Taita

Veb aplikacija za izdavanje soba i pronalaženje cimera

Detaljni arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 14.11.2022. | Nacrt | Inicijalna verzija | Dženisa |
| 16.11.2022. | 1.0 | Dopunjena i prerađena verzija | Aldina |
|  |  |  |  |
|  |  |  |  |

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 9

6.2.1 Registracija na sistem 9

6.2.2 Pretraživanje po ključnoj reči 9

6.2.3 Filtriranje 9

6.2.4 Sortiranje 9

6.2.5 Pregled podataka o oglasima 9

6.2.6 Prijavljivanje na sistem 9

6.2.7 Odjava sa sistema 9

6.2.8 Promena lozinke 9

6.2.9 Postavljanje novih oglasa 9

6.2.10 Ažuriranje postojećih oglasa 9

6.2.11 Brisanje oglasa 9

6.2.12 Dodavanje podataka o sopstvenom nalogu 9

6.2.13 Ažuriranje unetih podataka o sopstvenom nalogu 9

6.2.14 Prijavljeni korisnik – podnosilac zahteva gleda na koje oglase se prijavio 10

6.2.15 Prijavljeni korisnik – izdavač oglasa gleda ko se prijavio na oglas 10

6.2.16 Prijavljeni korisnik - podnosilac zahteva upućuje željenu poruku izdavaču oglasa 10

6.2.17 Prijavljeni korisnik - izdavač oglasa odgovara na poruku podnosiocu zahteva 10

6.2.18 Pregled korisničkih naloga 10

6.2.19 Izmena podataka 10

6.2.20 Brisanje korisničkog naloga 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 HTML 12

7.1.5 Node.js 12

7.1.6 MongoDB 12

8. Pogled na procese 12

8.1 Procesi 12

8.1.1 Veb čitač 12

8.1.2 Veb server 13

8.1.3 Node.js V8 JavaScript Engine 13

8.1.4 MongoDB server 13

9. Pogled na raspoređivanje sistema 13

9.1 Klijent 13

9.2 Frontend 13

9.3 Veb server 13

9.4 DBMS server 13

10. Pogled na implementaciju sistema 13

10.1 Model domena 14

10.2 Šema baze podataka 15

10.3 Komponente sistema 15

10.3.1 Komponente korisničkog interfejsa 16

10.3.2 Komponente aplikacione logike 16

10.3.3 Komponente za pristup podacima 16

11. Performanse 17

12. Kvalitet 17

Detaljni arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Taita aplikacije.

# Opseg dokumenta

Dokument se odnosi na Taita veb aplikaciju koja će biti razvijena od strane Tutim-a. Taita predstavlja veb aplikaciju za izdavanje soba i pronalaženje cimera. Namena joj je efikasno postavljanje, pretraživanje i prijava na oglase vezane za izdavanje soba i pronalaženje cimera.

# Reference

Spisak korišćene literature:

1. Taita – Dokumentacija projekta, Tutim-Taita, I, verzija 1.0, 2022, Tutim.
2. Taita – Plan realizacije projekta, Tutim-Taita, II, verzija 1.0, 2022, Tutim.
3. Taita – Raspored aktivnosti, Tutim-Taita, II, verzija 1.0, 2022, Tutim.
4. Taita – Vizija sistema, Tutim-Taita, III, verzija 1.0, 2022, Tutim.
5. Taita – Specifikacija zahteva, Tutim-Taita, IV, verzija 1.0, 2022, Tutim.

# 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. Taita aplikacija će biti implementirana kao veb aplikacija zasnovana na Node.js modules i MongoDB bazi podataka.
2. Klijentski deo Taita aplikacije će biti optimizovan za sledeće veb čitače: Google Chrome, Firefox, Microsoft Edge, Opera.
3. Svi zahtevi u pogledu performansi dati u odeljku 5 pod Reference moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.
4. Klasifikaciju oglasa treba standardizovati sa postojećom tipologijom propisanom Zakonom o turizmu, Pravilnikom o minimalnim tehničkim i sanitarno - higijenskim uslovima za pružanje usluga u domaćoj radinosti i u seoskom turističkom domaćinstvu, kao i drugim pravilnicima i odlukama.

# Pogled na slučajeve korišćenja

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

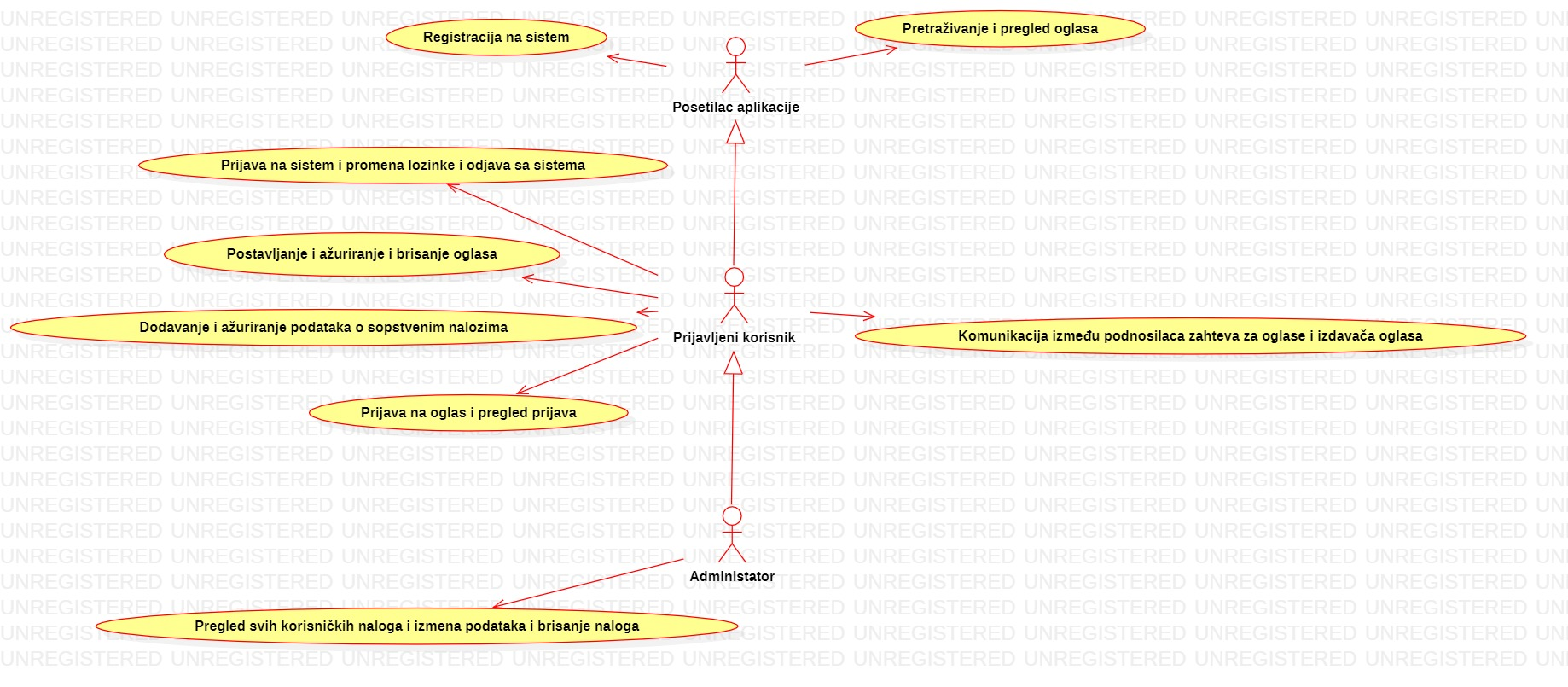
Slučajevi korišćenja Taita aplikacije su:

* *Registracija na sistem*
* *Pretraživanje i pregled oglasa*
  + Pretraživanje oglasa
    - Pretraživanje po ključnoj reči
    - Filtriranje
    - Sortiranje
  + Pregled podataka o oglasima
* *Prijava na sistem i promena lozinke i odjava sa sistema*
  + Prijavljivanje na sistem
  + Odjava sa sistema
  + Promena lozinke
* *Postavljanje i ažuriranje i brisanje oglasa*
  + Postavljanje novih oglasa
  + Ažuriranje postojećih oglasa
  + Brisanje postojećih oglasa
* *Dodavanje i ažuriranje podataka o sopstvenim nalozima*
  + Unos podataka o sopstvenom nalogu
  + Ažuriranje unetih podataka o sopstvenom nalogu
* *Prijava na oglas i pregled prijava*
  + Prijavljeni korisnik - podnosilac zahteva gleda na koje oglase se prijavio
  + Prijavljeni korisnik - izdavač oglasa gleda ko se prijavio na oglas
* *Komunikacija između podnosilaca zahteva za oglase i izdavača oglasa*
  + Prijavljeni korisnik - podnosilac zahteva upućuje željenu poruku izdavaču oglasa
  + Prijavljeni korisnik - izdavač oglasa odgovara na poruku podnosiocu zahteva
* *Pregled svih korisničkih naloga i izmena podataka i brisanje naloga*
  + Pregled korisničkih naloga
  + Izmena podataka
  + Brisanje korisničkog naloga

Ove slučajevi korišćenja mogu da iniciraju posetilac aplikacije, prijavljeni korisnik i administrator.

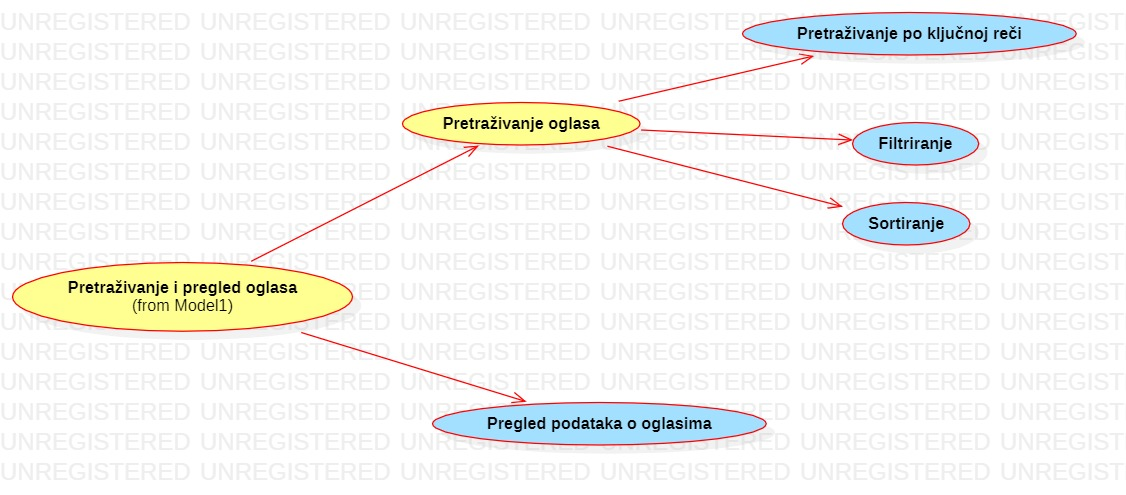
## Dijagrami slučajeva korišćenja

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

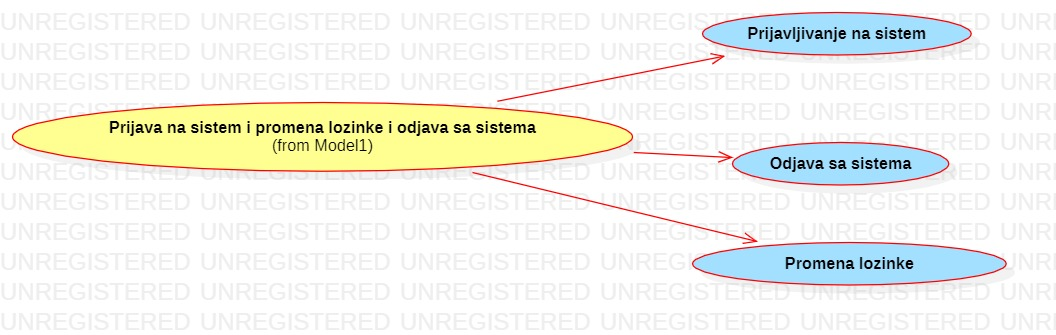


Slučajevi korišćenja *pretraživanje i pregled oglasa, prijava na sistem i promena lozinke i odjava sa sistema, postavljanje i ažuriranje i brisanje oglasa, dodavanje i ažuriranje podataka o sopstvenim nalozima, prijava na oglas i pregled prijava, komunikacija između podnosilaca zahteva za oglase i izdavača oglasa, pregled svih korisničkih naloga i izmena podataka i brisanje naloga* se mogu razložiti dalje na posebne slučajeve korišćenja.

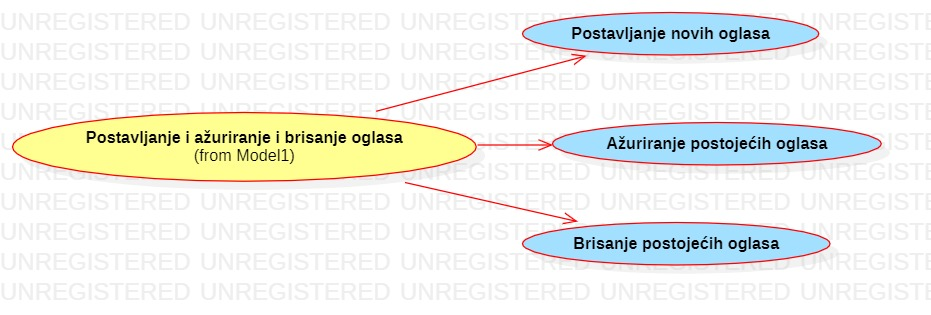
Detaljni UML dijagram za slučaj korišćenja *pretraživanje i pregled oglasa* je prikazan na sledećoj slici:



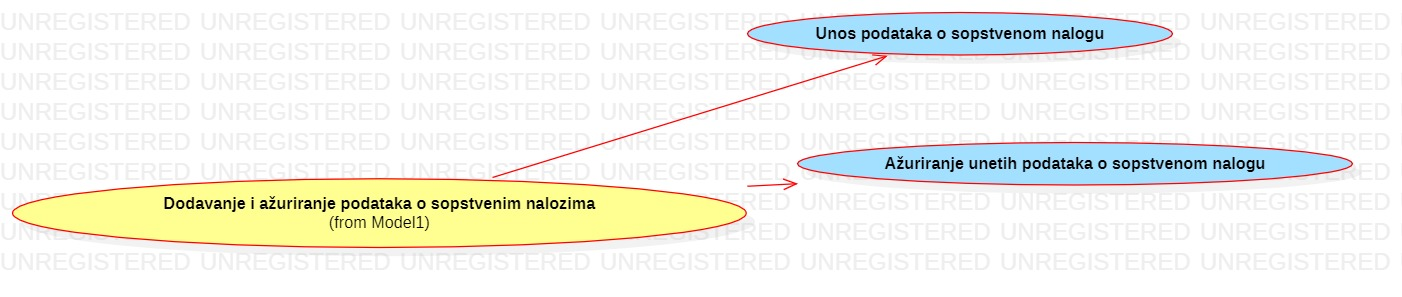
Detaljni UML dijagram za slučaj korišćenja *prijava na sistem i promena lozinke i odjava sa sistema* je prikazan na sledećoj slici:



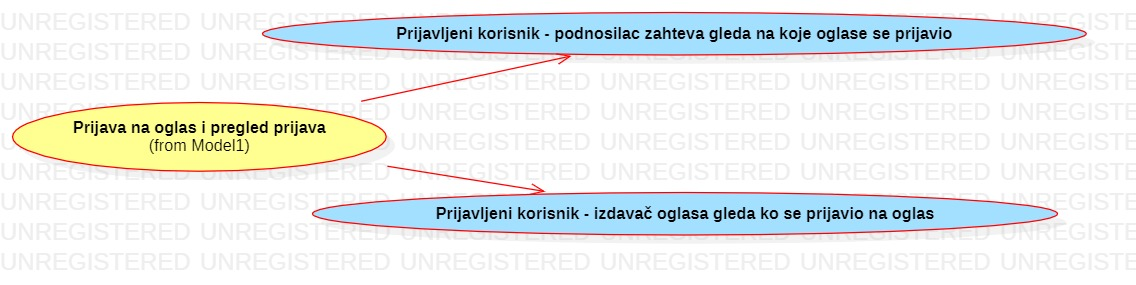
Detaljni UML dijagram za slučaj korišćenja *postavljanje i ažuriranje i brisanje oglasa* je prikazan na sledećoj slici:



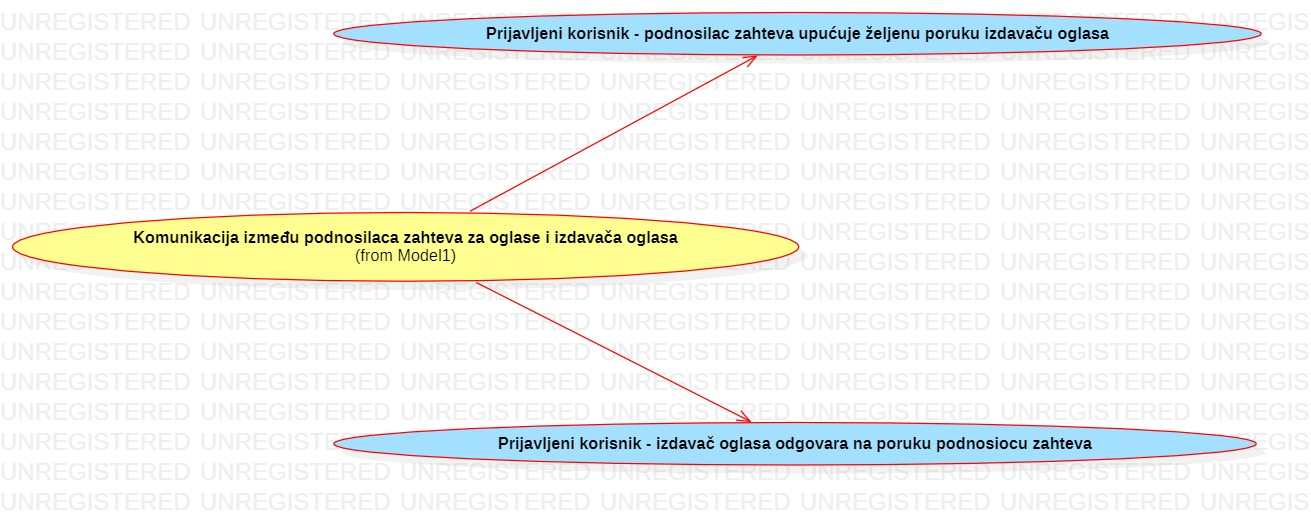
Detaljni UML dijagram za slučaj korišćenja *dodavanje i ažuriranje podataka o sopstvenim nalozima* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *prijava na oglas i pregled prijava* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *komunikacija između podnosilaca zahteva za oglase i izdavača oglasa* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *pregled svih korisničkih naloga i izmena podataka i brisanje naloga* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Registracija na sistem

Kratak opis: Posetioci aplikacije Taita će moći da naprave nalog uz registraciju sa imejl adresom i lozinkom.

Akteri koji iniciraju slučaj korišćenja Posetilac aplikacije.

### Pretraživanje po ključnoj reči

Kratak opis: Posetilac i prijavljeni korisnik pretražuju željene oglase po ključnoj reči.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije i prijavljeni korisnik.

### Filtriranje

Kratak opis: Posetilac i prijavljeni korisnik traže oglase prema izabranim filterima.

Akteri koji iniciraju slučaj korišćenja: Posetilac i prijavljeni korisnik.

### Sortiranje

Kratak opis: Posetilac i prijavljeni korisnik traže oglase prema odgovarajućim kriterijumima sortiranja.

Akteri koji iniciraju slučaj korišćenja: Posetilac i prijavljeni korisnik.

### Pregled podataka o oglasima

Kratak opis: Posetilac i prijavljeni korisnik pregledaju oglase.

Akteri koji iniciraju slučaj korišćenja: Posetilac i prijavljeni korisnik.

### Prijavljivanje na sistem

Kratak opis: Posetilac aplikacije i administrator se prijavljuju na sistem.

Akteri koji iniciraju slučaj korišćenja: Posetilac aplikacije i administrator.

### Odjava sa sistema

Kratak opis: Prijavljeni korisnik i administrator se odjavljuju sa sistema.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik i administrator.

### Promena lozinke

Kratak opis: Prijavljeni korisnik i administrator menjaju lozinku na svom nalogu.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik i administrator.

### Postavljanje novih oglasa

Kratak opis: Prijavljeni korisnici postavljaju novi oglas.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Ažuriranje postojećih oglasa

Kratak opis: Prijavljeni korisnik i administrator ažuriraju podatke o oglasima.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik i administrator.

### Brisanje oglasa

Kratak opis: Prijavljeni korisnik i administrator brišu oglase.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik i administrator.

### Dodavanje podataka o sopstvenom nalogu

Kratak opis: Prijavljeni korisnik dodaje podatke o sebi na svom nalogu.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Ažuriranje unetih podataka o sopstvenom nalogu

Kratak opis: Prijavljeni korisnik menja već unete podatke o sebi na svom nalogu.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Prijavljeni korisnik – podnosilac zahteva gleda na koje oglase se prijavio

Kratak opis: Prijavljeni korisnik – podnosilac zahteva gleda na koje oglase se prijavio.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Prijavljeni korisnik – izdavač oglasa gleda ko se prijavio na oglas

Kratak opis: Prijavljeni korisnik – izdavač oglasa gleda ko se prijavio na oglas.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Prijavljeni korisnik - podnosilac zahteva upućuje željenu poruku izdavaču oglasa

Kratak opis: Prijavljeni korisnik – podnosilac zahteva upućuje željenu poruku izdavaču oglasa.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Prijavljeni korisnik - izdavač oglasa odgovara na poruku podnosiocu zahteva

Kratak opis: Prijavljeni korisnik - izdavač oglasa odgovara na poruku podnosiocu zahteva.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik.

### Pregled korisničkih naloga

Kratak opis: Administrator pregleda korisničke naloge.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Izmena podataka

Kratak opis: Administrator menja podatke o korisničkim nalozima.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Brisanje korisničkog naloga

Kratak opis: Administrator briše korisničke naloge.

Akteri koji iniciraju slučaj korišćenja: 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.

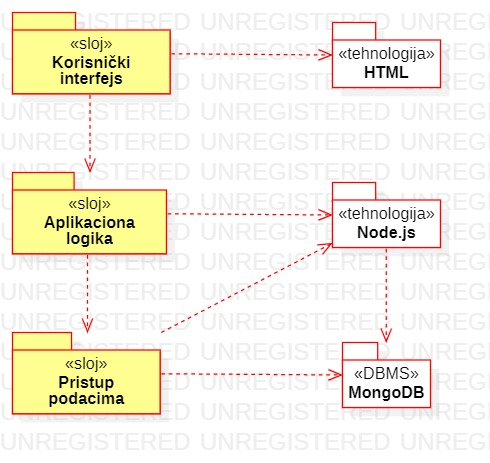
Logički pogled na Taita aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži veb stranice, HTML stranice 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 Node.js modules zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži Node.js modules koji 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 stranice i multimedijalni sadržaji koji generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i od HTML-a.

### Aplikaciona logika

<<sloj>>

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Taita veb aplikacije. Sadrži Node.js modules koji realizuju funkcionalnost karakterističnu za domen primene aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i Node.js paketa.

### Pristup podacima

<<sloj>>

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži Node.js modules 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 Node.js i MongoDB baze podataka.

### HTML

<<tehnologija>>

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

### Node.js

<<tehnologija>>

Tehnologija Node.js obezbeđuje mehanizam za upravljanje zahtevima koji stižu od strane klijenta. Kontroleri su ti koji su zaduženi za generisanje odgovora koji se prosleđuje klijentu.

### MongoDB

<<DBMS>>

MongoDB predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju Taita veb aplikacije.

# Pogled na procese

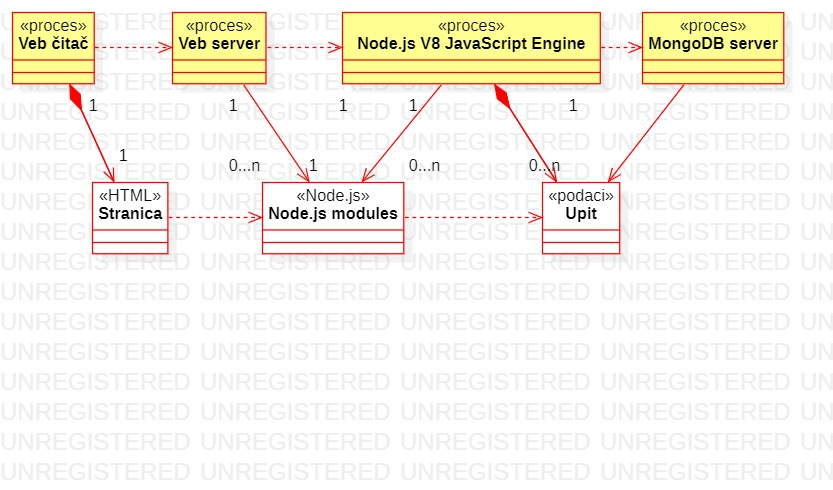
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Veb aplikacije zasnovane na Node.js imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom veb servera. Sa stanovišta projektanta Node.js veb aplikacije nije potrebno voditi računa o načinu rada veb servera i načinu izvršavanja modula.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Taita veb apliakcije kao veb aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Taita veb aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju veb aplikaciju zasnovanu na Node.js i MongoDB bazi podataka.



### Veb čitač

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

Veb čitač zavisi od veb servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

### Veb server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više veb čitača. Ukoliko je zahtevana stranica Node.js, veb server inicira izvršenje Node.js V8 JavaScript Engine procesa koji obrađuje odgovarajući modul i generiše sadržaj koji se vraća čitaču. Veb server može paralelno da inicira veći broj Node.js V8 JavaScript Engine procesa.

### Node.js V8 JavaScript Engine

Node.js V8 JavaScript Engine proces obavlja posao obrade zadatog Node.js modula. Za izvršenje Node.js modula ovaj proces može da zahteva usluge MongoDB servera.

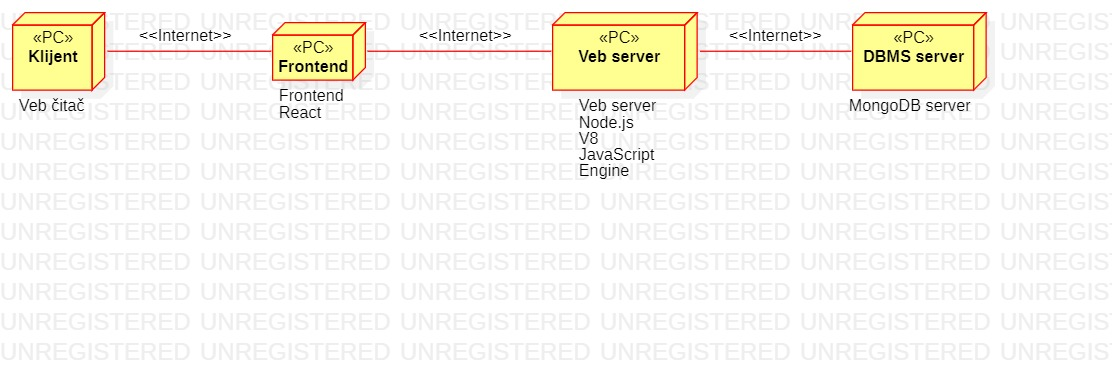
### MongoDB server

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 Taita veb aplikacije.



## Klijent

Pristup Taita veb aplikaciji se obavlja preko klijentskih računara na kojima se izvršava veb čitač. Za povezivanje između klijenta i veb servera koristi se internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Frontend

Za frontend se koristi React tehnologija kojom se gradi korisnički interfejs baziran na UI komponentama.

## Veb server

Računar na kome se izvršava veb server opslužuje više klijenata koji pristupaju preko interneta. Pored osnovnog procesa koji realizuje funkcionalnost veb servera, na ovom računaru mogu da se izvršavaju i procesi Node.js-a koji vrše obradu zadatog Node.js modula.

## DBMS server

DBMS server je računar na kome se izvršava MongoDB server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka.

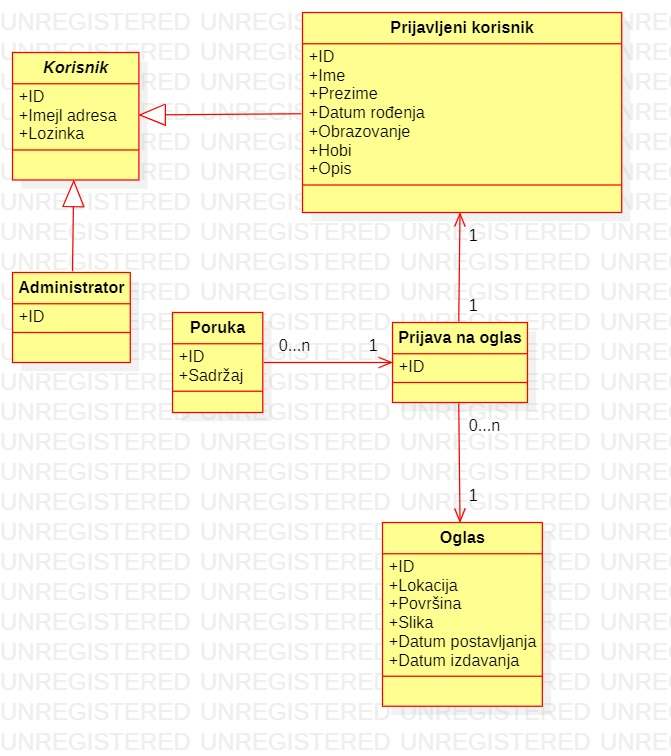
# Pogled na implementaciju sistema

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

## Model domena

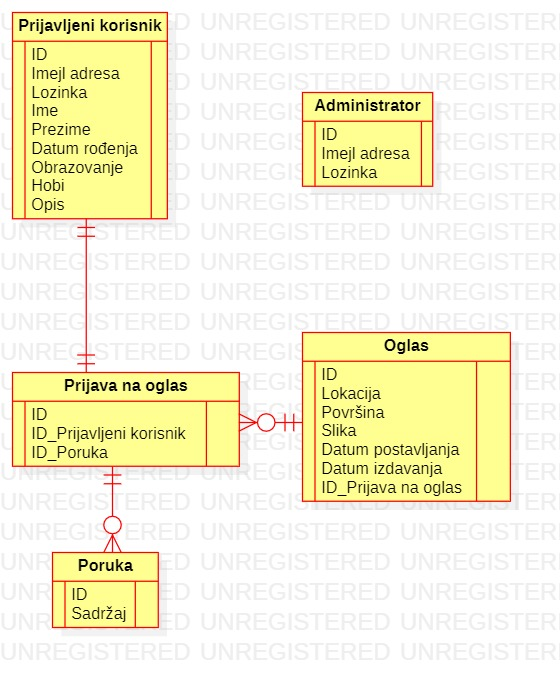
Model domena za koji se Taita veb aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti Node.js koje će biti implementirane.



## Šema baze podataka

Detaljna šema MongoDB baze podataka je prikazana na sledećem dijagramu. Dijagram je kreiran korišćenjemStarUML aplikacije .



## Komponente sistema

Komponente sistema Taita veb aplikacije su React komponente čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je Node.js prikazan kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru modula.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

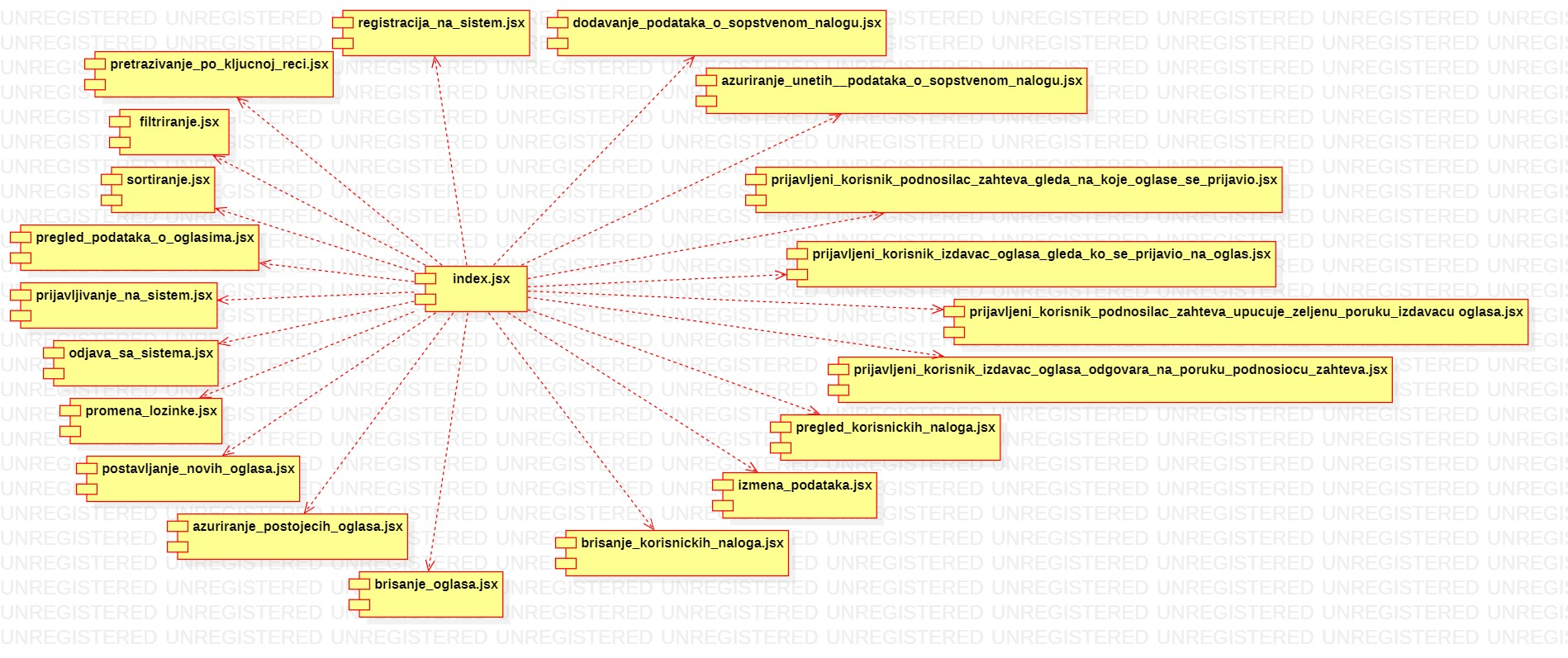


Komponenta **index.jsx** implementira stranicu aplikacije čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

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

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.jsx** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom modulu. 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 Node.js modules **backend.js**. Pomenuti modul se uključuje na početku index.jsx, 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 u odeljku 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 u odeljku 5.

# Kvalitet

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

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