

# Prostorno-vremenske baze podataka o kretanju vozila

---

Šutalo, Bojan

Undergraduate thesis / Završni rad

2019

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:119:520592>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-09-11**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences -  
Institutional Repository](#)



**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET PROMETNIH ZNANOSTI**

**Bojan Šutalo**

**Prostorno-vremenske baze podataka o kretanju vozila**

**ZAVRŠNI RAD**

**Zagreb, 2019**

SVEUČILIŠTE U ZAGREBU  
FAKULTET PROMETNIH ZNANOSTI  
ODBOR ZA ZAVRŠNI RAD

Zagreb, 10. travnja 2019.

Zavod: **Zavod za Inteligentne transportne sustave**  
Predmet: **Baze podataka**

## ZAVRŠNI ZADATAK br. 5420

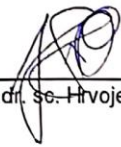
Pristupnik: **Bojan Šutalo (0135244051)**  
Studij: **Inteligentni transportni sustavi i logistika**  
Smjer: **Inteligentni transportni sustavi**

Zadatak: **Prostorno-vremenske baze podataka o kretanju vozila**

### Opis zadatka:

U radu je potrebno opisati prikupljanje i obradu povijesnih podataka o kretanju vozila cestovnom mrežom. Potrebno je oblikovati i izraditi prostorno vremensku bazu podataka u koju će se spremiti navedeni podaci te opisati alate koji su korišteni. Potrebno je prikazati i objasniti primjere osnovnih upita nad kreiranom prostornom vremenskom bazom podataka. Potrebno je izraditi jednostavno sučelje za prikaz i analizu podataka.

Mentor:

  
\_\_\_\_\_  
prof. dr. sc. Hrvoje Gold

Predsjednik povjerenstva za  
završni ispit:

\_\_\_\_\_

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET PROMETNIH ZNANOSTI**

**ZAVRŠNI RAD**

**PROSTORNO-VREMENSKE BAZE PODATAKA O  
KRETANJU VOZILA**

**SPATIOTEMPORAL DATABASE OF VEHICLE  
MOVEMENTS**

**Mentor:** prof.dr.sc. Hrvoje Gold

**Neposredni voditelj/komentor:**

Tomislav Erdelić, mag.ing.el.techn.inf

**Student:** Bojan Šutalo

**JMBAG:** 0135244051

**Zagreb, 2019**

## **Sažetak**

Zadatak ovog završnog rada je izraditi grafičko korisničko sučelje (GUI) u kojem će se prikazati i analizirati kretanje korisnika prometnom mrežom koja je ograničena na područje grada Zagreba. Prije same izrade sučelja u C#-u bilo je potrebno napraviti bazu podataka u koju se spremaju dobiveni GPS podaci koji su prilagođeni za pohranu u prostorno-vremensku bazu podataka. Podaci se mogu klasificirati ovisno o vremenu putovanja, brzini koja je zabilježena, rutama kojima je korisnik prometovao te označavanjem željenog područja za promatranje.

KLJUČNE RIJEČI: baza podataka, GPS, C#, veliki skupovi podataka

## **Summary**

The aim of this thesis is to create graphical user interface (GUI) to visualize and analyze the movement of users over the transport network restricted to the area of Zagreb. Prior to creating the GUI using C# programming language, it was necessary to create the database in which the GPS data will be saved. The GPS data had to be adjusted for storage in spatiotemporal database. The data can be classified according to the travel time, recorded speed, route the user used, etc.

KEY WORDS: databases, GPS, C#, big dataset

# SADRŽAJ

<b>1. UVOD .....</b>	<b>1</b>
<b>2. PRIKUPLJANJE I OPIS PODATAKA.....</b>	<b>2</b>
<b>2.1. Projekt SORDITO.....</b>	<b>2</b>
<b>2.2. GPS .....</b>	<b>4</b>
<b>2.3. Opis prikupljenih podataka .....</b>	<b>5</b>
<b>3. TEORIJSKA POZADINA PROSTORNO VREMENSKIH BAZA PODATAKA .....</b>	<b>7</b>
<b>4. POHRANA I OBRADA PODATAKA .....</b>	<b>10</b>
<b>4.1. Priprema podataka .....</b>	<b>10</b>
<b>4.2. Pohrana podataka u bazu podataka.....</b>	<b>11</b>
<b>4.3. Kreiranje baze podataka .....</b>	<b>11</b>
<b>4.4. Unos podataka u bazu.....</b>	<b>15</b>
4.4.1. Programski jezik C# .....	15
4.4.2. Opis procesa unosa podataka u bazu .....	17
<b>5. IZRADA SUČELJA ZA PRIKAZ I ANALIZA PODATAKA .....</b>	<b>23</b>
<b>5.1. Instalacija NuGet paketa .....</b>	<b>23</b>
<b>5.2. Izrada sučelja.....</b>	<b>24</b>
5.2.1. Prikaz podataka ovisno o vremenu i brzini .....	27
5.2.2. Prikaz podataka označavanjem miša .....	30
5.2.3. Prikaz podataka ovisno o ruti .....	32
<b>6. ANALIZA PODATAKA.....</b>	<b>34</b>
<b>7. ZAKLJUČAK.....</b>	<b>38</b>
<b>POPIS LITERATURE.....</b>	<b>39</b>
<b>POPIS SLIKA.....</b>	<b>40</b>

# 1. Uvod

Moderno vrijeme koje od nas iziskuje prilagodbu ubrzanom razvoju informatičke tehnologije također donosi i brojna olakšanja, kako u svakodnevnom životu, tako i u istraživanju, prikupljanju podataka te obradi i analizi istih s ciljem rješavanja problema s kojima se čovjek susreće svakodnevno. Taj je potencijal uvidjela i prometna struka te ga sve više nastoji uključiti u istraživanja, obrade i prikaze podataka.

Ovaj završni rad opisuje postupak izrade grafičkog korisničkog sučelja (GUI od engl. Graphical User Interface) koje prikazuje kretanje korisnika prometnom mrežom. Služeći se upitima nad prostorno-vremenskom bazom podataka prikazuju se lokacije kretanje korisnika koje se sortiraju prema vremenu kretanja, brzini ili se prikazuju cijele trajektorije kretanja korisnika. Za izradu samog sučelja korišten je programski jezik C#<sup>1</sup> dok je prostorno-vremenska baza podataka izrađena u SQL-u<sup>2</sup>. Podaci koji se obrađuju i koriste u izradi ovog rada dobiveni su od projekta pod nazivom SORDITO<sup>3</sup>, punog naziva „Sustav za optimizaciju ruta u dinamičkom transportnom okruženju“. Cilj rada je ponuditi teorijsku pozadinu za stvaranje istraživanja te prikazati dobivene rezultate. Rad se dijeli na šest poglavlja:

1. Uvod
2. Prikupljanje i opis podataka
3. Teorijska pozadina prostorno-vremenskih baza podataka
4. Pohrana i obrada podataka
5. Izrada sučelja za prikaz i analizu podataka
6. Zaključak

---

<sup>1</sup> C# <https://code.visualstudio.com/docs/languages/csharp>

<sup>2</sup> SQL <https://www.microsoft.com/en-us/download/details.aspx?id=29062>

<sup>3</sup> SORDITO Sustav za optimizaciju ruta u dinamičkom transportnom okruženju  
<https://www.fpz.unizg.hr/sordito/language/en/>

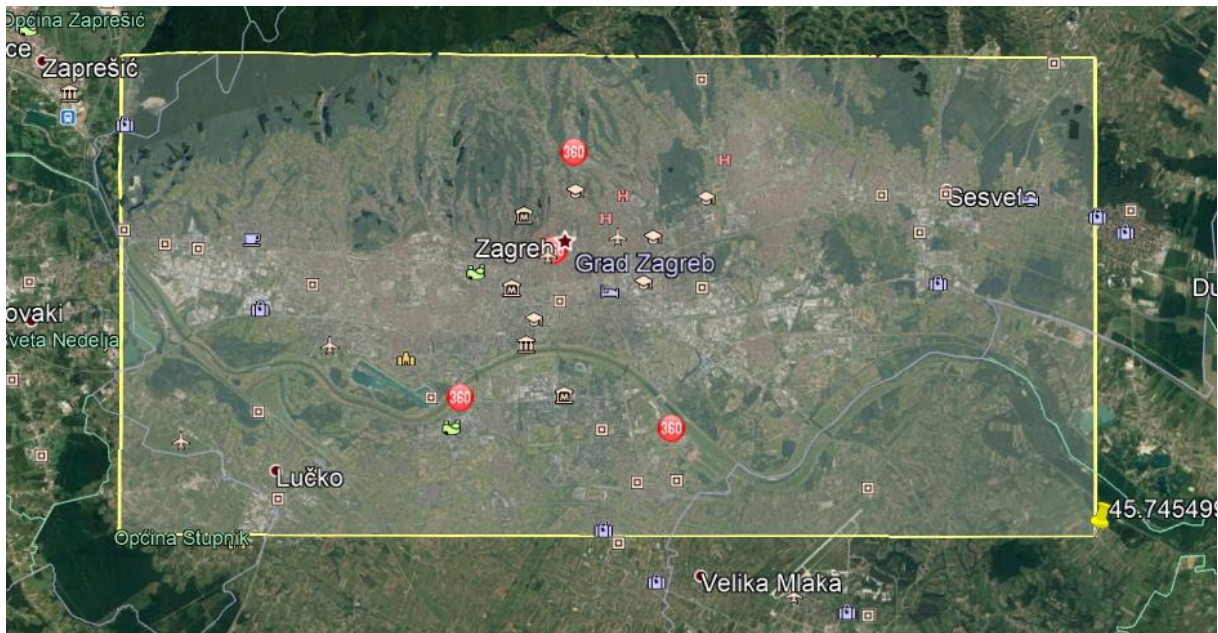
## **2. Prikupljanje i opis podataka**

Prikupljanje podataka je vrlo važan čimbenik u realizaciji projekata ili istraživanja. Podaci trebaju biti točni i korisni stoga je od velike važnosti izbor sustava ili senzora kojima se žele dobiti podaci. Odabir sustava ili senzora ovisi o tipu podataka koji će biti koristan za određenu vrstu istraživanja. Podaci korišteni u ovome radu rezultat su brojnih ispitivanja koje je provedeno kroz projekt SORDITO, o kojemu će se više biti riječi u nastavku teksta.

### **2.1. Projekt SORDITO**

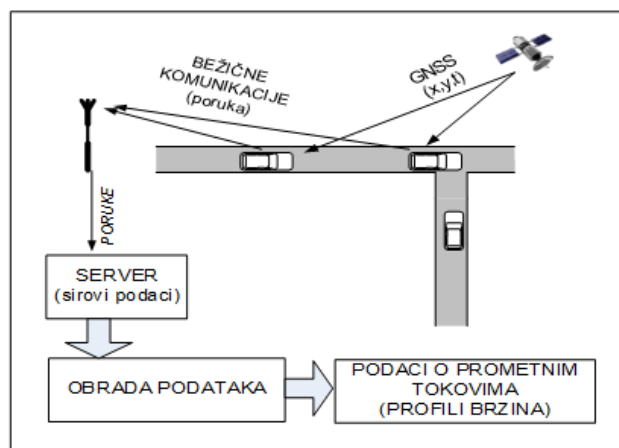
SORDITO (Sustav za optimizaciju ruta u dinamičkom transportnom okruženju) projekt je koji je započeo 17.10.2014. i trajao do 16.02.2016. Projekt je sufinanciran od strane Europske unije te je cjelokupan budžet koji je potreban za izvođenje iznosio 3.475.670,37 HRK od kojeg je 30% investirano u opremu kako bi se dobili kvalitetni i precizni podaci. Glavni je cilj projekta razviti algoritam koji će pronaći najbolju prometnu rutu na temelju predviđanja prometnih zagušenja koja se prate GPS jedinicama i radarskim detektorima koji su korišteni u izvođenju projekta s ciljem da individualni korisnici i aplikacije za planiranje ruta što kvalitetnije i točnije usmjere korisnika najbržim mogućim putem. Poligon promatranja kretanja vozila koja su praćena prostirao se prometnom mrežom grada Zagreba. U istraživanju je sudjelovalo oko 6000 vozila koja su u sebi imala ugrađene GPS zapisnike koji su pratili kretanje, brzinu i vrijeme zadržavanja svakog vozila [1], [2].





**Slika 1.** Prikaz poligona unutar kojeg su prikupljeni podaci

Nastavno na prethodno spomenutu opremu, slika 2 prikazuje proces prikupljanja GPS (engl. Global Positioning System) podataka. GNSS (engl. Global Navigation Satellite System) pruža geoprostorno pozicioniranje s globalnom pokrivenošću te omogućuje prijemnicima u automobilu određivanje lokacije s odmakom od nekoliko metara. Prijamnici računaju vrijeme i poziciju te se podaci šalju bežičnom komunikacijom u stanice koje sirove podatke prosljeđuju serveru. Sirovi podaci se obrađuju i dobiju se podaci o prometnim tokovima.



**Slika 2.** Shema prikupljanja podataka

## 2.2. GPS

Sustav za globalno pozicioniranje (GPS) predstavlja sustav satelita koji omogućuju precizno lociranje sustava koji koristi GPS prijemnik. Konfiguracija sustava sastoji se od 24 satelita koji su raspoređeni u 6 putanja te se u svakoj putanji nalaze 4 satelita, a u svakoj drugoj po jedan pričuvni te se na taj način iznad horizonta u svakom trenutku nalazi 5 ili više satelita čime je postignuta globalna pokrivenost. GPS sustav čine tri segmenta: svemirski segment, kontrolni segment i korisnički segment. Svemirski segment sastoji se od satelita koji se gibaju oko Zemlje u kružnim orbitama na visini oko 20 000 km te se time pokriva cijelu površinu Zemlje GPS signalom. Kontrolni segment ima ulogu nadzora i upravljanja cijelim sustavom. Kontrolne su stanice raspoređene duž cijele Zemlje, kontinuirno prate sve GPS satelite i šalju primljene signale u glavnu kontrolnu postaju na obradu. U glavnoj postaji obavlja se proces izračunavanja odstupanja pozicija svakog satelita od preciznih orbitalnih modela i odstupanje njihovih atomskih satova od GPS vremena. Korisnički segment obuhvaća sve korisničke uređaje koji koriste GPS sustav. Takvi uređaji imaju GPS prijemnike te obrađuju GPS podatke s ciljem dobivanja željenih informacija poput brzine, lokacije i sličnih podataka. Svi sateliti emitiraju na dvije frekvencije 1575,42 MHz (signal L1) i 1227,6 MHz (signal L2). Jedina razlika između signala L1 i L2 je ta što je signal L1 moduliran C/A i P kodom dok je signal L2 moduliran samo P kodom. C/A kod je niz od 1023 bita koji se emitira samo u signalu L1 te omogućuje pozicioniranje standardnom razinom točnosti, dok je P kod niz od oko  $10^{14}$  bitova, emitira se u L1 i L2 signalima i omogućuje pozicioniranje višom razinom preciznosti. Točnost samog sustava je veća kada u njegovoj okolini postoji manje objekata koji bi mogli potencijalno smetati prilikom slanja signala, odnosno kako bi se smanjila mogućnost reflektiranja signala u pogrešnom smjeru [3]. Slika 3 prikazuje primjer prikupljenih GPS podataka jedne rute vozila.



**Slika 3.** Prikaz GPS zapisa u jednoj ruti vozila

### 2.3. Opis prikupljenih podataka

Podaci koji su prikupljeni tijekom projekta opisuju kretanje svakog vozila na njegovoj trajektoriji. Prikupljeni podaci su takozvani sirovi podaci koji nisu prilagođeni niti obrađeni. Takvi neobrađeni podaci koji se nalaze u tekstualnoj datoteci su prikazani na slici 4. Jedan redak datoteke odgovara jednoj ruti koja ima određeni broj GPS zapisa ovisno o tome koliko je ruta duga, to jest koliko je vremena vozilo provelo u kretanju prije gašenja motora. Svaki GPS zapis sastoji se od točno devet (9) podataka koji daju određene informacije o točki zapisa te se



kontinuirano ponavljaju kroz rutu prilikom kretanja vozila. Jedan GPS zapis čine sljedeći atributi koji imaju isti redoslijed unutar svakog GPS zapisa, i prikazani su na slici 5.

```
1389367308;16.160558760166168;45.793641131601149;16.160628497600555;45.793769237134569;42;290;-
391666;107;1389367317;16.159245818853378;45.793990849661320;16.159311532974243;45.794112408900887;45;290;-
391666;109;1389367325;16.157977133989334;45.794328409664672;16.158048212528229;45.794461188939486;49;290;-
391666;105;1389367333;16.156609207391739;45.794692149318649;16.156686991453171;45.794835213364948;52;290;-
391666;113;1389367340;16.155349910259247;45.795019419471885;16.155430376529694;45.795170898149273;54;290;-
391666;104;1389367347;16.154065132141113;45.795353233048175;16.154142916202545;45.795500970631288;54;290;-
391666;106;1389367354;16.152765601873398;45.795671148880857;16.152842044830322;45.795827300993281;55;288;-
391664;107;1389367361;16.151447296142578;45.795988127861257;16.151519715785980;45.796137733833369;55;288;-
391664;108;1389367368;16.150122284889221;45.796292949604926;16.150192022323608;45.796448164944103;54;286;-
391664;108;1389367375;16.148826777935028;45.796579069110997;16.148899197578430;45.796738958783351;54;286;-
391664;106;1389367381;16.147517859935760;45.796869862267108;16.147587597370148;45.797024140978181;53;286;-
391664;106;1389367388;16.146222352981567;45.797177484243505;16.146288067102432;45.797299036532060;54;286;-
391663;106;1389367395;16.144985854625702;45.797505674814047;16.145028769969940;45.797584215860518;54;286;35
-391681;130;1389367517;16.134214103221893;45.809337009499579;16.134236901998520;45.809331400611875;60;18;
391681;117;1389367523;16.134691536426544;45.810189553862486;16.134700924158096;45.810185814661317;60;22;-
391681;102;1389367531;16.135280281305313;45.811186976810276;16.135299056768417;45.811180433325241;62;24;-
391681;120;1389367537;16.135799288749695;45.812047905779004;16.135828793048859;45.812038558087522;62;22;-
402536;104;1389367543;16.136316955089569;45.812906017175642;16.136337071657181;45.812899473892585;61;20;-
402536;103;1389367549;16.136780977249146;45.813760376389524;16.136806458234787;45.813753833206839;61;18;-
402536;102;1389367555;16.137188673019409;45.814624069755908;16.137199401855469;45.814621265578353;59;16;-
402535;101;1389367562;16.137538701295853;45.815623282706071;16.137584298849106;45.815615805033083;58;12;-
402534;114;1389367569;16.137835085391998;45.816616869573373;16.137872636318207;45.816611261418807;57;10;-
402534;113;1389367576;16.138053685426712;45.817603895982238;16.138089895248413;45.817600157278868;55;6;-
402533;111;1389367582;16.138191819190979;45.818509589481799;16.138215959072113;45.818508654821180;50;4;-
402533;101;1389367590;16.138237416744232;45.819504994137041;16.138266921043396;45.819504059493141;51;0;402
-402532;103;1389367605;16.138107329607010;45.821550892054475;16.138147562742233;45.821551826664027;55;354;-
402532;124;1389367613;16.137957125902176;45.822492970521381;16.137993335723877;45.822495774302546;53;352;-
402532;105;1389367621;16.137765347957611;45.823549986014307;16.137792170047760;45.823551855166265;53;352;-
402532;118;1389367628;16.137600392103195;45.824464928402072;16.137627214193344;45.824466797523328;52;352;-
402532;102;1389367636;16.137408730726242;45.825517233740004;16.137442141771317;45.825517233740004;52;352;
```

Slika 4. Prikaz neobrađenih podataka

Atribut	Primjer	Opis
UTC	1389367308	UTC vrijeme u sekundama (od 1.1.1970) – pretvoriti u lokalno vrijeme po potrebi (Za provjeru: 1389367308 → 10.01.2014. 16:21:48 ( <a href="https://www.epochconverter.com/">https://www.epochconverter.com/</a> ))
MATCH_X	15.952734500169754	Pridružena geografska dužina u karti [°]
MATCH_Y	45.780397919459887	Pridružena geografska širina u karti [°]
ORG_X	15.952774733304977	Originalna geografska dužina [°]
ORG_Y	45.7804016606603244	Originalna geografska širina [°]
ORG_S	50	Originalna brzina kretanja [km/h]
ORG_H	348	Originalni geografski smjer kretanja u odnosu na sjever [°] (360° = 0°)
LINK	-214696	ID pridruženog linka i digitalnoj karti
MATCH_D	110	Pridružena prijeđena cestovna udaljenost od prethodnog zapisa [m]

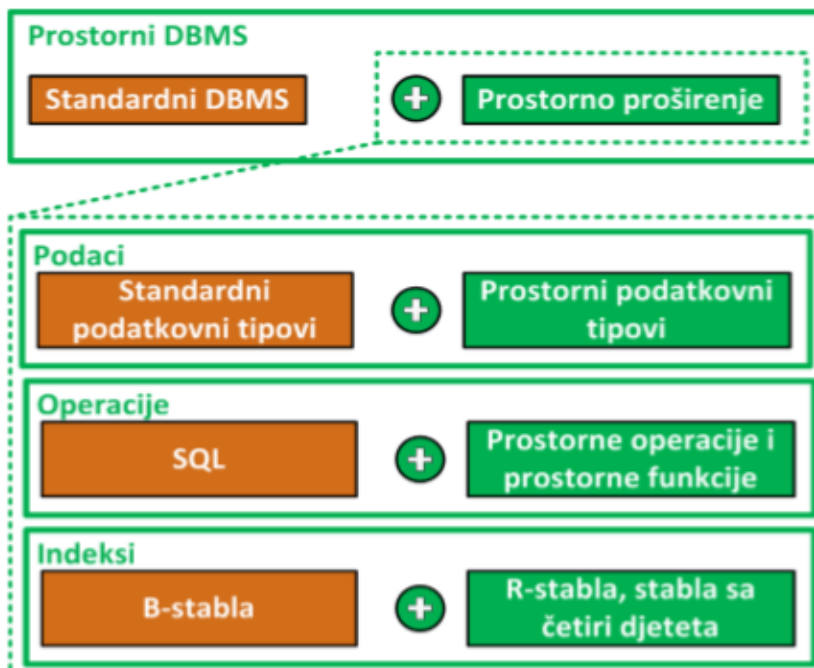
Slika 5. Opis podataka jednog GPS zapisa

### 3. Teorijska pozadina prostorno vremenskih baza podataka

Prostorno vremenske baze podataka služe za pohranu i obradu informacija koje sadrže vremenske i prostorne atribute. Najčešći primjer korištenja ove vrste baza podataka je praćenje pokretnih entiteta. Kao što je vidljivo iz naziva, prostorno vremenske baze podataka dijele na prostorne baze podataka i vremenske baze podataka [4],[5]. Prostorne baze podataka proširuju DBMS i jezik upita na takav način da on može predstavljati geometrijske oblike i obavljati upite nad njima. Vrste prostornih podataka koje treba sadržavati prostorni SQL definira OGC (od engl. Open Geospatial Consortium). Cilj OGC je da osigura konzistentan način opisivanja i upravljanja prostornim i geoprostornim podacima. Najznačajnije OGC specifikacije definiraju navedene oblike zapisa prostornih objekata [4], [6]:

- **WKT (Well Known Text)** je jezik koji opisuje i predstavlja vektorske geometrijske objekte na kartama, kartografske projekcije prostornih objekata i transformacije između kartografskih projekcija; [4];
- **WKB (Well Known Binary)** je binarni ekvivalent WKT koji je posebno prilagođen pohrani u bazama podataka; [4];
- **GML (Geography Markup Language)** je definirana gramatika za opisivanje geografskih podataka [4].

Na slici 6 prikazano je na koji način se standardni DBMS proširuje sa prostornim dijelom.



Slika 6. Prikaz proširenja DBMS-a

Dva su tipa prostornih podataka, a to su geometrijski i geografski tip. Geometrijski tip podataka služi za prikazivanje informacija u dvodimenzionalnom prostoru gdje je položaj točke na plohi definiran koordinatama x i y. Ovaj tip podataka se koristi za spremanje koordinata sljedećih tipova koordinatnog sustava: projekcijske koordinate, geografske koordinate i planarne koordinate. Geografski tip služi za spremanje geodetskih prostornih podataka kojima je vidljiva i zakrivljenost zemlje. Osnova geografskog tipa podataka je trodimenzionalnost i zakrivljeni model svijeta. Za određivanje pozicije svake točke koja definira geometriju ovog tipa potrebno je koristiti geografski koordinatni sustav. S obzirom da se geografski tip definira točku koristeći geografsku širinu i dužinu, koordinatne vrijednosti se mogu mjeriti stupnjevima [7].

Geometrijski i geografski tip podataka imaju različitu duljinu, što znači da iznos potreban za spremanje ovisi o kompleksnosti opisanog objekta. Takve tipove podataka SQL sprema kao niz binarnih podataka te svaki niz počinje s informacijom poput koji tip geometrije se koristi, prostorni referentni sustav i broj točaka, a nakon toga slijede koordinate. Primjer SQL naredbe za spremanje prostornih podataka prikazan je na slici 7. Potrebno je prvo definirati tablicu i stupac u koji se sprema podatak te potom navesti podatak koji se sprema.

```
INSERT INTO Podaci (PolozajNaKarti) VALUES (geography :: Point(45.815,15.98306,4326))
```

**Slika 7.** Prikaz insert naredbe za spremanje podataka u tip geography

Na slici 8 prikazan je izgled spremljenih podataka u tip geography, dok je na slici 9 prikazan select za dobavljanje podataka na način da su koordinate odvojene, a ne zapisane u nizu kao na slici 8.

IDZapis	PolozajNaKarti	
1	285011	0xE6100000010C5B7E580896E54640090000611A293040
2	285012	0xE6100000010C2C4FFE7DA1E5464006008055C4283040
3	285013	0xE6100000010C94E5A68DACE54640F6FF7F3071283040
4	285014	0xE6100000010C0DD8BEB78B8E54640F5FF7F8A17283040
5	285015	0xE6100000010C8A284332C3E54640F3FFFF02C5273040
6	285016	0xE6100000010CDE5A7E22CEE54640FCFFFCF70273040
7	285017	0xE6100000010C38685D8DD8E54640010080A51B273040
8	285018	0xE6100000010C939560F0E2E5464006000040C5263040
9	285019	0xE6100000010C514B68EDECE54640FAFFFF696E263040
10	285020	0xE6100000010C43418D4DF6E54640F8FFFF8219263040

**Slika 8.** Izgled podataka spremljenih u tip geography

```
select top 10 IDZapis, PolozajNaKarti.Lat AS 'Lat', PolozajNaKarti.Long AS 'Long' from podaci
```

	IDZapis	Lat	Long
1	285011	45,7936411316011	16,1605587601662
2	285012	45,7939908496613	16,1592458188534
3	285013	45,7943284096647	16,1579771339893
4	285014	45,7946921493186	16,1566092073917
5	285015	45,7950194194719	16,1553499102592
6	285016	45,7953532330482	16,1540651321411
7	285017	45,7956711488809	16,1527656018734
8	285018	45,7959881278613	16,1514472961426
9	285019	45,7962929496049	16,1501222848892
10	285020	45,796579069111	16,148826777935

Query executed successfully. DESKTOP-LD70TFH\SQLEXPRESS ...

**Slika 9.** Primjer selecta za dobavljanje koordinata u zasebne stupce

Vremenske baze podataka su baze podataka koje imaju ugrađene vremenske aspekte radi omogućivanja rada s takvim podacima putem prilagođenog i proširenog jezika. Vremenske baze imaju za cilj integraciju vremenskih koncepata u podatkovni model DBMS-a i jezik upita te proširenje sustava s tim kako bi izvršavanje upita bilo učinkovito. Kada je riječ o praćenju prostornih objekata kroz vrijeme, razlikuju se tri vrste vremena [4]:

- Transakcijsko vrijeme ( engl. transaction time) je vrijeme tijekom kojeg je neka tvrdnja ili podatak istinit u bazi podataka,
- Važeće vrijeme (engl. valid time) je vrijeme tijekom kojeg je neka činjenica istinita u stvarnom svijetu,
- Bitemporalno vrijeme (engl. bitemporal time) je vrijeme kojeg čini kombinacija važećeg vremena i transakcijskog vremena [4].

Vremenski koncepti mogu se opisati sljedećim skupom tipova podataka:

- Instant – točka na vremenskoj liniji kontinuiranog modela,
- Interval – ne usidreni interval na nekoj vremenskoj liniji, ima samo trajanje,
- Period – usidreni interval na vremenskoj liniji, za razliku od standardnih tipova podataka period nije uređen nego je definirano samo parcijalno uređenje,
- Periods – skup usidrenih intervala, naziva se još i vremenskim elementom

Postojeći sustav za upravljanje bazom podataka (SUBP) od tipova podataka sadrži date, time, timestamp, interval dok je za upravljanje vremenskim dijelom baze potrebna implementacija tipova instant, interval, period i periods te zbog nedostatka vremena nije implementirana stvarna prostorno-vremenska baza podataka

## 4. Pohrana i obrada podataka

Podaci dobiveni za obradu i korištenje nalaze se u tekstualnoj datoteci i kao takvi nisu pogodni manipulacijama. Kako bi se olakšao pristup i prikaz informacija za aplikaciju podaci su spremljeni u bazu podataka.

### 4.1. Priprema podataka

Dobivene podatke je potrebno prilagoditi i obraditi kako bi se mogli spremiti u bazu podataka ovisno o dostupnim tipovima podataka u bazi. Početni skup podataka se sastojao od UTC vremena, MATCH\_X, MATCH\_Y, ORG\_X, ORG\_Y, ORG\_S, ORG\_H, LINK i MATCH\_D. Svi podaci su bili pogodni za dodavanje u bazu podataka sukladno dostupnim tipovima podataka koje nudi SQL Server osim UTC vremena koji je bio zapisan u obliku: „1389367308“, odnosno broj sekundi od 1.1.1970., kao što je vidljivo u slici 2. Budući da je ovaj tip zapisa bilo potrebno dodati u bazu kao atribut kojem je tip podatka timestamp, a takav tip ne podražava ovakav oblik zapisa bilo je potrebno prilagoditi zapis tako što se UTC vrijeme preko zadane funkcije pretvorilo u datumski tip podatka oblika: „10.01.2014. 16:21:48“.

Prilikom unosa podataka u bazu treba biti oprezan kako bi se ispunila sva pravila koja nalažu način spremanja podataka. Ta pravila koja je potrebno pratiti poznata su po nazivu normalizacija. Normalizacija korištena u ovom radu dijeli se na tri normalne forme. Prva normalna forma nalaže da atribut sadrži nedjeljive vrijednosti, ne smije sadržavati skup podataka i da su neključni atributi ovisni o primarnom ključu. Budući da su dobiveni podaci već ispunjavali prvu normalnu formu nije ih bilo potrebno prilagođavati na toj razini. Sljedeća, druga normalna forma, zahtjeva ispunjavanje prve te da svaki atribut koji se nalazi u tablici, a da nije primarni ključ, treba biti ovisan o primarnom ključu u cijelosti, a ne samo njegovom dijelu. Nakon ispunjavanja te dvije normalne forme treća normalna forma nalaže da svaki atribut bude ovisan o primarnom ključu i niti jednom drugom atributu [8].



## 4.2. Pohrana podataka u bazu podataka

Prije nego bi se podaci mogli unijeti u SQL Server potrebno je napraviti bazu podataka. Kako bi bilo moguće spremirati podatke potrebno je kreirati bazu podataka koja odgovara svakom tipu podatka koje se bude nalazilo u pojedinom polju. Na slici 4. prikazana je skripta u SQL Serveru koja prilikom pokretanja kreira bazu podataka s zadanom tablicom i stupcima koji su prilagođeni tipovima podataka. Tipovi podataka koji su korišteni kako bi se izradila baza podataka su sljedeći:

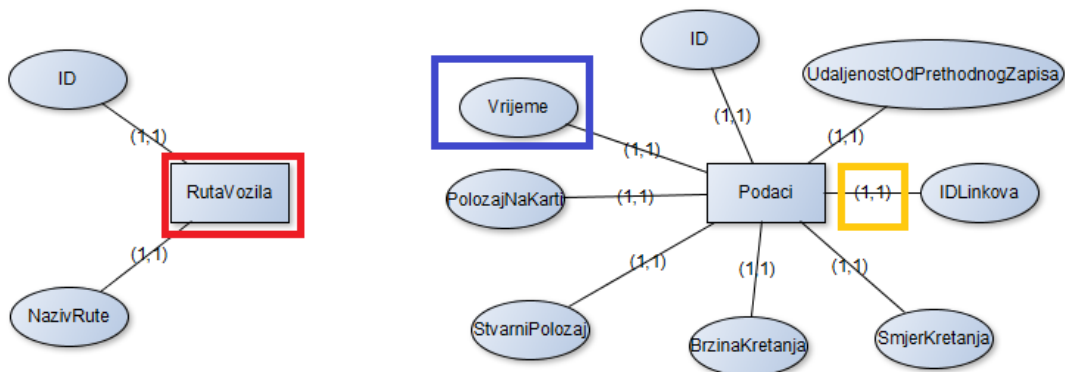
- *Int* - cjelobrojni zapis,
- *Timestamp* - definira godinu, mjesec, dan, sat, minutu i sekundu,
- *Geography* – objekt u prostornom (zemljopisnom) koordinatnom sustavu,
- *Float* - aproksimirani podatkovni tip koji koristi plivajući zarez,
- *Varchar* - reprezentacija teksta varijabilne veličine. Ne podržava č, ć, š i dr.

## 4.3. Kreiranje baze podataka

Prije početka kreiranja baze podataka potrebno je napraviti ER model (engl. Entity-Relationship Model). ER modeliranje je konceptualno dizajniranje baze podataka i sastoji se od dijagrama entiteta i ER dijagrama. Spomenuti dijagrami se modeliraju unutar alata Graph Editor koji je napravljen od strane tvrtke yWorks<sup>4</sup>. Dijagram entiteta definira logičku strukturu baze podataka te prikazuje i definira entitete i attribute. Entitet prilikom kreiranja baze podataka prelazi u istoimenu tablicu dok se iz atributa saznaju broj i nazivi stupaca koji sačinjavaju tablicu. Entiteti se označavaju pravokutnicima dok se atributi označavaju elipsama. Također je potrebno navesti kardinalitet svakog atributa. Kardinalitet je broj koji govori koliko vrijednosti jedan atribut može imati za opis pojedinog elementa entiteta. Prikaz dijagrama entiteta moguće je vidjeti na slici 10 gdje crvena boja prikazuje entitet, plava boja atribut, a žuta prikazuje kardinalitet atributa. Vrijednost jedan (1) kao donja granica označava da se prilikom unosa atribut mora unijeti, dok vrijednost jedan (1) kao gornja granica označava da se može unijeti maksimalno jedna vrijednost atributa. Na slici 10 su svi atributi kardinaliteta (1,1).

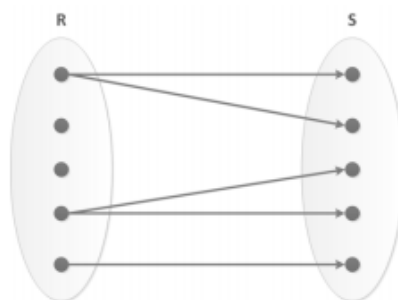
---

<sup>4</sup> yWorks <https://www.yworks.com/>

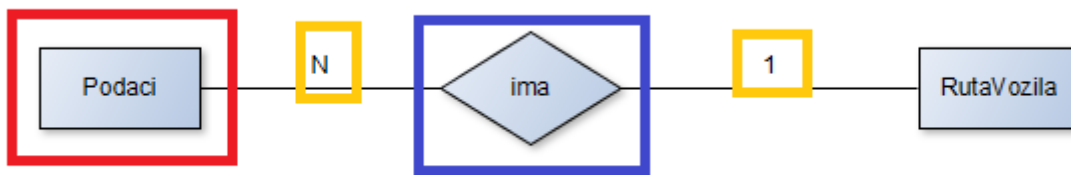


**Slika 10.** Prikaz dijagrama entiteta

Nakon izrade dijagrama entiteta potrebno je izraditi ER dijagram. ER dijagram (engl. Entity-Relationship diagram) prikazuje odnose između entiteta bez atributa. Entiteti su prikazani pravokutnikom, veze između entiteta prikazane su rombom u koji je upisan naziv veze. Nakon označavanja entiteta i naziva veze potrebno je odrediti tip veze koja je u ovom slučaju 1:N što je prikazano na slici 12. Ovaj tip veze označava da svaki element iz skupa R može biti povezan sa više elemenata skupa S, dok svaki element iz skupa S može biti povezan sa samo jednim elementom skupa R. Skupovi R i S su prikazani na slici 11. Odnosno, da jednoj ruti vozila može pripadati više redaka (N) (podataka) iz tablice podaci, a da svaki podatak pripada točno jednoj ruti. Rješenje ove relacijske sheme je da se u tablicu Podaci doda novi stupac RutaVozilaID koji je strani ključ.



**Slika 11.** Prikaz skupova R i S



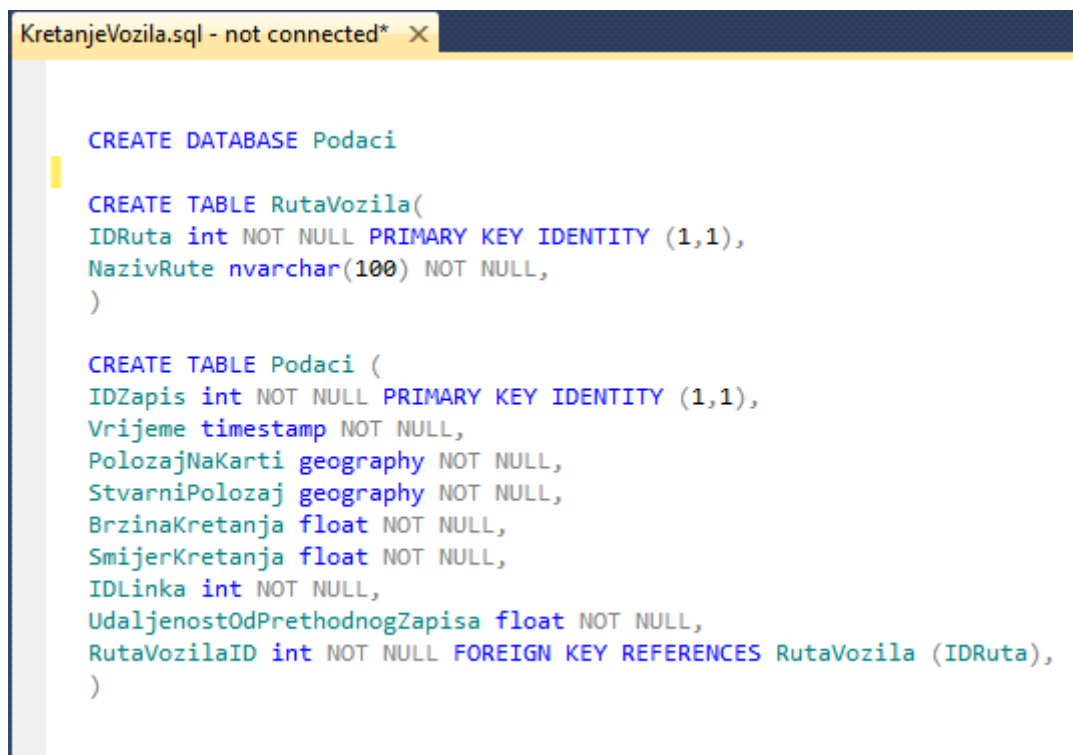
**Slika 12.** Prikaz ER dijagrama

Na slici 12 entiteti su označeni crvenom bojom, veza između entiteta plavom, dok je tip veze označen žutom bojom.

Baza podataka je kreirana pod imenom Podaci i sadrži dvije tablice naziva RutaVozila i Podaci kao što je prikazano na slici 13. Prva tablica naziva RutaVozila sadrži dva stupca IDRuta i NazivRute. IDRuta je primarni ključ koji program sam generira i brine se o njemu. Uloga primarnog ključa je jedinstveno odrediti svaki zapis koji se nalazi u tablici tako da se razlikuje od svih drugih zapisa. Tip podataka primarnog ključa je integer (int). Drugi stupac pod imenom NazivRute za svaki zapis sadrži naziv koji mu pripada ovisno na kojoj se ruti nalazi. Za svaki zapis moguće je odrediti mogućnost praznog mjesta unutar tablice tako da se uz tip podatka upise *null* odnosno *not null* ako se ne želi dati tu mogućnost. Druga tablica pod imenom Podaci sadrži devet stupaca koji su opisani u nastavku:

- IDZapis - primarni je ključ u tablici, postavljen je tako da se sam generira i brine o vrijednostima unutar stupca u kôdu zapisano „primary key identity (1,1)“,
- Vrijeme – vrijeme u kojem je nastao GPS zapis, tip podatka timestamp,
- PolozajNakarti – pridruženi geografski položaj na karti, tip podatka geography,
- StvarniPolozaj – originalni položaj, tip podatka geography,
- BrzinaKretanja – brzina vozila u datom trenutku, tip podatka float,
- SmjerKretanja – originalni geografski smjer kretanja u odnosu na sjever, tip podatka float,
- IDLinka – ID pridruženog linka na karti, tip podatka int,
- UdaljenostOdPrethodnogZapisa – udaljenost od prethodnog GPS zapisa koji se nalazi na istom linku u metrima, tip podatka float
- RutaVozilaID - strani ključ koji povezuje tablicu Podaci sa tablicom RutaVozila i dodjeljuje svakom zapisu rutu na kojoj se nalazio, tip podatka int.

Prilikom kreiranja tablica unutar baze podataka važno je slijediti redoslijed tako da se prvo kreira tablica RutaVozila pa potom Podaci jer tablica Podaci sadrži strani ključ koji se referencira na prvu tablicu. U suprotnom će SQL javiti pogrešku da izvršavanje kôda nije moguće.



```
KretanjeVozila.sql - not connected* X

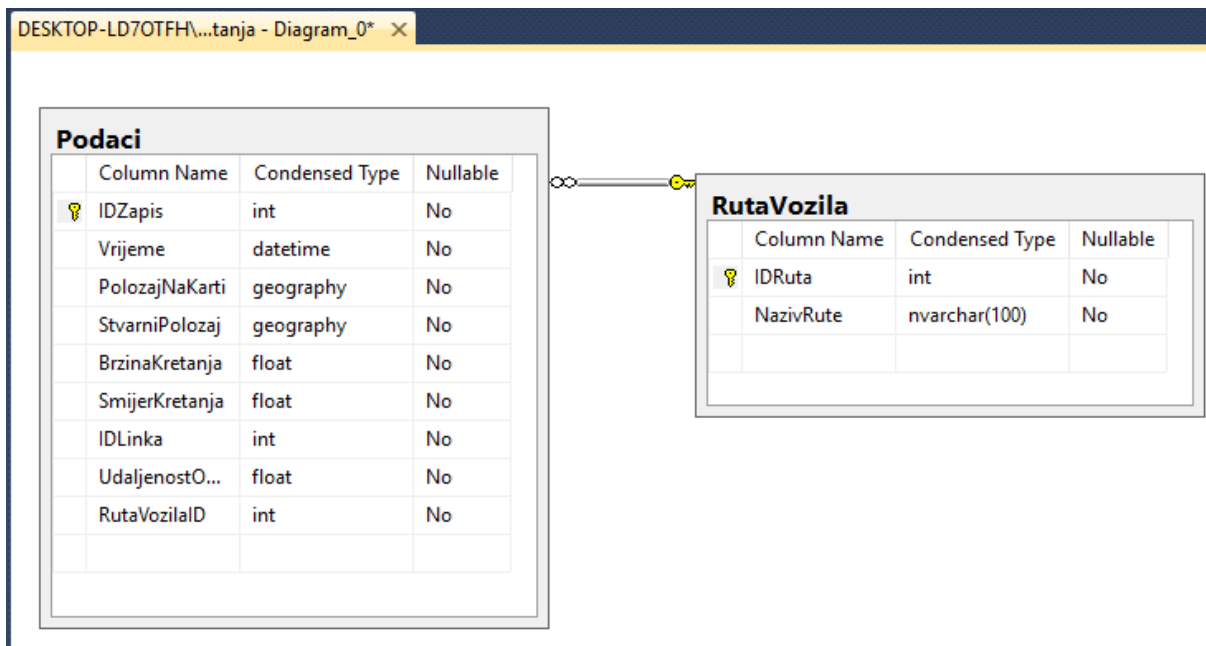
CREATE DATABASE Podaci

CREATE TABLE RutaVozila(
  IDRuta int NOT NULL PRIMARY KEY IDENTITY (1,1),
  NazivRute nvarchar(100) NOT NULL,
)

CREATE TABLE Podaci (
  IDZapis int NOT NULL PRIMARY KEY IDENTITY (1,1),
  Vrijeme timestamp NOT NULL,
  PolozajNaKarti geography NOT NULL,
  StvarniPolozaj geography NOT NULL,
  BrzinaKretanja float NOT NULL,
  SmijerKretanja float NOT NULL,
  IDLinka int NOT NULL,
  UdaljenostOdPrethodnogZapisa float NOT NULL,
  RutaVozilaID int NOT NULL FOREIGN KEY REFERENCES RutaVozila (IDRuta),
)
```

**Slika 13.** Skripta za kreiranje baze podataka „Podaci“

Radi lakšeg razumijevanja i pregleda kreirane baze SQL nudi grafički prikaz povezanosti između tablica na način da fizički prikazuje relaciju između primarnog i stranog ključa što je prikazano na slici 14. U ovom slučaju kada je kreirana baza jednostavna i ne sadrži mnogo relacija grafički prikaz će biti jednostavan, dok je za kompleksnije baze odličan način za pregled relacija.



Slika 14. Grafički prikaz povezanosti tablica

## 4.4. Unos podataka u bazu

Nakon izrade baze podatke potrebno je spremati podatke u bazu. Spremanje je moguće na dva načina. Prvi način je da se preko programa SQL podaci uvezu u bazu podataka, a drugi način spremanja je da se obavlja programskim jezikom gdje se vrši spajanje na bazu i unos podataka iz druge datoteke. Za potrebe ovog rada je korišten programski jezik C#.

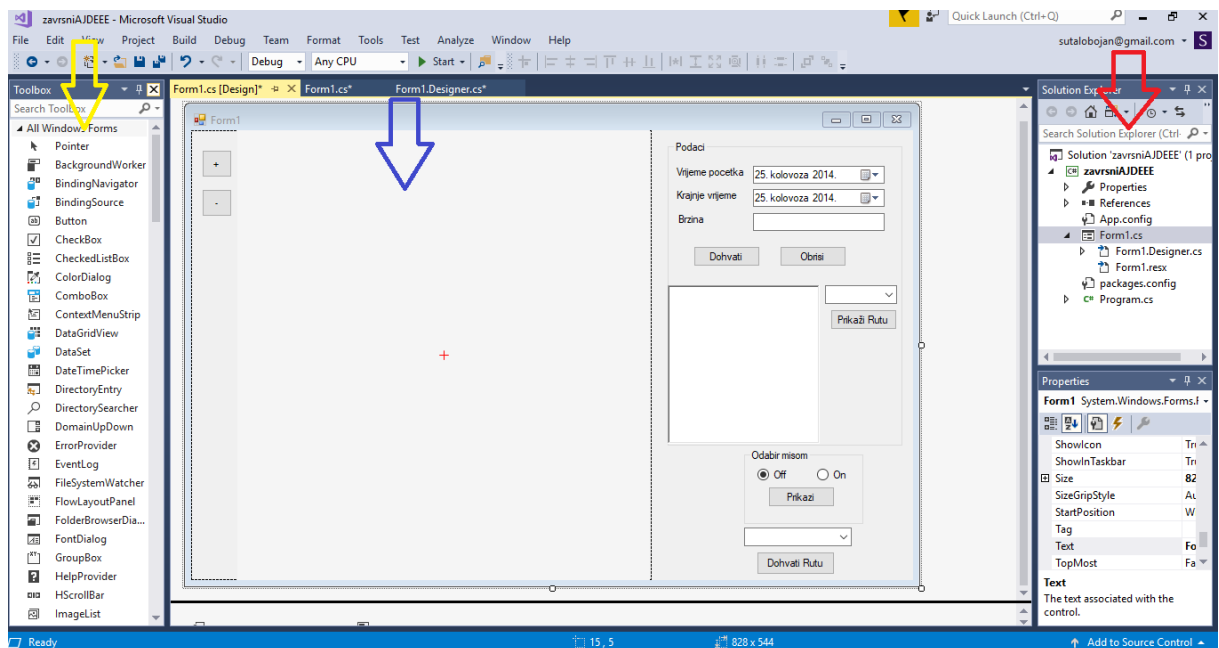
### 4.4.1. Programski jezik C#

Kako bi se podaci spremili u bazu podataka, a nakon spremanja i vizualizirali, potrebno je izraditi aplikaciju to jest korisničko sučelje koje će vizualizirati podatke i prikazivati ih na karti. Alat koji je korišten za izradu aplikacije je iz Microsoftovog paketa pod nazivom Visual Studio Express. Visual Studio Express alat je razvijen od strane Microsofta i namjenjen je razvoju aplikacija. Ovaj alat podržava razne programske jezike kao što su Python, C, C++ i C# koji je korišten prilikom izrade ovog rada.

U ovom radu koristit će se program s grafičkim sučeljem – koji se koristi za kompleksnije programe te detaljniji i jasniji prikaz podataka. Ovakav oblik programa korisniku omogućuje da sam oblikuje sučelje i dodaje razne kontrole koje su ponuđene. Prilikom izrade aplikacije korišten je predložak grafičkog sučelja zajedno s programskim jezikom C#. C# je objektno-

orijentirani programski jezik u kojem je program radije organiziran oko podataka ili objekta nego funkcija i logike. Objekt može biti definiran kao polje podataka koji ima jedinstvene atribute i ponašanje. Objektno – orijentirano programiranje je bazirano na sljedećim principima [9]:

- Definiranje podataka – implementacija i stanje svakog objekta spremljeni su unutar definiranih granica, nepripadajući objekti nemaju pristup ni mogućnost mijenjati ih.,
- Apstrakcija – objekti otkrivaju samo unutarnje mehanizme koji su nužni za korištenje drugih objekata, skrivajući pritom nepotrebni implementirajući kôd, zahvaljujući tome, promjene i dodaci su lako izvedivi,
- Nasljeđivanje – odnosi i podtipovi mogu biti dodijeljeni objektima što omogućuje korištenje zdravog razuma dok se održava posebna hijerarhija,
- Višeobličje – objekti mogu imati više od jednog oblika, ovisno o kontekstu, program će odrediti koje značenje je potrebno za određeni objekt, smanjujući potrebu za duplikacijom kôda.



**Slika 15.** Izgled sučelja Microsoft Visual Studia

Na slici 15 žutom strelicom je prikazan izbor alata (engl. Toolbox) koji su dostupni za izradu aplikacije, plavom strelicom je prikazan prostor u kojem korisnik može ovisno o želji kreirati izgleda sučelja i slagati potrebne alate iz toolboxa dok je crvenom strelicom prikazan prostor u

kojem se mogu mijenjati značajke i karakteristike svakog alata zasebno, primjerice naziv koji će biti prikazan na gumbu prilikom pokretanja sučelja ili boje slova.

#### 4.4.2. Opis procesa unosa podataka u bazu

Unos podataka iz tekstualne datoteke pod imenom „studentskirad.txt“ obavlja se preko programa napisanog u programskom jeziku C#. Prije samog unosa podataka potrebno je povezati se s bazom podataka naziva PodaciKretanja. Za stvaranje konekcije sa bazom koristi se klasa *SqlConnection* koja je definirana *connection stringom*. Connection string sadrži upute za lokaciju SQL Expressa i naziv baze podataka na koju se vrši spajanje. Za pristup bazi podataka korišten je *Integrated Security = True* što znači da je korištena windows autentifikacija. Nakon uspješnog povezivanja na bazu upisuje se naredba *StreamReader* koja pristupa tekstualnoj datoteci i čita podatke iz nje. Navedeno je vidljivo na slici 16.

```
sql = new SqlConnection("Data Source=DESKTOP-LD70TFH\\SQLEXPRESS;Database=PodaciKretanja; Integrated Security=True");
sql.Open();

StreamReader podaci = new StreamReader("studentskirad.txt");
```

**Slika 16.** Kôd za spajanje na bazu i čitanje tekstualne datoteke

Budući da SQL decimalni broj zapisuje sa zarezom, a u tekstualnoj datoteci su decimalni brojevi zapisani točkom potrebno je zamijeniti ta dva znaka kako bi se dobili valjane vrijednosti prilikom unosa podataka u tablice. Za promjenu znakova koristila se naredba „Replace('.', ',')“ . Tablica naziva RutaVozila sastoji se od dva stupca. Prvi je IDRuta o kojem se brine SQL dok je u drugi stupac NazivRuta potrebno ubaciti vrijednosti kako bi se razlikovali podaci koji pripadaju određenim rutama. Unutar while petlje naredbom „INSERT INTO RutaVozila " + "VALUES ('Ruta broj: " + brojRute + "'“ se popunjava stupac NazivRuta koji je tipa integer u kôdu zapisano int. Varijabla „brojRute“ je unaprijed deklarirana te je postavljena na početnu vrijednost 1, a nakon svakog izvršavanja while petlje njen se broj povećava za +1 što je vidljivo na slici 17.

```

int brojRute = 1;
while (podaci.EndOfStream == false)
{
    string line = podaci.ReadLine().Replace('.', ',');

    string naredbaRuta = "INSERT INTO RutaVozila " + "VALUES ('Ruta broj: " + brojRute + "')";
    SqlCommand cmdRuta = new SqlCommand(naredbaRuta, sql);
    cmdRuta.ExecuteNonQuery();
    brojRute++;
}

```

**Slika 17.** Ubacivanje u tablicu RutaVozila

U drugu tablicu naziva Podaci ubacuju se vrijednosti iz tekstualne datoteke „studentskirad.txt“. Nakon izvršavanja naredbe *StreamReader* i zamijene znakova točke i zareza podaci se odvajaju s znakom točka-zarez (;) i spremaju se u polje `lineSplit` zapisano u kôdu „`string[] lineSplit = line.Split(';');`“. Podaci su spremljeni na način da su sortirani redom jedan ispod drugog te svaki podatak ima svoj index počevši od broja nula što je prikazano na slici 18.

index	podaci
0	389367308
1	16.160558760166168
2	45.793641131601149
3	16.160628497600555
4	45.793769237134569
5	42
6	290
7	-391666
8	107
9	1389367317
10	16.159245818853378
11	45.793990849661320
12	16.159311532974243

**Slika 18.** Prikaz sortiranih podataka

Nakon što su podaci sortirani prema indeksima potrebno ih je spremati. Dobiveni podaci se spremaju u tablicu naziva Podaci. Tablica sadrži 9 stupaca te u svaki stupac ide po jedan atribut GPS zapisa koji se sastoji od 9 atributa. Podaci se spremaju na način da je definiran index za svaki atribut tako da se ide redoslijedom od indeksa 0 do indeksa 8 za prvi GPS zapis te se nakon izvršenog jednog niza za svaki idući zapis dodaje broj 9 budući da je toliko atributa koji ga sačinjavaju što je prikazano u kôdu na slici 19.



```

87     for (int i = 0; i < lineSplit.Length - 1; i += 9)
88     {
89         try
90         {
91             DateTime Vrijeme = UtcToRealTime(Convert.ToInt32(lineSplit[i]));
92             double MatchX = Convert.ToDouble(lineSplit[i + 1]);
93             double MatchY = Convert.ToDouble(lineSplit[i + 2]);
94             double OrgX = Convert.ToDouble(lineSplit[i + 3]);
95             double OrgY = Convert.ToDouble(lineSplit[i + 4]);
96             double OrgH = Convert.ToDouble(lineSplit[i + 6]);
97             double OrgS = Convert.ToDouble(lineSplit[i + 5]);
98             int LinkID = Convert.ToInt32(lineSplit[i + 7]);
99             double MatchD = Convert.ToDouble(lineSplit[i + 8]);
100        }
101        catch (Exception eex)
102        {
103            MessageBox.Show("For petlja u Form1_Loadu" + eex.Message);
104        }
105    }

```

**Slika 19.** Prikaz kôda za selektiranje atributa jednog zapisa

Kada su atributi jednog GPS zapisa selektirani i posloženi u varijable kojima pripadaju potrebno ih je spremiti u tablicu Podaci te će taj zapis sačinjavati prvi redak u toj istoj tablici. Ubacivanje u tablicu se vrši SQL naredbom „INSERT“. Podatke je potrebno prilagoditi formatu stupaca u bazi podataka. Primjerice, vrijeme je potrebno prilagoditi obliku kojeg podržava SQL pa je određeno da prilikom spremanja podatka u bazu podataka bude u formatu „godina-mjesec-dan sati:minute:sekunde. - Vrijeme.ToString("yyyy-MM-dd HH:mm:ss")“ što je vidljivo iz slike 20.

```

string naredba = "INSERT INTO Podaci " + "VALUES ('" + Vrijeme.ToString("yyyy-MM-dd HH:mm:ss") +
    "', geography::STPointFromText('POINT(" + toStr(MatchX) + " " + toStr(MatchY) + "'), 4326), " +
    "geography::STPointFromText('POINT(" + toStr(OrgX) +
    " " + toStr(OrgY) + "'), 4326)," + OrgS + "," + OrgH + "," + LinkID + "," + MatchD + "," + rutaID + ")";
cmd = new SqlCommand(naredba, sql);
cmd.ExecuteNonQuery();

```

**Slika 20.** Prikaz kôda za ubacivanje podataka u tablicu Podaci

Na slici 15 prikazan je način spremanja podataka u stupce PoložajNaKarti i StvarniPoložaj koji su prostornog tipa *geography*. Takav tip podatka u sebi sadrži dva atributa, a to su dvije koordinate, koje sačinjavaju točku na karti, i kao takav zahtjeva posebnu naredbu kako bi spremanje podataka bilo moguće. Naredba za spremanje u takav tip podatka glasi „geography::STPointFromText('POINT(" + toStr(MatchX) + " " + toStr(MatchY) + "'),4326')“ gdje su MatchX i MatchY dva atributa koja su prethodno selektirana.

Nakon što se izvrši jedan krug for petlje (slika 19) i podaci se sprema u tablicu naziva „Podaci“ ažurira se jedan redak tablice. Primjer jednog retka je prikazan na slici 21. Navedena iteracija se ponavlja sve dok petlja ne dođe do kraja datoteke.

IDZapis	Vrijeme	Polozaj\NaKarti	StvamiPolozaj	BrzinaKretanja	SmijerKretanja	IDLinka	UdajenostOdPrethodnogZapisa	RutaVozilaID
1	2014-01-10 16:21:48.000	0xE6100000010C5B7E580896E54640090000611A293040	0xE6100000010C3008F93A9AE546400D0000F31E293040	42	290	-391666	107	1

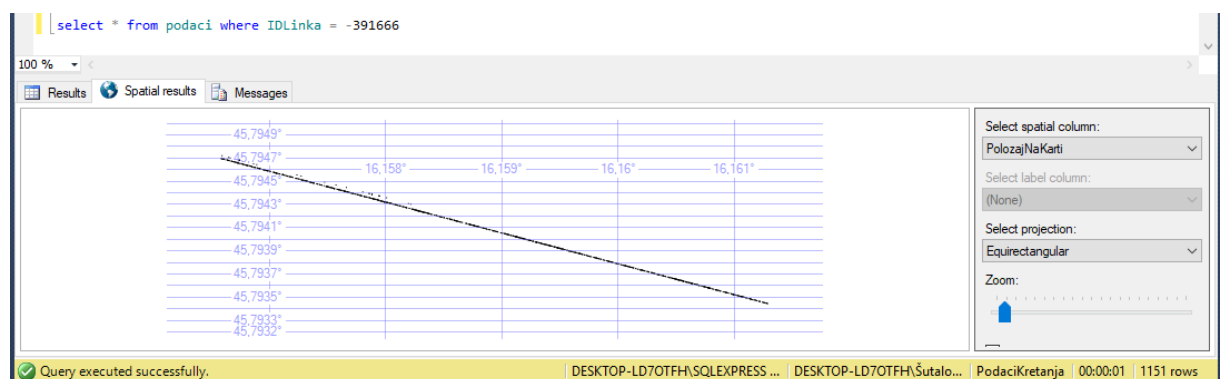
Slika 21. Jedan redak u tablici Podaci

Sljedeće slike prikazuju neke od primjera izvršavanja select naredbi i rezultate istih. Dva su načina prikazivanja podataka prvi je u obliku tablice dok drugi način naziva „Spatial results,, prikazuje položaje točaka na mreži karte.

IDZapis	Vrijeme	Polozaj\NaKarti	StvamiPolozaj	BrzinaKretanja	SmijerKretanja	IDLinka
1	2014-01-10 16:21:48.000	0xE6100000010C5B7E580896E54640090000611A293040	0xE6100000010C3008F93A9AE546400D0000F31E293040	42	290	-391666
2	285012	2014-01-10 16:21:57.000	0xE6100000010C2C4FFE7DA1E5464006008055C4283040	45	290	-391666
3	285013	2014-01-10 16:22:05.000	0xE6100000010C94E5A68DAC5E4640F6FF7F3071283040	49	290	-391666
4	285014	2014-01-10 16:22:13.000	0xE6100000010CDDDBEB78B8E54640F5FF7F8A17283040	52	290	-391666
5	321637	2012-12-07 11:32:56.000	0xE6100000010C50CF000EAE54640060000586F283040	87	290	-391666
6	321638	2012-12-07 11:32:56.000	0xE6100000010C50CF000EAE54640060000586F283040	87	290	-391666
7	321639	2012-12-07 11:33:00.000	0xE6100000010C318FA322B8E54640FAFFFC420283040	92	291	-391666
8	321640	2012-12-07 11:33:00.000	0xE6100000010C4B8146C89E54640080000C016283040	92	291	-391666
9	337655	2012-03-15 19:04:12.000	0xE6100000010C58745993AEE54640F6FFF4C6B283040	84	284	-391666
10	337656	2012-03-15 19:04:12.000	0xE6100000010C58745993AEE54640F6FFF4C6B283040	84	284	-391666

Slika 22. Prikaz podataka linku -391666

Na slici 22 prikazani su podaci i broj zapisa na linku -391666. Link sadrži 1151 zapis. Dok su na slici 23 prikazani isti podaci u obliku *Spatial results*. Vidljiva je prostorna raspodjela GPS podataka oko linka, koji je u digitalnoj karti reprezentiran geometrijskom linijom. GPS podaci se zbog svoje netočnosti nalaze u području oko linka ili na samom linku, te ih se pomoću metode *map matchinga* pridružuje odgovarajućem linku. U korištenim podacima, *map matching* je već unaprijed proveden.



Slika 23. Prikaz rezultata u obliku Spatial results

```
select * from podaci where BrzinaKretanja >200
```

IDZapis	Vrijeme	PolozajNaKarti	StvamiPolozaj	BrzinaKretanja	SmijerKretanja	IDLinka
382842	2012-08-01 13:05:36.000	0xE610000010CE10C15D6E9E646401C0000A575F22F40	0xE610000010C31D4C4ADF1E64640DFDFFFD76FF22F40	227	302	-217640
1538440	2012-10-18 16:35:31.000	0xE610000010C96130F5D25E84640FCFFF9FA4E62F40	0xE610000010C8A5B85B2AE54640E8FFFF1ADDEE62F40	244	55	214989
1584923	2012-06-15 14:34:29.000	0xE610000010C2AF63E5F1DE046400E00004872F82F40	0xE610000010C442F15FA12E04640110000EAB1F82F40	202	297	202867
1607452	2012-09-12 09:08:27.000	0xE610000010C33E5A01CC7DF464000000131CFC2F40	0xE610000010C68538435CADF46401200002E2EFC2F40	202	317	-202404

Query executed successfully. DESKTOP-LD7OTFH\SQLEXPRESS ... DESKTOP-LD7OTFH\Sutalo... PodaciKretanja | 00:00:00 | 4 rows

Slika 24. Prikaz zapisa u kojima je brzina veća od 200 km/h

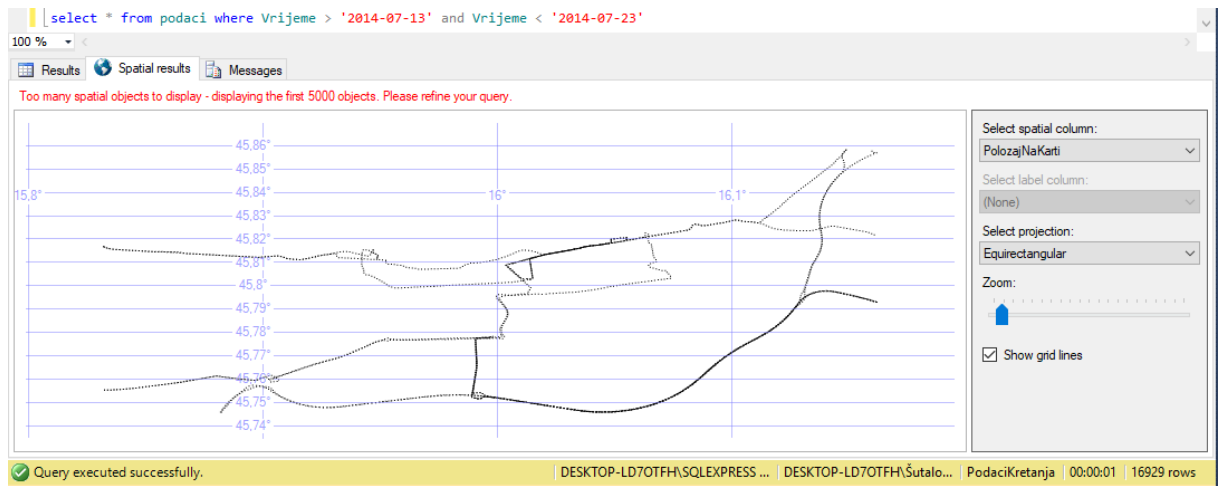
Slika 24 prikazuje zapise koji su dobiveni select naredbom. Tablica Podaci sadrži četiri zapisa koja imaju brzinu veću od 200 km/h te se svaki od zapisa nalazi na različitom linku. Ovakva analiza može se koristiti primjerice u policiji, za detekciju vozača koji su napravili određeni prometni prekršaj: prebrza vožnja, kretanje u zabranjenom smjeru, zabranjeno skretanje itd. Na slikama 25 i 26 prikazana je select naredba i podaci zapisani u vremenskom periodu od 13.07.2014 do 23.07.2014. U tom vremenskom periodu se nalazi 16929 GPS zapisa rasprostranjenih unutar područja Zagreba (prikazano je samo prvih 5000).

```
select * from podaci where Vrijeme > '2014-07-13' and Vrijeme < '2014-07-23'
```

IDZapis	Vrijeme	PolozajNaKarti	StvamiPolozaj	BrzinaKretanja	SmijerKretanja	IDLinka
537217	2014-07-16 07:36:26.000	0xE610000010CC842F63540E84640F0FFFF1D65E42F40	0xE610000010CC3A1472640E84640FFFFFDF66E42F40	29	194	217019
537218	2014-07-16 07:36:36.000	0xE610000010CB53D6ED22E846400E00005132E42F40	0xE610000010C0998A1C622E84640F6FFFA735E42F40	37	196	217020
537219	2014-07-16 07:36:58.000	0xE610000010C2917D3DD0EE846401A00009AADE32F40	0xE610000010C893F9A4B0FE846400900004EAE32F40	31	288	-216682
537220	2014-07-16 07:37:09.000	0xE610000010C90AD481317E846400F000020E0E32F40	0xE610000010C338144A817E84640FAFFF0E0DC32F40	29	284	-215972
537221	2014-07-16 07:37:28.000	0xE610000010C935925971FE84640FFFFF275DE22F40	0xE610000010C9B26E0D51FE84640160000905EE22F40	34	288	-215971
537222	2014-07-16 07:37:37.000	0xE610000010C0BF3CF628E84640180000DEB6E12F40	0xE610000010C743667572AE8464015000054B9E12F40	50	292	-215971
537223	2014-07-16 07:37:44.000	0xE610000010C1E6D5F0F32E84640EBFFFDD915E12F40	0xE610000010CD2008DB934E84640E7FFF51AE12F40	52	286	-215971
537224	2014-07-16 07:37:52.000	0xE610000010CC29CDF903CE84640A00002661E02F40	0xE610000010C4493CD3D40E846400300008868E02F40	44	286	-215970
537225	2014-07-16 07:38:02.000	0xE610000010C24FE4FE346E84640190000EDB8DF2F40	0xE610000010CE0F7CC7D49E846401000006BEDF2F40	41	286	-215969
537226	2014-07-16 07:38:12.000	0xE610000010CC2CFA2254EE846400700009E08DF2F40	0xE610000010C347806094FE84640BFFF2409DF2F40	33	274	-215968
537227	2014-07-16 07:38:22.000	0xE610000010C7B388BC74DE84640090000255DE2F40	0xE610000010CC565BA834EE846400D0000F854DE2F40	46	264	-215966

Query executed successfully. DESKTOP-LD7OTFH\SQLEXPRESS ... DESKTOP-LD7OTFH\Sutalo... PodaciKretanja | 00:00:01 | 16929 rows

Slika 25. Prikaz podataka u vremenskom periodu 13.07.2014.-23.07.2014.



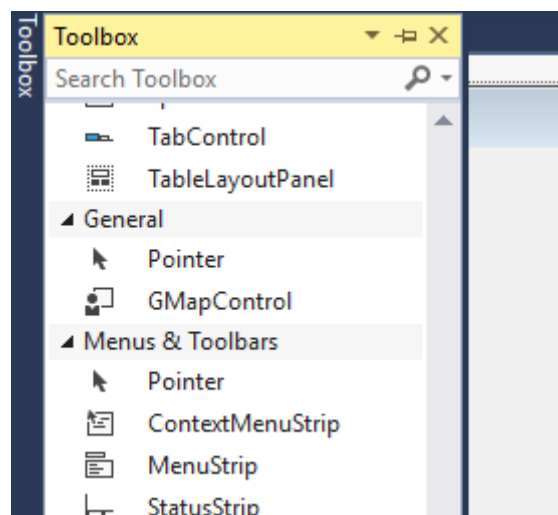
**Slika 26.** Prikaz rezultata s slike 25 u obliku Spatial results

## 5. Izrada sučelja za prikaz i analiza podataka

Nakon što su podaci obrađeni i spremljeni potrebno ih je vizualno prikazati. Za prikaz dobivenih podataka korištena je Google karta. Prije postupka prikaza podataka potrebno je instalirati NuGet paket, koji služi za proširenje funkcionalnosti programa i omogućuje pristup uslugama koje se temelje na digitalnim kartama te napisati kôd za implementaciju Google karte.

### 5.1. Instalacija NuGet paketa

NuGet paket je unaprijed definiran dio kôda koji mogu koristiti ostali programeri. Može se dodati projektu i koristiti njegove mogućnosti u bilo kojem dijelu projekta. NuGet se ponaša i kao platforma na kojoj se može kreirati i dijeliti kôd s ostalim korisnicima. Instalacija samog paketa je jednostavna, potrebno je otvoriti Nuget Package Manager i pronaći željenu verziju. U ovom radu je korišten Gmap.NET.Windows<sup>5</sup>. Nakon preuzimanja željenog paketa potrebno je dobivene alate spremiti u Toolbox kako bi se mogli koristiti [10]. Prikaz Toolboxa u kojem se nalazi spremljeni GMap alati je vidljiv na slici 27.

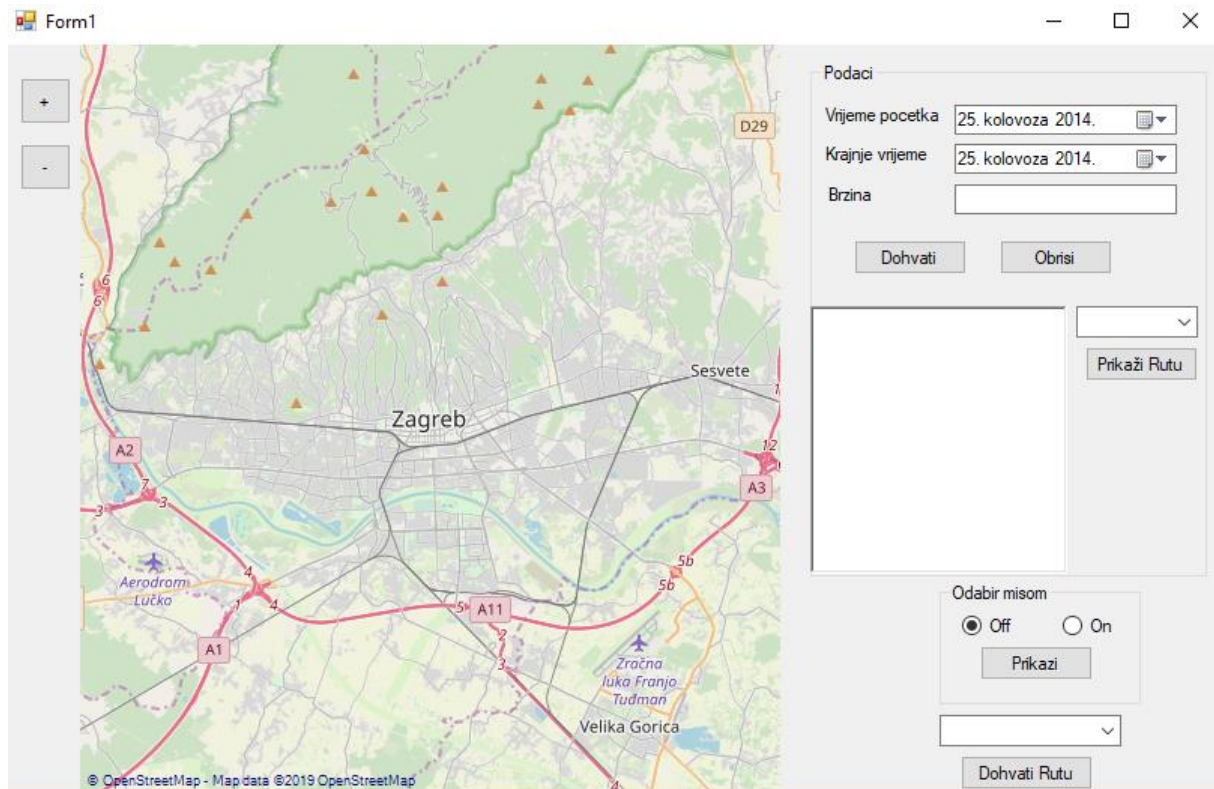


Slika 27. Prikaz Gmap.NET.Windows

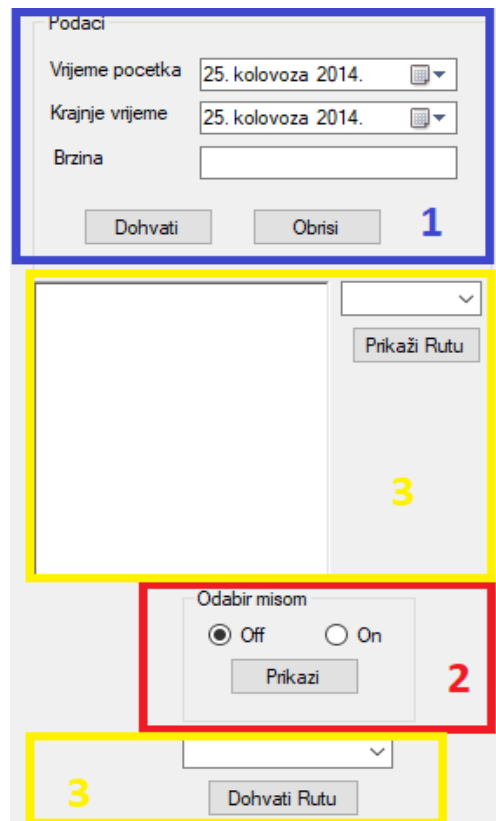
<sup>5</sup> Gmap.NET.Windows <https://www.nuget.org/packages/GMap.NET.Windows/>

## 5.2. Izrada sučelja

Nakon instalacije NuGet paketa može se početi s kreiranjem sučelja. Prvi korak je odabrati željene kontrole te ih smjestiti u u prozor aplikacije koja se izrađuje te odrediti nazive i funkciju koju će obavljati. Nakon odabira kontrola potrebno je napisati kôd kako bi izvršavale željenu funkciju unutar sučelja. Izgled korisničkog sučelja vidljiv je na slici 28.



Slika 28. Korisničko sučelje



**Slika 29.** Izgled kontrola

Za izradu rada su korištena tri načina filtriranja podataka koja su vidljiva na slici 29 a to su:

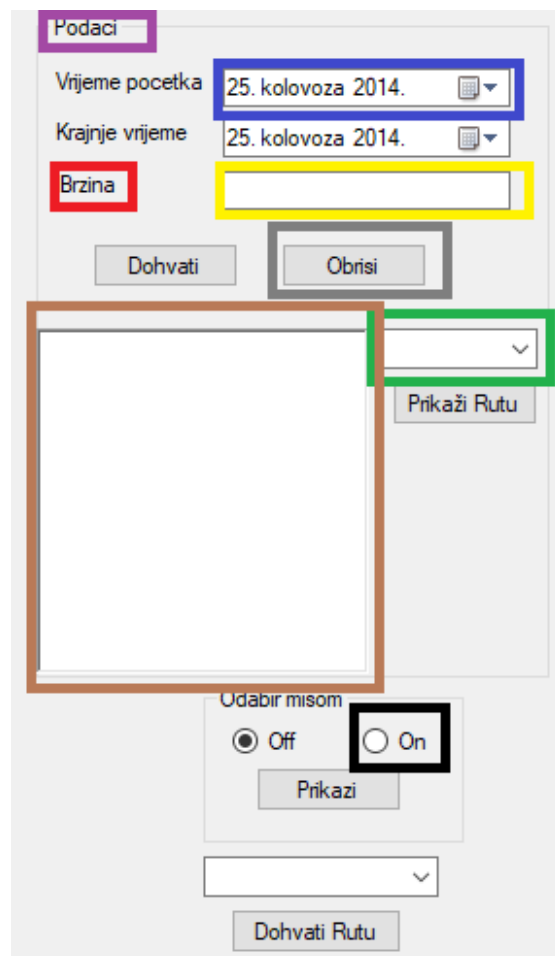
1. Prikaz podataka ovisno o vremenu i brzini – unutar ove funkcije moguće je odabrati početno i krajnje vrijeme promatranja te brzinu iznad koje su se vozila kretala te prikazati podatke unutar tih zadanih parametara,
2. Prikaz podataka označavanjem miša – prilikom označavanja željenog područja promatranja mišem moguće je dobiti vizualni prikaz svih zabilježenih točaka unutar tog poligona neovisno o vremenu u kojem su nastali,
3. Prikaz podataka ovisno o ruti – odabirom bilo koja od dva prethodna načina filtriranja moguće je dodatno filtrirati te podatke tako što se prilikom odabira jedne od ponuđenih ruta prikazuju oni podaci koji se nalaze samo na njoj.

Od ponuđenih kontrola unutar razvojnog okruženja Visual Studio za izradu rada su korišteni [11]:

- TextBox – koristi se za unos tekstualnih informacija, ne omogućava širenje na više redova, ali se to može omogućiti kroz svojstva kontrole,
- Label – koristi se za prikaz teksta. ne očekuje interakcija korisnika nego služi isključivo za prikaz teksta,

- Button – koristi se za pokretanje neke aplikacije ili dijela kôda,
- RadioButton – omogućava odabir samo jedne opcije odabirom kružića koji se nalazi ispred,
- GroupBox – omogućava sortiranje i grupiranje kontrola po funkcionalnijem rasporedu, daje mogućnost naziva GroupBoxa radi lakšeg manevriranja sučeljem,
- DateTimePicker – je kontrola koja omogućava odabir datuma u padajućem izborniku, moguće je prilagoditi željeni oblik zapisa datuma,
- ComboBox – kontrola koja omogućava prikaz podataka tako što koristi padajući izbornik,
- Splitter – omogućava razdvajanje sučelja na dvije cjeline koje imaju zasebne uloge,
- RichTextBox – služi za prikaz teksta. Tekst se može oblikovati po želji,
- MessageBox – može sadržati tekst, tipku i simbole koji informiraju i upućuju korisnika.

Na slici 30 su prikazane i bojom označene spomenute kontrole. GroupBox je označen ljubičastom bojom, DateTimePicker plavom, Label crvenom, TextBox žutom, sivom je označen Button, zelenom ComboBox, smeđom RichTextBox i crnom RadioButton.



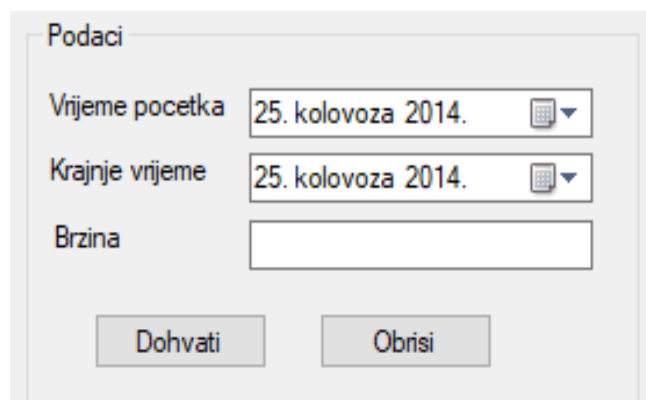
**Slika 30.** Korištene kontrole



### 5.2.1. Prikaz podataka ovisno o vremenu i brzini

Mogućnosti koje se nude u ovom dijelu sučelja na slici 31 pod nazivom „Podaci“ za prikaz podataka na karti su:

- Odabir početnog vremena promatranja,
- Odabir krajnjeg vremena promatranja,
- Upisivanje minimalne brzine vožnje,
- Gumb „Dohvati“ koji prilikom pokretanja postavlja markere na kartu,
- Gumb „Obrisi“ koji uklanja postavljene markere.



Podaci

Vrijeme pocetka 25. kolovoza 2014.

Krajnje vrijeme 25. kolovoza 2014.

Brzina

Dohvati Obrisi

**Slika 31.** Kontrole sučelja za prikaz podataka ovisno o vremenu i brzini

U pozadini programa iza svake kontrole nalazi se kôd koji omogućava funkcioniranje i obavljanje zadanih parametara i funkcija. Kôd za ovaj dio sučelja prikazan je na slikama 32 i 33.

```

134 private void Btn_Dohvati_Podatke_Click(object sender, EventArgs e)
135 {
136     try
137     {
138         if (Txt_Brzina.Text == string.Empty)
139         {
140             MessageBox.Show("Potrebno je unijeti sve podatke!");
141         }
142         else
143         {
144             DateTime vrijemePocetka = Convert.ToDateTime(dateTimePicker1.Text); //DateTime
145             DateTime vrijemeZavrsetka = Convert.ToDateTime(dateTimePicker2.Text);
146             double brzina = Convert.ToDouble(Txt_Brzina.Text);
147
148             string naredba1 = "Select RutaVozilaID, PolozajNaKarti from Podaci where Vrijeme between '"
149                 + vrijemePocetka.ToString("yyyy-MM-dd") + "' and '" + vrijemeZavrsetka.ToString("yyyy-MM-dd")
150                 + "' and BrzinaKretanja >= '" + brzina.ToString() + "'";
151             cmd = new SqlCommand(naredba1, sql);
152             myReader = cmd.ExecuteReader();
153             Rch_Txt_Ispis.Text = string.Empty;
154             map.Overlays.Clear();
155             List<string> lisat_IDeva = new List<string>();
156         }
157     }
158 }
159 catch(Exception ex)
160 {
161     MessageBox.Show(ex.Message);
162 }
163 }
164

```

**Slika 32.** Prikaz prvog dijela kôda za dio sučelja naziva „Podaci“

Služeći se s *if* naredbom (redak138) provjerava je li unesen izos minimalne brzine. Ako brzina nije unesena pojavljuje se MessageBox s porukom „Potrebno je unijeti sve podatke!“. U slučaju da je brzina unesena vrijednosti brzine i datuma se konvertiraju u tipove podataka double i DateTime redak (144-146) Nakon konvertiranja podataka pristupa se SQL upitu koji povlači sve podatke unutar zadanih vrijednosti iz baze podataka redak (148-150).

```

163 double x = Convert.ToDouble(value[value.Length - 3]);
164 string broj = x.ToString();
165 x = x / Math.Pow(10, (broj.Length - 2));
166 double y = Convert.ToDouble(value[value.Length - 2]);
167 broj = y.ToString();
168 y = y / Math.Pow(10, (broj.Length - 2));
169
170 PointLatLng tocka = new PointLatLng(y, x);
171
172 GMapOverlay marker = new GMapOverlay("marker");
173 GMap.NET.WindowsForms.GMapMarker markerType = new GMarkerGoogle(tocka, GMarkerGoogleType.blue_small);
174
175 marker.Markers.Add(markerType);
176 map.Overlays.Add(marker);

```

**Slika 33.** Prikaz drugog dijela kôda za dio sučelja naziva „Podaci“

Svaka dohvaćena koordinata iz stupca „PolozajNaKarti“ zasebno se razdvaja i sprema u varijable x i y te se stvara točka tipa PointLatLng kojoj se dodjeljuju spomenute varijable te se potom stvara marker na karti. PointLatLng je klasa iz paketa biblioteke Gmap koja služi za spremanje koordinata. Postupak se ponavlja sve dok se ne obradi i zadnja koordinata koja je dohvaćena iz baze. Gumb „Obriši“ omogućuje uklanjanje svih postavljenih markera kako bi svaka iduća radnja bila preglednija i jasnija korisniku. Prikaz kôda koji se izvršava pritiskom na gumb prikazan je na slici 34.

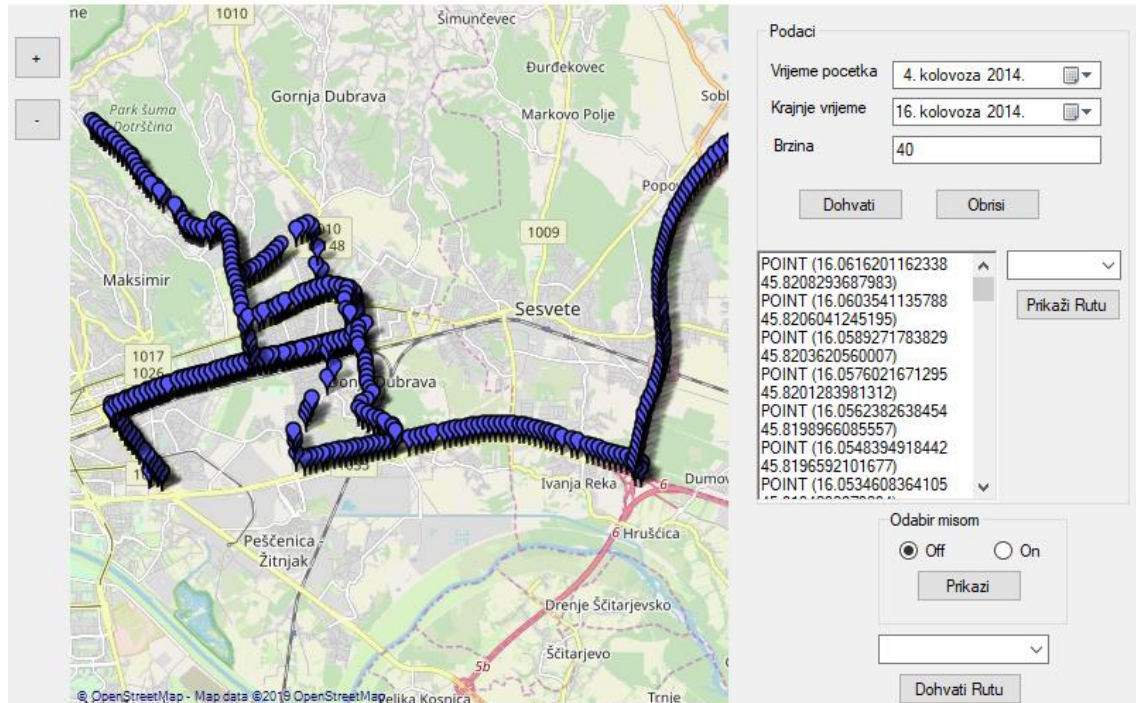
```

214 private void Btn_Clear_Map_Click(object sender, EventArgs e)
215 {
216     Rch_Txt_Ispis.Text = string.Empty;
217     map.Overlays.Clear();
218 }

```

**Slika 34.** Prikaz kôda za brisanje markera

Slika 35 prikazuje iscrtavanje markera na mapi. Prikazani podaci se nalaze u vremenskom periodu između 4. kolovoza 2014 i 16. kolovoza 2014, a minimalna brzina je 40 km/h. Na karti je prikazano 11915 označenih markera.

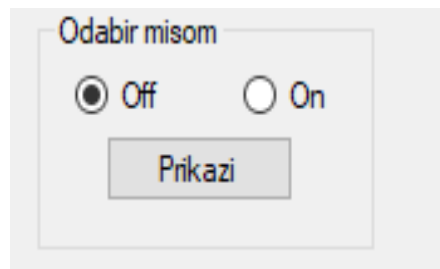


**Slika 35.** Prikaz markera ruta vozila na karti.

## 5.2.2. Prikaz podataka označavanjem miša

Drugi oblik selektiranja podataka za prikaz obavlja se označavanjem željenog područja promatranja na karti. Ovaj dio sučelja sadrži tri kontrole koje su ujedno i prikazane na slici 36:

- Kontrola za gašenje funkcije označavanja „Off“,
- Kontrola za paljenje funkcije označavanja „On“,
- Gumb „Prikazi“ koji dobavlja podatke iz označenog područja .



Slika 36. Prikaz kontrola za odabir mišom.

Na slici 37 je moguće vidjeti kôd koj se izvršava prilikom pritiska na gumb „Prikazi“.

```
310 private void btnPrikaz_Click(object sender, EventArgs e)
311 {
312     try
313     {
314         if (RB_Rect_On.Checked == true)
315         {
316             PointLatLng kockaLijevoDolje = new PointLatLng();
317             kockaLijevoDolje.Lng = LocationX1Y1.Lng;
318             kockaLijevoDolje.Lat = LocationX2Y2.Lat;
319             PointLatLng kockaDesnoGore = new PointLatLng();
320             kockaDesnoGore.Lng = LocationX2Y2.Lng;
321             kockaDesnoGore.Lat = LocationX1Y1.Lat;
322             bool longitudeChange = kockaLijevoDolje.Lng > kockaDesnoGore.Lng;
323             bool latitudeChange = kockaLijevoDolje.Lat > kockaDesnoGore.Lat;
324             PointLatLng pocetnaTockaKocke = LocationX1Y1;
325             kockaLijevoDolje_Lat = kockaLijevoDolje.Lat.ToString().Replace(',', '.');
326             kockaLijevoDolje_Lng = kockaLijevoDolje.Lng.ToString().Replace(',', '.');
327             kockaDesnoGore_Lat = kockaDesnoGore.Lat.ToString().Replace(',', '.');
328             kockaDesnoGore_Lng = kockaDesnoGore.Lng.ToString().Replace(',', '.');
329             string kocka = "select PolozajNaKarti.Lat as Latitude, PolozajNaKarti.Long as Longitude," +
330                 " RutaVozilaID from Podaci where PolozajNaKarti.Lat >= " + kockaLijevoDolje_Lat +
331                 " and PolozajNaKarti.Lat <= " + kockaDesnoGore_Lat + " and PolozajNaKarti.Long >= " +
332                 kockaLijevoDolje_Lng + " and PolozajNaKarti.Long <= " + kockaDesnoGore_Lng;
333             cmd = new SqlCommand(kocka, sql);
334             SqlDataReader myReader2 = cmd.ExecuteReader();
335             map.Overlays.Clear();
336         }
337     }
338     catch (Exception ex)
339     {
340         MessageBox.Show("btnPrikaz \n" + ex.Message);
341     }
342 }
```

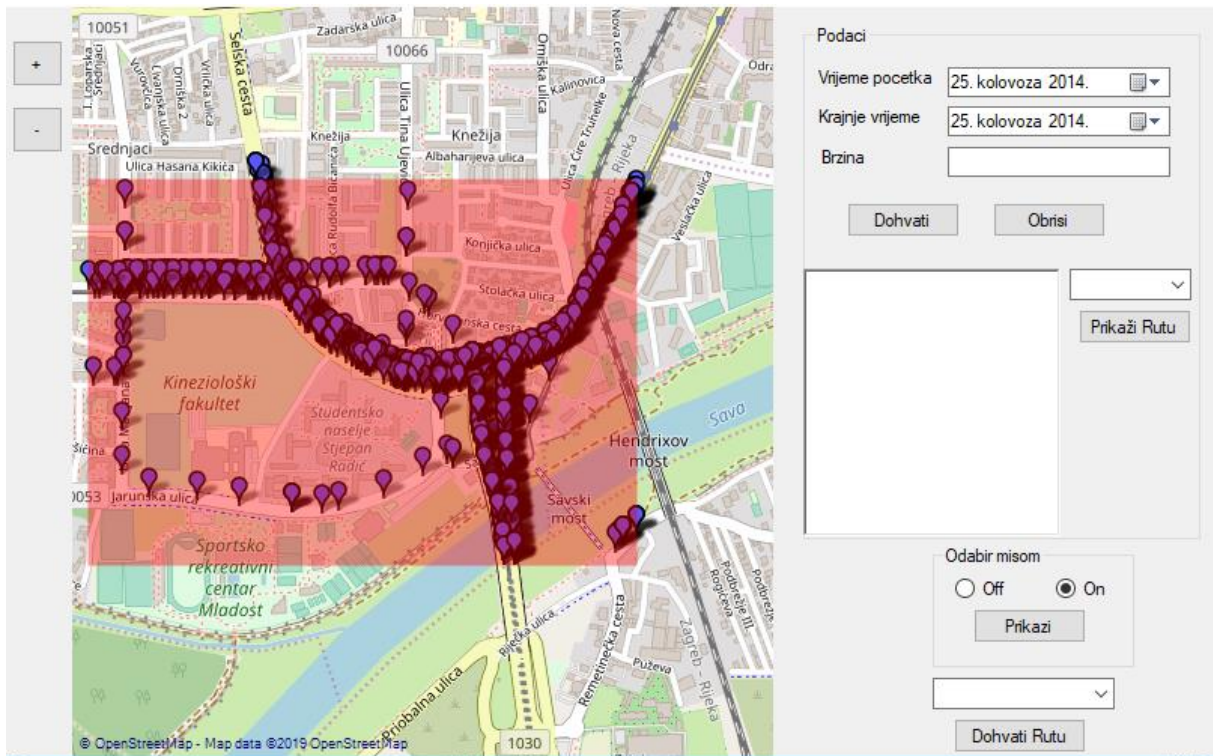
Slika 37. Prikaz kôda za gumb „Prikazi“

Nakon označavanja poligona na karti obodne točke poligona koje su prethodno dobivene se spremaju u varijable `kockaLijevoDolje_Lat`, `kockaLijevoDolje_Lng`, `kockaDesnoGore_Lat` i `kockaDesnoGore_Lng`. Potom se ti podaci koriste za formiranje SQL upita nad bazom gdje se pristupa svim podacima, iz stupca `PolozajNaKarti`, koji pripadaju svojim koordinatama unutar toga poligona (redak 342-346).

```
359 while (myReader2.Read())
360 {
361     string lineLat = myReader2["Latitude"].ToString();
362     string lineLng = myReader2["Longitude"].ToString();
363     string lineRutaVozilaID = myReader2["RutaVozilaID"].ToString();
364
365     double x = Convert.ToDouble(lineLng);
366     double y = Convert.ToDouble(lineLat);
367     //int ID = Convert.ToInt32(lineIDLinka);
368
369     PointLatLng tocka = new PointLatLng(y, x);
370
371     GMapOverlay marker2 = new GMapOverlay("marker2");
372     GMap.NET.WindowsForms.GMapMarker markerType2 = new GMarkerGoogle(tocka, GMarkerGoogleType.blue_small);
373
374     marker2.Markers.Add(markerType2);
375     map.Overlays.Add(marker2);
376
377     Cmb_Lista.Items.Add(lineRutaVozilaID);
378 }
379
```

**Slika 38.** kôd za postavljanje markera nakon označavanja mišem

Na slici 38 je prikazan nastavak kôda sa slike 37. Svaki od dobivenih podataka ima svoje koordinate na temelju kojih se određuje mjesto postavljanja markera. Koordinate se dijele na x i y koordinatu koje se konvertiraju u tip *double* (redak 365-366) te ih se koristi za kreiranje točke tipa *PointLatLng* koja se u konačnici postavlja kao marker (redak 369-375). U nastavku na slici 39 je prikaz poligona na karti s označenim markerima.

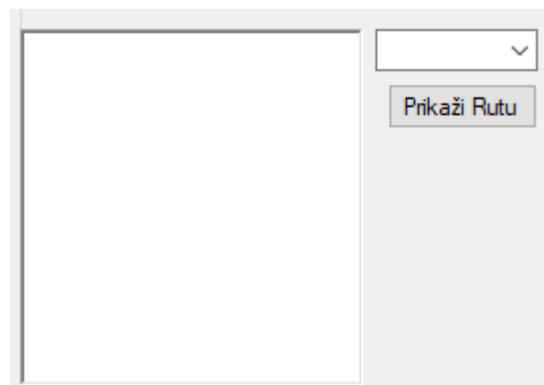


Slika 39. Prikaz karte s markerima unutar odabranog poligona

### 5.2.3. Prikaz podataka ovisno o ruti

Posljednji način odabira podataka koji se žele prikazati na karti je pomoću izbora rute na kojoj su zabiježeni GPS podaci. Prije odabira željene rute potrebno je označiti područje ili odabrati vremenski period i minimalnu brzinu kretanja kako bi se dobio popis ruta unutar zadanih parametara.

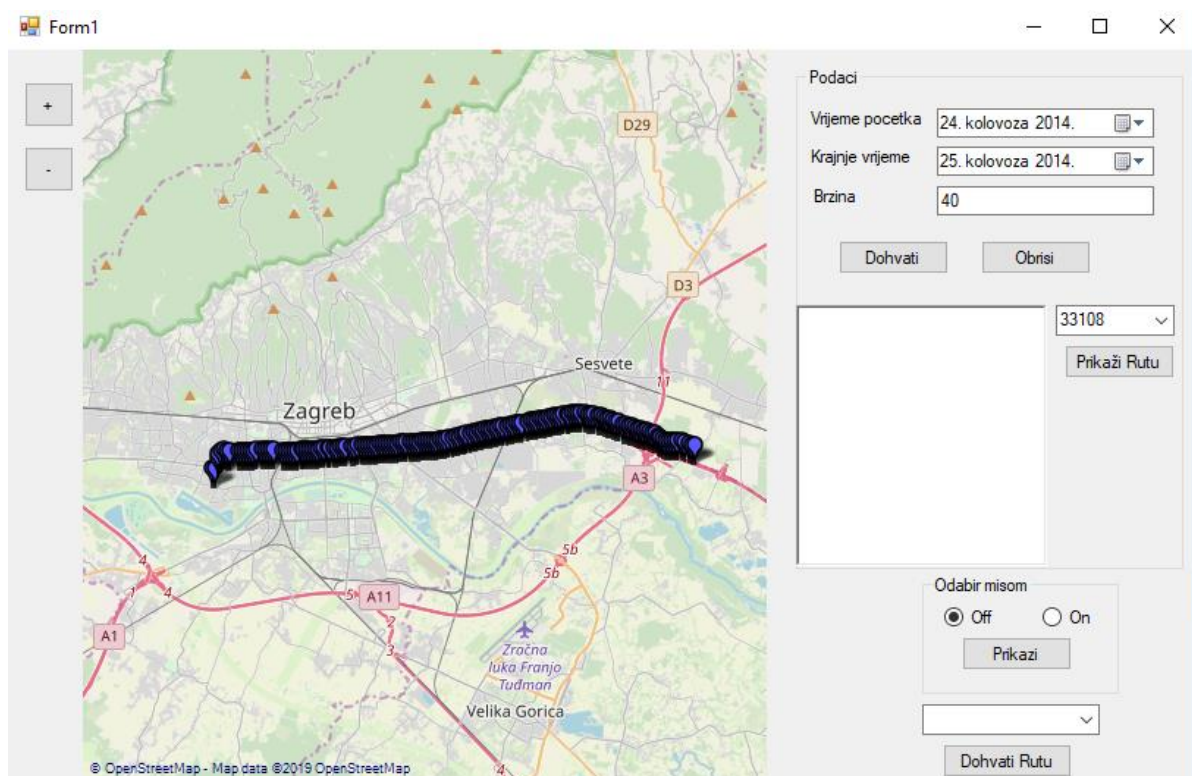
Na slici 40 prikazan je izgled dijela sučelja u kojem se prilikom klika na padajući izbornik prikažu rute. S lijeve strane nalazi se *RichTextBox* u kojem se ispisuju koordinate svake dohvaćene točke.



Slika 40. Prikaz izgleda sučelja za odabir rute



Nakon odabira željenih parametara i dohvaćanja podataka SQL select naredbom, dohvaća se i ID rute vozila što je vidljivo iz naredbe „ `string naredba1 = "Select RutaVozilaID, PolozajNaKarti from Podaci"` te se svaki ID sprema u padajući izbornik. Prilikom odabira željene rute i pritiska na gumb „Prikazi Rutu“ iscrtavaju se samo markeri koji se nalaze na odabranoj ruti. Prikaz primjera nalazi se na slici 41 gdje se prikazani markeri nalaze na ruti naziva „33108“. Ruta koja je označena na karti ima 155 zapisa, duljina rute je 19912 m, a ukupno vrijeme putovanja je 26 minuta i 34 sekunde.



**Slika 41.** Prikaz jedne rute

## 6. Analiza podataka

U ovom poglavlju napravljena je analiza podataka koji su korišteni u izradi aplikacije. Unutar tablice Podaci nalazi se 2407394 GPS zapisa (slika 42) dok su u tablici RutaVozila spremljene 34782 rute (slika 43) na kojima su raspoređeni spomenuti GPS zapisi. Naredne slike i dijagrami prikazuju karakteristike podataka s obzirom na broj zapisa unutar intervala brzine, broj zapisa ovisno o godinama te rutu s najviše zapisa.

IDZapisa	Vrijeme	Polozaj/NaKarti	StvamiPolozaj	BrzinaKretanja	SmijerKretanja	IDLinka	UdajlenostC
1	285011	2014-01-10 16:21:48.000	0xE610000010C5B7E580896E5464009000611A293040	0xE610000010C3008F93A9AE546400D0000F31E293040	42	290	-391666 107
2	285012	2014-01-10 16:21:57.000	0xE610000010C2C4FFE7DA1E5464006008055C4283040	0xE610000010CE0C9B479A5E54640F4FFFA3C8283040	45	290	-391666 109
3	285013	2014-01-10 16:22:05.000	0xE610000010C94E5A68DACE54640F6FF7F3071283040	0xE610000010CDD377CE7B0E54640F8FFFD875283040	49	290	-391666 105
4	285014	2014-01-10 16:22:13.000	0xE610000010CDD0BEB78B8E54640F5FF7F8A17283040	0xE610000010C438E0729BD0E54640080080A31C283040	52	290	-391666 113
5	285015	2014-01-10 16:22:20.000	0xE610000010C8A284332C3E54640F3FFF02C5273040	0xE610000010C1F24F528C8E546400200049CA273040	54	290	-391666 104
6	285016	2014-01-10 16:22:27.000	0xE610000010CDE5A7E22CEE54640FCFFFCF70273040	0xE610000010C1B66CE9D2E54640F3FFFE875273040	54	290	-391666 106
7	285017	2014-01-10 16:22:34.000	0xE610000010C38685D8DD8E54640010080A51B273040	0xE610000010CDD8343ABDDE54640FAFFFA720273040	55	288	-391664 107
8	285018	2014-01-10 16:22:41.000	0xE610000010C39956F0E2E5464006000040C5263040	0xE610000010C87F65CD7E7E54640060000FC9263040	55	288	-391664 108
9	285019	2014-01-10 16:22:48.000	0xE610000010C514B68EDECE54640FAFFFA696E263040	0xE610000010C73B27203F2E54640FEFFFA72263040	54	286	-391664 108
10	285020	2014-01-10 16:22:55.000	0xE610000010C43418D4DF6E54640F8FFFB219263040	0xE610000010C3FB6CD8AFBE54640F8FFFA11E263040	54	286	-391664 106

Slika 42. Prikaz podataka iz tablice Podaci

NazivRute	
1	Ruta broj: 1
2	Ruta broj: 10
3	Ruta broj: 100
4	Ruta broj: 1000
5	Ruta broj: 10000
6	Ruta broj: 10001
7	Ruta broj: 10002
8	Ruta broj: 10003
9	Ruta broj: 10004
10	Ruta broj: 10005
11	Ruta broj: 10006

Slika 43. Prikaz podataka iz tablice RutaVozila

RutaVozilaID	broj zapisa
1	35645
2	35652
3	35651
4	9477
5	9512
6	8344
7	35644
8	8619
9	8098
10	6125
11	5771

Slika 44. Prikaz 11 ruta s najviše zapisa



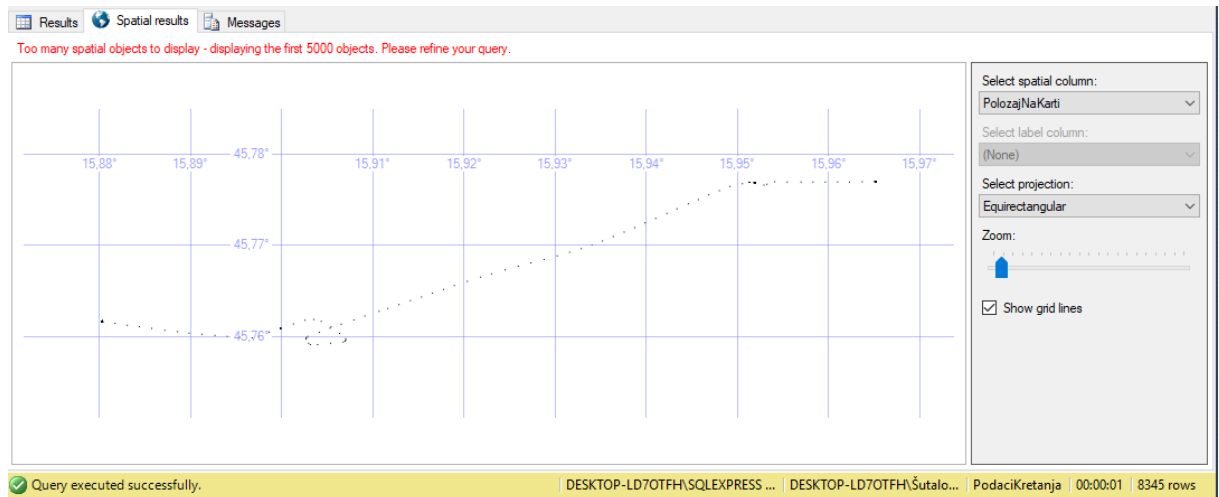
Na slici 44 je prikazano 11 ruta koje imaju najviše GPS zapisa. Od kojih je najveća s 8345 zapisa dok 11-ta s 539 zapisa, a najmanja ruta ima jedan (1) zapis.

select \* from Podaci where RutaVozilaID = 35645

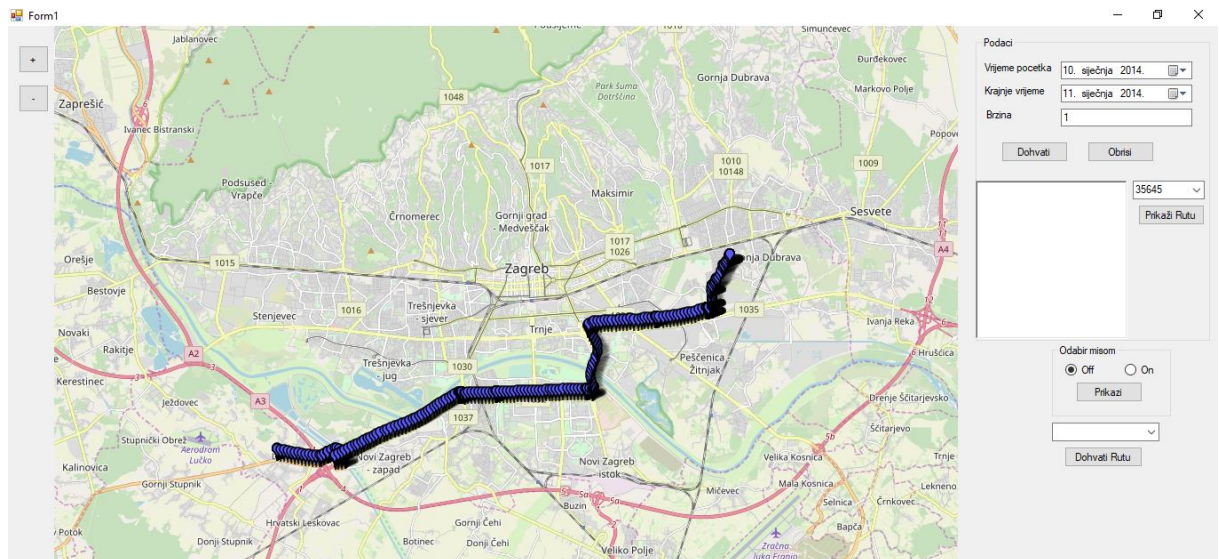
IDZapis	Vrijeme	PolozajNaKarti	StvamiPolozaj	BrzinaKretanja	SmijerKretanja	IDLinka
1	2014-01-10 14:29:50.000	0xE6100000010C2BB209F380E1464013000044BBC22F40	0xE6100000010C5D3319BC80E14640F3FFFF4BEC22F40	8	204	-200034
2	2291742	2014-01-10 14:29:57.000	0xE6100000010C8EB118B07CE14640FFFFFFA6B2C22F40	4	190	-200034
3	2291743	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034
4	2291744	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034
5	2291745	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034
6	2291746	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034
7	2291747	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034
8	2291748	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034
9	2291749	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034
10	2291750	2014-01-10 14:29:57.000	0xE6100000010CBEDA104A7CE14640100000F3B1C22F40	4	190	-200034

Query executed successfully. DESKTOP-LD70TFH\SQLEXPRESS ... DESKTOP-LD70TFH\Sutalo... PodaciKretanja 00:00:01 8345 rows

Slika 45. Prikaz rute 35645 s najviše GPS zapisa

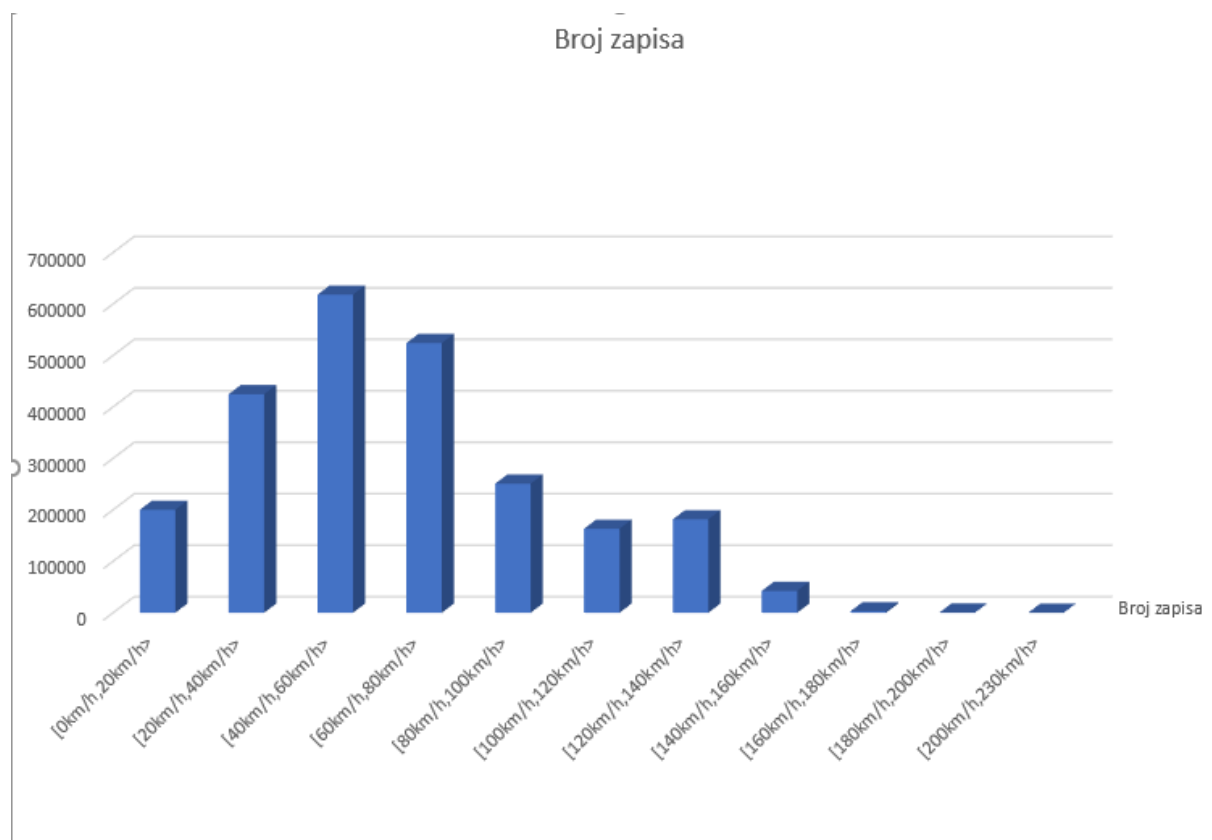


Slika 46. Prikaz rute 35645 u obliku Spatial results

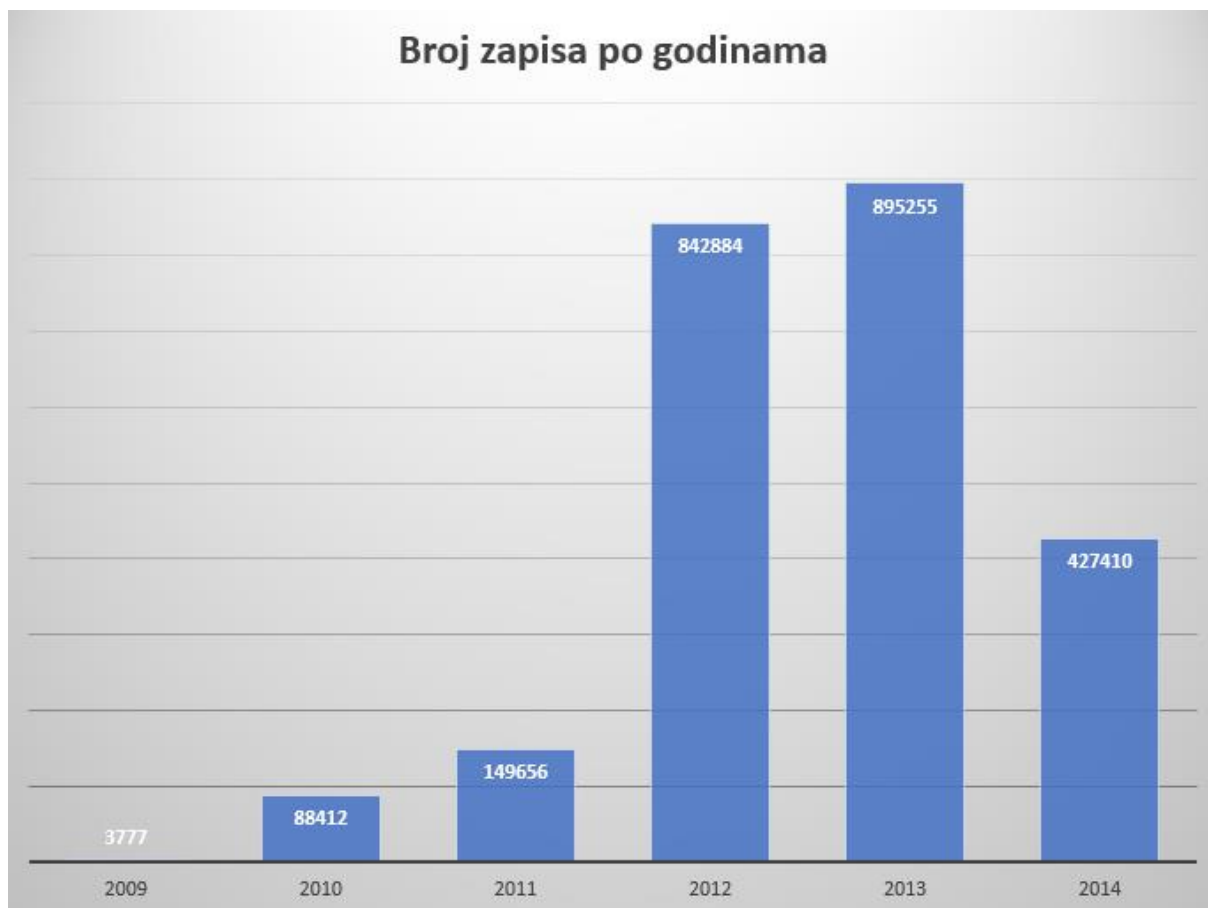


Slike 45 i 46 prikazuju rutu 35645 koja sadrži 8345 GPS zapisa što je čini rutu s najviše zapisa koja je ujedno i najduža s 47335 m, a slika 47 prikazuje rutu 35645 na karti.

Slike 47 i 48 prikazuju statistiku prikupljanja podataka. Na slici 47 vidljiv je broj zapisa ovisno o brzinama koje su zabilježene te su podaci grupirani prema intervalima istih, dok je na slici 48 prikazan broj prikupljenih GPS grupiran po godinama. Sukladno tome na slici 47 vidljivo je da je najviše GPS zapisa u intervalu brzine između 40 km/h i 60 km/h, a na slici 48 je najviše zapisa zabilježeno 2013. godine.



**Slika 48.** Prikaz broja GPS zapisa ovisno o intervalu brzine kretanja



**Slika 49.** Prikaz broja GPS zapisa ovisno o godini

## 7. Zaključak

Razvijanjem društva i tehnologija raste i potreba za prikupljanjem i obrađivanjem prometnih podataka kako bi se mogle uočiti kako trenutne potrebe tako i potrebe koje dolaze sa sve većim priljevom vozila na cestama. U završnom su radu obrađeni podaci o kretanjima vozila koji su nastali prilikom provedbe projekta SORDITO. To su GPS podaci koji sadrže podatke kao što su geografski položaj, brzina kretanja i udaljenost od prošlog zapisa. Za spomenute podatke izrađena je prostorno vremenska baza podataka koja je usklađena s korištenjem pravila pretvorbe normalnim formama. Nastavno, podatke je prije spremanja bilo potrebno prilagoditi tipovima podataka koje SQL podržava kako bi bili uspješno pohranjeni. Podaci su obrađeni i prilagođeni unutar razvojnog okruženja Visual Studio programskim jezikom C#. Kako bi podaci bili vidljivi krajnjem korisniku potrebno ih je vizualizirati. Izrađeno je sučelje koje ima odabrane kontrole za prikaz podataka na mapi pa je podatke moguće klasificirati prema vremenu u kojem su nastali i brzini kretanja, odabirom miša te odabirom željene rute. Ovakav način pristupa i obrade podataka i njihova vizualizacija omogućavaju korisniku da uoči trenutne probleme na prometnicama i primjeni odgovarajuću metodologiju rješavanja kako bi ih uklonio. Primjerice, analiza prekoračenja brzina na nekoj prometnici ili kritična točka na kojoj se događa veliki broj prometnih prekršaja ovisno o brzini kretanja vozila posebice u naseljenim dijelovima. Sukladno tome mogu se postaviti senzori za ograničenja ili nova regulacija prometa kako bi se smanjio broj stradalih u prometu.

## Popis literature

- [1] Rožić L, Carić T, Matulin M, Ravlić M, Fosin J, Milošević A, et al. Tehnički izvještaj rezultata eksperimentalnog razvoja projekta SORDITO. Sveučilište u Zagrebu, Fakultet prometnih Znanosti; 2016.
- [2] T. Erdelić, S. Vrbančić, L. Rožić, “A model of speed profiles for urban road networks using g-means clustering,” 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1081–1086, May 2015.
- [3] Ahmed El – Rabbany (2002.) Introduction to GPS: The Global Positioning System
- [4] Autorizirana predavanja iz kolegija Baze podataka: [https://moodle.srce.hr/2018-2019/pluginfile.php/1760048/mod\\_resource/content/5/Predavanja.pdf](https://moodle.srce.hr/2018-2019/pluginfile.php/1760048/mod_resource/content/5/Predavanja.pdf) (pristupljeno 23.8.2019.)
- [5] Autorizirana predavanja FER- Napredni modeli i baze podataka : Vremenske baze podataka [https://www.fer.unizg.hr/download/repository/1.\\_O\\_predmetu.pdf](https://www.fer.unizg.hr/download/repository/1._O_predmetu.pdf) (pristupljeno 24.8.2019.)
- [6] Norman, P. W. (2000.) Spatio-Temporal Databases: Contentions, Components and Consolidation
- [7] Autorizirana predavanja iz kolegija Baze podataka: [https://moodle.srce.hr/2018-2019/pluginfile.php/1760050/mod\\_resource/content/2/Vje%C5%BEba14.pdf](https://moodle.srce.hr/2018-2019/pluginfile.php/1760050/mod_resource/content/2/Vje%C5%BEba14.pdf) (pristupljeno 01.09.2019.)
- [8] Autorizirana predavanja iz kolegija Baze podataka: [https://moodle.srce.hr/2018-2019/pluginfile.php/1760026/mod\\_resource/content/7/11.tjedan.pdf](https://moodle.srce.hr/2018-2019/pluginfile.php/1760026/mod_resource/content/7/11.tjedan.pdf)
- [9] Margaret Rouse, TechTarget: Search App Architecture <https://searcharchitecture.techtarget.com/definition/object-oriented-programming-OOP> (pristupljeno 25.8.2019.)
- [10] Faisal Pathan, DZone <https://dzone.com/articles/install-nuget-packages-in-visual-studio> (pristupljeno 26.8.2019)
- [11] Saša Fajković, Uvod u programiranje kroz jezik Visual Basic <http://carpediem.hr/PublikacijeCarpeDiem/Publikacije/Visual%20Basic.pdf> (pristupljeno 26.8.2019.)

## Popis slika

- Slika 1.** Prikaz poligona unutar kojeg su prikupljeni podaci
- Slika 2.** Schema prikupljanja podataka
- Slika 3.** Prikaz GPS zapisa u jednoj ruti vozila
- Slika 4.** Prikaz neobrađenih podataka
- Slika 5.** Opis podataka jednog GPS zapisa
- Slika 6.** Prikaz proširenja DBMS-a
- Slika 7.** Prikaz insert naredbe za spremanje podataka u tip geography
- Slika 8.** Izgled podataka spremljenjih u tip geography
- Slika 9.** Primjer selecta za dobavljanje koordinata u zasebne stupce
- Slika 10.** Prikaz dijagrama entiteta
- Slika 11.** Prikaz skupova R i S
- Slika 12.** Prikaz ER dijagrama
- Slika 13.** Skripta za kreiranje baze podataka „Podaci“
- Slika 14.** Grafički prikaz povezanosti tablica
- Slika 15.** Izgled sučelja Microsoft Visual Studia
- Slika 16.** Kôd za spajanje na bazu i čitanje tekstualne datoteke
- Slika 17.** Ubacivanje u tablicu RutaVozila
- Slika 18.** Prikaz sortiranih podataka
- Slika 19.** Prikaz kôda za selektiranje atributa jednog zapisa
- Slika 20.** Prikaz kôda za ubacivanje podataka u tablicu Podaci
- Slika 21.** Jedan redak u tablici Podaci
- Slika 22.** Prikaz podataka linku -391666
- Slika 23.** Prikaz rezultata u obliku Spatial results
- Slika 24.** Prikaz zapisa u kojima je brzina veća od 200 km/h
- Slika 25.** Prikaz podataka u vremenskom periodu 13.07.2014.-23.07.2014.
- Slika 26.** Prikaz rezultata s slike 25 u obliku Spatial results
- Slika 27.** Prikaz Gmap.NET.Windows
- Slika 28.** Korisničko sučelje
- Slika 29.** Izgled kontrola
- Slika 30.** Korištene kontrole
- Slika 31.** Kontrole sučelja za prikaz podataka ovisno o vremenu i brzini

- Slika 32.** Prikaz prvog dijela kôda za dio sučelja naziva „Podaci“
- Slika 33.** Prikaz drugog dijela kôda za dio sučelja naziva „Podaci“
- Slika 34.** Prikaz kôda za brisanje markera
- Slika 35.** Prikaz markera ruta vozila na karti.
- Slika 36.** Prikaz kontrola za odabir mišom
- Slika 37.** Prikaz kôda za gumb „Prikazi“
- Slika 38.** Kôd za postavljanje markera nakon označavanja mišem
- Slika 39.** Prikaz karte s markerima unutar odabranog poligona
- Slika 40.** Prikaz izgleda sučelja za odabir rute
- Slika 41.** Prikaz jedne rute
- Slika 42.** Prikaz podataka iz tablice Podaci
- Slika 43.** Prikaz podataka iz tablice RutaVozila
- Slika 44.** Prikaz 11 ruta s najviše zapisa
- Slika 45.** Prikaz rute 35645 s najviše GPS zapisa
- Slika 46.** Prikaz rute 35645 u obliku Spatial results
- Slika 47.** Prikaz rute 35645 na karti
- Slika 48.** Prikaz broja GPS zapisa ovisno o intervalu brzine kretanja
- Slika 49.** Prikaz broja GPS zapisa ovisno o godini





Sveučilište u Zagrebu  
Fakultet prometnih znanosti  
10000 Zagreb  
Vukelićeva 4

### IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj \_\_\_\_\_ završni rad

isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu \_\_\_\_\_ završnog rada

pod naslovom **Prostorno-vremenske baze podataka o kretanju vozila**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 3.9.2019

Student/ica:

Bojka Šušić  
(potpis)