

ETF-RAILWAYS

Dokumentacija



Amina Salčin
Lejla Ramić

Opis teme

EtfRailways je aplikacija čiji primarni zadatak je upravljati, olakšati i rastetetiti rad same željezničke stanice i njenih uposlenika.

Cilj naše aplikacije je ponuditi korisniku sve potrebne informacije za samostalnu organizaciju prevoza do željene destinacije, ukoliko se korisnik odluči da kao prevozno sredstvo koristi voz. Gužve i redove čekanja na kupovinu karata se mogu izbjeći koristeći našu aplikaciju, koja je praktična i jednostavna za korištenje.

Aplikacija nije namijenjena samo za osobe koje žele koristiti uslge EtfRailways-a, već i za osoblje naše kompanije u cilju lakše organizacije i uvida u sve aktivnosti unutar EtfRailways-a. EtfRailways nudi velike olakšice svim korisnicima, od onih koji kupuju kartu, do zaposlenika, te i same uprave koja želi modernizirati svoje poslovanje i učiniti svoje usluge jednostavnijim i boljim.

Funkcionalnosti

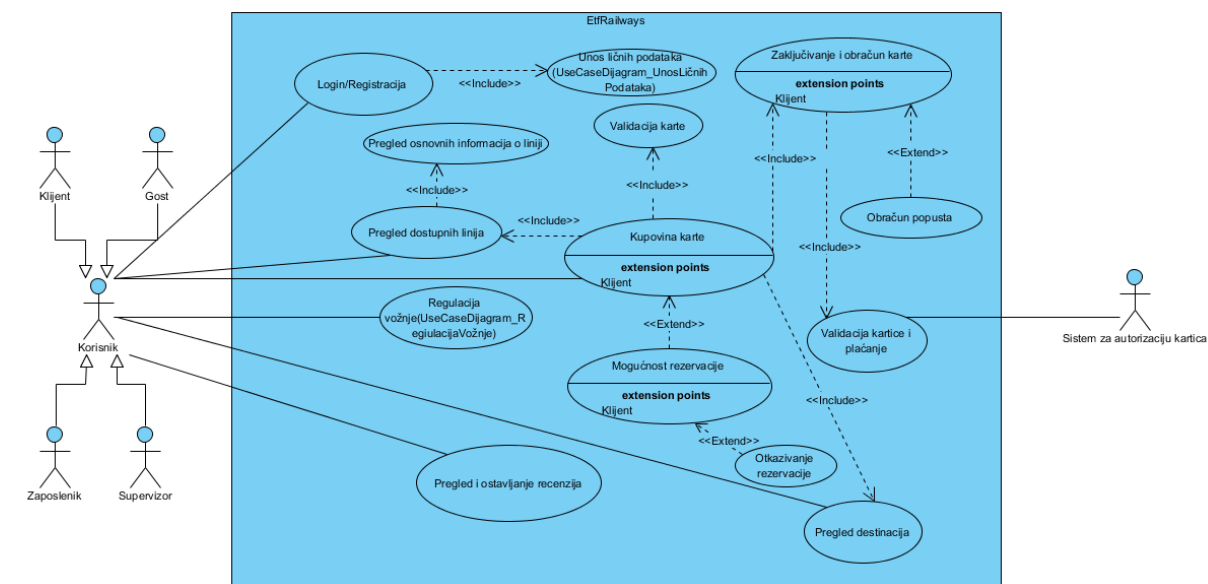
Predviđene funkcionalnosti naše aplikacije su:

- Mogućnost prijave na sistem sa različitim privilegijama (zaposlenik, korisnik aplikacije, gost, supervizor)
- Mogućnost registrovanja novog korisnika
- Kreiranje arhive registrovanog korisnika
- Mogućnost pregleda svih vožnji koje su ponuđene sa polascima i očekivanim dolascima
- Mogućnost dodavanja novih zaposlenika
- Mogućnost brisanja zaposlenika u slučaju otkaza
- Mogućnost registrovanja novih vožnji
- Mogućnost brisanja vožnje
- Mogućnost pregleda svih destinacija, do kojih nudimo prevoz, radi boljeg upoznavanja korisnika sa destinacijom koju izaberu
- Kupovina karte
- Mogućnost čitanja knjige žalbi i pohvala svih korisnika, kao i upisivanje utisaka korisnika
- Mogućnost prikaza informacija o našoj željezničkoj stanici(statistički podaci, informacije o vozovima...)

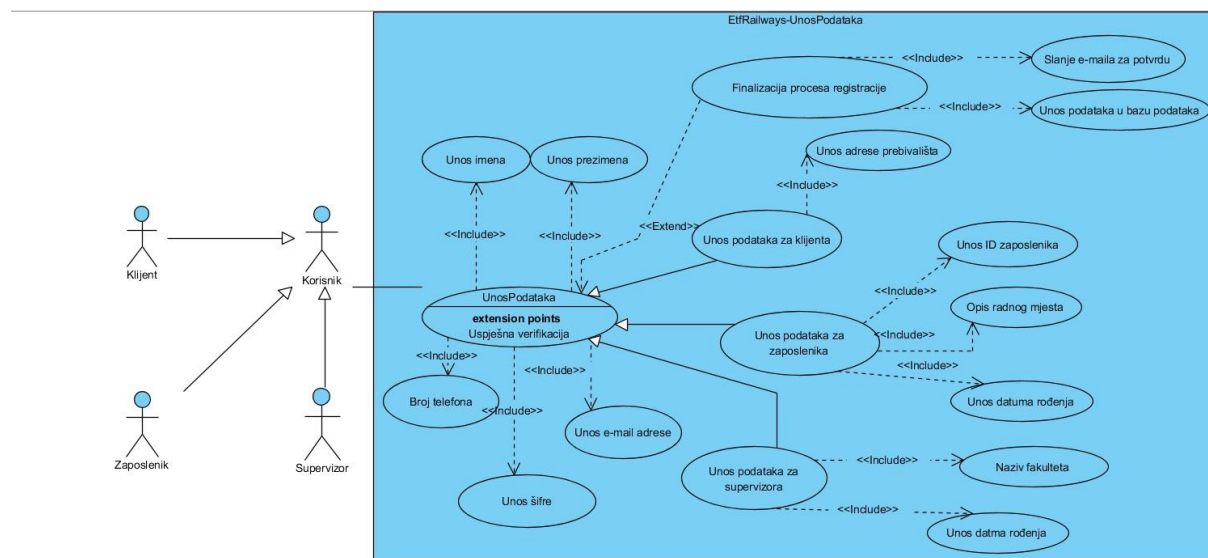
Dijagrami

Dizajnirani dijagrami tokom projekta su:

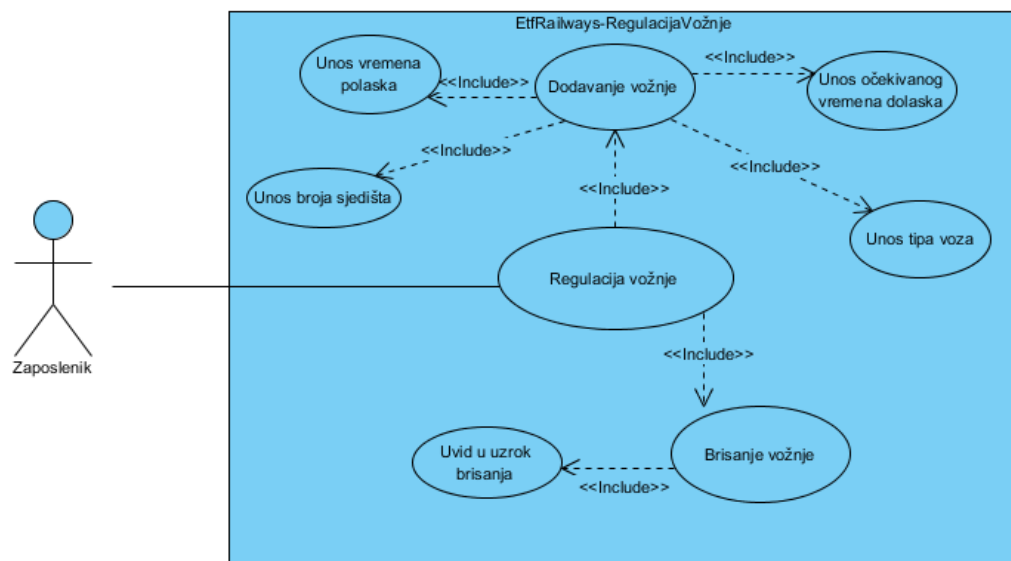
1. Use case diagram



glavni USE CASE dijagram

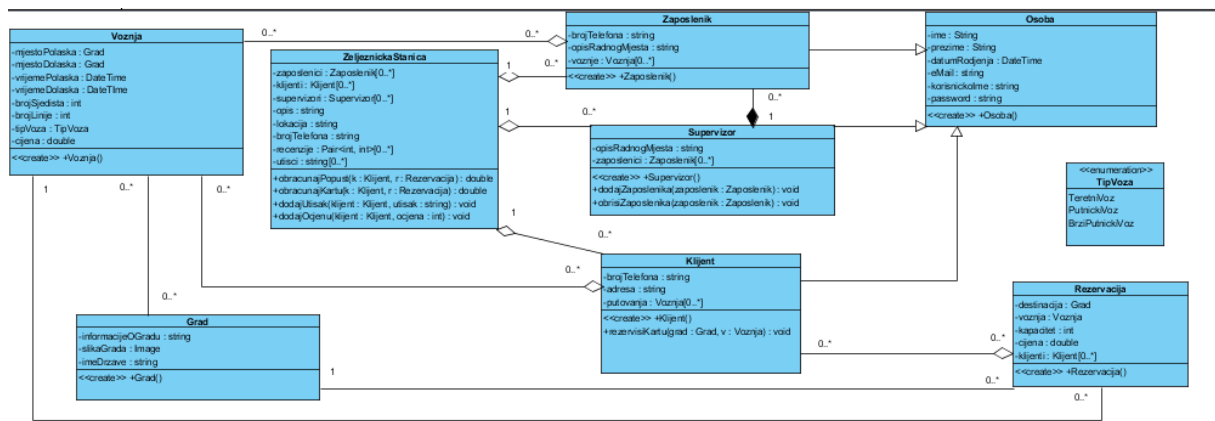


USE CASE za registraciju

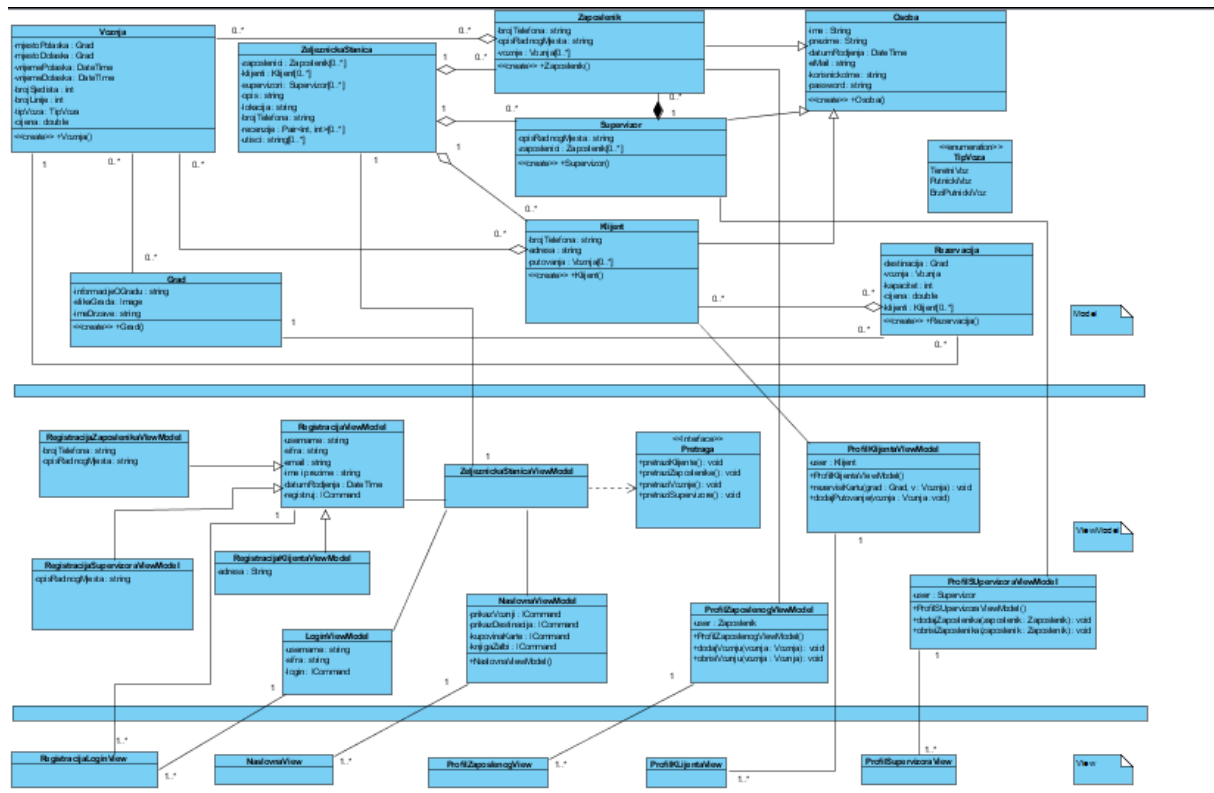


USE CASE za regulaciju vožnje

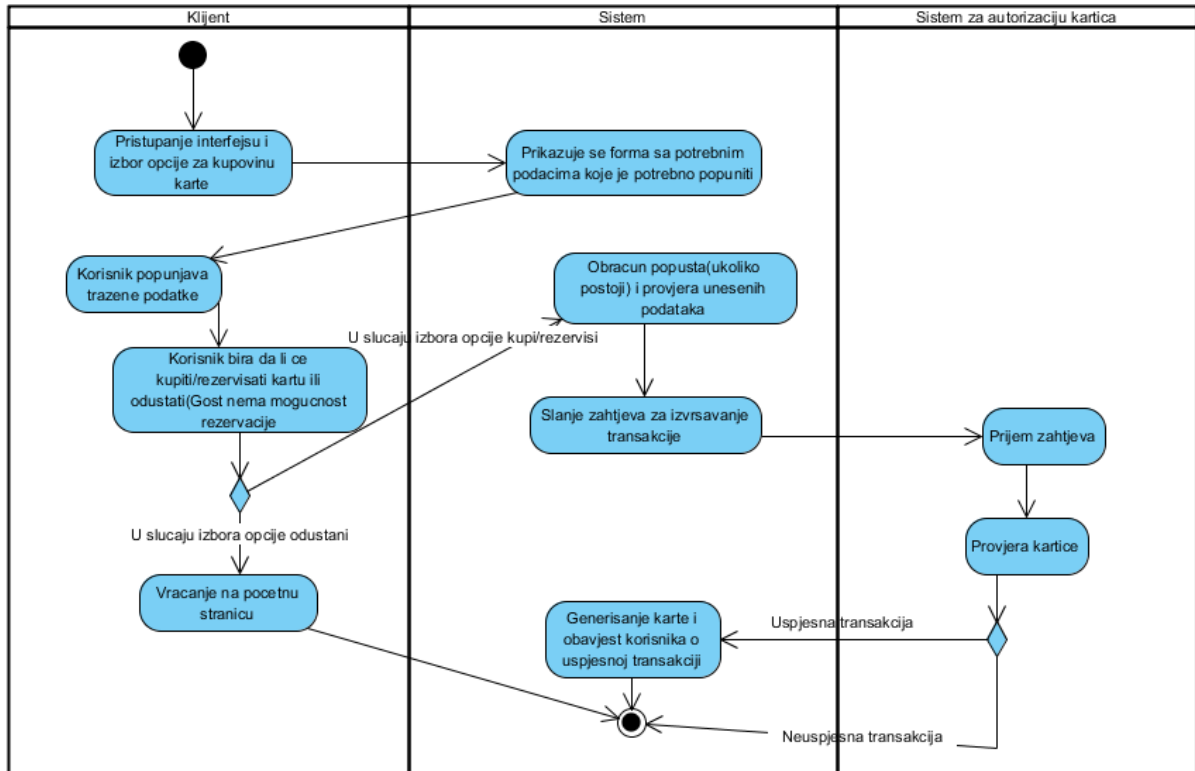
2. Dijagram klasa



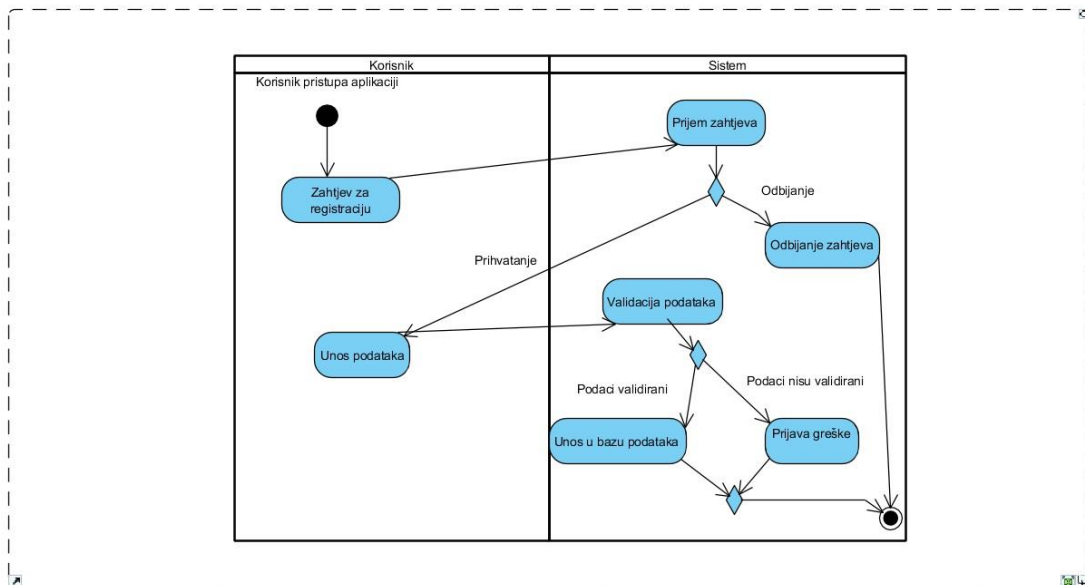
3. Redizajnirani dijagram klasa- MVVM



4. Dijagram aktivnosti

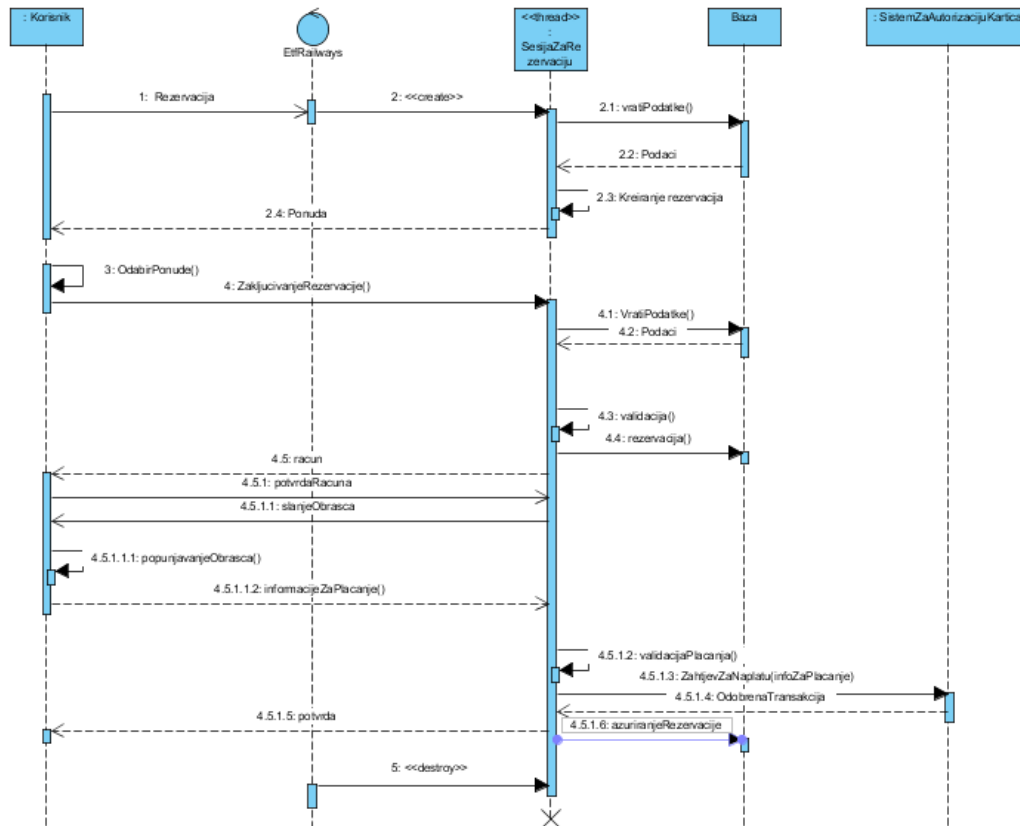


Dijagram aktivnosti za kupovinu karte



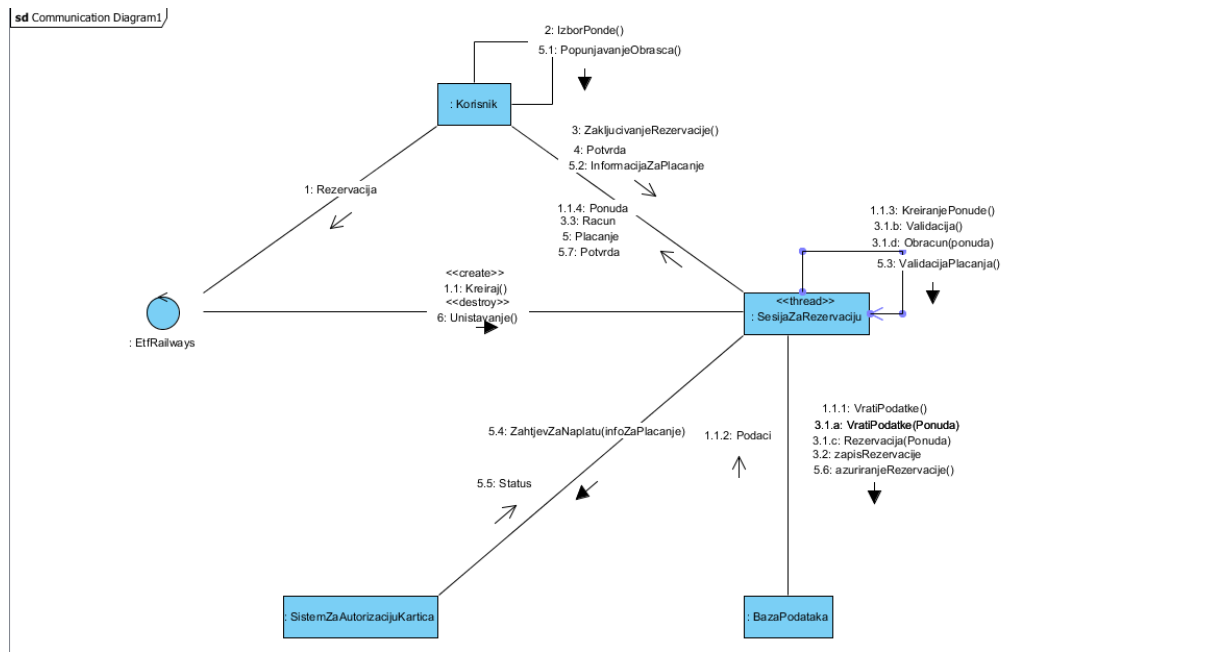
Dijagram aktivnosti za registraciju

5. Dijagram sekvenci



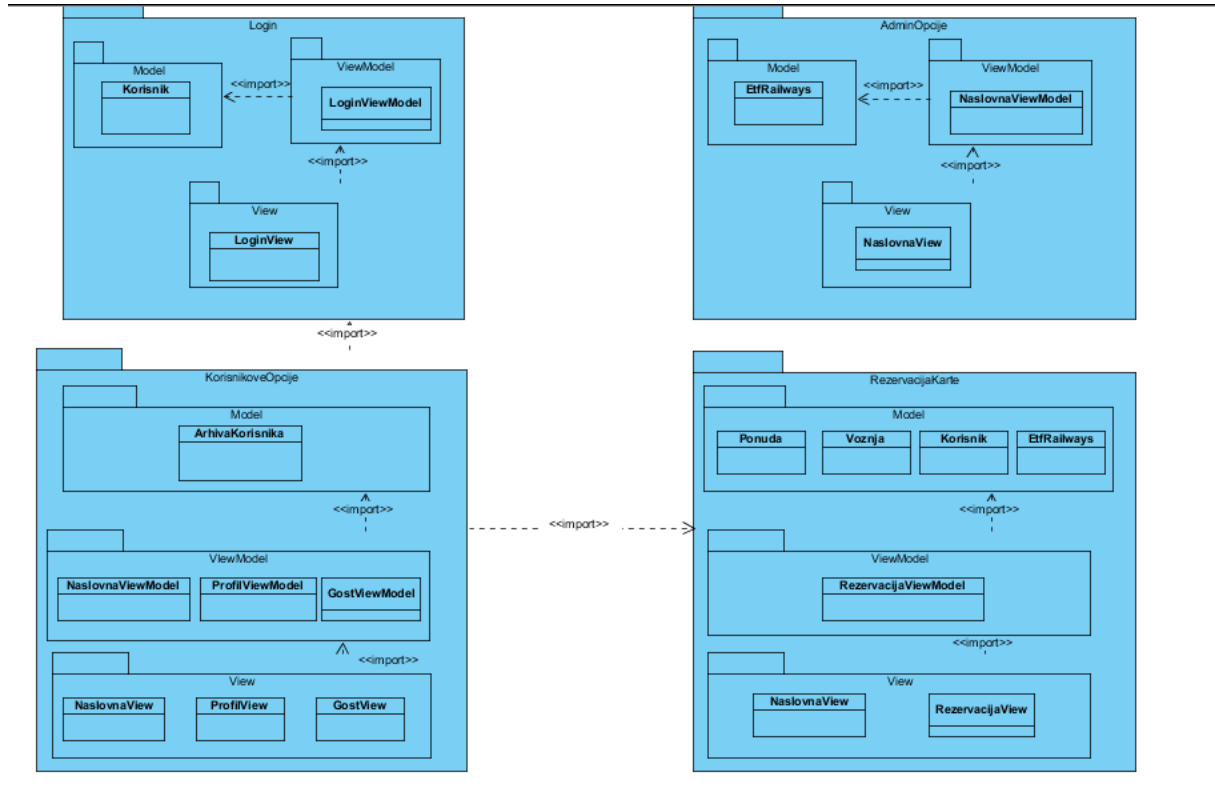
Dijagram sekvence za rezervaciju

6. Dijagram komunikacije

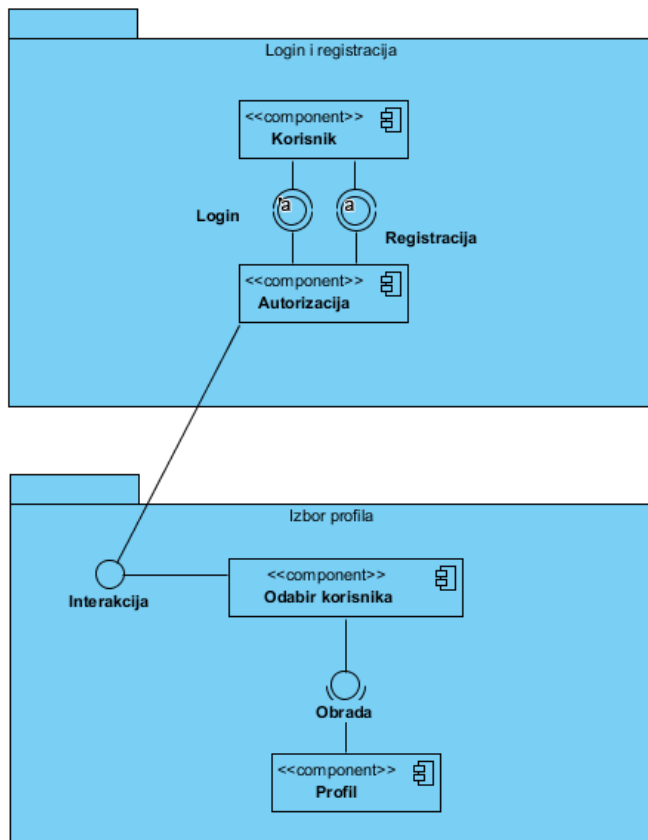


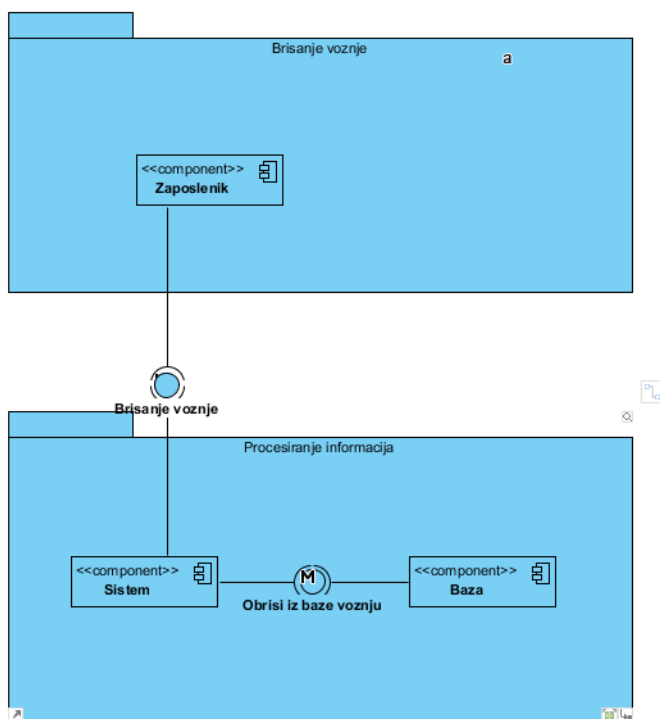
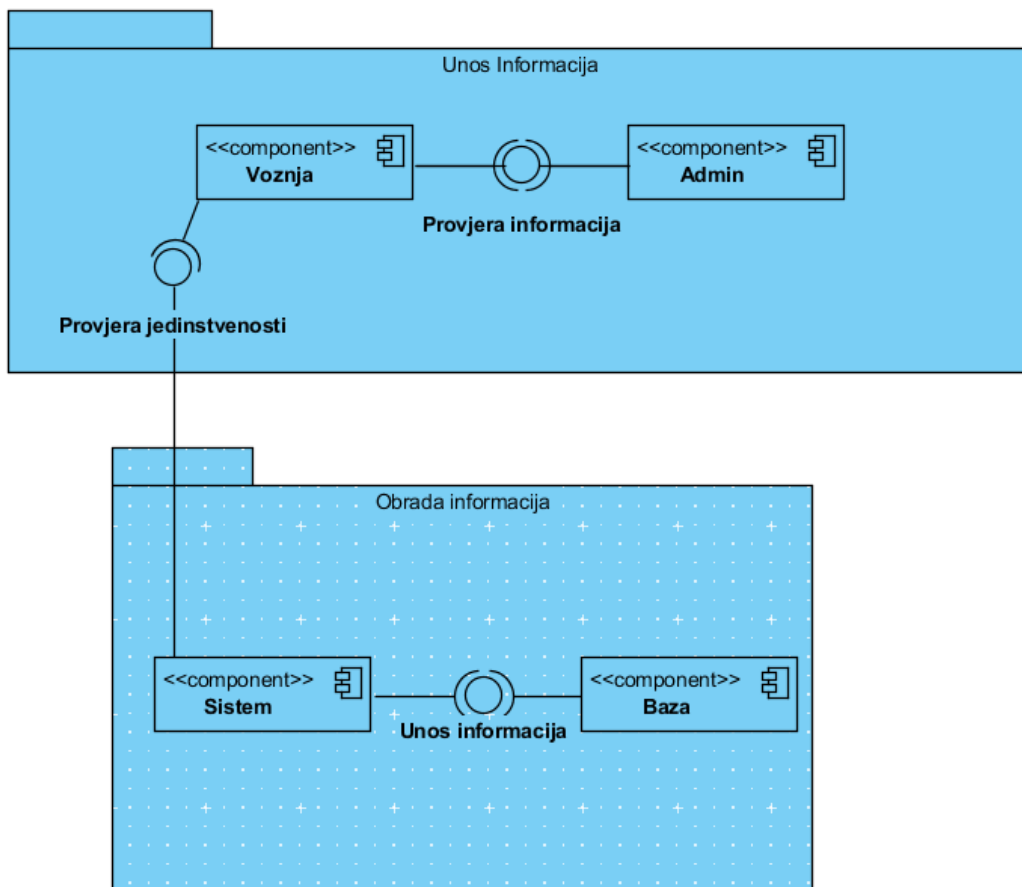
Dijagram komunikacije za rezervaciju

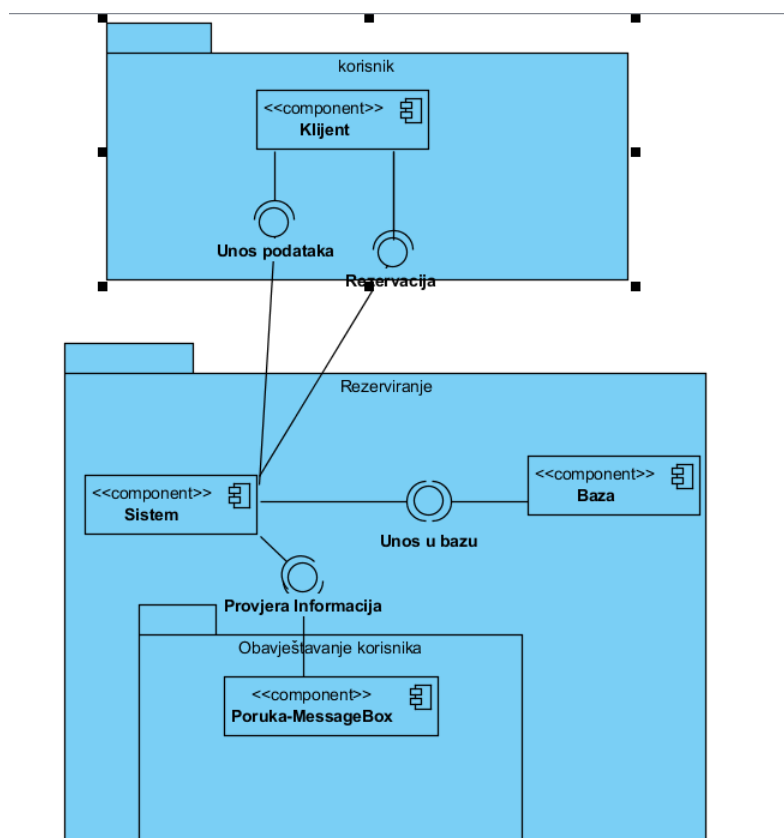
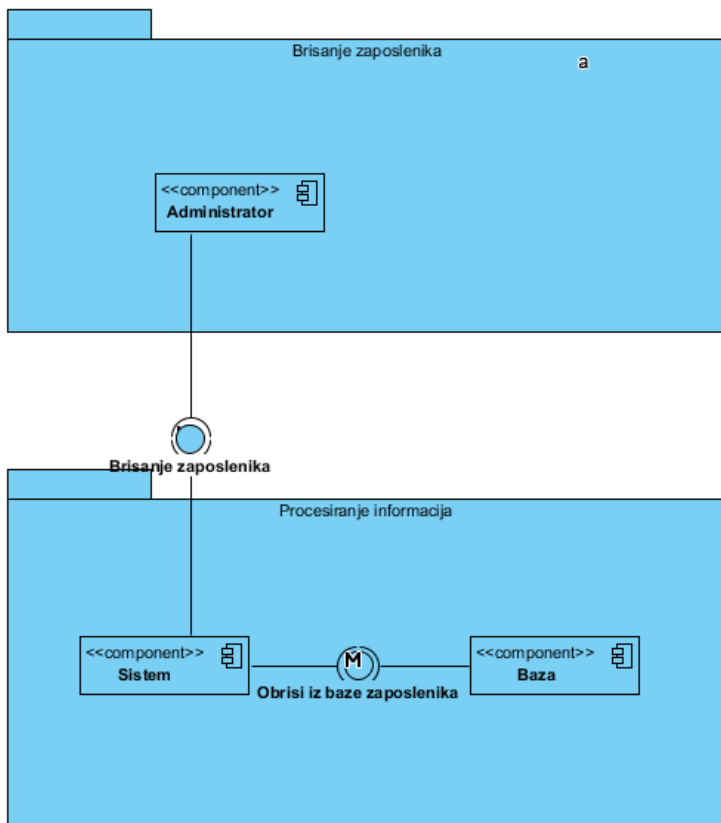
7. Dijagram paketa



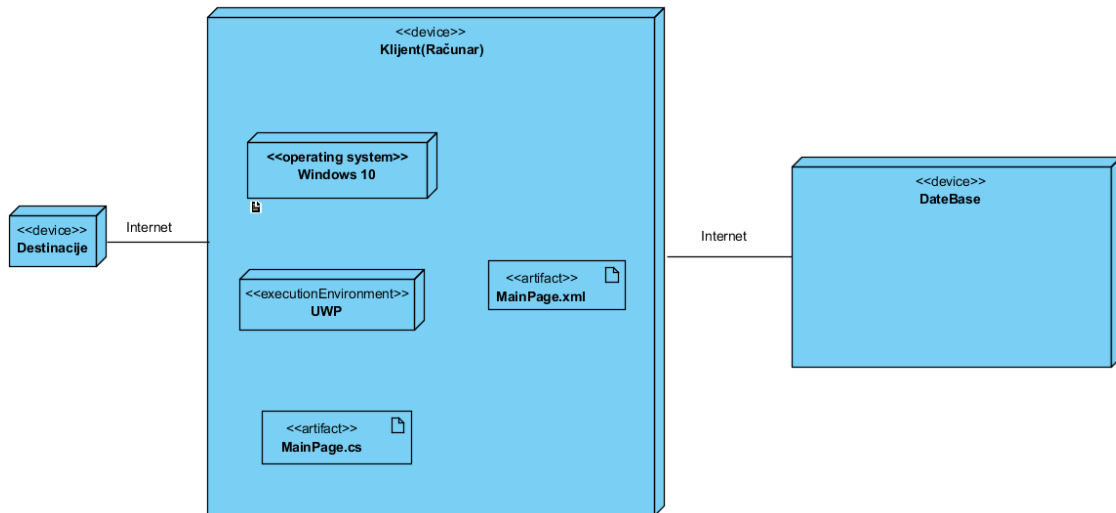
8. Dijagram komponenti







9. Dijagram raspoređivanja



Izvještaj o radu

Projekat je rađen timski. Svake sedmice su se odrađivali projektni zadaci predviđeni za tu sedmicu. Podjela rada je bila fer, i svako je radio svoje predviđene zadatke. Zadaci su rađeni zajedno, nalazeći se na određenom mjestu, ili od kuće putem Skype-a.

Svi traženi dijagrami su urađeni, neki i u više varijanti(npr USE CASE dijagram za kupovinu karte, regulaciju vožnje...).

UWP dio aplikacije, koji je zahtjevao implementaciju administrativnog dijela aplikacije je urađen do kraja. Koristio se MVVM patern, koji se bazira na Model-View-ViewModel organizaciji. Nakon toga aplikacija je povezana sa bazom(Rad sa Azure bazom podataka).

Pored administrativnog dijela aplikacije, u UWP smo implementirale i ostali dio naše aplikacije. Osim mogućnosti registracije i smještanja podataka na bazu, napravljene su forme za profile klijenata, supervizora, profili zaposlenika, te naslovna strana. Naslovna strana je obuhvatala podatke o ETF Railwaysu, mogućnost čitanja knjige žalbe, mogućnost klijenta da ostavi svoj komentar, te kupovinu karte. Ispoštovana je adaptivnost formi. ASP.NET dio aplikacije nije implementiran, ali smo u UWP dijelu implementirali sve forme, koje su trebale biti u ASP.NET dijelu.

ViewModel je predstavljao sponu između Modela i Pogleda. Komande na pogledu su povezivane sa odgovarajućim funkcionalnostima koristeći Binding. Binding smo koristile na način da smo u našem ViewModelu pravile komande pomoću ICommand.

```
30         public ICommand utisci { get; set; }
31         public ICommand regZaposlenika { get; set; }
32         public ICommand kupi{ get; set; }
33         public ICommand gost1 { get; set; }
34         public ICommand pretraziZaposlene { get; set; }

43
44         public ICommand PrijaviSe { get; set; }
45         public ICommand RegistrujSe { get; set; }
```

```

97         private ICommand login1;
98         private ICommand obrisiLiniju;
99         private String selektovanaVoznja;
100        private ICommand kl;
101        private ICommand zz;
102        private ICommand ad;

```

U konstruktor je bilo potrebno dodati za svaki od ovih ICommand-i kreiranje(sve se može vidjeti u kodu):

```

114         utisci = new RelayCommand<object>(dodajKomentar, mozeLiSeDodati);
115         kupi = new RelayCommand<object>(kupiKartu, mozeLiSeKupiti);
116         RegistrujSe = new RelayCommand<object>(registracijaAsync, moguCaRegistracija);

```

Iz xaml-a smo podatke prosljeđivali ViewModelu na sljedeći način: (neki od prosljeđenih podataka, sve se može vidjeti u projektu):

```

64         <TextBox Name="Ime" Text="{Binding Ime, Mode=TwoWay}" Grid.Column="1" HorizontalAlignment="Center" Grid.Row="3" PlaceholderText="Ime" />
65         <TextBox Name="Prezime" Text="{Binding Prezime, Mode=TwoWay}" Grid.Column="1" HorizontalAlignment="Center" Grid.Row="4" PlaceholderText="Prezime" />
66         <TextBox Name="Email" Text="{Binding Email, Mode=TwoWay}" Grid.Column="1" HorizontalAlignment="Center" Grid.Row="5" PlaceholderText="Email" />
67         <TextBox Name="Username" Text="{Binding UsernameRegistracija, Mode=TwoWay}" Grid.Column="1" HorizontalAlignment="Center" Grid.Row="6" PlaceholderText="Username" />
68         <PasswordBox Name="Pas" Password="{Binding PasswordRegistracija, Mode=TwoWay}" Grid.Column="1" HorizontalAlignment="Center" Grid.Row="7" PlaceholderText="Password" />
69         <PasswordBox Name="POtvrda" Password="{Binding Potvrda, Mode=TwoWay}" Grid.Column="1" HorizontalAlignment="Center" Grid.Row="9" PlaceholderText="Potvrda" />
70

```

Komande u xaml- u smo postavljali na sljedeći način:

```

="8" VerticalAlignment="Stretch" Click="Registruj_Click" Command="{Binding RegistrujSe, Mode=TwoWay}"/>
" VerticalAlignment="Stretch" Click="Prijava_Click" Command="{Binding PrijaviSe, Mode=TwoWay}"/>
Grid.Column="4" Grid.Row="9" Click="Button_Click" Command="{Binding gost1, Mode=TwoWay}"/>

```