



УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ
ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА
НОВИ САД
Департман за рачунарство и аутоматику
Одсек за рачунарску технику и рачунарске комуникације

ЗАВРШНИ (BACHELOR) РАД

Кандидат: Славко Радонић

Број индекса: 12600

Тема рада: Бежично праћења стања возила и путне инфраструктуре –
апликативни центар.

Ментор рада: др. Драган Самарџија

Нови Сад, јун, 2012.



КЉУЧНА ДОКУМЕНТАЦИЈСКА ИНФОРМАЦИЈА

Редни број, РБР:			
Идентификациони број, ИБР:			
Тип документације, ТД:	Монографска документација		
Тип записа, ТЗ:	Текстуални штампани материјал		
Врста рада, ВР:	Завршни (Bachelor) рад		
Аутор, АУ:	Славко Радонић		
Ментор, МН:	др. Драган Самарџија		
Наслов рада, НР:	Бежично праћења стања возила и путне инфраструктуре – апликативни центар.		
Језик публикације, ЈП:	Српски / латиница		
Језик извода, ЈИ:	Српски		
Земља публиковања, ЗП:	Република Србија		
Уже географско подручје, УГП:	Војводина		
Година, ГО:	2012.		
Издавач, ИЗ:	Ауторски репринт		
Место и адреса, МА:	Нови Сад; трг Доситеја Обрадовића 6		
Физички опис рада, ФО: (поглавља/страница/цитата/табела/слика/графика/прилога)			
Научна област, НО:	Електротехника и рачунарство		
Научна дисциплина, НД:	Рачунарска техника		
Предметна одредница/Кључне речи, ПО:			
УДК			
Чува се, ЧУ:	У библиотеци Факултета техничких наука, Нови Сад		
Важна напомена, ВН:			
Извод, ИЗ:			
Датум прихватања теме, ДП:			
Датум одбране, ДО:			
Чланови комисије, КО:	Председник:		
	Члан:		
	Члан, ментор:	Потпис ментора	



KEY WORDS DOCUMENTATION

Accession number, ANO:		
Identification number, INO:		
Document type, DT:	Monographic publication	
Type of record, TR:	Textual printed material	
Contents code, CC:	Bachelor Thesis	
Author, AU:	Slavko Radonić	
Mentor, MN:	dr. Dragan Samardžija	
Title, TI:	Wireless monitoring of vehicle and road infrastructure status – application centre.	
Language of text, LT:	Serbian	
Language of abstract, LA:	Serbian	
Country of publication, CP:	Republic of Serbia	
Locality of publication, LP:	Vojvodina	
Publication year, PY:	2012.	
Publisher, PB:	Author's reprint	
Publication place, PP:	Novi Sad, Dositeja Obradovica sq. 6	
Physical description, PD: (chapters/pages/ref./tables/pictures/graphs/appendices)		
Scientific field, SF:	Electrical Engineering	
Scientific discipline, SD:	Computer Engineering, Engineering of Computer Based Systems	
Subject/Key words, S/KW:		
UC		
Holding data, HD:	The Library of Faculty of Technical Sciences, Novi Sad, Serbia	
Note, N:		
Abstract, AB:		
Accepted by the Scientific Board on, ASB:		
Defended on, DE:		
Defended Board, DB:	President:	
	Member:	
	Member, Mentor:	Menthor's sign

Zahvalnost

Zahvaljujem se mentoru, dr. Draganu Samardžiji, na pomoći i savetima u toku izrade završnog rada.

Kolegi, Saši Kartaliji, koji je radio na drugom delu zadatka, na strpljenju i saradnji.

I takođe, svima koji su me ikad sprečili da odustanem.

SADRŽAJ

1.	Uvod	1
2.	Teorijske osnove	2
2.1	Prenosivi uređaji i LBS	2
2.2	Programski jezici i alati.....	2
2.2.1	Programski jezik C	2
2.2.2	Alati korišćeni za grafički prikaz.....	3
3.	Koncept rešenja	4
3.1	Arhitektura poslužioca	4
3.1.1	Prijem i pošiljka poruka	4
3.1.2	Preračunavanje udaljenosti na osnovu koordinata.....	5
3.1.3	Skladištenje podataka.....	6
3.2	Prikaz podataka na mapi	6
4.	Programsko rešenje	7
4.1	Poslužilac.....	7
4.1.1	Glavni modul poslužiloca	7
4.1.2	Modul za analizu poruka.....	8
4.1.3	Modul za rukovanje korisnicima	8
4.1.4	Modul za rukovanje događajima	9
4.1.5	Modul za preračunavanje udaljenosti tačaka na osnovu koordinata	11
4.1.6	Zaglavlje sa neophodnom infrastrukturom	12
4.2	Grafički prikaz podataka	13
5.	Rezultati	15
6.	Zaključak	17
7.	Literatura.....	18

SPISAK SLIKA

Slika 1.1 – Građa sistema, tok informacija	1
Slika 4.1 - Izgled struktura za čuvanje položaja	12
Slika 4.2 - Izgled strukture za čuvanje povratnih vrednosti funkcije za analizu poruke...	12
Slika 4.3. – Izgled i međusobne veze struktura za čuvanje podataka o korisnicima	12
Slika 4.4. – Izgled i međusobne veze struktura za čuvanje podataka o događajima.....	13
Slika 5.1 – Primer mape sa obeleženim mestima nekoliko prijavljenih događaja.....	16

SKRAĆENICE

LBS – eng. *Location Based Services* – Servisi zasnovani na položaju

API – eng. Application Programming Interface – Aplikativna Programska Sprega

UDP – eng. *User Datagram Protocol*

TCP – eng. *Transmission Control Protocol*

GPS – eng. *Global Positioning System*

EDGE – eng. *Enhanced Data rates for GSM Evolution*

GSM – fr. *Groupe Spécial Mobile*, eng. *Global System for Mobile Communications*

3G – eng. *3rd generation mobile telecommunications*

HTML – eng. *HiperText Markup Language*

JS – eng. *JavaScript*

AJAX – eng. *Asynchronous JavaScript And XML*

PHP – eng. *PHP: Hypertext Preprocessor, Personal Home Page: Hypertext Preprocessor*

Preprocessor

1. Uvod

Zadatak ovog rada je razvijanje arhitekture poslužioca sposobnog za prijem i obradu informacija o stanjima vozila i putne infrastrukture i generisanje odgovarajućih povratnih poruka. Pored komunikacije sa mobilnim terminalima, poslužilac treba da poseduje mogućnost da na zahtev korisnika u pretraživaču prikaže prikupljene podatke na mapi.

Poruke do poslužioca stižu putem bežičnog interneta. One mogu sadržati informaciju o nekom od događaja koje sistem prati i pomoću kojih formira bazu podataka ili o trenutnom položaju datog vozila na osnovu kojih se generiše povratna poruka. Baza se periodično pohranjuje u datoteku koja se koristi prilikom stvaranja mape.



Slika 1.1 – Građa sistema, tok informacija.

2. Teorijske osnove

2.1 Prenosivi uređaji i LBS

Zahvaljujući razvoju prenosivih uređaja, dolazi do povećanja u potražnji za servisima koji su usredsređeni na iskorištenje njihovih brojnih mogućnosti. Kao jedan od zastupljenijih primera ovakvih servisa su LBS odnosno servisi zasnovani na položaju korinika. To mogu biti svi servisi od lokalizovanih pretraga, koje kao rezultat daju tražene objekte u relativnoj blizini korisnika, do sistema za navigaciju, koji su sada dostupni kao aplikacije za razliku od ranijih specializovanih uređaja sposobnih samo za tu funkcionalnost. Postojanje i funkcionisanje LBS-a omogičava činjenica da gotovo svi današnji "pametni" telefoni i slični prenosivi uređaji imaju GPS senzor, a popularnost ovih servisa može se pripisati tome da pružaju izuzetno specifične informacije usko povezane za korisnika i time štede velike količine vremena.

Aplikacija realizovana u ovom radu može se svrstati u LBS i to kao proširenje sistema za navigaciju jer pruža dodatne informacije o putu kojim se korisnik kreće kao i mogućnost informisanijeg planiranja puta zahvaljujući grafičkom prikazu.

2.2 Programski jezici i alati

2.2.1 Programski jezik C

Kao osnovni alat za ostvarenje zadatka, odnosno za prokeltovanje samog poslužioca, koristi se programski jezik C. To je jezik niskog nivoa pogodan za primene u kojima se zahteva efikasnost koda i mala zahtevnost u pogledu sistemskih resursa, što je bitna stavka kod sistema koji je namenjen da uslužuje više korisnika, pretražuje velike količine podataka i to u realnom vremenu.

U ovom radu se neće zalaziti dublje u detalje vezane za programski jezik C.

2.2.2 Alati korišćeni za grafički prikaz

HTML je osnovni jezik za uobičavanje internet prezentacija, podržan u svim pretraživačima. Kao ni u prethodnom delu, neće biti detaljnijeg opisivanja ovog jezika.

JavaScript, koji ne treba pomešati sa programskim jezikom Java, je skript jezik koji se koristi, prvenstveno, kako bi se omogućila dodatna funkcionalnost i dinamičnost prezentacija na klijentskoj strani. JS je dinamičan, slabo tipiziran i poseduje skromnu podršku za objektno orientisano programiranje. Kod napisan u ovom jeziku se prilikom učitavanja stranice šalje pretraživaču u sklopu sadržaja stranice, gde se zatim prevodi i izvršava unutar pretraživača.

Jquery je besplatna biblioteka, otvorenog koda, za JS koja omogućava bolje prenošenje u različite pretraživače i pojednostavljuje skriptovanje HTML-a na klijentskoj strani. Najbitnija osobina ove biblioteke u vezi sa ovim projektom je ta da znatno olakšava razvoj AJAX aplikacija.

AJAX je skup povezanih tehnika pisanja koda koje omogućavaju razvoj asinhronih povezivanje i komunikaciju više aplikacija preko interneta. Asinhronost je u ovakvim primenama značajna jer pruža mogućnost razmene podataka između poslužioca i klijenta u toku izvršenja, a ne samo prilikom učitavanja stranice, što čini prezentacije interaktivnim.

Dok je JS uobičajen skript jezik na klijentskoj strani, PHP je ekvivalent na strani poslužioca. PHP je jezik opšte namene čija je prvobitna namena bila da proizvede dinamične internet stranice. Između ostalog, u sklopu sa JS i AJAX-om omogućava da stranica može biti osvežena novim podacima bez njenog ponovnog učitavanja.

Google Maps je besplatan servis koji pruža usluge vezane za digitalne mape i funkcionalnosti rukovanja mapama. Ova tehnologija dopušta jednostavnu implementaciju na različite internet stranice, kombinovanje sa drugim aplikacijama, razvoj dodataka i prilagođavanje specifičnim potrebama. Upotreba ovog servisa vrši se pomoću Google Maps JavaScript API V3. Skripte koja se izvršava na strani klijenta unutar pretraživača i koja je zadužena za slanje zahteva, Google-ovim serverima, za generisanje željene mape, kao i iscrtavanje tako dobijene mape.

3. Koncept rešenja

Na strani poslužioca razlikujemo dva dela koji su zaduženi za obavljanje dve osnovne funkcionalnosti. Za komunikaciju sa mobilnim terminalima i obradu i skladištenje podataka zadužen je poslužilac, dok za grafički prikaz podataka na mapi služi internet pretraživač.

3.1 Arhitektura poslužioca

Strukturu poslužioca je moguće podeliti u 3 glavna dela, to su:

- Prijem i pošiljka poruka
- Preračunavanje udaljenosti na osnovu koordinata
- Skladištenje podataka

3.1.1 Prijem i pošiljka poruka

Mobilni terminal na osnovu unosa korisnika, GPS i senzora ubrzanja (akcelerometra) formira poruke koje preko EDGE ili 3G internet veze, koristeći TCP/IP, dostavlja poslužiocu. Na strani poslužioca pristigle poruke se analiziraju i svrstavaju u odnosu na vrstu I podvrstu u odgovarajuće strukture namenjene za čuvanje informacija o sistemu i korisnicima. Vrste i odgovarajuće podvrste poruka su:

- Poruke o korisniku:
 - Praćenje (prijava, osvežavanje položaja i ođava)
- Poruke o događajima:
 - Policijska patrola (vidljiva i skrivena)
 - Udes (lakši i teži)
 - Gužva u saobraćaju (srednjeg, visokog intenziteta i zastoj)
 - Opasnost (na putu, pored puta i loši vremenski uslovi)
 - Kamera (na semaforu i radar)

Poruke praćenja omogućavaju poslužiocu da izračuna udaljenost korisnika od, do tada prijavljenih, događaja i obavesti korisnika o blizini događaja ukoliko ta udaljenost dostigne vrednost ispod određenog praga.

Poruke o događajima sadrže informacije koje su neophodne poslužiocu kako bi mogao da stvori korisnu sliku stanja sistema, odnosno da izvesti korisnika o tačkama od interesa i stvori prikaz na mapi.

Ukoliko su poruke ispravnog formata informacije dobijene iz njihovog sadržaja se spajaju sa već postojećim informacijama o sistemu, poruke praćenja tako što se korisnik dodaje, uklanja ili se njegov položaj osvežava, a događaji se se dodaju u niz odgovarajuće vrste odnosno stapaju u jedan, ukoliko se nalaze na dovoljno maloj udaljenosti. Ova udaljenost zavisi od vrste prijavljenog događaja, odnosno brzine kojom bi se vozilo u trenutku prijave moglo kretati, brže kretanje dovodi do većeg značaja različitog vremena reakcije korisnika.

Kada ustanovi da je korisnik prišao na određenu razdaljinu od prijavljenog događaja, koji bi mogao da bude od značaja, poslužilac generiše povratnu poruku i šalje je korisniku kao odgovor na njegovu poruku osvežavanja položaja.

3.1.2 Preračunavanje udaljenosti na osnovu koordinata

Udaljenost između tačaka na zemljinoj površini, predstavljenih uređenim parom koordinata geografske širine i dužine, se precizno izračunava složenim formulama koje nadoknađuju razlike nastale usled zakrivljenja zemlje i neravnina površine. Za potrebe ovog sistema dovoljnu preciznost pruža aproksimacija znatno jednostavnijom formulom koja kao pretpostavku uzima da je zemlja sfernog oblika i pri manjim udaljenostima pruža rezultate sa zanemarljivo malom greškom. Udaljenost tačaka na sfernoj površini projektovanoj na ravan računa se na sledeći način:

$$D = R * \sqrt{(\Delta x)^2 + (\cos(x_s) * \Delta y)^2}$$

pri čemu su:

- Δx i Δy – razlike x i y koordinata u radijanima - [rad]
- x_s – srednja vrednost x koordinata tačaka - [rad]
- $\cos(x_s)$ – faktor skraćenja razdaljine koju predstavlja jedan stepen geografske dužine u zavisnosti od geografske širine na kojoj se nalazi.
- R – srednja vrednost poluprečnika zemlje (~ 6371 km)
- D – udaljenost između tačaka – [km]

Jednostavnost ove formule je izuzetno značajna ako se uzme u obzir broj preračunavanja udaljenosti u sistemu ovakvog tipa. Dodatno ubrzanje može se postići ako se udaljenosti čuvaju kao kvadrirane vrednosti čime se zaobilazi potreba za korenovanjem.

3.1.3 Skladištenje podataka

U cilju brže pretrage, prikupljeni podaci se čuvaju u dinamičkim nizovima. Ovo omogućava jednostavan pristup podacima pomoću indeksnog adresiranja. Nedostaci ove strukture u pogledu uklanjanja podataka nadoknađena je time što se u nizu čuvaju pokazivači na strukture u kojima su podaci i to što se podaci ne uklanjaju zasebno već periodično u grupama pa se ponovna dodela memorije i prepisivanje preostalih članova obavlja ređe.

Bitna stavka je i izbegavanje višestrukog beleženja istog događaja, što je rešeno spajanjem sličnih prijava iste vrste događaja, na osnovu udaljenosti.

3.2 Prikaz podataka na mapi

Na zahtev korisnika moguće je prikazati podatke o sistemu, prikupljene do tog trenutka, na lokalnom primerku Google Map-e u sklopu internet pretraživača kako na računaru tako i bilo kom uređaju koji podržava neki od uobičajenih pretraživača. Mapu se može osvežiti docrtavanjem novih informacija zahvaljujući asinhronoj komunikaciji između pretraživača I poslužioca koju pruža AJAX.

4. Programsко rešenje

U ovom poglavlju sledi detaljniji opis realizacije sistema i njegovih delova.

4.1 Poslužilac

- Sačinjen je od pet modula i jednog dodatnog zaglavlja:
- Glavni modul poslužiloca (serverTCP.c)
- modul za analizu poruka (msgParser.c i msgParser.h)
- modul za rukovanje korisnicima (userHandler.c i userHandler.h)
- modul za rukovanje događajima (eventHandler.c i eventHandler.h)
- modul za preračunavanje udaljenosti tačaka na osnovu koordinata (coordToDist.c i coordToDist.h)
- zaglavje sa neophodnom infrastrukturom (dependancies.h)

4.1.1 Glavni modul poslužiloca

Obuhvata pokretanje servera, uspostavu veza i prijem i pošiljku poruka. Sadrži samo funkciju *main*

Funkcije: int main(int argc, char *argv[]);

Ulagne vrednosti:

- *argc – broj parametara koji je su prosleđeni funkciji
- *argv[] - pokazivač na niz sa vrednostima prosleđenim parametrima

4.1.2 Modul za analizu poruka

Pristigle poruke prosleđuju se ovom modulu kako bi bile pravilno podeljene i iz njih izvučene informacije o vrsti, podvrsti i ostalie neophodne za funkcionisanje sistema. Sadrži funkcije *parseMsgB* i *parseMsgS*.

Funkcije:

- int *parseMsgB*(char * *msgBuffer*, MPRS * *ret*);
- int *parseMsgS*(char * *msgBuffer*, MPRS * *ret*);

Opis:

Prva poruku posmatra kao niz bajtova i prepisuje predodređene količine u željene promenljive, dok druga posmatra poruku kao niz karaktera i poruku rastavlja tako što kroz nju prolazi dok ne nađe na dogovoren znak koji označava kraj vrednosti, tako dobijenu vrednost pretvara u brojčanu i ponavlja ovaj postupak do kraja poruke.

Ulazne vrednosti:

- **msgBuffer* – pokazivač na niz u kojem je smeštena poruka koju treba analizirati
- **ret* – pokazivač na strukturu u koju se smeštaju vrednosti dobijene analizom primljene poruke

Izlazne vrednosti:

- funkcije vracaju vrednost koja označava uspešnost izvršenja (1), grešku na strani poslužioca (-2) ili pogrešno uobičenu primljenu poruku (-1).

4.1.3 Modul za rukovanje korisnicima

Ukoliko analizirana poruka sadrži podatke o korisniku, za njih je zadužen ovaj modul. Na osnovu podataka iz strukture sa vrednostima iz primljene poruke, stvara se struktura koja predstavlja novog korisnika. Zatim se proverava da li korisnik već postoji u nizu korisnika i na osnovu toga ili dodaje novog korisnika ili osvežav podatke o postojećem.

Funkcije:

- UserArray * *createUserArray*();
- UserData * *createUserData*(int *userID*, Location * *currLoc*);

Opis:

Funkcije koje rezervišu memoriju i popunjavaju strukture početnim odnosno prosleđenim vrednostima.

Ulazne vrednosti:

- *userID* – broj dodeljen korisniku prilikom uključivanja u sistem
- **currLoc* – lokacija na koju je korisnik poslao kao svoj trenutni položaj

Izlazne vrednosti:

-
- pokazivači na strukture koje sadrže, redom, podatke o svim korisnicima, odnosno o pojedinačnom korisniku.

Funkcije:

- void destroyUserArray(UserArray * UA);
- void destroyUserData(UserData * UD);

Opis:

Funkcije koje oslobođaju memoriju dodeljenu u funkcijama za stvaranje.

Ulazne vrednosti:

- pokazivači na strukturu čiju memoriju treba osloboditi.

Funkcije:

- int checkForUser(UserArray * UA, UserData * UD);
- int addUser(UserArray * UA, UserData * UD) ;
- int updateUser(UserArray * UA, UserData * UD);

Opis:

Funkcija za proveru korisnika i funkcije koje se na osnovu njenog rezultata pozivaju i dodaju odnosno osvežavaju podatke o korisniku.

Ulazne vrednosti:

- *UA – pokazivač na strukturu niza korisnika na kojem se vrši funkcija.
- *UD – pokazivač na strukturu podataka o korisniku koji se trenutno obrađuje

Izlazne vrednosti:

- vrednost koja ukazuje na uspeh izvršenja funkcije.

4.1.4 Modul za rukovanje događajima

Ako analizirana poruka sadrži podatke o događaju, za njih je zadužen ovaj modul. Na osnovu podataka iz strukture sa vrednostima iz primljene poruke, stvara se struktura koja predstavlja novi događaj. Zatim se proverava da li događaj već postoji u odgovarajućem nizu događaja unutar strukture svih podataka i na osnovu toga ili dodaje novi događaj ili spaja sa postojećim postojećem.

Funkcije:

- AllData * createAllData();
- EventArray * createEventArray();
- Event * createEvent(int type, int subtype, int time, Location * loc, char* com);

Opis:

Funkcije koje rezervrvišu memoriju i popunjavaju strukture početnim odnosno prosleđenim vrednostima.

Ulazne vrednosti:

- type, subtype – vrsta I podvrsta događaja
- time – vreme kada je poruka primljena
- *loc – lokacija na koju je korisnik poslao kao položaj prijavljenog događaja
- *com – pokazivač na niz u kojem je smešten komentar dobijen iz poruke

Izlazne vrednosti:

- pokazivači na strukture koje sadrže, redom, podatke o svim događajima u sistemu, podatke o svim događajima jednog tipa, odnosno o pojedinačnom događaju.

Funkcije:

- void destroyAllData(AllData * AD);
- void destroyEventArray(EventArray * EA);
- void destroyEvent(Event * E);

Opis:

Funkcije koje oslobađaju memoriju dodeljenu u funkcijama za stvaranje.

Ulazne vrednosti:

- pokazivači na strukturu čiju memoriju treba osloboditi.

Funkcije:

- int addEventArray(AllData * AD, EventArray * EA);
- int addEvent(AllData * AD, int type, Event * E);

Opis:

Funkcija za dodavanje niza za novu vrstu događaja, odnosno novog događaja u odgovarajuću niz.

Ulazne vrednosti:

- *AD – pokazivač na strukturu svih podataka o događajima.
- *EA – pokazivač na strukturu niza događaja određene vrste, koji treba dodati.
- *E – pokazivač na strukturu podataka o pojedinačnom događaju koji treba dodati u niz odgovarajuće vrste.

Izlazne vrednosti:

- vrednost koja ukazuje na uspeh izvršenja funkcije.

Funkcije:

- int checkEDistance(AllData * AD, int type, Location * loc, int EorU);

Opis:

Funkcija koja proverava odnos udaljenosti prijavljenog korisnika ili događaja i ranije prijavljenih događaja i zadatog praga. U zavisnosti od tog odnosa (manji-veći) sistem odlučuje da li se dodaje novi ili menja postojeći korisnik, tj. Događaj.

Ulazne vrednosti:

- *AD – pokazivač na strukturu podataka sa informacijama o svim događajima.
- type – vrsta događaja, indeks niza događaja unutar strukture svih podataka, koje treba koristiti za poređenje.
- *loc – pokazivač na strukturu u kojoj se položaj korisnika, odnosno događaja, čiji odnos sa ostatkom sistema se traži.
- EorU – vrednost koja određuje da li se prosleđeni položaj odnosi na događaj ili korisnika (time utiče na prag poređenja).

Izlazne vrednosti:

- vrednost koja ukazuje na uspeh izvršenja funkcije, i ukoliko je uspešno izvršena indeks prvog događaja čija je udaljenost ispod praga.

Funkcija:

- int displayAllData(AllData * AD, FILE * dest);

Opis:

Funkcija koja periodično vrši upis važećih podataka iz sistema u datoteku kojoj se pristupa u cilju formiranja mape u pretraživaču.

Ulazne vrednosti:

- * AD – pokazivač na strukturu podataka sa informacijama o svim događajima.
- * dest – deskriptor odredišne datoteke.

Izlazne vrednosti:

- Funkcija vraca 1 u slučaju da su joj prosleđeni zadovoljavajući parametri, a -1 ako neki od parametara nije odgovarajuć

4.1.5 Modul za preračunavanje udaljenosti tačaka na osnovu koordinata

Jedina funkcija ovog modula zadužena je za određivanja udaljenosti korisnika ili događaja od ostalih događaja u sistemu.

Funkcije:

- int coordToDist(Location * A, Location * B);

Opis:

Funkcija primenjuje prethodno opisanu formulu da na osnovu koordinata tačaka A i B u decimalnom obliku izračuna udaljenost u metrima.

Ulazne vrednosti:

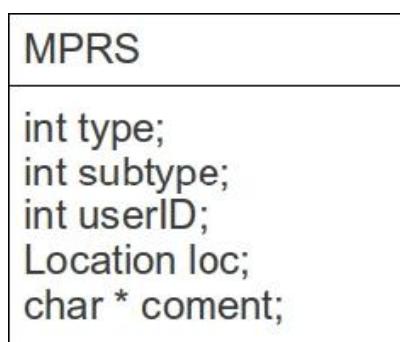
- *A i *B – pokazivači na strukturu koje sadrže položaje tih tačaka

Izlazne vrednosti:

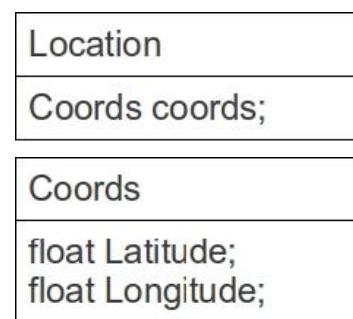
- udaljenost tačaka A i B u metrima.

4.1.6 Zaglavlj sa neophodnom infrastrukturom

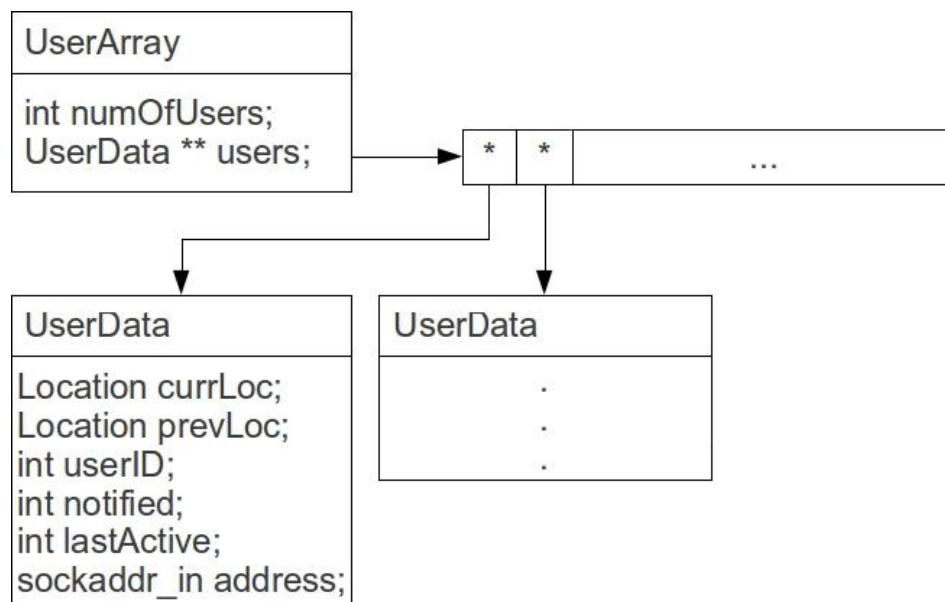
U sve strukture i konstantne vrednosti, kao i biblioteke potrebne u svim delovima projekta, definisani su u ovom zaglavljtu.



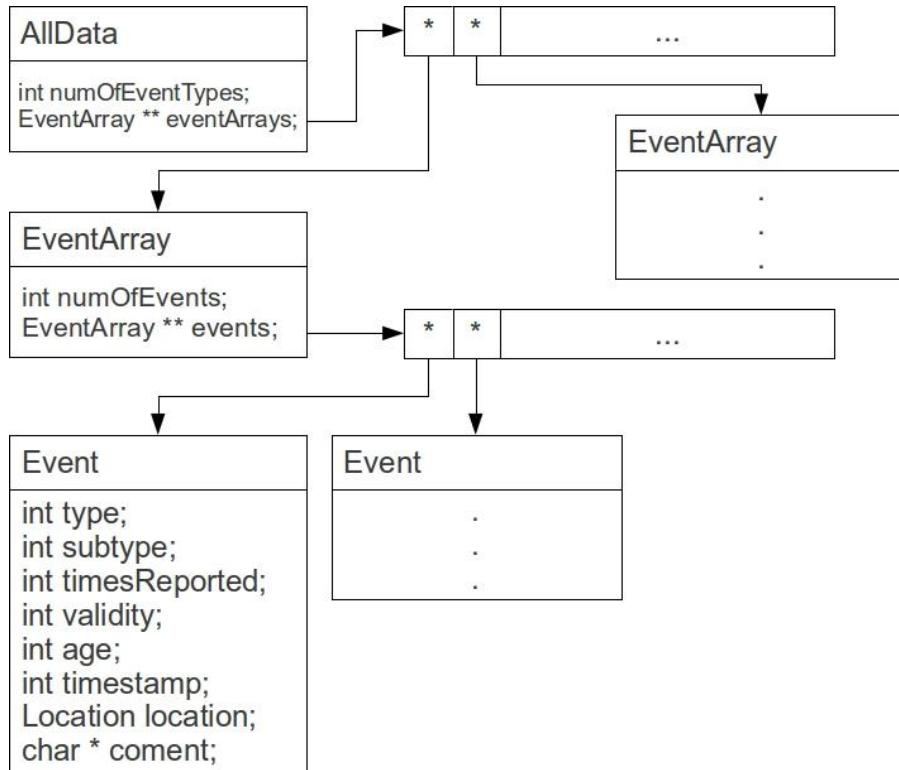
Slika 4.2 - Izgled strukture za čuvanje povratnih vrednosti funkcije za analizu poruke



Slika 4.1 - Izgled struktura za čuvanje položaja



Slika 4.3. – Izgled i međusobne veze struktura za čuvanje podataka o korisnicima



Slika 4.4. – Izgled i međusobne veze struktura za čuvanje podataka o događajima

4.2 Grafički prikaz podataka

Ovaj deo sistema je rešen u vidu javno dostupne internet prezentacije kojoj se pristupa pomoću pretraživača, dok je za sam proces insertovanje mape zadužen Google Maps odnosno API pomoću kojeg mu se pristupa. Demonstrativne je prirode i stoga realizovan u najjednostavnijem obliku, samo da bi bio u stanju da prikaže funkcionalnost sistema, bez obraćanja pažnje na estetiku stranice.

Delovi prezentacije:

- osnovna stranica (map.html), formira u pretraživaču praznu stranicu sa dva dugmeta. Pritiskom na prvo od njih poziva se Google Maps API funkcija koja šalje zahtev za generisanje mape, i vraća ručicu mape, a drugo poziva funkciju koja pokreće dobavljanje podataka za iscrtavanje i nakon što podaci stignu analiziraju se i na osnovu rezultata analize obeležavaju na mapi.
- Asinhroni pristup podacima sa poslužiocu (ajax.js), ovaj modul sadrži funkcije koje se pokreću pritiskom na dugme glavne stranice i šalju zahtev poslužiocu da pokrene izvršenje koda koji čita i prosleđuje sadržaj datoteke.

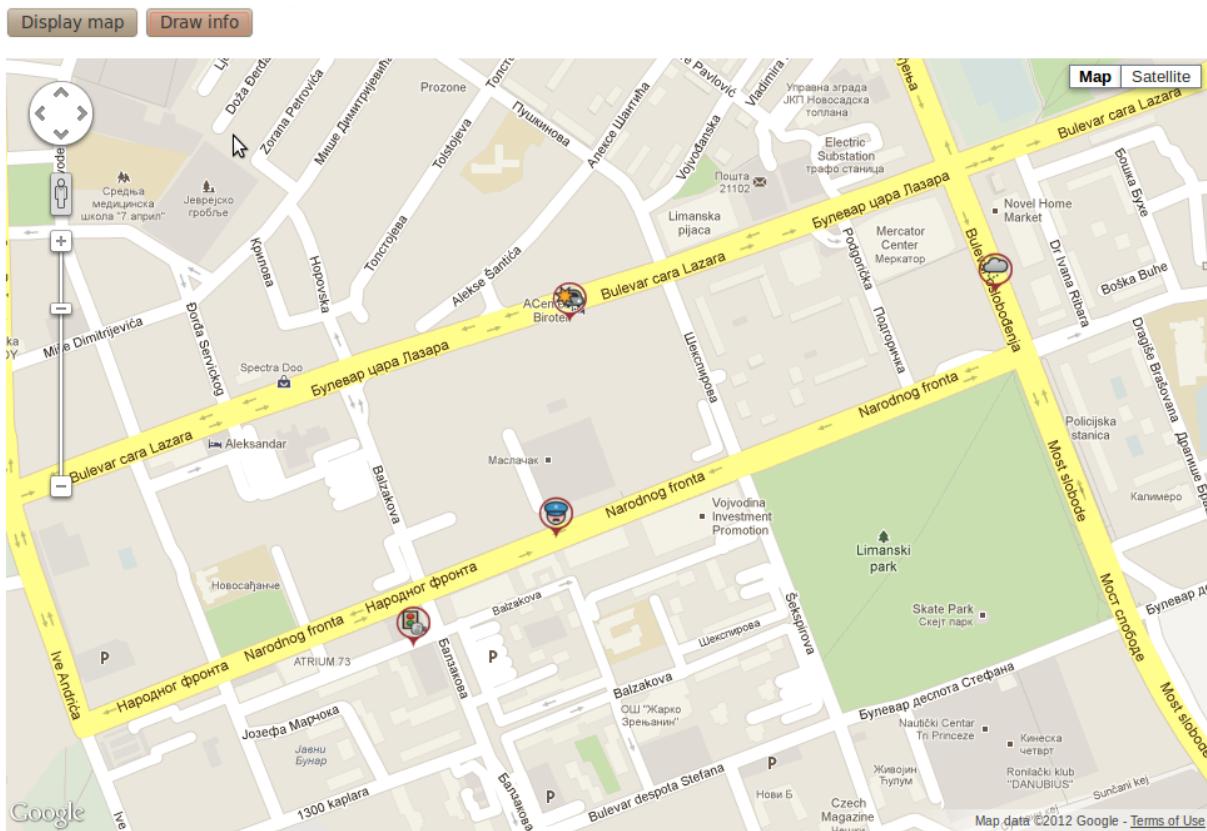
- Čitanje datoteke na poslužiocu i slanje sadržaja klijentu (basic.php), kod u ovom modulu se izvršava na zahtev klijenta i šalje mu podatke iz datoteke o događajima.
- Zahtev za stvaranje mape, njeno iscrtavanje i obeležavanje na osnovu podataka o sistemu (map.js i parseAndMark.js), funkcije koje započinju i okončavaju aktivnosti stranice do ponovnog zahteva korisnika.

5. Rezultati

Ispitivanje rada sistema vršeno je nizom testiranja na terenu. Upotrebljeni su različiti prenosivi uređaji i pokazalo se da se ne ponašaju na isti nacin pod istim uslovima. Najbitnija razlika je preciznost i kvalitet GPS primopredajnika, koji se menjaju od uređaja do uređaja. Ustanovljeno je da prilikom pozicioniranja na određenom položaju, u stanju mirovanja, GPS određenu količinu aditivnog šuma koji se linearno povećava sa vremenom provedenim sa nepomičnim uređajom. Ovaj šum se prenosi na ostale podatke sve dok se sistem ne ugasi i ponovo pokrene. Tokom kretanja šum je promenljivog intenziteta, ali i pravca pa se vremenom poništava i nema znatan uticaj na preciznost. Pored šuma u nekim slučajevima primećeno je i kašnjenje, tačnije nepreciznost koja nije nasumično usmerena, već je položaj na putanji kojom se vozilo kretalo do tog trenutka ali se na njemu ne nalazi već nekoliko sekundi.

Kod povratne povratne informacije, odnosno slanje poruke alarma, se izvršava predvidivo, ali buduće da su položaji dojava nasumični, a sistem ne proverava o kojem je doogađaju obavestio korisnja, dolazi do izostanka obaveštenja ukoliko su događaji dojavljeni na dovoljno maloj razdaljini (trenutno na manje od 200m).

Sistem redovno oslobađa nepotrebnu memoriju tako da ne dolazi do "curenja" (proveravano u toku 2 sata rada sa relativno malim opterećenjem i dotokom informacija).



Slika 5.1 – Primer mape sa obeleženim mestima nekoliko prijavljenih događaja

6. Zaključak

Postoji mnogo mesta za proširenje ovakvog sistema, kako u smislu dodatne funkcionalnosti i pokrivanja šireg skupa događaja tako i poboljšanja brzine i robustnosti trenutnog algoritma poslužioca.

Mogući dodaci u smislu funkcionalnosti:

- Uključivanje šireg skupa nadgledanih događaja, kao što su: praćenje ostalih učesnika u saobraćaju (ne samo automobila), korisnih objekata
- Rukovanje dodatnim informacijama u vidu komentara ili slika.
- Povezivanje korisnika i obaveštavanje o njihovoj udaljenosti.

Pobljšanja algoritma:

- Lokalizovana pretraga događaja. U zavisnosti od položaja vozila i pravca i smera kretanja bilo bi moguće, uz malu količinu dodatne obrade prilikom unosa podataka o događajima, odbaciti veći deo poznatih podataka jer sasvim sigurno nisu na mestu koje je od interesa za korisnika u datom trenutku. Ovakva pretraga bi takođe naglasila značaj događaja koji se nalaze ispred korisnika i olakšalo razaznavanje korisnih od beskorisnih informacija.

- Stalna registracija korisnika u cilju praćenja istorije korišćenja sistema, njihove pouzdanosti ili automatskog prilagođavanja sistema potrebama i navikama korisnika

- Zastarevanje podataka i njihovo automatsko izbacivanje nakon isteka određenog vremena bez nove dojave.

7. Literatura

- [1] Linux Man pages
- [2] www.stackowerflow.com
- [3] developers.google.com/maps
- [4] www.wikipedia.com