



**Univerzitet Singidunum**

**Departman za poslediplomske studije i međunarodnu saradnju**

**Savremene informacione tehnologije**

**Master studije**

**Razvoj geoinformacionog servisa za Android  
mobilne platforme**

**- master rad -**

**Mentor:**

prof. dr Mladen Veinović

**Student:**

Vladimir Lazović

410484/2011

Beograd, 2014. godina

# Razvoj geoinformacionog servisa za Android mobilne platforme

**Sažetak:** U radu je prikazan proces razvoja servisa za Android mobilne platforme korišćenjem lokacijski baziranih servisa.

Objašnjen je princip prikupljanja lokacijskih informacija korišćenjem GPS prijemnika, tornja mobilne telefonije i Wi-Fi konekcije.

Formula koja se u ispitivanju pokazala kao dovoljno tačna iskorišćena je za izračunavanje udaljenosti između mobilnog uređaja i objekata u bazi.

Korišćenjem savremenih tehnologija za razvoj aplikacija razvijen je servis koji na zahtev mobilne aplikacije izračunava udaljenost objekata u datom radijusu i dobijene rezultate prosleđuje mobilnoj aplikaciji koja ih prikazuje.

**Ključne reči:** Android, mobilni telefon, servis, geografski informacioni sistem, lokacijski bazirani sistemi, API, HTML5, PHP

## Development of a geo-information service for Android

**Abstract:** This paper elaborates on the process of service development for the Android mobile platform using location-based services (LBS). It also provides thorough explanation of the principle of location-based information collection using GPS receiver, mobile phone towers and Wi-Fi connection.

The formula that was successfully used in the study has also been implemented for calculating the distance between the mobile device and the object.

The use of contemporary technologies for application development has facilitated initiation of the service that calculates the distance of the object within the given radius when requested by the application and forwards those results to the mobile application that displays them.

**Keywords:** Android, mobile phone, service, geographic information system, location based services, API, HTML5, PHP

# Sadržaj

|          |                                                              |           |
|----------|--------------------------------------------------------------|-----------|
| <b>1</b> | <b>Uvod.....</b>                                             | <b>1</b>  |
| 1.1      | <i>Predmet istraživanja .....</i>                            | <i>1</i>  |
| 1.2      | <i>Ciljevi istraživanja .....</i>                            | <i>1</i>  |
| 1.3      | <i>Naučne metode.....</i>                                    | <i>1</i>  |
| <b>2</b> | <b>Geografsko pozicioniranje i određivanje lokacije.....</b> | <b>2</b>  |
| 2.1      | <i>GPS.....</i>                                              | <i>2</i>  |
| 2.2      | <i>IP adresa .....</i>                                       | <i>3</i>  |
| 2.3      | <i>GSM/CDMA .....</i>                                        | <i>3</i>  |
| 2.4      | <i>WiFi i Bluetooth.....</i>                                 | <i>4</i>  |
| <b>3</b> | <b>Usluge bazirane na lokaciji (LBS).....</b>                | <b>5</b>  |
| 3.1      | <i>Struktura LBS .....</i>                                   | <i>5</i>  |
| 3.2      | <i>Metode prikupljanja podataka o lokaciji .....</i>         | <i>6</i>  |
| 3.3      | <i>Primena LBS-a u praksi .....</i>                          | <i>7</i>  |
| 3.3.1    | <i>Marketing.....</i>                                        | <i>8</i>  |
| 3.3.2    | <i>Lokacijski bazirane društvene mreže .....</i>             | <i>8</i>  |
| 3.3.3    | <i>Lokacijski bazirane igrice za mobilne telefone .....</i>  | <i>8</i>  |
| 3.3.4    | <i>Praćenje .....</i>                                        | <i>8</i>  |
| 3.3.5    | <i>Turizam.....</i>                                          | <i>9</i>  |
| 3.4      | <i>Privatnost u LBS-u .....</i>                              | <i>9</i>  |
| <b>4</b> | <b>Pregled tehnologija .....</b>                             | <b>11</b> |
| 4.1      | <i>HTML 4.....</i>                                           | <i>11</i> |
| 4.2      | <i>HTML 5.....</i>                                           | <i>11</i> |
| 4.2.1    | <i>Strukturni markup.....</i>                                | <i>12</i> |
| 4.2.2    | <i>Aplikacioni programski interfejsi u HTML5.....</i>        | <i>13</i> |
| 4.3      | <i>XML .....</i>                                             | <i>14</i> |
| 4.3.1    | <i>Deklaracija.....</i>                                      | <i>14</i> |
| 4.3.2    | <i>Sintaksa.....</i>                                         | <i>14</i> |
| 4.3.3    | <i>Atributi.....</i>                                         | <i>15</i> |
| 4.3.4    | <i>Pravilnost XML dokumenta.....</i>                         | <i>15</i> |
| 4.4      | <i>MySQL .....</i>                                           | <i>16</i> |
| 4.5      | <i>PHP.....</i>                                              | <i>17</i> |
| 4.6      | <i>jQuery.....</i>                                           | <i>17</i> |
| 4.7      | <i>Android operativni sistem.....</i>                        | <i>18</i> |

|          |                                                                         |           |
|----------|-------------------------------------------------------------------------|-----------|
| 4.8      | <i>Geografski informacijski sistemi</i> .....                           | 19        |
| 4.9      | <i>WebGIS</i> .....                                                     | 20        |
| 4.9.1    | Arhitektura WebGIS-a.....                                               | 20        |
| 4.9.2    | Upotreba WebGIS-a u praksi .....                                        | 22        |
| 4.9.3    | Prednosti i razlike razvijenih lokacijskih servisa.....                 | 26        |
| <b>5</b> | <b>Metode za određivanje distanci između dve tačke u prostoru</b> ..... | <b>28</b> |
| 5.1      | <i>Vinsentijeva formula</i> .....                                       | 28        |
| 5.2      | <i>Formula velikog kruga</i> .....                                      | 30        |
| 5.3      | <i>Haversin formula</i> .....                                           | 31        |
| <b>6</b> | <b>Predloženo rešenje i eksperimentalni rezultati</b> .....             | <b>32</b> |
| 6.1      | <i>Poređenje tačnosti distanci</i> .....                                | 32        |
| 6.2      | <i>Poređenje brzine izvršavanja SQL upita u MySQL DBMS-u</i> .....      | 35        |
| <b>7</b> | <b>GIS portal i Android aplikacija</b> .....                            | <b>38</b> |
| 7.1      | <i>Google Maps API</i> .....                                            | 38        |
| 7.1.1    | Implementacija <i>Google Maps API</i> -ja u HTML5.....                  | 38        |
| 7.2      | <i>GIS portal</i> .....                                                 | 41        |
| 7.2.1    | Baza GIS Portala .....                                                  | 42        |
| 7.2.2    | Aplikacija GIS Portala .....                                            | 43        |
| 7.2.3    | Organizacija stabla i direktorijuma portala.....                        | 43        |
| 7.3      | <i>Aplikacija za pametne telefone</i> .....                             | 44        |
| 7.3.1    | Struktura klase.....                                                    | 44        |
| 7.3.2    | Definicija pomoćnih klase.....                                          | 45        |
| 7.3.3    | Definicija klase prikaza (Activity klase) .....                         | 46        |
|          | <b>Zaključak</b> .....                                                  | <b>50</b> |
|          | <b>Literatura</b> .....                                                 | <b>51</b> |
|          | <i>Spisak slika</i> .....                                               | 54        |
|          | <i>Spisak tabela</i> .....                                              | 54        |

# 1 Uvod

Razvoj veb aplikacija predstavlja jedan od trenutno najaktuelnijih pravaca istraživanja i razvoja u oblasti softverskog inženjerstva. Prednosti korišćenja Interneta kao osnovnog okruženja aplikacije su brojne. Veb aplikacije na klijentskoj strani koriste veb čitače koji su javno dostupni i besplatni. Veb čitači se odlikuju univerzalnošću, odnosno mogu predstavljati interfejs ka bilo kom obliku i formatu podataka. Takođe, veb aplikacije su nezavisne od platforme na kojoj se izvršavaju. Promena korisničkog interfejsa veb aplikacija svodi se na izmene izvornog koda, smeštenog na veb serveru, pri čemu učinjena izmene odmah postaju vidljive svim korisnicima. Aplikacije koje kao svoje okruženje za rad imaju Internet, menjale su svoj način korišćenja i izgradnje sa razvojem Internet tehnologija. Od statičkog sadržaja i izgleda aplikacije, preko korišćenja jezika za dinamičko kreiranje i izmenu izgleda stranice, veb aplikacije su dostigle nivo kolaborativnih sistema u kojima je uloga korisnika bliska ulozi projekatana ovakvih aplikacija. Moderne veb aplikacije predstavljaju pristupne tačke ka najrazličitijem sadržaju koji se prikazuje na zahtev korisnika. Sve veći uticaj na izgled i ulogu aplikacije imaju podaci koje je neophodno prikazati.

Pored prikupljanja podataka iz udaljenih resursa, veb aplikacije dozvoljavaju određeni stepen kolaborativnosti u radu. Korišćenjem interfejsa aplikacije korisnici mogu dodavati sopstveni sadržaj i podatke, a takođe mogu menjati i izgled aplikacije. Trenutno najaktuelniji arhitekturni model veb aplikacija je model u kome se pretpostavlja povezivanje aplikacije sa različitim distribuiranim izvorima podataka i servisima koje je moguće kombinovati.

## 1.1 Predmet istraživanja

Predmet istraživanja ovog rada predstavlja razvoj servisa koji će obraditi i proslediti podatke mobilnoj aplikaciji na osnovu lokacije mobilnog uređaja za što kraće vreme i sa što većom tačnošću. Analizirane su formule za izračunavanje udaljenosti između dve tačke, na osnovu rezultata odabrana je formula koja sa je zadovoljila kriterijume.

## 1.2 Ciljevi istraživanja

**Naučni cilj:** naučni cilj ovog rada je da se razvije servis koji će obraditi podatke kako bi se korisniku obezbedio rezultat u što kraćem vremenskom periodu i sa što većom geografskom preciznošću.

**Društveni ciljevi:** društveni cilj je upoznavanje javnosti sa razvojem servisa i mobilnih aplikacija pomoću savremenih tehnologija.

## 1.3 Naučne metode

Pri izradi ovog rada korišćene su osnovne naučne metode deskripcije, indukcije, dedukcije, analize, sinteze i generalizacije.

Metoda deskripcije je korišćenja pri objašnjavanju osnovnih pojmova i sistema funkcionisanja.

U teorijskim segmentima su primenjene metode indukcije, dedukcije i analize, a u izvođenju zaključaka metode sinteze i generalizacije.

Takođe u okviru ovog rada je napravljen eksperiment u kome su testirane metode za izračunavanje udaljenosti između dve tačke i pronalaženje načina za najbrže izvršenje SQL upita.

## 2 Geografsko pozicioniranje i određivanje lokacije

Ljudi su zbog stalnog kretanja od jednog do drugog mesta koristili razne metode sa različitim stepenom tačnosti, da približno izračunaju gde se nalaze u datom trenutku. Kako se tehnologija poboljšavala, tako se i njihova sposobnost da precizno utvrde tačnu poziciju povećavala. Termin geografske lokacije se može najbolje opisati kao određivanje geografskog položaja osoba, mesta ili stvari. U modernoj eri ovo podrazumeva upotrebu uređaja koji imaju pristup Internetu (računari, tableti, itd.), pametni telefoni ili GPS uređaji.

Pozicija ne mora da bude dobijena pomoću GPS uređaja. Postoji više načina da se dođe do podataka o lokaciji putem modernih uređaja. Načini na koji se može doći do ovih podataka su [1]:

- Globalno pozicioniranje (GPS)
- IP adresa
- GSM/CDMA
- *WiFi* i *Bluetooth* MAC adresa
- Ručni unos

GPS može da se koristi na bilo kojem savremenom mobilnom telefonu koji ima ugrađen GPS kao i na uređajima za GPS navigaciju.

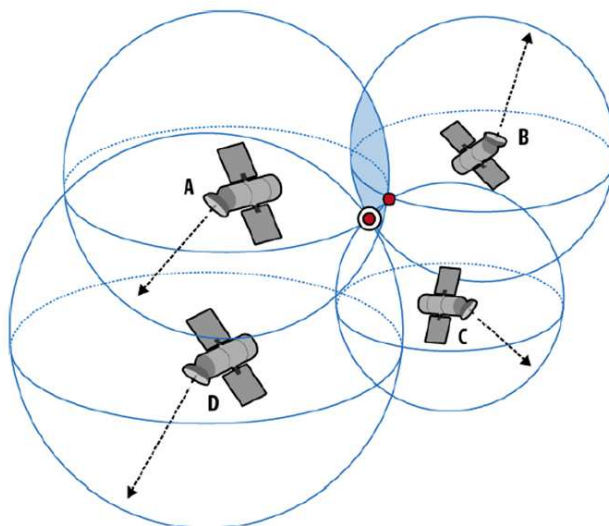
Lokacija koja se dobije putem IP adrese je dostupna sa bilo kog uređaja koji je povezan na globalnu računarsku mrežu. GSM/CDMA Mobilni ID koriste operateri mobilnih telefona širom sveta. Ručni unos je dostupan na bilo kom uređaju i softveru koji traži unos lokacije od korisnika, podatka kao što je poštanski kod itd.

### 2.1 GPS

GPS sateliti kontinuirano emituju signale koje GPS uređaj ili prijemnik prima i raščlanjuje. Prijemnik izračunava svoju poziciju tempirajući signale poslate od strane bilo kojeg satelita u nizu koji je vidljiv.

Prijemnik određuje vreme koje je potrebno da primi poruku, a zatim izračunava rastojanje do svakog satelita na osnovu ovih informacija. Parametri koji ulaze u proračun su: udaljenost satelita od prijemnika na osnovu vremena koje je prošlo od emitovanja do prijema signala, trenutne orbite satelita koja je poznata. Uređaj zatim trilateracijom proračunava poziciju i obavesti o trenutnoj poziciji.

Dok su u triangulaciji tri prijemnika dovoljna da se odredi lokacija, potrebno je razmotriti i faktor vremena satelita koji kruži. Potrebno je vreme, možda nekoliko sekundi, da signal stigne do Zemlje - bilo koja mala greška časovnika u satelitu, množi se sa tim vremenom, može da stvori velike greške pozicioniranja.



**Slika 1: Globalni sistem za pozicioniranje korišćenjem satelita**  
**Izvor: [1]**

Upotreba četvrtog satelita uklanja grešku iz jednačina. U većini slučajeva, prijemnik će koristiti četiri ili više satelita za izračunavanje svoje pozicije. Ovo nije striktno neophodno u slučajevima kada prijemnik već ima poznatu visinu (u slučaju nekih fiksnih prijemnika, na primer).

## 2.2 IP adresa

IP (Internet protokol) adresa je jedinstveni broj dodeljen svakom uređaju koji je povezan na Internet, koji omogućava komunikaciju sa drugim uređajima. Ovaj broj se može posmatrati na isti način kao i na kućnu adresu. IP adresa dodeljena uređaju kada se poveže na Internet omogućava slanje podataka drugim uređajima i prijem odgovora od njih.

U većini slučajeva, IP adresu dodeljuje Internet provajder (*Internet Service Provider – ISP*) u okviru blokova koji su na osnovu regiona dodeljeni lokalnom registru. Zbog toga se država, region, pa čak i grad uglavnom lako identifikuju za datu IP adresu.

## 2.3 GSM/CDMA

Mobilni ID je jedinstveni broj koji identifikuje svaki mobilni uređaj u određenoj mobilnoj mreži poput IP adresa.

Dve najpopularnije mreže na raspolaganju su [1]:

- *Global System for Mobile Communications (GSM)*
- *Code Division Multiple Access (CDMA)*

GSM je najstarija tehnologija mobilnih servisa. GSM je 2G tehnologija koja je dostupna u preko 200 zemalja, a najnoviji podaci pokazuju da je koristi više od 75% svetskih mobilnih korisnika.

CDMA je novija tehnologija od GSM, i dolazi i sa 2G i sa 3G vrstom usluge. Prednost CDMA tehnologije je u tome što omogućava većem broju korisnika da u isto vreme koriste istu frekvenciju, što GSM ne može da ponudi.

Koristeći triangulaciju baznih stanica, geografska širina i dužina mobilnog uređaja može lako da se otkrije. Što je više baznih stanica koji koriste triangulaciju, lokacija će biti tačnija. Zbog toga

ova tehnologija bolje funkcioniše u urbanim sredinama, gde postoji više baznih stanica nego u ruralnim područjima.

## **2.4 WiFi i Bluetooth**

WiFi i Bluetooth MAC adrese rade na sličan način kao IP adresa na uređaju. *Media Access Control* (MAC) adresa je jedinstveni broj dodeljen uređaju, obično od strane proizvođača. Ovaj broj je trajan i globalno je jedinstven identifikator uređaja u koji je ugrađen mrežni čip.

### 3 Usluge bazirane na lokaciji (LBS)

Usluge bazirane na lokaciji (*location-based services* – LBS) su se prvi put pojavile tokom 70-ih godina XX veka u SAD, kada je američko Ministarstvo odbrane otpočelo sa upotrebom sistema za globalno pozicioniranje (GPS), satelitskom infrastrukturom koja je služila za pozicioniranje ljudi i objekata, koje je imalo preciznost do 3 metra [2, p. 10]. GPS je prvenstveno zamišljen kao sistem koji bi se koristio u vojne svrhe, ali je 80-ih godina američka Vlada odlučila da se GPS pusti u slobodnu upotrebu širom sveta. Od tada, gotovo sve industrijske grane koriste tehnologiju GPS u svrhu poboljšanja svojih usluga i proizvoda, kao npr. automobilska industrija, koja integriše GPS prijemnike u automobile.

Informacija o lokaciji se uglavnom prikuplja upotrebom satelitskog sistema i GPS prijemnika. Međutim, krajem 90-ih godina zainteresovanost za LBS i pratećom tehnologijom se naglo proširila širom sveta kada se razvilo tržište mobilne telefonije.

LBS se prvenstveno koristi u tri oblasti [2, p. 11]: vojska i druge državne službe, hitne službe i komercijalni sektor. Značaj GPS preciznosti za pojedine tipove aplikacija je prikazan u sledećoj tabeli.

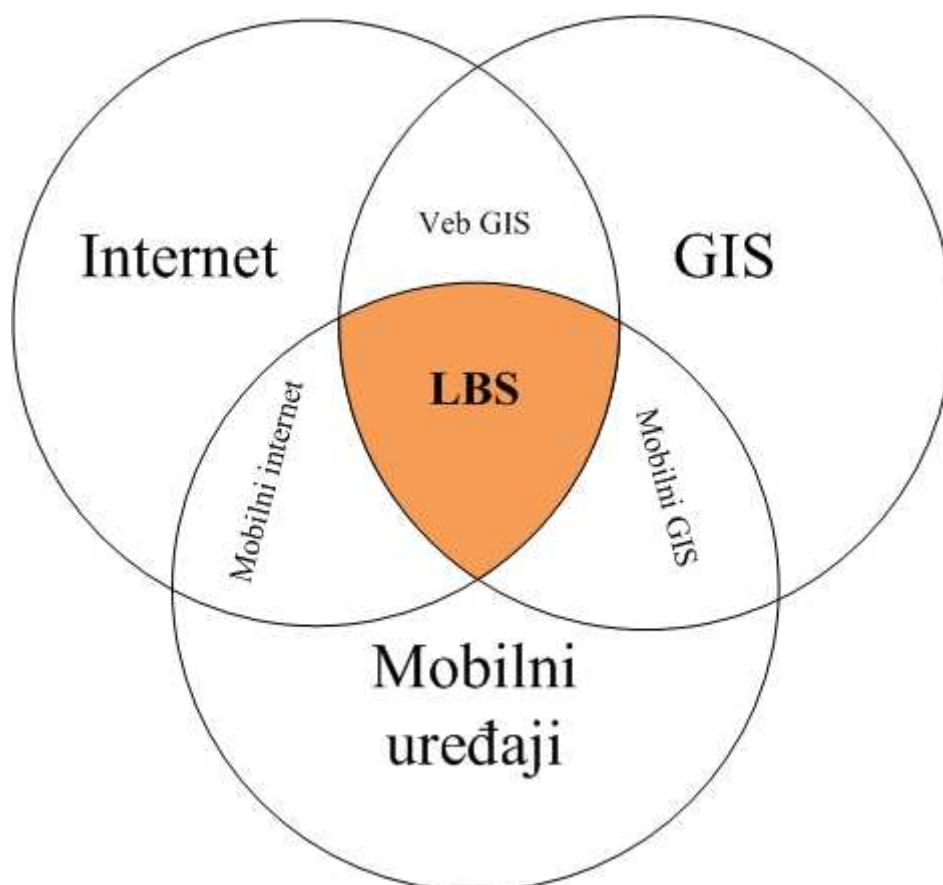
**Tabela 1: Značaj GPS preciznosti za pojedine tipove aplikacija**

| Tip aplikacije           | Nivo preciznosti | Tip aplikacije                      | Nivo preciznosti  |
|--------------------------|------------------|-------------------------------------|-------------------|
| Vesti                    | Nizak            | Igrice                              | Srednji           |
| Uputstva za kretanje     | Visok            | m-komerc                            | Srednje do visok  |
| Informacije o saobraćaju | Nizak            | Hitne službe                        | Visok             |
| Tačke od interesa        | Srednje do visok | Transport opasnih materija          | Visok             |
| Žute strane              | Nizak do srednji | Praćenje dece                       | Srednje do visok  |
| Auto navigacija          | Srednje do visok | Praćenje kućnih ljubimaca           | Srednje do visok  |
| Lična navigacija         | Visok            | Elektronska naplata putarine        | Srednje do visok  |
| Upravljanje flotom       | Nizak            | Daljinsko upravljanje radnom snagom | Nizak             |
| Praćenje automobila      | Srednje do visok | Lokalno reklamiranje                | Srednje do visoko |
| Praćenje imovine         | Visok            | Naplata bazirana na lokaciji        | Nizak do srednji  |

Izvor: [3]

#### 3.1 Struktura LBS

LBS aplikacije pružaju geografske usluge u realnom vremenu, npr. usluge mapiranja, planiranje ruta i slično [4]. Postoje dva načina na koji ove aplikacije funkcionišu [4]: korisnik traži geo-referenciranu uslugu ili administrator od korisnika zahteva geo-lokaciju. Kada se sazna lokacija korisnika (na osnovu geografskih koordinata), ona se sa ostalim relevantnim parametrima šalje u centar za procesiranje, gde se pohranjuju u bazu podataka. Tamo se, na osnovu podataka koji se nalaze u bazi, vrši analiza zahteva korisnika i korisniku se na uređaj dostavlja odgovor.



**Slika 2: Pozicija LBS-a u triangulaciji Interneta, GIS-a i mobilne tehnologije**  
**Izvor:** [4]

LBS aplikacije mogu da budu orijentisane u dva smera: ka korisniku i ka uređaju [2, p. 13]. LBS orijentisane ka korisniku se sastoje iz aplikacija čije su usluge bazirane na korisnicima, tj. da korisnik kontroliše uređaj i da na osnovu zadatih parametara dobije uslugu za koji je zainteresovan (npr. najbliža prodavnica kompjuterske opreme). LBS orijentisani ka uređaju su aplikacije koje ne zahtevaju da ih korisnik aktivira, već su implementirane u uređaje ili objekte (npr. GPS uređaj u automobilu). Kod ovih aplikacija korisnik uglavnom ne kontroliše direktno uređaj u kom se nalazi prijemnik.

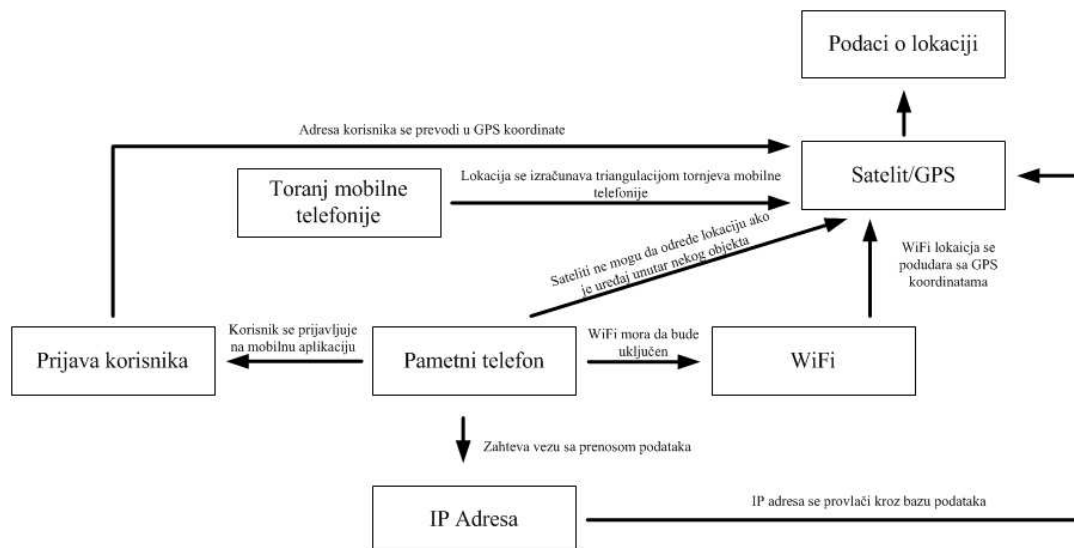
### 3.2 Metode prikupljanja podataka o lokaciji

Standardna metoda za prikupljanje podataka o lokaciji je upotreba GPS prijemnika. Međutim, postoji još najmanje četiri druge metode, koje se mogu i kombinovati, za prikupljanje podataka o lokaciji korisnika [5]:

- **Toranj mobilne telefonije:** Kada GPS signal ne može da dopre do čipa, što se dešava ukoliko je uređaj u zatvorenom prostoru, uređaj će u tom slučaju podatke o lokaciji dobiti od tornja mobilne telefonije na koji je povezan. Ovaj sistem je manje precizan od GPS pozicioniranja;
- **WiFi konekcija:** Ova metoda je precizna, ali zahteva aktivnu WiFi *hotspot*;
- **IP adresa:** Tačnost ovakvog pristupa zavisi od davaoca internet usluga (ISP);.

- **Korisnički definisana:** Kada se korisnik registruje na portal, često se od njih traži da unesu adresu stanovanja. Takvi podaci mogu biti prevedeni u GPS koordinate;

Slika 3 prikazuje proces prikupljanja podataka o lokaciji.



Slika 3: Proces prikupljanja podataka o lokaciji

Izvor: [6]

### 3.3 Primena LBS-a u praksi

Lokacijski bazirani servisi se mogu koristiti u sledeće svrhe [7]:

- **Marketing;**
- **Hitne službe** – jedan od osnovnih primene LBS je da se locira poziv pojedinca koji nije svestan svoje tačne lokacije ili nije u stanju da je otkrije zbog vanredne situacije. Na osnovu informacije o lokaciji hitna služba (npr. hitna pomoć, policija, vatrogasna služba) može da pruži pomoć na brz i efikasan način;
- **Informacioni servisi** – lokacijski bazirani informacioni servisi se odnose uglavnom na digitalne distribucije informacija na osnovu lokacije uređaja, vremenske specifičnosti i ponašanje korisnika. Ovo je jedan od najrasprostranjenijih i najranije implementiranih tipova LBS-a kao što je vremenska prognoza za određeni region/grad;
- **Navigacija** – navigacione usluge omogućavaju lociranje tačnog geografskog položaja mobilnog uređaja koristeći jedan od dostupnih sistema pozicioniranja i da dobije pravac za navigaciju korisnika do potrebnog mesta. Jedan od primera navigacije je mobilna aplikacija *Google Maps*;
- **Lokacijski bazirane društvene mreže;**
- **Lokacijski bazirane igre za mobilne telefone;**
- **Sport** – potencijal LBS-a i modernih mobilnih uređaja može se iskoristiti za praćenje sportske aktivnosti pomoću aplikacija kao što su *Nokia Sports Tracker*, *Nike+* itd. Funkcionalnost ovih aplikacija omogućava korisniku automatsko prikupljanje

podataka kao što su lokacija, udaljenost, brzina, dužina trajanja treninga ili sagorevanje kalorija, i njihovo skladištenje na serveru;

- **Geo označavanje** – geo označavanje (*geotaging*) se može definisati kao dodavanje geo-prostornih podataka digitalnim medijima, kao što su fotografije, video snimci, poruke, blogovi, veb stranice i GeoRSS. Geo označavanje dozvoljava pregledanje sadržaja na veb stranici, kao i vizualizacija sadržaja;
- **Praćenje;**
- **Turizam.**

### 3.3.1 Marketing

Lokacijskih bazirani marketing (LBM) je prisutan na tržištu već skoro jednu deceniju u formi SMS ili MMS poruka. LBM je novi termin marketing koji integriše mobilni marketing sa lokacijski baziranim servisima. Tehnologija je korišćena da označi lokaciju potrošača i obezbedi lokacijske oglase na svojim mobilnim uređajima. Postoje dve vrste usluga zasnovane na lokaciji: *push* i *pull*.

*Push* pristup je više svestran pristup i podeljen je u dve vrste, željene i ne željene oglase.

Neželjeni oglasi su češći slučaj, jer omogućavaju oglašivačima da ciljaju korisnike, dok kod željenih oglasa korisnici mogu da odrede tip reklama ili promotivnog materija koji će dobijati od oglašivača.

Nasuprot *Push* pristupu, *Pull* pristup je lakši za korisnike, koji mogu direktno tražiti informacije unošenjem ključnih reči.

### 3.3.2 Lokacijski bazirane društvene mreže

Društvene mreže kao što su *Facebook* i *Twitter* promenile su način kako ljudi komuniciraju i održavaju odnose sa prijateljima, kolegama, vršnjacima pa čak i sa porodicom. Razvoj pametnih telefona dalo je mogućnost integracije lokacije u okviru sadržaja. Ono što većina mreža ima zajedničko je „*check in*“, tj. mogućnost deljenja trenutne lokacije pojedinca na društvenoj mreži. Lokacijski bazirane društvene mreže marketarima su pružile nove mogućnosti za interakciju visoko ciljanih kupaca.

### 3.3.3 Lokacijski bazirane igrice za mobilne telefone

Lokacijski bazirane igrice (*Mobile Location Based Gaming* tj. *MLBG*) je rastući trend među LBG-om. *MLBG* ima elemente tradicionalnih igara na otvorenom kao što su žmurke, sa novim tehnologijama te igre su unapređene.

*MLBG* može biti definisan kao lokacijski bazirana igrice koja radi na mobilnom uređaju i kao komunikacioni kanal za razmenu informacija sa serverom ili drugim korisnikom koristi mobilni internet ili *WiFi* konekciju.

### 3.3.4 Praćenje

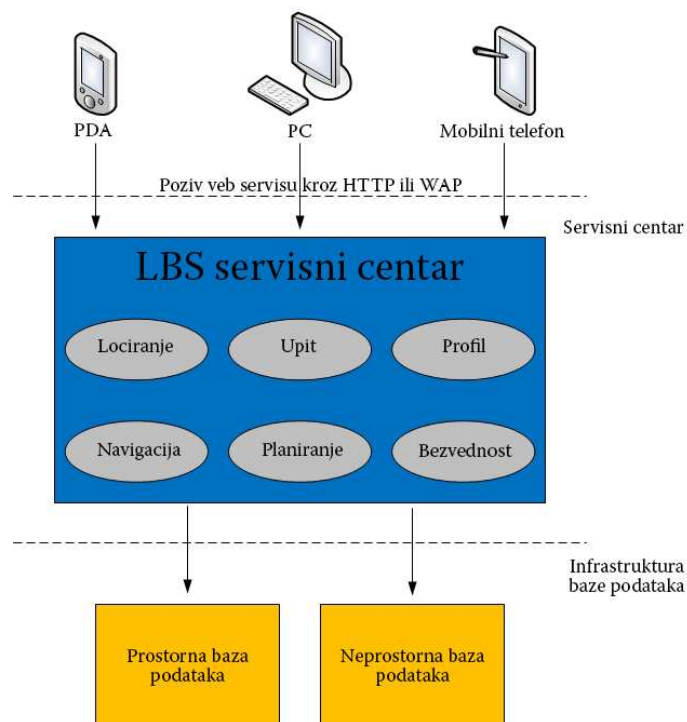
Praćenje u realnom vremenu je jedan od najkorisnijih aplikacija LBS-a. Može se koristiti za praćenje ljudi: deca, pacijenata sa demencijom, zatvorenici sa elektronskom narukvicom oko noge, poslodavcima da prate svoje radnike, kao i za praćenje životinja.

Praćenje vozila je još jedna primena LBS-a. Ovo se može odnositi na praćenje jednog vozila (npr. auto sigurnosni sistemi) kao i za kontrolu i koordinaciju celokupnih grupa vozila. Praktična primena ovog sistema je u *BusPlus-u*, korisnici mogu telefonima proveriti tačnu lokaciju autobusa.

### 3.3.5 Turizam

Turizam predstavlja jedno od polja u kojem je LBS našao veliku primenu, najviše kao sredstvo za planiranje ruta i navigaciju [8], gde turisti uz pomoć svog mobilnog telefona ili drugog prenosivog uređaja koji je svestan svoje lokacije mogu da planiraju rute kojima će da obiću atrakcije, dobiti informacije o atrakcijama kada dođu do njih ili dobijati informacije o lokaciji na kojoj se nalaze u realnom vremenu (vreme, gužva u saobraćaju i slično).

Jedan od sistema koji je razvije za potrebe turizam je Sistem za planiranje ruta i navigaciju (*Tourist route planning and navigation system – TRPNS*), koji je razvijen od strane grupe kineskih programera [9]. Sistem je koncipiran tako da pomaže turistima da određuju rutu putovanja i da se olakšano kreću duž određene putanje. Slika 4 prikazuje arhitekturu ovog sistema.



Slika 4: Arhitektura TRPNS sistema

Izvor: [8, 9]

## 3.4 Privatnost u LBS-u

Mnoge studije su pokazale da korisnici brinu za svoju privatnost i strahuju od bilo kakvih upada. Kao rezultat toga, operateri i marketari, ali i prijatelji moraju biti oprezni na koji način rukuju sa informacijama korisnika. Pored percepcije posmatrača, *spam* je postala druga pretnja u industriji. Lociranje bi znatno povećalo relevantnost poruka, a lokacijski bazirane *spam* poruke mogu ugroziti privatnost ljudi.

Kao rezultat ove potencijalne pretnje, Komisija EU je prepoznala problem u Direktivi o privatnosti i elektronskim komunikacijama (Direktiva 202/58/EC). Pored regulisanja mnogih aspekata elektronske komunikacije, poruke koje su „neželjene“ su eksplicitno pokriveni. Glavne tačke pokrivena u direktivi su sledeće [2, p. 17]:

- Automatsko pozivanje korisnika je dozvoljeno ukoliko su dali saglasnost;
- Ukoliko korisnik ima ugovor o kupovini usluga u kome je dozvoljeno korišćenje kontakt podataka u marketinške svrhe;
- Ukoliko marketar želi da koristi kontakt podatke mora da kontaktira korisnika za dozvolu za svaku poruku;
- Elektronske poruke koje prikrivaju identitet pošiljaoca ili su bez validne adrese za odgovor su zabranjene.

Takođe, prema članu 9 regulative, korišćenje podata o lokaciji je regulisano na sledeći način [10]:

- Podatke o lokaciji mogu obrađivati kada su podaci pribavljeni anonimno ili uz saglasnost korisnika za vreme trajanja usluge;
- Servis mora da obavesti korisnika, pre dobijanja saglasnosti, koju vrstu podataka prosleđuju, za koju svrhu i vremensko trajanje obrade, i da li će podaci biti dostupni trećem licu;
- Korisnici imaju mogućnost da povuku saglasnost za obradu podataka u bilo kom trenutku.

## 4 Pregled tehnologija

### 4.1 HTML 4

HTML je akronim od *HyperText Markup Language* i predstavlja jezik za označavanje koji se koristi za formatiranje sadržaja veb stranice. HTML predstavlja samu srž svih veb stranica. Kao modifikovanu i pojednostavljenu verziju SGML jezika, HTML standardizuje i održava (*World Wide Web Consortium* – W3C) [11].

Iako se HTML znakovi (*tags*) često nazivaju kodom, tehnički HTML nije kod, jer računarski kôd su instrukcije koje od računara traže da izvrši određenu operaciju. Svrha HTML znakova je da se struktura dokumenta označi tako da bi korisnikov pregledač (*user agent*), tj. Internet pretraživač mogao da prepozna strukturu dokumenta i ispravno je prikaže u prozoru koji korisnik koristi.

Za dodatno modifikovanje veb stranica, do sada su objavljeni neki dodaci [11]:

- CSS ili *Cascading Style Sheets* za izgled i položaj prezentacije
- Skriptni jezici (*Javascript*, *VBScript*) za omogućavanje dinamičnosti i interaktivnosti na veb prezentacijama,
- DOM ili *Document Object Model*, koji označava vezu između skripte i elementa na stranici

Pored HTML-a, na Internetu danas postoje jezici koji se koriste da korisnikovom pretraživaču dostave HTML koji će zavisiti od toga šta je korisnik tražio. Kôd se u tom slučaju izvršava na serverima i HTML koji je rezultat procesa biva dostavljen korisniku.

HTML strana sačinjena od HTML tagova se prikazuje pomoću namenski napisanog softvera koji se zove internet pretraživač. Trenutno najpopularniji internet pretraživači su *Google Chrome* i *Mozilla Firefox*. Kao što je već pomenuto, svrha veb pretraživača jeste da da pročita HTML dokumente i složi ih u, za korisnika, vizuelno prihvatljivu celinu. Pretraživač ne prikazuje HTML tagove, već te tagove interpretira i prikazuje sadržaj koji tagovi sadrže.

### 4.2 HTML 5

HTML 5 je poslednja revizija HTML jezika za označavanje, dizajnirana za kreiranje bogatijih i funkcionalnijih veb aplikacija. W3C, organizacija koja se bavi održavanjem HTML standarda počela je 2004. godine da radi na XHTML 2 koji je trebalo da zameni XHTML 1 i HTML 4 i postane primarni jezik za označavanje na internetu. Nisu svi bili zadovoljni ovom odlukom i tako se grupa veb profesionalaca, kompanija i proizvođača pretraživača odlučila da počne da razvija novi standard van W3C-a. Oni su oformili grupu WHATWG (*web hypertext application technology working group*) i počeli rad na svom standardu nazvanom *Web Applications 1.0*. Razlog ovome bilo je njihovo nezadovoljstvo načinom na koji je W3C razvijao novi XHTML 2 standard. Oni su smatrali da je W3C učinio premalo na rešavanju problema modernih veb aplikacija i da će nedostatak kompatibilnosti sa prethodnim verzijama usporiti napredak. W3C se složio sa WHATWG grupom 2007. godine i usvojio njihov standard *Web Applications 1.0* primenjujući ga u HTML 5. Od tog trenutka je obustavljen rad na XHTML 2 standardu. Za razliku od XHTML 2 HTML5 je zadržao kompatibilnost sa prethodnim verzijama što znači da njegova implementacija u već postojeće HTML 4 projekte neće prouzrokovati probleme ili prestanak funkcionalnosti već samo unapređenja. HTML 5 takođe donosi nove strukturalne, semantičke elemente i attribute,

omogućavajući dizajnerima i programerima da naprave kôd smislenijim. Kao što je već pomenuto HTML 5 ima veliki fokus na veb aplikacije. Donosi mnoštvo novih kontrola i specifičnih aplikacionih atributa i API-ja<sup>1</sup>. Takođe je bitno napomenuti da za razliku od prethodnih HTML specifikacija HTML 5 donosi striktna uputstva vezana za parsiranje i upravljanje greškama, a u cilju dobijanja identičnog ponašanja aplikacije u bilo kom pretraživaču.

HTML 5 dodaje mnogo novih sintaksnih funkcija. U te funkcije su uključeni [12]:

- `<video>`
- `<audio>`
- `<canvas>`
- `<svg>`

Ove funkcije su dizajnirane na taj način da olakšaju implementaciju i rukovanje multimedijalnim i grafičkim sadržajima na stranici, bez potrebe da se instaliraju dodaci i koriste API servisi.

Drugi novi elementi kao što su:

- `<header>`
- `<nav>`
- `<section>`
- `<article>`
- `<footer>`

su dizajnirani da se obogati semantički sadržaj dokumenta. Novi atributi su uvedeni u istu svrhu, a neki elementi i atributi su uklonjeni. Neki elementi kao što su:

- `<a>`
- `<cite>`
- `<menu>`

su promenjeni, redefinisani ili standardizovani [12].

#### 4.2.1 Strukturni markup

HTML 5 donosi niz novih elemenata i atributa koji definišu novi način strukturiranja HTML dokumenata.

Neki od tih elemenata su semantička zamena često korišćenih generičkog bloka `<div>` linijskog `<span>` elementa, npr. `<nav>` (namenjen za navigacioni deo sajta), `<footer>` (obično se odnosi na dno stranice ili poslednje linije HTML koda), ili `<audio>` i `<video>` umesto `<object>`.

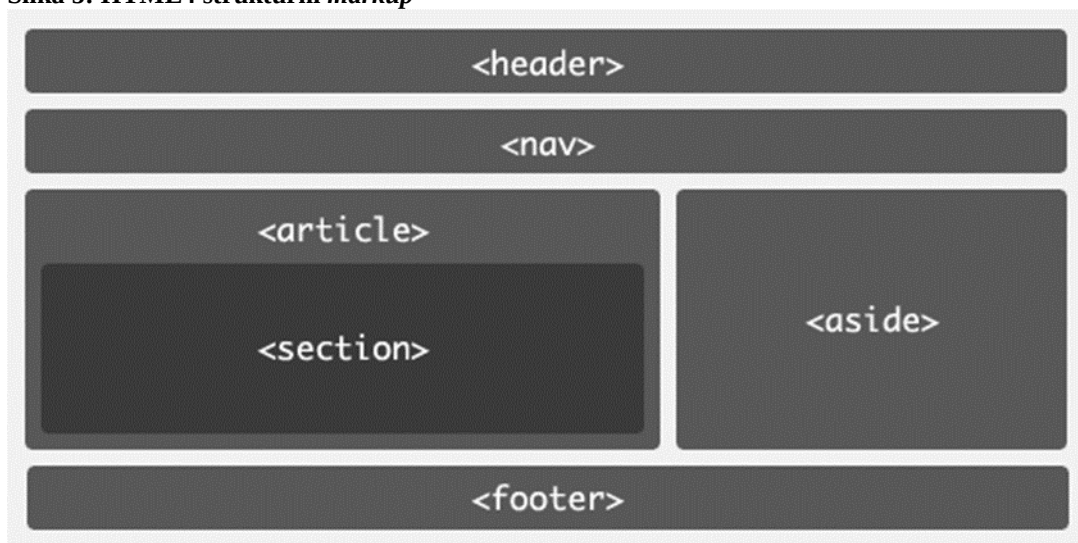
Neki zastareli elementi iz HTML 4.01 su uklonjeni, uključujući prezentacione elemente kao što su `<font>` i `<center>`, čiji efekti su postignuti korišćenjem CSS-a. Tu je istaknut značaj *JavaScript* jezika u ponašanju veb stranice.

---

<sup>1</sup> API – *Application programming interface* (Aplikacioni programski interfejs)



Slika 5: HTML4 strukturni *markup*



Slika 6: HTML5 strukturni *markup*

Izvor: [13]

#### 4.2.2 Aplikacioni programski interfejsi u HTML5

Pored novog označavanja, HTML 5 precizira i aplikacioni programski interfejs (API).

Novi API [12]:

- `<canvas>` element za 2D crtanje
- `<video>` i `<audio>` elementi za reprodukciju videa i zvuka
- *Web storage*
- *Drag&drop*
- *Microdata*
- GeoLokacija
- Veb SQL baza

## 4.3 XML

XML (*Extensible Markup Language*) je standardni skup pravila za definisanje formata podataka u elektronskoj formi koji je propisan od strane W3C. Sledeći pravila XML standarda, korisnici definišu sopstvene (XML) formate podataka, koje mogu koristiti za njihovo skladištenje, obradu i razmenu. XML je jezik orijentisan ka strukturi dokumenta i ne podrazumeva bilo kakvu definiciju prikaza. Ova osobina čini XML jezikom čiji su dokumenti češće korišćeni kao ulaz računarskog programa nego dokumenti koji će interpretirati veb klijent. XML dokumenta je moguće prikazati direktno u veb klijentima preko dva tipa transformacija [14, p. 13]:

- definisanja prikaza elemenata putem pomenutih CSS definicija
- prevođenja dokumenata u (X)HTML putem XSLT transformacija

XML omogućava stvaranje dugotrajnih formata podataka koji su nezavisni od platforme. Često se dokumenti pisani na jednoj platformi ne mogu čitati na drugim platformama, niti u različitim programima na istoj platformi, čak ni u ranijoj verziji jednog programa na istoj platformi. U XML dokumentima tekstualni su ne samo podaci već i etikete koje su smeštene u samoj XML datoteci. To znači da ih može čitati svaki program koju je u stanju da čita tekstualne datoteke. Tako se podaci mogu prenositi sa jednog sistema na drugi.

Osnovna svrha XML-a je da olakša deljenje podataka kroz različite informacione sisteme, posebno kroz one sisteme koji su povezani Internetom.

### 4.3.1 Deklaracija

Svaki XML dokument bi trebalo da otpočinje deklaracijom XML-a, ali to nije obaveza. Deklaracija sadrži *pseudoatribute* *version*, *standalone* i *encoding*. Na primer:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

Ako XML dokument poseduje deklaraciju XML-a, ona mora biti na početku dokumenta, a pre nje ne sme biti komentara, belina i sl. To je zato što XML analizator na osnovu prvih pet znakova (`<?xml`) prepoznaje način kodiranja znakova u dokumentu; ispred deklaracije XML-a može se naći jedino nevidljiva *Unicode* oznaka redosleda bajtova.

Atribut *version* uglavnom ima vrednost 1.0, jer zadavanje verzije 1.1 ograničava dokument na mali broj najnovijih verzija analizatora, dok svi analizatori za XML1.1 podržavaju i XML 1.0. Pošto neki sistemi ne pružaju podatke o skupu znakova, XML dopušta dokumentima da sami naznače svoj skup znakova koristeći deklaraciju kodiranja (*encoding declaration*). Ona se nalazi unutar deklaracije XML-a.

Atribut *encoding* je proizvoljan. Ako je izostavljen, analizator pretpostavlja da je upotrebljen *Unicode* skup znakova.

Atribut *standalone* (samostalan) je takođe neobavezan. Može imati vrednosti *yes* i *no*. Ukoliko je izostavljen, pretpostavka je da ima vrednost *no*.

### 4.3.2 Sintaksa

U XML dokumentima podaci su smešteni u obliku znakovnih nizova (*strings*), koji se nalaze između tekstualnih etiketa koje ih opisuju. U XML-u se osnovne jedinice etiketa i podataka nazivaju elementi.

Korisnik se pri pisanju etiketa mora pridržavati precizno definisane sintakse. Strukturu XML dokumenta opisuju njegove etikete, pomoću kojih se može videti koji su elementi pridruženi drugim elementima. Ako je XML dokument dobro projektovan, etikete opisuju i njegovu semantiku (npr. etikete ukazuju da li je element naslov, ime osobe, datum i sl.).

Primer sintakse u XML-u:

```
<osoba>Vladimir Lazović</osoba>
```

Složeniji XML dokument:

```
<osoba>
  <ime_i_prezime>
    <ime>Vladimir</ime>
    <prezime>Lazović</prezime>
  </ime_i_prezime>
  <zanimanje>web programer</zanimanje>
</osoba>
```

#### 4.3.3 Atributi

XML atributi daju podatke koji dodatno opisuju elemente XML dokumenta. Atributi imaju svoj naziv i vrednost. Ime atributa je razdvojeno od vrednosti znakom jednakosti i razmakom, koji je neobavezan. Vrednost atributa mora biti zatvorena u navodnike ili polunavodnike. Na primer:

```
<osoba rodjena="1988">
  Vladimir Lazović
</osoba>
```

#### 4.3.4 Pravilnost XML dokumenta

Za svaki XML dokument koji je kreiran u skladu sa sintaksičkim pravilima može se reći da je dobro oblikovan. Sve dokumente koji nisu u skladu sa ovim pravilima, programi za obradu XML-a odbijaju.

Unutar elementa XML dokumenta ne sme se naći znak < koji nema odgovarajuću izlaznu sekvencu </, jer se znak < uvek tumači kao početak etikete. Ovaj znak se, ukoliko korisniku zatreba u tekstu, može pretvoriti u izlaznu sekvencu pomoću reference entiteta **&lt;**, numeričke reference znaka **&#60;**, ili heksadecimalne reference znaka **&#x3c;**. Reference entiteta i numeričke reference znaka spadaju u markiranja; pri analiziranju XML dokumenta, markiranja se zamenjuju onim znakom na koji upućuje referenca.

XML poseduje pet unapred definisanih entiteta. To su:

- **&amp;** (&, ampersand ili ampersend);
- **&lt;** (<, znak „manje od“);
- **&gt;** (>, znak „veće od“);
- **&quot;** („„, običan (ravan) navodnik);

- **&apos;** (‘, ravan polunavodnik, apostrof).

U sadržaju elemenata se samo **&amp;** i **&lt;** moraju upotrebljavati umesto **&** i **<**, dok su ostale reference opcione.

Dobro oblikovan dokument mora da zadovoljava određena pravila. Na primer:

- za svaki XML dokument mora postojati tačno jedan korenski element;
- svaki element koji nije prazan mora imati početnu etiketu i odgovarajuću završnu etiketu;
- prazan element može biti označen oznakom praznog elementa (kao što je `<br />`, što je ekvivalentno `<br></br>`);
- sve vrednosti atributa moraju biti zatvorene u navodnike ili polunavodnike;
- elementi mogu biti ugnežđeni, ali se ne smeju preklapati, a svaki element, osim korenskog, mora se u potpunosti sadržati u drugom elementu;
- jedan element ne sme imati dva istoimena atributa;
- unutar oznaka ne smeju se koristiti komentari ni instrukcije za obradu;
- znakovi `<` ili **&** koji nisu pretvoreni u izlaznu sekvencu ne smeju se naći u znakovnim podacima elemenata i atributa;
- nazivi elemenata mogu sadržati slova, brojeve i posebne znakove, moraju počinjati slovom, ali ne smeju počinjati tekстом xml ni XML, a ne smeju sadržati ni praznine;
- takođe, imena elemenata su osetljiva na razliku između malih i velikih slova. Tako elementi OSOBA, osoba i Osoba za XML nisu isti.

Za XML dokument koji je formalno ispravan, tj. dobro oblikovan, a pri tom i zadovoljava određenu šemu, može se reći da je validan. Dokumenti koji šemu ne zadovoljavaju nisu validni. To znači da validnost dokumenta zavisi od šeme sa kojom se poredi.

## 4.4 MySQL

MySQL je besplatan, sistem otvorenog koda za upravljanje bazama podataka. PostgreSQL i MySQL su čest izbor sistema za upravljanje bazama podataka za projekte otvorenog koda, te se distribuira kao sastavni deo javnih repozitorijuma softvera, serverskih GNU/Linux distribucija, no takođe postoje instalacije i za ostale operativne sisteme poput Mac OS-a, Windows itd. [15].

MySQL baza je slobodna za većinu upotreba. Ranije u svom razvoju, MySQL se suočavao sa raznim protivnicima, jer su joj nedostajale neke osnovne funkcije definisane SQL standardom. Naime, MySQL je optimizovana kako bi bila brza nauštrb funkcionalnosti. Nasuprot tome, vrlo je stabilna i ima dobro dokumentirane module i ekstenzije, prema tome i podršku brojnih programskih jezika: PHP, Java, Perl, Python...

MySQL baze su relacionog tipa, koji se pokazao kao najbolji način skladištenja i pretraživanja velikih količina podataka i u suštini predstavljaju osnovu svakog informacionog sistema, tj. temelj svakog poslovnog subjekta koji svoje poslovanje bazira na dostupnosti kvalitetnih i brzih informacija.

MySQL i PHP su osvojili veliki deo tržišta jer su softver otvorenog koda i besplatni su što znači da se mogu besplatno koristiti u sve svrhe, pa i komercijalne.

## 4.5 PHP

PHP (PHP - *Hypertext Preprocessor*) je specijalizovani skriptni jezik koji se koristi za izradu dinamičkih veb stranica, a uglavnom za dinamičko generisanje HTML koda [16].

Upotrebom PHP-a moguće je kreirati HTML stranicu popunjenu dinamičkim sadržajem na serveru, pre njenog slanja klijentu. Na ovaj način PHP kôd koji je generisao stranu se ne može videti, već e prikazuje samo HTML.

Po sintaksi je sličan C programskom jeziku, a čak imaju i neke slične funkcije. Što znači da jednu radnju možete izvesti korišćenjem više različitih funkcija.

PHP je 1995. godine napisao Rasmus Lerdorf iz PHP/FI, upotrebom Perl skripte na svojim veb stranicama [16]. On je taj program nazvao „Alat za ličnu prezentaciju”, a kasnije je na to dodavao funkcije iz programskog jezika C za komunikaciju sa bazama podataka.

PHP 3.0 je predstavljao prvu verziju koja dosta liči na današnje verzije PHP-a. Kreirali su ga izraelski programeri Endi Gutmans i Zeev Suraski 1997. godine, kao potpunu preradu projekta PHP/FI [16]. Ime je promenjeno u PHP (bez dodatka 'FI'), čije se značenje izmenilo da predstavlja rekurzivni akronim „PHP: pre-procesor hiperteksta“. PHP 4.0 je nastao 1998. godine, a verzija PHP 5.0 je 2004. godine donela mnogo izmena u objektno orijentisanom programiranju [16].

## 4.6 jQuery

*JavaScript* je objektno orjentisan programski jezik namenjen dinamičkoj izradi veb sajtova. Koristi se u milionima veb sajtova za poboljšanje funkcionalnosti, validaciju formi, detekciju pretraživača, asinhronu komunikaciju sa serverom i još mnogo toga. Razvijen je 1995. godine od strane *Netscape*-a i koristi se kao *client-side* (na strani klijenta) Internet programski jezik. To znači da se njegova obrada vrši na korisnikovom računaru u samom pretraživaču, čime se postiže veća brzina rada, a veb aplikacije odgovaraju brže na korisnikove akcije i samim tim se postiže veća efikasnost. Predstavlja skriptni jezik.

*JavaScript* se u HTML dokument može ubaciti na dva načina. Pisanjem direktno u HTML kôd ili smeštanjem u posebnu datoteku koji se potom poziva u HTML dokumentu. Pretraživač čita HTML stranicu i prikazuje elemente onako kako oni nailaze. Kada naiđe na skript u stranici, obavlja se njeno izvršavanje pre nego što se nastavi čitanje ostalih elemenata.

*JavaScript* je *case-sensitive*, što znači da obraća pažnju na velika i mala slova. *JavaScript* kao i ostali programski jezici, koristi promenljive da bi zapamtio vrednosti, tako da se one kasnije mogu koristiti u drugim delovima programa.

*JavaScript* za razliku od nekih drugih programskih jezika ne zahteva navođenje tipa vrednosti promenljive, već to čak i ne dozvoljava. Ista promenljiva može da se koristi za smeštanje različitih tipova vrednosti, kao što su stringovi, celi brojevi, vrednosti realnog broja ili logičke vrednosti. *JavaScript* interpreter automatski konvertuje tip podataka koji se nalazi u promenljivoj zavisno od potrebe.

Postoje dve vrste promenljivih:

- Lokalne promenljive
- Globalne promenljive

Lokalne promenljive se oznakom `var` definišu unutar neke funkcije i dostupne su samo unutar iste, dok se globalne promenljive definišu izvan funkcije i njima se može pristupiti iz bilo koje funkcije. Ukoliko se promenljiva definiše unutar funkcije bez oznake `var`, ona će tada biti globalna.

*JavaScript* kao jedan od veoma laganih, ali ujedno i glavnih jezika veba, pruža pregršt mogućnosti za programere. Međutim potpuna moć *JavaScript*-a se otkriva korišćenjem njegovih biblioteka. Jedna od *JavaScript* biblioteka je *jQuery*.

*jQuery* je brza *JavaScript* biblioteka koja pojednostavljuje upravljanje i rukovanje HTML elementima, omogućava animacije i AJAX pozive za ubrzani veb razvoj. Biblioteka *jQuery* je osmišljena da promeni način na koji se piše *JavaScript*.

Pomoću *jQuery* biblioteke, mogu da se kreirati animirani efekti koji prijaju oku, a koji u isto vreme parira *Flash* tehnologiji animiranih sadržaja za veb.

*jQuery* je dobar za:

- Dodavanje efekata tekstu (efekti izbleđivanja teksta, nestajanja, ulaska/izlaska teksta)
- Pravljenje XML (*Ajax*) upita (za traženje dodatnih informacija od veb servera, bez potrebe za osvežavanjem stranice)
- Manipulacija DOM-om (jednostavno dodavanje, uklanjanje i premeštanje sadržaja veb stranice pomoću *JavaScript* koda)
- Pravljenje slajd šoua
- Pravljenje menija sa podmenijem
- Kreiranje drag & drop interfejsa (tako da na stranici elementi mogu da se premeštaju jednostavnim prevlačenjem).

## 4.7 Android operativni sistem

Android operativni sistem je trenutno najrasprostranjeniji operativni sistem za mobilne telefone. Zasnovan je na *Linux* kernelu i prilagođen tako da se može koristiti na većini mobilnih uređaja sa ekranom osetljivim na dodir, što pored mobilnih telefona uključuje i tablične računare, laptop računare, *smartbook* računare, čitače elektronskih knjiga, televiziju (Android TV), automobile (Android Auto) i ručne satove [17].

Kako Android OS odstupa od većine standardnih operativnih sistema baziranih na *Linux* kernelu, npr. Android nema standardni *X Window System* ili standardni skup GNU biblioteka pa tako nije u mogućnosti da pokreće aplikacije razvijene za druge standardne *DNU/Linux* bazirane sisteme.

Aplikacije na Android sistemima se ne pokreću direktno, već se pokreću u okruženju odvojenom od ostatka sistema, gde dobijaju samo određeni deo sistemskih resursa, pa tako nemaju pristup delovima sistema koji su im nepotrebni, što poboljšava sigurnost i stabilnost sistema. Takođe, pri instalaciji aplikacija korisnik dobija listu svih dozvola koje jedna aplikacija zahteva da bi se instalirala, što korisniku daje mogućnost da uoči potencijalno maliciozne aplikacije i obustavi njihovu instalaciju pre nego što dođe do oštećenja [17].

Sa tehničke strane Android predstavlja *Linux* bazirani operativni sistem razvijen za ARM i x86 arhitekturu i sastoji se od modifikovanog monolitnog *Linux* kernela zaduženog za podršku hardvera i funkcija niskog nivoa, skupa biblioteka zaduženih za dodatne podrške kao što su generisanje grafike, podrška za dekodovanje video snimaka, podrške za SSL šifrovanje itd. U

sklopu biblioteka se nalazi i odvojeni *Android Runtime* koji sadrži osnovne, biblioteke i Dalvik virtualna mašina zadužena za pokretanja aplikacija višeg nivoa napisanih u Java programskom jeziku. Na višem nivou od biblioteka su sistemske aplikacije neophodne za upotrebu sistema od strane korisnika, a među njima su *window manager*, menadžer resursa, menadžer instalacionih paketa, kao i aplikacije zadužene za obavljanje osnovnih funkcija vezanih za mobilne telefone ili uređaj na kom je instaliran Android. Na najvišem nivou se nalaze krajnje korisničke aplikacije, odnosno aplikacije koje direktno pokreće korisnik.

Ovakva arhitektura sistema nije iznenađujuća, jer predstavlja standardnu arhitekturu *Linux* sistema, gde su segmenti sistema razdvojeni po nivoima na kojim rade.

Za crtanje 3D grafike Android koristi biblioteku zasnovanu na *OpenGL ES 2.0* specifikaciji, što ovom sistemu daje mnoge napredne grafičke sposobnosti.

Kroz svoju istoriju Android je imao nekoliko verzija od kojih je svaka donosila neku novinu i poboljšanje, tako je npr. verzija 1.0 bila prva zvanično dostupna verzija Android operativnog sistema, V1.5 *Cupcake* je bila njena nadogradnja zasnovana na *Linux* kernelu 2.6.27, V1.6 *Donut* je koristio 2.6.29 *Linux* kernel i imao još više dodatnih mogućnosti u odnosu na prvu 1.0 verziju. Sa pojavom verzija 2.0 i 2.1 pod nazivom *Eclair* ispravljene su mnoge postojeće greške u samom sistemu i dodate dodatne podrške za rad sa kamerom, kao i poboljšana virtualna tastatura. Verzija 2.2 *Froyo* je prešla na novi kernel 2.6.32, ubrzala je rad sa memorijom i poboljšala performanse samog sistema. Verzija 2.3 *Gingerbread* takođe prelazi na novi kernel 2.6.35 i dodatno poboljšava korisnički interfejs, a takođe donosi i podršku za veće displeje, kao i za neke dodatne senzore. Verzija 3.0 poznata i kao *Honeycomb* je bila zasnovana na kernelu 2.6.36 i bila je prilagođena tabličnim računarima, a dodati su joj interfejs elementi kao što su *system bar* i *action bar* koji su prilagođeni za tablične računare. Takođe je pojednostavljena i upotreba multitaskinga, redizajnirana je i virtualna tastatura, tako da omogući lakše i brže kucanje, a uklonjeni su i neki sigurnosni propusti [17].

Verzija 4.4 koja nosi naziv *KitKat* je trenutno najnovija verzija Android operativnog sistema [18], a pored 4.4, aktuelne su još i 4.2 *Jelly Bean* i 4.3 *Jelly Bean*. Objavljena je preliminarne verzija Android 5.0 sistema.

## 4.8 Geografski informacioni sistemi

Savremeni geografski informacioni sistem (GIS) je organizovan skup računarskog hardvera, softvera, podataka, osoblja i mreža radi efikasnog prikupljanja, skladištenja, ažuriranja, rukovanja, analize, modelovanja, prenosa i prikaza svih oblika prostornih informacija.

GIS je oblast informacionih tehnologija koja se razvija velikom brzinom.

Savremenim GIS podrazumeva da je svet sačinjen od objekata. Prostorni objekti su objekti koji imaju svoju prostornu odrednicu, ili drugačije rečeno imaju svoju jedinstven ID i opisani su svojim geometrijskim, tematskim, radiometrijskim i vremenskim atributima.

U zadnjih nekoliko godina pojavile su se digitalne fotogrametrijske stanice bazirane na GIS tehnologiji. Digitalne fotogrametrijske stanice su izvor za prikupljanje podataka u mnogim GIS sistemima. Jednu od presudnih uloga u kvalitetu informacionog sistema daju upravo podaci. Stoga se posebna pažnja mora posvetiti održavanju kvaliteta prostornih podataka u GIS sistemu kroz ažuriranje i poboljšanje kvaliteta (*updating and refinement*). [19]

GIS čine tri važne komponente [20]:

- Kompjuterski hardver i softver;
- Prostorni podaci;
- Obučeni ljudi.

Kompjuterski hardver čine kompjuteri na kojima radi GIS. Softver radi na širokom spektru hardvera od servera do desktop računara. GIS softver obezbeđuje funkcije i alate koje su potrebne za unos i čuvanje geografskih informacija i pruža alate za upite, analizu i prikazivanje podataka u formi mape ili izveštaja. Svi GIS softverski paketi oslanjaju se na sistem za upravljanje bazama podataka (*DataBase Management System* - DBMS), sa kojima komunicira i obavlja upite koji zavise od potreba korisnika.

Prostorni podaci su jedni od najvažnijih, često i najskupljih, komponenata GIS. Geografski podaci, koji se sastoje od odgovarajućih informacija, unose se u sistem koristeći tehniku digitalizacije. Ovaj proces uključuje digitalno kodiranje geografskim karakteristika, kao što su zgrade, putevi ili granice zemalja.

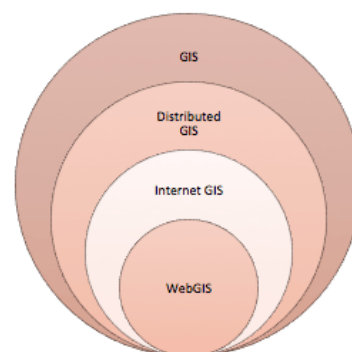
Digitalizacija se vrši praćenjem lokacije, puta ili administrativne granice geografskim karakteristikama na ekranu računara koristeći skeniranu mapu u pozadini, ili papirnu mapu koja je vezana za digitalizacionu tablu. Proces digitalizacije je veoma dugotrajan posao naročito kada se snimaju veliki skupovi podataka, kao što je zemljište, potoci ili topografske konture.

Prava moć geografskih informacionih sistema dolazi od ljudi koji ih koriste. Tokom protekle decenije, kompjuteri su postali mnogo lakši za korišćenje i pristupačniji za korisnike. Imajući u vidu ovu činjenicu, broj GIS korisnika je naglo porastao. Ljudi danas GIS koriste u različitim oblastima kao sredstvo koje im omogućava da što efikasnije obavljaju svoje poslove.

## 4.9 WebGIS

WebGIS je vrsta distribuiranog informacionog sistema, koji sadrži GIS aplikacioni server, a klijent može biti internet pretraživač (*web browser*), desktop aplikacija ili mobilna aplikacija (Slika 7).

Server ima svoj URL kako bi ga klijenti pronašli na vebu. Klijent se oslanja na HTTP specifikacije da bi slao zahteve ka serveru. Server sprovodi zatražene GIS operacije i šalje odgovor klijentu, ponovo putem HTTP-a. Format odgovora može biti u HTML-u, koji koriste veb čitači, ali može biti i u drugim formatima, kao što je binarna slika, XML (*Extensible Markup Language*) ili JSON (*JavaScript Object Notation*).



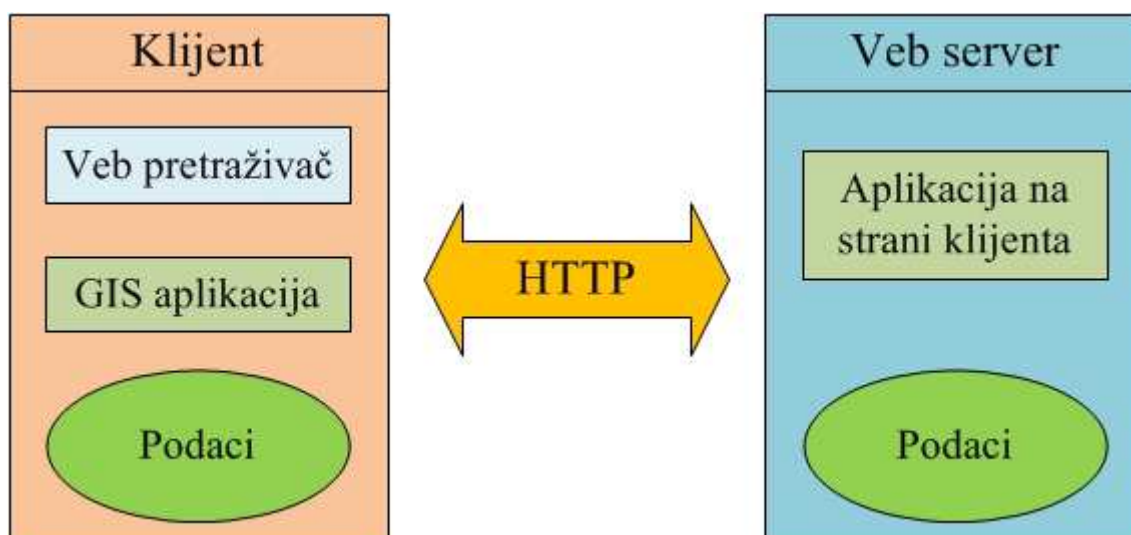
Slika 7: Odnos pojmova od GIS-a do Web GIS-a

Izvor: [21, p. 154]

### 4.9.1 Arhitektura WebGIS-a

Postoje dve vrste arhitekture kada se govori o razvoju WebGIS aplikacija: *client-side* (na strani klijenta) i *server-side* (na strani servera).

U *client-side* arhitekturi, aplikacije teže da prebace deo obrade zahteva na računar korisnika, umesto da server uradi veći deo obrade. Neke aplikacije zahtevaju instalaciju dodatka za Internet pretraživač, kako bi se omogućio pristup serveru, dok postoji mogućnost instalacija kompletne aplikacije. U oba slučaja *client-side* aplikacije zahtevaju instalaciju neophodnog softvera.



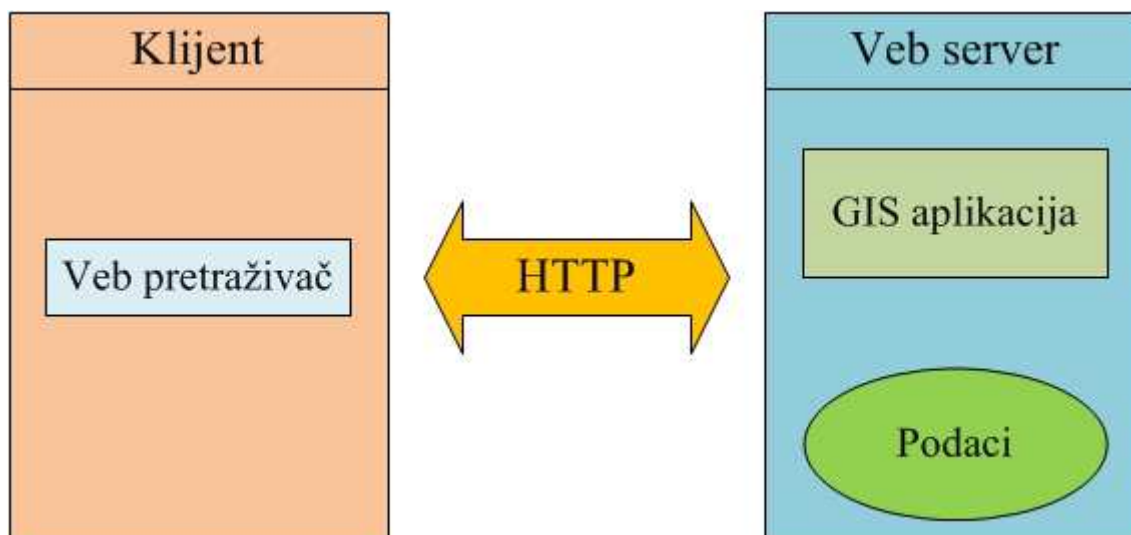
Slika 8: *Client-side* arhitektura  
Izvor: [22]

Tabela 2: Prednosti i nedostaci *client-side* arhitekture

| Prednosti <i>client-side</i> arhitekture                                                   | Nedostaci <i>client-side</i> arhitekture                                                                    |
|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| aplikacije koriste resurse računara korisnika                                              | mogućnost servera da šalje velike količine podataka                                                         |
| veća kontrola procesa analize podataka                                                     | veliki i kompleksni podaci mogu biti komplikovani za obradu ukoliko klijent nema dovoljno dobre performanse |
| nakon dostavljanja podataka korisnik nastavlja sa radom bez potrebe da šalje/prima podatke | kompleksne analize se izvode sporije na računarima sa lošijim performansama                                 |

Izvor: [22]

U *server-side* arhitekturi korisnik šalje zahtev serveru, server obrađuje zahtev i vraća rezultate korisniku preko standardnog HTTP protokola. Odgovor koji korisnik dobije od server je standardna veb strana koji internet pretraživač može da prikaže. Prednost ove arhitekture je ta što aplikacija i podaci ostaju centralizovani na serveru, to pojednostavljuje razvoj, primenu i održavanje.



Slika 9: *Server-side* arhitektura

Izvor: [22]

Tabela 3: Prednosti i nedostaci *server-side* arhitekture

| Prednosti <i>server-side</i> arhitekture | Nedostaci <i>server-side</i> arhitekture             |
|------------------------------------------|------------------------------------------------------|
| Lakši razvoj                             | svaki zahtev upućen serveru mora biti tamo i obrađen |
| Lakše održavanje                         | na performanse obrade utiču brzina interneta         |
| Podržava internet standarde              | ne koristi se procesna mogućnost računara klijenta   |
| Zahteva standardni internet pretraživač  |                                                      |

Izvor: [22]

## 4.9.2 Upotreba WebGIS-a u praksi

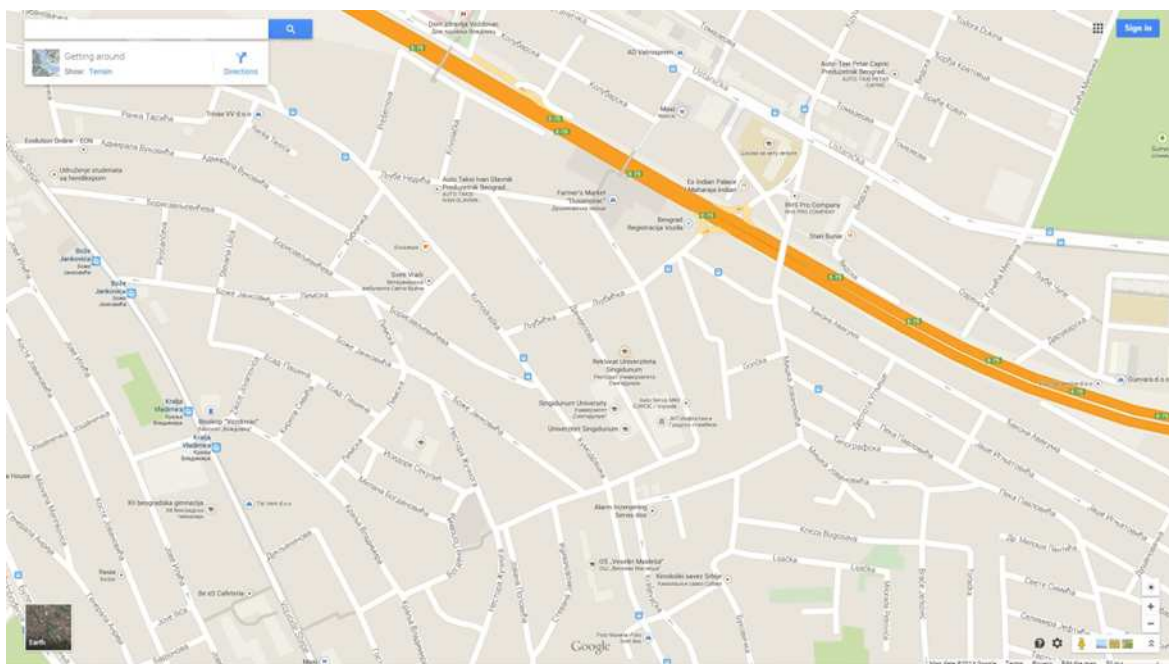
### 4.9.2.1 Google maps

*Google maps* je usluga na Internetu, omogućena od strane kompanije *Google*, koja pruža detaljne informacije o geografskim regionima i lokacijama širom sveta. Pored uobičajenih mapa puta, *Google Maps* nudi satelitske snimke mnogih gradova. U nekim gradovima je moguće videti fotografije snimljene iz vozila.

*Google Maps* nudi nekoliko usluga kao deo veb aplikacije [23]:

- Mogućnost planiranja putanje vozilom, biciklom, javim prevozom ili pešaka od jednog do drugog mesta;
- *Google Maps* aplikacioni servis omogućava ugradnju mapa u sisteme kao što su sajtovi, aplikacije za desktop računare, aplikacije za mobilne uređaje i dr.;
- *Google Maps* za mobilne uređaje nudi servis za vozače koji koristi podatke o GPS lokaciji mobilnog uređaja, kao i podatke bežičnih ili mobilnih mreža;
- *Google Street View* omogućava korisnicima pregledanje i navigaciju kroz panoramski pogled 360° horizontalno i 290° vertikalno sa sredine ulica, gradova širom sveta;
- Dodatne usluge nude slike Meseca, Marsa i svemira za astronome amatere.

Satelitski snimci *Google* mapa se ne ažuriraju u realnom vremenu, ali *Google* na dnevnom nivou dodaje podatke u svoju bazu, tako da većina slika nije starija od tri godine.



Slika 10: Google Maps

#### 4.9.2.2 Bing Maps

*Bing Maps* je usluga za veb mapiranje koja je integrisana u *Microsoft Bing* paket. Korisnici mogu da pregledaju i pretražuju topografski osenčene mape mnogih gradova širom sveta. Mape sadrže ugrađene tačke od interesa, kao što su metroi, stadioni, bolnice, škole, restorani i drugi objekti. Moguće je pretraživanje javno dostupnih tačaka koje su korisnici kreirali.

*Bing Maps* predstavlja sledeću generaciju *Virtual Earth* platformi koja je u ponudu *Microsoft* kompanije.

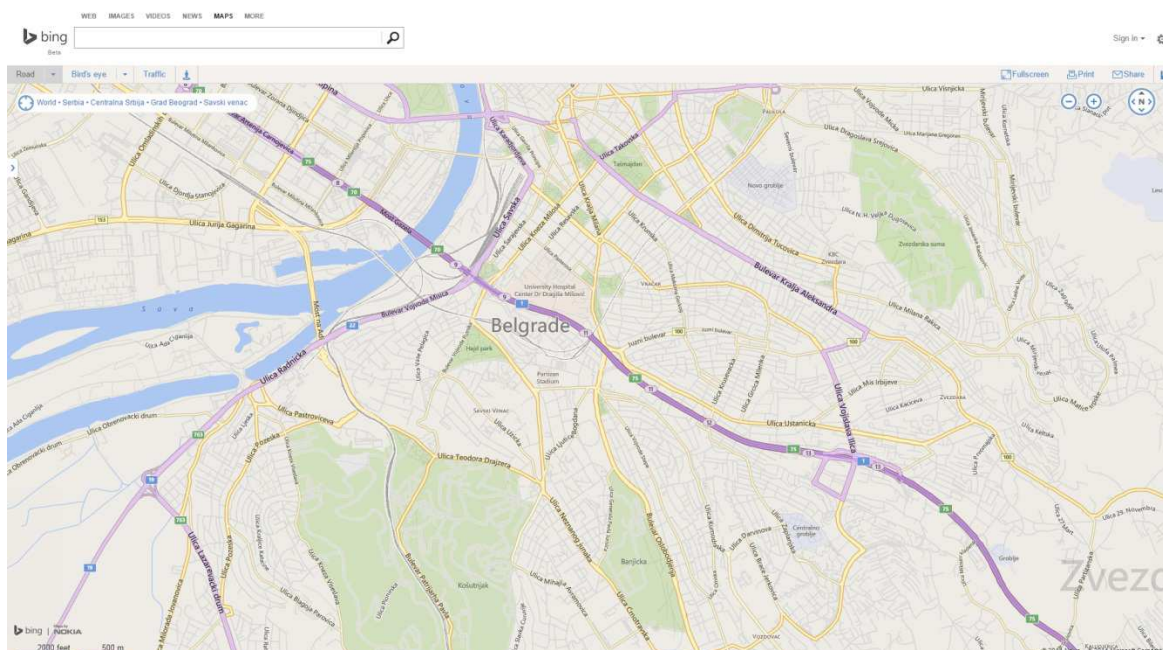
Programeri imaju fleksibilnost da se koriste šest *Bing Maps* API servisa [24]:

- *Bing Maps* API, omogućava programerima da kreiraju aplikacije koje koriste *Bing Maps*;
- *Silverlight Map* kontrola, kôd izgrađen na *Silverlight 3*;
- *Bing Map AJAX Control*, omogućava korisnicima da šalju zahteve preko *Javascript*-a do *AJAX* objekta;
- *Bing Maps SOAP i Rest Services*, omogućavaju laku integraciju funkcionalnosti mapa u sve vrste aplikacija;
- *MapPoint Web Service*, koristi *SOAP XML* kako bi komunicirao sa aplikacijom korisnika.

*Bing Maps* nudi pet vrsta pogleda [24]:

- *Road View* – podrazumevani prikaz mape koji prikazuje ulice;
- *Aerial View* – prikazuje satelitske snimke na karti i naglašava ulice i znamenitosti;
- *Bird's Eye View* – prikazuje mapu iz ptičije perspektive, gde su slike snimane pod uglom od 45°;

- *StreetSide View* – obezbeđuje 360° prikaz slike koje su snimljene kamerom koja je montirana na automobil;
- *3D View* – prikaz zgrada u tri dimenzije.



**Slika 11: Bing Maps**

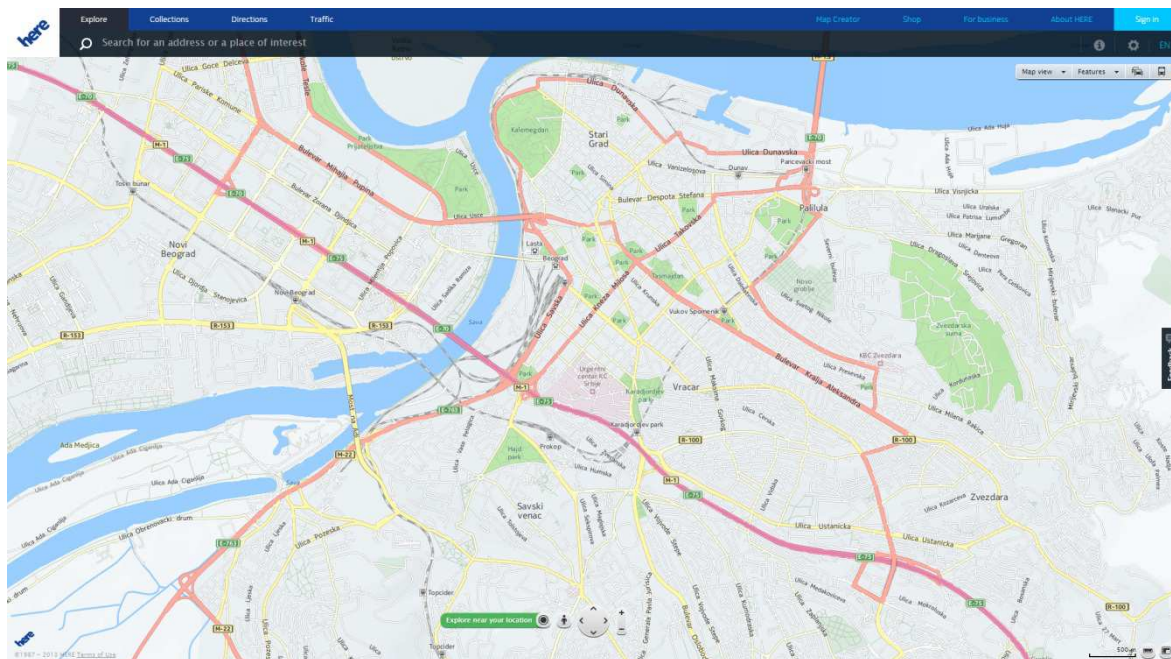
#### 4.9.2.3 Here

*Here*, nekada *Ovi Maps* i *Nokia Maps*, je poslovna jedinica kompanije *Nokia* koja integriše kreiranje mapa i lokacijske usluge pod jedan brend. Tehnologija koju koristi *Here* je bazirana na *cloud-computing* modelu, sto znači da se podaci i usluge čuvaju na serverima kako bi korisnici imali pristup bez obzira na uređaj koji koriste.

Here mape su najzastupljenije na *Windows mobile* operativnim sistemima, dok su od novembra 2012. godine su dostupne i za *Android* i *iOS*.

*Here* veb sajt nudi [25, 26]:

- Podršku rutiranja između više tačaka;
- 3D mape u 25 gradova, sa opcijom rutiranja;
- Pretraga javnog prevoza;
- Sinhronizacija sajta i korisnikovog mobilnog uređaja;
- Vizuelno prikazivanje područja koje je popularno za izlaske, hranu, šoping i znamenitosti u gradu.



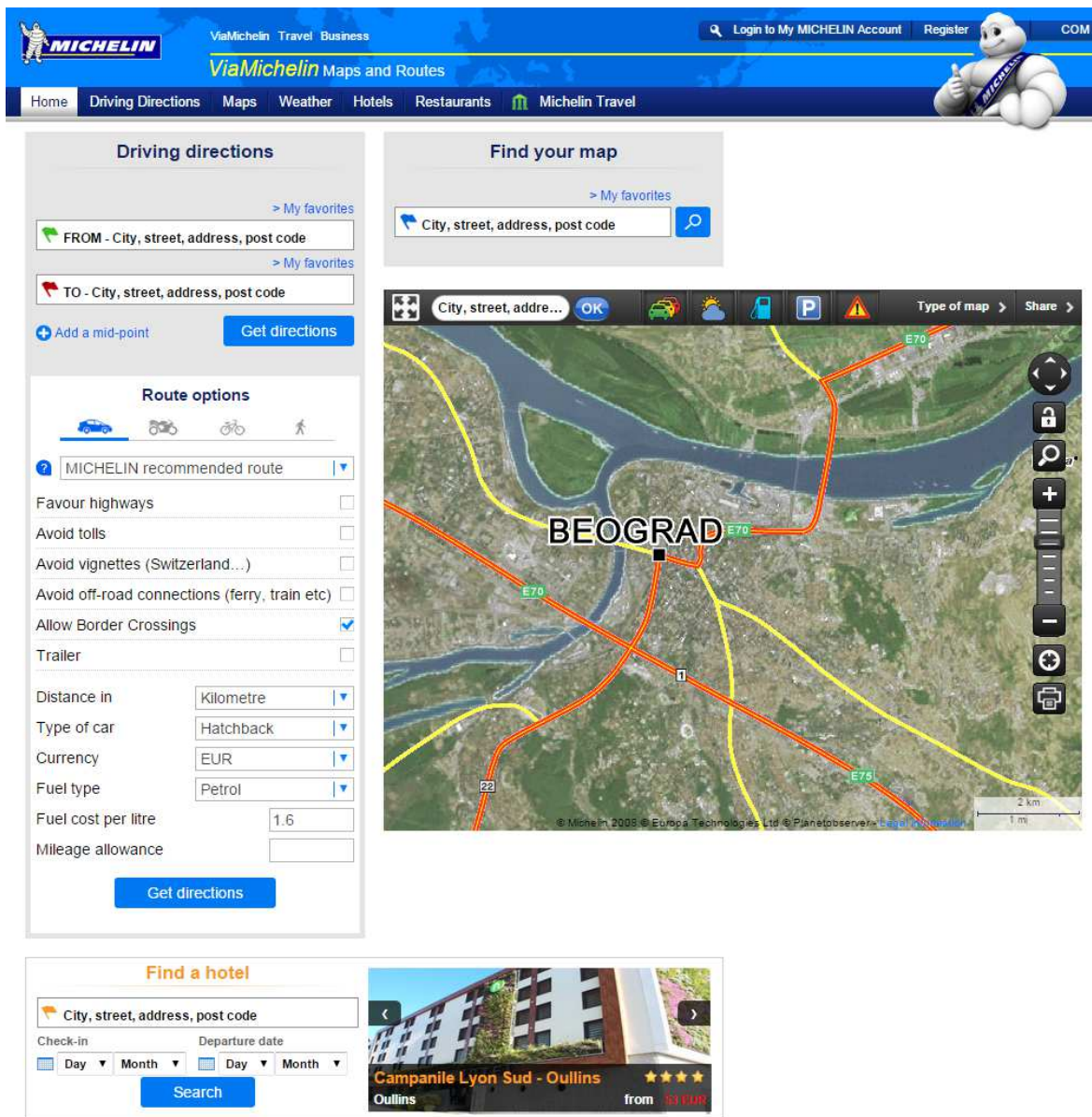
Slika 12: Here

#### 4.9.2.4 ViaMichelin

Kompanija *Michelin* pored uspešne prodaje pneumatika za putnička i teretna vozila, razvila je i portal na kome je moguće dobiti sve informacije na jednom mestu kada je putovanje u pitanju.

*ViaMichelin* je *on-line* usluga koja služi za planiranje putovanja. Delovi usluge su planiranje rute, rezervacija hotela, popisi restorana i turističkih atrakcija i informacije o stanju u saobraćaju i vremenska prognoza u stvarnom vremenu. Nakon planiranja puta, plan je moguće preuzeti i snimiti u GPS uređaj, nakon čega je potrebno samo njegovo aktiviranje i praćenje unapred pripremljenog plana puta [27].

*ViaMichelin* sajt pokriva preko sedam miliona kilometara puteva i ulica u preko 42 evropskih zemalja.



Slika 13: ViaMichelin

#### 4.9.3 Prednosti i razlike razvijenih lokacijskih servisa

Sva tri, gore navedena lokacijska servisa, imaju svoje prednosti i mane. Razlike su vidljive već prilikom otvaranja sajtova.

Kod *Google Maps*-a, sve opcije koje se koriste za prilagođavanje izgleda mape se nalaze u donjem desnom uglu, kod *Bing Maps*-a u gornjem delu mape sa leve i desne strane, dok kod *Here* mapa se jedan deo podešavanja nalazi u gornjem desnom delu mape drugi deo u donjem delu u sredini.

*Google Maps* ima najbolju dokumentaciju za korišćenje API-ja, pomoću kojeg korisnici mogu da integrišu mape u svoje aplikacije.

*Bing Maps* imaju prednost prilikom pretrage adresa i prikazivanja plana puta, kao i slike veće rezolucije. Prednost *Bing Maps*-a je „bird's eye“ opcija, *Google* nudi opciju pregleda slika u krugu od 360 stepeni, ta opcija se zove *Google Earth* koja prikazuje mapu u satelitskom obliku sa 3D zgradama koje su korisnici sami kreirali. *Bing*-ova opcija „bird's eye“ sa druge strane je dostupna zahvaljujući slikama iz aviona koje je sakupio Microsoft-ov ambiciozni *Global Ortho* projekat [28].

U slučaju *Bing Maps*-a neuspeh dolazi u obliku neadekvatne mobilne aplikacije, verzija najkorisnija za ljude kojima su potrebne mape. U ovom trenutku, *Bing Maps* ne nude aplikaciju za *Android* i *iOS* operativne sisteme, tj. za pametne telefone i tablete koje danas koristi najveći broj ljudi, dok je aplikacija dostupna samo za *Windows Phone 8* korisnike [28].

Od sve tri prikazane usluge, *Google* nudi kvalitetniju aplikaciju za mape. Aplikacija je laka za upotrebu i sadrži više korisnih i bolje razvijenih funkcija. Opcija „*turn-by-turn*“ radi brzo i efikasno i sposobna je da proračuna novu rutu ukoliko se promaši planirano skretanje. *Bing Maps* usluga nema toliko razvijene funkcionalnosti.

## 5 Metode za određivanje distanci između dve tačke u prostoru

Preciznost geolokacije predstavlja daljinu izmerene lokacije od stvarne. Najčešći način da se koristi preciznost u GIS se izražava iskazivanjem maksimalnog odstupanja, tj. Maksimalne apsolutne greške. Na primer: lokacija tačke je u krugu od 20 metara, tj. preciznosti je 20 metara.

Jedan od glavnih faktora preciznosti je način na koji se lokacija meri. IP adresa je manje precizna od *Cell* ID-a, koji je manje precizan od GPS-a. Najuočljiviji razlog za navedene razlike je taj da se za lokaciju pomoću IP adrese koristi lokacija rutera, AP-a ili *firewall*-a koji može biti stotinama metara udaljen od lokacije sa koje je poslat zahtev za geolociranje. Ovo može biti scenario u velikim korporativnim sredinama. *Cell* ID je precizniji od IP adrese, jer se geolokacija meri iz triangulacije mrežnih tornjeva koji se nalaze u blizini korisnika. GPS je precizniji od *Cell* ID-a, jer se geolokacija određuje na osnovu kompleksnijih kalkulacija upoređivanjem svojstava i sadržaja signala više satelita.

Pored navedenih postoje i hardverski problemi, među kojima su: radio smetnje, vremenski uslovi i dr. koji utiču na jačinu signala i smanjuju preciznost prilikom geolociranja. Preciznost informacija o geolokaciji će se povećavati sa napretkom tehnologije.

Postoji mnogo načina za izračunavanje rastojanja između dve tačke na Zemljinoj površini, opisanim geografskom širinom i dužinom. Metode se razlikuju po složenosti i tačnosti, kao i prema tome da li se u zavisnosti od udaljenosti, može zanemariti zakrivljenost Zemlje.

### 5.1 Vinsentijeva formula

Vinsentijeve formule su dve uvezane iterativne metode koje se koriste u geodeziji da bi se izračunale razdaljine između dve tačke na površini sferoida, koje je razvio Tadeus Vinsenti 1975. godine [29]. One su bazirane na pretpostavci da je oblik Zemlje sferoid spljošten na polovima, pa je ovaj metod precizniji za izračunavanje od metoda kao što su velika kružnica, koja podrazumeva da je Zemlja loptastog oblika.

Formula je precizna u okviru od 0,5 mm (odnosno 0,000015"). Kalkulacije koje se baziraju na sferičnom modelu, kao što je dosta jednostavniji haversin model, su precizne oko 0,3% (što je dovoljno precizno za široku upotrebu) [30].

Prvi (direktni) metod izračunava lokaciju tačke koja ima svoju daljinu i pravac od druge tačke. Drugi (inverzivni) metod računa geografsku razdaljinu i pravac između dve tačke. Metode imaju široku upotrebu u geodeziji jer su precizne do 0,5 mm na Zemljinom elipsoidu [30].

**Tabela 4: Oznake u Vinsentijevoj formuli**

|                                               |                                                             |
|-----------------------------------------------|-------------------------------------------------------------|
| $a, b$                                        | velike i male poluose elipsoida                             |
| $f$                                           | spljoštenost elipsoida ( $f = \frac{a-b}{a}$ )              |
| $\phi$                                        | geografska širina tačaka                                    |
| $L$                                           | geografska dužina tačaka                                    |
| $s$                                           | elipsoidna razdaljina između dve tačke.                     |
| $\alpha_1, \alpha_2$                          | prednje azimute tačaka                                      |
| $z$                                           | azimut na ekvatoru                                          |
| $u^2 = \frac{\cos^2 \alpha (a^2 - b^2)}{b^2}$ |                                                             |
| $U$                                           | smanjena širina, definisana $\tan U = (1 - f) \tan \varphi$ |
| $\lambda$                                     | razlika u dužini na pomoćnoj sferi                          |
| $\sigma$                                      | ugaona razdaljina $P_1P_2$ na sferi                         |
| $\sigma_1$                                    | ugaona razdaljina na sferi od ekvatora do $P_1$             |
| $\sigma_m$                                    | ugaona razdaljina na sferi od ekvatora do sredine linije    |

**Izvor:** [29]

U skripti priloženoj u ovom radu korišćena je Vinsentijeva inverzna formula koja glasi [29]:

$$\sin \sigma = \sqrt{(\cos U_2 \sin \lambda)^2 + (\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda)^2} \quad (1)$$

$$\cos \sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos \lambda \quad (2)$$

$$\sigma = \arctan \frac{\sin \sigma}{\cos \sigma} \quad (3)$$

$$\sin \alpha = \frac{\cos U_1 \cos U_2 \sin \lambda}{\sin \sigma} \quad (4)$$

$$\cos^2 \alpha = 1 - \sin^2 \alpha \quad (5)$$

$$\cos(2\sigma_m) = \cos \sigma - \frac{2 \sin U_1 \sin U_2}{\cos^2 \alpha} \quad (6)$$

$$C = \frac{f}{16} \cos^2 \alpha [4 + f(4 - 3 \cos^2 \alpha)] \quad (7)$$

$$\lambda = L + (1 - C)f \sin \alpha \{ \sigma + C \sin \sigma [\cos(2\sigma_m) + C \cos \sigma (-1 + 2 \cos^2(2\sigma_m))] \} \quad (8)$$

Kada se  $\lambda$  konvergira na željeni nivo preciznosti ( $10^{-12}$  što odgovara približno 0,06 mm), procenjuje se:

$$u^2 = \cos^2 \alpha \frac{a^2 - b^2}{b^2} \quad (9)$$

$$A = 1 + \frac{u^2}{16348} \{ 4096 + u^2 [-768 + u^2 (320 - 175u^2)] \} \quad (10)$$

$$B = \frac{u^2}{1024} \{ 256 + u^2 [-128 + u^2 (74 - 47u^2)] \} \quad (11)$$

$$\Delta \sigma = B \sin \sigma \left\{ \cos(2\sigma_m) + \frac{1}{4} B \left[ \cos \sigma \left( -1 + 2 \cos^2(2\sigma_m) - \frac{1}{6} B \cos(2\sigma_m) (-3 + 4 \sin^2 \sigma) (-3 + 4 \cos^2(2\sigma_m)) \right) \right] \right\} \quad (12)$$

$$s = bA(\sigma - \Delta \sigma) \quad (13)$$

$$\alpha_1 = \arctan \left( \frac{\cos U_2 \sin \lambda}{\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos \lambda} \right) \quad (14)$$

$$\alpha_2 = \arctan \left( \frac{\cos U_1 \sin \lambda}{-\sin U_1 \cos U_2 + \cos U_1 \sin U_2 \cos \lambda} \right) \quad (15)$$

Između dve približno dijametralno različite tačke, iterativna formula možda ne konvergira, što se dešava kada je prvi pokušaj za izračunavanje  $\lambda$  gornjom formulom veći od apsolutne vrednosti  $\pi$ .

Najprecizniji i najčešće korišćen model Zemljinog elipsoida je WGS-84. Ostali elipsoidi koji se koriste su Airy 1830 u Velikoj Britaniji, Hajfordov međunarodni elipsoid 1924 u ostatku Evrope, Clarke 1880 u Africi, GRS-67 u Južnoj Americi i mnogi drugi. Američki NAD83 i australijski GDA koriste GRS-80 elipsoid, koji je funkcionalno ekvivalentan WGS-84 elipsoidu.

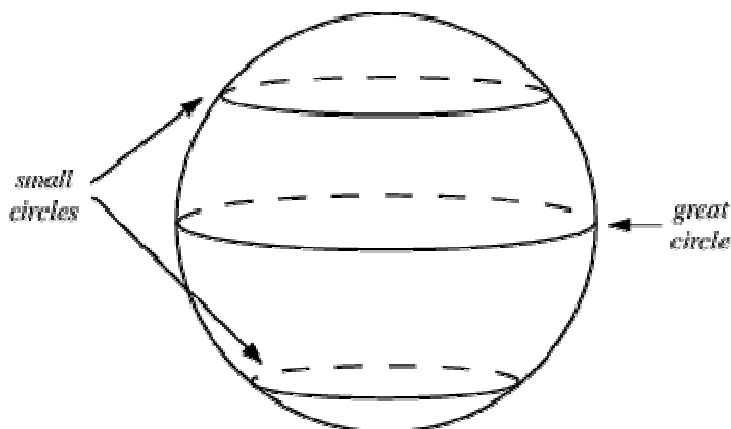
**Tabela 5: Tačke geografskih širina i dužina na različitim tipovima elipsoida**

| Ellipsoid          | a                        | b                              | f                           |
|--------------------|--------------------------|--------------------------------|-----------------------------|
| WGS-84             | 6 378 137 m ( $\pm 2$ m) | $\approx 6\,356\,752.314245$ m | $\approx 1 / 298.257223563$ |
| GRS-80             | 6 378 137 m              | $\approx 6\,356\,752.314140$ m | $1 / 298.257222101$         |
| Airy 1830          | 6 377 563.396 m          | 6 356 256.910 m                | $\approx 1 / 299.3249646$   |
| International 1924 | 6 378 388 m              | $\approx 6\,356\,911.946$ m    | $1 / 297$                   |
| Clarke 1880        | 6 378 249.145 m          | $\approx 6\,356\,514.86955$ m  | $1 / 293.465$               |
| GRS-67             | 6 378 160 m              | $\approx 6\,356\,774.719$ m    | $1 / 298.247167$            |

Izvor: [30]

## 5.2 Formula velikog kruga

Rastojanje velikog kruga (Ortodromsko rastojanje) je najkraće rastojanje između bilo koje dve tačke na površini sfere, izmereno putanjom površine sfere (umesto kroz sferu) [31]. S obzirom da je sferična geometrija drugačija od standardne Euklidove geometrije, jednačine za rastojanje imaju drugačiju formu. Rastojanje između dve tačke u Euklidovom prostoru je dužina prave linije od jedne do druge tačke. Kod sferičnih oblika, s druge strane, nema pravih linija, već su zamenjene geodezijskim linijama. Geodezijske linije na sferi predstavljaju velike krugove (krugovi na sferi su oni krugovi čiji se centar podudara sa centrom sfere).



**Slika 14: Delovi sfere**

Izvor: [31]

Pošto bilo koje dve tačke na sferi obrazuju kružnicu, spajanjem onih koje nisu direktno naspram jedna druge dobija se jedinstvena kružnica, koja ima dva luka. Dužina kraćeg luka predstavlja rastojanje velikog kruga između te dve tačke. Spajanjem onih koje su direktno jedna naspram druge dobija se neograničen broj velikih krugova, ali je svaki polukrug isti, tj. polovina obima kružnice  $\pi r$ , gde je  $r$  poluprečnik kružnice.

S obzirom da je Zemlja skoro sferična, jednačine za rastojanje velikog kruga se koriste da se grubo izračunaju najkraća rastojanja između bilo koje dve tačke na Zemlji, pa stoga imaju primenu u navigaciji.

Razdaljina se izračunava [31]:

$$r_i = \begin{bmatrix} \cos \lambda_i \cos \delta_i \\ \sin \lambda_i \cos \delta_i \\ \sin \delta_i \end{bmatrix} \quad (16)$$

Tako da je razdaljina na velikom krugu:

$$d = a \cos^{-1} [\cos \delta_1 \cos \delta_2 \cos(\lambda_1 - \lambda_2) + \sin \delta_1 \sin \delta_2] \quad (17)$$

### 5.3 Haversin formula

Haversin formula je jednačina vrlo važna u navigaciji, koja daje rastojanje između dve tačke na sferi na osnovu geografske dužine i geografske širine [32].

Ona glasi [32]:

$$d = 2r \arcsin \left( \sqrt{\text{haversin}(\phi_2 - \phi_1) + \cos(\phi_1) \cos(\phi_2) \text{haversin}(\lambda_2 - \lambda_1)} \right) =$$

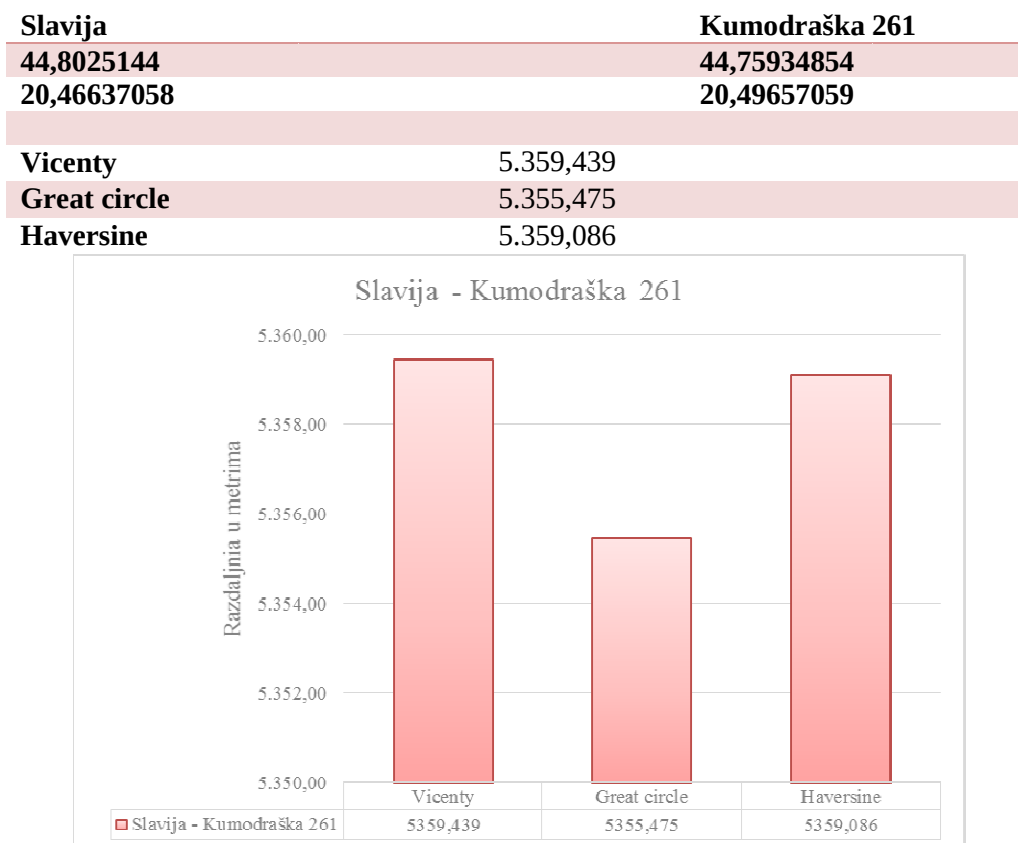
$$2r \arcsin \left( \sqrt{\sin^2 \left( \frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left( \frac{\lambda_2 - \lambda_1}{2} \right)} \right) \quad (18)$$

Gde je  $d$  udaljenost između dve tačke koja je izračunava, a  $r$  prečnik Zemlje (6.371 km).

## 6 Predloženo rešenje i eksperimentalni rezultati

### 6.1 Poređenje tačnosti distanci

**Primer 1:** Udaljenost u metrima između Slavije i zgrade Univerziteta Singidunum u Kumodraškoj 261

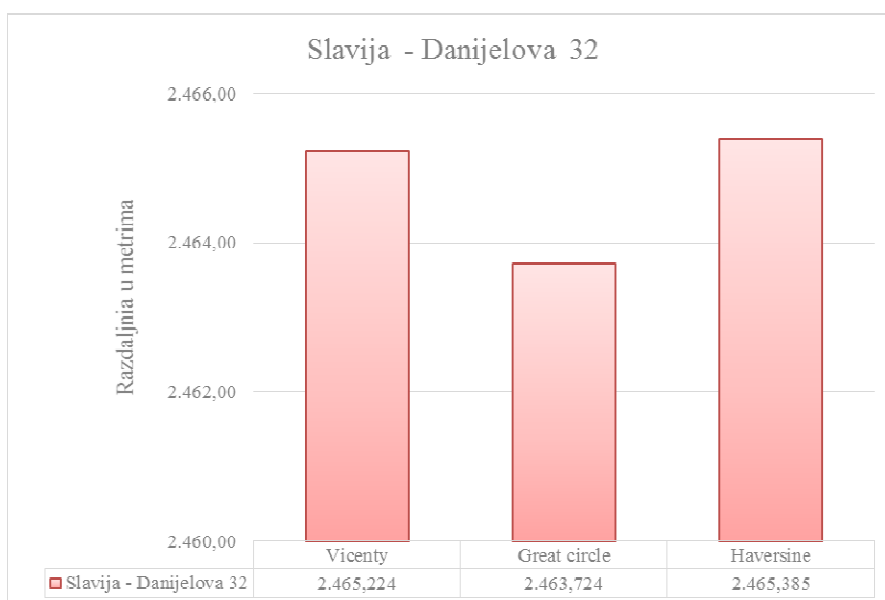


**Primer 2:** Udaljenost u metrima između Slavije i zgrade Univerziteta Singidunum u Danijelovoj 32

| Slavija     | Danijelova 32 |
|-------------|---------------|
| 44,8025144  | 44,78217773   |
| 20,46637058 | 20,47881603   |

|              |           |
|--------------|-----------|
| Vicenty      | 2.465,224 |
| Great circle | 2.463,724 |
| Haversine    | 2.465,385 |



**Primer 3:** Udaljenost u metrima između Slavije i zgrade Univerziteta Singidunum u Dušana Popovića 7

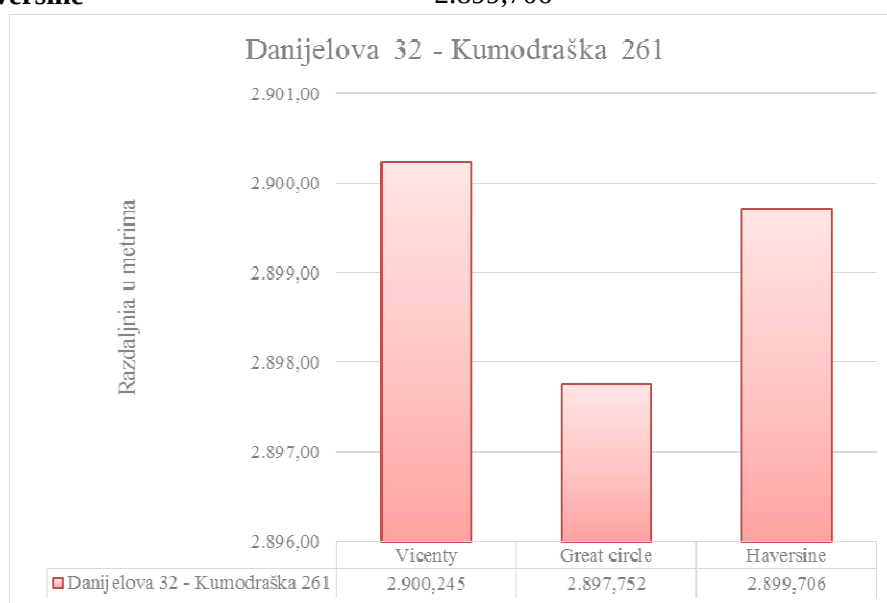
|                     |                          |
|---------------------|--------------------------|
| <b>Slavija</b>      | <b>Dušana Popovića 7</b> |
| <b>44,8025144</b>   | <b>44,79060716</b>       |
| <b>20,46637058</b>  | <b>20,51792264</b>       |
|                     |                          |
| <b>Vicenty</b>      | <b>4.288,33</b>          |
| <b>Great circle</b> | <b>4.274,909</b>         |
| <b>Haversine</b>    | <b>4.277,791</b>         |



**Primer 4:** Udaljenost u metrima između zgrada Univerziteta Singidunum u Danijelovoj 32 i Kumodraškoj 261

|                      |                       |
|----------------------|-----------------------|
| <b>Danijelova 32</b> | <b>Kumodraška 261</b> |
| <b>44,78217773</b>   | <b>44,75934854</b>    |
| <b>20,47881603</b>   | <b>20,49657059</b>    |
|                      |                       |

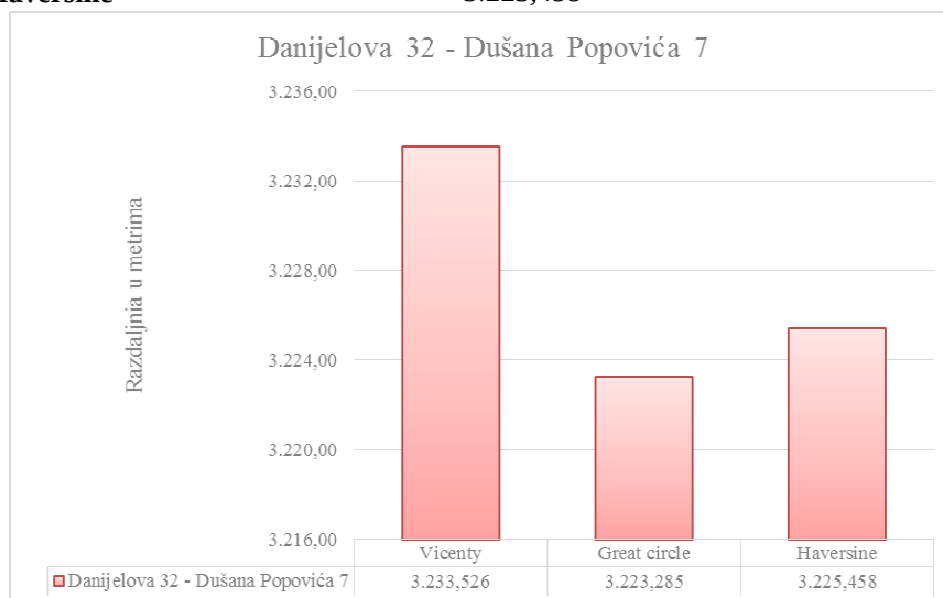
|                     |           |
|---------------------|-----------|
| <b>Vicenty</b>      | 2.900,245 |
| <b>Great circle</b> | 2.897,752 |
| <b>Haversine</b>    | 2.899,706 |



**Primer 5:** Udaljenost u metrima između zgrada Univerziteta Singidunum u Danijelovoj 32 i Dušana Popovića 7

|                      |                          |
|----------------------|--------------------------|
| <b>Danijelova 32</b> | <b>Dušana Popovića 7</b> |
| 44,78217773          | 44,79060716              |
| 20,47881603          | 20,51792264              |

|                     |           |
|---------------------|-----------|
| <b>Vicenty</b>      | 3.233,526 |
| <b>Great circle</b> | 3.223,285 |
| <b>Haversine</b>    | 3.225,458 |



**Primer 6:** Udaljenost u metrima između zgrada Univerziteta Singidunum u Dušana Popovića 7 i Kumodraškoj 261

|                          |                       |
|--------------------------|-----------------------|
| <b>Dušana Popovića 7</b> | <b>Kumodraška 261</b> |
| 44,79060716              | 44,75934854           |

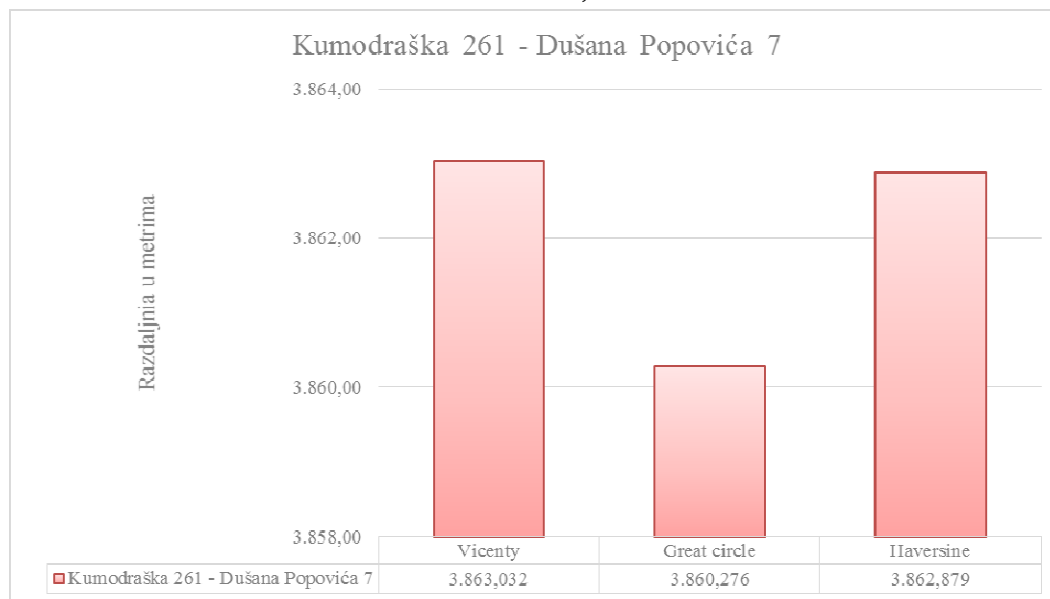
20,51792264

20,49657059

Vicenty 3.863,032

Great circle 3.860,276

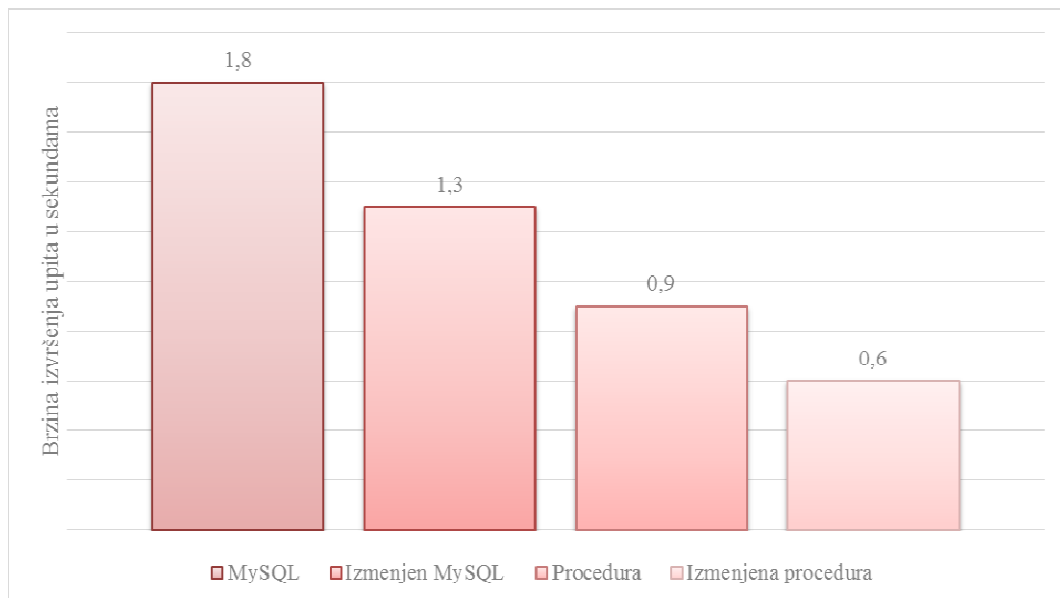
Haversine 3.862,879



## 6.2 Poređenje brzine izvršavanja SQL upita u MySQL DBMS-u

Na osnovu prethodnog merenja odabrana je Haversinova formula jer je pružila dovoljnu tačnost za potrebe rada.

Na osnovu odabrane formule kreiran je SQL upit u MySQL DBMS-u koji se nije dovoljno brzo izvršavao, nakon čega je prvobitni upit modifikovan. Time je izvršavanje upita ubrzano za oko pola sekunde, što nije predstavljalo dovoljno poboljšanje performansi. Prvobitni upit je uvršten u proceduru, gde je izvršavanje ubrzano za još dodatne 0,4 sekunde, da bi na kraju u proceduru uvrstili modifikovan upit kojim je postignuta brzina izvršavanja od 0,6 sekundi na tabeli od 5.000 zapisa.



Slika 15: Brzine izvršavanja upita

**Kod 1:** Prikazuje Haversinovu formulu u formi SQL upita koja je korišćena u radu

```
set @orig_lat = 44.8024516277;
set @orig_lon = 20.4662723541;
set @dist = 10;

SELECT stores.*,
6371 * 2 * ASIN(SQRT(POWER(SIN((@orig_lat - abs(stores.latitude)) *
pi()/180 / 2), 2) + COS(@orig_lat * pi()/180 ) * COS(abs(stores.latitude)
* pi()/180) * POWER(SIN((@orig_lon - stores.longitude) * pi()/180 / 2),
2) )) as distance
FROM stores
HAVING distance <= @dist
ORDER BY distance ASC;
```

**Kod 2:** Prikazuje izmenjen SQL upit

```
set @orig_lat = 44.8024516277;
set @orig_lon = 20.4662723541;
set @dist = 10;

set @lon1 = @orig_lon-@dist/abs(cos(radians(@orig_lat))*111.22176);
set @lon2 = @orig_lon+@dist/abs(cos(radians(@orig_lat))*111.22176);

set @lat1 = @orig_lat-(@dist/111.22176);
set @lat2 = @orig_lat+(@dist/111.22176);

SELECT stores.*,
6371 * 2 * ASIN(SQRT(POWER(SIN((@orig_lat - abs(stores.latitude)) *
pi()/180 / 2), 2) + COS(@orig_lat * pi()/180 ) * COS(abs(stores.latitude)
* pi()/180) * POWER(SIN((@orig_lon - stores.longitude) * pi()/180 / 2),
2) )) as distance
FROM stores
WHERE
stores.longitude BETWEEN @lon1 AND @lon2 AND
stores.latitude BETWEEN @lat1 AND @lat2
HAVING distance <= @dist
ORDER BY distance ASC;
```

**Kod 3:** Prikazuje SQL upit u formi procedure

```

CREATE PROCEDURE `distance`(IN `orig_lat` DOUBLE(12,10), IN `orig_lon`
DOUBLE(12,10), IN `dist` FLOAT(4))
BEGIN
  set @lon1 = orig_lon-dist/abs(cos(radians(orig_lat))*111.22176);
  set @lon2 = orig_lon+dist/abs(cos(radians(orig_lat))*111.22176);

  set @lat1 = orig_lat-(dist/111.22176);
  set @lat2 = orig_lat+(dist/111.22176);

  SELECT stores.*,
  6371 * 2 * ASIN(SQRT(POWER(SIN((orig_lat - stores.latitude) * pi()/180 /
2), 2) + COS(orig_lat * pi()/180 ) * COS(stores.latitude * pi()/180) *
POWER(SIN((orig_lon - stores.longitude) * pi()/180 / 2), 2) )) as
distance,
  store_logo(stores.id) AS store_logo,
  store_score(stores.id) AS score
FROM stores
HAVING distance <= dist
ORDER BY distance ASC;
END

```

**Kod 4:** Prikazuje izmenjenu proceduru

```

CREATE PROCEDURE `distance`(IN `orig_lat` DOUBLE(12,10), IN `orig_lon`
DOUBLE(12,10), IN `dist` FLOAT(4))
BEGIN
  set @lon1 = orig_lon-dist/abs(cos(radians(orig_lat))*111.22176);
  set @lon2 = orig_lon+dist/abs(cos(radians(orig_lat))*111.22176);

  set @lat1 = orig_lat-(dist/111.22176);
  set @lat2 = orig_lat+(dist/111.22176);

  SELECT stores.*,
  6371 * 2 * ASIN(SQRT(POWER(SIN((orig_lat - stores.latitude) * pi()/180 /
2), 2) + COS(orig_lat * pi()/180 ) * COS(stores.latitude * pi()/180) *
POWER(SIN((orig_lon - stores.longitude) * pi()/180 / 2), 2) )) as
distance,
  store_logo(stores.id) AS store_logo,
  store_score(stores.id) AS score
FROM stores
WHERE
stores.longitude BETWEEN @lon1 AND @lon2 AND
stores.latitude BETWEEN @lat1 AND @lat2
HAVING distance <= dist
ORDER BY distance ASC;
END

```

## 7 GIS portal i Android aplikacija

### 7.1 Google Maps API

Google Maps API pruža programerima nekoliko načina da ugrade Google mape u svoje aplikacije i omogućava jednostavnu upotrebu, kao i mogućnosti za prilagođavanje. Trenutno postoji nekoliko API usluga:

- Google Maps Javascript API
- Google Static Maps API
- Google Maps Embed API

U zavisnosti od potreba, moguće je koristiti jedan ili kombinaciju sva tri navedena API-ja istovremeno.

Većina aplikacija može da koriste API za Google mape besplatno, međutim ukoliko aplikacija konstantno stvara veliki broj poziva ka API servisu, usluga se naplaćuje. Google Maps API ima ograničenja, dnevno je moguće napraviti do 25.000 poziva ka API servisu. Ako se ovo ograničenje prekorači, Google zahteva naplatu prekoračenja korišćenja koju je moguće odobriti u Konzoli Google API-ja (Google API Console).

| Usluga                 | Dnevno ograničenje | 1.000 dodatnih poziva API-ja (u Američkim dolarima \$) |
|------------------------|--------------------|--------------------------------------------------------|
| JavaScript Maps API v3 | 25.000             | 0,50                                                   |
| Static Maps API        | 25.000             | 0,50                                                   |
| Street View Image API  | 25.000             | 0,50                                                   |

#### 7.1.1 Implementacija Google Maps API-ja u HTML5

Sve aplikacije u koje se ugrađuje Google Maps moraju da navedu API ključ prilikom uključivanja Google Maps API JavaScript koda. Pomoću ovog ključa Google prati korišćenje API-ja.

Da bi ključ bio izdat programeru, potrebno je na adresi <https://code.google.com/apis/console> izvršiti prijavu za određeni API servis pomoću Google naloga. Nakon prijave, potrebno je kreirati projekat i poslati zahtev za izdavanje ključa koji će biti korišćen u aplikaciji.

Moguće je kreiranje četiri vrste ključa:

- Ključ za serversku aplikaciju
- Ključ za aplikaciju koja se pokreće u pretraživaču
- Ključ za android aplikaciju
- Ključ za iOS aplikaciju

Projekat

Overview

Services

Team

API Access

Billing

Reports

Quotas

BigQuery

Google Cloud SQL

Google Cloud Storage

## API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting requests back to your project.

### Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 64 client IDs. [Learn more](#)

[Create an OAuth 2.0 client ID...](#)

### Simple API Access

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

| Key for browser apps (with referers) |                                        | <a href="#">Generate new key...</a>      |
|--------------------------------------|----------------------------------------|------------------------------------------|
| API key:                             | AIzaSyBwhffqoERlYuky5OMCqQDES4tkdibH0g | <a href="#">Edit allowed referers...</a> |
| Referers:                            | Any referer allowed                    | <a href="#">Delete key...</a>            |
| Activated on:                        | Nov 2, 2014 7:11 AM                    |                                          |
| Activated by:                        | @gmail.com – you                       |                                          |

| Key for Android apps (with certificates) |                                         | <a href="#">Generate new key...</a>          |
|------------------------------------------|-----------------------------------------|----------------------------------------------|
| API key:                                 | AIzaSyDnQ5QjrBpy1lHOaavwdiLPJeMirLmKYh0 | <a href="#">Edit allowed Android apps...</a> |
| Android apps:                            | Any app allowed                         | <a href="#">Delete key...</a>                |
| Activated on:                            | Oct 20, 2014 3:11 AM                    |                                              |
| Activated by:                            | @gmail.com – you                        |                                              |

[Create new Server key...](#) [Create new Browser key...](#) [Create new Android key...](#) [Create new iOS key...](#)

### Notification Endpoints

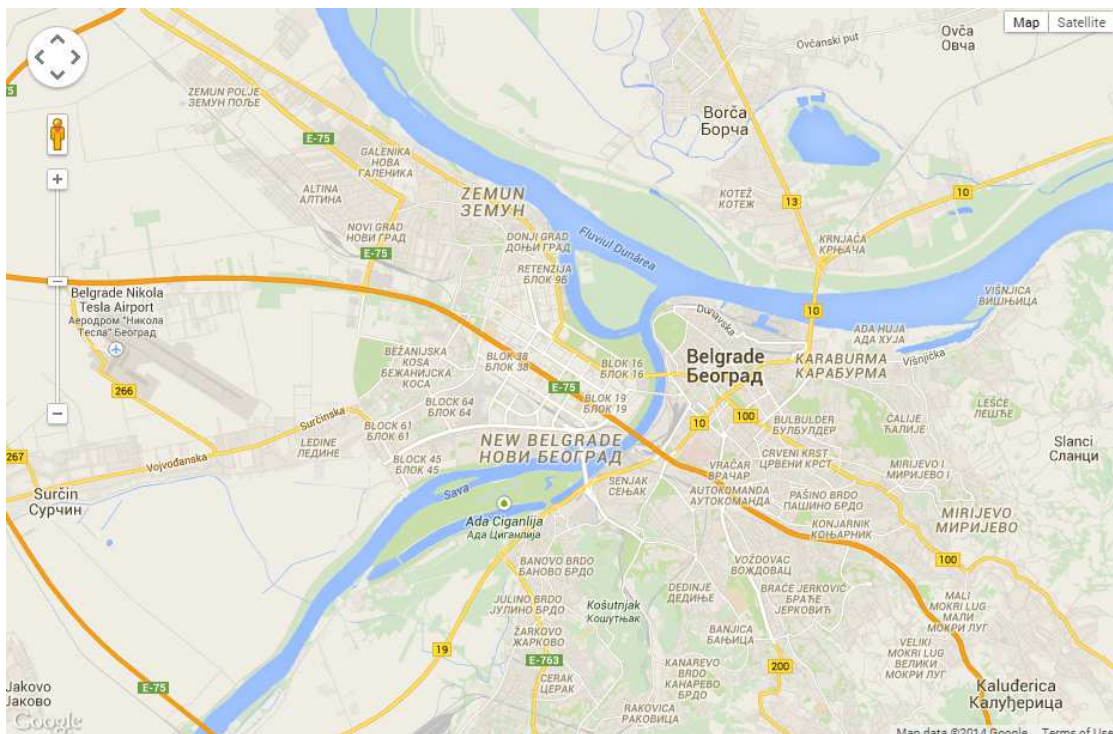
Use notification endpoints to identify domains that may receive webhook notifications from your API. [Learn more](#)

|                  |                    |                         |
|------------------|--------------------|-------------------------|
| Allowed Domains: | No domains allowed | <a href="#">Edit...</a> |
|------------------|--------------------|-------------------------|

Slika 16: Google konzola – prikaz prozora za kreiranje ključeva

## Kod 5: Implementacija API-ja u HTML5 stranicu

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="initial-scale=1.0, user-
scalable=no">
    <meta charset="utf-8">
    <title>Implementacija Google Maps API-ja u HTML5</title>
    <style>
      html, body, #map-canvas { height: 100%; margin: 0px;
padding: 0px }
    </style>
    <script
src="https://maps.googleapis.com/maps/api/js?key=API_KEY"></script>
    <script>
      function initialize() {
        var myLatLng = new google.maps.LatLng(44.782006,
20.478960);
        var mapOptions = {
          zoom: 12,
          center: myLatLng
        }
        var map = new
google.maps.Map(document.getElementById('map-canvas'), mapOptions);
      }
      google.maps.event.addDomListener(window, 'load',
initialize);
    </script>
  </head>
  <body>
    <div id="map-canvas"></div>
  </body>
</html>
```



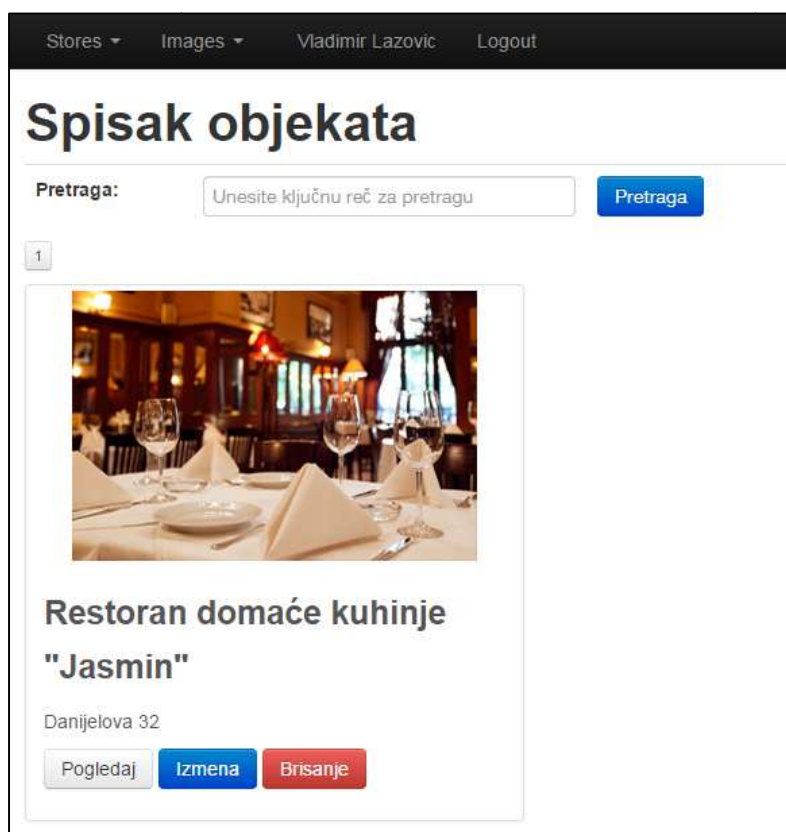
Slika 17: Primer prikaza mape generisane pomoću Google Maps API-ja

## 7.2 GIS portal

Informacioni sistem je izgrađen na platformi otvorenog koda, korišćenjem savremenih veb tehnologija. Osnova platforme čini veb hosting arhitektura bazirana na GNU/Linux operativnom sistemu čiji resursi se u konkretnoj implementaciji koriste kroz uslugu deljenog hosting paketa lociranog na serverima u data centru hosting kompanije u Nemačkoj, sa brzom Internet vezom sa centralnim nodovima u Srbiji.

Aplikativni nivo platforme čini softver pisan PHP skriptnim jezikom, koji se izvršava na strani servera i na tržištu veb aplikacija ima udeo od preko 40%. Osnovno svojstvo PHP skriptinog jezika je da može da se kombinuje sa HTML opisnim jezikom u istoj datoteci, ali je u konkretnoj implementaciji ovog informacionog sistema korišćeno proširenje PHP skripte za dodavanje mogućnosti šablonskog kreiranja sadržaja. Šablonski mehanizam koji je korišćen omogućava softverski paket po imenu *Smarty Templating Engine* za PHP.

*Smarty* je razvio svoju nezavisnu filozofiju pisanja tagova i predstavljanja funkcija koje se primenjuju nad sadržajem na mestu prikaza u šablonu. Ovaj sistem pisanja tagova se razlikuje od standardnog PHP modela kod kojeg se sadržaj prosleđuje funkcijama. U *Smarty* modelu postoje modifikatori koji se primenjuju na sadržaj i tako ga menjaju i prilagođavaju pre prikaza na adekvatnom mestu u šablonu.



Slika 18: Izgled spiska objekta na portalu

---

```

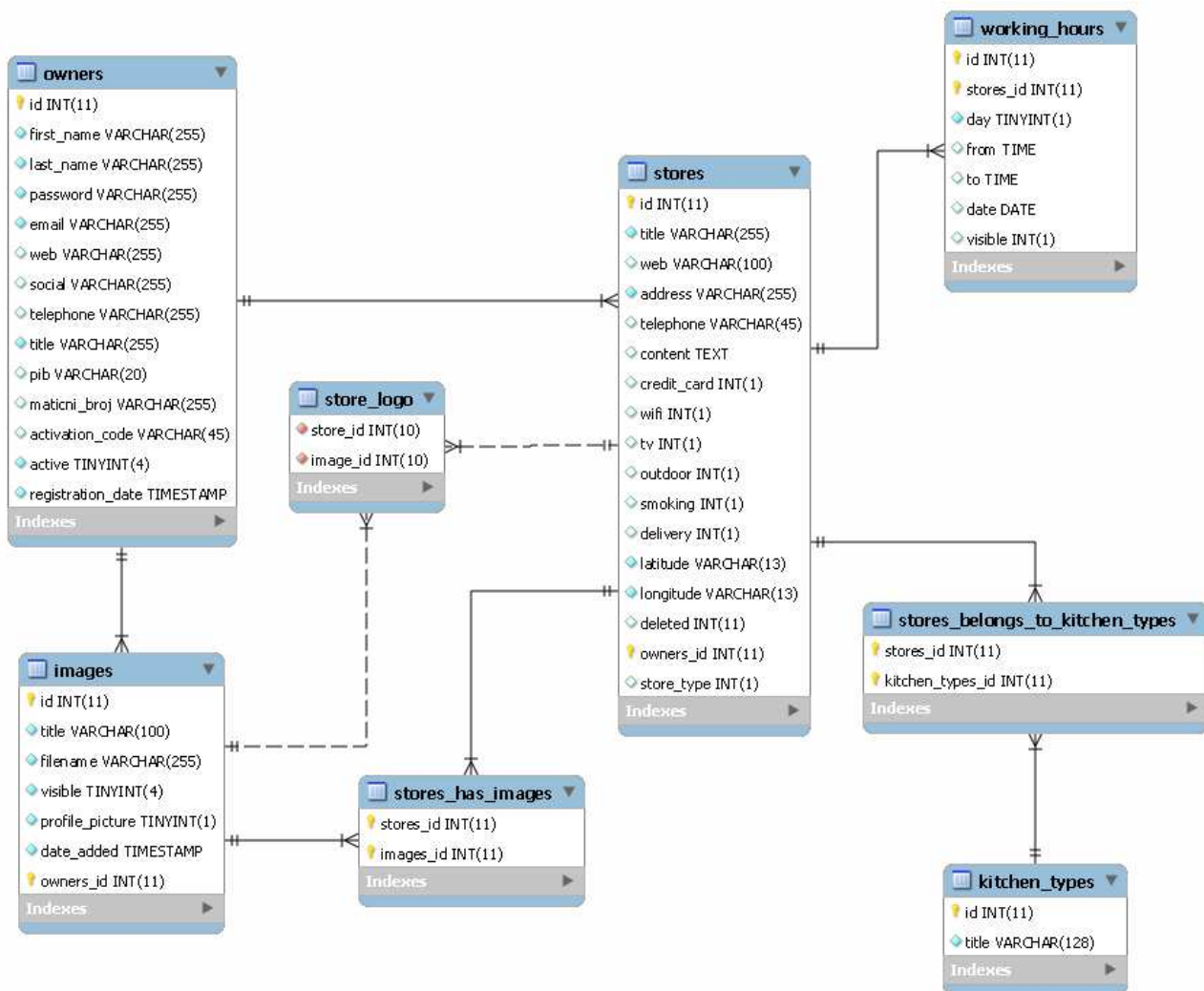
<stores>
  <store>
    <id>4529</id>
    <title>Restoran domaće kuhinje "Jasmin"</title>
    <web>http://singidunum.ac.rs</web>
    <address>Danijelova 32</address>
    <telephone>0113093069</telephone>
    <content>Restoran se nalazi u okviru Univerziteta Singidunum. Svojim korisnicima nudi raznovrsna jela i pica.</content>
    <latitude>44.7823515</latitude>
    <longitude>20.4787783</longitude>
    <credit_card>1</credit_card>
    <wifi>1</wifi>
    <tv>1</tv>
    <outdoor>0</outdoor>
    <smoking>0</smoking>
    <delivery>0</delivery>
    <score>0</score>
    <image_count>2</image_count>
    <comment_count>0</comment_count>
    <store_logo>2013/07/small/6208_135469182227_5502913_n.jpg</store_logo>
    <types>Sendvići, Paste, Pizza, Palačinke, Roštilj, Zdrava hrana, Napitci, Poslastice</types>
  </store>
</stores>

```

Slika 19: XML struktura podataka o objektu

### 7.2.1 Baza GIS Portala

Odvojeni deo informacionog sistema čini sistem za skladištenje i obradu podataka koji je, kao i ostatak informacionog sistema, softver otvorenog koda. Za informacioni sistem je odabran sistem za upravljanje bazama podataka sa relacionim modelom i to MySQL DBMS (*DataBase Management System*). MySQL je drugi sistem relacionih baza podataka otvorenog koda po zastupljenosti na Internetu. MySQL ima izuzetno široku podršku i dostupan je kroz konektore za sve bitnije programske jezike: C/C++, C#, Java, Python, Perl itd. Baza podataka je u konkretnoj implementaciji informacionog sistema instalirana na istom serveru na kojem je i aplikativni deo sistema, tj. PHP skripte, iako je moguće da se baza podataka nalazi na odvojenom serveru, posvećenom isključivo izvršavanju procesa baze podataka.



Slika 20: Dizajn baze podataka portala

### 7.2.2 Aplikacija GIS Portala

Najvažnija celina aplikativnog dela informacionog sistema je PHP aplikacija koja predstavlja spregu između baze podataka i mobilne aplikacije, koja je glavni interfejs između korisnika i informacija koje informacioni sistem pruža. Aplikacija je pisana za PHP verziju 5.5 i korišćena je objektno orijentisana programska paradigma.

### 7.2.3 Organizacija stabla i direktorijuma portala

Aplikacija je podeljena u dva odvojena stabla direktorijuma, od kojih je jedno smešteno u zaštićenom delu servera čiji sadržaj na pripada ni jednoj virtuelnoj host sekciji *Apache* veb servera, dok sadržaj drugog stabla pripada javno dostupnom direktorijumu veb sajta. Zaštićeni deo sadrži datoteke koje se direktno uključuju kroz PHP skripte iz javnog dela, ali nije moguće njihovo neposredno izvršavanje.

Među zaštićenim datotekama su:

- Konfiguraciona datoteka koja sadrži osnovne parametre sajta, kao što su osnovna putanja sajta i parametri za pristup bazi podataka;
- Datoteka sa često korišćenim funkcijama, koja sadrži definicije globalno dostupnih funkcija;

- Datoteka sa definicijom klase za rad sa bazom podataka;
- *Smarty Templating Engine* softverski paket sa datotekom za inicijalizaciju sistema za rad sa šablonima;
- Osnovna klasa modula veb sajta, kreirana za potrebe izrade izvedenih modula informacionog sistema.

Među javno dostupnim datoteka su:

- Indeks datoteka aplikacije koja se prva izvršava i obavlja poslove pokretanja, provere validnosti zahteva, obrade izlaznog bafera i prosleđivanja rezultata izvršavanja serverske skripte klijentu;
- Direktorijum sa osnovnim sredstvima za prilagođavanje izlaza aplikacije;
- Izvedene module informacionog sistema, organizovane na taj način da jedan modul ima svoju istoimenu datoteku, koje se sve nalaze u pod direktorijumu namenjenom za module;
- Šablonske datoteke za sve javne metode svih modula, organizovane na taj način da svaki javni metod modula ima svoju istoimenu datoteku u pod direktorijumu istog imena kao modul kojem pripada, a svi oni smešteni u direktorijum namenjen za šablone.

## 7.3 Aplikacija za pametne telefone

Aplikacija za pametne telefone razvijena je za telefone sa Android operativnim sistemom, komunikacija sa serverom se odvija preko API-ja.

API sadrži skup PHP funkcija koja preko MySQL baze podataka prosleđuje podatke u XML formatu.

### 7.3.1 Struktura klasa

Struktura klasa aplikacije je organizovana u tri osnovna direktorijuma prema funkciji klasa koje se u aplikaciji koriste kao apstraktne klase pristupanjem njihovim statičnim članovima podacima i metodima ili tako što se instanciraju objekti pojedinih i na taj način koriste.

Tri osnovna direktorijuma su:

- Direktorijum `/src/com/njamnjam/droid`
- Direktorijum `/src/com/njamnjam/misc`
- Direktorijum `/src/com/njamnjam/views`

Ovi direktorijumi direktno odgovaraju paketima u Java programskom jeziku u koje su smeštene klase iz pojedinačnih direktorijuma i to paketi:

- `com.njamnjam.droid`
- `com.njamnjam.misc`
- `com.njamnjam.views`

Spisak klasa sa nadovezanim pripadajućim paketom upisanim kao prefiks imena klase broji ukupno 10 klasa i to:

- `com.njamnjam.droid.GalleryActivity`

- com.njamnjam.droid.**LauncherActivity**
- com.njamnjam.droid.**MainActivity**
- com.njamnjam.droid.**PropertiesActivity**
- com.njamnjam.droid.**StoreInformationActivity**
- com.njamnjam.misc.**Misc**
- com.njamnjam.misc.**Model**
- com.njamnjam.misc.**Provider**
- com.njamnjam.misc.**Settings**
- com.njamnjam.views.**KitchenTypeMenuItemView**

U aplikaciji se koriste dve apstraktne klase i to:

- com.njamnjam.misc.**Misc**
- com.njamnjam.misc.**Settings**

### 7.3.2 Definicija pomoćnih klasa

Klasa **Misc** u sebi sadrži 6 statičnih metoda opšte namene i jedan *LocationListener* objekat koji se koristi za detektovanje promene GPS koordinata uređaja na kojem se aplikacija izvršava. Spisak statičnih metoda je:

- **public static boolean** isNetworkAvailable( *Context* context );  
Ovaj metod proverava da li je dostupna mrežna konekcija u trenutku izvršavanja metoda. Vraća **true** ako jeste ili **false** ako nije.
- **public static boolean** isServerAvailable( *Context* context );  
Ovaj metod proverava da li je dostupan server u trenutku izvršavanja metoda. Vraća **true** ako jeste ili **false** ako nije.
- **public static void** initLocationManager( *Context* context );  
Ovaj metod inicijalizuje glavni statički član podatak tipa *LocationListener* u klasi *Misc*.
- **public static final String** md5( **final String** s );  
Ovaj metod izračunava MD5 heš vrednost prosleđenog argumenta i vraća rezultat u obliku niza karaktera koji predstavlja heš vrednost.
- **public static Drawable** getImage( *String* url );  
Ovaj metod preuzima sa Interneta sa prosledene URL adrese grafički resurs u obliku *Android Drawable* tipa.
- **public static Drawable** getStaticMap( *double* lat, *double* lon );  
Ovaj metod generiše statičnu sliku *Google Maps* prikaza lokacije na koordinatama prosleđenim kroz argumente u pozivu metoda.

Klasa **Settings** u sebi sadrži četiri statična tekstualna člana podatka koji predstavljaju konfiguraciju putanja ka Internet resursima na serveru sa koje aplikacije preuzima. Članovi podaci sa podrazumevanim vrednostima za veb server su:

- **public static String** domain = "http://domen aplikacije";
- **public static String** API\_path = Settings.domain + "xml/";
- **public static String** API\_serve\_check\_path = Settings.API\_path + "serverCheck";
- **public static String** Image\_path = Settings.domain + "images/";

Klasa **Provider** je osnovna klasa za preuzimanja podataka sa servera koja iz XML resursa dobijenog sa servera formira listu modela, tj. listu objekata klase **Model**. Osobina objekata klase **Provider** je da se prilikom instanciranja pomoću konstruktora navodi URL putanja ka API zahtevu za vraćanje XML resursa, kao i naziv glavnog noda sa strukturom vraćenog modela. Ostali metodi klase **Provider** su:

- **public Model get( int i );**  
Metod vraća model na i-toj poziciji u listi.
- **public int size();**  
Metod vraća ukupan broj vraćenih stavki u XML rezultatu.
- **public String get( int i, String key );**  
Metod vraća tekstualnu reprezentaciju vrednosti podatka iz XML noda na i-toj poziciji u listi, sa nazivom podatka definisanim ključem *key*.

Klasa **Model** je osnovna klasa za ekstrakciju vrednosti iz XML noda dobijenog pomoću objekata klase **Provider**. Konstruktor objekata klase **Model** zahteva kao argument XML Element noda za koji treba da preuzme vrednosti. Pored konstruktora postoji još jedan metod:

- **public String get( String key );**  
Metod vraća tekstualnu reprezentaciju vrednosti podatka iz XML noda, sa nazivom podatka definisanim ključem *key*.

Klasa **KitchenTypeMenuItemView** je klasa koja definiše funkciju grafičke jedinice za prikaz u listi tipova kuhinja u glavnom meniju. Klasa nasleđuje Android klasu *TextView*. Osnova funkcija posebne izvedene klase je da definiše stil prikaza i da uveže *OnClickListener* objekat koji će se okidati klikom na stavku u meniju.

### 7.3.3 Definicija klasa prikaza (Activity klase)

Klasa **GalleryActivity** je izvedena iz klase **Activity** i predstavlja deo android aplikacije zadužen za prikaz galerije slika za određenu radnju.

Osnovni metod klase aktivira asinhrono preuzimanje spiska slika iz galerije radnje čiji je identifikacioni broj prosleđen kroz Android *Bundle* paket zapakovan u *Intent* kojim je **Activity** otvoren. Nakon preuzimanja spiska slika, pokreće se asinhrono preuzimanje slika sa servera ukoliko već ne postoji preuzeta slika u kešu. Za svaku prikazanu sliku u galeriji je vezan *OnClickListener* koji otvara sliku preuzetu u većoj rezoluciji u podrazumevanom programu za pregled multimedijalnih sadržaja na uređaju na kojem se aplikacija pokreće.

Klasa **LauncherActivity** je izvedena iz klase **Activity** i predstavlja deo aplikacije koji se prvi pokreće. Ova klasa prikazuje *Splash Screen* sa logotipom veb sajta i posle definisanog vremenskog intervala pokreće intent koji prikazuje *MainActivity* prozor.

Klasa **MainActivity** je izvedena iz klase **Activity** i predstavlja deo aplikacije koji je zadužen za prikaz centralnog prozora aplikacije. Klasa *MainActivity* vrši asinhrono preuzimanje spiska vrsta kuhinja i generiše glavni menu sa objektima klase *KitchenTypeMenuItemView*. Druga funkcija prilikom pokretanja je da zahteva od API-ja na serveru da dostavi XML spisak radnji koje se nalaze u definisanom prečniku u okolini GPS koordinata preuzetih iz **Misc** klase. Nakon završetka asinhronog preuzimanja spiska radnji, aplikacija formira listu pogleda definisanim u *store.xml*

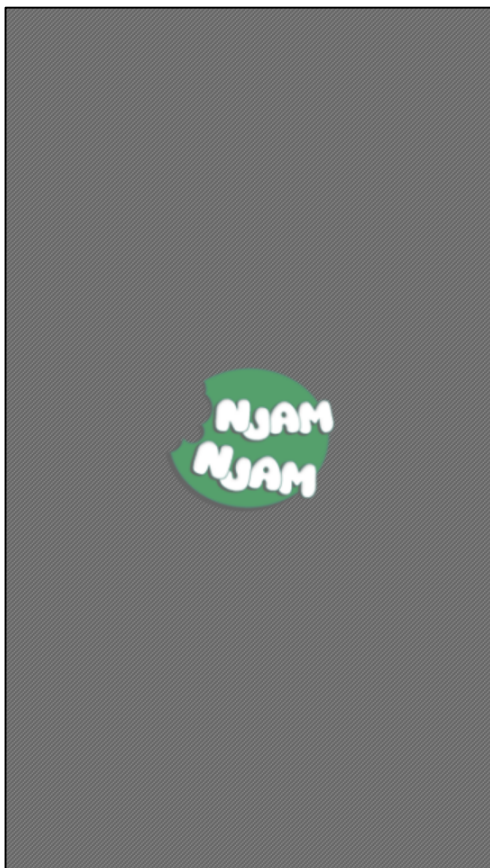
*layout* datoteci, čiji je osnovni okvir tipa *LinearLayout*. Klasa vezuje *OnClickListener* na svaku stavku koji nakon klika formira intent koji prikazuje prozor za prikaz detalja radnje. Takođe, klasa vezuje osluškivač događaja koji prikazuje prozor podešavanja aplikacije.

Klasa **PropertiesActivity** je izvedena iz klase **Activity** i predstavlja deo aplikacije koji je zadužen za prikaz prozora podešavanja, gde korisnik može da izabere prečnik u čijem dometu želi da aplikacija nalazi radnje oko njegove trenutne GPS lokacije.

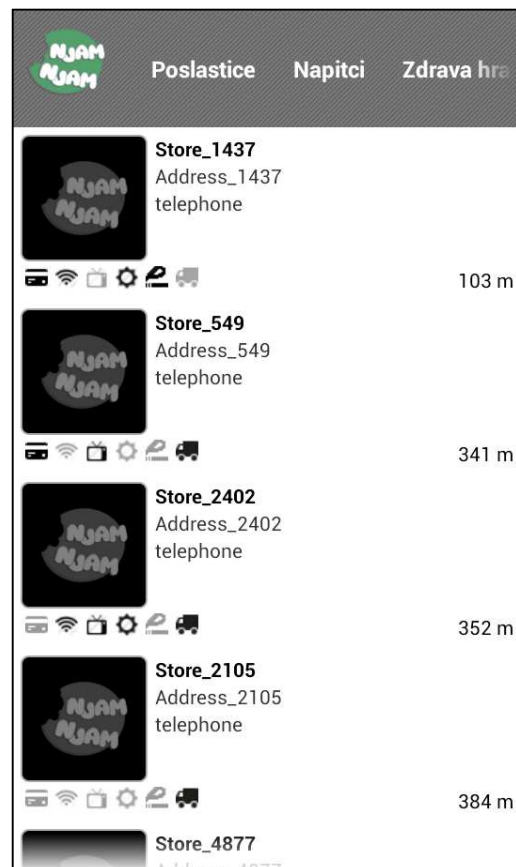
Klasa **StoreInformationActivity** je izvedena iz klase **Activity** i predstavlja drugi po važnosti deo aplikacije koji je zadužen za prikaz detalja o radnji u grafički uređenom prozoru definisanim XML *layout* datotekom *activity\_store\_information.xml*. Struktura ovog *layout*-a je izgrađena unutar Android *RelativeLayout* okvira.

Po pokretanju prozora koji definiše ova klasa, asinhrono se preuzimaju podaci o radnji, kao i tabela radnog vremena, uslovljena parametrom aktivnog datuma, preuzetog sa servera. Ovako prilagođena tabela se ispisuje ispod podataka o radnji. Prikaz pojedinačnih aktivnih elemenata, kao što je dugme za otvaranje galerije slika prikazuju se samo ako XML resurs vrati informaciju da postoji galerija slika za prikaz.

Ako postoje slike u galeriji slika za prikazanu radnju, vezuje se *OnClickListener* na dugme za prikaz galerije, tako da se nakon klika, pokreće intent koji prikazuje prozor za prikaz galerije.



Slika 21: Izgled ekrana prilikom učitavanja aplikacije



Slika 22: Početni izgled aplikacije

**Restoran domaće kuhinje "Jasmin"**

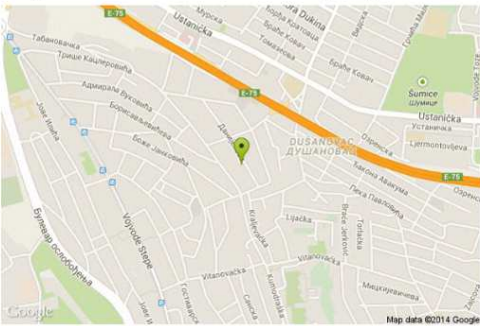


Danijelova 32  
011 3093069  
<http://singidunum.ac.rs>



Sendviči, Paste, Pizza, Palačinke, Roštilj, Zdrava hrana, Napitci, Poslastice

Restoran se nalazi u okviru Univerziteta Singidunum. Svojim korisnicima nudi raznovrsna jela i pica.



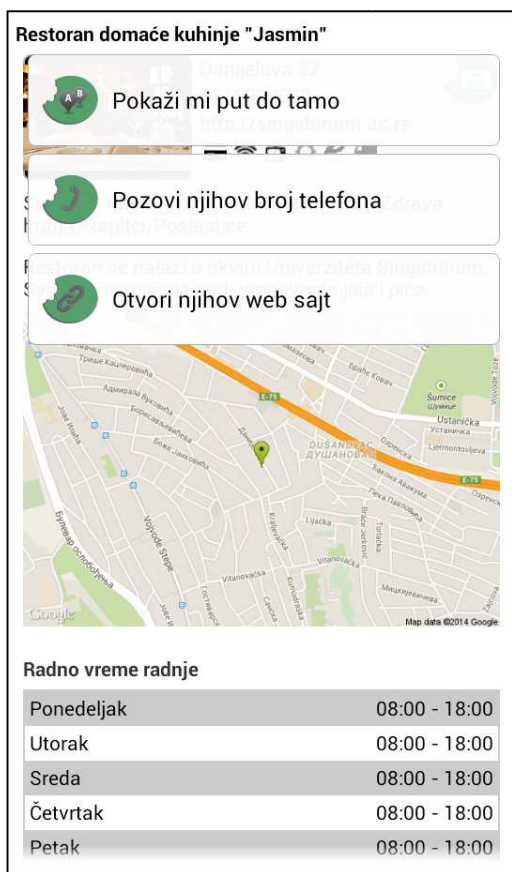
**Radno vreme radnje**

|             |                      |
|-------------|----------------------|
| Ponedjeljak | 08:00 - 18:00        |
| Utorak      | 08:00 - 18:00        |
| Sreda       | 08:00 - 18:00        |
| Četvrtak    | 08:00 - 18:00        |
| Petak       | 08:00 - 18:00        |
| Subota      | 09:00 - 14:00        |
| 01.11.2014. | Subota 09:00 - 14:00 |
| 15.11.2014. | Subota 09:00 - 14:00 |
| 29.11.2014. | Subota 09:00 - 14:00 |
| 12.12.2014. | Subota 09:00 - 14:00 |

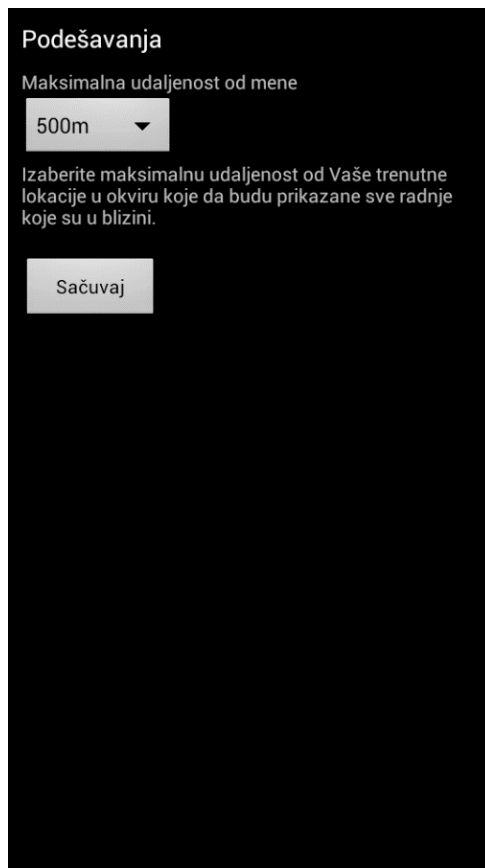
**Slika 23: Prikaz detalja radnje**



**Slika 24: Galerija slika odabrane radnje**



Slika 25: Izgled menija za navigaciju, pozivanje broja i otvaranje sajta odabrane radnje



Slika 26: Podešavanje aplikacije

## Zaključak

Razvoj pametnih telefona sa *Android* ili *iOS* operativnim sistemom pruža uređaj koji može da zameni više drugih. Telefoni su u velikoj meri sustigli računare kako procesorskom snagom tako i memorijom za skladištenje informacija.

Ljudi i objekti postaju sve mobilniji. Kako se tehnologija razvija, stvaraće se nove mogućnosti za ponudu personalizovanog, ciljanog sadržaja, koji je zasnovan na lokaciji.

Lokacijski bazirani servisi će sve više postajati deo svakodnevnog života, a prve usluge su već obezbeđene za korisnike mobilnih telefona, kao što su pretraga prijatelja po lokaciji, vremenske informacije ili dešavanja u okolini korisnika.

Lokacijski bazirani servisi transformišu način života ljudi širom planete. Ovi servisi mogu da uštede vreme, novac, pa čak i da spasu život. Međutim, zbog tehnologije koja ga omogućava, ovakav servis ima sposobnost da beleži aktivnosti korisnika, koje mogu da sadrže informacije lične prirode.

Geografske mape se u poslednje dve-tri decenije konstantno razvijaju i napreduju, i prešle su put od papirnih mapa do digitalnih, interaktivnih mapa na mobilnim telefonima sa bogatim sadržajima.

Koristeći *server-side* potencijale WebGIS-a, *Google Map-s* i *Android* operativnog sistema, dokazano je da je moguće kreirati lokacijski baziranu GIS aplikaciju za mobilne telefone koja će uz pomoć Haversinove formule moći da isporuči neophodne podatke, efikasno i za velikom preciznošću.

Kako bi ceo sistem bio operativan potreban je sistem za skladištenje podataka, koji će čuvati sve lokacijske podatke, kao i veb server koji će lokacijske podatke prosleđivati mobilnoj aplikaciji preko HTTP protokola u XML formatu. Za slanje podataka mobilnoj aplikaciji korišćen je aplikacioni programski interfejs (API) sa kojim se komunicira upitima i koji predstavlja interfejs između korisnika i baze podataka, a koji je zadužen da rezultate prosledi mobilnoj aplikaciji za dalju obradu i prezentaciju.

## Literatura

- [1] A. T. Holdener III, HTML5 Geolocation, O'Reilly Media, Inc., 2011.
- [2] S. Spiekermann, „General Aspects of Location-Based Services,“ u *Location-Based Services*, Elsevier, 2004, pp. 9-26.
- [3] V. Bellocci, S. Genovese, D. Inuaggiato / M. Tucci, Mobile Location-Aware Services: 2002 Market Perspective, Ericsson, Divison Service Architecture and Interactive Solutions, 2002.
- [4] „What location-based services (LBS) are?,“ 2011. [Na mreži]. Available: <http://www.onbile.com/info/what-lbs-are/>. [Poslednji pristup 23 mart 2014].
- [5] Business Insider, „INFOGRAPHIC: How Location Data Is Collected To Power The New Generation Of Mobile Marketing Campaigns,“ 2013. [Na mreži]. Available: <http://www.businessinsider.com/mobile-location-based-services-and-location-data-2013-8>. [Poslednji pristup 22 mart 2014].
- [6] Business Insider, „BII REPORT: How Location-Based Services Are Transforming The Mobile Industry,“ 2013. [Na mreži]. Available: <http://www.businessinsider.com/bii-report-how-location-based-data-is-transforming-the-entire-mobile-industry-2013-1>. [Poslednji pristup 22 mart 2014].
- [7] A. Buczkowski, „Location-Based Services – Applications,“ 2011. [Na mreži]. Available: <http://geoawesomeness.com/knowledge-base/location-based-services/location-based-services-applications/>. [Poslednji pristup 22 mart 2014].
- [8] N. Minić, A. Njeguš / J. Tulić Ceballos, „The impact of Web 3.0 technologies on tourism information systems,“ u *Singidunum University International Scientific Conference Sinteza 2014*, Beograd, 2014.
- [9] Y.-F. Nie, X. Fu / J.-X. Zeng, „A Tourist Route Planning and Navigation System Based on LBS,“ u *Proceedings of the 2009 International Symposium on Web Information Systems and Applications (WISA'09)*, Nanchang, P. R. China, 2009.
- [10] European Council, „Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications),“ *Official Journal L 201*, pp. 0037-0047, 2002.
- [11] W3C, „Introduction to HTML 4,“ [Na mreži]. Available: <http://www.w3.org/TR/html4/intro/intro.html>. [Poslednji pristup 15 februar 2014].
- [12] W3Schools, „HTML5 Introduction,“ 2013. [Na mreži]. Available: [http://www.w3schools.com/html/html5\\_intro.asp](http://www.w3schools.com/html/html5_intro.asp). [Poslednji pristup 15 februar 2014].
- [13] A. Marreiros, „HTML 5 Basic,“ 2013. [Na mreži]. Available: <http://digitalmindignition.wordpress.com/2013/02/03/html-5-basic/>. [Poslednji pristup 15 februar 2014].
- [14] N. Minić, Savremene informacione tehnologije u oblasti turizma, master rad, Beograd:

Univerzitet Singidunum, 2009.

- [15] Oracle, „What is MySQL?“, 1997, 2014. [Na mreži]. Available: <http://dev.mysql.com/doc/refman/5.1/en/what-is-mysql.html>. [Poslednji pristup 15 februar 2014].
- [16] D. Marković, „Šta je PHP“, 2009. [Na mreži]. Available: <http://www.draganmarkovic.net/weblog.php?file=sta-je-php.php>. [Poslednji pristup 10 april 2014].
- [17] A. Bjelošević, „Android OS“, [Na mreži]. Available: <http://aleksandar.bjelosevic.info/android-os/#.VDuPLfmSzg0>. [Poslednji pristup 10 april 2014].
- [18] Android, „Android - 4.4 KitKat“, [Na mreži]. Available: <http://www.android.com/versions/kitkat-4-4/>. [Poslednji pristup 10 april 2014].
- [19] MapSoft d.o.o, „GIS platforma“, [Na mreži]. Available: <http://www.mapsoft.rs/index.php/sr/gis-sofтвер/photosoft/gis-platforma>. [Poslednji pristup 12 oktobar 2014].
- [20] „The Components of GIS“, [Na mreži]. Available: <http://www.essex-countynj.org/GIS.pdf>. [Poslednji pristup 12 oktobar 2014].
- [21] V. Jovanović, B. Đurđev, Z. Srdić and U. Stankov, Geografski informacioni sistemi, Beograd, Novi Sad: Univerzitet Singidunum i Univerzitet u Novom Sadu, 2012.
- [22] J. Marshall, „Developing Internet-Based GIS Applications“, *GIS India*, br. 11, pp. 16-19, 2002.
- [23] S. Gibilisco, „What is Google Maps?“, 2013. [Na mreži]. Available: <http://whatis.techtarget.com/definition/Google-Maps>. [Poslednji pristup 12 oktobar 2014].
- [24] Microsoft, „Microsoft® Bing™ Maps Platform APIs' Terms Of Use“, 2014. [Na mreži]. Available: <http://www.microsoft.com/maps/product/terms.html>. [Poslednji pristup 12 oktobar 2014].
- [25] G. Spector / L. A. Varney, „C3 Technologies Now Part of Nokia Ovi Maps“, 2011. [Na mreži]. Available: <http://www10.giscale.com/nbc/articles/1/947086/C3-Technologies-Now-Part-Nokia-Ovi-Maps>. [Poslednji pristup 12 oktobar 2014].
- [26] P. Bonetti, „Find your way to HERE.com“, 2013. [Na mreži]. Available: <http://360.here.com/2012/11/23/find-your-way-to-here-com/>. [Poslednji pristup 12 oktobar 2014].
- [27] Michelin, „About ViaMichelin.com“, [Na mreži]. Available: <http://www.viamichelin.com/tpl/psg/divers/htm/AproposdeViaMichelin.htm>. [Poslednji pristup 12 oktobar 2014].
- [28] A. Coutts, „Don't laugh! Microsoft has Google beat on maps, but you would never know it“, 2013. [Na mreži]. Available: <http://www.digitaltrends.com/opinion/bing-maps-google-maps/>. [Poslednji pristup 12 oktobar 2014].
- [29] T. Vincenty, „Direct and Inverse Solutions of Geodesics on the Ellipsoid with application of nested equations“, *Survey Review*, t. 23, br. 176, pp. 88-93, 1975.

- [30] C. Veness, „Vincenty formula for distance between two Latitude/Longitude points,“ 2012. [Na mreži]. Available: <http://www.movable-type.co.uk/scripts/latlong-vincenty.html>. [Poslednji pristup 15 januar 2014].
- [31] E. W. Weisstein, „Great Circle,“ MathWorld - A Wolfram Web Resource, 2010. [Na mreži]. Available: <http://mathworld.wolfram.com/GreatCircle.html>. [Poslednji pristup 15 januar 2014].
- [32] R. W. Sinnott, „Virtues of the Haversine,“ *Sky and Telescope*, t. 68, br. 2, p. 159, 1984.

## Spisak slika

|                                                                                      |    |
|--------------------------------------------------------------------------------------|----|
| Slika 1: Globalni sistem za pozicioniranje korišćenjem satelita .....                | 3  |
| Slika 2: Pozicija LBS-a u triangulaciji Interneta, GIS-a i mobilne tehnologije ..... | 6  |
| Slika 3: Proces prikupljanja podataka o lokaciji .....                               | 7  |
| Slika 4: Arhitektura TRPNS sistema .....                                             | 9  |
| Slika 5: HTML4 strukturni <i>markup</i> .....                                        | 13 |
| Slika 6: HTML5 strukturni <i>markup</i> .....                                        | 13 |
| Slika 7: Odnos pojmova od GIS-a do Web GIS-a .....                                   | 20 |
| Slika 8: <i>Client-side</i> arhitektura .....                                        | 21 |
| Slika 9: <i>Server-side</i> arhitektura .....                                        | 22 |
| Slika 10: <i>Google Maps</i> .....                                                   | 23 |
| Slika 11: <i>Bing Maps</i> .....                                                     | 24 |
| Slika 12: <i>Here</i> .....                                                          | 25 |
| Slika 13: <i>ViaMichelin</i> .....                                                   | 26 |
| Slika 14: Delovi sfere .....                                                         | 30 |
| Slika 15: Brzine izvršavanja upita .....                                             | 36 |
| Slika 16: <i>Google</i> konzola – prikaz prozora za kreiranje ključeva .....         | 39 |
| Slika 17: Primer prikaza mape generisane pomoću <i>Google Maps API</i> -ja .....     | 40 |
| Slika 18: Izgled spiska objekta na portalu .....                                     | 41 |
| Slika 19: XML struktura podataka o objektu .....                                     | 42 |
| Slika 20: Dizajn baze podataka portala .....                                         | 43 |
| Slika 21: Izgled ekrana prilikom učitavanja aplikacije .....                         | 47 |
| Slika 22: Početni izgled aplikacije .....                                            | 47 |
| Slika 23: Prikaz detalja radnje .....                                                | 48 |
| Slika 24: Galerija slika odabrane radnje .....                                       | 48 |
| Slika 25: Izgled menija za navigaciju, .....                                         | 49 |
| Slika 26: Podešavanje aplikacije .....                                               | 49 |

## Spisak tabela

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| Tabela 1: Značaj GPS preciznosti za pojedine tipove aplikacija .....               | 5  |
| Tabela 2: Prednosti i nedostaci <i>client-side</i> arhitekture .....               | 21 |
| Tabela 3: Prednosti i nedostaci <i>server-side</i> arhitekture .....               | 22 |
| Tabela 4: Oznake u Vinsentijevoj formuli .....                                     | 29 |
| Tabela 5: Tačke geografskih širina i dužina na različitim tipovima elipsoida ..... | 30 |