

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

GPS tracker



2011

Ondřej Vyhlídal

Anotace

GPS tracker je aplikace sloužící pro monitorování polohy pomocí souřadnic GPS. Práce obsahuje aplikaci pro mobilní zařízení s operačním systémem Android. Práce dále obsahuje popis, jak operační systém Android a aplikace v tomto operačním systému fungují, popis technologie GPS, postup při programování a stručnou uživatelskou příručku. Aplikace je navržena tak, aby ji byl schopen nainstalovat a používat i méně zkušený uživatel. K implementaci jsem použil programovací jazyk Java, Android SDK a Google APIs.

Děkuji Mgr. Jiřímu Zaccpalovi, Ph.D. za vedení mého bakalářského projektu, za cenné rady při konzultacích. Také bych rád poděkoval všem, kteří se podíleli na testování výsledného projektu.

Obsah

Seznam obrázků

1. Úvod

GPS tracker byl vytvořen pro potřeby uživatelů, kteří si chtějí pomocí svých chytrých telefonů vybavených operačním systémem Android a GPS čipem zaznamenávat svůj pohyb. Aplikace umožňuje zaznamenávat prošlé trasy a zobrazovat aktuální informace o trase. Dále také zobrazovat informace o aktuální pozici případně i vybranou trasu v mapě, exportovat či importovat trasy z gpx souborů. Pomocí akustické navigace se mohou uživatelé nechat navigovat po zvolené trase.

Požadavky na bakalářskou práci:

- Vytvoření programu pro sledování a zaznamenávání trasy pomocí mobilního přístroje vybaveného GPS modulem.
- Program by měl zobrazovat základní aktuální informace (polohu, rychlost, ...).
- Program by měl exportovat (importovat) naměřená data do souboru (případně na web).
- Program by měl zobrazovat informace o naměřených datech (rychlost, nadmořská výška, ...).
- Program by měl umožňovat akustickou navigaci.
- Program by měl umožňovat online a offline zobrazení trasy na mapě.

2. Teoretický úvod

2.1. Operační systémy pro mobilní zařízení

Současný trend vývoje mobilních zařízení je příznivý především pro chytré telefony. Každý telefon tohoto typu je vybaven operačním systémem, jehož účelem je se starat o veškerý chod zařízení. Použití speciálních operačních systémů v chytrých telefonech přináší spoustu výhod, jako jsou například: možnost instalace nových aplikací, lepší využití hardwaru telefonu, větší množství aplikací atd. Uživatelé běžných telefonů většinou nemají tak rozsáhlé možnosti využití svých zařízení, nicméně nespornou výhodou klasických telefonů je jejich výdrž na baterie. I ten nejlepší chytrý telefon, při běžném používání nevydrží na jedno nabití baterie více jak 3 dny. Při využívání funkcí jako jsou například Wi-fi, GPS senzor či přehrávač videí se výdrž neúměrně zkracuje. Je to daň, kterou uživatelé musí platit za to, že mají místo klasických telefonů malé počítače. S příchodem operačních systémů pro mobilní zařízení se pro vývojáře aplikací otevírají nové možnosti využití jejich znalostí v oblastech vývoje softwaru. V současné době mezi nejvýznamnější takovéto operační systémy patří:

- **iOS** – Operační systém od Apple Inc., určený nejen pro telefony, ale i pro tablety a multimediální přehrávače téže značky.
- **Symbian** – Systém vyvíjený Symbian Foundation. Symbian byl v roce 2010 uvolněn pod Eclipse Public License, čímž se z něj stala otevřená platforma. Symbian založený na Unixovém mikrojádře, je použitelný pouze pro procesory ARM. Všechny chytré telefony značky Nokia byly vybaveny tímto operačním systémem. V únoru 2011 Nokia ohlásila, že veškeré další chytré telefony budou vybaveny operačním systémem Windows Phone 7.
- **Windows Phone 7** – Je operační systém vyvinutý společností Microsoft Corporation. Tento operační systém by měl být nástupcem již nevyvíjeného Windows Mobile 6.5. Windows Phone 7 je šířen pod proprietární licenci, snaží se konkurovat operačním systémům jako jsou iOS či Android. V současné době se zařízení s Windows Phone 7 netěší velké oblibě u zákazníků. Tento systém zcela podporuje multi-touch technologie, mezi jeho hlavní výhodu patří možnost napojení se na Xbox pomocí technologie XBox live, což umožňuje hraní her na hostitelském zařízení.
- **Android** – Otevřená platforma, která je navržena pro mobilní zařízení. Jeho architektura mu umožňuje být hardwarově nezávislý (může běžet na různých zařízeních s rozdílnými procesory) [?] [?] [?] [?].

2.2. Vývoj aplikací pro mobilní zařízení

Vývoj aplikací pro mobilní telefony je odvětví, jenž se poslední dobou velice dynamicky rozvíjí. Rozvoj úzce souvisí s nárůstem prodeje chytrých telefonů, díky jejich dostupnosti pro běžné uživatele. Na aplikace vyvíjené pro mobilní operační systémy jsou kladené mnohem vyšší nároky, než na aplikace vyvíjené pro osobní počítače. Vývojáři aplikací musí mít na mysli, že nemají k dispozici takové množství výpočetního výkonu. Jsou tedy limitováni následujícími prvky:

- nižší takt procesoru
- omezené množství operační paměti
- menší velikost datových úložišť
- malé rozlišení displejů
- vyšší ceny datových přenosů
- pomalé přenosy dat s velkou odezvou
- nespolehlivá datová připojení
- omezené kapacity baterií

I přes to, že nová generace mobilních zařízení má jistá hardwarová vylepšení, stále se nemohou rovnat s hardwarem osobních počítačů. Výrobci mobilních zařízení, především mobilních telefonů, upřednostňují malé rozměry a dlouhou výdrž před zvyšováním výpočetního výkonu telefonů. Pro vývojáře to znamená ztrátu náskoku, který by jim zaručoval Moorův zákon (každé dva roky zdvojnásobení počtu tranzistorů v integrovaných obvodech). V praxi to znamená, že by se vývojáři měli soustředit i na optimalizace kódu tak, aby běžel svižně, protože předpokládat takové nárůsty výkonu jako u osobních počítačů je nepravděpodobné.

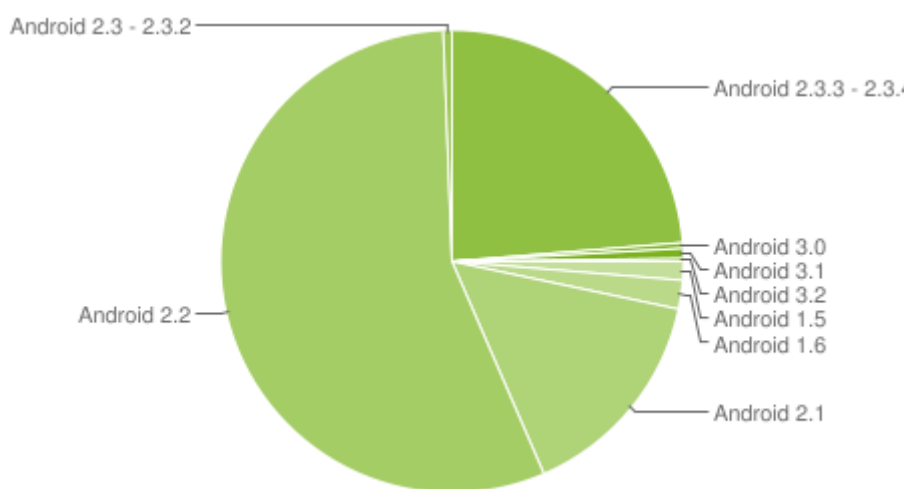
2.3. Operační systém Android

Slovy Andyho Rubina `Android is the first truly open and comprehensive platform for mobile devices`. volně přeloženo: Android je první, skutečně komplexní a otevřená platforma pro mobilní zařízení. [?]

Historie

Společnost Android Inc. byla založena v roce 2003, v roce 2005 byla koupena společností Google Inc. Od roku 2005 do 2007 tým pod vedením Andyho Rubina pracoval na linuxovém jádře, na jehož základě Google získal několik patentů v oblasti mobilních technologií. Koncem roku 2007 bylo utvořeno konzorcium Open Handset Alliance, jež sdružovalo společnosti zabývající se vývojem

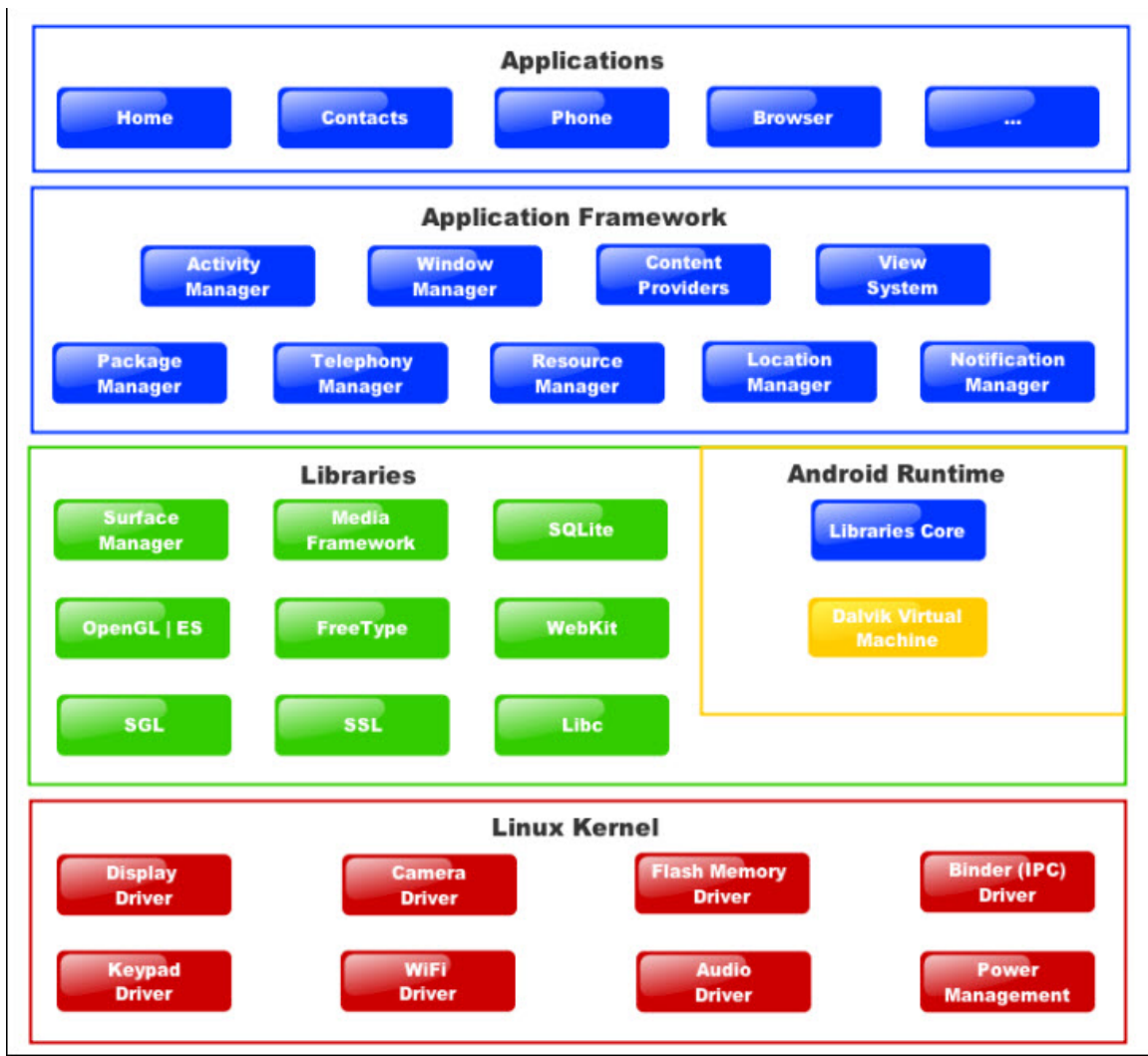
mobilních telefonů, jak po softwarové stránce, tak i po hardwarové stránce. Jedním z cílů konzorcia bylo inovovat stávající technologie a lépe reagovat na potřeby uživatelů. Klíčem k tomuto měla být otevřená platforma Android. Rok 2008 byl pro Android zlomový, protože, v září ohlásil T-Mobile první mobilní telefon s tímto operačním systémem – T-Mobile G1, ke konci roku Google uvolnil zdrojové kódy pro platformu Android pod Apache open source licencí a také stihl zveřejnit novou verzi 1.1 v níž opravoval některé zásadní chyby předešlé verze. V září následujícího roku byla vydána verze 1.6 a o měsíc později verze 2.0, což mělo za následek velký nárůst mobilních zařízení opatřených tímto operačním systémem. Ve verzi 2.0 byly představeny inovace, jako například: text-to-speech, multitouch, gesta, podpora HTML 5 a další. V současnosti je aktuální verze 2.3.4 určená především pro mobilní telefony a verze 3.1 používaná v tabletech. Nejvíce uživatelů používá verzi 2.2. Larry Page (spoluzakladatel společnosti Google). Ten na svém Google+ profilu zveřejnil zajímavá čísla: Android je prodáván společně se 400 různých zařízení od 39 výrobců ve 123 zemích. Celkový počet zařízení je 130 000 000, přičemž každý den přibude přibližně 550 000 nových. Android market obsahuje přes 250 000 aplikací, celkový počet stáhnutí všech aplikací překračuje 6 000 000 000 [?] [?] [?] [?].



Obrázek 1. Poměrné zastoupení různých verzí operačního systému Android [?].

Operační systém Android má pevně danou architekturu. V nejvyšší vrstvě jsou aplikace, pod ní se nachází aplikační framework, další vrstva je tvořená knihovnamy, jejichž součástí je i Android runtime, nejnižší vrstvu tvoří Linuxové jádro. Podrobná struktura viz obrázek 2.

- **Aplikační vrstva** – Zahrnuje všechny aplikace (jak nativní, tak i aplikace třetích stran).



Obrázek 2. Struktura operačního systému Android [?].

- **Aplikační framework** – Poskytuje soubor hlavních knihoven využívaných při tvorbě aplikací a vytváří abstraktní vrstvu pro přístup k hardwaru zařízení.
- **Android runtime** – Je soubor klíčových knihoven a Dalvik Virtual Machine, tyto dvě části společně formují základ pro aplikační framework. Dalvik navrhnul a napsal Dan Bornstein, je to v podstatě Java Virtual Machine pouze s několika rozdíly. Je optimalizovaný pro běh v mobilních zařízeních a umožňuje plynulý běh více instancí najednou na jednom zařízení. Veškerý přístup k systémovým službám a hardwaru je realizován prostřednictvím Dalvika.
- **Knihovny** – Je soubor C/C++ knihoven, obsahující knihovny jako jsou

například knihovny pro přehrávání médií, grafické knihovny pro 2D a 3D grafiku, SQLite knihovny pro nativní podporu databází a další.

- **Linuxové jádro** – Využívá se pro správu paměti a dalších komponent systému. Obsahuje také ovladače pro veškerý hardware daného zařízení.

[?][?]

2.3.1. Architektura aplikací pro operační systém Android

Architektura aplikací pro operační systém android podporuje koncept znovupoužitelnosti komponent. Tento mechanismus umožňuje vytvořit například nové uživatelské rozhraní pro standardní vytáčecí obrazovku nebo využít komponenty již nainstalované aplikace. Následující seznam služeb představuje základní kameny pro vývoj aplikací pro Android.

- **Aktivity Manager** – Řídí životní cyklus jednotlivých aktivit.
- **Pohled** – Používá se pro vytváření grafického uživatelského rozhraní pro aktivity.
- **Správce notifikací** – Poskytuje ucelený mechanismus pro zobrazování notifikací uživateli.
- **Poskytovatel Obsahu** – Umožňuje aplikacím sdílet data.
- **Správce zdrojů** – Podporuje "ne-kódové" zdroje, jako jsou například obrázky či předdefinované texty.

2.3.2. Standardní typy aplikací pro Android

Většina aplikací vytvořených pro operační systém Android bude spadat do jedné z následujících kategorií:

- **Aplikace běžící v popředí** – Aplikace, které jsou aktivní pouze když jsou zobrazeny uživateli, nevykonávají žádnou práci pokud nejsou aktivní. Typickým příkladem těchto aplikací mohou být hry.
- **Aplikace běžící v pozadí** – Aplikace, které většinu svého aktivního času tráví v pozadí, neočekávají od uživatele téměř žádnou interakci. Typickým příkladem takovéto aplikace je automatické rozesílání odpovědí na příchozí SMS zprávy.

- **Aplikace kombinující oba předchozí přístupy** – Aplikace očekává od uživatele interakci, ale provádí většinu své práce v pozadí a jen občasně informuje uživatele. Typickým příkladem jsou multimediální přehrávače.
- **Widget** – Aplikace většinou neprovádí složité výpočty, hlavním účelem je zpříjemnit uživateli práci s mobilním zařízením.

2.4. Popis technologie GPS

2.4.1. Co je to GPS

GPS (Global Positioning System) je síť družic obíhajících kolem Země po přesně určených drahách. Globální polohový systém provozuje a udržuje ministerstvo obrany USA. Je tvořen 31 družicemi (září 2010), obíhajícími povrch země ve výšce 20 000 km. Celý tento systém umožňuje komukoli, kdo je vybaven GPS-přijímačem, určit svoji polohu s přesností do deseti metrů. Pomocí dalších zpřesňujících metod se můžeme dostat na přesnost v řádech desítek centimetrů. Kromě již zmíněného GPS existují i jiné alternativy, například v podobě Ruského GLONASS, nebo evropského komerčního projektu GALILEO. [?] [?] [?]

2.4.2. Jak GPS funguje

System GPS se skládá ze tří základních segmentů – uživatelského, kosmického a řídicího.

1. **Uživatelský segment** představují koncová zařízení uživatelů vybavené anténami pro příjem gps signálu a procesory pro zpracování přijatých signálů. Přijímače vypočítávají polohu antény, nadmořskou výšku a získávají přesné datum i čas.
2. **Kosmický segment** je tvořen soustavou až 32 družic (projektovaný je přitom na 24 družic). Družice jsou umístěny po 5 až 6 na 6 kruhových drahách. Družice provede jeden oběh okolo Země za 11 hodin a 58 minut. Každá družice má svůj unikátní identifikátor a vysílá jiný kód, na základě těchto údajů je přijímač schopen přesně určit každou družici. Družice jsou několikrát do roka plánovaně odstavovány kvůli údržbě a seřizování atomových hodin.
3. **Řídicí segment** tento segment plní řadu úloh:
 - monitoruje signály družic kosmického systému
 - vyhodnocuje chování družic na oběžných drahách
 - určuje parametry oběžných drah
 - určování korekčních parametrů pro hodiny a oběh
 - sledování a vyhodnocení stavu družic
 - vysílá aktualizované parametry pro družice
 - udržuje družice
 - řídí celý systém

V tomto segmentu rozlišujeme tři druhy stanic:

- **Monitorovací stanice** – Jsou strategicky rozmístěny, aby umožňovaly sledování co největšího počtu družic.
- **Hlavní řídicí stanice** – Ke každé hlavní stanici je i záložní, zpracovává signály z monitorovacích stanic, provádí modelování chování kosmického segmentu, určování parametrů oběžných drah a korekčních parametrů hodin na družicích.
- **Stanice pro komunikaci s družicemi** – Totožné s monitorovacími stanicemi, přenáší nově určené korekční parametry oběžných drah a atomových hodin a také slouží pro ovládání družic.

2.4.3. Metody určování polohy z GPS

Přijímač určuje svoji vzdálenost k několika družicím a svoji polohu potom stanovuje protínáním vzdáleností. K určování vzdálenosti od družic se využívají následující tři způsoby:

- **Kódové měření** – Základním principem je určení vzdáleností mezi přijímačem a družicemi, využívají se dálkoměrné kódy vysílané jednotlivými družicemi. Přijímač přijme dálkoměrný kód příslušné družice, zjistí čas odesílání a přijetí jedné sekvence kódu, ze zjištěného časového rozdílu určuje aktuální pozici.
- **Fázové měření** – Jsou založena na využití nosných vln vyslaných družicemi. Pokud spočítáme počet celých vln, jejichž délku známe, tak dopočtení poslední ne celé vlny je snadné. Toto měření ve srovnání s kódovým způsobem měření je mnohem přesnější, ale také poměrně pomalejší díky složitému a zdlouhavému zjištění počtu celých vln před počátkem měření.
- **Dopplerovská měření** – Založena na Dopplerově jevu. Dopplerův jev popisuje změnu frekvence a vlnové délky přijímaného signálu vůči vysílanému signálu, pokud se vysílač nebo přijímač pohybuje. Dopplerova jevu se využívá například i v měření rychlosti vozidel nebo v lékařských sonografech.

V běžných přijímačích se většinou používá jen metoda kódového měření. [?] [?]

2.5. Srovnání vybraných GPS trackerů

Náhodným výběrem jsem pro srovnání kvalit různých GPS trackerů v Android Marketu vybral trackery z prvních pozic a tyto jsem následně srovnával z několika úhlů. Jedná se o následující GPS:

2.5.1. GPS tracker

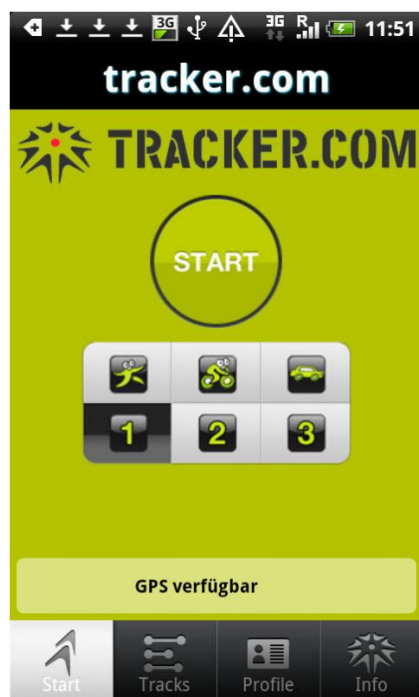
Aplikace s názvem GPS tracker zobrazovala pouze adresu v mapě, dále nabízel možnost poslat svoji aktuální pozici přes sms, emailem, do mapy či přímo na facebook. Aplikaci jsem zkoušel v offline módu i online módu, ale bohužel GPS tracker nebyl schopen si uchovat nakešované mapy, dále nenabízel možnost export do souboru. [?]

2.5.2. Mobile GPS tracker

Mobile GPS tracker měl velice jednoduché a přehledné rozhraní. Umožňoval standardní nastavení, které bylo velice kvalitně zpracováno. Neustále vypínal a zapínal GPS senzor. Mobile GPS tracker uměl zobrazit polohu na mapě, ale bohužel ukazoval jen přibližnou oblast výskytu, aktuální pozice nebyla v mapě nijak zvýrazněna. Procházená trasa nešla uložit do souboru. Po ukončení aplikace standardním způsobem aplikace stále běžela na pozadí a neustále vypínala a zapínala GPS senzor. [?]

2.5.3. Tracker.ch

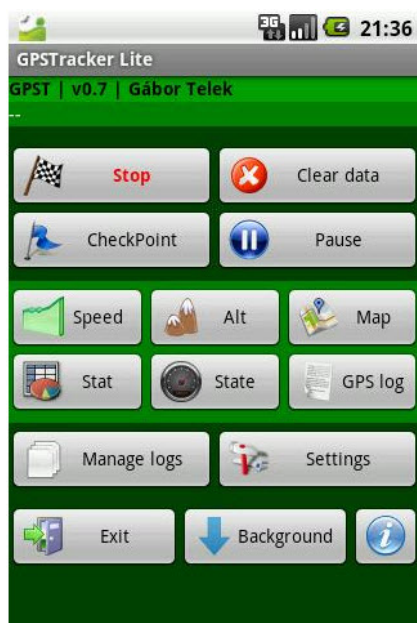
U této aplikace bylo uvedeno, že je zdarma a není potřeba žádná dodatečná registrace. Avšak po nainstalování aplikace vyžadovala registraci a i přes to bohužel běžela v bezplatné fázi pouze v omezeném módu. [?]



Obrázek 3. Hlavní obrazovka Tracker.ch. [?].

2.5.4. GPS Tracker Lite

Dle mého názoru velice dobrá aplikace, mající všechny standardní funkce trackeru a navíc nabízela možnost uploadu vybrané trasy na web. Dále nabízela při ukládání souboru zvolit, zda uživatel šel pěšky, jel na bruslích, kole či autem. Přehledně zobrazovala naměřené informace. Chyba aplikace se projevila při testování, pokud chceme nechat vykreslit graf rychlosti. Během vykreslování grafu se mohou zobrazit nesmyslné údaje, tak jako v mém případě. GPS Tracker Lite nabízel export trasy do gpx nebo textového souboru. [?]



Obrázek 4. Hlavní obrazovka GPS Tracker Lite. [?].

2.5.5. GPS tracker - Iplussoft

Tento GPS tracker měl plynulé nastavení intervalu aktualizace, umožňoval pouze nastavení časového intervalu pro aktualizaci souřadnic. Po nainstalování nebylo možné tuto aplikaci používat, bylo nutné se nejdříve zaregistrovat na stránkách výrobce. bohužel i přes opakované pokusy registrovat se, se nezdařilo dokončit registraci ani jednou. Další chybu spatřuji v nemožnosti ukončit aplikaci po jejím zavření, stále běží v systému a nebylo možné ji žádným způsobem ukončit. [?]



Obrázek 5. Hlavní obrazovka trackeru. [?].

2.5.6. GPS essentials

Po spuštění aplikace nabízí uživateli možnost nastavit si do dvanácti polí, jaké se mu mají zobrazovat informace. Aplikace dále nabízí standardní nastavení aktualizace souřadnic. Je zde i možnost vynutit si zapnutí podsvícení displeje, dále možnost hybridního zobrazení (část displeje mapa a část aktuální informace). Považuji za vhodné upozornit na nevhodnost takového zobrazení u mobilních zařízení s malým displejem. [?]



Obrázek 6. Hlavní obrazovka trackeru. [?].

2.5.7. Live view GPS

Po instalaci je vyžadována dodatečná registrace a získání aktivačního klíče. Celá tato operace byla velmi obtěžující. Aplikace fungovala pouze jako sledovač daného zařízení a nebylo možné žádným způsobem uložit pozici či trasu. [?]

LIVEVIEWGPS
TRACKING YOU CAN TRUST

User Id

Password

Activation Key

Login

Remember me

Sales/Support: 661.294.6805
29021 Ave Sherman STE 103
Valencia, CA 91355
www.LiveViewGPS.com

Obrázek 7. Obrazovka přihlášení [?].

2.5.8. Tracked lite

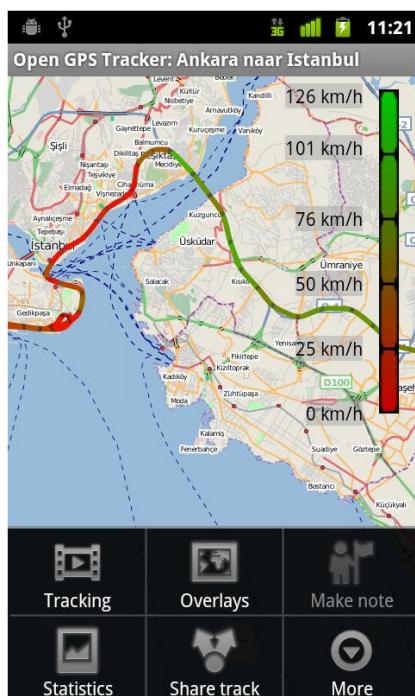
Tracked lite po spuštění přestala reagovat na veškeré uživatelské vstupy. Zkoušel jsem několikrát reinstall aplikace, nicméně ani toto řešení nebylo úspěšné. [?]



Obrázek 8. Obrazovka s aktuálními informacemi [?].

2.5.9. Open GPS tracker

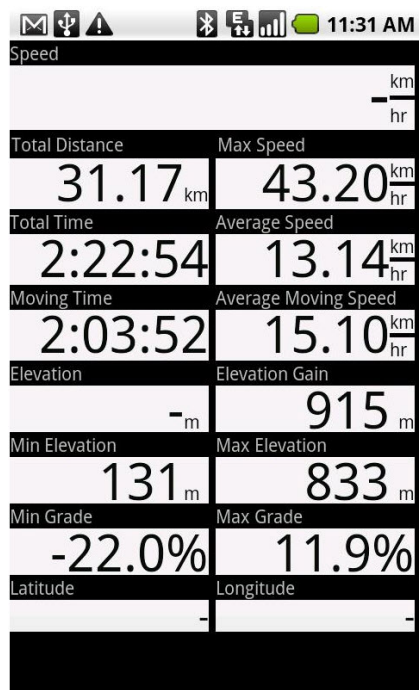
Velice přehledně zpracované statistiky, široká nabídka exportu souřadnic do souborů gpx, kml a textových souborů. Aplikace umožňovala odeslat svoji pozici či trasu přes email, avšak chybně zobrazovala trasu na mapě, pokud byl záznam pozastaven. Velice hlasité oznámení ztráty signálu GPS senzoru. Vygenerované statistiky byly hůře čitelné. [?]



Obrázek 9. Přehledová obrazovka [?].

2.5.10. My tracks

Jako jediná aplikace obsahovala rozsáhlou integrovanou nápovědu. Velmi dobře zpracované uživatelské rozhraní, jež bylo jednoduché na ovládání, byla zde možnost připojit i externí senzor na snímání tlukotu srdce. Trasy bylo možno exportovat do gpx, csv, kml a poslat je emailem či exportovat přes google account. [?]



Speed	
	km/hr
Total Distance	Max Speed
31.17 km	43.20 km/hr
Total Time	Average Speed
2:22:54	13.14 km/hr
Moving Time	Average Moving Speed
2:03:52	15.10 km/hr
Elevation	Elevation Gain
-m	915 m
Min Elevation	Max Elevation
131 m	833 m
Min Grade	Max Grade
-22.0%	11.9%
Latitude	Longitude
-	-

Obrázek 10. Přehledová obrazovka [?].

2.5.11. Souhrn

Jako nejlepší aplikace se mi jevil *Open GPS tracker*, jelikož z pohledu uživatele byl navržen opravdu intuitivně. I přesto, že jsem nečetl uživatelskou příručku, tak mi nečinilo žádné problémy tuto aplikaci plnohodnotně ovládat. Jako nejméně vhodnou aplikaci bych volil *Tracked Lite*, protože aplikace nešla nainstalovat i na zařízení, které mělo vyhovovat požadavkům. Výzkum jsem prováděl v průběhu měsíce února 2011, je pravděpodobné, že autoři zmíněných aplikací již vydali novější verze, ve kterých odstranili výše zmíněné nedostatky.

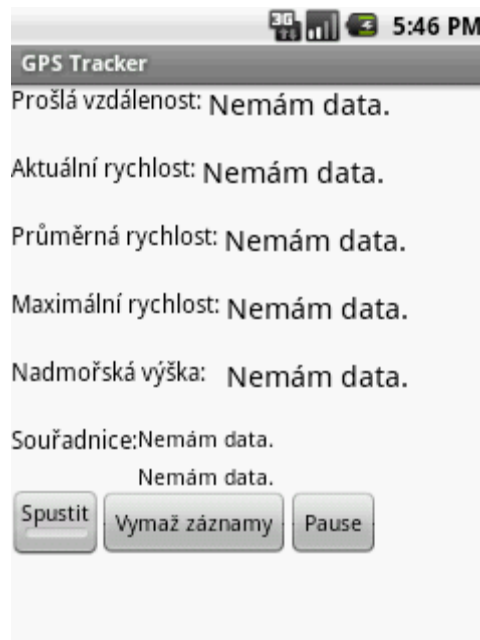
3. Uživatelská příručka

3.1. Požadavky, instalace a spuštění aplikace

Aplikace GPS tracker je určena pro mobilní zařízení s operačním systémem Android, která jsou vybavena GPS čipem. Pro plynulý běh aplikace je nutná minimální verze operačního systému Android 2.1 Eclair. Dále je nutné mít možnost instalovat aplikace pomocí instalačního souboru z paměťové karty. Tento požadavek bude odstraněn, jakmile se mi podaří umístit aplikaci na oficiální Android Market. Instalace probíhá spuštěním instalačního souboru `GPSTracker.apk`. Po nainstalování se přidá ikona s názvem GPS Tracker do hlavní nabídky. Spuštění aplikace probíhá výhradně přes tuto ikonu.

3.2. Hlavní obrazovka

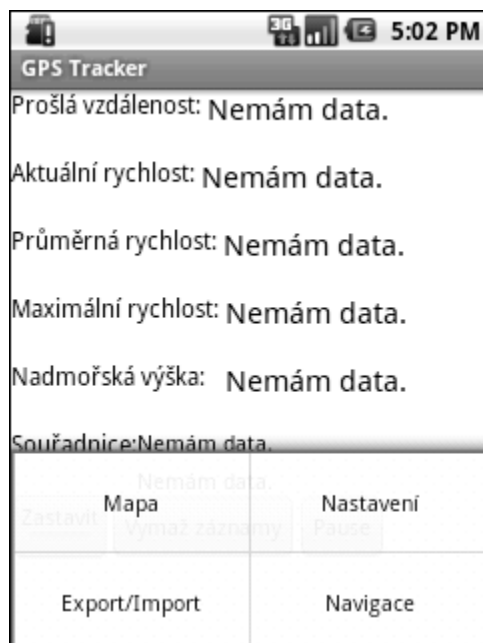
Po spuštění aplikace se zobrazí hlavní obrazovka. Zde jsou uživateli zobrazována veškerá aktuální data týkající se jeho pozice či procházené trasy. Konkrétně se jedná se o informace ohledně prošlé vzdálenosti, aktuální rychlosti, průměrné rychlosti, maximální rychlosti, nadmořské výšce a gps souřadnicích. Dále zde nalezneme několik nezbytných tlačítek pro ovládání aplikace. První z nich je zapínací tlačítko `Spustit` sloužící pro zapínání, vypínání záznamu trasy a signalizaci, zda je záznam spuštěn či vypnut. Pokud není zaplý GPS senzor a uživatel se pokusí o zapnutí záznamu objeví se hláška o možnosti zapnutí GPS senzoru, která uživatele přesměruje do nastavení, kde může zapnout gps senzor. Vpravo od `Spustit` je umístěno tlačítko `Vymaž záznamy`, jež slouží na vymazání všech uložených záznamů v databázi. Posledním tlačítkem je `Pauza`, kterou je možné použít pouze pokud probíhá zaznamenávání trasy a umožňuje pozastavit a znova spustit zaznamenávání trasy.



Obrázek 11. Hlavní obrazovka aplikace.

3.3. Menu

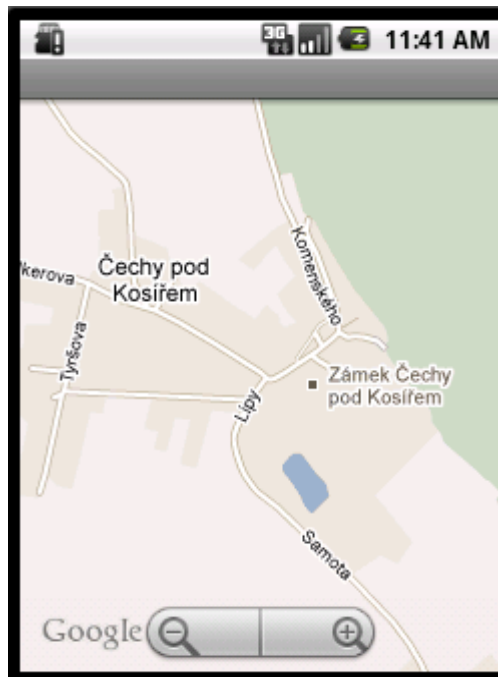
Po stisknutí hardwarového tlačítka menu na zařízení se vyvolá nabídka se seznamem obrazovek, do nichž se uživatel může přepnout. Pro návrat ze zvolené obrazovky je vhodné použít hardwarové tlačítko zařízení zpět. Těmito tlačítky jsou vybavena všechna zařízení s operačním systémem Android.



Obrázek 12. Menu aplikace.

3.4. Mapa

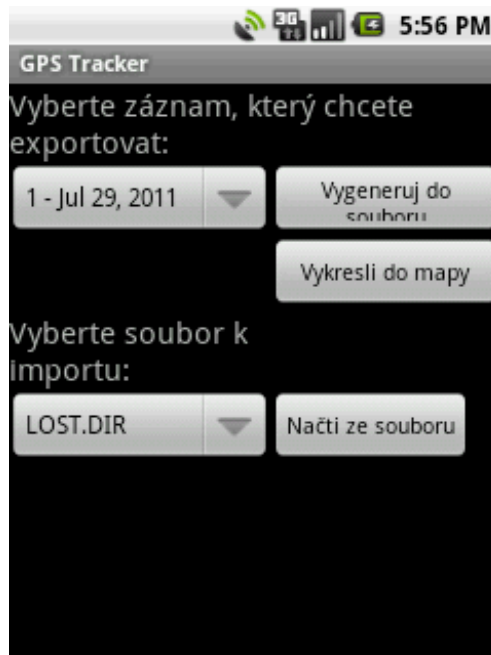
Při zobrazení mapy se zobrazí mapové podklady Google Maps. V případě, že je spuštěný záznam zobrazí se navíc i všechny body již prošlé trasy pomocí malých zelených bodů. Mapu lze pouze přibližovat, oddalovat, posouvat, není možné vyhledávání adres a podobně. Tlačítka pro ovládání mapy se zobrazí v okamžiku, kdy se dotknete displeje. Jsou umístěna ve spodní části obrazovky.



Obrázek 13. Mapa.

3.5. Export, import

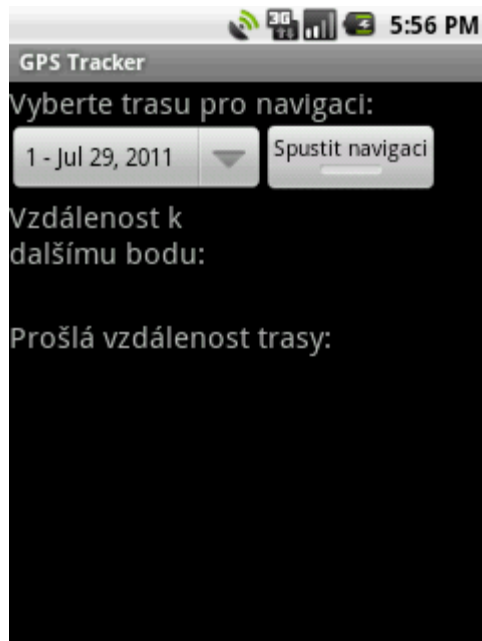
Obrazovka exportu je rozdělena na dva celky. Horní část slouží pro export tras. Nachází se zde rozbalovací seznam, z něhož můžeme vybrat trasu, kterou chceme exportovat do souboru gpx. Dále jsou zde umístěna dvě tlačítka, první slouží pro export trasy do gpx souboru a druhé vykreslí zvolenou trasu do mapy. Dolní část okna je určena pro import tras uložených v souborech gpx v kořenovém adresáři paměťové karty zařízení. Po rozkliknutí rozbalovací nabídky se zobrazí seznam všech souborů. Kliknutím na název vybereme soubor a následným stisknutím tlačítka **Načti ze souboru** se provede načtení trasy do interní databáze aplikace.



Obrázek 14. Export a import tras.

3.6. Navigace

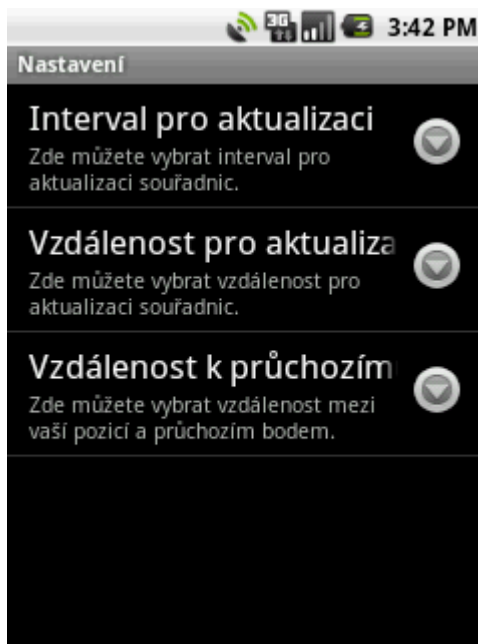
Navigace obsahuje rozbalovací seznam tras, po kterých se můžeme nechat navigovat pomocí akustické navigace. Vybereme název trasy a navigaci spustíme tlačítkem **Spustit navigaci**. Po stisknutí tlačítka signalizuje, zda je navigace aktivní. Dále se zde zobrazují důležité informace pro uživatele: vzdálenost k dalšímu bodu trasy, či jakou již urazil část cesty. Navigace vydává tři různé tóny pro určení správného směru k dalšímu bodu. První tón je vyšší a určuje, že jdeme správným směrem, pro signalizaci, naopak hluboký tón signalizuje špatný směr. Poslední tón je určen pro signalizaci úspěšného průchodu přes bod.



Obrázek 15. Akustická navigace.

3.7. Nastavení

V nastavení uživatel nalezne veškeré nastavení aplikace, především možnost nastavit intervaly pro aktualizaci GPS pozice. Jsou zde tři možnosti, první nastavení minimální vzdálenosti mezi aktualizací GPS pozice a druhé je možnost nastavit minimální čas pro aktualizaci GPS pozice. Třetí slouží pro nastavení přesnosti akustické navigace.



Obrázek 16. Nastavení aplikace.

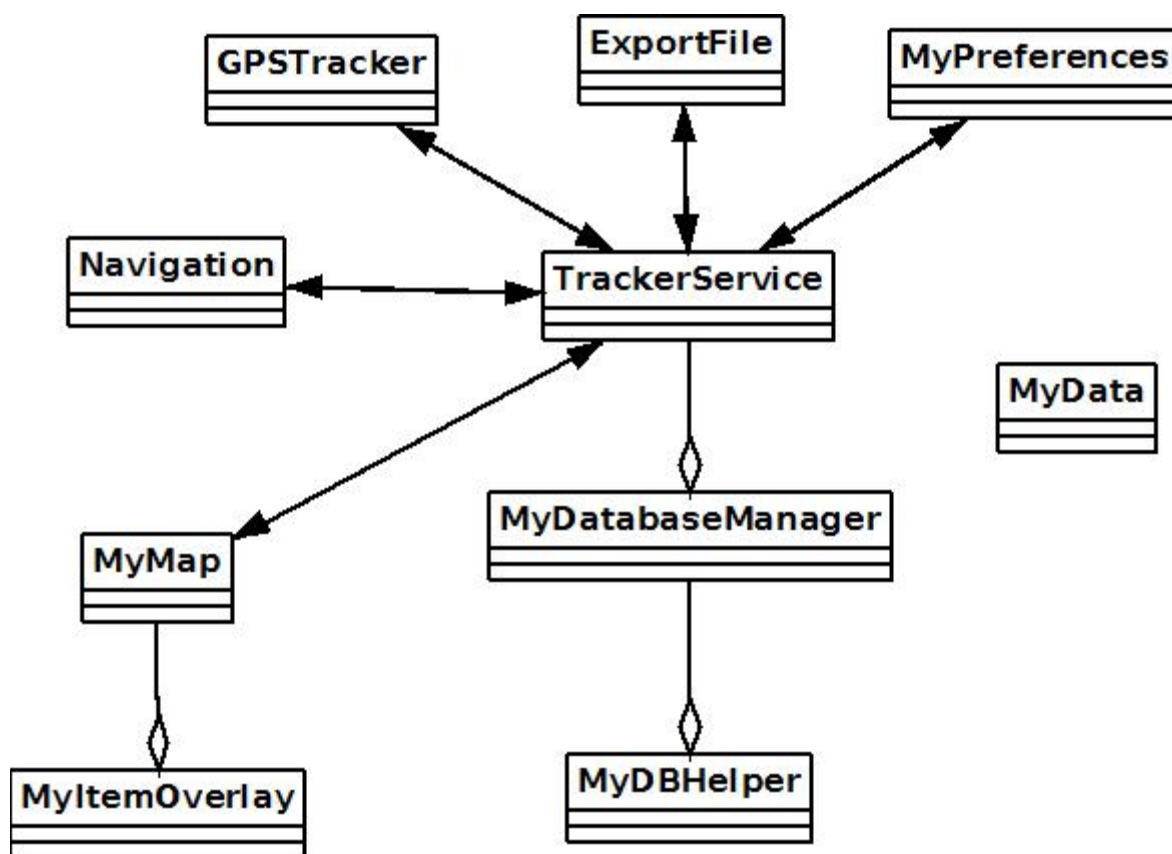
3.8. Vytvoření záznamu a následný export

V první řadě spustíme aplikaci. Nyní se zobrazí hlavní okno aplikace, obsahující základní ovládací prvky. Pro spuštění záznamu stačí stisknout tlačítko **Spustit**. Pokud aplikace zjistí, že GPS čip je vypnutý vyvolá kontextové okno s otázkou ohledně zapnutí GPS čipu. Po kladné odpovědi se otevře nastavení zařízení, zde je nutné povolit příjem GPS signálu. Stisknutím klávesy zpět se opět zobrazí aplikace. Nyní stačí opět stisknout tlačítko **Spustit**. Do okamžiku, než GPS čip zachytí dostatečný signál ze satelitů, se zobrazuje ve všech textových polích hláška **Nemám data**. Jakmile dojde k zachycení signálu, hláška je nahrazena aktuálními hodnotami. Nyní můžeme vyrazit na cestu a nechat zařízení sledovat náš pohyb. Pokud chceme během cesty zastavit a pozastavit tak záznam stačí stisknout tlačítko **Pause**. Opětovným stisknutím tlačítka obnovíme záznam aktuální trasy. Pro dokončení záznamu stiskneme tlačítko **Zastavit**. Pokud nyní chceme exportovat trasu do gpx souboru, či mapy, stiskneme tlačítko menu a zvolíme možnost **Export/import**. Zobrazí se možnosti exportu či importu tras. Z rozbalovací nabídky zvolíme trasu pro export a nyní stačí kliknout na **Vygeneruj do souboru** či na **Vykresli do mapy**. Vygenerovaný gpx soubor se uloží do kořenového adresáře na paměťové kartě. Pokud v okamžiku ukládání není paměťová karta v zařízení, zobrazí se hláška o nezdařeném uložení souboru.

4. Programátorská příručka

4.1. Architektura GPS trackeru

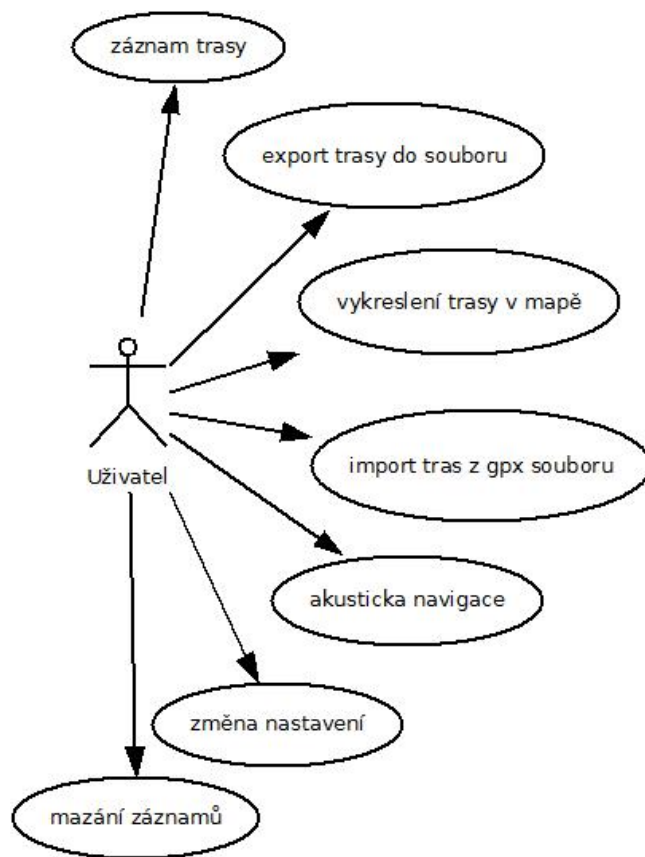
Při návrhu aplikace jsem si položil otázku, jak bych chtěl, aby aplikace fungovala. Zkoumal jsem několik možných řešení, až jsem nakonec dospěl k finálnímu návrhu, viz obrázek 9. Celá aplikace je rozdělena do několika částí. Nejdůležitější částí je služba běžící na pozadí po celou dobu spuštění aplikace. Zmíněná služba se stará o zaznamenávání tras, aktualizaci GPS souřadnic a komunikuje s ostatními komponentami aplikace. Pro ukládání tras jsem zvolil SQLite databázi. Operační systém má nativní podporu těchto databází. Pro zobrazování tras jsem použil Google Maps. Google poskytuje API knihovny přímo pro Android, díky tomuto se řešení založené na Google Maps přímo nabízí. Google Maps nabízí i možnost jistého kešování map.



Obrázek 17. Diagram tříd aplikace.

4.1.1. Příklad spuštění aplikace a odchyčení souřadnic

Na tomto příkladu bych chtěl demonstrovat princip fungování aplikace. Po spuštění aplikace se zobrazí hlavní okno aplikace, současně proběhne broadcast o spuštění aplikace, jež spustí inicializaci služby běžící na pozadí. Když uživatel klikne na tlačítko `Spustit` provede se kontrola, zda je GPS čip zapnut, v případě negativní odpovědi dostane uživatel možnost jej zapnout. V případě, že je GPS zapnut aktivita `GPSTracker` vyšle broadcast o tom, že se kliklo na tlačítko `start` a má se začít zaznamenávat. Tento broadcast je odeslán do systému, ten následně vybere nejvhodnější broadcast receiver pro zpracování. Aplikace je navržena tak, aby pro zpracování broadcastu byl vždy vybrán broadcast receiver, který je součástí služby běžící na pozadí. Služba odchytil broadcast, dekoduje jej a na základě předaných informací postupuje dále. Následně čeká na aktualizace souřadnic. Jakmile se objeví nové GPS souřadnice, služba provede jejich uložení do databáze a odešle souřadnice broadcastem do systému. Systém opět vybere nejvhodnější receiver pro příjem tohoto broadcastu. V aplikaci jsou dvě aktivity, obsahující vhodný receiver pro zpracování, systém vybere výhradně tu aktivitu zobrazenou uživateli. Receiver obsažený v této aktivitě provede dekodování a podle předaných informací pokračuje v práci. Daný postup se využívá dokud uživatel nezastaví, nebo nepozastaví záznam trasy. V případě, že se tak stane, vyšle se opět broadcast, informující službu o změně stavu. Služba tento broadcast přijme, dekoduje a přizpůsobí své chování. Na podobném principu je založena celá aplikace.



Obrázek 18. Diagram případu užití aplikace.

4.2. Popis tříd

4.2.1. Třída TrackerService

Třída, jež je potomkem třídy `Service` a zároveň implementuje `LocationListener`. To vše zajišťuje, že instance třídy bude schopná běžet na pozadí a zachytávat změny GPS souřadnic.

Metody

- `private void initilaizeAll()` – Uvedení všech prvků servisu do výchozího stavu.
- `private void loadFromSharedPrefs()` – Načtení posledního stavu instance.
- `private void updateUpdatePreferences()` – Je volána vždy při přechodu instance do pozadí.
- `public void onCreate()` – Volána vždy, když je instance prvně spuštěna, obsahuje inicializaci prvků.
- `public void onDestroy()` – Volána vždy, když je instance ukončena.
- `public void onStart(Intent intent, int startid)` – Základní metoda, slouží pro spuštění služby.
- `private void stopLocService()` – Ukončí běh služby.
- `private void startLocationService()` – Spustí odchyťávání změn GPS souřadnic.
- `public void enableRecord(int updateTime)` – Spustí odchyťávání souřadnic v určených časových intervalech.
- `public void enableLocBroadcastToNav(int updateTime)` – Zavoláním této metody se nastaví, že odesílané souřadnice bude zpracovávat aktivita navigace.
- `public void pauseRecord()` – Pozastaví záznam.
- `public void resumeRecord(int updateTime)` – Povolí záznam.
- `public void disableRecord()` – Vypne záznam.
- `public void setUpdateTime(int t)` – Nastaví interval pro aktualizace souřadnic.

- `private void removeAllRecords()` – Vymaže všechny záznamy databáze,
- `public ArrayList<MyData> getAllRecords()` – Vrábí seznam všech datových struktur, které jsou naplněny záznamy databáze.
- `public ArrayList<Object> getRecord(int pos)` – Vrací jeden konkrétní záznam z pozice.
- `public ArrayList<MyData> getRecordsById(int id)` – Vrací seznam všech záznamu daného id.
- `private ArrayList<Integer> getAllRecordsId()` – Vrací seznam všech id.
- `private ArrayList<String> getAllRecordsIdAndDate()` – Vrací seznam všech id a datumů.
- `private void addRecordToDb(...)` – Přidá záznam do databáze.
- `private void addRecordsFromFileToDb(ArrayList<MyData> newRecords)` – Přidá záznamy ze souboru do databáze.
- `private void initializeReceiver()` – Inicializuje broadcast receiver, aby služba mohla přijímat informace od ostatních komponent aplikace.
- `private void registerAllReceivers()` – Zaregistruje všechny potřebné receivers pro službu.
- `private void stopReceivingIntents()` – Odebere všechny zaregistrované přijímače.
- `public void onLocationChanged(Location location)` – Zpracovává každou aktualizaci souřadnic.
- `public void onStatusChanged(String arg0, int i, Bundle arg2)` – Podle statusu GPS čipu zobrazuje informace.

4.2.2. Třída GPSTracker

Třída `GPSTracker` je potomkem aktivity. Hlavním účelem této aktivity je zobrazovat informace o trase uživateli. Přijímá data od služby, která je potom zpracovává. Například: vypočítává průměrnou rychlost, či počítá prošlou vzdálenost.

Metody

- `protected void onCreate(Bundle savedInstanceState)` – Volána vždy, když je aktivita prvně spuštěna, obsahuje inicializaci prvků
- `private void enableReceiver()` – Zaregistrování instance třídy `BroadcastReceiver` pro příjem komunikace od ostatních komponent aplikace.
- `private void disableReceiver()` – Odregistrování instance třídy `BroadcastReceiver` pro veškerou komunikaci od komponent aplikace.
- `public void onStart()` – Metoda volána vždy při přesunu aktivity do popředí.
- `public void onResume()` – Metoda volána vždy při přesunu aktivity do popředí, obsahuje všechny nezbytné inicializace prvků a komponent.
- `public void onDestroy()` – Metoda volána vždy před vypnutím aktivity
- `public void onPause()` – Volána vždy při přechodu aktivity do pozadí, jsou zde všechny metody nezbytné pro bezpečné uložení stavu aktivity.
- `private void initializeGui()` – Inicializuje veškeré grafické prvky včetně metod pro jejich obsluhu.
- `public boolean onCreateOptionsMenu(Menu menu)` – Vytvoří menu.
- `public boolean onOptionsItemSelected(MenuItem item)` – Obsluha kliknutí na položku v menu.
- `private void initializeReceiver()` – Vytvoří instanci třídy `BroadcastReceiver` a nastaví všechny požadované filtry pro příjem komunikace.
- `public void startMyService()` – Vyšle komunikaci s žádostí o zapnutí služby.
- `private void stopMyService()` – Vyšle komunikaci s žádostí o vypnutí služby.
- `private void removeValuesAfterEnd()` – Nastavení nulových hodnot do proměnných pro uchování dat o trase.
- `private void launchGPSOptions()` – Vyvolá nabídku zapnutí GPS senzoru.
- `private void makeBroadcastToService(String action, int updateInterval)` – Vyšle komunikaci pro službu.

- `private void updateGuiIfRecord(boolean isRecording, double lat, double lon, double alt, float speed, long time, float bearing)` – Při příjmu souřadnic aktualizuje veškerá data aktuální trasy a zobrazí je uživateli.
- `private double calculateAvgSpeed(long time, double distance)` – Vypočítá průměrnou rychlost.
- `private double calculateDistanceHaversine(float lata, float lnga, float latb, float lngb)` – Použije Haversinovu formuli pro výpočet vzdáleností mezi 2 body.
- `private void buildAlertMessageNoGps()` – Vytvoří hlášku pro případ, že není zaplý GPS senzor.
- `private void updateSharedPreference()` – Ukládá aktuální stav aktivity.
- `private void loadSharedPreferences()` – Nahrává poslední uložený stav aktivity.
- `private void updateDistanceAfterResume()` – Vytvoří požadavek na aktualizaci dat v aktivitě.
- `private void updateDistanceAfterDataReceived(ArrayList<MyData> tmpArrayListID)` – Po přijetí dat od služby provede jejich aktualizaci.

4.2.3. Třída MyMap

Třída `MyMap` je potomkem třídy `MapActivity`, slouží pro zobrazení mapy, nahraných tras a právě procházených tras.

- `public void onCreate(Bundle savedInstanceState)` – Volána vždy, když je aktivita prvně spuštěna, obsahuje inicializaci prvků.
- `public void onResume()` – Metoda volána vždy při přesunu aktivity do popředí, obsahuje všechny nezbytné inicializace prvků a komponent.
- `private void enableReceiver()` – Zaregistrování instance třídy `BroadcastReceiver` pro příjem komunikace od ostatních komponent aplikace.
- `private void disableReceiver()` – Odregistrování instance třídy `BroadcastReceiver` pro veškerou komunikaci od komponent aplikace.

- `public void onStart()` – Metoda volána vždy při přesunu aktivity do popředí.
- `public void onDestroy()` – Metoda volána vždy před vypnutím aktivity.
- `public void onPause()` – Volána vždy při přechodu aktivity na pozadí.
- `public boolean onCreateOptionsMenu(Menu menu)` – Vytvoří menu.
- `public boolean onOptionsItemSelected(MenuItem item)` – Obsluha kliknutí na položku v menu.
- `private void makeOverlayForSingleLocation(double lat, double lon)` – Podle předaných argumentů vykreslí nad mapu značku.
- `private void makeMyOverlayForRoute(ArrayList<MyData> locList)` – Podle předané trasy vykreslí nad mapu trasu.

4.2.4. Třída MyOverlay

Třída MyOverlay dědí ze třídy Overlay, obsahuje metodu draw pro vykreslení mapy. **Metody**

- `public void draw(Canvas canvas, MapView mapv, boolean shadow)` – Nastavuje vlastnosti čáry pro kreslení trasy a zajišťuje její vykreslení.

4.2.5. Třída ExportFile

ExportFile je potomkem třídy Activity, hlavním účelem této třídy je poskytnout uživateli možnost importovat a exportovat trasy. **Metody**

- `protected void onCreate(Bundle savedInstanceState)` – Volána vždy, když je aktivita prvně spuštěna, obsahuje inicializaci prvků
- `private void enableReceiver()` – Zaregistrování instance třídy BroadcastReceiver pro příjem komunikace od ostatních komponent aplikace.
- `private void disableReceiver()` – Odregistrování instance třídy BroadcastReceiver pro veškerou komunikaci od komponent aplikace.
- `public void onResume()` – Metoda volána vždy při přesunu aktivity do popředí, obsahuje všechny nezbytné inicializace prvků a komponent.

- `public void onStart()` – Volána vždy před přechodem aktivity do popředí.
- `public void onDestroy()` – Metoda volána vždy před vypnutím aktivity.
- `public void onPause()` – Volána vždy při přechodu aktivity na pozadí.
- `private void generateFile(ArrayList<MyData> dataToFile)` – Vytvoří kompletní obsah gpx souboru.
- `private void loadGPXSecond(String filename)` – Načte gpx soubor a pošle jeho obsah do servisu, který vloží záznamy trasy do databáze.
- `private void displayTrackInMap(Intent i)` – Pošle data s trasou mapě.
- `private void broadcastToService(String action, int recordId)` – Vyšle komunikaci službě.

Struktura gpx souboru

```
<?xml version='1.0' encoding='UTF-8' standalone='yes' ?>
<gpx version="1.1" creator="GPSTracker"
n0:schemaLocation="http://www.topografix.com/GPX/1/1
http://www.topografix.com/gpx/1/1/gpx.xsd" xmlns="http://www.topografix.com/GPX/1/1"
xmlns:n0="http://www.w3.org/2001/XMLSchema-instance">
  <wpt lat="" lon="">
    <ele> </ele>
    <time> </time>
  </wpt>
```

Obrázek 19. Struktura gpx souboru.

4.2.6. Třída Navigation

Třída `Navigation` je potomkem třídy `Activity` a slouží pro realizaci akustické navigace po zvolené trase. **Metody**

- `protected void onCreate(Bundle savedInstanceState)` – Volána vždy, když je aktivita prvně spuštěna, obsahuje inicializaci prvků.
- `public void onStart()` – Volána vždy před přechodem aktivity do popředí.
- `public void onDestroy()` – Metoda volána vždy před vypnutím aktivity.
- `public void onPause()` – Volána vždy při přechodu aktivity na pozadí.
- `public void onResume()` – Metoda volána vždy při přesunu aktivity do popředí, obsahuje všechny nezbytné inicializace prvků a komponent.
- `private void initSounds()` – Nahraje z `resources` všechny dostupné zvuky do `soundpoolu` a připraví je k přehrávání.
- `private Runnable goodBeep()` – Na základě zadaného `soundID` přehraje zvuk s vypočítanou hlasitostí,
- `private Runnable badBeep()` – Na základě zadaného `soundID` přehraje zvuk s vypočítanou hlasitostí.
- `private void navigate(double lat, double lon)` – Metoda je volána vždy, když jsou přijaty nové souřadnice, dostane je v podobě argumentů a provede výpočty všech hodnot nezbytných k navigaci a nastaví timer k akustické signalizaci.
- `private void initializeTimerGood(int time)` – Podle předaného času nastaví nový timer, který bude signalizovat přibližování k dalšímu bodu.
- `private void initializeTimerBad(int time)` – Funguje stejně, jako výše uvedená metoda, jen výsledný timer vydává zvuk signalizující pohyb směrem od průchozího bodu.
- `private double calculateDistanceHaversine(float lata, float lnga, float latb, float lngb)` – K výpočtu vzdálenosti mezi dvěma body je zde opět použita Haversinova formule.

4.2.7. Třída MyData

Třída `MyData` je implementací třídy `Parcelable`, byla vytvořena především jako datový nosič, obsahuje proměnné pro uchování dat. Je použita při přenosu dat z databáze, do databáze a při posílání mezi aktivitami. **Metody**

- `public MyData(int rId, double Long, double Lat, double Alt, String date)` – Konstruktor.
- `public void writeToParcel(Parcel dest, int arg1)` – Zapiše data do úložiště.

4.2.8. Třída MyDatabaseManager

Hlavním úkolem třídy `MyDatabaseManager` je vytvořit abstraktní vrstvu nad samotnou databází a implementovat metody pro manipulaci s daty. **Metody**

- `public MyDatabaseManager open() throws SQLException` – Vytvoří instanci třídy `MyDBHelper` a vytvoří tabulku.
- `public void close()` – Ukončuje práci s databází.
- `public void addRow(...)` – Přidá záznam do tabulky.
- `public void deleteRow(int rowId)` – Vymaže vybraný záznam.
- `public void clearTable()` – Vymaže celou databázi.
- `public void updateRow(...)` – Aktualizuje záznam v tabulce.
- `public ArrayList<MyData> getAllRowsAsArrays()` – Vrátí všechny záznamy z tabulky.
- `public ArrayList<String> getAllRecordsIdAndDate()` – Vrátí ze všech záznamů jejich id a datum porřízení.
- `public ArrayList<Integer> getAllRecordsId()` – Vrátí unikátní identifikátor všech záznamů tras.
- `public ArrayList<Object> getRowAsArray(long rowId)` – Vrátí řádek s příslušným identifikátorem.
- `public ArrayList<MyData> getAllRecordsById(int id)` – Vrátí všechny záznamy k příslušnému identifikátoru trasy.

4.2.9. Třída MyDBHelper

MyDBHelper je potomkem třídy SQLiteOpenHelper. Hlavním účelem třídy je vytvořit databázi a tabulku pro uložení zaznamenaných dat.

Metody

- `public void onCreate(SQLiteDatabase database)` – Vytvoří databázi.
- `public void onUpgrade(SQLiteDatabase database, int oldVersion, int newVersion)` – V případě přechodu na novou verzi databáze je zavolána tato metoda.
- `public void deleteDatabase(SQLiteDatabase database)` – Vymaže celou databázi.

Struktura tabulky pro uložení dat ukazuje následující obrázek.

	id	longitude	latitude	altitude	time	record id	date
1	1	17.2511683333333	49.59405	0.0	1312502406000	1	Aug 5, 2011
2	2	17.250755	49.5938183333333	0.0	1312502407000	1	Aug 5, 2011
3	3	17.251045	49.5935383333333	0.0	1312502408000	1	Aug 5, 2011

Obrázek 20. Struktura tabulky databáze pro ukládání naměřených dat.

Tabulka databáze je navržena tak, aby bylo možné z uložených dat v dalších verzích programu vytvořit přehledné grafy a statistiky tras.

4.2.10. Třída MyItemOverlay

Třída MyItemOverlay tvoří grafickou reprezentaci jednoho bodu a jeho vykreslení do mapy.

Metody

- `public MyItemOverlay(Drawable defaultMarker, Context context)` – Konstruktor, beroucí jako argumenty kontext aplikace a ikonu, kterou bude daný bod reprezentovaný v mapě.
- `public void addOverlay(OverlayItem overlay)` – Přidá prvek do seznamu prvků.

4.2.11. Třída MyPreferences

Třída `MyPreferences` je potomkem `PreferenceActivity` a implementuje `OnSharedPreferenceChangeListener`. Slouží především jako obrazovka nastavení aplikace, kde si uživatel může zvolit časový interval pro aktualizaci GPS souřadnic nebo minimální vzdálenost mezi aktualizacemi GPS souřadnic.

Metody

- `public void onCreate(Bundle savedInstanceState)` – Vytvoření aktivity.
- `public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, String key)` – Pokud uživatel změní nějaké hodnoty v nastavení, tak se pomocí této metody změna projeví i v datech aplikace.

Závěr

Cílem práce bylo vytvořit aplikaci, která by uživatelům umožnila sledovat, zaznamenávat jejich pohyb případně využít k pohybu po trase akustickou navigaci. Důležitou funkcí aplikace je také import a export tras v uložených v gpx souborech. Samozřejmostí je i zobrazování tras v mapě. Během tvorby aplikace jsem zjistil, že nejvhodnější pro zobrazení tras jsou Google Maps. V tomto ohledu by bylo možné do budoucna GPS tracker rozšířit o serverovou část, zde by se uživatel mohl přihlásit a umisťovat nasbíraná data. Dále by mohla být i část, která by z nahraných souborů generovala statistiky a přehledné grafy. Během studia problematiky operačního systému Android jsem se dověděl mnoho podstatných informací ohledně fungování a využití tohoto systému do budoucna. Myslím si, že již brzy tento systém zaujme dominantní postavení na trhu s chytrými telefony. Tato práce mě také obohatila o zkušenost s vývojem aplikací pro mobilní zařízení.

Conclusions

The goal of this bachelor work was to create an application where users can watch, record their movement and use acoustic route navigation. Very important feature is the import and export routes stored in gpx files. There is also possibility to show route in map. During the development of application, I found that the best way to show route in map is to use Google Maps. In the future the application could be expand with server part, where would be users data logs, statistics and graphs from data. During the study of problematic of Android operating system I found quite lot of information about how this system work and its possible use in future. I think that in nearly future Android will take the dominant standing in market with smart phones. This thesis gave me many experiences with developing application for mobile devices.

Reference

- [1] Meier, Reto. *Professional Android 2 Application Development*. Wrox, duben 2010.
- [2] Hashimi, Sayed; Komatineni, Satya; McLean, Dave. *Pro Android 2* Apress, březen 2010.
- [3] Mednieks, Zigurd; Dornin, Laird; Meike, G. Blake; Nakamura, Masumi. *Programming Android* <http://ofps.oreilly.com/titles/9781449389697/index.html>lektronická publikace, 2011.
- [4] L. Murphy, Mark. *Beginning Android 2* Apress, 2010.
- [5] Android. <http://www.android.com/>
- [6] Android Developers. <http://developer.android.com/resources/dashboard/platform-versions.html>
- [7] Cell Phone Rewiews Blog. <http://www.cellphoneanswers.info/android-architecture/>
- [8] Google Plus. <https://plus.google.com/u/0/106189723444098348646/posts/dRtqKJCbpZ7>
- [9] Android (operační systém). [http://cs.wikipedia.org/wiki/Android_\(opera%C4%8Dn%C3%AD_syst%C3%A9m\)](http://cs.wikipedia.org/wiki/Android_(opera%C4%8Dn%C3%AD_syst%C3%A9m))
- [10] Apache License. http://en.wikipedia.org/wiki/Apache_License
- [11] Gpx Schema Documentation. http://www.topografix.com/gpx/1/1/#type_wptType
- [12] Android XML Parser Performance. http://www.developer.com/ws/article.php/10927_3824221_2/Android-XML-Parser-Performance.htm
- [13] Calculate Distance, Bearing Between Latitude/Longitude. <http://www.movable-type.co.uk/scripts/latlong.html>
- [14] PacDv Free Sound Effects. http://www.pacdvd.com/sounds/interface_sounds-3.html
- [15] Global Positioning System. http://cs.wikipedia.org/wiki/Global_Positioning_System
- [16] Globální Družicový Polohový Systém. http://cs.wikipedia.org/wiki/Glob%C3%A1ln%C3%AD_dru%C5%BEicov%C3%BD_polohov%C3%BD_syst%C3%A9m

- [17] O technologii GPS. http://eu.mio.com/cs_cz/global-positioning-system.htm
- [18] iOS (apple). [http://cs.wikipedia.org/wiki/IOS_\(Apple\)](http://cs.wikipedia.org/wiki/IOS_(Apple))
- [19] Apple iOS 5. <http://www.apple.com/ios/ios5/>
- [20] Symbian OS. http://cs.wikipedia.org/wiki/Symbian_OS
- [21] Symbian OS. <http://www.root.cz/zpravicky/symbian-os-je-ode-dneska-otevreny/>
- [22] Windows Mobile. http://cs.wikipedia.org/wiki/Windows_Mobile
- [23] Windows Phone 7. http://cs.wikipedia.org/wiki/Windows_Phone_7
- [24] Icon drawer. <http://www.icondrawer.com/free.php>
- [25] Android forum. <http://androidforum.cz/>
- [26] Android Market. <https://market.android.com/?hl=cs>
- [27] Android Developers. <http://developer.android.com/index.html>
- [28] My Tracks. <https://market.android.com/details?id=com.google.android.maps.mytracks>
- [29] Open GPS Tracker. <https://market.android.com/details?id=nl.sogeti.android.gpstracker>
- [30] Live View GPS Tracking Utility. <https://market.android.com/details?id=com.liveviewgps>
- [31] GPS Essentials. <https://market.android.com/details?id=com.mictale.gpsessentials>
- [32] GPS Tracker <https://market.android.com/details?id=dev.gps>
- [33] GPS Tracker <https://market.android.com/details?id=com.instamapper.gpstracker>
- [34] tracker.ch - GPS tracker. <https://market.android.com/details?id=com.nth.android.tracker>

A. Obsah přiloženého CD

V samotném závěru práce je uveden stručný popis obsahu přiloženého CD/DVD, tj. závazné adresářové struktury, důležitých souborů apod.

`bin/`

Instalátor GPSTRACKER.APK programu.

`doc/`

Dokumentace práce ve formátu PDF, vytvořená dle závazného stylu KI PŘF pro diplomové práce, včetně všech příloh, a všechny soubory nutné pro bezproblémové vygenerování PDF souboru dokumentace (v ZIP archivu), tj. zdrojový text dokumentace, vložené obrázky, apod.

`src/`

Kompletní zdrojové texty programu GPS TRACKER se všemi potřebnými (převzatými) zdrojovými texty, knihovnami a dalšími soubory pro bezproblémové vytvoření spustitelných verzí programu (v ZIP archivu).

`readme.txt`

Instrukce pro instalaci a spuštění programu GPS TRACKER, včetně požadavků pro jeho provoz.

Navíc CD/DVD obsahuje:

`data/`

Ukázková a testovací data použitá v práci a pro potřeby obhajoby práce.

`install/`

Instalátory aplikací, knihoven a jiných souborů nutných pro provoz programu, které nejsou standardní součástí operačního systému.

U veškerých odjinud převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro materiály, u kterých toto není splněno, je uveden jejich zdroj (webová adresa) v textu dokumentace práce nebo v souboru `readme.txt`.