

# Aplikacija za automatizaciju procesa simulacije prometnih podataka

---

**Garašić, Andre**

**Undergraduate thesis / Završni rad**

**2021**

*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:119:979001>

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

*Download date / Datum preuzimanja:* **2024-08-09**



*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET PROMETNIH ZNANOSTI

Andre Garašić

**APLIKACIJA ZA AUTOMATIZACIJU PROCESA  
SIMULACIJE PROMETNIH PODATAKA**

ZAVRŠNI RAD

Zagreb, 2021.

Zagreb, 25. svibnja 2021.

Zavod: **Zavod za inteligentne transportne sustave**  
Predmet: **Algoritmi i programiranje**

## ZAVRŠNI ZADATAK br. 6303

Pristupnik: **Andre Garašić (0135251021)**  
Studij: **Inteligentni transportni sustavi i logistika**  
Smjer: **Inteligentni transportni sustavi**

Zadatak: **Aplikacija za automatizaciju procesa simulacije prometnih podataka**

### Opis zadatka:

Prilikom korištenje simulacijskog alata Simulation of Urban MObility (SUMO) često je potrebno podesiti značajan broj parametara prometnih scenarija. Proces podešavanja parametara je moguće automatizirati. Potrebno je izraditi program u programskog jeziku Python koji će automatizirati proces određivanja Traffic Assignment Zone (TAZ) pomoću novoizrađenog sučelja koje koristi slobodno dostupnu digitalnu kartu Open Street Map. U radu je potrebno opisati korišteni sustav Django, JavaScript Framework Leaflet, makro i mikro simulacijske alate i detaljno objasniti proces automatizacije koji je izrađen za simulacijski alat SUMO.

Mentor:

Predsjednik povjerenstva za  
završni ispit:

---

prof. dr. sc. Tonči Carić

Sveučilište u Zagrebu  
Fakultet prometnih znanosti

## **ZAVRŠNI RAD**

# **APLIKACIJA ZA AUTOMATIZACIJU PROCESA SIMULACIJE PROMETNIH PODATAKA**

## **APPLICATION FOR AUTOMATION OF THE SIMULATION PROCESS OF TRAFFIC DATA**

Mentor: prof. dr. sc. Tonči Carić

Student: Andre Garašić

Neposredni voditelj: Željko Majstorović, mag.  
ing. traff. JMBAG: 0135251021

Zagreb, rujan 2021.

## SAŽETAK

Simulacijski alati su uvelike doprinijeli bržem razvoju i implementaciji prometnih rješenja. Od početka njihova korištenja bili su u velikoj prednosti naspram klasičnih metoda istraživanja i evaluacije prometnih rješenja pogotovo u pogledu brzine izvedbe i financijskog aspekta. Napretkom tehnologije i sve većim računalnim kapacitetima prometne simulacije su postajale sve kompleksnije pogotovo u pogledu postavljanja simulacije, sa sve većim brojem mogućnosti i detalja koji se mogu podesiti pri postavljanju simulacije. Kako je kompleksnost postavljanja simulacije na visokoj razini, ima smisla promotriti proces i pokušati ga pojednostaviti. U ovom radu prezentirana je aplikacija za automatizaciju dijela postavljanja simulacije korištenjem Simulation of Urban MObility (SUMO) simulacijskog paketa, ideja je automatizirati dio procesa postavljanja simulacije uz pomoć web aplikacije koja će generirati zadane datoteke za postavljanje simulacije u SUMO simulacijskom alatu.

KLJUČNE RIJEČI: SUMO, simulacijski alati, prometni podaci, automatizacija

## SUMMARY

Simulation software has greatly contributed to the faster development and implementation of traffic solutions. Since the beginning traffic simulation software was at a big advantage against classic methods of researching and evaluating traffic solutions especially in the financial aspect and the time necessary to perform them. With the development in technology and computer power traffic simulations were getting more complex especially the process of setting up the simulation required more knowledge and resources because of the increase in features and parameters that can be adjusted. As the complexity of setting up the traffic simulation is at a high level, it makes sense to try and simplify it. In this case the traffic simulation software of choice is Simulation of Urban Mobility (SUMO) simulation package and the idea is to automate part of the process of setting up the simulation with the help of Python scripts running on a web application which will generate files necessary for the setup of the simulation in the SUMO simulation package.

KEY WORDS: SUMO, simulation tools, traffic data, automation

## SADRŽAJ

1. Uvod.....	1
2. Pregled simulacijskih alata .....	3
2.1 Mikroskopski simulacijski model.....	3
2.2 Makroskopski simulacijski model.....	4
2.3 Simulacijski alati.....	4
2.4 Simulation of urban mobility - SUMO.....	5
2.4.1 SUMO mrežne karte.....	6
2.4.2 SUMO generiranje prometne potražnje .....	7
2.4.3 SUMO pokretanje simulacije .....	7
2.4.4 SUMO izlazne datoteke.....	8
3. Problem procesa simulacije prometnih podataka.....	9
3.1 Pregled prometnih podataka .....	10
3.2 Ručno generiranje prometnih podataka.....	10
3.3 Automatizirani postupak generiranja prometnih podataka .....	12
4. Tehnologije korištene pri izradi aplikacije .....	15
5. Korištenje aplikacije.....	19
6. Zaključak.....	23

# 1 Uvod

Provođenje eksperimenata nad prometnim sustavom u stvarnom svijetu s ciljem poboljšanja njegove učinkovitosti je složen, dugotrajan i financijski zahtjevan postupak. U današnje doba dugogodišnji razvoj tehnologije u pogledu računalne snage i programskih alata omogućio je prikazivanje i simuliranje prometnog sustava u simulacijskim programima. Simulacijski alati omogućuju jednostavnije postavljanje i provođenje istraživanja nad prometnim sustavom, najveća prednost simulacije je brzina izrade i provođenja te jednom kada se izradi simulacijski model može se koristiti koliko god puta je potrebno. U ovom radu koristit će se Simulation of Urban MObility (SUMO) simulacijski paket. SUMO je simulator prometa otvorenog koda, dizajniran je za rukovanje velikim mrežama te izradu intermodalnih prometnih modela. Zbog svoje kompleksnosti i količine posebnih značajki postoji mnoštvo parametara koji se trebaju podesiti prije nego što se simulacija može pokrenuti, da bi se dodatno skratio proces postavljanja simulacije automatizirati će se određeni dio prikupljanja ulaznih parametara (postavki). Program koji će automatizirati proces prikupljanja ulaznih parametara izrađen je u Python programskom jeziku uz korištenje Django radnog okruženja za izgradnju web aplikacije. Cilj ovog završnog rada je automatizirati proces postavljanja određenih parametara prometne potražnje za pokretanje simulacije u SUMO simulacijskom alatu. Rad je podijeljen na šest poglavlja:

1. Uvod
2. Pregled simulacijskih alata
3. Problem procesa simulacije prometnih podataka
4. Tehnologije korištene pri izradi aplikacije
5. Korištenje aplikacije
6. Zaključak.

U drugom poglavlju opisuju se dvije osnovne strategije simuliranja prometa, te se dobiva uvid u neke od dostupnih simulacijskih programa na tržištu.

Treće poglavlje opisuje prometne podatke koji se automatiziraju i uvodi u problematiku simuliranja.

Četvrto poglavlje prikazuje tehnologije korištene pri izradi aplikacije. Uz opis svake od korištenih tehnologija prikazane su prednosti korištenja svake.

U petom poglavlju detaljno je prikazan proces korištenja izrađene web aplikacije za automatizaciju postavljanja procesa simulacije koristeći SUMO simulatora.

U posljednjem poglavlju dan je zaključak rada.



## 2 Pregled simulacijskih alata

Računalne simulacije postaju sve popularniji alat području znanosti. Znanstvenici rješavaju različite probleme koristeći modernu računalnu snagu za pokretanje kompleksnih simulacijskih okolina. Simulacijske okoline koriste se za ispitivanje znanstvenih modela da bi se ustanovila njihova izvedivost i točnost.

Današnja računalna moć omogućila je provođenje simulacija brzinama puno većim nego što je to moguće izvesti u stvarnom svijetu. Također, simulacije omogućavaju istraživačima sigurnu okolinu u kojoj je moguće provoditi eksperimente bez negativnih posljedica u stvarnom svijetu.

Jedan od sustava koji je najbolje proučavan koristeći računalnu simulaciju jest prometni sustav. Uobičajeno je provoditi eksperimente nad prometnim sustavom u simuliranom okruženju iz razloga praktičnosti, omjera uloženog i dobivenog te s gledišta financijskog aspekta koji je znatno povoljniji za razliku od stvarnog svijeta.

Računalni znanstvenici razvili su dvije strategije simuliranja prometa:

1. Makroskopska simulacija
2. Mikroskopska simulacija [1].

### 2.1 Mikroskopski simulacijski model

Mikro simulacijski modeli su modeli gdje se kretanje pojedinih vozila po prometnoj mreži definira jednostavnim pravilima slijeđenja vozila, prestrojavanja vozila i razmaku između vozila [2]. Dinamične varijable modela predstavljaju mikroskopske vrijednosti poput pozicije i brzine pojedinačnih vozila. Ovaj model može se podijeliti u dvije kategorije: *Cell automata model*, gdje su vrijeme i prostor diskretne varijable i *continuous model*, gdje je vrijeme kontinuirana varijabla u vremenu.

Prvi mikroskopski modeli razvijeni su u šezdesetim godinama prošlog stoljeća. Slijedeća strategija se koristila za izgradnju dinamičnih modela. Jednadžba gibanja za svako vozilo temeljena je na pretpostavci da pojedini vozači reagiraju na podražaj od drugih vozila na neki poseban način. Ta reakcija je izražena u vidu akceleracije. Pojedini podražaj može biti funkcija položaja vozila u prostoru i njihove vremenske derivacije. Funkcija položaja vozila dobiva se tako što se pretpostavlja da vozači pojedinih vozila poštuju prometne zakone i znakove u svakom trenutku u cilju izbjegavanja prometnih nesreća [3].

Većina prvih mikroskopskih simulacijskih modela zasnivala se na *follow-the-leader* modelu kojeg su koncipirali Gazis, Herman i Rothery. Također postoje još *Optimal Velocity Model* i *Generalized Force Model*.

## 2.2 Makroskopski simulacijski model

Makroskopski prometni model pretpostavlja dovoljno velik broj vozila na prometnici u tolikoj mjeri da se svaki tok vozila može sagledati kao neprekinuti tok vozila u cijevi. Postoje tri bitne varijable u makroskopskom prometnom modelu one su:

- Protok vozila,  $q$  [voz/h]: broj vozila koji prođu kroz promatrani presjek prometnice u jedinici vremena u jednom smjeru za jednosmjerne prometnice ili u oba smjera za dvosmjerne prometnice.
- Brzina prometnog toka,  $v$  [km/h]: Srednja vrijednost brzina svih vozila koja sudjeluju u promatranom prometnom toku.
- Gustoća prometnog toka,  $\rho$  [voz/km]: broj vozila na jedinici duljine prometnice, po prometnoj traci [4].

## 2.3 Simulacijski alati

Postoji mnoštvo simulacijskih alata, svaki posjeduje drugačije značajke te zbog toga svaki ima svoje prednosti i mane. Simulacijske alate možemo uspoređivati po slijedećim značajkama:

1. Otvoreni kod
2. Kompatibilnost s operacijskim sustavima
3. Dokumentacija
4. Korisničko sučelje
5. Postavljanje simulacije
6. Grafički prikaz simulacije
7. Rezultati simulacije

U ovom radu se koristi SUMO simulacijski paket koji će se detaljnije opisati kasnije, također neki od dobro poznatih i često korištenih simulacijskih alata su slijedeći:

- PTV Vissim – mikroskopski simulacijski alat koji omogućuje detaljan i realističan prikaz prometnog toka. Jednostavan je za korištenje i nisu potrebne dodatne skripte za njegovu uporabu. Podržava detaljno modeliranje prometne potražnje, ponude i ponašanja sudionika u prometu [5].
- Aimsun Next – simulacijski alat koji spaja mikrosimulaciju, mezoskopsku simulaciju, makroskopske funkcionalnosti, modeliranje prometne potražnje te dvije hibridne simulacije (makro-meso i mikro-meso) sve u jednoj aplikaciji [6].
- Matsim –simulacijsko radno okruženje otvorenog koda koji omogućuje implementaciju „agent-based“ modela na prometnoj simulaciji velikih razmjera. Matsim se sastoji od nekoliko modula koji se mogu kombinirati ili koristiti zasebno [7].

## 2.4 Simulation of urban mobility - SUMO

SUMO je mikroskopski simulacijski alat otvorenog koda dizajniran za velike prometne mreže. SUMO podržava intermodalne simulacije uključujući pješački promet i dolazi s velikim brojem alata za kreiranje raznih scenarija. Postoji mogućnost simuliranja kako se određeni prometni tok koji se sastoji od pojedinačnih vozila kreće kroz zadanu prometnu mrežu. Simulator je isključivo mikroskopski, svako vozilo se zasebno modelira, ima svoju rutu i kreće se individualno kroz mrežu. Uz navedeno, važna značajka SUMO simulatora je mogućnost pokretanja na više operacijskih sustava koja je dijelom dobivena korištenjem C++ programskog jezika i pripadajućih biblioteka, također bitna je njegova interoperabilnost dobivena korištenjem Extensible Markup Language (XML) strukturiranih datoteka [8].

SUMO simulacijski alat dolazi sa svim potrebnim aplikacijama potrebne za postavljanje i provedbu simulacije. Slijedeće aplikacije se mogu koristiti za izradu simulacije:

- NetEdit – aplikacija s grafičkim sučeljem za uređivanje mrežnih karti,
- Netconvert – python skripta koja služi za pretvaranje uvezene digitalne karte u mrežnu kartu formata koji SUMO podržava,

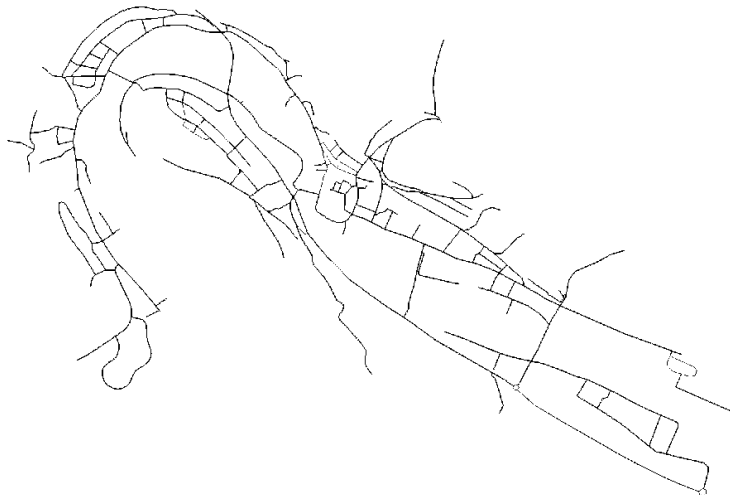
- Od2trips – python skripta korištena za pretvaranje Odredišno-Dolazišnih (OD) matrica u putovanja pojedinačnih vozila,
- Duarouter – python skripta korištena za računanje najbrže rute,
- SUMO - aplikacija naredbenog retka bez grafičkog sučelja

### 2.4.1 SUMO mrežne karte

SUMO mrežna karta predstavlja dio karte vezan uz promet, pod time se smatraju ceste i križanja po kojima se kreću simulirana vozila (slika 1.). Ugrubo SUMO mrežna karta predstavlja usmjereni graf. Vrhovi grafa, su u SUMO kontekstu raskrižja a bridovi grafa predstavljaju prometnice. Mrežne karte sadrže sljedeće informacije:

- Prometnice, kao i zasebne prometne trake
- Semaforska logika referencirana raskrižjima
- Raskrižja, uključujući prednost prolaska

Karte koriste Kartezijev koordinatni sustav s metričkim koordinatama. Kako bi se digitalna karta uspješno učitala u SUMO simulator mora se prvo prilagoditi koristeći prije spomenuti alat netconvert, netconvert podržava razne formate kao što su Open Street Map (OSM), VISUM, Vissim itd. [8].



Slika 1. SUMO mrežna karta

## 2.4.2 SUMO generiranje prometne potražnje

Nakon što se generira mrežna karta sljedeći korak bi bio dodavanje automobila u simulaciju, taj korak se naziva modeliranje prometne potražnje. Za modeliranje prometne potražnje bitni su sljedeći pojmovi: putovanje definirano kao kretanje entiteta od jednog mjesta do drugog definiranom početnim bridom (ulicom), završnim bridom i vremenom početka. Ruta je prošireno putovanje definirano ne samo početnim i završnim bridom nego i svim bridovima kojim će entitet proći [8].

Postoji mnogo načina kako generirati prometnu potražnju, odabir ovisi o vrsti ulaznih podataka koje imamo:

- Koristeći rute- koristi se kada se potražnja želi kreirati ručno ili uz pomoć posebno izrađene skripte
- Koristeći definiciju toka - sličan pristup poput korištenja ruta samo što se entiteti koji imaju isti početni i završni brid grupiraju
- Koristeći nasumično generirane rute – koristi se kada ne postoji pristup prometnim podacima međutim, u tom slučaju rezultati simulacije nisu realni
- Koristeći OD matrice
- Koristeći podatke prometnih detektora

## 2.4.3 SUMO pokretanje simulacije

Za pokretanje prometne simulacije u SUMO simulacijskom alatu potrebne su sljedeće ulazne datoteke: datoteka koja sadrži mrežnu kartu i datoteka koja sadrži rute vozila, također mogu postojati i dodatne datoteke koje učitavaju dodatne entitete kao što su modovi rada semafora, induktivne petlje i stanice za bus. Postoji redoslijed učitavanja datoteka u simulaciju, prvo se učitava mrežna karta zatim se dodatne datoteke učitaju nakon toga se otvaraju datoteke koje sadrže rute i učitaju se prvih n koraka te svakih n vremenskih koraka čitaju se rute za sljedećih n vremenskih koraka.

Svaka simulacija zahtjeva definiranje vremenskog perioda unutar kojeg će se simulacija odvijati. Vrijeme trajanja simulacije zadaje se opcijama početak i kraj. Simulacija završava u sljedećim slučajevima: Zadano je vrijeme završetka simulacije i to vrijeme je dostignuto, vrijeme završetka simulacije nije definirano ali su sva vozila

„prošla“ simulaciju i posljednji slučaj kad korisnik zadaje naredbu zaustavljanja simulacije [8].

#### 2.4.4 SUMO izlazne datoteke

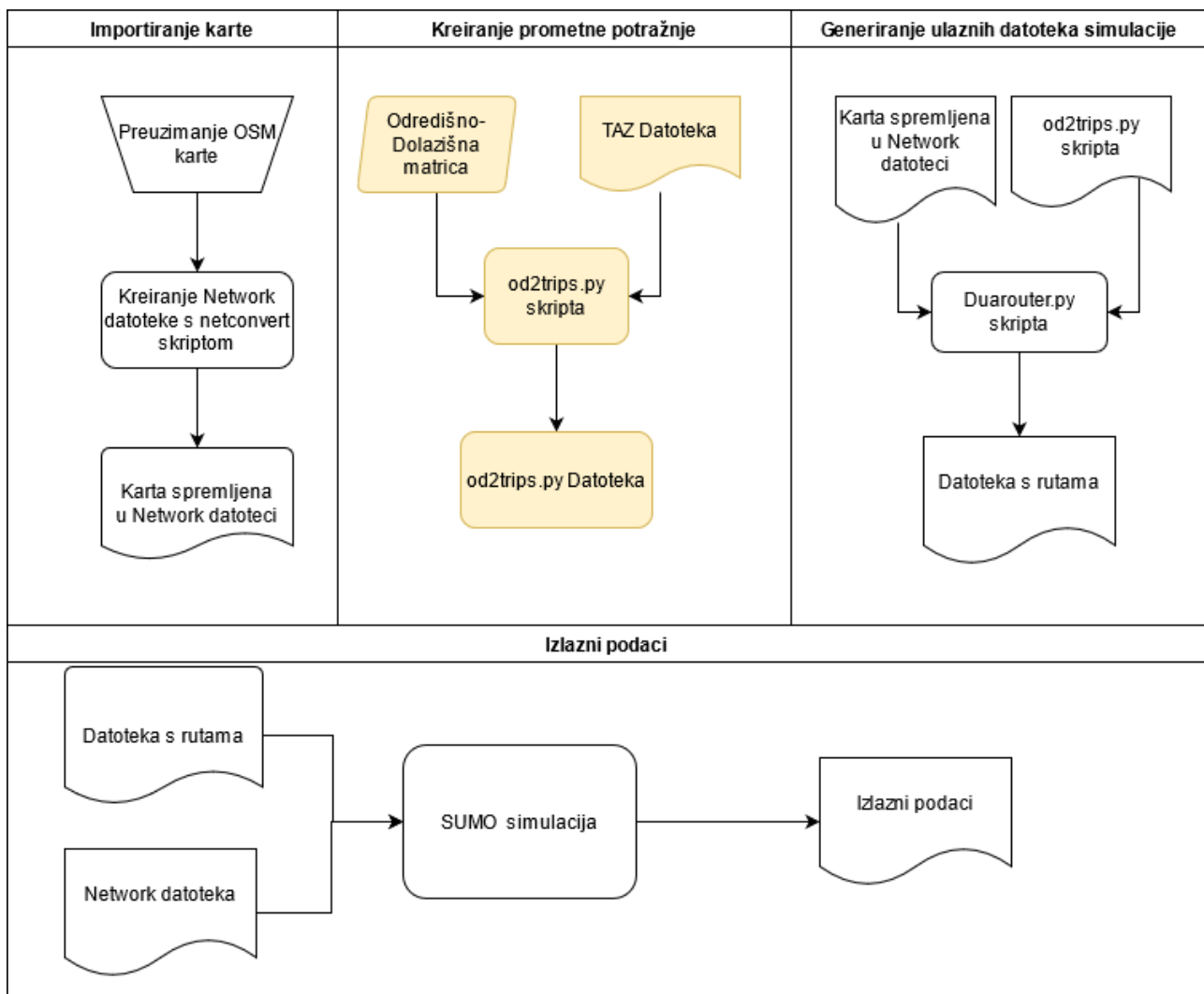
SUMO dozvoljava generiranje velikog broja raznolikih podataka o simulaciji. Oni se zapisuju u datoteke ali mogu se ispisivati u komandni redak. U zadanom stanju oni su onemogućeni i mogu se pojedinačno aktivirati ili preko komandnog retka ili preko dodatnih datoteka pri postavljanju simulacije [8].

Sve izlazne datoteke generirane od strane SUMO simulatora su u *Extensible Markup Language* (XML) formatu. Dostupne su izlazne datoteke koje sadrže informacije o vozilima:

- Pozicija vozila: pozicija i brzina svih vozila za sve simulirane vremenske korake
- Emisije: Brojke o emisijama svih vozila za sve simulirane vremenske korake
- *Floating Car Data* (FCD): uključuje ime, poziciju, kut i tip svakog vozila
- Razni podaci o bridovima, prometnim trakama i vozilima
- Podaci o poziciju i brzini za vizualizaciju.

### 3 Problem procesa simulacije prometnih podataka

Metodologija kreiranja simulacije u SUMO simulacijskom alatu prikazana je u četiri koraka na slici 2. Prvi korak je preuzimanje karte i generiranje mrežne datoteke. Drugi korak je generiranje datoteke koja sadrži pojedinačna putovanja vozila iz OD matrica i *Traffic Assignment Zone* (TAZ) datoteke, taj proces se ujedno i automatizirao uz pomoć web aplikacije. Treći korak je generiranje datoteke koja sadrži rute pojedinačnih vozila, i četvrti korak je generiranje izlaznih podataka iz simulacije [9].



Slika 2. Metodologija izrade simulacije pomoću SUMO simulatora

izvor: [9]

### 3.1 Pregled prometnih podataka

Zbog svoje kompleksnosti, za uspješno pokretanje simulacije u SUMO simulacijskom alatu potrebno je ispravno kreirati određene datoteke koje će se koristiti u postavljanju simulacije od strane SUMO aplikacija. Te datoteke sadrže parametre i postavke za pokretanje simulacije i bitno je da su pojedinačno ispravno formatirane, isto tako je bitno da su datoteke međusobno kompatibilne iz razloga što pojedine SUMO aplikacije koriste više datoteka u procesu postavljanja simulacije te ako ne postoji konzistentnost u imenovanju pojedinih parametara simulacija se neće moći uspješno izvršiti.

Prometni podaci potrebni za pokretanje simulacije [8]:

- Kartografska projekcija (engl. *network*) koja sadrži sve prometnice (engl. *edges*), raskrižja (engl. *junctions*), te spojeve (engl. *connectionn*)
- Polazište (engl. *Origin*) je parametar koji predstavlja točku na prometnici na karti od koje vozilo započinje putovanje
- Odredište (engl. *Destination*) je parametar koji predstavlja točku na prometnici na karti gdje vozilo završava putovanje
- Datoteka koja sadrži moguće polazišne i odredišne lokacije TAZ datoteka. TAZ predstavlja područje na karti koje korisnik proizvoljno odredi. U tom području karte nalaze se prometnice koje se u kasnijem postupku mogu koristiti kao početna i ili završna odredišta pojedinih putovanja.
- Datoteka koja sadrži listu Odredišno-Dolazišnih (OD) matrica sastoji se od polazišnog TAZ-a, odredišnog TAZ-a te broja vozila koja se kreću od polazišne do odredišne točke.

### 3.2 Ručno generiranje prometnih podataka

Ručno generiranje prometnih podataka odnosi se na ručno definiranje datoteka prometne potražnje (TAZ datoteka i OD datoteka) ili uređivanje već postojećih datoteka. Prije početka, bitno je znati da se vozilo u SUMO alatu sastoji od tri čimbenika:

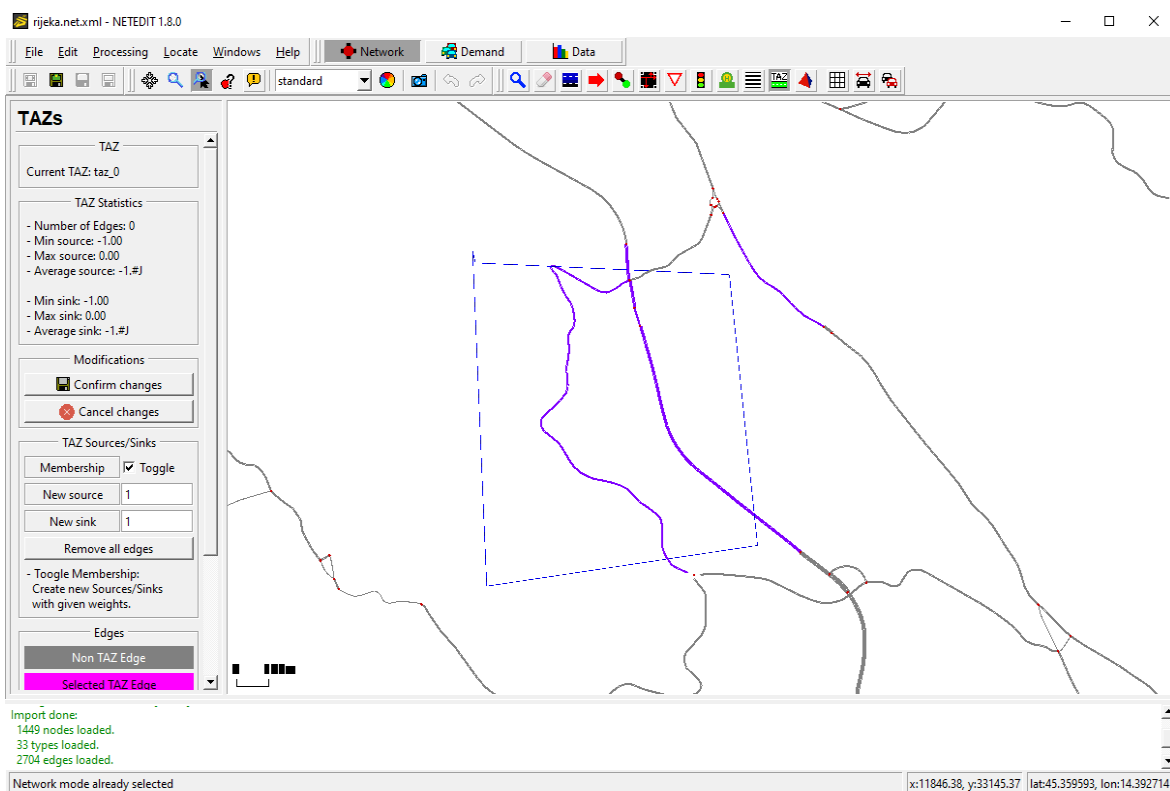
- Tip vozila koji opisuje njegove fizičke proporcije,
- ruta po kojoj će vozilo voziti,



- te identifikacijski broj vozila.

Ruta i tip vozila se oboje mogu dijeliti s više vozila. Nije obavezno definirati tip vozila, ako se ne definira uzima se zadana vrijednost. Također vozač vozila se ne treba modelirati eksplicitno.

Pri ručnom generiranju prometnih podataka prvi korak je otvoriti kartu područja na kojemu želimo provoditi simulaciju, u ovome slučaju to je napravljeno u ranije spomenutom alatu Nedit. Nedit (slika 3) ima mogućnost odabira tri načina rada, jedan za uređivanje mrežnih karti, drugi za uređivanje objekata vezano uz promet i treći način rada vezan za uređivanje prometnih podataka.



Slika 3. Prikaz definiranja TAZ zone u Nedit programu

Slijedeći korak nakon što se učita karta u program je pritisnuti gumb TAZ koji se nalazi na alatnoj traci u gornjem dijelu alata vidljivom na 3. slici te početi crtati poligon koji će definirati TAZ zonu. Kao što se može vidjeti na 3. slici TAZ zona označena je poligonom s isprekidanom plavom linijom, unutar poligona mogu se vidjeti prometnice koje prolaze kroz taj poligon označene ljubičastom bojom, isto tako po želji se mogu i označiti prometnice neposredno izvan poligona isto tako se mogu odznačiti prometnice

koje se nalaze u poligonu. Nakon crtanja poligona te odabirom željenih prometnica može se kreirati TAZ zona pritiskom na gumb „Save changes“. Ako se sve uspješno spremi dobije se datoteka naziv\_datoteke.taz.xml. Uz TAZ datoteku potrebno je također izraditi datoteku naziv\_datoteke.od formata koja sadrži rute vozila te broj vozila koja se njome kreću. Zatim se uz pomoć skripte od2trips.py prikazane na slici 2. koja prima .taz.xml datoteku i .od datoteku kreira nova datoteka formata naziv\_datoteke.odtrips.xml. Nakon toga preostaje zadnji korak poziv duarouter.py skripte koja prima kartu spremljenu u formatu .net.xml te datoteku generiranu u prošlom koraku formata odtrips.py te se dobije datoteka naziv\_datoteke.odtrips.rou.xml i uz pomoć te datoteke i datoteke koja sadrži informacije o karti može se pokrenuti SUMO grafičko korisničko sučelje te pokrenuti i izvršiti simulacija.

### **3.3 Automatizirani postupak generiranja prometnih podataka**

Cilj automatiziranog postupka generiranja prometnih podataka je jednak kao kod ručnog, a to je na kraju uspješno pokretanje simulacije. Međutim velika razlika je u broju koraka potrebnih da se dođe do krajnjeg cilja između ručnog načina i automatiziranog, to se naravno manifestira u kraćem vremenu potrebnom za postavljanje simulacije kod automatiziranog postupka generiranja prometnih podataka nego kod ručnog.

Već na samom početku postoji velika prednost a to je automatski otvorena mrežna karta bez ikakve potrebe da ju korisnik sam otvara. Kod automatiziranog postupka ne treba se crtati poligon već se samo treba odabrati lokacija na karti gdje se želi postaviti TAZ zona, a aplikacija automatski nacrtava poligon oko odabrane lokacije i pronađe najbliže prometnice oko odabrane lokacije. Paralelno uz odabir TAZ zone i generiranje datoteke .taz.xml koja je prikazana na slici 4. izrađuje se .od datoteka koja je prikazana na slici 4. i ona još prima dodatan argument a to je broj vozila koji korisnik jednostavno unosi u tekstni okvir nazvan „broj vozila“.

```

-<taz id="6" color="blue" shape="14944.260139764636,9335.472591821104 14791.606582477456,9111.332436813973">
  <tazSource id="-304148" weight="1"/>
  <tazSink id="-304148" weight="1"/>
</taz>
-<taz id="7" color="red" shape="11307.525706762448,7750.113682623953 11154.702088792692,7526.065443934873">
  <tazSource id="-304694" weight="1"/>
  <tazSink id="-304694" weight="1"/>
</taz>
-<taz id="8" color="blue" shape="14990.112328207353,9404.271815098822 14837.46217993286,9180.13046988938">
  <tazSource id="-305375" weight="1"/>
  <tazSink id="-305375" weight="1"/>
</taz>

```

Slika 4. Prikaz generirane TAZ datoteke

Na slici 4. se vidi struktura TAZ datoteke te isto tako parametri pojedinog TAZ-a. Parametar „id“ služi označavanju svakog TAZ-a te je on jedinstven broj, parametar „color“ predstavlja boju nacrtanog poligona u svrhu lakšeg snalaženja i prikaza, parametar „shape“ predstavlja koordinate poligona kojeg je aplikacija sama generirala na temelju lokacije za TAZ koju je korisnik odabrao. „tazSource“ i „tazSink“ su također opisani parametrom id koji je jedinstven za pojedinu prometnicu međutim može se ponavljati više puta u TAZ datoteci ako se odabere ista prometnica više puta.

```

1 $OR;D2
2 * From-Time To-Time
3 5.30 6.45
4 * Factor
5 1.00
6 *
7 * some
8 * additional
9 * comments
10 1 2 12
11 3 4 45
12 5 6 56
13 7 8 45
14 9 10 15
15 11 12 20
16 13 14 23

```

Slika 5. Prikaz .od datoteke

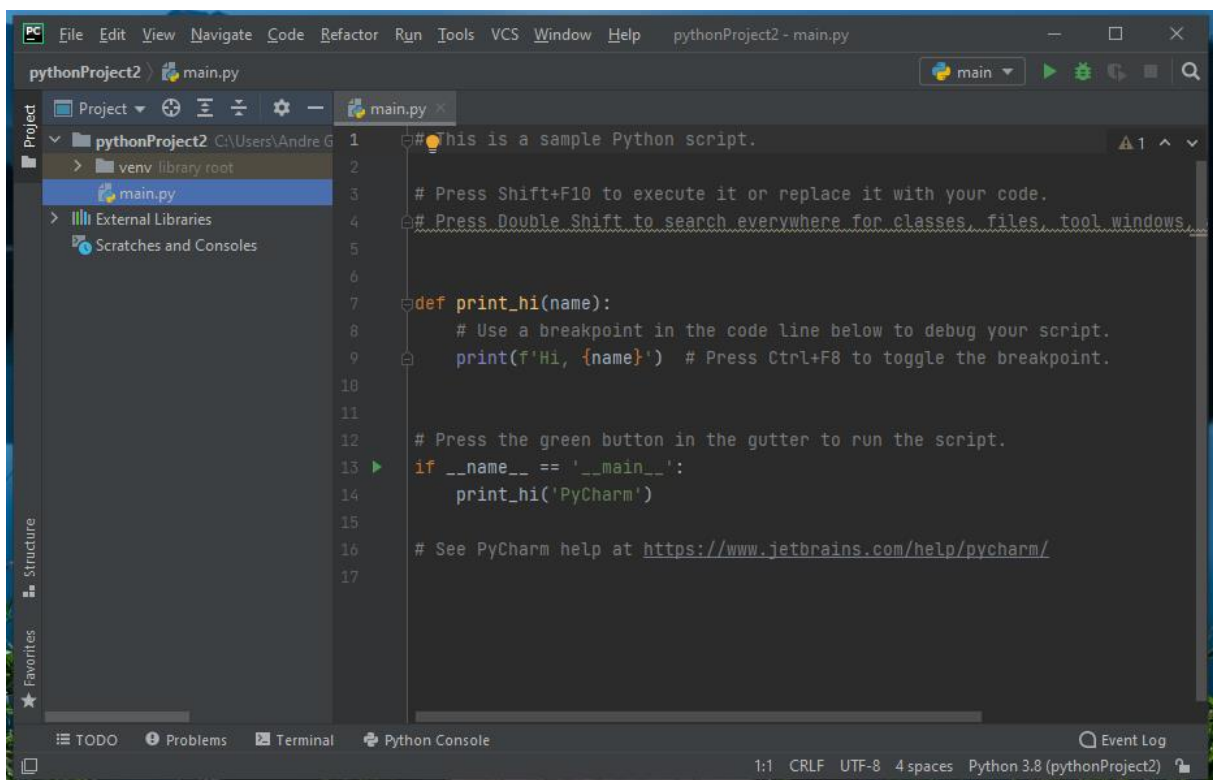
Na slici 5. se vidi generirana .od datoteka koja sadrži rute vozila te broj vozila koja se njome kreću, također mora se obratiti pažnja na strukturiranje dokumenta kada se ručno radi što ovdje nije slučaj zato što aplikacija automatski generira datoteku ispravnog formata.

Aplikacija ujedno automatizira i posljednje korake te se ne mora kao u ručnom postupku otvarati naredbeni redak i pozivati od2trips.py i duarouter.py skripta već to aplikacija sve sama radi i na kraju se dobije datoteke .odtrips.rou.xml koja se zajedno s kartom pokreće u SUMO grafičkom sučelju za početak simulacije.

## 4 Tehnologije korištene pri izradi aplikacije

Završni rad izrađen je u formatu web aplikacije, u većem dijelu procesa izrade koristio se programski jezik Python, ponešto Javascript programski jezik te HTML i CSS.

Razvojno okruženje u kojem se razvijala web aplikacija je Pycharm od Češke tvrtke JetBrains. Pycharm je specijalizirani *Integrated Development Enviroment* (IDE) za Python programski jezik [10]. Postoji *Professional Edition* verzija koji se plaća, te besplatna verzija programa - *Community Edition* (slika 6). Na slici 6. vidi se početni zaslon kada se prvi put otvori projekt u Pycharm programskom okruženju. Razvojno okruženje Pycharm nudi u *Community Edition* verziji koja se koristila u izradi ovog rada brojne mogućnosti koji olakšavaju i ubrzavaju proces razvoja programskog rješenja poput inspekcije koda, restrukturiranje koda, inteligentnog Python uređivača i mnoge druge.

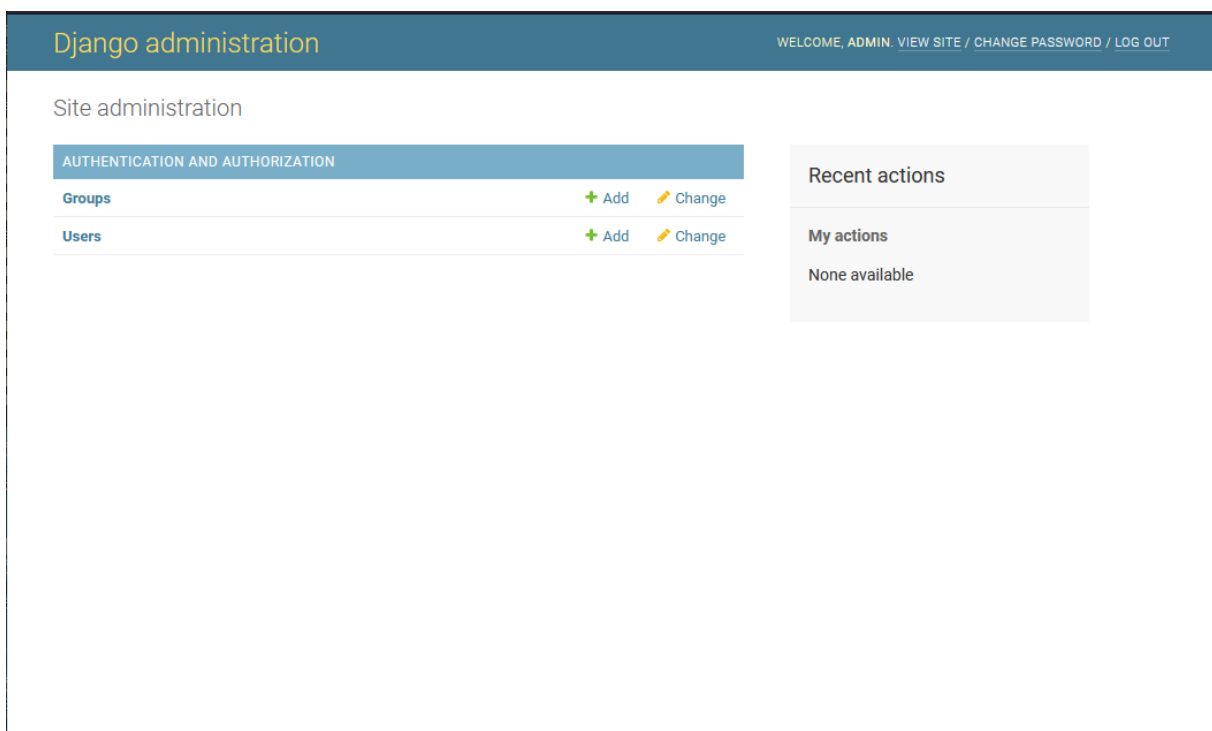


Slika 6. Prikaz novog projekta u Pycharm razvojnom okruženju

Web aplikacija izrađena je u Python programskom jeziku uz pomoć Django radnog okruženja. Django je Python radno okruženje za razvijanje web aplikacija koji

omogućuje brzu izradu, sigurno upravljanje korisničkim podacima i skalabilnu web aplikaciju [11].

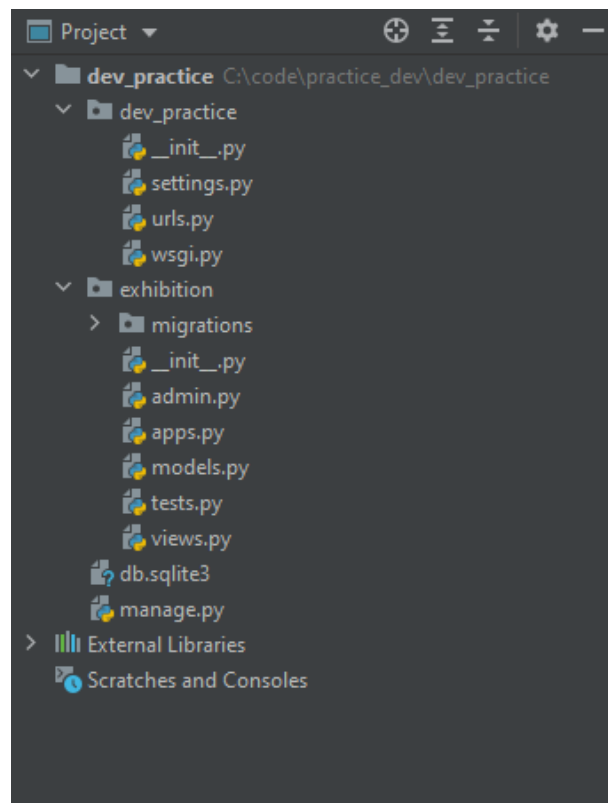
Django je odabran iz razloga što nudi širok raspon značajki i funkcionalnosti što znači da se programer može više fokusirati na bitnije stvari. Jedna takva značajka je ugrađeni *Object-Relational Mapper* (ORM) koji omogućuje interakciju s relacijskim bazama podataka kao što su SQLite, PostgreSQL i MySQL bez pisanja *Structured Query Language* (SQL) koda već Django ORM obavlja sve od izrade baze podataka, kreiranja tablica i odnosa između tablica te samog unosa, brisanja i modificiranja podataka unutar baze podataka. Također Django koristi *Model Template View* (MVT) koncept, gdje *Model* omogućuje lako upravljanje podacima koji se povlače iz baze podataka, *Template* je prezentacijski sloj gdje se definira koji podaci se pokazuju na web stranici i na koji način se oni prezentiraju i *View* je zaslužan za uspješnu vezu između *Model* i *Template* sloja. Još jedna zgodna značajka Django radnog okruženja je ugrađeno administratorsko sučelje (slika 7) koje se vrlo jednostavno postavi i u svega par linija koda omogućuje izradu korisničkih profila, upravljanjem bazom podataka i mnogim drugim parametrima kao posjećenost stranice i pojedine IP adrese s kojih se stranica posjetila [11].



Slika 7. Django administratorsko sučelje

Jedan od glavnih razloga zašto koristiti Django je brzina razvijanja, Django nudi gotovu „šprancu“ za razvoj aplikacije koja kad se pokrene izradi sve potrebne datoteke i generira sav potreban kod kako bi se odmah moglo početi s razvijanjem aplikacije (slika 8). Također je dobro što je sigurnost na visokoj razini tako što Django ima ugrađenu sigurnost protiv slijedećih prijetnji: *Cross Site Scripting (XSS)* omogućuje zlonamjernom korisniku da učita skriptu u preglednik drugih korisnika tako što skriptu pospremi u bazu podataka gdje će ona biti korištena od strane drugih korisnika, *Cross Site Request Forgery (CSRF)* kod kojeg zlonamjerni korisnik izvodi radnje koristeći se kredencijama od drugog korisnika bez njegovog znanja, *SQL Injection* je tip napada gdje zlonamjerni korisnik uspije pokrenuti proizvoljan SQL kod na bazi podataka i potencijalno izbrisati ili podijeliti osjetljive podatke iz baze [11].

Jedna od glavnih nedostataka Django radnog okruženja je vrijeme potrebno da se dobro shvati, iako je rađen u Python programskom jeziku što pojednostavljuje sintaksu zbog velikog broja značajki treba ipak ponešto vremena da se savlada, također nije pogodan za male projekte jer zbog svoje kompleksnosti koristi puno više resursa nego drugi alati što povećava troškove.



Slika 8. Struktura projekta koju je Django izgenerirao

Za dodavanje mrežne karte u web aplikaciju koristila se JavaScript biblioteka otvorenog koda Leaflet. Leaflet omogućuje jednostavno dodavanje mrežne karte po izboru na web stranicu. Uz to ima dodanih mogućnosti poput dodavanja markera, crtanja poligona po karti, podrška za mobilne uređaje te razne mogućnosti nadogradnje za vlastite potrebe [12]. Leaflet se koristio zbog njegove jednostavnosti u vidu implementacije koja je uz to dobro objašnjena i dokumentirana (slika 9). Sve što je potrebno za implementiranje karte je definirati *tileLayer*, poveznicu za kartu koju želimo koristiti te definirati početnu točku u kojoj će se karta otvoriti.

```
var map = L.map('map').setView([51.505, -0.09], 13);

L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
  attribution: '&copy; <a href="https://www.openstreetmap.org/copyright">OpenStreetMap</a> contributors'
}).addTo(map);

L.marker([51.5, -0.09]).addTo(map)
  .bindPopup('A pretty CSS3 popup.<br> Easily customizable.')
  .openPopup();
```

Slika 9. Pokretanje OSM karte s Leaflet bibliotekom, [12]



## 5 Korištenje aplikacije

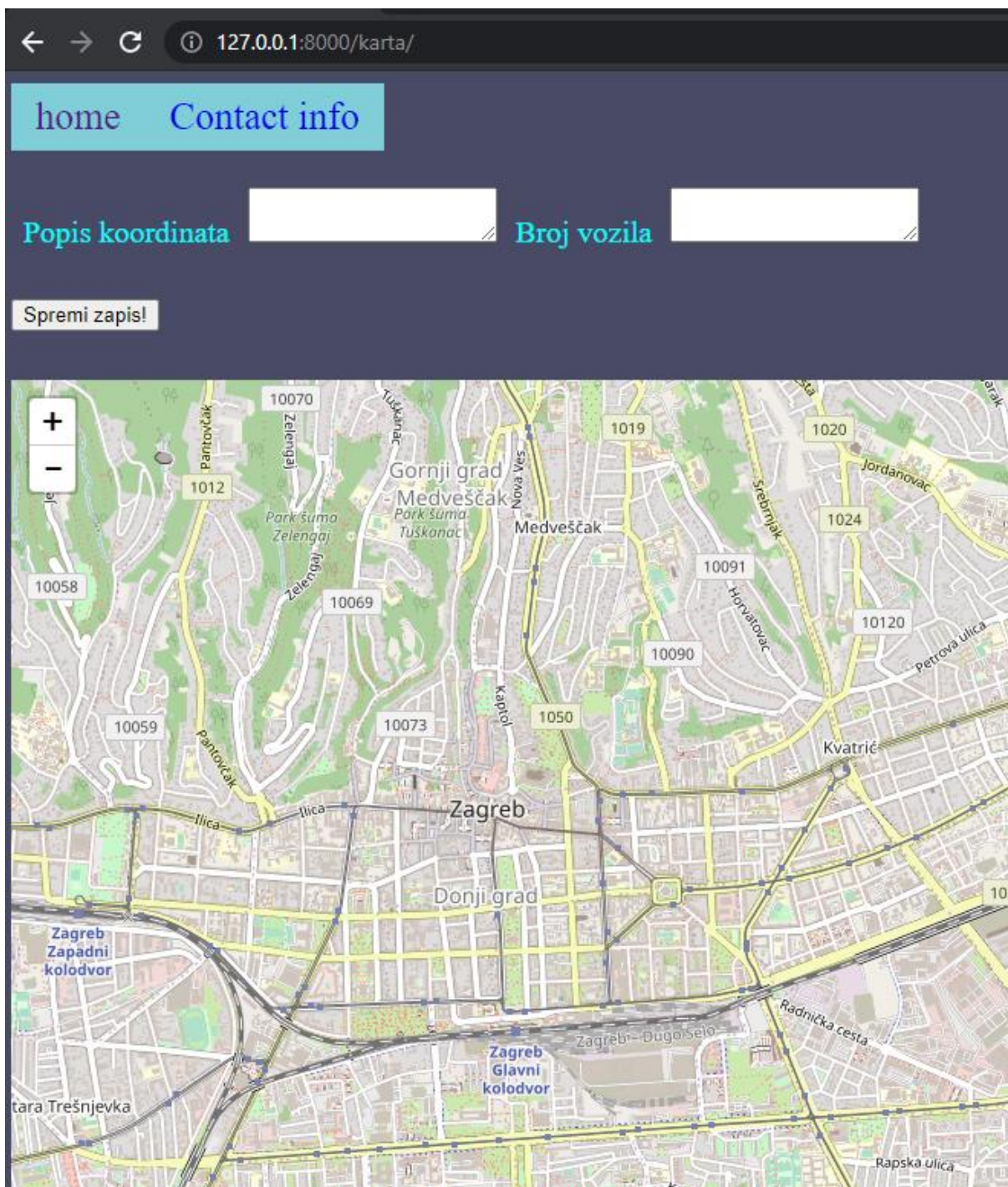
Web aplikacija je izrazito jednostavna za korištenje, ona se pokreće iz Windows naredbenog retka. Za pokretanje web aplikacije na računalu potrebno je preuzeti i instalirati Python programski jezik verziju 3.8, slijedeće Python pakete: Django, Sumo, Elementtree, uz Python pakete također je potrebno preuzeti Javascript biblioteku Leaflet. Slijedeći korak je otvoriti Windows naredbeni redak navigirati do mjesta pohrane web aplikacije na računalu, ulasku u samu mapu te izvršiti slijedeću naredbu: „python manage.py runserver“. Uspješno pokrenuta web aplikacija iz Windows naredbenog retka te njezina internetska adresa koja se jednostavno potraži na mrežnom pregledniku (slika 10).

```
C:\ag\Zavrsni>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
July 05, 2021 - 10:30:39
Django version 2.2.14, using settings 'Zavrsni.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

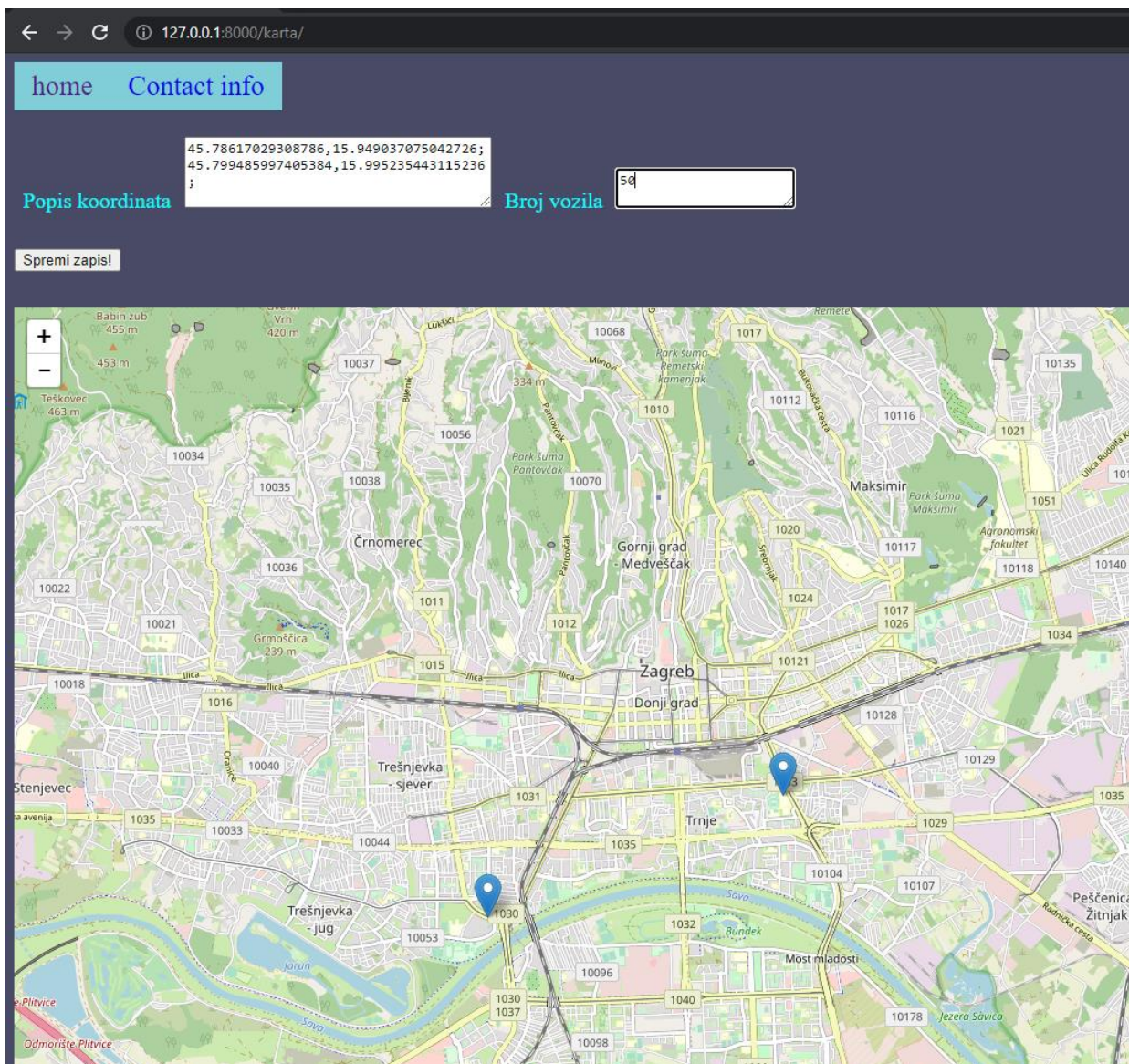
Slika 10. Pokretanje web aplikacije iz Windows naredbenog retka

Nakon što se pokrenula aplikacija i otvorila u mrežnom pregledniku stisne se gumb stvori zapis koji vodi do stranice na kojoj se nalazi karta koja služi za odabir početnih i završnih lokacija pojedinih putovanja te isto tako broja vozila koje će se kretati od početne do završne točke. Na slici 11. vidi se polje za unos koordinata s nazivom „Popis koordinata“ ona služi za prikazivanje koordinata koje je korisnik odabrao pritiskom na kartu, međutim isto tako se mogu direktno u tekstnu kutiju unijeti koordinate ako se želi postaviti neku specifičnu lokaciju pri tome treba biti pažljiv i odvojiti dvije koordinate sa znakom točka-zarez ( ; ).



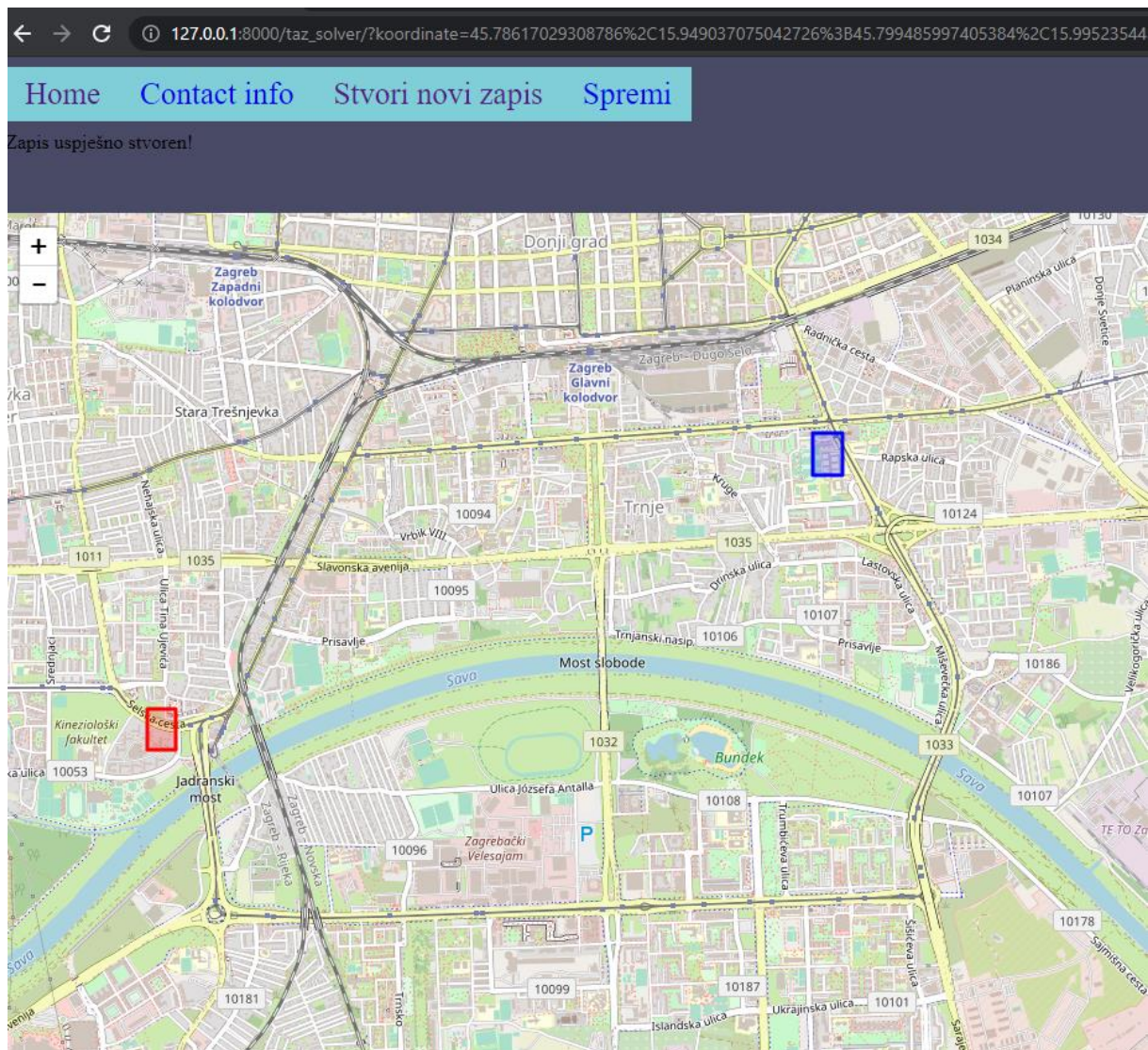
Slika 11. Prikaz stranice za odabir točaka interesa

Nakon uspješno odabranih koordinata početne i završne točke za pojedino putovanje te upis broja vozila u za to predviđenu tekstnu kutiju zaslon izgleda kao prikazan na slici 12. Slijedeći korak je pritisnut gumb „Spremi zapis!“ kako bi se zapis dodao. Dobiveni zapis sprema se u ranije spomenutu TAZ datoteku te isto tako OD datoteku.



Slika 12. postavljanje početne i završne točke putovanja te broja vozila

Na slici 13. prikazan je zaslon koji potvrđuje da je zapis uspješno spremljen. Isto tako se prikazuje mrežna karta na kojoj su označena dva pravokutnika koja predstavljaju lokacije koje je korisnik odabrao, crvenom bojom prikazana je početna točka putovanja a plavom bojom odredišna točka putovanja.



Slika 13. prikaz uspješno stvorenog zapisa

Također možemo stisnuti gumb „Stvori novi zapis“ na slici 13. i napraviti novi zapis koji će se nadodati u spomenutu TAZ datoteku i OD datoteku. Taj korak možemo ponavljati sve dok ne izradimo dovoljan broj zapisa.

Nakon što je korisnik gotov s odabirom TAZ zona treba se stisnuti gumb spremi kako bi se pokrenula od2trips.py skripta koja prima TAZ datoteku i OD datoteku kao ulaz a vraća datoteku naziv\_datoteke.odtrips.xml koja se u kombinaciji s kartom postavi kao ulaz duarouter.py skripti te ona generira datoteku naziv\_datoteke.odtrips.rou.xml koja se otvori u SUMO grafičkom sučelju zajedno s kartom kako bi se simulacija pokrenula te kako bi se dobila izlazna datoteka simulacije.

## 6 Zaključak

Simulacijski alati su uvelike promijenili proces implementacije i istraživanja prometnih rješenja. Svojom brzinom izvedbe, raznim mogućnostima, širokom dostupnošću pokazali su se iznimno vrijednim alatom bilo kojeg inženjera. Naravno s pojavom brojnih mogućnosti konfiguracije simulacije i njenih parametara kompleksnost pokretanja simulacije se povećala.

U ovom završnu radu razvijena je web aplikacija za potrebe jednostavnijeg i bržeg postavljanja ulaznih parametara prometne simulacije u SUMO simulacijskom alatu. Web aplikacija zasniva se na Django radnom okruženju koji se koristi za razvijanje web rješenja u Python programskom jeziku. Cilj web aplikacije je generirati dvije datoteke, prva je TAZ datoteka koja sadrži listu svih koordinata početnih i završnih točki koje je korisnik odabrao u TAZ zoni, druga je OD datoteka koja sadrži uređeni par početnih i završnih lokacija s brojem vozila koji se kreće na toj relaciji. Također nakon postavljanja ulaznih parametara moguće je pokrenuti simulaciju direktno iz web aplikacije.

## Literatura

- [1] Kotusevski G, Hawick K. A review of traffic simulation software. Institute of information and mathematical sciences, Massey University, New Zealand. 2009.
- [2] Introduction to micro simulation. Preuzeto sa: <http://www.microsimulation.drfox.org.uk/intro.html> (24.05.2021).
- [3] Popping J. An overview of microscopic and macroscopic traffic models. Leiden University, Netherlands. 2013.
- [4] Dadić I, Kos G, Ševrović M. Teorija prometnog toka. Fakultet prometnih znanosti, Zagreb. Preuzeto sa: <http://files.fpz.hr/Djelatnici/msevrovic/Teorija-prometnih-tokova-2014-skripta.pdf> (25.05.2021).
- [5] PTV group. Preuzeto sa: <https://www.ptvgroup.com/en/solutions/products/ptv-vissim/> (08.06.2021).
- [6] Aimsun Next. Preuzeto sa: <https://www.aimsun.com/aimsun-next/> (08.06.2021).
- [7] Matsim. Preuzeto sa: <https://www.matsim.org/about-matsim> (09.06.2021).
- [8] Lopez PA, Behrisch M, Bieker-Walz L, Erdmann J, Flotterod YP, Hilbrich R, et al. Microscopic Traffic Simulation using SUMO. IEEE Conference on Intelligent Transportation Systems, ITSC. 2018.
- [9] Majstorović Ž, Garašić A, Tišljarić L, Carić T. Simulation of the urban mobility using cellular network data: case study for the City of Rijeka Croatia. Proceedings of the International Scientific Conference "The Science and Development of Transport" (ZIRP), 2021. (Prihvaćeno za objavu)
- [10] Pycharm. Preuzeto sa: <https://www.jetbrains.com/pycharm/guide/> (21.07.2021).
- [11] Django. Preuzeto sa: <https://docs.djangoproject.com/en/3.2/> (21.07.2021).

[12] Leaflet. Preuzeto sa: <https://leafletjs.com/reference-1.7.1.html>  
(21.07.2021).

## Popis slika

Slika 1. SUMO mrežna karta .....	6
Slika 2. Metodologija izrade simulacije pomoću SUMO simulatora .....	9
Slika 3. Prikaz definiranja TAZ zone u Netedit programu .....	11
Slika 4. Prikaz generirane TAZ datoteke .....	13
Slika 5. Prikaz .od datoteke .....	13
Slika 6. Prikaz novog projekta u Pycharm razvojnom okruženju .....	15
Slika 7. Django administratorsko sučelje .....	16
Slika 8. Struktura projekta koju je Django izgenerirao .....	17
Slika 9. Pokretanje OSM karte s Leaflet bibliotekom, [12] .....	18
Slika 10. Pokretanje web aplikacije iz Windows naredbenog retka .....	19
Slika 11. Prikaz stranice za odabir točaka interesa .....	20
Slika 12. postavljanje početne i završne točke putovanja te broja vozila .....	21
Slika 13. prikaz uspješno stvorenog zapisa .....	22





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 **APLIKACIJA ZA AUTOMATIZACIJU PROCESA**  
**SIMULACIJE PROMETNIH PODATAKA**

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

Student/ica:

U Zagrebu, 1.9.2021 \_\_\_\_\_

\_\_\_\_\_ *Andre Gerašić*  
(potpis)