

Programsko inženjerstvo

Ak. god. 2021./2022.

PassDirect

Dokumentacija, Rev. 1

Grupa: *CozySolutions*

Voditelj: *Tomislav Pavković*

Datum predaje: *19.11.2021.*

Nastavnik: *Daria Primorac*

Sadržaj

1 Dnevnik promjena dokumentacije	2
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	16
3.2 Ostali zahtjevi	22
4 Arhitektura i dizajn sustava	23
4.1 Baza podataka	25
4.1.1 Opis tablica	25
4.1.2 Dijagram baze podataka	28
4.2 Dijagram razreda	29
Popis literature	31
Indeks slika i dijagrama	32
Dodatak: Prikaz aktivnosti grupe	33

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Mirta Vučinić	25.10.2021.
0.2	Funkcionalni zahtjevi, UC.	Ivan Hajpek	28.10.2021.
0.3	UC dijagram.	Martina Galić	28.10.2021.
0.4	DB dijagram i sekvencijski dijagrami.	Martina Galić	03.11.2021.
0.5	DB dijagram.	Ivan Hajpek	03.11.2021.
0.6	Dorađeni UC-ovi i sekvencijski dijagrami.	Ivan Hajpek	04.11.2021.
0.7	Dorađeni UC-ovi i dijagram baze podataka.	Martina Galić	04.11.2021.
0.8	Napisan opis za entitete.	Martina Galić	07.11.2021.
0.9	Napisan kratak opis baze i pojedinih entiteta.	Ivan Hajpek	08.11.2021.
0.10	Opis arhitekture sustava.	Martina Galić	10.11.2021.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.11	Opis dizajna sustava i ostali zahtjevi.	Ivan Hajpek	10.11.2021.
0.12	Dorađeni sekv. dijagrami i pregled dokumentacije.	Ivan Hajpek	15.11.2021.
0.13	Dorađeni sekv. dijagrami i opis projektnog zadatka.	Martina Galić	15.11.2021.
0.14	Izmjena BP i opis Controller Diagrama.	Ivan Hajpek	16.11.2021.
0.15	Class Diagrams.	Martina Galić	16.11.2021.
0.16	Dodatak, literatura, opis sastanaka.	Ivan Hajpek	19.11.2021.
0.17	Tablica aktivnosti i izmjena pojedinih dijagrama.	Martina Galić	19.11.2021.

2. Opis projektnog zadatka

Velik problem sadašnjice, a i prošlosti u području željezničke strukture napokon je dobio rješenje. U prošlosti je zasigurno bilo teže riješiti problem nejednolike raspodjele mase unutar vagona vlaka zbog nedostatka naprednije tehnologije, što je uzrokovalo povećanje istrošenosti kotača, slabljenje strukture osovine te postojanje veće vjerojatnosti iskliznuća vlaka iz tračnica. Mehaničari su stoga periodično morali održavati sustav kotača radi njegove rekonfiguracije i optimizacije što nekada možda i nije bilo nužno, no mehaničari nisu bili sigurni te su svakako napravili potrebno održavanje, što u nekim zemljama moraju i danas, no ne i u Nizozemskoj, jednoj od najnaprednijih država po pitanju željezničkog prometa. Utvrđivanje dijela vagona na kojem je potrebno provesti održavanje je dugotrajan i kompleksan posao. Kako bi smanjili broj nesreća uzrokovanih nastalim problemima zbog nejednolike raspodjele mase te sačuvali infrastrukturu, odnosno produžili joj životni vijek, Nizozemci su osmislili sustav senzora Gotcha.

Gotcha se sastoji od senzora postavljenih duž tračnica kolosijeka na odabranim točkama. Pri prolasku vlaka preko senzora, senzori mjere opterećenje kotača u stvarnom vremenu. Iz vremenskog niza podataka se tada može procijeniti akumulativna šteta na svakom kotaču i odrediti optimalan trenutak održavanja svakog vagona.

Uz to što će senzor zabilježiti podatke koji pomažu pri određivanju optimalnog trenutka održavanja vagona, cilj je i doći do manjeg broja popravaka odnosno održavanja. To se postiže na način da se ravnomjerno rasporedi masa u vagonu, odnosno da putnici sjede na odgovarajućim mjestima. Dakle, postiže se pomoć u prevenciji oštećenja nastalim nejednolikom raspodjelom mase, smanjuje se trošak održavanja i povećava vrijeme dostupnosti vagona. Ideja aplikacije PassDirect je upućivanje putnika u odgovarajuće vagone (i perone na kojima ih mogu čekati) u svrhu optimalne distribucije njihove težine, uporabom podataka sustava senzora Gotcha. Vlakovi nad kojima se provode mjerenja namijenjeni su brzom povezivanju prigradskih naselja te nemaju omogućenu rezervaciju sjedala.

Korisnik:

Kako bi korisnik mogao pristupiti funkcionalnostima web stranice PassDirect, mora pristupiti registraciji, odnosno kreiranju svog korisničkog računa te prijaviti na isti. Prilikom registracije, korisnik unosi svoje podatke, i to: ime, prezime, e-mail adresu i lozinku. Korisnik može imati ili ulogu administratora ili konkretnog korisnika. Administrator može pregledati i ukloniti korisnike te pregledati njihove transakcije, no nije zadužen za mijenjanje voznog reda. Također ima pristup svim provedenim transakcijama, bazi podataka korisnika i redu vožnje. PassDirect upravlja rasporedom putnika dok je za vozni red zadužen drugi ogranak željeznice.

Nakon prijave i registracije:

Nakon uspješne registracije i prijave, korisniku se na pregledniku prikazuju ponuđena stajališta, od kojih on može odabrati jedno. Odabir ga vodi na listu dolazećih vlakova na to stajalište (u stvarnom vremenu). Informacije koje korisnik treba dobiti odabirom stajališta su: identifikacijska oznaka vlaka, krajnje odredište, predviđeni kolosijek i vrijeme dolaska, preporučeni broj perona, preporučeni broj vagona i preporučeni dio vagona (front/end). Ukoliko postoji moguće kašnjenje, crvena boja će sugerirati na njega. Zadnje tri informacije bi trebale biti istaknute i dane samo za one vlakove koji su napustili prethodno stajalište. Do zaključka je li vlak napustio prethodno stajalište dolazimo uporabom senzora koji se nalazi na prošlom stajalištu. Ako je mjerenje detektirano, vlak je napustio prethodno stajalište. Vlak se zadržava na svakom stajalištu 10 minuta. Vrijeme u kojem senzor detektira vlak, koristimo za određivanje kašnjenja vlaka na sljedeće stajalište, točnije određujemo hoće li kasniti i koliko će kasniti. Senzor je vremenski udaljen od stajališta 1 minutu, te strojovođa može dodatno nadoknaditi kašnjenje od 1 minute bez kršenja zakona o sigurnosti željezničkog prometa. Stoga granična vrijednost iznosi 2 minute, odnosno ako senzor očitava podatke težine vagona više od 2 minute nakon što je vlak trebao napustiti stajalište, može se izdati upozorenje o njegovom kašnjenju na sljedeće stajalište. Korisnik u svakom trenutku može klasično pretražiti red vožnje. Klasično pretraživanje uključuje: odabir mjesta polaska, odabir mjesta dolaska i datum. Na zaslonu se potom izlistavaju odgovarajuće linije vlakova zajedno s ostalim informacijama (npr. identifikacijska oznaka vlaka, mjesto i vrijeme polaska, mjesto i vrijeme dolaska, kolosijek dolaska, kašnjenje vlaka i cijena karte).

Kupovina karte:

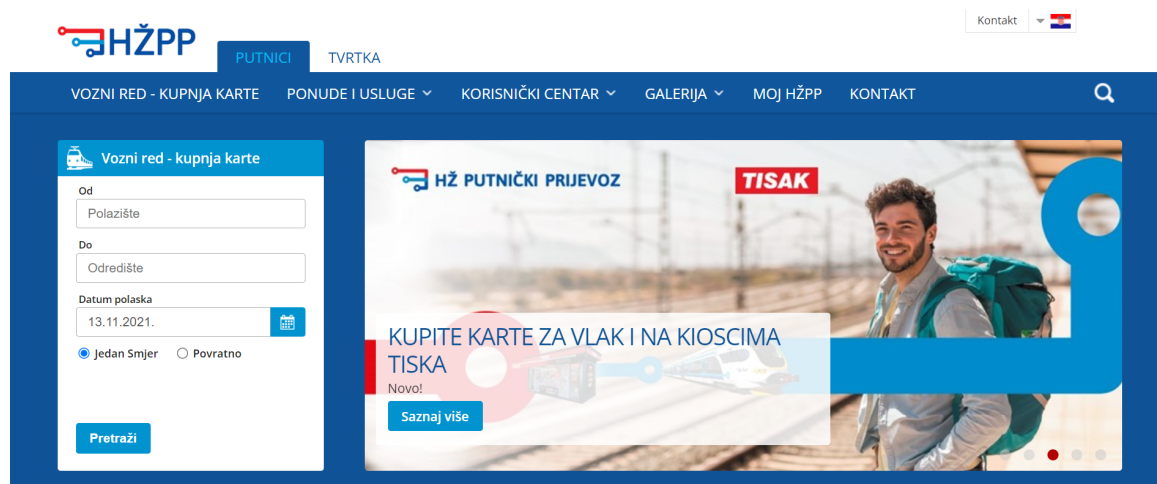
Prilikom prikazivanja linija, korisnik može kupiti kartu za svaku od njih. PassDirect sadrži upitnik koji oponaša plaćanje. Pri prvoj transakciji, korisnik popunjava formular, no pri svakoj narednoj, formular se popunjava sam. Nakon kupovine, korisniku stižu dva e-maila. U jednom e-mailu se nalazi kupljena karta, a u drugom obavijest koja ga upućuje na određeni kolosijek i peron koji je procijenjen kao najpogodniji. Drugi e-mail treba stići tek nakon što vlak napusti prethodno stajalište. Drugi e-mail treba sadržavati: informativni tekst, šifru vlaka, vrijeme njegovog dolaska, kolosijek te uputu za korisnika o prikladnom vagonu, dijelu vagona te broju perona na kojem taj vagon može pričekati. Naravno, cilj aplikacije jest uputiti putnika u praznji dio najmanje okupiranog vagona.

Sustav senzora:

Svako mjerenje senzora daje podatak koji se sastoji od: položaja senzora na pruzi, identifikacijske oznake vlaka, brzine vlaka, vrijeme mjerenja te niza prirodnih brojeva, po dva za svaki vagon. Niz prirodnih brojeva označava višak detektirane mase u kilogramima u svakom vagonu i to jedan podatak za prednji dio vagona, a drugi za stražnji dio. Na primjer: 80, 60, 110, 220, 756 i 923. Prvi vagon (80,60) je uravnotežen, drugi vagon (110, 220) također, dok treći vagon (756, 923) ima najviše putnika no i najveću razliku između prednjeg i stražnjeg dijela. Sva mjerenja na pojedinačnom vlaku obavljaju su istovremeno. Zanimljivo je da razliku između lijeve i desne strane vagona. PassDirect je implementiran kao web aplikacija te podržava rad više korisnika u stvarnom vremenu.

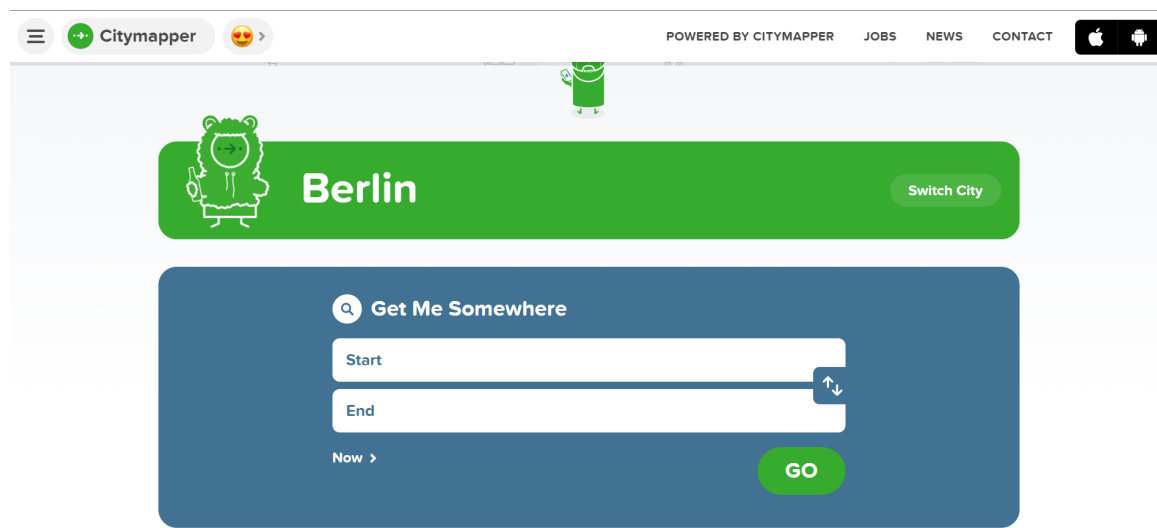
Slična programska rješenja:

Postoje mnoga slična programska rješenja koja omogućavaju korisniku kupovinu karte, odabir početnog i krajnjeg stajališta koji rezultira izlistavanjem svih linija. Jedna od njih je naravno HŽPP-ova stranica. Navedena stranica ima implementirane sve funkcionalnosti kao i PassDirect poput kupovine karte, pregleda svih linija, pregleda svih linija s obzirom na početno i krajnje stajalište, datum, vrijeme i slično, no s obzirom da u Hrvatskoj još nije u primjenu stupio Gotcha sustav senzora, na ovako napredno rješenje kao u Nizozemskoj ćemo morati sačekati.



Slika 2.1: Web stranica HŽPP-a

Također je tu i web aplikacija CityMapper koja sugerira korisniku mjesto sjedenja, no ne s obzirom na Gotcha sustav, nego s obzirom na korisnikove potrebe. Konkretnije, predlaže najbolje mjesto s obzirom na korisnikovo krajnje stajalište ili presjedanje. CityMapper posjeduje sve navedene funkcionalnosti kao i HŽPP web stranica. No, jako je malo web aplikacija koje sugeriraju korisniku koje mjesto bi bilo poželjno koristiti.



Slika 2.2: Web stranica cityMapper

Moguća nadogradnja:

Web aplikaciju PassDirect bi se moglo unaprijediti dodavanjem novih funkcionalnosti poput zaključivanja o potrebi obnove infrastrukture na osnovu mjerenja sen-

zora. Predviđanjem obnove, mogli bismo pravovremeno ukloniti određene vozne redove kako korisnik ne bi dobivao krivu informaciju pri ranoj kupovini karte ili ako dođe do otkazivanja linije, aplikacija bi trebala poslati e-mail korisniku. Uz to bi bila poželjna i funkcionalnost koja bi pri određivanju sjedala za korisnika, uz podatke mjerenja, uzimala u obzir i krajnje stajalište korisnika ili presjedanje i na taj način poticala korisnika još više da koristi upravo to određeno sjedalo, jer u tom slučaju, korist bi imao i korisnik i željeznički prijevoz.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Klijent
2. Administrator
3. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) se registrirati kako bi se mogao prijaviti u aplikaciju
2. Klijent (inicijator) može:
 - (a) se prijaviti u stranicu kako bi mogao koristiti aplikaciju
 - (b) odabrati jedno od ponuđenih stajališta vlaka i dobiti prikaz informacija za svaku ponuđenu vožnju (identifikacijsku oznaku vlaka, krajnje odredište, predviđeni kolosijek i vrijeme dolaska, preporučeni broj perona, preporučeni broj vagona i preporučeni dio vagona)
 - (c) klasično pretražiti red vožnje u bilo kojem trenutku (odabir mjesta polaska, odabir mjesta dolaska i datum)
 - (d) kupiti kartu za svaku prikazanu liniju na aplikaciji
3. Administrator (inicijator) može:
 - (a) pristupati podacima o klijentima te pregledati njihove transakcije
 - (b) pristupati redu vožnje
 - (c) brisati klijente
4. Aplikacija (sudionik) mora:
 - (a) poslati mail s kupljenom kartom klijentu nakon što je klijent plati
 - (b) poslati drugi mail kojim ga upućuje na određeni kolosijek i peron

5. Baza podataka (sudionik):

- (a) pohranjuje podatke o korisnicima i svim njihovim provedenim transakcijama(kupljenim kartama)
- (b) pohranjuje podatke o voznom redu, poput polaznog i krajnjeg stajališta, cijene karte, datuma linije i slično

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvaranje korisničkog računa za prijavu u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** –
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju registracije
 2. Korisnik upisuje potrebne vlastite podatke
 3. Korisnik potvrđuje registraciju
 4. Sustav validira podatke i sprema ih u bazu podataka u slučaju ispravnosti
- **Opis mogućih odstupanja:**
 - 2.a Odabir već postojećeg korisničkog imena, odnosno e-maila, upis korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnog formata e-maila
 1. Sustav obavijesti korisnika o neuspješnom upisu podatka i vraća ga na stranicu za registraciju
 2. Korisnik mijenja pogrešne podatke ili odustaje od registracije

UC2 - Prijava

- **Glavni sudionik:** Klijent
- **Cilj:** Prijava za pristup aplikaciji za kupnju karata
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik unosi korisničko ime i lozinku
 2. Korisnik potvrđuje navedene podatke
 3. Korisnik dobiva pristup aplikaciji
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime ili lozinka
 1. Sustav obavijesti korisnika o neuspješnoj prijavi i vrati ga na stranicu za prijavu

UC3 - Pregled vlakova za stajalište

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati sve dolazeće vlakove na stajalište koje je klijent odabrao
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava
- **Opis osnovnog tijeka:**
 1. Korisnik odabire stajalište koje mu treba
 2. Korisnik potvrđuje odabir
 3. Aplikacija prikazuje popis dolazećih vlakova na odabrano stajalište
- **Opis mogućih odstupanja:**
 - 2.a Korisnik nije odabrao stajalište
 1. Sustav obavijesti korisnika o neuspješnom zahtjevu i zatraži odabir stajališta

UC4 - Pretraživanje voznog reda

- **Glavni sudionik:** Klijent
- **Cilj:** Pretražiti vozni red
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava
- **Opis osnovnog tijeka:**
 1. Korisnik odabire mjesto polaska i dolaska
 2. Korisnik odabire željeni datum
 3. Korisnik potvrđuje odabir
 4. Aplikacija prikazuje vozni red za odabrane podatke
- **Opis mogućih odstupanja:**
 - 3.a Korisnik nije odabrao mjesto polaska, dolaska ili datum
 1. Sustav obavijesti korisnika o neuspješnom zahtjevu i zatraži ponovni odabir i unos

UC5 - Kupnja karte

- **Glavni sudionik:** Klijent
- **Cilj:** Kupiti kartu za vlak
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijava
- **Opis osnovnog tijeka:**

1. Korisnik odabire vlak za koji želi kupiti kartu
 2. Korisnik upisuje potrebne osobne podatke za kupnju(ime, prezime, podaci o kartici itd.)
 3. Korisnik potvrđuje kupnju
- **Opis mogućih odstupanja:**
 - 2.a Korisnik nije dobro upisao neki podatak
 1. Sustav obavijesti korisnika o neuspješnoj kupnji zbog neispravno unesenih podataka
 2. Korisnik ima mogućnost ponovno unijeti podatke ili odustati od kupnje

UC6 - Pregled klijenata

- **Glavni sudionik:** Administrator
- **Cilj:** Pristupiti i pregledati podatke klijenata
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i registriran s dodijeljenim pravom administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabere opciju pregledavanja klijenata
 2. Na aplikaciji se izlistaju podatci svakog korisnika aplikacije

UC7 - Pregled transakcija

- **Glavni sudionik:** Administrator
- **Cilj:** Pristupiti i pregledati podatke o provedenim transakcijama
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i registriran s dodijeljenim pravom administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabere opciju pregledavanja transakcija
 2. Na aplikaciji se izlistaju podatci svake provedene transakcije kroz aplikaciju

UC8 - Pregled reda vožnje

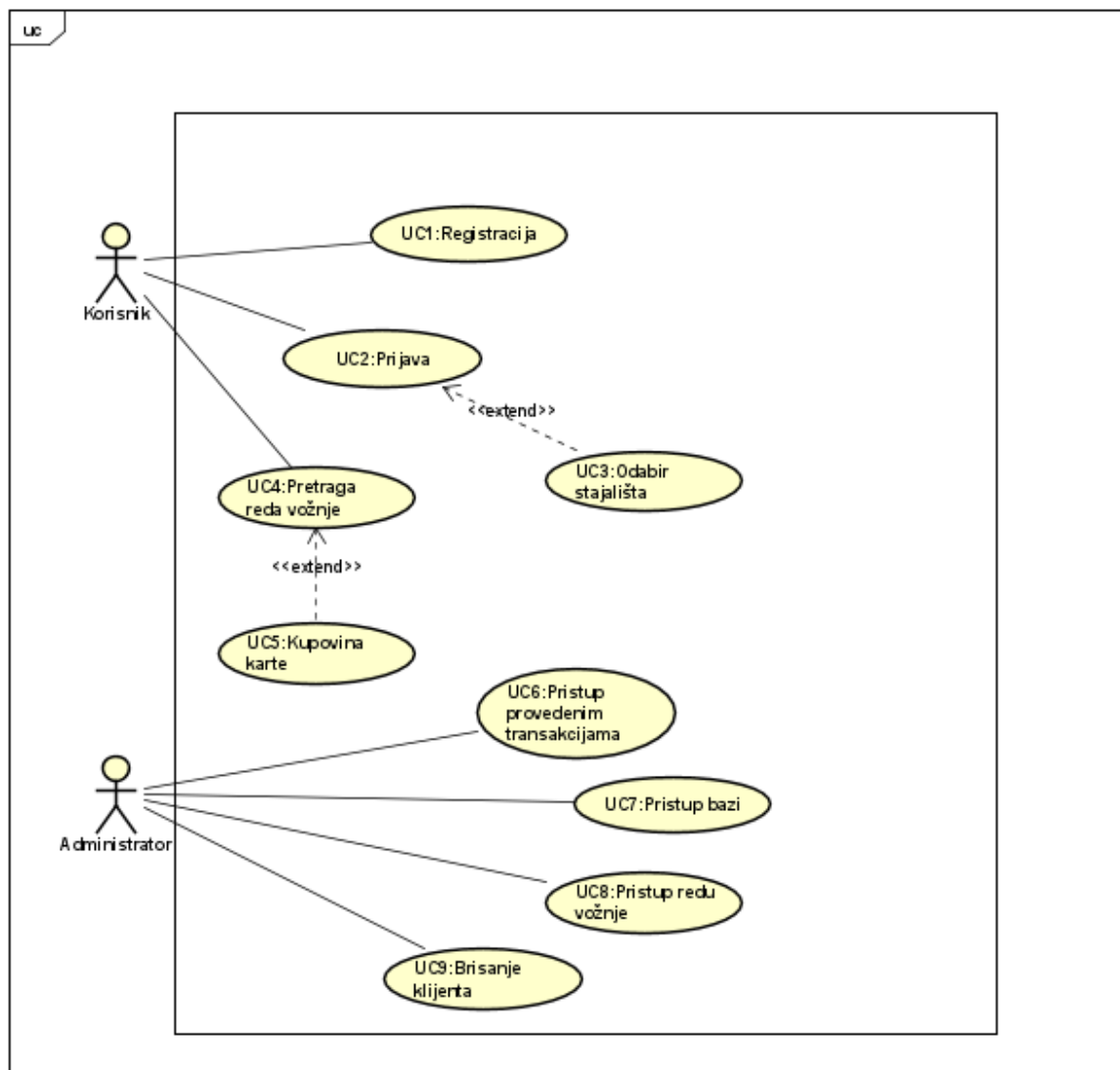
- **Glavni sudionik:** Administrator
- **Cilj:** Pristupiti i pregledati vozni red
- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik mora biti prijavljen i registriran s dodijeljenim pravom administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabere opciju pregledavanja voznog reda
 2. Na aplikaciji se izlista vozni red

UC9 - Brisanje klijenta

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati klijente i obrisati odabrane
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i registriran s dodijeljenim pravom administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabere klijenta na listi klijenata
 2. Administrator obriše odabranog klijenta opcijom brisanja
 3. Aplikacija ga vraća na osvježenu listu klijenata

Dijagrami obrazaca uporabe

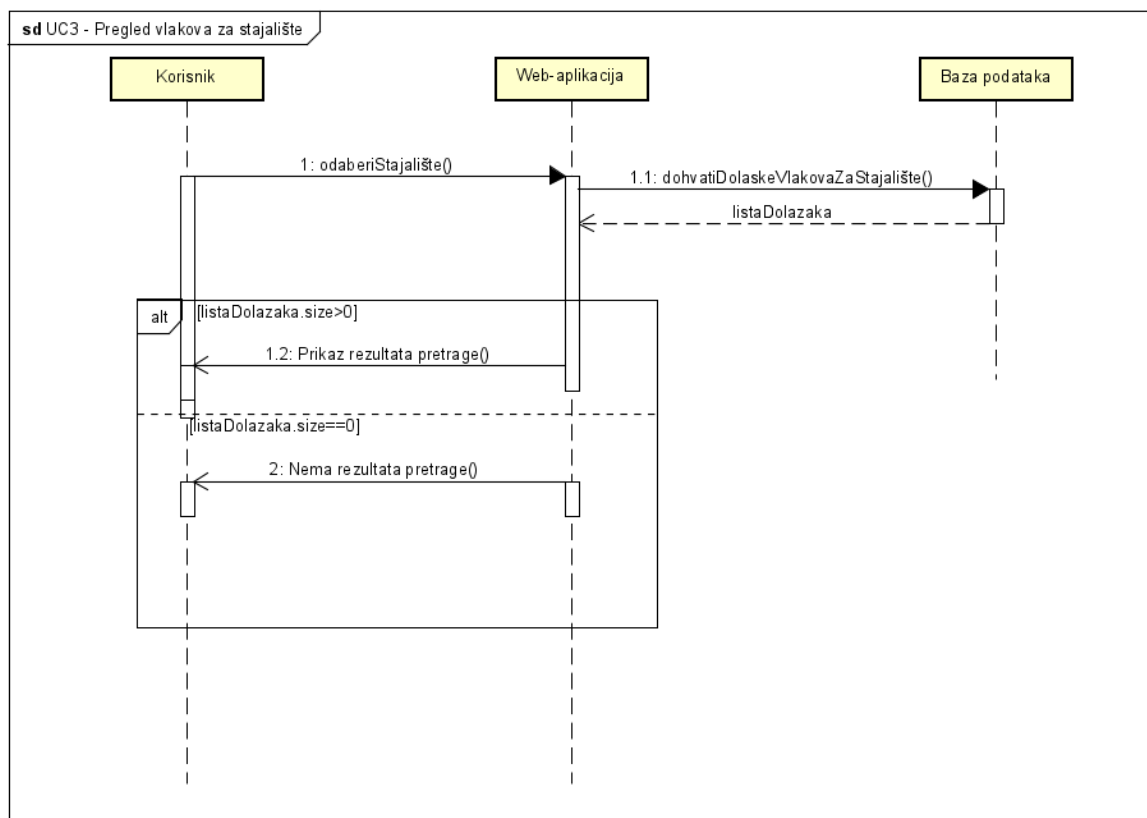


Slika 3.1: UC dijagram

3.1.2 Sekvencijski dijagrami

UC3 - Pregled vlakova za stajalište

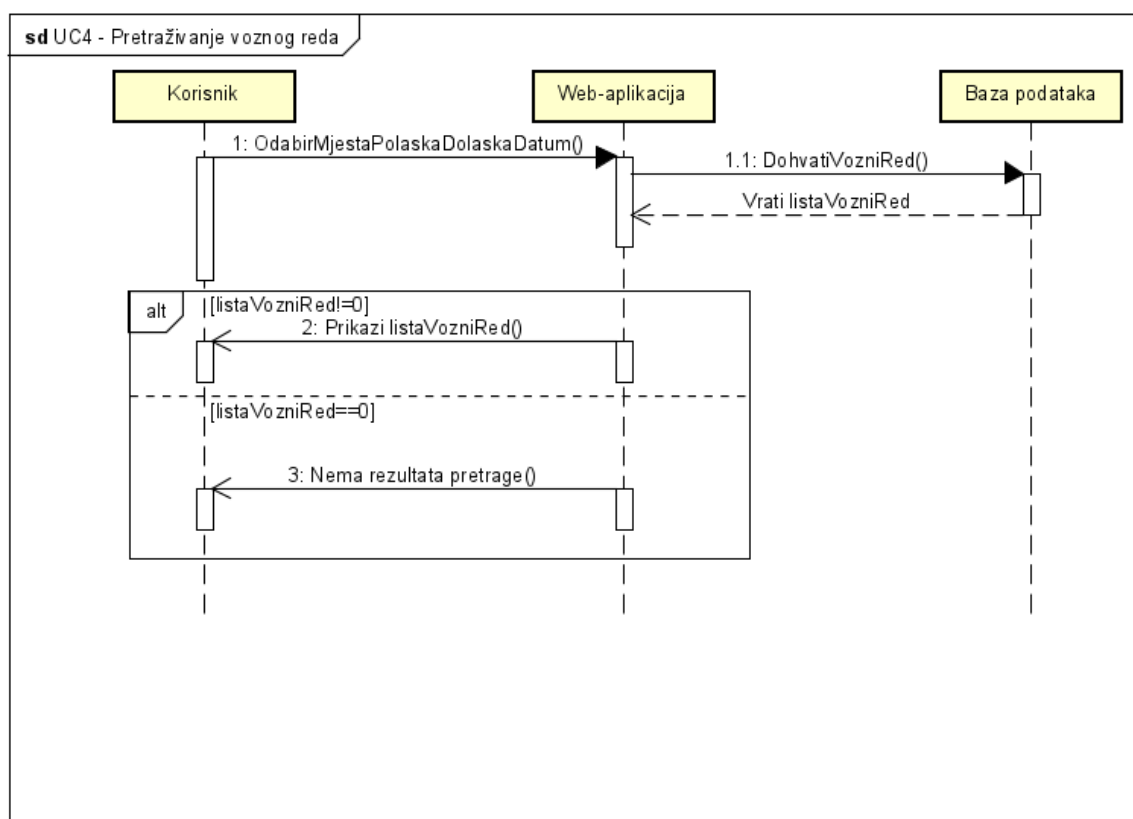
Klijent odabire i potvrđuje stajalište za koje želi dobiti nadolazeće vlakove. Poslužitelj dohvaća i prikazuje popis i informacije o vlakovima koji stižu na odabrano stajalište.



Slika 3.2: UC3 - Pregled vlakova za stajalište

UC4 - Pretraživanje voznog reda

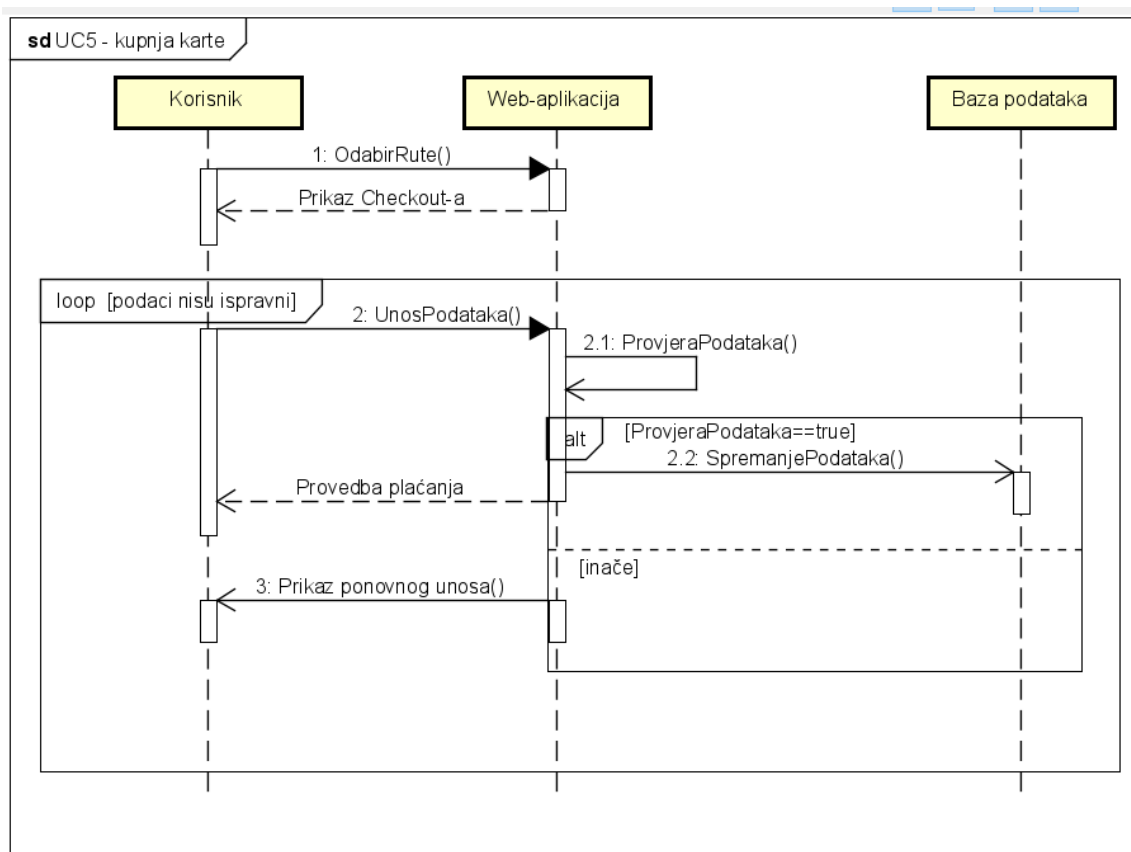
Klijent odabire mjesto polaska i dolaska te datum polaska, a poslužitelj mu vraća listu mogućih linija za unesene podatke.



Slika 3.3: UC4 - Pretraživanje voznog reda

UC5 - Kupnja karte

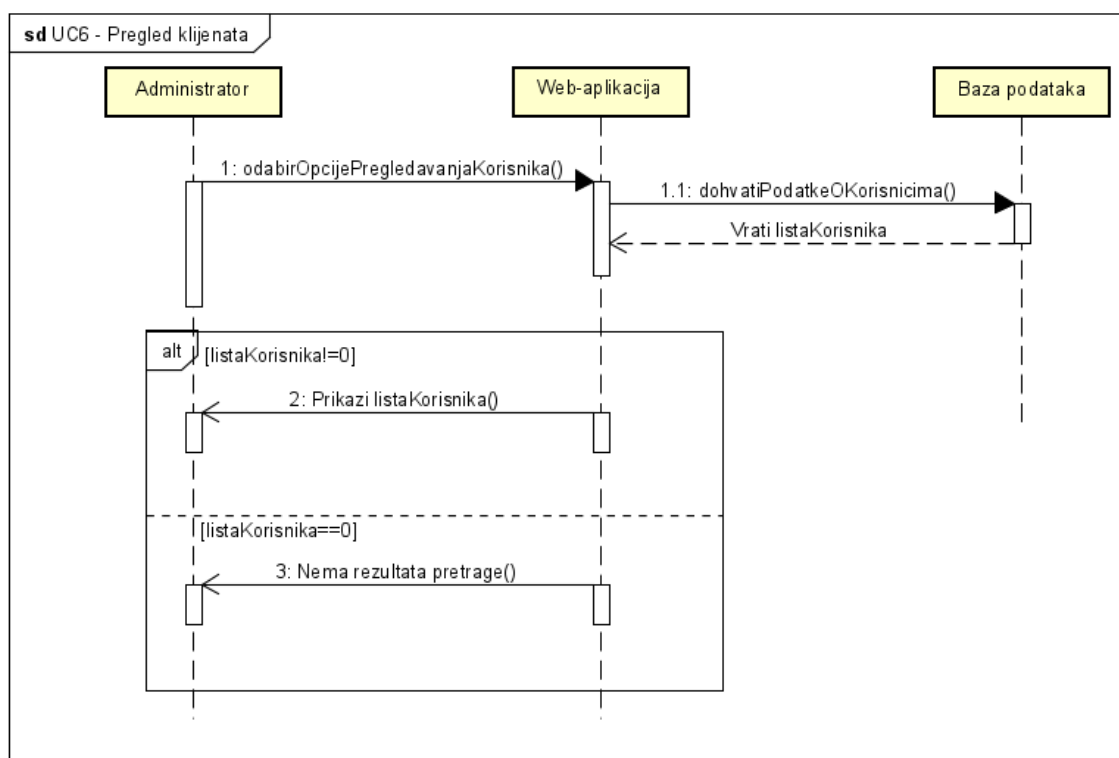
Klijent odabire vlak za koji želi kupiti kartu pa ga poslužitelj odvede na stranicu za checkout. Ovdje unosi sve potrebne podatke za provesti transakciju, nakon čega potvrdi kupnju. Ukoliko su neki podaci netočni ili neispunjeni, poslužitelj daje ponovno mogućnost checkout-a ili odustajanje od kupnje. Kada su podaci točni i ispunjeni, klijentu se javlja uspješna kupnja.



Slika 3.4: UC5 - Kupnja karte

UC6 - Pregled klijenata

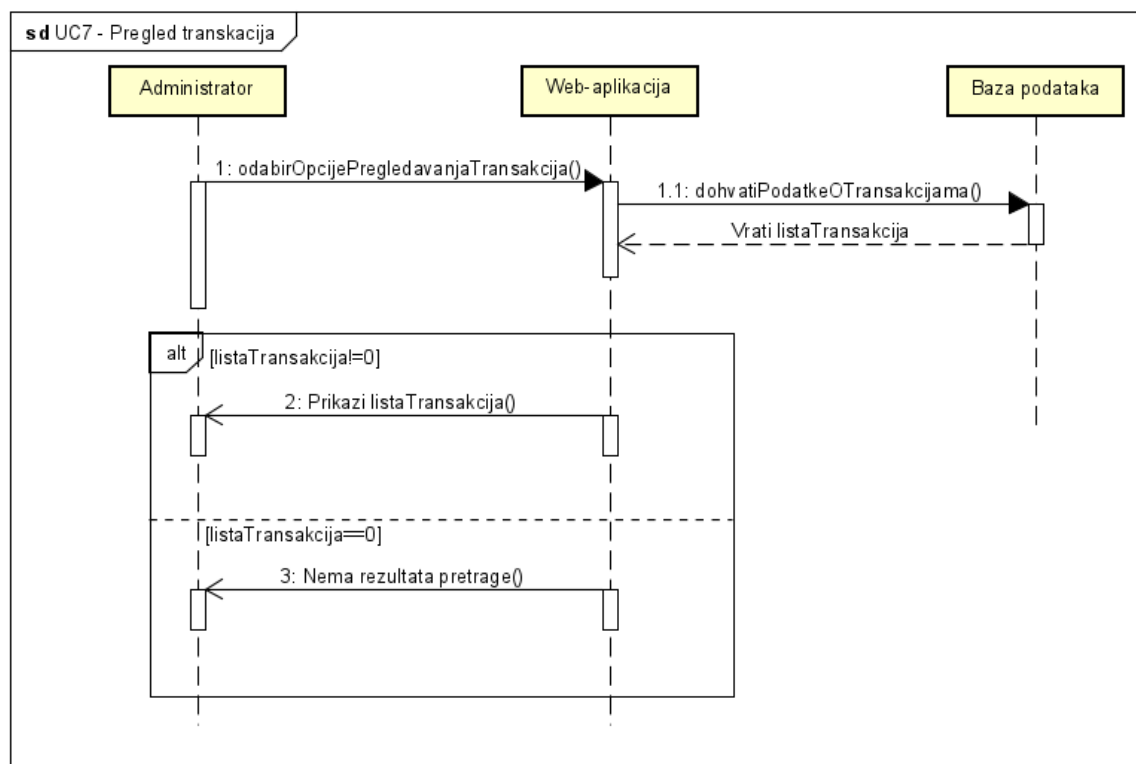
Administrator odabire opciju pregleda podataka o svim klijentima. Poslužitelj dohvaća i prikazuje popis i informacije o klijentima.



Slika 3.5: UC6 - Pregled klijenata

UC7 - Pregled transakcija

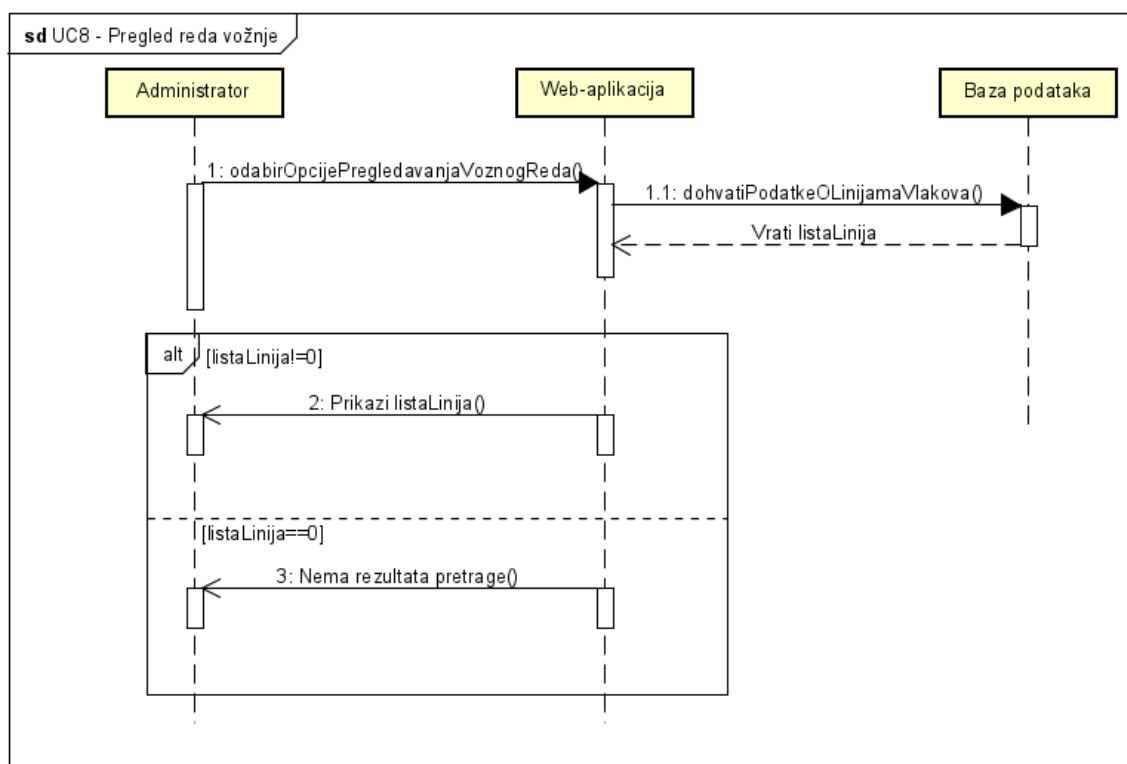
Administrator odabire opciju pregleda podataka o svim provedenim transakcijama. Poslužitelj dohvaća i prikazuje popis i informacije o svim provedenim kupnjama karata.



Slika 3.6: UC7 - Pregled transakcija

UC8 - Pregled reda vožnje

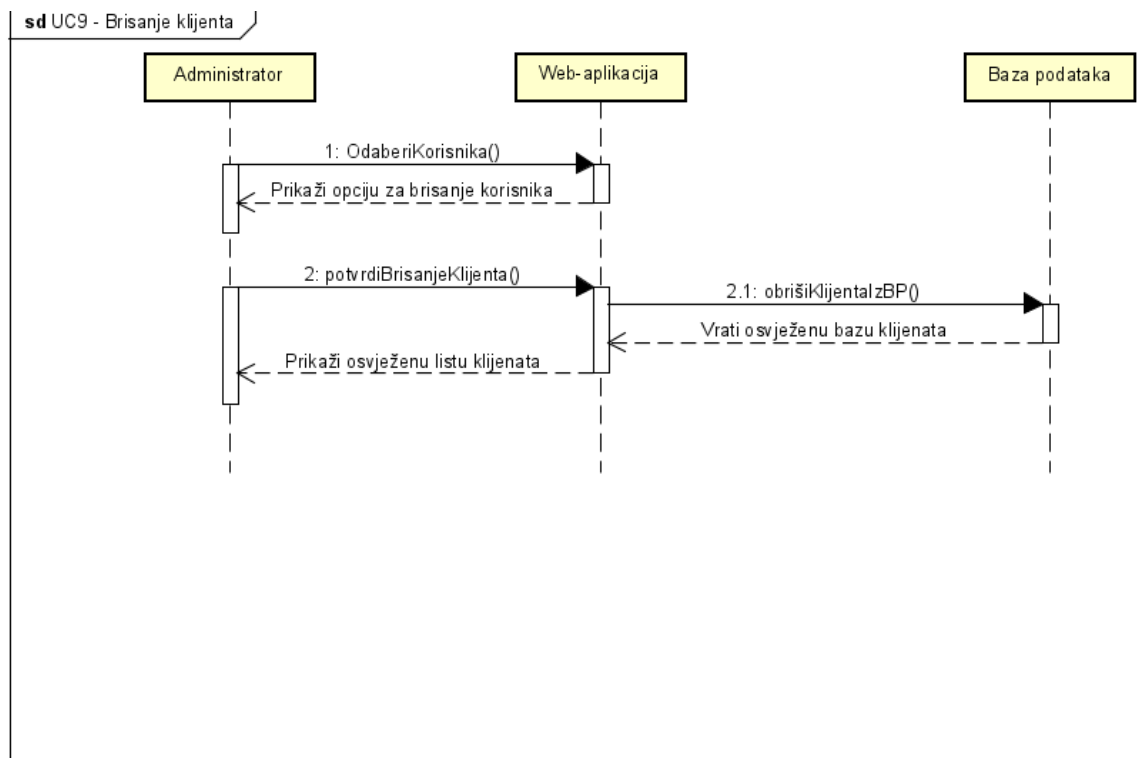
Administrator odabire opciju pregleda podataka o voznom redu, odnosno linijama vlakova. Poslužitelj dohvaća i prikazuje popis i informacije o svim voznim linijama vlakova.



Slika 3.7: UC8 - Pregled voznog reda

UC9 - Brisanje klijenta

Administrator odabire klijenta te mu se izbacuje opcija za brisanje odabranog klijenta. Nakon odabira opcije, poslužitelj briše klijenta i vraća administratora na osvježenu listu klijenata.



Slika 3.8: UC9 - Brisanje klijenta

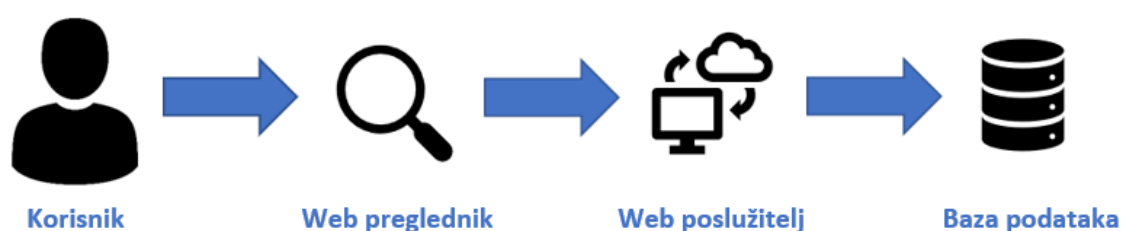
3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u isto vrijeme.
- Sustav mora biti izveden kao web aplikacija s objektno orijentiranim jezikom.
- Komunikacija između korisnika i sustava mora biti brza i jednostavna.
- Aplikacija mora prikazivati aktualne, odnosno ažurirane podatke u svakom trenutku.
- Baza podataka treba biti brza, učinkovita i dobro povezana sa sustavom, otporna na bilo kakve greške korisnika i administratora.
- U slučaju pogrešaka u unosu od strane korisnika, sustav mora izbaciti odgovarajuće upozorenje.
- Dodatna poboljšanja(senzor za pomoć očuvanju vagona) ne smiju ugroziti već postojeće osnovne funkcionalnosti aplikacije.

4. Arhitektura i dizajn sustava

Arhitektura programske potpore predstavlja strukturu sustava ili više njih koji sadrži elemente, njihova obilježja i odnose među njima. Koristimo objektno orijentiranu arhitekturu koja najbolje odgovara razvoju složene web aplikacije namijenjene za više korisnika u stvarnom vremenu. Možemo ju klasificirati na četiri ključna dijela koji osiguravaju izvršavanje naredbi korisnika:

- 1. Web preglednik
- 2. Web poslužitelj
- 3. Web aplikacija
- 4. Baza podataka



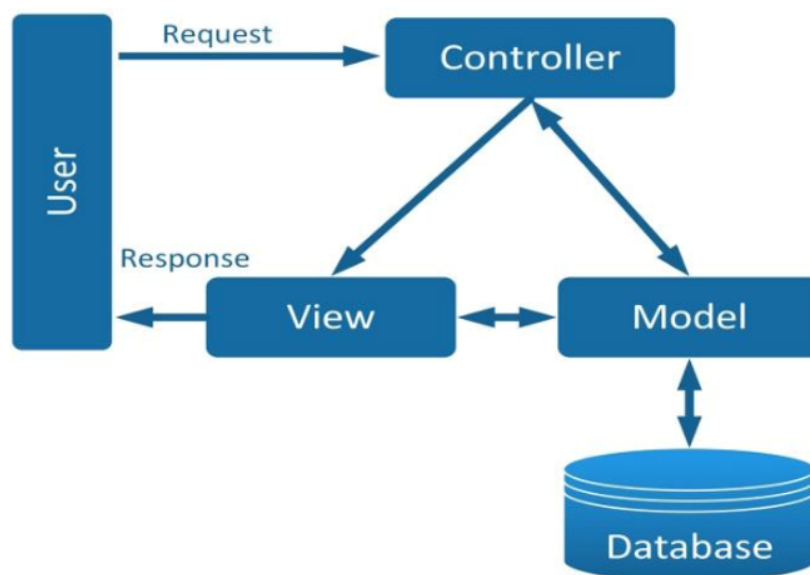
Slika 4.1: Arhitektura sustava

PassDirect je web aplikacija koja svojim korisnicima omogućuje praktičniju, bržu i efikasniju kupovinu karata za vožnju vlakom te pretragu putovanja. Kupovina karte preko ove aplikacije ujedno doprinosi i očuvanju infrastrukture željezničkog prometa. Prilikom pristupa web aplikaciji korisniku na jednostavan način postaje moguće pretražiti putovanja s odabranog polazišnog stajališta i kupiti kartu. Da bi korisnik mogao pristupiti navedenim funkcionalnostima treba pristupiti web pregledniku koji za njega vrši daljnju interakciju s ostalim podsustavima koji su sastavni dijelovi arhitekture sustava. Web preglednik dalje mora komunicirati s web poslužiteljem na kojem je sama implementacija web aplikacije. Preko web poslužitelja se obavljaju svi mogući funkcionalni zahtjevi aplikacije koji su prethodno definirani, a oni su izravno povezani s poslužiteljem baze podataka. U bazi podataka pohranjeni su svi podaci nužni za potpuno ispravan rad aplikacije, ali i

podaci koje ručno unose korisnici poput osobnih podataka. Web poslužitelj šalje objekte na zahtjev klijenata, ti se zahtjevi realiziraju u obliku HTTP poruka. Web poslužitelju se pristupa preko nekog web preglednika.

Za izradu frontend dijela web aplikacije korišten je programski jezik Javascript s razvojnim okvirom React.js. Za izradu backend dijela korišten je Typescript s razvojnim okvirom Express.js. Baza podataka je implementirana pomoću PostgreSQL.

Za realizaciju arhitekture sustava korišten je koncept Model-View-Controller. Model-View-Controller je obrazac koji razdvaja aplikaciju u tri glavne logičke komponente: Model, View i Controller. Svaka od nabrojanih komponenti ima zadatak rukovati s određenim razvojnim aspektima aplikacije. Također, one su nezavisne jedna od druge i kao rezultat toga je jednostavno dodavanje i preoblikovanje svojstava.



Slika 4.2: Model-View-Controller

- **Model**

Poznat je kao najniža razina što znači da je odgovoran za održavanje podataka s kojima korisnik radi. Glavni zadatak je dohvat, manipulacija podacima odnosno suradnja s bazom podataka. Sprema tražene podatke u objekte i na taj način ih šalje bazi podataka. Reagira na zahtjeve Controllera jer on nikada sam ne razgovara s bazom podataka i nakon komunikacije proslijeđuje potrebne podatke Controller-u. Jedna od bitnijih stvari za napomenuti je da Model nikada izravno ne komunicira s View.

- **View**

Služi za prikazivanje podataka tako što generira korisničko sučelje za korisnika. Ti podaci su rezultat rada Model-a, ali se oni ne preuzimaju izravno nego putem Controller-a tako da View surađuje samo s Controller-om.

- **Controller**

Upravlja komponentama Model i View. Ne mora brinuti o rukovanju logikom podataka, već samo govori Model-u što treba učiniti. Nakon primanja podataka od Model-a, on ih obrađuje i prosljeđuje rezultat do View-a.

4.1 Baza podataka

Za potrebe našeg sustava odabrali smo relacijsku bazu podataka koja nam olakšava modeliranje stvarnog svijeta. Baza je definirana skupom tablica, odnosno relacija koje su definirane svojim imenima i skupovima atributa (i ključevima). Zbog pouzdanosti, jednostavnosti i zadovoljavajućih preformansi, odabrali smo rad s PostgreSQL bazom. Baza podataka omogućava nam brz i lak dohvat i spremanje podataka, odnosno obradu i izmjenu podataka korištenih u aplikaciji. Naša baza sastoji se od sljedećih entiteta:

- User
- Ticket
- Journey
- TrainRoute
- Station
- SensorData

4.1.1 Opis tablica

User - Entitet User sadrži sve potrebne informacije o korisniku aplikacije koji može biti kupac ili administrator što određuje atribut UserType. User ima također i attribute FirstName, LastName, Email, Password te CreatedDate što označava datum registracije korisnika. Email je jedinstven atribut, tako da se svaki User razlikuje po svom e-mailu. Ključ entiteta je Email.

Korisnik		
Email	VARCHAR	jedinstveni e-mail korisnika
FirstName	VARCHAR	ime korisnika
LastName	VARCHAR	prezime korisnika
Password	VARCHAR	lozinka registriranog korisnika
UserType	VARCHAR	vrsta korisnika,može biti kupac ili administrator
CreatedDate	DATE	datum registracije korisnika

Ticket - Entitet Ticket je slabi entitet koji ovisi o entitetu User iz razloga što karta ne može biti izdana bez da korisnik zatraži njezino izdavanje. Ključ entiteta Ticket je Id ticket-a. Atribut email je strani ključ koji referencira entitet User te se radi o *Many-to-Many* vezi. Atributi IdJourney i IdTrainRoute rute su strani ključevi koji referenciraju entitet Journey te se radi o *Many-To-One* vezi.

Ticket		
Id	INT	jedinstveni identifikacijski broj karte koja je kupljena
Email	VARCHAR	strani ključ entiteta User
IdJourney	INT	strani ključ entiteta Journey
IdTrainRoute	INT	strani ključ entiteta TrainRoute

Journey - Ovaj entitet sadržava sve važne podatke za jedno putovanje vlakom. Sadrži attribute: IdJourney,DepartureTime i ArrivalTime, Price, DepartureStation, ArrivalStation te IdTrainRoute. Atributi DepartureStation i ArrivalStation su strani ključevi koji referenciraju entitet Station te se radi o *Many-To-One* vezi. Atribut IdTrainRoute je strani ključ koji referencira entitet TrainRoute putem *Many-To-One* veze.

Journey		
IdJourney	INT	jedinstveni identifikacijski broj putovanja
DepartureTime	DATETIME	datum i vrijeme polaska s polazišnog stajališta
ArrivalTime	DATETIME	datum i vrijeme dolaska na odredišno stajalište
Price	FLOAT	cijena karte za putovanje
DepartureStation	VARCHAR	početno stajalište putovanja i strani ključ entiteta Station
ArrivalStation	VARCHAR	završno stajalište putovanja i strani ključ entiteta Station
IdTrainRoute	INT	jedinstveni identifikacijski broj TrainRoute-e

TrainRoute - Entitet TrainRoute ima attribute: Stations(lista koja sadržava sve stanice kojima cjelokupna ruta prolazi pa i početna stanica cijele rute,kao i završna).U ruti Zagreb-Karlovac-Knin-Split, početak cjelokupne rute je Zagreb, a završetak Split. Entitet TrainRoute će imati sve moguće kombinacije podruta za primjer navedene rute. Journeys navedene rute će biti Zagreb-Karlovac, Zagreb-Knin, Zagreb-Split, Karlovac-Knin, Karlovac-Split, Knin-Split. Zbog potrebe zadatka da za svako putovanje imamo početno i završno odredište cjelokupne rute tog putovanja, nastao je ovaj entitet TrainRoute koji je povezan s Journeys *One-To-Many* vezom. Entitet TrainRoute ima također i attribute: IdTrainRoute koji je ujedno i ključ te TrainId.

TrainRoute		
IdTrainRoute	INT	jedinstveni identifikacijski broj cjelokupne rute
Stations	Station[]	lista svih stajališta kojima prolazi vlak u svojoj ruti
TrainId	INT	jedinstveni identifikacijski broj vlaka koji vozi tu rutu

Station - Ovaj entitet sadržava imena svih stajališta. Entitet Stajalište povezan

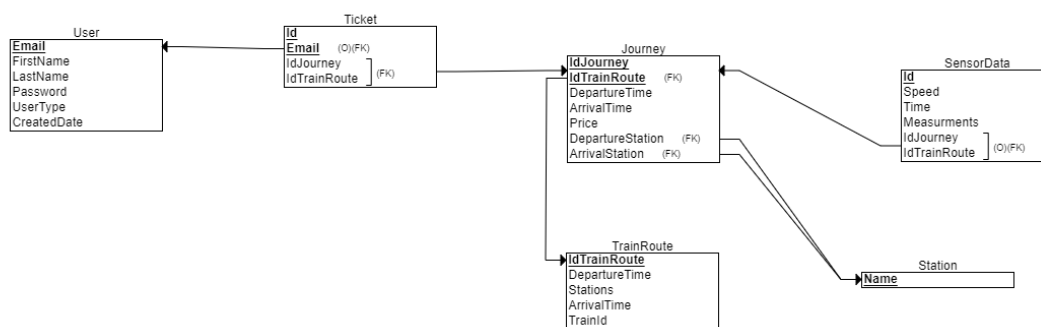
je putem dvije *One-To-Many* veze s entitetom Putovanje, gdje jedna veza označava početno stajalište, a druga završno.

Station		
Name	VARCHAR	ime stajališta

SensorData - Ovaj entitet sadržava sve važne podatke za jedno mjerenje senzora. Sadrži attribute: Id, Time, Speed, Measurments, IdJourney, IdTrainRoute. Entitet SensorData povezan je putem *Many-To-One* veze s entitetom Journey.

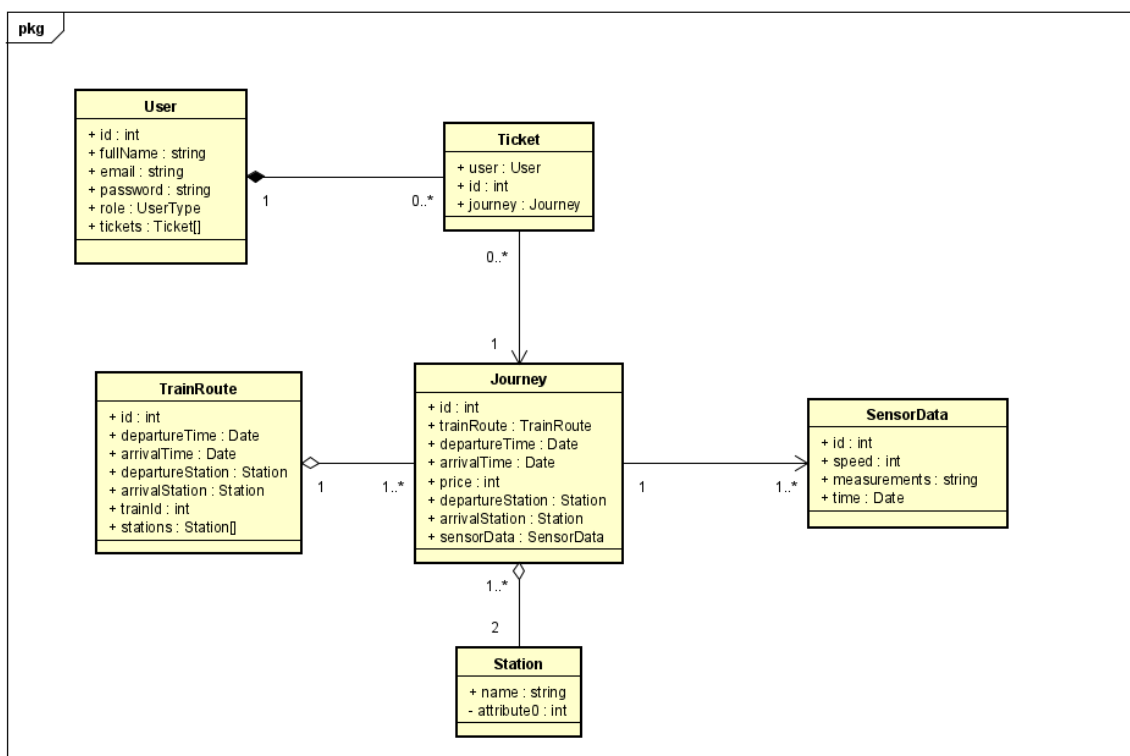
SensorData		
Id	INT	jedinstveni identifikacijski broj mjerenja
Time	DATETIME	datum i vrijeme mjerenja senzora
Measurments	VARCHAR	vrijednost mjerenja senzora
Speed	INT	brzina vlaka u trenutku mjerenja senzora
IdJourney	INT	strani ključ entiteta Journey
IdTrainRoute	INT	strani ključ entiteta TrainRoute

4.1.2 Dijagram baze podataka



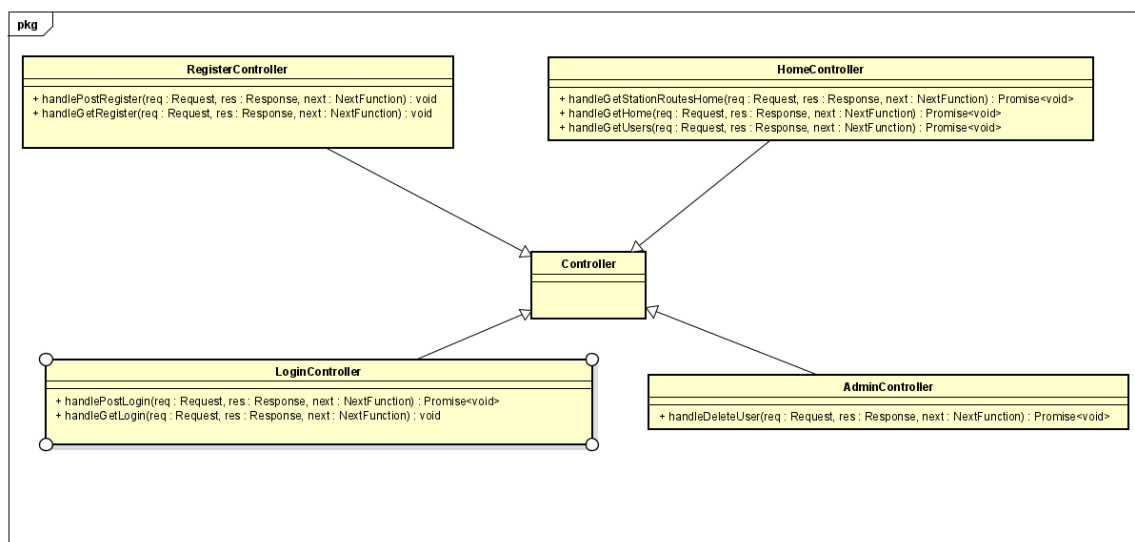
Slika 4.3: Dijagram baze podataka

4.2 Dijagram razreda



Slika 4.4: Dijagram razreda Modela

Model razredi preslikavaju strukturu baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka te vraćaju tražene podatke. Razred *User* predstavlja registriranog i prijavljenog korisnika koji može koristiti njegove osnovne funkcionalnosti. Razred *Ticket* predstavlja jednu kartu koju je kupio *User* za jedno putovanje - *Journey*. Razred *Journey* predstavlja relaciju putovanja za koju korisnik(*User*) kupuje željezničku kartu. Razred *TrainRoute* predstavlja kompletnu rutu na kojoj putuje vlak na kojem je korisnik kupio kartu za određeno putovanje. Razred *Station* predstavlja listu svih stajališta kojima vozi naš prijevoznik, a razred *SensorData* predstavlja mjerenja sustava senzora Gotcha za svako putovanje.



Slika 4.5: Dijagram razreda Controllera

Dijagram se sastoji od nekoliko nadglednika: HomeController, LoginController, RegisterController, AdminController te svi naslijeđuju apstraktni razred Controller. HomeController se koristi prilikom ulaska korisnika na Home ekran i prikaz voznog reda na stranici. RegisterController se koristi prilikom registracije korisnika na stranicu, a LoginController prilikom prijave korisnika/admina u stranicu. AdminController se koristi prilikom funkcionalnosti koje su omogućene za Admina(pregleda, brisanje korisnika).

Popis literature

Kontinuirano osvježavanje

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. The Unified Modeling Language, <https://www.uml-diagrams.org/>
3. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Web stranica HŽPP-a	7
2.2	Web stranica cityMapper	7
3.1	UC dijagram	15
3.2	UC3 - Pregled vlakova za stajalište	16
3.3	UC4 - Pretraživanje voznog reda	17
3.4	UC5 - Kupnja karte	18
3.5	UC6 - Pregled klijenata	19
3.6	UC7 - Pregled transakcija	20
3.7	UC8 - Pregled voznog reda	21
3.8	UC9 - Brisanje klijenta	22
4.1	Arhitektura sustava	23
4.2	Model-View-Controller	24
4.3	Dijagram baze podataka	28
4.4	Dijagram razreda Modela	29
4.5	Dijagram razreda Controllera	30

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

1. sastanak

- Datum: 07.10.2021.
- Prisustvovali: T. Pavković, M. Vučinić, A. Vinković, K. Pavlović, F. Cindrić, M. Galić, I. Hajpek
- Teme sastanka:
 - Razgovor o poznavanju tehnologija
 - Dogovor o tutorialima i upute za okviran način rad

2. sastanak

- Datum: 16.10.2021.
- Prisustvovali: T. Pavković, M. Vučinić, A. Vinković, K. Pavlović, F. Cindrić, M. Galić, I. Hajpek
- Teme sastanka:
 - Komentiranje dobivene teme i razjašnjavanje nejasnoća
 - Upoznavanje s GitLab-om i LaTeX-om

3. sastanak

- Datum: 23.10.2021.
- Prisustvovali: T. Pavković, M. Vučinić, A. Vinković, K. Pavlović, F. Cindrić, M. Galić, I. Hajpek
- Teme sastanka:
 - Dogovorene razvojne okoline i tehnologije
 - Rasprava o izradi generičkih sposobnosti
 - Dogovor za početak razvijanje aplikacija
 - Dogovoren početak rada na predlošku dokumentacije i izgledu stranice

4. sastanak

- Datum: 30.10.2021.
- Prisustvovali: T. Pavković, M. Vučinić, A. Vinković, K. Pavlović, F. Cindrić, M. Galić, I. Hajpek
- Teme sastanka:
 - Pregled prethodno zadanih zadataka
 - Određen kostur aplikacije
 - Završene početne verzije UC-ova i dijagrama
 - Početak dizajna baze podataka

5. sastanak

- Datum: 06.11.2021.
- Prisustvovali: T. Pavković, M. Vučinić, A. Vinković, K. Pavlović, F. Cindrić, M. Galić, I. Hajpek
- Teme sastanka:
 - Doradivanje dokumentacije
 - Modeliranje entiteta i dizajn kontrolera
 - Doradivanje dosadašnjeg front-end-a

6. sastanak

- Datum: 14.11.2021.
- Prisustvovali: T. Pavković, M. Vučinić, A. Vinković, K. Pavlović, F. Cindrić, M. Galić, I. Hajpek
- Teme sastanka:
 - Provjere na front-endu
 - Pravljenje arhitekture i dizajna sustava u dokumentaciji
 - Daljnji rad na back-endu, kontrolerima i modelima

7. sastanak

- Datum: 16.11.2021.
- Prisustvovali: T. Pavković, M. Vučinić, A. Vinković, K. Pavlović, F. Cindrić, M. Galić, I. Hajpek
- Teme sastanka:
 - Završne promjene na dokumentaciji i bazi podataka
 - Povezivanje back-enda i front-enda
 - Dizanje baze na Heroku i deployanje aplikacije

Tablica aktivnosti

Kontinuirano osvježavanje

	Tomislav Pavković	Ante Vinković	Martina Galić	Filip Cindrić	Krešimir Pavlović	Mirta Vučinić	Ivan Hajpek
Upravljanje projektom	15	5					
Opis projektnog zadatka			10			5	
Funkcionalni zahtjevi							8
Opis pojedinih obrazaca							10
Dijagram obrazaca			5				
Sekvencijski dijagrami			13				8
Opis ostalih zahtjeva							7
Arhitektura i dizajn sustava	1	5	10				8
Baza podataka		3	20				15
Dijagram razreda			7				4
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja		5					2

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Tomislav Pavković	Ante Vinković	Martina Galić	Filip Cindrić	Krešimir Pavlović	Mirta Vučinić	Ivan Hajpek
Zaključak i budući rad							
Popis literature							1
<i>Izrada početne stranice</i>	6				20	15	
<i>Izrada baze podataka</i>	10	2					
<i>Spajanje s bazom podataka</i>	7	8				30	
<i>Back end</i>	30	5					
<i>Login</i>	10				10	15	
<i>Registracija</i>	7					15	
<i>Admin home</i>	9			20		4	
<i>Pregled vlakova</i>							
<i>Kupnja karata</i>				30			
<i>Deploy</i>	7	10	10				10
<i>Izrada dizajna</i>		5		5	10		
<i>Izrada testnih podataka</i>		2		2			