WEBUS

Softver za rezervisanje autobuskih karti

Detaljni arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 29.11.2022. | 1.0 | Inicijalna verzija | Aladin Lakota |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc122106484)

[2. Opseg dokumenta 5](#_Toc122106485)

[3. Reference 5](#_Toc122106486)

[4. Predstavljanje arhitekture 5](#_Toc122106487)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc122106488)

[6. Pogled na slučajeve korišćenja 5](#_Toc122106489)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc122106490)

[6.2 Kratak opis slučajeva korišćenja 9](#_Toc122106491)

[6.2.1 Registracija 9](#_Toc122106492)

[6.2.2 Prijavljivanje 9](#_Toc122106493)

[6.2.3 Verifikacija mejla 9](#_Toc122106494)

[6.2.4 Odjavljivanje 10](#_Toc122106495)

[6.2.5 Brisanje korisnika 10](#_Toc122106496)

[6.2.6 Kreiranje korisnika kao moderatora 10](#_Toc122106497)

[6.2.7 Blokiranje korisnika 10](#_Toc122106498)

[6.2.8 Deblokiranje korisnika 10](#_Toc122106499)

[6.2.9 Pregled podataka o korisniku 10](#_Toc122106500)

[6.2.10 Postavljanje rute 10](#_Toc122106501)

[6.2.11 Izmena rute 10](#_Toc122106502)

[6.2.12 Brisanje rute 10](#_Toc122106503)

[6.2.13 Pretraga po gradu 10](#_Toc122106504)

[6.2.14 Pretraga po datumu 10](#_Toc122106505)

[6.2.15 Rezervacija sedišta 10](#_Toc122106506)

[6.2.16 Plaćanje karte 10](#_Toc122106507)

[6.2.17 Kopiranje karte 11](#_Toc122106508)

[6.2.18 Pregled informacija o rutama 11](#_Toc122106509)

[6.2.19 Popunjava formu za kontakt 11](#_Toc122106510)

[6.2.20 Skenira QR kod 11](#_Toc122106511)

[7. Pogled na logičku arhitekturu sistema 11](#_Toc122106512)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 12](#_Toc122106513)

[7.1.1 Korisnički interfejs 12](#_Toc122106514)

[7.1.2 Aplikaciona logika 12](#_Toc122106515)

[7.1.3 Pristup podacima 12](#_Toc122106516)

[7.1.4 HTML 12](#_Toc122106517)

[7.1.5 NodeJS 12](#_Toc122106518)

[7.1.6 MongoDB 12](#_Toc122106519)

[8. Pogled na procese 13](#_Toc122106520)

[8.1 Procesi 13](#_Toc122106521)

[8.1.1 Veb čitač 13](#_Toc122106522)

[8.1.2 Veb server 13](#_Toc122106523)

[8.1.3 MongoDB Server 13](#_Toc122106524)

[9. Pogled na raspoređivanje sistema 13](#_Toc122106525)

[9.1 Klijent 14](#_Toc122106526)

[9.2 Veb server 14](#_Toc122106527)

[9.3 DBMS server 14](#_Toc122106528)

[10. Pogled na implementaciju sistema 14](#_Toc122106529)

[10.1 Model domena 14](#_Toc122106530)

[10.2 Komponente sistema 14](#_Toc122106531)

[10.2.1 Komponente korisničkog interfejsa 15](#_Toc122106532)

[10.2.2 Komponente aplikacione logike 15](#_Toc122106533)

[10.2.3 Komponente za pristup podacima 16](#_Toc122106534)

[11. Performanse 17](#_Toc122106535)

[12. Kvalitet 17](#_Toc122106536)

Detaljni arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture WEBUS veb aplikacije.

# Opseg dokumenta

Dokument se odnosi na WEBUS veb aplikaciju koja će biti razvijena od strane FOREX team-a. WEBUS predstavlja skraćenicu za Portal traženje poslova. Namena sistema jeste da olakša nezaposlenima da brže dođu do polsa.

# Reference

Spisak korišćene literature:

1. WEBUS – Predlog projekta, FOREX-WEBUS-01, V1.0, 2022, FOREX Team.
2. WEBUS – Planirani raspored aktivnosti na projektu, V1.0, 2022, FOREX Team.
3. WEBUS – Plan realizacije projekta, V1.0, 2022, FOREX Team.

# 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. WEBUS veb aplikacija će biti implementirana kao Veb aplikacija zasnovana na NodeJS-u i Mongo bazi podataka [4].
2. Klijentski deo WEBUS veb aplikacije će biti optimizovan za sledeće Veb čitače: Internet Explorer 7.0 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].

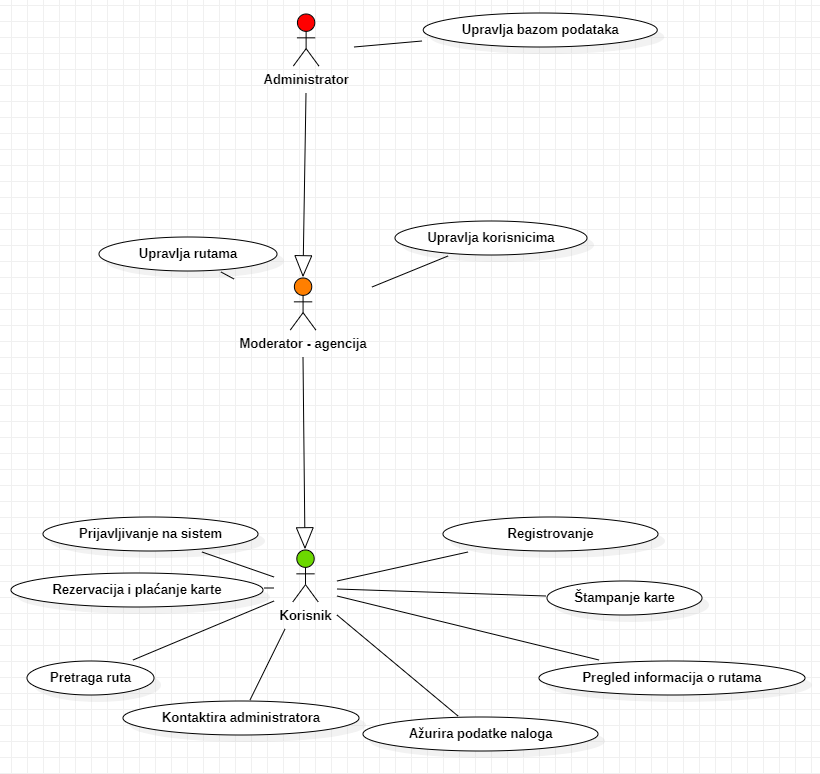
Slučajevi korišćenja WEBUS portala su:

* **Prijavljivanje na sistem**
  + Prijavljivanje
  + Odjavljivanje
  + Restartovanje lozinke
* **Upravlja korisnicima**
  + Tabela korisnika
    - Pregled podataka o korisniku
  + Dodeljivanje korisniku admina
  + Brisanje korisnika
  + Blokiranje korisnika
  + Deblokiranje korisnika
* **Upravlja rutama**
  + Dodavanje rute
  + Ažuriranje podataka
  + Brisanje rute
* **Pretraga ruta**
  + Po gradu
  + Po datumu
  + Po ceni
* **Rezervacija i plaćanje karte**
  + Rezervacija sedišta
  + Plaćanje karte
* **Kontaktira administratora**
  + Popunjava formu
  + Skenira QR kod
* **Registrovanje**
  + ***Verifikacija mejla***
* **Štampanje rezervisane karte**
* **Ažurira podatke naloga**
* **Upravlja bazom podataka**

Ove slučajevi korišćenja mogu da iniciraju registrovan Korisnik, registrovana Moderator(agencija) ili Administrator.

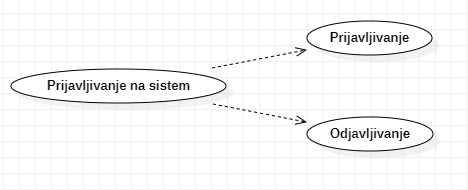
## Dijagrami slučajeva korišćenja

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

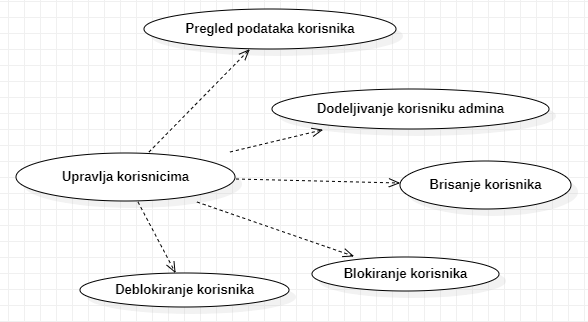


Slučajevi korišćenja ***prijavljivanje na sistem****,* ***upravlja korisnicima****,* ***upravlja rutama****,* ***pretraga ruta****,* ***rezervacija i plaćanje karte****,* ***kontaktira administratora***obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

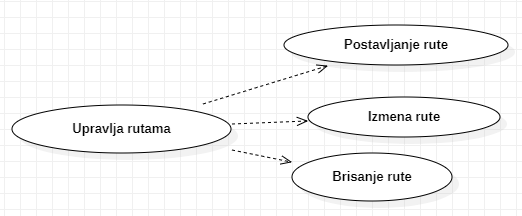
Detaljni UML dijagram za slučaj korišćenja ***prijavljivanje na sistem***je prikazan na sledećoj slici:



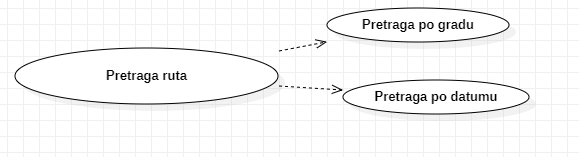
Detaljni UML dijagram za slučaj korišćenja ***upravlja korisnicima***je prikazan na sledećoj slici:



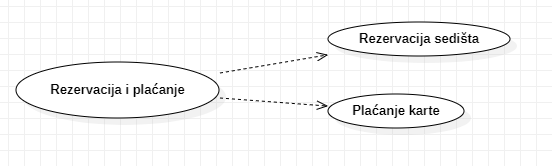
Detaljni UML dijagram za slučaj korišćenja ***upravlja rutama*** je prikazan na sledećoj slici:

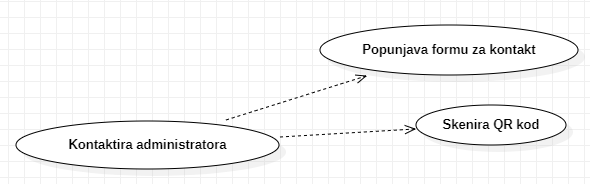


Detaljni UML dijagram za slučaj korišćenja ***pretraga ruta*** je prikazan na sledećoj slici:



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



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

## Kratak opis slučajeva korišćenja

### Registracija

Kratak opis: Nakon što se korisnik registruje uspešno na veb aplikaciju njemu je omogućeno da se uloguje.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija) i Administrator.

### Prijavljivanje na sistem

Kratak opis: Mogućnost pregleda korisnika koji su registrovani na veb aplikaciju.

Akteri koji iniciraju slučaj korišćenja: Administrator, Moderator(agencija) i Korisnik.

### Verifikacija mejla

Kratak opis: Prikaz verfikacije mejla u toku registracije korisnika, korisniku se proverava validnost mejla.

Akteri koji iniciraju slučaj korišćenja: Neregistrovani korisnik.

### Odjavljivanje

Kratak opis: Korisnik veb aplikacije se odjavljuje sa naloga.

Akteri koji iniciraju slučaj korišćenja: Administrator, Moderator(agencija) i korisnik.

### Brisanje korisnika

Kratak opis: Administrator uklanja korisnika iz baze.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Kreiranje korisnika kao moderatora

Kratak opis: Izmena uloge ili funkcije korisnika na veb aplikaciji.

Akteri koji iniciraju slučaj korišćenja: Administrator i moderator.

### Blokiranje korisnika

Kratak opis: Administrator blokira korisnika iz baze.

Akteri koji iniciraju slučaj korišćenja: Administrator i moderator.

### Deblokiranje korisnika

Kratak opis: Administrator deblokira korisnika iz baze.

Akteri koji iniciraju slučaj korišćenja: Administrator i moderator.

### Pregled podataka o korisniku

Kratak opis: Administrator ima tabelarni prikaz svih registrovanih korisnika.

Akteri koji iniciraju slučaj korišćenja: Administrator i moderator.

### Dodavanje rute

Kratak opis: Moderator(agencija) postavlja podatke o autobuskoj ruti.

Akteri koji iniciraju slučaj korišćenja: Moderator(agencija), Administrator.

### Ažuriranje rute

Kratak opis: Moderator(agencija) pravi izmene nad određenom rutom.

Akteri koji iniciraju slučaj korišćenja: Moderator(agencija), Administrator.

### Brisanje rute

Kratak opis: Moderator(agencija) briše rutu.

Akteri koji iniciraju slučaj korišćenja: Moderator(agencija), Administrator.

### Pretraga po gradu

Kratak opis: Pretraga oglasa po traženom gradu.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija), Administrator.

### Pretraga po datumu

Kratak opis: Pretraga oglasa po traženom datumu.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija), Administrator.

### Pretraga po ceni

Kratak opis: Pretraga oglasa po traženom datumu.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija), Administrator.

### Rezervacija sedišta

Kratak opis: Korisnici aplikacije rezervišu autobusku kartu.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija), Administrator.

### Plaćanje karte

Kratak opis: Korisnici aplikacije popunjavaju formu za plaćanje karte bankovni indentifikacionim brojevima.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija), Administrator.

### Štampanje karte

Kratak opis: Korisnici veb aplikacije imaju mogućnost štampanja rezervisane karte.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija), Administrator.

### Pregled informacija o rutama

Kratak opis: Registrovani i ulogovani korisnici aplikacije imaju prikaz svih dostupnih autobuskih karti odnosno ruta.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija), Administrator.

### Popunjava formu za kontakt

Kratak opis: Korsinik veb aplikacije ima mogućnost kontaktiranja popunjavanjem jedinstvene forme za kontakt administratora putem mejla.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija).

### Skenira QR kod

Kratak opis: Korisnik veb aplikacije pored mogućnosti popunjavanja forme za kontakt, ima mogućnost skeniranja QR koda kojim dobija kontakt telefon od administratora aplikacije.

Akteri koji iniciraju slučaj korišćenja: Korisnik, Moderator(agencija).

# 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.

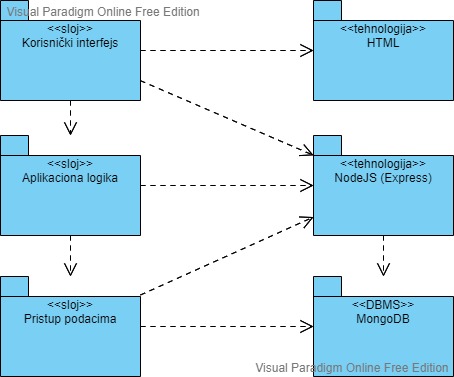
Logički pogled na WEBUS veb aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži veb 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 JavaScript fajlove zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži fajlove 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 veb aplikacije. U njemu su sadržane sve HTML, multimedijalni sadržaji i JS fajlovi koji generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom. Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i JS.

### Aplikaciona logika

sloj

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

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži kontrolere 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 NodeJS-a 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.

### NodeJS

tehnologija

Tehnologija NodeJS obezbeđuje mehanizam za upravljanje zahtevima koji stižu od strane klijenta. NodeJS je takođe zadužen i za generisanje odgovora koji se prosleđuje klijentu.

### MongoDB

DBMS

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

# Pogled na procese

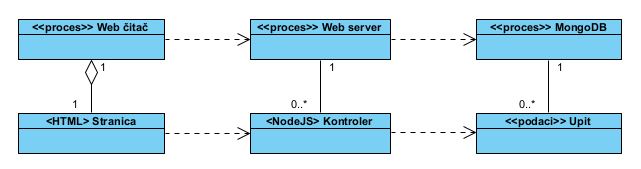
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe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

Veb aplikacije zasnovane na NodeJS-u imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Veb servera.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje WEBUS kao Veb aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju WEBUS veb aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Veb aplikaciju zasnovanu na NodeJS-u i Mongo bazi podataka.



### Veb čitač

Veb čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Veb servera.

Veb čitač zavisi od Veb servera koji generiše i vraća odgovarajuće informacije (JSON objekat) na zahtev.

### Veb server

Veb server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više veb čitača.On odlučuje koji kontroler će biti zadužen za generisanje odgovora.

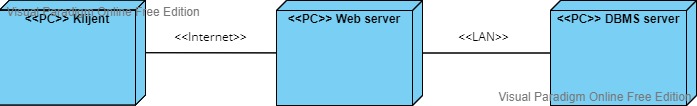
### MongoDB Server

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 WEBUS veb aplikacije.



## Klijent

Pristup WEBUS 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.

## Veb server

Računar na kome se izvršava Veb server opslužuje više klijenata koji pristupaju preko Interneta. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Veb 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).

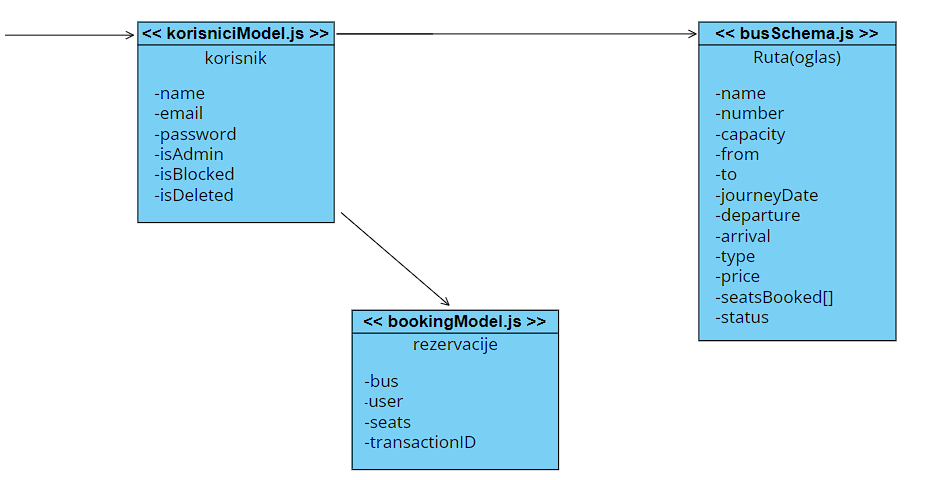
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju WEBUS 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 WEBUS veb aplikacija projektuje je ilustrovan UML dijagramom klasa koji je radjen preko online visual paradigm-a. 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 kontrolera) koje će biti implementirane.

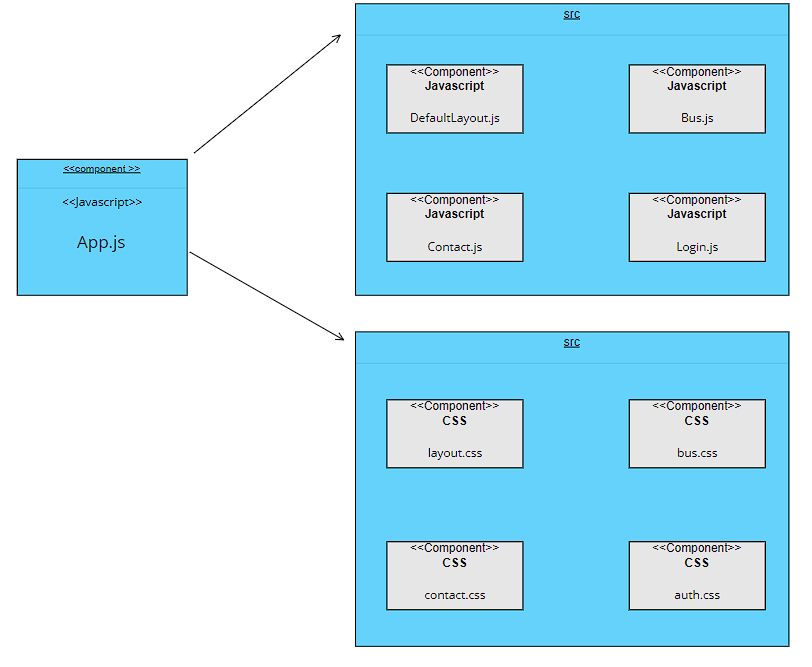


## Komponente sistema

Komponente sistema WEBUS veb aplikacije su React komponente i kontroleri čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. Koja od komponenti će se prikazati zavisi isključivo od putanje (URL-a).

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



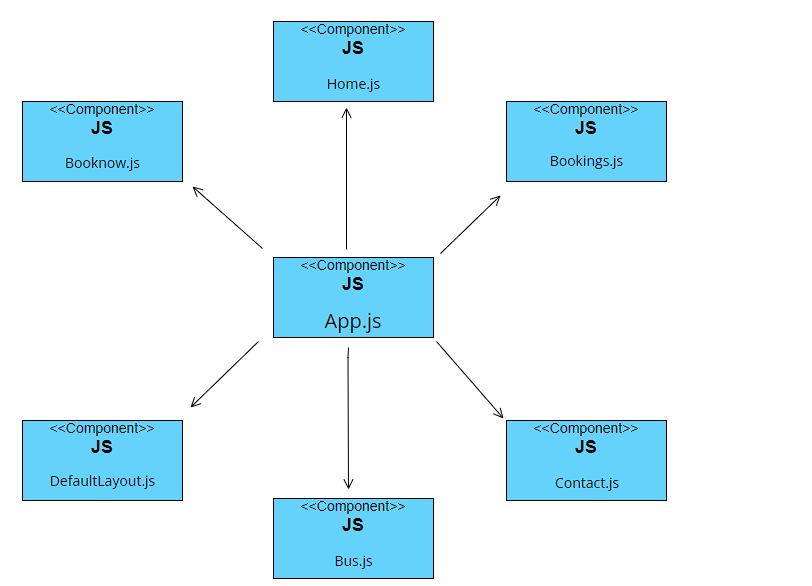
Komponenta **App.js** implementira stranicu portala čiji sadržaj može da varira zavisno od URL-a zahteva.

Komponenta **DefaultLayaut.js** je jedna od kopmonenti koja može biti sadržaj stranica App.js.

Komponenta **layout.css** predstavlja jedan od opisa stilova za pojedine HTML elemente koji se javljaju na stranici DefaultLayout.js.

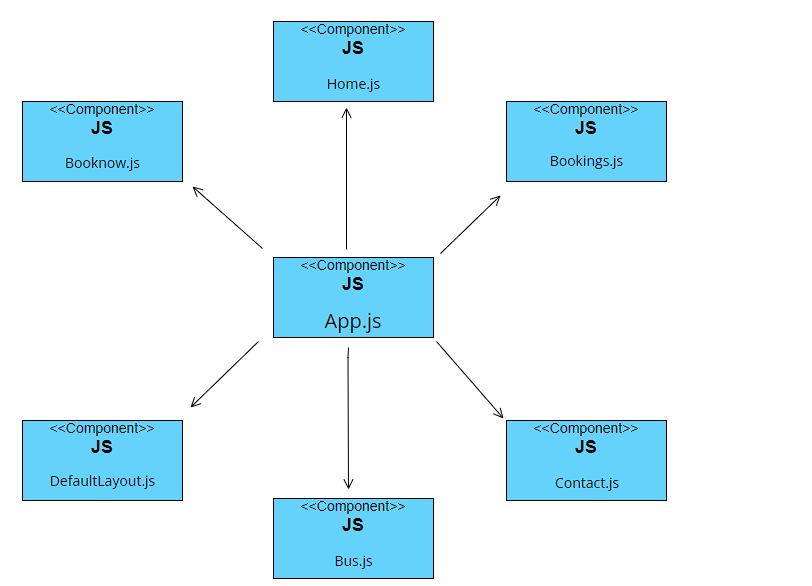
### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **App.js** komponente korisničkog interfejsa.Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti.



### Komponente za pristup podacima

Bazi se pristupa unutar kontrolera a pošto se koristi Mongoose onda nema potrebe za pravljenjem posebnih fajlova iz kojih bi se pristupalo bazi jer Mongoose sam po sebi može pristupiti iz 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 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. WEBUS veb aplikacija će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada veb aplikacija nije dostupan ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.