Tripko web aplikacija

Arhitekturni projekat

Verzija1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 19.04.2023. | 1.0 | Inicijalna verzija | Miljana, Ksenija, Dimitrije, Stefan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 8

6.2 Kratak opis slučajeva korišćenja 8

6.2.1 Registrovanje 8

6.2.2 Prijavljivanje 8

6.2.3 Kreiranje rute 8

6.2.4 Oglašavanje rute 8

6.2.5 Odabir rute i priključivanje turi 8

6.2.6 Organizacija budžeta 8

6.2.7 Sortiranje 8

6.2.8 Sortiranje po datumu 8

6.2.9 Sortiranje po prosečnoj oceni 8

6.2.10 Sortiranje po vodiču 9

6.2.11 Pretraga ruta 9

6.2.12 Pretraga ruta po lokaciji 9

6.2.13 Pretraga ruta po kategoriji znamenitosti 9

6.2.14 Povratne informacije 9

6.2.15 Pregled i prikazodređene rute 9

6.2.16 Ažuriranje podataka o korisniku 9

6.2.17 Ažuriranje podataka o znamenitostima 10

6.2.18 Zahtev za promociju u power-vodiča 10

6.2.19 Pregled osnovnih činjenica o znamenitostima 10

6.2.20 Brisanje postojećeg korisničkog naloga 10

6.2.21 Kontrola korisničkih naloga 10

6.2.22 Odobravanje power-vodiča 10

7. Pogled na logičku arhitekturu sistema 10

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10

7.1.1 Korisnički interfejs 10

7.1.2 Aplikaciona logika 10

7.1.3 Pristup podacima 11

7.1.4 HTML 11

7.1.5 React 11

7.1.6 .NET 11

7.1.7 MySQL 11

8. Pogled na procese 11

8.1 Procesi 11

9. Pogled na raspoređivanje sistema 12

9.1 Klijent 12

9.2 Web server 12

9.3 DBMS server 12

10. Pogled na implementaciju sistema 12

10.1 Model domena 12

10.2 Šema baze podataka 13

10.3 Komponente sistema 14

10.3.1 Komponente korisničkog interfejsa 14

10.3.2 Komponente aplikacione logike 15

10.3.3 Komponente za pristup podacima 16

11. Performanse 17

12. Kvalitet 17

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Tripko web aplikacije.

# Opseg dokumenta

Dokument se odnosi na Tripko web aplikaciju koja će biti razvijena od strane quadSquad tima. Namena Web aplikacije Tripko je planiranje i organizacija putovanja, kao i mogućnost pronalaska i oglašavanja turističkog vodiča.

# Reference

Spisak korišćene literature:

1. PeNcIL – Predlog projekta, SWE-PeNcIL-01, V1.0, 2007, SWETeam.
2. Tripko – Predlog projekta, quadSquad
3. PeNcIL – Plan realizacije projekta, V1.0, 2007, SWETeam.
4. Tripko – Plan realizacije projekta, quadSquad
5. PeNcIL – Vizija sistema, V1.0, 2007, SWETeam
6. Tripko – Vizija sistema, quadSquad.
7. PeNcIL – Arhitektura projekta, SWE-PeNcIL-05, v1.0, 2007, SWETeam.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. Tripko web aplikacija će biti implementirana kao Web aplikacija zasnovana na JavaScript jeziku i MySQL bazi podataka [4].
2. Klijentski deo Tripko web aplikacije će biti optimizovan za sledeće Web čitače: Microsoft Edge, Opera 8.0 i noviji, Google Chrome, kao i Firefox (Mozilla).[4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Rad aplikacije treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za trgovinu, turizam i telekomunikacije.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

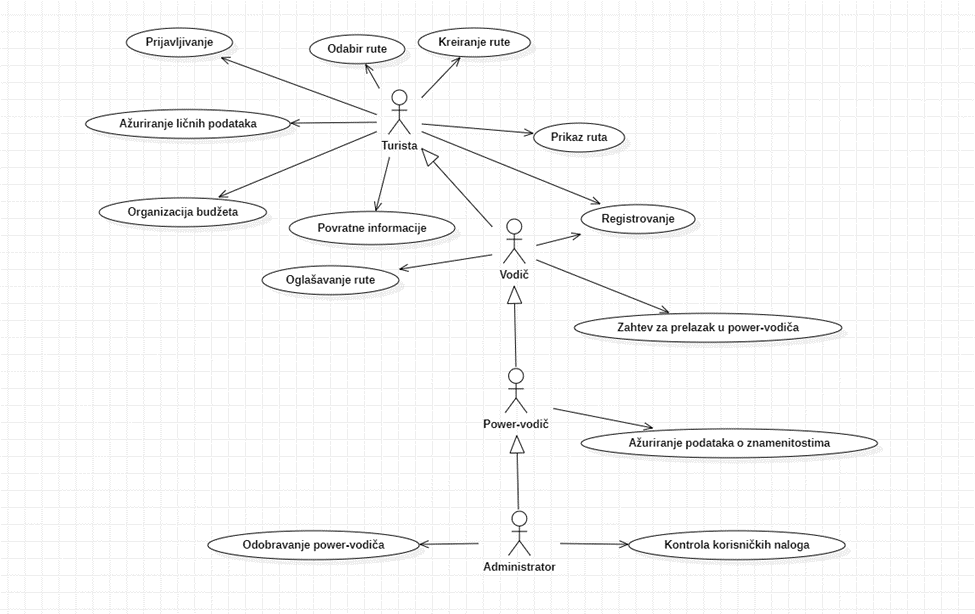
Slučajevi korišćenja Tripko web aplikacije su:

* Registrovanje
* Prijavljivanje
* *Pretraga ruta*
  + Prikaz osnovnih informacija o ruti
  + *Pretraga ruta po lokaciji*
    - Pretraga ruta na odredjenoj lokaciji
    - Pretraga ruta na osnovu trenutne lokacije
  + *Pretraga ruta po kategoriji znamenitosti*
* Ažuriranje podataka o korisniku
* *Ažuriranje podataka o znamenitostima*
* Kontrola korisničkih naloga
  + Kreiranje novog korisnika
  + Brisanje postojećeg korisnika
* Kreiranje rute
* Oglašavanje rute
* Odabir rute i priključivanje turi
* Organizacija budžeta
* Sortiranje
  + Sortiranje po datumu
  + Sortiranje po prosečnoj oceni
  + Sortiranje po vodiču
* Povratne informacije
* Pregled i prikaz određene rute
* Zahtev za promociju u power-vodiča
* Odobravanje zahteva za promociju u power-vodiča
* Brisanje sopstvenog naloga
* Pregled osnovnih činjenica o znamenitostima

Ove slučajevi korišćenja mogu da iniciraju turista, vodič, power vodič ili administrator.

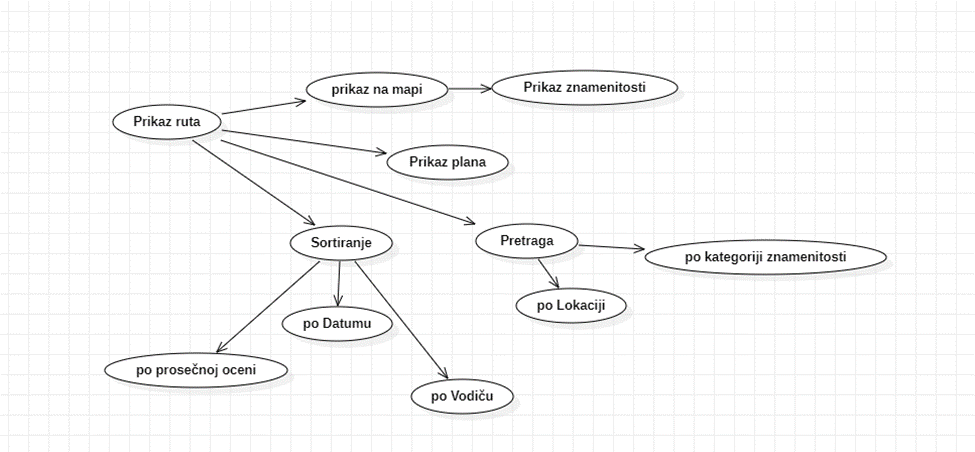
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Tripko web aplikacije prikazan je na sledećoj slici:

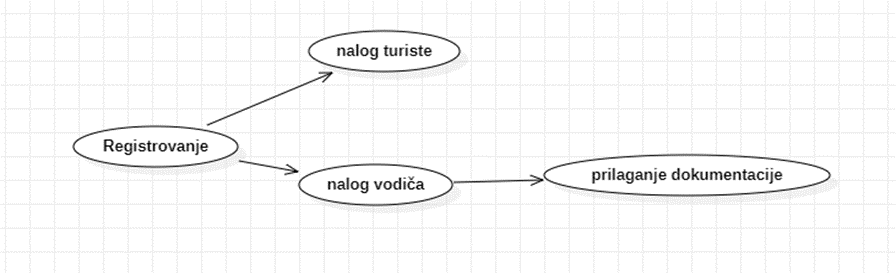


Slučajevi korišćenja *prikaza ruta* obuhvataju složenije radnje koje se mogu razložiti dalje na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *prikaz ruta* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *registrovanja* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Registrovanje

**Kratak opis:**

Kreiranje novog korisničkog naloga.

**Akteri:**

Turista, Vodič.

### Prijavljivanje

**Kratak opis:**

Prijavljivanje korisnika na web aplikaciju u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

**Akteri:**

Turista, vodič, power-vodič, Administrator.

### Kreiranje rute

**Kratak opis:**

Kreiranje rute sa znamenitostima koje korisnik želi da obiđe.

**Akteri:**

Turista, Vodič, Power-vodič.

### Oglašavanje rute

**Kratak opis:**

Oglašavanje rute da bi bila javna, što otvara mogućnost za unajmljivanje vodiča od strane turista.

**Akteri:**

Vodič, Power-vodič.

### Odabir rute i priključivanje turi

**Kratak opis:**

Turista bira jednu od ruta koje su objavili vodiči i priključuje turi.

**Akteri:**

Turista.

### Organizacija budžeta

**Kratak opis:**

Prilikom kreiranja sopstvene rute, aplikacija računa cenu na osnovu dodatih znamenitosti.

**Akteri:**

Turista, Vodič, Power-vodič.

### Sortiranje

**Kratak opis:**

Sortiranje prikazanih ruta po određenom kriterijumu.

**Akteri:**

Turista, Vodič, Power-vodič.

### Sortiranje po datumu

**Kratak opis:**

Sortira se lista ruta po datumu i vremenu polaska.

**Akteri:**

Turista, Vodič, Power-vodič

### Sortiranje po prosečnoj oceni

**Kratak opis:**

Sortira se lista ruta po prosečnoj oceni rute.

**Akteri:**

Turista, Vodič, Power-vodič

### Sortiranje po vodiču

**Kratak opis:**

Sortira se lista ruta po prosečnoj oceni vodiča.

**Akteri:**

Turista, Vodič, Power-vodič

### Pretraga ruta

**Kratak opis:**

Korisnik ima mogucnost pretrage rute na osnovu lokacije na kojoj se ruta nalazi i na osnovu kategorija znamenitosti koje se obilaze u ruti.

**Akteri:**

Turista

### Pretraga ruta po lokaciji

**Kratak opis:**

Korisnik pretražuje rute po lokaciji.

**Akteri:**

Turista.

### Pretraga ruta po kategoriji znamenitosti

**Kratak opis:**

Korisnik pretrazuje rute koje sadrze odredjene kategorije znamenitosti.

**Akteri:**

Turista.

### Povratne informacije

**Kratak opis:**

Ostavljanje recenzije o ruti i vodiču, nakon završene rute.

**Akteri:**

Turista.

### Pregled i prikaz određene rute

**Kratak opis:**

Prikaz stranice sa informacijama o određenoj selektovanoj ruti.

**Akteri:**

Turista, Vodič, Power-vodič, Administrator.

### Ažuriranje podataka o korisniku

**Kratak opis:**

Ažuriranje podataka o sebi od strane prijavljenog korisnika aplikacije.

**Akteri:**

Turista, Vodič, Power-vodič, Administrator.

### Ažuriranje podataka o znamenitostima

**Kratak opis:**

Izmena podataka o znamenitosti od strane Power-vodiča, kako bi bili aktuelni i tačni.

**Akteri:**

Power-vodič.

### Zahtev za promociju u power vodiča

**Kratak opis:**

Promena uloge korisnika iz Vodiča u Power-vodiča.

**Akteri:**

Vodič, Power-vodič.

### Pregled osnovnih činjenica o znamenitostima

**Kratak opis:**

Prikazuju se osnovne informacije o izabranoj znamenitosti.

**Akteri:**

Turista, Vodič, Power-vodič, Administrator.

### Brisanje postojećeg korisničkog naloga

**Kratak opis:**

Brisanje korisničkog naloga i podataka za postojećeg korisnika web aplikacije.

**Akteri:**

Turista, Vodič, Power-vodič, Administrator.

### Kontrola korisničkih naloga

**Kratak opis:**

Kontrola i nadgledanje korisničkih naloga, i moguće preduzimanje neophodnih akcija.

**Akteri:**

Administrator

### Odobravanje power vodiča

**Kratak opis:**

Administrator odobrava promociju vodiča u power-vodiča

**Akteri:**

Administrator.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

Logički pogled na Tripko web aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

*Korisnički interfejs* sadrži Web stranice, JS skripte i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

*Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži klase napisane u JavaScript-u zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

*Pristup podacima* sadrži C# klase (modeli) koje predstavljaju perzistencioni sloj to jest interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs web apliakcije. U njemu su sadržane sve HTML stranice, multimedijalni sadržaji i JS skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i React.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Tripko web aplikacije. Sadrži JS skripte koje realizuju funkcionalnost karakterističnu za domen primene web aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i React i .NET paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži C# klase zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MySQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa .NET i MySQL baza podataka.

### HTML

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### React

tehnologija

Tehnologija React obezbeđuje mehanizam za pisanje i izvršavanje JS skripti na strani klijenta. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs i pozivaju odgovarajuće funkcije koje pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### .NET

Tehnologija

Tehnologija .NET služi za pisanje kontrolera koji sadrže funkcije za pristup bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Tripko web aplikacije.

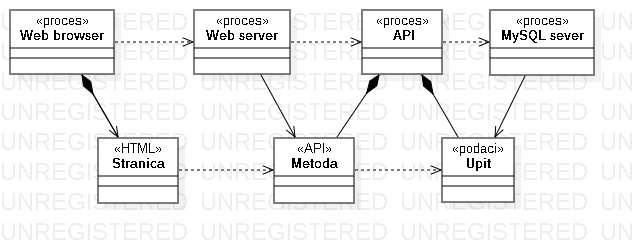
# Pogled na procese

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Tripko web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju PeNcIL portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na PHP-u i MySQL bazi podataka.



### Web browser

Web browser je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu HTML stranicu.

Web browser zavisi od Web servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web browsera. Ukoliko je zahtevana stranica JS skripta, Web server inicira izvršenje .NET API procesa koji obrađuje odgovarajuci API poziv i generiše sadržaj koji se vraća browseru.

### .NET API

.NET API proces obavlja posao obrade zadatog JS skripta i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web browseru. Za izvršenje API-a ovaj proces može da zahteva usluge MySQL servera-a. Komunikacija između .NET API procesa i MySQL servera se obavlja preko prosleđivanja upita i vraćanja rezultat.

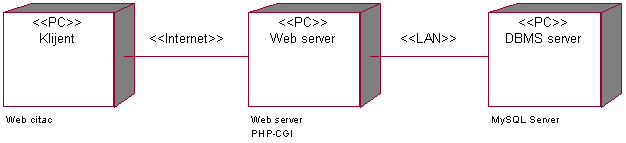
### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja Tripko web aplikacije.



## Klijent

Pristup Tripko web aplikacije se obavlja preko klijentskih računara na kojima je otvoren Web browser. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi .NET API-a koji vrše obradu zadatih JS skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server

DBMS server je računar na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

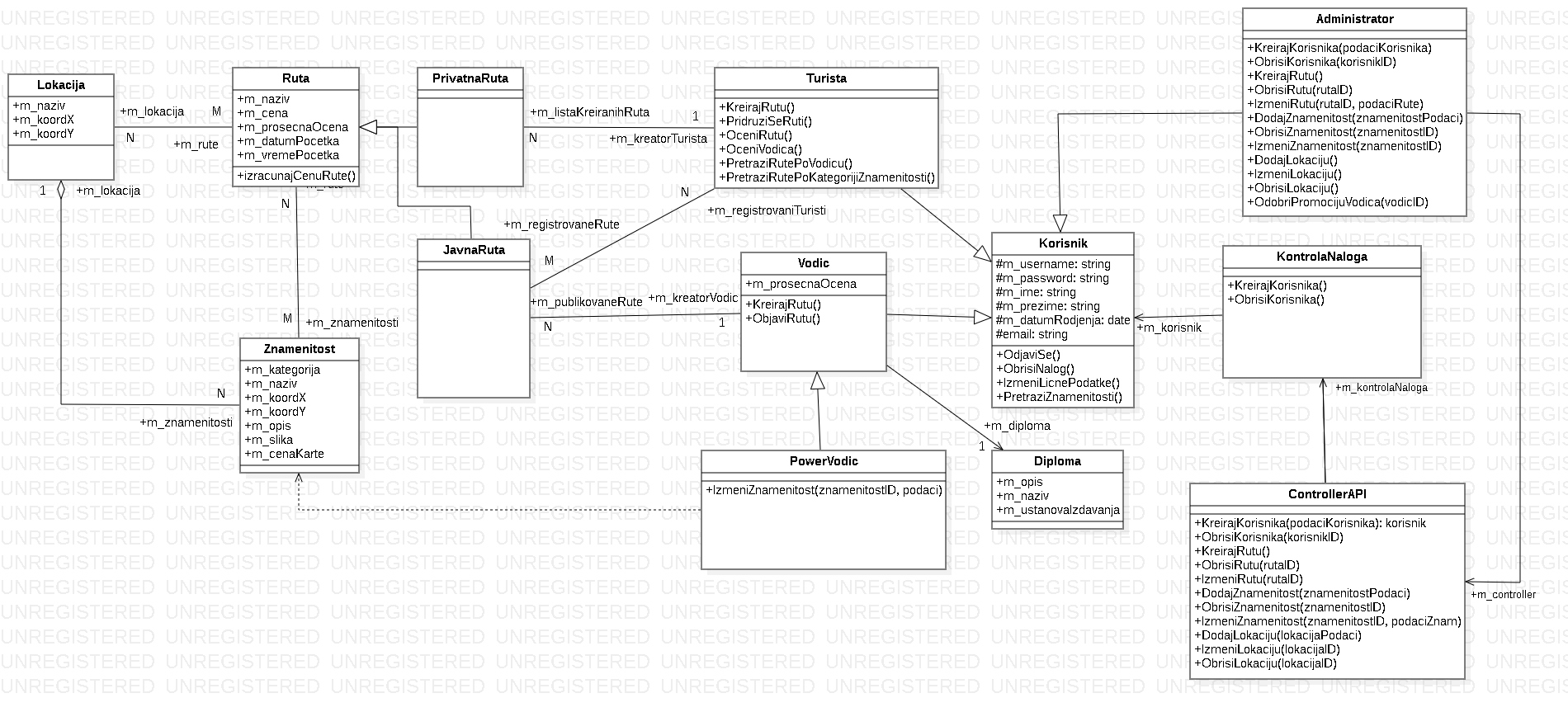
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju Tripko web aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

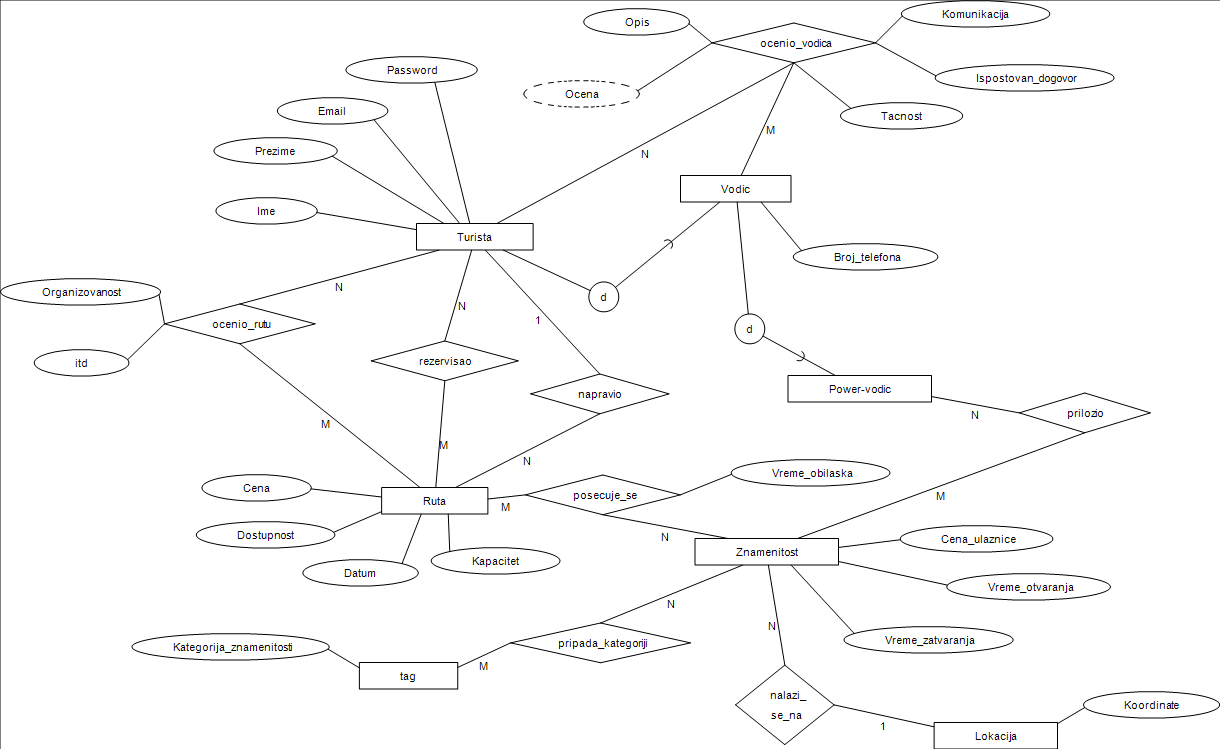
Model domena za koji se Tripko web aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (JS skripti i C# dotnet klasa) koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *SharpEER* .

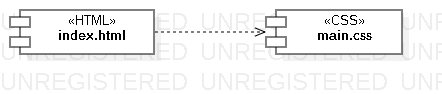


## Komponente sistema

Komponente sistema Tripko web aplikaicja su JS fajlovi čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je JS skripta prikazana kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



Komponenta **index.html** implementira pocetnu stranicu web sajta.

Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

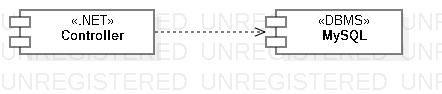
### Komponente aplikacione logike

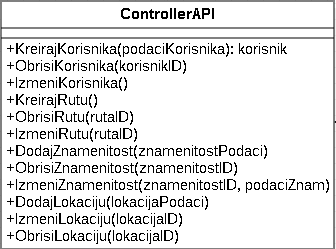
Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru .NET kontrolera.





# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će biti u mogućnosti da podrži do 1000 simultanih pristupa korisnika web stranici.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. Tripko web aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.