Prostorno vremenska analiza podataka o prometnim nesrećama

Crnko, Karlo

Undergraduate thesis / Završni rad

2019

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

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:119:326412

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**



Repository / Repozitorij:

Faculty of Transport and Traffic Sciences - Institutional Repository







SVEUČILIŠTE U ZAGREBU

FAKULTET PROMETNIH ZNANOSTI

Karlo Crnko

PROSTORNO VREMENSKA ANALIZA PODATAKA O PROMETNIM NESREĆAMA

ZAVRŠNI RAD

SVEUŠILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

ODBOR ZA ZAVRŠNI RAD

Zagreb, 1. travnja 2019.

Zavod: **Zavod za inteligentne transportne sustave**

Predmet: **Baze podataka**

ZAVRŠNI ZADATAK br. 5163

Pristupnik: **Karlo Crnko (0135243444)**

Studij: Inteligentni transportni sustavi i logistika Smjer: Inteligentni transportni sustavi

Zadatak: **Prostorno vremenska analiza podataka o prometnim nesrecama**

Opis zadatka:

Zadatak ovog završnog rada je izraditi korisničko sučelje pomoću kojeg će se na digitalnoj karti prikazati prometne nesreće na području Republike Hrvatske za razdoblje jedne godine. Prije izrade samog sučelja dobivene podatke je potrebno filtrirati i prilagoditi formatom za pohranu u relacijsku bazu podataka. U grafičkom korisničkom sučelju potrebno je omogućiti odabir vremenskog razdoblja za koje se žele prikazati lokacije prometnih nesreća. Na ovaj način vidljive su žarišne točke na području na kojem su prometne nesreće učestalije.

Mentor: Predsjednik povjerenstva za završni ispit:

prof. dr. sc. Hrvoje Gold

Zagreb, 2019.

Sveučilište u Zagrebu

Fakultet prometnih znanosti

ZAVRŠNI RAD

PROSTORNO VREMENSKA ANALIZA PODATAKA O PROMETNIM NESREĆAMA

SPATIAL-TEMPORAL ANALYSIS OF TRAFFIC ACCIDENTS

Mentor:

prof. dr. sc. Hrvoje Gold

Student:

Karlo Crnko, 0135243444

Neposredni voditelj:

mag. ing. traff. Martina Erdelić

Zagreb, 2019.

PROSTORNO VREMENSKA ANALIZA PODATAKA O PROMETNIM NESREĆAMA

SAŽETAK

Zadatak završnog rada jest izrada grafičkog korisničkog sučelja koji prikazuje kritična mjesta, odnosno prometne nesreće unutar vremenskog perioda od jedne godine. Podaci su dobiveni od strane MUP-a, koji su zabilježeni u 2014. godini. Podatke je potrebno obraditi i pohraniti unutar baze podataka, koji se kasnije prikazuju u sučelju. Podaci sadrže informacije o geografskoj lokaciji koji se prikazuju na digitalnoj karti, uz odgovarajući opis kritičnih mjesta. Cilj je prikazati manipulaciju velikih količina podataka, koji se mogu koristiti za izradu statističkih analiza.

KLJUČNE RIJEČI: kritična mjesta, baze podataka, geografska lokacija, grafičko korisničko sučelje

SUMMARY

The task of the final paper is to create a graphical user interface that shows the critical points, is traffic accidents within a period of one year. The data were obtained by the Ministry of Interior, which was recorded in 2014. The data must be processed and stored within the database, which will later be displayed in the interface. The data captures geographical location information that is displayed on the digital map, with an appropriate description of the critical locations. The goal is to show the manipulation of large amounts of data, which can be used to make statistical analysis.

KEYWORDS: critical locations, databases, geographical location, graphical user interface

SADRŽAJ

1. UVOD 1
2. ODREĐIVANJE KRITIČNIH MJESTA PROMETNE MREŽE 2
3. METODE PRIKUPLJANJA I FILTRIRANJA PODATAKA O KRITIČNIM MJESTIMA 5
3.1. Induktivne petlje 5
3.2. Video kamere 6
3.3. Sustav za globalno pozicioniranje 6
3.4. Globalni sustav pokretnih komunikacija 7
4. POHRANA PODATAKA O KRITIČNIM MJESTIMA U RELACIJSKU BAZU 9
4.1. Kreiranje relacijske baze podataka 11
4.1.1. Dijagram entiteta 11
4.1.2. ER dijagram 13
4.1.3. Relacijski model podataka 15
4.2. Pohrana podataka u kreiranu relacijsku bazu podataka 16
5. PRIKAZ PROSTORNE DISTRIBUCIJE PODATAKA I ODREĐIVANJE KRITIČNIH MJESTA 18
5.1. Microsoft Visual Studio 18
5.2. Povezivanje baze podataka s aplikacijom 19
5.2.1. NuGet paketi 20
5.2.2. Programiranje sučelja i filtracija podataka 20
6. ZAKLJUČAK 28
LITERATURA 29
POPIS SLIKA 30
POPIS GRAFIKONA 31
IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST 32

1. UVOD

Razvoj društva kroz razvoj tehnologije uvelike su promijenili način na koji živimo. U svakodnevnici teži se prema maksimalnoj optimizaciji, efikasnosti i preciznosti organizacijskih sustava. Ono što sustav čini efikasnijim i preciznijim jest brza obrada i analiza podataka dobivenih iz realnog svijeta. Upravo u informacijskim znanostima razvoj tehnologije omogućava prikupljanje i manipulaciju velike količine podataka. U tom aspektu informacijskih znanosti leži problematika pohranjivanja i pretraga većih količina podataka čime se bave baze podataka.

Prostorno vremenska analiza podataka primijenjena na prometnu mrežu predstavlja postupak prikupljanja podataka koji opisuju pojave na određenom dijelu prometne mreže u nekom vremenskom trenutku. Neuobičajene pojave ili incidentne situacije na prometnoj mreži predstavljaju događaje koji negativno utječu na prometne uvjete. Kako bi se mogli identificirati uzroci koji su pridonijeli zagušenju prometnih tokova, prikupljaju se podaci o kretanju prometnog toka. Ako se podaci o incidentnim situacijama na prometnoj mreži prikupljaju kroz duži vremenski period, iz podataka se mogu raspoznati ponavljajući uzorci takvih događaja. Tako se može detektirati kroz učestalost prometnih nesreća na nekom dijelu prometne mreže kritično mjesto, odnosno mjesto na kojem je veća vjerojatnost ponavljanja istog ili sličnog događaja.

U sklopu završnog rada obrađeni su i pohranjeni u bazu podataka podaci o prometnim nesrećama koje je prikupilo Ministarstvo unutarnjih poslova Republike Hrvatske. Izrađena baza podataka povezana je s aplikacijom pomoću koje je omogućena vizualizacija pohranjenih podataka. U drugom poglavlju opisan je pojam kritičnih mjesta prometne mreže te način na koji se kritična mjesta određuju. Metode prikupljanja i filtriranja podataka o kritičnim mjestima prikazane su u trećem poglavlju. U četvrtom poglavlju prikazana je struktura izrađene baze podataka te su opisani korišteni podaci. Izrada i funkcionalnost aplikacije koja se koristi za vizualizaciju upita iz baze podataka opisana je u petom poglavlju. U zadnjem poglavlju dani su zaključci rada.

2. ODREĐIVANJE KRITIČNIH MJESTA PROMETNE MREŽE

Promet se definira kao složeni dinamički sustav koji ima za cilj prijevoz robe i ljudi te prijenos informacija s jednog mjesta na drugo uz pomoć prijevoznih i prijenosnih sredstava u određenom vremenskom periodu. Da bi takav dinamički sustav mogao normalno funkcionirati, djelovanje svih sudionika u tom sustavu kao i čimbenici koji utječu na sustav moraju biti međusobno povezani, a njihovi postupci usklađeni. Ako dođe do nekog slučajnog, nepredvidivog ili neplaniranog događaja (sudar, vremenske neprilike, zatvorena prometnica i sl.) onda dolazi do poremećaja u djelovanju sustava koji se zove incident. Incident je događaj koji uzrokuje smanjenje kapaciteta prometnice ili neuobičajeno povećanje potražnje. Incidenti u prometu koji se događaju na određenim prometnicama, tj. mjestima s većom učestalošću mogu se nazvati kritičnim mjestima.

Kako bi se odredila kritična mjesta provode se statističke analize koje uspoređuju registrirani i uobičajeni broj nesreća. Određeno mjesto u prometnoj mreži bit će definirano kao kritično mjesto ako je registrirani broj prometnih nesreća veći od kritične razine broja prometnih nesreća. Za smanjenje prometnih nesreća i poboljšanje sigurnosti na cestama od presudne je važnosti razumjeti kako, gdje i kada su se dogodile prometne nesreće. Postoje razni utjecaji zbog kojih se događaju prometne nesreće. Nesreće mogu biti uzrokovani radi čovjekove pogreške u prijevozu, fizičkog okruženja (strmi nagib, oštar zavoj), vremenskih prilika (kiša, snijeg, vjetar, magla), nepravilnog projektiranja, održavanjem cesta i slično [2]. Razumijevanje uzroka prometnih nesreća može smanjiti broj prometnih nesreća zbog toga što zakonodavne vlasti mogu svojim mjerama poboljšati regulaciju i nadzor prometa nad tim kritičnim mjestima. Na primjer na nekim dionicama ceste mogu se ograničiti brzine kretanja, omogućavanje veće preglednosti u zavoju i slično [8].

Prema podacima Svjetske zdravstvene organizacije**,** u prometnim nesrećama svakoga dana između 20 i 50 milijuna osoba zadobije tjelesne ozljede. U tim nesrećama više od 3600 osoba smrtno nastrada, a oko 15 000 osoba zadobije tjelesne ozljede koje imaju za posljedicu trajni invaliditet. U Europskoj uniji tijekom 2017. godine poginulo je 49 sudionika u prometu na milijun stanovnika, dok je u Hrvatskoj poginulo njih 80.

Kako bi se identificirala kritična mjesta postoje razne tehnologije prikupljanja podataka o prometnoj mreži o kojima će više biti rečeno u idućem poglavlju. Razvoj tehnologija u prometu je uvelike promijenio način mjerenja prometnih parametara, što je rezultiralo razvojem nove grane uprometu inteligentni transportni sustavi (engl. Intelligent Transport Systems, ITS). ITS

se može definirati kao holistička, upravljačka i informacijsko-komunikacijska nadgradnja klasičnog sustava prometa i prijevoza kojim se postiže znatno poboljšanje performansa, odvijanje prometa, učinkovitiji prijevoz putnika i roba, poboljšanje sigurnosti u prometu, udobnost i zaštita putnika, manja onečišćenja okoliša, itd. [9]. Jedna od temeljnih usluga kojima se bavi ITS je nadzor i otklanjanje incidenata. Nadzor i otklanjanje incidenata u prometu predstavlja skup koordinacijskih aktivnosti od strana viših aktera koja ima za cilj povratak prometnog toka u stanje normalnog režima rada nakon odvijanja incidentnog događaja.

U Republici Hrvatskoj Zakonom o sigurnosti prometa na cestama prometna nesreća definirana je kao događaj na cesti u kojem je sudjelovalo najmanje jedno vozilo u pokretu i u kojem je najmanje jedna osoba ozlijeđena ili poginula ili u roku od 30 dana preminula od posljedica te prometne nesreće ili je izazvana materijalna šteta [7].

Uzroci prometnih nesreća su više ili manje povezani s različitim prostornim parametrima, a iz analize parametara točkastih događaja (engl. Point Patern Analysis, PPA) razvile su se razne metode za otkrivanje kritičnih mjesta. Metode koje se koriste mogu se svrstati u dva široka tipa. Prvi tip su metode ispitivanja efekata prvog reda prostornog procesa, odnosno mjerenje varijacija udaljenosti točkastih događaja. Drugi tip su metode ispitivanja efekata drugog reda, odnosno ispitivanje strukture prostornih interakcija (ovisnosti) točaka. Prvi tip metoda temelji se na udaljenostima između točaka (srednja udaljenost do najbliže susjedne točke), dok se drugi tip metoda temelji na području, oslanjajući se na različite karakteristike frekvencijski distribuiranim točaka u promatranim podregijama istraživanog područja („kvadrati“) [2].

Prva metoda, ujedno i najjednostavnija, bazira se na analizi najbližeg susjeda gdje se ispituju udaljenosti svih točaka, a zatim se određena točka pridjeljuje pojedinoj grupi. Grupiranje točaka provodi se na način da se raščlanjuje skup točaka na manje podskupove koji se međusobno ne preklapaju. Svaka točka se mora razvrstati, te svaka točka može biti član samo jedne grupe. Točka se pridjeljuje u onaj podskup čije su udaljenosti između točaka u najmanje, odnosno najbliže [2].

Procjena gustoće zrna (engl. Kernel Densitiy Estimation, KDE) predstavlja jednu od najpopularnijih metoda za analizu svojstava prvog reda točkastih događaja. Svrha KDE metode jest procjena gustoće jezgre koja se temelji na intenzitetu pojavljivanja događaja za procjenu gustoće [2].

Za određivanjem kritičnih mjesta potrebno je odrediti mjerila temeljena po prikupljenim podacima o prometnim nesrećama po kojima se mogu određena područja u prometnoj mreži

klasificirati kao kritično mjesto. Iz navedenih metoda statističkim ispitivanjem pojedinih područja dobivaju se kritične vrijednosti na način uspoređivanja područja prometnih nesreća s drugim područjima sličnih karakteristika. Kao mjerila za klasificiranjem kritičnih mjesta uzimaju se u obzir prometno opterećenje, duljina lokacije i vremenski period određenog područja koja prelazi definiranu kritičnu razinu.

Mjerila koja se moraju udovoljiti da bi se kritično mjesto moglo nazvati kao takvo jesu:

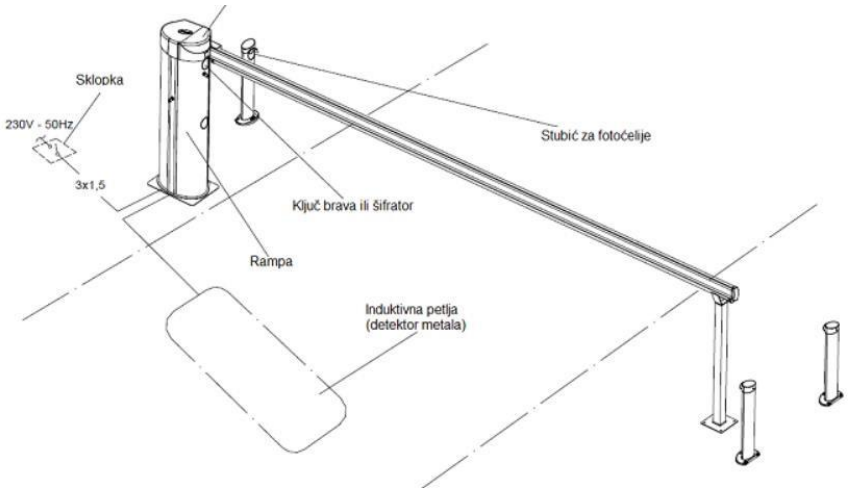
- ako se na kritičnoj lokaciji u prethodne 3 godine dogodilo 12 ili više prometnih nesreća s ozlijeđenim osobama;
- ako je u prethodne 3 godine na prometnoj lokaciji evidentirano 15 ili više prometnih nesreća bez obzira na posljedice;
- ako su se na kritičnoj lokaciji, u prethodne 3 godine, dogodile 3 ili više istovrsnih prometnih nesreća, u kojima su sudjelovale iste skupine sudionika, s istim pravcima kretanja, na istim konfliktnim površinama [3].

3. METODE PRIKUPLJANJA I FILTRIRANJA PODATAKA O KRITIČNIM MJESTIMA

Za detekciju prometnih nesreća najprije mora postojati sustav koji omogućuje prikupljanje podataka o prometnoj mreži. Smisao ITS-a jest umreženi sustav u kojem se međusobno prosljeđuju informacije između infrastrukture i entiteta putem komunikacijskih medija. Da bi se informacije mogle detektirati, u infrastrukturu i entitete moraju biti ugrađene tehnologije koje omogućavaju prikupljanje, obradu i slanje informacija. Prikupljeni podaci najčešće se šalju u centralizirani prometni centar gdje se podaci obrađuju. Podaci se uglavnom prikupljaju putem senzora. Senzore možemo podijeliti na statične i dinamične. Statični senzori predstavljaju fiksne objekte na prometnicama kao što su induktivne petlje, prometne kamere, magnetski senzori i slično. Dinamički senzori predstavljaju pokretne objekte na prometnicama, a sadrže sustav za globalno pozicioniranje (engl. Global Positioning System, GPS) za određivanje položaja na zemlji. Dalje u tekstu objasnit će se pojedini senzori.

3.1. Induktivne petlje

Induktivna petlja se koristi za detekciju vozila po principu promjene induktiviteta žičane petlje. Žičani dio induktivne petlje se ugrađuje u beton ili asfalt i zaljeva se specijalnom smjesom radi sprječavanja prodora vode u kanal u kome se nalazi žičani dio petlje.



Slika 1. Shematski prikaz induktivne petlje

Izvor: http://www.automatskisistemi.rs/parking-rampa/rampe-daphne-4/

Kada naiđe vozilo preko žičane petlje, u magnetnom polju petlje dolazi do promjene induktiviteta i detektor induktivne petlje to registrira. Detektor induktivne petlje šalje signal elektronici na koju je povezan kao što je prikazano na slici 1. Induktivne petlje mogu se koristiti za različite namjene u prometu: parking rampe, semafore, brojanje prometa, mjerenje brzine vozila, itd.

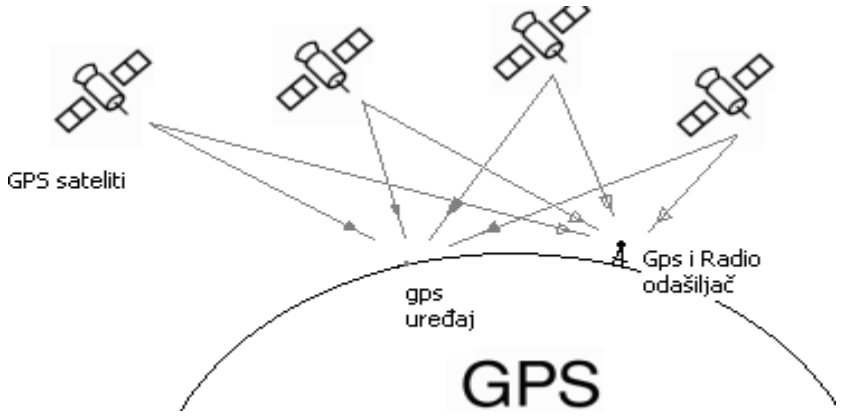
3.2. Video kamere

Video kamere predstavljaju sustav video detekcije koji radi na principu slanja slika u procesor koji analizira podatke. Najčešće se postavljaju na visokim stupovima pored prometnica. Video kamere koriste se najčešće za mjerenje brzine vozila po traku, brojanje i klasifikaciju vozila.

3.3. Sustav za globalno pozicioniranje

Sustav za globalno pozicioniranje predstavlja satelitski radionavigacijski sustav za određivanje položaja na zemlji. GPS sustav podijeljen je u tri segmenta: svemirski, kontrolni i korisnički. Na slici 2 prikazan je svemirski segment koji čine GPS sateliti, kontrolni segment koji čini kontrolnu stanicu (na slici GPS i radio odašiljač), te korisnički segment koji čini GPS uređaj.

Svemirski segment čine 32 satelita koji se gibaju oko Zemlje u kružnim orbitama na visini 20 000 km. Raspored satelita unutar 6 orbitalnih ravnina je tako odabran da se u svakom trenutku iznad horizonta nalazi barem pet ili više satelita, čime se postiže globalna pokrivenost. Sateliti emitiraju signale na dvjema prijenosnim frekvencijama na koje se moduliraju kodovi za pozicioniranje i navigacijske poruke. Unutar satelita nalaze se precizni i stabilni atomski satovi koji osiguravaju potrebnu stabilnost signala i usklađenost s jedinstvenim GPS vremenom.



Slika 2. Prikaz lociranja pozicije putem GPS-a

Izvor: https://usporedi.hr/teme/pozicioniranje-na-mobitelu-bez-gps-a

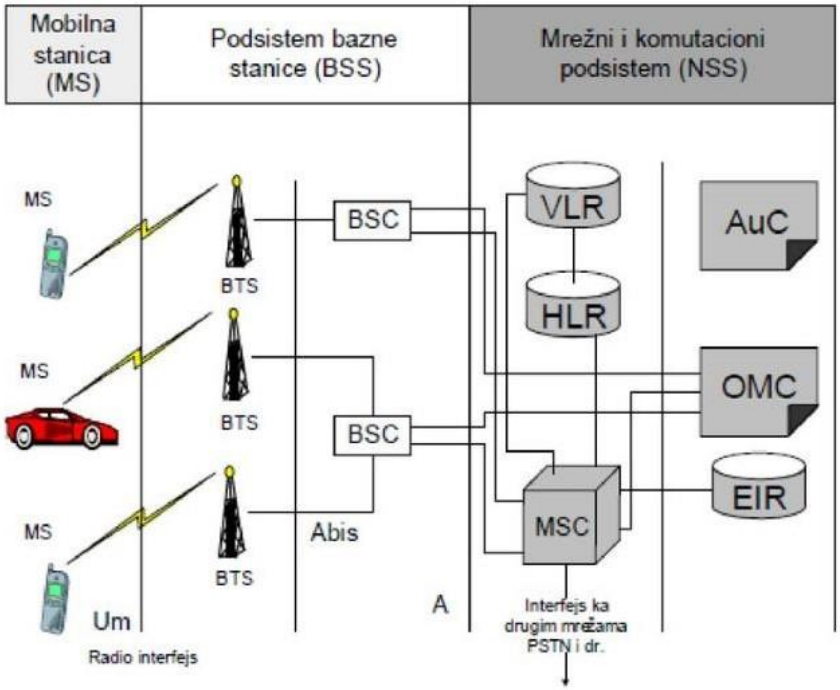
Kontrolni segment obavlja nadzor i upravlja cijelim sustavom. Po čitavoj Zemlji su raspoređene kontrolne stanice koje kontinuirano prate sve GPS satelite i prosljeđuju primljene satelitske signale u glavnu kontrolnu postaju na obradu. U glavnoj postaji izračunavaju se odstupanja pozicija svakog satelita od preciznih orbitalnih modela i odstupanja njihovih atomskih satova od GPS vremena.

Korisnički segment čine dvije kategorije korisnika – autorizirani korisnici među kojima spadaju američka vojska i državne službe, a neautorizirani su svi ostali korisnici širom svijeta. Pozicioniranje neautoriziranih korisnika odstupa 10 metara od pozicioniranja autoriziranih korisnika [11].

U prometu prikupljanje podataka može se provesti metodom „podataka plutajućih vozila“ (engl. Floating Car Data, FCD). FCD metoda je prikupljanje podataka o prometu u stvarnom vremenu lociranjem vozila putem GPS-a ili GSM-a. Dakle, svako vozilo opremljeno je mobilnim telefonom i li GPS-om koji djeluju kao senzori u prometnoj mreži. Prikupljanje podataka putem GPS-a mogu dati preciznu lokaciju koja može odstupati 30 metara. FCD metodom prikupljaju se podaci poput lokacije automobila, brzine i smjera vozila koja se anonimno šalju u prometni centar. Nakon prikupljanja i obrade informacija novoobrađeni podaci mogu biti predani svim korisnicima u pojedinom dijelu mreže kao informacija o stanju na prometnicama [5].

3.4. Globalni sustav pokretnih komunikacija

Globalni sustav pokretnih komunikacija (engl. Global System for Mobile Communication, GSM) predstavlja standard koji je u potpunosti digitaliziran. GSM omogućuje upotrebu prijenosa podataka na velikim područjima, koja su prekrivena ćelijskom strukturom. Širina frekvencijskog pojasa iznosi 25 MHz. Za uspostavljanje veze mobilna jedinica – bazna stanica iznosi 890-915 MHz, a za vezu bazna stanica – mobilna jedinica 935-960 MHz.



Slika 3. Prikaz arhitekture GMS mreže

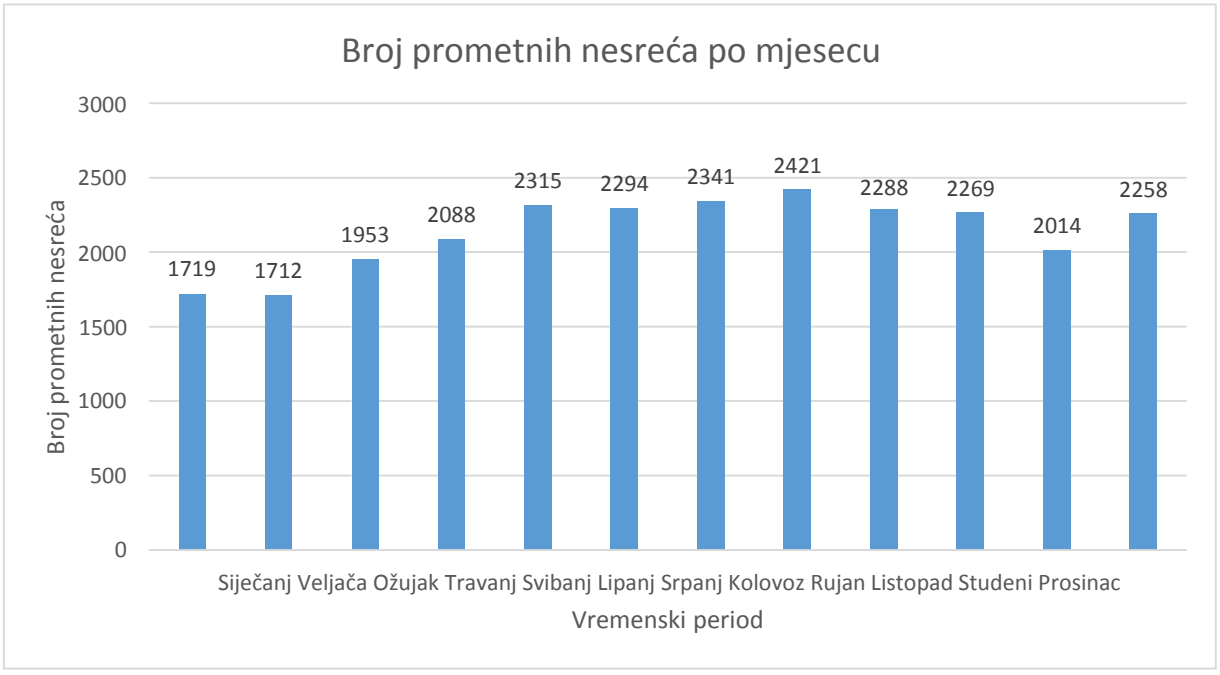
Izvor: https://slideplayer.com/slide/14089294/

GSM mreža sastoji se od tri glavna dijela: mobilna stanica, bazna stanica i mrežni podsistem kao što je prikazano na slici 3. Mobilna stanica sastoji se od SIM (engl. Subscriber Identity Module) kartice koja omogućuje mobilnost. Bazna stanica sastoji se od komunikacijskog i kontrolnog dijela. Komunikacijski dio definira ćelije i barata protokolima za radiovezu s mobilnim stanicama. Kontrolni dio podešava radio kanale, izmjenjuje frekvencije i predstavlja vezu između mobilne stanice i mrežnog podsistema. Središnja komponenta mrežnog podsistema je mobilni komutacijski centar (engl. Mobile Switching Center, MSC) čije su funkcije registracija, provjera autentičnosti, lociranje, omogućavanje veze s digitalnim mrežama integriranih usluga (engl Integrated Services Digital Network, ISDN).

Kao što je već rečeno u prethodnom poglavlju prikupljanje podataka o vozilima omogućeno je putem GPS ili GSM sustava. Većina vozača posjeduje najmanje jedan ili više mobilnih uređaja koji služe kao senzori u prometnoj mreži. Pozicioniranje mobilnog uređaja provodi se tehnikom triangulacije, odnosno određivanjem položaja glavnih točaka. Zatim se vrijeme putovanja i ostali podaci mogu procijeniti u nizu ostalih segmenata prije nego što ih prometni centri pretvore u korisne informacije. Ovaj pristup je posebno dobro prilagođen za pružanje relativno točnih informacija u urbanim područjima zbog manje udaljenosti između antena [5].

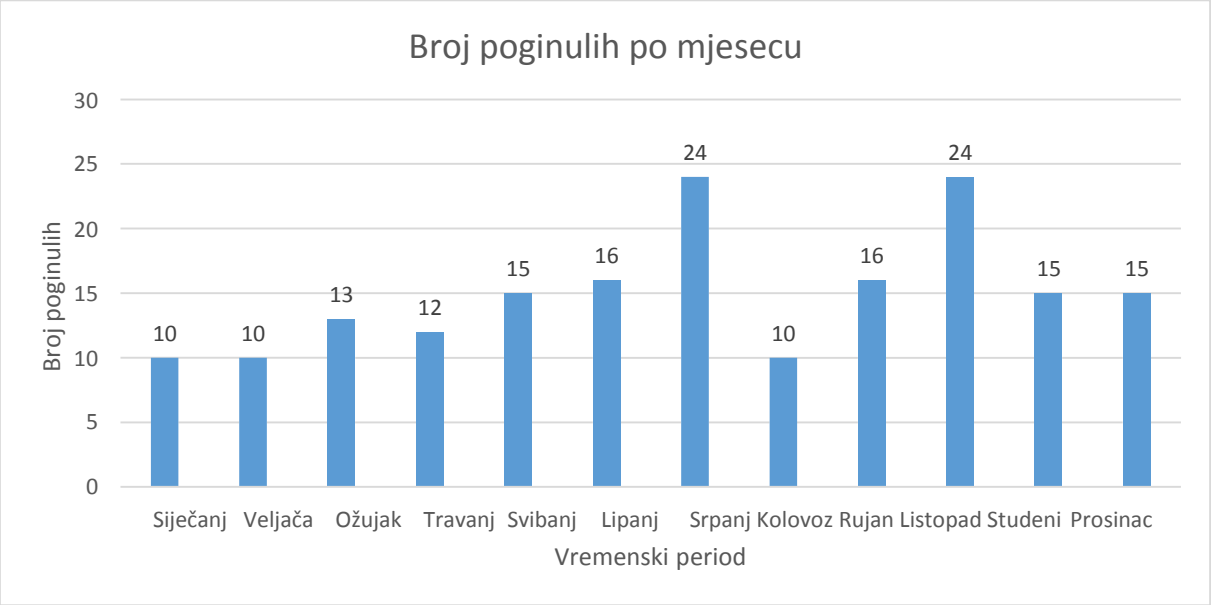
4. POHRANA PODATAKA O KRITIČNIM MJESTIMA U RELACIJSKU BAZU

Policijski službenici su ovlašteni i dužni izaći na mjesto događaja na prometnoj mreži o kojem su dojavljeni. Ako je dojava istinita, odnosno ako je u prometnoj nesreći sudionik zadobio tjelesne ozljede i/ili je poginuo i/ili je nastupila materijalna šteta, policija je dužna napraviti uviđaj te prikupiti sve relevantne činjenice. Svaki policijski službenik popunjava „Upitnik o prometnoj nesreći“ gdje se unose podaci o prometnoj nesreći. Podaci u upitniku mogu biti uneseni u brojčanom, znakovnom i tekstualnom obliku. Svaki brojčani ili znakovni podatak jest šifra koja ima značenje u već izrađenom znakovniku za popunjavanje upitnika o prometnoj nesreći (UPN). Na tom upitniku nalaze se atributi koje opisuju pojedinu prometnu nesreću, kao što su posljedice, vrste, okolnosti prometne nesreće, karakteristike ceste i sl. Za sudionike bilježe se podaci o spolu, količini alkohola, upotreba droge, prijave i dr. Kasnije se ti isti podaci upisuju u informacijski sustav policije ili Excel tablicu. Na slici 4 prikazani su navedeni atributi u Excel tablici.



Grafikon 1. Prikaz broja prometnih nesreća

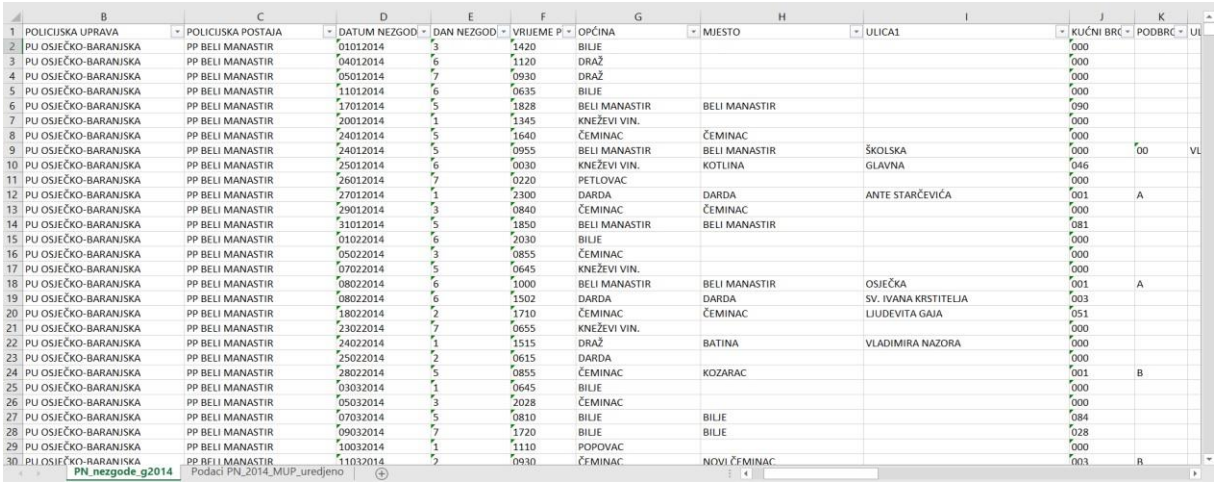
Izvor: Izradio i prilagodio autor



Grafikon 2. Prikaz broja poginulih po mjesecu

Izvor: Izradio i prilagodio autor

Grafikon 1. i 2. prikazuju broj prometnih nesreća i broj poginulih u prometnim nesrećama u vremenskom periodu od mjesec dana. Podaci su obrađeni na temelju dobivenih podataka o prometnim nesrećama 2014. godine. Na grafikonima je vidljivo da broj i vrsta prometnih nesreća variraju po mjesecima. Najveći broj prometnih nesreća u 2014. godini dogodio se u kolovozu, a najmanji u veljači. Također broj prometnih nesreća s poginulim osobama je namanji u siječnju u veljači dok je najveći broj zabilježen u srpnju i listopadu.



	B	C	D	E	F	G	H	I	J	K
1	POLICIJSKA UPRAVA	POLICIJSKA POSTAJA	DATUM NEZGOD	DAN NEZGOD	VRIJEME P	OPĆINA	MJESTO	ULICA1	KUĆNI BR	PODBR
2	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	01012014	3	1420	BILJE			000	
3	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	04012014	6	1120	DRAŽ			000	
4	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	05012014	7	0930	DRAŽ			000	
5	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	11012014	6	0635	BILJE			000	
6	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	17012014	5	1828	BELI MANASTIR	BELI MANASTIR		090	
7	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	20012014	1	1345	KNEŽEVI VIN.			000	
8	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	24012014	5	1640	ČEMINAC	ČEMINAC		000	
9	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	24012014	5	0955	BELI MANASTIR	BELI MANASTIR	ŠKOLSKA	000	00
10	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	25012014	6	0030	KNEŽEVI VIN.	KOTLINA	GLAVNA	046	
11	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	26012014	7	0220	PETLOVAC			000	
12	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	27012014	1	2300	DARDA	DARDA	ANTE STARČEVIĆA	001	A
13	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	29012014	3	0840	ČEMINAC	ČEMINAC		000	
14	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	31012014	5	1850	BELI MANASTIR	BELI MANASTIR		081	
15	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	01022014	6	2030	BILJE			000	
16	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	05022014	3	0855	ČEMINAC			000	
17	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	07022014	5	0645	KNEŽEVI VIN.			000	
18	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	08022014	6	1000	BELI MANASTIR	BELI MANASTIR	OSJEČKA	001	A
19	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	08022014	6	1502	DARDA	DARDA	SV. IVANA KRSTITELJA	003	
20	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	18022014	2	1710	ČEMINAC	ČEMINAC	LJUDEVITA GAJA	051	
21	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	23022014	7	0655	KNEŽEVI VIN.			000	
22	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	24022014	1	1515	DRAŽ	BATINA	VLADIMIRA NAZORA	000	
23	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	25022014	2	0615	DARDA			000	
24	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	28022014	5	0855	ČEMINAC	KOZARAC		001	B
25	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	03032014	1	0645	BILJE			000	
26	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	05032014	3	2028	ČEMINAC			000	
27	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	07032014	5	0810	BILJE	BILJE		084	
28	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	09032014	7	1720	BILJE	BILJE		028	
29	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	10032014	1	1110	POPOVAC			000	
30	PU OSJEČKO-BARANJSKA	PP BELI MANASTIR	11032014	2	0930	ČEMINAC	NOVI ČEMINAC		003	B

PN_nezgode_g2014 | Podaci PN_2014_MUP_uredjeno

Slika 4: Podaci o prometnim nesrećama

Izvor: Dokument preuzet iz MUP-a

4.1. Kreiranje relacijske baze podataka

Baza podataka je skup međusobno povezanih podataka, pohranjenih u vanjskoj memoriji računala. Podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima. Upisivanje, promjena, brisanje i čitanje podataka obavlja se posredstvom posebnog softvera, tzv. Sustava za upravljanje bazom podataka (Database Managament System, DBMS). [6]

Sustav za upravljanje bazom podataka jest poslužitelj (server) baze podataka koji omogućava upravljanje podacima u bazi. Oblikuje fizički izgled baze prema logičkoj strukturi baze podatka. Ono što definira logičku strukturu baze jest model podataka. Model podataka definira strukturu baze pomoću tipa podataka, odnosa između podataka i ograničenja. Postoje različiti modeli podataka kao što su:

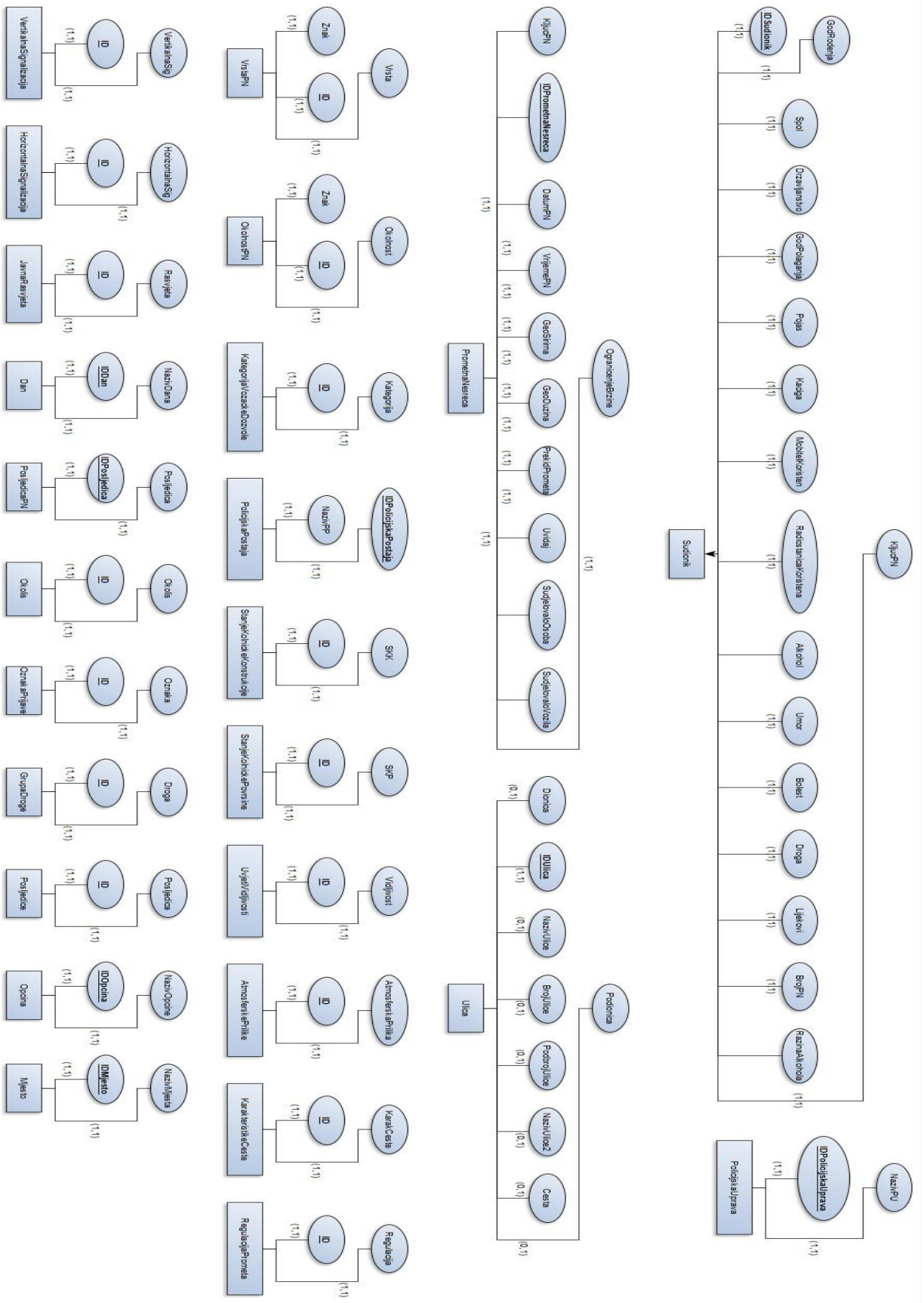
- relacijski model;
- mrežni model;
- hijerarhijski model;
- objektni model;
- dimenzijski model;
- objektno-relacijski model. [6]

U radu je primijenjen relacijski model podataka koji je postavio E. Codd. Podaci su strukturirani kroz relacije. Osnovni koncept modela jesu relacije, entiteti, atributi, ključevi. Podaci i veze među podacima se prikazuju kao redci i stupci koji čine tablice.

U završnom radu koristi se Microsoftov SQL poslužitelj (engl. "Microsoft Search Query Language") za izradu baze podataka i mogućnošću izvođenja funkcija upisivanja, promjena, brisanja i čitanja podataka. Za komunikaciju korisnika s DBMS-a odvija se pomoću jezika za postavljanje upita (engl. "Query Language") koji omogućuje interaktivno pretraživanje baze.

4.1.1. Dijagram entiteta

Prije samog kreiranja baze podataka, najprije je potrebno opisati realni svijet odnosno prometne nesreće putem modela kao što su dijagram entiteta i dijagram relacija (engl. 'Entity-Relationship Diagram', ER dijagram). Dijagram entiteta prikazuje entitete s njihovim atributima. Entiteti predstavljaju objekte ili događaje koji nas zanimaju dok atributi predstavljaju svojstva entiteta [10]. Na slici 5 prikazan je dijagram entiteta koji prikazuje entitete i atribute prometne nesreće.



Slika 5. Dijagram entiteta

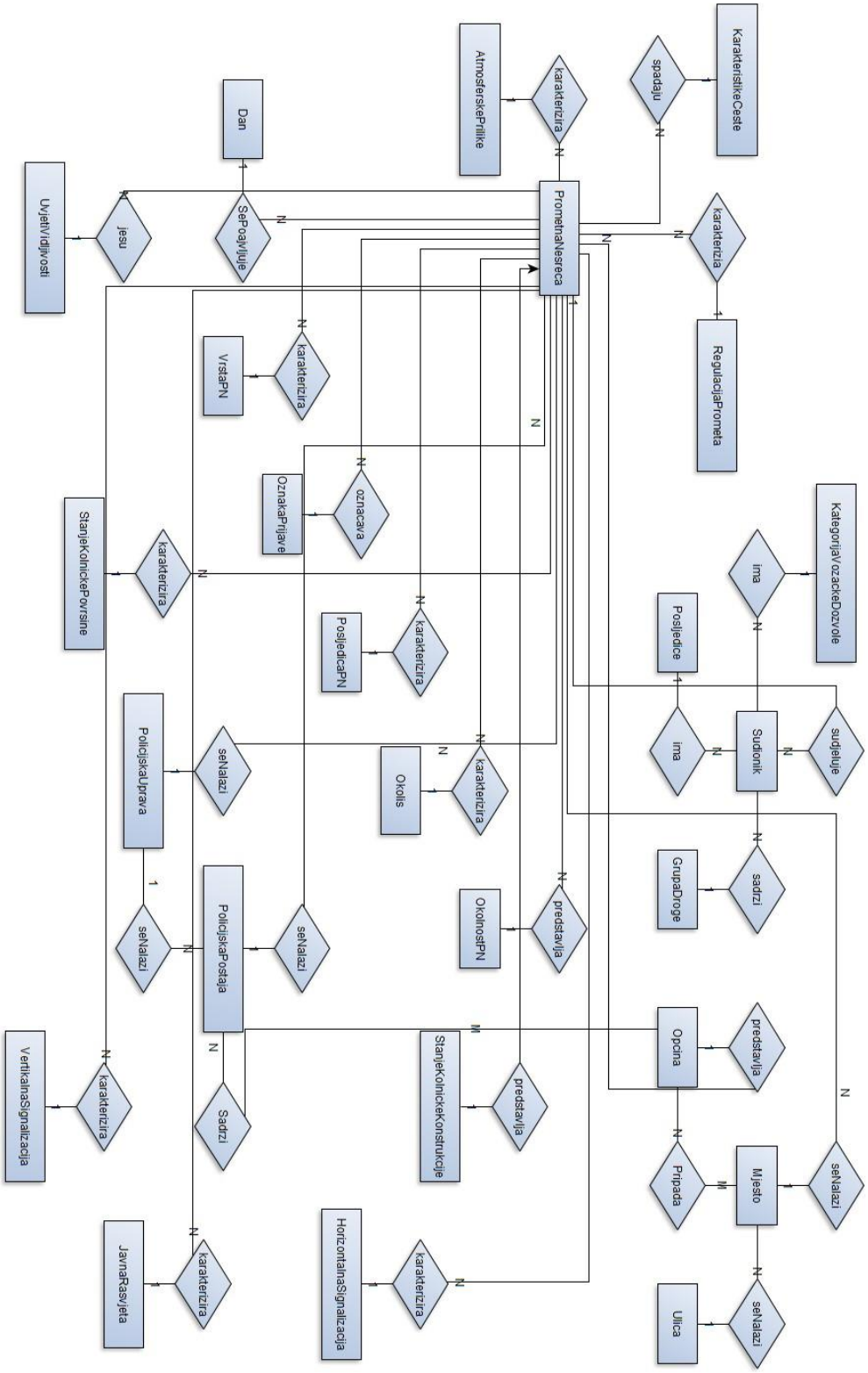
Izvor: Izradio i prilagodio autor

4.1.2. ER dijagram

Nakon izrade dijagrama entiteta, slijedi izrada ER dijagrama koja prikazuje odnose između entiteta koji sudjeluju u relaciji, bez atributa. Postoje različite veze među relacijama kao što su jednostavne veze (1:1, 1:N, M:N) i složene veze. U ovome radu objasnit će se jednostavna 1:N veza pošto je ona najzastupljenija u radu.

U 1:N vezi svaki element skupa prometne nesreće može biti povezan s više elemenata skupa sudionik (ne nužno), dok svaki element skupa sudionik može biti povezan sa samo jednim elementom skupa prometne nesreće [10]. Na slici 6 ER-dijagram prikazuje veze između entiteta prometnih nesreća.

U M:N vezi svaki element skupa općina može biti povezan s više elemenata skupa mjesto, ali i ne mora. Isto tako vrijedi i za elemente skupa mjesto. Dakle, u jednome mjestu može biti više općina, a u jednoj općini može biti isto tako više mjesta.



Slika 6. ER-dijagram

Izvor: Izradio i prilagodio autor

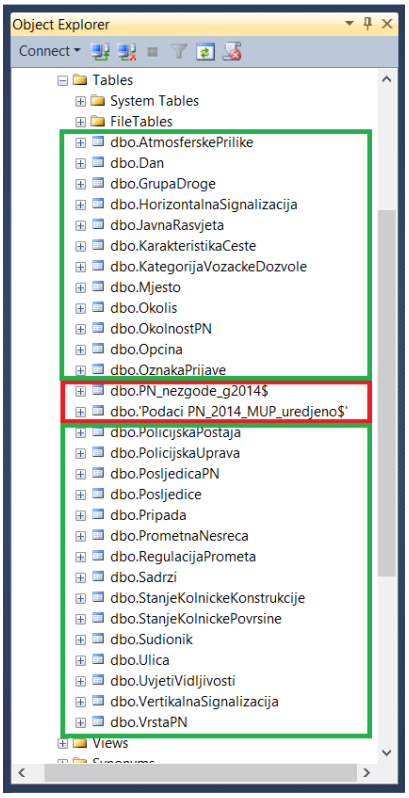
4.1.3. Relacijski model podataka

Kada su izrađeni dijagram entiteta i ER dijagram, potrebno je kreirati relacije, odnosno tablice putem SQL naredbi. Prilikom izrade tablica potrebno je pripaziti na izgled tablice, odnosno svaka relacija (ili tablica) sadrži svoju listu atributa (ili stupaca), a svaki atribut ima svoju domenu (ili tip).

Potrebno je napomenuti da svaka relacija sadrži primarni ključ, koji na jedinstven način identificira svaki redak u tablici. Primarni ključ koji se nalazi kao veza u drugoj tablici, a ne u svojoj originalnoj tablici naziva se strani ključ.

Prilikom kreiranja tablica, potrebno je paziti na redoslijed kreiranja i izvršavanja tablica. Tako da npr. tablicu Sudionik ne može se kreirati i izvršiti dok se najprije ne kreira i izvrši tablica PrometnaNesreća pošto se u tablici Sudionik nalazi strani ključ PrometnaNesrećaID.

Na slici 7. vidljiv je stablasti prikaz baze podataka. U Zelenim okvirima su izrađene tablice, dok su u crvenom okviru dodane tablice iz Excela.



Slika 7. Stablasti prikaz baze podataka

Izvor: Izradio i prilagodio autor

4.2. Pohrana podataka u kreiranu relacijsku bazu podataka

Prije same pohrane podataka u bazu podataka potrebno je najprije uvesti podatke iz Excel tablice u bazu podataka. Nakon toga je potrebno putem upita odabrati željene podatke te ih pohraniti u kreirane tablice. Podatke iz Excel tablice prebacuju se u SQL poslužitelj pomoću funkcije uvoz podataka (engl. Import Data). Tada SQL poslužitelj stvara nove dvije tablice, pošto se Excel tablica sastoji od dva lista kako je prikazano podataka kako je prikazano na slici 4. SQL poslužitelj ne pazi na domenu podataka, već je svim stupcima primijenio tip podatka nvarchar(255).

Kod pohrane podataka iz Excel tablica u kreirane tablice potrebno je paziti na redoslijed izvršavanja pohrane podataka, isto kao i kod kreacije tablica, zbog ograničenja koja dopuštaju strani ključevi. Za prebacivanje podataka iz Excel tablice u kreirane tablice koristi se naredba „insert into“. Naredba koja dohvaća podatke koji se dodaju je „select“ .

U insert dijelu navodi se ime tablice te u zagradi stupac u koji se želi pohraniti podatke. U select dijelu selektiraju se svi različiti podaci iz Excel tablice uz pomoć kodne riječi „distinct“. U „order by“ dijelu želi se abecedno poredati nazivi kodna riječ „asc“.

Prije samog pohranjivanja podataka u tablicu PrometnaNesreca, potrebno je popuniti tablice koje nije moguće popuniti upitom iz razloga što u Excel tablici se nalaze samo brojevi koji imaju značenje u Znakovniku za popunjavanje upitnika o prometnoj nesreći. Primjerice oznaka 2 za vrstu prometne nesreće znači bočni sudar. Na primjeru ispod je prikazano unošenje jednog takvog primjera u bazu podataka.

```
set identity_insert VrstaPN  on
insert into VrstaPN (ID, ZNAK, Vrsta)
values (1, '1', 'IZ SUPROTNIH SMJEROVA'),
(2, '2', 'BOČNI SUDAR'),
(3, '3', 'USPOREDNA VOŽNJA'),
(4, '4', 'VOŽNJA U SLIJEDU'),
(5, '5', 'VOŽNJA UNATRAG'),
(6, '6', 'UDAR U PARKIRANO VOZILO'),
(7, '8', 'SLIJETANJE VOZILA S CESTE'),
(8, '9', 'NALET NA BICIKL'),
(9, '10', 'NALET NA PJEŠAKA'),
(10, '11', 'NALET NA MOTOCIKL ILI MOPED'),
(11, '12', 'SUDAR SA ŽELJEZNIČKIM VOZILOM'),
(12, '14', 'OSTALO'),
(13, '15', 'UDAR VOZILA U OBJEKT NA CESTI'),
(14, '16', 'UDAR VOZILA U OBJEKT KRAJ CESTE'),
(15, '17', 'NALET NA DOMAĆU ŽIVOTINJU'),
(16, '18', 'NALET NA DIVLJU ŽIVOTINJU'),
(17, '19', 'NALET NA PTICU')
set identity_insert VrstaPN off
```

Nakon što su ispunjene tablice, tablica PrometnaNesreca je spremna za pohranu podatka. Spomenuto je da SQL poslužitelj ne pazi na domenu podataka, odnosno svim podacima je pridijeljen isti tip podataka. Upravo zbog toga potrebno je korigirati select naredbu, kako bi se poštivao format određenog tipa podataka. Tako da format u Excel tablici u stupcu DATUM NEZGODE jest 'DD-MM-YY', a naš traženi format jest 'YY-MM-DD'. Da bi se riješilo to formatiranje koristila se sljedeća naredba:

```
(right([DATUM NEZGODE], 4)+'-'+substring([DATUM NEZGODE], 3,2)+'-'+left([DATUM NEZGODE],2))
```

Isto tako, potrebno je korigirati format VRIJEME PN koje je u formatu 'hh:mm', a traženi format je 'hh:mm:ss'. Naredba jest:

```
(left([VRIJEME PN],2)+':'+right([VRIJEME PN],2)+':00'
```

Stupci SUDJELOVALO OSOBA, SUDJELOVALO VOZILA, KLJUČ PN i OGRANIČENJE BRZINE potrebno je pretvoriti podatke iz znakovnog tipa u brojevni tip podatka sljedećom naredbom:

```
convert(int, [SUDJELOVALO VOZILA]),
convert(int, [SUDJELOVALO OSOBA]),
convert(int, [KLJUČ PN]),
convert(int, [OGRANIČENJE BRZINE]),
```

Nadalje, stupci u Excel tablici DROGA, LIJEKOVI i ALKOHOL imaju tri različite vrijednosti, a to su D,N ili O odnosno da, ne ili odbijeno. Naredbom „case when“ podaci su preimenovani u DA, NE ili ODBIJENO, ovisno o vrijednosti. Naredba je sljedeća:

```
case when DROGA = 'D' then 'DA'
       when DROGA = 'N' then 'NE'
       when DROGA = 'O' then 'ODBIJENO'
end as Droga,
case when LIJEKOVI = 'D' then 'DA'
       when LIJEKOVI = 'N' then 'NE'
       when LIJEKOVI = 'O' then 'ODBIJENO'
end as Lijekovi,
case when ALKOHOL = 'D' then 'DA'
       when ALKOHOL = 'N' then 'NE'
       when ALKOHOL = 'O' then 'ODBIJENO'
end as Alkohol
```

5. PRIKAZ PROSTORNE DISTRIBUCIJE PODATAKA I ODREĐIVANJE KRITIČNIH MJESTA

Nakon pohrane podataka u bazu podataka, slijedeći korak jest vizualizirati obrađene podatke putem grafičkog korisničkog sučelja pomoću kojeg korisnik može manipulirati podacima putem računala. Grafičko sučelje izrađeno je pomoću programa Visual Studio programskog alata koji nudi mnoge mogućnosti.

5.1. Microsoft Visual Studio

Microsoft Visual Studio predstavlja alat za razvoj različitih aplikacija kao što su Windows, Windows Form, Windows Presentation Foundation, Windows Store i Microsoft Silverlight. Program nudi različite programske jezike poput C++, C#, Java, F#, JavaScript, Python, Query Language, TypeScript, Visual Basic.

Program za prikaz prostorne razdiobe nesreća i kritičnih mjesta je napisan u C# programskom jeziku. C# je nastao s ciljem da bude jednostavan, siguran, moderan, objektno orijentiran jezik visokih performansi za .NET platformu. Nastao je na temelju objektnih jezika Java, C++ i Visual Basic. Vrlo je sličan Javi i C++ jeziku (sintaksa i semantika je dobrim dijelom preuzeta iz Jave, koja je kao i C# potpuno objektno orijentirani jezik), ali C# za razliku od Jave nije neovisan o platformi, tj. operativnom sustavu, već je kreiran za izradu stolnih (desktop) i Internet aplikacija u .Microsoft .NET okruženju [5].

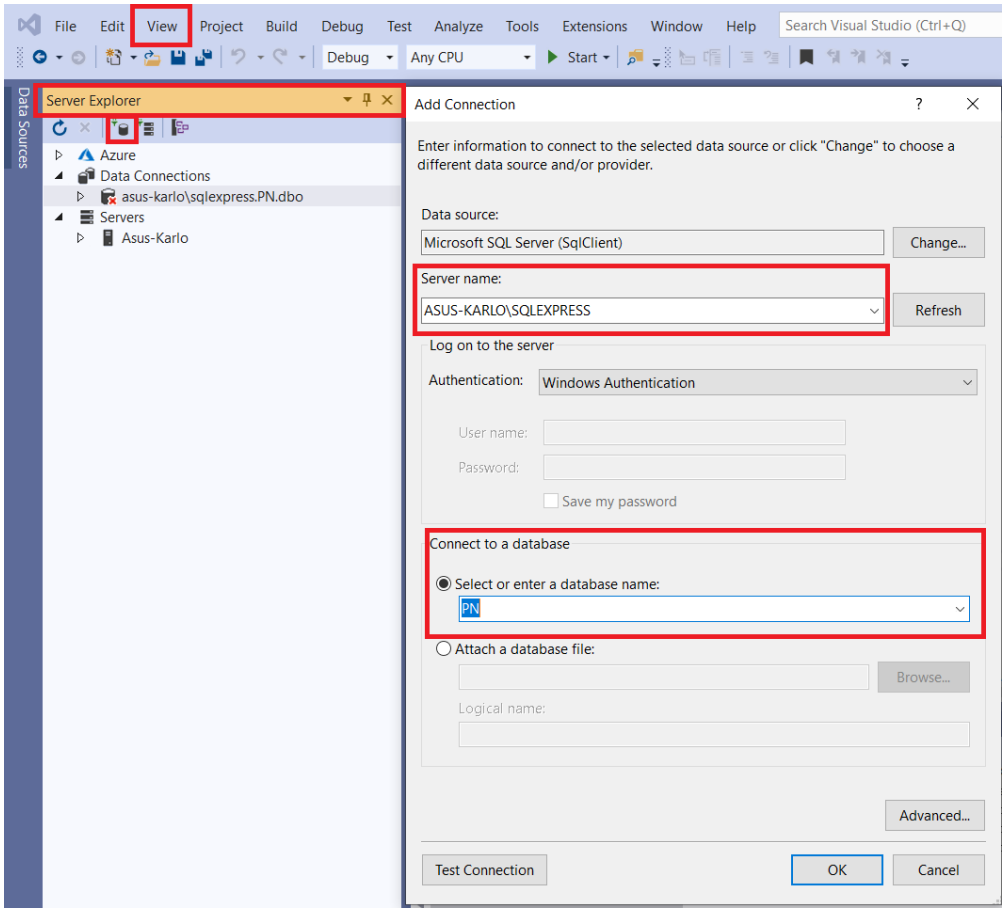
U ovome radu aplikacija je izrađena u Windows Form projektu. Windows Form aplikacije su aplikacije za izradu grafičkog sučelja. Forma predstavlja strukturirani prozor koji sadrži prezentacijske elemente za prikaz i unos podataka i one su najučinkovitiji i najjednostavniji način komunikacije korisnika sa programom. Dvije osnovne vrste formi su:

- SDI forme (engl."Single Document Interface") – forme za sučelje jednog dokumenta. Za prikaz svakog dokumenta koristi se posebna forma, a svaka forma promatra se kao odvojeni prozor;
- MDI forme (engl."Multiple Document Interface") – forme za sučelje više dokumenata. Postoji jedan glavni prozor (forma), dok se svi ostali prozori (forme) otvaraju preko njega. Glavna forma naziva se MDI roditelj, a ostale povezane forme nazivaju se MDI potomci [10].

Windows Form pruža jednostavno oblikovanje formi pomoću povuci-ispusti (engl. "drag-drop") sučelja.

5.2.Povezivanje baze podataka s aplikacijom

Da bismo mogli prikazati podatke u aplikaciji potrebno je najprije povezati SQL Server sa Visual Studiom. Unutar samog projekta potrebno je kliknuti na *View ->Server Explorer -> Connect to Database*. Zatim program izbacuje prozor unutar kojeg biramo poslužitelj, te bazu podataka na koju se želimo spojiti kao što je prikazano na slici 8. Spajanje na bazu podataka zahtjeva spojni niz (engl. Connection string), koji sadrži podatke na koji server i bazu podataka se želi spojiti i s kojim će se korisničkim podacima prijaviti.



Slika 8: Prikaz povezivanja baze podataka s aplikacijom

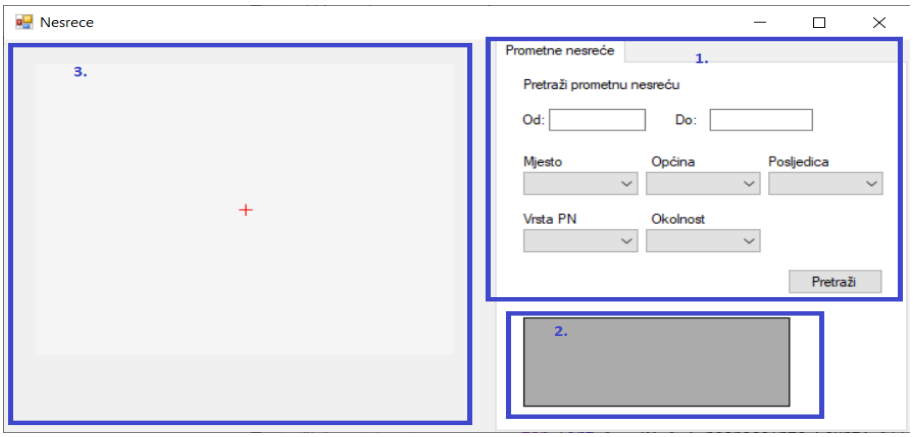
Izvor: Izradio i prilagodio autor

5.2.1. NuGet paketi

Za prikaz geoprostornih podataka koriste se NuGet paketi. Oni su Microsoftov mehanizam za dijeljenje koda. Jednostavno rečeno, NuGet paket je jedna ZIP datoteka s .nupkg ekstenzijom koja sadrži sastavljeni kod (engl. Dynamic Link Library), ostale datoteke povezane s tim kodom i opisni manifest koji uključuje informacije poput broja verzije paketa. Programeri s kodom za dijeljenje stvaraju pakete i objavljuju ih javnom ili privatnom domaćinu. Potrošači paketa nabavljaju te pakete od odgovarajućih domaćina, dodaju ih u svoje projekte, a zatim pozivaju funkcionalnost paketa u šifri projekta. NuGet sam zatim obrađuje sve međufazne detalje [12].

5.2.2. Programiranje sučelja i filtracija podataka

Nakon povezivanja baze sa aplikacijom i instalacijom NuGet paketa, aplikacija je spremna za izradu. Kao što smo naveli ranije, Visual Studio u *Windows Form* aplikacijama nudi opciju povuci-ispusti (engl. Drag-Drop) alate za programiranje. Najprije potrebno je odabrati kontrole koje će aplikacija sadržavati te ih kasnije programirati.



Slika 9: Izgled aplikacije

Izvor: Izradio i prilagodio autor

Na slici 9 jest izgled same aplikacije. Pod brojem 1. nalazi se dio pomoću kojeg je moguće dohvaćati podatke prema zadanim parametrima. Podaci se mogu dohvaćati prema vremenskom periodu, mjestu, općini, posljedici nesreće, vrsti nesreće i okolnosti nesreće. Pod brojem 2. nakon klika na pretraži, filtracijom podataka u pravokutnik ispisuju se ključevi ili šifre prometnih nesreća pod kojim se one pronalaze. Pod brojem 3. nalazi se karta na kojoj će se

pozicionirati prometna nesreća nakon dohvaćanja podataka. Klikom na ključ prometne nesreće na karti se prikazuje odabrana prometna nesreća. Ako se na karti prikaže veći broj prometnih nesreća, onda se mjesto, na kojem se primjećuje veća gustoća točaka, može nazvati kritičnim mjestom.

U dijelu filtriranja podataka nalaze se *TextBox-evi*, *ComboBox-evi* i *Button* kontrole. Za filtraciju podataka o vremenskom periodu koristi se dva *TextBox-a*, unutar kojih se određuje početni datum i završni datum pretrage o prometnim nesrećama. Prilikom unosa mora se paziti na format datuma. Kao što već znamo podaci uneseni u bazu podataka jest u formatu 'YY-MM-DD' tako da se taj format mora pratiti.

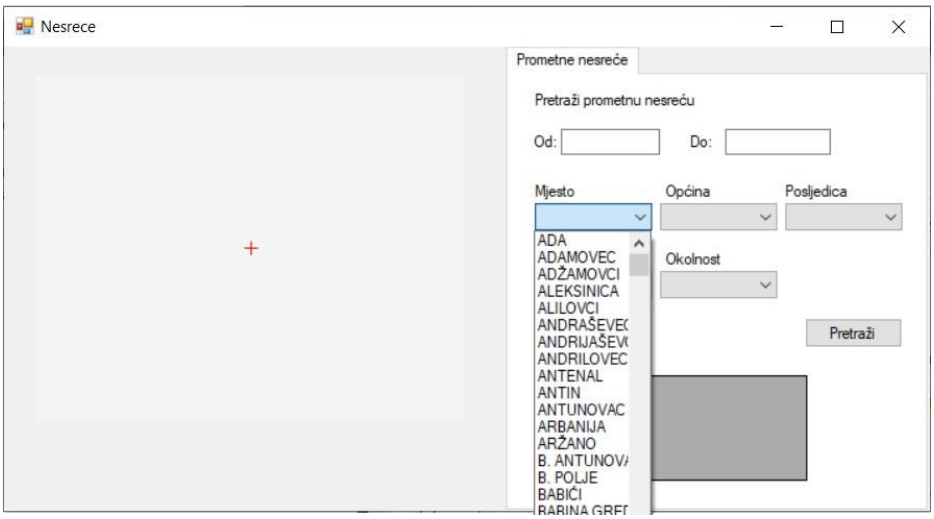
Ispod filtracije vremenskog perioda nalazi se pet *ComboBox-a*. Za svaki *Combobox* potrebno je unijeti odgovarajuće podatke.

```
48
                1 reference
49              private void FillInComboboxes()
50              {
51                  string connetionString;
52                  SqlConnection cnn;
53                  connetionString = @"Data Source=ASUS-KARLO\SQLEXPRESS;Initial Catalog=PN;Integrated Security=True";
54                  cnn = new SqlConnection(connetionString);
55                  cnn.Open();
56
57
58
59                  String sql = "SELECT NazivMjesta FROM mjesto";
60                  SqlCommand cmnd = new SqlCommand(sql, cnn);
61                  SqlDataReader reader = cmnd.ExecuteReader();
62
63                  while (reader.Read())
64                  {
65                      comboBox1.Items.Add(String.Format("{0}", reader[0]));
66                  }
67                  reader.Close();
68
69
```

Slika 10: Filtracija podataka – 1. dio

Izvor: Izradio i prilagodio autor

Na slici 10 je vidljivo najprije (redak 51 - redak 55) povezivanje tablice sa bazom podataka putem *connection stringa*. Zatim se inicijalizira (redak 59 – redak 61) znakovni niz *sql*, te instance klase *cmnd* i *reader*. *Sql* je znakovni niz tip podatka koji se predaje drugoj klasi *SqlCommand cmnd* koja prima dva argumenta, naredbu u obliku znakovnog niza „SELECT NazivMjesta FROM mjesto“ koju želi izvršiti unutar funkcije *SqlConnection* koji otvara konekciju. Klasa *SqlDataReader* je klasa koja izvršava čitanje komande *cmnd*. U *while* petlji (redak 63 – redak 67), dok čitač čita *string* naredbu u *ComboBoxu* se predaju podaci u tipu podatka *string*.



Slika 11: Izgled aplikacije – 2. dio

Izvor: Izradio i prilagodio autor

Na slici 11 je prikazan ispunjeni *ComboBox* nakon izvršavanja koda. Isti princip vrijedi za ispunjavanje preostala četiri *ComboBoxa*.

Sljedeći korak jest programiranje kontrole *Button* ili gumba koji će primiti filtrirane podatke, te dobiti listu ključeva unutar kontrole *DataGridView*.

```
391
392         sql = File.ReadAllText(@"SQLUpitBas3.sql", Encoding.Defa
393
394         String mjesto = comboBox1.Text;
395         String opcina = comboBox2.Text;
396         String posljedica = comboBox3.Text;
397         String vrstaPN = comboBox4.Text;
398         String okolnost = comboBox5.Text;
399
400         String datumStart = textBox2.Text;
401         String datumEnd = textBox3.Text;
402
403         String wherePart = " where ";
404
405         int cnt = 0;
406
407         if (mjesto != "")
408         {
409             cnt++;
410             wherePart += "NazivMjesta = " + "'" + mjesto + "'";
411         }
412
413         if (opcina != "")
414         {
415             cnt++;
416             if (cnt > 1)
417             {
418                 wherePart += " AND ";
419             }
420             wherePart += "NazivOpcine = " + "'" + opcina + "'";
421         }
```

```
422
423         if (posljedica != "")
424         {
425             cnt++;
426             if (cnt > 1)
427             {
428                 wherePart += " AND ";
429             }
430             wherePart += "PosljedicaPN.Posljedica = " + "'" + posljedica + "'";
431         }
432
433         if (vrstaPN != "")
434         {
435             cnt++;
436             if (cnt > 1)
437             {
438                 wherePart += " AND ";
439             }
440             wherePart += "Vrsta = " + "'" + vrstaPN + "'";
441         }
442
443         if (okolnost != "")
444         {
445             cnt++;
446             if (cnt > 1)
447             {
448                 wherePart += " AND ";
449             }
450             wherePart += "Okolnost = " + "'" + okolnost + "'";
451         }
452
```

Slika 12: Filtracija podataka – 2. dio

Izvor: Izradio i prilagodio autor

Na slici 12 prikazan je isječak iz koda gdje se najprije filtrirani podaci predaju u novonastale varijable (redak 394 – redak 403) mjesto, opcina, posljedica , vrstaPN, okolnost, datumStart i

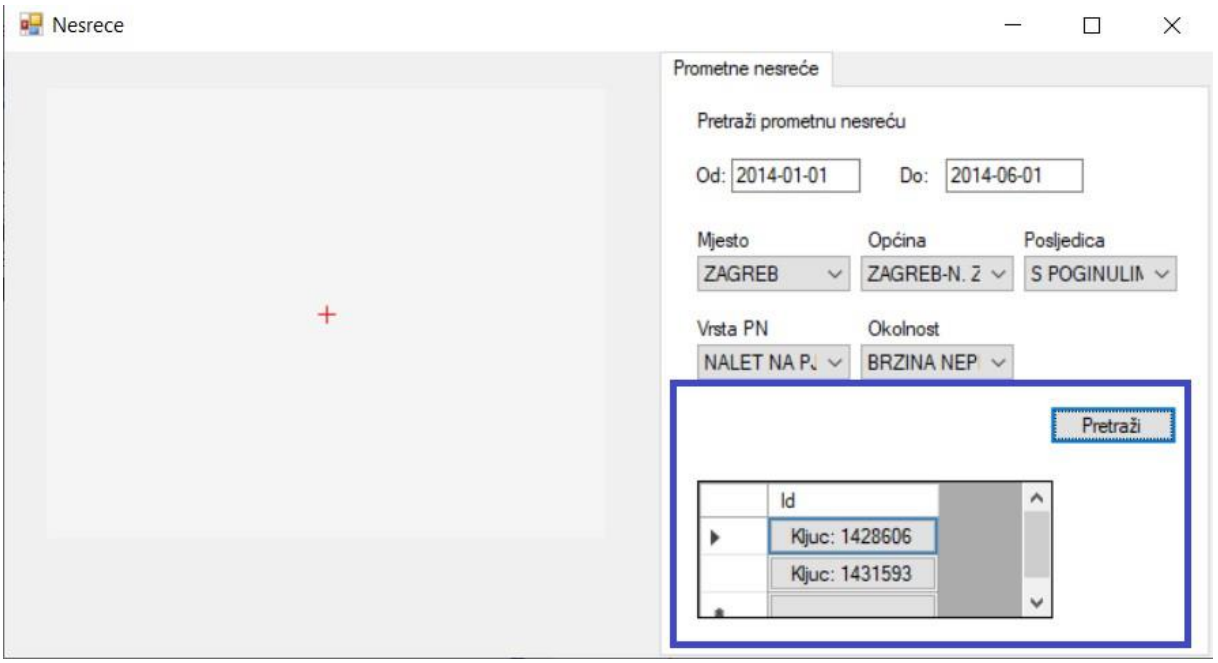
datumEnd (slika 14). Inicijalizira se brojač „int cnt = 0;“ koji prolazi kroz grananje („if“) (redak 405 – redak 452). U grananju se predaju odabrane vrijednosti te na kraju dohvaćaju vrijednosti izgledom na *Sql* upit *string* tipa podatka. Upit se opet predaje pozivanjem kontrole *cmnd* i otvaranjem konekcije koja izvršava upit.

```
472
473             int row = 0;
474             while (reader.Read())
475             {
476                 String kljucPN = String.Format("{0}", reader["KljucPN"]);
477                 if (!opisnikOsoba.ContainsKey(kljucPN))
478                 {
479                     String opisNesreceStr = "";
480                     for (int i=0; i < nesrecaInfo.Count; i++)
481                     {
482                         String data = String.Format("{0}", reader[nesrecaInfo[i]]).Trim();
483                         if (data == "")
484                         {
485                             data = "NULL";
486                         }
487
488                         opisNesreceStr += data;
489                         if (i < nesrecaInfo.Count - 1)
490                         {
491                             opisNesreceStr += ", ";
492                         }
493                     }
494                     opinikNesrece.Add(opisNesreceStr);
495                     keys.Add(kljucPN);
496                     sirine.Add(String.Format("{0}", reader["GeoSirina"]));
497                     duzine.Add(String.Format("{0}", reader["GeoDuzina"]));
498
499                     String text = "Kljuc: " + kljucPN + "\n";
500
501                     Button btn = new Button();
502                     btn.Text = text;
503
504                     dataGridView1.Rows.Add(btn);
505                     dataGridView1.Rows[row].Cells[0].Value = text;
506
507                     row++;
```

Slika 13: Filtracija podataka – 3. dio

Izvor: Izradio i prilagodio autor

Izvršavanjem upita, odnosno klikom na kontrolu *Button*, dobivaju se ključevi prometnih nesreća unutar *DataGridView-a*, koji se trebaju popuniti u liste kao što prikazuje isječak koda na slici 13. Dok čitač izvršava čitanje, liste dobivaju vrijednosti u obliku znakovnog niza koje se kasnije ispunjavaju u liste.



Slika 14: Izgled aplikacije - 3. dio

Izvor: Izradio i prilagodio autor

Završno je još ostalo samo prikaz kritičnih mjesta na digitalnoj karti. Dakle, sljedeće što se treba još izvršiti jest prilikom klika unutar *DataGridView-a* prikazati te ispisati podatke o kritičnom mjestu, prikazano na slici 14.

Najprije je potrebno omogućiti pritisak mišem na stupce unutar *DataGridView-a* . U tu svrhu koristi se sljedeća naredba:

```
DataGridViewButtonColumn btnColumn = new DataGridViewButtonColumn();
            btnColumn.HeaderText = "Id";
            dataGridView1.Columns.Add(btnColumn);
```

```
221         if (numberClicked == 0)
222         {
223             dataGridView1.CellClick += (sender3, e3) =>
224             {
225                 try
226                 {
227                     var senderGrid = (DataGridView)sender3;
228
229                     if (senderGrid.Columns[e3.ColumnIndex] is DataGridViewButtonColumn &&
230                         e3.RowIndex >= 0)
231                     {
232                         gMapControl1.DragButton = MouseButtons.Left;
233                         gMapControl1.MapProvider = GMapProviders.GoogleMap;
234                         double sirina = Convert.ToDouble(sirine[e3.RowIndex]) / 100000.0;
235                         var sirinaCorrected = Math.Floor(sirina);
236                         var sirinaCorrectedDecimals = sirina - sirinaCorrected;
237                         var sirinaMinute = Math.Floor((sirina - sirinaCorrected) * 100);
238                         var sirinaSecond = ((sirina - sirinaCorrected) * 100) - sirinaMinute;
239                         var sirinaFinal = sirinaCorrected + (sirinaMinute / 0.6) / 100 + (sirinaSecond / 0.6) / 10000;
240                         double duzina = Convert.ToDouble(duzine[e3.RowIndex]) / 100000.0;
241                         var duzinaCorrected = Math.Floor(duzina);
242                         var duzinaCorrectedDecimals = duzina - duzinaCorrected;
243                         var duzinaMinute = Math.Floor((duzina - duzinaCorrected) * 100);
244                         var duzinaSecond = ((duzina - duzinaCorrected) * 100) - duzinaMinute;
245                         var duzinaFinal = duzinaCorrected + (duzinaMinute / 0.6) / 100 + (duzinaSecond / 0.6) / 10000;
246
247                         gMapControl1.Position = new GMap.NET.PointLatLng(sirinaFinal, duzinaFinal);
248                         gMapControl1.MinZoom = 5;
249                         gMapControl1.MaxZoom = 100;
250                         gMapControl1.Zoom = 10;
```

Slika 15: Filtracija podataka – 4. dio

Izvor: Izradio i prilagodio autor

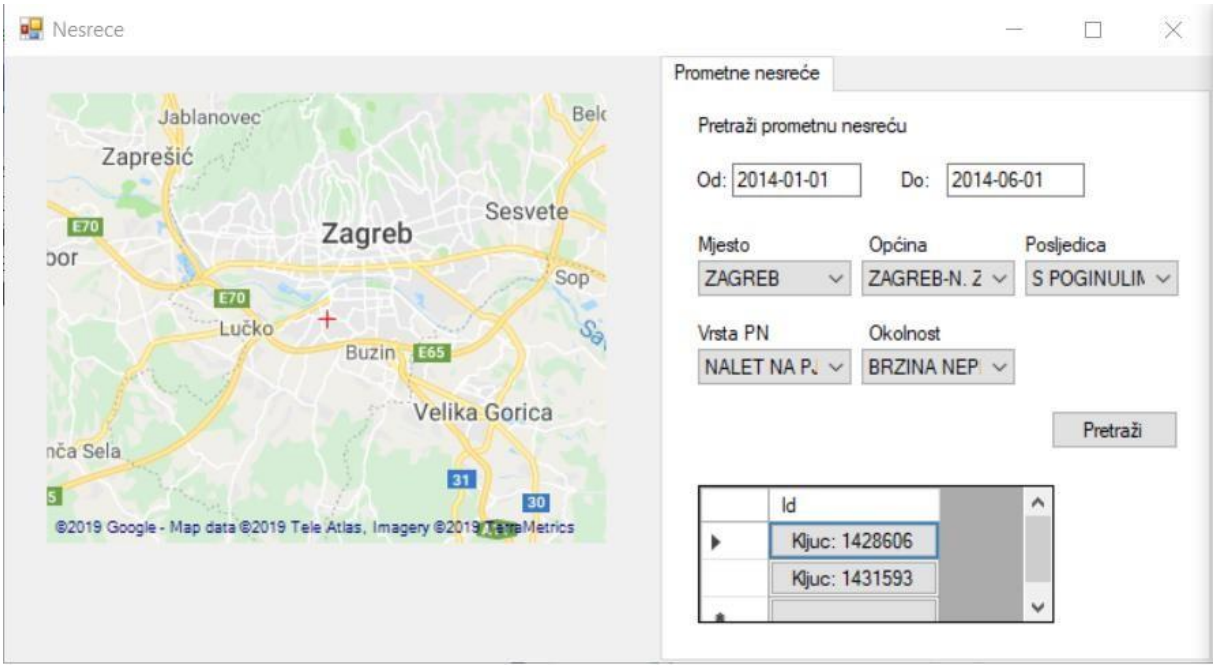
Prilikom klika na kontrolu unutar *DataGridView-a*, odnosno retka iz listi širina i dužina predaju se geografske pozicije o mjestu koje se pretvaraju u decimalni tip podatka (slika 15 prikazuje isječak iz koda) (redak 234 i redak 240). On prima vrijednosti koje se najprije moraju korigirati kako bi precizno odredili lokaciju kritičnog mjesta (redak 234 – redak 239 i redak 241 – redak 245). Zatim se te vrijednosti dodjeljuju digitalnoj karti koja prikazuje kritično mjesto (redak 247). Karta ima mogućnost pomicanja držanjem lijeve kontrolne tipke miša (redak 232), te mogućnost zumiranja (redak 249).

```
256
257             for (int i = 1; i < nesrecaInfo.Count; i++)
258             {
259                 if (i < nesrecaInfo.Count - 1)
260                 {
261                     opisNesreceMessage += String.Format("{0}: {1}\n", nesrecaInfo[i], opisList[i]);
262                 }
263             }
264             MessageBox.Show(opisNesreceMessage);
265
266             String opisSudionikaStr = "";
267             for (int i = 0; i < osobeInfo.Count; i++)
268             {
269                 opisSudionikaStr += osobeInfo[i];
270                 if (i < osobeInfo.Count - 1)
271                 {
272                     opisSudionikaStr += ", ";
273                 }
274             }
275             List<String> sudionici = opisnikOsoba[keys[e3.RowIndex]];
276             string sudioniciMessage = "Popis sudionika:";
277
278             for (int i = 0; i < sudionici.Count; i++)
279             {
280                 sudioniciMessage += "\n";
281                 var sudionikList = sudionici[i].Split(',').ToList();
282                 for (int j = 0; j < osobeInfo.Count; j++)
283                 {
284
285                     if (j < osobeInfo.Count - 1)
286                     {
287                         sudioniciMessage = sudioniciMessage + String.Format("{0}: {1}\n", osobeInfo[j], sudionikList[j]);
288                     }
289                 }
290             }
291             MessageBox.Show(sudioniciMessage);
292         }
```

Slika 16: Filtracija podataka – 5. dio

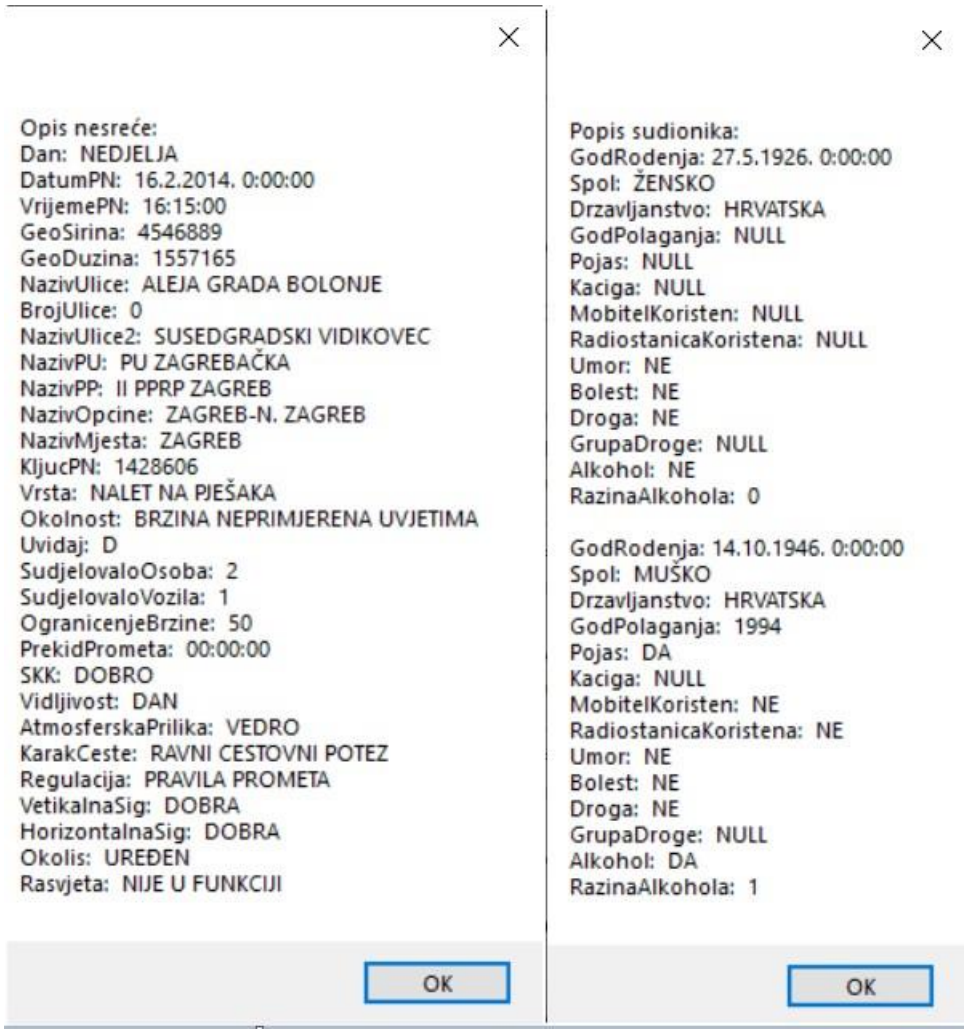
Izvor: Izradio i prilagodio autor

Uz prikaz digitalne karte pojavljuju se dva *MessageBox-a*, jedan za drugim u kojima se ispisuje najprije atributi o prometnoj nesreći, a zatim atributi o sudionicima u nesreći prikazano slikama 17 i 18.



Slika 17: Izgled aplikacije – 4. dio

Izvor: Izradio i prilagodio autor



Slika 18: Izgled aplikacije - 5. dio

Izvor: Izradio i prilagodio autor

6. ZAKLJUČAK

Prometni sustav predstavlja nepredvidiv i dinamičan sustav gdje su negativne pojave prometne nesreće koje mogu rezultirati materijalnim štetama i/ili ljudskim stradanjima. U ovome završnom radu predstavljena je problematika slučajnih pojava u prometu gdje su objašnjeni termini incident, prometna nesreća te kritično mjesto. Sustav kao takav je mjerljiv, odnosno moguće je prikupiti podatke koje omogućavaju analizu pojava prometne nesreće. Navedene su metode i kriteriji identificiranja kritičnih mjesta, te tehnologije koje omogućavaju prikupljanje podataka o takvim sustavima.

Postupanje u svezi prometnih nesreća izvršavaju nadležne institucije, odnosno policija. U ovome radu obrađeni su podaci o prometnim nesrećama na području Republike Hrvatske za razdoblje jedne godine dobiveni od strane MUP-a. Razvojni proces izrade baze podataka opisano je kroz tri faze oblikovanja. Prva faza predstavlja konceptualni dio gdje su prikazani entiteti, atributi i veze koje opisuje sadržaj baze i načine povezivanja. U drugoj fazi prikazane su relacije, odnosno logička shema koja opisuje njeno logičko ponašanje. U posljednjoj fizičkoj fazi opisana je fizička građa baze. Ona predstavlja naredbe pomoću kojih su se prebacivali podaci iz jednih u druge tablice.

Završna faza izrade završnog rada jest vizualni prikaz obrađenih podataka. Vizualizacija je izvršena pomoću grafičkog korisničkog sučelja gdje je omogućena pretraga i filtracija podataka. Svrha završnog rada jest interakcijskim sučeljem prikazati kritična mjesta putem digitalne karte na području Republike Hrvatske. Kritična mjesta obuhvaćaju podatke o svojim lokacijama, koji su pogodna za izradu statističkih obrada podataka. Izrađeno grafičko sučelje omogućava jednostavan, brz i pregledan način manipulacije velikih količina podataka koje mogu služiti za izradu statističkih ili analitičkih podataka.

LITERATURA

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

[2] Zhixiao, X., Jun, J.: Kernel Density Estimation of Traffic Accidents in a Network Space. Computers, Environment and Urban Systems 32 (2008) 396–406

[3] Zovak, G., Šarić, Ž.: Prometno-tehničke ekspertize i sigurnost – nastavni materijal, Fakultet prometnih znanosti, Sveučilište u Zagrebu, Zagreb, 2015.

[4] Gatrell, A., Bailey, T., Diggle, B., Rowlingson, B.: Spatial Point Pattern Analysis and Its Application in Geographical Epidemiology, Transactions of the Institute of British Geographers, Vol. 21, No. 1 (1996), pp. 256-274

[5] Guillaume, L.: Road Traffic Data: Collection Methods and Applications, Working Papers on Energy, Transport and Climate Change N.1

[6] Manger, R.: Baze Podataka, Udžbenik Sveučilišta u Zagrebu, Zagreb, 2014.

[7] Zakon o sigurnosti prometa na cestama, čl. 182., st. 1., NN 67/08., 48/10., 80/13., 158/13., 92/14., 64/15., 108/17., Zagreb: Narodne novine

[8] Šarić, Ž., Zovak, G., Kunštek, A. Kučinić, T: Metodologija za identifikaciju opasnih mjesta u cestovnoj prometnoj mreži, Fakultet prometnih znanosti, Zagreb, 2017.

[9] Autorizirana predavanja iz kolegija Inteligentni transportni sustavi I: https://moodle.srce.hr/2017-2018/pluginfile.php/1205351/mod_resource/content/2/PrezentacijePREDAVANJA/predavanj e%205%20201617.pdf

[10] Autorizirana predavanja iz kolegija Baze podataka : https://moodle.srce.hr/2018-2019/pluginfile.php/1759966/mod_resource/content/6/2.tjedan.pdf

[11] Autorizirana predavanja iz kolegija Lokacijsko navigacijskih sustava: https://moodle.srce.hr/2017-2018/pluginfile.php/1670215/mod_resource/content/1/07-Radionavigacijski%20sustavi.pdf

[12] Internet izvor :https://docs.microsoft.com/en-us/nuget/what-is-nuget (pristupljeno 31. kolovoza 2019.)

POPIS SLIKA

Slika 1: Shematski prikaz induktivne petlje 5
Slika 2: Prikaz lociranja pozicije putem GPS-a..... 6
Slika 3: Prikaz arhitekture GMS sustava 8
Slika 4: Podaci o prometnim nesrećama..... 10
Slika 5: Dijagram entiteta 12
Slika 6: ER-Dijagram 14
Slika 7: Stablasti prikaz na bazu podataka 15
Slika 8: Prikaz povezivanja baze podataka s aplikacijom 19
Slika 9: Izgled aplikacije – 1. dio 20
Slika 10: Filtracija podataka – 1. dio 21
Slika 11: Izgled aplikacije – 2. dio 22
Slika 12: Filtracija podataka – 2. dio 22
Slika 13: Filtracija podataka – 3. dio 23
Slika 14: Izgled aplikacije – 3. dio 24
Slika 15: Filtracija podataka – 4. dio 25
Slika 16: Filtracija podataka – 5. dio 26
Slika 17: Izgled aplikacije – 4. dio 26
Slika 18: Izgled aplikacije – 5. dio 27

POPIS GRAFIKONA

Grafikon 1. Prikaz broja prometnih nesreća..... 9

Grafikon 2. Prikaz broj poginulih po mjesecu 10

Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj završni rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.
Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.
Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.
Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu završnog rada pod naslovom **PROSTORNO VREMENSKA ANALIZA PODATAKA O PROMETNIM NESREĆAMA** na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, 9.9.2019

(potpis)