

Tramvaji

Mihael Petrinjak

26. siječnja 2024.

Sadržaj

1	Predgovor	2
2	Uvod	3
2.1	Originalni tekst zahtjeva:	3
2.2	Kontekst sustava	3
2.2.1	Tramvaji	3
2.2.2	Centralni poslužitelj	3
2.2.3	Baza podataka	3
2.2.4	Displeji	4
3	Zahtjevi na sustav	5
3.1	Funkcionalni zahtjevi	5
3.2	Nefunkcionalni zahtjevi	5
4	Modeli sustava	6
4.1	Opis upotrebe sustava	6
4.1.1	Specifikacija use caseova	7
4.1.2	Activity dijagrami	16
4.2	Klase	17
4.2.1	Sequence dijagrami	18
5	Oblikovanje	21
5.1	Class dijagram	21
5.2	Sequence dijagrami	23
5.2.1	DisplayClientInit, DisplayData	23
5.2.2	TramClientSetup, SendMessage, UpateTramLocation	23
5.2.3	AnalyzeSignal, TransmitToStation, EstimateStationData	25
5.3	Component dijagram	29

1 Predgovor

Ovaj dokument je projektni zadatak za kolegij *Softversko inženjerstvo* na Matematičkom odsjeku PMF-a. Bavi se projektiranjem jednog softverskog sustava na razini analize i oblikovanja.

2 Uvod

2.1 Originalni tekst zahtjeva:

Riječ je o distribuiranom sustavu koji omogućuje putnicima da na tramvajskim stanicama dobiju informaciju o skorom dolasku tramvaja. Sustav **preko bežične mreže** povezuje **tramvaje, središnji poslužitelj i displeje** na stanicama. Tramvaj ima ugrađeni **GPS** uređaj i u kratkim vremenskim razmacima dojavljuje svoj položaj poslužitelju. Na osnovi nekoliko zadnjih dojava **poslužitelj računa** brzinu tramvaja te predviđa koliko će vremena tramvaju trebati do koje stanice. **Poslužitelj na displejima periodički osvježava** podatke o tramvajima i njihovim vremenima dolaska. U slučaju većeg zastoja, kvara ili opasnosti, **vozač tramvaja može poslati odgovarajuću poruku** poslužitelju, a poslužitelj tada ponovo ažurira podatke na displejima te po potrebi zove ekipu za **servisiranje ili policiju**.

2.2 Kontekst sustava

2.2.1 Tramvaji

U svakom tramvaju je instaliran uređaj s vlastitim operacijskim sustavom i sučeljem. Na njega je moguće instalirati komponente ovog sustava „tramvajski klijent”.

Osim toga, ugrađen je i GPS uređaj koji je povezan s njim. Standardno vrijeme osvježavanja koordinata koje računa GPS prijamnik je 1 sekunda.

Tramvaju je ugrađeno sklopovlje za bežičnu komunikaciju. Odnosno, postoji antena za primanje i odašiljanje radiovalova. Pripadni čipovi imaju određeni memorisjki kapacitet za zapisivanje prijevoda nadolazećih signala i kriptiranje podataka za odašiljanje. Spomenuti klijent ima sučelje prema antenskom sustavu pa može pisati i čitati iz memorije te slati poruke.

2.2.2 Centralni poslužitelj

Centralni poslužitelj je jako računalo u mogućnosti paralelizirati mnogo procesa. Ima hardversku podršku jake radio antene za odašiljanje i primanje signala velikih udaljenosti. Poslužitelj je povezan na sučelja informacijskih sustava policije i servisne službe za tramvaje. Za dohvat podataka o udaljenosti postaja i voznom redu koristi postojeću bazu podataka.

2.2.3 Baza podataka

Postoji baza podataka izvan sustava koja sadrži informacije o svim tramvajskim linijama i stanicama. Točnije, svaka linija je reprezentirana svojom rutom, odnosno nizom stanica od polazišta do odredišta. Zapisana su očekivana vremena putovanja između svake dvije susjedne stanice. O svakoj stanici se zna na kojim GPS koordinatama se

nalazi. Postoje specijalne rute koje prometni nadzornik dodjeljuje tramvajskim linijama u slučaju izmjena u prometu.

2.2.4 Displeji

Na tramvajskim stanicama su postavljeni uređaji s ekranom za prikaz podataka. Radio sklop svake stanice namješten je na jedinstvenu frekvenciju odaziva, odnosno sluša samo jednu frekvenciju. Ako neki uređaj želi poslati podatke displeju, to radi na toj frekvenciji. Operacijski sustav koji upravlja displejem može čitati iz memorije antene. Na displeje su instalirane komponente ovog sustava.

3 Zahtjevi na sustav

3.1 Funkcionalni zahtjevi

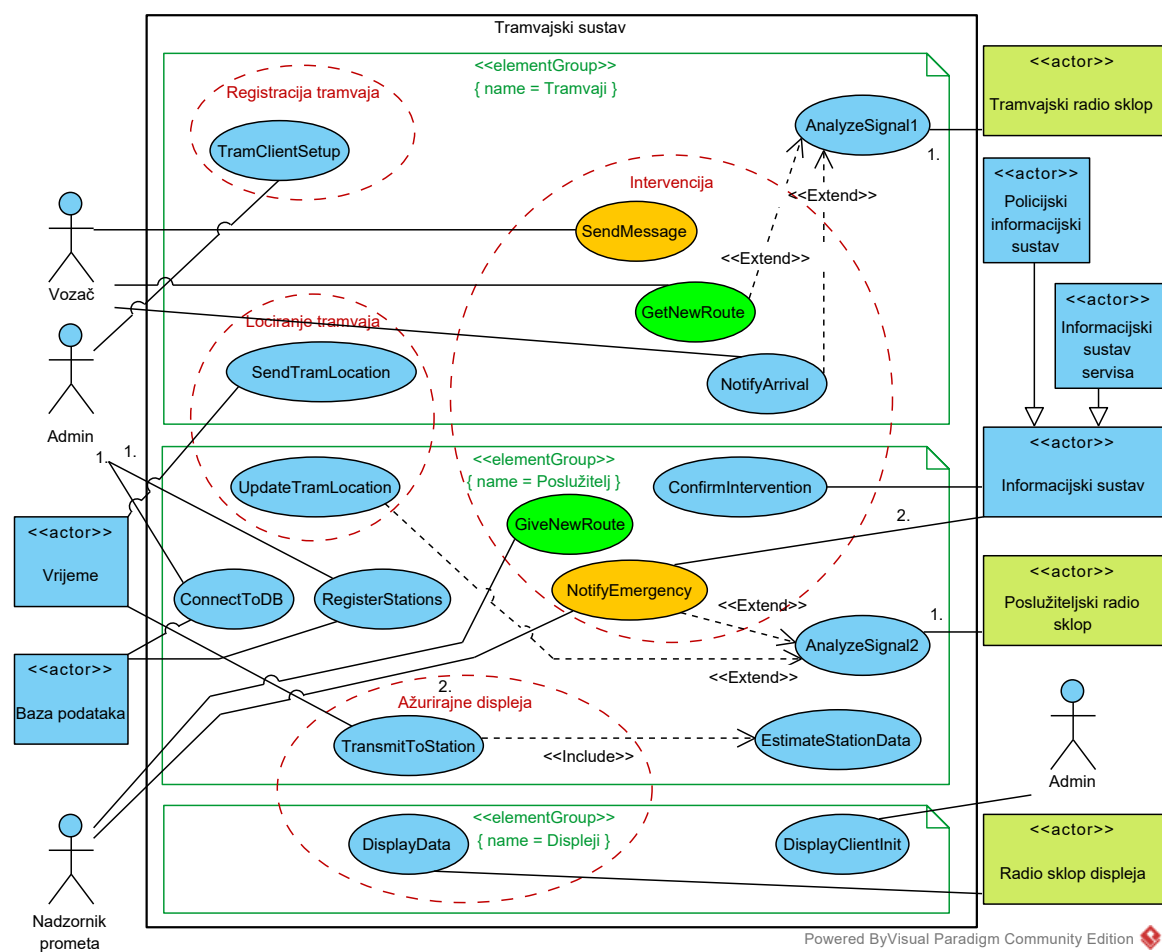
1. Displej mora prikazivati broj linije, odredište za tu liniju i procijenjeno vrijeme do dolaska na stanicu gdje je postavljen. Prikazane informacije su najbržih nadolazećih tramvaja sortiranih po vremenu dolaska u rastućem poretku.
2. Klijent u tramvaju mora periodički dojavljivati svoj položaj centralnom poslužitelju.
3. Vozač mora moći poslati poruku poslužitelju, koji radi odgovarajuću akciju. U slučaju odabira poruke za opasnost, poslužitelj kontaktira policiju i šalje GPS lokaciju tramvaja za koji je potrebna intervencija. Isto tako u slučaju kvara zove ekipu za servisiranje. Svaka poruka odlazi i nadzorniku.
4. Vozač dobiva povratnu informaciju o uzbuni nakon obrade kod odgovarajućeg nadzornog tijela, odnosno izmjene u prometu i potvrdu da je prikladna interventna ekipa poslana na njegovu lokaciju.
5. Vozač na početku radnog vremena dojavljuje poslužitelju koji broj linije je dodeljen dotičnom tramvaju.
6. Administrator postavlja identifikator tramvajskog klijenta kako bi mogao predstavljati fizičku jedinku tramvaja.
7. Administrator povezuje sustav na bazu podataka u kojoj su informacije o tramvajskim rutama.
8. Administrator stvara veze između frekvencija odaziva displeja i stanica na kojima su postavljeni. Ima dopuštenje to zapisati u bazu.
9. Prometni nadzornik dobiva informacije u iznimnim situacijama i donosi odluke o izmjenama u prometu te šalje upute vozačima tramvaja.

3.2 Nefunkcionalni zahtjevi

1. Poslužitelj mora biti u mogućnosti istovremeno primiti pozicije svih tramvaja.
2. Razlika između prikazanog i stvarog vremena dolaska bi trebala biti minimalna.

4 Modeli sustava

4.1 Opis upotrebe sustava



Slika 1: Use case dijagram

4.1.1 Specifikacija use caseova

Use case: SendTramLocation
ID: 1
Brief description: Tramvajski klijent šalje koordinate dobivene preko GPS prijamnika centralnom poslužitelju.
Primary actors: 1. Vrijeme
Secondary actors: None
Preconditions: 1. Prošlo je vrijeme čekanja između dva slanja lokacija.
Main flow: 1. Počinje kad u sustavu završi vrijeme čekanja za ponovno slanje lokacije. 2. Tramvajski klijent dohvaća koordinate. 3. Podaci se upakiravaju u skladu s definiranim komunikacijskim protokolom. 4. Podaci se šalju antenskom sklopovlju. 5. Anteni se signalizira da pošalje podatke.
Postconditions: 1. Antenski sustav u tramvaju je poslao podatke centralnom poslužitelju.
Alternative flows: None

Use case: SendMessage
ID: 5
Brief description: Slanje vozačeve poruke centralnom poslužitelju.
Primary actors: 1. Vozač
Secondary actors: None
Preconditions: 1. Vozač je ispunio tekstualno polje u sučelju za stvaranje poruke. 2. Vozač je odabrao tip poruke u hitnom slučaju. 3. Vozač je pritisnuo tipku za slanje poruke na uređaju u tramvaju.
Main flow: 1. Klijent dohvaća vozačevu tekstualnu poruku i odabrani tip uzbune. 2. Podaci se upakiravaju u skladu s definiranim komunikacijskim protokolom. 3. Podaci se šalju antenskom sklopovlju. 4. Anteni se signalizira da pošalje paket.
Postconditions: 1. Antenski sustav je poslao paket.
Alternative flows: None

Use case: AnalyzeSignal2
ID: 2
Brief description: Sustav analizira signal na poslužiteljskoj anteni kako bi utvrdio što treba činiti.
Primary actors: 1. Poslužiteljski radio sklop.
Secondary actors: None
Preconditions: 1. Radio sklop je primio podatke poslane od nekog tramvajskog klijenta.
Main flow: 1. Počinje kad sustav dobije podražaj od radio sklopa. 2. Sustav dohvaća podatke od antenskog sklopovlja. 3. Sukladno protokolu zapakiravanja dohvaća se kôd namjene primljene poruke. 4. Ako kôd poruke odgovara ažuriranju tramvajske lokacije a) extend(UpdateTramLocation) 5. Ako kôd poruke odgovara notificiranju u slučaju uzbune a) extend(NotifyEmergency)
Postconditions: None
Alternative flows: None

Use case: UpdateTramLocation
ID: 3
Brief description: Ažuriranje lokacije tramvaja u softverskoj reprezentaciji stanja u prometu.
Primary actors: 1. Poslužiteljski radio sklop
Secondary actors: None
Preconditions: 1. U 4. koraku AnalyzeSignal2 kôd namjene poruke odgovara ažuriranju lokacije tramvaja.
Main flow: 1. Raspakirava se paket poruke i dohvaćaju poslane koordinate tramvaja i ID klijenta koji ju je generirao. 2. U sustavu se traži memorijska lokacija odgovarajućeg tramvaja na temelju ID-ja. 3. Spremaju se novi podaci na tu lokaciju.
Postconditions: 1. U memoriju su zapisane nove koordinate tramvaja koji je poslao poruku.
Alternative flows: None

Use case: NotifyEmergency
ID: 4
Brief description: Prosljeđivanje poruke o hitnom slučaju u prometu.
Primary actors: 1. Poslužiteljski radio sklop
Secondary actors: 1. Informacijski sustav 2. Prometni nadzornik
Preconditions: 1. U 4. koraku AnalyzeSignal2 kôd namjene poruke odgovara hitnom slučaju.
Main flow: 1. Raspakirava se paket poruke. Dohvaća se odredište poruke, ID klijenta koji ju je generirao i vozačeva bilješka. 2. Ako je odredište policija a) Generira se poruka u format koji odgovara policijskom sustavu. Uključuje koordinate, opis uzbune i vozačevu bilješku. b) Korištenjem sučelja policijskog informacijskog sustava šalje se generirana poruka. 3. Ako je odredište servis a) Generira se poruka u format koji odgovara sustavu servisa. Uključuje koordinate, opis uzbune i vozačevu bilješku. b) Korištenjem sučelja sustava servisa šalje se generirana poruka. 4. Prometnom nadzorniku šalje se kopija poruke.
Postconditions: 1. Informacijski sustav kojem je poslana poruka javlja uspješno slanje. 2. Nadzornik potvrđuje da je primio poruku.
Alternative flows: 1. FailedSend

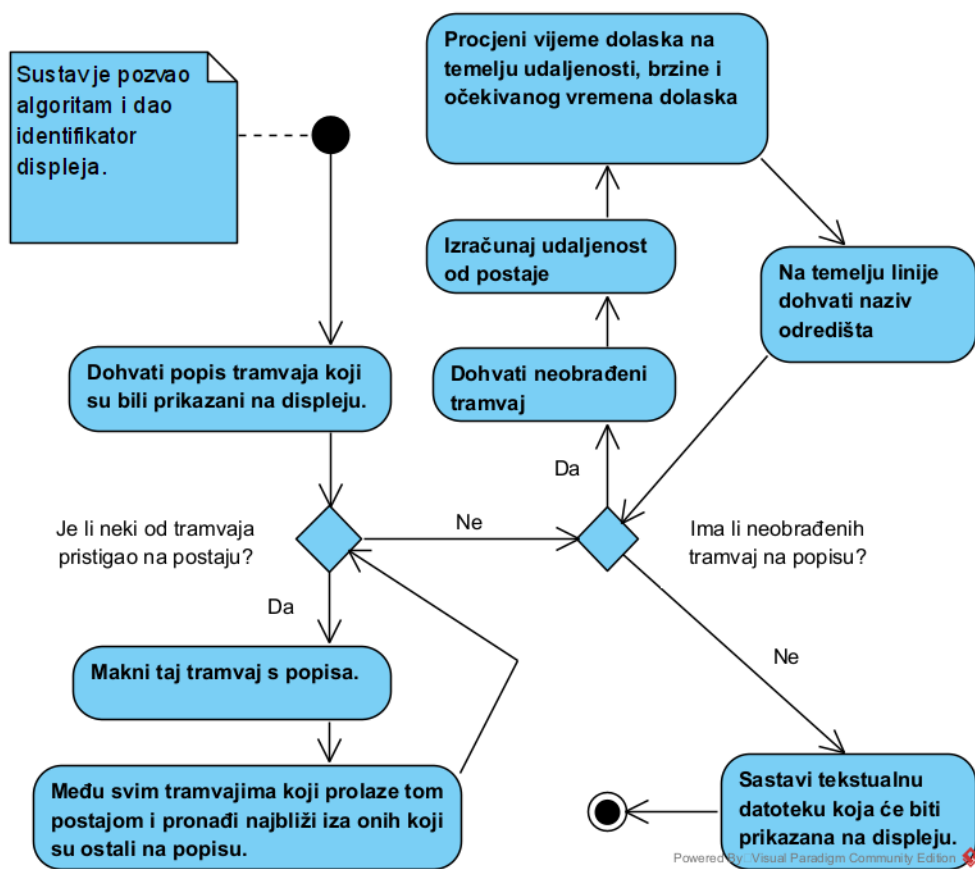
Use case: FailedSend
ID: 4.1
Brief description: Ponovni pokušaj slanja poruke interventnim službama.
Primary actors: 1. Informacijski sustav
Secondary actors: 1. Informacijski sustav
Preconditions: 1. Informacijski sustav je vratio poruku o neuspješnom slanju.
Main flow: 1. Ponavljaju se koraci 2. i 3. iz NotifyEmergency
Postconditions: 1. Poslana je poruka informacijskom sustavu.
Alternative flows: 1. FailedSend

Use case: DisplayData
ID: 6
Brief description: Displej prikazuje podatke primljene preko antene.
Primary actors: 1. Radio sklop displeja
Secondary actors: None
Preconditions: 1. Radio sklop je primio podatke od centralnog poslužitelja.
Main flow: 1. Počinje kad sustav dobije podražaj od antenskog sklopovlja. 2. Iz memorije antene se dohvaća primljeni paket. 3. Paket se raspakirava sukladno definiranom protokolu. 4. Na ekranu se preko sučelja OS-a displeja daje naredba za prikazivanje liste tramvaja.
Postconditions: 1. Na ekranu su prikazane informacije o tramvajima.
Alternative flows: None

Use case: TransmitToStation
ID: 7
Brief description: Centralni poslužitelj šalje ažurirane podatke odgovarajućem displeju.
Primary actors: 1. Vrijeme
Secondary actors: None
Preconditions: 1. Završilo je vrijeme čekanja u ciklusu slanja.
Main flow: 1. Počinje kad završi vrijeme čekanja između dva slanja podataka istom displeju. 2. include(EstimateStationData) 3. Dohvaćaju se ažurirani podaci. 4. Pakiraju se sukladno komunikacijskom protokolu. 5. Traži se podatak o frekvenciji odaziva antene displeja. 6. Radio sklopovlju poslužitelja se predaje paket i frekvencija. 7. Signalizira se anteni da pošalje paket.
Postconditions: 1. Radio sklop je poslao poruku.
Alternative flows: None

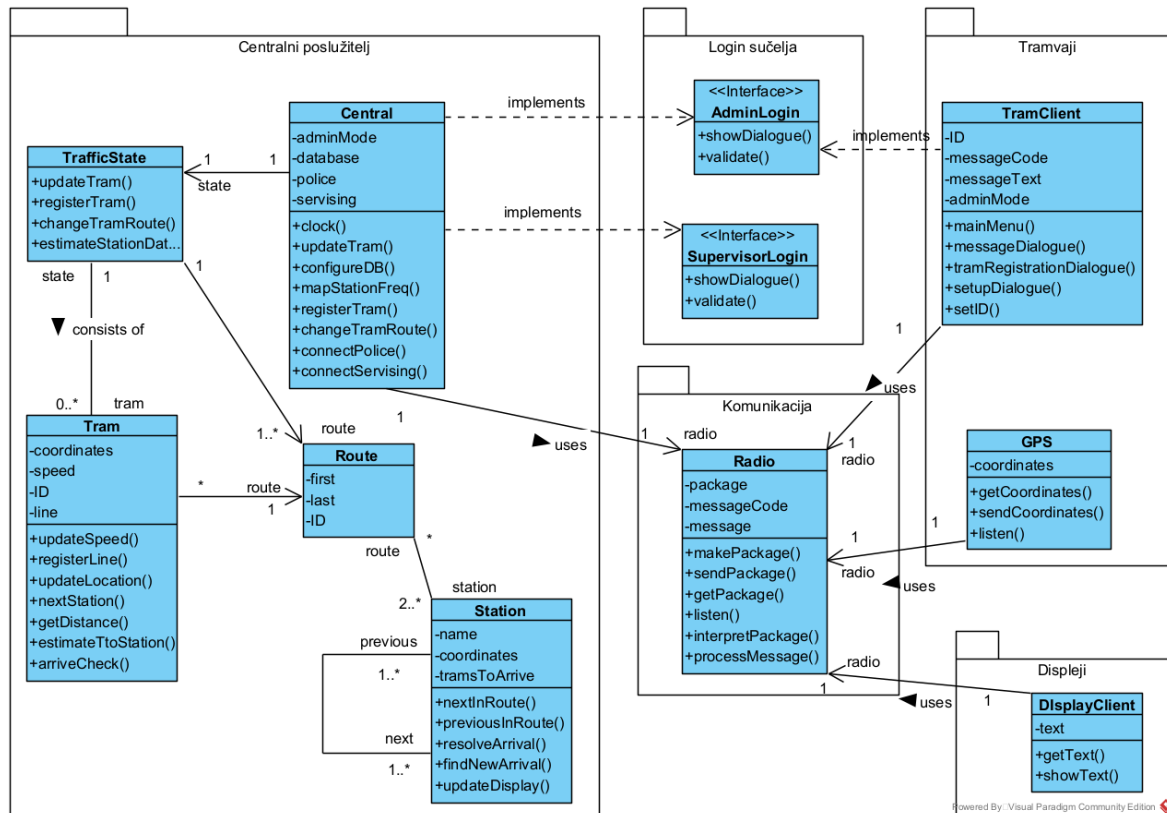
Use case: EstimateStationData
ID: 8
Brief description: Računa se koje linije i vremena moraju biti prikazane na nekom displeju prilikom sljedećeg osvježavanja.
Primary actors: 1. Vrijeme
Secondary actors: None
Preconditions: 1. Sustav je izrazio potrebu za ažuriranjem informacija o tramvajima za neku postaju.
Main flow: 1. Prikuplja se ID postaje. 2. Signalizira se početak računanja. 3. Počinje algoritam za procjenu dolazaka za tu stanicu. Slika 2 4. Završava algoritam procjene.
Postconditions: 1. U sustavu postoji datoteka s linijama i vremenima koju treba prikazati na displeju.
Alternative flows: None

4.1.2 Activity dijagrami



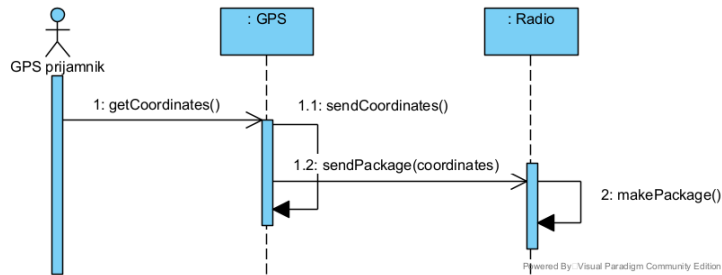
Slika 2: Algoritam procjene vremena dolazaka

4.2 Klase



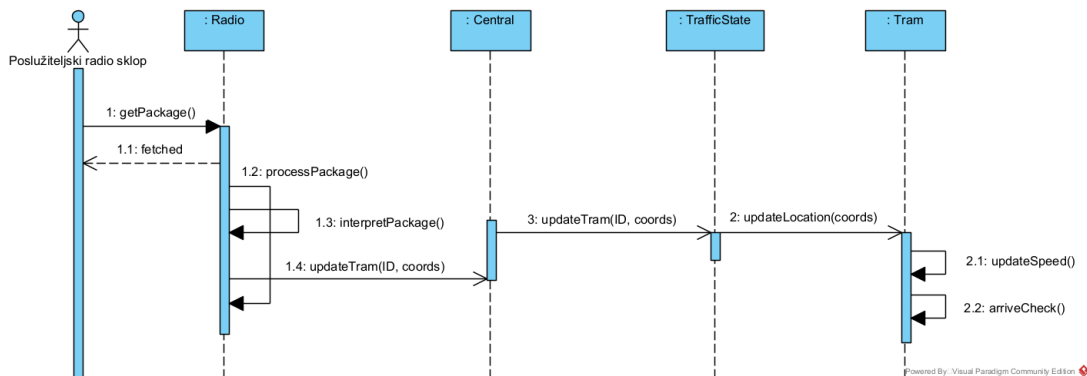
Slika 3: Class diagram

4.2.1 Sequence dijagrami



Slika 4: SendTramLocation

Slika 4 opisuje realizaciju use case-a u kojem GPS prijamnik poziva metodu `getCoordinates()` koja iz memorije u kojoj su koordinate dohvaća podatke. Objekt klase `GPS` šalje koordinate iz svog atributa `coordinates` pozivajući u svojoj instanci `Radio` klase `sendPackage(coordinates)` u koju je uključen poziv `makePackage()` kodiran u skladu s protokolom.

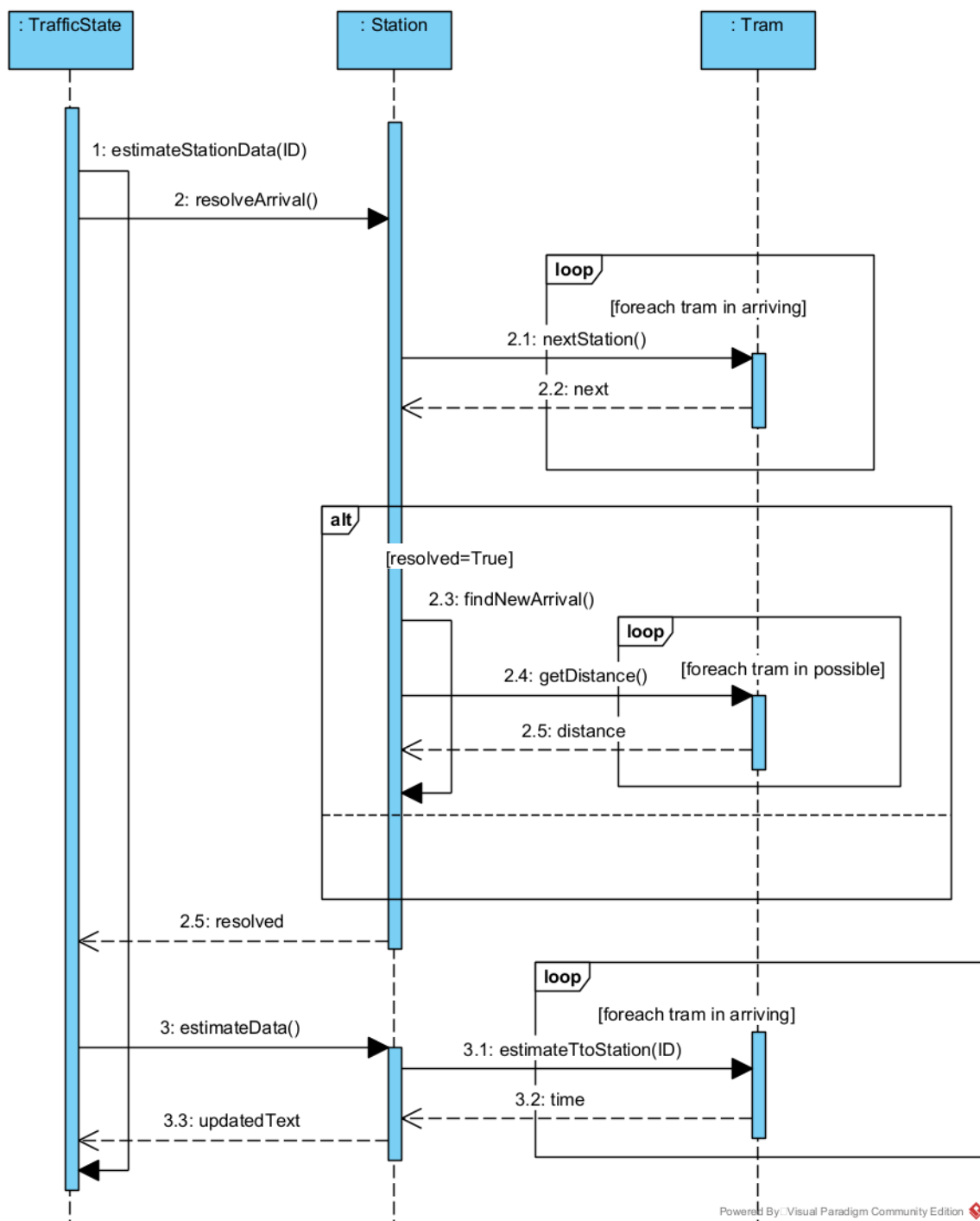


Slika 5: UpdateTramLocation

Slika 5 prikazuje situaciju kada je paket koji sadrži koordinate poslan centralnom poslužitelju. Njegov radio sklop u `(central:Central).(radio:Radio)` poziva metodu dohvata paketa iz memorije antene. U klasi `Radio` metoda `processPackage()` obavlja sve potrebno za analizu paketa i popratne akcije. `interpretPackage()` raspakirava paket sukladno protokolu. U ovom use case-u se poruka interpretirala kao dostava lokacijskih podataka tramvaja. To znači da `radio` kao atribut od `central` u njemu poziva `updateTram()` s argumentima identifikatora tramvaja i novim koordinatama. Odgovarajuća instanca tramvaja sprema nove koordinate, računa brzinu i provjerava je li tramvaja stigao na stanicu.

Prije ažuriranja informacija na displejima potrebno je procijeniti koliko daleko su se pomaknuli tramvaji i treba li neke maknuti s popisa jer su već pristigli. Slika 6 prikazuje kako to postići. Uz identifikator displeja odgovarajuća `stanica:Station` prvo provjerava je li neki od tramvaja koji su u prethodnom ciklusu bili nadolazeći pristigao. To

se realizira prolaskom po svim instancama klase `:Tram` koji su bili na popisu. Njihovom metodom `nextStation()` saznajemo je su li već pristigli. Ako je za barem jedan bilo istina, traže se novi. Postoji popis tramvaja `possible` sa svim tramvajima koji prolaze promatranom stanicom. Među njima se pomoću `getDistance()` može naći najbliži izuzev one i dalje na popisu. Nakon osvježavanja popisa `arriving` svaki tramvaj ima svoju `estimateTtoStation(ID)` metodu za procjenu vremena. To uključuje računanje s podacima o očekivanom vremenu iz baze podataka, trenutnoj brzini i udaljenosti od stanice. Udaljenost se računa u metrici tračnica, odnosno nije zračna udaljenost. Konačno, generira se tekstualni zapis za prikaz na displeju. To će se zapakirati pomoću metoda iz `radio:Radio` objekta i poslati odgovarajućem displeju.



Slika 6: EstimateStationData

5 Oblikovanje

U ovom dijelu naglasak je na fazi oblikovanja softverskog produkta. Bavimo se profinjavanjem klasa, sequence dijagrama i radimo pogled na sustav kroz njegove komponente.

5.1 Class dijagram

Slika 7 predstavlja nadogradnju sustava pogodnu za implementaciju. Sve veze su profinjene. Uvedena su neka sučelja radi preglednosti. Napravljena je poveznica s kontekstom sustava. Vanjski sustavi i paketi bitni za funkcionalnosti tramvajskog su obojani žutom bojom.

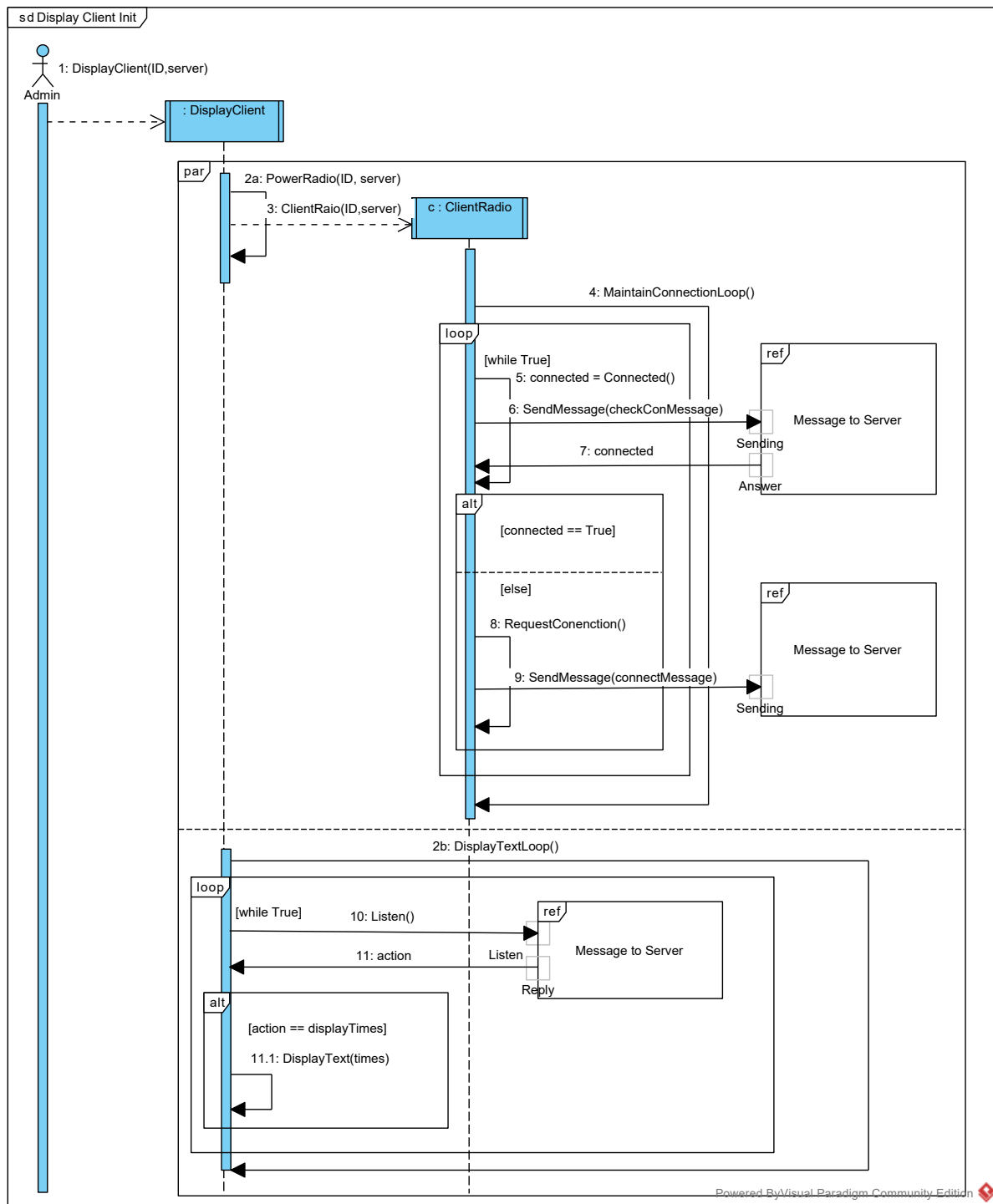
5.2 Sequence dijagrami

5.2.1 DisplayClientInit, DisplayData

Dijagram 8 prikazuje što se sve događa kad radnik koji postavlja displej pokrene sustav. Svakom displeju je u bazi podataka preko ID-ja dodijeljen naziv stanice i GPS lokacija. Dakle, kod inicijalizacije je dovoljno navesti taj ključ i neki mrežni identifikator servera s kojim će pokušati komunicirati. Postoje dva paralelna procesa. Jedan se brine o održavanju veze sa serverom, a drugi dohvaćanju podataka koje treba prikazati na ekranu. Referenca *Message to Server* se odnosi na dijagram 9. Kroz gate *Sending* prikazan je proces slanja poruke serveru. *Listen* je dio procesa slanja poruke, ali može se zasebno koristiti kao što je slučaj kod prikazivanja informacija na displeju.

5.2.2 TramClientSetup, SendMessage, UpateTramLocation

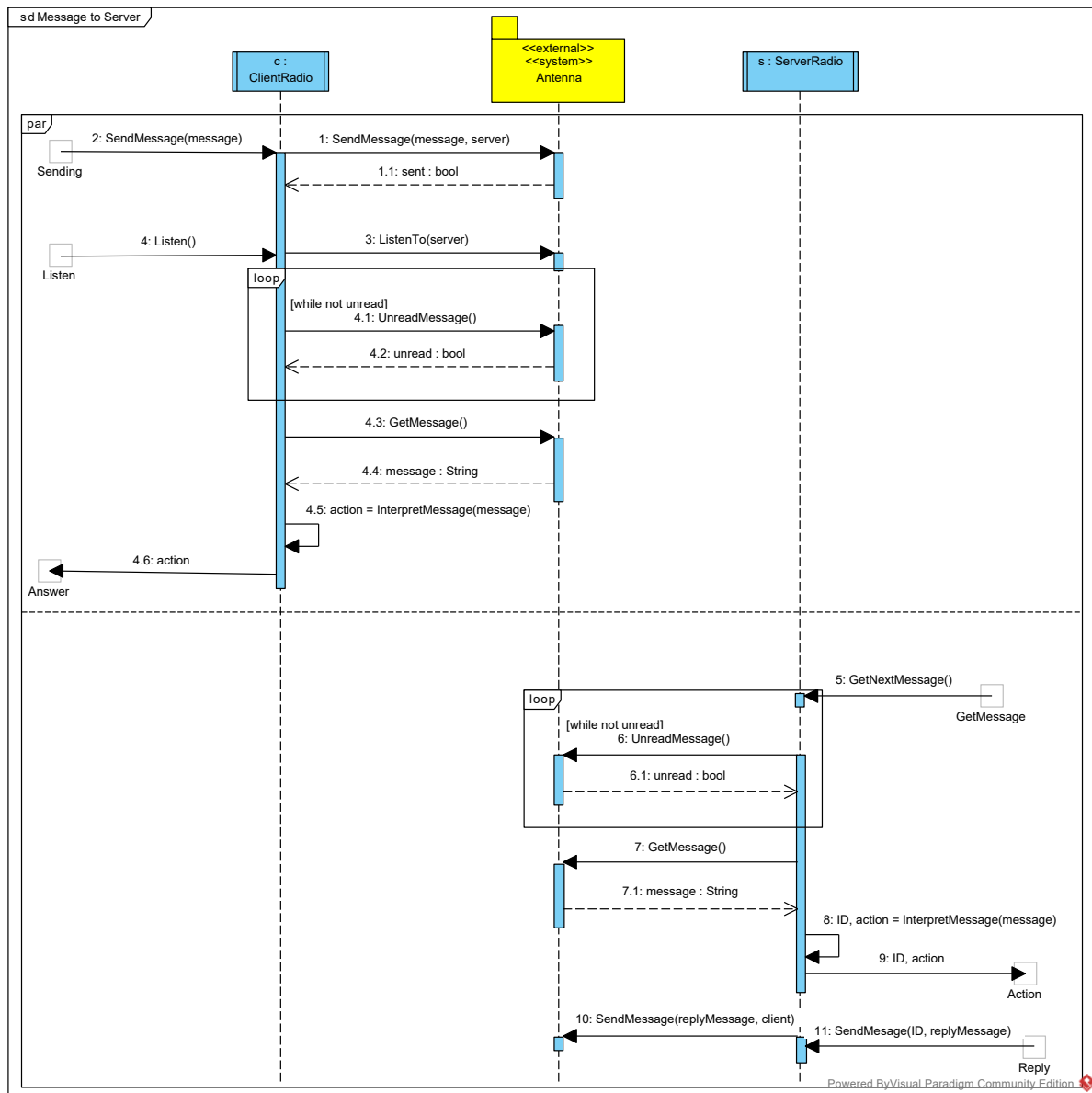
Na sličan način bi se oblikovao i životni ciklus tramvajskog klijenta. Samo što bi postojali dodatni asinkroni procesi za interakciju s osobljem u tramvaju — *DriverMessage*, *CheckLoginInfo* i slušanju uputa servera — *NewRoute*, *InterventionNotification*. Još jedan paralelni proces bi dohvaćao lokaciju pomoću vanjskog GPS sučelja i slao poruku serveru sa svojom lokacijom. Nakon primanja poruke, server bi ažurirao podatke o tramvaju sukladno dobivenim podacima.



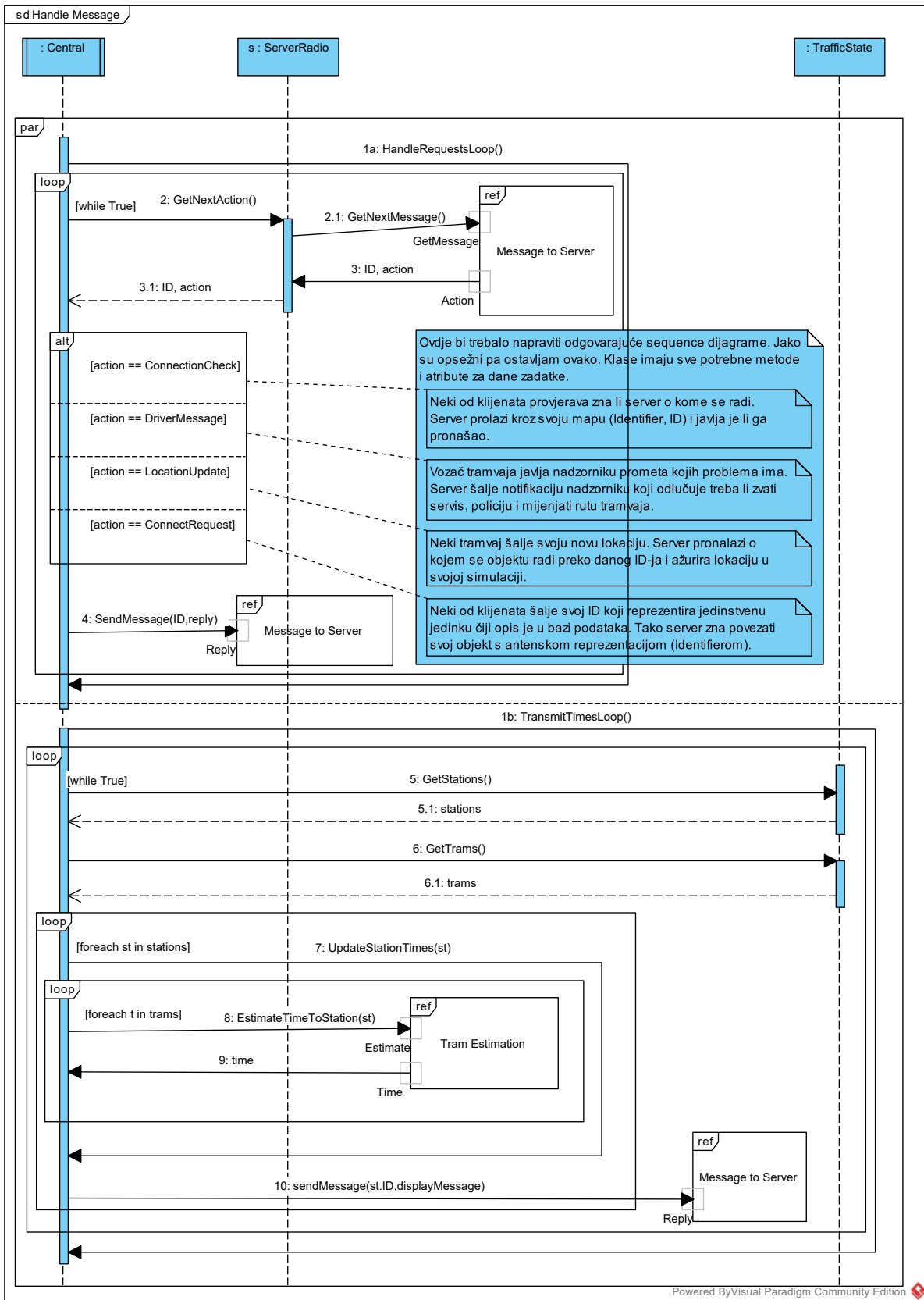
Slika 8: Život displeja

5.2.3 AnalyzeSignal, TransmitToStation, EstimateStationData

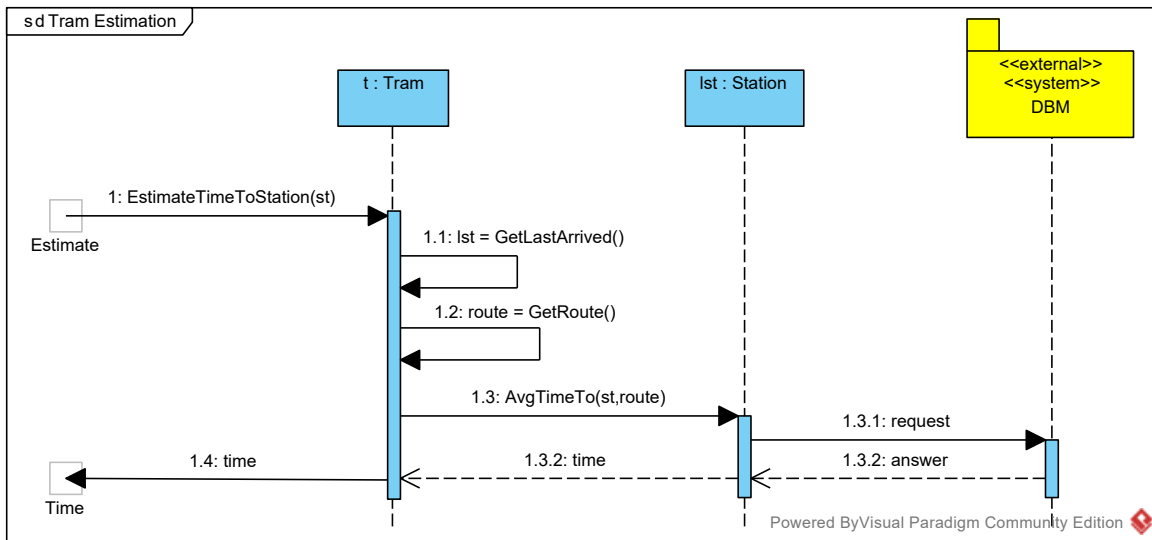
Na dijagramu 9 možemo vidjeti kako klijentski i serverski radio izmjenjuju poruke koristeći sučelje sustava *Antenna* koji se bavi tehnikacijama izmjene podataka radiovalovima. U centrali su dva istovremena procesa prikazana na dijagramu 10. Jedan se brine za obradu zahtjeva klijenata ovisno o tome kakvu akciju oni sugeriraju. Nakon odgovarajuće obrade se nazad pošiljatelju daje odgovor. Koristi se *Reply* gate na slici 9. Klijentski radio prima tu poruku ako je pozvana metoda `Listen()` i interpretira kako bi znao što raditi. U slučaju displeja to može biti `DisplayText` ili odgovor `Yes` na upit o povezanosti sa serverom. Nedostatak odgovora `Yes` klijent shvaća kao odgovor „no”. Drugi proces prolazi po svim stanicama u svojem popisu aktivnih. Svaki prolazak kroz petlju provjerava je li neki novi klijent (koji može biti displej ili tramvaj) zatražio spajanje sa serverom. Nakon što su svi pobrojani prolazi se po svim mogućim stanicama i tramvajima. Za fiksnu stanicu `s:Station` se kroz taj postupak saznaje kojim tramvajima je na ruti. Ovdje proces prolazi kroz gate *Estimate* čija obrada je prikazana na slici 11. Svaki od spomenutih tramvaja u ovisnosti o zadnjoj stanici na koju je pristigao i ruti šalje zahtjev stanici. Objekt stanica ima pristup bazi podataka u kojoj je dovoljno informacija za izračun. Korištenjem sučelja database managera i računom s tim podacima pronalazi traženo vrijeme i ono se šalje korijenu zahtjeva, odnosno centrali koja preko radija šalje odgovarajućem displeju podatke za prikaz.



Slika 9: Komunikacija



Slika 10: Aktivnosti poslužitelja

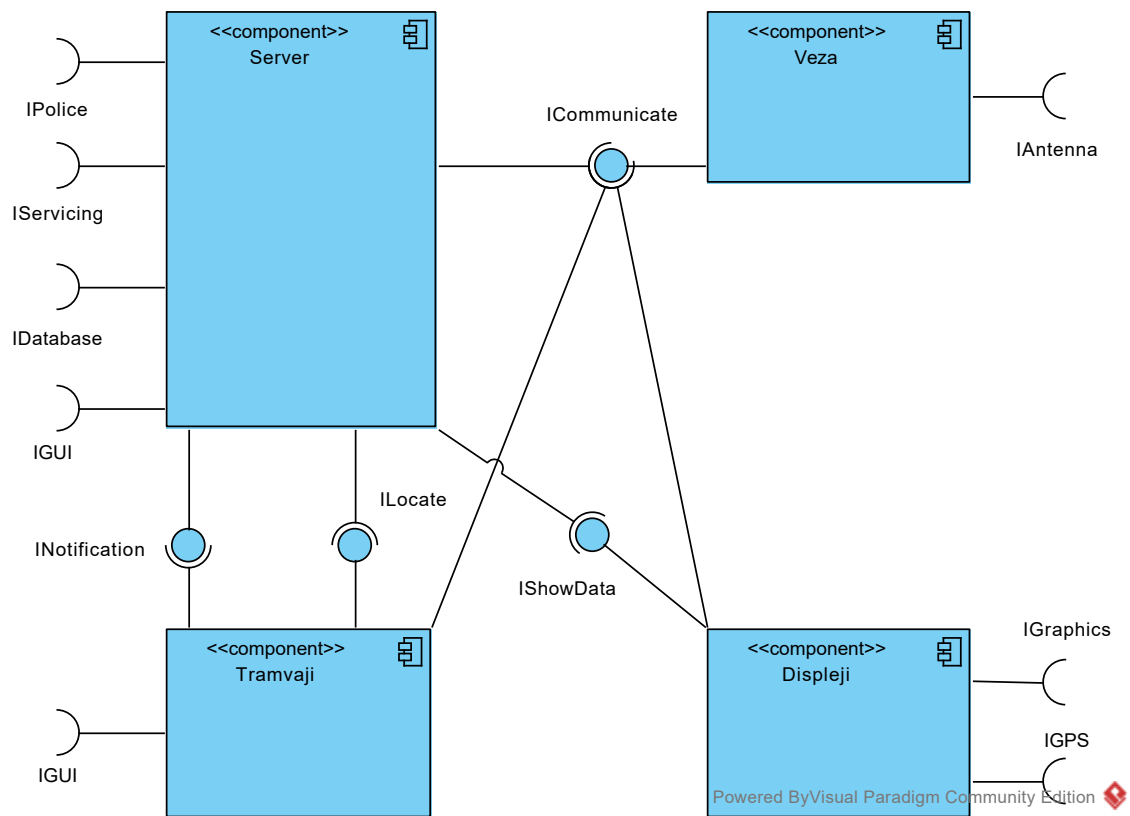


Slika 11: Procjena vremena

5.3 Component dijagram

Na dijagramu 12 je pojednostavljeni prikaz sustava. Kako bi u potpunosti funkcionirao traži realizaciju sučelja vanjskih sustava — policije, servisa, baze podataka, GPS uređaja, antene i grafičkih instrumenata uređaja na kojima je instaliran sustav.

- `ICommunicate` specificira standardne metode slanja poruka na neku adresu (identifikator) preko mreže.
- Preko `IShowData` server prikazuje podatke o dolascima tramvaja na displejima.
- Sučelje `ILocate` omogućuje serveru ažuriranje podataka o tramvajima u prometu.
- Pomoću `INotification` tramvajski klijenti mogu slušati obavijesti i upute poslužitelja.



Slika 12: Komponentni prikaz