Sadržaj

[1. Cilj dokumenta 3](#_Toc6528879)

[2. Opseg dokumenta 3](#_Toc6528880)

[3. Reference 3](#_Toc6528881)

[4. Predstavljanje arhitekture 3](#_Toc6528882)

[5. Ciljevi i organizacija arhitekture 3](#_Toc6528883)

[6. Pogledi na slučajeve korišćenja 3](#_Toc6528884)

[6.1 Dijagram slučajeva korišćenja 4](#_Toc6528885)

[6.2 Kratak opis slučajeva korišćenja 4](#_Toc6528886)

[6.2.1 Registracija 4](#_Toc6528887)

[6.2.2 Prijavljivanje 5](#_Toc6528888)

[6.2.3 Pregled i filtriranje ponuda 5](#_Toc6528889)

[6.2.4 Prijavljivanje za vožnju 5](#_Toc6528890)

[6.2.5 Postavljanje nove ponude za vožnju 5](#_Toc6528891)

[6.2.6 Pregled i ocenjivanje profila drugih korisnika 5](#_Toc6528892)

[6.2.7 Ažuriranje podataka o sebi i svojim ponudama za vožnju 5](#_Toc6528893)

[6.2.8 Brisanje naloga korisnika 5](#_Toc6528894)

[6.2.9 Brisanje ponude za vožnju 5](#_Toc6528895)

[7. Pogled na logičku arhitekturu Sistema 6](#_Toc6528896)

[7.1 Korisnički interfejs (sloj) 6](#_Toc6528897)

[7.2 Aplikaciona logika (sloj) 6](#_Toc6528898)

[7.3 Pristup podacima (sloj) 6](#_Toc6528899)

[7.4 HTML (tehnologija) 7](#_Toc6528900)

[7.5 PHP (tehnologija) 7](#_Toc6528901)

[7.6 MySQL (DBMS) 7](#_Toc6528902)

[7.7 React.js (tehnologija) 7](#_Toc6528903)

[8. Pogled na procese 7](#_Toc6528904)

[8.1 Procesi 7](#_Toc6528905)

[8.1.1 Web čitači 8](#_Toc6528906)

[8.1.2 Web server 8](#_Toc6528907)

[8.1.3 PHP 8](#_Toc6528908)

[8.1.4 MySQL server 8](#_Toc6528909)

[9. Pogled na raspoređivanje sistema 9](#_Toc6528910)

[9.1 Klijent 9](#_Toc6528911)

[9.2 Web server 9](#_Toc6528912)

[9.3 DBMS server 9](#_Toc6528913)

[10. Pogled na implementaciju sistema 10](#_Toc6528914)

[10.1 Model domena 10](#_Toc6528915)

[10.2 Šema baze podataka(ER dijagram) 10](#_Toc6528916)

[10.3 Komponente sistema 11](#_Toc6528917)

[10.3.1 Komponente korisničkog interfejsa 11](#_Toc6528918)

[10.3.2 Komponente aplikacione logike 11](#_Toc6528919)

[10.3.3 Komponente za pristup podacima 12](#_Toc6528920)

[11. Performanse 13](#_Toc6528921)

[12. Kvalitet 13](#_Toc6528922)

# Cilj dokumenta

Cilj ovog dokumenta je detaljan opis arhitekture Prevezi Me aplikacije.

# Opseg dokumenta

Dokument se odnosi na Prevezi Me aplikaciju koja će biti razvijena od strane RBP grupe.

Namena je lakše nalaženje jeftinijeg prevoza na različitim relacijama Republike Srbije.

# Reference

1. Prevezi me - Predlog projekta, 03.03.2019, RBP grupa

2. Prevezi me - Vizija sistema, 08.03.2019, RBP grupa

3. Prevezi me - Plan realizacije projekta, 20.03.2019, RBP grupa

4. Prevezi me - Specifikacija zahteva, 31.03.2019, RBP grupa

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao niz 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 organizacija arhitekture

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

1. Prevezi Me web aplikacija će biti zasnovana na React.js framework-u, PHP skriptnom jeziku i MySQL bazi podataka.
2. Klijentski deo Prevezi Me aplikacije će biti optimizovan za sledeće web engine:

* Blink (Google Chrome, Opera)
* Gecko (Mozilla Firefox)
* EdgeHTML, Chakra (Microsoft Edge)
* Webkit (Safari)

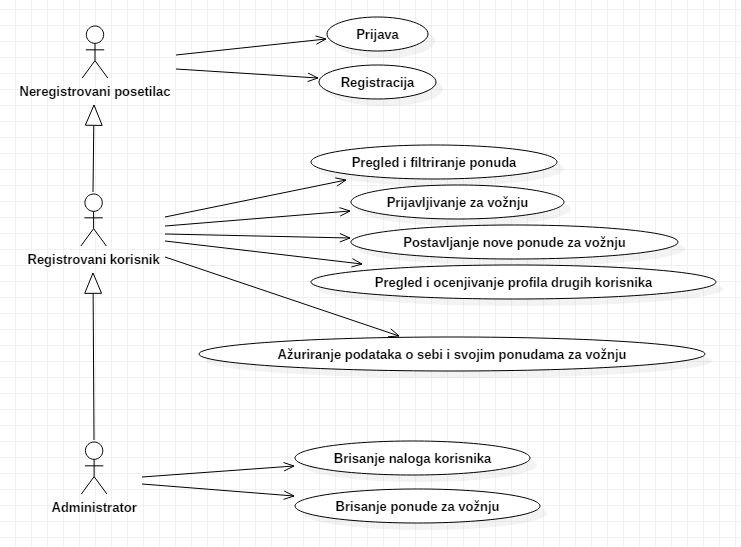
# Pogledi na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva.Slučajevi korišćenja Prevezi Me aplikacije su:

* + - Registracija
    - Prijavljivanje
    - Pregled i filtriranje ponuda
    - Prijavljivanje za vožnju
    - Postavljanje nove ponude za vožnju
    - Pregled i ocenjivanje profila drugih korisnika
    - Ažuriranje podataka o sebi i svojim ponudama za vožnju
    - Brisanje naloga korisnika
    - Brisanje ponude za vožnju

## Dijagram slučajeva korišćenja

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

****

## Kratak opis slučajeva korišćenja

### **Registracija**

Kratak opis: Otvaranje novog naloga za aplikaciju.

Akteri: Neregistrovani posetilac

### **Prijavljivanje**

Kratak opis: Prijavljivanje na aplikaciju u cilju pristupa glavnim funkcijama aplikacije.

Akteri: Neregistrovani posetilac, korisnik i administrator

### **Pregled i filtriranje ponuda**

Kratak opis: Prikaz stranice sa trenutnim ponudama prevoza.

Akteri: Korisnik i administrator

### **Prijavljivanje za vožnju**

Kratak opis: Potvrđivanje od strane korisnika da se želi prijaviti za željenu ponudu.

Akteri: Korisnik i administrator

### **Postavljanje nove ponude za vožnju**

Kratak opis: Objavljivanje svih potrebnih informacija vezanih za nuđenu vožnju

Akteri: Korisnik i administrator

### **Pregled i ocenjivanje profila drugih korisnika**

Kratak opis: Pregled informacija o drugim korisnicima i ocenjivanje.

Akteri: Korisnik i administrator

### **Ažuriranje podataka o sebi i svojim ponudama za vožnju**

Kratak opis: Ažuriranje podataka o sebi i svojim ponudama za vožnju

Akteri: Korisnik i administrator

### **Brisanje naloga korisnika**

Kratak opis: Brisanje naloga korisnika od strane administratora.

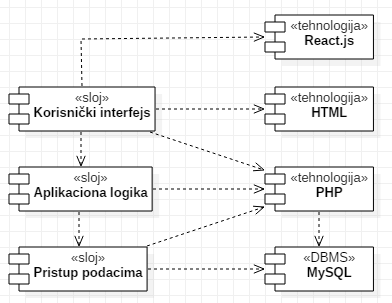
Akteri: Administrator

### **Brisanje ponude za vožnju**

Kratak opis: Brisanje ponude za vožnju od strane administratora.

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 klase.

## Korisnički interfejs (sloj)

Ovaj sloj realizuje korisnički interfejs portala. U njemu su sadržane sve HTML, multimedijalni sadržaji, React.js skripte i PHP skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

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

## Aplikaciona logika (sloj)

Sloj aplikacione logike sadrži PHP skripte koje realizuju funkcionalnost karakterističnu za domen primene portala i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima. Ovaj sloj zavisi od sloja za pristup podacima i PHP paketa.

## Pristup podacima (sloj)

Sloj za pristup podacima sadrži PHP skripte 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 PHP 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.

## PHP (tehnologija)

Tehnologija PHP-a obezbeđuje mehanizam za pisanje i izvršavanje skripti na strani servera. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs i pristupaju bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

## MySQL (DBMS)

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Prevezi-me aplikacije.

## React.js (tehnologija)

React.js je JavaScript framework koji generiše većinu HTML koda korisničkog interfejsa web aplikacije.

# Pogled na procese

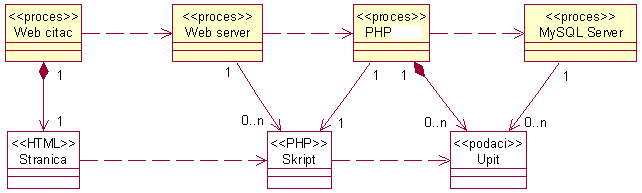
Ovaj odeljak sadrži pogled na procesnu arhitekturu sistema. Ovaj opis treba sadržati specifikaciju različitih zadataka koji su uključeni u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa određenim zadacima takođe spada u opis procesne arhitekture.

Web aplikacije koje su zasnovane na PHP-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa pogleda projektanta PHP Web aplikacije, nije potrebno voditi računa o načinu rada Web servera i načinu izvršavanja skripti.

Primera radi, u nastavku je dat opis procesa uključenih u izvršenju Prevezi-me kao Web aplikacije.

## Procesi

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



### **Web čitači**

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

Web čitač 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 čitača. Ukoliko je zahtevana stranica PHP skripta, Web server inicira izvršenje PHP procesa koji obrađuje odgovarajuću skriptu i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj PHP procesa.

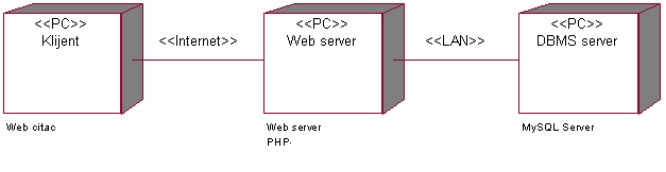
### **PHP**

PHP proces obavlja posao obrade zadate PHP skripte i generiše odgovarajući tekstualni sadržaj koji Web server šalje Web čitaču. Za izvršenje PHP skripte ovaj proces može da zahteva usluge MySQL servera. Komunikacija između PHP procesa i MySQL servera se obavlja preko prosleđivanja upita i vraćanja rezultata.

### **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



Na prethodnoj slici dat je UML diagram raspoređivanja ‘Prevezi me’ aplikacije

## 9.1 Klijent

Pristup ‘Prevezi me’ aplikaciji se obavlja preko klijentskih računara na kojima se izvršava Web čitač. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## 9.2 Web server

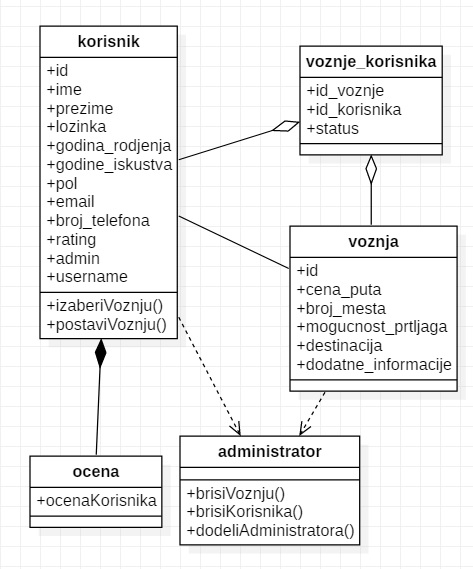
Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi u PHP-u koji vrše obradu zadatih PHP skripti. U najopštioj konfiguraciji DBMS se izvršava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

## 9.3 DBMS server

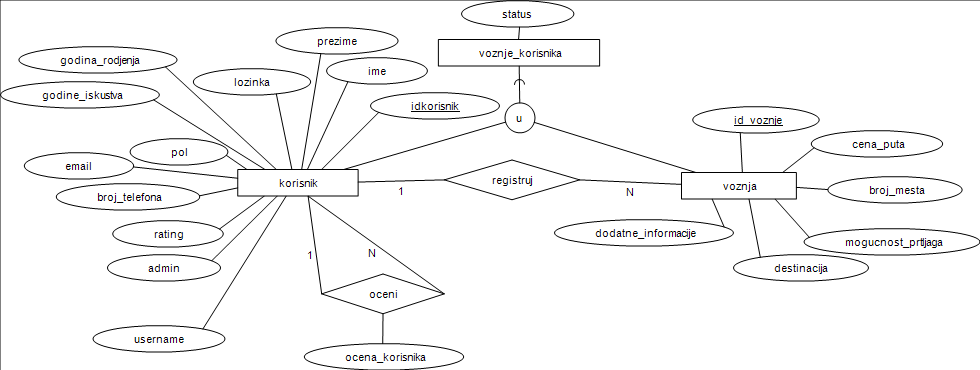
DBMS server je računar na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

# Pogled na implementaciju sistema

## Model domena



## Šema baze podataka(ER dijagram)

****

## Komponente sistema

Komponente sistema Prevezi-me aplikacije su PHP skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i klasni dijagrami. U slučajevima gde je PHP skripta prikazan kao klasa, atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije koje su definisane u okviru skripte.

### **Komponente korisničkog interfejsa**

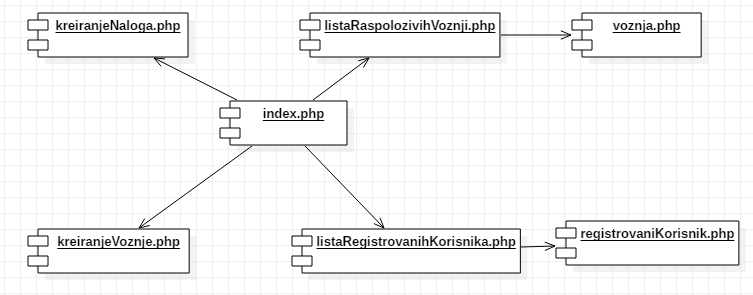
Dizajn korisničkog interfejsa je obuhvaćen sledećim komponentama:



Komponenta **index.php** implementira stranicu aplikacije čiji sadržaj može da varira od parametra koji joj se prosledi u zahtevu.

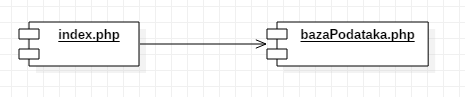
Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

### **Komponente aplikacione logike**

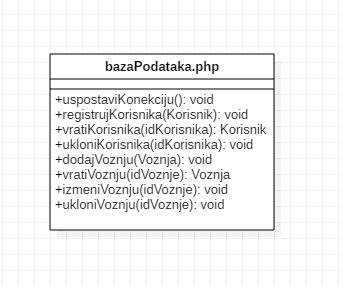
Komponente koje realizuju domen problema su omogućene isključivo preko **index.php** komponente korisničkog interfejsa. Na taj način se zadržavaju sva podešavanja stila koja su definisana u ovoj skripti. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:

### **Komponente za pristup podacima**

Pristup bazi podataka je u potpunosti zatvoren unutar funkcija definisanih u okviru PHP skripte **bazaPodataka.php**. Pomenuta skripta se uključuje na početku index.php-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



Na sledećem UML dijagramu klasa se nalaze funkcije za pristup podacima iz baze:



Iako su funkcije imenovane na takav način da se prostim pogledom na njihove nazive može jednoznačno zaključiti šta rade, ipak će dejstvo par njih biti navedeno:

* uspostaviKonekciju() – uspostavlja konekciju sa bazom podataka radi manipulacije nad podacima
* registrujKorisnika(Korisnik) – u bazu podataka se ubacuje novi korisnik koji nakon registracije može koristiti ovu aplikaciju
* ukloniKorisnika(idKorisnika) – na osnovu jedinstvenog identifikatora se pronalazi korisnik koji se uklanja iz baze
* izmeniVoznju(idVoznje) – na osnovu jedinstvenog identifikatora se menjaju detalji vožnje pod tim identifikatorom i u skladu sa tim ažurira u bazi

# Performanse

Izabrana arhitektura softvera bi trebala da podrži do 2419 simultanih pristupa, a vreme potrebno za pristupanje bazi podataka, u cilju izvršavanja nekog upita, ne sme biti veća od 5 minuta. Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen.

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza navedenih u zahtevima u pogledu pouzdanosti:

1. Prevezi-me servis će biti dostupan 24 časa dnevno, 7 dana u nedelji, 365 dana u godini. Vreme kada portal nije dostupan ne sme da pređe 1 dan
2. Srednje vreme između dva uzastopna otkaza ne sme pasti ispod 120 sati.