihned po jejím spuštění. Service, definovaný třídou *ObsluhaZprav*, která rozšiřuje třídu *Service* [36] a dědí její metody, je spuštěn ve vlákně aktuální aktivity a slouží pro vykonávání operací na pozadí. Díky vykonávání operací, zjišťujících aktualizaci jednotlivých informací o skupině na pozadí, nedochází k zatěžování samotné aktivity komunikací se serverem. Jelikož při nedostatku RAM může operační systém zařízení ukončit libovolný service, aby část paměti uvolnil, je nutné nastavit mód spuštění pro service na *START_STICKY* [38], což zajistí, že v případě ukončení service operačním systémem jej systém znovu obnoví ihned, jak to bude možné. V případě aplikace Cyclo je v service spuštěn časovač, který v pravidelném intervalu 10 sekund zasílá na server požadavek na aktualizaci informací o pozicích všech členů dané skupiny a skupinových zpráv. V rámci požadavku je také na server zaslána aktuální pozice uživatele, který daný požadavek odesílá.

Zpracování požadavku na serveru zajišťuje funkce *aktualizujSkupinu()*, která nejdříve ověří, jestli je skupina stále aktivní. Pokud ano, uloží nové souřadnice uživatele do databáze a odešle uživateli zpět odpověď obsahující jména a aktuální pozice ostatních členů skupiny a všechny uživatelem nepřečtené skupinové zprávy. Zpracování odpovědi ze serveru je také prováděno na pozadí v service a po zpracování je výsledek odeslán prostřednictvím broadcastového vysílání. Broadcastovou zprávu s výsledkem zpracování mohou přijímat všechny aktivity, které mají nastavený *BroadcastReceiver* s identifikátorem daného service. V případě aplikace Cyclo naslouchá broadcastovým zprávám pouze třída *MapaSkupina()*, ve které je nadefinovaný *BroadcastReceiver*, filtrující všechny přijaté broadcastové zprávy dle identifikátoru zadaného v service. Po přijetí zprávy ze service spustí třída *MapaSkupina()* akci dle přijatého výsledku. V případě přijetí nových pozic uživatelů dojde k překreslení všech bodů značících jednotlivé členy skupiny na mapě. Při doručení skupinové zprávy je daná zpráva zobrazena v *AlertDialogu* a její zobrazení je provázeno krátkou vibrací, aby o dané zprávě uživatel ihned věděl i v případě vypnutého displeje.

4.8.7 Skupinové zprávy

Pro odesílání a příjem skupinových zpráv musí být uživatel součástí nějaké skupiny a musí mít spuštěnou aktivitu se skupinovou mapou. K odeslání nové zprávy slouží ikonka zprávy, umístěná v levé části obrazovky. Po kliknutí na tuto ikonu je zobrazen dialog pro napsání skupinové zprávy, který lze vidět na obrázku 4.12. Komponentou pro zadání zprávy je *EditText*, který je ve výchozím stavu nastavený jako jednořádkový. Aby bylo možné psát zprávu na více řádků a docházelo k zalamování textu, je nutné nastavit komponentě vlastnost *setSingleLine(false)* a výchozí výška pole je určena pomocí nastavení *setMinLines(4)*, což značí výšku 4 řádky. Po odeslání zprávy se o její zpracování na serveru postará funkce *skupinovaZprava()*, která nejdříve zjistí, jestli je skupina, pro kterou je daná zpráva určená aktivní a pokud ano, je daná zpráva uložená do databáze. Je ale nutné zajistit, že zpráva bude doručena všem členům skupiny a

všem pouze jednou, i když má každý uživatel jinak stabilní připojení k internetu, takže se někomu zobrazí zpráva ihned a někomu například až po minutě. Jako nejjednodušší řešení mi přišlo vložit danou zprávu do databáze zvlášť pro každého člena skupiny. Takže při každé aktualizaci skupinových informací funkce zkontroluje, jestli v tabulce se skupinovými zprávami existuje zpráva se jménem uživatele, který požadavek poslal a ID jeho aktuální skupiny a pokud ano, tak mu danou zprávu předá a zároveň ji odstraní z databáze, jelikož již byla uživateli doručena. Jelikož je ale povolené zadat do zprávy odřádkování, které by *BufferedReader* považoval při přijetí zprávy aplikací Cyclico za oddělení jednotlivých prvků, je nutné před odesláním zprávy uživateli na serveru odřádkování nahradit mezerou, což zajišťuje příkaz `str_replace(PHP_EOL, " ", $zprava)`, kde konstanta `PHP_EOL` označuje odřádkování a proměnná `$zprava` obsahuje danou zprávu.



Obrázek 4.12 – Náhled obrazovky s mapou ve skupinovém módu

5 Testování

Testování aplikace probíhalo průběžně po celou dobu jejího vývoje. Aplikace byla testována na mobilních telefonech HTC Desire X, Gigabyte GSmart Essence 4, Samsung Galaxy Xcover 3 a HTC Desire 516. Přehled parametrů jednotlivých zařízení, obsahující verzi operačního systému Android (OS), verzi API, velikost paměti RAM v MB a rozlišení displeje se nachází v tabulce 5.1.

Testování aplikace bylo zaměřeno na zobrazení a ovládání aplikace uživatelem, správnost a přesnost jednotlivých naměřených a vykreslovaných dat a praktické testování jednotlivých scénářů, které mohou při používání aplikace nastat.

| Testované zařízení | Verze OS | API | RAM (MB) | Rozlišení displeje |
|---------------------------|----------|-----|----------|--------------------|
| HTC Desire X | 4.1 | 16 | 768 | 800x480 |
| Gigabyte GSmart Essence 4 | 4.4.2 | 19 | 512 | 800x480 |
| Samsung Galaxy Xcover 3 | 5.1.1 | 22 | 1536 | 800x480 |
| HTC Desire 516 | 4.3 | 18 | 1024 | 960 x 540 |

Tabulka 5.1 – Přehled parametrů u testovacích zařízeních

5.1 Zobrazení a ovládání aplikace

Zobrazení všech aktivit aplikace je na všech testovaných zařízeních korektní a všechny prvky jsou dostupné a správně vykreslené při natočení zobrazení obrazovky na šířku i na výšku. Vykreslování mapy je plynulé a zobrazovaná mapa odpovídá aktuální pozici uživatele. Ani chvilkové výpadky internetového spojení v oblastech s horší dostupností signálu neovlivnily správné vykreslení mapy, pouze občas došlo ke zhoršení kvality zobrazení mapy na daném úseku trasy.

Ovládání aplikace je velmi intuitivní a při testování s uživateli, kteří aplikaci Cyclo viděli poprvé a byli seznámeni jen se základními informacemi o tom, jaké funkce aplikace obsahuje, neměl nikdo z testerů žádný problém najít a spustit požadovanou funkci.

5.2 Běh aplikace

Běh aplikace je velmi plynulý a reakce na všechny akce vyvolané uživatelem je téměř okamžitá. Občasnou výjimkou je načítání aktivit s mapou, kdy v některých případech dochází k menší prodlevě před spuštěním aktivity, z důvodu čekání na inicializaci vykreslované mapy. Menší prodleva může nastat také při načítání úvodní obrazovky módu jednotlivce v případě, že má uživatel již uložené větší množství tras, jelikož jsou při jejím spuštění načítány do *ListView* informace o všech uložených trasách. Ale ve všech testovaných případech byla doba načítání aktivit maximálně do 2 sekund, což se dá považovat vzhledem k povaze zobrazovaných dat za zanedbatelný čas.

Funkčnost aplikace byla ve většině testovaných případů bezproblémová a naprosto v pořádku. Pouze v některých případech došlo v průběhu spuštěného záznamu trasy k neočekávanému ukončení aktivity, zapříčiněnému nedostatkem paměti RAM. K této chybě ale docházelo pouze výjimečně, většinou pokud na daném testovacím zařízení byly spuštěné další aplikace na pozadí a telefon měl tedy k dispozici omezené množství RAM, například méně než 100MB, což není mnoho, vzhledem k tomu že kolem 50MB z dané paměti zaberou jen služby Google Play, které jsou využívány pro vykreslování mapy v aplikaci.

5.3 Správnost a přesnost dat

Aby aplikace byla v reálném provozu použitelná a její využití mělo smysl, je nutné, aby zobrazovaná a zaznamenávaná data byla co nejpřesnější a nedocházelo k příliš velkým nepřesnostem.

5.3.1 Přesnost lokalizace

Přesnost lokalizace mobilního zařízení ovlivňuje mnoho faktorů a nelze garantovat žádný maximální rozsah odchylek a nepřesností, ke kterým může dojít, viz kapitola 2.1.5. Důležitým předpokladem, pro správné určení aktuální polohy, je povolení určování polohy pomocí systému GPS. Pokud uživatel spustí aktivitu využívající polohu uživatele bez povolení GPS, je mu vypsán dialog s upozorněním, že lokalizace nemusí bez tohoto povolení fungovat správně.

Na základě všech testování, provedených na různých mobilních zařízeních a v různých podmínkách, které mohou při cyklistickém výletu nastat, lze konstatovat, že určování polohy je i při mírně omezeném výhledu na oblohu, nebo při nepříznivém počasí velmi přesné (s odchylkou maximálně v jednotkách metrů).

5.3.2 Záznam trasy

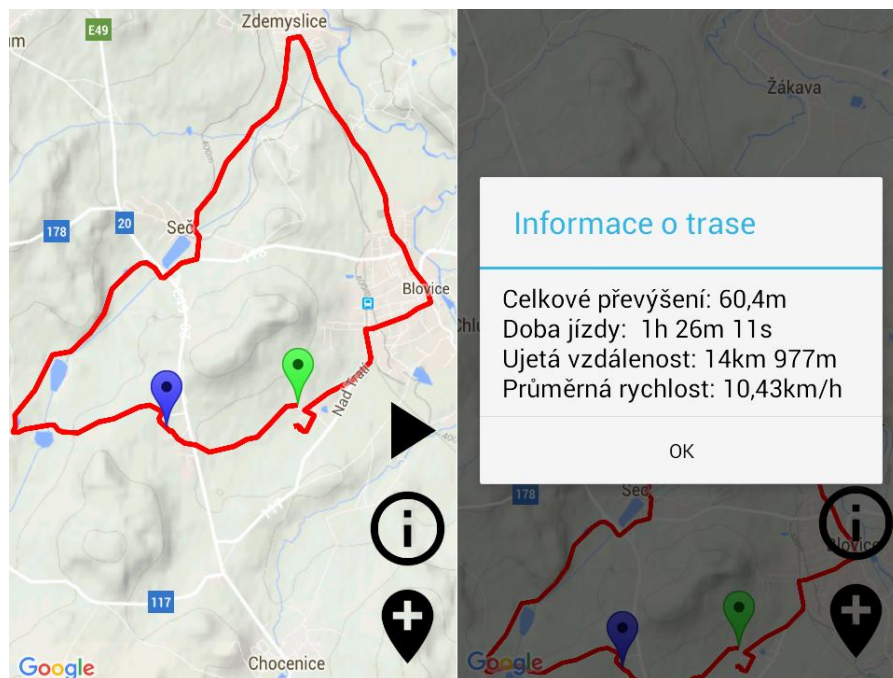
Záznam trasy je plynulý a odpovídá téměř přesně skutečné trase, po které se daný uživatel pohybuje. Kladně se projevilo také nastavení, že se dva po sobě jdoucí body musí lišit nejméně o 10 metrů, aby byla nová pozice zaznamenána. Díky tomu bylo značně omezeno vykreslování „úskoků“, ke kterým docházelo z důvodu drobných odchylek lokalizace, pokud uživatel v průběhu záznamu trasy zastavil na jednom místě a nepohyboval se.

Při záznamu nové trasy v aktivitě, která zároveň vykresluje uloženou trasu, jsou obě trasy zobrazeny korektně. A pokud se uživatel pohybuje po stejné trase, jako je trasa uložená, obě trasy se dle očekávání téměř shodují a navzájem opisují.

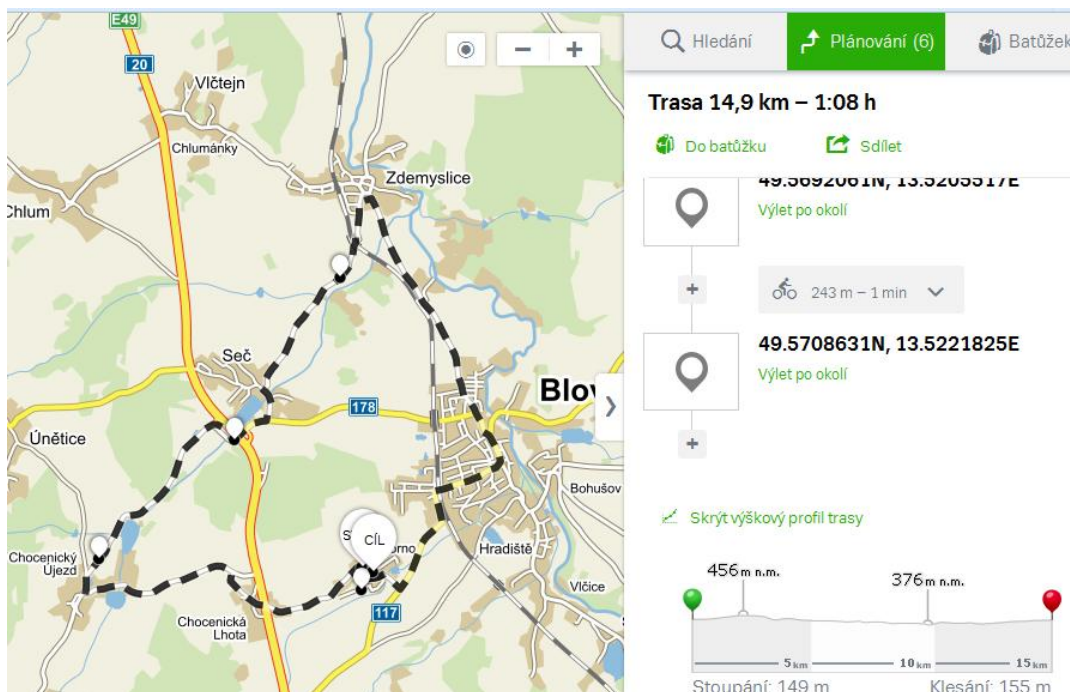
5.3.3 Ukládání informací o trase

Při spuštěném záznamu trasy jsou jednotlivé souřadnice trasy, body zájmu a informace o dané trase ukládány bez jakýchkoliv problémů a jejich následné zobrazení odpovídá

reálným údajům. Na obrázcích 5.1 a 5.2 lze vidět srovnání vykreslení trasy, zaznamenané při testovacím cyklistickém výletě aplikací Cyclo, spolu s naměřenými statistickými údaji k této trase (obr 5.1) a hodnot pro totožnou trasu vypsanych na serveru mapy.cz (obr 5.2).



Obrázek 5.1 – Trasa a statistické údaje k trase zaznamenané aplikací Cyclo



Obrázek 5.2 – Totožná trasa k obr 5.1 a statistické údaje k trase vypočtené serverem mapy.cz (zdroj: mapy.cz)

Jak lze ze srovnání obrázků 5.1 a 5.2 vidět, vykreslení trasy aplikací Cyclo na mapě je naprosto přesné a statistické údaje odpovídají skutečnosti. Čas potřebný pro ujetí trasy

je také korektní, jelikož byla v rámci záznamu trasy aplikací Cyclo průměrná rychlost 10,4 km/h, kdežto server mapy.cz počítá s průměrnou rychlostí 13,4 km/h. Jen údaj o celkovém převýšení byl různý přibližně o 20 metrů, tudíž došlo na základě testování ještě k drobné úpravě výpočtu převýšení, viz kapitola 4.6.5.

Pokud při spuštěném záznamu trasy dojde k neočekávanému pádu aplikace, všechny zaznamenané informace o dané trase jsou uloženy naprosto správně a daná trasa je v seznamu uložených tras označena názvem „Neuložená trasa“.

5.3.4 Skupinové funkce

Skupinový mód vyžaduje pro svou správnou funkčnost připojení k internetu. Bez internetového připojení je uživateli vypsána hláška s informací o této skutečnosti. V případě funkčního internetového připojení fungují všechny skupinové funkce správně a dle očekávání. Jednotlivé vstupy zadávané uživatelem jsou ošetřeny a správnost všech podmínek nutných pro připojení se ke skupině/založení skupiny jsou ověřovány korektně. Aktualizace pozic jednotlivých členů skupiny a jejich zobrazení na mapě funguje bezproblémově a odpovídá skutečnosti. Skupinové zprávy dorazí všem uživatelům téměř okamžitě a v případě, že má některý z uživatelů výpadek internetového připojení, je mu zpráva korektně doručena ihned po obnovení internetového připojení. Testování skupinových funkcí probíhalo při reálných výletech 2 a 3 osob, ale také byla v rámci testování simulována skupina o 20 členech. Skupinové funkce aplikace fungovaly během všech testování správně a plynule a to i při větším množství členů skupiny.

5.3.5 Hlídací funkce

Pokud není v aplikaci nastaveno číslo pro případ nouze, je tato funkce automaticky deaktivována. V opačném případě funguje naprosto správně. Pokud se uživatel 4 minuty nepohybuje, je vypsán dialog doprovázený vibrací. Po kliknutí na tlačítko dialogu je hlídací funkce pozastavena a automaticky se obnoví až při zaznamenání pohybu uživatele. Pokud na dialog uživatel do 2 minut nezareaguje, je odeslána zpráva na zadané nouzové číslo, která obsahuje přesné aktuální souřadnice, kde se uživatel nachází.

5.3.6 Další testované scénáře

Aplikace byla testována na funkčnost bez připojení zařízení k internetu. Skupinový mód nelze vůbec spustit a je pouze vypsána hláška informující uživatele, že pro spuštění skupinového módu musí mít přístup k internetu. Ostatní části aplikace fungují i bez nutnosti internetového připojení korektně. Při spuštění aktivity s mapou, je uživatel informován hláškou, že je internetové připojení nutné pro správné vykreslování mapy a mapa nemusí být zobrazena vůbec, nebo bude mít horší kvalitu. Nicméně funkčnost aplikace není omezena a záznam trasy, spolu se zobrazováním aktuální polohy funguje korektně.

Při drobných výpadcích internetového připojení v rámci módu jednotlivce i skupiny, nedochází k žádným problémům, aplikace funguje korektně a nedochází k jejímu zamrzání nebo neočekávaným pádům. Jen ve skupinovém módu nelze v daném okamžiku přijímat a odesílat skupinové informace, což je ale pochopitelné a aktuální informace jsou zobrazeny ihned po obnovení spojení.

Při testování byla odhalena chyba, že pokud v testovacím zařízení nebyla vložena paměťová karta, nebylo možné spustit žádnou aktivitu zobrazující mapu. Tento problém byl způsoben chybou ve starších verzích Google Play Services, která již byla odstraněna a od verze Google Play Services 9.0.83 k dané chybě nedochází [39].

Zobrazení a funkčnost aplikace se vždy korektně přizpůsobí preferencím uživatele uloženým v nastavení aplikace. Toto nastavení je správně uloženo v zařízení po celou dobu, až do změny uživatelem, případně do odinstalování aplikace.

Nutnou podmínkou pro správný záznam trasy je mít v zařízení povolené určování polohy pomocí systému GPS. Pokud uživatel spustí záznam trasy bez tohoto povolení, záznam trasy je spuštěn, ale uživatel je upozorněn hláškou, že je nutné určování polohy povolit, jinak nebude záznam trasy fungovat korektně, případně vůbec.

6 Možnosti rozšíření aplikace

V rámci vývoje aplikace Cyclo a na základě provedeného testování jsem dospěl k několika možnostem, o jaké funkce by bylo vhodné realizovanou aplikaci dále rozšířit a jak vylepšit současnou funkcionalitu.

Sdílení ujetých tras

Aby byla aplikace pro uživatele zajímavější, bylo by určitě dobré přidat možnost sdílení ujetých tras se známými prostřednictvím sociálních sítí, případně pro inspiraci ostatních uživatelů umožnit sdílení trasy se všemi uživateli aplikace.

Body zájmu

Dalším rozšířením by mohlo být volitelné zobrazování všech bodů zájmu uložených uživatelem na mapě. Díky tomu by uživatel při záznamu nové trasy mohl vidět, jestli není například poblíž nějakého bodu zájmu, který si uložil někdy předtím u jiné trasy. Také by mohlo být užitečné umožnit zadávání skupinových bodů zájmu, aby bod zájmu přidáný jedním uživatelem byl zobrazen i všem ostatním členům dané skupiny.

Statistické údaje

Statistické údaje ukládané k trase by bylo možné rozšířit o nespočet dalších údajů, jako například aktuální počasí, maximální rychlost, které uživatel dosáhnul. Případně na základě statistických údajů vypisovat globální statistiku s údaji, kolik kilometrů uživatel již ujel celkově od nainstalování aplikace a další.

Správa skupiny

U skupinového módu by bylo vhodné přidat zakladateli skupiny možnost zobrazení jednotlivých připojených členů a možnost správy těchto členů a údajů skupiny.

Výpis uložených tras

Výpis uložených tras na úvodní stránce módu jednotlivce by mohl být načítán až po spuštění dané aktivity, aby nedocházelo k časové prodlevě při jejím spuštění. Navíc by vypadalo lépe a přehledněji, kdyby byl každý záznam s uloženou trasou doplněn o obrázek s vykreslenou danou trasou.

Pauza záznamu trasy

Při spuštění záznamu nové trasy by mohlo být také kromě tlačítka stop také tlačítko pauze. Po kliknutí na toto tlačítko by byly záznam trasy a případně také hlídací funkce pozastaveny a po opětovném kliknutí znovu obnoveny.

Obnovení záznamu trasy

Pokud by došlo k pádu aplikace, bylo by vhodné rozšířit aplikaci o funkci, umožňující pokračovat po obnovení aplikace v záznamu aktuální trasy od bodu ukončení.

7 Závěr

Cílem této práce bylo vytvoření systému pro podporu cyklistických výletů jednotlivců i skupin, s aplikací na platformě Android a serverovou částí pro obsluhu skupinových funkcí.

Výsledkem práce je teoretický rozbor a především funkční systém pro podporu cyklistických výletů jednotlivců i skupin s klientskou aplikací na platformě Android. Díky teoretickému rozboru lze získat lepší představu, jaké možnosti je možné využít pro lokalizaci uživatele mobilního zařízení a jaké lze použít mapy a mapové podklady v rámci vytváření aplikací pro zařízení s operačním systémem Android. Tyto znalosti byly následně uplatněny při návrhu aplikace a její tvorbě.

Výsledná aplikace nabízí základní funkce, jako například vykreslování aktuální trasy uživatele do mapy, vypsání informací o dané trase, možnost uložení zaznamenané trasy, včetně automatického průběžného ukládání trasy, nebo vkládání bodů zájmu do aktuálně zaznamenané trasy.

Oproti obdobným aplikacím výsledná aplikace Cyclio nabízí také z mého pohledu velmi užitečné funkce, jako například hlídací funkci, která automaticky odešle nouzovou SMS, obsahující aktuální souřadnice uživatele, na uživatelem zvolené číslo, pokud se jeho pozice delší dobu nezmění a mohlo například dojít k jeho pádu a ztrátě vědomí. Další přidanou hodnotou je rozšíření aplikace o skupinový mód, který umožňuje zobrazení aktuálních pozic jednotlivých členů stejné skupiny na mapě a zasílání skupinových zpráv všem členům v rámci dané skupiny.

Testování celého systému bylo provedeno na netriviálním vzorku dat, v různých podmínkách a s různými testovacími scénáři, které by mohli nastat. Všechny chyby nalezené při testování byly odstraněny a aplikace by měla fungovat naprosto korektně.

Práci považuji za úspěšnou, měla by splňovat všechny požadavky a cíle, které byly určeny v rámci zadání práce. Zároveň si navzdory názvu práce myslím, že má díky svým funkcím mnohem širší možnost využití, než jen pro cyklistické výlety.

Uživatelům, kteří by chtěli začít s vývojem aplikací pro platformu Android, nebo se jím už zabývají, bych doporučil knihu Vývoj aplikací pro Android [41]. Kniha obsahuje rady pro úplné začátečníky, ale i informace užitečné pro pokročilejší uživatele.

Přehled zkratk

GPS, *Global Positioning System* - globální polohovací systém, sloužící k určení polohy

A-GPS, *Assisted GPS* - GPS využívající informace ze vzdálených serverů, sloužící pro urychlení lokalizace

GSM, *Global System for Mobile Communications* - systém standardizující mobilní komunikaci

BTS, *Base Transceiver Station* - základnová stanice vysílače/přijímače mobilního signálu

CGI, *Cell Global Identity* - identifikační číslo vysílače v rámci BTS

TA, *Timing Advance* - vzdálenost mobilního zařízení od vysílače

WiFi - v rámci práce užívané pro označení bezdrátové sítě

SSID, *Service Set Identifier* - název WiFi sítě

BSSID, *Basic Service Set Identification* - fyzická adresa vysílače WiFi signálu

API, *Application Programming Interface* - rozhraní, poskytující aplikaci určitou funkcionalitu

OSM, *OpenStreetMap* - projekt poskytující volně dostupné mapové podklady

SMS, *Short Message Service* - v rámci práce užívané pro označení krátkých textových zpráv

RAM, *Random Access Memory* - operační paměť zařízení

MD5 - hashování funkce

HTML, *HyperText Markup Language* - značkovací jazyk, používaný pro tvorbu hypertextových dokumentů

XML, *Extensible Markup Language* - obecný značkovací jazyk

HTTP, *HyperText Transfer Protocol* - protokol pro výměnu hypertextových dokumentů

TCP, *Transmission Control Protocol* - protokol pro spolehlivý přenos dat

URL, *Unique Resource Locator* - jednoznačné určení zdroje v rámci internetové sítě

PHP - skriptovací programovací jazyk

Literatura

- [1] National Coordination Office for Space-Based Positioning, Navigation, and Timing, *GPS Accuracy* [online], [cit. 5. 12. 2015]. Dostupné z: <<http://www.gps.gov/systems/gps/performance/accuracy>>.
- [2] National Coordination Office for Space-Based Positioning, Navigation, and Timing, *How GPS works* [online], [cit. 18. 12. 2015]. Dostupné z: <<http://www.gps.gov/multimedia/poster/poster-web.pdf>>.
- [3] Information-analytical centre, *Frequently Asked Questions* [online], [cit. 15. 12. 2015]. Dostupné z: <<https://www.glonass-iac.ru/en/guide/navfaq.php>>.
- [4] Jiří Kuruc, *Pořádek ve zkratkách: co znamená A-GPS?* [online], [cit. 7. 11. 2015]. Dostupné z: <<http://navigovat.mobilmania.cz/clanky/poradek-ve-zkratkach-co-znamenava-a-gps/sc-265-a-1314495>>.
- [5] Information-analytical centre, *GLONASS* [online], [cit. 15. 12. 2015]. Dostupné z: <<https://www.glonass-iac.ru/en/index.php>>.
- [6] Odbor ITS, kosmických aktivit a VaVal, *GALILEO - Evropský globální navigační družicový systém* [online], [cit. 3. 12. 2015]. Dostupné z: <<http://www.czechspaceportal.cz/3-sekce/gnss-systemy/galileo>>.
- [7] Ondřej Franěk, *Lokalizace volajícího při tísňovém volání z mobilního telefonu* [online], [cit. 24. 11. 2015]. Dostupné z: <http://www.zachrannasluzba.cz/odborna/0306_lokmt.htm>.
- [8] *Location Strategies* [online], [cit. 9. 12. 2015]. Dostupné z: <<http://developer.android.com/guide/topics/location/strategies.html>>.
- [9] Tim Bray, *A Deep Dive Into Location* [online], [cit. 16. 11. 2015]. Dostupné z: <<http://android-developers.blogspot.cz/2011/06/deep-dive-into-location.html>>.
- [10] Steven J. Vaughan-Nichols, *How Google--and everyone else--gets Wi-Fi location data* [online], [cit. 13. 12. 2015]. Dostupné z: <<http://www.zdnet.com/article/how-google-and-everyone-else-gets-wi-fi-location-data>>.
- [11] National Coordination Office for Space-Based Positioning, Navigation, and Timing, *Frequently Asked Questions* [online], [cit. 10. 12. 2015]. Dostupné z: <<http://www.gps.gov/support/faq>>.
- [12] *LocationRequest* [online], [cit. 14. 6. 2016]. Dostupné z: <<https://developers.google.com/android/reference/com/google/android/gms/location/LocationRequest>>.

- [13] *Introduction to the Google Maps Android API* [online], [cit. 19. 12. 2015]. Dostupné z: <<https://developers.google.com/maps/documentation/android-api/intro>>.
- [14] *OpenStreetMap* [online], [cit. 17. 12. 2015]. Dostupné z: <<http://www.openstreetmap.org/about>>.
- [15] *Mapnik* [online], [cit. 5. 6. 2016]. Dostupné z: <<http://wiki.openstreetmap.org/wiki/Mapnik>>.
- [16] *Blocked applications* [online], [cit. 20. 12. 2015]. Dostupné z: <http://wiki.openstreetmap.org/wiki/Blocked_applications>.
- [17] *Platform Versions* [online], [cit. 7. 4. 2016]. Dostupné z: <<http://developer.android.com/about/dashboards/index.html>>.
- [18] *Overview of Google Play Services* [online], [cit. 7. 4. 2016]. Dostupné z: <<https://developers.google.com/android/guides/overview>>.
- [19] *Material icons* [online], [cit. 20. 3. 2016]. Dostupné z: <<https://design.google.com/icons>>.
- [20] *Making Your App Location-Aware* [online], [cit. 8. 4. 2016]. Dostupné z: <<http://developer.android.com/training/location/index.html>>.
- [21] *Google Maps Android API* [online], [cit. 8. 4. 2016]. Dostupné z: <https://developers.google.com/maps/documentation/android-api/config#specify_android_permissions>.
- [22] *Google Maps Directions API* [online], [cit. 10. 4. 2016]. Dostupné z: <<https://developers.google.com/maps/documentation/directions>>.
- [23] *Storage Options* [online], [cit. 6. 4. 2016]. Dostupné z: <<http://developer.android.com/guide/topics/data/data-storage.html>>.
- [24] *Android Custom ListView Items "Row"* [online], [cit. 13. 3. 2016]. Dostupné z: <<http://hmkcode.com/android-custom-listview-items-row>>.
- [25] *Saving Data in SQL Databases* [online], [cit. 6. 4. 2016]. Dostupné z: <<http://developer.android.com/training/basics/data-storage/databases.html>>.
- [26] Obaro Ogbo, *How to use SQLite to store data for your Android app* [online], [cit. 30. 3. 2016]. Dostupné z: <<http://www.androidauthority.com/use-sqlite-store-data-app-599743>>.

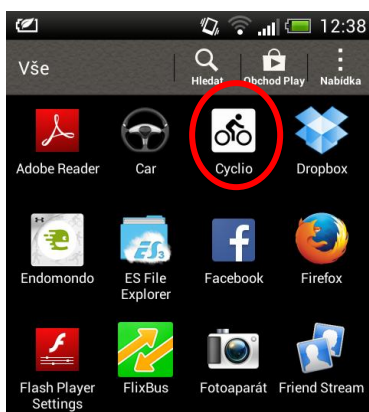
- [27] *Android: How to get accurate altitude* [online], [cit. 12. 4. 2016]. Dostupné z: <<http://stackoverflow.com/questions/9361870/android-how-to-get-accurate-altitude>>.
- [28] *App Manifest* [online], [cit. 20. 4. 2016]. Dostupné z: <<http://developer.android.com/guide/topics/manifest/manifest-intro.html>>.
- [29] Abhay Anand, *Android Httpurlconnection Post and Get Request Tutorial* [online], [cit. 9. 5. 2016]. Dostupné z: <<https://studytutorial.in/android-httpurlconnection-post-and-get-request-tutorial>>.
- [30] *AsyncTask* [online], [cit. 28. 5. 2016]. Dostupné z: <<https://developer.android.com/reference/android/os/AsyncTask.html>>.
- [31] *Jaký je vztah mezi úhlem a procentem stoupání svahu* [online], [cit. 29. 5. 2016]. Dostupné z: <http://www.offroad-rc.info/WordPress/?page_id=6168>.
- [32] *URLConnection* [online], [cit. 29. 5. 2016]. Dostupné z: <<https://developer.android.com/reference/java/net/URLConnection.html>>.
- [33] Dušan Janovský, *HTTP protokol* [online], [cit. 16. 6. 2016]. Dostupné z: <<https://www.jakpsatweb.cz/server/http-protokol.html>>.
- [34] *HTTP protokol - Typy odpovědí serveru* [online], [cit. 16. 6. 2016]. Dostupné z: <<https://www.jakpsatweb.cz/server/http-protokol.html#odpovedi>>.
- [35] *SQL Injection* [online], [cit. 16. 6. 2016]. Dostupné z: <http://www.w3schools.com/sql/sql_injection.asp>.
- [36] *Service* [online], [cit. 16. 6. 2016]. Dostupné z: <<https://developer.android.com/reference/android/app/Service.html>>.
- [37] Lionite, *Using PHP with MySQL - the right way* [online], [cit. 17. 6. 2016]. Dostupné z: <<https://www.binpress.com/tutorial/using-php-with-mysql-the-right-way/17>>.
- [38] *START_STICKY and START_NOT_STICKY* [online], [cit. 17. 6. 2016]. Dostupné z: <<http://stackoverflow.com/questions/9093271/start-sticky-and-start-not-sticky>>.
- [39] *NullPointerException from Google maps* [online], [cit. 5. 6. 2016]. Dostupné z: <<http://stackoverflow.com/questions/34188394/nullpointerexception-from-google-maps>>.
- [40] *Apache* [online], [cit. 20. 6. 2016]. Dostupné z: <<https://httpd.apache.org/>>.
- [41] LACKO, Ľuboslav. Překlad HERODEK, Martin. *Vývoj aplikací pro Android*. 1. vyd. Brno: Computer Press, 2015, 472 s. ISBN 978-80-251-4347-6.

A1 Uživatelská příručka

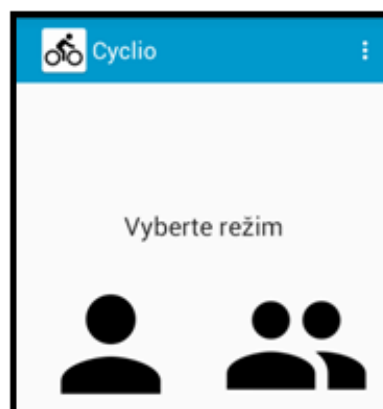
Tato uživatelská příručka slouží jako průvodce používáním aplikace Cyclo a k seznámení uživatele s jednotlivými funkcemi této aplikace.

Spuštění aplikace

Aplikaci lze spustit kliknutím na její ikonu v základním menu, viz obrázek 1, na kterém je ikona aplikace Cyclo vyznačena červeným ohraničením. Po kliknutí na tuto ikonu je zobrazena úvodní obrazovka aplikace, zobrazená na obrázku 2.



Obrázek 1 - spuštění aplikace ze základního menu



Obrázek 2 – úvodní obrazovka aplikace Cyclo

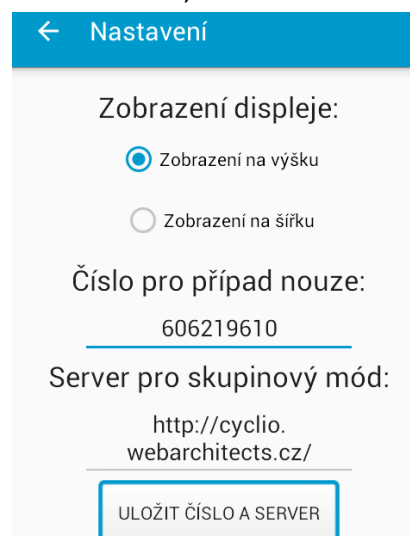
Úvodní obrazovka aplikace slouží jako rozhraní pro výběr módu jednotlivce nebo skupinového módu, případně pro přechod do nastavení a zobrazení nápovědy.

Nastavení a nápověda

Provést změnu nastavení aplikace, případně zobrazit nápovědu, je možné prostřednictvím menu v pravé horní části úvodní obrazovky. Po kliknutí na ikonku značící menu je zobrazena nabídka pro výběr požadované akce, viz obrázek 3.



Obrázek 3 – nabídka zobrazená po kliknutí na menu

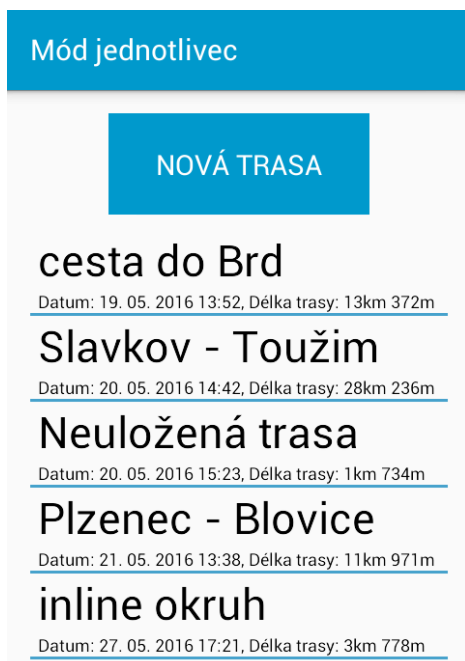


Obrázek 4 – nastavení aplikace Cyclo

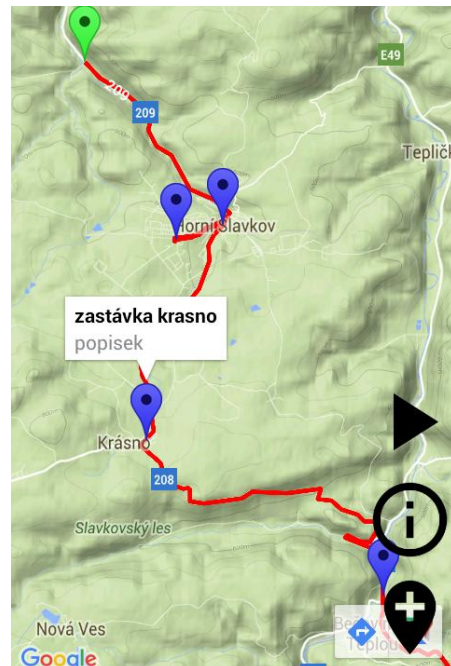
V případě výběru položky „Nastavení“ je zobrazena aktivita sloužící pro změnu nastavení aplikace, viz obrázek 4. Prostřednictvím nastavení lze zadat/změnit číslo pro případ nouze, zvolit preferované natočení displeje (jelikož je v rámci aplikace zakázáno měnit natočení displeje dle natočení telefonu) a nastavit adresu serveru, který slouží pro obsluhu skupinového módu. Možnost změny serveru je určena pouze pro testovací účely, kdyby chtěl někdo spustit aplikaci na svém vlastním serveru. Při výběru položky „Nápověda“ je zobrazen dialog s nápovědou, obsahující popis základních funkcí aplikace a požadavky nutné pro její správnou funkčnost.

Mód jednotlivce

Po kliknutí na ikonu jednotlivce, která se nachází v levé části úvodní obrazovky, je spuštěn mód jednotlivce, který slouží pro individuální používání aplikace. Úvodní obrazovka módu jednotlivce obsahuje seznam všech uživatelem uložených tras a také tlačítko pro spuštění záznamu nové trasy, viz obrázek 5. Pro správnou funkčnost zobrazení a záznamu trasy je nutné mít v zařízení po celou dobu záznamu trasy povolené určování polohy pomocí systému GPS a připojení k internetu. Záznam trasy je plně funkční i bez internetového připojení, ale může dojít ke zhoršení kvality vykreslované mapy, případně nebude mapa vykreslena vůbec.



Obrázek 5 – úvodní obrazovka módu jednotlivce



Obrázek 6 – záznam trasy s vloženými body zájmu

Pokud uživatel stiskne tlačítko pro záznam nové trasy, je zobrazena aktivita s mapou a třemi ovládacími tlačítky, viz obrázek 6. Tato tlačítka, popisována od shora dolů, slouží pro spuštění/uložení záznamu trasy, zobrazení statistických informací o dané trase a přidání bodu zájmu do mapy. Pokud není spuštěn záznam trasy, na mapě je vykreslován pouze bod označující aktuální pozici uživatele, ale žádné jiné funkce nejsou aktivní. V opačném případě je na mapě vykreslována aktuální trasa, po které se uživatel pohybuje, začínající zeleným bodem. Zaznamenávají se také aktuální

statistické informace k dané trase a lze vkládat body zájmu. Bod zájmu je vložen na aktuální pozici uživatele, obsahuje nadpis a popis a je označen modrou barvou, viz obrázek 6. Výslednou trasu, včetně všech informací, je následně možné uložit do seznamu uložených tras a kdykoliv v budoucnu si jí zobrazit. Uložení lze provést kliknutím na tlačítko „stop“, které je po spuštění záznamu trasy vykresleno místo tlačítka „play“.

Po kliknutí na jakoukoliv trasu ze seznamu uložených tras, je zobrazen dialog pro výběr akce, kterou uživatel chce provést. Na výběr je změna informací o dané trase (název a doplňující informace), odstranění dané trasy nebo zobrazení trasy na mapě. Při zvolení zobrazení trasy je daná trasa vykreslena na mapě včetně všech statistických informací a vložených bodů zájmu, takže je zobrazení stejné jako v případě záznamu trasy, viz obrázek 6. Uživatel může zároveň při zobrazení uložené trasy spustit záznam trasy nové, přičemž uložená trasa zůstane zobrazená na mapě. Díky tomu lze uloženou trasu následovat a uživatel ihned na mapě uvidí, pokud by se od dané trasy odchýlil.

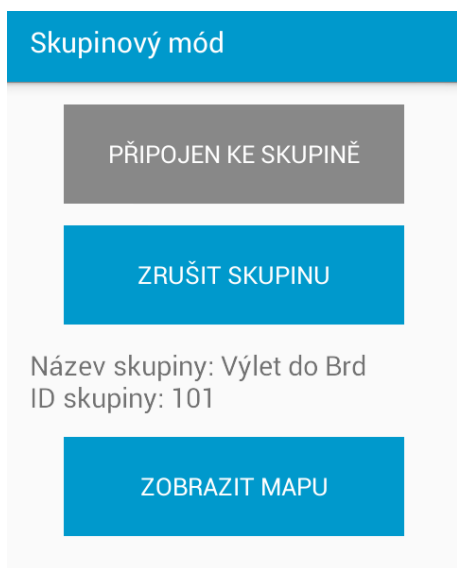
Skupinový mód

Tento mód je určen pro uživatele, kteří se vydají na společný cyklistický výlet. Ve skupinovém módu si každý člen skupiny může zobrazit aktuální pozice ostatních členů stejné skupiny na mapě a zároveň je možné přijímat a odesílat skupinové zprávy všem členům skupiny. Pro spuštění a správnou funkčnost skupinového módu je nutné mít v zařízení po celou dobu povolené určování polohy pomocí systému GPS a připojení k internetu.

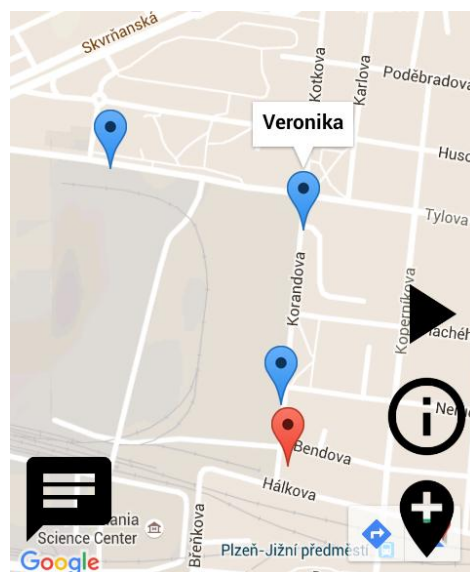
Na začátku výletu je nutné, aby jeden z uživatelů založil novou skupinu kliknutím na tlačítko „založit novou skupinu“. U nové skupiny musí být vyplněn název a heslo skupiny, maximální počet členů, kteří se mohou ke skupině připojit a jméno uživatele, který skupinu zakládá. Po úspěšném založení skupiny se danému uživateli zobrazí na úvodní obrazovce informace o dané skupině a tlačítko pro zobrazení skupinové mapy, viz obrázek 7. Uživatel řekne ostatním členům ID skupiny a heslo a ti se zadáním daných údajů mohou ke skupině připojit. Pro úspěšné připojení ke skupině musí být vyplněny všechny údaje, přičemž heslo musí být shodné s heslem skupiny s daným ID, jméno uživatele musí být v rámci dané skupiny jedinečné a ke skupině nesmí již být připojen maximální počet členů. Po úspěšném připojení ke skupině se uživateli zobrazí název dané skupiny a tlačítko pro zobrazení skupinové mapy.

Pro využívání skupinových funkcí je nutné mít spuštěnou aktivitu se skupinovou mapou, kterou lze spustit kliknutím na tlačítko „zobrazit mapu“. Aktivita má stejné funkce, které jsou dostupné při záznamu nové trasy v módu jednotlivce. Navíc ale zobrazuje na mapě aktuální polohu jednotlivých členů skupiny, kteří jsou značeni světle modrými body, jak lze vidět na obrázku 8. Vlastní aktuální poloha uživatele je zobrazena jako červený bod. Při zobrazení skupinové mapy je také možné přijímání a odesílání skupinových zpráv. Pro odeslání skupinové zprávy slouží ikonka zprávy v levé

dolní části obrazovky, viz obrázek 8. Odeslaná zpráva je zobrazena všem členům připojeným k dané skupině.



Obrázek 7 – úvodní obrazovka skupinového módu po založení nové skupiny

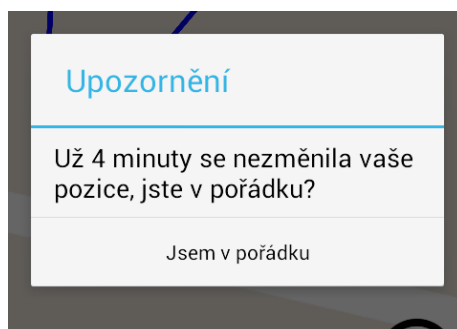


Obrázek 8 – zobrazení skupinové mapy s členy skupiny a aktuální polohou uživatele

Hlídací funkce

Součástí aplikace je také hlídací funkce, která umožňuje zaslání nouzové SMS na předem definované číslo v případě, že by se uživateli mohlo něco stát. Například pokud by došlo k pádu z kola a uživatel by byl v bezvědomí.

Nutnou podmínkou, aby hlídací funkce fungovala, je mít uložené nouzové číslo v nastavení aplikace, mít spuštěný záznam trasy a danou trasu zahájit (mít alespoň jednou aktualizovanou pozici). Hlídací funkce kontroluje, jestli se uživatel pohybuje a v případě že 4 minuty nezaznamená žádný pohyb, je zobrazen dialog s dotazem, jestli je uživatel v pořádku, viz obrázek 9. Pokud si uživatel udělal například jen pauzu, stačí daný dialog skrýt kliknutím na tlačítko „Jsem v pořádku“ a nic se neděje. Pokud ale není další 2 minuty provedena žádná akce (a uživatel se tedy celých 6 minut nepohnul, ani nekliknul na zobrazený dialog), je na nouzové číslo uvedené v nastavení odeslána nouzová SMS s aktuální pozicí uživatele.



Obrázek 9 – zobrazení upozornění od hlídací funkce

A2 Instalační příručka

Instalační příručka obsahuje návod, jak nainstalovat aplikaci Cyclo na uživatelské zařízení a požadavky potřebné pro spuštění serveru sloužícího pro obsluhu skupinového módu.

Instalace aplikace Cyclo

Na adrese <http://cyclo.webarchitects.cz/Cyclo.apk>, případně z přiloženého CD, si lze stáhnout instalační soubor APK s aplikací Cyclo. Stažený soubor je nutné přemístit do zařízení, na kterém má být nainstalován a poté jej stačí v daném zařízení nalézt a kliknutím na něj je spuštěna instalace aplikace Cyclo. Po úspěšné instalaci naleznete ikonku pro spuštění aplikace v základním menu daného zařízení.

Serverová část

Pro zprovoznění serverové části, sloužící pro obsluhu skupinového módu na vlastním serveru, je nutné mít k dispozici webový server, který podporuje služby HTTP dle standardů protokolu HTTP, například tedy volně dostupný Apache [40]. Webový server musí také podporovat skriptovací jazyk PHP verze 5 a vyšší. Další nutnou součástí je databázový server MySQL verze 5.1 a vyšší.

Na webový server je nutné nahrát soubor *config.ini*, ve kterém budou umístěné údaje pro připojení se k databázi a soubor *answer.php*, který slouží pro obsluhu požadavků přijatých od aplikace Cyclo a pro komunikaci s databázovým serverem. Na databázovém serveru je nutné vytvořit databázi s tabulkami pro ukládání skupinových informací a v souboru *config.ini* přepsat údaje pro připojení se k databázi. Jednotlivé tabulky s požadovanou strukturou, které jsou nutné pro ukládání skupinových informací, lze do databáze importovat pomocí souboru *DB.sql*. Všechny uvedené soubory se nacházejí na přiloženém CD v adresáři *serverova_cast..*