

**VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky**

**Asistent profesionálního řidiče na platformě Android
Truck driver assistant for Android platform**

2012

Tomáš Pašek

Zadání bakalářské práce

Student: **Tomáš Pašek**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R059 Mobilní technologie

Téma: **Asistent profesionálního řidiče na platformě Android
Truck Driver Assistant for Android Platform**

Zásady pro vypracování:

Cílem práce je tvorba aplikace, která usnadní řidičům kamiónů a jejich dispečerům. Aplikace bude zaznamenávat údaje o jízdě a poskytovat doplňující informace. Hlavními funkcemi bude záznam trasy a rychlosti vozidla na základě údajů z GPS (jako dodatečná kontrola k tachogramům a elektronickému mýtnému), upozorňovat řidiče na blížící se povinnou přestávku a její konec. Řidič a dispečink budou mít možnost v sdílet další aktuální informace (veřejně, pro dispečink, v rámci firmy) - zejména o umístění odstavných parkovišť a jejich volné kapacitě, ceně PHM, dopravních omezeních a omezení jízd, apod.

1. Zjistěte, jaké funkce jsou řidiči a dispečery požadovány, vyhledejte případné konkurenční aplikace, které poskytují podobnou funkčnost a srovnajte jejich funkce.
2. Určete, jakým způsobem budou informace sdíleny (HTTP, cloudové řešení, webové služby, apod.), zdokumentujte formát, v němž budou informace předávány. Vyberte, které části Android API využijete a v jaké verzi. Popište případné nestandardní knihovny a nástroje, které budete využívat.
3. Analyzujte, navrhnete a implementujte mobilní aplikaci a serverovou část.
4. Výsledné řešení otestujte alespoň na dvou různých mobilních zařízeních a zhodnoťte dosažené výsledky. Srovnajte funkčnost Vaší aplikace s aplikacemi z bodu 1.

Seznam doporučené odborné literatury:

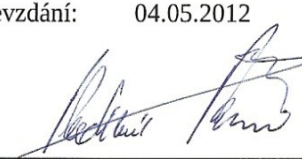
1. Burnette, E. Hello, Android: Introducing Google's Mobile Development Platform. Pragmatic Bookshelf, 2008. ISBN: 978-1-93435-617-3.
2. Meier, R. Professional Android 2 Application Development. Wrox Press, 2010. ISBN: 0-47056-552-0.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. Pavel Moravec, Ph.D.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012


prof. RNDr. Vladimír Vašínek, CSc.
vedoucí katedry




prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne: 26. 4. 2012

Tomáš Pásek

.....
Podpis

Poděkování

Rád bych poděkoval Ing. Pavlu Moravcovi Ph.D. za odbornou pomoc a konzultaci při vytváření této bakalářské práce.

Abstrakt

Tématem bakalářské práce je návrh a implementace asistenta profesionálního řidiče na platformě Android. Aplikace obsahuje serverovou i klientskou část pro sdílení informací jako jsou čerpací stanice nebo parkoviště s danými záznamy. Součástí aplikace je i záznam trasy a záznam rychlosti na dané trase. Na serverové části se ukládají veškeré záznamy, ke kterým má přístup dispečink. Sdílení informací mezi klientem a serverem probíhá přes protokol HTTP.

Klíčová slova

Android, aplikace, GPS, HTTP, záznam jízd, databáze

Abstract

The topic of the bachelor thesis is the design and implementation of an professional driver Assistant on the Android platform. The application includes server and client parts designed for information sharing (such as gas stations or parking lot availability). Additional parts of the application are used to record routes and speed details on a given route. The server part provides the control centre with access to all recorded information. Sharing information between the client and the server is done via HTTP.

Key words

Android, application, GPS, HTTP, driving record, database

Seznam použitých zkratek

Zkratka	Anglický význam	Český význam
ADR	Accord Dangereuses Route	Evropská dohoda o mezinárodní silniční přepravě nebezpečných věcí
AETR	European Road Transport Agreement	Evropská dohoda o práci osádek vozidel v mezinárodní silniční dopravě
API	Application Programming Interface	Rozhraní pro programování aplikací
CSS	Cascading Style Sheets	Kaskádové styly
GPS	Global Positioning System	Družicový polohový systém
GUI	Graphical User Interface	Grafické uživatelské rozhraní
HTTP	Hyper Text Transfer Protocol	Protokol pro přenos hypertextu
JSF	Java Server Faces	Grafické rozhraní stránek
MySQL		Databázový systém
PHP	Hypertext Preprocessor	Skriptovací programovací jazyk
URL	Uniform Resource Locator	Jednoznačné určení zdroje
XML	Extensible Markup Language	Značkovací jazyk

Obsah

1	Úvod.....	1
2	Požadavky na aplikaci.....	2
	2.1 Požadované funkce pro řidiče.....	2
	2.2 Požadované funkce pro dispečery.....	3
3	Konkurenční programy.....	4
	3.1 LogiAssist.....	4
	3.2 TruckerTimer.....	5
	3.3 tudyNE.....	5
	3.4 City Traffic.....	6
	3.5 Tabulka srovnání aplikací.....	6
4	Využívané součásti Androidu a sdílení informací.....	7
	4.1 Android API.....	7
	4.1.1 Location a Google Maps.....	7
	4.1.2 Database.....	7
	4.1.3 Graphics.....	8
	4.1.4 HTTP.....	8
	4.2 Možnosti sdílení informací.....	8
	4.2.1 Cloud computing.....	9
	4.2.2 Webové služby.....	9
	4.2.3 HTTP.....	9
	4.3 Mé řešení sdílení.....	9
5	Analýza a návrh aplikace.....	11
	5.1 Klientská část.....	11
	5.2 Serverová část.....	13
6	Implementace.....	16
	6.1 Klientská část.....	16
	6.1.1 Databáze.....	16

6.1.2	Práce s GPS a mapou.....	17
6.1.3	Záznam nové trasy.....	18
6.2	Serverová část.....	20
6.3	Nasazení do provozu	23
7	Testování	24
8	Porovnání s konkurenčními aplikacemi	26
9	Závěr.....	27
	Použitá literatura	28
	Přílohy	xxix

1 Úvod

Hlavním cílem této bakalářské práce je navrhnout možné řešení a následně zrealizovat program, který by měl ulehčit práci profesionálním řidičům a popřípadě jejich dispečerům. Hlavním úkolem vytvořeného programu je zaznamenávat denní dobu řízení, přestávky a rychlost na základě údajů z GPS. Dále bude obsahovat databázi s čerpacími stanicemi a parkovišti, která bude přístupná přes webové rozhraní. Dispečink tak bude moct podle daných údajů navrhnout trasy, které budou výhodné, jak pro samotnou přepravu, tak i pro řidiče.

V této oblasti se nějaký čas pohybují, a proto jsem zvolil toto téma, protože řidiči by aplikaci s touto funkcí přivítali.

První kapitola obsahuje požadavky, které jsou na aplikaci vyžadovány. Jedná se o výčet požadavků, které přijdou vhod řidičům i dispečerům.

Do druhé kapitoly jsou zařazené konkurenční aplikace. Tyto aplikace jsou následně v tabulce srovnány.

V další části této práce jsou uvedené jednotlivé součásti Androidu, které byly při vývoji aplikace použity, jedná se o jednotlivá API. Dále je zde uvedeno sdílení informací mezi klientskou částí a serverovou částí.

Čtvrtá a pátá kapitola obsahuje analýzu, návrh a samotnou implementaci aplikace. Kapitoly obsahují také diagramy, které byly při vývoji dodržovány.

Další v řadě je kapitola o testování výsledné aplikace, obsahující výčet některých chyb, které se při vývoji aplikace objevily.

Předposlední je kapitola, kde jsou srovnány aplikace z kapitoly dvě s touto aplikací.

Poslední kapitola obsahuje závěr a konečné zhodnocení dosažených výsledků.

2 Požadavky na aplikaci

Tato kapitola obsahuje soupis požadavků na tuto aplikaci. Ve výčtu samozřejmě neuvedu všechny požadavky, protože každý má jiné nároky a očekává jiné požadavky od aplikace.

Jeden z hlavních požadavků na každou aplikaci je, aby byla pokud možno co nejvíce graficky přívětivá k uživateli. I ten nejméně zdatný uživatel, by měl být bez větších problémů schopen se rychle zorientovat a aplikaci umět ovládat.

Další požadavek by mohl být: dobrá čitelnost za jakýchkoli podmínek. Toto je sice dáno nejvíce typem displeje mobilního zařízení, ale aplikace by tomu mohla alespoň trochu napomáhat. Měli by být vhodně zvolené ovládací prvky a barvy.

2.1 Požadované funkce pro řidiče

Jeden z nejdůležitějších požadavků pro řidiče zajisté bude již zmíněná dobrá čitelnost na displeji. Aplikace bude používána jak ve vozidle tak i mimo něj a všichni určitě ví, co dokáže slunce s displejem mobilního zařízení.

Dalším důležitým požadavkem jsou mapové podklady. Měli by být pokud možno, co nejpodrobnější a podporovat zobrazování své aktuální pozice pomocí GPS mobilního zařízení.

Možnost odpočtu jednotlivých dob řízení a povinných přestávek, které řidiči ukládá mezinárodní úmluva AETR. Tahle funkce je velice užitečná pro uživatele a značně jim ulehčuje práci, aby nemuseli pořád myslet na hodiny. Více informací o úmluvě AETR zde [7].

Možnost přidávání do mapy své body zájmu jako jsou čerpací stanice a parkoviště pro nákladní vozidla.

Dalším požadavkem je záznam projeté trasy. Trasy by měly být přístupné, jak řidiči, tak dispečinku. Tohle bude možné pomocí toho, že záznamy budou odesílány na server, kde budou přístupné a bude možné si je zobrazit na mapě.

Záznam rychlostí na dané trase by mohl být další požadavek od téhle aplikace. Bylo by možné tyto údaje použít jako dodatečnou kontrolu k tachografu vozidla.

2.2 Požadované funkce pro dispečery

Práce dispečera může být velice různorodá a závisí na velikosti společnosti, ve které tuto profesi vykonává. Vykonávat profesi dispečera znamená shánět přepravy, mít přehled o aktuálních polohách vozidel, které má na starost. Dále informuje řidiče, kde a kdy má přistavit vozidlo k naložení popř. složení nákladu. Hlavní činnosti, by ale mělo být shánění jednotlivých přeprav a výpočet nákladu na danou přepravu. Při plánování přepravy musí zohlednit cenu pohonných hmot, mýtného, amortizaci vozidla a spoustu dalších faktorů, které ovlivňují konečnou cenu přepravy.

Jedním z nejdůležitějších požadavků je, aby měl přehled o vozidlech, která má na starost, kde se nachází a zda jsou prázdná nebo naložena.

Další možností je nahlédnout do zaznamenaných tras a zobrazit si je na mapě. Podle toho může prozkoumat, zda jsou trasy výhodné, jak pro řidiče, tak pro přepravce, co se týká financí. Podle toho by mohl upravit plánování dalších jízd.

3 Konkurenční programy

Na poli konkurenčních programů jsou dostupné aplikace, ať už více či méně zdařilé, které se zabývají pouze jedním problémem. Nejčastěji se jedná o dopravní informace, kamerový systém křižovatek atd. S funkcí, kterou obsahuje má aplikace, moc společného nemají. Nejvíce se mé aplikace přibližují níže zmíněné programy. Zahrnul bych zde i telematické programy, ať už od jednotlivých automobilek nákladních vozů, nebo od různých společností. Ty, ovšem jsou na jiné úrovni, než je má aplikace a nejsou pro mobilní zařízení, ale jsou již většinou součástí palubního počítače nákladního vozidla. Tyto programy analyzují výkony vozidla, hlídají servisní intervaly, hodnotí řidiče, určují pozici vozidla a obsahují mnoho dalších více, či méně důležitých informací.

Z českých mobilních aplikací jsou nejvíce podobny mé aplikaci City Traffic a tudyNE. Tyto aplikace jsou zaměřené pouze na dopravní informace a neobsahují další možnosti, které by se daly srovnat s mou vytvořenou aplikací.

3.1 LogiAssist

Mobilní aplikace vytvořena v Německu za podpory významných německých společností. Aplikace má být komplexním řešením všech situací, které mohou potkat řidiče nákladních vozů na území Německa.

Samotná aplikace se velmi jednoduše a intuitivně ovládá. Na mapě se vybere oblast, kterou si uživatel zvolí a vybere si také body zájmu nebo události, a následně se zobrazí na mapě. Může to být zastávka kamionu (Autohof), čerpací stanice, kolona, práce na silnici a mnoho dalších bodů zájmu. Na mapě se rovněž zobrazují uživatelé této aplikace, kteří mohou navzájem mezi sebou komunikovat formou chatu. Pod svým profilem si může uživatel vytvářet nové body zájmu nebo události. Také si může ukládat své kamarády do oblíbených a poté v adresáři vidí, jak jsou daleko od něho, může mu zavolat, poslat zprávu nebo si na mapě zobrazit, kde se právě nachází.

Součástí je i samostatná aplikace DocStop, ve které jsou zaznamenány lékařské služby. Vzhledem k aktuální pozici si může řidič na mapě zobrazit potřebnou lékařskou službu. Může to být buď nemocnice, nebo si může zvolit jednotlivé lékařské odvětví jako je zubař, oční klinika atd. Další součástí je aplikace, ve které se zadá čtyřčíslí nebezpečného nákladu náležícího pod ADR. Řidič obdrží informace o druhu nákladu, omezení které se ho týkají atd. Pro více informací doporučuji shlédnout [9]

Aplikace je velmi zdařilá a dobře propracovaná, což napovídá tomu, že se na ní podíleli profesionální společnosti a daná kvalita se dá očekávat.

Ve své aplikaci při vytváření parkoviště, bude možné zadávat jeho aktuální stav – zda jsou na parkovišti nějaká volná místa nebo je plno. Tyto informace bude moci vložit každý, kdo se zrovna nachází na parkovišti a ostatní si ji budou moci zhlédnout. Podobnou věc bych chtěl použít i u čerpacích stanic. Zaznamenávala by se aktuální cena pohonných hmot. Dále bych chtěl ve své aplikaci použít záznam doby řízení a odpočinku, což mi u téhle aplikace chybí.

3.2 TruckerTimer

Mobilní aplikace představující dokonalý digitální tachograf. Je tvořena pro řidiče nákladních automobilů, kterým zaznamenává jednotlivé doby řízení a povinné přestávky.

TruckerTimer pomáhá orientovat se v právních předpisech pro řízení nákladních vozidel, zahrnující předpisy nařízení podle EU a AETR. Graficky znázorňuje a odpočítává čas, jak dlouho řidič ujel za den, kolik času ještě může řídit do konce týdne a do čtrnácti dnů. Každý den je samozřejmě rozdělen podle pravidel, kolik hodin se může řídit a jak dlouhou chce řidič pauzu, zdali si ji chce zkrátit,... Podle GPS určuje práci jako řízení vozidla nebo jinou práci. Samozřejmostí je shrnutí za den, týden a čtrnáct dnů, které je možné odesílat pomocí Bluetooth a zpracovávat v Excelu. Více informací o téhle aplikaci zde [10].

Tato aplikace je velice zdařilá a po grafické stránce uživatelsky příjemná a jednoznačná. Grafické znázornění s odpočtem času je velice propracované.

V mé aplikaci bych rád měl také vytvořený odpočet do konce doby řízení, konce přestávky, a aby bylo možné vidět celkový sumář za týden, popř. 14 dní. Určitě nebude na takové úrovni jako je v téhle aplikaci, ale účel bude plnit.

3.3 tudyNE

Je zdařilá aplikace, ovšem taky zaměřena jenom na dopravní informace jako jsou dopravní nehody, uzavírky a zobrazuje i hustotu provozu. Aplikace přebírá dopravní informace od Policie ČR, což je velký klad, předpokládá se pravdivost a aktuálnost situace. Podrobnosti o této mobilní aplikaci [12]

3.4 City Traffic

Má v sobě obsaženou databázi s parkovišti a zobrazují se i kolik je volných parkovacích míst. Ovšem jsou to parkovací místa pro osobní vozidla, tudíž s její aplikací se nedá srovnávat. Aplikace má velice dobře propracované dopravní informace, dopravní nehody, seznam kamer, hustotu provozu. Aplikace obsahuje velice malé pokrytí České republiky, pokrytá jsou jenom tři největší města, jako je Praha, Brno a Ostrava. Více informací o zmíněné aplikaci naleznete zde [11].

3.5 Tabulka srovnání aplikací

	LogiAssist	TruckerTimer	City Traffic	tudyNE
Databáze parkovišť / čerpacích stanic	ano / ano	ne / ne	ano / ne	ne / ne
Odpočet dob řízení	ne	ano	ne	ne
Dopravní informace	ano	ne	ano	ano
Platforma	Android	Android	Android	Android
Zpoplatněná	ne	ano	ne	ne
Plusy / mínusy	+ součástí aplikace + komplexní řešení - Nemá odpočet dob řízení	+ velice propracovaný odpočet dob řízení - Žádné jiné funkce	+ dopravní informace - malé pokrytí ČR	+ Dopravní informace od Policie ČR - další funkce

Tabulka 3.1: Porovnání aplikací

4 Využívané součásti Androidu a sdílení informací

Tato kapitola obsahuje výběr API a dalších součástí, které jsem použil při vlastní implementaci této aplikace. V této části je i vysvětleno sdílení mezi mobilní a serverovou částí v mé aplikaci.

4.1 Android API

API je jakýsi soubor všech tříd a funkcí, které je možno použít při vývoji a definuje, jak jsou knihovny volány ze zdrojového kódu programu. Každý programovací jazyk má svoje API, aby práce programátora byla co nejvíce ulehčena používáním jednotlivých knihoven. Popis a informace o jednotlivých API [1][2][3][4].

4.1.1 Location a Google Maps

Android.location obsahuje třídy, založené na získávání polohy mobilního zařízení. Hlavní součástí location API je Location Manager, který poskytuje určit polohu zařízení pomocí GPS nebo pomocí Wi-Fi či jiné sítě.

Pro zobrazení pozice na mapě slouží externí knihovny Google Maps. Hlavní třída je MapView, která zobrazuje mapu, zachycuje dotyky v mapě pro samotný posun zobrazené mapy, přiblížení a oddálení. Tato třída obsahuje také mnoho součástí pro jednotlivé překrývání map, ať už mapou jinou, či různými ikonami. Tato externí knihovna s mapami není součástí knihovny Android, a proto se mohou, vyskytnou problémy s kompatibilitou u některých zařízeních.

K samotnému zobrazení map ať už pomocí emulátoru nebo na fyzickém zařízení je potřeba získat klíč k Maps API. Klíč je vlastně otisk certifikátu, který se používá k podpisu aplikace. Poté se už jenom vygenerovaný klíč vloží k MapView v dané aplikaci. Jeden klíč je možno použít pro více aplikací.

4.1.2 Database

Balíček android.database.sqlite obsahuje třídy pro správu SQLite databáze.

Součástí tohoto balíčku je třída SQLiteDatabase, která má hlavní metody pro vytvoření a smazání databáze. Také dovoluje vyvolávat SQL dotazy a provádět běžné úpravy databáze.

Další část, kterou jsem použil je třída SQLiteOpenHelper, což je pomocná třída pro správu databáze. Tato třída se stará o otevření databáze před zápisem, vytvoření pokud není vytvořená. Stará se také o upgrade databáze.

4.1.3 Graphics

Poskytuje základní grafické nástroje pro vykreslování přímo na obrazovku. Nedílnou součástí je také třída Canvas kreslící jednotlivé symboly, ikony a různé markery na zvolené plátno. Z balíčku jsem také použil třídu Point, která obsahuje dva parametry, což jsem využil pro zobrazení aktuální pozice a jako její dva parametry jsem použil zeměpisnou šířku (latitude) a zeměpisnou délku (longitude).

4.1.4 HTTP

Rozhraní pro HTTP komponenty. Zabývá se službami pro HTTP protokol, jako je příprava zprávy pro odeslání, záhlaví zprávy, spojení, ve kterém je zpráva odeslána. Umožňuje definovat jednotlivé metody jako je GET, POST apod.

Zde jsou uvedeny jen ty Android API, které v aplikaci jsou využívány v největší míře. Samotná aplikace využívá daleko více Android API, než je výše uvedeno.

4.2 Možnosti sdílení informací

Pod pojmem sdílení informací, mám na mysli výměnu dat mezi mobilní částí a serverovou částí. Jak je vidět, už ze zadání vyplývá, že se naskytlo více možností sdílení informací a budoucí komunikace mezi klientem a serverem.

4.2.1 Cloud computing

Cloud computing je sdílení softwarového a hardwarového prostředku pomocí sítě. Jednoduché připojení k daným službám odkudkoli a nezávislé na platformě. Pro přístup ke sdíleným informacím uživatelé používají webový prohlížeč nebo nějakého klienta.

Cloudové řešení je zbytečně složité pro moji aplikaci. Navíc by muselo být závislé na poskytovateli služeb. Více informací zde [5].

4.2.2 Webové služby

Webové služby jsou aplikace pro výměnu dat mezi uživatelem a serverem. Jsou ovládány uživatelem běžným internetovým prohlížečem. Nejsou závislé na programovacím jazyku – používají svůj standart ve formátu XML pro komunikaci.

4.2.3 HTTP

Internetový protokol určený pro zasilání HTTP souborů. Protokol funguje formou dotaz – odpověď. Klient se serverem si vyměňují při komunikaci zprávy v textové podobě. Je to nejrozšířenější komunikační protokol na internetu. Tento protokol používá URL pro jednotné určení daného zdroje. Informace čerpány z těchto stránek [6].

4.3 Mé řešení sdílení

V mé aplikaci jsem použil sdílení aplikací pomocí protokolu HTTP.

Uživatel má na svém mobilním telefonu nainstalovanou tuto aplikaci, která odesílá data po připojení k internetu. Data, to jsou konkrétně databáze čerpacích stanic a parkovišť. Po připojení mobilního zařízení k internetu, se data odesílají na server, kde se vloží do databáze. Na serveru je vytvořená databáze s příslušnými tabulkami, do kterých se vkládají odeslaná data.

V aplikaci se vytváří QueryString. Odesílaná data jsou přidávána jako parametry. Výsledný QueryString je přidán do URL. Data jsou odeslána na server metodou GET. Každý QueryString obsahuje jako první parametr, parametr t. Tento parametr slouží k rozeznání, o jaký záznam se jedná.

t = 1.....čerpací stanice

t = 2.....parkoviště

t = 3.....trasa

```
String s = ("http://10.0.2.2:8080/WebApplication2/Muj?t=3&jmeno=" +  
jmeno + "&casZac=" + casZac + "&start=" + start + "&zacatekstav=" +  
zacatekstav + "&konCas=" + konCas + "&konec=" + konec +  
&stavKonec=" + stavKonec + "&rok=" + rok + "&mesic=" + mesic +  
&den=" + den);
```

Tohle řešení mi přišlo jako zcela dostačující pro mou aplikaci, v téhle fázi vývoje v jaké se momentálně nachází. V dalších fázích vývoje bude toto řešení nahrazeno odesíláním celého souboru na server.

5 Analýza a návrh aplikace

V téhle kapitole se věnuji analýze a návrhu mnou vytvořené aplikace. Analýza je rozdělena na dvě části, každá vyžaduje něco jiného. Na část klientskou, což je mobilní aplikace a část serverovou.

Při analýze jsem využíval i diagramy aktivit některých operací, které probíhají. Obrázek (Obr. 5.3) zobrazuje, jak probíhá přihlášení uživatele k aplikaci a následné vytvoření čerpací stanice. Čerpací stanice bude odeslána na server. Podobně tomu bude, když uživatel bude chtít vytvořit záznam s novým parkovištěm. Na obrázku (Obr. 5.1) je prvotní návrh klientské části.

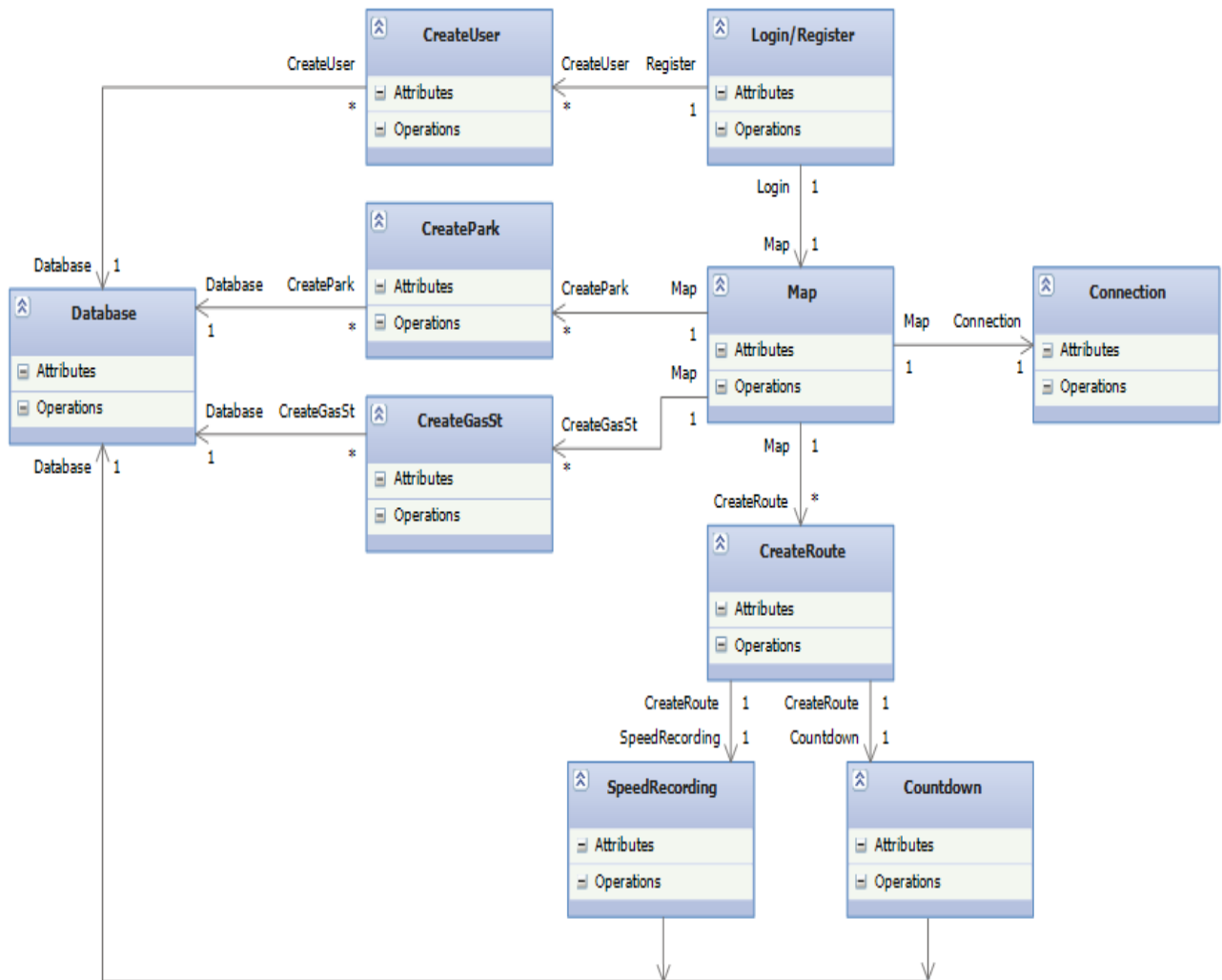
5.1 Klientská část

K vývoji bude použito vývojové prostředí Eclipse, do kterého bylo použito Android SDK. Aplikace jako prvotně je vyvíjena pro verzi Androidu 2.3.4. Tuto verzi jsem si vybral, protože vlastním mobilní telefon s touto verzí systému a tudíž nejčastější testování proběhne na tomhle mobilním zařízení.

Při analýze klientské části jsem zohlednil všechny požadavky, které by měla aplikace splňovat, aby byla pokud možno co nejvíce uživatelsky přívětivá. Požadavky jsou: intuitivní uživatelské rozhraní, zobrazování poloh na mapě, záznam trasy a rychlosti, odpočet dob řízení, vkládání vlastních bodů zájmů a komunikace se serverem.

Celá aplikace je postavena na práci s mapou. Prvotní myšlenka byla použít mapové podklady, které jsou přístupné offline. Ovšem po prozkoumání možností, volba padla na mapové podklady od společnosti Google i když jsou přístupny pouze v režimu online. Tyhle mapy jsou velice podrobné a proto splňují požadavky pro danou aplikaci. Navíc mapové podklady společnosti Google jsou již přímo ve většině zařízení podporující mobilní platformu Android. V dnešní době, když je velice dobré pokrytí mobilním internetem, není na škodu použití zvolených mapových podkladů.

Při návrhu grafické rozhraní byl kladen důraz na jednoduchost, intuitivnost, přehlednost a co možná nejjednodušší ovládání dané aplikace. Výsledná mobilní aplikace bude převážně používána v automobilu nebo venku mimo něj, takže musí být velice dobře čitelná.



Obr. 5.1 Diagram klientské části

5.2 Serverová část

K implementaci serverové části bude použito vývojové prostředí NetBeans. Tato část bude implementována v programovacím jazyce Java s využitím GlassFish serveru. Jednotlivé záznamy budou obsluhovány pomocí Entity Managed Bean. Celý server bude pro testovací účely spuštěn na localhostu mého počítače.

Analýza serverové části byla značně jednodušší oproti klientské části. Server se stará jenom o ukládání příchozích záznamů do vytvořené databáze. Když na server přijde klientem vytvořený queryString, server ho rozkóduje a výsledné hodnoty vloží do příslušných tabulek databáze. Jakmile uživatel odešle z mobilu záznam s novou trasou, automaticky je odeslán i soubor kml s projetou cestou a textový soubor se záznamem rychlostí dané trasy. Server tento záznam vloží do tabulky s projetými cestami a vytvoří u tohoto záznamu odkaz na mapu. Po kliknutí na odkaz se v novém okně internetového prohlížeče vykreslí na Google mapách projetá cesta. Obdobně tomu bude i při vložení čerpací stanice nebo parkoviště.

Co se týče grafického rozhraní serverové části, tak to je velice jednoduché a celkově nenáročné, což se projevuje v rychlosti procházení jednotlivých záznamů. GUI bude obsahovat jenom tabulku se záznamy z databáze + odkazy na mapu. Bude také možné vytvořit nový záznam. Vytvořit půjde i nová trasa, ale už nebude obsahovat projetou cestu a soubor s rychlostmi.

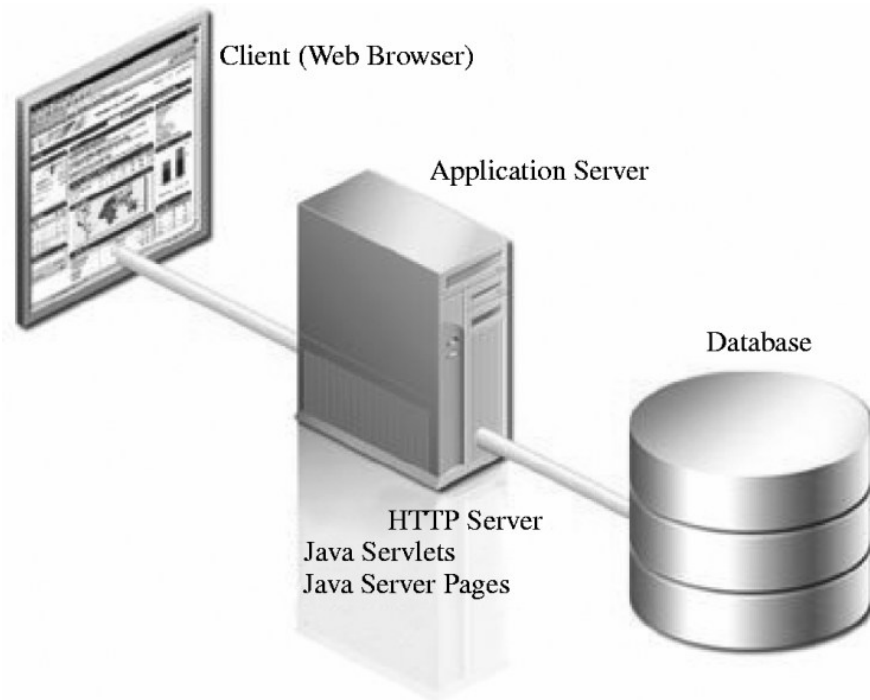
Při samotné implementaci aplikace, budu vycházet z třívrstvého modelu aplikace

(Obr. 5.2)[8]. Model je rozdělen na tři části:

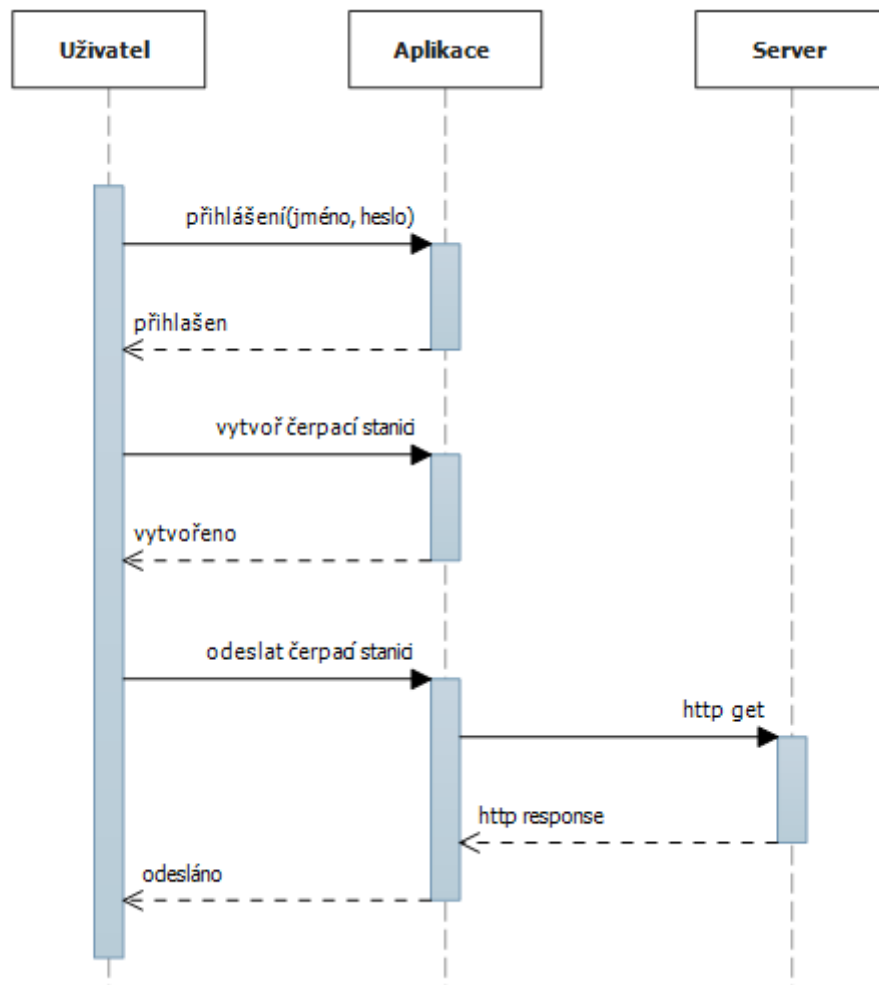
Prezentační úroveň – uživatel, který používá svůj internetový prohlížeč

Aplikační vrstva – server s veškerou řídicí logikou

Databáze – obsahuje jednotlivé záznamy v databázi



Obr. 5.2 Třístupňová architektura



Obr. 5.3 Diagram aktivit – přihlášení uživatel s vytvořením a odesláním záznamu

6 Implementace

V této kapitole se budu věnovat samotné implementaci mé aplikace. Implementace je rozdělena do dvou částí. Klientskou část a část serverovou.

6.1 Klientská část

Aplikace, kterou má uživatel nainstalovanou na svém zařízení podporující operační systém Android. Aplikace vyžaduje přístup k úložišti daného zařízení, přístup k internetu pro následné odeslání vytvořených záznamu na server a také používá GPS daného zařízení.

6.1.1 Databáze

Před uložením jednotlivých záznamu je potřeba si vyčlenit místo a způsob, jakým budou záznamy ukládány. Já jsem zvolil způsob takový, že se v zařízení vytvoří lokální SQLite databáze, do které se vytvořené záznamy vkládají. Při vytváření databáze se hned vytvoří jednotlivé tabulky pro záznamy. Databáze obsahuje čtyři tabulky:

čerpací stanice – Do této tabulky se ukládají záznamy s vytvořenými čerpacími stanicemi (Obr. 6.1). Do tabulky se ukládá id, název, aktuální poloha tzn. latitude (zeměpisná šířka) a longitude (zeměpisná délka), den, měsíc, rok a jednotlivé ceny nafty, benzínu a adBlue.

parkoviště – Obsahuje záznamy s vytvořenými parkovišti (Obr. 6.2). Ukládá se id, místo (obecný popis), aktuální poloha tzn. latitude(zeměpisná šířka) a longitude(zeměpisná délka), počet parkovacích míst, počet volných míst, čas (hodiny, minuty) a datum (rok, měsíc, den).

trasy – Tato tabulka obsahuje záznamy tras, které uživatel projel. Je zde uloženo id, čas zahájení jízdy, místo vyjetí, stav tachometru, čas konce jízdy, místo dojetí, stav tachometru a datum (rok, měsíc, den).

uživatelé – Už podle názvu vyplývá, že tabulka obsahuje záznamy o uživateli. Do tabulky se ukládá uživatelské jméno a heslo, které si uživatel vytvořil při první registraci.

Souběžně s vložením záznamu do databáze příslušné tabulky se vytváří i textový soubor s názvem tabulky. Do tohoto souboru jsou ukládány jednotlivé záznamy oddělené čárkou + jako poslední hodnota se uloží i uživatelské jméno uživatele, který daný záznam vytvořil. Do tohoto souboru se může ukládat více záznamů. Tento soubor je vytvářen proto, když chce uživatel odeslat

vytvořené záznamy na server. Při odesílání dat jsou jednotlivé záznamy vkládány jako parametry do URL.

Obr. 6.1 Nový záznam
čerpací stanice

Obr. 6.2 Nový záznam
parkoviště

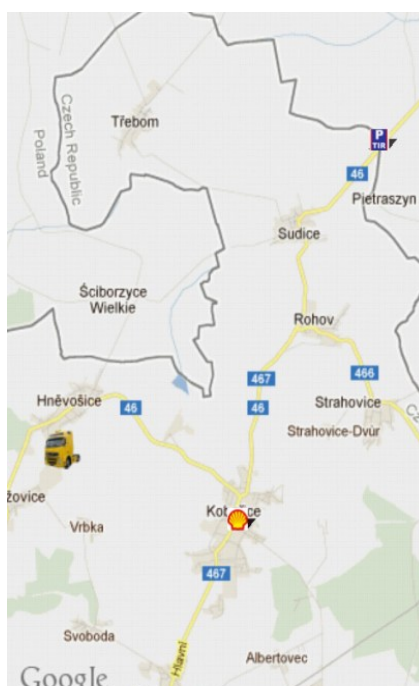
6.1.2 Práce s GPS a mapou

Abychom mohli vůbec pracovat se systémem GPS je implementována třída `LocationManager` a jeho rozhraní `LocationListener`. V `LocationListeneru` je připravená metoda `onLocationChanged`. Tato metoda se volá, když se změní poloha zařízení. `LocationListener` se zapíná pomocí metody `requestLocationUpdates`. Tato metoda obsahuje parametry provider, vzdálenost a čas. Providerem určujeme, zdali se bude určovat poloha pomocí GPS nebo pomocí mobilní či WIFI sítě. Vzdálenost určuje, jak daleko se musí zařízení posunout, aby byla načtena další okamžitá poloha. Tento údaj se udává v metrech. Čas, to je interval, který musí uplynout, než se načte aktuální poloha, údaj je v milisekundách.

Na mapě se uživateli zobrazuje jeho aktuální poloha pomocí markeru ve tvaru kamionu (Obr. 6.3), dále čerpací stanice a parkoviště. Tyto objekty jsou na mapě zobrazovány pomocí třídy `Overlay`, která je součástí Android SDK. Jednotlivé čerpací stanice a parkoviště jsou ukládány do

List<Overlay>. Tyto objekty jsou následně ukázány, když MapView zavolá `getOverlays()`. Jednotlivé záznamy se zobrazí na mapě ihned po vytvoření nového záznamu uživatelem.

Po kliknutí na daný objekt v mapě, ať už se jedná o parkoviště nebo čerpací stanici, se zobrazí u objektu bublina, která obsahuje záznamy. U čerpací stanice to je název a cena nafty, benzinu a adBlue. Po kliknutí na ikonu parkoviště se zobrazí poloha, počet parkovacích míst a aktuální počet volných míst na parkovišti.



Obr. 6.3 Mapa

6.1.3 Záznam nové trasy

Když uživatel zahájí novou trasu, spustí se současně odpočet dob řízení (Obr. 6.5), záznam rychlosti a záznam trasy. Okamžitě se začne odpočítávat čtyři a půl hodiny, což je maximální možná doba jízdy řidiče v kuse. Po celou dobu se do souboru ukládá projetá cesta. Do druhého souboru se ukládá rychlost, jakou se vozidlo pohybuje.

Odpočet dob řízení je znázorněn na `progressBaru`, který ukazuje jednotlivé doby do konce daného úkonu. Samotné kroky `progressBaru` jsou implementovány pomocí třídy `Handler()`.

```
Handler handler=new Handler()
{
    @Override
    public void handleMessage(Message msg)
    {
        total=total+1;
        String minuty=String.valueOf(total).toString();
        txt.setText(minuty + " / 270");
        bar.incrementProgressBy(1);
    }
}
```

Tento kód běží ve vlákne, které je spouštěno v cyklu `for` o 270 krocích, což je počet minut, kolik trvá doba řízení. Vlákno se po každém kroku uspí na jednu minutu, aby to odpovídalo počtu kroků, které se musí provést v cyklu. Možnost jaká je momentálně realizována v aplikaci, je nejvíce užívána mezi řidiči.

Projektá cesta se ukládá do souboru `.txt`, v kterém se ukládají hodnoty zeměpisné šířky, zeměpisné délky a nadmořské výšky. Ukládají se buď každých 500 metrů nebo co jednu minutu. Tyto tři záznamy jsou odděleny čárkou, po třetím záznamu následuje mezera a další záznam. Takovou strukturu souboru jsem zvolil, protože následně je celý soubor přečten a nakopírován se všemi hodnotami do výsledného souboru `.kml`. V tomto souboru už je připravená struktura a pouze se zde vloží všechny záznamy s projetou cestou. Soubor `.kml` obsahuje nadefinovanou strukturu s jmenným prostorem `http://earth.google.com/kml/2.x`.

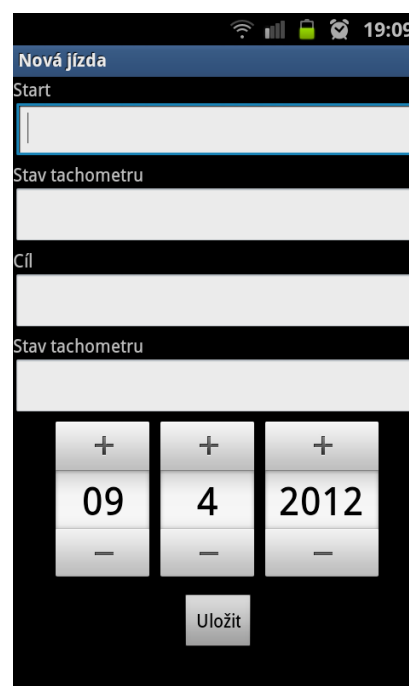
Možná se to zdá jako krkolomné řešení tohoto překopírování souboru z jednoho do druhého. Ovšem, když jsem zkoušel ukládat hodnoty hned do souboru `.kml` tak to trvalo daleko déle než, když se hodnoty ukládají do textového souboru a následně se vloží do výsledného souboru, který je tak připraven k odeslání na server.

Záznam rychlosti je zaznamenávána aktuální rychlost + maximální rychlost. Rychlost se získává ze systému GPS a je udávána v metrech za sekundu, proto je vynásobená hodnotou 3.6, aby byla výsledná rychlost v kilometrech za hodinu. Tyhle údaje by měly sloužit pouze jako orientační pro možnou kontrolu s údaji z tachografu vozidla, které by měly být přesnější než tyto záznamy.

Po skončení druhé jízdy se zobrazí formulář (Obr. 6.4), kde řidič vyplní potřebné údaje a záznam je vložen do databáze a do souboru k případnému odeslání na server.



Obr. 6.5 Odpočet doby jízdy



Obr. 6.4 Nový záznam trasa

6.2 Serverová část

Při implementaci serverové části jsem postupoval podle mého prvotního návrhu a tak je tato kapitola rozdělena do 3 částí. Po prohlížení záznamu na serveru stačí uživateli pouze internetový prohlížeč.

Grafické rozhraní je první část (Obr. 6.7). Grafické rozhraní na serveru je velice jednoduché a přehledné. Bylo vytvářeno pomocí JSF. Na stránce se zobrazuje tabulka s jednotlivými záznamy. Je také možnost vložit nový záznam. U každého záznamu je i odkaz s mapou. Na mapě se zobrazuje poloha čerpací stanice nebo parkoviště. U záznamu s projetými trasami se zobrazí odkaz na mapu, na které se vykreslí projetá trasa (Obr. 6.6). Projetá cesta je zobrazená žlutou barvou.

```
<h:dataTable value="#{mBenzinka.findAllBenzinky()}" var="b"
    styleClass="table" headerClass="table-header"
    rowClasses="table-odd-row,table-even-row">
  <h:column>
    <f:facet name="header">Název</f:facet>
    #{b.nazev}
  </h:column>
    ..... Další jednotlivé záznamy
  <h:column>
    <h:outputLink id="link1"
    value="http://maps.google.com/maps">
    <f:param name="q" value="#{b.latitude}
    #{b.longitude}" />
    <h:outputText value="Mapa" />
    </h:outputLink>
  </h:column>
</h:dataTable>
```

Zdrojový kód je ukázka grafického rozhraní stránky s čerpacími stanicemi. Na začátku tabulky jsou definovány CSS styly. Následují jednotlivé sloupce tabulky se záznamy vložených z databáze. Poslední sloupec je odkaz na Google mapy, na kterých se zobrazuje poloha čerpací stanice. Jako parametry k odkazu jsou vloženy hodnoty zeměpisné šířky a zeměpisné délky.

Název	Latitude	Longitude	Rok	Měsíc	Den	Nafta	Benzin	adBlue	Uživatel	
Shell_Orlova	49.866187	18.422847	2012	4	20	36.0	37.0	0.0	Tom	Mapa
Shell	49.774942	18.097411	2012	4	20	37.0	39.0	11.0	Tom	Mapa

Přidat nový záznam

Název :

Latitude :

Longitude :

Rok :

Měsíc :

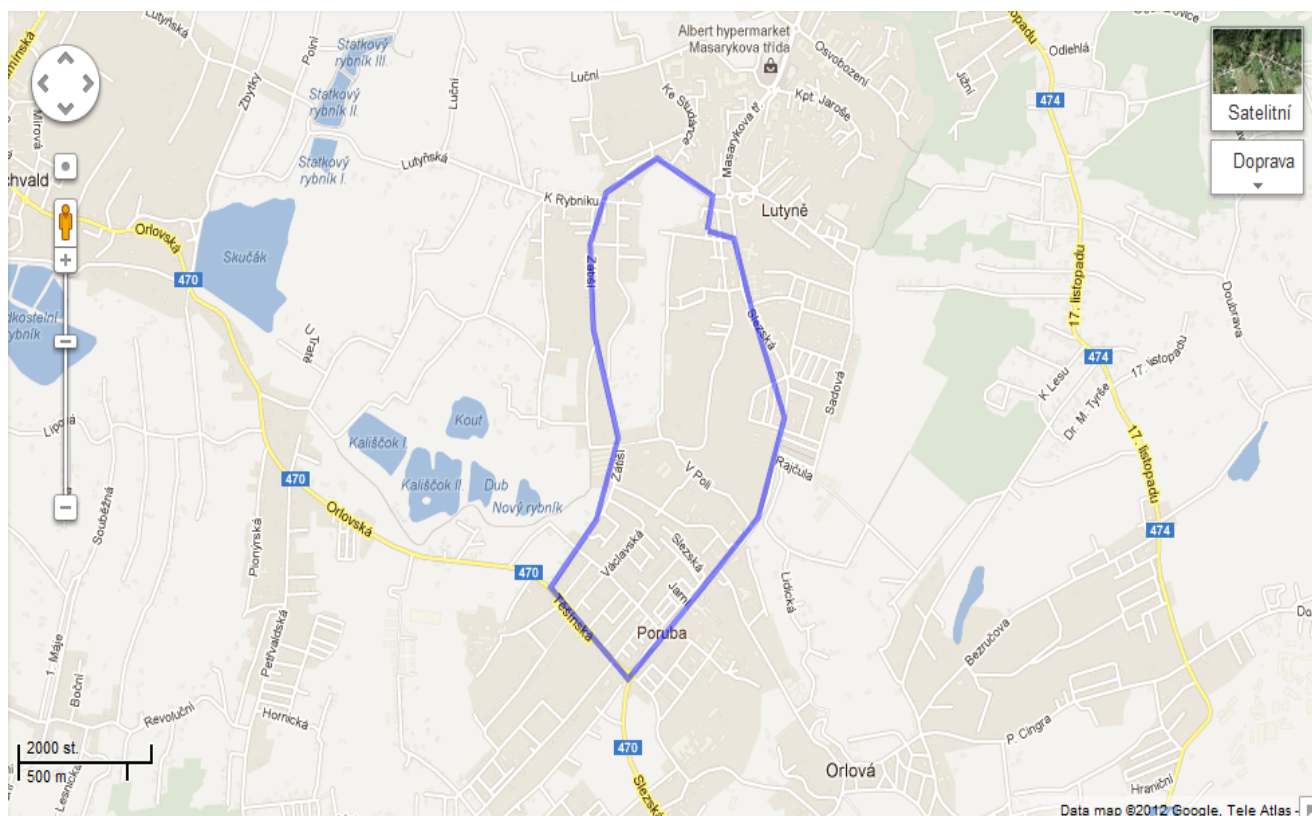
Den :

Nafta :

Benzin :

adBlue :

Obr. 6.7 Grafické rozhraní - server



Obr. 6.6 Zobrazení projeté cesty

Řídící logika je druhou částí. Tato část obsahuje, jak už z názvu vyplývá, jednotlivé metody pro ovládání stránek, práci s databází apod. Obsahuje také třídu, která má na starost komunikaci s klientem.

Třída pro komunikaci s klientem aby mohla přijatá data vložit do databáze, používá pro připojení k databázi driver `org.apache.derby.jdbc.ClientDriver`. Obsahuje také metody pro vkládání jednotlivých příchozích záznamů do příslušných tabulek databáze. kde tříděny a vkládány do tabulek pomocí parametru `t` viz. 4.3Mé řešení sdílení

Databáze je částí číslo tři. Databáze na serveru je velice podobná s databází, kterou má klient vytvořenou ve svém zařízení. Obsah tabulek je stejný a navíc je u každé tabulky přidán jeden sloupec, který obsahuje uživatelské jméno uživatele, který daný záznam vytvořil a odeslal na server.

6.3 Nasazení do provozu

Pro případné budoucí komerční nasazení programu již nebude možné používat server na localhostu jak tomu bylo do teď. Je tedy možnost použít webhosting, na kterém by běžel doposud vytvořený server pomocí JSP. S tímto řešením by mohl nastat problém, kdyby na serveru byly tisíce záznamů. Proto s největší pravděpodobností by byl zvolen webhosting, který by podporoval MySQL databázi + PHP.

Odesílané záznamy by byly posílané jako celé soubory. Na serveru následně rozpársovány a vloženy do příslušných tabulek databáze. Toto řešení jsem bral v potaz už při prvotním návrhu, proto se už nyní záznamy ukládají do souboru.

Tento způsob již byl vyzkoušen, zdali je vhodným řešením. Omezovaly mě jenom znalosti PHP programování. Základní funkčnost, jako je odesílání dat a následný příjem a rozpársování s uložením do databáze, bylo použitelné a bude bráno v potaz při dalším vývoji aplikace.

7 Testování

Při prvotním psaní kódu a jeho následné spuštění probíhalo pomocí emulátoru, který je součástí Android SDK. Když už výsledná aplikace měla zčásti ucelenou formu a byla již alespoň trochu funkční, probíhal test na mém vlastním mobilním telefonu. Jedná se o mobilní telefon Samsung Galaxy S2. Mobilní telefon je vybaven operačním systémem Android 2.3.4 GINGERBREAD, dvou jádrovým procesorem 1,2Ghz a operační paměti 1GB.

Při spuštění aplikace v emulátoru bylo možné pozorovat, jak při složitějších operacích se prodlužuje délka, než se operace provede. Totéž lze říci o vykreslování, přiblížení, oddálení a pohybu mapy.

Naproti tomu, když je aplikace spuštěna v mém mobilním telefonu, nic takové nebylo zaznamenáno. Operace probíhají bez pozorovatelné prodlevy. Pohyby v mapě jsou limitovány pouze rychlostí připojení k internetu. I když připojení k internetu je stejné jak v emulátoru tak v mobilním telefonu, vykreslování map probíhá rychleji v mobilním telefonu než v emulátoru.

Po téhle zkušenosti jsem začal vždy při vývoji aplikace vlastní mobilní telefon připojovat pomocí USB k počítači a spouštění probíhalo v něm a ne v emulátoru. V mobilním telefonu bylo jenom potřeba povolit ladění pomocí USB. Následné ladění probíhalo už jenom tímhle způsobem.

Co se týče testování práce s GPS, to už bylo testováno pouze na mobilním zařízení.

Problém nastal až tehdy, když jsem si na svém zařízení vytvořil nějaký záznam, ať už se jednalo o čerpací stanici nebo parkoviště. Po vytvoření a následném zobrazení na mapě neodpovídala poloha. Poté co jsem zkusil vytvořit další záznam a použil jsem aktuální polohu, zjistil jsem, kde nastal problém. Problém byl v tom, že emulátor používal pro oddělení desetinných míst tečku. Kdežto můj mobil používá čárku. Prvotní testy probíhali v emulátoru, kde to fungovalo správně, tudíž jsem tomu nevěnoval další pozornost. Po úpravě zdrojových kódů aplikace funguje správně.

Další nemile překvapení nastalo, když jsem si pro účely testu nechal zobrazovat na displeji aktuální rychlost. Testování probíhalo v autě. Rychlost co se zobrazovala, za žádných okolností nepřekročila hodnotu 26. Tahle hodnota se mi zdála nějaká nízká v porovnání s tachometrem auta při testovací jízdě. Věděl jsem, že bude nějaká odchylka mezi hodnotami z GPS a tachometrem automobilu, ale že až tak velká to jsem nečekal. Po zhlédnutí dokumentaci k danému API jsem zjistil, že rychlost získána z GPS je v metrech za sekundu. Proto ta velká odchylka od tachometru automobilu. Následně byla rychlost vždy vynásobená hodnotou 3.6 a výsledná rychlost už je udávána v kilometrech za hodinu. Po téhle úpravě a následné další testovací jízdě už bylo vše v naprostém pořádku.

Celý server je spuštěn na localhostu. Pro dosavadní testování aplikace mi tahle možnost stačila. Pro další rozšiřování funkčnosti by tato možnost již nebyla dostačující. viz. 6.3 Nasazení do provozu

Z celkového testování této aplikace vyplývá, že v aktuálním stádiu, v jakém se momentálně nachází, je funkčnost na velice dobré úrovni. Aplikace je zcela funkční a požadavky, které se od aplikace očekávaly, splňuje.

8 Porovnání s konkurenčními aplikacemi

V této kapitole se budu snažit porovnat aplikace z kapitoly 3 s mou aplikací. V třetí kapitole byly srovnány pouze programy mezi sebou, ale nebylo tam srovnání s mou, již vytvořenou aplikací.

Do tabulky (Tabulka 3.1) porovnání aplikací jsem přidal i svou aplikaci pro rychlejší přehled nejdůležitějších funkcí. (Tabulka 8.1)

	LogiAssist	TruckerTimer	City Traffic	tudyNE	Má aplikace
Databáze parkovišť / čerpacích stanic	ano / ano	ne / ne	ano / ne	ne / ne	ano / ano
Odpočet dob řízení	ne	ano	ne	ne	ano
Dopravní informace	ano	ne	ano	ano	ne*
Platforma	Android	Android	Android	Android	Android
Zpoplatnění	ne	ano	ne	ne	ne

Tabulka 8.1: Porovnání s mou aplikací

*S touto funkcí se počítá do dalšího vývoje aplikace.

Z mého hlediska, z vybraných aplikací, se mi nejvíce líbí aplikace LogiAssist. Tato aplikace obsahuje nejvíce funkcí podobných s mou aplikací. TruckerTimer je zase aplikace zaměřena na velice podrobný odpočet dob řízení, tudíž z části obsahuje funkci, kterou mám implementovanou i ve své aplikaci. Ostatní dvě aplikace jsou zaměřené pouze na dopravní informace, takže doposud nespĺňují s mou aplikací nic společného. V další fázi vývoje budou dopravní informace součástí, takže se budou dát porovnat.

Nejlepší aplikace by vznikla sloučením všech čtyř aplikací do jedné. Taková kombinace by byla velice dobrým komplexním řešením. Nebudu se tajit myšlenkou, že k takovému závěru bych se chtěl dostat s mou aplikací.

9 Závěr

Cílem bakalářské práce bylo vytvořit mobilní aplikaci pro platformu Android, která by měla sloužit profesionálním řidičům případně i jejím dispečerům. Výsledná aplikace umí to, co bylo od ní očekáváno, tudíž se dá říct, že úkol byl splněn. Toto téma jsem si vybral, protože v tomto oboru se dost orientuji. Další důvod byl i ten, že jsem si mohl rozšířit své dosavadní znalosti o programování pro zařízení s operačním systémem Android.

V budoucnu bych chtěl ve vývoji této aplikace pokračovat. Určitě bych rozšířil celkovou funkčnost na obou částech této aplikace a nemalá část by byla věnovaná i celkovému zabezpečení.

V první řadě by byl server implementován na webhosting a s tím by souvisely změny v komunikaci mezi serverem a klientem. Dále by přišla řada na klientskou část.

Tato část by byla rozšířená o další záznamy, jako jsou objízdné trasy, uzavírky, dopravní nehody a další situace, které mohou na cestě nastat. Mohla by se rozšířit funkčnost i za hranice naší republiky, to by znamenalo zaznamenat do mapy ambasády, apod. Další možná funkce by byla vzájemná komunikace mezi řidiči a zobrazování jejich polohy na mapě, možnost nouzového volání popř. linky na zelenou vlnu. Rozšířil by se i odpočet dob řízení.

Aplikace nebude momentálně vystavena k veřejnému užívání, protože není dostatečně zabezpečená a bude sloužit prozatím jen pro soukromé účely a pro účely testování. S budoucím vylepšením by stálo za pokus aplikaci nasadit do ostrého provozu. Z počátku by to byla jen menší skupina vybraných uživatelů, kteří by informovali o vlastních zkušenostech z aplikace, popřípadě sdělovali svoje návrhy a námitky. Postupem času by se počet uživatelů mohl zvětšovat.

Použitá literatura

- [1] BURNETTE, Ed. *Hello, Android: introducing Google's mobile development platform*. Version 2008-12-5. Raleigh, N.C.: Pragmatic Bookshelf, c2008, 218 s. ISBN 978-193-4356-173.
- [2] MEIER, Reto. *Professional Android 2 application development: introducing Google's mobile development platform*. Version 2008-12-5. Indianapolis: Wiley, c2010, 543 s. Wrox programmer to programmer. ISBN 978-0-470-56552-0.
- [3] MURPHY, Mark L. *Android 2: průvodce programováním mobilních aplikací*. Vyd. 1. Brno: Computer Press, 2011, 375 s. Wrox programmer to programmer. ISBN 978-80-251-3194-7.
- [4] Android Developers [online]. [cit. 2012-04-25]. Dostupné z: <http://developer.android.com/>
- [5] Cloud Computing. [online]. [cit. 2012-04-26]. Dostupné z:
<http://www.cloudcomputing.cz/index.html>
- [6] HTML v příkladech. [online]. [cit. 2012-04-26]. Dostupné z:
<http://www.nti.tul.cz/~satrapa/docs/wwwprikl/html9.html>
- [7] AETR. [online]. [cit. 2012-04-30]. Dostupné z: <http://doprava.vpraxi.cz/aetr.html>
- [8] Emerald. [online]. [cit. 2012-04-30]. Dostupné z:
http://www.emeraldinsight.com/content_images/fig/3330080202004.png
- [9] LogiAssist. [online]. [cit. 2012-04-30]. Dostupné z:
<http://www.logiassist.de/web/guest/die-app>
- [10] TruckerTimer. [online]. [cit. 2012-04-30]. Dostupné z:
<https://play.google.com/store/apps/details?id=com.spottydog.tt>
- [11] City Traffic (ČR, Evropa...). [online]. [cit. 2012-04-30]. Dostupné z:
<https://play.google.com/store/apps/details?id=org.myklos.citytraffic>
- [12] Dopravní informace a nehody. [online]. [cit. 2012-04-30]. Dostupné z:
<http://www.tudyne.cz/>

Přílohy

- I. Aplikace v souboru Asistent řidiče.zip
- II. Uživatelská příručka v souboru Příručka.pdf
- III. Programátorská dokumentace.zip