Ready Vet

Sistem za poboljšanje rada veterinarske stanice

Arhitekturni projekat

Verzija 2.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 30.04.2020. | 1.0 | Inicijalna verzija | Pet Friends |
| 29.06.2020. | 2.0 | Ispravka | Nikola Lozo |
|  |  |  |  |
|  |  |  |  |

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 8

6.2.2 Prijavljivanje 8

6.2.3 Pregled osnovnih podataka o veterinarskoj stanici 8

6.2.4 Pregled spiska veterinara 8

6.2.5 Pregled podataka o određenom veterinaru 8

6.2.6 Pregled spiska usluga 9

6.2.7 Pregled podataka o određenoj usluzi 9

6.2.8 Pregled zdravstvenog kartona kućnog ljubimca 9

6.2.9 Online zakazivanje usluge 9

6.2.10 Dodavanje slike 9

6.2.11 Odobravanje usluga u zakazanom terminu 9

6.2.12 Pregled sopstvenog profila 9

6.2.13 Uvid u odgovor veterinara 9

6.2.14 Odobravanje registrovanja veterinara 9

6.2.15 Uređivanje zdravstvenih kartona 9

6.2.16 Uvid u ocene i komentare 9

6.2.17 Pregled obaveza 10

6.2.18 Pregled profila korisnika 10

6.2.19 Brisanje profila korisnika 10

6.2.20 Pregled profila veterinara 10

6.2.21 Brisanje profila veterinara 10

6.2.22 Uređivanje usluga 10

6.2.23 Dodavanje nove usluge 10

6.2.24 Brisanje usluge 10

6.2.25 Uvid u angažovanja 10

7. Pogled na logičku arhitekturu sistema 10

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 11

7.1.1 Korisnički interfejs 11

7.1.2 Aplikaciona logika 11

7.1.3 Pristup podacima 11

7.1.4 Angular 11

7.1.5 Node.js 12

7.1.6 MongoDB 12

8. Pogled na procese 12

8.1 Procesi 12

8.1.1 Web čitač 12

8.1.2 Web server 12

8.1.3 MongoDB Server 13

9. Pogled na raspoređivanje sistema 13

9.1 Klijent 13

9.2 Web server 13

9.3 DBMS server 13

10. Pogled na implementaciju sistema 13

10.1 Model domena 13

10.2 Šema baze podataka 14

10.3 Komponente sistema 15

10.3.1 Komponente korisničkog interfejsa 15

10.3.2 Komponente aplikacione logike 16

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 Ready Vet portala.

# Opseg dokumenta

Dokument se odnosi na Ready Vet portal koji će biti razvijen od strane Pet Friends. Ready Vet predstavlja pseudonim za sistem čija je svrha poboljšanje rada veterinarske stanice. Namena sistema je efikasno prezentovanje rada veterinarske stanice, online zakazivanje pregleda kod veterinara i lakša komunikacija između korisnika, veterinara i uprave u veterinarskoj stanici.

# Reference

Spisak korišćene literature:

1. Ready Vet – Predlog projekta, SWE\_01\_Predlog\_Projekta, V1.0, 2020, Pet Friends.
2. Ready Vet– Vizija sistema, SWE\_02\_Vizija\_Sistema, V2.0, 2020, Pet Friends.
3. Ready Vet – Plan realizacije projekta, SWE\_03\_Plan\_Realizacije, V1.0, 2020, Pet Friends.
4. Ready Vet – Specifikacija zahteva, SWE\_04\_Specifikacija\_Zahteva, V2.0, 2020, Pet Friends.

# 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. Ready Vet portal će biti implementiran kao Web aplikacija zasnovana na Angular-u, NodeJS-u (NestJS framework) i MongoDB bazi podataka [4].
2. Klijentski deo Ready Vet portala će biti optimizovan za savremene Web čitače. [4].
3. Svi zahtevi u pogledu performansi dati u [5] morajubitiuzetiu obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

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

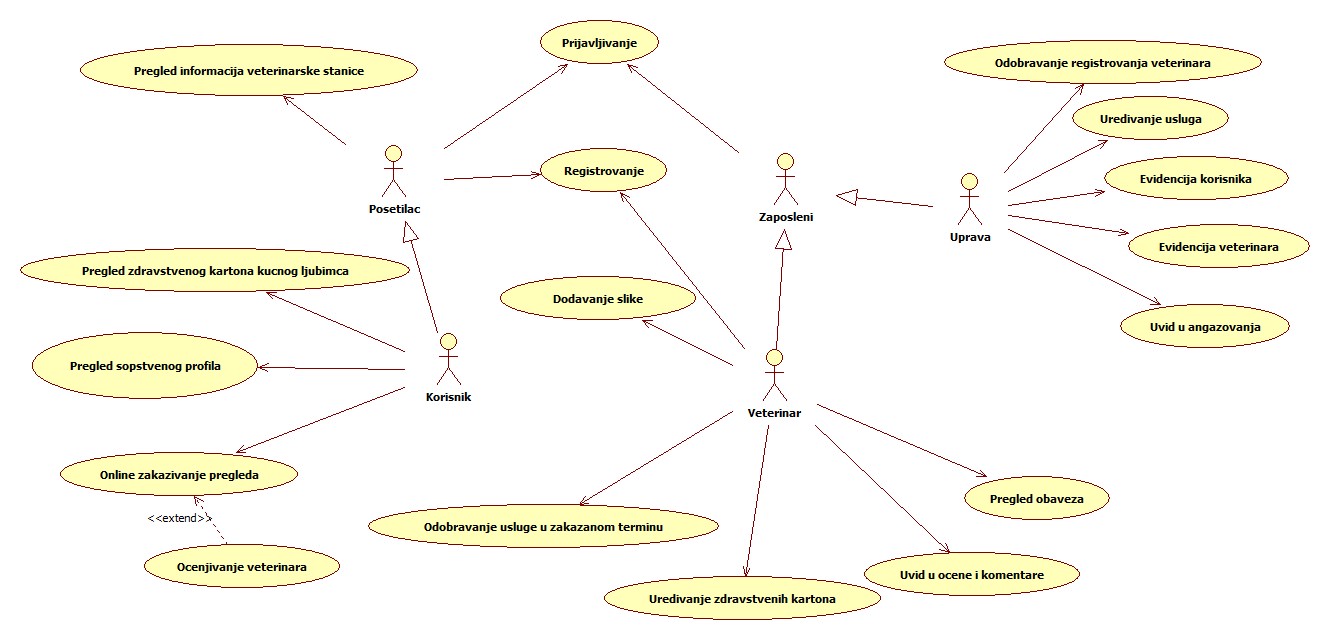
Slučajevi korišćenja Ready Vet portala su:

* Registracija
* *Prijavljivanje*
* *Pregled informacija veterinarske stanice*
  + Pregled osnovnih podataka o veterinarskoj stanici
  + *Pregled podataka o veterinarima* 
    - Pregled spiska veterinara
    - Pregled podataka o određenom veterinaru
  + *Pregled podataka o uslugama* 
    - Pregled podataka o određenoj usluzi
    - Pregled spiska usluga
* Online zakazivanje pregleda
* Uvid u odgovor veterinara
* Odobravanje usluga u zakazanom terminu
* Uređivanje zdravstvenih kartona
* Uvid u ocene i komentare
* Pregled obaveza
* Dodavanje slike
* *Evidencija profila korisnika*
  + Brisanje profila korisnika
  + Pregled profila korisnika
* *Evidencija profila veterianra*
  + Brisanje profila veterinara
  + Pregled profila veterinara
* *Ažuriranje usluga*
  + Uređivanje usluga
  + Dodavanje nove usluge
  + Brisanje usluge
* Pregled zdravstvenog kartona kućnog ljubimca
* Uvid u angažovanja
* Odobravanje registrovanja veterinara
* Pregled sopstvenog profila

Ove slučajevi korišćenja mogu da iniciraju posetilac portala, korisnik, veterinar i uprava.

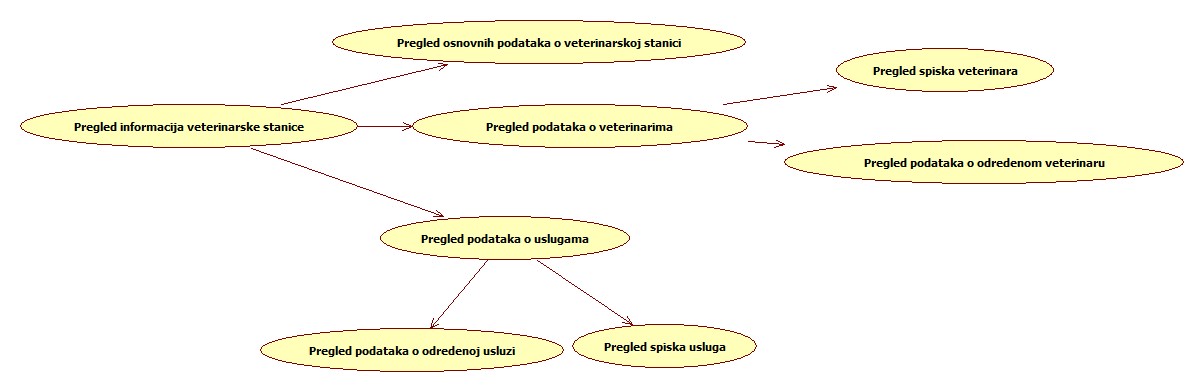
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Ready Vet portala prikazan je na sledećoj slici:

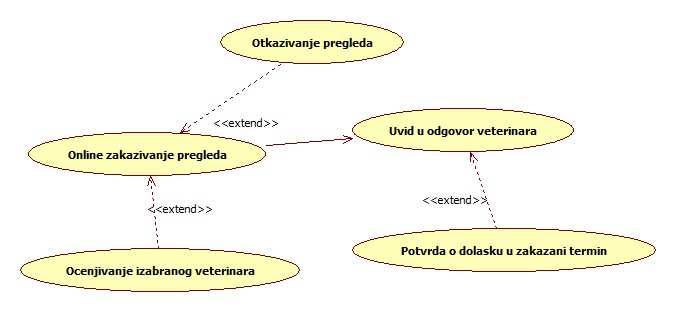


Slučajevi korišćenja *pregled informacija veterinarske stanice*, *online zakazivanje pregleda, evidencija korisnika, evidencija veterinara i ažuriranje usluga* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

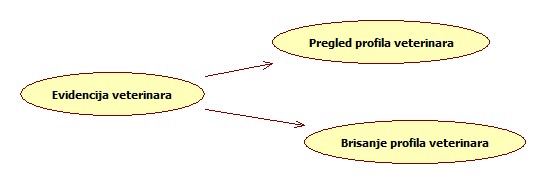
Detaljni UML dijagram za slučaj korišćenja *pregled informacija veterinarske stanice* je prikazan na sledećoj slici:



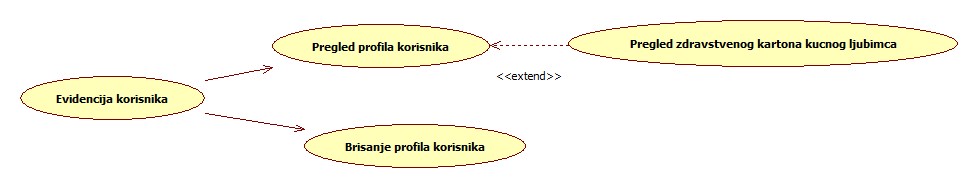
Detaljni UML dijagram za slučaj korišćenja *online zakazivanje pregleda* je prikazan na sledećoj slici:



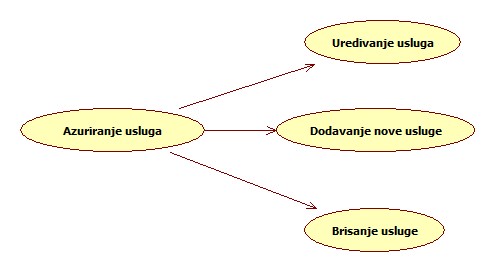
Detaljni UML dijagram za slučaj korišćenja *evidencija veterinara* je prikazan na sledećoj slici:



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



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



## Kratak opis slučajeva korišćenja

### Registracija

Kratak opis:Prikaz stranice portala sa formom za unos osnovnih ličnih podataka i mogućnost kreiranja koriničkog i veterinarskog naloga koji nudi dodatne servise.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala.

### Prijavljivanje

Kratak opis:Prijava na portal koristeći mejl ili lozinku.

Akteri koji iniciraju slučaj korišćenja: Zaposlen(Veterinar, Uprava), Korisnik.

### Pregled osnovnih podataka o veterinarskoj stanici

Kratak opis:Prikaz početne stranice portala sa osnovnim informacijama kao što je fotografija i tekst dobrodošlice.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, korisnik.

### Pregled spiska veterinara

Kratak opis:Prikaz stranice sa spiskom veterinara zaposlenih u stanici.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, korisnik.

### Pregled podataka o određenom veterinaru

Kratak opis:Prikaz stranice sa informacijama o određenom veterinaru, uključujući ocene i komentare drugih korisnika.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, korisnik.

### Pregled spiska usluga

Kratak opis:Prikaz stranice sa spiskom usluga koje veterinarska stanica nudi.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, korisnik.

### Pregled podataka o određenoj usluzi

Kratak opis:Prikaz stranice sa detaljnim opisom usluge, cenom, ukoliko je potrebno i predostrožnostima za vlasnike.

Akteri koji iniciraju slučaj korišćenja: Posetilac portala, korisnik.

### Pregled zdravstvenog kartona kućnog ljubimca

Kratak opis:Prikaz evidencije lečenja kućnog ljubimca.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Online zakazivanje usluge

Kratak opis:Prikaz stranice za online zakazivanje termina kod izabranog veterinara sa mogućnošću otkazivanja termina.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Dodavanje slike

Kratak opis: Dodavanje slike na profil veterinara.

Akteri koji iniciraju slučaj korišćenja: Veterinar

### Odobravanje usluga u zakazanom terminu

Kratak opis: Odobravanje usluge i termina izabranih od strane korisnika.

Akteri koji iniciraju slučaj korišćenja: Veterinar

### Pregled sopstvenog profila

Kratak opis: Prikaz profila korisnika, sa osnovnim informacijama o njemu, spiskom životinja i zahtevima za pregled.

Akteri koji iniciraju slučaj korišćenja: Korisnik.

### Uvid u odgovor veterinara

Kratak opis:Prijavljeni korisnik ima uvid u odgovor na prosledjeni zahtev veterinaru sa mogućnošću potvrdjivanja termina.

Akteri koji iniciraju slučaj korišćenja:Korisnik.

### Odobravanje registrovanja veterinara

Kratak opis: Uprava odobrava registrovanje veterinara na portal.

Akteri koji inicirati slučaj korišćenja: Uprava

### Uređivanje zdravstvenih kartona

Kratak opis:Prikazana strancica pruža mogućnost uredjivanja zdravstvenih kartona.

Akteri koji iniciraju slučaj korišćenja: Veterinar.

### Uvid u ocene i komentare

Kratak opis:Prikazuju se ocene svih veterinara i komenari dosadašnjih korisnika(vlasnici kućnih ljubimaca).

Akteri koji iniciraju slučaj korišćenja: Veterinar, posetilac

### Pregled obaveza

Kratak opis:Prikaz stranice sa svim obavezama veterinara u narednom periodu.

Akteri koji iniciraju slučaj korišćenja: Veterinar

### Pregled profila korisnika

Kratak opis: Pregled profila korisnika, sa svim njegovim životinjama.

Akteri koji iniciraju slučaj korišćenja: Uprava

### Brisanje profila korisnika

Kratak opis: Brisanje korisničkog naloga.

Akteri koji iniciraju slučaj korišćenja: Uprava

### Pregled profila veterinara

Kratak opis: Pregled profila veterinara, sa svim njegovim podacima i utiscima.

Akteri koji iniciraju slučaj korišćenja: Uprava

### Brisanje profila veterinara

Kratak opis:Brisanje naloga veterinara.

Akteri koji iniciraju slučaj korišćenja: Uprava

### Uređivanje usluga

Kratak opis: Uprava ima mogućnost uređivanja usluga: menjanja cene ili osnovnih informacija određene usluge.

Akteri koji iniciraju slučaj korišćenja: Uprava

### Dodavanje nove usluge

Kratak opis:Kreiranje nove usluge veterinarske stanice.

Akteri koji iniciraju slučaj korišćenja: Uprava

### Brisanje usluge

Kratak opis:Brisanje usluge veterinarske stanice.

Akteri koji iniciraju slučaj korišćenja: Uprava

### Uvid u angažovanja

Kratak opis:  Uprava ima mogućnost uvida u zakazane, odrađene i odbijene preglede.

Akteri koji iniciraju slučaj korišćenja: Uprava

# 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.

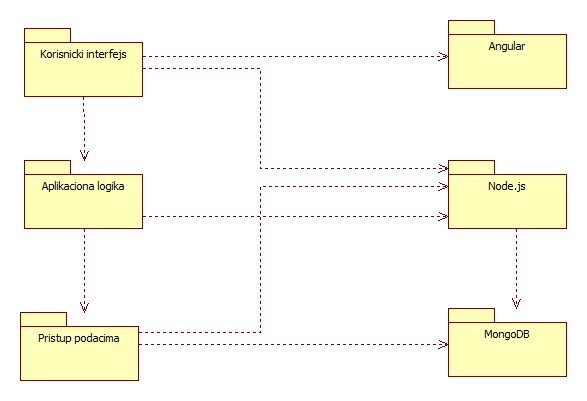
Logički pogled na Ready Vet portal obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice realizovane pomoću Angular skripti.

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

Paket *Pristup podacima* sadrži Node.js skripte koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u MongoDB bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

Ovaj sloj realizuje korisnički interfejs portala. U njemu su sadržane sve Angular skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa Angular i Nest.js.

### Aplikaciona logika

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Ready Vet portala. Sadrži Node.js 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 Nest.js paketa.

### Pristup podacima

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži Node.js skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MongoDB bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa Nest.js i MongoDB baza podataka.

### Angular

Tehnologija Angular-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.

### Node.js

Tehnologija Node.js-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.

### MongoDB

MongoDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Ready Vet portala.

# Pogled na procese

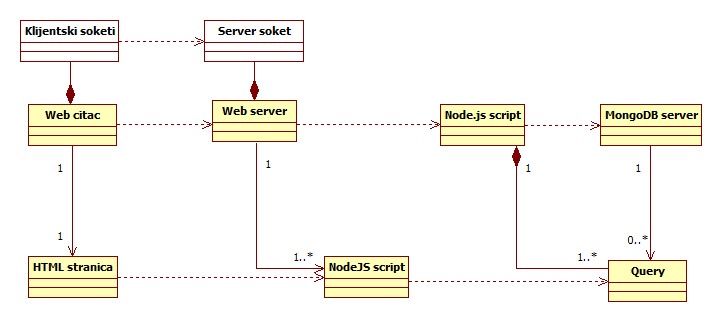
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Web aplikacije zasnovane na Node.js-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera. Sa stanovišta projektanta Node.js 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 Ready Vet portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Ready Vet portala. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na Node.js-u i MongoDB 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 dobija ažuriranja podataka od servera kroz sokete.

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. Kada je reč o našoj Web aplikaciji, na serverskoj strani se izvršavaju Node.js skripte koje obrađuje klijentske zahteve koji stižu iz Web čitača preko Angular skripti. Node.js ima pristup MongoDB bazi za preuzimanje i ažuriranje podataka.

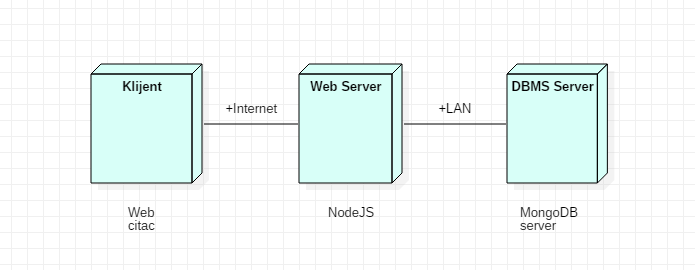
### MongoDB Server

MongoDB Server je proces koji izvršava funkcionalnost MongoDB 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 Ready Vet portala.



## Klijent

Pristup Ready Vet portal 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 Node.js koji vrše obradu zadatih Node.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 MongoDB 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).

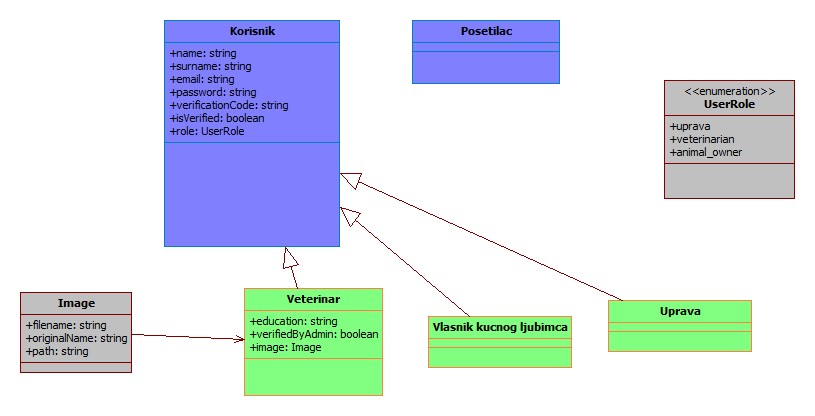
# Pogled na implementaciju sistema

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

## Model domena

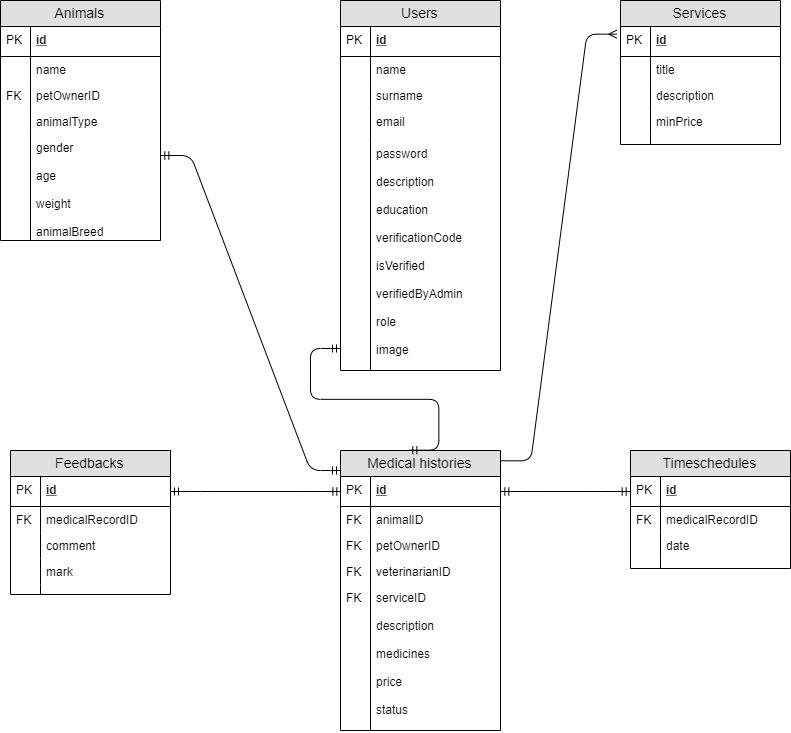
Model domena za koji se Ready Vet portal projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (NodeJS i Angular skripti) koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem NodeJs skripti. Sledeći dijagram je UML dijagram tipa ER kojim je modelovan izgled baze jer je MongoDB nerelaciona baza.



## Komponente sistema

Komponente sistema Ready Vet portala su NodeJS i Angular skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je NodeJS i Angular 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

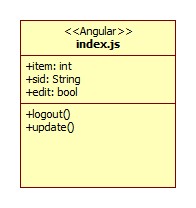
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



Komponenta **index.js** je implementira stranicu portala čiji sadržaj može da varira od parametra koji korisnik izabere.

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

Parametri koji utiču na izbor i jezik za prikaz stranice ilustrovani su sledećim dijagramom klasa:

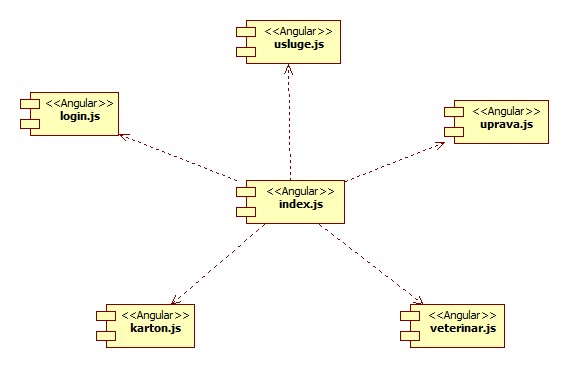


Značenje atributa je sledeće:

* item – identifikator stavke koja se detaljno prikazuje na stranici
* sid – identifikator sesije kada je korisnik ulogovan
* edit – definiše da li se radi o modu za ažuriranje podataka

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.js** 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:

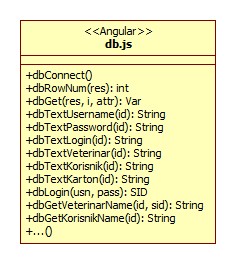


### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru NodeJS skripta **db.js**Pomenuti skript se uključuje na početku main.js-a, tako da su sve funkcije za pristup podacima dostupne svim komponentama.



Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze:



Navedene funkcije obavljaju sledeće zadatke:

* dbConnect – povezivanje na bazu podataka, poziva se u samom skriptu
* dbRowNum – vraća broj slogova sadržanih u rezultatu upita
* dbGet – vraća vrednost polja iz rezultata upita
* dbText... – vraća vrednost pojedinih konstanti koje se definišu uz ID
* dbLogin – loguje korisnika i vraća SID
* dbGetVeterinarName – vraća ime člana po ID-u i SID-u
* dbGetKorisnikName – vraća naziv korisnika po ID-u i SID-u
* ... – ostale funkcije

# 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. Ready Vet portal će biti dostupan 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.