RENT-A-BIKE

Veb aplikacija za iznajmljivanje bicikala

Detaljni arhitekturni projekat

Verzija 2.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 16.11.2021. | 1.0 | Inicijalna verzija | Almedina Hasanovic |
| 19.11.2021. | 2.0 | Konačna verzija | Almedina Hasanovic |
|  |  |  |  |
|  |  |  |  |

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 6

6.2 Kratak opis slučajeva korišćenja 8

6.2.1 Registracija 6

6.2.2 Prijavljivanje 7

6.2.3 Prikaz informacija 7

6.2.4 Brisanje korisnika 8

6.2.5 Ažuriranje informacija 8

6.2.6 Ažuriranje osnovnih podataka 9

6.2.7 Promena avatara 9

6.2.8 Promena lozinke 10

6.2.9 Unos nove fotografije 10

6.2.10 Brisanje fotografije 11

6.2.11 Pretraga servisera 12

6.2.12 Pretraga korisnika 13

6.2.13 Kreiranje obaveštenja 14

6.2.14 Brisanje obaveštenja 15

6.2.15 Kreiranje termina 15

6.2.16 Brisanje termina 16

6.2.17 Dodavanje kategorije 17

6.2.18 Zakazivanje termina 17

6.2.19 Otkazivanje termina

6.2.20 Unos korisnika

6.2.21 Ažuriranje korisnika 18

6.2.22 Brisanje korisnika 18

7. Pogled na logičku arhitekturu sistema 9

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 10

7.1.4 HTML 10

7.1.5 PHP 10

7.1.6 JavaScript 11

7.1.7 MySQL 11

8. Pogled na procese 11

8.1 Procesi 11

8.1.1 Web čitač 11

8.1.2 Web server 11

8.1.3 PHP-CGI 11

8.1.4 MySQL Server 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 Komponente sistema 13

10.2.1 Komponente korisničkog interfejsa 13

10.2.2 Komponente aplikacione logike 13

10.2.3 Komponente za pristup podacima 14

11. Performanse 14

12. Kvalitet 14

Detaljni arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture RENT-A-BIKE aplikacije.

# Opseg dokumenta

Dokument se odnosi na RENT-A-BIKE veb aplikaciju koja će biti razvijena od strane DinaTeam-a. RENT-A-BIKE je veb aplikacija namenjena za iznjamljivanje ili zakazivanje bicikala. Namena sistema je efikasno prezentovanje, kreiranje, rezervacije termina i održavanje sadržaja vezanih za rad aplikacije.

# Reference

Spisak korišćene literature:

1. Definicija projekta \_RENT-A-BIKE\_final, V1.0, 2021, DinaTeam .
2. Plan realizacije projekta\_RENT-A-BIKE\_final, V1.0, 2021, DinaTeam .
3. Raspored\_Aktivnosti\_RENT-A-BIKE\_final, V1.0, 2021, DinaTeam .
4. Vizija sistema \_RENT-A-BIKE\_final, V1.0, 2021, DinaTeam .
5. Specifikacija Zahteva \_RENT-A-BIKE\_final, V1.0, 2021, DinaTeam .

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, 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. RENT-A-BIKE aplikacija će biti implementiran kao Web aplikacija zasnovana na PHP skripting jeziku, JS-u i MySQL bazi podataka [4].
2. Klijentski deo RENT-A-BIKE aplikacija će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, 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. Klasifikaciju publikacija treba standardizovati sa postojećom tipologijom propisanom od strane Ministarstva za nauku.

# Pogled na slučajeve korišćenja

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

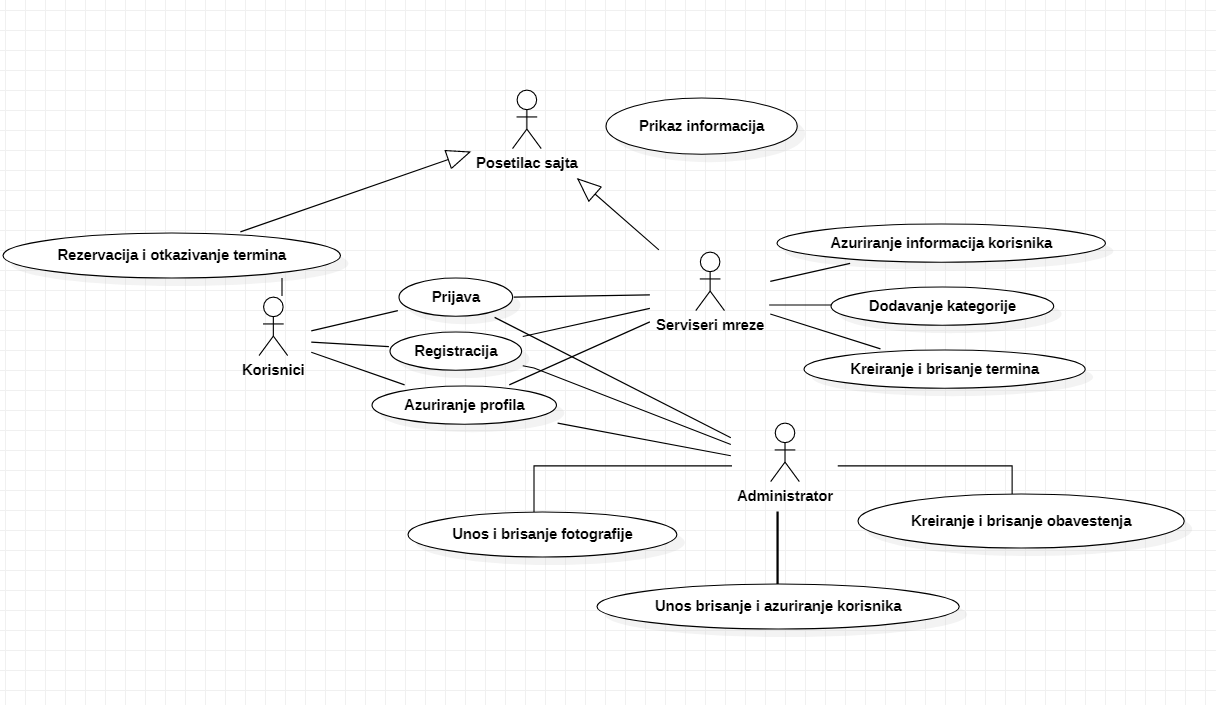
Slučajevi korišćenja RENT-A-BIKE aplikacije su:

* Registracija
* Prijavljivanje
* Prikaz informacija
* Ažuriranje informacija korisnika
  + *Brisanje korisnika*
  + *Ažuriranje korisnika*
* Ažuriranje profila
  + *Ažuriranje osnovnih podataka*
  + *Promena avatara*
  + *Promena lozinke*
* Unos i brisanje fotografije
  + *Unos nove fotografije*
  + *Brisanje fotografije*
* Pretraga servisra i korisnika
  + *Pretraga servisera*
  + *Pretraga korisnika*
* Kreiranje i brisanje obaveštenja
  + *Kreiranje obaveštenja*
  + *Brisanje obaveštenja*
* Kreiranje i brisanje termina
  + *Kreiranje termina*
  + *Brisanje termina*
* Dodavanje kategorije
* Zakazivanje i otkazivanje termina
  + *Zakazivanje termina*
  + *Otkazivanje termina*
* Unos, ažuriranje i brisanje korisnika
  + *Unos korisnika*
  + *Ažuriranje korisnika*
  + *Brisanje korisnika*

Ove slučajevi korišćenja mogu da iniciraju administrator, serviser, posetilac ili korisnik.

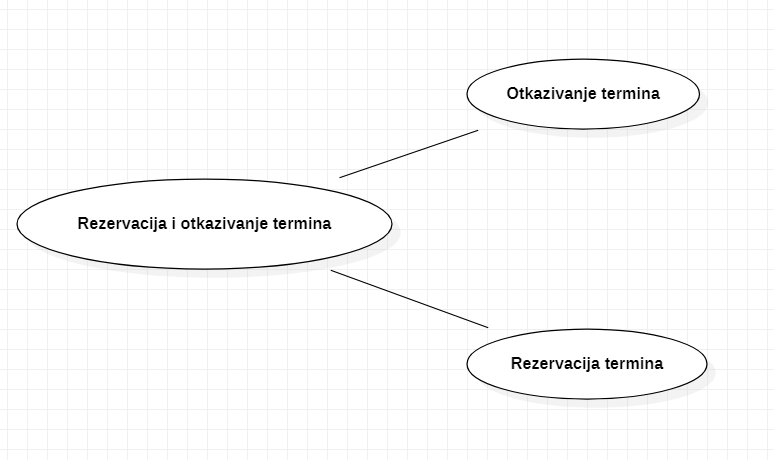
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja RENT-A-BIKE aplikacije prikazan je na sledećoj slici:

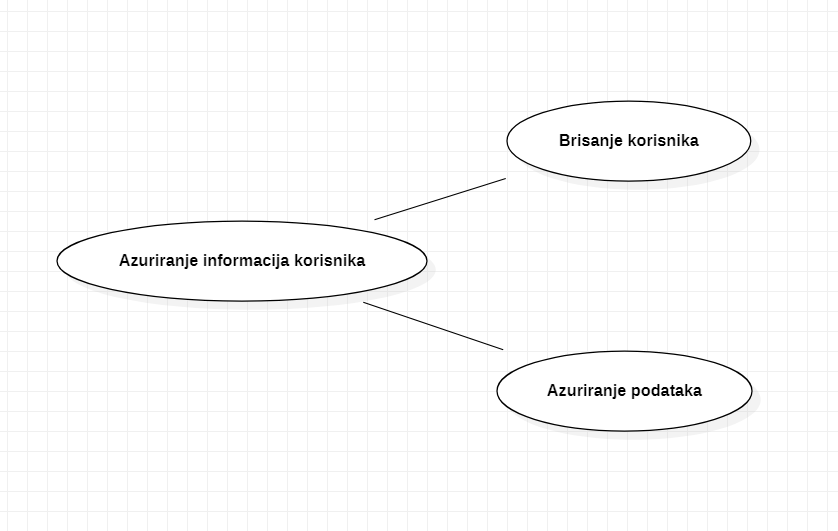


Slučajevi korišćenja  *rezervacija i otkazivanje termina,*  *ažuriranje informacija korisnika, ažuriranje profila, unos i brisanje fotografije, pretraga servera i korisnika, kreiranje i brisanje obaveštenja i termina i unos, brisanje i ažuriranje korisnika* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

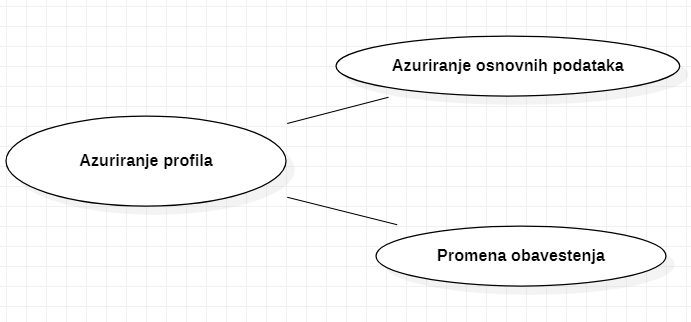
Detaljni UML dijagram za slučaj korišćenja *rezervacija i otkazivanje termina* je prikazan na sledećoj slici:



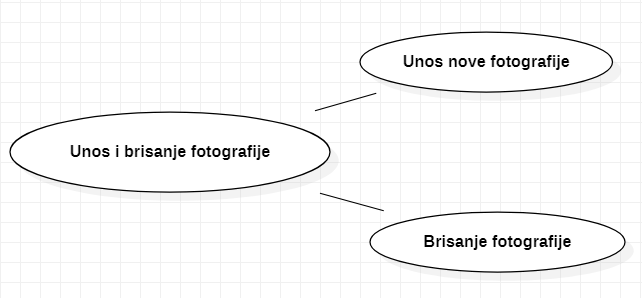
Detaljni UML dijagram za slučaj korišćenja *ažuriranje informacija korisnika* je prikazan na sledećoj slici:



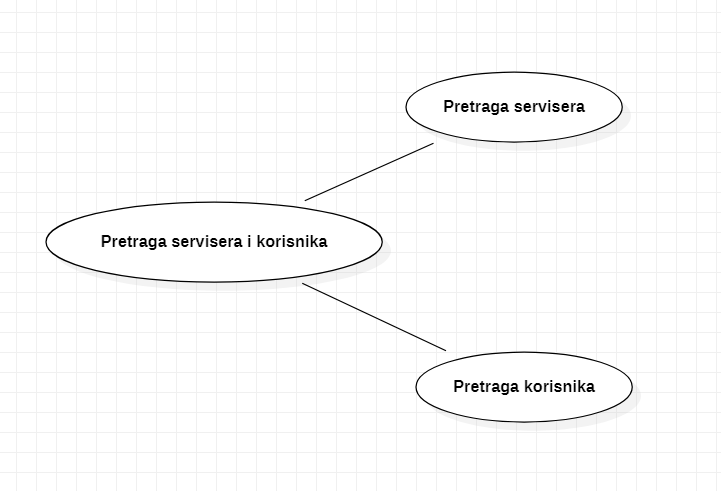
Detaljni UML dijagram za slučaj korišćenja *ažuriranje profila* je prikazan na sledećoj slici:



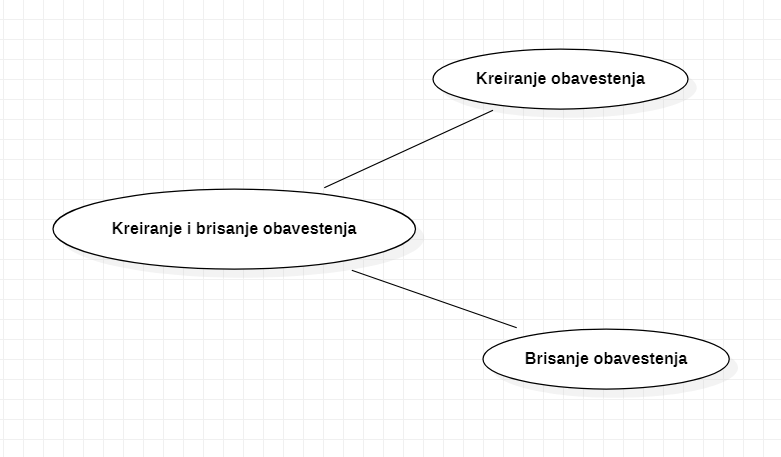
Detaljni UML dijagram za slučaj korišćenja *unos i brisanje fotografije* je prikazan na sledećoj slici:



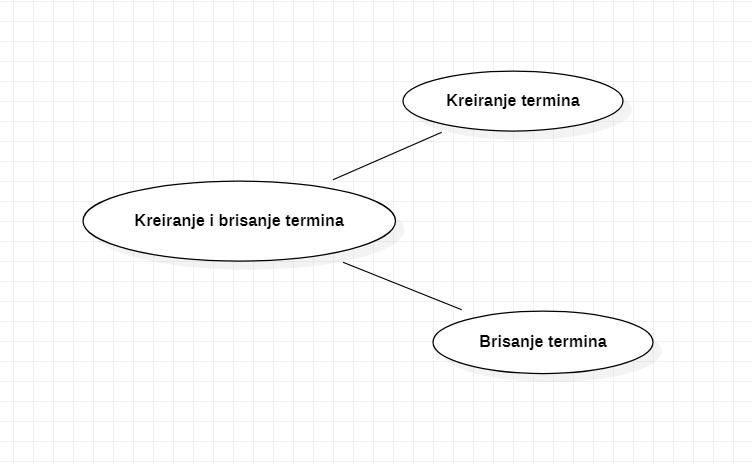
Detaljni UML dijagram za slučaj korišćenja *pretraga servisera i korisnika*  je prikazan na sledećoj slici:



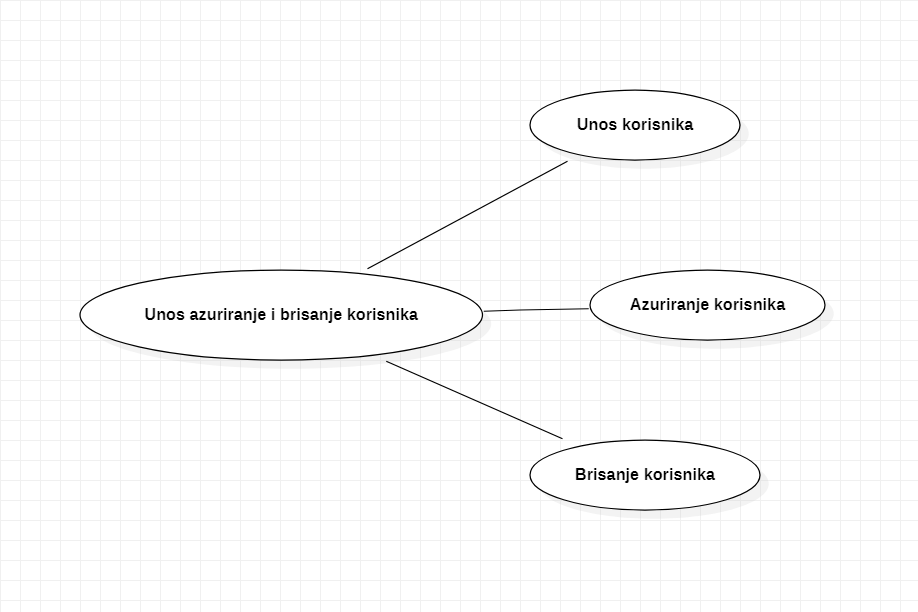
Detaljni UML dijagram za slučaj korišćenja *kreiranje i brisanje obaveštenja* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *kreiranje i brisanje termina* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *unos, ažuriranje i brisanje korisnika* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

## *6.2.1 Registracija*

Kratak opis: Korisnik sajta unosi svoje podatke. Unošenjem korisničkog imena, lozinke i ličnih podataka kao što su mesto, država, jmbg, email, pol, korisnik kreira svoj lični nalog.

Akteri koji iniciraju slučaj korišćenja: korisnik sajta.

## *6.2.2. Prijavljivanje*

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

Akteri koji iniciraju slučaj korišćenja: Administrator, serviser, posetilac i korisnik.

## *6.2.3. Prikaz informacija*

Kratak opis: Prikaz stranice sistema sa osnovnim podacima o aplikaciji

Akteri koji iniciraju slučaj korišćenja: administrator, serviser, posetilac i korisnik.

## *6.2.4. Ažuriranje informacija korisnika*

Kratak opis: Izmena i unos podataka u informacijama korisnika.

Akteri koji iniciraju slučaj korišćenja: Serviser.

## *6.2.5. Brisanje korisnika*

Kratak opis: Brisanje korisnika iz baze od strane servisera.

Akteri koji iniciraju slučaj korišćenja: Serviser.

## *6.2.6. Ažuriranje osnovnih podataka*

Kratak opis: Izmena osnovnih podataka o ulogovanom korisniku.

Akteri koji iniciraju slučaj korišćenja: Administrator, serviser, posetilac ili korisnik.

## *6.2.7. Promena avatara*

Kratak opis: Izmena profilne fotografije ulogovanog korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator, serviser, posetilac ili korisnik.

## *6.2.8. Promena lozinke*

Kratak opis: Promena lozinke ulogovanog korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator, serviser, posetilac ili korisnik.

## *6.2.9. Unos nove fotografije*

Kratak opis: Unos nove fotografije u galeriju.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.10. Brisanje fotografije*

Kratak opis: Brisanje postojeće fotografije iz galerije.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.11. Pretraga servera*

Kratak opis: Prikaz stranica sa listom servera mogućnost njihove pretrage po imenu ili prezimenu.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.12. Pretraga korisnika*

Kratak opis: Prikaz stranica sa listom korisnika i mogućnost njihove pretrage po imenu ili prezimenu.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.13. Kreiranje obaveštenja*

Kratak opis: Dodavanje novog obaveštenja od strane administratora.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.14. Kreiranje termina*

Kratak opis: Dodavanje novog termina od strane servisera.

Akteri koji iniciraju slučaj korišćenja: Serviser.

## *6.2.15. Brisanje obaveštenja*

Kratak opis: brisanje obaveštenja od strane administratora.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.16. Brisanje termina*

Kratak opis: brisanje termina od strane servisera.

Akteri koji iniciraju slučaj korišćenja: Serviser.

## *6.2.17. Ažuriranje korisnika*

Kratak opis: Ažuriranje podataka o bilo kom registrovanom korisniku od strane administratora.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.18. Brisanje korisnika*

Kratak opis: Biranje postojećeg korisnika od strane administratora.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.19. Unos korisnika*

Kratak opis: Unos novog korisnika u sisem od strane administratora.

Akteri koji iniciraju slučaj korišćenja: Administrator.

## *6.2.20. Dodavanje kategorije*

Kratak opis: Unos novog korisnika u sisem od strane administratora.

Akteri koji iniciraju slučaj korišćenja: Serviser.

## *6.2.21. Zakazivanje termina*

Kratak opis: Zakazivanje slobodnog termina od strane korisnika kod servisera koji je kreirao isti.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

## *6.2.22. Otkazivanje termina*

Kratak opis: Otkazivanje zakazanog termina od strane korisnika.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

# 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.

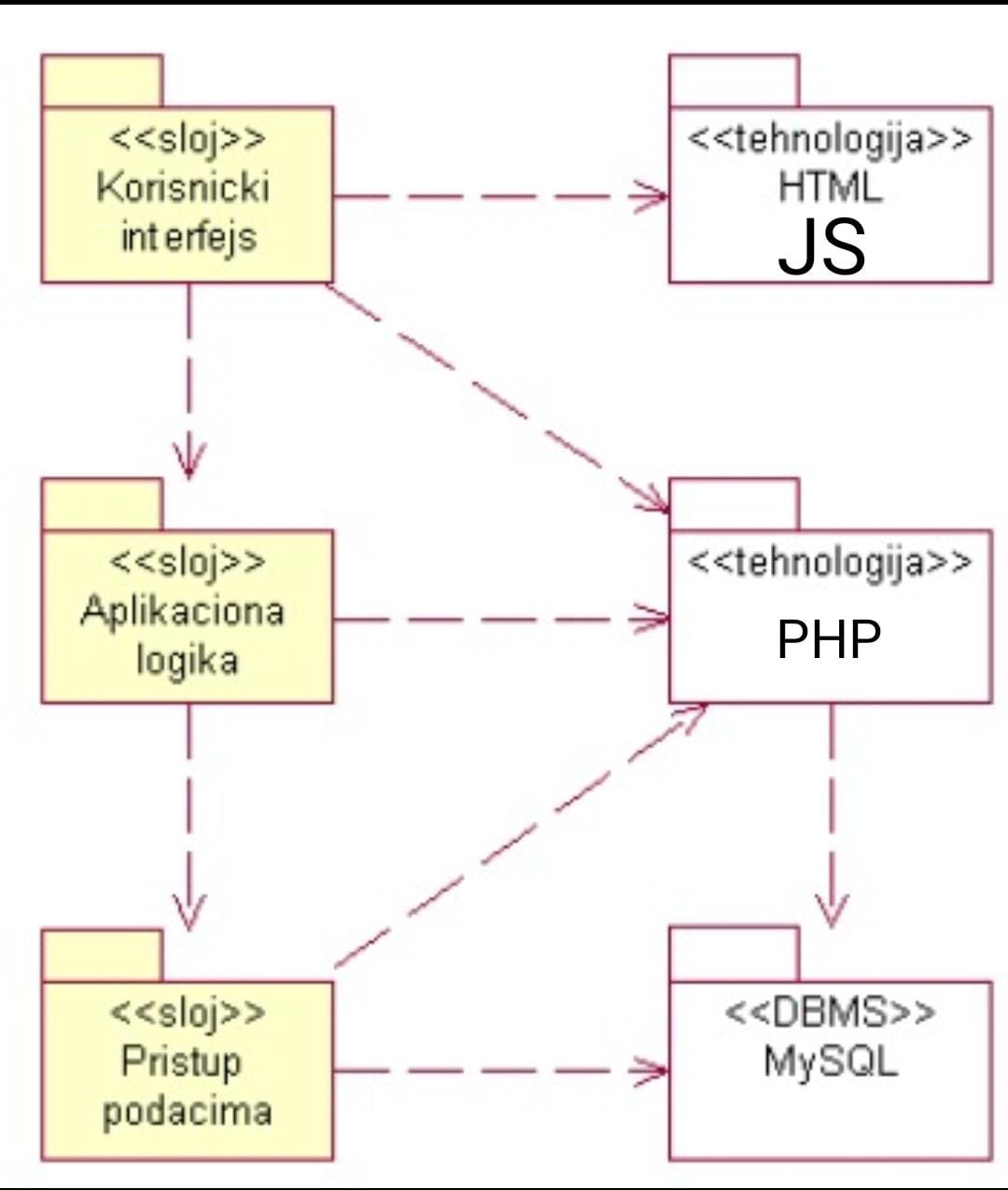
Logički pogled na Bibliophile portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

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

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži PHP skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži PHP skripte koje predstavljaju 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 portala. U njemu su sadržane sve HTML, multimedijalni sadržaji, JS 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, JS i PHP.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi RENT-A-BIKE aplikacije. Sadrži PHP skripte koje realizuju funkcionalnost karakterističnu za domen primene aplikacijei 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 se nalazi na dnu troslojne arhitekture i 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.

### JavaScript

tehnologija

Tehnologija JavaScript-a obezbeđuje mehanizam za pisanje i izvršavanje skripti. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs..

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju RENT-A-BIKE aplikacije.

# Pogled na procese

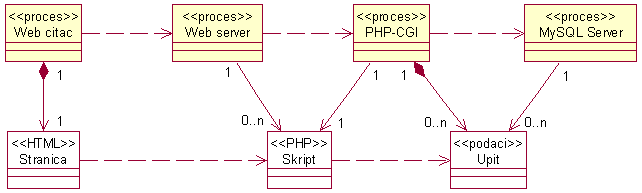
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

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

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje RENT-A-BIKE aplikacije kao Web aplikacije.

## Procesi

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



### Web čitač

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 da prikazuje 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 skript, Web server inicira izvršenje PHP-CGI procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj PHP-CGI procesa.

### PHP-CGI

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

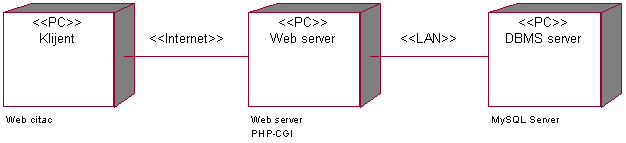
### MySQL Server

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 RENT-A-BIKE aplikacije.



## Klijent

Pristup RENT-A-BIKE 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.

## 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 PHP-CGI koji vrše obradu zadatih PHP 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).

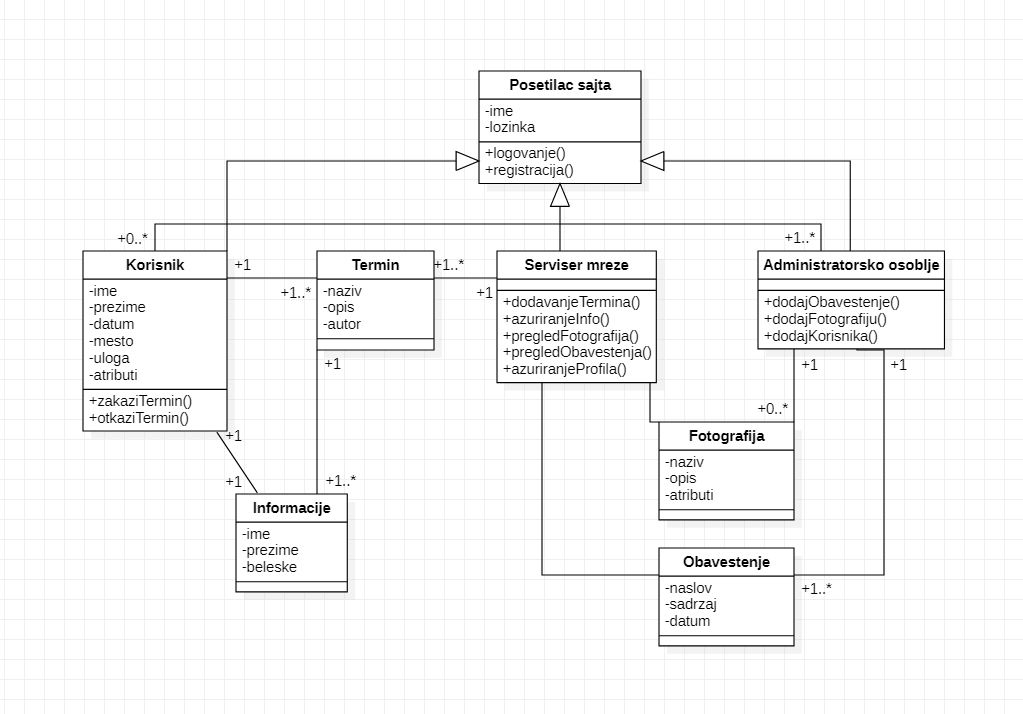
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju RENT-A-BIKE 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 RENT-A-BIKE 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 (PHP skripti) koje će biti implementirane.

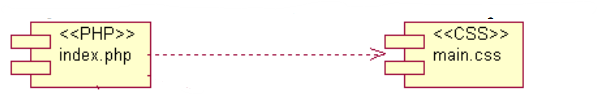


## Komponente sistema

Komponente sistema RENT-A-BIKE aplikacija su PHP skripti čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je PHP skript prikazan 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:

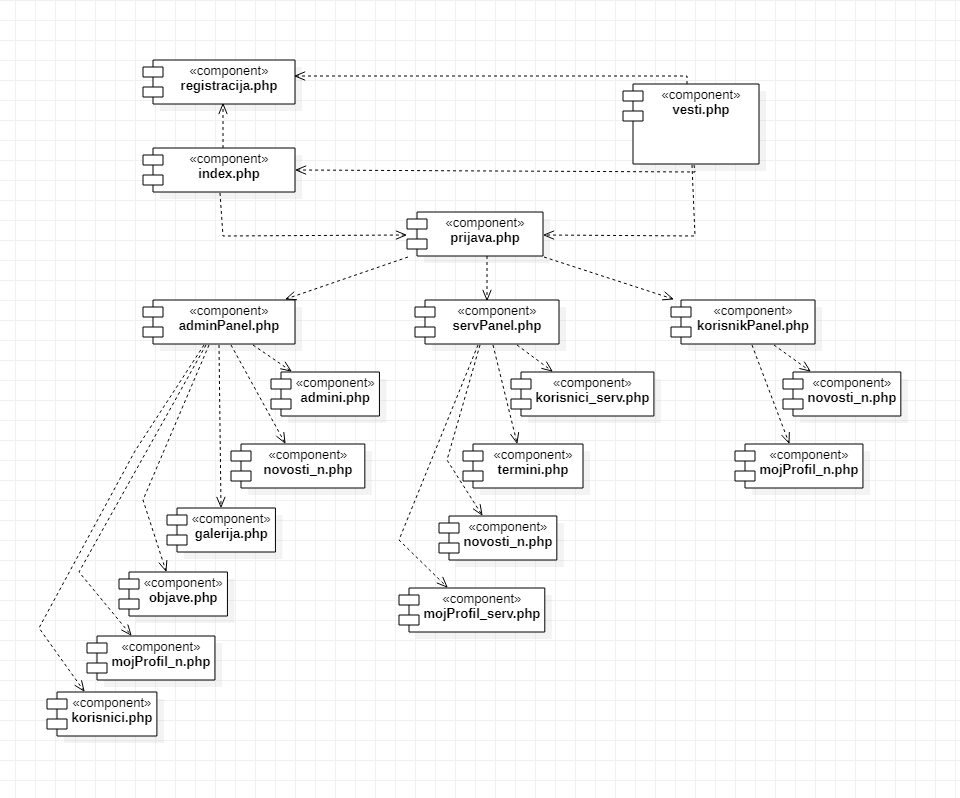


Komponenta **index.php** je implementira stranicu aplikacije čiji sadržaj može da varira od parametara koji joj se proslede u zahtevu.

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

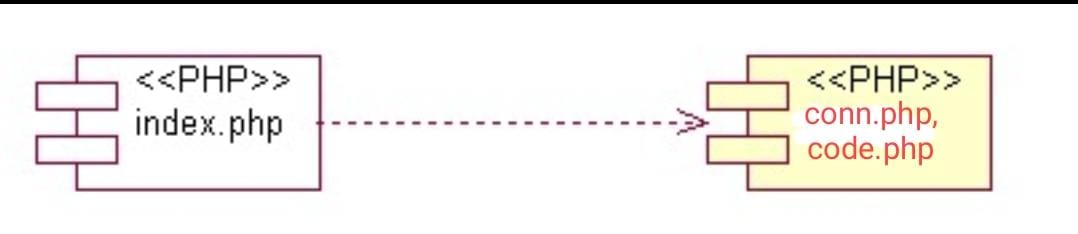
### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju preko **index.php** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. 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 PHP skripta **conn.php i code.php**. Pomenuti skript se uključuje na početku index.php-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



# 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 da podrži do 1000 simultanih pristupa korisnika portalu.
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. RENT-A-BIKE aplikacija će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada portal nije dostupan ne sme da pređe 10%.

Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.