

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Aplikace pro mobilní sledování pohybových aktivit

Bc. Jakub Eliáš

Diplomová práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2012/2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jakub Eliáš**
Osobní číslo: **I11371**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Aplikace pro mobilní sledování pohybových aktivit**
Zadávací katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem úvodní části práce bude vysvětlit pojem "pohybové aktivity", zhodnotit existující nástroje pro měření kondice a dalších parametrů např. během sportovních aktivit, a to s důrazem na mobilní aplikace. Dále popsat vhodné technologie pro vývoj aplikací tohoto typu. Praktická část bude spočívat v analýze, návrhu a implementaci aplikace pro snímání polohy jedince pomocí mobilního zařízení s OS Android a následném vizuálním a statistickém vyhodnocení desktopovou aplikací realizovanou ve vybraném programovacím jazyce.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

MURPHY, Mark L. Android 2: průvodce programováním mobilních aplikací.

Vyd. 1. Brno: Computer Press, 2011, 375 s. ISBN 978-80-251-3194-7.

BOLLINGER, Gary a Bharathi NATARAJAN. JSP - Java Server Pages:

podrobný průvodce začínajícího tvůrce. 1. vyd. Praha: Grada, 2003, 418 s.

Moderní programování. ISBN 80-247-0340-8.

LEHNERT, Michal. Trénink kondice ve sportu. 1. vyd. Olomouc: Univerzita

Palackého v Olomouci, 2010, 143 s. ISBN 978-802-4426-143.

Vedoucí diplomové práce:

Ing. Zdeněk Šilar

Katedra informačních technologií

Datum zadání diplomové práce: **31. října 2012**

Termín odevzdání diplomové práce: **17. května 2013**

prof. Ing. Simeon Karamazov, Dr.

děkan



L.S.

prof. Ing. Antonín Kavička, Ph.D.

vedoucí katedry

V Pardubicích dne 15. listopadu 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10. 5. 2013

Jakub Eliáš

Poděkování

Rád bych poděkoval svým blízkým příbuzným za veškerou podporu v rámci celého mého studia. Dále bych rád poděkoval mému vedoucímu, panu Ing. Zdeňku Šilarovi za veškerou ochotu a pomoc během vypracování této práce.

Anotace

Práce se zabývá sledováním pohybových aktivit s důrazem na mobilní zařízení. Vysvětluje jejich důležitost a smysl vývoje aplikace pro jejich sledování. Praktická část je věnována popisu vlastní aplikace, jejímu návrhu a implementaci.

Klíčová slova

Pohybové aktivity, mobilní aplikace, GPS, Google Maps, OpenStreetMap, vizualizace map, sledování polohy

Title

Application for mobile tracking of movement activities

Annotation

This thesis deals with tracking of movement activities with focus on mobile devices and explains its importance and meaning of development application to track them. The practical part describes the custom application, its design and implementation.

Keywords

Movement activities, mobile application, GPS, Google Maps, OpenStreetMap, map vizualization, position tracking

Obsah

Seznam zkratk	8
Seznam obrázků	9
Seznam tabulek	9
Úvod	10
1 Pohybové aktivity	11
1.1 Výdej energie.....	11
1.1.1 Výpočet vydané energie	11
1.1.2 Index tělesné hmotnosti	13
2 Mobilní zařízení	15
2.1 Vývoj pro mobilní zařízení.....	15
2.2 Operační systém Android	15
2.2.1 Architektura OS Android	16
2.2.2 Vývoj pro OS Android	18
2.2.3 Základní stavební prvky aplikace Android.....	19
2.3 Sledování polohy	21
3 Vizualizace map	23
3.1 OpenStreetMap.....	23
3.2 Google Maps	24
4 Dostupná řešení	27
4.1 Sports Tracker	27
4.2 Runstar™.....	28
4.3 RunKeeper.....	29
4.4 Endomondo.....	32
5 Návrh a implementace aplikace	36
5.1 Mobilní aplikace.....	36
5.1.1 Získání polohy	38
5.1.2 Měření výkonů.....	38
5.1.3 Uložení dat.....	40
5.2 Desktopová aplikace.....	42
5.2.1 Datový model	42
5.2.2 Komponenty aplikace	44

5.2.3	JMapViewr	47
5.2.4	JFreeChart.....	48
5.2.5	GUI a ovládání	50
5.2.6	Sestavení aplikace.....	51
Závěr		54
Literatura		55
Příloha A – Obsah přiloženého DVD		56

Seznam zkratk

GPS	Global Positioning Systém
BMI	Body mass index
OS	Operační systém
API	Application Programming Interface
XML	Extensible Markup Language
JDBC	Java Database Connectivity
MVC	Model View Controller
AVD	Android Virtual Devices
SDK	Software Development Kit
ADT	Android Development Tools
DVM	Dalvik Virtual Machine
USB	Universal Serial Bus
StAX	Streaming API for XML

Seznam obrázků

Obrázek 1 - Graf hodnoty BMI [8].....	13
Obrázek 2 - Životní cyklus aktivity [3]	20
Obrázek 3 - Ukázka OpenStreetMap.....	24
Obrázek 4 – Ukázka Google Maps.....	25
Obrázek 5 – Ukázka z aplikace Sports Tracker.....	27
Obrázek 6 - Ukázka rozhraní aplikace Runstar [11]	28
Obrázek 7 - Ukázka aplikace Runstar [11].....	29
Obrázek 8 - Ukázka aplikace RunKeeper [11].....	30
Obrázek 9 - Ukázka aplikace RunKeeper [11].....	31
Obrázek 10 - Ukázka aplikace Endomondo [11].....	32
Obrázek 11 - Ukázka aplikace Endomondo č. 2 [11].....	33
Obrázek 12 - Ukázka aplikace Endomondo č. 3 [11].....	34
Obrázek 13 - Ukázka aplikace Endomondo č. 4 [11].....	35
Obrázek 14 - Ukázka výběru aktivity v aplikaci	37
Obrázek 15 - Ukázka výběru trasy	37
Obrázek 16 - Ukázka trati.....	39
Obrázek 17 - Ukázka aplikace před a po startu aktivity.....	40
Obrázek 18 - Ukázka struktury souboru <i>trasy.dat</i>	41
Obrázek 19 - Ukázka struktury souboru <i>vykon.dat</i>	41
Obrázek 20 - ER diagram databáze aplikace.....	43
Obrázek 21 - Aplikace jako celek.....	45
Obrázek 22 - Tabulka výkonů	45
Obrázek 23 – Tabulka vypočtených hodnot	46
Obrázek 24 - Dialogové okno s údaji o sportovci	46
Obrázek 25 - Ukázka zobrazení trati v desktopové aplikaci	47
Obrázek 26 - Výškový profil tratě (XY graf).....	48
Obrázek 27 - Porovnání časových výkonů	49
Obrázek 28 - Graf spálených kilojoulů	49
Obrázek 29 - Graf celkových časů výkonů (sloupcový graf).....	50
Obrázek 30 - Obsah souboru <i>config.xml</i>	51
Obrázek 31 - Přidaný kód do souboru <i>pom.xml</i>	52
Obrázek 32 - Přidání tagů <i><dependency></i>	53
Obrázek 33 - Adresářová struktura projektu v Maven	53

Seznam tabulek

Tabulka 1 – hodnoty vydané energie.....	12
Tabulka 2 - rozdělení hodnot BMI [8].....	14

Úvod

Mobilní zařízení si již vybudovala velmi důležitý podíl na trhu s elektronikou, který si již řadu let drží. Jde o tuhý konkurenční boj, kterého se účastní hned několik velkých firem. V posledních letech prokazují dominantní postavení zařízení postavená na platformě Android, která je předmětem i této práce.

V souvislosti s nárůstem mobility zařízení by se dalo mluvit o poklesu mobility lidí a s tím spojeným poklesem fyzické zdatnosti celých generací obyvatel po celém světě. Představa ovládání okolí bez nutnosti většího pohybu vytváří návyk pro sedavá zaměstnání a trávení času před elektronickými zařízeními. Právě nejrůznější aplikace v mobilním zařízení se však pokouší o impuls, který dokáže „vyhnat“ člověka ven a donutit ho vzdorovat těmto návykům pohybem. Aplikací vzniká celá řada pro různé platformy a mají různé možnosti dle toho, co je možné pomocí daného zařízení získávat.

Důležité je však, aby uživatele zaujaly a zpříjemňovaly mu tím chuť pracovat na sobě a zdokonalovat svoji fyzickou kondici. V rámci práce nás budou zajímat aplikace zaměřené na pohybové aktivity. V úvodní části bude pojem vysvětlen společně s vysvětlením významu důležitosti těchto aktivit. Dále bude uvedeno, jak může člověk jejich prostřednictvím na sobě pracovat a jak sledovat výsledky svého snažení, zejména výdej energie, který je jednou z hlavních hodnot. Tato informace bude doplněna o výpočet a význam BMI indexu, který je často sledovanou hodnotou v souvislosti s lidským zdravím.

Teoretická část pokračuje nastíněním současné situace trhu s mobilními zařízeními a dále se zaměřuje na platformu Android a popisuje její architekturu a základní stavební části. Zde se práce zabývá i vývojem pro tato zařízení a bude uvedeno i několik informací spojených se sledováním polohy pomocí GPS v těchto přístrojích.

Pro lákavý vizuální vzhled je dnes běžné využití map. O tom je právě následující kapitola. Popsány budou zejména zdarma využitelné knihovny pro tyto účely, a to OpenStreetMap a Google Maps. Důvodem je především jejich využití v praktické části.

Nadcházející kapitola potom popisuje dostupná řešení, která mohou uživatelé ve svých zařízeních využít v souvislosti s pohybovými aktivitami. Zde bude zmíněno několik řešení a budou zhodnoceny jejich přednosti a nedostatky.

Následně bude předveden návrh vlastního řešení rozdělený na dvě části o mobilní aplikaci, která zastává funkci snímače pohybových aktivit v přenosném zařízení, a o desktopové aplikaci, která následně bude sloužit pro vizuální a statistické zhodnocení výsledků aktivit. Zde bude uveden postup řešení a samotná implementace zajímavých částí aplikací společně s vysvětlením funkcionality a způsobu prezentace a ovládáním jednotlivých komponent.

1 Pohybové aktivity

Pod pojmem pohybová aktivita si představme druh tělesné aktivity, která je výsledkem svalové práce provázené zvýšením energetického výdeje, charakterizované svébytnými vnitřními determinantami a vnější podobou. Pohybové aktivity představují mnohovýznamový konstrukt a podle kontextu jsou dále různě označovány jako strukturované, nestrukturované, zdraví podporující, bazální, běžné každodenní, sportovní apod.

Nás dále budou v rámci práce zajímat běžné každodenní aktivity, konkrétně transportní pohybové aktivity. Významným příkladem v této oblasti je chůze a jízda na kole jako prostředek transportu. Dospělým se dnes doporučuje na základě zjištěných zdravotních benefitů těchto aktivit zařadit chůzi a jízdu na kole do denního režimu pro přesun do zaměstnání.

Průvodním jevem současného životního stylu je pohybová nedostatečnost a z ní vyplývající zdravotní komplikace. Nejčastěji zmiňovaným důsledkem je významný vzestup nadváhy a obezity, jak u dětské, tak i dospělé populace. To má za následek řadu zdravotních komplikací, které zpětně rozhodujícím způsobem ovlivňují životní styl.

Dalším problémem, který je spojen s nedostatkem pohybového zatížení lokomočního typu, je stále se snižující zdatnost populace. Zdatnost není v současnosti chápána jen jako předpoklad pro realizaci výkonu, ale ve stále větší míře je zdůrazňováno její léčebně preventivní působení. Je důležitým předpokladem aktivního životního stylu. Chápeme ji jako schopnost organismu vyrovnat se s vnějším stresem nebo mu odolávat. Zdatnost lze ovlivňovat pouze přiměřenými pohybovými aktivitami a vhodně sestaveným pohybovým programem [4].

1.1 Výdej energie

Každá práce či sportovní výkon má určitou energetickou hodnotu. Tento výdej energie závisí mimo jiné na tělesné hmotnosti. Pohyb silnějšího jedince je pochopitelně energeticky náročnější. Orientačně lze odhadnout, že dvakrát těžší jedinec vydává přibližně dvakrát více energie při stejné rychlosti pohybu. Platí však, že obézní lidé chodí obvykle pomaleji než štíhlí.

1.1.1 Výpočet vydané energie

V kalkulaci energetického výdeje se musíme rozhodnout pro vyjadřování v kilojoulech či kaloriích. Vzhledem k tomu, že energetické hodnoty potravin se dnes uvádějí v kilojoulech, má menší smysl starší vyjadřování náročnosti pohybových aktivit v kaloriích. Dále je obvyklý minutový či hodinový způsob vyjadřování. Energetické hodnoty sportovních aktivit jsou uvedeny v tabulce 1.

Tabulka 1 – hodnoty vydané energie

Aktivita	Vydaná energie
Běh 8 km/h	0.559 kJ/min/kg
Běh 12 km/h	0.942 kJ/min/kg
Běh 16 km/h	1.117 kJ/min/kg
Chůze - intenzivní	0.28 kJ/min/kg
Chůze - pomalá	0.174 kJ/min/kg
Chůze - střední	0.245 kJ/min/kg
Jízda na kole - pomalá	0.28 kJ/min/kg
Jízda na kole - rychlá	1.117 kJ/min/kg
Jízda na kole - střední	0.559 kJ/min/kg
Plavání – klidné	0.419 kJ/min/kg

K výpočtu celkově vydané energie slouží následující vzorec:

Celkově vydaná energie

$$= \text{hodnota z tabulky} * \text{délka trvání sportu} * \text{hmotnost jedince}$$

Výsledná vydaná energie je v jednotkách kilojoule (kJ), přičemž délka trvání sportu se uvádí v minutách a hmotnost v kilogramech. V praxi se stále často objevuje kalorie (značka cal) jako jednotka energie popř. tisíckrát větší kilokalorie (kcal). Přepočet potom spočívá ve vynásobení resp. vydělení konstantou 4,185 kJ. Tato konstanta je dána z definice kalorie, která říká, že kalorie je množství energie, které dokáže zvýšit teplotu 1 gramu vody ze 14,5°C na 15,5°C, a jelikož měrná tepelná kapacita vody je asi 4185 J * kg⁻¹ * K⁻¹, kde K je jednotkou teplotních stupňů v kelvinech. Platí tedy:

$$1 \text{ cal} = 4,185 \text{ J}$$

U sportů dochází k výrazně vyšší zátěži než při běžné práci. Minimálním cílem by mělo být spálení zhruba 8000 kJ týdně organizovanými pohybovými aktivitami, optimálně rozdělenými do třicetiminutových intervalů denních pohybových aktivit nebo do hodinových intervalů alespoň třikrát týdně. Jak vyplývá z uvedené tabulky, je tímto minimem například více než třicetiminutová intenzivní chůze denně. Jiný typ

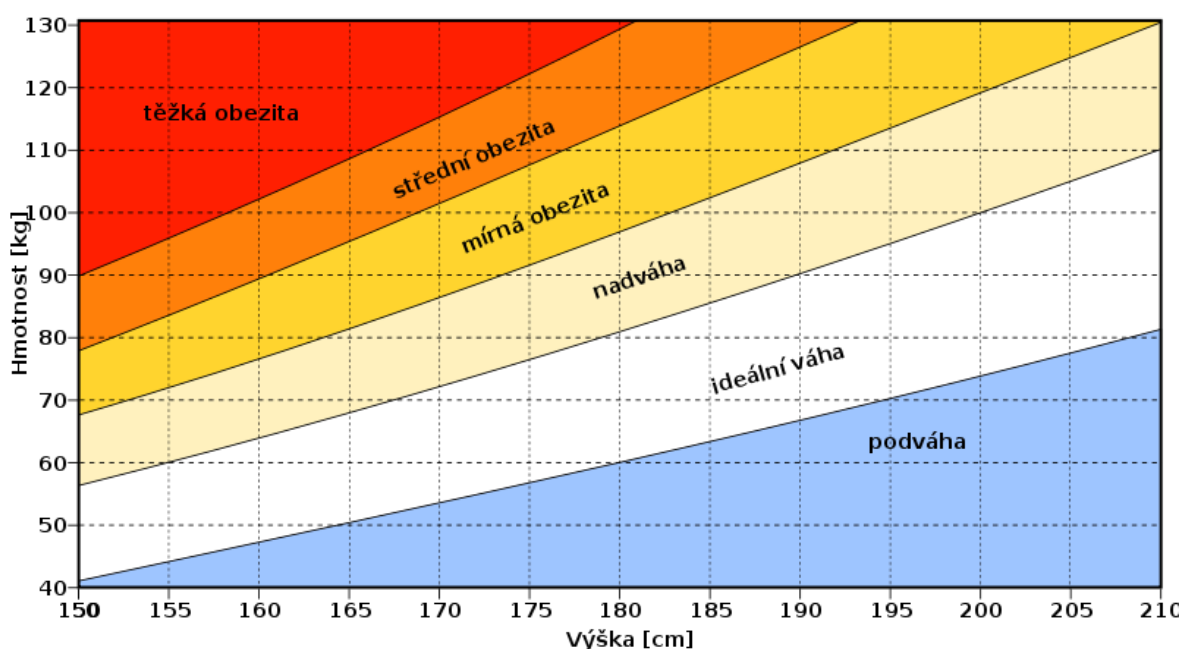
doporučení uvádí jako minimum výdej 20 až 30 kJ na 1 kg hmotnosti denně u štíhlejších a alespoň 15 kJ na 1 kg hmotnosti u obézních [4].

1.1.2 Index tělesné hmotnosti

Index tělesné hmotnosti, obvykle označovaný zkratkou BMI z anglického body mass index, je číslo používané jako indikátor podváhy, normální tělesné hmotnosti, nadváhy a obezity umožňující statistické porovnávání tělesné hmotnosti lidí s různou výškou. Index se spočítá vydělením hmotnosti daného člověka druhou mocninou jeho výšky:

$$BMI = \frac{\text{hmotnost}}{\text{výška}^2}$$

Do tohoto vzorečku se dosazuje hmotnost v kilogramech a výška v metrech a výsledná jednotka kg/m² se často vynechává [8].



Obrázek 1 - Graf hodnoty BMI [8]

Výsledná hodnota BMI se obvykle následně klasifikuje dle hodnot z tabulky vytvořené Světovou zdravotnickou organizací. V populaci se objevují hodnoty indexu v rozmezí od přibližně 15 (závažná podvýživa) až přes 40 (morbidní obezita). Přesné hranice mezi jednotlivými kategoriemi (závažná podvýživa, podvýživa, optimální váha atd.) se mezi různými odborníky liší, ale všeobecně je BMI pod 18,5 považováno za podváhu, která může být příznakem nějaké poruchy stravování či jiného zdravotního problému, zatímco BMI nad 25 se považuje za nadváhu a nad 30 za příznak obezity. Tyto hranice platí pro dospělé starší 20 let.

Tabulka 2 - rozdělení hodnot BMI [8]

Kategorie	Rozsah BMI – kg/m ²	Základní BMI	Hmotnost osoby vysoké 180 cm
těžká podvýživa	≤ 16,5	méně než 0,6	méně než 53,5 kg
podváha	16,5 – 18,5	0,6 – 0,74	od 53,5 do 60 kg
ideální váha	18,5 – 25	0,74 – 1	od 60 do 81 kg
nadváha	25 – 30	1 – 1,2	od 81 do 97 kg
mírná obezita	30 – 35	1,2 – 1,4	od 97 do 113 kg
střední obezita	35 – 40	1,4 – 1,6	od 113 do 130 kg
morbidní obezita	> 40	nad 1,6	nad 130 kg

BMI se využívá obvykle jako statistický nástroj. U konkrétního jedince je BMI příliš jednoduchým prostředkem, který ignoruje velké množství důležitých faktorů (např. stavbu těla, množství svalstva apod.), např. kulturista může mít hodnotu BMI nad 30, a přesto nebýt obézní, protože velké hodnota indexu je u něj dána velkým množstvím svalové hmoty a naopak starší lidé s malým množstvím svalstva mohou být obézní, a přesto jejich hodnotou BMI řazeni do kategorie ideální váhy. Hranice hodnot BMI se také liší pro různé rasy. Asiaté například používají o něco nižší hranice, za obézní se považují lidé s hodnotou nad 27,5 a jako ideální je stanoveno rozmezí 18,5 – 23. Pro populaci Spojených států amerických je dle hlediska nejmenší úmrtnosti optimální hranice v rozmezí 18 – 27, pro Evropu je to hodnota 25.

V klinické praxi se proto obvykle používají přesnější testy jako měření tloušťky podkožního tuku, impedanční měření atd. BMI je nejužitečnější pro statistické průzkumy mezi rozsáhlejšími vzorky populace, nejčastěji pro zkoumání korelace mezi obezitou a jinými faktory. Důvodem pro použití BMI je, že pro jeho použití stačí v datech uvádět výšku a hmotnost [8].

V současné době se na internetu objevuje čím dál tím více aplikací, které pomáhají sledovat či vyhodnocovat pohybovou aktivitu jedince. Zejména aplikace pro mobilní zařízení jsou vhodnou variantou sledování pohybových aktivit.

2 Mobilní zařízení

Mobilní zařízení jsou v dnešní době nedílnou součástí každodenního života většiny lidí po celém světě. Příkladem mobilního zařízení, které se stalo již před několika lety naprostým standardem, je mobilní telefon. Dnes si však mnoho lidí s pouhým mobilním telefonem nevystačí, a proto si pořizují jejich chytřejší následovníky tzv. chytré telefony, které jsou také nazývány jako smartphony. Tato zařízení hravě nahrazují běžné mobilní telefony a navíc disponují mnoha dalšími funkcemi, které se mnohdy liší podle výrobce. V současné době mají na trhu dominantní podíl firmy Samsung, Nokia a Apple následované dalšími neméně významnými výrobci jako jsou ZTE, LG Electronics, Huawei Technologies atd.

Mobilní telefony však nejsou jedinými zařízeními, která řadíme do kategorie mobilních. Dále sem patří i notebooky, netbooky, PDA, tablety, navigace, popřípadě MP3 přehrávače a další zařízení. V následujících podkapitolách se budu zabývat převážně mobilními telefony, protože právě pro mobilní telefon byla vytvořena mobilní aplikace, která je předmětem této práce.

2.1 Vývoj pro mobilní zařízení

S rostoucím prodejem mobilních zařízení roste i zájem a potřeba vývoje aplikací pro tato zařízení. Již zmínění výrobci Apple a jejich iPhone s podporou operačního systému iOS a Samsung s podporou OS Android jsou tedy hlavními cíli pro vývoj aplikací v současné době. Ačkoliv Android na trhu iOS značně převyšuje, často se aplikace tvoří hlavně pro tyto dvě platformy. Příkladem dalších méně významných operačních systémů pro mobilní zařízení jsou například BlackBerry a Windows Phone.

2.2 Operační systém Android

Společnost Android Inc. byla založena v Kalifornii v říjnu 2003 Andym Rubinem, Richem Minerem, Nickem Searsem a Chrisem Whitem. Google Inc. v srpnu roku 2005 odkoupil v té době nepříliš známou „startup“ firmu Android Inc. a udělal z ní svoji dceřinou společnost.

Po odkupu společnosti tým Googlu pod vedením Andyho Rubina vyvinul platformu založenou na Linuxovém jádře a v září roku 2007 Google získal několik patentů v oblasti mobilních technologií. Odborná veřejnost začala po akvizici spekulovat, že Google chce tímto krokem vstoupit na trh „chytrých“ mobilních telefonů a chystá vydání vlastního telefonu.

5. listopadu v roce 2007 bylo vytvořeno uskupení Open Handset Alliance. Konsorcium, které zahrnovalo společnosti zabývající se výrobou mobilních telefonů, čipů nebo mobilních aplikací, např. Google, HTC, Intel, LG, Motorola, Samsung, Qualcomm, NVIDIA a dalších 26 společností. Cílem tohoto konsorcia bylo vyvinout otevřený standard

pro mobilní zařízení. O týden později byl vydán první Android SDK pro vývojáře pod licenci open-source.

V říjnu roku 2008 byl ve Spojených státech amerických uveden první komerční telefon vyrobený firmou HTC s operačním systémem Android (v České republice byl uveden v lednu 2009) a zároveň s tím bylo uvolněno SDK 1.0. V roce 2009 roste počet zařízení používající Android na více jak dvacet. Na konci roku 2010 se Android stal vedoucí smartphone platformou a na počátku roku 2012 už jednoznačně dominoval trhu se smartphony s 59% tržním podílem.

Android představuje rozsáhlou open source platformu, která vznikla zejména pro mobilní zařízení, jako jsou chytré telefony, PDA, navigace, tablety atd. Podíl na trhu prodeje zařízení s tímto operačním systémem je na prvním místě a předpokládá se, že tento trend bude nadále pokračovat. To je jeden z hlavních důvodů, proč jsem si vývoj pro tuto platformu vybral pro svoji diplomovou práci.

Při vývoji systému byla brána v úvahu omezení, kterými disponují klasické mobilní zařízení, jako výdrž baterie, menší výkonnost a málo dostupné paměti. Zároveň bylo jádro Androidu navrženo pro běh na různém hardwaru. Systém tak může být použit bez ohledu na použitý chipset, velikost či rozlišení obrazovky.

Samotná platforma Android dává k dispozici nejen operační systém s uživatelským prostředím pro koncové uživatele, ale i kompletní řešení nasazení operačního systému (specifikace driverů aj.) pro mobilní operátory a výrobce zařízení a v neposlední řadě pro vývojáře aplikací poskytuje efektivní nástroje pro jejich vývoj – *Software Development Kit* [1].

2.2.1 Architektura OS Android

Architektura operačního systému Android je rozdělena do 5 vrstev. Každá z nich má svůj účel a nemusí být přímo odděleny od sebe:

- jádro OS
- knihovny
- *Android Runtime*
- *Android framework*
- základní aplikace

Nejnižší vrstva architektury je jádro operačního systému, které tvoří abstraktní vrstvu mezi používaným hardwarem a zbytkem softwaru ve vyšších vrstvách. Jádro systému Androidu je postaveno na Linuxu ve verzi 2.6. Je využito jeho mnoha vlastností, jako podpora správy paměti, správa sítí, zabudované ovladače nebo správy procesů,

například souběžného běhu aplikací, které běží jako samostatné procesy s oprávněním stanoveným systémem.

Další vrstvou jsou knihovny napsané v jazyce C/C++. Slouží různým komponentám systému a jejich funkce jsou vývojářům poskytnuty prostřednictvím *Android Application Framework*. Jedná se například o knihovny pro přehrávání různých druhů formátů audia a videa, *LibWebCore* knihovna webového prohlížeče, *FreeType* knihovna pro vykreslování bitmapových a vektorových fontů, *OpenGL* knihovna pro vykreslování 3D grafiky a další.

Vrstva Android Runtime obsahuje aplikační virtuální stroj nazvaný *Dalvik*, který byl týmem v Googlu vytvořen speciálně pro Android. *Dalvik Virtual Machine* (DVM) je registrově orientovaná architektura, která využívá základních vlastností Linuxového jádra, jako je práce s vlákny nebo správa paměti. Důvodem vzniku nového virtuálního stroje byla zejména licenční práva, kdy jazyk Java a jeho knihovny jsou volně šiřitelné, zatímco *Java Virtual Machine* není. Dalším důvodem byla optimalizace virtuálního stroje pro mobilní zařízení, a to především v oblasti poměru výkonu a úspory energie. V této vrstvě jsou také obsaženy základní knihovny programovacího jazyka Java. Knihovny se svým obsahem blíží platformě *Java Standard Edition*. Hlavní rozdíl je v nepřítomnosti knihoven pro uživatelské rozhraní (AWT, Swing), které byly nahrazeny knihovnami uživatelského rozhraní pro Android nebo přidáním knihovny Apache pro práci se sítí. Překlad aplikace napsané pro Android probíhá zkompileováním zdrojového Java kódu do Java byte kódu pomocí stejného kompilátoru, jako je používán v případě překladu Java aplikací. *Javabyte kód* se poté překompileje pomocí *Dalvik* kompilátoru a výsledný *Dalvik byte kód* je možné spustit na DVM. Spuštění každé Android aplikace běží ve svém vlastním procesu, s vlastní instancí DVM.

Nejdůležitější vrstvou pro vývojáře je *Application framework*, která poskytuje přístup k velkému množství služeb často používaných přímo v aplikacích. Jsou to služby zpřístupňující data jiných aplikací, prvky uživatelského rozhraní, aplikace běžící na pozadí, hardware používaného zařízení a mnoho dalších služeb a funkcí. Základní sada služeb zahrnuje především:

- Sada prvků *View* – slouží pro vytvoření grafického uživatelského rozhraní a skládá se z prvků, jako jsou tlačítka, seznamy, textová pole atp.
- *Content Providers* – tato služba umožňuje přístup k datům jiných aplikací.
- *Resource Manager* – poskytuje přístup k „nekódovým“ zdrojům, jako například textové řetězce, obrázky a jiné přidané soubory.
- *Notification Manager* – to je služba, která umožňuje aplikacím zobrazit upozornění ve stavovém řádku.
- *Activity Manager* – poskytuje orientaci v zásobníku aplikací a řídí životní cyklus aplikací.

Nejvyšší vrstvou systému jsou základní aplikace využívané běžnými uživateli. Jedná se o aplikace předinstalované jako SMS program, e-mailový klient, kalkulačka, kalendář nebo dodatečně stažené z Android Marketu [1].

2.2.2 Vývoj pro OS Android

Oficiálně podporované vývojové prostředí pro aplikace Android je Eclipse. Do něj je možné nainstalovat *ADT plugin*, který ulehčuje práci s Android projektem.

Nástroje pro vývoj aplikací pro platformu Android jsou obsaženy v *Android Software Development Kit* (SDK), který je dostupný pro všechny hlavní platformy operačních systémů a je rozdělen na tři druhy:

- základní konfigurace vývojového prostředí
- doporučená konfigurace vývojového prostředí
- plná konfigurace vývojového prostředí

Základní konfigurace SDK zahrnuje:

- *SDK Tools* – obsahuje nástroje pro debugování (ddms), analýzu grafického layoutu, testování aplikace, správu *Android Virtual Devices* (AVD), Android emulátor a další potřebné nástroje.
- *SDK Platform-tools* – obsahuje další potřebné nástroje pro vývoj závislé na verzi platformy. Příkladem může být *Android Debug Bridge* potřebný pro nahrávání souborů do zařízení.
- *Android SDK platforms* – se skládá z knihoven, ukázkových kódů atp. Alespoň jedna platforma musí být přítomna pro kompilaci aplikace, nastavení a běh AVD.

Doporučená konfigurace navíc obsahuje:

- USB Driver
- dokumentace
- příklady kódů

USB Driver je nezbytný pro testování a ladění nainstalovaných aplikací ve Windows.

Plná konfigurace:

- Google API – je API zpřístupňující rozhraní Google Maps.

Součástí SDK je také Emulátor operačního systému Android pro účely testování vytvořených aplikací bez nutnosti fyzického zařízení. Za pomoci *AVD Manageru* a Android SDK je možné konfigurovat volbu síťového připojení, SD karty atd. a spouštět

jednotlivá virtuální zařízení. Aplikace se většinou chovají ve virtuálním prostředí stejně jako ve fyzickém zařízení, jsou však situace, které se těžko virtualizují nebo virtualizace není možná vůbec, např. přijímání hovorů, funkce bluetooth, úroveň nabití baterie nebo audio/video vstup.

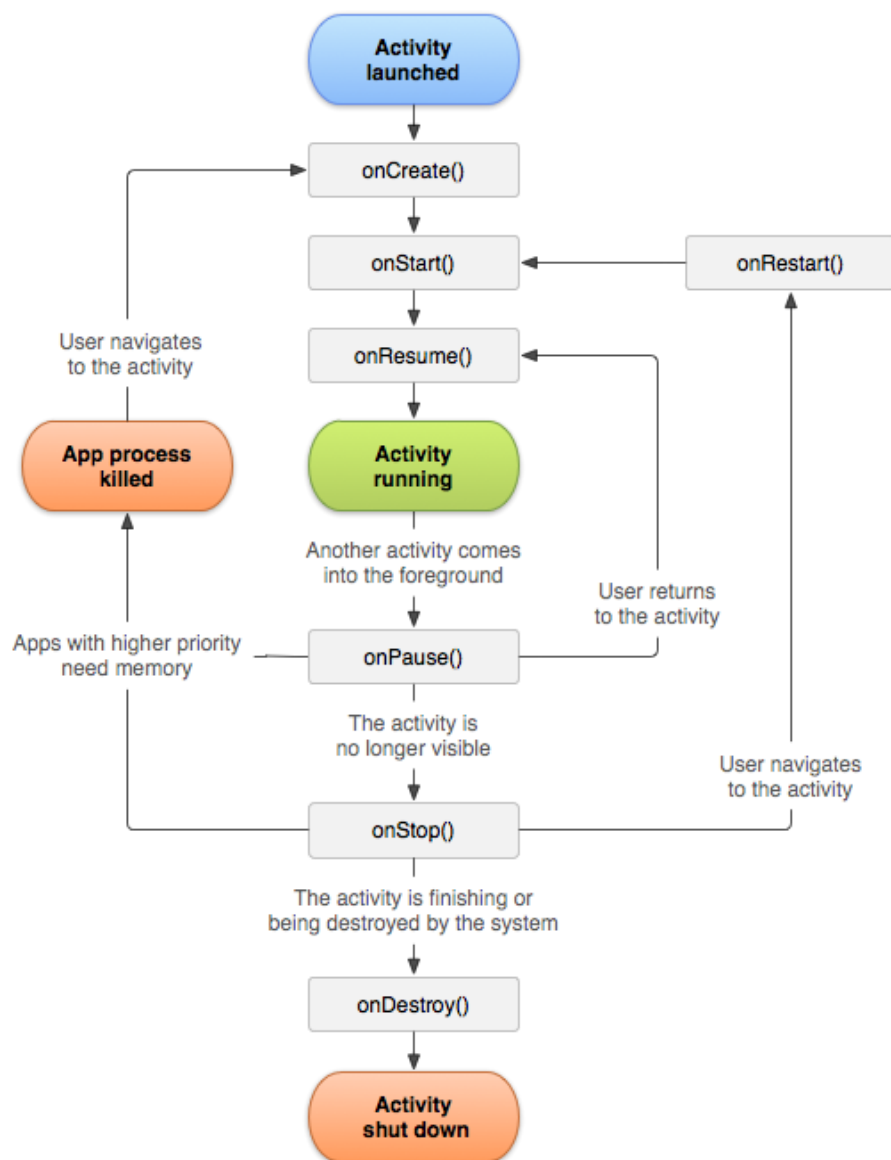
2.2.3 Základní stavební prvky aplikace Android

Základními stavebními kameny Android aplikace jsou komponenty nazývané:

- *Activity* – aktivita reprezentující obrazovku.
- *Service* – služba dovolující provádět akce na pozadí.
- *Content providers* – poskytovatel obsahu pro přístup k datům.
- *Broadcast receiver* – přijímač přenosu reagující na příchozí oznámení.

Nadefinování všech těchto komponent je nutné v souboru `AndroidManifest.xml`, který je uložen v kořenovém adresáři aplikace. Existují zde také zprávy zvané *intenty*, které kromě *content provideru* také umožňují komunikaci mezi komponentami.

Jedna aktivita odpovídá jedné obrazovce. Každá tedy obsahuje grafické uživatelské rozhraní dané XML souborem určujícím layout. V rámci aplikace obvykle existuje více aktivit, mezi kterými se uživatel přepíná, a ony si mohou přitom předávat informace. Zahájení aktivity je poměrně náročná záležitost. Je nutno vytvořit nový proces, alokovat paměť pro objekty uživatelského rozhraní, které se rozloží do layoutu obrazovky, a obrazovku zobrazit. Aby nedocházelo k zbytečnému plýtvání výpočetních prostředků při vzniku, zániku a opětovném vzniku aktivity, k čemuž může jednoduše docházet při přecházení mezi obrazovkami pomocí tlačítek, je zde *Activity Manager*, který je zodpovědný za celkovou správu životního cyklu aktivity. Tento manažer pracuje se zásobníkem, do kterého ukládá informace o spuštěných aktivitách, přičemž na vrcholu je ta aktuálně zobrazovaná. Životní cyklus je graficky znázorněn na následujícím obrázku.



Obrázek 2 - Životní cyklus aktivity [3]

Stavy aktivity jsou následující:

- *Activity launched* – označuje počátek a započetí aktivity.
- *Activity running* – je stav, kdy je aktivita zobrazena uživateli a může s ním interagovat. V tomto stavu může být v jednom okamžiku tedy právě jedna aktivita.
- *App process killed* – k tomuto stavu dochází, když *Activity Manager* zruší aktivitu z důvodu nedostatku paměti, a to v případě, že je aktivita viditelná např. při zobrazeném dialogovém okně, ale častěji v případě, že je neviditelná.
- *Activity shut down* – tj. stav, kdy je aktivita ukončena a nevyužívá žádnou paměť.

Service komponenta představuje dlouho vykonávající se úkol běžící na pozadí nebo přístup k vzdáleným zdrojům, kde není známa doba odezvy jako například v případě připojení k serveru. Tato služba se spouští dvěma způsoby, a to za použití metody *bindService* vyvolanou jinou komponentou neboli klientem. Za její pomoci může být k službě navázáno více komponent naráz a k jejímu ukončení dochází až v okamžiku odpojení všech komponent. Pouze stejný klienty ji potom může ukončit. Druhý způsob je metodou *startService*. Zde není striktně určeno, která komponenta může službu ukončit. Stav, ve kterých se může nacházet, jsou následující:

- *Component calls* – stav, kdy se služba inicializuje po zavolání nebo navázání komponentou.
- *Service is running* – okamžik, kdy služba vykonává na pozadí svoji funkci.
- *Service is shut down* – služba byla ukončena.

Content provider se označuje jako aplikační rozhraní, a to z důvodu, že v aplikaci slouží ke sdílení dat ať už mezi aplikacemi, tak i mezi jednotlivými aktivitami. Aplikace tedy mohou přistupovat k datům jiných aplikací uložených v *SQLite* databázi, souborech či na webu, pokud k tomu mají povolení. Tvoří jej relativně jednoduché rozhraní se standartními metodami *delete*, *insert*, *update* apod. Oddělení uživatelského rozhraní a dat umožňuje pracovat s daty a nahradit tak původní aplikace novými a vylepšenými.

Broadcast receiver je komponentou, která naslouchá nově přichozím oznámením a reaguje na ně příslušnou operací jako výpisem na stavový řádek popř. spuštěním jiné komponenty. Používá se například jako reakce na oznámení o doručení SMS zprávy, o vyfocení fotografie, o stažení dat nebo o nízkém stavu baterie [3].

2.3 Sledování polohy

Nejrozšířenější systém pro sledování polohy je *Global Positioning System*, zkráceně GPS, tj. vojenský globální družicový polohový systém provozovaný Ministerstvem obrany Spojených států amerických, s jehož pomocí je možné určit polohu a přesný čas kdekoli na Zemi nebo nad Zemí s přesností do deseti metrů. Přesnost GPS lze s použitím dalších metod ještě zvýšit až na jednotky centimetrů. Část služeb tohoto systému s omezenou přesností je volně k dispozici i civilním uživatelům.

Sledování polohy pomocí GPS je pro „chytré“ mobilní telefony s operačním systémem Android standardem. Obvykle zde lze nalézt dva různé druhy zjišťování polohy. Jedním z nich je tzv. síťové umístění, to zjistí polohu pouze zhruba. Druhé je satelitní umístění, které se používá pro navigaci. Android telefony, stejně jako mnoho jiných smartphonů, také používají *Assisted GPS* (aGPS). To jim umožňuje vypočítat polohu družice pomocí sítě a získat místo rychleji.

Telefony Android mají v sobě zabudovaný skutečný GPS čip a ten jim umožňuje získat umístění z GPS satelitů. Avšak získat první umístění v některých oblastech vyžaduje

trpělivost. V některých oblastech to může trvat 10 sekund, v jiných třeba až 5 minut. Se získáním umístění také souvisí, že je obvykle nutné mít výhled na oblohu, to právě kvůli satelitům.

GPS je v telefonu poměrně náročná na baterii, když sledování polohy momentálně nevyužíváte, je vhodnější GPS deaktivovat a šetřit tak její energii [7].

3 Vizualizace map

Vizualizace map slouží pro vykreslení mapových podkladů, tak aby co nejdříve kopírovali reálný povrch terénu.

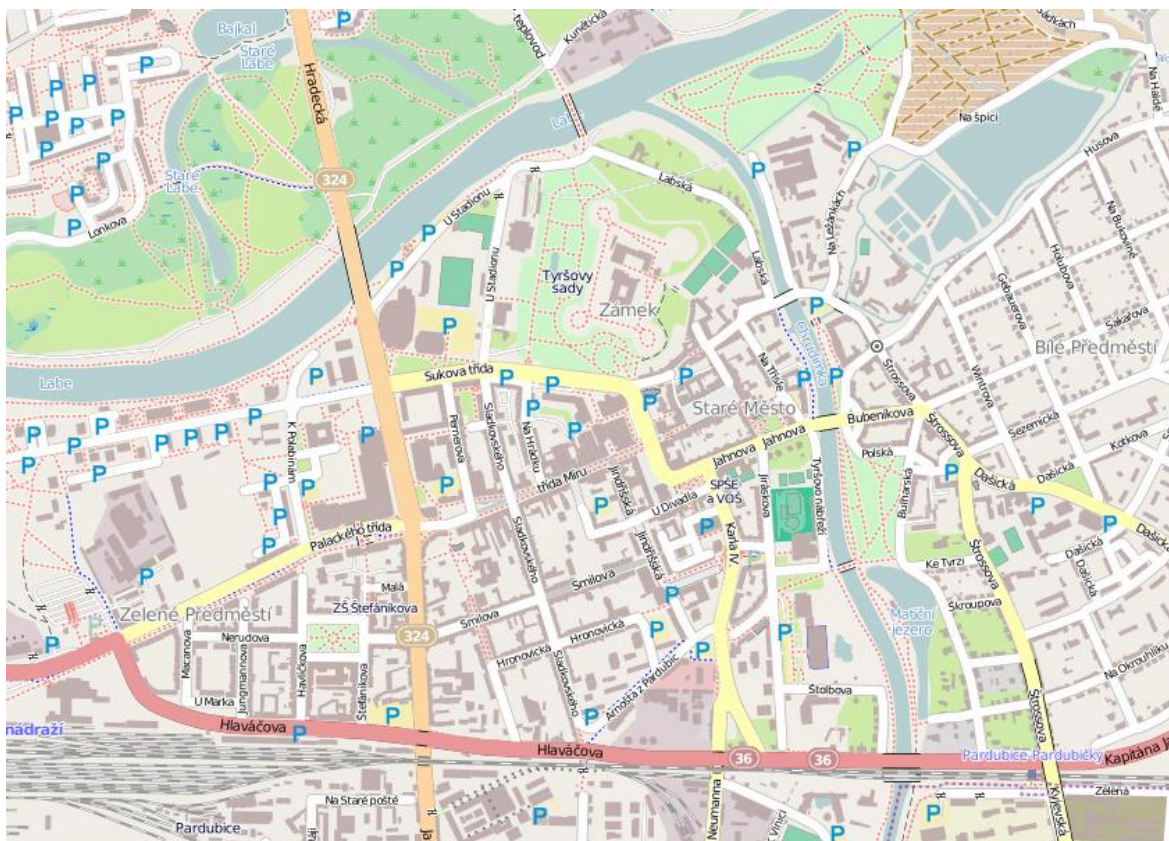
Abychom mohli mapy vizualizovat, je nutnost mít přístup k datovým podkladům takových map. Problémem, na který můžeme narazit, jsou licenční ujednání, která nám mohou zamezit použití takových podkladů v aplikaci. Obvykle podkladů, které jsou zpřístupněny zdarma je podstatně méně a otázkou je potom i jejich kvalita, nicméně OpenStreetMap je projekt, který právě tato kritéria splňuje.

3.1 OpenStreetMap

OpenStreetMap kolektivní spoluprací na koncepci Open source, jehož cílem je tvorba volně dostupných geografických dat a následně jejich vizualizace do podoby topologických map (např. silniční mapa, uliční mapa města, cyklomapa, atd.). Pro tvorbu geodat se jako podklad využívá záznamů z přijímačů GPS nebo jiné licenčně kompatibilní a zpravidla digitalizované mapy. Data jsou poskytována pod licencí „Open Database License“. Projekt OpenStreetMap je inspirován projekty jako Wikipedia, kdy umožňuje jednoduchou editaci dat, zaznamenává kompletní historii změn a výsledky práce jsou přístupné veřejnosti.

Cílem je vytvářet geodata pro celý svět, přičemž nejlépe je zmapovaná západní Evropa, kde projekt vznikl a USA. Zdrojem dat jsou od počátku projektu dobrovolníci, kteří systematicky mapují oblasti pomocí ručních GPS přijímačů. GPS data jsou poté zpracována na počítači a posléze nahrána do databáze OpenStreetMap. Mapovat lze při procházkách, na kole, v autě apod.

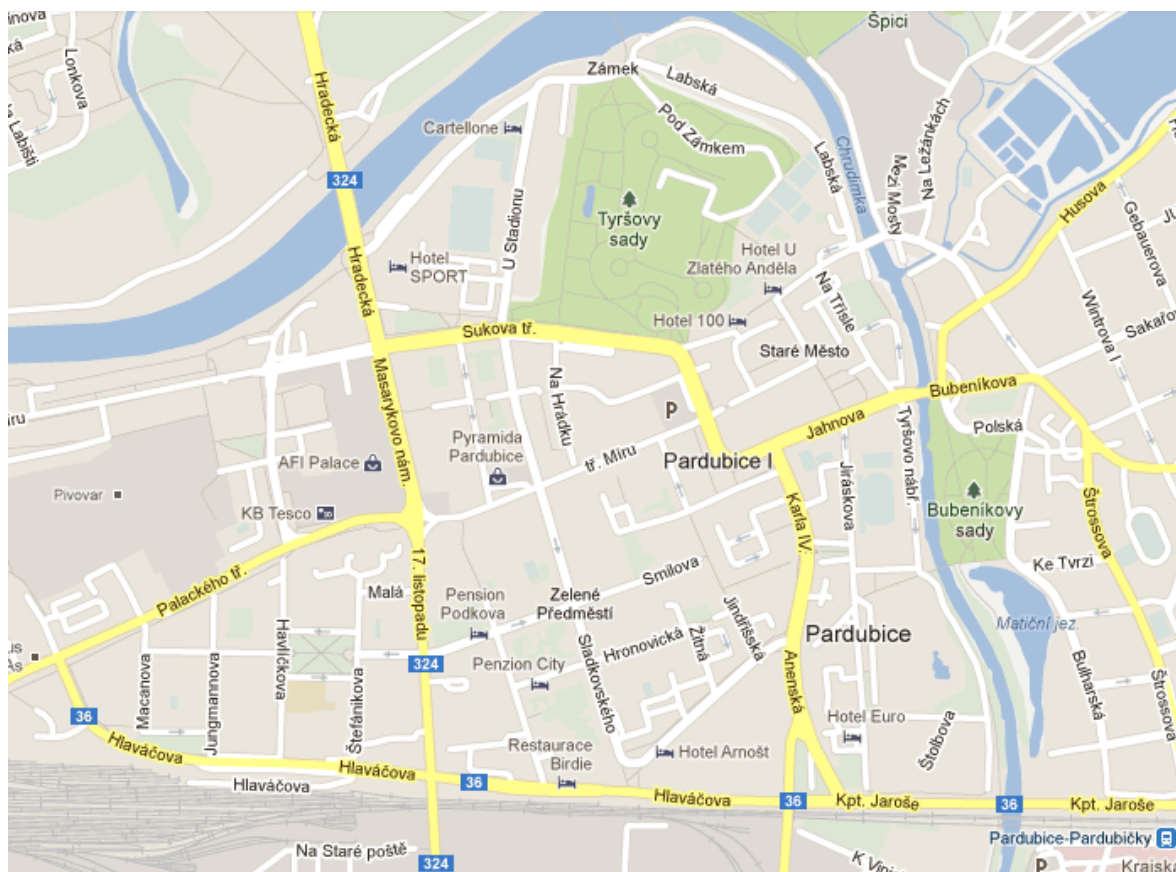
Nedávné zpřístupnění leteckých snímků a další dat z komerčních či veřejných zdrojů přispělo ke zrychlení a zpřesnění mapových podkladů, např. využití půdy. Projekt podpořila i řada vládních organizací poskytnutím satelitních snímků, katastrálních map apod., ale i některé komerční firmy, které se rozhodly část svých dat poskytnout pro potřeby projektu [9].



Obrázek 3 - Ukázka OpenStreetMap

3.2 Google Maps

Jinou variantou kvalitních a zdarma dostupných map jsou Google Maps od společnosti Google. Jedná se o technologii a internetovou aplikaci, která pohání mnoho mapových služeb včetně stránek Map Google. Na internetové stránky třetích stran se mapy vkládají pomocí Google Maps API. Tato služba nabízí např. mapy ulic, plánovač cest pro cestování pěšky, automobilem nebo veřejnou hromadnou dopravou a polohu mnoha podniků po celém světě. Google maps se mohou pyšnit satelitními snímky ve vysokém rozlišení mnoha městských oblastí v Kanadě, Spojených státech amerických, Spojeného království, Austrálii a mnoha dalších zemí, nicméně mnoho z nich jsou snímky letecké.



Obrázek 4 – Ukázka Google Maps

V roce 2006 Google představil Java aplikaci s názvem *Google Maps for Mobile*, která je určena pro provoz na jakémkoliv telefonu nebo zařízení založeném na Javě s mnoha funkcemi dostupných ve webové verzi aplikace. Od prosince roku 2008 je k dispozici pro následující platformy:

- Android
- iPhone OS
- Windows Mobile
- Nokia/Symbian
- Symbian OS
- BlackBerry
- Phones with Java-Platform
- Palm OS [10]

Pro zobrazení mapy v aplikaci je nutné získání klíče a nadefinovat ho v kódu aplikace. Je nutné získat zvlášť klíč pro testování aplikace a následně nový při publikaci aplikace.

Pro zobrazení mapy se vyžaduje připojení k datovým službám. Nicméně pokud Vám to daná aplikace dovolí, fungují i bez něj, avšak místo silnic, domů a dalších objektů zakreslených na mapě uvidíte jen mřížkovanou šedivou plochu.

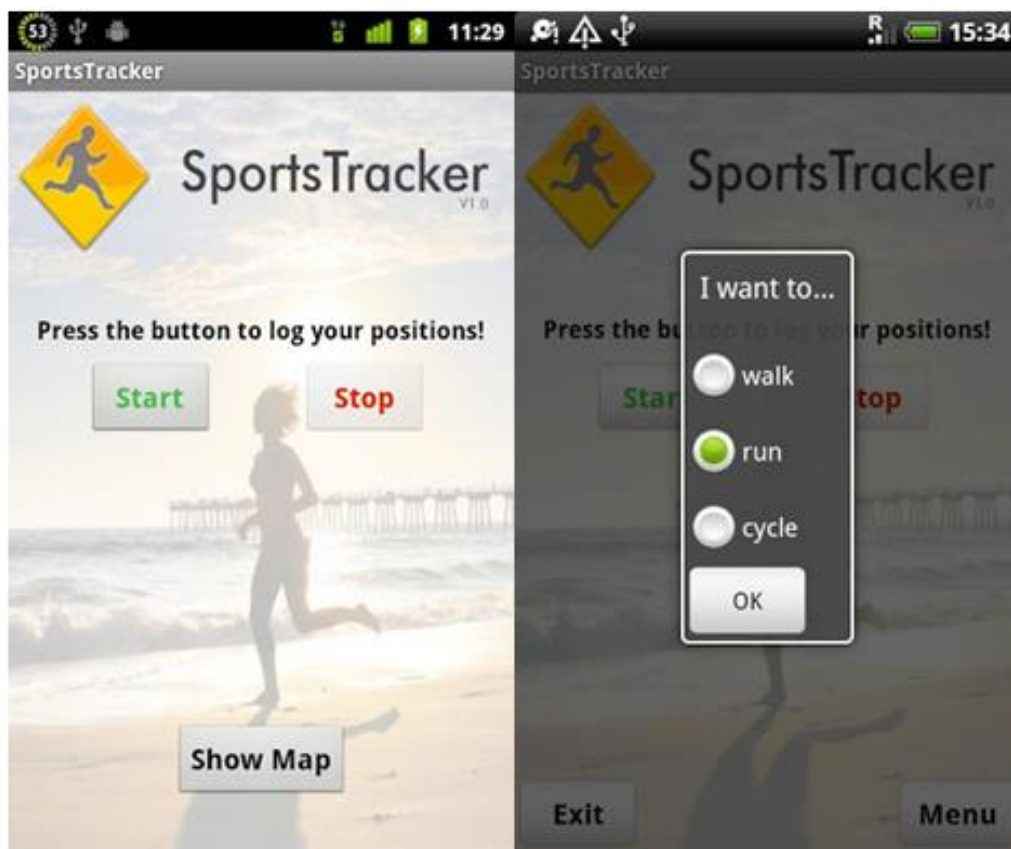
4 Dostupná řešení

V rámci práce bylo otestováno několik mobilních aplikací zabývajících se podobnou problematikou. Aplikace tohoto druhu jsou obvykle založeny na sledování ураžené vzdálenosti, výpočtech spálených kalorií při aktivitě, nadmořské výšce, rychlosti, času apod.

Některé aplikace jsou placené, zdarma nebo částečně zdarma, to znamená, že poskytují jen omezené množství informací o vykonaných aktivitách a pokud si uživatel přeje získávat pokročilejší statistiky o svých výkonech, je nucen za poplatek upgradovat svůj účet. Většina takových aplikací obvykle umožňuje sledovat prognózu výkonů z dlouhodobějšího hlediska, u některých to však není možné bez toho, aby si uživatel prostřednictvím internetu založil účet nebo se propojil s aplikací na sociální síti. Zkrátka u některých aplikací si sportovec bez datového připojení v zařízení nemá šanci nic zaznamenávat.

4.1 Sports Tracker

Sports Tracker je jednoduchou aplikací, která nenabízí mnoho možností. Vybrat si zde můžete v poněkud nevzhledném seznamu ze tří aktivit, a to chůzi, běh anebo cyklistiku. Její předností je, že není nutnost se nikde registrovat, není potřeba být tedy ani připojen k internetu. Na druhou stranu nenabízí nic extra. Jednoduše jen zaznamená trasu sportovce při jeho pohybové aktivitě.



Obrázek 5 – Ukázka z aplikace Sports Tracker

Na konci aktivity umožňuje uložit trasu do souboru na SD kartě s názvem aktuálního data a času za účelem pozdějšího zobrazení trasy v aplikaci Google Earth [11]. Nezaznamenává však ani rychlost při výkonu, ani čas, zkrátka žádné další informace.

4.2 Runstar™

Runstar™ je designově celkem pěkná aplikace zaměřená pouze na běh. Umožňuje sportovci nastavit si svůj měsíční cíl, kterého hodlá dosáhnout, sledovat jeho průběh a sdílet ho s přáteli prostřednictvím sociálních sítí.

Tato aplikace již zaznamenává více údajů o výkonu a zároveň je zobrazuje ve vzhledných barevných grafech. Uživatel si tedy přímo v aplikaci nechá vykreslit graf celkové vzdálenosti, průměrné rychlosti nebo celkového času v posledním měsíci, posledním roce nebo celkově za celou dobu užívání. Pro jednotlivé měsíce je zde celkem přehledně vytvořen seznam výkonů s jejich parametry. Těmi jsou čas, délka, datum a po kliknutí na detailnější informace i graf vývoje rychlosti a nadmořské výšky. Zajímavým údajem je zde průměrné tempo, to udává průměrnou hodnotu v minutách potřebnou ke zdolání jednoho kilometru. Naopak příliš nepraktická mi zde přijde možnost úpravy naběhané vzdálenosti, kterou se i po změně můžete „chlubit“ na sociálních sítích. Ke každému cvičení je potom dovoleno zobrazit mapu s vyznačenou trasou běhu.



Obrázek 6 - Ukázka rozhraní aplikace Runstar [11]

Runstar je příkladem aplikace, která je zdarma, avšak pokud chcete využívat všech implementovaných funkcí, jste nuceni si koupit plnou verzi. Ta nabízí:

- přednastavení max. času pro běh
- přednastavení vzdálenosti
- nastavení běhu na běžícím páse
- zvuková zpětná vazba
- upload cvičení na web

Jako největší nevýhodu této aplikace bych určil její specifickou pro jedinou pohybovou aktivitu. Aplikaci to sice umožňuje zaměřit se pouze na jeden druh sportu, nedá se však použít obecně [11].

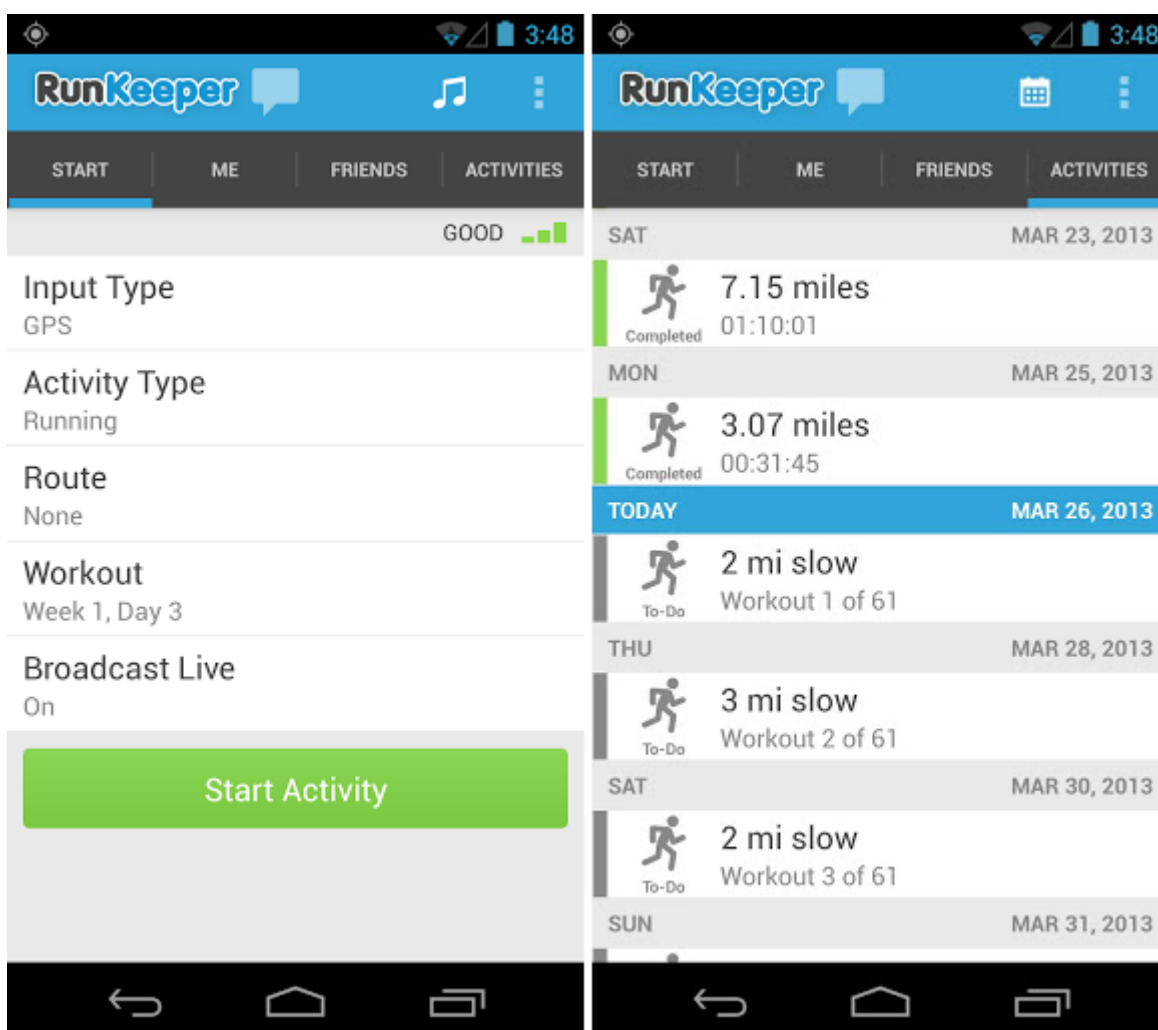


Obrázek 7 - Ukázka aplikace Runstar [11]

4.3 RunKeeper

RunKeeper je jednou z povedených, dobře hodnocených a poměrně oblíbených aplikací pro sledování pohybových aktivit. Hned v úvodu bez předchozí nutnosti registrace a přihlašování si uživatel vybere z rozmanitého seznamu aktivit např. běh, cyklistiku, horskou cyklistiku, chůzi, bruslení, lyžování, snowboarding, veslování, jízdu na invalidním vozíku apod. Může si vybrat vytvořené cvičení s určitou dobou zahřátí a závěrečným protažením a počtem opakování různě dlouhých a rychlých cvičení. Délku si může volit

v závislosti na čase nebo délce trati. Umožněno je však i vytvořit cvičení vlastní nebo zkrátka jen určit požadované tempo.

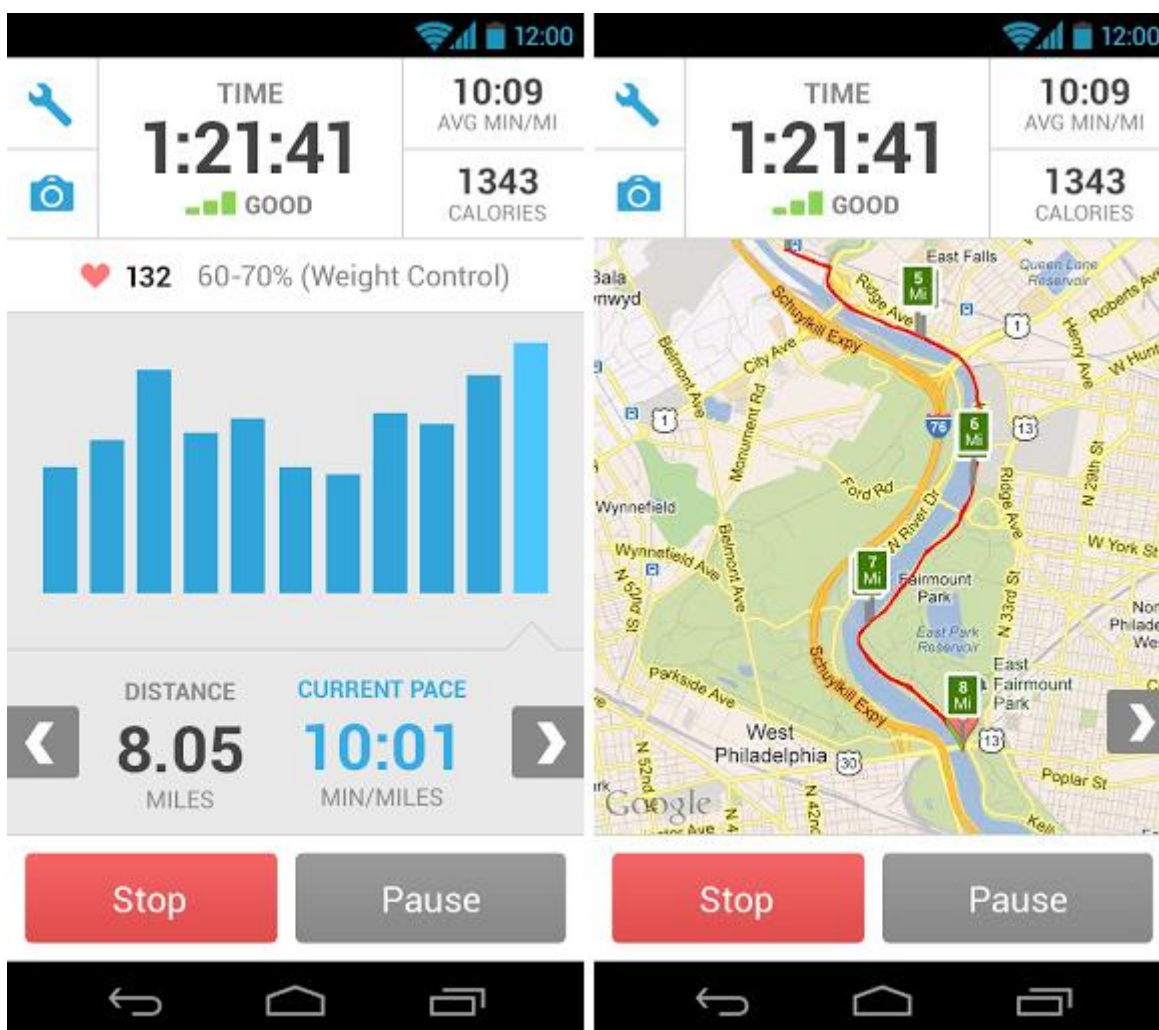


Obrázek 8 - Ukázka aplikace RunKeeper [11]

Během celého cvičení je sportovec provázen zvukovými instrukcemi např. o započítí různých částí cvičení a může kontrolovat, ve které části se právě nachází a kolik času nebo vzdálenosti zbývá do konce právě běžící fáze. Zároveň se nabízí možnost sledování trasy a pozastavení tréninku. Po ukončení aktivity je uživatel ihned vybídnut k uložení záznamu o tom, jak si vedl. O výsledku může informovat své okolí prostřednictvím sociálních sítí, to však až při spojení k internetové síti. RunKeeper tedy nevyžaduje datové připojení, ale pokud se pro něj uživatel rozhodne, může se spojit s komunitou sportovců využívající aplikaci, přidávat si přátele a sdílet s nimi své výsledky. To je možné i prostřednictvím sociální sítě Twitter a Facebook. Dosažené výkony se buď „zahodí“, nebo uloží a poté se zobrazují spolu se zaznamenanými záznamy o celkovém čase, tempu v částech cvičení a průměrném tempu, kaloriích, celkové vzdálenosti a mapě.

Mnoho uživatelů jistě ocení možnosti:

- volba frekvence a druh zvukových podnětů
- nastavení upomínek na cvičení
- online podpora monitoringu srdce
- automatické pozastavení
- online podpora poslechu a ovládání hudby během aktivity

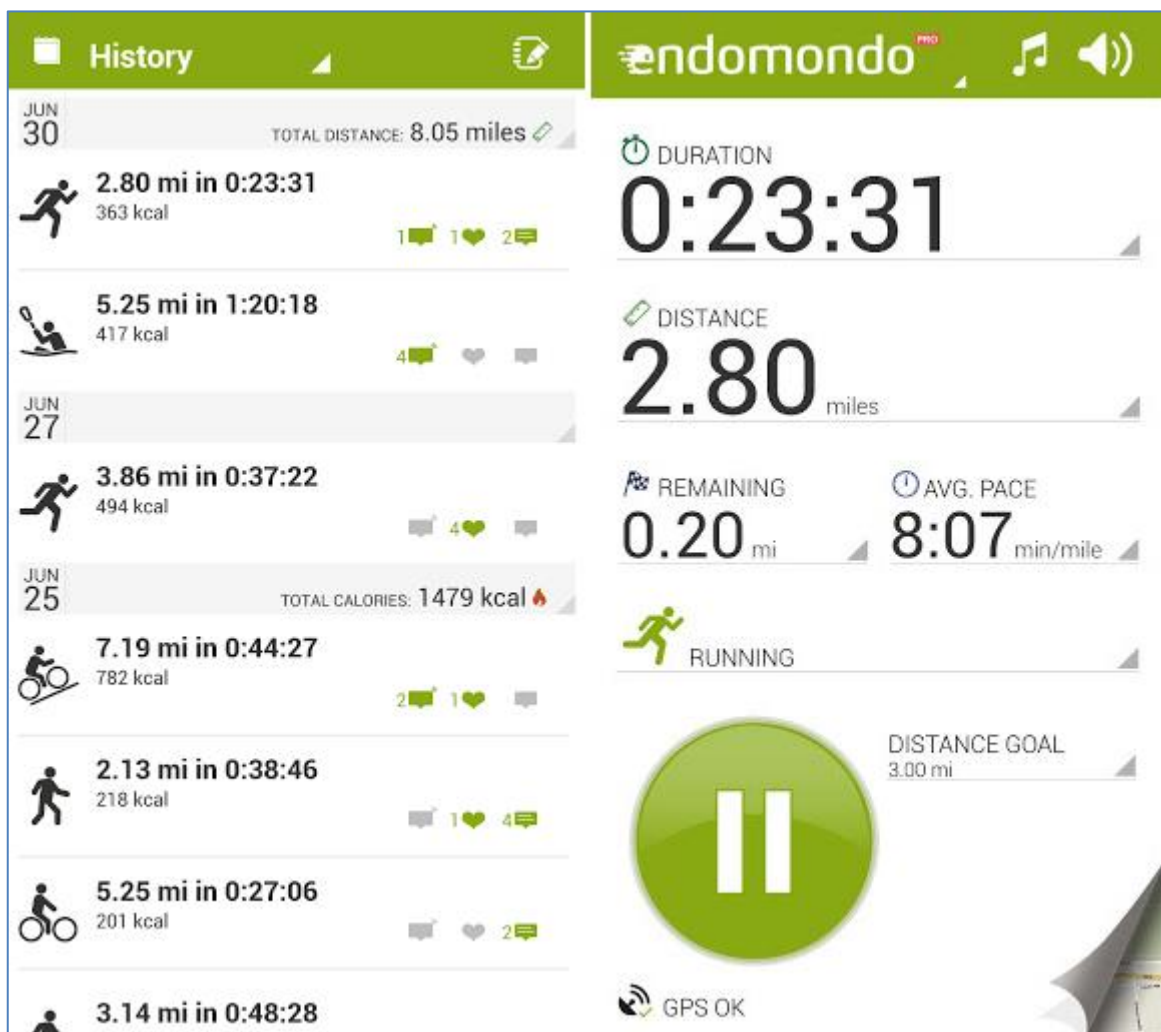


Obrázek 9 - Ukázka aplikace RunKeeper [11]

Přes všechny šikovně provedené části aplikace a možnosti, které nabízí, zde schází grafy k jednotlivým měřením a vypočteným údajům, které by efektivně odrážely sportovcovi výkony a zlepšení. Sloupcový graf udávající tempo během výkonu je jediný grafický údaj vypovídající o průběhu cvičení [11].

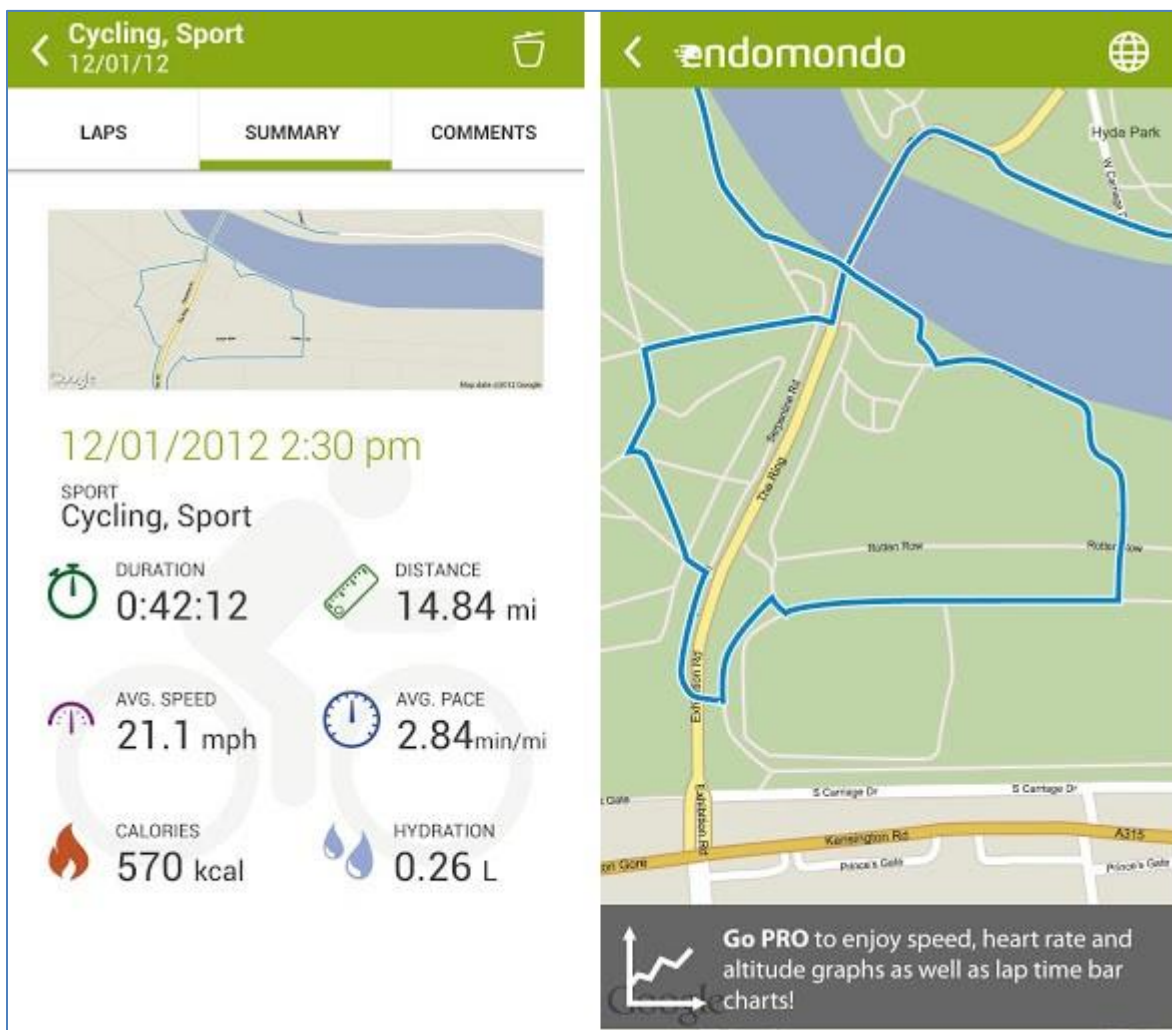
4.4 Endomondo

Mezi nejlépe hodnocené aplikace pro vzdálenostně orientované aktivity patří Endomondo. Nabízí zaznamenávat výkony v mnoha aktivitách jako běh, cyklistika, chůze, lezení, běh na běžeckém páse, skateboarding a mnoha dalších. Tato aplikace se opět dělí podle toho, zda máte placenou nebo bezplatnou verzi.



Obrázek 10 - Ukázka aplikace Endomondo [11]

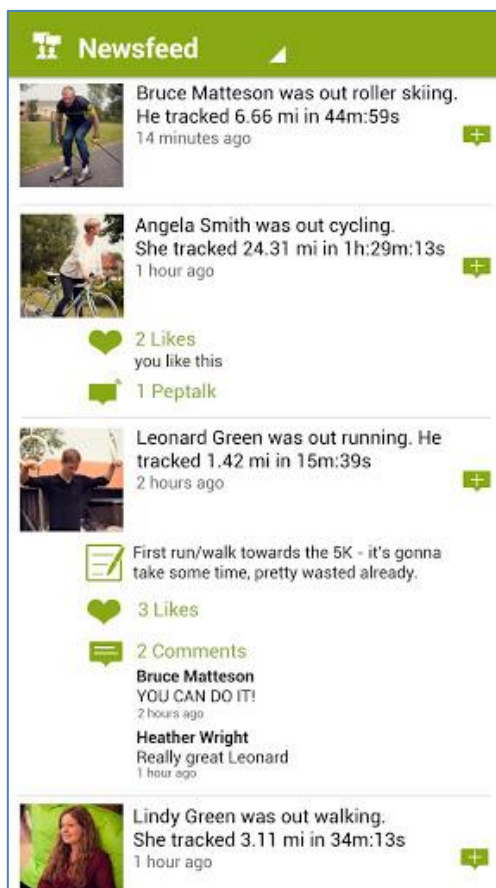
Bezplatná verze nabízí sledovat všechny venkovní sporty včetně jejich trvání, vzdálenosti, rychlosti a kalorií. Dále přidávat cvičení manuálně. Zde jde o cvičení, která se vykonávají na místě jako běh na běžícím páse nebo zátěžová posilovací cvičení. Součástí je také možnost zvukového povzbuzování během každého kilometru popř. míle cvičení. Zadarmo se uživateli dostane i klasicky sledování mapy trasy, prohlížení historie výkonů a mezičasů.



Obrázek 11 - Ukázka aplikace Endomondo č. 2 [11]

Aplikace disponuje i mnoha drobnými vylepšeními jako nastavení zvukové hlášky, jakmile sportovec dosáhne určité vzdálenosti nebo udržování hudebních seznamů ke cvičením. Vhodnou funkcí je i automatické pozastavení v případě, že se uživatel přestane pohybovat nebo např. odpočítávání. V neposlední řadě potěší i implementace výběru nejzajímavějších záznamů na hlavní obrazovce.

Prostřednictvím sociální sítě je možné umístit cvičení na „Facebook Timeline“ a zároveň vidět poslední cvičení přátel. Díky tomuto propojení je k dispozici např. pokoření nejlepšího času kamaráda a soutěžení na konkrétní trase.

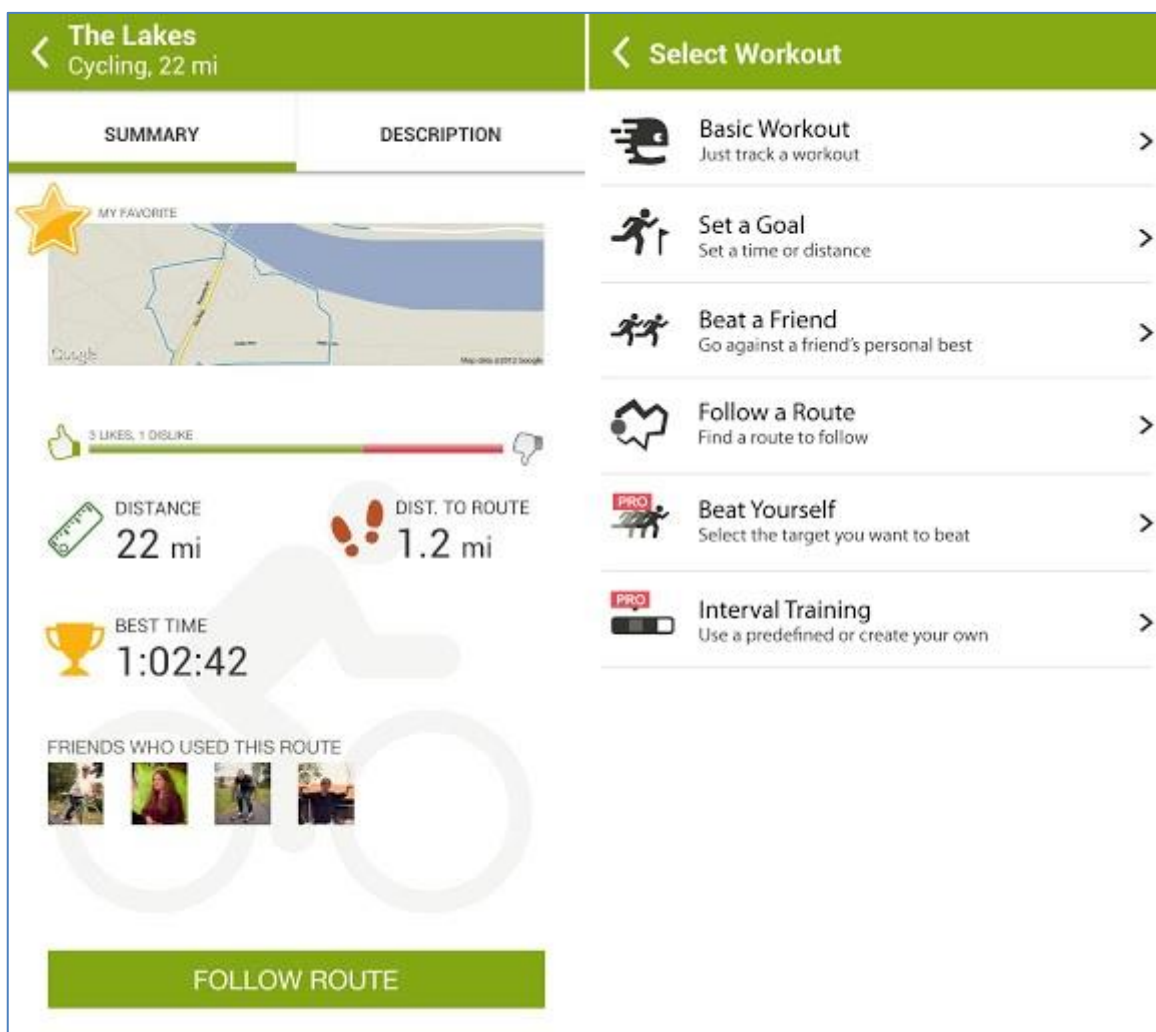


Obrázek 12 - Ukázka aplikace Endomondo č. 3 [11]

Špatnou zprávou pro ty, kteří dávají přednost bezplatným aplikacím je, že až placená verze nabízí:

- intervalový trénink
- grafy
- mód pro nízkou spotřebu
- krokoměr
- možnost závodit proti sobě, času nebo kaloriím

Intervalový trénink nabízí výběr ze tří programů vytvoření vlastního s doprovodem zvukových pokynů. Standardní grafy bývají v podobných aplikacích dostupné bezplatně, v Endomondo ne. Přitom neposkytují nic navíc oproti grafům konkurence. Je zde možnost grafu s časem trasy, rychlostí, nadmořskou výškou atp. Zajímavě už zní spuštění pohotovostního režimu pro úsporu baterie. Novinkou, která by měla fungovat i ve vnitřním prostředí bez signálu GPS je krokoměr. Ten, jak již název napovídá, slouží pro měření počtu kroků. Vhodným vylepšením pro ty, jež by rádi korigovali množství zvukového povzbuzení je jeho přizpůsobení a výběr druhu informací. Nakonec mnozí ocení výběr kalorického nebo časového cíle před cvičením [11].



Obrázek 13 - Ukázka aplikace Endomondo č. 4 [11]

Je to však jedna z aplikací, ve které se bez registrace a přihlášení dál nedostanete, takže pro uživatele bez datových služeb nepoužitelné. Navíc dle zpětné vazby se právě kvůli internetovému připojení vyskytlo dost chyb, které mnoha sportovcům mohlo cvičení znepříjemnit.

5 Návrh a implementace aplikace

Přes všechny výhody a nedostatky zmíněných aplikací považuji za vhodné vytvoření aplikace vlastní z následujících důvodů:

- cena
- komplexnost
- rozsáhlost
- uživatelská přívětivost

Cílem je tedy nalézt vhodný kompromis mezi existujícími aplikacemi, který bude zdarma všem dostupný a nebude se specializovat pouze na vybranou aktivitu, ale nabídne sledování různých běžných pohybových aktivit. Aplikace by neměla být ani příliš rozsáhlá, ani moc jednoduchá. Základem by mělo být sledování důležitých aspektů, které se pomocí Android platformy nabízí a dále vypočítávat a porovnávat základní hodnoty související s trasovacími aktivitami jako jsou výdej kalorií, čas, vzdálenost, nadmořská výška a rychlost. Vizuální a statistické vyhodnocení výsledků bude provádět až desktopová aplikace, zejména z hlediska dodržení zadání práce. Ovládání obou částí aplikace by mělo být uživatelsky přívětivé, a to s důrazem na jednoduchost.

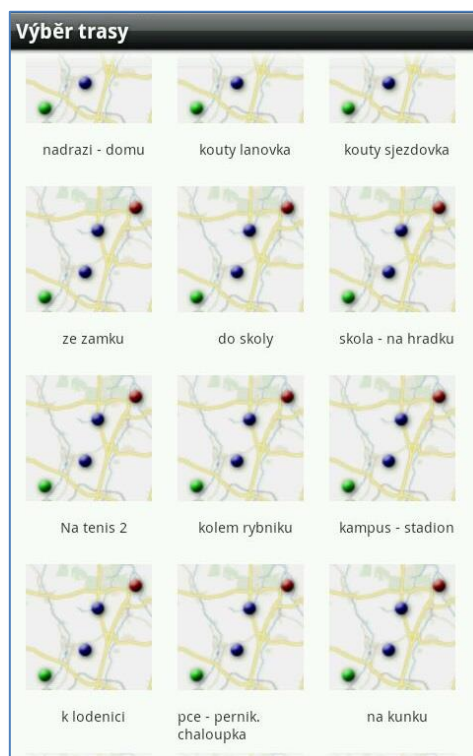
5.1 Mobilní aplikace

Aplikace spolu s mobilním zařízením zastupují funkci snímače. Jejich hlavním úkolem je tedy zaznamenat data o výkonu sportovce spolu s informacemi o vybrané aktivitě, trase popř. zaznamenat trasu novou. Vývoj probíhal v programovacím prostředí Eclipse a pro testování byl použit HTC Desire HD.

Úvodní obrazovkou, ještě před započítáním měření, je výzva uživatele k výběru aktivity a poté trasy. Tyto položky jsou uspořádány v ovládací prvku *GridView*. Každý z výběrů představuje jinou obrazovku resp. jinou třídu dědicí od třídy *Activity*. Součástí layoutu výběru trasy je menu, které lze vyvolat použitím tlačítek přístroje, a to nabízí volbu „Nová trasa“, které uživatele přeměruje také na obrazovku mapy, ale bez bodů trati.



Obrázek 14 - Ukázka výběru aktivity v aplikaci



Obrázek 15 - Ukázka výběru trasy

5.1.1 Získání polohy

Po výběru nezbytných parametrů se dostáváme do jádra aplikace. Tím je obrazovka s mapou a tedy třída, která dědí vlastnosti třídy *MapActivity*, což umožňuje zobrazení mapy Google Maps, a implementuje metody třídy *LocationListener*. Nejdůležitější a prakticky jedinou využívanou metodou, kterou *LocationListener* implementuje je *onLocationChanged(Location location)*, která reaguje na změnu GPS souřadnic. Parametrem této metody je nový aktuální bod typu *Location*. Z proměnné tohoto typu lze získat různá aktuální data jako zeměpisnou šířku a délku, nadmořskou výšku, rychlost apod. a právě informace o nadmořské výšce a rychlosti jsou v přetížené metodě *onLocationChanged(Location location)* vypisovány do textového pole *TextLabel* nad mapou.

Do tohoto stavu se však aplikace nedostane, pokud nebude povolena funkce GPS. Kontrola její aktivity probíhá hned po vytvoření instance třídy a při negativním výsledku se vytvoří instance *AlertDialog.Builder*, která na tuto skutečnost upozorní uživatele dialogovým oknem a nabídne mu přechod do nastavení zařízení, kde má možnost GPS aktivovat.

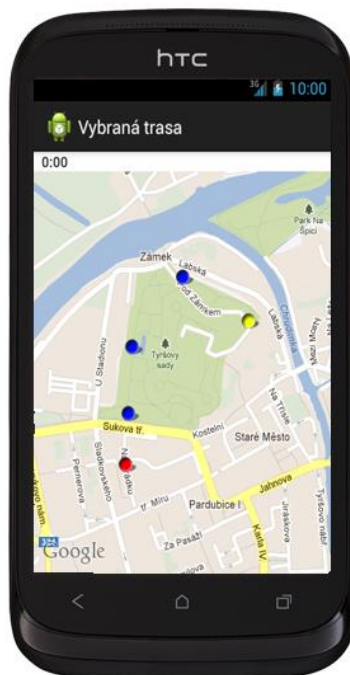
Další nezbytnou podmínkou zobrazení informací ohledně aktuální pozice je nalezení GPS signálu. To může obvykle trvat i několik minut. Jakmile se tak stane, aktuální nalezená pozice se na mapě zobrazí v podobě světle modrého pulsujícího puntíku s šedivým lemováním a poloprůhledným kruhem okolo, značící možnou odchylku. Pokud se stane, že se uživatel na mapě „ztratí“, existuje zde funkcionalita rychlého zaměření. To lze vyvolat tlačítkem v menu „Moje pozice“. Následně se využije tzv. *MapController* a jeho metoda *animateTo(Location location)*. Obdobnou funkci zastává položka menu „Trať“, jež má za cíl se naopak po mapě přesunout na místo s tratí.

5.1.2 Měření výkonů

Pro vykreslení tratě se používá třída *MyOverlay*, která dědí od třídy *ItemizedOverlay<OverlayItem>*. Jedná se vlastně o seznam prvků tzv. *OverlayItem*, které překrývají základní mapu Google Maps.

Každá trať je v aplikaci dána body. Jejich počet je závislý na délce trati. Při tvorbě nové trati se na startu vytvoří první tzv. startovní bod označený žlutým puntíkem. Každý další bod je potom označen tmavě modrou barvou s výjimkou posledního červeného puntíku. Nové body přibývají, jakmile zařízení zaznamená vzdálenost 100 metrů od posledního vyznačeného bodu. Tato hodnota byla experimentálně zvolena na základě testování při aktivitách, jako je chůze a běh, a to v poměrně „zatáčkovitém“ prostředí, kde by při větší vzdálenosti mohlo dojít k možnosti zkracování si tratě při opakovaném výkonu.

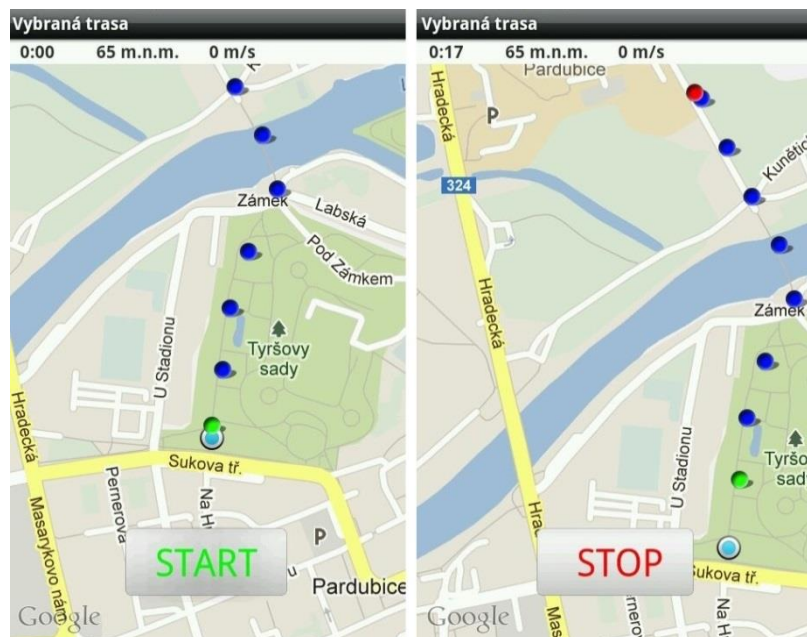
Jelikož některé tratě nemusí být přímočaré a tudíž nemusí být hned jasné, který modrý bod bude následovat, bylo v aplikaci implementováno zabarvení následujícího bodu tratě na zelenou barvu. V praxi to vypadá tak, že startovní bod popř. každý další, kterého sportovec dosáhne, zmizí a následující se zbarví na zeleno, aby bylo jasné, kam dál pokračovat.



Obrázek 16 - Ukázka trati

Společně s vyznačováním bodů dochází k ukládání průběžných dat o rychlosti, čase, GPS souřadnicích, nadmořské délce a délce trati. Pro ukládání těchto hodnot se využívá datová struktura *ArrayList*, a to zejména z důvodu, že tuto datovou strukturu není potřeba dopředu inicializovat na pevný počet prvků. Není tedy potřeba realokace paměti proměnné. K zjištění vzdálenosti mezi dvěma místy, zde spolehlivě slouží metoda *distanceTo(Location location)*, přičemž k ověření dochází po uplynutí 1000 milisekund. Pro tuto funkci je nastaven *Handler*, jež kontrolu vyvolá metodou *postDelayed(Runnable r, long delayMillis)*.

K odstartování aktivity slouží tlačítko „START“, to se zobrazuje v dolní části mapy, jakmile je nalezena aktuální GPS pozice. V případě, že chce uživatel vykonávat pohybovou aktivitu na již vytvořené trati, musí se ještě nacházet v blízkém okolí startovního bodu. Tolerance je z důvodu nepřesnosti určení pozice, ale i z důvodu lepšího nalezení startu, nastavena na 15 metrů. Tato tolerance je nastavena i na další body. Důvod je obdobný, nepřesnost určení GPS pozice a lepší míra zaručení průběhu bodem. Pokud by uživateli nestačil ani tento 15 metrový poloměr, může se stát, že ačkoliv okolím místa „proběhl“ nebude započítáno a musel by se vrátit zpět, aby se mu počítaly i další body. Takto je ošetřeno, aby nedocházelo k výraznému zkracování trati.



Obrázek 17 - Ukázka aplikace před a po startu aktivity

Opačnou funkci nese tlačítko „STOP“. Slouží k předčasnému ukončení aktivity nebo k ukončení zaznamenávání nové trati a následnému vybitnutí k uložení. Před uložení, je nutné zadat jediný parametr a to název trati. K tomuto účelu je implementováno dialogové okno se vstupním textovým polem a tlačítky pro potvrzení popř. přerušení. Následně se zobrazuje dialogové okno s informacemi o průběhu v rámci daného výkonu. Dialog se tedy objeví i po dosažení cíle na již vytvořené trati. Obsahuje základní informace, a to:

- celkový čas výkonu
- vzdálenost v metrech
- průměrná rychlost

5.1.3 Uložení dat

Ještě před tím se data uloží do souborů. Pro ukládání se využívají prosté textové soubory, které se vytvoří společně s adresáři na SD kartě zařízení. V kořenovém adresáři se při prvním použití vytvoří adresář *KondiMetr* s podadresáři *trasy* a *vykony*. Soubory jsou dva, přičemž jeden je určen pro trasy a druhý pro výkony. Data se do souboru *trasy.dat* ukládá ve specifické struktuře, která je důležitá zejména pro pozdější čtení ze souboru (Obrázek 18 - Ukázka struktury souboru *trasy.dat*). Nová trasa se vždy ukládá na konec souboru, přičemž odděleny jsou od sebe prázdným řádkem. Jako první je na samostatném řádku uložen název trasy následovaný celkovou délkou trasy. Poté je vždy na samostatném řádku dvojice GPS souřadnic a nadmořská výška. Všechny tři prvky jsou odděleny středníkem.

```

Z tenisu
1345
15.773069;50.046722;269
15.773883;50.045593;269
15.774566;50.044783;277
15.775184;50.043946;260
15.774726;50.04305;254
15.774101;50.042217;270
15.7739;50.041318;258
15.773715;50.040416;253
15.773562;50.039511;264
15.773404;50.038601;263
15.77348;50.038057;262

Na hradku - nadrazi
1598
15.773756;50.0379;280
15.769095;50.036964;286
15.767342;50.036034;291
15.76624;50.03546;299

```

Obrázek 18 - Ukázka struktury souboru *trasy.dat*

Podobně je tomu u výkonů. Ty se v souboru *vykon.dat* ukládají opět na jeho konec a opět jsou od sebe odděleny prázdným řádkem (Obrázek 19 - Ukázka struktury souboru *vykon.dat*). U každého je nejprve uveden název tratě, ke které se vztahuje. Na dalším řádku je identifikační číslo aktivity vybrané před spuštěním měření. Třetí řádek obsahuje datum a čas v době, kdy došlo k ukončení měření popř. dosažení cíle. Na zbylých je potom vždy čtveřice hodnot zaznamenaných v jednotlivých bodech trati. Jsou jimi celkový čas od startu, průměrná rychlost v posledním měřeném úseku, délka úseku v metrech a nadmořská výška v tomto bodě. Odděleny jsou opět středníkem.

```

do skoly
2
2013-09-03 19:29:25
0:00;0.0;0;276.0
1:00;1.0;116;271.0
2:05;1.0;126;275.0
3:13;0.0;112;278.0
4:14;1.0;125;276.0
5:25;0.0;119;275.0
6:37;1.0;125;282.0
7:12;1.0;53;273.0

kolem rybniku
2
2013-03-17 15:43:17
0:00;0.0;0;261
2:36;0.0;195;256

```

Obrázek 19 - Ukázka struktury souboru *vykon.dat*

Mobilní aplikace je tedy poměrně jednoduchá, přesto zastává důležitou úlohu snímání dat o trase a výkonu sportovní aktivity a ty zaznamenává pro pozdější využití v desktopové aplikaci. Klade důraz na možnost výběru stejných tras pro stejné nebo rozdílné aktivity a informovat především o tom podstatném, a to poloze, času, rychlosti a vzdálenosti. Daří se jí zaznamenat vše důležité pro zpracování a vyhodnocení výsledků.

5.2 Desktopová aplikace

Pro implementaci desktopové aplikace byl zvolen programovací jazyk Java. Databázovou část zde zastává Oracle Database 10g Express Edition. Pro přístup k databázovému serveru bylo využito ovladače JDBC, který představuje jednotné rozhraní pro přístup k relačním databázím. Tento ovladač představuje Java třídu poskytnutou společností Oracle, přičemž jeho výhodou je, že klient nemusí být nijak složitě konfigurován a nejsou nutné žádné lokální klientské instalace. Jako vývojové prostředí byl zvolen NetBeans IDE 7.2.1. K závěrečnému sestavení aplikace byl použit nástroj Apache Maven, který slouží pro správu, řízení a automatizaci „buildů“ aplikací. Důvodem použití tohoto nástroje bylo sestavení aplikace jako jednoho .jar souboru včetně použitých externích knihoven.

5.2.1 Datový model

Použité externí knihovny jsou nezbytné pro správné fungování aplikace, zejména pak pro zobrazování grafů a map, ale i pro přístup k databázi. Databáze zde představuje hlavní uložisko dat. Jsou v ní uloženy jednotlivé trasy spolu s informací o její délce názvu a přezdívce uživatele, který ji vytvořil. Všechna tato data se přebírají z mobilní aplikace. Atribut Kdo nesoucí přezdívku uživatele je zde v této fázi bezvýznamný. Jeho využití je nastíněno spíše do budoucna, kdy by bylo umožněno absolvovat tratě jiných uživatelů.

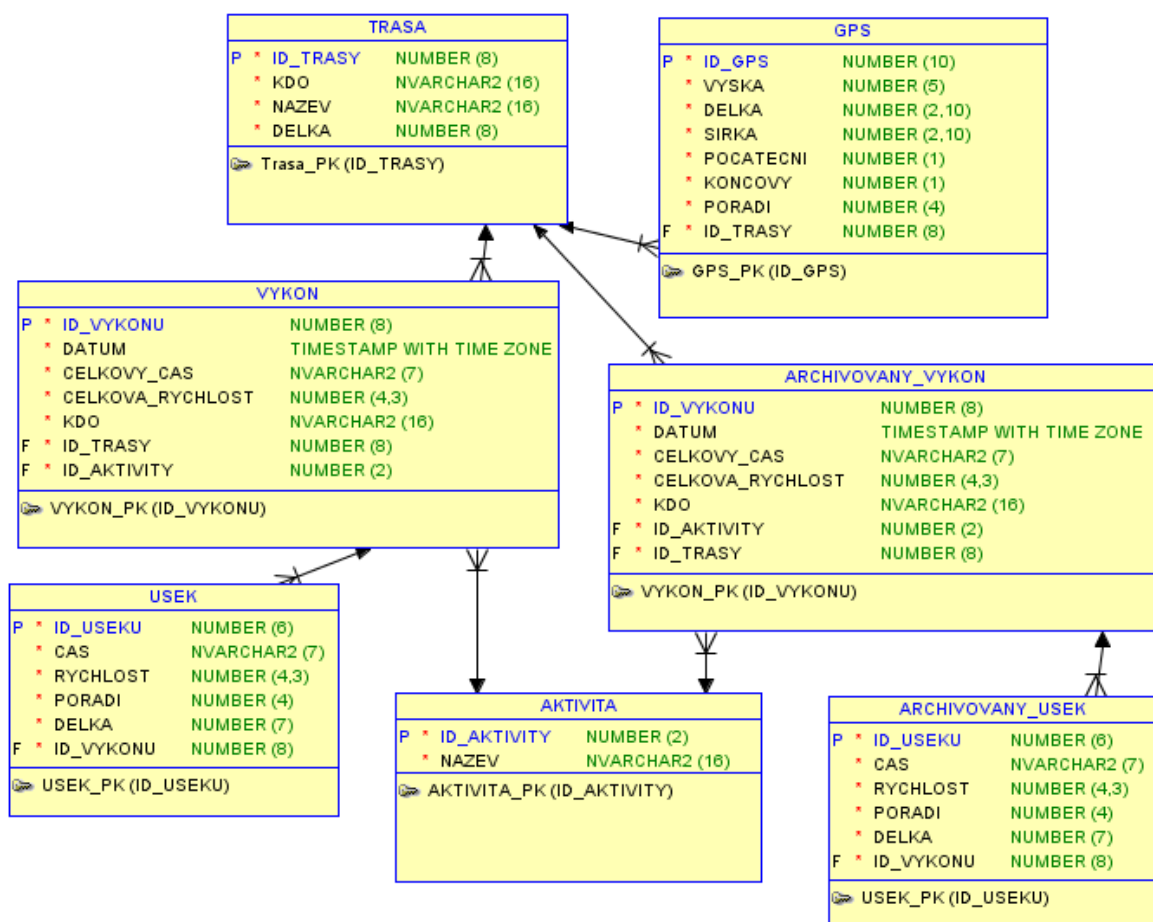
Každá trasa se skládá ze dvou a více GPS souřadnic. K jejich uchování je k dispozici tabulka GPS. Krom identifikátoru (ID_Gps) obsahuje atributy nesoucí informace o nadmořské výšce (Vyska), zeměpisné délce (Delka) a šířce (Sirka), pořadí (Poradi) v rámci tratě a další dva pro jednoznačné označení prvního (Pocatecni) a posledního (Koncovy) bodu tratě.

V rámci jedné trasy je možné evidovat více výkonů. Tabulka VYKON ukládá datum dokončení výkonu (Datum), jeho celkový čas (Celkovy_cas), průměrnou rychlost v průběhu celého výkonu (Celkova_rychlost) a atribut s přezdívkou sportovce (Kdo), který aktivitu absolvoval. Poslední sloupec tabulky má stejné využití, jako výše zmíněný obdobný atribut v tabulce tras.

Pro uchování typu aktivity je určena tabulka AKTIVITA, které obsahuje sloupce ID_Aktivity a Nazev.

USEK je důležitou tabulkou, která obsahuje data o jednotlivých úsecích resp. bodech výkonu. Každý se vyznačuje identifikačním číslem (ID_Useku), celkovým časem na konci úseku (Cas), průměrnou rychlostí v rámci úseku (Rychlost), pořadím (Poradi) a jeho délkou (Delka). Jedná se vlastně o data přenesená z aplikace na mobilním zařízení.

Tabulky ARCHIVOVANY_VYKON a ARCHIVOVANY_USEK jsou obdobou tabulek VYKON a USEK. Jejich význam je pro uchování uživatelem odebraných dat v aplikaci.



Obrázek 20 - ER diagram databáze aplikace

O přístup do databáze se stará třída *DatabaseUtil*, která nejprve naváže spojení metodou *getConnection()*. Zde je využita knihovna ovladače JDBC od společnosti Oracle. Navázání spojení se skládá z několika částí. Nejprve se nahraje a registruje ovladač. K tomu slouží příkaz *Class.forName("oracle.jdbc.OracleDriver")*. Ten je umístěn v bloku *try{}*, který vyvolá výjimku v případě, že systém nedokázal třídu ovladače najít. Registrace je vázána na *DriverManager*, což je třída s metodami pro správu a práci s JDBC. Představuje vrstvu mezi kódem a JDBC ovladačem, která ve chvíli provedení předešlého příkazu *forName()* vykoná registraci. To se však děje na pozadí v rámci třídy ovladače. Další částí je samotné spojení s databází prováděné metodou *DriveManager.getConnection()* se třemi parametry:

- databázová URL začínající vždy řetězcem „jdbc.“
- přihlašovací jméno
- heslo

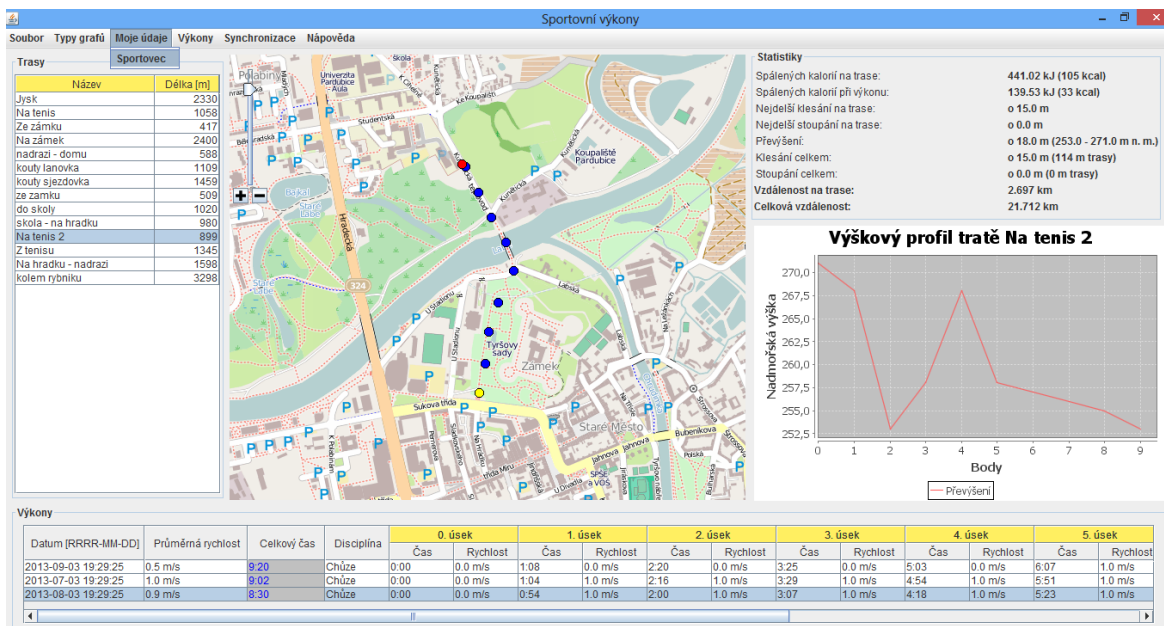
Jakmile je spojení s databází navázáno, je možné vykonávat datové operace. Určena je k tomu třída *Statement*, poskytnutá vytvořeným objektem *Connection*. Instance této třídy potom umožňuje posílání SQL příkazů databázi, kde jsou zpracovány, a je zaslána příslušná odpověď. Třída obsahuje velké množství metod jako *execute()*, *executeQuery()*, *executeUpdate()*, přičemž v aplikaci je obvykle využito pro vykonání dotazů metody *execute()*. Výsledky dotazů jsou potom objekty třídy *ResultSet*, který už obsahuje skutečná data, které se následně dají získat např. v cyklu *while* a metodami *getInt()*, *getString()*, apod. v závislosti na datovém typu proměnné. Pro posunutí na další pozici výsledku dotazu se využívá metody *next()*.

Import dat do databáze probíhá prostřednictvím aplikace. Je nutné, aby bylo zařízení kabelem připojeno a bylo možné přistupovat na SD kartu. Jakmile je toto zajištěno, stačí jen v menu aplikace vybrat volbu „Synchronizace“ a „Synchronizovat“. Pokud dojde k chybě, aplikace upozorní uživatele dialogovým oknem. V opačném případě se vyhledají na SD kartě soubory *trasy.dat* a *vykon.dat* a projde se jejich obsah. Jako první přijdou na řadu trasy. Načte se obsah souboru do bufferu a rozdělí se po řádcích. Poprvé se soubor projde, aby se zjistil počet tras, a z kolika se skládají bodů. Podle toho se ve druhé fázi algoritmu inicializují pole a může začít druhý průchod, ve kterém již dochází k samotnému získávání dat a ukládání do nově vytvořených instancí tříd GPS a Trasa. Těmito instancemi se zároveň naplňují seznamy, které jsou následně v cyklu zpracovány. Před pokusem o vložení do databáze se provede dotaz, zda v databázi již není trasa s identickým názvem a délkou trati. Pokud ano, tak je evidentní, že bychom se pokoušeli do databáze vložit duplicitní data, a proto se tento prvek seznamu vynechá. Jestliže se ukáže, že se jedná o novou trasu, dojde k jejímu uložení a poté i k uložení dat o GPS souřadnicích této trasy do tabulky GPS.

Podobně je tomu u výkonů, s tím rozdílem, že se ukládá do tabulek VYKON a USEK. Pro kontrolu duplicit zde stačí zjistit, jestli v databázi již není záznam se stejným datem.

5.2.2 Komponenty aplikace

Grafické uživatelské rozhraní je sestaveno pomocí aplikačního rozhraní knihovny Swing. Je to knihovna uživatelských prvků na platformě Java pomocí níž je možno vytvářet okna, dialogy, tlačítka, seznamy, rámečky atp. Než budou popsány jednotlivé části aplikace, na následujícím obrázku je možné nahlédnout na hlavní okno výsledné desktopové části práce.



Obrázek 21 - Aplikace jako celek

Součástí knihovny Swing je i třída *JTable*, která umožňuje zobrazovat a editovat data ve formě tabulky. Za její pomoci byly zobrazeny tabulky Trasy a Výkony. Komponenta *JTable* stejně jako všechny prvky uživatelského rozhraní ve Swingu využívá princip MVC. Základním principem architektury MVC je rozdělení datového modelu aplikace, uživatelského rozhraní a řídicí logiky do nezávislých komponent. Důsledkem je možnost snadné výměny některé z těchto tří komponent bez vlivu na ostatní dvě komponenty, vyšší flexibilita, snazší údržba atp.

Krom vlastní třídy *JTable*, která slouží jako ovladač tzv. „controller“, zde figuruje několik dalších komponent. Příkladem může být *TableModel*, který reprezentuje a poskytuje data, která jsou v tabulce zobrazena. Pro tabulky tras a výkonů to jsou třídy *TrasaModel* a *RozsireneVykonyModel*, které rozšiřují abstraktní třídu *AbstractTableModel*. Jejich implementace spočívá v překrytí abstraktních metod *getRowCount()*, *getColumnCount()*, *getValueAt()* a přizpůsobení pro účely aplikace. Navíc pro přizpůsobení názvu sloupce je překryta i metoda *getColumnName()*.

Datum [RRRR-MM-DD]	Průměrná rychlost	Celkový čas	Disciplína	0. úsek		1. úsek	
				Čas	Rychlost	Čas	Rychlost
2013-09-03 19:29:25	0.5 m/s	9:20	Chůze	0:00	0.0 m/s	1:08	0.0 m/s
2013-07-03 19:29:25	1.0 m/s	9:02	Chůze	0:00	0.0 m/s	1:04	1.0 m/s
2013-08-03 19:29:25	0.9 m/s	8:30	Chůze	0:00	0.0 m/s	0:54	1.0 m/s

Obrázek 22 - Tabulka výkonů

Třída *ColumnGroup* slouží pro vytvoření skupin sloupců a je potom možné např. vytvořit společné hlavičky několika sloupců. V případě aplikace bylo právě vhodné spojit hlavičky

času a rychlosti týkající se stejného úseku tratě. Tato úprava však nijak neomezuje možnost řazení záznamů dle jejich hodnot v jednom sloupci.

Další komponentou zobrazující se v aplikaci je pole statistik. Jedná se o tabulku vypočtených a sumarizovaných hodnot týkající se obvykle vybrané trati popř. výkonu. Její ukázka (Obrázek 23) demonstruje, o které hodnoty se jedná.

Statistiky	
Spálených kalorií na trase:	441.02 kJ (105 kcal)
Spálených kalorií při výkonu:	139.53 kJ (33 kcal)
Nejdelší klesání na trase:	o 15.0 m
Nejdelší stoupání na trase:	o 0.0 m
Převýšení:	o 18.0 m (253.0 - 271.0 m n. m.)
Klesání celkem:	o 15.0 m (114 m trasy)
Stoupání celkem:	o 0.0 m (0 m trasy)
Vzdálenost na trase:	2.697 km
Celková vzdálenost:	21.712 km

Obrázek 23 – Tabulka vypočtených hodnot

První hodnota spálených kalorií resp. kilojoulů se vztahuje celkově na vybranou trasu. Pokud tedy sportovec např. intenzivně trénuje na určité trati, má zde přehled o celkové vydané energii. Druhá hodnota udává stejnou hodnotu, ale v rámci právě jednoho vybraného výkonu. Obě hodnoty jsou kromě druhu aktivity závislé i na hmotnosti sportovce. Tyto údaje se zobrazují v dialogovém okně po výběru položky menu „Moje údaje“ a následně „Sportovec“ (Obrázek 24). Po stisku tlačítka „Upravit“ lze prostřednictvím formuláře údaje změnit.

Údaje o sportovci

Jméno:	Jakub
Váha:	67 kg
Výška:	190 cm
Věk:	24 let
BMI:	18.6 (optimální váha)

Upravit OK

Obrázek 24 - Dialogové okno s údaji o sportovci

Jméno a věk, zde hrají roli spíše orientační, ale pomocí výšky a váhy se počítá právě BMI index, za nímž je hned uvedeno, do jaké kategorie tělesné váhy uživatel spadá.

Další řádky v tabulce statistik se soustřeďují na výškové rozdíly na trase. Nejprve jde o hodnoty, které informují o hodnotě největšího stoupání resp. poklesu na trati a o kolik metrů během tohoto úseku sportovec vystoupá resp. sestoupí. Výsledky menší

než 10 metrů se neuvažují, jelikož trasy jsou obvykle delší vzdálenosti než 100 metrů a při nich takový rozdíl není příliš znatelný. Zároveň se tímto omezením alespoň částečně eliminuje nepřesnost měření, ke které může dojít. Následující položka nazvaná „Převýšení“ udává rozdíl nejmenšího a největšího bodu trati a zároveň uvádí hodnoty nadmořské výšky v těchto bodech. „Klesání celkem“ a „Stoupání celkem“ jsou hodnoty, které informují o změně terénu trati vzhůru resp. dolů jako celku. Doplnuje tak předcházející údaje tím, že udává jasnou informaci o tom, zda je trať převážně „z kopce“ nebo „do kopce“. Zároveň zde tentokrát nechybí vzdálenost - délka trati, která má tento trend. Při tomto výpočtu se opět uvažuje souvislá změna výšky alespoň o 10 metrů.

Nakonec jsou uvedeny hodnoty celkové vzdálenosti, která byla na vybrané trase sportovními výkony uskutečněna a hodnota spočítaná v rámci všech výkonů a všech tratí celkem.

5.2.3 JMapViewer

Pro lepší představu o trase i v desktopové aplikaci byla využita knihovna JMapViewer. Jedná se o Java komponentu, která dovoluje jednoduše integrovat online OpenStreetMap do Java aplikace. Poskytuje např. ovládání zoomu mapy pomocí posuvníku a tlačítek, přidání seznamu značek přes rozhraní *MapMarker* aj.



Obrázek 25 - Ukázka zobrazení trati v desktopové aplikaci

Komponenta JMapView je tedy obohacena pouze o obarvené body trasy vybrané z databáze a umístěné do mapy pomocí GPS souřadnic. Při výběru mapy je nastaveno i vycentrování na umístění aktuálně vybrané trasy. Pokud není dostupné připojení k síti internet, zobrazují se body trasy bez podkladu mapy.

5.2.4 JFreeChart

JFreeChart je zdarma poskytovanou Java knihovnou, která programátorům umožňuje ve svých aplikacích snadno zobrazit grafy dosahující profesionální kvality. Tato knihovna podporuje celou řadu druhů grafů včetně jejich kombinací např.:

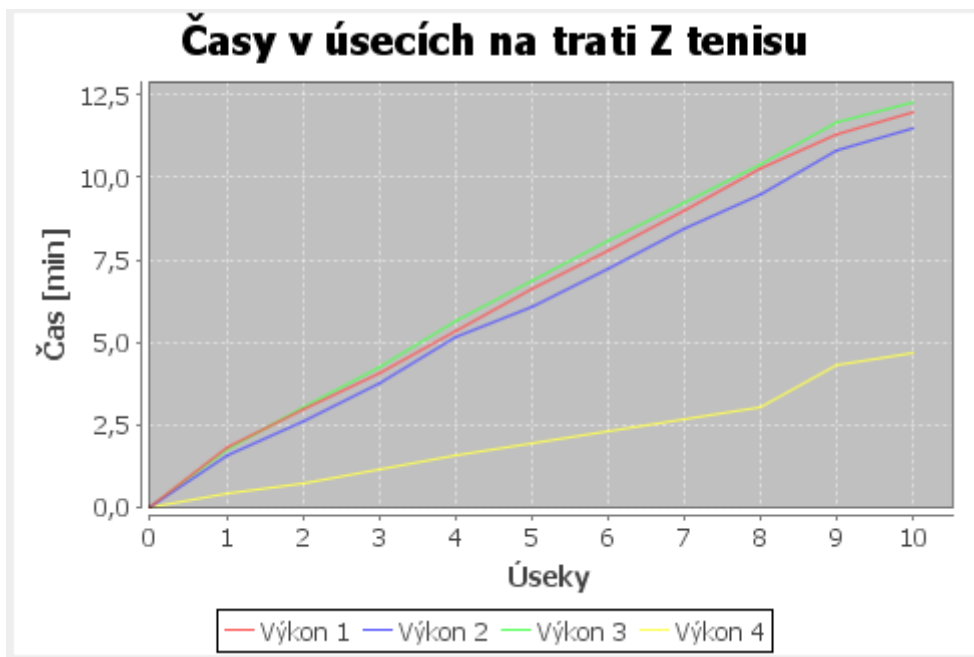
- XY grafy
- koláčové grafy
- Ganttův diagram
- sloupcové grafy
- grafy s jedinou hodnotou (teploměr)

Samozřejmě součástí jsou i popisky os, nadpis grafu a legenda. Další funkcionalitou je přibližování za pomoci myši a změna nastavení za pomoci lokálního menu. V závislosti na zmapované nadmořské výšce v bodech, pak mohou být zobrazeny grafy udávající výškový profil tratě pro vhodnou představu o náročnosti tratě z hlediska převýšení (Obrázek 26).



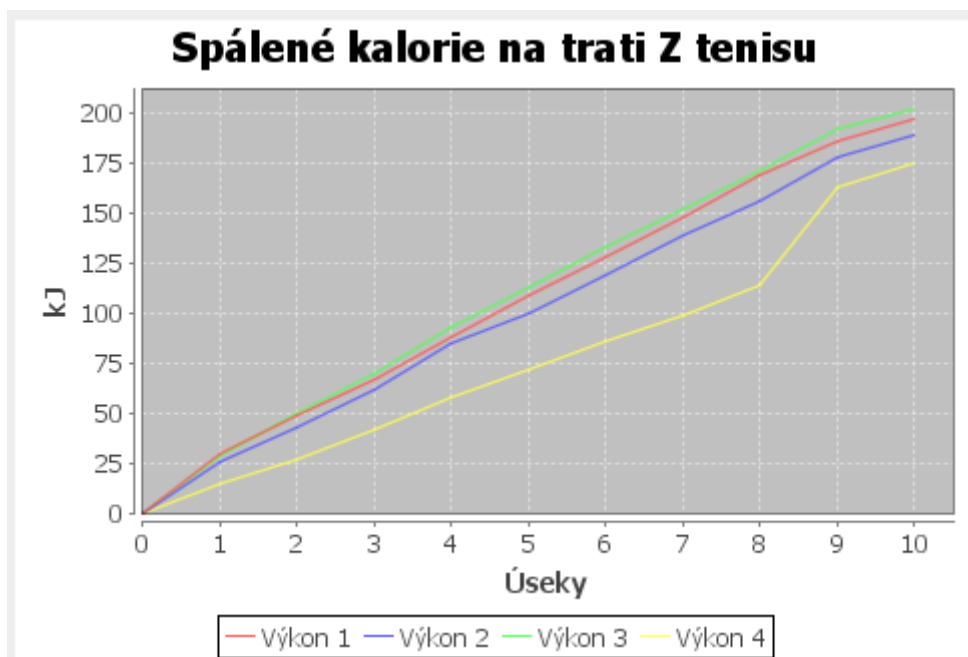
Obrázek 26 - Výškový profil tratě (XY graf)

Obrázek 27 potom demonstruje grafické znázornění o časovém průběhu čtyř různých výkonů v rámci stejné tratě.



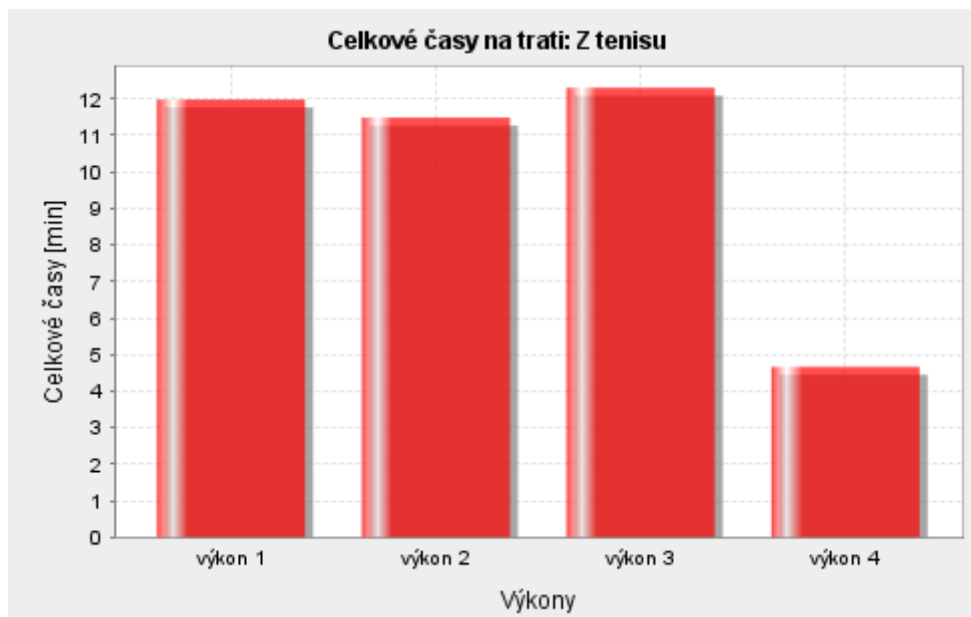
Obrázek 27 - Porovnání časových výkonů

Obdobou je potom graf (Obrázek 28) vykreslující množství spálených kilojoulů od začátku do konce pohybové aktivity.



Obrázek 28 - Graf spálených kilojoulů

Sloupcový graf (Obrázek 29) byl využit pro grafické znázornění celkových časů výkonů. Přehledně tedy demonstruje zlepšení nebo naopak zhoršení z hlediska časových výsledků.



Obrázek 29 - Graf celkových časů výkonů (sloupcový graf)

Implementace grafů pomocí frameworku JFreeChart spočívá v podstatě ve volbě typu grafu, nastavení popisků a zejména pak naplnění daty např. *dataset DefaultXYDataset* je nutné naplnit dvourozměrným polem hodnot pro výše demonstrované zobrazení XY grafu s více výkony.

5.2.5 GUI a ovládání

Jádro aplikace pak představuje třída *MainWindow*, ve které je v podstatě poskládané grafické uživatelské rozhraní.

Pro vhodné umístění komponent byl použit *GridBagLayout* s pomocnou třídou *GBCBuilder*, což je tabulkově založený správce rozvržení. Každý element má svoji přesnou polohu a rozlohu v mřížce, prostor zvenku (pad), zevnitř (inset), způsob vyplnění (fill) a zarovnání (anchor).

K překreslení komponent aplikace dochází po výběru příslušného řádku tabulky. V závislosti na výběru trasy se překreslí všechny komponenty hlavní obrazovky a zobrazí podrobnější informace k výběru (mapu trasy, odpovídající graf, výkony na trati a vypočtené hodnoty). Stejně tak se v závislosti na výběru výkonu překresluje vypočtená hodnota s množstvím spálených kalorií při něm. Jednotlivé komponenty jako mapa, graf, řazení výkonů je možné ovládat nezávisle na sobě.

Jak již bylo zmíněno, hlavní okno aplikace obsahuje i menu, které zastává vedlejší funkce pro ovládání. Kromě klasických možností pro ukončení aplikace a nápovědy se stručným popisem významu položek menu, obsahuje možnosti právě pro změnu zobrazení typu grafu. To je realizováno komponentou *JRadioButtonMenuItem*. Jedná se o prvek, kde je jasně uvedeno, která položka je právě vybrána, přičemž zvolena může být právě jedna.

Další položka představuje výše zmíněné zobrazení a změnu údajů o sportovci, na základě kterých se počítají spálené kilojouly. Zde by mělo být doplněno, že vyplněné údaje se po ukončení aplikace nikam nevytrácejí, ale využívají pro ukládání XML soubor umístěný v domovském adresáři systému právě přihlášeného uživatele a cestou „\KondiMetr\config.xml“.

```
<?xml version="1.0" encoding="UTF-8"?>
<sportovec>
  <jmeno>Jakub</jmeno>
  <vaha>67</vaha>
  <vyska>190</vyska>
  <vek>24</vek>
</sportovec>
```

Obrázek 30 - Obsah souboru config.xml

Stejně záznamy poté inicializují strukturu dat předávanou v aplikaci přes třídu *Sportovec* při příštím spuštění aplikace. Nově zadané údaje se do souboru nezapisují ihned, ale až před ukončením aplikace. K tomu slouží třída *ExitThread*, jejíž instance je vytvořena při startu aplikace a má za úkol zachytit ukončení aplikace. Jakmile se tak stane, začne proces ukládání. Pro práci s XML souborem bylo využito implementace založené na StAX, což je tzv. „Streaming API for XML“. Třídy *StAXParser* a *StAXWriter* jsou tedy určeny pro jednodušší práci čtení a zápisu z resp. do souboru XML s využitím standartních tříd z „javax.xml.stream“.

Následující položka menu „Výkony“ s možným výběrem „Odebrat vybraný“ umožňuje odstranění pro uživatele přebytečných výkonů. Podmínkou je pouze mít vybraný příslušný řádek v tabulce výkonů, jinak dojde k upozornění na tuto skutečnost dialogovým oknem. Při úspěchu této operace se zvolené záznamy z tabulky USEK a VYKON odeberou, ale ještě předtím se jejich duplicity uloží do tabulek ARCHIVOVANY_USEK a VYKON pro archivaci, jiné pozdější využití, sběr dat nebo zpětné jednodušší dolování ztracených dat.

Funkce výše zmíněné položky pro synchronizaci již byla popsána.

5.2.6 Sestavení aplikace

V samotném závěru byl využit nástroj Apache Maven, a to za účelem, aby vznikl jediný soubor s příponou jar, kterým bude možné spustit aplikaci bez nutnosti nastavování cest k externím knihovnám.

S využitím NetBeans, kde byl Maven součástí základní instalace, stačilo vytvořit nový projekt a nakopírovat do „Source Packages“ strukturu vytvořených zdrojových kódů. Pro kompilaci je stěžejní „předvygenerovaný“ soubor *pom.xml*, který se při vzniku projektu automaticky vytvořil ve složce „Project Files“. Zde je nutné provést několik změn, pro dosažení požadovaného efektu. Mezi tagy *<project>* je potřeba přidat tagy *<build>* s následující strukturou říkající překladači, aby tzv. „dependencies“ byly

obsaženy v cílovém souboru. Vhodné je upravit jen tag `<mainClass>` nastavující hlavní třídu.

```
<build>
  <plugins>
    <plugin>
      <artifactId>maven-assembly-plugin</artifactId>
      <executions>
        <execution>
          <phase>package</phase>
          <goals>
            <goal>attached</goal>
          </goals>
        </execution>
      </executions>
      <configuration>
        <archive>
          <manifest>
            <mainClass>dp.Main</mainClass>
          </manifest>
        </archive>
        <descriptorRefs>
          <descriptorRef>jar-with-dependencies</descriptorRef>
        </descriptorRefs>
      </configuration>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.3.2</version>
      <configuration>
        <source>1.7</source>
        <target>1.7</target>
      </configuration>
    </plugin>
  </plugins>
</build>
```

Obrázek 31 - Přidaný kód do souboru pom.xml

Druhou neméně důležitou částí úpravy konfiguračního souboru *pom.xml* je nastavení knihoven, které mají být slinkovány. Pravděpodobně existuje více způsobů, avšak jako vhodný postup se ukázalo nejprve přidat pro každou externí knihovnu tag `<dependency>` s vhodně nazvanými parametry.

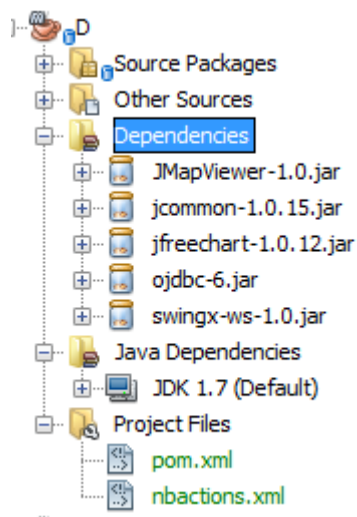
```

<dependencies>
  <dependency>
    <groupId>jfreechart</groupId>
    <artifactId>jfreechart</artifactId>
    <version>1.0.12</version>
  </dependency>
  <dependency>
    <groupId>ojdbc</groupId>
    <artifactId>ojdbc</artifactId>
    <version>6</version>
  </dependency>
  ...

```

Obrázek 32 - Přidání tagů `<dependency>`

Po uložení souboru pom.xml se ve složce „Dependencies“ objeví knihovna s názvem uvedeným v tagu `<artifactId>`. Nakonec je nutné každou z takto manuálně navolených knihoven označit pravým tlačítkem myši a zvolit volbu „Manually install artifact“, a poté najít externí knihovnu, kterou si přejeme slinkovat s projektem.



Obrázek 33 - Adresářová struktura projektu v Maven

Tento postup platí pro případ, že jsou knihovny lokálně dostupné. Maven umožňuje i přidat knihovny online skrze databázi na internetu pomocí volby „Add Dependency“ nad adresářem „Dependencies“ v projektu, ale vzhledem k tomu, že knihovny byly lokálně dostupné, byl využit tento postup.

Závěr

Podařilo se vytvořit funkční mobilní aplikaci, která dovede snímat polohu jedince pomocí mobilního zařízení s OS Android a stejně tak funkční desktopovou aplikaci, která následně vizuálně a statisticky vyhodnocuje zaznamenané údaje o výkonech, trasách a sportovci. Ačkoliv se aplikace ubírá jiným směrem než řešení uvedené v kapitole 4, tak by mohla těmto řešením konkurovat. Hlavním rozdílem je především v rozdělení na část mobilní a desktopovou. Dále neuplatňuje a nenutí k napojení na sociální síť a s tím spojenou propagaci, ačkoliv to je z marketingového hlediska jistě významný krok. Jako výhodu desktopové části bych uvedl její přehlednost a jednoduchost pro ovládání. Zejména umístění dat na jednom místě a možnost je snadno řadit a srovnávat.

Co se týče výhledu do budoucna, tak by bylo vhodné po vzoru stávajících řešení napojit aplikaci na síť internet pro vzdálené bezdrátové porovnávání výsledků sportovců na stejných trasách mezi sebou. Nicméně bych dal přednost řešení, kdy by uživatel nebyl omezen datovým připojením a nucen se registrovat a přihlašovat. Nejvhodnější by pravděpodobně byla varianta „svobodné volby výběru“ po spuštění aplikace.

Testování ukázalo, že by pravděpodobně bylo i vhodné nechat na uživateli výběr délky měřených úseků trati. Sportovní aktivity jsou také možným a nenáročným rozšířením, avšak ty nejběžnější a nejoblíbenější jsou zahrnuty.

Ačkoliv vyhodnocení výsledků bylo soustředěno do desktopové aplikace, možným vylepšením by mohlo být i vytvoření jednoduššího vyhodnocení výkonů přímo v aplikaci mobilního zařízení. Popřípadě přidat alespoň informaci o posledním nebo nejlepším dosaženém výsledku sportovce pro větší motivaci zlepšení přímo při provádění aktivity.

V závěru testování se taktéž ukázalo, že množství tras v aplikaci může poměrně rychle nabývat. Vhodným doplněním by tedy bylo vytvoření možnosti filtrování a vyhledávání tras dle různých kritérií jako délka, název nebo druh aktivity, který byl na trase vykonán nejčastěji popř. poprvé.

Pro ukládání dat v mobilní aplikaci by mohlo být vhodnější využít XML soubor, namísto prostého textového. Význam to však má až případně, kdy by pro to existovalo využití, jako například propojení s nějakou webovou službou. Do té doby bych tomu přikládal význam pouze estetický.

Jako rozšíření desktopové aplikace bych navrhoval přidání dalších zajímavých grafů např. rychlosti, ale i jiných.

Literatura

- [1] Android. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-03-31]. Dostupné z: [http://en.wikipedia.org/wiki/Android_\(operating_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [2] iOS (Apple). In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-03-31]. Dostupné z: [http://cs.wikipedia.org/wiki/IOS_\(Apple\)](http://cs.wikipedia.org/wiki/IOS_(Apple))
- [3] Android developers. Application Fundamentals [online]. 2011 [cit. 2013-04-09]. Dostupné z: <http://developer.android.com/guide/components/fundamentals.html>
- [4] HENDL, Jan a Lubomír DOBRÝ. Zdravotní benefity pohybových aktivit: monitorování, intervence, evaluace. Vyd. 1. Praha: Karolinum, 2011, 300 s. ISBN 978-802-4620-008.
- [5] MÁČEK, Miloš a RADVANSKÝ, Jiří. Fyziologie a klinické aspekty pohybové aktivity. Praha: Galén, 2011, 245 s. ISBN 978-80-7262-695-3.
- [6] Global Positioning System. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-09]. Dostupné z: http://cs.wikipedia.org/wiki/Global_Positioning_System
- [7] The Android GPS. In: Backcountry navigator [online]. [cit. 2013-04-09]. Dostupné z: <http://www.backcountrynavigator.com/android-gps/>
- [8] Index tělesné hmotnosti. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-09]. Dostupné z: http://cs.wikipedia.org/wiki/Body_mass_index
- [9] OpenStreetMap. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-10]. Dostupné z: <http://cs.wikipedia.org/wiki/OpenStreetMap>
- [10] Google Maps. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-10]. Dostupné z: http://en.wikipedia.org/wiki/Google_Maps
- [11] Google play. GOOGLE. Google play [online]. 2012 [cit. 2013-04-12]. Dostupné z: <https://play.google.com/store/apps>

Příloha A – Obsah přiloženého DVD

- zdrojové soubory desktopové aplikace
- zdrojové soubory mobilní aplikace
- zkompileovaná desktopová aplikace včetně externích knihoven
- zkompileovaná mobilní aplikace
- manuál README
- elektronická podoba diplomové práce ve formátu PDF