UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Zoran Fetah

RAZŠIRITEV LOKACIJSKE STORITVE TRIP TRACKER ZA SATELITSKE NAPRAVE

DIPLOMSKO DELO
NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Igor Rožanc

Ljubljana, 2011

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00018/2010
Datum: 01.10.2010

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ZORAN FETAH**

Naslov: **RAZŠIRITEV LOKACIJSKE STORITVE TRIP TRACKER ZA SATELITSKE NAPRAVE**

UPGRADE OF TRIP TRACKER LOCATION BASED SERVICE FOR SATELLITE DEVICES

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Glavna značilnost lokacijskih storitev je zmožnost določanja lokacije uporabnika. Primer tovrstne storitve je Trip Tracker, ki zna slediti določeni napravi uporabnika in hraniti, analizirati in posredovati podatke o tem. Za prenos podatkov o sledenju se uporablja mobilno omrežje.

V diplomski nalogi preučite in predstavite področje ter na ustrezen način razširite to storitev še za satelitske naprave. V ta namen zasnujte, izvedite in opišite spremembe ustreznih nastavitev in predvsem programski modul storitve. Na koncu novo storitev preverite v praksi in analizirajte opravljeno delo.

Mentor:

viš. pred. dr. Igor Rožanc

Dekan:

prof. dr. Nikolaj Zimic

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Zahvala

Zahvaljujem se dr. Igorju Rožancu za vodenje in vse nasvete pri izdelavi diplomske naloge. Zahvaljujem se tudi Gregorju Rebolju in ostalim zaposlenim v podjetju Klika d.o.o., za pomoč in svetovanje pri izdelavi diplomskega dela.

Še posebej pa se zahvaljujem svoji družini, prijateljem in Nataši, za podporo v času študija in za pomoč pri izdelavi diplomskega dela.

Kazalo

Slike

Seznam uporabljenih kratic in pojmov

Almanah	Okviren podatek o tirnicah satelitov, ki vključuje tudi model za popravke napak. Vsi sateliti oddajajo podatke o tirnicah za vse satelite.
Efemeride	Bolj podrobni podatki o odstopanju satelitov z načrtovanih tirnic. Vsak satelit oddaja le podatke za sebe.
GPS	Global Positioning System - Satelitski sistem za določanje položaja.
GSM	Global System for Mobile Communications (sprva tudi Group Spécial Mobile) – Globalni sistem za mobilne komunikacije
HTTP	Hipertekstni transportni protokol – komunikacijski protokol, ki se uporablja za prenos informacij po spletu
IMEI	International Mobile Equipment Identity – enolična številka, ki jo imajo vse GSM naprave.
SIM	Subscribers Identity Module - Modul za identificiranje naročnika
Spot Personal Tracker	GPS naprava podjetja Spot.
TripTracker	Sledilni sistem oz. ena izmed lokacijsko osnovanih storitev, ki je dostopna uporabnikom.
URL	Uniform Resourse Locatior - Enolični določevalec vira ali običajno naslov spletne strani.
USB	Universal Serial Bus – Način vzpostavitve komunikacije med računalnikom in perifernimi napravami.
XML	Extensible Markup Language – jezik za opis sheme podatkov.

Povzetek

Diplomsko delo opisuje postopek razširitve sledilno lokacijske storitve imenovane TripTracker podjetja Klika d.o.o. Zajema proces razvoja modula od zajema zahtev preko razvoja do testiranja. Podrobneje opisuje implementacijo modula in integracijo tega preko obstoječih vmesnikov.

Razširitev sistema omogoča uporabnikom širšo izbiro naprav za sledenje. Naprava za sledenje in sistem morata biti najprej ustrezno nastavljena, da lahko ta pošilja informacije o lokaciji na strežnik. Od to jih modul prenese, ustrezno pretvori in shrani v zbirko podatkov. Pregled podatkov oziroma opravljenih sledilnih poti je uporabnikom omogočeno preko spletne aplikacije TripTacker.

Uvod razloži pojem lokacijskih storitev, njihovega razvoja, prednosti, slabosti in možnosti uporabe. Sledi opis delovanja sistema GPS in sistema TripTracker. Sama izvedba je razdeljena na dva dela. V prvem delu je opis integracije še ne implementiranega modula v obstoječi sistem TripTracker, v drugem delu pa je opisana programska rešitev modula.

Diploma opisuje tudi glavna orodja in tehnologije, ki smo jih uporabili pri izdelavi modula. Na koncu postavimo tudi smernice za nadaljnje delo.

Ključne besede

GPS, lokacijske storitve, Spot, TripTracker

Abstract

This work describes an upgrading process of a location based service TripTracker made by company called Klika d.o.o. Specifically it covers the process of developing and integration of module to the backend system via existing interface. It includes all phases from requirements definiton to testing.

Our work will give user a wider choices of devices to use for tracking. Tracking device and system must be properly configured, as device sends tracking points to the server. From this point the module can transfer, convert and store them to database. Review of track is possible for user through a web application.

Introduction explains the meaning of the location based service, its development, advantages and potential uses, followed by short description of how GPS system and TripTracker service works. The realization is devided into two parts. First part describes integration of the module into existing TripTracker system, while the second part describes a software solution of the module.

In the work we describe main tools and techniques used in the development of module, and the guidelines for further work as well.

Key words

GPS, location based services, Spot, TripTracker

1. Uvod

V zadnjem času se informacijska tehnologija zelo hitro razvija. Računalniki, mobilna telefonija in razne storitve in aplikacije nam na takšen ali drugačen način omogočajo lažje in boljše življenje. Ena izmed pomembnejših in hitrejše rastočih vrst storitev so tudi lokacijske storitve.

Kot nam že ime pove, gre za storitve, pri katerih potrebujemo napravo, ki zna določiti položaj uporabnika in ga nato posredovati naprej, drugače rečeno nam zna na podlagi tega položaja podati koristno informacijo (npr. turista zanimajo podatki o najbližjih mestnih znamenitostih). Ena izmed lokacijsko osnovanih storitev je tudi TripTracker, ki omogoča sledenje vozil, oseb ali živali z napravo za določanje položaja. Teh je veliko in ker ne delujejo vse na isti način, smo se odločili razširiti TripTracker z podporo za napravo, katere način delovanja v TripTracker sistemu še ni podprt, da bi omogočili uporabnikom širši nabor naprav za sledenje. Odločili smo se za napravo imenovano Spot Personal Tracker, ker so obstoječi uporabniki TripTracker storitve izrazili željo po uporabi te naprave.

2. Lokacijske storitve, sistemi in uporabljene tehnologije

2.1. Lokacijske storitve

Lokacijsko storitve (ang. location based services) so se razvile na podlagi zahtev služb za prvo pomoč. Njihova želja je bila, da bi bila ob klicu poleg številke kličočega dostopna tudi informacija o njegovi lokaciji.

Danes nam lokacijske storitve omogočajo recimo:

- Pomoč na cesti – če ima uporabnik v avtu nameščeno GPS napravo, ki vsebuje senzorje za zaznavo trka, lahko v kombinaciji z mobilnim telefonom ali kar preko svoje GSM enote o tem obvesti ustrezne službe s točnim podatkom o lokaciji, kar omogoča hitrejšo intervencijo.

- Sledenje – spremljanje določenega predmeta, živali ali osebe. Najbolj popularno in ekonomsko upravičeno je sledenje vozilom. Pri tem se lahko uporablja mobilni telefon voznika ali pa v avtomobilu vgrajena namenska GPS naprava. Vgrajene GPS naprave omogočajo, da se na ta način lažje najde odtujene avtomobile ali nadzirati službeno pot zaposlenih.

- Navigacija – s pomočjo podatka o položaju uporabnika se lahko uporabnika usmerja in mu tako čim bolj olajša pot iz točke A v točko B. Če želimo tako storitev podpreti moramo nujno poznati možne povezave med obema točkama, kar praviloma pomeni, da je na navigacijski napravi potrebno imeti zemljevid celotnega cestnega omrežja in ga po potrebi tudi posodabljati.

- Zabavne storitve –iskanje prijateljev ali zanimivih točk. Trenutna socialna omrežja omogočajo, da s pomočjo telefona ali računalnika objavimo svojo lokacijo ter iščemo prijatelje oz. zanimive točke, ki se nahajajo v naši bližini. V zadnjem času se pojavlja vedno več iger, za mobilne telefone z vgrajenim GPS sprejemnikom, ko mora uporabnik najti nek predmet oz. lokacijo glede na svoj položaj.

2.2. Sistemi za določanje pozicije

Ameriški GPS sistem je trenutno edini popolnoma delujoči svetovni sistem, v razvoju pa so tudi sistemi drugih držav, ki bi rade postale bolj neodvisne od Združenih držav Amerike. Ameriški GPS sistem omogoča zelo veliko natančnost pri določanju položaja, vendar je ta zares dostopna samo za vojaške namene preko šifriranega signala. [2]

Globalni sistemi za določanje položaja so:

- Ameriški GPS

- Ruski GLONASS [3]

- Kitajski Compass [4]

- Galileo – projekt Evropske unije v sodelovanju z Indijo, Izraelom, Južno Korejo, Kitajsko, Marokom, Savdsko Arabijo in Ukrajino. [5]

2.2.1. Sistem za določanje pozicije GPS

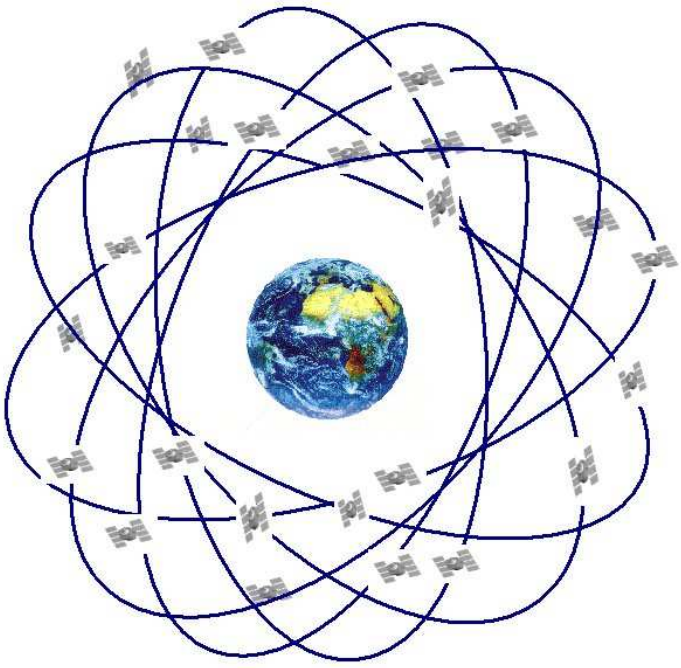
GPS je razvilo ameriško obrambno ministrstvo leta 1978, od leta 1983 pa je njegova uporaba dovoljena tudi za civilno prebivalstvo. GPS sestavljajo trije segmenti:

- uporabniški – GPS sprejemniki,

- vesoljski – sateliti, ki krožijo nad Zemljo ter

- nadzorni – centri na Zemlji, ki skrbijo za nemoteno delovanje satelitov.

GPS sistem sestavlja najmanj 24 satelitov, ki krožijo v šestih enakomerno razmaknjenih tirnicah okoli Zemlje, kot je prikazano na sliki 1. Obhodni čas satelitov je 12 ur. Povprečna življenjska doba satelita je 7 let in pol, tako do sedaj je bilo utirjenih že 60 satelitov. Sateliti so 20.200 km nad zemeljsko površino. S poljubne točke na Zemlji je v določenem trenutku vidnih od 6 do 11 satelitov, za uspešno določitev položaja morajo biti vidni vsaj 4 sateliti. [6]

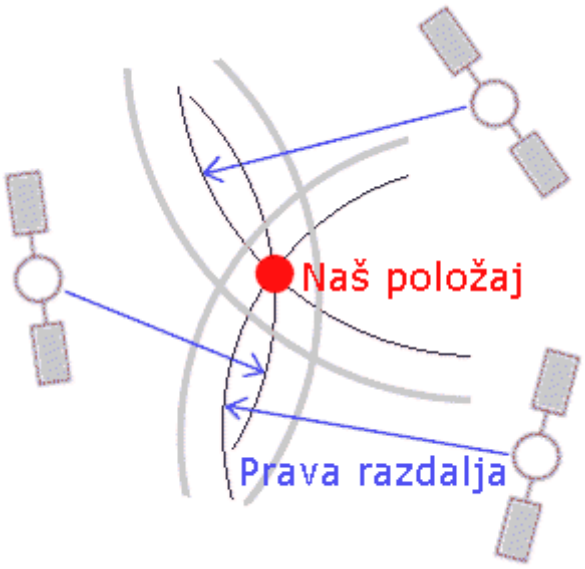


Slika 1: Tirnice GPS-satelitov.

2.2.1.1. Delovanje

Vsak satelit za delovanje uporablja zelo natančno atomsko uro. Vsi sateliti neprestano oddajajo čas po svoji uri ter svoj položaj na tirnici (t.i. almanah), kot tudi položaje vseh ostalih satelitov, saj so ti zaradi stalnega kroženja po tirnici določljivi. Zaradi zunanjih dejavnikov lahko pride do manjših odstopanj od načrtovane poti, te pa odkrivajo v nadzornih centrih na zemlji. Centralno nadzorno središče je v Colorado Springsu (ZDA). Poleg centralnega nadzornega središča so ob dolžini ekvatorja razporejena še štiri nadzorna središča, ki skupaj skrbijo za delovanje satelitov, popravljanje njihove tirnice in sinhronizacijo njihovih ur.

Določanje položaja temelji na merjenju razdalj do vsaj treh satelitov, kot prikazuje slika 2. Vsak satelit oddaja signal, katerega hitrost je skoraj enaka hitrosti svetlobe. Razdalja do satelita se izračuna iz hitrosti signala in časa potovanja od satelita do sprejemnika. Tako dobimo polmer sfere, ki jo določa satelit. Sferi dveh satelitov, bi se lahko sekali v eni točki, kar pa je možno samo, če se uporabnik nahaja na isti premici. Ker je to praktično nemogoče, je presek dveh sfer krožnica. Ko k temu dodamo še podatek s tretjega satelita, dobimo dve možni točki. Če bi se uporabnik nahajal nekje na zemeljski površini, je možno položaj uporabnika določiti tudi samo s tremi sateliti, kjer bi četrto sfero določala Zemlja. [1]



Slika 2: Določanje položaja.

V primeru, da se uporabnik ne nahaja na Zemlji, druge točke ne moremo ignorirati. Problem je tudi, da bi tak način pozicioniranja zahteval enako natančno uro v sprejemnikih, kot je v oddajnikih (satelitih). Za rešitev se uporablja še četrti satelit, zaradi česar ima lahko sprejemnik kvarčno uro, ki jo stalno spreminja. Vsakič, ko dobi podatek iz 4 satelitov, mora popraviti svoj čas, to lahko naredi samo, če se vse 4 sfere sekajo točno v eni točki, tako lahko

s pomočjo štirih enačb izračunamo vrednosti štirih spremenljivk (zemljepisne dolžine, širine, višine in popravka časa).

Natančnost sistema GPS za civilno prebivalstvo je od 5 do 30 m. Pri tem največji del, okrog 5m, predstavljajo ionosferske napake, 2-3 m je posledica nenatančnega podatka o satelitu, dodatna 2 m pa prinese nenatančnost ure.

2.2.1.2. GPS Signali

Edini podatek s satelita je navigacijsko sporočilo (ang. navigation message ali NAV). Navigacijsko sporočilo vsebuje čas, podatke o satelitu, efemeride in almanah. Sporočila se pošiljajo v okvirih, ki so dolgi 1500 bitov in razdeljeni v podokvirje po 300 bitov. Pošiljajo pa se s hitrostjo 50 bit/s. [7]

Prvih 300 bitov opisuje čas satelita in njegovo morebitno odstopanje od GPS-časa. To dvoje je najpomembnejši podatek za določanje položaja. Poleg tega vsebuje ta del tudi podatek o tem, kakšno je stanje satelita (zdravje satelita). Naslednjih 600 bitov predstavlja efemeride. Te so specifične za določen satelit in opisujejo točne popravke njegove tirnice. V zadnjih 600 bitih se prenese še almanah oziroma natančno 1/25 celotnega almanaha. Če bi almanah pridobivali samo z enega satelita, bi v najslabšem primeru trajalo 12,5 minut za pridobitev celotnega almanaha.

GPS sateliti za oddajanje uporabljajo dva signala:

- C/A-koda (ang. Coarse/Acquistion)
  To je psevdonaključno 1023-bitno zaporedje, ki se ponavlja vsako milisekundo in razširja signal s hitrostjo 1,023 Mchip/s.

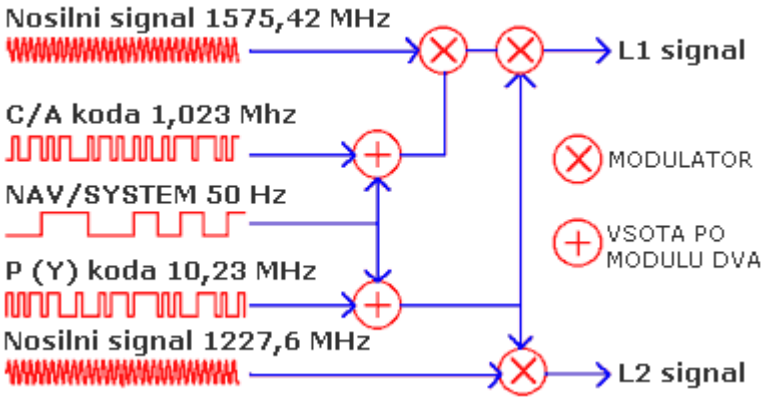
- P-koda (ang. Precision)
  Je zaporedje s frekvenco 10,23 MHz in periodo ponavljanja 1 teden. Signal razširja s hitrostjo 10,23 Mchip/s.

Signal se potem še fazno modulira in oddaja na nosilnih frekvencah GPS-sistema. Ti sta označeni z L1 (1575,42 MHz) in L2 (1227,60 MHz).

P-koda je običajno šifrirana z (encrYpted) kodo. Tako je zagotovljeno, da lahko signal P uporablja samo vojska, ki je zaradi šifriranja lahko prepričana o tem, da signal res prihaja z GPS satelita. C/A-kodo za določen satelit lahko po drugi strani reproducirajo vsi GPS-sprejemniki.

Bistvena razlika v natančnosti med civilno in vojaško rabo je posledica tega, da se navigacijsko sporočilo sporočilo oddaja s kodnega multipleksa na podlagi kode C/A le na frekvenci L1, medtem ko se isto sporočilo, obdelano s P-kodo, oddaja tako na L1 kot na L2, kot je prikazano na sliki 3. Druga prednost takšnega delovanja je, da lahko ameriška vojska v

primeru vojaških konfliktov moti L1 signal in tako ohromi delovanje civilnih GPS sprejemnikov (vključno s sovražnikovimi) na določenem območju. Sama lahko medtem še vedno uporablja podatke, ki jih sateliti pošiljajo preko frekvence L2.



Slika 3: Prikaz kodiranja navigacijskega sporočila.

2.2.2. GPS naprave

GPS naprave bi lahko razdelili na dva dela in sicer na sprejemnik GPS signala in računalnik, ki izvaja druge operacije. Sprejemnik GPS signala nam pomaga določiti naš trenutni položaj, vendar bi bila večina uporabnikov samo nad obliko predstavitve te informacije razočarana, zato imamo zraven vsakega GPS sprejemnika še računalnik s programsko opremo, ki nam pomaga to informacijo obdelati in predstaviti na uporaben način. Slika 4 prikazuje različne GPS naprave.

Slika 4: GPS naprave.

2.2.2.1. Teltonika GH1202

Eno izmed podjetij, ki izdeluje in razvija GPS naprave se imenuje Teltonika. Za njihove naprave je značilno, da se uporabljajo le za sledenje, kar pomeni, da nimajo zaslona na katerem bi prikazovali podatke. Namesto prikazovanja podatkov le-te pošiljajo na prednastavljen strežnik. To pomeni, da imajo naprave tudi notranjo GSM enoto, ki jim omogoča pošiljanje podatkov preko mobilnega omrežja, zato potrebujejo tudi svojo SIM kartico. V nadaljevanju bomo predstavili GPS napravo Teltonika model GH1202 (slika 5). [8]

Slika 5: Teltonika GH1202 GPS naprava.

GH1202 vsebuje GPS sprejemnik za določanje pozicije in GSM enoto za pošiljanje podatkov o trenutni lokaciji na strežnik. Da naprava normalno deluje, mora biti ustrezno nastavljena. Nastavi se jo lahko tako, da se jo preko USB priključka poveže z računalnikom in se za nastavljanje uporabi ustrezno programsko opremo ali pa se na številko SIM kartice pošlje SMS sporočilo, ki vsebuje nastavitve.

Ena izmed pomembnejših nastavitev, brez katere je naprava nekoristna, je URL pot do strežnika, kamor GH1202 pošilja podatke o trenutni lokaciji. Poleg tega lahko nastavimo pogostost pošiljanja podatkov o lokaciji, kar je zelo pomembno, saj pogostejše pošiljanje občutno skrajša čas delovanja naprave. Pri GH1202 baterija zdrži od 10 do 12 ur normalne uporabe, če pa nastavimo zelo pogosto pošiljanje podatkov, čas delovanja prepolovimo.

Ker naprava GH1202 za pošiljanje podatkov uporablja mobilno omrežje, ne dela povsod. Ko mobilnega omrežja ni ne pošilja podatkov na strežnik, ampak jih shranjuje v spomin. V spomin lahko spravi med 6000 in 8000 točk, ki jih pošlje v pravilnem vrstnem redu takoj, ko naprava pride v doseg mobilnega omrežja. Pri tem moramo upoštevati tudi strošek, ki ga

naredimo zaradi prenosa podatkov pri mobilnem operaterju, zlasti če sledimo vozilu ali osebi v tujini.

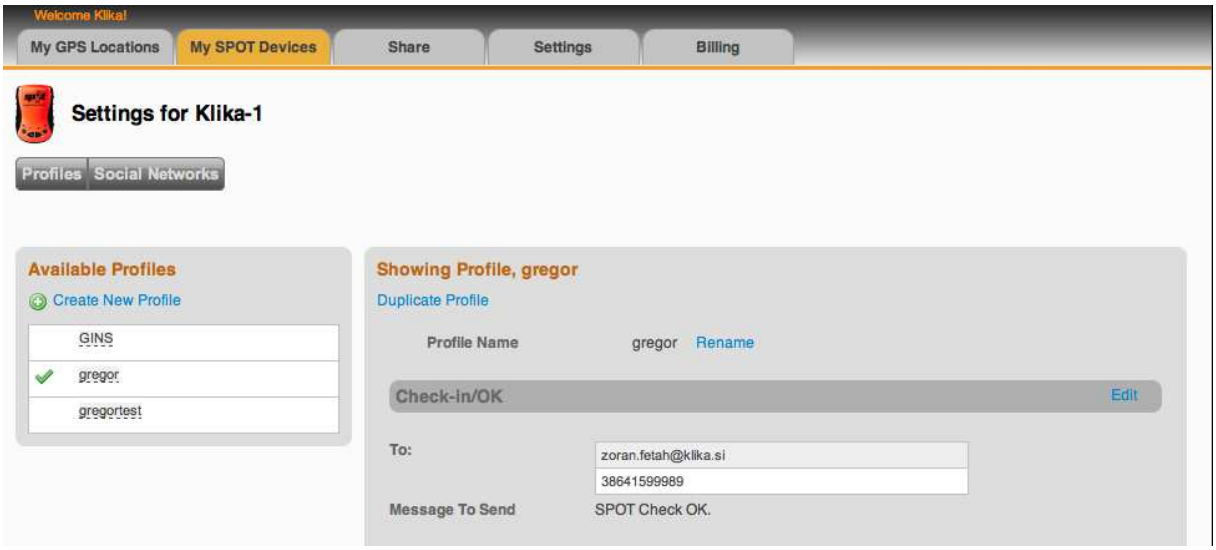
Teltonika GH1202 nam omogoča, da nastavimo posebno telefonsko številko kamor se izvrši klic ali pošlje SMS sporočilo v primeru, da potrebujemo pomoč. To se zgodi, ko pritisnemo na gumb Klic v sili, ki se nahaja na napravi.

2.2.2.2. Spot Personal Tracker

Spot je ameriško podjetje, ki prav tako izdeluje in razvija GPS naprave za sledenje. Najpomembnejša značilnost njihovih naprav je, da za prenos podatkov ne potrebujejo mobilnega omrežja, ampak namesto tega uporabljajo satelitsko povezavo, zato so uporabni predvsem za alpiniste. Za osebno rabo so razvili model Spot Personal Tracker, prikazan na sliki 6. [9]

Slika 6: Spot Personal Tracker GPS naprava.

Ob nakupu katerekoli Spot naprave je potrebno odpreti uporabniški račun na njihovi spletni strani in letno plačevati njihovo storitev sledenja. V nasprotnem primeru je naprava neuporabna, saj podatke o lokaciji pošilja samo na Spot-ov strežnik. Ob prijavi na spletno stran Spot dobimo seznam kupljenih naprav in možnost urejanja njihovih nastavitev, kot je prikazano na sliki 7.



Slika 7: Administracija Spot naprave.

Zato da bi se Spot GPS naprave lahko uporabljale tudi z drugimi lokacijskimi storitvami, ponuja Spot strežnik svojim uporabnikom posnete točke. Do njih pridemo preko HTTP zahtevka na Spotovem strežniku s ključem naprave, za katero nas zanimajo posnete točke. Vsaka naprava ima unikaten ključ, kateri je uporabniku dostopen preko administracijske strani.

Spot Personal Tracker ima naslednje funkcije:

- Vklop/Izklop – ob vsakem vklopu naprave se zažene test, ki preveri delovanje naprave. Potem začne naprava iskati signal GPS satelitov in ko ga najde, uporabniku to sporoči z utripajočo zeleno lučko. Za izklop naprave je potrebno gumb držati 3 sekunde.

- Sledenje – ko je naprava vklopljena beleži trenutne točke in jih preko satelitske povezave pošilja na Spot-ov strežnik.

- V redu – pošlje SMS sporočilo, da je vse v redu, na prednastavljeno telefonsko številko, vključno s trenutnim položajem

- Pomoč – pošlje SMS sporočilo na prednastavljeno telefonsko številko, da je potrebna pomoč in se zanašate, da bo sporočilo, vključno s trenutno lokacijo, posredoval reševalcem. Zaradi resnosti takšne akcije je gumb pomoč rahlo zamaknjen v notranjost naprave, da ne bi nenamerno prišlo do stiska na gumb.

- Klic v sili – obvesti reševalno službo, ki je najbližja trenutni lokaciji, da je potrebna hitra pomoč. Tudi gumb klic v sili je skrit v notranjosti naprave.

2.2.3. TripTracker

Če želimo uporabljati naprave za sledenje moramo biti prijavljeni v vsaj eno izmed lokacijsko osnovanih storitev. Taka storitev je TripTracker, ki je poleg storitve sledenja tudi socialno omrežje, ki omogoča nalaganje slik, posnetih poti in deljenje le-teh s prijatelji.

TripTracker sestavljajo:

- TripTracker spletna aplikacija,
- TripTracker zaledje sistema ter
- TripTracker spletna aplikacija za poslovne uporabnike.

2.2.3.1. TripTracker spletna splikacija

Spletna aplikacija TripTracker, prikazana na sliki 8, je neodvisna od ostalih komponent TripTracker-ja. Namenjena je vsem, ki radi potujejo in bi svoja potovanja in izkušnje radi delili z drugimi, ali pa bi jih radi shranili za kasnejše pregledovanje. Aplikacija v zaledju uporablja svojo zbirko podatkov za uporabnike, storitev pa je brezplačna za uporabo. [10]

Slika 8: Spletna aplikacija TripTracker.

Ko želi uporabnik naložiti novo potovanje, ga mora najprej prenesti iz GPS naprave na svoj računalnik preko USB vmesnika ali na kakršenkoli drug način. Ko ima datoteko s točkami na računalniku, se lahko prijavi v TripTracker spletno aplikacijo, kjer lahko datoteko naloži. Takoj po obdelavi datoteke se pot izriše na zemljevidu.

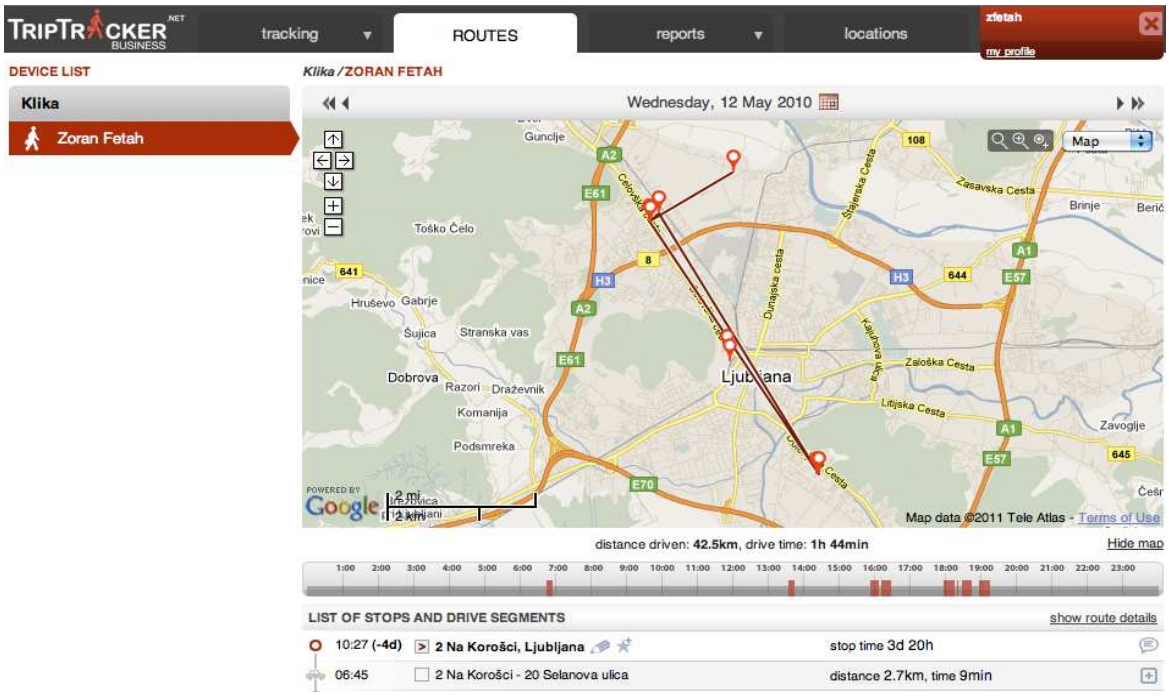
V primeru, da je uporabnik med potovanjem posnel še fotografije, jih lahko naknadno naloži ter označi h kateremu potovanju spadajo. Sistem jih bo dodal na tisti del poti, kjer so bile posnete. To lahko naredi na dva načina. Najprej preveri če imajo slike označeno lokacijo, torej je ob sliki zapisana zemljepisna dolžina in širina. Če te ne najde, preveri kdaj je bila slika posneta in kje na poti je bil uporabnik tisti čas.

2.2.3.2. TripTracker zaledje sistema

Zaledje sistema TripTracker je samostoječ servis, ki ima za vsako podprto GPS napravo v zaledju sistema svoj modul. Vsak modul na svojih vratih na vhodu pričakuje podatke. Podatki, ki pridejo na vhod, so sestavljeni iz identifikacijske številke (ki je kar številka SIM kartice GPS naprave) ter točk (zemljepisna širina in dolžina). Ko pridejo podatki za določeno GPS napravo, jih ustrezni modul prevzame, obdela (po potrebi pretvori format zapisa) in zapiše v zbirko podatkov. Zaledje sistema TripTracker uporablja PostgreSQL zbirko podatkov. Največja prednost te je, da je odprto kodna in ima veliko podpore v internetni skupnosti.

2.2.3.3. TripTracker spletna aplikacija za poslovne uporabnike

Za poslovne uporabnike je na voljo podobna spletna aplikacija kot za navadne uporabnike, le da je izgled spletne aplikacije drugačen, kar prikazuje slika 9. Storitev je potrebno plačevati z mesečno naročnino, v to pa je vključena tudi GPS naprava s pripadajočo SIM kartico. GPS napravo se pred uporabo ustrezno nastavi, da je spletna aplikacija povezana z zalednim sistemom.



Slika 9: TripTracker spletna aplikacija za poslovne uporabnike.

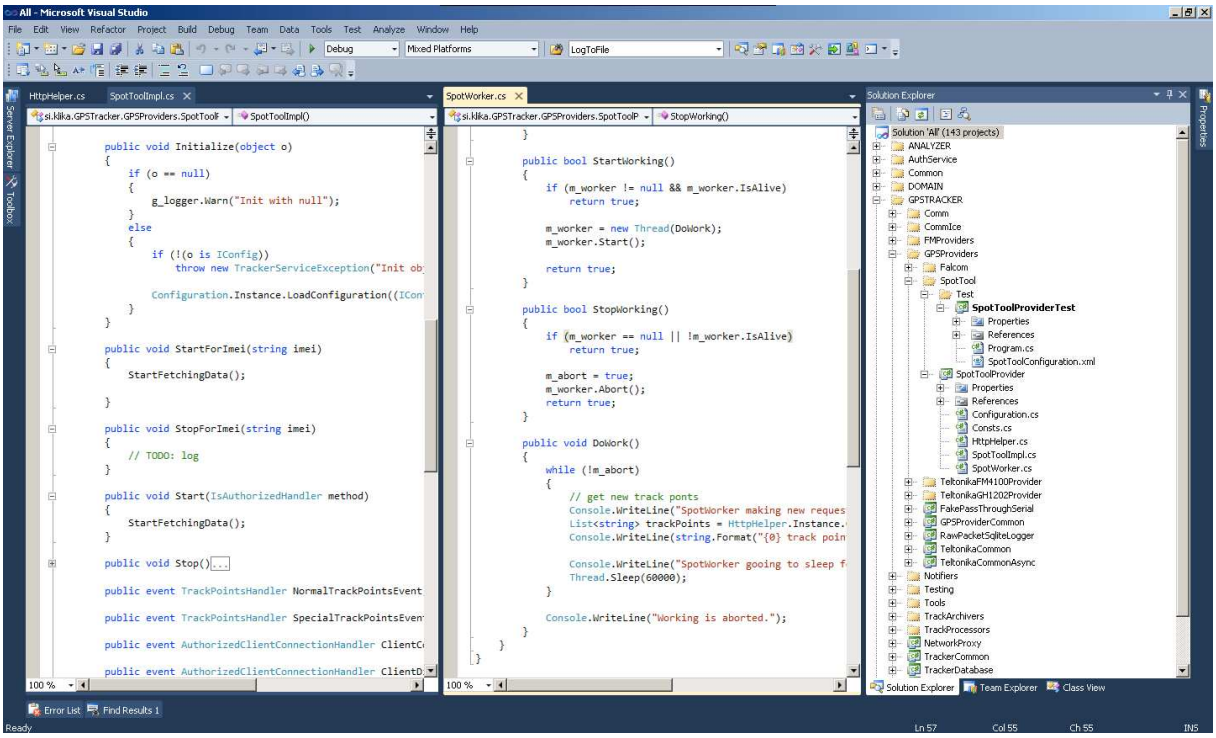
Spletna aplikacija za poslovne uporabnike je narejena tako, da se lahko sledenje spremlja v živo, kar pomeni, da se na mapi izrisujejo nove točke takoj, ko jih zaledni sistem obdela in shrani.

2.3. Uporabljena orodja in tehnologije

Modul smo razvili v programskem orodju Microsoft Visual Studio 2010. Razvoj je potekal v programskem jeziku C#. Strežnik, kamor GPS naprava pošilja podatke o lokaciji, posreduje podatke v obliki XMLja.

2.3.1. Microsoft Visual Studio 2010

Microsoft Visual Studio 2010 je integrirano okolje, ki poenostavi celoten življenjski cikel razvijanja aplikacije od načrtovanja do uvajanja. Visual Studio 2010 ponuja orodja za izdelovanje prototipov, modeliranje in vizualno načrtovanje, s katerimi programer lažje oživi svoje vizije. Nudi integrirano okolje, v katerem lahko razvijalci s svojim znanjem modelirajo, kodirajo, odpravljajo napake, preizkušajo in uvajajo različne vrste programov. Visual Studio 2010 poenostavi pogosta opravila in omogoča razvijalcem raziskovanje v globino platforme. Ponuja zmogljiva orodja za upravljanje projekta, ohranjanje izvorne kode in iskanje napak. Preizkuševalci in razvijalci lahko uporabljajo ročno in samodejno preizkušanje ter napredna orodja za odpravljanje napak in tako zagotovijo, da sestavljajo pravi program na pravi način. Na sliki 10 je prikaza delovna površina omenjenega orodja. [11]

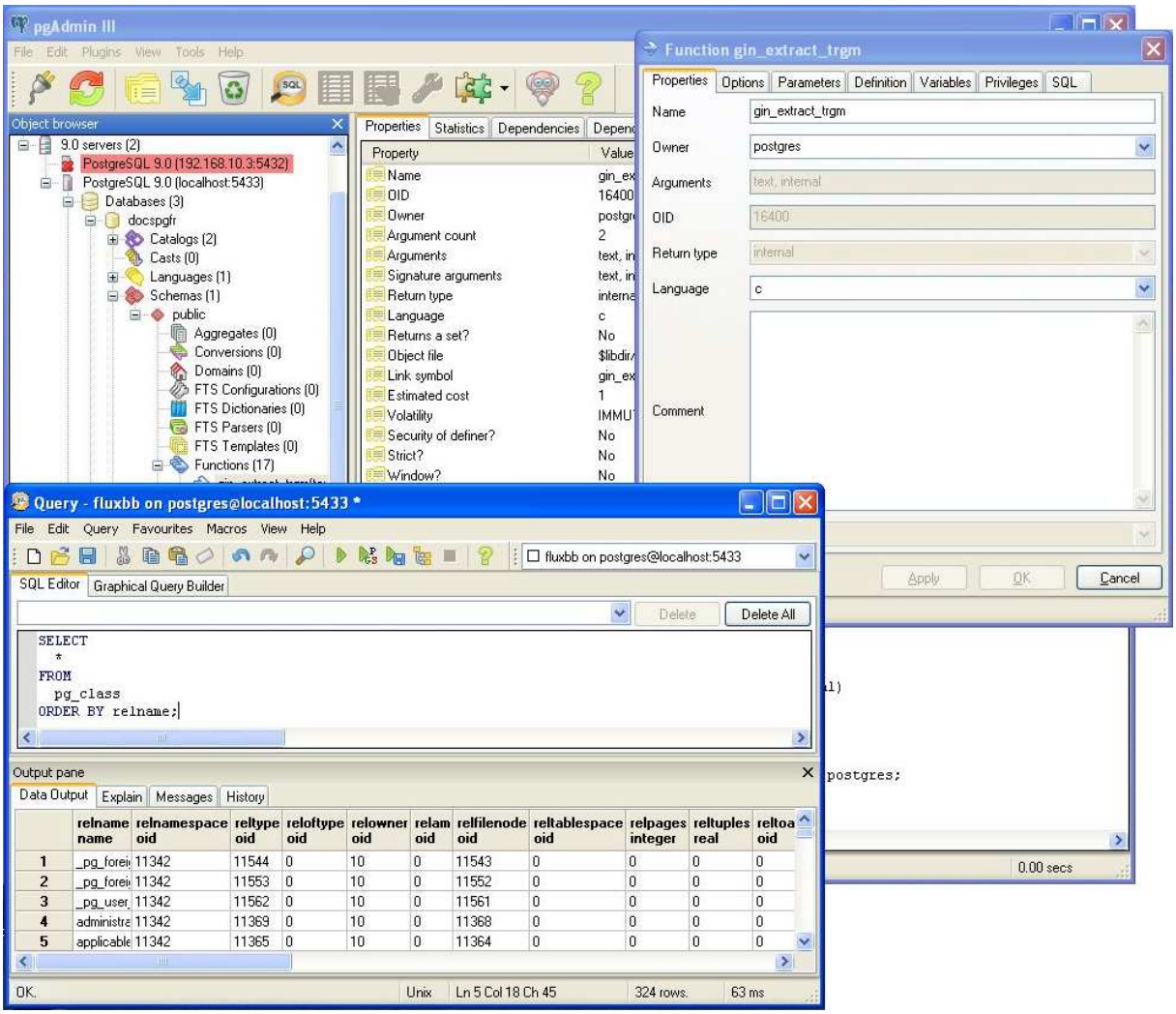


Slika 10: Razvojno orodje Microsoft Visual Studio 2010.

2.3.2. PostgreSQL in PGAdmin

PostgreSQL je zmogljiv odprtokodni sistem za upravljanje z podatkovnimi zbirkami oz. bazami. Razvija se že 15 let in ima sloves zelo zanesljivega delovanja. Podpira vse glavne operacijske sisteme. [12]

PGAdmin aplikacija preko preprostega uporabniškega vmesnika omogoča celovito upravljanje s podatkovno bazo PostgreSQL. Podpira večino operacijskih sistemov. Primerna je za vse vrste uporabnikov (od začetnikov do izkušenih in zahtevnih uporabnikov), saj podpira vse najnovejše funkcije PostgreSQL. Slika 11 prikazuje omenjeni uporabniški vmesnik. [13]



Slika 11: PGAdmin uporabniški vmesnik.

2.3.3. Jezik XML

Standard XML (ang. Extensible Markup Language) je oblikoval konzorcij svetovnega spleta W3C (World Wide Web Consortium). Različica 1.0 je bila odobrena in objavljena 10. februarja 1998. XML je splošen, standarden in od proizvajalcev neodvisen jezik, s katerim lahko avtorji definirajo (opišejo) oznake, ki so uporabljene pri opisu vsebine dokumenta. Možnost definiranja lastnih oznak dovoljuje oblikovanje lastnih tipov dokumentov in s tem zadovoljitev specifičnih potreb. XML dokumenti imajo strukturo, katero lahko poljubno mnogokrat naprej delimo v podstrukture. [14]

Zaradi moči in prilagodljivosti je jezik XML predvsem aplikacijski jezik in ne samo jezik za prikaz vsebin. Pravilnost XML dokumentov lahko preverjamo in jih šele nato ustrezno obdelujemo. Zaradi objektno usmerjene narave je jezik XML inteligenten format zapisa

podatkov, saj omogoča hranjenje dodatnih informacij, ki jih lahko skrijemo, prikažemo ali obdelamo po potrebah uporabnika. Enostavni XML dokument je prikazan na sliki 12.

```
<Books>
    <Book ISBN="0553212419">
        <title>Sherlock Holmes: Complete Novels...
        <author>Sir Arthur Conan Doyle</author>
    </Book>
    <Book ISBN="0743273567">
        <title>The Great Gatsby</title>
        <author>F. Scott Fitzgerald</author>
    </Book>
    <Book ISBN="0684826976">
        <title>Undaunted Courage</title>
        <author>Stephen E. Ambrose</author>
    </Book>
    <Book ISBN="0743203178">
        <title>Nothing Like It In the World</title>
        <author>Stephen E. Ambrose</author>
    </Book>
</Books>
```

Slika 12: Primer XML dokumenta.

3. Programska rešitev

Sistem TripTracker predvideva razširitve podpore za druge naprave, tako da so potrebni vmesniki že narejeni. Naša naloga je v okviru samostojnega projekta narediti nov delujoč modul, pri čemer bomo uporabili obstoječe vmesnike. Ta mora biti preko vmesnikov integriran v sistem TripTracker, njegova naloga pa bo zbiranje in obdelava točk iz naprave Spot Personal Tracker.

3.1. Zajem zahtev

Namen zajema zahtev je zbrati čim več informacij o potrebah bodočih uporabnikov aplikacije. Ker smo se odločili, da bomo razširili že obstoječi sistem z modulom, ki mora opravljati enake naloge kot ostali moduli, smo zahteve že poznali.

3.1.1. Uvod

Dandanes se informacijska tehnologija zelo hitro razvija in nekaj kar je danes novo, je jutri že zastarelo. Sistem smo se odločili razširiti z napravo, katere način delovanja v sistemu še ni podprt. Ker nam je sistem dobro poznan in predvsem, ker so uporabniki sistema izrazili željo po podprtju naprave Spot Personal Tracker, smo si to nalogo zadali v okviru diplomskega dela.

3.1.2. Naloge

Glavna zahteva je implementirati modul, ki bo integriran v sistem preko že narejenih sistemskih vmesnikov in bo deloval brez temeljitejših posegov v jedro sistema.

Pri raziskovanju delovanja naprave Spot Personal Tracker, smo ugotovili, po kakšnem postopku bomo prišli do končne rešitve:

- kreiranje samostojnega servisa, ki implementira sistemski vmesnik,
- konfiguracija modula,
- zajem točk iz strežnika Spot ter
- obdelava točk in zapis v zbirko podatkov.

Za testiranje naše rešitve smo se odločili narediti ločeno namizno aplikacijo (ang. application), ki bo vsebovala teste za vsak funkcionalni sklop modula. Različne teste bomo sprožili s pomočjo argumentov kar v ukazni vrstici.

3.2. Izvedba rešitve

3.2.1. Opis in začetek

Po tem, ko smo je bila naloga jasno definirana, je bilo potrebno namestiti razvojno okolje. Namestili smo Visual Studio 2010 Professional Edition [6] in izbrali programski jezik C#. Iz strežnika smo prenesli izvorno kodo zalednega sistema TripTracker.

Najprej smo ustvarili nov projekt. Ker smo imeli načrt razvoja že narejen, smo dodali vse razrede, ki jih bomo potrebovali, nato pa smo začeli s kodiranjem.

3.2.2. Implementacija vmesnika

Vsak modul mora imeti implementiran vmesnik `IGPS`. Ta vmesnik je del TripTracker sistema in omogoča zalednemu sistemu, da ob zagonu prepozna module, ki jih mora naložiti in zagnati. Ko smo našemu glavnemu razredu pripisali dedovanje vmesnika `IGPS` smo že poznali funkcije, ki jih je potrebno implementirati. Pomembne funkcije pri našem modulu so:

- `void Initialize(object o);`
- `void StartForImei(string imei);`

- `void StopForImei(string imei);`
- `void Start(IsAuthorizedHandler method);`
- `void Stop();`
- `event TrackPointsHandler NormalTrackPointsEvent;`
- `event TrackPointsHandler SpecialTrackPointsEvent;`

Vsak modul deluje kot samostojni servis, ki ga TripTracker zažene (preko funkcije `Start`) in po potrebi ustavi (preko metode `Stop`), pred tem pa ga mora ustrezno nastaviti s funkcijo `Initialize`, preko katere mu posreduje nastavitve. Funkciji `StartForImei` in `StopForImei` sporočita modulu za katere GPS naprave naj sprejema oz. naj ne sprejema točke. Preko dogodkov `NormalTrackPointsEvent` in `SpecialTrackPointsEvent` posredujemo prejete točke TripTrackerju, ki jih zapiše v zbirko podatkov.

Delovanje našega modula se razlikuje od ostalih. Metodi `StartForImei` in `StopForImei` kot parameter prejmeta identifikator GPS naprave, ki je določen v TripTrackerju in ta je trenutno IMEI številka, ki jo imajo vse GSM naprave. Ker Spot Personal Tracker nima IMEI številke smo soočeni s problemom, na kakšen način bi lahko modulu sporočili, za katere naprave naj sprejema točke in za katere ne.

Ker vsak modul potrebuje minimalne nastavitve za normalno delovanje, smo se odločili, da bomo identifikatorje naprav posredovali preko nastavitev modula. Slaba stran takšne implementacije je, da bi morali za dodajanje in odstranjevanje naprave, spreminjati nastavitve modula, vendar je glede na delovanje TripTracker-ja to edina možna rešitev.

3.2.3. Konfiguracija modula

Konfiguracija oz. nastavitve modula so shranjene v XML datoteki, ki jo prikazuje slika 13. V konfiguracijo modula spadajo naslednje nastavitve:

- `ServerURL` – URL do Spot strežnika, kjer so shranjene točke.
- `ServerTimeout` – interval, ki nam pove koliko časa se najdlje počaka na odgovor Spot strežnika (vrednost je zapisana v milisekundah)
- `FetchInterval` – interval, ki nam pove na koliko časa se naredi zahteva na Spot-ov strežnik za nove točke (vrednost je zapisana v milisekundah)
- `KeyList` – je seznam elementov `Key`. Vsak od teh elementov vsebuje identifikator naprave, za katero moramo spremljati točke sledenja. S tem smo začasno rešili problem IMEI identifikatoja na TripTrackerju.

```
<SpotTool>
  <ServerURL>http://share.findmespot.com/messageService/guestlinkservlet</ServerURL>
  <ServerTimeout>60000</ServerTimeout> <!-- in miliseconds -->
  <FetchInterval>10000</FetchInterval> <!-- in miliseconds -->
  <KeyList>
    <key>0br41PNcuO5nFsDpUYNMl4NXNH4vbX0Q7</key>
    <key>951678kjhREleo125saHTL65PASef9456</key>
    <key>asdGR548DTiU965VSDFggfs156saPo9la</key>
  </KeyList>
</SpotTool>
```

Slika 13: XML konfiguracija modula.

Konfiguracija modula se nahaja v skupni konfiguraciji TripTracker sistema, v tej je prostor namenjen posebej modulom za GPS naprave, pod sekcijo GpsProviders. V sekciji GpsProviders imamo tako nastavitve za vse module, naloga modula pa je, da zna iz vseh najti svoje nastavitve. To smo dosegli tako, da smo nastavitve za naš modul dali še pod dodatno sekcijo, znotraj GpsProviders, ki ima naziv SpotTool.

Pri inicializaciji modula dobimo nastavitve v obliki XML dokumenta. Za ta dokument smo že na začetku naredili razred, ki bo znal delati s takšno obliko XML-ja. Razred, poleg logike za delo z XML dokumentom vsebuje tudi lastnosti, preko katerih lahko dostopamo do vseh nastavitev modula.

Ko smo naredili konfiguracijo, smo lahko pripravili prvi testni scenarij, s katerim bi testirali, če se je konfiguracija pravilno naložila. Najprej smo morali ustvariti testni projekt. V projekt smo dodali XML dokument, ki je vseboval konfiguracijo modula, nato pa smo napisali testni scenarij, ki preko vmesnika `IGPS` naloži konfiguracijo in izpiše vrednosti (slika 14).

Slika 14: Izpis testa konfiguracije.

3.2.4. Zajem točk

Naslednja naloga na seznamu je zajem točk. Ker Spot Personal Tracker pošilja točke na Spot-ov strežnik, smo naredili poseben razred, ki bo tekel v svoji niti (ang. thread) in bo glede na določeno časovno obdobje v konfiguraciji, pošiljal HTTP zahtevke za točke na Spot-ov strežnik. Ker je zahtevana pri delovanju modula tudi pravilna zaustavitev, smo to naredili s pomočjo zastavice. To ves čas preverjamo da ugotovimo, kdaj se morajo HTTP zahtevki nehati pošiljati. Funkcije v nadaljevanju prikazujejo na kakšen način smo to naredili.

```
public bool StartWorking() {
    if (m_worker != null && m_worker.IsAlive) {
        return true;
    }

    m_worker = new Thread(DoWork);
    m_worker.Start();

    return true;
}

public bool StopWorking() {
    if (m_worker == null || !m_worker.IsAlive) {
        return true;
    }

    m_abort = true;
    m_worker.Abort();
    return true;
}

public void DoWork() {
    while (!m_abort) {
        // get new track ponts
        List<string> trackPoints =
HttpHelper.Instance.GetTrackPoints();

        Thread.Sleep(Configuration.Instance.FetchInterval);
    }
}
```

Za pošiljanje zahtevkov smo imeli vnaprej pripravljen razred za delo s HTTP zahtevki. Zahtevki morajo vsebovati ključ naprave (ali več ključev, v primeru če sledimo več napravam), naslovljeni morajo biti na URL Spot-ovega strežnika, ki je nastavljen v konfiguraciji, tako kot tudi ključi naprav. Ključe naprav dobimo preko Spotovih administracijskih strani.

Ker Spot-ov strežnik ne dovoljuje drugačnega načina, smo morali pobirati točke za vsako napravo posebej, kar pomeni, da smo preverili vse ključe v konfiguraciji in nato naredili

zahtevo za točke. Zaradi, takega načina delovanja, smo morali tudi odgovore strežnika obdelati za vsako napravo posebej. Slika 15 prikazuje odgovor Spotovega strežnika.


```
- <messageList>
    <totalCount>12</totalCount>
  - <message>
      <esn>0-7436373</esn>
      <esnName>Klika-1</esnName>
      <messageType>TEST</messageType>
      <messageDetail>SPOT Check OK.</messageDetail>
      <timestamp>2011-03-02T12:13:07.000Z</timestamp>
      <timeInGMTSecond>1299067987</timeInGMTSecond>
      <latitude>46.09467</latitude>
      <longitude>14.49927</longitude>
    </message>
  - <message>
      <esn>0-7436373</esn>
      <esnName>Klika-1</esnName>
      <messageType>TEST</messageType>
      <messageDetail>SPOT Check OK.</messageDetail>
      <timestamp>2011-03-02T06:54:28.000Z</timestamp>
      <timeInGMTSecond>1299048868</timeInGMTSecond>
      <latitude>46.08847</latitude>
      <longitude>14.47447</longitude>
    </message>
  - <message>
      <esn>0-7436373</esn>
      <esnName>Klika-1</esnName>
      <messageType>TEST</messageType>
      <messageDetail>SPOT Check OK.</messageDetail>
      <timestamp>2011-03-01T07:42:26.000Z</timestamp>
      <timeInGMTSecond>1298965346</timeInGMTSecond>
      <latitude>46.09457</latitude>
      <longitude>14.49955</longitude>
    </message>
</messageList>
```


Slika 15: Odgovor strežnika Spot na zahtevek o točkah naprave.

Po implementaciji HTTP zahtevkov smo dodali nov testni scenarij. Ker še nimamo shranjevanja smo zaenkrat preverjali, ali se modul obnaša normalno, če ga sredi delovanja poskusimo ustaviti. Tako smo nastavili interval zahtevkov na 10 sekund, po petih sekundah pa smo iz testnega razreda sprožili zaustavitev modula preko `IGPS` vmesnika. Rezultat (Slika 16) je bil zadovoljiv, saj se je nit, ki sproža zahtevke, nemudoma ustavila.

Slika 16: Test prekinitve modula.

3.2.5. Razčlenjevanje XML dokumenta in shranjevanje točk

Naslednji korak je razčlenjevanje XML dokumenta, ki ga dobimo kot odgovor Spot-ovega strežnika. XML dokument vsebuje dve poglavitni sekciji:

- `totalCount` – število vrnjenih točk za zahtevano napravo
- `message` – podatki o točki

Kolikor je vrednost elementa `totalCount` v XML dokumentu, toliko elementov `message` oz. točk vsebuje ta dokument. Element `message` je sestavljen iz:

- `esn` – enolični identifikator GPS naprave,
- `esnName` – ime GPS naprave,
- `messageType` – tip sporočila,

- `messageDetail` – podroben opis sporočila,
- `timeStamp` – čas, ko je bila točka posneta,
- `timeInGMTSecond` – globalni čas, ko je bila točka posneta, izražen v sekundah ,
- `latitude` – zemljepisna širina ter
- `longitude` – zemljepisna dolžina.

Za zapis točk v zbirko podatkov smo morali točke, iz XML-ja, pretvoriti v zapis točk, kot ga uporablja TripTracker. V XML dokumentu preverimo, če se vrednost elementa `totalCount` ujema s številom `message` elementov. V kolikor se števili ne ujemata, odgovor ignoriramo in nadaljujemo z naslednjimi zahtevki, drugače pa pretvorimo vsak `message` element v spremenljivko tipa String in ga dodamo na seznam novo dodanih točk. Pomembne so samo sledeče lastnosti elementa `message`:

- `timeStamp`
- `latitude`
- `longitude`

V nadaljevanju je prikazan del funkcije, ki prikazuje, kako smo pretvorili točke iz XML dokumenta v format, ki je sprejemljiv za TripTracker.

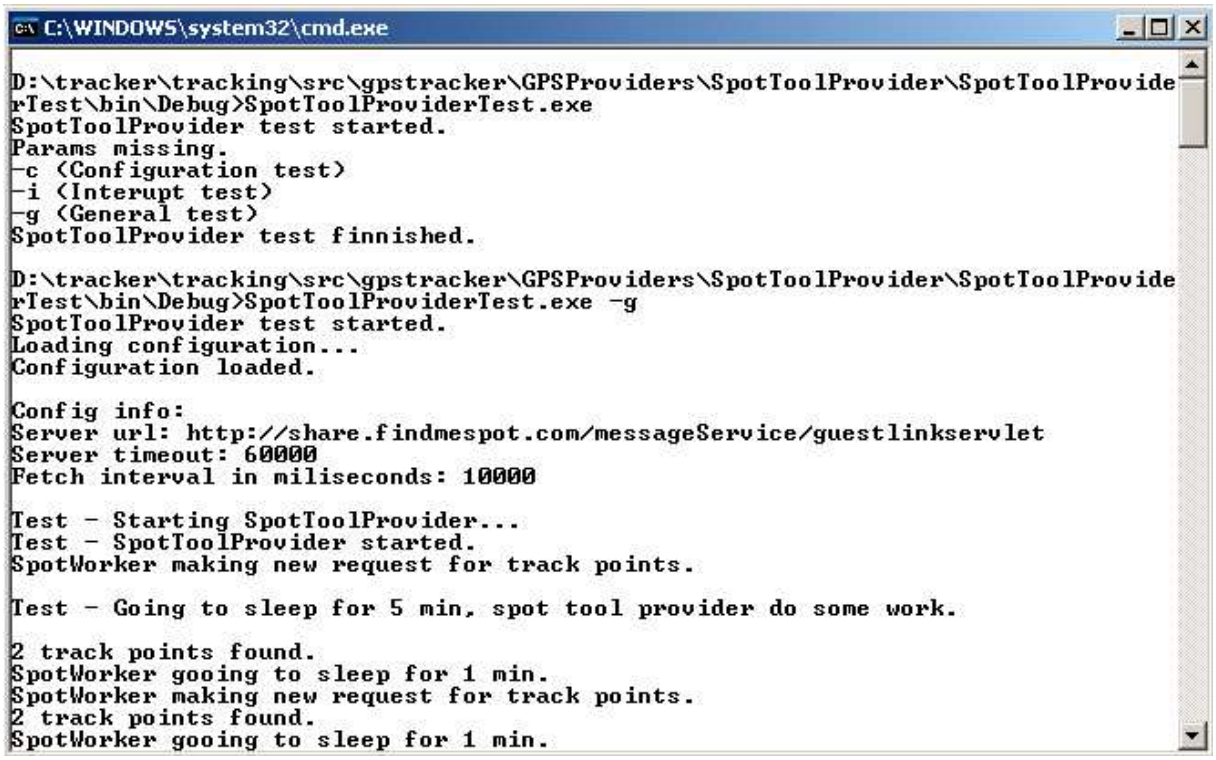
```
XmlNodeList messages = messageList.ChildNodes;
for (int nI = 0; nI < messages.Count; nI++) {
    if (messages[nI].Name == "message") {
        try {
            string trackPoint = string.Empty;
            trackPoint =
string.Format("{0};{1};{2};0;0;0;0;0;1",
messages[nI].SelectSingleNode("timestamp").InnerText,
messages[nI].SelectSingleNode("longitude").InnerText,
messages[nI].SelectSingleNode("latitude").InnerText);

     trackPointsList.Add(trackPoint);

        }
        catch (Exception ex) {
            return;
        }
```

Ko smo dobili seznam novo dobljenih točk smo sprožili dogodek `NormalTrackPointsEvent`, z argumentoma `identifikator naprave` in `seznam točk`. Tega je prestregel TripTracker in točke zapisal v zbirko podatkov.

Prišli smo do faze razvoja, ko imamo delujoči modul. Sedaj lahko napišemo testni scenarij za celotno delovanje modula in tako odkrijemo morebitne napake. Da bi TripTracker lahko shranjeval točke je potrebno vnesti identifikator naprave (IMEI) v zbirko podatkov. Zaradi preverjanja formata identifikatorja je bilo to nemogoče narediti preko obstoječega uporabniškega vmesnika, zato smo napravo vpisali kar ročno s programom PGAdmin, ki je urejevalnik za PostgreSQL zbirko podatkov. Poleg tega smo ustavarili še testnega uporabnika, kateremu smo pripisali to napravo.

Za tem je bilo potrebno posneti testno pot z GPS napravo, tako da smo na Spot strežniku lahko imeli nekaj testnih točk. Ko smo zaključili s tem smo, napisali testni scenarij, ki je najprej naložil konfiguracijo, nato pa začel zbirati točke in jih vpisovati v zbirko podatkov.

Pri testiranju smo najprej dobili sporočilo o napaki, da se točke niso shranile v zbirko podatkov zaradi napačnega formata datuma. Ko smo preverili format, ki ga pričakuje TripTracker in tistega, ki ga dobimo od Spot strežnika, smo na podlagi razlike napisali funkcijo, ki pretvori format datuma, nato pa smo popravili funkcijo za pretvorbo točk in zagnali test še enkrat (slika 17).



Slika 17: Polni test funkcionalnosti modula.

Ponovni test ni sporočal nobenih napak in točke so se uspešno shranjevale v zbirko podatkov. Po temeljitejšem pregledu izpisov testnega razreda pa smo hitro ugotovili, da z vsakim zahtevkom na Spot strežnik dobimo v odgovoru nazaj vse posnete točke. Zato smo se odločili preveriti še zbirko podatkov, vendar nismo našli podvojenih zapisov.

V nadaljevanju smo preverili zakaj od Spot strežnika dobivamo stare zapise točk. Na internetu smo hitro ugotovili, da podjetje Spot na svojem strežniku hrani točke, ki niso starejše od enega tedna. Čeprav napake tukaj pravzaprav ni bilo, smo se odločili še malo spremeniti našo funkcijo za pretvorbo točk. Dodali smo globalno spremenljivko, ki je za vsako napravo sledila, katere točke so že shranjene v zbirki podatkov. Pred dodajanjem nove točke v seznam točk, smo preverili, če ta v globalni spremenljivki že obstaja. Če je tako jo ignoriramo, drugače pa dodamo v seznam novih točk. V nadaljevanju je spremenjen del funkcije, ki pretvarja točke.

```
string trackPoint = string.Empty;
trackPoint = string.Format("{0};{1};{2};0;0;0;0;0;1",
FormatDateTime(messages[nI].SelectSingleNode("timestamp").Inne
rText),
messages[nI].SelectSingleNode("longitude").InnerText,
messages[nI].SelectSingleNode("latitude").InnerText);

if(!pointList.Contains(trackPoint)) {
    trackPointsList.Add(trackPoint);
    pointList.Add(trackPoint);
}
```

Pred ponovnim testiranjem smo nastavili čas testiranja na 30 minut, čas zahtevkov pa na eno minuto in zagnali test. Med testiranjem smo imeli napravo vključeno in modul ni javil nobene napake, točke so se v redu prenašale in shranjevale. Po končanem testiranju smo preverili še zbirko podatkov, ker so bile vse točke pravilno shranjene smo nadaljevali z razvojem.

3.2.6. Dnevnik

Po končani implementaciji modula nam je ostalo samo še zapisovanje pomembnih informacij v dnevnik sistema (ang. logging). Vpisovanje v dnevnik je pri vseh vrstah aplikacij zelo pomembno, saj v primeru napak lahko preverimo informacije, ki so bile zapisane v dnevnik in tako ugotovimo, na katerem delu kode je prišlo do napake in zakaj.

Pisanje dnevnika v TripTracker-ju je zelo enostavno, saj je zato narejen poseben razred imenovan `Logger`. `Logger` je narejen v projektu, ki je skupna točka vsem projektom znotraj TripTracker sistema. V nadaljevanju sledi del kode, kjer se kreira nov objekt `Logger` razreda.

```
protected static readonly Logger g_logger =
LogManager.GetLogger(typeof(SpotToolImpl));
```

`Logger` vsebuje sledeče funkcije:

- `Info` – se uporablja za zapis informativnih podatkov,
- `Warn` – se uporablja na mestih, kjer bi zaradi vrednosti podatkov lahko kasneje drugje prišlo do napake
- `Debug` – se uporablja na kritičnih mestih kode, kjer je verjetnost za napake zelo visoka
- `Error` – se uporablja, ko je že prišlo do napake in bi radi zapisali čimveč informacij o napaki.

TripTracker ima v svoji konfiguraciji vse potrebne nastavitve, kot so pot na disku, za zapisovanje, velikost dnevnika ipd. V našem modulu smo uporabili sledeče zapise:

- `Info` – pri vstopu v funkcijo in izhodu iz nje,
- `Warn` – v primeru inicializacije modula s praznimi nastavitvami,
- `Debug` – pri pretvarjanju točk iz XML dokumenta ter
- `Error` – povsod kjer imamo try/catch blok.

3.2.7. Končno testiranje

Na koncu smo izvedli še postopek celotnega testiranja. Celotni zaledni sistem TripTracker-ja (vključno z zbirko podatkov), smo namestili na testno okolje. Ko smo nastavili konfiguracijo za celotni sistem smo zagnali TripTracker servis in začeli uporabljati Spot Personal Tracker.

Po enem tednu uporabe naprave, ni bilo slabosti modula ali TripTracker storitve. Tudi po pregledu dnevnika nismo opazili nobene napake v delovanju, vendar menimo, da bi za konkretnejše testiranje modula morali uporabljati več naprav. Le tako bi lahko videli, kako se modul obnaša pod večjo obremenitvijo, vendar več tovrstnih naprav nismo mogli dobiti. Slika 18 prikazuje eno izmed naših poti v času glavnega testiranja modula.

Slika 18: Izris posnete poti na mapi.

Zaradi pomanjkanja časa tudi nismo napisali avtomatskih testov, s pomočjo katerih bi lahko simulirali več naprav in več različnih poti po celem svetu, na podlagi česar bi lahko z večjo gotovostjo potrdili, da modul dela tudi pod večjimi obremenitvami.

4. Analiza programske rešitve

4.1. Možne izboljšave

Naša naloga je bila razviti modul s polno funkcionalnostjo in ga integrirati v že obstoječo in delujočo rešitev. Modul smo funkcionalno razvili, vendar integracija v sistem TripTracker ni popolna. TripTracker za identifikator naprav uporablja IMEI številko, ki je pa Spot Personal Tracker naprava nima. Šele po nadgradnji TripTracker-ja, kjer bo izbran skupni identifikator vsem napravam, bo možno naš modul popolnoma integrirati v sistem, s tem da bodo potrebne tudi manjše spremembe v nastavitvah modula.

4.2. Težave pri razvoju

Pri delu smo naleteli na nekaj težav.

4.2.1. Identifikacija naprave

Kot smo že omenili, je največjo težavo povzročila implementacija sistema TripTracker, ki ni predvidevala, da naprave brez GSM enote za identifikator ne uporabljajo IMEI. Problem smo rešili tako, da smo identifikatorje naprav prestavili v konfiguracijo modula.

S tem smo vso kontrolo prepustili modulu. Če se odločimo slediti neko dodatno napravo, moramo v konfiguracijo vnesti njeno identifikacijsko številko in ponovno zagnati TripTracker, da se bo nova številka preko konfiguracije prenesla v modul.

S to težavo je povezano tudi urejanje naprav in uporabnikov, ki je sicer omogočeno preko spletnega vmesnika. Naprave se brez IMEI številke žal ne da vnesti, prav tako se uporabnika, ne da povezati z napravo. Ta problem je rešljiv preko kateregakoli urejevalnika za PostgreSQL zbirko podatkov, kjer se lahko ročno vnese in poveže uporabnika z napravo.

4.2.2. Časovni format

Pri prvem resnejšem testiranju smo naleteli na napako, in sicer pri shranjevanju točk v zbirko podatkov. Pri pregledu sporočila napake smo hitro ugotovili, da je do napake prišlo zaradi neujemanja formata datuma in časa med GPS napravo in TripTracker-jem. Problem smo odpravili s funkcijo, ki je pretvorila čas v pravilni format.

4.2.3. Podvajanje točk

Zaradi nepoznavanja delovanja strežnika Spot smo dolgo časa iskali razlog, zakaj dobivamo pri ponavljajočih zahtevah za točke v odgovoru iste točke. Po dolgem raziskovanju na internetu smo končno prišli do odgovora. Spot strežnik hrani točke za teden dni nazaj. Po tem odkritju smo priredili kodo, da podvojenih točk ne pošilja naprej TripTracker sistemu.

5. Sklepne ugotovitve

V sklopu diplomske naloge smo razširili že obstoječo programsko rešitev z modulom, ki končnim uporabnikom omogoča uporabo večjega nabora GPS naprav. Naš cilj je bil razviti modul in ga polno integrirati v sistem TripTracker. Slednje nam žal ni popolnoma uspelo zaradi omejitev na sistemu TripTracker, saj ta ni predvideval uporabe naprav, ki za delovanje uporabljajo satelitsko povezavo. Spoznali smo, da je potrebno v primeru razširitve sistema delovanje tega zelo dobro poznati. Prav tako je potrebno podrobneje raziskati delovanje novih razširitev in narediti načrt razvoja, saj nam dober načrt lahko prihrani veliko časa pri kasnejšem programiranju.

Med razvojem se je pokazalo nekaj težav, a smo vse uspešno rešili. Podrobneje smo uspeli spoznati standard XML in delovanje spletnih storitev, veliko smo se naučili tudi o samem delovanju GPS sistema in napravah, ki ta sistem uporabljajo. Vpliva na izbiro razvojnega okolja in programskega jezika nismo imeli, saj smo morali uporabiti isto okolje in jezik kot ga uporablja TripTracker. Vendar bi v vsakem primeru izbrali isto, predvsem zaradi razširjenosti, podpore in enostavne uporabe.

Na začetku smo si zadali nalogo razširiti TripTracker sistem, da bo v njem podprta sledilna naprava Spot Personal Tracker. Dejansko smo z našim modulom podprli delovanje vseh naprav podjetja Spot, saj vse njihove naprave delujejo na isti način. Ugotovili smo tudi pomanjkljivost z identifikatorji sledilnih naprav sistema TripTracker, ki jo bo treba v prihodnosti odpraviti, da bo način delovanja vseh modulov poenoten.

Navkljub večletnim izkušnjam s programiranjem, smo z izdelavo diplomske naloge pridobili nova praktična znanja in izkušnje, za katere smo prepričani, da nam bodo koristila tudi v prihodnje.

6. Viri

[1] Elliott D. Kaplan, Christopher J. Hegarty, Understanding GPS : principles and applications, London : Artech House, 2006

[2] (2010) Satellite navigation- Wikipedija, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/Global_navigation_satellite_system

[3] (2010) GLONASS - Wikipedija, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/GLONASS

[4] (2010) Compass navigation system- Wikipedija, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/COMPASS_navigation_system

[5] (2010) Galileo (satellite navigation) - Wikipedija, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/Galileo_(satellite_navigation)

[6] (2010) GPS - Wikipedija, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/GPS

[7] (2010) GPS signals - Wikipedija, prosta enciklopedija. Dostopno na: http://en.wikipedia.org/wiki/GPS_signals

[8] (2010) Teltonika. Dostopno na: http://www.teltonika.lt/en/

[9] (2010) Spot. Dostopno na: http://www.findmespot.eu/en/index.php

[10] (2010) TripTracker. Dostopno na: http://triptracker.net/

[11] (2010) Microsoft Visual Studio 2010. Dostopno na: http://www.microsoft.com/business/smb/sl-SI/strezniki-in-orodja/visual-studio-pro.mspx

[12] (2010) PostgreSQL. Dostopno na: http://www.postgresql.org/

[13] (2010) PGAdmin. Dostopno na: http://www.pgadmin.org/index.php

[14] (2010) XML. Dostopno na: http://www.w3.org/XML/