

# Obrada i vizualizacija podataka pružatelja javnog gradskog prijevoza u stvarnom vremenu

---

Šarić, Ante

Undergraduate thesis / Završni rad

2023

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

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:119:617149>

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

*Download date / Datum preuzimanja:* **2025-02-07**



*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET PROMETNIH ZNANOSTI

Ante Šarić

**OBRADA I VIZUALIZACIJA PODATAKA PRUŽATELJA  
JAVNOG GRADSKOG PRIJEVOZA U STVARNOM  
VREMENU**

**ZAVRŠNI RAD**

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU  
FAKULTET PROMETNIH ZNANOSTI

ZAVRŠNI RAD

**OBRADA I VIZUALIZACIJA PODATAKA PRUŽATELJA  
JAVNOG GRADSKOG PRIJEVOZA U STVARNOM  
VREMENU**

**REAL-TIME DATA PROCESSING AND VISUALIZATION  
OF URBAN PUBLIC TRANSPORTATION PROVIDERS**

**Mentor:** dr. sc. Tomislav Erdelić

**Student:** Ante Šarić

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

**JMBAG:** 0135259815

Zagreb, lipanj 2023.

# OBRADA I VIZUALIZACIJA PODATAKA PRUŽATELJA JAVNOG GRADSKOG PRIJEVOZA U STVARNOM VREMENU

Sažetak:

U ovom radu istražuje se obrada i vizualizacija podataka pružatelja javnog gradskog prijevoza u realnom vremenu, s fokusom na primjeru ZET-a. Cilj je pružiti korisnicima jednostavno i učinkovito sučelje za praćenje podataka o javnom prijevozu kako bi se olakšalo planiranje putovanja i poboljšala kvaliteta usluge. Rad koristi Python i C# kao glavne programske jezike, te SQL bazu podataka za pohranu prikupljenih podataka. Python se koristi za prikupljanje podataka s poslužitelja javnog prijevoza ZET-a. Korištenjem odgovarajućih biblioteka i tehnika, podaci se preuzimaju u stvarnom vremenu i obrađuju kako bi se dobio koristan skup informacija. Zatim se prikupljeni podaci pohranjuju u SQL bazu podataka radi daljnje analize i obrade. Baza podataka omogućava povijesni pregled podataka i omogućuje dublju analizu trendova i uzoraka. Obrada podataka uključuje segmentaciju podataka, filtriranje, i izračunavanje koje će korisnicima pružiti korisne informacije o voznim redovima, kašnjenjima i drugim aspektima javnog prijevoza. C# programski jezik se koristi za izradu grafičkog korisničkog sučelja koje omogućuje vizualizaciju podataka. Korisnici mogu pregledavati različite informacije o vozilima, rutama, stanicama i vremenima dolaska. Interaktivno sučelje omogućuje korisnicima prilagodbu prikaza i pretraživanje podataka prema svojim potrebama. Kombinacija Pythona, SQL baze podataka i C# programa omogućava sveobuhvatnu obradu, pohranu i vizualizaciju podataka pružatelja javnog gradskog prijevoza. Ova integracija tehnologija pruža korisnicima praktičan alat za praćenje i planiranje putovanja javnim prijevozom. Dodatno, omogućuje analitičarima i stručnjacima da istraže i razumiju trendove i performanse javnog prijevoza te na temelju toga predlože poboljšanja.

**Ključne riječi:** obrada podataka, vizualizacija podataka, javni gradski prijevoz, ZET, Python, C#, SQL baza podataka, analiza podataka, stvarno vrijeme.

# REAL-TIME DATA PROCESSING AND VISUALIZATION OF URBAN PUBLIC TRANSPORTATION PROVIDERS

## Abstract:

This paper explores the processing and visualization of real-time public urban transportation data, with a focus on the case of ZET. The aim is to provide users with a simple and efficient interface for tracking public transportation data, facilitating trip planning and improving service quality. The paper utilizes Python and C# as the main programming languages, along with an SQL database for storing the collected data. Python is used for data retrieval from the ZET public transportation server. By employing appropriate libraries and techniques, data is acquired in real-time and processed to obtain a useful set of information. The collected data is then stored in an SQL database for further analysis and processing. The database allows for a historical overview of the data and enables in-depth analysis of trends and patterns. Data processing involves data segmentation, filtering, and calculation to provide users with valuable insights into schedules, delays, and other aspects of public transportation. C# programming language is employed to develop a graphical user interface that facilitates data visualization. Users can access various information about vehicles, routes, stops, and arrival times. The interactive interface allows users to customize the display and search for data according to their needs. The combination of Python, SQL database, and C# programming enables comprehensive processing, storage, and visualization of public urban transportation data. This integration of technologies provides users with a practical tool for tracking and planning public transportation trips. Additionally, it enables analysts and experts to explore and understand trends and performance in public transportation, leading to potential improvements.

Key words: data processing, data visualization, public urban transportation, ZET, Python, C#, SQL database, data analysis, real-time.

Zagreb, 27. travnja 2023.

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

## ZAVRŠNI ZADATAK br. 7194


Pristupnik: **Ante Šarić (0135259815)**  
Studij: **Inteligentni transportni sustavi i logistika**  
Smjer: **Inteligentni transportni sustavi**

Zadatak: **Obrada i vizualizacija podataka pružatelja javnog gradskog prijevoza u stvarnom vremenu**

### Opis zadatka:

Cilj rada je izraditi programsku podršku za prikupljanje podataka o javnom gradskom prijevozu s toka podataka u stvarnom vremenu. Prvotno je potrebno napisati program koji se spaja na javno objavljeni stvarno-vremeni tok podataka, te potom proučiti formate, standard i vrstu podataka dobivenih stvarno-vremenog toka. Na temelju analize podataka potrebno je dizajnirati relacijsku bazu podataka i napisati programsku podršku za kontinuirano spremanje podataka iz stvarno-vremenog toka u bazu podataka. Potom je potrebno provesti detaljnu analizu prikupljenih podataka o vozilima javnog gradskog prijevoza, na temelju sljedećih kriterija: vremena putovanja, duljine putovanja, vremena stajanja, učestalosti u rasporedu i sl. Dodatno, za potrebe lakše analize potrebno je napraviti vizualizaciju podataka u formi grafova i slika.

Mentor:



---

dr. sc. Tomislav Erdelić

Predsjednik povjerenstva za  
završni ispit:

---

## Sadržaj

1	UVOD .....	1
2	PRUŽATELJ JAVNOG GRADSKOG PRIJEVOZA ZET .....	3
2.1	Vozni park ZET .....	5
2.2	Linije ZET .....	7
2.2.1	Tramvajske linije .....	8
2.2.2	Autobusne Linije .....	8
2.3	GTFS ZET .....	11
2.3.1	GTFS .....	11
2.3.2	Statički podaci .....	11
2.3.3	Realtime podaci .....	13
2.3.4	Baze podatka .....	15
3	INFORMIRANJE PUTNIKA I VOZAČA .....	20
3.1	Predputno informiranje .....	20
3.2	Putno informiranje .....	20
3.3	Informiranje o javnom prijevozu .....	21
4	OPIS PODATAKA I STANDARDA PRUŽATELJA JAVNOG GRADSKOG PRIJEVOZA .....	23
4.1	Python skripta za skidanje podataka .....	24
4.2	Python skripta za spremanje podataka u bazu podataka .....	28
5	PROGRAMSKO SUČELJE ZA VIZUALIZACIJU I ANALIZU STVARNO VREMENSKIH PODATAKA .....	29
5.1	Visual Studio .....	29
5.2	Kod za programsko sučelje ZET_GTFS .....	31
5.3	Programsko sučelje ZET_GTFS .....	35
6	ANALIZA POVIJESNO PRIKUPLJENIH PODATAKA .....	36
7	<b>ZAKLJUČAK .....</b>	<b>39</b>
	<b>LITERATURA .....</b>	<b>40</b>
8	DODATAK 1- Popis ZET linija .....	41
	<b>POPIS SLIKA .....</b>	<b>49</b>
	<b>POPIS TABLICA .....</b>	<b>51</b>

# 1 UVOD

U suvremenim gradovima, javni gradski prijevoz igra ključnu ulogu u osiguravanju učinkovite i održive mobilnosti građana. Uzimajući u obzir rastući broj putnika i sve složenije prometne mreže, pružatelji javnog gradskog prijevoza suočavaju se s izazovom pružanja ažurnih informacija o rasporedima, kašnjenjima i drugim relevantnim podacima putnicima. Obrada i vizualizacija podataka pružatelja javnog gradskog prijevoza u realnom vremenu predstavljaju ključne komponente za poboljšanje iskustva putnika i optimizaciju prometnog sustava. Kroz adekvatnu obradu podataka pružatelji mogu pratiti položaj vozila, analizirati prometne tokove i detektirati prometne gužve, dok vizualizacija podataka omogućuje putnicima da na intuitivan način pristupe relevantnim informacijama.

Cilj ovog završnog rada je istražiti i razviti sustav obrade i vizualizacije podataka pružatelja javnog gradskog prijevoza u stvarnom vremenu. Korištenjem suvremenih tehnologija, kao što su Python, SQL i C# za vizualizaciju, ovaj rad će se fokusirati na razvoj aplikacije koja će omogućiti pružateljima gradskog prijevoza praćenje, analizu i prikaz podataka u realnom vremenu. Pored toga, vizualizacija podataka kroz intuitivna i interaktivna sučelja omogućit će putnicima da lakše planiraju svoje putovanje, prilagode se izmjenama i imaju bolji pregled nad cijelim prometnim sustavom.

Ovaj završni rad ima za cilj pridonijeti razvoju inteligentnih prometnih sustava, unaprijediti iskustvo putnika i doprinijeti učinkovitijem i održivijem javnom gradskom prijevozu. Kroz primjenu naprednih tehnologija i analitičkih metoda, očekuje se da će rezultati ovog istraživanja imati praktičnu primjenu u stvarnom svijetu, te će pružateljima i putnicima javnog gradskog prijevoza omogućiti donošenje informiranih odluka i povećanje kvalitete usluge. U nastavku rada, detaljno će se analizirati metode obrade podataka, tehnologije vizualizacije i primjene u kontekstu javnog gradskog prijevoza. Također, opisat će se metodologija razvoja sustava, implementacija aplikacije i evaluacija rezultata. Kroz sve ove faze, cilj je stvoriti inovativan sustav obrade i vizualizacije podataka koji će pridonijeti daljnjem razvoju prometnih sustava i poboljšati iskustvo putnika. Naslov završnog rada je obrada i vizualizacija podataka pružatelja javnog gradskog prijevoza u realnom vremenu te je rad podijeljen u 7 tematskih cjelina:

1. Uvod
2. Pružatelj javnog gradskog prijevoza ZET
3. Informiranje putnika
4. Opis podataka i standarda pružatelja javnog gradskog prijevoza
5. Programsko sučelje za vizualizaciju i analizu stvarno vremenskih podataka
6. Analiza povijesno prikupljenih podataka
7. Zaključak.



Svaka od ovih tematskih cjelina ima svoju ulogu u istraživanju i razvoju sustava obrade i vizualizacije podataka pružatelja javnog gradskog prijevoza u stvarnom vremenu. Navedeni naslovi pružaju strukturiranu organizaciju rada i omogućuju čitateljima da se usredotoče na specifične aspekte teme. U uvodu se uvodi čitatelje u temu istraživanja, postavlja se kontekst i ciljevi rada te se daje pregled strukture rada.

Druga cjelina je posvećena pružatelju javnog gradskog prijevoza ZET. U ovoj se cjelini opisuju karakteristike pružatelja, njegova mreža i usluge, kao i važnost informiranja putnika o prometnom sustavu. Treća cjelina istražuje informiranje putnika i analizira postojeće metode i alate koji se koriste za pružanje informacija putnicima javnog gradskog prijevoza. Ovdje se istražuje važnost pravovremenih i ažurnih informacija za putnike.

Četvrta cjelina fokusira se na opis podataka i standarda pružatelja javnog gradskog prijevoza. U ovoj cjelini analiziraju se vrste podataka koje pružatelj prikuplja, njihova struktura i kvaliteta te postojeći standardi koji se primjenjuju. Peta cjelina se bavi razvojem programskog sučelja za vizualizaciju i analizu stvarno vremenskih podataka. Ovdje se istražuju tehnologije i alati za obradu podataka u stvarnom vremenu, kao i razvoj sučelja koje omogućuje putnicima pristup relevantnim informacijama.

Šesta cjelina se fokusira na analizu povijesno prikupljenih podataka. Ovdje se istražuju metode analize i interpretacije povijesnih podataka kako bi se izveli zaključci o prometnom sustavu i mogućim poboljšanjima. U zaključku se sumiraju glavni rezultati istraživanja i razvoja, iznose se zaključci o učinkovitosti i primjenjivosti razvijenog sustava te se raspravlja o mogućim daljnjim koracima i istraživačkim smjerovima.

## 2 PRUŽATELJ JAVNOG GRADSKOG PRIJEVOZA ZET

Od godine 1891., kada su 5. rujna prvi put planirano i organizirano gradskim ulicama krenula tramvajska kola, pokretana jednom konjskom snagom, u dnevnik se Zagrebačkog električnog tramvaja upisalo mnogo događaja. Tramvajski sustav u Zagrebu postao je ključna komponenta gradskog prijevoza i igrao je vitalnu ulogu u povezivanju različitih dijelova grada te olakšavanju svakodnevnih putovanja građana. Uvod tramvaja u Zagrebu predstavljao je revoluciju u prijevozu i bio je prvi korak prema razvoju modernog javnog gradskog prijevoza. Na **slici 1** prikazan je prvi zagrebački tramvaj, [1].



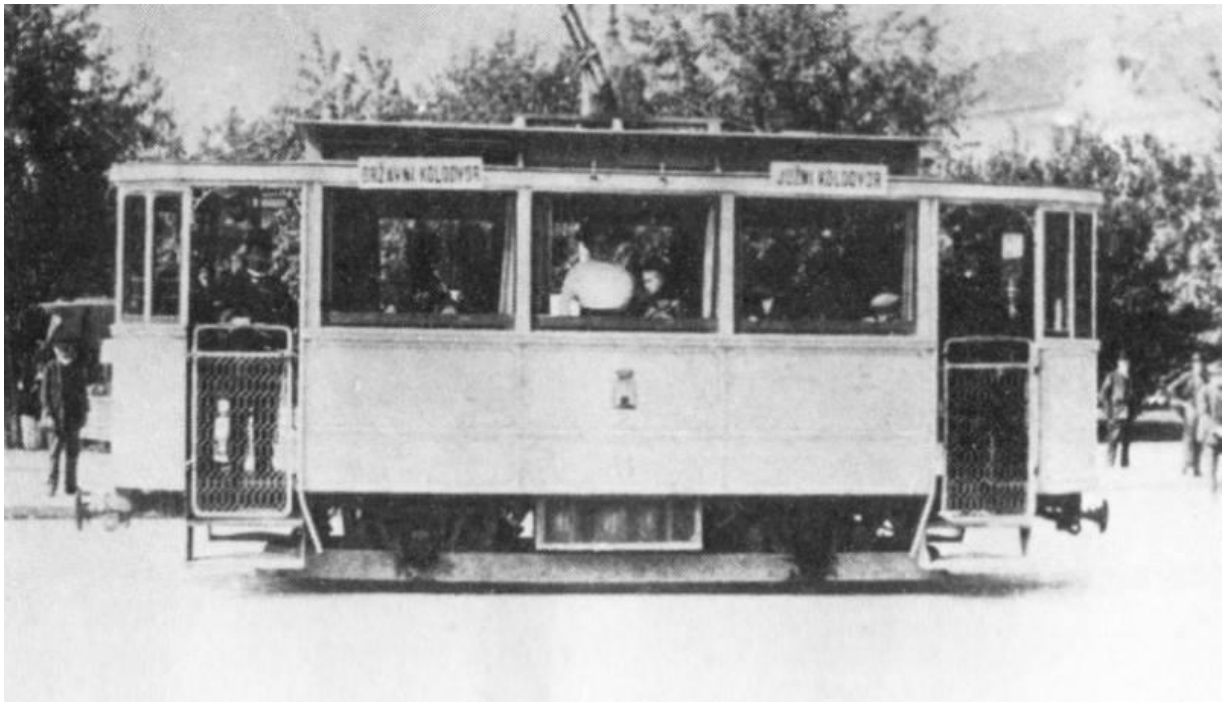
**Slika 1.** Prvi Zagrebački tramvaj, [2].

Na prvim stranicama elektrifikacija tramvaja u Zagrebu dogodila se 1910. godine prikazana na **slici 2**, čime je zamijenjena konjska snaga s električnim pogonom. Ova promjena omogućila je brže i učinkovitije putovanje putnika te je postavila temelje za daljnji razvoj javnog gradskog prijevoza u Zagrebu. Posebno se ističe godina 1931., kada se autobusni promet pridružio tramvajskom sustavu. Ova integracija omogućila je još veću dostupnost i fleksibilnost prijevoza za putnike, čineći gradski prijevoz u Zagrebu sveobuhvatnijim, [1].

Uspinjača prikazana na **slici 3**, koja je u funkciji još od 1890. godine, i sljemenska žičara koja je bila u ZET-ovom prometu od '63. do 2007. godine, imaju neizostavnu ulogu u zagrebačkom prijevozu putnika. Uspinjača je postala prepoznatljiv simbol Zagreba te je pružala jedinstveno iskustvo putnicima koji su se vozili uzbrdo prema Gornjem gradu. Nakon stanke zbog izgradnje, sljemensku žičaru naslijedila je nova i suvremenija verzija koja je puštena u promet u travnju 2022. godine. Ove posebne vrste prijevoza doprinose raznolikosti gradskog prijevoza i pružaju jedinstvene mogućnosti putovanja u Zagrebu, [1].

Kroz godine, tramvajski vozni park u Zagrebu prošao je kroz proces modernizacije, osobito od 2005. godine kada je prvi niskopodni tramvaj domaće proizvodnje počeo voziti zagrebačkim ulicama. Ova modernizacija vozila rezultirala je udobnijim i pristupačnijim prijevozom za putnike te je unaprijedila kvalitetu usluge. Dovršetak investicijskih projekata uključuje i nabavku novih autobusnih vozila, uključujući ona specijalizirana za prijevoz osoba s

invaliditetom. Zahvaljujući tim investicijama, Zagreb se danas ubraja među vodeće europske gradove u pogledu javnog gradskog prijevoza, [1].



**Slika 2.** Prvi električni tramvaj, [8].

Posebna pozornost pridaje se praćenju svjetskih industrijskih dostignuća i uvođenju novih tehnologija u javni gradski prijevoz Zagreba. Sustav za nadzor i upravljanje prometom omogućuje praćenje aktualne prometne situacije, koordinirano postupanje u izvanrednim okolnostima te digitalnu razmjenu važnih podataka u sklopu prometne mreže i voznog parka ZET-a. Ovaj kompleksni računalni sustav omogućuje automatiziranu internu distribuciju i obradu informacija, stvarajući pretpostavke za niz dodatnih usluga koje podižu standard javnog gradskog prijevoza u Zagrebu na najvišu razinu. Implementacija sofisticiranih tehnoloških rješenja stvara pretpostavke za suvremene prijevoznike principe, kao i prateće poslovne principe koji često dovode zagrebački javni prijevoznik do inovativnih poslovnih modela. S velikim brojem svakodnevnih korisnika, koji broje nešto manje od milijun na dan, zagrebački javni prijevoz postaje ključni aspekt gradskog života, [1].

Razvoj javnog gradskog prijevoza u Zagrebu od njegovih početaka do danas predstavlja impresivnu priču o napretku i inovacijama. Kontinuirana modernizacija vozila, implementacija novih tehnologija i unaprjeđenje sustava nadzora i upravljanja prometom omogućili su Zagrebu da postane vodeći europski grad u pogledu javnog gradskog prijevoza. Sveobuhvatna mreža tramvaja, autobusa, uspinjače i žičare pruža putnicima raznolike opcije za putovanje po gradu. Budući razvoj trebao bi se usredotočiti na daljnje unaprjeđenje usluga, poboljšanje ekološke održivosti i prilagodbu novim potrebama putnika. Zagrebački javni gradski prijevoz i dalje će igrati ključnu ulogu u povezivanju građana i doprinositi kvaliteti života u glavnom gradu Hrvatske, [1].



**Slika 3.** Uspinjača, [7].

## 2.1 Vozni park ZET

Trenutno stanje ZET-a obuhvaća širok spektar vozila i usluga koje omogućuju efikasan i raznolik javni gradski prijevoz u Zagrebu. U nastavku će se detaljno opisati svaka kategorija vozila i njihova uloga u sustavu ZET-a.

1. Autobusi: ZET raspolaže s 476 autobusnih vozila koja obavljaju prijevoz putnika diljem grada. Prosječna starost autobusa iznosi 11,9 godina, što ukazuje na potrebu za stalnom obnovom voznog parka kako bi se osigurala sigurnost i udobnost putnika. Autobusi su vitalni za povezivanje različitih dijelova grada i pružanje prijevoza stanovnicima i posjetiteljima Zagreba. Na **slici 4** prikazani su novi autobusi ZET-a.



**Slika 4.** Novi autobusi ZET-a, [9].

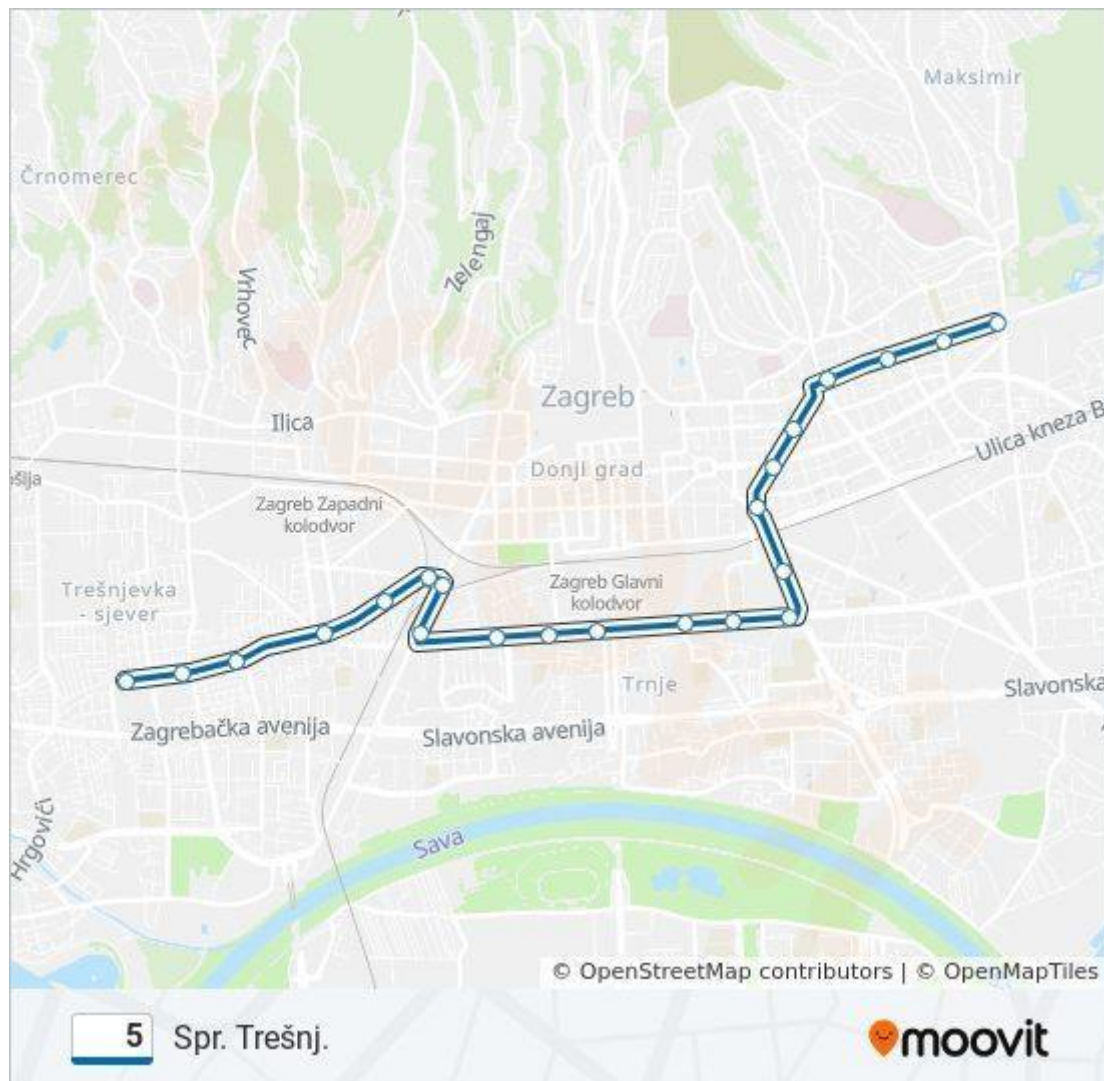
2. Tramvaji: U ZET-ovom voznom parku nalazi se 266 tramvajskih vozila. Od tog broja, 142 su niskopodna tramvajska vozila. Niskopodni tramvaji omogućuju lakši ulazak i izlazak putnicima, posebno osobama s invaliditetom ili ograničenom pokretljivošću. Tramvaji su jedan od najpopularnijih načina javnog prijevoza u Zagrebu te povezuju glavne gradske četvrti i stvaraju vitalnu mrežu javnog prijevoza.
3. Žičara: ZET također ima žičaru koja pruža jedinstvenu vrstu prijevoza u Zagrebu. Žičara je važan prijevozni segment jer povezuje različite dijelove grada koji su vertikalno udaljeni, omogućujući putnicima da se brzo i udobno kreću između njih. Ova vrsta prijevoza često je popularna među turistima koji žele istražiti zagrebačke atrakcije s visine.
4. Uspinjača: Uspinjača, dugogodišnji simbol Zagreba, također je dio ZET-ovog sustava javnog prijevoza. Ova povijesna uspinjača pruža putnicima jedinstveno iskustvo vožnje prema Gornjem gradu. Uspinjača je atrakcija sama po sebi te privlači turiste i lokalne stanovnike koji žele uživati u prekrasnom pogledu na grad dok se penju prema Gornjem gradu.
5. Vozila za prijevoz osoba s invaliditetom: ZET ima 31 vozilo posebno prilagođeno za prijevoz osoba s invaliditetom i djece s poteškoćama u razvoju. Ova vozila omogućuju pristupačnost i udobnost za putnike s posebnim potrebama, pružajući im siguran i pouzdan način putovanja.
6. Autobusi za prijevoz školske djece: ZET također ima autobuse namijenjene prijevozu školske djece. Ova vozila osiguravaju siguran prijevoz učenika do škole i natrag te su prilagođena njihovim potrebama i sigurnosnim standardima.
7. Turistički tramvaj, panoramski autobusi i vlakići: U sklopu turističke ponude, ZET nudi razne turističke usluge koje uključuju turistički tramvaj, panoramske autobuse i vlakiće. Ova vozila omogućuju turistima da istraže atrakcije i znamenitosti grada na udoban i informativan način.

8. Turistička elektro vozila: ZET također nudi turistička elektro vozila koja su ekološki prihvatljivija i tiša u usporedbi s drugim vozilima. Ova vozila pružaju alternativni način prijevoza za turiste i lokalne stanovnike koji žele istraživati grad na održiv način, [1].

Svi ovi aspekti ZET-ovog voznog parka i usluga čine sustav javnog gradskog prijevoza u Zagrebu raznolikim i prilagođenim potrebama putnika. Uzimajući u obzir potrebu za stalnom modernizacijom i obnovom vozila, ZET nastoji osigurati visoku kvalitetu usluge, sigurnost putnika i udobnost tijekom putovanja, [1].

## 2.2 Linije ZET

Prijevoz putnika u Zagrebu je danas izuzetno raznolik i obuhvaća širok spektar prijevoznih opcija. Dok su tramvaji neizostavan dio zagrebačkog prometnog pejzaža, autobusi su također ključni čimbenik u povezivanju različitih dijelova grada i okolnih područja. Sustav autobusnih linija pruža vitalnu podršku putnicima, omogućujući im da brzo i učinkovito stignu na svoje odredište. **Slika 5** prikazuje tramvajsku liniju 5., [1].



**Slika.5** Prikaz tramvajske linije 5, [6].

U gradu Zagrebu i okolnim područjima postoji ukupno 147 dnevnih autobusnih linija koje osiguravaju prometno povezivanje. Ove linije povezuju područja Grada Zagreba, Veliku Goricu i Zaprešić, pružajući putnicima mogućnost da se lako kreću između ovih mjesta. Osim toga, autobusni prijevoz je organiziran i u općinama Bistra, Luka i Stupnik, čime se omogućuje prijevoz putnika i izvan samog grada, [1].

### 2.2.1 Tramvajske linije

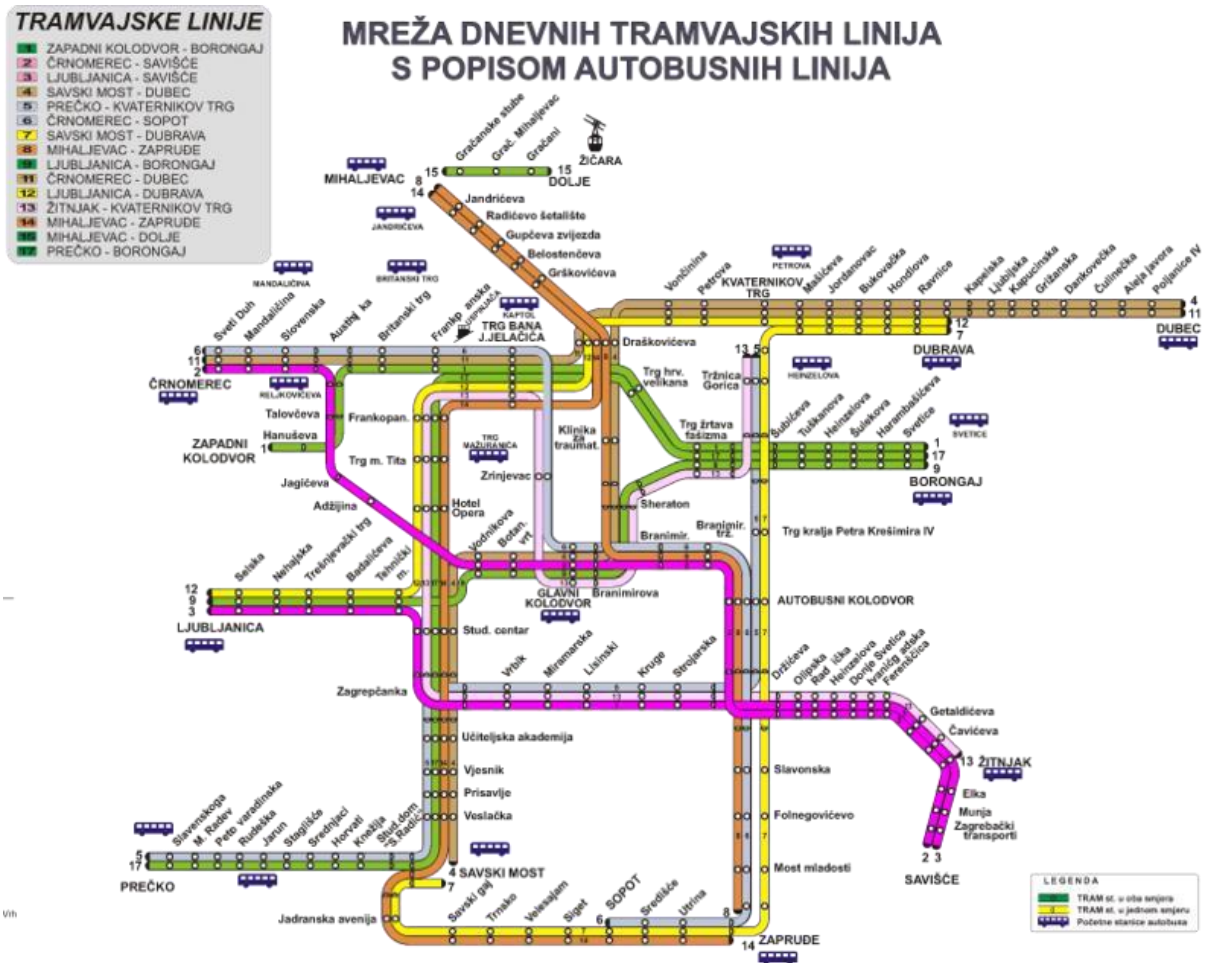
Tramvajski sustav u Zagrebu prikazan na **slici 6** obuhvaća različite linije koje povezuju različite dijelove grada. Evo pregleda nekih od tramvajskih linija u Zagrebu:

1. Linija 1: Zapadni kolodvor - Borongaj: ova linija povezuje zapadni dio grada, počevši od Zapadnog kolodvora, s naseljem Borongaj na istoku Zagreba. Tramvaji na ovoj liniji prolaze kroz središnje dijelove grada, pružajući važnu vezu između zapadnog i istočnog dijela grada.
2. Linija 2: Čnomerec - Savišće: linija 2 povezuje Čnomerec na zapadu grada s naseljem Savišće na istoku. Ova linija također prolazi kroz središnje dijelove grada i pruža prijevoz putnicima između različitih dijelova Zagreba.
3. Linija 3: Ljubljanića - Savišće: ova linija prolazi kroz naselje Ljubljanića i povezuje ga s naseljem Savišće. Putnici na ovoj liniji imaju pristup važnim dijelovima grada i mogu se lako kretati između njih.
4. Linija 4: Savski most - Dubec: linija 4 spaja naselje Savski most s naseljem Dubec na istoku grada. Ova linija omogućuje putnicima da se kreću između različitih dijelova grada, uključujući i prijelaz preko rijeke Save.
5. Linija 5: Prečko - Park Maksimir: ova linija povezuje naselje Prečko s parkom Maksimir. Putnici mogu uživati u prijevozu do popularnog parka i istraživati prirodnu ljepotu koju nudi, [1].

### 2.2.2 Autobusne Linije

U Zagrebu, glavnom gradu Hrvatske, postoji razvijena mreža autobusnih perona namijenjenih za dnevne autobusne linije. Ovi peroni predstavljaju važan čvorišni punkt za organizaciju prometa i omogućuju putnicima učinkovit prijevoz po gradu i okolnim područjima. Ukupno postoji 31 peron koji služe kao polazišta i odredišta za autobuse koji povezuju različite dijelove grada. **Slika 7** prikazuje jedan gradski peron i autobus na njemu, [1].

Svaki peron ima svoj jedinstveni broj i obično je označen oznakama koje ukazuju na destinacije i linije koje polaze ili dolaze na određenom peronu. Peroni su smješteni na prometno povoljnim lokacijama, često u blizini željezničkih postaja, tramvajskih stajališta ili drugih ključnih prometnih čvorišta. Uz to, autobusni peroni često su opremljeni informacijskim pločama koje pružaju putnicima važne podatke o rasporedu dolazaka i polazaka autobusa, kao i o destinacijama i linijama koje koriste određeni peron. Informacije pomažu putnicima da se lakše snađu u prometnoj mreži grada i pravovremeno planiraju svoje putovanje. Autobusni peroni u Zagrebu također često imaju prateće sadržaje poput nadstrešnica, klupa, rasvjete i digitalnih zaslona s informacijama o linijama i dolascima što prikazuje **slika 8**. Ovi sadržaji pružaju udobnost i sigurnost putnicima, posebno kad su vremenski nepovoljni uvjeti., [1].



Slika 6. Tramvajski promet u Zagrebu, [9].

Osim ovih linija, postoje i druge tramvajске linije koje povezuju različite dijelove grada, uključujući linije 6, 7, 8, 9, 11, 12, 13, 14 i 17. Svaka od ovih linija pruža putnicima pristup različitim dijelovima grada i omogućuje im brzo i učinkovito putovanje tramvajem. Također, za putnike koji trebaju prijevoz tijekom noći, postoje i noćne tramvajске linije. Linije 31, 32, 33 i 34 pružaju prijevoz tijekom noćnih sati i pomažu putnicima da stignu na svoja odredišta čak i kasno u noć, [1].





Slika 7. Prikaz perona sa autobusom, [5].



Slika 8. Prikaz perona i njegove infrastrukture, [4].

Zahvaljujući dobro organiziranoj mreži autobusnih perona i raznovrsnim dnevnim linijama, putnici u Zagrebu imaju mogućnost jednostavnog i učinkovitog putovanja autobusom

unutar grada i do okolnih područja. Autobusni peroni predstavljaju važan dio javnog prijevoza u Zagrebu, pružajući putnicima mogućnost brzog i praktičnog kretanja po gradu.

## 2.3 GTFS ZET

### 2.3.1 GTFS

General Transit Feed Specification (GTFS) je otvoreni standard koji se koristi za distribuciju relevantnih informacija o sustavima javnog prijevoza putnicima. Omogućuje javnim prijevoznim agencijama da objavljuju svoje podatke o javnom prijevozu u formatu koji može biti korišten od strane različitih softverskih aplikacija. Danas se GTFS format podataka koristi od strane tisuća pružatelja usluga javnog prijevoza, [3].

GTFS je podijeljen na komponentu rasporeda koja sadrži raspored, cijene i geografske informacije o prijevozu te komponentu u stvarnom vremenu koja sadrži predviđanja dolazaka, pozicije vozila i obavijesti o uslugama. Informacije o rutama, rasporedima i cijenama prikazane su u jednostavnim tekstualnim datotekama, što znači da se mogu stvarati i održavati bez korištenja kompliciranog ili vlasničkog softvera. GTFS se podržava diljem svijeta, a njezina upotreba, važnost i obuhvat sve više rastu. Vjerojatno većina agencija koje poznajete koristi GTFS za prikaz ruta, rasporeda, lokacija stajališta i drugih informacija, dok putnici već koriste te informacije putem različitih aplikacija, [3].

Neki putnici vole koristiti različite aplikacije ovisno o svojim potrebama. GTFS omogućuje putnicima da odaberu najbolju aplikaciju za planiranje putovanja. Nekim aplikacijama može biti lakše pristupiti ili bolje pružati informacije putnicima s invaliditetom, neke mogu biti jednostavnije za korištenje, a ponekad putnici jednostavno žele najnoviju aplikaciju. GTFS feed, koji sadrži statičke informacije o javnom prijevozu, sastoji se od nekoliko tekstualnih (txt) datoteka koje su smještene u jednoj ZIP datoteci. Svaka datoteka opisuje određeni aspekt informacija o prijevozu: stajališta, rute, putovanja, cijene karata itd., [3].

### 2.3.2 Statički podaci

Na službenim stranicama ZET-a, Zagrebačkog električnog tramvaja, građanima se pružaju dragocjeni javno dostupni statički podaci koji se konstantno održavaju i ažuriraju. Podaci su od suštinskog značaja za funkcioniranje gradskog prijevoza koji omogućuju građanima brz i jednostavan uvid u ključne informacije. Statički podaci koji se nude na tim stranicama pružaju stabilnu osnovu za planiranje putovanja i praćenje javnog gradskog prijevoza. Primjerice, podaci o koordinatama položaja svih stanica u Zagrebu omogućuju precizno lociranje svake stanice na mreži javnog prijevoza. Ovo je posebno korisno za turiste i nove stanovnike grada koji se žele snaći u nepoznatom okruženju.

Jedna od ključnih komponenti ovih statičkih podataka su informacije o vremenima polazaka javnog gradskog prijevoza. Građani mogu unaprijed saznati točno vrijeme kada će tramvaj ili autobus stići na određenu stanicu, čime se znatno smanjuje čekanje na stajalištima. Ovakve informacije su dragocjene, posebno u uvjetima brzog tempa života, kada je važno maksimalno iskoristiti svoje vrijeme. Prikaz podataka u **tablici 1**, olakšava građanima da brzo i lako prepoznaju ključne informacije. U tablici se nalaze podatci o stanici, njenim koordinatama, rasporedu polazaka i možda čak i informacije o povezanim linijama.

**Tablica 1.** Prikaz i opis statičkih podataka ZET-a.

Naziv	Opis	Sadržaj
agency.txt	Sadrži informacije o agenciji za javni prijevoz koja operira uslugom javnog prijevoza	Ime agencije, web stranicu, vremensku zonu, jezik, telefonski broj i URL za naknadu"
calendar.txt	Sadrži ID usluga za svaku uslugu dostupnu u određenom vremenskom rasponu	"ID usluge, dani u tjednu kada je usluga dostupna, početni i krajnji datumi te eventualne iznimke redovnog rasporeda."
calendar_dates.txt	"Sadrži sve dodane ili uklonjene ID-ove usluga za određene datume."	ID usluge, datum i vrsta iznimke (dodana ili uklonjena)."
feed_info.txt	"Sadrži informacije o samom GTFS feedu, kao što su izdavač i broj verzije."	Naziv izdavača, URL izdavača, jezik, početni i krajnji datumi važenja podataka te broj verzije.
routes.txt	Sadrži informacije o tranzitnim rutama.	Identifikacijski broj rute, prijevoznik koji operira rutu, kratko i dugo ime rute te vrsta prijevoza (npr. autobus, podzemna željeznica).
stops.txt	Sadrži informacije o stanicama javnog prijevoza, poput njihove lokacije i imena.	Stop ID, naziv, lokacija (geografska širina i dužina), i informacija o tome da li je stajalište pristupačno za osobe s invaliditetom.
stop_times.txt	Sadrži informacije o vremenima dolaska i odlaska vozila na svakoj stanici na njegovoj ruti.	Sadrži vrijeme dolaska i odlaska svakog vozila na svakoj stanici na svom putu. Trip ID, ID stanice, vrijeme dolaska i odlaska, redni broj stanice u putu i bilo kakva ograničenja u usluzi prikupljanja ili iskrcavanja putnika.
trips.txt	Sadrži informacije o putovanjima koja se definiraju kao niz od dva ili više zaustavljanja koja se javljaju u određenom redosljedu.	Trip ID, ID rute, ID usluge, naziv (head sign) i smjer putovanja.

### 2.3.3 Realtime podaci

Real-time podaci su iznimno važni u prometnom sektoru kako bi putnici dobili ažurirane informacije o dolascima i odlascima vozila. U slučaju ovog poslužitelja, podaci se ažuriraju svakih 5 sekundi kako bi osigurali preciznost i pouzdanost informacija. Što se tiče real-time podataka od strane ZET-a, mogu se pronaći na URL-u: <https://zet.hr/gtfs-rt-protobuf>. Objavljivanje real-time podataka, kao kod mnogih prijevoznih sustava, donosi brojne prednosti za korisnike i agenciju:

- Poboljšanje korisničkog iskustva: real-time podaci omogućuju točan prikaz dolaska vozila na stanicu, smanjujući vrijeme čekanja i olakšavajući planiranje putovanja.
- Povećana pouzdanost i transparentnost: objavljivanje ovih podataka povećava povjerenje korisnika u sustav i poboljšava odnos s agencijom putem preciznih vremena polazaka.
- Optimizacija putovanja: real-time podaci pomažu korisnicima da se brže kreću gradom, smanjujući gužve na stanicama i omogućujući bolji raspored putnika.
- Pouzdanost informacija: ovi podaci osiguravaju aktualne i točne informacije, omogućujući donošenje informiranih odluka o putovanju.
- Poticanje održivog prijevoza: pristup real-time podacima potiče ljude da koriste javni prijevoz, doprinoseći smanjenju gužvi i emisija CO<sub>2</sub> u gradu.
- Tehnološki napredak: suvremene tehnologije omogućuju brzu distribuciju podataka, čime se postiže napredak u odnosu na tradicionalne rasporede.
- Poticanje inovacija: dostupnost ovih podataka potiče razvoj aplikacija i rješenja koja dodatno poboljšavaju korisničko iskustvo.
- Konkurentska prednost: prijevozne agencije s real-time podacima privlače više korisnika i postaju preferirani izbor u usporedbi s drugim opcijama.

Objavljivanje real-time podataka od strane ZET-a ima za cilj unaprijediti iskustvo korisnika, optimizirati upravljanje prijevozom i potaknuti održivu mobilnost. U **tablici 2** prikazani su i opisani real-time podaci te navedeni primjeri.

**Tablica 2.** Prikaz i opis real-time podataka ZET-a.

Naziv	Opis	Primjer
ID	Identifikacijski broj vozila. Svako vozilo ZET-a ima jedinstveni identifikacijski broj koji se koristi za praćenje i identifikaciju određenog vozila.	XQD7RXR02N
Trip_id	Jedinstveni identifikator putovanja: Svako putovanje (trip) ima svoj jedinstveni identifikator koji se koristi za praćenje i identifikaciju određenog putovanja.	0_1_713_7_11934

Route_id	Broj rute: svaka ruta ima svoj jedinstveni broj koji se koristi za identifikaciju određene rute kojom se putovanje odvija.	7
Stop_id	Identifikacijski broj sljedeće stanice: ovo je identifikacijski broj sljedeće stanice na koju će tramvaj stići. Taj broj se ažurira kako se vozilo približava sljedećoj stanici.	163_4
Stopname	Naziv stanice: ovo je naziv sljedeće stanice na koju će tramvaj stići. Putnicima pomaže prepoznati svoju stanicu ili odredište.	Šubićeva.
Stoplat I stoplon	Koordinate sljedeće stanice: ove koordinate predstavljaju geografsku širinu i dužinu sljedeće stanice. Ažuriraju se kako se vozilo približava stanici.	45,810019, 15,992974
Route_long_name	Puni naziv rute: ovo je puni naziv rute kojom se putovanje odvija. Pruža putnicima dodatne informacije o odabiru pravilne rute.	Savski most-Dubrava
Geometry	Koordinate trenutnog položaja tramvaja (x,y): ove koordinate predstavljaju trenutni položaj tramvaja. Ažuriraju se kako se vozilo kreće po ruti.	15,992974 , 45,810019
stop sequence	Redoslijed stanica u slijedu putovanja: ovaj podatak prikazuje redoslijed stanica u slijedu putovanja. Putnici mogu pratiti napredak putovanja i znati koja je sljedeća stanica.	9
Delay	Kašnjenje (u sekundama) odlaska: ovaj podatak prikazuje kašnjenje tramvaja u odnosu na vremenski raspored.	163_4

Problematika real-time podataka leži u izazovima s kojima se suočavaju korisnici prilikom njihova korištenja. Naime, ovi podaci često dolaze u specifičnom formatu koji nije intuitivan za svakodnevne ljude, što može uzrokovati poteškoće u razumijevanju. Ovaj oblik prezentacije podataka može biti prepreka za običnog čovjeka. Detaljni podaci koji su prikazani na **slici 9** omogućuju putnicima da budu informirani o dolascima i odlascima tramvaja, kao i o trenutnom položaju vozila i mogućim kašnjenjima. Sve te informacije čine putovanje javnim prijevozom učinkovitijim i olakšavaju putnicima planiranje svojih ruta.

id	trip_id	route_id	stop_sequence	arrival_time	departure_time	stop_id	delay	stop_name	stop_lat	stop_lon	route_long_name	geometry
0	XZWINTR00E	0_11_17606_176_10301	176	nan	1682444230.00000	99_32	-3770.00000	Črnomerec	45.81606	15.93333	Zg.(Črn.)-G.Bitra	POINT(15.933327 45.816059)
1	XRWINTR07H	0_11_606_6_12752	6	1682447716.00000	1682447716.00000	185_3	-126.00000	Zaprude	45.77945	16.00173	Črnomerec - Sopot	POINT(16.001734 45.779448)
2	XQWINTR01J	0_11_404_4_11700	4	nan	nan	206_14	0.00000	Dubec	45.82785	16.07770	Savski most - Dubec	POINT(16.0777 45.827847)
3	XVWINTR06V	0_11_15902_159_10108	159	1682447864.00000	1682447864.00000	674_23	14.00000	Buhimi	45.70612	15.91370	S.most-Strm.Odranski	POINT(15.913701 45.706124)
4	XXWINTR0HV	0_11_13603_136_10456	136	1682447896.00000	nan	99_23	nan	Črnomerec	45.81532	15.93323	Črnomerec-Špansko	POINT(15.933231 45.815318)
5	XXWINTR0HW	0_11_13603_136_10456	136	nan	nan	99_32	0.00000	Črnomerec	45.81606	15.93333	Črnomerec-Špansko	POINT(15.93327 45.816059)
6	XRWINTR0L6	0_11_715_7_13212	7	1682447546.00000	1682447546.00000	231_2	-298.00000	Jordanovac	45.81761	16.00757	Savski most-Dubrava	POINT(16.007574 45.817612)
7	XRWINTR0TF	0_11_904_9_12011	9	nan	1682447466.00000	245_1	-486.00000	Ljubljanića	45.79728	15.93856	Ljubljanića-Boronjaj	POINT(16.0098564 45.797279)
8	XUWINTR0GO	0_11_10911_109_10638	109	nan	1682447835.00000	568_22	75.00000	Kaularićev prilaz	45.76498	16.00233	Črnomerec - Dugave	POINT(16.002334 45.764875)
9	X4XINTR04E	0_11_29501_295_10073	295	nan	1682446880.00000	184_31	-2320.00000	Zaprude	45.78169	16.00317	Zapr.-Jakušev.-Zapr.	POINT(16.003167 45.781688)
10	X1XINTR0PV	0_11_22901_229_10061	229	nan	nan	110_41	0.00000	Glavni kolodvor	45.80386	15.97904	Gl.kolodvor-M.Mlaka	POINT(15.979041 45.803857)
11	X4XINTR01L	0_11_28301_283_10064	283	1682447804.00000	1682447804.00000	1027_22	64.00000	Bistrička-Zagrebačka	45.82781	16.10555	Dubec-Brest.-Sesvete	POINT(16.105547 45.827811)
12	XRWINTR0AH	0_11_611_6_12727	6	1682446991.00000	1682446991.00000	112_1	-885.00000	Branim. tržnica	45.80614	15.99198	Črnomerec - Sopot	POINT(15.99198 45.80614)
13	XRWINTR0AI	0_11_611_6_12802	6	nan	nan	179_1	0.00000	Sopot	45.77770	15.98668	Črnomerec - Sopot	POINT(15.986683 45.777699)
14	XRWINTR0GZ	0_11_707_7_13195	7	nan	nan	226_1	0.00000	Arena Zagreb	45.77718	15.95388	Savski most-Dubrava	POINT(15.953875 45.777177)
15	XUWINTR02F	0_11_10202_102_10135	102	1682447797.00000	1682447797.00000	401_23	-83.00000	Tuškanac 78	45.83138	15.97143	Brit.trg-Mihaljevac	POINT(15.971427 45.831377)
16	XUWINTR02G	0_11_10202_102_10138	102	nan	nan	176_28	0.00000	Mihaljevac	45.84161	15.97519	Brit.trg-Mihaljevac	POINT(15.975189 45.841614)
17	XSWINTR0K3	0_11_1213_12_11896	12	1682447585.00000	1682447585.00000	275_2	-293.00000	Selka	45.79772	15.94228	Ljubljanića - Dubrava	POINT(15.942279 45.79772)
18	XQWINTR06F	0_11_203_2_12307	2	1682447751.00000	1682447751.00000	109_1	-182.00000	Glavni kolodvor	45.80521	15.97928	Črnomerec - Savišće	POINT(15.979278 45.805213)
19	XQWINTR06G	0_11_203_2_12328	2	nan	nan	266_3	0.00000	Savišće	45.78940	16.03492	Črnomerec - Savišće	POINT(16.034918 45.7894)
20	XTWINTR02I	0_11_1712_17_12557	17	1682447670.00000	1682447670.00000	106_1	-19.00000	Tra J. Jelčića	45.81291	15.97731	Prečko - Boronjaj	POINT(15.977312 45.812906)

Slika 9. Prikaz real time podataka ZET-a

### 2.3.4 Baze podataka

Baze podataka predstavljaju organizirane skupove povezanih podataka koji se pohranjuju na strukturiran način kako bi omogućili efikasno i pouzdano upravljanje informacijama. Za upravljanje ovim bazama podataka koristimo posebne softverske sustave poznate kao sustavi za upravljanje bazama podataka (eng. DataBase Management System, DBMS), [11].

DBMS je programski alat koji omogućuje da učinkovitu manipulaciju, pohranjivanje i izvlačenje podatka iz baze podataka. On djeluje kao posrednik između korisnika i samih podataka, omogućujući da se koriste podaci bez potrebe za poznavanjem detalja o njihovom fizičkom pohranjivanju, [11].

Sam sustav za upravljanje bazama podataka ima tri ključna sloja:

1. Vanjska razina: ovaj sloj predstavlja sučelje između korisnika i baze podataka. On omogućuje korisnicima da definira način na koji žele vidjeti podatke, kako bi imali jednostavan pristup informacijama koje ih zanimaju. Korisnici mogu koristiti upite i izvještaje za pregledavanje, uređivanje i upravljanje podacima na način koji je za njih najprikladniji.
2. Logička razina: ovaj sloj određuje međuovisnosti između vanjske i unutarnje razine baze podataka. On mapira vanjske sheme i zahtjeve korisnika u logičku strukturu baze podataka koja koristi relacije, tablice i ključeve. Ovdje se definiraju entiteti, njihovi atributi i veze među njima. Logička razina osigurava dosljednost i integritet podataka te olakšava rad s bazom podataka na apstraktnoj razini.

3. Fizička razina: ovaj sloj prikazuje kako DBMS percipira i upravlja stvarnim podacima na disku. On se bavi organizacijom podataka na fizičkoj razini, uključujući načine pohrane, pristupa i indeksiranja. Cilj je optimizirati performanse baze podataka i minimizirati vrijeme pristupa podacima, [11].

Logička razina baze podataka definirana je pomoću različitih modela podataka. Modeli podataka su formalni sustavi koji opisuju strukturu baze podataka, uključujući tipove podataka, odnose među njima i ograničenja koja vrijede u bazi. Postoji nekoliko vrsta modela podataka, uključujući hijerarhijski model podataka, mrežni model podataka, objektni model podataka, relacijski model podataka, objektno-relacijski model podataka i dimenzijski model podataka, [11].

Svaki od ovih modela ima svoje prednosti i primjene, a odabir odgovarajućeg modela ovisi o potrebama i zahtjevima projekta. Kroz adekvatno upravljanje bazama podataka i primjenu odgovarajućih modela, osiguravamo efikasno i pouzdano rukovanje informacijama, što je ključno za uspješno vođenje modernih sustava i aplikacija, [11].

#### 2.3.4.1 Relacijski model

Relacijski model podataka je konceptualni model koji se koristi u relacijskim bazama podataka, a za upravljanje takvim bazama koristimo sustave poznate kao RDBMS (eng. Relational DataBase Management System). Ovi sustavi koriste SQL (eng. Structured Query Language) jezik za izvođenje upita i ažuriranje podataka u bazama, [11].

Dvanaest Coddovih pravila, definiranih od strane Edgara F. Codd-a 1970. godine, opisuju ključne karakteristike relacijskih baza podataka. Nulto pravilo naglašava da sustav koji tvrdimo da je relacijski mora upravljati bazom podataka samo na relacijski način. Ostala pravila, od 1. do 12., definiraju aspekte kao što su predstavljanje podataka, pravila pristupa, rukovanje pogledima, visok nivo unosa, ažuriranja i brisanja, nezavisnost fizičkih i logičkih podataka, te očuvanje integriteta podataka, [11].

Današnji komercijalni sustavi za upravljanje bazama podataka koriste SQL jezik, a iako ne ispunjavaju sva Coddova pravila, moraju zadovoljavati najmanje dva od njih. Tako komercijalne relacijske baze podataka prikazuju podatke korisnicima kao relacije (tablice) i koriste relacijske operatore za upravljanje podacima, [11]. Relacijski model podataka temelji se na konceptu relacija, koje su u biti cijele tablice. Svaka relacija sadrži stupce koji predstavljaju attribute s definiranim tipovima podataka (domenama). Redci u relacijama predstavljaju n-torke koje sadrže vrijednosti za svaki atribut. Kroz relacijski model podataka, osigurava se organizacija, dosljednost i integritet podataka u bazama, što čini ovaj model vrlo popularnim i široko korištenim u suvremenom informacijskom sustavu, [11].

Primarni ključ je neki atribut ili skup atributa koji na jedinstveni način identificiraju svaki redak u tablici. Njega opisuje drugo Coddovo pravilo koje opisuje da se do svakog zapisanog podatka može doći pomoću primarnog ključa. Svaki primarni ključ u tablici mora zadovoljavati tri uvjeta. Ti osnovni uvjeti su: jedinstvenost, minimalnost, pravilo integriteta primarnog ključa, [11].

Za analizu i pregledavanje podataka, ključni alat su SQL upiti. Na **slici 10** prikazan je jedan takav upit koji ima zadatak izvlačenja važnih informacija iz tablice **ZET\_PODACI**. Evo detaljnog razjašnjenja koraka po korak:

```
SELECT DISTINCT CAST(route_id AS INT) AS route_id
FROM ZET_PODACI where time_input >'2023-07-14 16:00:00 'and time_input<'2023-07-14
19:00:00' order by route_id asc
```

**Slika 10.** Primjer Sql upita.

- **SELECT DISTINCT**, opis: ovom naredbom bazi podataka se nalaže da izabere samo jedinstvene vrijednosti iz stupca koji je naveden u idućem dijelu upita.
- **CAST(route\_id AS INT) AS route\_id**, opis: ovdje se koristi CAST funkcija kako bi se **route\_id** pretvorio u cjelobrojnu (int) vrijednost. Ova transformacija može biti korisna ako je **route\_id** pohranjen kao neki drugi tip podataka, poput niza znakova. Dodatak **AS route\_id** dodjeljuje novo ime stupcu nakon transformacije.
- **FROM ZET\_PODACI**, opis: u ovoj dijelu navodi se ime tablice iz koje želimo dohvatiti podatke, u ovom slučaju to je ZET\_PODACI.
- **WHERE time\_input > '2023-07-14 16:00:00' AND time\_input < '2023-07-14 19:00:00'**, opis: ova naredba postavlja uvjet koji filtrira retke iz tablice prema vremenskom rasponu. Samo će redci čiji je **time\_input** (vrijeme unosa) između 16:00:00 i 19:00:00, 14. srpnja 2023. godine, biti uzeti u obzir.
- **ORDER BY route\_id ASC**, opis: ova naredba zahtijeva da upit sortira rezultate prema **route\_id** stupcu uzlazno. To znači da će rezultati biti prikazani od najmanjeg prema najvećem **route\_id**.

Kako bi se kreirala tablica za analizu nekih podataka potrebno je stvoriti novu bazu podataka, a to se postiže na način prikazan u ovom kodu:

```
CREATE DATABASE GTFS
```

-Nakon toga potrebno je stvoriti novu tablicu za spremanje određenih podataka. Podaci moraju imati svoj naziv i tip podataka kao što je prikazano u sljedećem kodu:

```
create table osobe(
    id VARCHAR(255) primary key,
    ime nvarchar (50),
    broj_osobe int not null identity(1,1),
)
```

Kreirana je tablica „osobe “ koja se sastoji od id, imena, rednog broja osobe. Stupac „id“ je definiran kao tip podatka **VARCAHR** gdje svaki znak zauzima jedan bajt. To znači da SE MOŽE pohraniti nizove različitih duljina, ali neće nužno podržavati sve jezične znakove i simbole (č,ć,š). Ujedno je i primarni ključ. Stupac „ime “ je tip podatka **NVARCHAR**. Ovaj tip podataka se koristi za pohranu varijabilnih nizova znakova, ali s razlikom da svaki znak zauzima dva bajta. To omogućuje da **NVARCHAR** podržava širi raspon jezičnih znakova i simbola, uključujući različite jezike i pisma. Stupac „broj\_osobe“ je tip podatka **INT**. U ovom slučaju, **INT**



označava cjelobrojni tip podataka. **IDENTITY(1,1)** je naredba koja govori bazi podataka da automatski generira vrijednosti za stupac. Prva "1" označava početnu vrijednost odnosno od koje vrijednosti će početi generiranje. Druga "1" označava korak generiranja, odnosno koliko se vrijednost povećava za svaki novi red.

#### 2.3.4.2 GTFS baza podataka

Na slici 11. je prikazana stvorena tablica **ZET\_PODACI** u bazi podataka **GTFS\_ZET**. Tablica je namijenjena spremanju podataka koje se skidaju s ZET servera putem Python koda. Tablica **ZET\_PODACI** ima različite stupce koji služe za pohranu važnih informacija o putovanjima vozila javnog prijevoza unutar ZET-a. Stupci sadrže podatke kao što su identifikatori putovanja, identifikatori ruta, redoslijed stajališta, vremena dolaska i odlaska, identifikatori stajališta, kašnjenje vozila, nazivi stajališta, geografske koordinate stajališta te vremenski zapis o unosu podataka.

Ova tablica omogućuje analizu i praćenje kretanja vozila, njihovih dolazaka i odlazaka na stajalištima te pruža informacije o kašnjenjima. Pohranjeni podaci omogućuju stvaranje relacija između različitih elemenata putovanja, kao što su rute, stajališta, vremena dolaska i odlaska itd. Omogućuje razumijevanje funkcioniranja javnog prijevoza i pruža korisne informacije korisnicima aplikacije. Tablica **ZET\_PODACI** je ključan dio cjelokupne aplikacije koja se bavi skidanjem, obradom i vizualizacijom podataka javnog prijevoza. Sustav koji koristi ovu bazu podataka omogućuje korisnicima praćenje i planiranje putovanja, prikazivanje informacija o stajalištima, ruta i kašnjenjima te pruža jasnu sliku o funkcioniranju sustava javnog prijevoza u realnom vremenu.

```
create table ZET_PODACI(  
    id VARCHAR(255),  
    trip_id VARCHAR(255),  
    route_id INT,  
    stop_sequence INT,  
    arrival_time VARCHAR(255),  
    departure_time VARCHAR(255),  
    stop_id VARCHAR(255),  
    delay_zet INT,  
    stop_name VARCHAR(255),  
    stop_lat FLOAT ,  
    stop_lon FLOAT default 0,  
    route_long_name VARCHAR(255) default 0,  
    geometry_x FLOAT default 0,  
    geometry_y FLOAT default 0 ,  
  
    time_input datetime default getdate()  
)
```

**Slika 11.** Prikaz koda za stvaranje tablice ZET\_PODACI

- Stupac: id (VARCHAR(255)), opis: ovaj stupac predstavlja identifikator (ID) za zapise u tablici. Koristi se za jedinstveno označavanje svakog retka podataka.
- Stupac: trip\_id (VARCHAR(255)), opis: ovaj stupac sadrži identifikator (ID) putovanja (trip) za koje se podaci odnose. Putovanje je pojedinačna vožnja vozila koje pripada određenoj ruti.

- Stupac: route\_id (INT), opis: ovaj stupac sadrži cjelobrojnu vrijednost koja predstavlja identifikator (ID) rute za koju je vezano putovanje. Svaka ruta ima jedinstveni broj koji je pohranjen u ovom stupcu.
- Stupac: stop\_sequence (INT), opis: ovaj stupac sadrži cijeli broj koji predstavlja redni broj stajališta u okviru putovanja. Redni brojevi stajališta određuju redosljed stajanja vozila tijekom putovanja.
- Stupac: arrival\_time (VARCHAR(255)), opis: ovaj stupac sadrži vremenski zapis dolaska vozila na određeno stajalište tijekom putovanja.
- Stupac: departure\_time (VARCHAR(255)), opis: ovaj stupac sadrži vremenski zapis polaska vozila s određenog stajališta tijekom putovanja.
- Stupac: stop\_id (VARCHAR(255)), opis: ovaj stupac sadrži identifikator (ID) stajališta na kojem se vozilo zaustavlja tijekom putovanja.
- Stupac: delay\_zet (INT), opis: ovaj stupac sadrži cjelobrojnu vrijednost koja predstavlja kašnjenje (u sekundama) vozila na određenom stajalištu.
- Stupac: stop\_name (VARCHAR(255)), opis: ovaj stupac sadrži naziv stajališta na kojem se vozilo zaustavlja tijekom putovanja.
- Stupac: stop\_lat (FLOAT), opis: ovaj stupac sadrži decimalni broj koji predstavlja geografsku širinu (latitude) stajališta.
- Stupac: stop\_lon (FLOAT, default 0), opis: ovaj stupac sadrži decimalni broj koji predstavlja geografsku dužinu (longitude) stajališta. Ako za određeno stajalište nije dostupna informacija o geografskoj dužini, tada će vrijednost ovog stupca biti postavljena na 0.
- Stupac: route\_long\_name (VARCHAR(255), default 0), opis: ovaj stupac sadrži naziv rute kojoj pripada putovanje. Ako za određenu rutu nije dostupna informacija o nazivu, tada će vrijednost ovog stupca biti postavljena na 0.
- Stupac: geometry\_x (FLOAT, default 0), opis: ovaj stupac sadrži decimalni broj koji predstavlja x-koordinatu točke stajališta na karti. Ova koordinata se koristi za vizualizaciju stajališta na karti. Ako za određeno stajalište nije dostupna informacija o x-koordinati, tada će vrijednost ovog stupca biti postavljena na 0.
- Stupac: geometry\_y (FLOAT, default 0), opis: ovaj stupac sadrži decimalni broj koji predstavlja y-koordinatu točke stajališta na karti. Ova koordinata se koristi za vizualizaciju stajališta na karti. Ako za određeno stajalište nije dostupna informacija o y-koordinati, tada će vrijednost ovog stupca biti postavljena na 0.
- Stupac: time\_input (DATETIME, default GETDATE()), opis: ovaj stupac sadrži vremenski zapis koji označava vrijeme unosa podataka u tablicu. Podrazumijevano, ova vrijednost će biti postavljena na trenutno vrijeme kada se unesu novi podaci.

Tablica **ZET\_PODACI** služi za pohranu podataka o putovanjima vozila javnog prijevoza (tramvaji, autobusi) u okviru ZET-a. Podaci u tablici omogućuju analizu i praćenje kretanja vozila, njihovih dolazaka i odlazaka na stajalištima te pružaju informacije o kašnjenjima. Stupci s koordinatama (stop\_lat, stop\_lon, geometry\_x, geometry\_y) koriste se za prikazivanje stajališta na karti pomoću geografske točke (latitude i longitude) i kartografske točke (x-koordinata i y-koordinata). Uz sve navedene podatke, tablica također bilježi vrijeme unosa novih podataka u stupac "time\_input", što pruža informaciju o vremenu.

### 3 INFORMIRANJE PUTNIKA I VOZAČA

Jedan od područja koje obuhvaća ITS je skupina usluga putnih informacija namijenjenih putnicima i vozačima. Ova skupina usluga uključuje niz usluga koje pružaju informacije prije putovanja i tijekom putovanja, obavještanja u javnom prijevozu, navigaciju za osobna vozila do odredišta i slično. Ove informacije omogućuju određivanje optimalne rute, načina prijevoza, vođenje do odredišta i druge relevantne informacije. Usluge se ostvaruju putem posebnih sustava ili integriranih sustava koji podržavaju više povezanih usluga, [10].

Skupina usluga putnih informacija ITS-a obuhvaća sljedeće temeljne usluge:

1. Predputne informacije
2. Putne informacije vozaču i putniku
3. Putne informacije u javnom prijevozu
4. Osobne informacijske usluge
5. Izbor rute i navigacija

Usluge pružaju korisnicima potrebne informacije kako bi imale bolji uvid u svoje putovanje, omogućavajući im da donose informirane odluke o ruti, načinu prijevoza i drugim aspektima putovanja, [10].

#### 3.1 Predputno informiranje

U funkcionalnom području informiranja putnika, prva usluga koja se pruža je predputno informiranje. Usluga može biti samostalni komercijalni paket ili integrirana s drugim uslugama u odgovarajućem paketu na tržištu. Sustav ima svrhu pružiti korisnicima informacije prije početka putovanja kako bi im olakšao donošenje odluka o različitim aspektima putovanja, kao što su način putovanja, moda, ruta, vrijeme polaska i slično, [10].

Informacije koje se pružaju mogu se odnositi na različite aspekte putovanja, uključujući planiranje putovanja javnim prijevozom, stanje na cestama, vremenske nepogode, dostupnost parkirnih mjesta, vozne redove u željezničkom, zračnom i vodnom prometu, ugostiteljske i turističke sadržaje te korisne informacije vezane uz putovanje, [10].

Predputne informacije dostupne su putem različitih medija i telekomunikacijskih terminalnih uređaja, kao što su žični/fiksni telefoni, radio, telefaks, RDS/TMC, računala povezana na internet, mobilni uređaji te javni interaktivni kiosci. Na taj način korisnici mogu pristupiti potrebnim informacijama na različite načine, ovisno o svojim preferencijama i raspoloživim tehnološkim sredstvima, [10].

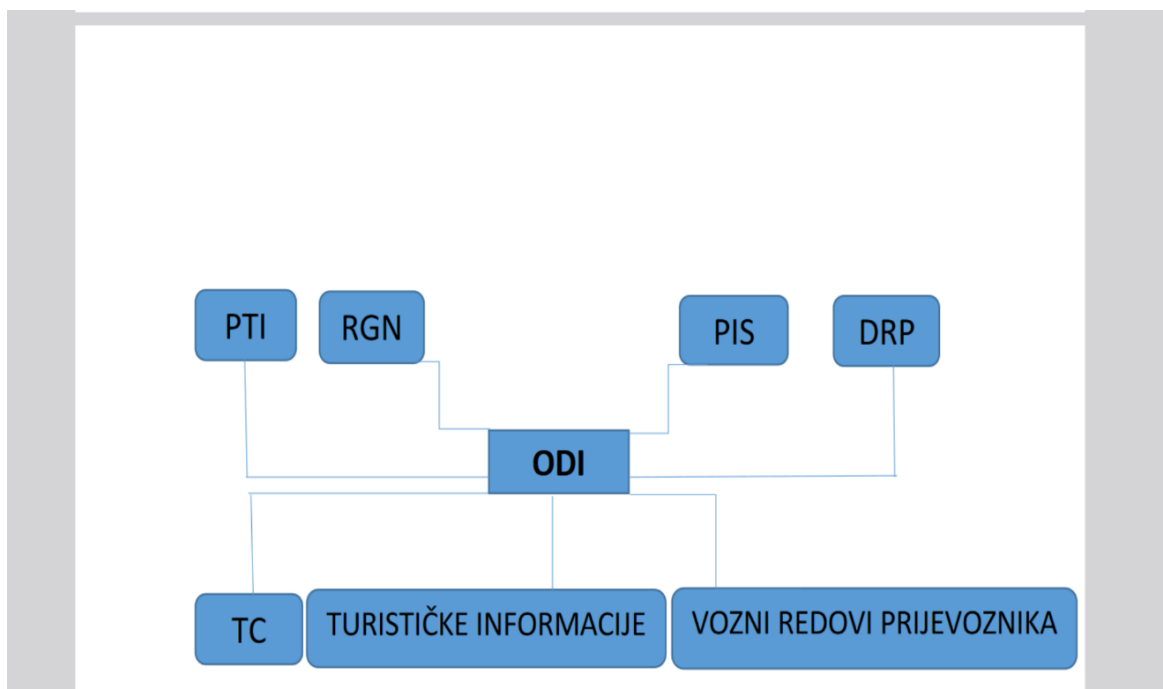
#### 3.2 Putno informiranje

Glavna svrha usluge putnog informiranja je pružiti vozaču informacije o prometnim uvjetima prije i tijekom putovanja. Ove informacije omogućuju vozaču, kao i putnicima u vozilu, da donesu bolje odluke u vezi s rutom putovanja ili promjenom prijevoznog sredstva, [10].

Putne informacije vozaču obuhvaćaju sljedeće aspekte:

- Uvjete na prometnicama, kao što su gužve, zastoje ili radovi na cestama.
- Nesreće na cesti koje mogu utjecati na tijek prometa.
- Događaje koji mogu imati utjecaj na promet, poput sportskih ili kulturnih manifestacija.
- Atraktivna turistička ili zabavna događanja koja se odvijaju u blizini, [10].

Informacije pružaju vozačima vrijedan uvid u trenutačno stanje prometa i omogućuju im da planiraju svoje putovanje na temelju te informacije. Tako vozači mogu izbjeći zastoje, odabrati optimalnu rutu ili čak odlučiti koristiti drugo prijevozno sredstvo ako su uvjeti nepovoljni. **Slika 12** prikazuje odnos putnog informiranja s drugim sustavima, [10].



**Slika 12.** Integracija ODI (putno informiranje vozača) s drugim sustavima, [10].

### 3.3 Informiranje o javnom prijevozu

Usluga putnih informacija o javnom prijevozu u sklopu TS-a (transportnog sustava) spada u kategoriju putnih informacija i ostvaruje se kao samostalan sustav. Glavna svrha ove usluge je potaknuti veće korištenje javnog gradskog prijevoza pružanjem boljih informacija korisnicima tih usluga. Informacije o vozilima i uslugama javnih prijevoznika dostupne su korisnicima u njihovim domovima, radnim mjestima, na ulicama, kolodvorima i drugim mjestima. Komunikacija se može ostvariti putem telefonskog poziva prema odgovarajućem centru, osobnog računala i interneta ili putem javnih interaktivnih terminala dostupnih na javnim mjestima, [10].

Nedostatak odgovarajućih putnih informacija za putnike može značajno otežati korištenje javnog prijevoza, posebno za posjetitelje i turiste. Neke od glavnih poteškoća su:

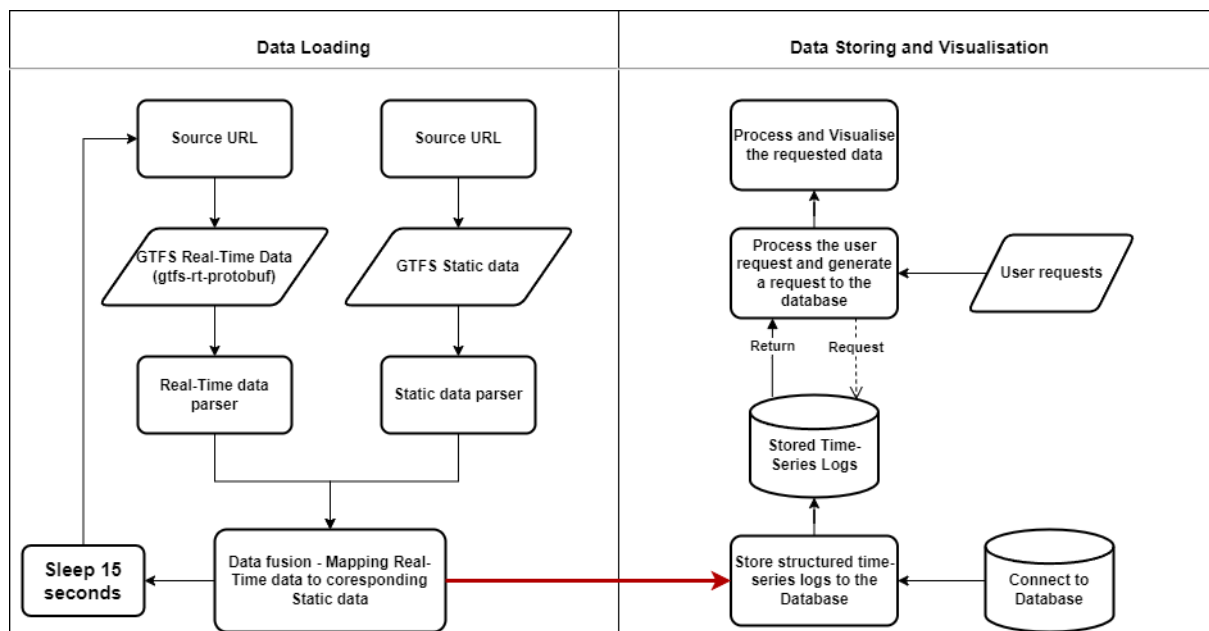
- Odabir prikladne linije javnog prijevoza.
- Gdje i kako platiti uslugu.
- Vrijeme čekanja na prijevozno sredstvo.

Istraživanja su pokazala da je razina psihološkog stresa smanjena kada putnik ima informaciju o očekivanom vremenu čekanja. Na primjer, kada putnik zna da će čekati 10 minuta na autobus, to smanjuje napetost i nelagodu u usporedbi sa situacijom kada putnik čeka 4 minute, a ne zna kada će autobus stići. [10].

## 4 OPIS PODATAKA I STANDARDA PRUŽATELJA JAVNOG GRADSKOG PRIJEVOZA

Skidanje, obrada i vizualizacija podataka je kompleksan problem koji zahtijeva upotrebu tri različita programska jezika: Python, SQL i C#. Pri skidanju podataka korišten je programski jezik Python, koji se pokazao iznimno korisnim za obradu velikih količina podataka u obliku GTFS-RT (eng. General Transit Feed Specification - Real Time). Python je omogućio jednostavno parsiranje podataka i pretvaranje protobuf formata u oblik rječnika. Također korištene **Pandas** i **GeoPandas** biblioteke za rad s tabličnim i geoprostornim podacima. Za preuzimanje podataka s interneta korištena je biblioteka requests. Nakon skidanja i obrade podataka, korišten je SQL za pohranu tih podataka u bazu podataka.

Aplikacija pruža korisnicima konkretne i aktualne podatke o putovanjima tramvaja i autobusa. Uz pomoć geografskih podataka, korisnici mogu pratiti putanje vozila u stvarnom vremenu. Podaci o dolascima i odlascima na stajalištima osiguravaju točnost i pouzdanost informacija, što je ključno za korisnike koji se oslanjaju na aplikaciju za svoje svakodnevne aktivnosti. Na **slici 13.** prikazan je dijagram toka cijelog sustava. Kroz ovaj dijagram, korisnici mogu jasno vidjeti kako skidanje podataka, njihova obrada i vizualizacija funkcioniraju u sustavu kako bi pružili aktualne i pouzdane informacije korisnicima.



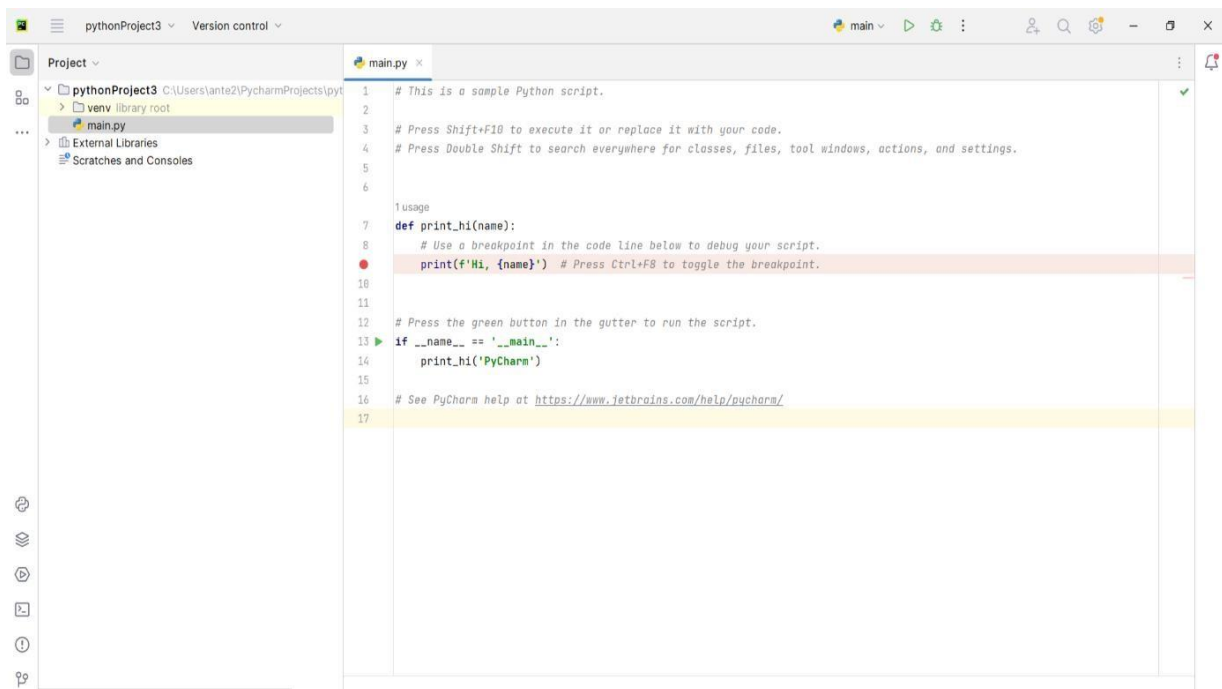
**Slika 13.** Prikaz dijagrama toka sustava.

Svojim višeslojnim pristupom, pouzdanošću i intuitivnim korisničkim sučeljem, aplikacija pruža potrebne informacije korisnicima kako bi se olakšalo i poboljšalo njihovo putovanje i kretanje kroz grad. Dijagram toka sustava pruža korisnicima vizualni prikaz kako

cijeli sustav funkcionira, što doprinosi boljoj interakciji i razumijevanju aplikacije. Sve navedeno osigurava da korisnici mogu osloniti na razvijenu aplikaciju kako bi im olakšala svakodnevno putovanje i kretanje gradom.

#### 4.1 Python skripta za skidanje podataka

Uspjeh ideje o aplikaciji ovisi o učinkovitom strukturiranju i skidanju podataka s ZET servera. Za rješavanje ovog problema, korišten je programski jezik Python i platformu PyCharm, koja omogućava lakše i efikasnije razvijanje aplikacije. Implementirana je skripta koja automatizira proces skidanja podataka sa servera, te je također korištena je za pohranu statičkih podataka koji su potrebni za rad aplikacije. Na **slici 14** prikazano je sučelje PyCharm platforme koje je korišteno za razvoj i uređivanje koda. PyCharm je moćan i intuitivan razvojni alat koji pomaže u produktivnom radu. Omogućuje brzo testiranje i ispravljanje koda, te pruža mogućnost lako pratiti i upravljati verzijama projekta. Prilikom procesa skidanja podataka, potrebno je bilo instalirati određene pakete i biblioteke koje su ugrađene u Python. Ovi paketi su ključni za učinkovitu obradu i organizaciju podataka s ZET servera.



Slika 14. Prikaz sučelja platforme PyCharm.

Popis korištenih paketa:

1. **google.transit.gtfs\_realtime\_pb2**: ovaj paket omogućuje parsiranje GTFS-RT protobuf podataka s ZET servera. GTFS-RT (eng. General Transit Feed Specification - Real Time) je format za razmjenu u stvarnom vremenu koji pruža ažurirane informacije o javnom prijevozu.
2. **protobuf\_to\_dict**: ova biblioteka omogućuje pretvaranje protobuf podataka u format rječnika, što olakšava daljnju obradu i manipulaciju podacima.

3. **flatten\_json**: ovaj paket omogućuje "ravnjanje" ili pretvaranje složenih JSON objekata u jednostavne rječnike, što olakšava organizaciju podataka i izvlačenje relevantnih informacija.
4. **requests**: ova biblioteka omogućuje izvršavanje HTTP zahtjeva kako bi se pristupilo ZET serveru i preuzeli podatke.
5. **pandas**: Pandas je biblioteka koja omogućuje rad s tabličnim podacima. Korištena je za organizaciju i obradu preuzetih podataka u obliku DataFrame-a.
6. **geopandas**: Geopandas je biblioteka za rad s geoprostornim podacima. Omogućuje stvaranje geografskih točaka i rad s geometrijom. Kako bi se vizualizirali podaci na karti.
7. **time**: Time omogućuje kontrolu vremenskih intervala, što je važno za periodično skidanje podataka sa servera.
8. **pyodbc**: Pyodbc omogućuje povezivanje i interakciju s Microsoft SQL Server bazom podataka, što se koristi za spremanje preuzetih podataka.

Svim ovim paketima i bibliotekama zajedno sa Pythonom i PyCharm platformom omogućeno je automatizirano i efikasno skidanje, obradu i organizaciju podataka sa ZET servera. Na **slici 15** kod predstavlja Python funkciju **parse\_zet\_feed\_data(url)** koja obavlja postupak preuzimanja, parsiranja i obrade GTFS-RT podataka s određenog URL-a pomoću Python paketa i biblioteka. Funkcija prvo šalje zahtjev na navedeni URL putem **requests** biblioteke kako bi preuzela GTFS-RT podatke s ZET servera. Nakon preuzimanja, koristi **gtfs\_realtime\_pb2** biblioteku za parsiranje podataka i pretvaranje ih u strukturirani objekt **feed**, koji sadrži ažurirane informacije o javnom prijevozu.

Sljedeći korak je pretvorba objekta **feed** u običan Python rječnik pomoću **protobuf\_to\_dict** biblioteke. To omogućuje lakšu obradu podataka i pristup pojedinačnim atributima, što je ključno za daljnje organiziranje i analizu podataka. Nakon što su podaci pretvoreni u rječnik, funkcija koristi **Pandas** biblioteku kako bi učitale statički podatci iz CSV datoteka i povezala ih s relevantnim podacima iz GTFS-RT. Ovo omogućuje da se dobiju sve potrebne informacije o linijama, stajalištima i vremenima dolaska i odlaska.

Funkcija stvara **GeoDataFrame** pomoću **geopandas** biblioteke kako bi stvorila geografske točke na temelju dobivenih lokacija stajališta. Nakon obrade i organiziranja podataka, funkcija koristi **pyodbc** biblioteku kako bi uspostavila vezu s bazom podataka i spremila relevantne podatke u SQL Server bazu podataka. To omogućuje da aplikacija ima konkretnu bazu podataka s aktualnim podacima o javnom prijevozu. Cijeli ovaj postupak se periodično izvršava u petlji (while petlja svakih 15 s) koja omogućuje redovito osvježavanje podataka sa ZET servera i ažuriranje baze podataka kako bi korisnici mogli uvijek imati pristup aktualnim informacijama o javnom prijevozu u realnom vremenu.



```

def parse_zet_feed_data(url):
    # Preuzimanje GTFS-RT protobuf podatka s URL-a
    response = requests.get(url, allow_redirects=True)

    # Parsiranje protobuf podatka pomoću gtfs_realtime_pb2 biblioteke
    feed = gtfs_realtime_pb2.FeedMessage()
    feed.ParseFromString(response.content)

    zet_dict = protobuf_to_dict(feed)

    # Učitaj statičke podatke (e.g. stop name, location, route name, etc.)
    absolute_path = os.path.dirname(__file__)
    relative_path = 'data'
    static_data_path = os.path.join(absolute_path, relative_path)
    stops = pd.read_csv(static_data_path + '\\stops.txt', sep=',')
    routes = pd.read_csv(static_data_path + '\\routes.txt', sep=',')

    # Pravljenje zet okvira podataka
    zet_df = pd.DataFrame(flatten(record, '.') for record in zet_dict['entity'])

    # Nazivi stupaca
    zet_df = zet_df[['id', 'trip_update.trip.trip_id', 'trip_update.trip.route_id',
                    'trip_update.stop_time_update.0.stop_sequence',
                    'trip_update.stop_time_update.0.arrival.time',
                    'trip_update.stop_time_update.0.departure.time',
                    'trip_update.stop_time_update.0.stop_id',
                    'trip_update.stop_time_update.0.departure.delay']]

    # Formiranje naziva stupaca okvira podataka

```

**Slika 15.** Prikaz koda za skidanje i pretvaranje podataka u riječnik.

Na slici 16 prikazano je formiranje naziva stupaca u okviru podataka **zet\_df**. To je postignuto metodom **set\_axis**, kojom se dodjeljuju odgovarajući nazivi stupcima. Svaki naziv stupca predstavlja određenu informaciju o podacima koja se želi zadržati. Nakon toga, povezani su statički podaci o stajalištima s glavnim okvirom podataka **zet\_df**. Statički podaci, kao što su naziv stajališta, geografska širina i dužina, bili su pohranjeni u tablici **stops**. Metoda **merge** spaja statičke podatke s odgovarajućim redovima u okviru podataka **zet\_df**, temeljem identifikatora stajališta (**stop\_id**). Promijenjena je vrijednosti stupca **route\_id** u cjelobrojne vrijednosti (integer) metodom **astype**. Analogno povezivanje stajališta, izvršeno je i povezivanje podataka o rutama s glavnim okvirom **zet\_df**. Statički podaci o rutama, poput naziva ruta, nalazili su se u tablici **routes**. Ponovnim korištenjem metode **merge** postiže se spajanje podataka s odgovarajućim redovima u okviru podataka **zet\_df**, temeljem identifikatora ruta (**route\_id**). Kako bi se mogla vizualizirati stajališta na karti, korištena je biblioteka **geopandas**. S njezinom pomoći, stvorene se geografske točke za svako stajalište u okviru podataka **zet\_df**, koristeći zemljopisne širine (**stop\_lat**) i dužine (**stop\_lon**).

```

# Formiranje naziva stupaca okvira podataka
zet_df = zet_df.set_axis(
    ['id', 'trip_id', 'route_id', 'stop_sequence', 'arrival_time', 'departure_time', 'stop_id', 'delay'],
    axis=1)

# Dodaj stacičke podatke u glavni okvir podataka feeda "u stvarnom vremenu" (za svaku rutu pronađite njene stacičke podatke)
zet_df = zet_df.merge(stops[['stop_id', 'stop_name', 'stop_lat', 'stop_lon']],
    left_on='stop_id',
    right_on='stop_id',
    how='left')

zet_df = zet_df.astype({"route_id": int})
zet_df = zet_df.merge(routes[['route_id', 'route_long_name']],
    left_on='route_id',
    right_on='route_id',
    how='left')

# Primjer: Route_ID -> 6; Route_Long_Name -> Crnomerec -Sopot

# Stvorite geo-točku iz zemljopisne dužine i širine
zet_df = geopandas.GeoDataFrame(zet_df, geometry=geopandas.points_from_xy(zet_df.stop_lon, zet_df.stop_lat))

```

**Slika 16.** Prikaz koda za formiranje okvira podataka i povezivanja sa statičkim podacima.

Na slici 17 prikazan je dio koda: prvo se formira okvir podataka **veh\_route\_info** koji sadrži jedinstvene brojeve ruta i njihove nazive iz okvira podataka **zet\_df**. To se radi pomoću metode **drop\_duplicates()**, koja uklanja duplicirane redove na temelju stupaca **route\_id** i **route\_long\_name**. Nakon toga, provjeravanjem svakog broja rute u okviru **veh\_route\_info** određuje se je li linija tramvajska ili autobusna. Ako je broj rute manji od 100, klasificira se kao tramvajska linija, inače kao autobusna. To je postignuto pomoću metode **loc** kojom se pristupa recima koji zadovoljavaju određeni uvjet, a zatim se koristi metoda **astype** kako bi se broj rute pretvorio u niz (string). Pomoću ovih operacija dodaje se novi stupac **label** u okvir podataka **veh\_route\_info**.

Sortiranjem rute vozila (tramvaje i autobuse) prema njihovim brojevima, pohranjuje se rezultat u rječnik **veh\_route\_info**. Svaki zapis u ovom rječniku sadrži jedinstveni broj vozila (vrijednost) i odgovarajući opis (naziv linije) koji se prethodno stvara. Jedinstveni brojevi vozila pohranjuju se u listu **veh\_route\_numbers**, što olakšava daljnje korištenje tih brojeva u aplikaciji.

```

# Izvoz jedinstvenih brojeva ruta i naziva

veh_route_info = zet_df[['route_id', 'route_long_name']].drop_duplicates()

# Ako je broj rute manji od 100 - klasificira se kao tramvajska linija, inače je autobusna linija
veh_route_info.loc[veh_route_info['route_id'] < 100, 'label'] = veh_route_info['route_id'].astype(str) + \
    ": Tram line " + \
    veh_route_info["route_long_name"]
veh_route_info.loc[veh_route_info['route_id'] >= 100, 'label'] = veh_route_info['route_id'].astype(str) + \
    ": Bus line " + \
    veh_route_info["route_long_name"]

# Preimenujte stupac route_id u vrijednost
veh_route_info.rename(columns={'route_id': 'value'}, inplace=True)

# Sortirajte rute vozila prema njihovom broju (vrijednosti) i pohranite rezultat u rječnik
veh_route_info = veh_route_info[['value', 'label']].sort_values(by=['value']).to_dict('records')

# Jedinstveni brojevi vozila
veh_route_numbers = [o["value"] for o in veh_route_info]

# Brisanje lokalnih varijabli koje se ne koriste - radi lakšeg čitanja
del zet_df, absolute_path, relative_path, static_data_path, feed, response

```

**Slika 17.** Prikaz koda za sortiranje vozila po broju rute.

## 4.2 Python skripta za spremanje podataka u bazu podataka

Na slici 18 prikazana je funkcija `save_to_database`, ključan dio ovog programa koji omogućuje spremanje podataka u bazu podataka. Ova funkcija koristi `pyodbc` biblioteku za uspostavljanje veze s bazom podataka. Kada se uspostavi veza, kreira se kursor koji služi za izvršavanje SQL upita. Kako bi se podaci mogli pohraniti u bazu, koristi se SQL upit definiran u varijabli `query`. Ovdje se koristi parametrizirani upit gdje su nazivi stupaca i vrijednosti postavljeni na znak pitanja (?). Nakon toga, konkretne vrijednosti podataka koje se žele spremiti izdvojene su iz rječnika i pohranjene u varijablu `values`. Nakon što su SQL upit i vrijednosti pripremljeni, izvršava se upit pomoću naredbe `cursor.execute(query, values)`. Tako podaci se zapisuju u tablicu 'ZET\_PODACI' unutar baze podataka 'ZET\_GTFS'.

Važan korak koji slijedi je potvrda promjena i spremanje podataka u bazi korištenjem `connection.commit()`. Nakon što su podaci uspješno spremljeni, kursor se zatvara pomoću `cursor.close()` i veza s bazom se prekida pozivanjem `connection.close()`. Ova funkcija omogućuje da podaci skupljeni iz ZET-ovog servera budu strukturirano i sigurno pohranjeni u SQL bazu podataka.

```
# Funkcija za spremanje podataka u SQL bazu podataka
def save_to_database(data):
    # Uspostavi vezu s bazom podataka
    connection = pyodbc.connect(
        "Driver={SQL Server};"
        "Server=LAPTOP-GMK2UDCO;"
        "Database=ZET_GTFS;"
        "Trusted_Connection=yes;"
    )

    # Kreiraj kursor za izvršavanje SQL upita
    cursor = connection.cursor()

    # Izvrši SQL upit za spremanje podataka

    query = "INSERT INTO ZET_PODACI (id, trip_id, route_id, stop_sequence, arrival_time, departure_time, stop_id, delay_zet, stop_name, stop_lat, " \
            "stop_lon, route_long_name, geometry_x, geometry_y) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)"
    values = (data['id'], data['trip_id'], data['route_id'], data['stop_sequence'], data['arrival_time'],
              data['departure_time'], data['stop_id'], data['delay'], data['stop_name'], data['stop_lat'],
              data['stop_lon'], data['route_long_name'], data['geometry'].x, data['geometry'].y)

    cursor.execute(query, values)

    # Potvrdi promjene u bazi podataka
    connection.commit()

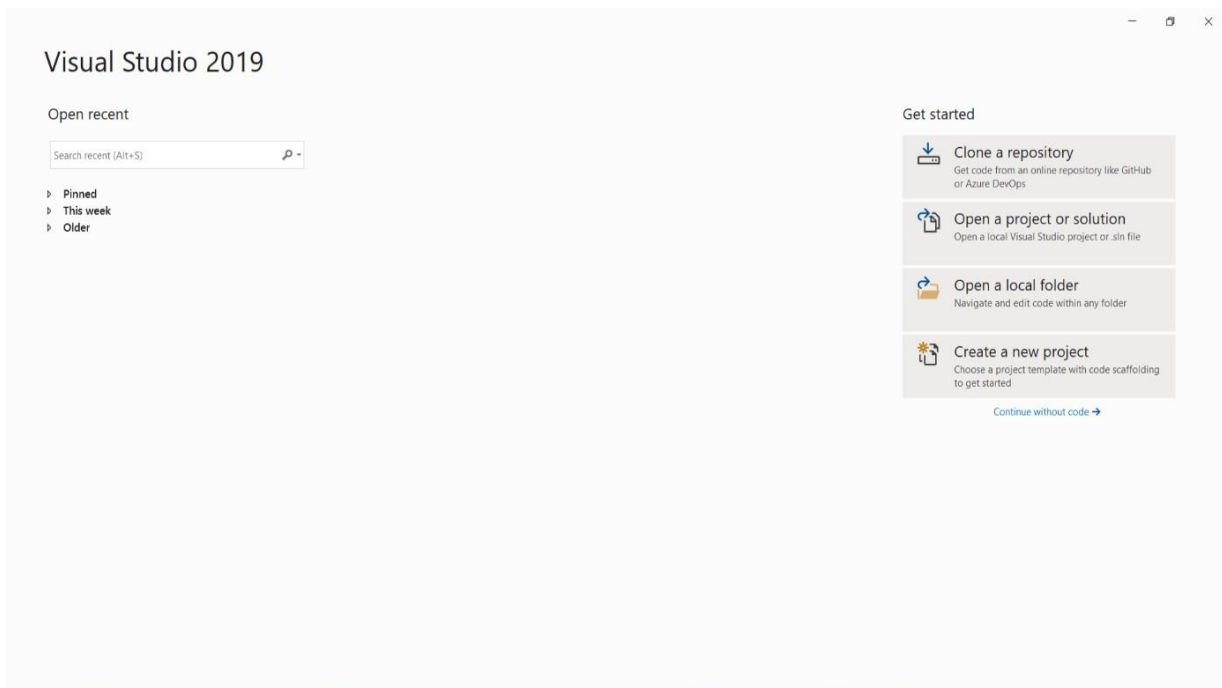
    # Zatvori kursor i vezu s bazom podataka
    cursor.close()
    connection.close()
```

Slika 18. Prikaz koda za spremanje podataka u bazu podataka.

## 5 PROGRAMSKO SUČELJE ZA VIZUALIZACIJU I ANALIZU STVARNO VREMENSKIH PODATAKA

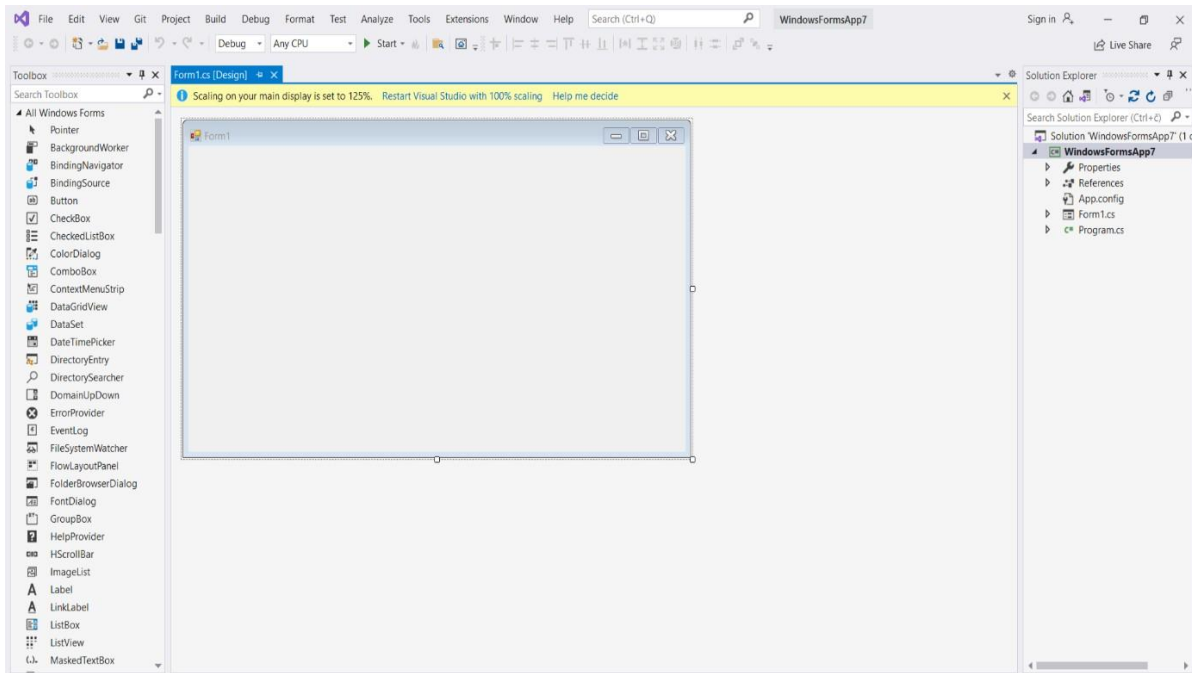
### 5.1 Visual Studio

Visual Studio je integrirano razvojno okruženje (IDE) razvijeno od strane Microsofta, koje omogućuje programerima izradu različitih aplikacija kao što su grafička korisnička sučelja, konzolne aplikacije, web aplikacije, mobilne aplikacije, aplikacije u oblaku te web usluge. Visual Studio nudi podršku za različite programske jezike, no u ovom radu korišten je C# programskog jezika za implementaciju programa. Na **Slici 19** prikazano je sučelje Visual studia.



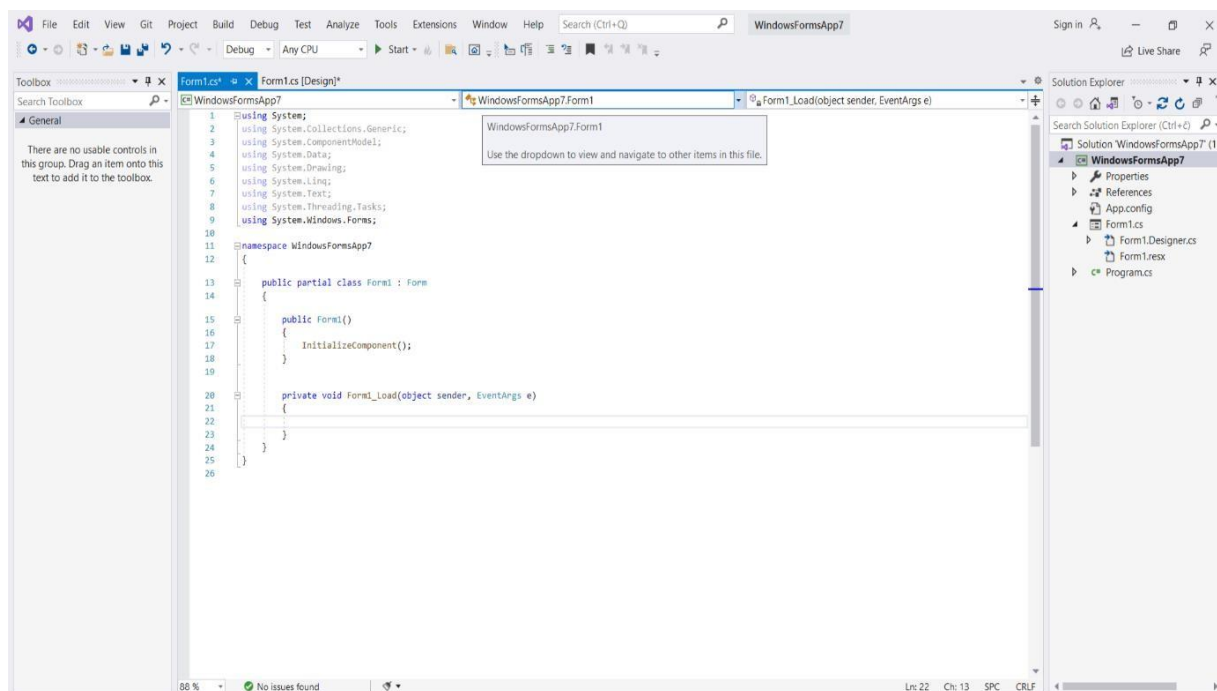
**Slika 19.** Prikaz sučelja Visual studia.

Za izradu korisničkog sučelja, korištena je Windows Forms aplikacija koja se razvija na platformi .NET Framework. Za programiranje korisničkog sučelja u ovom kontekstu korišten je programski jezik C#. Windows Forms je tehnologija koja omogućuje izradu klasičnih desktop aplikacija s grafičkim korisničkim sučeljem na platformi Windows. Ona pruža bogate kontrole i mogućnosti za stvaranje intuitivnih i interaktivnih korisničkih sučelja. Zahvaljujući .NET Frameworku, razvoj Windows Forms aplikacija u C# programskom jeziku je brz i učinkovit. Na **slici 20** prikazano je sučelje programa Visual Studio, koje se sastoji od nekoliko dijelova koji pružaju korisne funkcionalnosti za razvoj aplikacija.



**Slika 20.** Prikaz dijelova sučelja.

1. Gornji dio: u gornjem dijelu prozora nalazi se alatna traka koja sadrži velik broj korisnih funkcija za razvoj i upravljanje projektom. Također, tu se nalazi i gumb za pokretanje programa kako bi se mogla testirati aplikacija.
2. Lijevi dio ispod alatne trake: u ovom dijelu nalazi se povlačenje komponenti za izradu korisničkog sučelja. Ovdje se mogu dodavati gumbi, tekstualna polja (textboxove) i druge komponente povlačenjem iz padajućeg izbornika s lijeve strane. Na ovaj način može se brzo i jednostavno dizajnirati izgled našeg sučelja.
3. Output prozorčić: ispod dijela za pravljenje sučelja nalazi se prozorčić koji prikazuje izlaz ili rezultate programa. Ovdje se mogu vidjeti ispisane poruke, greške ili ostale informacije koje su važne za razumijevanje rada aplikacije.
4. Desni dio: desni dio sučelja podijeljen je na dva dijela:
  - a) Solution Explorer: ovdje se nalazi struktura projekta, gdje su prikazane sve datoteke koje korisnik koristi u programu. Solution Explorer pomaže u organizaciji projekta i omogućuje lako pronalaženje i upravljanje datotekama.
  - b) Properties prozorčić: ovdje se prikazuju svojstva odabrane komponente ili datoteke. Kada se odabere određeni element u sučelju, u ovom prozorčiću mogu se mijenjati razna svojstva kako bi se prilagodio izgled i ponašanje aplikacije.
5. Pisanje koda: kada se dva puta klikne na određeni gumb (ili drugu komponentu) u dijelu za pravljenje sučelja, otvara se prozor za pisanje koda što je prikazano na **slici 21**. Ovdje se mogu definirati funkcije i logika za odabrani element kako bi imao željenu funkcionalnost.



**Slika 21.** Prikaz dijela sučelja za pisanje koda.

Ovo su glavni dijelovi sučelja programa Visual Studio koji omogućuju razvoj aplikacija na efikasan i intuitivan način. Kombinacija grafičkog dizajna sučelja i pisanja koda olakšava razvoj kompleksnih aplikacija i omogućuje programerima da brzo testiraju i implementiraju funkcionalnosti.

## 5.2 Kod za programsko sučelje ZET\_GTFS

**Slika 22** prikazuje metodu `gMap_Load`. U metodi `gMap_Load` definiraju se postavke i konfiguracije za `GMapControl`, kontrolu koja omogućuje prikazivanje karte u korisničkom sučelju. Prvo se postavlja opcija da se karta povlači (pomiče) koristeći desnu tipku miša, što je standardni način kretanja kroz kartu. Zatim se postavlja `MapProvider`, odnosno izvor karte koji će se koristiti. U ovom slučaju, koristi se `OpenStreetMapProvider`, što omogućuje prikazivanje karte s podacima iz `OpenStreetMap-a`. `GMaps.Instance.Mode` postavlja se na `ServerOnly`, što znači da će se koristiti podaci s poslužitelja (online način rada). Nakon toga, postavlja se početna pozicija karte na određene koordinate (45.813, 15.977). Definiraju se i minimalna i maksimalna razina zumiranja kako bi se ograničila mogućnost korisnika. Karta će se moći približiti minimalno na razinu 3 i najviše na razinu 19. Na kraju, postavlja se trenutna razina zumiranja na 12, što će prikazati kartu u umjerenom zoomu. Opcija `ShowCenter` postavlja se na `false` kako bi se skrila ikona centra mape koja bi inače bila vidljiva.

Na slici **23** prikazana je metoda `StaviPlaviMarker`. U ovoj metodi `StaviPlaviMarker` stvaraju se plavi markeri na karti u korisničkom sučelju. Prvo, kreira se novi `GMarkerGoogle` objekt koristeći predane koordinate `tocka` i tip markera `GMarkerGoogleType.blue_dot`. Ovaj tip markera predstavlja plavi marker u obliku malog kruga.

```

private void gMap_Load(object sender, EventArgs e)
{
    gMap.DragButton = MouseButtons.Right;
    gMap.CanDragMap = true;
    gMap.MapProvider = GMap.NET.MapProviders.OpenStreetMapProvider.Instance;
    GMap.NET.GMaps.Instance.Mode = GMap.NET.AccessMode.ServerOnly;
    gMap.Position = new GMap.NET.PointLatLng(45.813, 15.977);
    gMap.MinZoom = 3;
    gMap.MaxZoom = 19;
    gMap.Zoom = 12;
    gMap.ShowCenter = false;
}

```

**Slika 22.** Metoda gMap\_Load.

```

private void StaviPlaviMarker(PointLatLng tocka)
{
    var marker = new GMarkerGoogle(tocka, GMarkerGoogleType.blue_dot);
    //marker.ToolTipText = "ID ovog linka je " + pocetak.linkID;
    //marker.ToolTip.Fill = Brushes.Black;
    //marker.ToolTip.Foreground = Brushes.White;
    //marker.ToolTip.Stroke = Pens.Transparent;
    //marker.ToolTip.TextPadding = new Size(20, 20);
    //marker.ToolTipMode = MarkerTooltipMode.OnMouseOver;
    gMap.Overlays.Add(markerOverlay);
    markerOverlay.Markers.Add(marker);
}

```

**Slika 23.** Metoda StaviPlaviMarker.

Nakon što je marker stvoren, moguće je postaviti dodatne informacije za prikazivanje tooltip-a (balončića s informacijama) kada korisnik pređe mišem preko markera. U trenutnom kodu, linije koje se odnose na postavljanje tooltip-a su komentirane (//), što znači da se ne koriste. Ove linije koda omogućuju postavljanje dodatnih tekstualnih informacija, boje pozadine, boje teksta i drugih vizualnih svojstava tooltip-a. Nakon što je marker kreiran i po potrebi postavljene informacije za tooltip, marker se dodaje u **markerOverlay** (sloj) na **gMap** kartu koja se mora prethodno instalirati. **MarkerOverlay** predstavlja skup markera koji će se prikazivati na karti. Na kraju, plavi marker se dodaje u listu markera unutar **markerOverlay** kako bi se prikazao na karti.

Na **slici 25** prikazan je dio koda aplikacije. Ovaj kod predstavlja događaj „button1\_Click“ koji se izvršava kada korisnik klikne na određeni gumb u korisničkom sučelju. Detalji o tome koji je gumb i gdje se nalazi određuju se putem **sender** i **EventArgs e** parametara. U ovom slučaju, kod sadrži beskonačnu petlju unutar kojeg se neprestano

dohvaćaju i prikazuju podaci iz baze podataka te se postavlja određeno vrijeme čekanja između svakog dohvaćanja. Nakon što korisnik klikne na određeni gumb u korisničkom sučelju, petlja **while** će se pokrenuti i izvršavati cijeli blok koda unutar nje. Postavlja se varijabla **Zaustavi** na **false**, što znači da petlja treba nastaviti s radom. Zatim se neprestano dohvaća trenutno vrijeme i smanjuje za 15 sekundi kako bi se dobilo vrijeme koje je 15 sekundi prije trenutnog vremena. Ovo vrijeme koristi se za SQL upite prema bazi podataka. Pomoću navedenih podataka za spajanje, ostvaruje se veza s bazom podataka putem objekta **sql**.

Uzastopno se izvršavaju tri SQL upita prema bazi podataka. Sva tri upita traže podatke iz tablice **ZET\_PODACI**, gdje je **route\_id** jednak 7 i **time\_input** manji od trenutnog vremena koje se dobije na početku petlje. Prva dva upita dohvaćaju koordinate **geometry\_x** i **geometry\_y**, dok treći upit dohvaća sve podatke. Dohvaćeni podaci koriste se za stvaranje plavog markera na korisničkom sučelju, kako bi se to postiglo potrebno je pozvati metodu kao što je prikazano na **slici 24** te se istovremeno prikazuju u tablici "tablica".

```
StaviPlaviMarker(new PointLa
```

**Slika 24.** Pozivanje metode StaviPlaviMarker.

Nakon što se izvrše SQL upiti i podaci se prikažu u korisničkom sučelju, petlja koristi **Thread.Sleep(15000)** kako bi čekala 15 sekundi prije nego što ponovno izvrši cijeli blok koda, stvarajući tako interval između svakog dohvaćanja podataka iz baze podataka i prikazivanja na korisničkom sučelju. Ova funkcionalnost omogućuje neprekidno ažuriranje podataka u korisničkom sučelju svakih 15 sekundi, što može biti korisno za praćenje i vizualizaciju



```

private void button1_Click(object sender, EventArgs e)
{
    bool Zaustavi = false;
    while (!Zaustavi)
    {

        DateTime dohvatVremena = DateTime.Now;
        dohvatVremena = dohvatVremena.AddSeconds(-15);
        //string trenutno_vrijeme = dohvatVremena.ToString("yyyy-MM-dd
hh:m:ss.fff");
        string trenutno_vrijeme = "" + dohvatVremena.ToString("s") + "";

        // Beskonačna petlja za povećanje vremena

        try
        {
            sql = new SqlConnection(@"Server=LAPTOP-
GMK2UDCO;Database=ZET_GTFS;Trusted_Connection=True;");

            sql.Open();
            MessageBox.Show("Podaci su učitani");

        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }

        SqlCommand cmd1 = new SqlCommand("select top 1 geometry_x from
ZET_PODACI where route_id=7 and time_input<" + trenutno_vrijeme, sql);
        SqlDataReader sdr = cmd1.ExecuteReader();
        sdr.Read();
        double točka_x = Convert.ToDouble(sdr.GetValue(0));
        sdr.Close();

        SqlCommand cmd2 = new SqlCommand("select top 1 geometry_y from
ZET_PODACI where route_id=7 and time_input<" + trenutno_vrijeme, sql);
        sdr = cmd2.ExecuteReader();
        sdr.Read();
        double točka_y = Convert.ToDouble(sdr.GetValue(0));
        sdr.Close();

        StaviPlaviMarker(new PointLatLng(točka_y, točka_x));

        SqlCommand cmd = new SqlCommand("select top 1 * from ZET_PODACI where
route_id=7 and time_input<" + trenutno_vrijeme, sql);

        SqlDataReader reader = cmd.ExecuteReader();

        DataTable dt = new DataTable();
        dt.Load(reader);
        tablica.DataSource = dt;

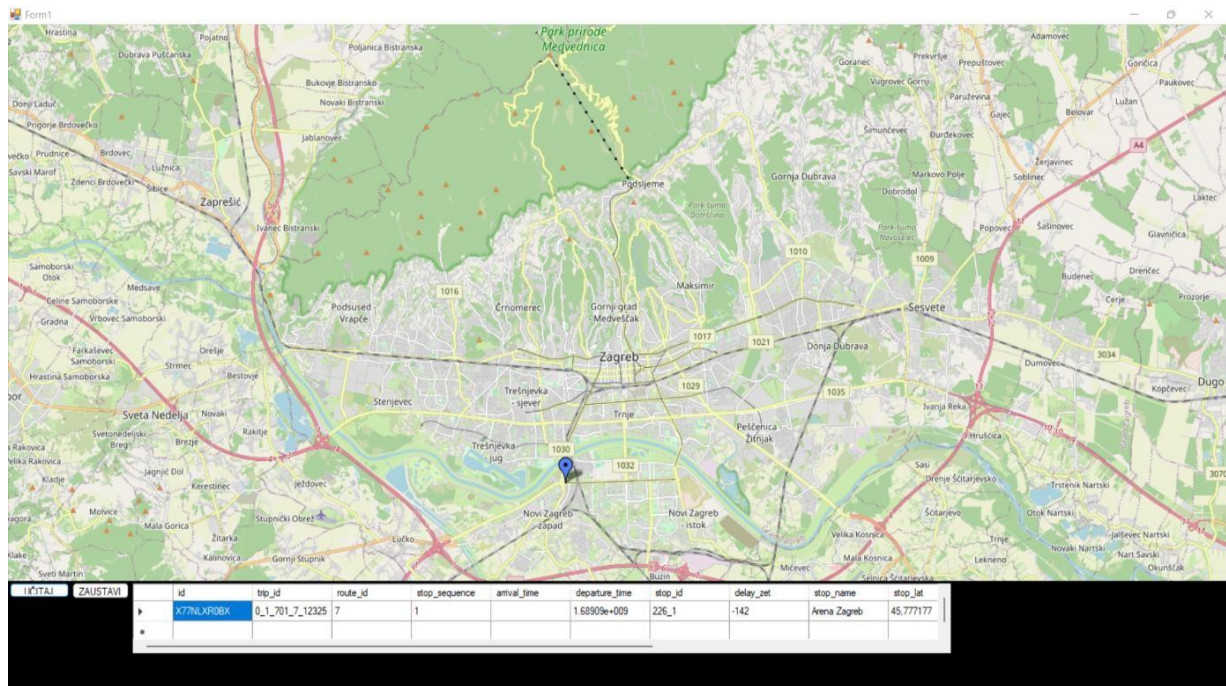
        Thread.Sleep(15000);
    }
}

```

**Slika 25.** Kod za dohvaćanje podataka iz baze i pozivanje metode za vizualizaciju.

### 5.3 Programsko sučelje ZET\_GTFS

Razvoj aplikacije za praćenje tramvaja u javnom gradskom prijevozu ZET bio je ključan korak u poboljšanju iskustva putnika i unaprjeđenju kvalitete usluge. Ova inovativna aplikacija omogućuje putnicima da lakše i brže prate svoje odabrane tramvaje i autobuse, smanjujući vrijeme čekanja i povećavajući efikasnost u prijevozu. Korištenjem C# programskog jezika napravljeno je korisničko sučelje koje omogućuje vizualizaciju položaja tramvaja i autobusa korisnicima javnog gradskog prijevoza prikazano na **slici 26**.



**Slika 26.** Prikaz sučelja aplikacije

Sučelje se sastoji od dva gumba i jednog polja za prikaz podataka određenog tramvaja. Klikom na gumb „**Učitaj**“ aplikacija dohvaća podatke iz baze podataka i prikazuje ih korisnicima u stvarnom vremenu, omogućujući im da vide gdje se njihov tramvaj nalazi na karti. Kada korisnik pritisne drugi gumb „**Zaustavi**“, aplikacija se zaustavlja, prekidajući prikaz podataka na karti.

Aplikacija pruža putnicima mogućnost da u svakom trenutku prate svoj odabrani tramvaj ili autobus u stvarnom vremenu. To smanjuje neizvjesnost i olakšava putnicima da se pravovremeno organiziraju i stignu na željeno odredište. Također, ZET bi mogao imati koristi od ove aplikacije jer omogućuje bolje praćenje i upravljanje vozilima, što rezultira poboljšanom kvalitetom usluge i zadovoljnijim putnicima. Uz kontinuirano praćenje i analizu podataka, ZET može koristiti povratne informacije putnika kako bi unaprijedili svoju uslugu i podigli razinu javnog gradskog prijevoza. Aplikacija za praćenje tramvaja i autobusa u stvarnom vremenu predstavlja korak naprijed u modernizaciji i optimizaciji javnog prijevoza, pružajući putnicima bolje iskustvo i veću efikasnost u putovanju.

## 6 ANALIZA POVIJESNO PRIKUPLJENIH PODATAKA

Prikupljeni podaci prilikom analize povijesno spremljenih podataka u bazu o putovanjima tramvaja i autobusa, pružaju uvid u promet i korištenje javnog prijevoza u određenom vremenskom intervalu. Tijekom analize od 2023-07-14 16:00:00 do 2023-07-14 19:00:00, analizirane su različite linije tramvaja i autobusa koje se koriste na mreži. Pomoću upita:

```
SELECT DISTINCT CAST(route_id AS INT) AS route_id
FROM ZET_PODACI where time_input >'2023-07-14 16:00:00 'and time_input<'2023-07-14
19:00:00' order by route_id asc
```

vidljivo je da u tom intervalu putuju 134 linije, 13 tramvajskih i 121 autobusna linija. Prvo su analizirane tramvajske linije. Ustanovljeno je da su sljedeće tramvajske linije putovale tijekom analiziranog vremenskog intervala koje su prikazane u **tablici 3**: 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14 i 17. Svaka od tih linija ima svoju rutu i frekvenciju putovanja. Ovi podaci ukazuju na prometne rute koje su bile aktivne tijekom spomenutog vremena.

**Tablica 3.** Prikaz tramvajskih linija koje putuju u intervalu od 16-19h.

Tramvajska linija	
1.	2
2.	3
3.	4
4.	5
5.	6
6.	7
7.	8
8.	9
9.	11
10.	12
11.	13
12.	14
13.	17

Nadalje, fokus se prebacuje na autobusne linije. Tijekom analiziranog vremena ima ukupno 120 autobusnih linija koje su korištene tijekom navedenog intervala. Od svih autobusnih linija, linija 118 se istaknula kao najkorištenija. Linija 118 je zabilježila čak 1176 zapisa u tih 3 sata. Ova linija je posebno zanimljiva jer povezuje trg Mažuranić i Voltino. To znači da je ova ruta izuzetno popularna među putnicima. Analizira se top 5 autobusnih linija koje su imale najviše zapisa u navedenom vremenskom intervalu. Osim linije 118, koje je dominirala sa svojim brojem zapisa, postoje i još četiri linije koje su bile vrlo korištene. To se linije 268, 172, 220 i 209. Svi ovi podaci ukazuju na važnost javnog prijevoza i omogućuju uvid u popularne trase koje koriste građani tijekom navedenog vremena. Pomoću upita:

```
SELECT TOP 5 route_id, COUNT(*) AS broj_pojavljivanja FROM ZET_PODACI where route_id>100
and time_input >'2023-07-14 16:00:00 'and time_input<'2023-07-14 19:00:00' GROUP BY
route_id
ORDER BY COUNT(*) DESC
```

dolazi se do pet linija koje su imale najviše spremljenih zapisa u bazu podataka što je prikazano na **slici 27**.

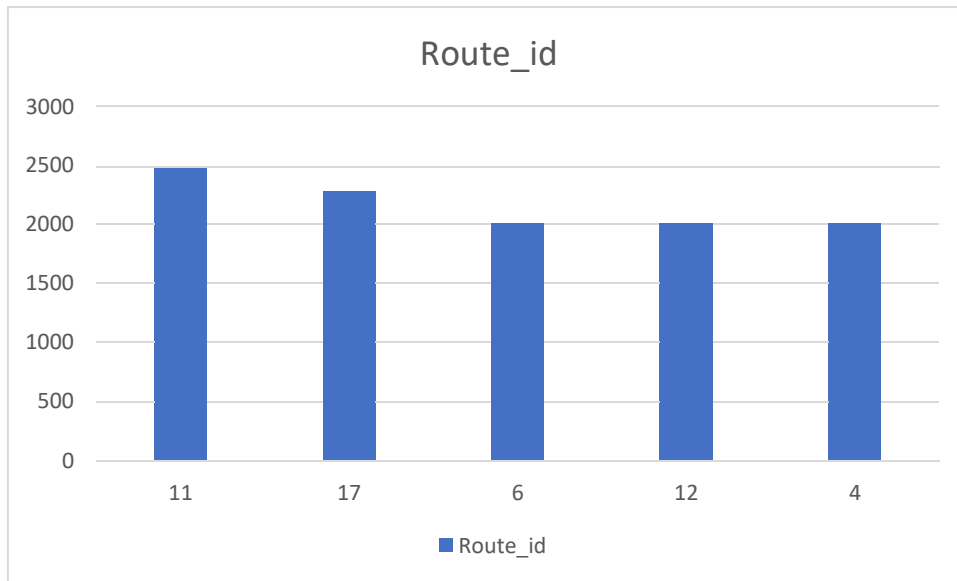


**Slika 27.** Prikaz grafikona o najprometnijim autobusnim linijama.

Dodatno se analizira broj vozila za svaku od linija tijekom tri satnog intervala. Pomoću toga otkriveno je da linija 11 ima najviše vozila, a zabilježeno je čak 2495 zapisa u analiziranom vremenskom rasponu. Linija 11 povezuje Črnomerec i Dubec, što su dva značajna dijela grada. Očito je da postoji velika potražnja za ovom rutom, što objašnjava njen značajan broj putovanja u spomenutom intervalu. **Slika 26** prikazuje grafikon koji nam ukazuje na važnost tih linija. Linija 11 bila je na vrhu s najvećim brojem zapisa, a linije 17, 6, 12 i pratile su je s velikim brojem putovanja. Pomoću upita:

```
SELECT TOP 5 route_id, COUNT(*) AS broj_pojavljivanja FROM ZET_PODACI where
route_id<100 and time_input >'2023-07-14 16:00:00 'and time_input<'2023-07-14
19:00:00' GROUP BY route_id
ORDER BY COUNT(*) DESC
```

dolazi se do pet linija koje su imale najviše spremljenih zapisa u bazu podataka što je prikazano na **slici 28**.



**Slika 28.** Prikaz grafikona o najprometnijim tramvajskim linijama

Sve ove analize omogućuju dublje razumijevanje korištenja javnog prijevoza u određenom vremenskom periodu. Ovo je od vitalnog značaja za optimizaciju i planiranje javnog prijevoza kako bi se udovoljile potrebama putnika i poboljšala kvaliteta usluge.

## 7 ZAKLJUČAK

Prikupljanje i analiza podataka o putovanjima javnog prijevoza, poput tramvaja i autobusa, te vizualizacija tih podataka na karti, predstavlja ključan alat za razumijevanje i optimizaciju javnog prijevoza u gradu. Ovaj proces omogućuje razvoj aplikacija koje mogu pružiti korisne informacije i poboljšati iskustvo putnika u gradskom prometu.

Prikupljanje podataka u bazu omogućuje stvaranje obuhvatne zbirke informacija o putovanjima javnog prijevoza. To uključuje podatke o vremenima dolaska i odlaska, rutama koje se koriste, broju putnika i drugim relevantnim informacijama. Ovaj skup podataka može biti ključan za analizu prometa, identificiranje popularnih ruta i stajališta, te planiranje optimalnih voznih redova.

Vizualizacija tramvajskih linija na karti pruža pregled mreže javnog prijevoza i njegovu povezanost s drugim dijelovima grada. Korisnici mogu jednostavno prepoznati rasprostranjenost stajališta, rute koje se koriste i lokacije ključnih prometnih čvorišta. Ova vizualizacija pomaže korisnicima u bržem pronalaženju najbližeg stajališta i identifikaciji mogućih alternativa za putovanje.

Analiza povijesnih podataka omogućuje dublje razumijevanje ponašanja putnika tijekom određenih vremenskih intervala. Kroz analizu možemo otkriti popularne rute i linije tijekom specifičnih sati, dana ili sezona. To pomaže u optimizaciji resursa javnog prijevoza, kao što su povećanje broja vozila ili promjena voznih redova kako bi se zadovoljile potrebe putnika.

## LITERATURA

1. ZET. [Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s: <https://www.zet.hr>
2. Grad Zagreb službene stranice. Od konjskog di niskopodnog tramvaja. [Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s: <https://www.zagreb.hr/od-konjskog-do-niskopodnog-tramvaja/19152>
3. General Transit Feed Specification. GTFS: Making Public Transit Data Universally Accessible [Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s: <https://gtfs.org/>
4. Zagreb info. Autobusi Grada Zgreba.[Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s:<https://www.zagreb.info/aktualno/zg/preskakanje-voznje-busevi-se-ne-drze-rasporeda-neke-linije-sve-praznije-cekali-50-minuta-a-ide-svaki-10/474360/>
5. ZGPortal. Autobusne linije ZET-a od broja 201 do broja 299 – popis linija i trasa .[Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s:<https://www.zgportal.com/zginfo/autobusne-linije-zet-a-od-broja-201-do-broja-299-popis-linija-i-trasa/>
6. Moovit. Linija 5 [Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s:[https://moovitapp.com/index/hr/javni\\_prijevoz-line-5-Zagreb-3761-1433367-7758559-2](https://moovitapp.com/index/hr/javni_prijevoz-line-5-Zagreb-3761-1433367-7758559-2)
7. Povjest.hr. U promet puštena Zagrebačka uspinjača (1890.) [Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s: <https://povijest.hr/nadanasnjidan/u-promet-pustena-zagrebacka-uspinjača-1890/>
8. Povijes.hr. Prvi električni tramvaj. [Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s: <https://povijes.hr/nadanasnjidan/zagreb-dobio-prvi-elektricni-tramvaj-1910/>
9. Wikipedia. ZET Zagreb.png. [Online]. [Pristupljeno: 2023 Lipanj.] Preuzeto s:[https://hr.wikipedia.org/wiki/Datoteka:ZET\\_Zagreb.png](https://hr.wikipedia.org/wiki/Datoteka:ZET_Zagreb.png)
10. Bošnjak I., Inteligentni Transportni Sustavi 1, Sveučilište u Zagrebu fakultet prometnih znanosti, Zagreb 2006.

## 8 DODATAK 1- Popis ZET linija

### NOĆNE LINIJE

- 116 LJUBLJANICA - PODSUSED MOST
- 172 ZAGREB (ČRNOMEREC) - ZAPREŠIĆ
- 212 DUBEC - SESVETE
- 268 ZAGREB (GLAVNI KOLODVOR) - VELIKA GORICA

### IZVANREDNE LINIJE

- 607 SAVSKI MOST - SOPOT
- 617 GRAČANSKO DOLJE - DRAŠKOVIĆEVA

### BORONGAJ - položaj perona

Trase autobusnih linija terminala Borongaj

- 231 BORONGAJ - DUBEC
- 269 BORONGAJ - SESVETSKI KRALJEVEC - IVER

### BRITANSKI TRG - položaj perona

Trase autobusnih linija terminala Britanski trg

- 101 BRITANSKI TRG - GORNJE PREKRIŽJE
- 102 BRITANSKI TRG - MIHALJEVAC
- 103 BRITANSKI TRG - KRALJEVEC
- 105 KAPTOL - BRITANSKI TRG
- 138 BRITANSKI TRG - ZELENGAJ - BRITANSKI TRG

### ČRNOMEREC - položaj perona

Trase autobusnih linija terminala Britanski trg

- 109 ČRNOMEREC - DUGAVE
- 119 ČRNOMEREC - PODSUSED MOST
- 120 ČRNOMEREC - GAJNICE - PREČKO
- 121 ČRNOMEREC - KARAŽNIK - GAJNICE
- 122 ČRNOMEREC - PODSUSED-CENTAR
- 123 ČRNOMEREC - PODSUSEDKO DOLJE
- 124 ČRNOMEREC - GORNJI STENJEVEC
- 125 ČRNOMEREC - GORNJE VRAPČE
- 126 ČRNOMEREC - GORNJA KUSTOŠIJA - KRVARIĆ
- 127 ČRNOMEREC - MIKULIĆI
- 128 ČRNOMEREC - LUKŠIĆI
- 130 ČRNOMEREC - BORČEC



- 131 ČRNOMEREC - BIZEK
- 134 ČRNOMEREC - PREČKO
- 135 ČRNOMEREC - GRABERJE
- 136 ČRNOMEREC - ŠPANSKO
- 137 ČRNOMEREC - PERJAVICA - BORČEC
- 172 ZAGREB (ČRNOMEREC) - ZAPREŠIĆ
- 176 ZAGREB (ČRNOMEREC) - GORNJA BISTRA
- 177 ZAGREB (ČRNOMEREC) - POLJANICA - GORNJA BISTRA

## ČUČERJE

Trasa autobusne linije iz Čučerja

- 239 ČUČERJE - DANKOVEC - MARKUŠEVEČKA TRNAVA

## DUBEC - položaj perona

Trase autobusnih linija terminala Dubec

- 211 DUBEC - BRANOVEČINA - DUBEC
- 212 DUBEC - SESVETE
- 223 DUBRAVA - TRNOVČICA - DUBEC
- 224 DUBEC - NOVOSELEC
- 231 BORONGAJ - DUBEC
- 261 DUBEC - SESVETE - GORANEC
- 262 DUBEC - SESVETE - PLANINA DONJA
- 263 DUBEC - SESVETE - KAŠINA - PLANINA GORNJA
- 264 DUBEC - SESVETE - JESENOVEC
- 267 DUBEC - MARKOVO POLJE
- 270 DUBEC - SESVETE - BLAGUŠA
- 271 DUBEC - SESVETE - GLAVNICA DONJA
- 272 DUBEC - SESVETE - MORAVČE
- 273 DUBEC - SESVETE - LUŽAN
- 274 ZAGREB (DUBEC) - SESVETE - LAKTEC
- 279 DUBEC - NOVI JELKOVEC
- 280 DUBEC - SESVETE - ŠIMUNČEVEC
- 283 DUBEC - BRESTJE - SESVETE

## DUBRAVA - položaj perona

Trase autobusnih linija terminala Dubrava

- 205 DUBRAVA - MARKUŠEVEC - BIDROVEC
- 206 DUBRAVA - MIROŠEVEC
- 208 DUBRAVA - VIDOVEC
- 209 DUBRAVA - ČUČERJE
- 210 DUBRAVA - STUDENTSKI GRAD - NOVI RETKOVEC
- 213 DUBRAVA - JALŠEVEC

- 223 DUBRAVA - TRNOVČICA - DUBEC
- 230 DUBRAVA - GRANEŠINSKI NOVAKI
- 232 DUBRAVA - ČRET
- 235 DUBRAVA - TRNAVA - KOZARI BOK

GLAVNI KOLODVOR - položaj perona

Trase autobusnih linija terminala Glavni kolodvor

- 108 GLAVNI KOLODVOR - SAVSKI MOST
- 166 ZAGREB (GLAVNI KOLODVOR) - DONJI DRAGONOŽEC
- 218 GLAVNI KOLODVOR - SAVICA - BOROVJE
- 219 GLAVNI KOLODVOR - SLOBOŠTINA
- 220 GLAVNI KOLODVOR - DUGAVE
- 221 GLAVNI KOLODVOR - TRAVNO
- 229 GLAVNI KOLODVOR - ODRA - MALA MLAKA
- 234 GLAVNI KOLODVOR - KAJZERICA - LANIŠTE
- 241 GLAVNI KOLODVOR - VELIKO POLJE
- 242 GLAVNI KOLODVOR - PODBREŽJE
- 243 GLAVNI KOLODVOR - KAJZERICA (RADOVI)
- 268 ZAGREB (GLAVNI KOLODVOR) - VELIKA GORICA
- 281 GLAVNI KOLODVOR - NOVI JELKOVEC
- 310 ZAGREB (GLAVNI KOLODVOR) - PETROVINA
- 311 ZAGREB (GLAVNI KOLODVOR) - CEROVSKI VRH
- 313 ZAGREB (GLAVNI KOLODVOR) - VUKOMERIĆ
- 330 ZAGREB (GLAVNI KOLODVOR) - VELIKA GORICA (BRZA LINIJA)

JANDRIĆEVA

Trasa autobusne linije iz Jandrićeve

- 104 JANDRIĆEVA - DOM UMIROVLJENIKA

JANKOMIR

Trase autobusnih linija iz Jankomira

- 107 JANKOMIR - ŽITNJAK
- 115 LJUBLJANICA - ŠPANSKO - JANKOMIR
- 146 RELJKOVIĆEVA - MALEŠNICA - JANKOMIR

KAMPUS

Trasa autobusne linije iz Kampusu

- 236 KAMPUS - ČAVIĆEVA

## KAPTOL - položaj perona

### Trase autobusnih linija terminala Kaptol

- 105 KAPTOL - BRITANSKI TRG
- 106 KAPTOL - MIROGOJ - KREMATORIJ
- 201 KAPTOL - KVATERNIKOV TRG
- 203 SVETICE - VINEC - KAPTOL
- 226 KAPTOL - REMETE - SVETICE
- 238 KAPTOL - KOZJAK

## RELJKOVIĆEVA - položaj perona

### Trase autobusnih linija iz Reljkovićeve

- 139 RELJKOVIĆEVA - JELENOVAC - RELJKOVIĆEVA
- 141 RELJKOVIĆEVA - VINOGRADI - RELJKOVIĆEVA
- 146 RELJKOVIĆEVA - MALEŠNICA - JANKOMIR
- 148 RELJKOVIĆEVA - HERCEGOVAČKA - BOSANSKA - RELJKOVIĆEVA

## KOLEDINEČKA

### Trasa autobusne linije iz Koledinečke

- 214 KOLEDINEČKA - TRNAVA - KOZARI BOK

## KUNIŠČAK

### Trase autobusnih linija iz Kuniščaka

- 129 KUNIŠČAK - ŠESTINSKI DOL
- 149 KUNIŠČAK - VRHOVEC

## KUSTOŠIJANSKA

### Trasa autobusne linije iz Kustošijanske

- 151 KUSTOŠIJANSKA - ZAVRŠJE - VRAPČANSKA

## KVATERNIKOV TRG - položaj perona

### Trase autobusnih linija terminala Kvaternikov trg

- 201 KAPTOL - KVATERNIKOV TRG
- 202 KVATERNIKOV TRG - KOZJAK
- 204 KVATERNIKOV TRG - HORVATOVAC - VOĆARSKA - KVATERNIKOV TRG

- 207 KVATERNIKOV TRG - RIM - KVATERNIKOV TRG
- 215 KVATERNIKOV TRG - TRNAVA
- 216 KVATERNIKOV TRG - RESNIK - IVANJA REKA
- 217 KVATERNIKOV TRG - STRUGE - PETRUŠEVEČKO NASELJE
- 237 KVATERNIKOV TRG - KOZARI PUTEVI
- 276 ZAGREB (KVATERNIKOV TRG) - IVANJA REKA - DUMOVEC
- 290 ZAGREB (KVATERNIKOV TRG) - ZRAČNA LUKA - VELIKA GORICA

LJUBLJANICA - položaj perona

Trase autobusnih linija terminala Ljubljana

- 113 LJUBLJANICA - JARUN
- 114 LJUBLJANICA - PREČKO
- 115 LJUBLJANICA - ŠPANSKO - JANKOMIR
- 116 LJUBLJANICA - PODSUSED MOST

MIHALJEVAC - položaj perona

Trase autobusnih linija terminala Mihaljevac

- 102 BRITANSKI TRG - MIHALJEVAC
- 140 MIHALJEVAC - SLJEME (RADOVI)
- 233 MIHALJEVAC - MARKUŠEVEC

NOVI JELKOVEC

Trase autobusnih linija iz Novog Jelkovca

- 279 DUBEC - NOVI JELKOVEC
- 281 GLAVNI KOLODVOR - NOVI JELKOVEC
- 282 SESVETE - NOVI JELKOVEC

PREČKO - položaj perona

Trase autobusnih linija iz Prečkog

- 114 LJUBLJANICA - PREČKO
- 120 ČRNOMEREC - GAJNICE - PREČKO
- 134 ČRNOMEREC - PREČKO
- 168 SAVSKI MOST - JEŽDOVEC - PREČKO

SAVSKI MOST - položaj perona

Trase autobusnih linija terminala Savski most

- 108 SAVSKI MOST - GLAVNI KOLODVOR
- 110 SAVSKI MOST - BOTINEC
- 111 ZAGREB (SAVSKI MOST) - DONJI STUPNIK - STUPNIČKI OBREŽ
- 112 SAVSKI MOST - LUČKO
- 132 SAVSKI MOST - GOLI BREG - BREZOVICA
- 133 SAVSKI MOST - SVETA KLARA - ČEHI
- 159 SAVSKI MOST - STRMEC ODRANSKI
- 160 SAVSKI MOST - LIPNICA - HAVIDIĆ SELO
- 161 SAVSKI MOST - KUPINEČKI KRALJEVEC - ŠTRPET
- 162 SAVSKI MOST - AŠPERGERI
- 163 SAVSKI MOST - DONJI TRPUCI - GORNJI TRPUCI
- 164 ZAGREB (SAVSKI MOST) - HORVATI
- 168 SAVSKI MOST - JEŽDOVEC - PREČKO
- 315 ZAGREB (SAVSKI MOST) - LUKAVEC

SESVETE - položaj perona

Trase autobusnih linija terminala Sesevete

- 212 DUBEC - SESVETE
- 225 SESVETE - RESNIK - KOZARI BOK
- 275 SESVETE - SESVETSKA SOPNICA
- 277 SESVETE - SESVETSKA SELNICA
- 278 SESVETE - SESVETSKA SELA - KRALJEVEČKI NOVAKI
- 282 SESVETE - NOVI JELKOVEC
- 283 DUBEC - BRESTJE - SESVETE
- 284 SESVETE - IVANJA REKA

SVETICE - položaj perona

Trase autobusnih linija terminala Svetice

- 203 SVETICE - VINEC - KAPTOL
- 226 KAPTOL - REMETE - SVETICE
- 227 SVETICE - GORNJI BUKOVAC - GRAČANSKO DOLJE
- 228 SVETICE - REBRO - SVETICE

TRG BANA JOSIPA JELAČIĆA

- 150 TUŠKANAC GARAŽA - GORNJI GRAD - TRG BANA JOSIPA JELAČIĆA (RADOVI)

TRG MAŽURANIĆA - položaj perona

Trasa autobusne linije s Trga Mažuranića

- 118 TRG MAŽURANIĆA - VOLTINO

## VELIKA GORICA - položaj perona

### Trase autobusnih linija terminala Velika Gorica

- 268 ZAGREB (GLAVNI KOLODVOR) - VELIKA GORICA
- 290 ZAGREB (KVATERNIKOV TRG) - ZRAČNA LUKA - VELIKA GORICA
- 302 VELIKA GORICA - VELIKA BUNA
- 303 VELIKA GORICA - KOZJAČA
- 304 VELIKA GORICA - MRACLIN
- 305 VELIKA GORICA - TUROPOLJE
- 309 VELIKA GORICA - SASI
- 310 ZAGREB (GLAVNI KOLODVOR) - PETROVINA
- 319 VELIKA GORICA - LUKAVEC
- 321 VELIKA GORICA - STRMEC BUKEVSKI
- 322 VELIKA GORICA - CEROVSKI VRH
- 323 VELIKA GORICA - RIBNICA - LAZINA
- 324 VELIKA GORICA - ČIČKA POLJANA
- 325 VELIKA GORICA - VUKOJEVAC
- 326 VELIKA GORICA - ZRAKOPLOVNO TEHNIČKI CENTAR
- 330 ZAGREB (GLAVNI KOLODVOR) - VELIKA GORICA (BRZA LINIJA)
- 335 VELIKA GORICA - PLESO - DONJA LOMNICA

## VRAPČANSKA ALEJA - položaj perona

### Trase autobusnih linija s Vrapčanske aleje

- 142 VRAPČANSKA ALEJA - JAČKOVINA - VRAPČANSKA ALEJA
- 143 VRAPČANSKA ALEJA - OREŠJE
- 145 VRAPČANSKA ALEJA - ORANICE

## ZAPREŠIĆ - položaj perona

### Trase autobusnih linija terminala Zaprešić

- 172 ZAGREB (ČRNOMEREC) - ZAPREŠIĆ
- 174 ZAPREŠIĆ - ŽEJINCI
- 182 TRG MLADOSTI - GROBLJE ZAPREŠIĆ - ŠIBICE

## ZAPRUĐE - položaj perona

### Trase autobusnih linija terminala Zapruđe

- 295 ZAPRUĐE - JAKUŠEVEC - ZAPRUĐE
- 307 ZAGREB (ZAPRUĐE) - STRMEC BUKEVSKI
- 308 ZAGREB (ZAPRUĐE) - SASI

## ŽITNJAK - položaj perona

### Trase autobusnih linija iz Žitnjaka

- 107 JANKOMIR - ŽITNJAK
- 222 REMETINEC - ŽITNJAK[1]

## POPIS SLIKA

**Slika 1.** Prvi Zagrebački tramvaj.

**Slika 2.** Prvi električni tramvaj.

**Slika 3.** Uspinjača.

**Slika 4.** Novi autobusi ZET-a.

**Slika.5** Prikaz tramvajske linije 5.

**Slika 6.** Tramvajski promet u Zgrebu.

**Slika 7.**Prikaz perona sa autobusom.

**Slika 8.** Prikaz perona i njegove infrastrukture.

**Slika 9.** Prikaz real time podataka ZET-a

**Slika 10.** Primjer Sql upita.

**Slika 11.** Prikaz koda za stvaranje tablice ZET\_PODACI.

**Slika 12.** Integracija ODI (putno informiranje vozača) s drugim sustavima.

**Slika 13.** Prikaz dijagrama toka sustava.

**Slika 14.** Prikaz sučelja platforme PYCharm.

**Slika 15.** Prikaz koda za skidanje i pretvaranje podataka u rječnik.

**Slika 16.** Prikaz koda za formiranje okvira podataka i povezivanja sa statičkim podacima.

**Slika 17.** Prikaz koda za sortiranje vozila po broju rute.

**Slika 18.** Prikaz koda za spremanje podataka u bazu podataka.

**Slika 19.** Prikaz sučelja Visual studia.

**Slika 20.** Prikaz dijelova sučelja.

**Slika 21.** Prikaz dijela sučelja za pisanje koda.

**Slika 22.** Metoda gMap\_Load.

**Slika 23.** Metoda StaviPlaviMarker.



**Slika 24.** Pozivanje metode StaviPlaviMarker.

**Slika 25.** Kod za dohvaćanje podataka iz baze i pozivanje metode za vizualizaciju.

**Slika 26.** Prikaz sučelja aplikacije.

**Slika 27.** Prikaz grafikona o najprometnijim autobusnim linijama.

**Slika 28.** Prikaz grafikona o najprometnijim tramvajskim linijama.

## POPIS TABLICA

**Tablica 2.** Prikaz tramvajskih linija koje putuju u intervalu od 16-19h.

**Tablica 1.** Prikaz i opis statičkih podataka ZET-a.

**Tablica 3.** Prikaz i opis real-time podataka ZET-a.

Sveučilište u Zagrebu  
Fakultet prometnih znanosti  
Vukelićeva 4, 10000 Zagreb

## IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOSTI

Izjavljujem i svojim potpisom potvrđujem da je \_\_\_\_\_  
*[završni rad]*  
(vrsta rada)

isključivo rezultat mogega vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju upotrijebljene bilješke i bibliografija. Izjavljujem da nijedan dio rada nije napisan na nedopušten način, odnosno da je prepisan iz necitiranog rada te da nijedan dio rada ne krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu završnog/diplomskog rada obrada i vizualizacija podataka pružatelja pod naslovom javnog gradskog prijevoza u stvarnom vremenu, u Nacionalni repozitorij završnih i diplomskih radova ZIR.

U Zagrebu, 06.09.2023

Student/ica:

*Ante Šarić*

(ime i prezime, potpis)