Web aplikacije za prikaz prometnih zagušenja na cestovnoj mreži

Pavčec, Filip

Undergraduate thesis / Završni rad

2023

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://urn.nsk.hr/urn:nbn:hr:119:906668

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-12**

Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Filip Pavčec

Web aplikacija za prikaz prometnih zagušenja na cestovnoj mreži

Završni rad

SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Filip Pavčec

Web aplikacija za prikaz prometnih zagušenja na cestovnoj mreži

Web appliaction for visualizing traffic congestion on the road network

Mentor: dr. sc. Tomislav Erdelić

Neposredni voditelj: Nikola Mardešić. mag. ing. traff.

Student: Filip Pavčec

JMBAG: 0135257511

Sažetak:

U današnje doba, zagušenja u cestovnom prometu su postala sve veći probem koji uzrokuju gubitak vremena i produktivnosti te nezadovoljstvo korisnika odnosno vozača. S obzirom na povećani broj vozila i nedostatak učinkovitije prometne infrastrukture potrebno je razvijanje sustava za prikaz i praćenje te izbjegavanje zagušenja u prometu. U ovom završnom radu razvijena je ASP.NET web aplikacija koja prikazuje zagušenost na cestovnoj mreži za cijelu Hrvatsku. Podatci o profilima brzina i cestovnim segmentima u RH pohranjeni su u bazu podataka. Dani podaci o cestovnim segmentima imaju atribute poput koordinata, tipu segmenata nazivima ulica i dr. Cilj ovog rada je na praktičan i jednostavan način prikazati pohranjene podatke. Korisnici aplikacije će kroz aplikaciju moći pregledavati podatke,analizirati podatke i promjene te identificirati područja gdje se nalaze zagušenja.

Ključne riječi: C#, ASP.NET web aplikacija, profili brzina, zagušenje

Summary:

In today's day and age, road traffic congestion has become more of an increasing issue that causes loss of time and productivity and dissatisfaction for the average user and driver. With the increased number of vehicles and the lack of more efficient traffic infrastructure, it is necessary to develop a system to display, monitor and avoid traffic congestion. In this thesis an ASP.NET web application was developed that shows congestion on the road network for the whole of Croatia. Data on speed profiles and road segments in the Republic of Croatia are stored in the database. The given data on road segments have attributes such as coordinates, type of segments, street names, etc. The goal of this thesis is to present the stored data in a simple and practical way. Through the application, users will be able to view data, analyze data and changes, and identify areas where there is congestion.

Keywords: C#, ASP.NET web application, speed profiles, congestion

SADRŽAJ

1.UVOD.....1

2.OPIS PODATAKA O PROMETNIH ZAGUŠENJIMA.....3

2.1.Projekt SORDITO.....4

2.2 GPS.....4

2.3 Struktura pohranjenih podataka.....5

3. OBRADA I SPREMANJE PODATAKA U RELACIJSKU BAZU PODATAKA.....13

3.1.SQL Server Managment Studio.....13

3.2.Vizualizacija podataka.....14

3.3.Pohrana podataka.....19

3.4 Unos podataka.....24

4.PROGRAMSKI OKVIR ZA IZRADU WEB APLIKACIJE I OPIS RADA APLIKACIJE.....31

4.1 Nuget paket.....31

4.2 ASP.NET i C#.....32

4.3 Izrada frontend i backend dijela aplikacije.....33

4.3.1 Backend dio aplikacije.....33

4.3.2 Frontend dio aplikacije.....39

5. ANALIZA REZULTATA.....40

6. ZAKLJUČAK.....53

LITERATURA.....54

POPIS SLIKA.....56

POPIS TABLICA.....59

SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI
ODBOR ZA ZAVRŠNI RAD

Zagreb, 5. svibnja 2023.

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

ZAVRŠNI ZADATAK br. 7267

Pristupnik: **Filip Pavčec (0135257511)**
Studij: Inteligentni transportni sustavi i logistika
Smjer: Inteligentni transportni sustavi

Zadatak: **Web aplikacije za prikaz prometnih zagušenja na cestovnoj mreži**

Opis zadatka:

Cilj rada je izraditi web aplikaciju za prikaz prometnih zagušenja na cestovnoj mreži. Prvo je potrebno proučiti podatke o cestovnim segmentima u digitalnoj karti i njihovim profilima brzina spremljenih u tekstualnim datotekama. Potom je potrebno izraditi relacijsku bazu podataka i spremiti navedene podatke u bazu podataka. Nakon spremanja podataka, potrebno je povezati bazu podataka s web aplikacijom, te unutar web aplikacije koristiti javno dostupnu digitalnu kartu za prikaz podataka. Nastavno na navedeno, potrebno je u web aplikaciju dodati interaktivne elemente i događaje na karti (dvostruki klik), te iz baze podataka dohvatiti tražene podatke i prikazati ih u vizualno prihvatljivoj formi na karti. Na kraju je potrebno provesti kratku analizu podataka spremljenih u bazu podataka.

Mentor:

dr. sc. Tomislav Erdelić

Predsjednik povjerenstva za završni ispit:

1 UVOD

Zagušenje na prometnicama u današnjem svijetu predstavlja ozbijlan izazov. Zagušenja na prometnicama, zbog porasta broja vozila i ograničenim infrastrukturnim kapacitetima, postaju sve učestalija. Zbog nastalih zagušenja dolazi do posljedica kao što su povećanje vremena putovanja vozača odnosno putnika, povećanje emisije štetnih plinova i negativnih ekonomskih posljedica. Implementacija inteligentnih transportnih sustava je jedno od rješenja pomoću kojeg se problem zagušenja na cestovnoj mreži može rješavati.

Inteligentni transportni sustav se definira kao područje koje se odnosi na holističku, odnosno informacijsku i komunikacijsku te upravljačku nadogradnju na već postojeću prometnu infrastrukturu. Inteligentni transportni sustav (ITS) se temelji na pružanju mogućnosti povećanja performansi, osigurava bolji tok prometa te efikasniji i sigurniji prijevoz putnika i robe. Na ovaj način dolazi do smanjenja zagađenja okoliša te se istovremeno povećeava sama udobnost putovanja. ITS sustavi predstavljaju iznimno složene sustave koji su sastavljaeni od složenih elemenata koji imaju svoju svrhu. Te svrhe su ograničene ukoliko se koriste zasebno. Ukoliko se elementi povezivaju u jednu cijelinu tada ti elementi mogu međusobno razmjenjivati informacije na temelju čega dolazi do izrade kvalitetnog sustava na koji se svakodnevni korisnici odnosno ljudi mogu osloniti. Područje ITS-a obuhvaća sve vrste transporta: cestovni promet, zračni promet, željeznički promet, pomorski promet, informacijsko-komunikacijski odnosno telekomunikacijski promet, [1].

Primjenom naprednih tehnologija kao što su senzori, napredni komunikacijski sustavi i različiti algoritmi, ITS tehnologija će omogućiti stabilnije praćenje prometa u realnom vremenu, dinamičko upravljanje prometnom signalizacijom i semaforiziranim raskrižjima, predupitno informiranje putnika te informiranje putnika i vozača o alternativnim rutama. Integracijom tehnologija ITS-a će se omogućiti harmonizaciju prometnih tokova, smanjenje zagušenja na cestovnoj mreži te cjelokupno poboljšanje učinkovitosti prometnog sustava.

Ovaj završni rad analizira podatke o prometnim zagušenjima, vizualizira navedena zagušenja temeljem podataka o prosječnoj brzini vozila na prometnoj mreži. Vizualizacija navedenih zagušenja će se izvršiti kroz web aplikaciju izrađenu u programskom jeziku C#.

Cjelokupna tematika završnog rada je obrađena kroz 6 poglavlja:

1. Uvod
2. Opis podataka o prometnim zagušenjima
3. Obrada i spremanje podataka u relacijsku bazu podataka
4. Programski okvir za izradu web aplikacije i opis rada aplikacije
5. Analiza rezultata
6. Zaključak

Drugo poglavlje opisuje podijelu podataka i općenito dobivene podatke o prometnim zagušenjima na cestovnoj mreži. Također će se opisati značajke pojedinih podatake koji će se koristiti unutar baze podataka.

Treće poglavlje se bavi evidencijom i spremanjem podataka u relacijsku bazu podataka iz koje će se podaci koristiti za izradu aplikacije. S obzirom na to da je relacijska baza podataka direktno povezana s web aplikacijom također će se opisivati obrada i pohrana tih podataka.

Četvrto poglavlje ovog rada pruža detaljan pregled postupaka vezanih uz izradu same web aplikacije te postupke iz programskog koda C# i potrebne NuGet pakete koji su potrebni u izradi web aplikacije. Također je opisan dizajn programskog okvira koji je korišten za razvoj aplikacije.

Peto poglavlje opsiuje općenite podatke dobivene za izradu aplikacije.

U šestom poglavlju je dan zaključak rada.

2 OPIS PODATAKA O PROMETNIM ZAGUŠENJIMA

Pri prikupljanju podataka, potrebno je točno bilježiti korisne informacije. Kvalitetna oprema za prikupljanje podataka je presudan čimbenik u realizaciji samog projekta i izradi aplikacije.

U ovom poglavlju su opisani prikupljeni podaci o agregiranim GPS podacima praćene flote vozila u formi profila brzina cestovnih segmenata u digitalnoj karti - njihove prosječne brzine, brzine za dnevne i noćne vrijednosti te prosječne brzine za pojedine 5-minutne intervale za dane u tjednu.

2.1 Projekt SORDITO

Projekt SORDITO odnosno Sustav za optimizaciju ruta u dinamičkom i transportnom okruženju je projekt koji je trajao od 17.10.2014 do 16.02.2016. U sklopu projekta izrađeni su podaci koji su se koristili za izradu aplikacije i koji su bili korištenii za projekt SORDITO ,a prikupljeni su od partnerske tvrtke Mireo d.d.

Prikupljeno je približno 7 milijardi GPS zapisa u Republici Hrvatskoj u razdoblju od kolovoza 2009. do listopada 2014. Podaci su prikupljeni uz pomoć navigacijskih uređaja ugrađenih u praćena vozila, te su konačni podaci prikupljenih FCD (eng. Floating Car Data) odnosno tehnikom pokretnih vozila korištenjem navigacijskog uređaja u vozilu. Da bi se FCD tehnika mogla koristiti vozila koje se prate moraju biti opremljena globalnim satelitskim navigacijskim sustavom odnosno GNSS-om (eng. Global Navigation Satelite System), [17].

FCD tehnologija djeluje kao mobilni senzori koji se temelje na lokaciji na cestovnoj mreži te bilježe formatirane podatke kao što su:

- Geografske koordinate
- Vremenske oznake
- Trenutna brzina
- Smjer

Prikupljanje FCD podatke u duljem vremenskom razdoblju rezultira detaljnim i opsežnim podacima koji se nameću na postupke rudarenja podataka za izdavanje prostorno vremenskih obrazaca.

2.2 GPS

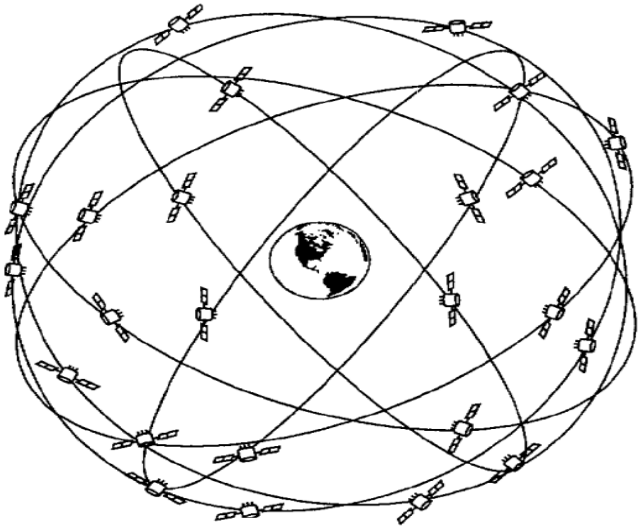
GPS odnosno NAVSTAR-GPS (NAVigation System with Time And Ranging- Global Positioning System) je sustav sustav satelita koji se koristi za globalno pozicioniranje i lociranje bilo kojeg sustava koji koristi GPS prijemnik. Raspoloživost GPS sustava je praktički stalna i globalna. Sustav se temelji na mjerenju vremena koje je potrebno za propagaciju signala od satelita do prijemnika[3].

GPS sustav se sastoji od tri segmenta:

- Svemirski segment,
- Kontrolni segment,
- Korisnički segment.

Svemirski segment GPSa se sastoji od 32 satelita od kojih je 31 trenutno aktivan. Raspored satelita se nalazi unutar 6 orbitalnih ravnina pri čemu se u svakom trenutku iznad horizonta nalazi najmanje 5 ili više satelita zbog čega je postignuta globalna pokrivenost (Slika 1.). Sateliti se gibaju oko Zemlje u kružnim orbitama na visini oko 20000km.

Sateliti emitiraju signale na dvije prijenosne frekvencije na kojima se moduliraju kodovi za pozicioniranje i navigacijske poruke. Uporabom jedne frekvencije točnost lociranja iznosi 22,5m, a uvođenjem druge frekvencije točnost se povećava za 37,8% odosno na udaljenost od 8,5m, [3].



Slika 1. Konstelacija satelita GPS sustava.

Kontrolni segment se bavi nadzorom i upravljanjem cijelog sustava. Kontrolne stanice sustava su raspoređene po cijeloj Zemlji te konstantno prate sve GPS satelite i njihove primljene satelitske signala prosljeđuju u glavno kontrolnu postaju za obradu podataka. U glavnoj postaji se računaja odstupanja pozicija svakog satelita od preciznih orbitalnih modela. Korekcijske veličini koje su izračunate za svaki satelit se šalju natrag satelitima nekoliko puta dnevno, pri čemu se uključuju u navigacijsku poruku. Sateliti mogu raditi nekoliko mjeseci iako se izgubi kontakt sa glavnim kontrolnim sustavom, [3].

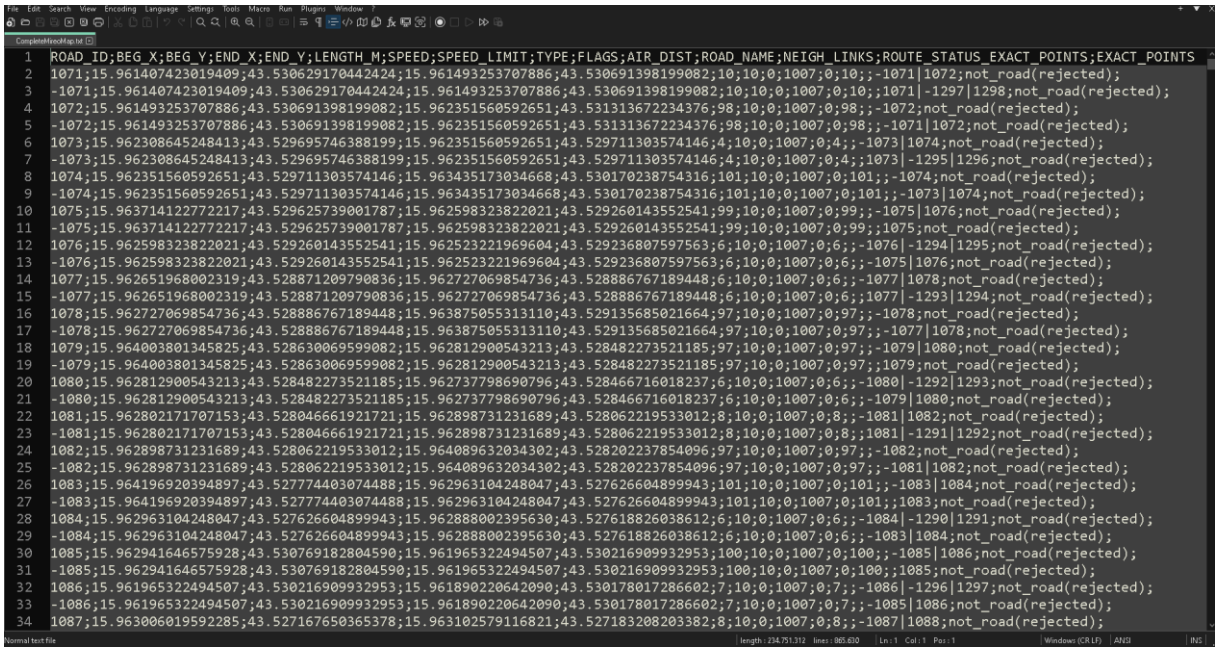
Korisnički segment GPS-a se sastoji od dvije kategorije korisnika:

- Autorizirani korisnici,
- Neautorizirani korisnici.

Autorizirani korisnici su američka vojske i druge državne službe, dok su neautorizirani korisnici svi ostali korisnici koji širom svijeta koriste sustav. Pozicioniranje sa višom razinom točnosti namijenjeno je samo autoriziranim korisnicima koji koriste dvofrekvencijski GPS prijamnik i kodove za dekriptiranje, [3].

2.3 Struktura pohranjenih podataka

Podaci su pohranjeni kroz četiri različite datoteke. Od tih četiri datoteka najbitni podaci za pojedine linkove odnosno cestovne segmente se nalazi u tekstualnoj datoteci pod nazivom „CompleteMireoMap.txt“ te su prikazani slikom (Slika 2.) i ER dijagramom. Podatke o cestovnim segmentima u digitalnoj prometnoj karti ustupila je tvrtka Mireo d.d.



```
ROAD_ID;BEG_X;BEG_Y;END_X;END_Y;LENGTH_M;SPEED;SPEED_LIMIT;TYPE;FLAGS;AIR_DIST;ROAD_NAME;NEIGH_LINKS;ROUTE_STATUS_EXACT_POINTS;EXACT_POINTS
1071;15.961407423019409;43.530629170442424;15.961493253707886;43.530691398199082;10;10;0;1007;0;10;;-1071|1072;not_road(rejected);
-1071;15.961407423019409;43.530629170442424;15.961493253707886;43.530691398199082;10;10;0;1007;0;10;;1071|-1297|1298;not_road(rejected);
1072;15.961493253707886;43.530691398199082;15.962351560592651;43.531313672234376;98;10;0;1007;0;98;;-1072;not_road(rejected);
-1072;15.961493253707886;43.530691398199082;15.962351560592651;43.531313672234376;98;10;0;1007;0;98;;-1071|1072;not_road(rejected);
1073;15.962308645248413;43.529695746388199;15.962351560592651;43.529711303574146;4;10;0;1007;0;4;;-1073|1074;not_road(rejected);
-1073;15.962308645248413;43.529695746388199;15.962351560592651;43.529711303574146;4;10;0;1007;0;4;;1073|-1295|1296;not_road(rejected);
1074;15.962351560592651;43.529711303574146;15.963435173034668;43.530170238754316;101;10;0;1007;0;101;;-1074;not_road(rejected);
-1074;15.962351560592651;43.529711303574146;15.963435173034668;43.530170238754316;101;10;0;1007;0;101;;-1073|1074;not_road(rejected);
1075;15.963714122772217;43.529625739001787;15.962598323822021;43.529260143552541;99;10;0;1007;0;99;;-1075|1076;not_road(rejected);
-1075;15.963714122772217;43.529625739001787;15.962598323822021;43.529260143552541;99;10;0;1007;0;99;;1075;not_road(rejected);
1076;15.962598323822021;43.529260143552541;15.962523221969604;43.529236807597563;6;10;0;1007;0;6;;-1076|-1294|1295;not_road(rejected);
-1076;15.962598323822021;43.529260143552541;15.962523221969604;43.529236807597563;6;10;0;1007;0;6;;-1075|1076;not_road(rejected);
1077;15.962651968002319;43.528871209790836;15.962727069854736;43.528886767189448;6;10;0;1007;0;6;;-1077|1078;not_road(rejected);
-1077;15.962651968002319;43.528871209790836;15.962727069854736;43.528886767189448;6;10;0;1007;0;6;;1077|-1293|1294;not_road(rejected);
1078;15.962727069854736;43.528886767189448;15.963875055313110;43.529135685021664;97;10;0;1007;0;97;;-1078;not_road(rejected);
-1078;15.962727069854736;43.528886767189448;15.963875055313110;43.529135685021664;97;10;0;1007;0;97;;-1077|1078;not_road(rejected);
1079;15.964003801345825;43.528630069599082;15.962812900543213;43.528482273521185;97;10;0;1007;0;97;;-1079|1080;not_road(rejected);
-1079;15.964003801345825;43.528630069599082;15.962812900543213;43.528482273521185;97;10;0;1007;0;97;;1079;not_road(rejected);
1080;15.962812900543213;43.528482273521185;15.962737798690796;43.528466716018237;6;10;0;1007;0;6;;-1080|-1292|1293;not_road(rejected);
-1080;15.962812900543213;43.528482273521185;15.962737798690796;43.528466716018237;6;10;0;1007;0;6;;-1079|1080;not_road(rejected);
1081;15.962802171707153;43.528046661921721;15.962898731231689;43.528062219533012;8;10;0;1007;0;8;;-1081|1082;not_road(rejected);
-1081;15.962802171707153;43.528046661921721;15.962898731231689;43.528062219533012;8;10;0;1007;0;8;;1081|-1291|1292;not_road(rejected);
1082;15.962898731231689;43.528062219533012;15.964089632034302;43.528202237854096;97;10;0;1007;0;97;;-1082;not_road(rejected);
-1082;15.962898731231689;43.528062219533012;15.964089632034302;43.528202237854096;97;10;0;1007;0;97;;-1081|1082;not_road(rejected);
1083;15.964196920394897;43.527774403074488;15.962963104248047;43.527626604899943;101;10;0;1007;0;101;;-1083|1084;not_road(rejected);
-1083;15.964196920394897;43.527774403074488;15.962963104248047;43.527626604899943;101;10;0;1007;0;101;;1083;not_road(rejected);
1084;15.962963104248047;43.527626604899943;15.962888002395630;43.527618826038612;6;10;0;1007;0;6;;-1084|-1290|1291;not_road(rejected);
-1084;15.962963104248047;43.527626604899943;15.962888002395630;43.527618826038612;6;10;0;1007;0;6;;-1083|1084;not_road(rejected);
1085;15.962941646575928;43.530769182804590;15.961965322494507;43.530216909932953;100;10;0;1007;0;100;;-1085|1086;not_road(rejected);
-1085;15.962941646575928;43.530769182804590;15.961965322494507;43.530216909932953;100;10;0;1007;0;100;;1085;not_road(rejected);
1086;15.961965322494507;43.530216909932953;15.961890220642090;43.530178017286602;7;10;0;1007;0;7;;-1086|-1296|1297;not_road(rejected);
-1086;15.961965322494507;43.530216909932953;15.961890220642090;43.530178017286602;7;10;0;1007;0;7;;-1085|1086;not_road(rejected);
1087;15.963006019592285;43.527167650365378;15.963102579116821;43.527183208203382;8;10;0;1007;0;8;;-1087|1088;not_road(rejected);
```

Slika 2. Prikaz podataka u programu Notepad++

Svaki red u prikazuje pojedini cestovni segment odnosno link i njegove podatke. Cestovni segment u digitalnoj karti predstavlja linijsku reprezentaciju stvarnog cestovnog segmenta. Prikupljeni podaci su strukturirani u CSV (Coma Seperated Values) formatu. Podatci koji su se koristili su odvojeni točka-zarezom, [4]. Datoteka se sastoji od 865,630 redaka od čega su 865,629 cestovni segmenti. Svaki redak je cestovni segment koji se sastoji od jedinstvenog identifikacijskog broja ili ID-a, X geografske koordinate početne točke linka, Y geografske koordinate početne točke linka, X koordinate završne točke linka, Y koordinate završne točke linka, duljinu linka izmjerena u metrima, brzina linka izmjerena u km/h, ograničenje brzine na linku, kategorija linka odnosno vrsta ceste, zastavice smjera, zračna udaljenost između početne i završne točke, naziv ulice odnosno ceste, susjedni

linkovi trenutnom linku, niz geografskih točaka x i y koji preciznije opisuju link i njihov status (Slika 3).

```
ROAD_ID;BEG_X;BEG_Y;END_X;END_Y;LENGTH_M;SPEED;SPEED_LIMIT;TYPE;FLAGS;AIR_DIST;ROAD_NAME;NEIGH_LINKS;ROUTE_STATUS_EXACT_POINTS;EXACT_POINTS
```

Slika 3. Format zapisa svakog linka

Podaci su detaljnije prikazani u sljedećoj tablici (Tablica 1.).

Tablica 1. Detaljan opis pojedinog elementa za svaki link

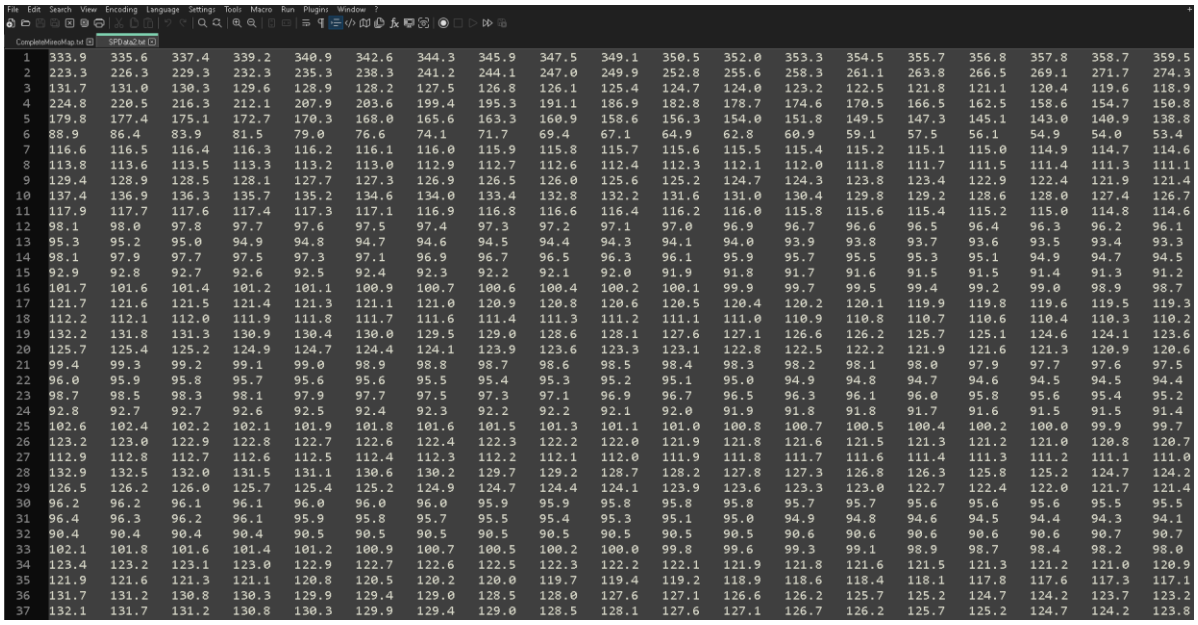
Atribut	Opis	Vrijednost
ROAD_ID	ID cestovnog segmenta (cijeli broj)	1071
BEG_X	X kooridnata (dužina) početne točke linka (B)	15.961407423019409
BEG_Y	Y kooridnata (dužina) početne točke linka (B)	43.530629170442424
END_X	X kooridnata (dužina) završne točke linka (E)	15.961493253707886
END_Y	Y kooridnata (dužina) završne točke linka (E)	43.530691398199082
LENGTH_M	Duljina linka u metrima	10
SPEED	Brzina linka u km/h koju je postavio Mireo (statična brzina – ne mijenja se ovisno o vremenu u danu/tjednu)	10

SPEED_LIMIT	Ograničenje brzine na linku koje je postavio Mireo	0
TYPE	Kategorija linka koju je postavio MIREO - 1010 su autoceste, a 1080 kvartovske male ceste, gradacija ide po 10 između tih vrijednosti. Sve manje od 1010 ili veće od 1080 su pješačke staze i slično	1007
FLAGS	Zastavica smjera: Moguće vrijednosti 0,1,2 ili 3. 0 – po linku se može putovati u dva smjera – u tom slučaju u datoteci postoje 2 retka za link, jedan za pozitivni i jedan za negativni ID linka. Kao primjer pogledajte link 215380, odnosno – 215380 1 – po linku se putuje u jednom smjeru, a ID linka je pozitivna vrijednost, i to od točke B prema točci E 2- po linku se putuje u jednom smjeru, a ID link je negativne vrijednosti, i to od točke E prema točki B 3- sve isto kao i za vrijednost 0, ali 3 predstavlja da je cesta zbog nekog razloga zatvorena	0

AIR_DIST	Zračna udaljenost u metrima između točaka B i E	10
ROAD_NAME	Naziv ulice/ceste kojoj pripada link. Neki linkovi nemaju dodijeljen naziv pa je ovo polje prazno.	Splitska
NEIGH_LINKS	Susjedni linkovi trenutnom linku. Ovo je bitno za formiranje grafa cestovne mreže. Iz jednog linka se može dalje ići na niti jedan link, na jedan link ili na više njih. Znači između dvije susjedne ; nalazi se lista susjednih linkova, pri čemu su ID-evi linkova međusobno odvojeni vertikalnom crtom (|)	-1071|1072
ROUTE_STATUS_EXACT_POINTS	S obzirom da je link zadan početnom i završnom koordinatom, mi tehnički link prikazujemo linijom. No u Mireovoj karti postoji detaljniji prikaz linkova, npr. onih zavijenih prilikom spajanja na autocestu. S obzirom da to nismo dobili,	not_road(rejected) ili ok

	dosjetkom se u 95% slučajeva može dobiti i precizniji prikaz. Poziva se Mireova funkcija za izračun rute između točaka B i E (ovisno o smjeru) koja vraća niz „preciznijih točaka“ u toj ruti. Taj funkcija kao rezultat vraća route status, i samo ako je route status vrijednosti „OK“, u stupcu EXACT_POINTS biti će zapisane točke rute.	
EXACT_POINTS	Niz geografskih točaka (x,y) koji preciznije opisuju link. Koordinate točaka su međusobno odvojene vertikalnom crtom (|).	„“(prazni string) ili 15.971170663833618|43.525401809676708|15.971331596374512|43.526514217545312

Druga datoteka pod nazivom „SPData2.txt“ sadrži podatke o profilima brzine (Slika 4) za razmatrane cestovne segmente. Datoteka se sastoji od 564,113 redaka i 198 stupaca. Svaki stupac predstavlja 5 minutni interval od 5:30 do 22:00 sata.

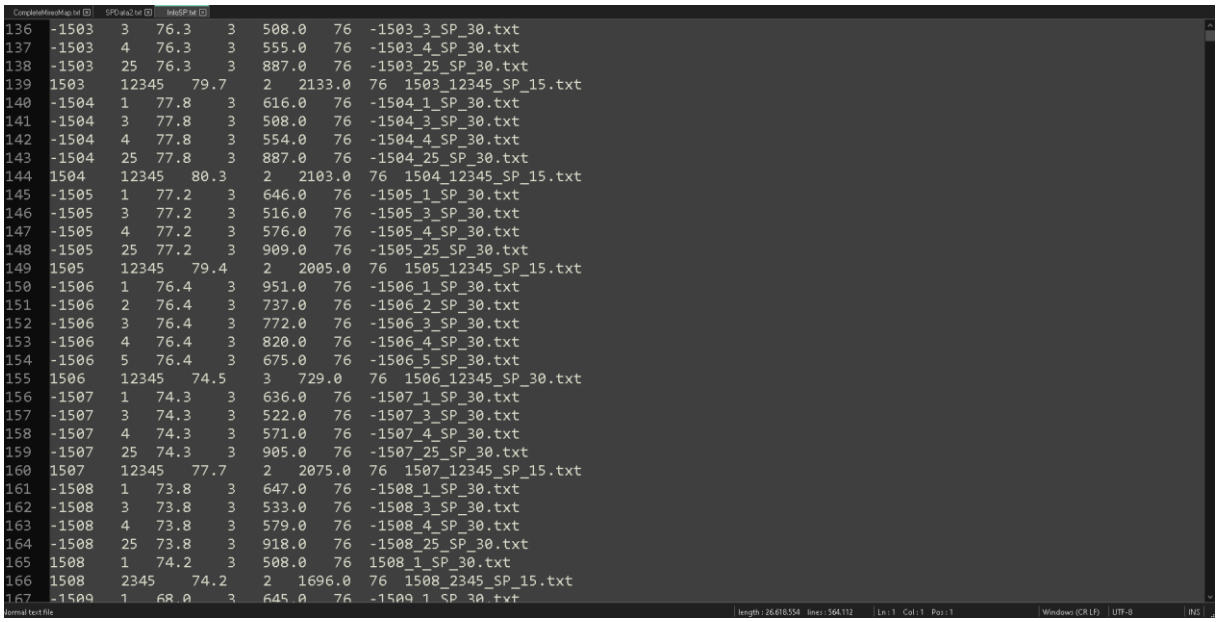


1	333.9	335.6	337.4	339.2	340.9	342.6	344.3	345.9	347.5	349.1	350.5	352.0	353.3	354.5	355.7	356.8	357.8	358.7	359.5
2	223.3	226.3	229.3	232.3	235.3	238.3	241.2	244.1	247.0	249.9	252.8	255.6	258.3	261.1	263.8	266.5	269.1	271.7	274.3
3	131.7	131.0	130.3	129.6	128.9	128.2	127.5	126.8	126.1	125.4	124.7	124.0	123.2	122.5	121.8	121.1	120.4	119.6	118.9
4	224.8	220.5	216.3	212.1	207.9	203.6	199.4	195.3	191.1	186.9	182.8	178.7	174.6	170.5	166.5	162.5	158.6	154.7	150.8
5	179.8	177.4	175.1	172.7	170.3	168.0	165.6	163.3	160.9	158.6	156.3	154.0	151.8	149.5	147.3	145.1	143.0	140.9	138.8
6	88.9	86.4	83.9	81.5	79.0	76.6	74.1	71.7	69.4	67.1	64.9	62.8	60.9	59.1	57.5	56.1	54.9	54.0	53.4
7	116.6	116.5	116.4	116.3	116.2	116.1	116.0	115.9	115.8	115.7	115.6	115.5	115.4	115.2	115.1	115.0	114.9	114.7	114.6
8	113.8	113.6	113.5	113.3	113.2	113.0	112.9	112.7	112.6	112.4	112.3	112.1	112.0	111.8	111.7	111.5	111.4	111.3	111.1
9	129.4	128.9	128.5	128.1	127.7	127.3	126.9	126.5	126.0	125.6	125.2	124.7	124.3	123.8	123.4	122.9	122.4	121.9	121.4
10	137.4	136.9	136.3	135.7	135.2	134.6	134.0	133.4	132.8	132.2	131.6	131.0	130.4	129.8	129.2	128.6	128.0	127.4	126.7
11	117.9	117.7	117.6	117.4	117.3	117.1	116.9	116.8	116.6	116.4	116.2	116.0	115.8	115.6	115.4	115.2	115.0	114.8	114.6
12	98.1	98.0	97.8	97.7	97.6	97.5	97.4	97.3	97.2	97.1	97.0	96.9	96.7	96.6	96.5	96.4	96.3	96.2	96.1
13	95.3	95.2	95.0	94.9	94.8	94.7	94.6	94.5	94.4	94.3	94.1	94.0	93.9	93.8	93.7	93.6	93.5	93.4	93.3
14	98.1	97.9	97.7	97.5	97.3	97.1	96.9	96.7	96.5	96.3	96.1	95.9	95.7	95.5	95.3	95.1	94.9	94.7	94.5
15	92.9	92.8	92.7	92.6	92.5	92.4	92.3	92.2	92.1	92.0	91.9	91.8	91.7	91.6	91.5	91.5	91.4	91.3	91.2
16	101.7	101.6	101.4	101.2	101.1	100.9	100.7	100.6	100.4	100.2	100.1	99.9	99.7	99.5	99.4	99.2	99.0	98.9	98.7
17	121.7	121.6	121.5	121.4	121.3	121.1	121.0	120.9	120.8	120.6	120.5	120.4	120.2	120.1	119.9	119.8	119.6	119.5	119.3
18	112.2	112.1	112.0	111.9	111.8	111.7	111.6	111.4	111.3	111.2	111.1	111.0	110.9	110.8	110.7	110.6	110.4	110.3	110.2
19	132.2	131.8	131.3	130.9	130.4	130.0	129.5	129.0	128.6	128.1	127.6	127.1	126.6	126.2	125.7	125.1	124.6	124.1	123.6
20	125.7	125.4	125.2	124.9	124.7	124.4	124.1	123.9	123.6	123.3	123.1	122.8	122.5	122.2	121.9	121.6	121.3	120.9	120.6
21	99.4	99.3	99.2	99.1	99.0	98.9	98.8	98.7	98.6	98.5	98.4	98.3	98.2	98.1	98.0	97.9	97.7	97.6	97.5
22	96.0	95.9	95.8	95.7	95.6	95.6	95.5	95.4	95.3	95.2	95.1	95.0	94.9	94.8	94.7	94.6	94.5	94.5	94.4
23	98.7	98.5	98.3	98.1	97.9	97.7	97.5	97.3	97.1	96.9	96.7	96.5	96.3	96.1	96.0	95.8	95.6	95.4	95.2
24	92.8	92.7	92.7	92.6	92.5	92.4	92.3	92.2	92.2	92.1	92.0	91.9	91.8	91.8	91.7	91.6	91.5	91.5	91.4
25	102.6	102.4	102.2	102.1	101.9	101.8	101.6	101.5	101.3	101.1	101.0	100.8	100.7	100.5	100.4	100.2	100.0	99.9	99.7
26	123.2	123.0	122.9	122.8	122.7	122.6	122.4	122.3	122.2	122.0	121.9	121.8	121.6	121.5	121.3	121.2	121.0	120.8	120.7
27	112.9	112.8	112.7	112.6	112.5	112.4	112.3	112.2	112.1	112.0	111.9	111.8	111.7	111.6	111.4	111.3	111.2	111.1	111.0
28	132.9	132.5	132.0	131.5	131.1	130.6	130.2	129.7	129.2	128.7	128.2	127.8	127.3	126.8	126.3	125.8	125.2	124.7	124.2
29	126.5	126.2	126.0	125.7	125.4	125.2	124.9	124.7	124.4	124.1	123.9	123.6	123.3	123.0	122.7	122.4	122.0	121.7	121.4
30	96.2	96.2	96.1	96.1	96.0	96.0	96.0	95.9	95.9	95.8	95.8	95.8	95.7	95.7	95.6	95.6	95.6	95.5	95.5
31	96.4	96.3	96.2	96.1	95.9	95.8	95.7	95.5	95.4	95.3	95.1	95.0	94.9	94.8	94.6	94.5	94.4	94.3	94.1
32	90.4	90.4	90.4	90.4	90.5	90.5	90.5	90.5	90.5	90.5	90.5	90.5	90.6	90.6	90.6	90.6	90.6	90.7	90.7
33	102.1	101.8	101.6	101.4	101.2	100.9	100.7	100.5	100.2	100.0	99.8	99.6	99.3	99.1	98.9	98.7	98.4	98.2	98.0
34	123.4	123.2	123.1	123.0	122.9	122.7	122.6	122.5	122.3	122.2	122.1	121.9	121.8	121.6	121.5	121.3	121.2	121.0	120.9
35	121.9	121.6	121.3	121.1	120.8	120.5	120.2	120.0	119.7	119.4	119.2	118.9	118.6	118.4	118.1	117.8	117.6	117.3	117.1
36	131.7	131.2	130.8	130.3	129.9	129.4	129.0	128.5	128.0	127.6	127.1	126.6	126.2	125.7	125.2	124.7	124.2	123.7	123.2
37	132.1	131.7	131.2	130.8	130.3	129.9	129.4	129.0	128.5	128.1	127.6	127.1	126.7	126.2	125.7	125.2	124.7	124.2	123.8

Slika 4. Prikaz profila brzine za prvih 19 stupaca (5:30-7:05) u programu Notepad++

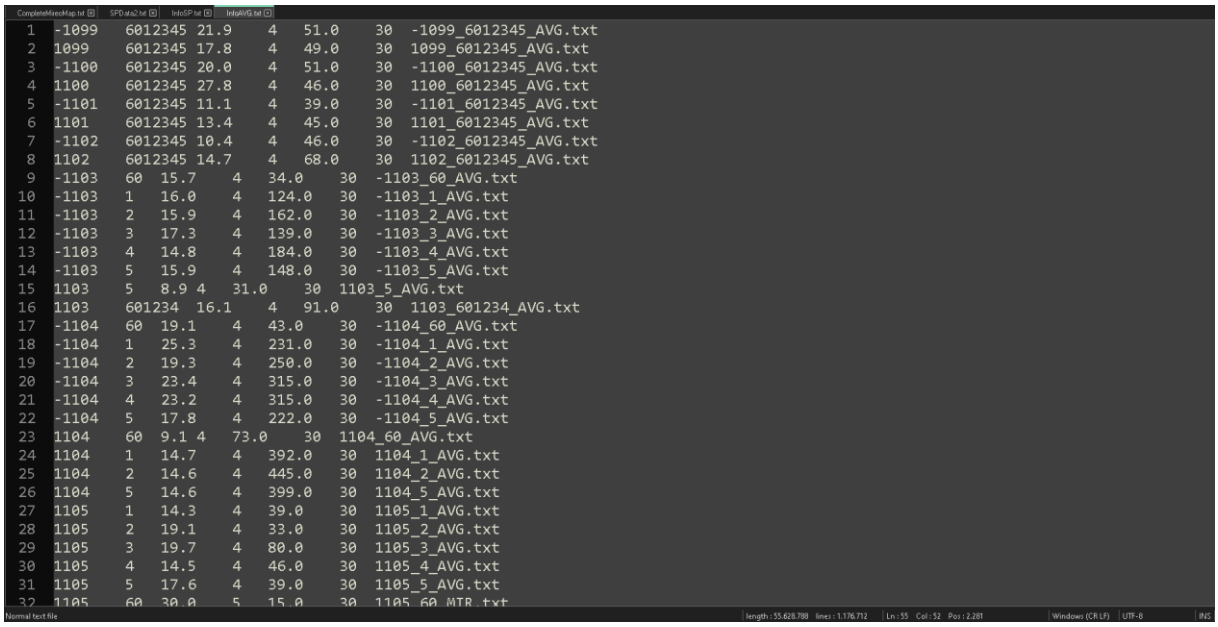
Podaci su u tekstualnoj datoteci s razgraničenjima (.txt), pri čemu je znak TAB na tipkovnici (ASCII kod znaka 009) znak koji razdvaja tekst, [5].

Treća datoteka pod nazivom „InfoSP.txt“ se sastoji od 564,112 redaka i 7 stupaca, a za aplikaciju su potrebni prvi, drugi i treći stupac. Prvi stupac prikazuje ID linka za koji je izračunat odgovarajući profil brzine. Drugi stupac prikazuje dane u tjednu za koje su izračunati profili brzine. Dani u tjednu su označeni brojevima od 0-6, pri čemu 0 označava nedjelju,1 ponedjeljak, 2 utorak, 3 srijedu, 4 četvrtak, 5 petak i 6 subotu. Ukoliko je vrijednost samo npr. 3 onda je izračunat profile brzine za srijedu, a ako je vrijedost npr. 12345 znači da nije bilo dovoljno podataka te je jedan profile izračunat i korišten za sve dane koji se nalaze u tom stringu. Treći stupac predstavlja free-flow (noćnu brzinu) odnosno brzinu u periodu od 22:00 do 5:30 (Slika 5.). Podaci u datoteci razdvojeni znakom TAB.



Slika 5. Prikaz podataka iz datoteke InfoSP.txt u programu Notepad++

Četvrta datoteka pod nazivom „InfoAVG.txt“ se sastoji od 1,176,712 i 7 stupaca. U datoteci se nalaze podaci koji nisu imali dovoljno podataka da se izračunaju profili brzini, te je izračunata samo jedna prosječna brzina (Slika 6.). Datoteka se sastoji od 7 stupaca te su za aplikaciju potrebni prvi, drugi i treći stupac. Prvi stupac prikazuje odgovarajuće ID-eve linka. Drugi stupac prikazuje dane u tjednu za skoje je izračunata prosječna brzina. Dani su također označeni brojevima od 0-6 , pri čemu 0 označava nedjelju,1 ponedjeljak, 2 utorak, 3 srijedu, 4 četvrtak, 5 petak i 6 subotu. Ako je vrijednost samo npr. 5 onda je izračunata prosječna brzina samo za četvrtak, a ako je vrijednost npr 06 onda je izračunata brzina za subotu i nedjelju. Treći stupac je prosječna brzina kroz cijeli dan izražena u km/h.



```
-1099   6012345 21.9    4   51.0    30  -1099_6012345_AVG.txt
1099    6012345 17.8    4   49.0    30  1099_6012345_AVG.txt
-1100   6012345 20.0    4   51.0    30  -1100_6012345_AVG.txt
1100    6012345 27.8    4   46.0    30  1100_6012345_AVG.txt
-1101   6012345 11.1    4   39.0    30  -1101_6012345_AVG.txt
1101    6012345 13.4    4   45.0    30  1101_6012345_AVG.txt
-1102   6012345 10.4    4   46.0    30  -1102_6012345_AVG.txt
1102    6012345 14.7    4   68.0    30  1102_6012345_AVG.txt
-1103   60  15.7    4   34.0    30  -1103_60_AVG.txt
-1103   1   16.0    4   124.0   30  -1103_1_AVG.txt
-1103   2   15.9    4   162.0   30  -1103_2_AVG.txt
-1103   3   17.3    4   139.0   30  -1103_3_AVG.txt
-1103   4   14.8    4   184.0   30  -1103_4_AVG.txt
-1103   5   15.9    4   148.0   30  -1103_5_AVG.txt
1103    5   8.9 4   31.0    30  1103_5_AVG.txt
1103    601234  16.1    4   91.0    30  1103_601234_AVG.txt
-1104   60  19.1    4   43.0    30  -1104_60_AVG.txt
-1104   1   25.3    4   231.0   30  -1104_1_AVG.txt
-1104   2   19.3    4   250.0   30  -1104_2_AVG.txt
-1104   3   23.4    4   315.0   30  -1104_3_AVG.txt
-1104   4   23.2    4   315.0   30  -1104_4_AVG.txt
-1104   5   17.8    4   222.0   30  -1104_5_AVG.txt
1104    60  9.1 4   73.0    30  1104_60_AVG.txt
1104    1   14.7    4   392.0   30  1104_1_AVG.txt
1104    2   14.6    4   445.0   30  1104_2_AVG.txt
1104    5   14.6    4   399.0   30  1104_5_AVG.txt
1105    1   14.3    4   39.0    30  1105_1_AVG.txt
1105    2   19.1    4   33.0    30  1105_2_AVG.txt
1105    3   19.7    4   80.0    30  1105_3_AVG.txt
1105    4   14.5    4   46.0    30  1105_4_AVG.txt
1105    5   17.6    4   39.0    30  1105_5_AVG.txt
1105    60  30.0    5   15.0    30  1105_60_MTR.txt
```

Slika 6. Prikaz podataka iz datoteke InfoAVG.txt u programu Notepad++

Datoteke „InfoSP.txt“ i „SPData2.txt“ imaju jednak broj redak i time prva linije u datoteci „InfoSP.txt“ gdje piše „1104 3 4.4“, ID linka je 1104 i njen profil je dan za srijedu te je u prvoj liniji datoteke „SPData2.txt“

3 OBRADA I SPREMANJE PODATAKA U RELACIJSKU BAZU PODATAKA

Relacijska baza podataka je virtualni prostor gdje se spremaju prikupljeni podaci iz stvarnog vremena u formatu u kojem se omogućuje da prikupljeni podaci manipuliraju. Podaci koji su bili dobiveni za obradu se nalazi u txt. datoteci pa ih se mora prebaciti u bazu podataka kako bi se mogli jednostavnije koristiti za obradu. U ovom poglavlju završnog rada će biti opisano programsko okruženje SQL Server Managment Studio te dizajn i obrada podataka koji su se koristili.

3.1 SQL Server Managment Studio

SQL Server Managment Studio (SSMS) je integrirano okruženje za upravljanju SQL infrastrukturom. SQL Server Managment Studio se koristi za pristup, konfiguraciju, upravljanje, administraciju i razvoj podataka unutar svih SQL-ovih komponenti kao što su SQL Server i SQL baze podataka, [6].

Kako bi se podaci mogli uspješno prenijeti u SQL Server potrebno je napravit bazu podataka. Za uspješni prijenos podataka potrebno je izraditi bazu podataka u kojoj će svaki tip pohranjenog podatka odgovarati odnosno poklapati s tipovima podataka koji se već nalazi u bazi.

Tipovi podataka koji su bili korišteni u radu da bi se mogla izraditi baza podataka su:

- Int- Egzaktni cjelobrojni iznos,
- Float- Aproksimirani podatkovni tip koji koristi plivajući zarez,
- Nvarchar- UNICODE reprezentacija teksta varijabilne veličine.

3.2 Vizualizacija podataka

Vizualizacija podataka je ključni element za analizu i manipulaciju složenih podataka za donošenje odluka na temelju dobivenih informacija. Vizualizacija podataka omogućuje jednostavno prepoznavanje i razumjevanje odnosa, obrazaca i odstupanja.

Prije izrade baze podataka potrebno je napraviti Entity Relationship Model odnosno ER dijagram za vizualizaciju podataka. ER dijagram se dizajnirao pomoću alata yEd Graph Editor koji je napravljane od tvrtke yWorks. ER dijagram odnosno dijagram entiteta definira strukturu baze podataka te prikazuje atribute i entitete te veze između njih. Svaki entitet je u nekakvom odnosu ili vezi sa drugim entitetima i atributima iz svog okruženja. Veze se mogu podijeliti na jednostavne i složene veze.

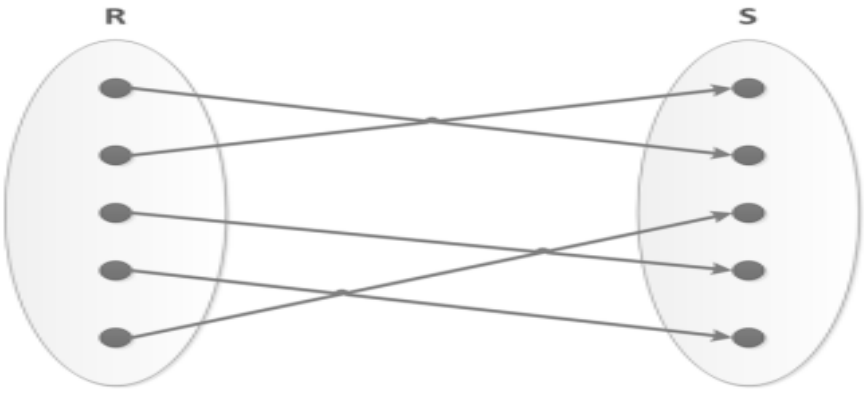
Složene veze se dijele na:

- Involuirane veze,
- Podtip veze,
- Ternarne veze.

Dok se jednostavne veze dijele na:

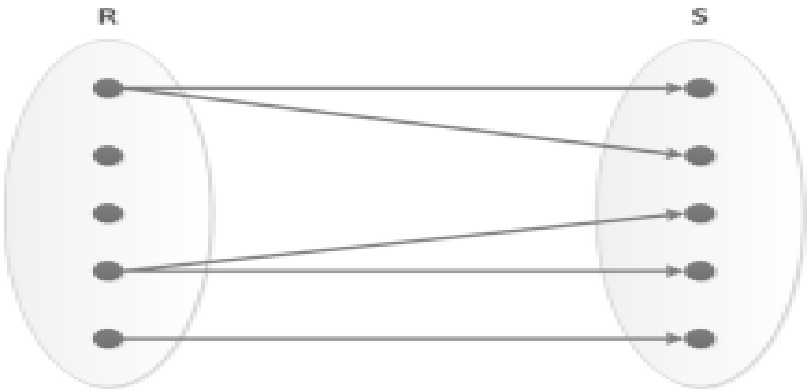
- Veza 1:1
- Veza 1:N
- Veza M:N

Kod 1:1 veze svaki element skupa R može biti povezan sa samo jednim elementom iz skupa S (Slika 7.). Isto tako vrijedi i za elemente iz skupa R.



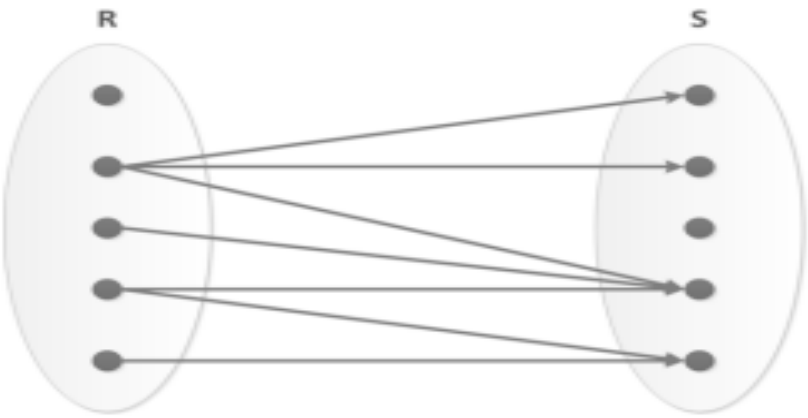
Slika 7. Prikaz 1:1 veze

Kod 1:N veze svaki element skupa R može biti povezan sa više elemenata skupa S iako nije nužno. Dok svaki element skupa S može biti povezan sa samo jednim elementom iz skupa R (Slika 8.).



Slika 8. Prikaz 1:N veze

Kod M:N veze svaki element skupa R može biti povezan sa više elemenata skupa S iako nije nužno. Isto tako svaki element skupa S može biti povezan s više elemenata skupa R iako nije nužno (Slika 9.), [7].



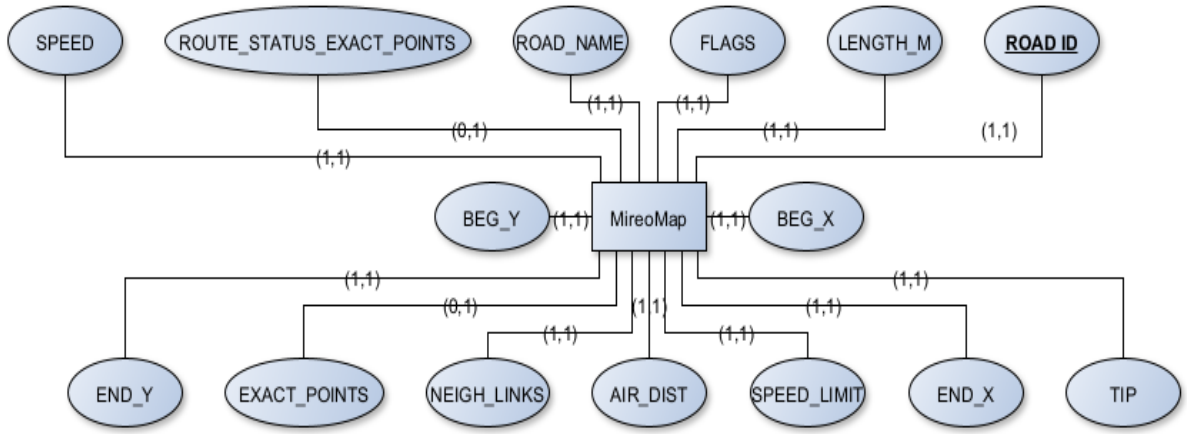
Slika 9. Prikaz M:N veze

Kod kreiranja baze podataka svaki entitet prelazi u tablicu dok se iz atributa odrede brojevi i kardinaliteti pojedinih atributa te naziv svakog stupca koji će biti unutar tablice. Za sve 4 tekstualne datoteke su napravljani dijagrami entiteta, dani njihovi kardinaliteti te odnosi odnosno veze unutar dijagrama. Zbog potrebe samo nekih podataka iz tablice „InfoSP“ i lakše pohrane podataka iz „SPData2“ tablice spojeni su u jedan dijagram entiteta.

Dijagram entiteta pod nazivom 'MireoMap' (Slika 10.):

- Road_ID - atribut i ujedno primarni ključ, kardinaliteta (1,1)
- BEG_X - atribut za početnu koordinatu X, kardinaliteta (1,1)
- BEG_Y - atribut za početnu koordinatu Y, kardinaliteta (1,1)
- END_X - atribut za završnu koordinatu X, kardinaliteta (1,1)
- END_Y - atribut za završnu koordinatu Y, kardinaliteta (1,1)
- LENGTH_M - atribut za duljinu linka, kardinaliteta (1,1)
- SPEED - atribut za brzinu linka, kardinaliteta (1,1)
- SPEED_LIMIT - atribut za ograničenje brzine, kardinaliteta (1,1)
- TIP - atribut za kategorizaciju ceste (1,1)
- FLAGS - atribut za određivanje smjera vožnje na cesti, kardinaliteta (1,1)
- AIR_DIST - atribut za zračnu udaljenost između koordinata, kardinaliteta (1,1)
- ROAD_NAME - atribut za naziv ulice kojoj cesta pripada, kardinaliteta (1,1)
- NEIGH_LINKS - atribut za susjedne linkove, kardinaliteta (1,1)
- ROUTE_STATUS_EXACT_POINTS - atribut za točke između točaka pojedinog linka, kardinaliteta(0,1)
- EXACT_POINTS - atribut na niz točaka kardinaliteta(0,1)

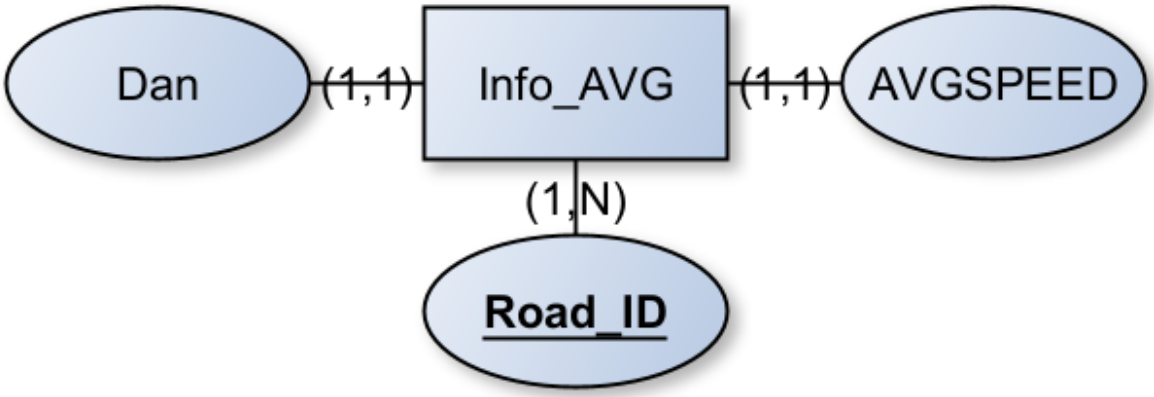
Ukoliko je kardinalitet atributa ROUTE_STATUS_EXACT_POINTS kod nekog linka (1,1) tada će i kardinalitet atributa EXACT_POINTS biti (1,1) odnosno ako Mireovoj karti postoji detaljan prikaz data će vrijednost za atribut ROUTE_STATUS_EXACT_POINTS postati 'OK' i atribut EXACT_POINTS će istovremeno vraćati geografske točke koji će opisati link.



Slika 10: Dijagram entiteta 'MireoMap'

Dijagram pod nazivom 'Info_AVG' (Slika 11.):

- Road_ID - atribut i ujedno primarni ključ, kardinaliteta (1,N)
- Dan - atribut za dane za koje je izračunata prosječna brzina, kardinaliteta (1,1)
- AVGSPEED - atribut za prosječnu brzinu, kardinaliteta (1,1)

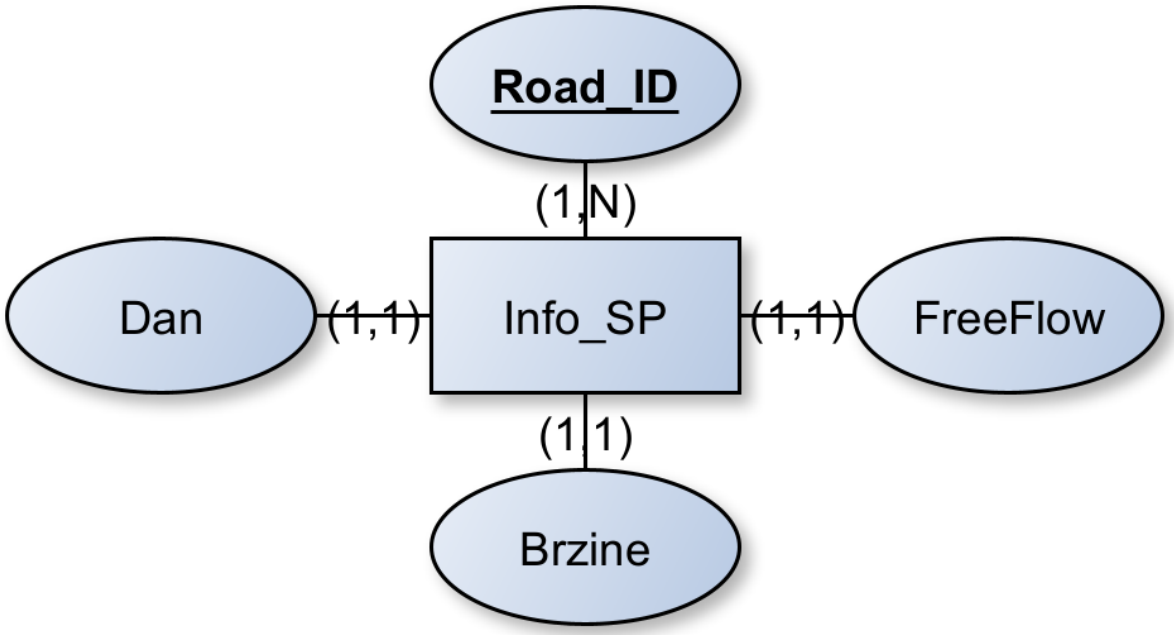


Slika 11: Dijagram entieta 'Info_AVG'

Dijagram pod nazivom 'Info_SP' (Slika 12.) koji je sastavljen od ID-a, dane za profile brzine, noćne odnosno free-flow brzine iz datoteke 'Info_SP' te 198 brzina iz datotetke

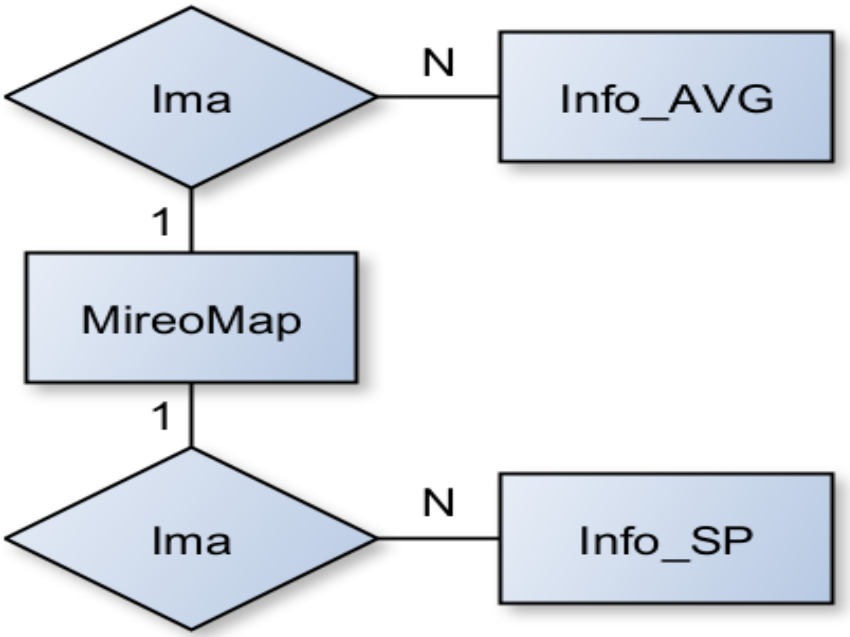
'SPData2' koji su bili spremljeni pod jednim atributom radi lakše pohrane i vizualizacije dijagrama (Slika 12.):

- Road_ID - atribut i ujedno primarni ključ, kardinaliteta (1,N)
- Dan - atribut za dane za koje su dani profili brzina, kardinaliteta (1,1)
- FreeFlow - atribut za noćnu brzinu, kardinaliteta (1,1)
- Brzine - atribut za 198 brzina, kardinaliteta (1,1)



Slika 12: Dijagram entiteta 'Info_SP'

Na slici 13 je prikazan ER dijagram između MireoMap, Info_SP i Info_AVG(Slika 13.) te njihove veze. Entitet 'MireoMap' se povezuje s entitetom Info_AVG preko veze 'Ima' s vezom 1:N što znači da isključivo 1 element iz skupa 'MireoMap' može biti povezan sa više elemenata iz skupa 'Info_AVG', ali i ne mora biti povezan niti sa jednim elementom. Također je 'MireoMap' sa istom 1:N vezom povezan sa entitetom 'Info_SP' što znači da isključivo 1 element iz skupa 'MireoMap' može biti povezan sa više elemenata iz skupa 'Info_SP' , ali i ne mora. Rješenje navedene relacijske sheme prikazano je u sljedećem potpoglavlju.



Slika 13. Prikaz ER dijagrama relacijske baze podataka

3.3 Pohrana podataka

Podaci se pohranjuju u bazu podataka pod imenom 'Zavrsni' (Slika 13.) i sastoji se od 3 tablice pod nazivom MireoMap, Info_AVG i Info_SP. Prva tablica koja se mora kreirati unutar baze podataka je pod nazivom MireoMap zatim tablice Info_AVG i Info_SP jer sadrže strani ključ koji se referencira na početnu tablicu. Ukoliko se tablice ne naprave u tom redoslijedu SQL će javiti da je došlo do greške.

Tablica pod imenom MireoMap se sastoji od 15 stupaca koji su detaljnije opisani u nastavku (Slika 14.):

- ROAD_ID - primarni ključ tablice koji će biti referenca za Road_ID iz tablica Info_AVG i Info_SP,tip podatatka cijeli broj odnosno integer (int),
- BEG_X - početni geografski položaj x-kordinate na karti, tip podatka broj s plivajućim zarezom (float),
- BEG_Y - početni geografski položaj y-kordinate na karti, tip podatka broj s plivajućim zarezom (float),

- END_X - krajnji geografski položaj x-kordinate na karti, tip podatka broj s plivajućim zarezom (float),
- END_Y - krajnji geografski položaj y-kordinate na karti, tip podatka broj s plivajućim zarezom (float),
- LENGTH_M - duljina link u metrima, tip podatka broj s plivajućim zarezom (float),
- SPEED - brzina na linku, tip podatka broj s plivajućim zarezom (float),
- SPEED_LIMIT - postavljeno ograničenje brzine , tip podatka broj s plivajućim zarezom (float),
- TIP - vrsta ceste, pomoću danog broja se određuje radi li se o kvartovskoj cesti, autocesti, pješačkim stazama i slično, tip podatka cijeli broj odnosno integer (int),
- FLAG - ovisno o danom broju određuje se radi li se o jednosmjernoj, dvosmjernoj ili zatvorenoj cesti, tip podatka cijeli broj odnosno integer (int),
- AIR_DIST - zračna udaljenost između početne i završne točke linka, tip podatka broj s plivajućim zarezom (float),
- ROAD_NAME - naziv ceste odnosno ulice, tip podatka tekst varijabilne veličine (nvarchar),
- NEIGH_LINKS - susjedni linkovi, tip podatka tekst varijabilne veličine (nvarchar),
- ROUTE_STATUS_EXACT_POINTS - označuje postojanje točaka na označenom linku, tip podatka tekst varijabilne veličine (nvarchar),
- EXACT_POINTS - koordinate točke na označenom linku, tip podatka tekst varijabilne veličine (nvarchar) .

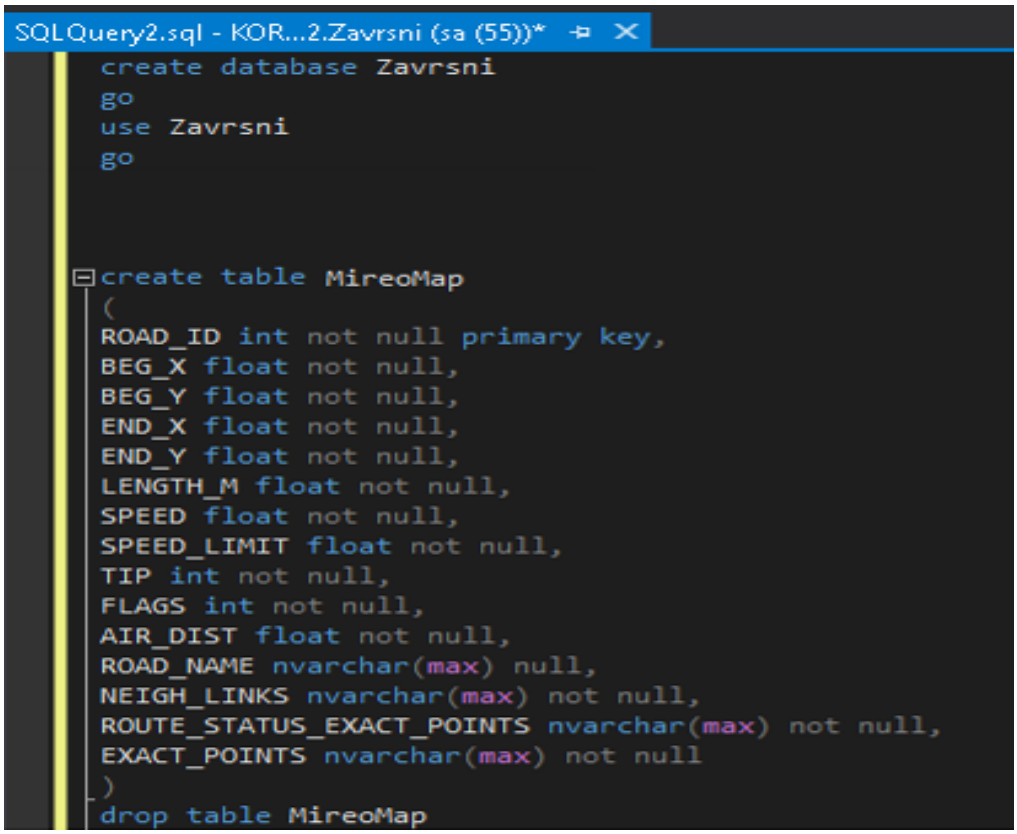
Na slici (Slika 14.) je prikazan SQL kod za izradu tablice MireoMap. Svaki stupac se sastoji od naziva stupca, podatkovnog tipa i mogućnost unošenja NULL vrijednosti ili NOT NULL vrijednosti. Svaki stupac je međusobno odvojen sa zarezima.

Osim naziva stupaca i njihovi tipovi podataka za svaki stupac odnosno tablicu potrebno je dodati određena ograničenja. Korisnik pomoću postavljenih ograničenja očuva integritet baze podataka. Što se više ograničenja postavi kod kreiranja strukture baze podataka to će održavanje kasnije biti jednostavnije. Ograničenja koja su korištena u radu su:

- NULL ili NOT NULL – ovo ograničenje sprječava odnosno dopušta unošenje NULL vrijednosti u stupac gdje je tako postavljeno. NULL vrijednost označava prazan string

ili odsutnost podatka unutar stupca. Postavljanjem ovog ograničenja na stupac spriječava potencijalne probleme ukoliko se odnosi na ubacivanje nepoznate vrijednosti.

- PRIMARY KEY – ovo ograničenje jedinstveno identificira svaki redak unutar tablice
- FOREIGN KEY – atribut koji se koristi za uspostavljanje mogućih veza između tablica



Slika 14. Skripta za kreiranje tablice MireoMap i baze podataka Zavrsni

Tablica pod imenom Info_AVG se sastoji se sastoji od 3 stupaca koji su detaljnije opisani (Slika 15.):

- Road_ID - strani ključ koji referencira ROAD_ID iz tablice MireoMap, tip podatka cijeli broj odnosno integer (int),
- Dan- dani za koji su izračunate prosječne brzine, tip podatka tekst koji može primiti maksimalno 10 dijakritčkih znakova (nvarchar(10)),

- AVGSPEED - prosječna brzina za te dobivene dane, tip podatka broj s plivajućim zarezom (float) .

S naredbom FOREIGN KEY References, stupac Road_ID postaje strani ključ koji u tablici Info_AVG se refencira na tablicu MireoMap i njen primarni ključ odnosno istoimeni stupac Road_ID. Kako bi se stupac Road_ID iz tablice Info_AVG mogao referencirat, tablica MireoMap je morala biti kreirana prva (Slika 15.).

```
create table Info_AVG
(
    Road_ID int not null FOREIGN KEY References MireoMap(Road_ID),
    Dan nvarchar(10) not null,
    AVGSPEED float,
)
```

Slika 15. Skripta za kreiranje tablice Info_AVG

Tablica pod imenom Info_SP se sastoji od 4 stupaca, pri čemu se brzine spremljene u jedan stupac radi lakše pohrane i preglednosti (Slika 16.):

- Road_ID - strani ključ koji referencira ROAD_ID iz tablice MireoMap, tip podatka cijeli broj odnosno integer (int),
- Dan - dani za koje su dani profili brzine, tip podatka tekst koji može primiti maksimalno 10 dijakritčkih znakova (nvarchar(10)),
- FreeFlow - prosječna nočna brzina, tip podatka broj s plivajućim zarezom (float),
- Brzine - prosječna brzina izmjerena u intervalima od pet minuta, počevši od 5:30 do 22:00, tip podatka broj s plivajućim zarezom (float).

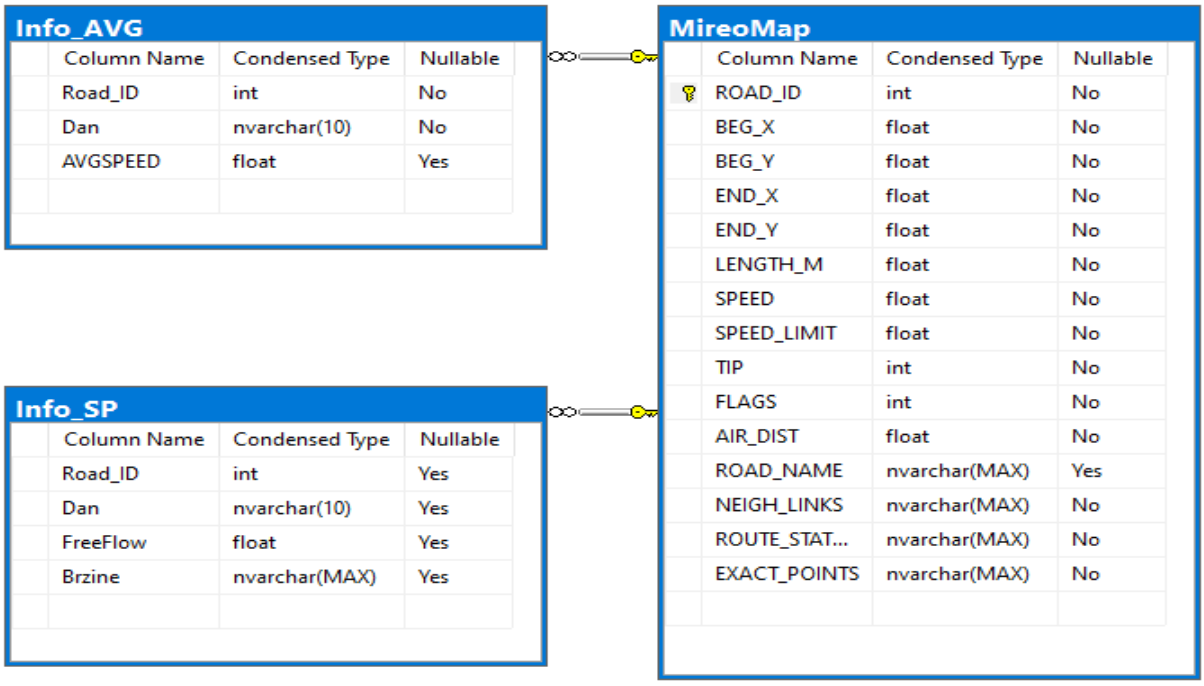
```
CREATE TABLE Info_SP (
    Road_ID int null FOREIGN KEY References MireoMap(Road_ID),
    Dan nvarchar(10) null,
    FreeFlow float null,
    Brzine nvarchar(max),
)
```

Slika 16. Skripta za kreiranje tablice Info_SP

S naredbom FOREIGN KEY References, stupac Road_ID postaje strani ključ koji u tablici Info_SP se refencira na tablicu MireoMap i njen primarni ključ odnosno istoimeni stupac Road_ID. Kako bi se stupac Road_ID iz tablice Info_SP mogao referencirat, tablica MireoMap je morala biti kreirana prva (Slika 15.).

Radi lakšeg prikaza povezanosti pojedinih tablica unutar SQL, koristi se grafički prikaz povezanosti između podataka kao što je prikazano na slici (Slika 17.). Dijagrami baze podataka se koriste za grafičkog prikazivanja strukture baze podataka. Pomoću dijagrama baze podataka se mogu stvarati i mijenjati tablice, stupci, onosi i ključevi. Također se mogu mijenjati indeksi i ograničenja, [8].

Dijagram baze podataka može pomoći u dokumentiranju i prepoznavanju zahtjeva sustava. Također daje bolje razumijevanje ukupne strukture i time omogućuje lakše mapiranje tablice i odnose njihovih primarnih i stranih ključeva, [9]. Zbog male količine tablica, dijagram baze podataka je vrlo jednostavan te ne sadrži mnogo relacija pa je samim ti i grafički prikaz jednostavan. Ovakav način prikaza tablica je odličan način za pregled relacija kod kompleksnijih baza.



Slika 17. Dijagram baze podataka

3.4 Unos podataka

Za unos podataka iz tekstualnih datoteka se koristio program napisan u programskom jeziku C# u razvojnom okruženju Visual Studio 2022. C# okruženje pruža osnovnu implementaciju pomoću koje se omogućuje rad s podacima funkcijama koje se mogu koristiti u kodu. ADO.NET koristi niz klasa koje su potrebne da bi programski jezik mogao raditi s drugim aplikacijama ili xml i tekstualnim datotekama. Klase omogućuju pristupanje podacima u bazi podataka, [10].

Klase koje su potrebne za prebacivanje iz tekstualne datoteke u bazu podataka (Slika 18.):

- System.Data - sadrži klase koje čine ADO.NET arhitekturu,
- System.Data.SqlClient - sadrži klase za spajanje i izvršavanje koda na SQL bazi podataka,
- System.IO - sadrži više klasa za čitanje, zapisivanje, i rad sa datotekam.

```
1   using System;
2   using System.Data;
3   using System.Data.SqlClient;
4   using System.IO;
```

Slika 18. Klase potrebne za unos podataka

Prije unosa potrebno je povezati C# kod za bazom podataka. Spajanje na bazu podataka se radilo putem SQL autentifikacije preko koje će se predati podaci na koji server i na koju bazu podataka se želi spojiti te sa kojim će se korisničkim podacima prijaviti. Naredba „sql.Open(); će otvoriti konekciju prema bazi podataka(Slika 19.). U navedenoj konekciji 'SqlConnection' pod nazivom 'sql' se spaja na već imenovanu bazu podataka pod nazivom 'Zavrsni' s User Id-om 'sa' te Password-om 'Baze123'.

```
SqlConnection sql = new SqlConnection(@"Server=KORISNIK-PC\MSSQLSERVER2;Database=Zavrsni;User Id=sa;Password=Baze123");
sql.Open();
```

Slika 19. SqlConnection sa podacima za spajanje na bazu

S obzirom da SQL zapisuje decimalni broj pomoću zareza, a svi decimalni brojevi unutar tekstualne datoteke su zapisani sa decimalnom točkom. Za promjenu iz decimalnog zareza u decimalnu točku potrebno je koristiti naredbu „.Replace('.', ',');“. Nakon što se uspostavi uspješna konekcija koristi se naredba pod nazivom StreamReader (Slika 20.). StreamReader je klasa namjenjena za čitanje datoteka. Naredba StreamReader se koristi za čitanje redaka i informacija iz standardne tekstualne datoteke. Naredba postavlja UTF-8 kodiranje osim ako nije drugačije navedeno. Umjesto zadanog se automatski postavljaju ANSI kodne stranice za trenutni sustav. UTF-8 ispravno obrađuje UNICODE znakove te se na temelju lokaliziranih verzija operativnog sustava dobiju dosljedni rezultati, [11].

Prije while petlje da bi se počeli čitati podaci iz tekstualne datoteke stavljena je naredba „citanje.ReadLine()“ (Slika 20.) koja čita prvi red unutar tekstualne datoteke „CompleteMireoMap.txt“. Ukoliko bi while petlja počela čitati od prvog reda program bi javio greško iz razloga što bi inače čitalo prvi red gdje se nalazi HEADER string: ROAD_ID;BEG_X;BEG_Y;END_X;END_Y;LENGTH_M;SPEED;SPEED_LIMIT;TYPE;FLAGS;AIR_DIST;ROAD_NAME;NEIGH_LINKS;ROUTE_STATUS_EXACT_POINTS;EXACT_POINTS (Slika 2.). Unutar while petlje se prvo čitaju podaci iz tekstualne datoteke počevši od drugoga reda. Pojedini podaci se spremaju ko decimalni brojevi odnosno brojevi s plivajućim zarezom pa se pohranjuju kao tip podatka double, podaci kao što su BEG_X, BEG_Y, END_X, END_Y, LENGTH_M, SPEED, SPEED_LIMIT i AIR_DIST. Svi podaci koji se čitaju iz tekstualne datotetke se automatski spremaju kao niz znakova koji se koristi za predstavljanje teksta, odnosno tip podatka string. Za te podatke potrebno je koristiti naredbu „Convert.ToDouble()“(Slika 20), koja pretvara bilo koji tip podataka koji se sastoji od brojeva u tip podatka broja s plivajućim zarezom odnosno double. Drugi podaci kao što su ROAD_ID, TIP, FLAGS se moraju spremiti kao cijeli broj odnosno tip podatka integer (int). Kako se svi podaci automatski spremaju kao niz znakova odnosno tip podatka string, navedeni podaci se moraju pretvoriti koristeći naredbu koja pretvara bilo koji tip podatka koji se sastoji od brojeva u 32-bitni cijeli broj s predznakom „Convert.ToInt32()“ (Slika 20.). Ostali podaci odnosno ROAD_NAME, NEIGH_LINKS, ROUTE_STATUS_EXACT_POINTS i EXACT_POINTS se direktno čitaju iz tekstualne datoteke, te ih nije potrebno pretvariti u drugi tip podatka jer su automatski spremljeni u tip podatka koji je potreban, niz znakova koji se koristi za predstavljanje teksa odnosno string.

```
SqlConnection sql = new SqlConnection(@"Server=KORISNIK-PC\MSSQLSERVER2;Database=Zavrsni;User Id=sa;Password=Baze123");
sql.Open();
StreamReader citanje = new StreamReader("CompleteMireoMap.txt");
string linija = "";
citanje.ReadLine();
while (citanje.EndOfStream==false)
{
    linija = citanje.ReadLine();
    string []stupac= linija.Split(';');
    int ROAD_ID = Convert.ToInt32(stupac[0]);
    double BEG_X = Convert.ToDouble(stupac[1].Replace('.', ','));
    double BEG_Y = Convert.ToDouble(stupac[2].Replace('.', ','));
    double END_X = Convert.ToDouble(stupac[3].Replace('.', ','));
    double END_Y = Convert.ToDouble(stupac[4].Replace('.', ','));
    double LENGTH_M = Convert.ToDouble(stupac[5].Replace('.', ','));
    double SPEED = Convert.ToDouble(stupac[6].Replace('.', ','));
    double SPEED_LIMIT = Convert.ToDouble(stupac[7].Replace('.', ','));
    int TIP = Convert.ToInt32(stupac[8]);
    int FLAGS = Convert.ToInt32(stupac[9]);
    double AIR_DIST = Convert.ToDouble(stupac[10].Replace('.', ','));
    string ROAD_NAME = stupac[11];
    string NEIGH_LINKS = (stupac[12].Replace('.', ','));
```

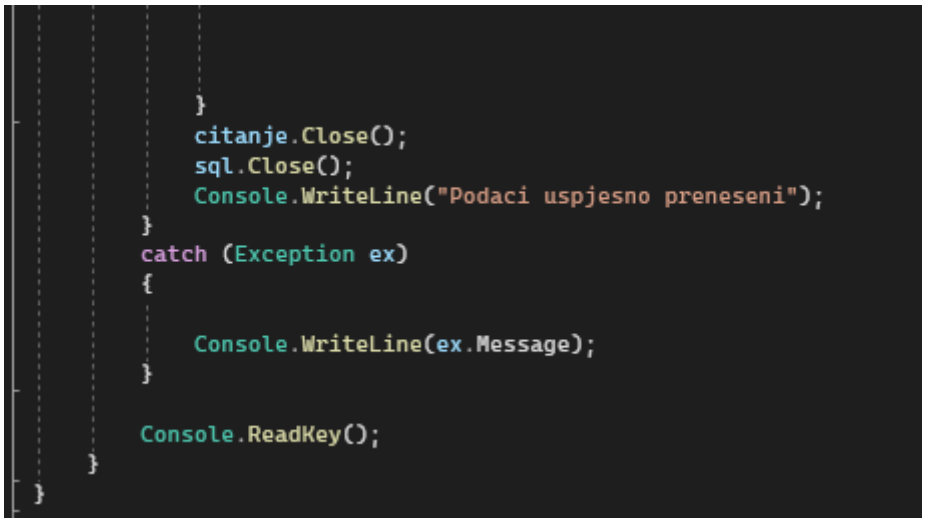
Slika 20. C# koda za čitanje tekstualne datoteke MireoMap

Prije while petlje se deklarirala 'linija' kao tip podatka string pod naredbom „string linija = '' ''“ (Slika 18.)koja se koristi za učitavanje niz znakove iz datoteke. Unutar while petlje podatak EXACT_POINTS poprima vrijednost ovisno o vrijednosti koju poprimi ROUTE_STATUST_EXACT POINTS (Slika 21.). Ako ROUTE_STATUS_EXACT_POINTS poprimi vrijednost „not_road(rejected)“ odnosno ukoliko link koji se isčitava nema istaknute preciznije točke koje se nalazi na linku, tada EXACT POINTS poprima praznu vrijednost. Ako ROUTE_STATUS_EXACT_POINTS poprimi vrijednost „ok“ odnosno, ako postoje preciznije točke na linku tada vrijednost EXACT_POINTS poprimi vrijednosti niza preciznijih geografskih točka na linku koji se u tom trenutku čita.

```
string ROUTE_STATUS_EXACT_POINTS = stupac[13];
if (ROUTE_STATUS_EXACT_POINTS == "not_road(rejected)")
{
    string EXACT_POINTS = "";
    SqlCommand cmd = new SqlCommand("Insert into MireoMap(ROAD_ID,BEG_X,BEG_Y,END_X,END_Y,LENGTH_M,SPEED,SPEED_LIMIT,TIP,FLAGS,AIR_DIST,ROAD_NAME,NEIGH_LINK
    cmd.Parameters.AddWithValue("@ROAD_ID", ROAD_ID);
    cmd.Parameters.AddWithValue("@BEG_X", BEG_X);
    cmd.Parameters.AddWithValue("@BEG_Y", BEG_Y);
    cmd.Parameters.AddWithValue("@END_X", END_X);
    cmd.Parameters.AddWithValue("@END_Y", END_Y);
    cmd.Parameters.AddWithValue("@LENGTH_M", LENGTH_M);
    cmd.Parameters.AddWithValue("@SPEED", SPEED);
    cmd.Parameters.AddWithValue("@SPEED_LIMIT", SPEED_LIMIT);
    cmd.Parameters.AddWithValue("@TIP", TIP);
    cmd.Parameters.AddWithValue("@FLAGS", FLAGS);
    cmd.Parameters.AddWithValue("@AIR_DIST", AIR_DIST);
    cmd.Parameters.AddWithValue("@ROAD_NAME", ROAD_NAME);
    cmd.Parameters.AddWithValue("@NEIGH_LINKS", NEIGH_LINKS);
    cmd.Parameters.AddWithValue("@ROUTE_STATUS_EXACT_POINTS", ROUTE_STATUS_EXACT_POINTS);
    cmd.Parameters.AddWithValue("@EXACT_POINTS", EXACT_POINTS);
    cmd.ExecuteNonQuery();
}
else
{
    string EXACT_POINTS = (stupac[14].Replace('.', ','));
    SqlCommand cmd = new SqlCommand("Insert into MireoMap(ROAD_ID,BEG_X,BEG_Y,END_X,END_Y,LENGTH_M,SPEED,SPEED_LIMIT,TIP,FLAGS,AIR_DIST,ROAD_NAME,NEIGH_LINK
    cmd.Parameters.AddWithValue("@ROAD_ID", ROAD_ID);
    cmd.Parameters.AddWithValue("@BEG_X", BEG_X);
    cmd.Parameters.AddWithValue("@BEG_Y", BEG_Y);
    cmd.Parameters.AddWithValue("@END_X", END_X);
    cmd.Parameters.AddWithValue("@END_Y", END_Y);
    cmd.Parameters.AddWithValue("@LENGTH_M", LENGTH_M);
    cmd.Parameters.AddWithValue("@SPEED", SPEED);
    cmd.Parameters.AddWithValue("@SPEED_LIMIT", SPEED_LIMIT);
    cmd.Parameters.AddWithValue("@TIP", TIP);
    cmd.Parameters.AddWithValue("@FLAGS", FLAGS);
    cmd.Parameters.AddWithValue("@AIR_DIST", AIR_DIST);
    cmd.Parameters.AddWithValue("@ROAD_NAME", ROAD_NAME);
    cmd.Parameters.AddWithValue("@NEIGH_LINKS", NEIGH_LINKS);
    cmd.Parameters.AddWithValue("@ROUTE_STATUS_EXACT_POINTS", ROUTE_STATUS_EXACT_POINTS);
    cmd.Parameters.AddWithValue("@EXACT_POINTS", EXACT_POINTS);
    cmd.ExecuteNonQuery();
```

Slika 21. Ubacivanje u SQL tablicu MireoMap

Pomoću naredbe „SqlCommand cmd“ se poziva na izvršavanje zadane naredbe za Sql server odnosno bazu podataka na kojoj je bila konekcija. Pomoću navedene naredbe ubacuju se podaci u svih 15 stupaca iz tablice MireoMap (Slika 21). Naredbe „cmd. Parameters. AddWithValue();“ zamjenjuju metodu SqlParameterCollection za izbjegavanje moguću dvosmislenosti za svaki podatak koji se ubacuje u tablicu. Nakon što se svi podaci iz tekstualne datoteke ubace u Sql tablicu, koristi se naredba „citanje.Close();“ koja zatvara početni StreamReader. Naredba „sql.Close();“(Slika 21). zatvara konekciju na početnu Sql bazu podataka koja se odredila u početku aplikacije. Try-catch blok primjećuje te ispisuje moguće greške koje bi se mogle dogoditi prilikom učitavanja podataka iz datoteke.



```
            }
            citanje.Close();
            sql.Close();
            Console.WriteLine("Podaci uspjesno preneseni");
        }
        catch (Exception ex)
        {

            Console.WriteLine(ex.Message);
        }

        Console.ReadKey();
    }
}
```

Slika 22. Zatvaranje StreamReader-a i konekcije na bazu

Kada se izvrši jedna iteracija while petlje, ažurira se redak unutar tablice MireoMap iz baze podataka. Tablica MireoMap prolazi kroz onoliko iteracije koliko ima redaka unutar tekstualne datoteke „CompleteMireoMap.txt“ u trajanju od 5:30 minuta. Primjer popunjenih redaka je prikazan na slici (Slika 23.).

Results | Messages

	ROAD_ID	BEG_X	BEG_Y	END_X	END_Y	LENGTH_M	SPEED	SPEED_LIMIT	TIP	FLAGS	AIR_DIST	ROAD_NAME	NEIGH_LINKS	ROUTE_STATUS_EXACT_POINTS	EXACT_POINTS
1	-504093	18,8511765003204	45,8517007055923	18,8512194156647	45,8521640189 08	59	5	0	1086	0	52	Crkvena	504060|-504092|504093	ok	18,851219415664673
2	-504092	18,8512301445007	45,8515960859224	18,8511765003204	45,8517007055923	12	5	0	1086	0	12		-504059|504092	ok	18,851176500320435
3	-504091	18,8457155227661	45,8532101960685	18,8450074195862	45,8530756886789	63	5	0	1086	0	57	Planina zapad	-504080|504081|504091	ok	18,845007419586182
4	-504090	18,8448035717011	45,8515437760136	18,8461661338806	45,8529710715954	191	5	0	1086	0	190	Planina istok	-504081|504082|504090	ok	18,846166133880615
5	-504089	18,8435912132263	45,8533297579195	18,8445782661438	45,8512299155278	266	5	0	1086	0	246	Planina zapad	504089	ok	18,844578266143799
6	-504088	18,8400936126709	45,8559226923266	18,8391816616058	45,8504826215279	645	15	0	1086	0	609	Planina zapad	-504086|504087|504088	ok	18,839181661605835
7	-504087	18,8400936126709	45,8559226923266	18,840479850769	45,856042248346	33	15	0	1086	0	33	Planina zapad	-504086|504087|504088	ok	18,840479850769043
8	-504086	18,8391923904419	45,8563560816741	18,8400936126709	45,8559226923266	115	15	0	1086	0	85		-504085|504086	ok	18,840093612670898
9	-504085	18,8363921642303	45,8643955887261	18,8391923904419	45,8563560816741	983	15	0	1086	0	919		504085	ok	18,839192390441895
10	-504084	18,8049137592316	45,8443992748586	18,828216791153	45,86144441774 13	2751	15	0	1086	0	2616		-503767|504084	ok	18,828216791152954
11	-504083	18,8536870479584	45,8518053250655	18,853976726532	45,8509608908445	96	25	0	1080	0	96	Dunavska	504083	ok	18,853976726531982
12	-504082	18,8448035717011	45,8515437760136	18,8445782661438	45,8512299155278	39	25	0	1080	0	39	Planina zapad	-504081|504082|504090	ok	18,844578266143799
13	-504081	18,8457155227661	45,8532101960685	18,8448035717011	45,8515437760136	199	25	0	1080	0	198	Planina zapad	-504080|504081|504091	ok	18,844803571701050
14	-504080	18,847324848175	45,8541592111836	18,8457155227661	45,8532101960685	174	25	0	1080	0	163	Planina zapad	504057|504080	ok	18,845715527661 13
15	-504079	18,8493740558624	45,8526198556615	18,8461661338806	45,8529710715954	325	25	0	1080	0	251	Crkvena	-504077|504078|504079	ok	18,846166133880615
16	-504078	18,8493740558624	45,8526198556615	18,8490736484528	45,8524330377717	36	25	0	1080	0	31	Crkvena	-504077|504078|504079	ok	18,849073648452759
17	-504077	18,8511121273041	45,8526945826417	18,8493740558624	45,8526198556615	139	25	0	1080	0	135	Crkvena	-504073|504074|-504076|504077	ok	18,849374055862427
18	-504076	18,8512086868286	45,8527244734057	18,8511121273041	45,8526945826417	8	25	0	1080	0	8	Crkvena	-504075|504076	ok	18,851112127304077

Slika 23. Prikaz rezultata dohvata podataka iz tablice MireoMap

Na isti način se popunjuje tablica Info_AVG. Nakon što se spoji na bazu podataka pomoću naredbe „sql.Open()“ deklarira se naredba StreamReader koja će čitati prva 3 stupca iz tekstualne datotetke „InfoAVG.txt“. Unutar while petlje se deklarira naredba „string[] stupac= linija.Split('\t')“ pomoću koje se razdvajuju podaci unutar tekstualne

datoteke. Podaci unutar datoteke InfoAVG su razdvojeni razmakom koji se u programskom jeziku C# označuje kao '\t'. Nakon što se podaci razdvoje, počinju se čitati samo prva tri podatka unutar svakog retka. Podatak tipa int, odnosni cijeli broj pohranjen pod nazivom AVG_ID poprima vrijednosti prvog podatka iz svakog retka u tekstualnoj datoteci. Podatak tipa string, odnosno podatak za predstavljanje niza znakova pod nazivom Dan prima drugi podatak koji se nalazi u svakom retku. odatak tipa double ili broj s plivajućim zarezom pod nazivom AVGSPEED prima treći podatak,odnosno prosječnu brzinu iz svakog retka. Nakon što se prođe potreban broj iteracija s naredbama SqlCommand, cmd.Parameters.AddWithValue i ExecuteNonQuery ubacuju se podaci u tablicu Info_AVG u trajanju od 3:30 minute (Slika 24.).

```
try
{
    SqlConnection sql = new SqlConnection(@"Server=KORISNIK-PC\MSSQLSERVER2;Database=Zavrsni;User Id=sa;Password=Baze123");
    sql.Open();
    StreamReader citanje = new StreamReader("InfoAVG.txt");
    string linija = "";
    while (citanje.EndOfStream == false)
    {
        linija = citanje.ReadLine();
        string[] stupac = linija.Split('\t');
        int AVG_ID = Convert.ToInt32(stupac[0]);
        string Dan = stupac[1];
        double AVGSPEED = Convert.ToDouble(stupac[2].Replace('.', ','));
        SqlCommand cmd = new SqlCommand("INSERT INTO Info_AVG (Road_ID, Dan, AVGSPEED) VALUES (@AVG_ID, @Dan, @AVGSPEED)", sql);
        cmd.Parameters.AddWithValue("@AVG_ID", AVG_ID);
        cmd.Parameters.AddWithValue("@Dan", Dan);
        cmd.Parameters.AddWithValue("@AVGSPEED", AVGSPEED);
        cmd.ExecuteNonQuery();
    }
    citanje.Close();
    Console.WriteLine("AVG uspjesno preneseno");
}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
Console.ReadKey();
```

Slika 24. Skripta za prebacivanje podataka iz tekstualne datoteke u tablicu Info_AVG

Tablica Info_SP kombinira dvije datoteke unutar jedne tablice što je moguće zbog jednakog broja podataka odnosno redaka. Isto vremeno se rade dva StreamReader-a, prvi pod nazivom 'citanje' koji će čitati podatke iz InfoSP-a, a drugi pod nazivom 'citanje2' će čitati podatke iz SpData2 (Slika 25).

```
StreamReader citanje = new StreamReader("InfoSP.txt");
StreamReader citanje2 = new StreamReader("Spdata2.txt");
```

Slika 25. StreamReader za datoteke InfoSP i Spdata2

Unutar while petlje se istovremeno čitaju i ubacuju podaci u tablicu Info_SP u bazi podataka. Varijabla 'linija2' se koristi primanje teksta varijabilne veličine iz datoteke Info_SP, 'Brzine2' postaje vrijednost koja je učitana iz varijable 'linija2'. Varijabla SP_ID tipa podatka cijelog broja prima prvi podatak, varijabla Dan tipa podatka cijelog broja prima drugi podatak dok varijabla FreeFlow prima podatke tipa broja s plivajućim zarezom prima treći podatak (Slika 26.). Naredba „cmd.ExecuteNonQuery()“ izvršava sve naredbe koje se povezuju na „SqlCommand cmd“ odnosno izvršava naredbu za ubacivanje učitanih podataka u tablicu Info_SP u trajanju od 3:15 minute.

```
    while (citanje.EndOfStream == false)
    {
        linija = citanje.ReadLine();

        string linija2 = citanje2.ReadLine();
        string Brzine2 = linija2.Replace('.', '.');
        string[] stupac = linija.Split('\t');
        int SP_ID = Convert.ToInt32(stupac[0]);
        int Dan = Convert.ToInt32(stupac[1]);
        double FreeFlow = Convert.ToDouble(stupac[2].Replace('.', ','));


        SqlCommand cmd = new SqlCommand("Insert into Info_SP(Road_ID, Dan, FreeFlow,Brzine) Values(@SP_ID, @Dan, @FreeFlow,@Brzine)", sql);
        cmd.Parameters.AddWithValue("@SP_ID", SP_ID);
        cmd.Parameters.AddWithValue("@Dan", Dan);
        cmd.Parameters.AddWithValue("@FreeFlow", FreeFlow);
        cmd.Parameters.AddWithValue("@Brzine", Brzine2);

        cmd.ExecuteNonQuery();


    }
    citanje.Close();
    citanje2.Close();
    Console.WriteLine("Podaci uspjesno preneseni");
    sql.Close();
    Console.WriteLine("Uspjesno odspajanje");

}
catch (Exception ex)
{
    Console.WriteLine(ex.Message);
}
```

Slika 26. Skripta za prebacivanje podataka iz tekstualne datoteke u tablicu Info_SP

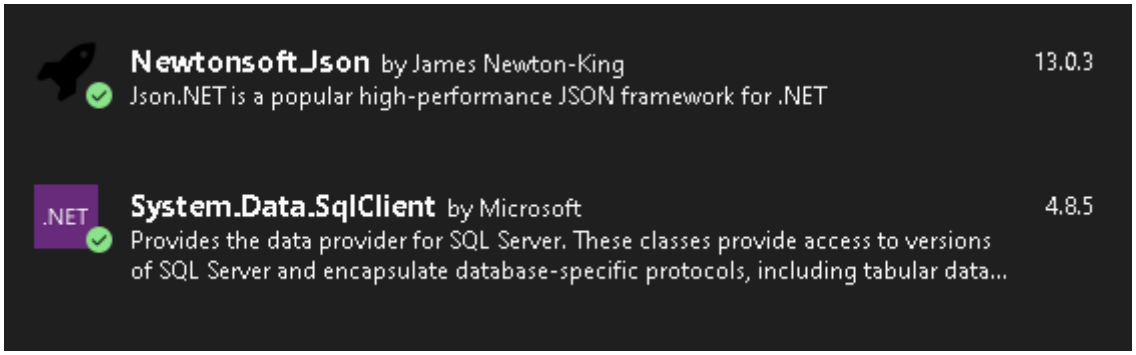
4.PROGRAMSKI OKVIR ZA IZRADU WEB APLIKACIJE I OPIS RADA APLIKACIJE

U ovom poglavlju će se opisivati način rada i korištenja web aplikacije. Aplikacija je izrađena u programskom jeziku C# koristeći razvojno okruženje Visual Studio 2022 i okruženje ASP.NET Core. Na karti će se vizualizirati linkovi te će se za odabran link prikazati graf profila brzina. Za potrebu vizualizacije koristila se OSM odnosno OpenStreetMap karta. OpenStreetMap je besplatna, otvorena geografska baza podataka koja se ažurira i održava kroz grupu volontera putem otvorene suradne.

4.1 NuGet paket

Bitan alat za svaku modernu razvojnu platformu je metoda pomoću kojeg se mogu stvarati, dijeliti i koristit korisni kodovi. Takav oblik koda je često povezan u pakete koji sadrže sastavljen kod s ostalim sadržajem koji je potreban u projektima u kojim se koriste paketi. Za .NET Core mehanizam koji se koristi za dijeljenje koda koji podržava MicroSoft se zove NuGet [12]. NuGet paket je ZIP daotteka s ekstenzijom '.nupkg' sadrži komprimirani kod, druge datoteke koje su povezane s tim kodom i dodatne informacije kao što su verzije paketa.

U ovom radu se koristio paket System.Data.SqlClient (Slika 26.). Paket pruža mogućnost spajanja i prijenos podataka za SQL Server. Klase od paketa (Slika 28.)omogućuju pristup verzijama SQL Servera i upotreba protokola specifičnih za bazu podataka uključujući tok tabličnih podataka.



Slika 27. Korišten Nuget paket

```
using System.Data.SqlClient;
using System.Data;
```

Slika 28. Klase za pristup Sql bazi podataka

4.2 ASP. NET i C#

.NET je razvojna platforma koja je sastavljena od alata, programskih jezika te biblioteka za izradu različitih vrsta aplikacija. Osnovna .NET platforma se sastoji od komponenanta koji se primjenjuju na različite vrse drugih aplikacija. Dodatni okviri kao što je ASP.NET proširiuju .NET s različitim komponentama za izgradnju određenih vrsta aplikacija [12].

C# je široko korišten programski jezik koji se može koristiti za izradu web aplikacija. C# omogućuje izradu mnogo različitih vrsta sigurnih i robusnih aplikacija koji se koriste u .NET-u. C# također pruža jezične konstrukcije za izravnu podršku svim ovim konceptima i time se C# smatra prirodnim jezikom za stvaranje i korištenje softverskih komponent, [13].

HTML odnosno Hypertext Markup Language je jedan od tri glavnih komponenata koji se koriste kod izrade modernih web stranica. Za razliku od C# što je programski jezik, HTML predstavlja označni jezik u kojem se zekst označuje tako da se može putem web stranice prikazati na određeni način. Budući da je HTML jednostavan jezik za označavanje teksta, ne može donositi odluke, odgovarati korisniku ili automatizirati zadatke koji se ponavljaju, [14].

Tehnologija koja se koristi za stilske listove naziva se CSS odnosno Cascading Style Sheets. CSS je jezik koji definira izgled odnosno stilske konstrukcije web stranica kao što su fontovi teksta, boje i pozicioniranje, a koriste se u svrhu opisivanja načine na koje su informacije oblikovane i prikazane na web stranici. CSS stilovi se mogu dodatno pohranjivati na HTML stranici, [14].

JavaScript je prvi programski jezik za skriptiranje web stranica koji preglednici podržavaju, a još uvijek je idalje daleko najpopularniji. JavaScript je razvila firma Netscape Communications Corporations, tvorac web preglednika Netscape. Neke od stvari koje se mogu napraviti u JavaScriptu su:

- Prikazivanje korisniku poruka kao dio web stranica,

- Potvrđivanje sadržaja nekog obrasca,
- Izradu izračuna,
- Animiranje slika ili stvaranje slika koji se mijenjaju kada kursorom se pomakne preko njih,
- Izrada oglasa koji mogu biti u interakciji s korisnikom umjesto da prikazuju jednostavnu sliku, [14].

4.3 Izrada frontend i backend dijela aplikacije

Frontend i Backend su dva najpopularnija pojma koja se koriste u web razvoju. Frontend dio se odnosi na ono što korisnici vide i s čime komuniciraju dok se backend dio odnosni na način na koji sve funkcionira.

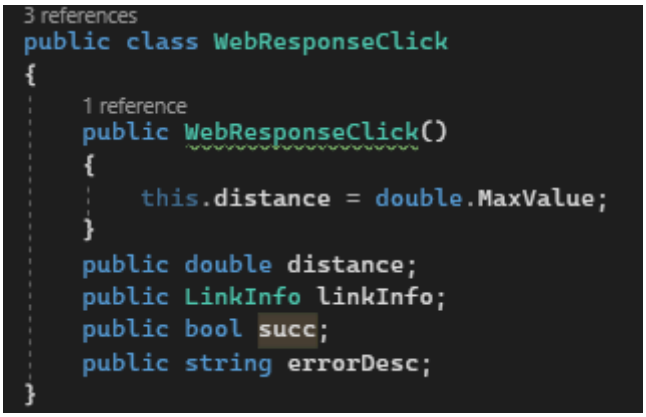
Frontend je dio koji korisnici mogu vidjeti i skojim mogu komunicirati. Komunikacija se odnosi na GUI odnosno grafičko korisničko sučelje, naredbeni redak ukjlučujući dizajn, tekstove, slike,videozapise, navigacijske izbornike te sve ostale vizualne aspekte web stranice koje korisnici mogu vidjeti i iskusiti. Frontend se također naziva i klijentskom stranom aplikacije zbog boja, stilova teksta, grafikona i gumba koje korisnici izravno doživljavaju. Glavni zadatak FrontEnd dijela je da je web stranica ispravna i da se ispravno prikazuje na svim uređajima neovisno o veličini te da se niti jedan dio web stranice ne ponaša drukčije bez obzirana na veličine zaslona, [15].

Backend je dio koji se odnosi na serversku stranu web stranice. Backend dio pohranjuje i raspoređuje podatke i također osigurava da sve na klijentskoj strani web stranice radi dobro. Sa Backend dijelom se ne može vidjeti ni komunicirati te ne dolazi u izravan dodir s korisnicima. Neke od aktivnosti Backend dijela su: stvaranje knjižica, rad s komponentama sustava bez korisničkih sučelja ili pisanja API-ja ,[15].

4.3.1 Backend dio aplikacije

Početna klasa pod nazivom 'WebResponseClick' se sastoji od četiri podatka koje će se vraćati natrag u samu aplikaciju Slika(29.). Podatak 'distance' za udaljenost, podatak

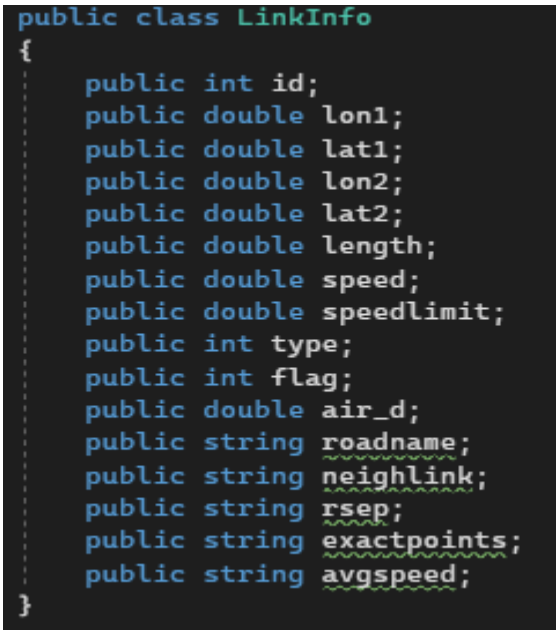
'linkInfo' za informacije o linku, podatak pod nazivom 'succ' za valjanost odabranog linka i podatak 'errorDesc' koji vraća informaciju o mogućoj greški prilikom odabira linka.



Slika 29. Podaci koji se vraćaju natrag u web

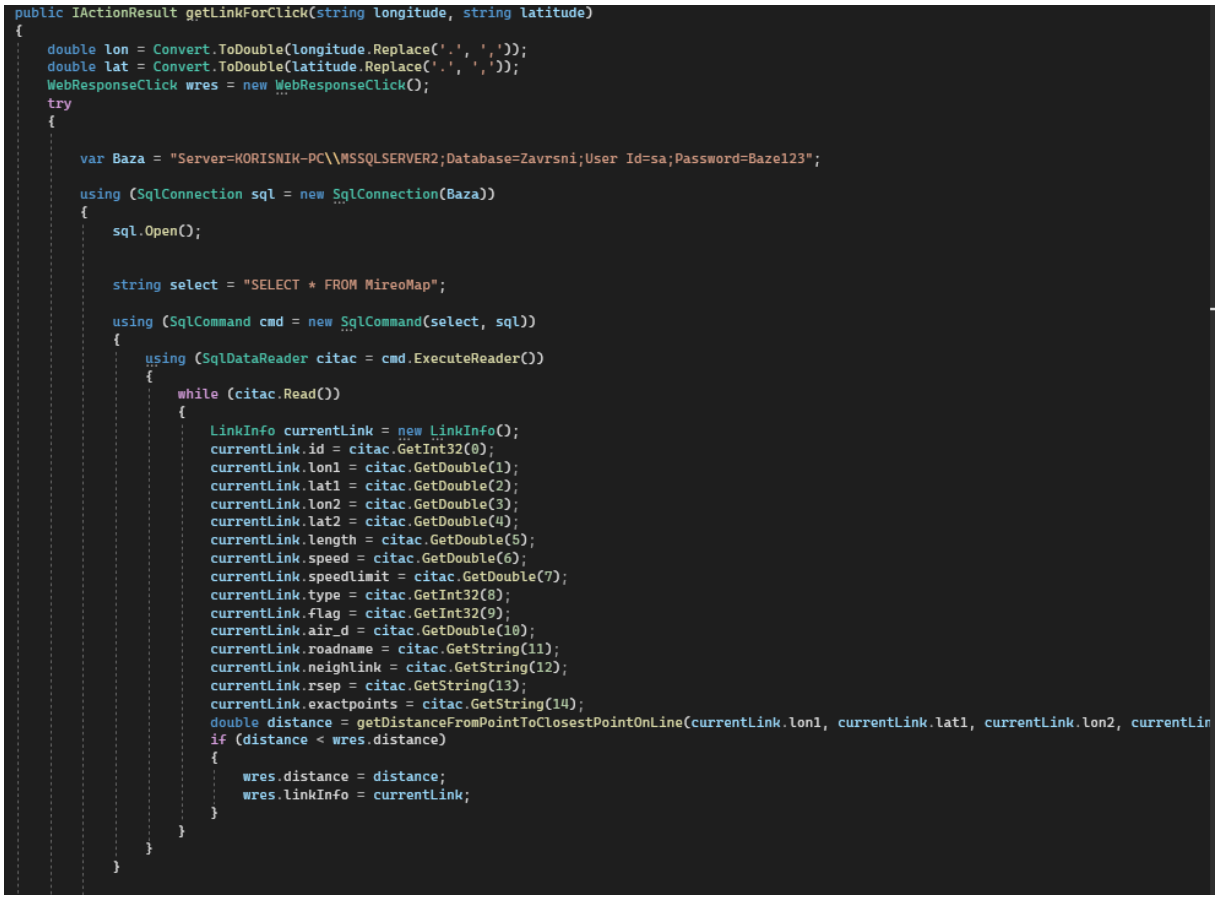
Klasa pod nazivom LinkInfo je klasa koja je definirana za spremanje podataka označenog linka. Sve navedene informacije koje su pružaju o pojedinom linku imaju javno postavljenu 'vidljivost' (public) što znači da ih je moguće upotrijebit iz drugih klasa te su dalje navedene:

- Javna (public) cijelobrojna vrijednost (int) pod nazivom 'id',
- Javne (public) decimalne vrijednosti (double) 'lon1' odnosno BEG_X, 'lat1' odnosno BEG_Y, 'lon2' odnosno END_X, 'lat2' odnosno END_Y,
- Javna (public) decimalna vrijednost (double) pod nazivom 'length' odnosno LENGTH_M,
- Javne (public) decimalne vrijednosti (double) pod nazivom 'speed' i 'speedlimit' ,
- Javne (public) cijelobrojne vrijednosti (int) pod nazivom 'type' i 'flag',
- Javna (public) decimalna vrijednost (double) pod nazivom 'air_d',
- Javni (public) tekst varijabilne veličine (string) 'roadname', 'neighlink', 'rsep' odnosno ROUTE_STATUS_EXACT_POINTS, 'exactpoints', 'avgspeed' (Slika 30.).



Slika 30. Klasa LinkInfo

Metoda IactionResult getLinkForClick se poziva iz web stranice kada se dva puta klikne na neku lokaciju na karti, u varijabilne longitude i latitute se predaje određena geografska dužina, odnosno širina, iz klika na karti. Metoda stvara konekciiju direktno na bazu podataka pomoću varijable pod nazivom 'Baza' te kod decimalnih vrijednosti lon1, lon2, lat1, lat2 mijenja decimalnu točku u decimalni zarez putem naredbe „Replace('.', ','));“. U string variablu naziva 'select' se pohranjuje naredba „SELECT * FROM MireoMap“iz SQL-a koja uzima svaki redak iz tablice MireoMap te kroz naredbu 'SqlDataReader',koja omogućuje način čitanja toka redaka za baze iz Sql Servera, prolazi kroz svaki pojedini redak te u while petlji učitava svaku pojedinu varijablu koja se u njoj nalazi te ih predaje u 'currentlink' koji se natrag predaje u početnu klasu LinkInfo (Slika 31.).



```
public IActionResult getLinkForClick(string longitude, string latitude)
{
    double lon = Convert.ToDouble(longitude.Replace('.', ','));
    double lat = Convert.ToDouble(latitude.Replace('.', ','));
    WebResponseClick wres = new WebResponseClick();
    try
    {

        var Baza = "Server=KORISNIK-PC\\MSSQLSERVER2;Database=Zavrsni;User Id=sa;Password=Baze123";

        using (SqlConnection sql = new SqlConnection(Baza))
        {
            sql.Open();


            string select = "SELECT * FROM MireoMap";

            using (SqlCommand cmd = new SqlCommand(select, sql))
            {
                using (SqlDataReader citac = cmd.ExecuteReader())
                {
                    while (citac.Read())
                    {
                        LinkInfo currentLink = new LinkInfo();
                        currentLink.id = citac.GetInt32(0);
                        currentLink.lon1 = citac.GetDouble(1);
                        currentLink.lat1 = citac.GetDouble(2);
                        currentLink.lon2 = citac.GetDouble(3);
                        currentLink.lat2 = citac.GetDouble(4);
                        currentLink.length = citac.GetDouble(5);
                        currentLink.speed = citac.GetDouble(6);
                        currentLink.speedlimit = citac.GetDouble(7);
                        currentLink.type = citac.GetInt32(8);
                        currentLink.flag = citac.GetInt32(9);
                        currentLink.air_d = citac.GetDouble(10);
                        currentLink.roadname = citac.GetString(11);
                        currentLink.neighlink = citac.GetString(12);
                        currentLink.rsep = citac.GetString(13);
                        currentLink.exactpoints = citac.GetString(14);
                        double distance = getDistanceFromPointToClosestPointOnLine(currentLink.lon1, currentLink.lat1, currentLink.lon2, currentLin
                        if (distance < wres.distance)
                        {
                            wres.distance = distance;
                            wres.linkInfo = currentLink;
                        }
                    }
                }
            }
```

Slika 31. Skripta koja čita tablicu MireoMap te ih predaje u LinkInfo

Također se istovremeno na temelju odabranog link izvršava select naredba koja odabire svaki redak čiji se id podudara sa odabranim id-em nakon duplog klika na karti koji se dalje koristi za prikaz grafa profila brzine (Slika 32).

```
    string select1 = "SELECT * FROM Info_SP where Road_ID =" + wres.linkInfo.id + "";
    if (string.IsNullOrEmpty(select1))
    {
        wres.linkInfo.avgspeed = "x";
    }
    else
    {
        using (SqlCommand cmd1 = new SqlCommand(select1, sql))
        {
            using (SqlDataReader citac1 = cmd1.ExecuteReader())
                while (citac1.Read())
                {
                    wres.linkInfo.avgspeed = citac1.GetString(3);
                }
        }

    }


    sql.Close();

}
```

Slika 32. Odabir redaka iz tablice Info_SP

Podatak errorDesc se koristi ukoliko se pojavi greška prilikom odabira linka na karti. Također se koristi ukoliko je odabir korisnika na karti više od 70 metara udaljen od najbližeg linka i time vraća poruku korisniku da je odabrana lokacija korisnika dalja od 70 metara od najbližeg linka (Slika 33).

```
    if (wres.distance > 70)
    {
        wres.succ = false;
        wres.errorDesc = "Location of the point is too far from any link!";
    }
    else
    {
        wres.succ = true;

    }

}
catch (Exception ex)
{
    wres.errorDesc = ex.Message;
    wres.succ = false;
}
```

Slika 33. Prikaz moguće greške i udaljenosti veće od 70 metara

Pomoćna metoda naziva 'airalDistHaversine' prima lon1, lat1, lon2 i lat2 te pomoću njih računa zračnu udaljenost u metrima između dvije odabrane lokacije te ju vraća kao varijablu pod nazivom 'd'. Metoda se koristi ukoliko se projekcija nalazi izvan granica linija, tada se pomoći navedene metode korigira na najbližu točku na liniji (Slika 34.).

```
public static double airalDistHaversine(double lon1, double lat1, double lon2, double lat2)
{

    double R = 6371000; // metres
    double phi1 = lat1 * Math.PI / 180; // φ, λ in radians
    double phi2 = lat2 * Math.PI / 180;
    double deltaphi = (lat2 - lat1) * Math.PI / 180;
    double deltaLambda = (lon2 - lon1) * Math.PI / 180;

    double a = Math.Sin(deltaphi / 2) * Math.Sin(deltaphi / 2) +
               Math.Cos(phi1) * Math.Cos(phi2) *
               Math.Sin(deltaLambda / 2) * Math.Sin(deltaLambda / 2);
    double c = 2 * Math.Atan2(Math.Sqrt(a), Math.Sqrt(1 - a));

    double d = R * c;
    return d;
}
```

Slika 34. Metoda za računanje pomoćne zračne udaljenosti.

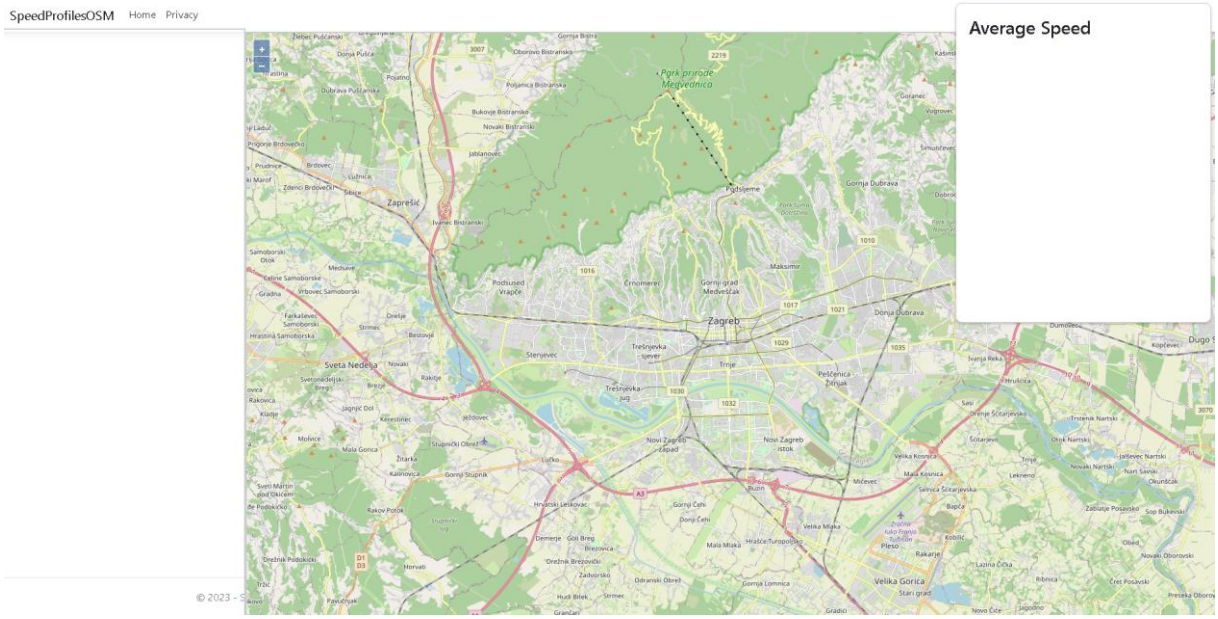
Klasa getDistanceFromPointToClosestPointOnLine istovremeno računa udaljenost od odabrane točke do najbliže točke na liniji i korigira odabir korisnika na najbližu točku na liniji ukoliko je odabir bio izvan granica linija (Slika 35.).

```
public static double getDistanceFromPointToClosestPointOnLine(double lx1, double ly1, double lx2, double ly2, double px, double py)
{
    //Vektorski racuna najblizu posctku
    double[] vec_l1P = new double[2] { px - lx1, py - ly1 };
    double[] vec_l1l2 = new double[2] { lx2 - lx1, ly2 - ly1 };

    double mag = Math.Pow(vec_l1l2[0], 2) + Math.Pow(vec_l1l2[1], 2);
    double prod = vec_l1P[0] * vec_l1l2[0] + vec_l1P[1] * vec_l1l2[1];
    double normDist = prod / mag;

    double clX = lx1 + vec_l1l2[0] * normDist;
    double clY = ly1 + vec_l1l2[1] * normDist;

    //Ukoliko je projkekcije izvan granica linije, korigirati na najblizu tocku na liniji
    double minLX = Math.Min(lx1, lx2);
    double minLY = Math.Min(ly1, ly2);
    double maxLX = Math.Max(lx1, lx2);
    double maxLY = Math.Max(ly1, ly2);
    if (clX < minLX)
    {
        clX = minLX;
    }
    if (clY < minLY)
    {
        clY = minLY;
    }

    if (clX > maxLX)
    {
        clX = maxLX;
    }
    if (clY > maxLY)
    {
        clY = maxLY;
    }

    return airalDistHaversine(px, py, clX, clY);
}
```

Slika 35. Metoda getDistanceFromPointToClosestPointOnLine

4.3.2 Frontend dio aplikacije

CSHTML dio aplikacije kontrolira što će se prikazati na web aplikaciji. Unutar CSHTML dijela koda se postavlja OSM karta (Slika 36.). Za rad s OSM kartom korištena je biblioteka OpenLayers. Centar karte je postavljan na kordinate 15,981179 i 45,805821 koje odgovaraju centru grada Zagreba (Slika 37.). Navedene koordinate su reprojicirane u koordinatni sustav OSM karte. Razina prikaza postavljena je na 12. Cijelokupna karta smještena je u div element divMap, te spremljena u varijablu map.



Slika 36. OSM karta

```
var map = new ol.Map({
    interactions: ol.interaction.defaults({ doubleClickZoom: false }),
    target: 'divMap',
    layers: [
        new ol.layer.Tile({
            source: new ol.source.OSM()
        })
    ],
    view: new ol.View({
        center: ol.proj.fromLonLat([15.981179, 45.805281]),
        zoom: 12
    })
});
```

Slika 37. Dio koda koji postavlja Zagreb kao centar

U div elementu pod nazivom 'divOptions' se nalazi kod koji omogućuje prikaz podataka s lijeve strane (Slika 38.). Unutar elementa se također postavlja stil pozadine kao transparentna pomoću „background – color: transparent;“, stil teksta kao serif pomoću „font – family: serif“ i boja teksta kao plava pomoću „color: dodgerblue;“.Prilikom odabira linka na karti se sa lijeve strane na web aplikaciji prikazuju svi podaci o odabranom linku, na način da se popuni unutarnji dio HTMLa varijable linkInfoDiv (Slika 39.).

```
<div id="divOptions" style="background-color: transparent; font-family: serif; font-weight: bold; color: dodgerblue;">

    <div id="linkInfo"></div>
</div>
```

Slika 38. Div element za prikaz podataka o odabranom linku.

```
linkInfoDiv.innerHTML = `
        <h3>Link Information</h3>
        <ul>
            <li><strong>ROAD_ID:</strong> ${li.id}</li>
            <li><strong>BEG_X: </strong> ${li.lon1}</li>
            <li><strong>BEG_Y: </strong> ${li.lat1}</li>
            <li><strong>END_X: </strong> ${li.lon2}</li>
            <li><strong>END_Y: </strong> ${li.lat2}</li>
            <li><strong>Length: </strong> ${li.length} m</li>
            <li><strong>Speed: </strong> ${li.speed} km/h</li>
            <li><strong>SpeedLimit: </strong> ${li.speedlimit} km/h</li>
            <li><strong>Type:</strong> ${li.type}</li>
            <li><strong>Flag:</strong> ${flagRijec}</li>
            <li><strong>AirDistance:</strong> ${li.air_d} m</li>
            <li><strong>RoadName:</strong> ${li.roadname}</li>
            <li><strong>NeighboringLinks:</strong> ${li.neighlink}</li>
            <li><strong>RouteStatusExactPoints:</strong> ${li.rsep}</li>
            <li><strong>ExactPoints:</strong> ${li.exactpoints}</li>

            </ul>
    `;
```

Slika 39. Prikaz svakog elementa za odabrani link.

Div element pod nazivom 'popup' (Slika 42.) nalazi se na gornjoj desnoj strani web aplikacije sa naslovom 'Average Speed'. Element istovremeno prima izgled odnosno 'dizajn' samog grafa i pretvara profil brzina koji je predan kao tekst varijabilne veličine u brojčane vrijednosti.

Za izradu grafa koristila se JavaScript biblioteka za izradu grafova Chart.js. Chart.js je otvorena biblioteka na Githubu koaj omogućuje izradu različitih vrsta grafova kroz HTML element platna.

Biblioteka podržava 8 vrsta grafova:

- Line chart odnosno linijski grafikon,
- Bar chart odnosno linijski grafikon,
- Doughnut chart odnosno prstenasti grafikon,
- Pie chart odnosno tortni grafikon,
- Radar chart odnosno radarski grafikon,
- Polar chart odnosno polarni grafikon,
- Bubble chart odnosno mjehurićni grafikon i
- Scatter chart odnosno raspršeni grafikon, [15]

Kod izrade grafa prvi korak je uključiti chart.js u HTML-u (Slika 40.). Za samu izradu se mora deklarirati klasa pod nazivom „chartCanvas“ pri čemu Average Speed predstavlja naslov samog popup-a(Slika 41.). S obzirom da su podaci spremljeni kao string potrebno ih je pravilno pripremiti za pohranu na način da se po potrebi razdvoje pomoću znaka „/t“ odnosno praznog stringa. Kod dizajna grafa sa naredbom type: 'line' se koristio linijski graf zbog boljeg prikaza velikih količina podataka unutar malog polja, podaci su se sa naredbom 'data: brzine' koristili iz varijable pod nazivom brzine, dok oznake na samom grafu su napravljene pomoću naredbe „labels: numberArray“ te predstavljaju vrijeme iz petminutnih intervala u trajanju od 5:30 do 22:00 sata (Slika 43.).

```
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Slika 40. Link chart.js-a

```
<div id="popup" class="popup">
    <h3>Average Speed</h3>
    <canvas id="chartCanvas"></canvas>
</div>
```

Slika 41. Klasa chartCanvas

```
// Prosjek brzina, pretvorba u brojeve
var avgspeed = li.avgspeed;
var brzine = avgspeed.split('\t').map(parseFloat);
var numberArray = [];
```

Slika 42. Priprema podataka za pohranu

```
for (var i = 5.5; i <= 22; i = i + (5 / 60.0)) {
    numberArray.push(i.toFixed(2));
}
// Dizajn za graf
var ctx = graf.getContext('2d');
var chart = new Chart(ctx, {
    type: 'line',
    data: {
        labels: numberArray,
        datasets: [{
            label: 'Average Speed',
            data: brzine,
            borderColor: 'rgba(0, 123, 255, 1)',
            backgroundColor: 'rgba(0, 123, 255, 0.2)',
            tension: 0.3
        }]
    },
    options: {
        responsive: true,
        scales: {
            y: {
                beginAtZero: true
            }
        }
    }
});
```

Slika 43. Dizajn grafa

Dimenzije grafa se postavljene pomoću naredbi graf.width te graf.height. Element također pomoću naredbe „chartContainer.innerHTML = '';“ briše prijašnji graf iz popup-a za mogućnost prikaza novog grafa od novo odabranog linka (Slika 41).

```
// Dimenzije

var graf = document.createElement('canvas');
graf.id="";
graf.id = 'chartCanvas';
graf.width = 300;
graf.height = 200;


// Graf
var chartContainer = document.getElementById('popup');
chartContainer.innerHTML = '';
chartContainer.appendChild(graf);
```

Slika 44. Kod za izgled grafa i za pohranu odnosno pretvorbu podatka 'avgspeed'

Duplim klikom na kartu se stvara konekcija na bazu podataka pri ćemu se dohvate sve potrebne informacije za prikaz podataka na karti te se poziva funkcija pod nazivom 'dblclick' (Slika 45). Linija „if (activePlotedLink != undefined)“ provjerava ukoliko postoji prethodna veza na mapi i ako postoji onda se već postojeća veza uklanja s mapa. Linija „var latlong = ol.proj.transform(event.coordinate, 'EPSG:3857', 'EPSG:4326');“ uzima koordinatne točke na mapi i pretvara ih iz Mercator projekcije odnosno 'EPSG:3857' u geografski koordinatni sustav odnosno EPSG:4326. Linija „url: "@Url.Action("getLinkForClick")"“ predstavlja URL na koji će zahtjev biti poslan na temelju server-side metode 'getLinkForClick'. Linija „data: { "longitude": latlong[0], "latitude": latlong[1] }“ predstavlja podatke koji se šalju odnosno geografske duljine i geografske širine. Linija „failure: function (response)“ se koristi u slučaju neuspješnog povratnog poziva koja se poziva ukoliko se pojavi greška. Poruka prikazuje upozorenje s objašnjenjem o pogrešci.

```
//Dupli click na kartu
map.on('dblclick', function (event) {
    //Pobrisi prethodno iscrtani link
    if (activePlotedLink != undefined) {
        vectorSource.removeFeature(activePlotedLink);
        activePlotedLink = undefined;
    }
    //Pretvori koordinate iz pixela u karti u standardni 4326 sustav s geografskim duzinama i sirinama
    var latlong = ol.proj.transform(event.coordinate, 'EPSG:3857', 'EPSG:4326');
    //Pozovi funkciju u backendu za ucitavanje podataka o linku
    $.ajax({
        type: "POST",
        url: "@Url.Action("getLinkForClick")",
        data: { "longitude": latlong[0], "latitude": latlong[1] },
        success: succCallGetLinkForClick,//Funkcija koja se poziva ako je poziv bio uspješan
        failure: function (response) {
            //Ispiši grešku u malom prozorčiću
            alert(response.error);
        }
    });
});
```

Slika 45. Metoda za dupli klik na kartu

Navedena funkcija poziva funkciju „succCallGetLinkForClick“ koja je prikazana na slici (Slika 46.). Navedena funkcija prvo provjerava vrijednosti svojstva 'succ' te ako je vrijednost 'false' onda je došlo do greške prilikom dohvaćanja podataka. Ukoliko nije došlo do greške onda će se kod izvršiti. Linija 'var points' kreira dvodimenzionalno polje koje sadrži točke koje definiraju početnu i završnu točku polilinije. Linija 'var polyline' stvara polilinije koristeći koordinate iz polja pod nazivom 'points'. Linija 'polyline.transform' natrag pretvara koordinate iz geografskog koordinatnog sustava u Mercatorovu projekciju zbog pravilnog prikaza na karti.

```
function succCallGetLinkForClick(response) {
    if (response.succ == false) {
        // Display error message in an alert
        alert(response.errorDesc);
    }
    else {
        var li = response.linkInfo;
        var points = [[li.lon1, li.lat1], [li.lon2, li.lat2]];
        //Polilinije za link
        var polyline = new ol.geom.LineString(points);
        //Pretvori nazad iz 4326 u 3857 koji korisiti OSM
        polyline.transform('EPSG:4326', 'EPSG:3857');
```

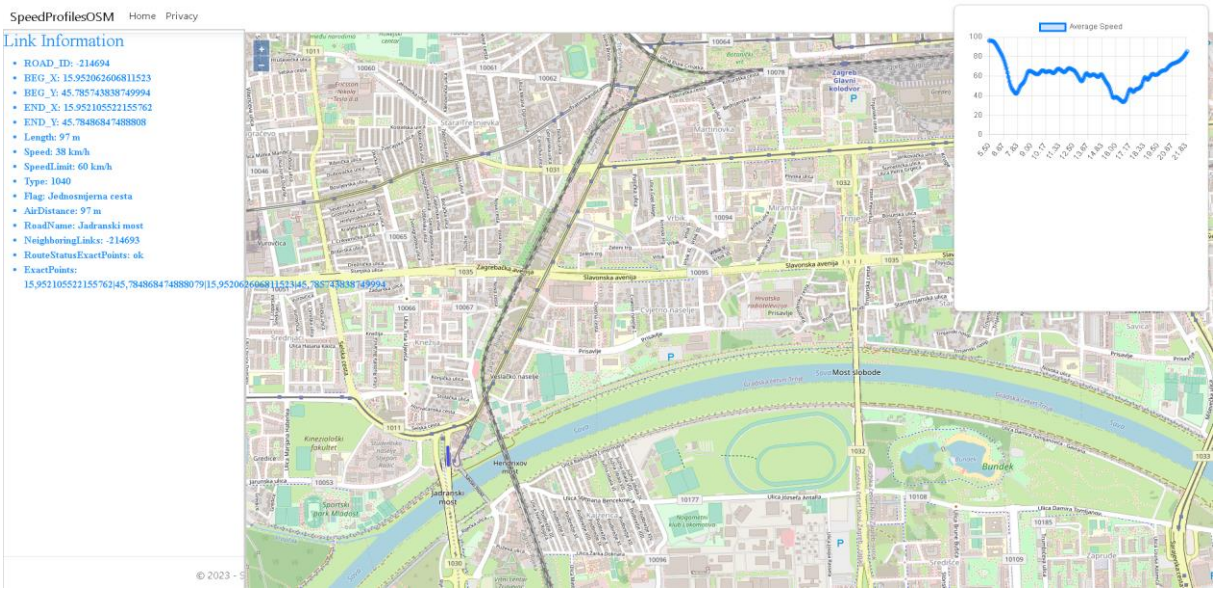
Slika 46. Funkcija succCallGetLinkForClick

Varijabla 'iconFeature' stvara novu značajku koja koristi polilinije „polyline“ kao svoju geometriju. Naredba 'iconFeature.setStyle postavlja stil za novu značajku. Također se definira stil linije i postavlja joj boju kao plavu. „vectorSource.addFeature(iconFeature)“ dodaje stvorenu ikonu. Linija „activePlotedLink“ omogućuje kasnije brisanje iscrtane ikone ako je potrebno (Slika 47.).

```
//Napravi novu OSM ikonu/znacajku
var iconFeature = new ol.Feature({
    geometry: polyline,
});
//Postavi stil
iconFeature.setStyle(new ol.style.Style({
    stroke: new ol.style.Stroke({
        color: 'rgba(0,0,255,0.7)',
        width: 5
    })
}));
//Dodaj je u vektorski sloj
vectorSource.addFeature(iconFeature);

//Postavi da je trenutno iscrtan bas ta ikona, tako da je kasnije mozes lako pobrisati
activePlotedLink = iconFeature;
```

Slika 47. Funkcija za izradu nove OSM ikone

Ukoliko je odabrana točka na karti više od 70 metara udaljena od najbližeg linka tada se pojavljuje popup odnosno skočni prozor s porukom „Location of the point is too far from any link!“. U slučaju odabira izvan granica linije dolazi do korigiranja na najbližu točku na liniji. Nakon uspješnog duplog klika označuje se prikaz najbliže točke plavom linijom te se istovremeno prikazuje geografska dužina i širina, duljina linka u metrima, brzina te ograničenje brzine, ime ceste, ID najbližeg linka i niz geografskih točka koje detaljno opisuju link, te se na popup-u pod nazivom 'Average Speed' iscratava graf s profilima brzine o odabranom linku (Slika 48.).



Slika 48. Rezultat duplog klika na kartu

5. ANALIZA REZULTATA

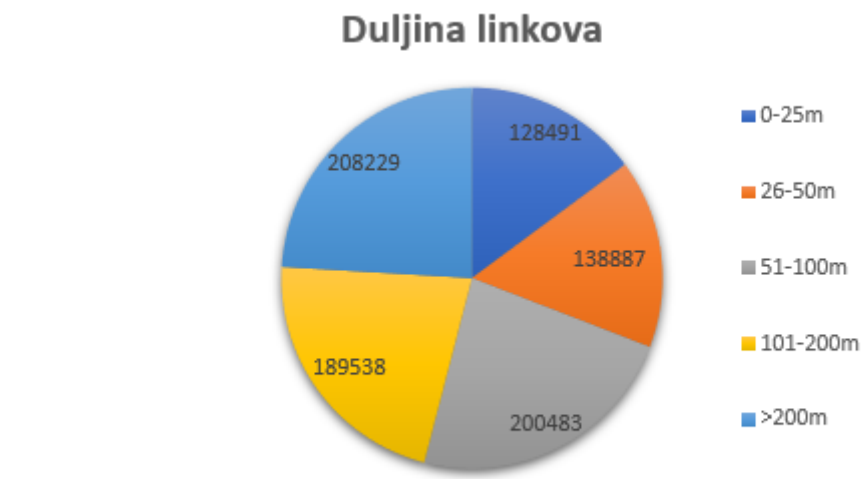
U ovom dijelu rada će se provest analiza dobivenih podataka dobivenih od strane Mireo d.d. Za analizu dobivenih podataka će se istovremeno koristit Microsoft SQL Server i te za njihov prikaz će se koristit Microsoft Excel.Podaci koji će se analazirati su:

- Raspodjela po duljinama linkova
- Raspodjela postavljenih brzina
- Raspodjela ograničenja brzina
- Raspodjela vrsta cesta

Ukupan broj podataka iznosi 865,628. Raspodjela za analizu podjele duljine linkova je razdvojena u pet grupa. Prvi skup je za linkove u duljini od 0 do 25 metara, drugi skup za linkove u duljini od 26 do 50 metara, treći skup za linkove u duljini od 51 do 100, četvrti skup za linkove u duljini od 101 do 200 metara i peti odnosno zadnji skup u duljini većoj od 200 metara. Potrebne SQL naredbe za prikaz iznosa za svaki pojedini skup je prikazan na slici (Slika 49.). Za vizualizaciju podataka je prikazan graf i tablica navedenih podataka (Slika 50). Iz grafa se vidi da najveći udio linkova čine linkovi s duljinom većom od 200 m dok najmanji udio čine segmenti u duljini od 0 do 25 m i 25 do 50 m.

```
select LENGTH_M from MireoMap
where LENGTH_M>0 and LENGTH_M<26

select LENGTH_M from MireoMap
where LENGTH_M>25 and LENGTH_M<51

select LENGTH_M from MireoMap
where LENGTH_M>50 and LENGTH_M<101

select LENGTH_M from MireoMap
where LENGTH_M>100 and LENGTH_M<201

select LENGTH_M from MireoMap
where LENGTH_M>200
```

Slika 49. Sql Naredbe za prikaz iznosa



Slika 50. Graf za podatke

Tablica 2. Prikaz podataka o duljini linkova s postocima

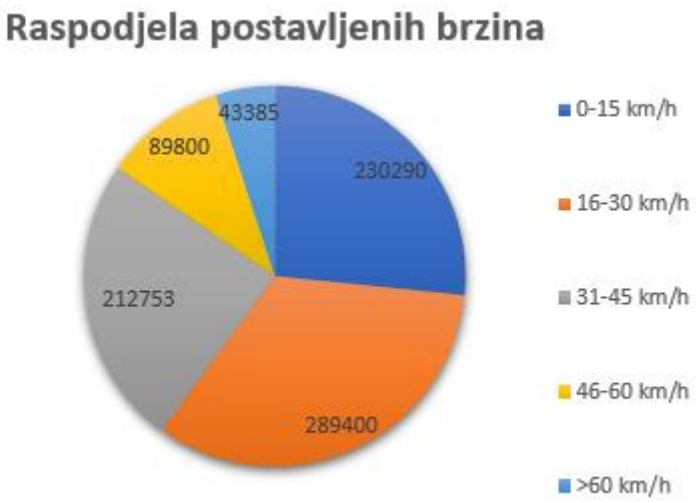
Duljina linkova	Broj linkova	Postotak
0-25m	128 491	14,844%
26-50m	138 887	16,045%
51-100m	200 483	23,160%
101-200m	189 538	21,896%
>200m	208 229	24,055%

Raspodjela za analizu postavljenih brzina od strane Mireo koja se odnosi na statične brzine koje se ne mijenjaju ovisno o vremene u danu ili tjednu je razdvojena u četiri grupa. Prvi skup je za brzine od 0-15 km/h, drugi skup je za brzine od 16-30 km/h, treći skup je za brzine od 31-45 km/h, četvrti skup je za brzine od 46-60 km/h dok je peti skup za brzine veće od 60 km/h. SQL naredbe za prikaz iznosa za svaki pojedini skup je prikazan na slici (Slika 51.). Za vizualizaciju podataka je prikazan graf i tablica navedenih podataka s postotnim udjelom (Slika 52.). Iz grafa je vidljivo da najveći udio čine postavljene brzine od 16-30 km/h

dok najamnji udio čine postavljene brzine u iznosima većim od 60 km/h koje se većinskim udjelom odnosi na brze ceste i autoceste.

```
select SPEED from MireoMap
where SPEED>0 and SPEED<16

select SPEED from MireoMap
where SPEED>15 and SPEED<31

select SPEED from MireoMap
where SPEED>30 and SPEED<46

select SPEED from MireoMap
where SPEED>45 and SPEED<61

select SPEED from MireoMap
where Speed>60
```

Slika 51. Sql naredbe za prikaz iznosa



Slika 52. Graf za podatke

Tablica 3. Prikaz podataka o brzinama s postocima

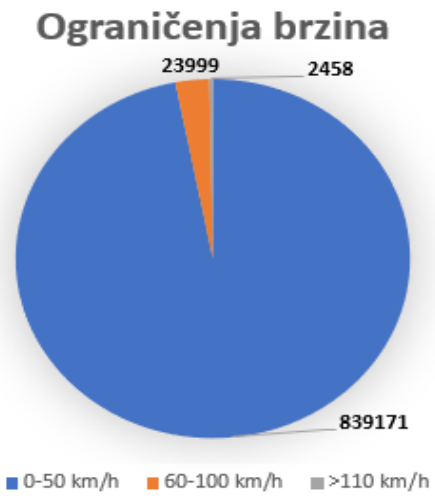
Brzina linkova	Broj linkova	Postotak
0-15 km/h	230290	26,604%

16-30 km/h	289400	33,432%
31-45 km/h	212753	24,578%
46-60 km/h	89800	10,374%
>60 km/h	43385	5,012%

Raspodjela za analizu postavljenih ograničenja brzina od strane Mireo je razdvojena u tri skupa. Prvi skup je za ograničenja od 0 do 50 km/h, drugi skup je za ograničenja od 60 do 100 km/h, treći skup je za ograničenja veća od 110 km/h. U prvoj skupini od 839171 linkova 791058 njih čine linkovi čija su ograničenja postavljena na 0 km/h. SQL naredbe za prikaz iznosa za svaki pojedini skup je prikazan na slici (Slika 53.). Za vizualizaciju podataka je prikazan graf i tablica navedenih podataka s postotnim udjelom (Slika 54.). Graf za postavljena ograničenja brzina prikazuje daleko veći udio ograničenja u iznosima od 0 do 50 m dok manje od 1% čine postavljena ograničenja u iznosima većim od 110 km/h koje se većinom odnose na autoceste.

```
SELECT SPEED_LIMIT, COUNT(*) AS Count
FROM MireoMap
GROUP BY SPEED_LIMIT;
```

Slika 53. Sql naredba za prikaz iznosa



Slika 54. Graf za podatke

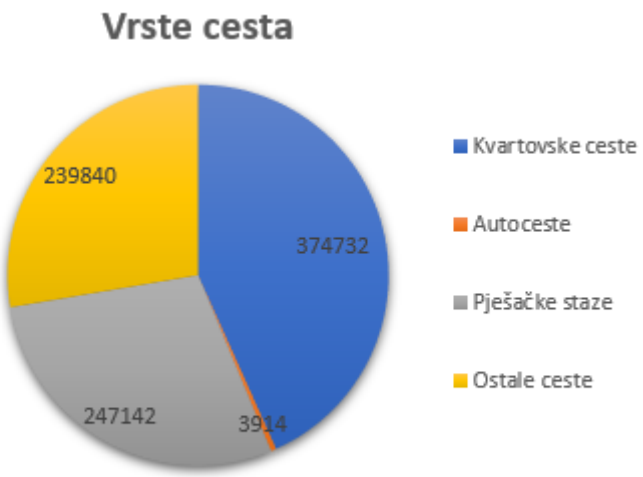
Tablica 4. Prikaz podataka o ograničenjima brzina

Ograničenja brzina	Broj linkova	Postotak
0-50 km/h	839171	96,945%
60-100 km/h	23999	2,772%
>110 km/h	2458	0,283%

Raspodjela za kategorije link koju je postavio Mireo je razdvojena u 4 skupine. Prvi skup je za autoceste, drugi skup je za kvartovske ceste, treći skup je za pješačke staze dok je četvrti skup za ostale ceste. SQL naredbe za prikaz iznosa za svaki pojedini skup je prikazan na slici (Slika 55.).Za vizualizaciju podataka je prikazan graf i tablica navedenih podataka s postotnim udjelom (Slika 56.). Iz grafa se vidi da vrlo mali udio čine autoceste dok te daleko veća zastupljenost kvartovskih cesta.

```
select TIP from MireoMap
where TIP=1080

select TIP from MireoMap
where TIP=1010

select TIP from MireoMap
where TIP>1080 or TIP<1010

select TIP from MireoMap
where TIP<1080 and TIP>1010
```

Slika 55. Sql naredba za prikaz iznosa



Slika 56. Graf za podatke

Tablica 5. Prikaz podataka o vrsti cesta

Vrste cesta	Broj linkova	Postotak
Kvartovske ceste	374732	43,290%
Autoceste	3914	0,452%
Pješačke staze	247142	28,551%
Ostale ceste	239840	27,707%

6. Zaključak

U ovom radu su obrađeni podaci koji su preuzeti s projekta SORDITO. Podaci koji sadrže informacije kao što geografska pozicija i profili brzina kretanja. Cilj rada je za navedene podatke izraditi bazu podataka i sukladno tome web aplikaciju koja pomaže pri vizualizaciji korisnicima i određivanju zagušenja na prometnicama. Za vizualizaciju je izrađeno grafičko web sučelje pomoću ASP.NET platforme i potrebnih NuGet paketa. Kroz poglavlja u radu su opisane metode i potrebni postupci koji su potrebni da bi se došlo do konačnog rješenja.

S povećanjem osobnih i teretnih vozila na prometnicama zagušenja su sve više prisutnija i samim time potreba za tehnologije koje mogu na učinkovit i optimalan način vodit promet u gradovima. Posljedice optimalnog vođenja prometa su smanjenja zagušenja, smanjenja repove čekanja, smanjenje frustracije vozača odnosno putnika, smanjenje kašnjenja žurnih službi, smanjenje vremena dostave, bolje upravljanje rizicima i sukladno tome povećanje sigurnosti na prometnica i smanjen broj prometnih nezgoda i stradalih u prometu.

LITERATURA

[1] Bošnjak I.*, Inteligentni transportni sustavi – ITS 1*, Fakultet prometnih znanosti, Sveučilište u Zagrebu, Zagreb, 2006.

[2] 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.

[3] Autorizirana predavanja iz kolegija Lokacijski i navigacijski sustavi: https://moodle.srce.hr/2021-2022/pluginfile.php/6092525/mod_resource/content/5/05-Satelitski%20pozicijski%20sustavi.pdf

[4] ArcGIS Enterprise, CSV, TXT, and GPX files: https://enterprise.arcgis.com/en/portal/10.7/use/csv-gpx.htm#:~:text=CSV%20and%20TXT%20files%20store,comma%2C%20semicolon%2C%20or%20tab.

[5] Microsoft, Import or export .txt or .csv files: https://support.microsoft.com/en-au/office/import-or-export-text-txt-or-csv-files-5250ac4c-663c-47ce-937b-339e391393ba

[6] Microsoft, What is SQL Server Management Studio: https://learn.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16

[7] Autorizirana predavanja iz kolegija Baze podataka: https://moodle.srce.hr/2022-2023/pluginfile.php/6852898/mod_resource/content/11/Vje%C5%BEba3.pdf

[8] Microsoft, Visual Database Tools: https://learn.microsoft.com/en-us/sql/ssms/visual-db-tools/create-a-new-database-diagram-visual-database-tools?view=sql-server-ver16

[9] Hasura, Entity Relationship Model: https://hasura.io/learn/database/microsoft-sql-server/er-modeling/

[10] Autorizirana predavanja iz kolegija Baze podataka: https://moodle.srce.hr/2022-2023/pluginfile.php/6852999/mod_resource/content/5/Vje%C5%BEba14.pdf

[11] Microsoft, StreamReader Class: https://learn.microsoft.com/en-us/dotnet/api/system.io.streamreader?view=net-7.0

[12] Microsoft, What is ASP.NET: https://dotnet.microsoft.com/en-us/learn/aspnet/what-is-aspnet

[13] Microsoft, A tour of the C# language: https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/

[14] Julie C. Meloni, HTML, CSS and JavaScript: https://www.doc-developpement-durable.org/file/Projets-informatiques/cours-&-manuels-informatiques/htm-html-xml-ccs/Sams%20Teach%20Yourself%20HTML,%20CSS,%20and%20JavaScript%20All%20in%20One.pdf

[15] GeeksForGeeks, FrontEnd vs BackEnd:https://www.geeksforgeeks.org/frontend-vs-backend/

[16] GeeksForGeeks,JavaScript Chart.js:https://www.geeksforgeeks.org/javascript-chart-js/?ref=gcse

[17] T. Erdelić, M. Erdelić, T. Carić, L. Tišljarić, A. Turković, N. Jelušić**,** Estimating congestion zones and travel time indexes based on the floating car data: https://www.sciencedirect.com/science/article/pii/S0198971521000119#bb0340

POPIS SLIKA

Slika 1. Konstelacija satelita GPS sustava.....5

Slika 2. Prikaz podataka u programu Notepad++.....6

Slika 3. Format zapisa svakog linka.....7

Slika 4.Prikaz profila brzine za prvih 19 stupaca (5:30-7:05) u programu Notepad++.....11

Slika 5. Prikaz podataka iz datoteke InfoSP.txt u programu Notepad++.....12

Slika 6. Prikaz podataka iz datoteke InfoAVG.txt u programu Notepad++.....13

Slika 7. Prikaz 1:1 veze.....15

Slika 8. Prikaz 1:N veze.....16

Slika 9. Prikaz M:N veze.....16

Slika 10. Dijagram entiteta 'MireoMap'.....18

Slika 11: Dijagram entieta 'Info_AVG'.....18

Slika 12: Dijagram entiteta 'Info_SP'.....19

Slika 13. Prikaz ER dijagrama relacijske baze podataka.....20

Slika 14. Skripta za kreiranje tablice MireoMap i baze podataka Zavrsni.....22

Slika 15. Skripta za kreiranje tablice Info_AVG.....23

Slika 16. Skripta za kreiranje tablice Info_SP.....23

Slika 17. Dijagram baze podataka.....24

Slika 18. Klase potrebne za unos podataka.....25

Slika 19. SqlConnection sa podacima za spajanje na bazu.....25

Slika 20. C# koda za čitanje tekstualne datoteke MireoMap.....27

Slika 21. Ubacivanje u SQL tablicu MireoMap.....28

Slika 22. Zatvaranje StreamReader-a i konekcije na bazu.....29

Slika 23. Sql upit: Select * from MireoMap.....29

Slika 24. Skripta za prebacivanje podataka iz tekstualne datoteke u tablicu Info_AVG.....30

Slika 25. StreamReader za datoteke InfoSP i Spdata2.....30

Slika 26. Skripta za prebacivanje podataka iz tekstualne datoteke u tablicu Info_SP.....31

Slika 27. Korišten Nuget paket.....32

Slika 28. Klase za pristup Sql bazi podataka.....32

Slika 29. Podaci koji se vraćaju natrag u web.....35

Slika 30. Klasa LinkInfo.....36

Slika 31. Skripta koja čita tablicu MireoMap te ih predaje u LinkInfo.....37

Slika 32. Odabir redaka iz tablice Info_SP.....38

Slika 33. Prikaz moguće greške i udaljenosti veće od 70 metara.....38

Slika 34. Metoda za računanje pomoćne zračne udaljenosti.....39

Slika 35. Metoda getDistanceFromPointToClosestPointOnLine.....39

Slika 36. OSM karta.....40

Slika 37. Dio koda koji postavlja Zagreb kao centar.....40

Slika 38. Div element za prikaz podataka o odabranom linku.....41

Slika 39. Prikaz svakog elementa za odabrani link.....41

Slika 40. Link chart.js-a.....42

Slika 41. Klasa chartCanvas.....42

Slika 42. Priprema podataka za pohranu.....43

Slika 43. Dizajn grafa.....43

Slika 44. Kod za izgled grafa i za pohranu odnosno pretvorbu podatka 'avgspeed'.....44

Slika 45. Metoda za dupli klik na kartu.....44

Slika 46. Funkcija succCallGetLinkForClick.....45

Slika 47. Funkcija za izradu nove OSM ikone.....45

Slika 48. Rezultat duplog klika na kartu.....46

Slika 49. Sql Naredbe za prikaz iznosa.....47

Slika 50. Graf za podatke.....48

Slika 51. Sql Naredbe za prikaz iznosa.....49

Slika 52. Graf za podatke.....49

Slika 53. Sql Naredbe za prikaz iznosa.....50

Slika 54. Graf za podatke.....50

Slika 55. Sql Naredbe za prikaz iznosa.....51

Slika 56. Graf za podatke.....52

POPIS TABLICA

Tablica 1. Detaljan opis pojedinog elementa za svaki link.....7

Tablica 2. Prikaz podataka o duljini linkova s postocima.....48

Tablica 3. Prikaz podataka o brzinama s postocima.....49

Tablica 4. Prikaz podataka o ograničenjima brzina.....51

Tablica 5. Prikaz podataka o vrsti cesta.....52

Sveučilište u Zagrebu
Fakultet prometnih znanosti
Vukelićeva 4, 10000 Zagreb

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOSTI

Izjavljujem i svojim potpisom potvrđujem da je _____završni rad_____
(*vrsta rada*)
isključivo rezultat mojega vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju upotrijebljene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedopušten način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da 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/diplomskog rada pod naslovom ___Web_aplikacija_za_prikaz_prometnih_zagušenja_na_cestovnoj mreži____ , u Nacionalni repozitorij završnih i diplomskih radova ZIR.

Student/ica:

U Zagrebu, ____2023._____ _____
(ime i prezime, *potpis*)