ReceByte

Personalizovani recepti za vaš frižider

Arhitekturni projekat

Verzija 2.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 13.04.2021. | 1.0 | Inicijalna verzija | SANTim |
| 21.06.2021 | 2.0 | Prepravka funkcionalnosti | SANTim |
|  |  |  |  |
|  |  |  |  |

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 Pregled osnovne stranice web-aplikacije 8

6.2.2 Pregled recepta 8

6.2.3 Dodavanje komentara 8

6.2.4 Kreiranje naloga 8

6.2.5 Ocenjivanje recepata 8

6.2.6 Verifikacija recepata 8

6.2.7 Dodavanje privilegovanih korisnika 8

6.2.8 Pretraga recepata 8

6.2.9 Pretraga recepata na osnovu namirnica iz frižidera 8

6.2.10 Dodavanje namirnica u virtuelni frižider 8

6.2.11 Čuvanje recepata 8

6.2.12 Dodavanje recepta 9

6.2.13 Lajkovanje komentara 9

7. Pogled na logičku arhitekturu sistema 9

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 9

7.1.1 Korisnički interfejs 9

7.1.2 Aplikaciona logika 9

7.1.3 Pristup podacima 10

7.1.4 HTML 10

7.1.5 .NET Core 10

7.1.6 MySQL 10

8. Pogled na procese 10

8.1 Procesi 10

8.1.1 Web čitač 10

8.1.2 Web server 11

8.1.3 MySQL Server 11

9. Pogled na raspoređivanje sistema 11

9.1 Klijent 11

9.2 Web server 11

9.3 DBMS server 11

10. Pogled na implementaciju sistema 11

10.1 Model domena 11

10.2 Šema baze podataka 12

10.3 Komponente sistema 13

10.3.1 Komponente korisničkog interfejsa 13

10.3.2 Komponente aplikacione logike 15

10.3.3 Komponente za pristup podacima 15

11. Performanse 17

12. Kvalitet 17

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture ReceByte web aplikacije.

# Opseg dokumenta

Dokument se odnosi na ReceByte web aplikaciju koji će biti razvijen od strane SANTim-a. Namena sistema je efikasno kreiranje i pretraga recepata,saveta kao i presonalizovani frižider prijavljenih korisnika.

# Reference

Spisak korišćene literature:

1. ReceByte – Predlog projekta, SWE-Recebyte-01, V2.0, 2021, SANTim.
2. ReceByte – Vizija sistema, V2.0, 2021, SANTim.
3. ReceByte – Plan realizacije projekta, V1.0, 2021, SANTim.
4. ReceByte – Planirani raspored aktivnosti na projektu, V1.0, 2021, SANTim.
5. ReceByte – Specifikacija zahteva, V1.0, 2021, SANTim.

# 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. ReceByte aplikacija će biti implementiran kao Web aplikacija zasnovana na JavaScript-u i MySQL bazi podataka [4].
2. Klijentski deo ReceByte aplikacije 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].

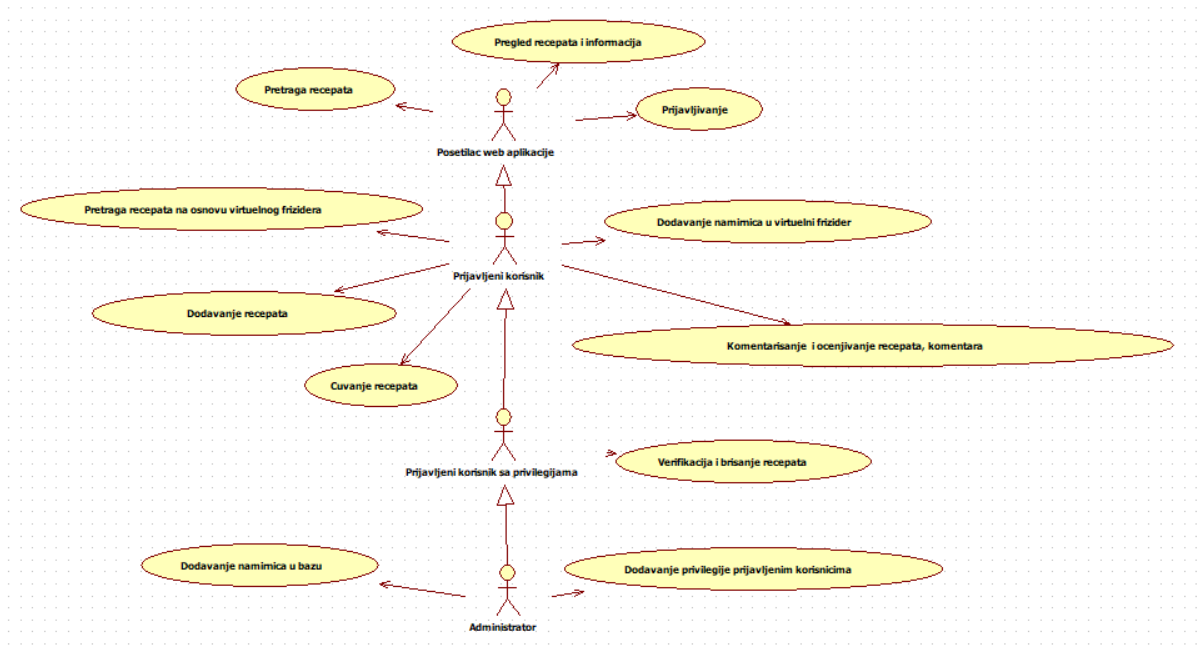
Slučajevi korišćenja ReceByte aplikacije su:

* Pretraga recepata
* *Pregled recepata i informacija*
  + Pregled stranice recepta
  + *Pregled informacije o receptu*
    - Pregled ocene
    - Pregled autora
    - Pregled komentara
    - Pregled sadržaja
* Prijavljivanje
* Dodavanje namirnica u virtuelni frižider
* Pretraga recepata na osnovu virtuelnog frižidera
* *Komentarisanje i ocenjivanje recepata, komentara*
  + *Komentarisanje i ocenjivanje recepata*
    - * Komentarisanje recepata
      * Ocenjivanje recepata
  + Ocenjivanje komentara
* Dodavanje recepata
* Čuvanje recepata
* *Verifikacija* 
  + Verifikacija recepata
* Dodavanje privilegije prijavljenim korisnicima
* Dodavanje namirnica u bazu

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

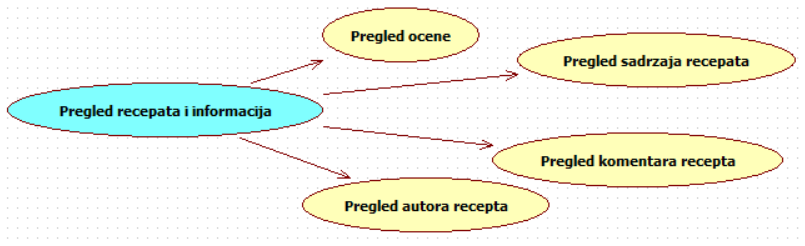
## Dijagrami slučajeva korišćenja

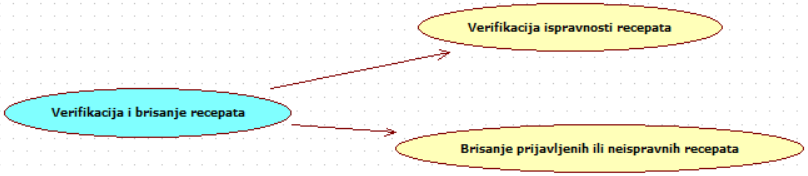
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja ReceByte aplikacije prikazan je na sledećoj slici:

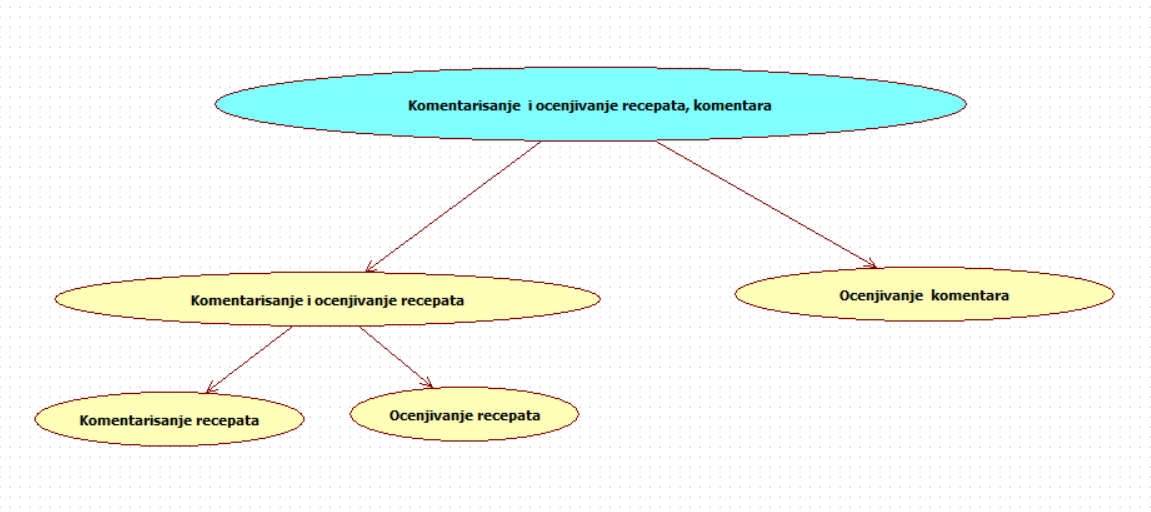


Slučajevi korišćenja *pregled recepata i informacija, verifikacija i brisanje recepata, komentarisanje i ocenjivanje recepata i komentara* obuhvataju složenije radnje koje se mogu dalje razložiti na pojedinačne slučajeve korišćenja.

Detaljni UML dijagram za slučaj korišćenja *pregled recepata i informacija* je prikazan na sledećoj slici:



 Detaljni UML dijagram za slučaj korišćenja *verifikaciju i brisanje recepata* je prikazan na sledećoj slici:

Detaljni UML dijagram za slučaj korišćenja *komentarisanje i ocenjivanje recepata, komentara* je prikazan na sledećoj slici:

## Kratak opis slučajeva korišćenja

### Pregled osnovne stranice web-aplikacije

Kratak opis: Prikaz početne stranice web-aplikacije

Akteri koji iniciraju slučaj korišćenja: Posetilac web-aplikacije, Prijavljeni korisnik, Prijavljeni korisnik sa privilegijama, Administrator

### Pregled recepta

Kratak opis: Prikaz stranice sa konkretnim receptom.

Akteri koji iniciraju slučaj korišćenja: Posetilac web-aplikacije, Prijavljeni korisnik, Prijavljeni korisnik sa privilegijama, Administrator

### Dodavanje komentara

Kratak opis: Dodavanje komentara o receptima.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Prijavljeni korisnik sa privilegijama, Administrator

### Kreiranje naloga

Kratak opis: Kreiranje novog naloga za prijavljenog korisnika.

Akteri koji iniciraju slučaj korišćenja: Posetilac web-aplikacije

### Ocenjivanje recepata

Kratak opis: Ocenjivanje recepata odabirom ocene od 1 do 5.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Prijavljeni korisnik sa privilegijama, Administrator

### Verifikacija recepata

Kratak opis: Verifikacija recepata koji prijavljeni korisnici žele da objave.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik sa privilegijama, administrator

### Dodavanje privilegovanih korisnika

Kratak opis: Unapređivanje prijavljenog korisnika u prijavljenog korisnika sa privilegijama.

Akteri koji iniciraju slučaj korišćenja: Administrator.

### Pretraga recepata

Kratak opis: Pretraga recepata na osnovu zadatih kriterijuma

Akteri koji iniciraju slučaj korišćenja: Posetilac web-aplikacije, Prijavljeni korisnik, Prijavljeni korisnik sa privilegijama, Administrator

### Pretraga recepata na osnovu namirnica iz frižidera

Kratak opis: Pretraga recepata na osnovu namirnica iz frižidera.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Prijavljeni korisnik sa privilegijama, Administrator

### Dodavanje namirnica u virtuelni frižider

Kratak opis: Korisnik dodaje namirnice u svoj virtuelni frižider.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Privilegovani korisnik,Administrator

### Čuvanje recepata

Kratak opis: Čuvanje recepata radi kasnijeg pregleda.

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, Privilegovani korisnik, Administrator

### Dodavanje recepta

Kratak opis: Korisnik dodaje recept u bazu podataka

Akteri koji iniciraju slučaj korišćenja: Prijavljeni korisnik, privilegovani korisnik.

### Lajkovanje komentara

Kratak opis: Dodavanje reakcije like/dislike na komentar.

Akteri koji iniciraju slučaj korišćenja: Prijavljani korisnik, Privilegovani korisnik, 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.

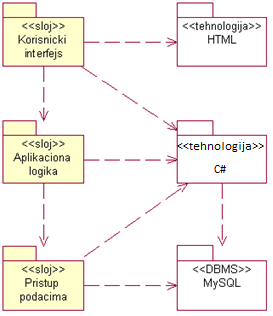
Logički pogled na ReceByte aplikaciju obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice,JS skripte, .NET Core fajlove 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 .NET Core fajlove zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži .NET Core fajlove 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

Ovaj sloj realizuje korisnički interfejs web aplikacije. U njemu je sadržan HTML, multimedijalni sadržaji, CSS u vidu Bootstrapa, .NET Core i JavaScript koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom. Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML, JavaScript i CSS.

### Aplikaciona logika

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi ReceByte web aplikacije. Sadrži web stranice koje 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 .NET Core paketa.

### Pristup podacima

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži .NET Core zadužen 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 .NET Core i MySQL baza podataka.

### HTML

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.

### .NET Core

Tehnologija C#-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 ReceByte aplikacije.

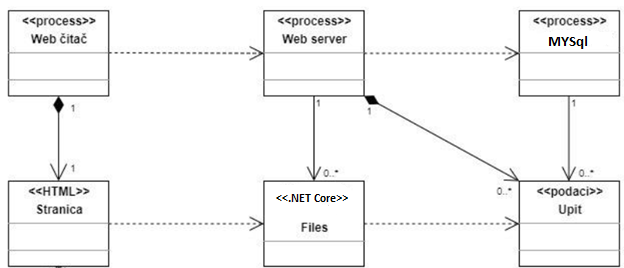
# Pogled na procese

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.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje Recebyte Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju ReceByte web aplikacije. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na .NET Core-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. Web server dobija zahteve od više Web klijenata, a zatim se obraća Aplikacionom serveru koji sadrži logiku koja obrađuje zahteve. Kada se ovi procesi završe, vraćaju se odgovarajući podaci za dati zahtev Web serveru koji generiše HTML, JavaScript i CSS fajlove kako bi se generisala stranica na klijentovoj strani.

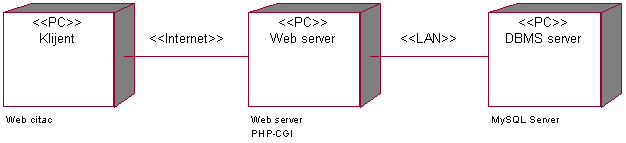
### MySQL Server

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 ReceByte aplikacije.



## Klijent

Pristup ReceByte aplikacije 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. 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).

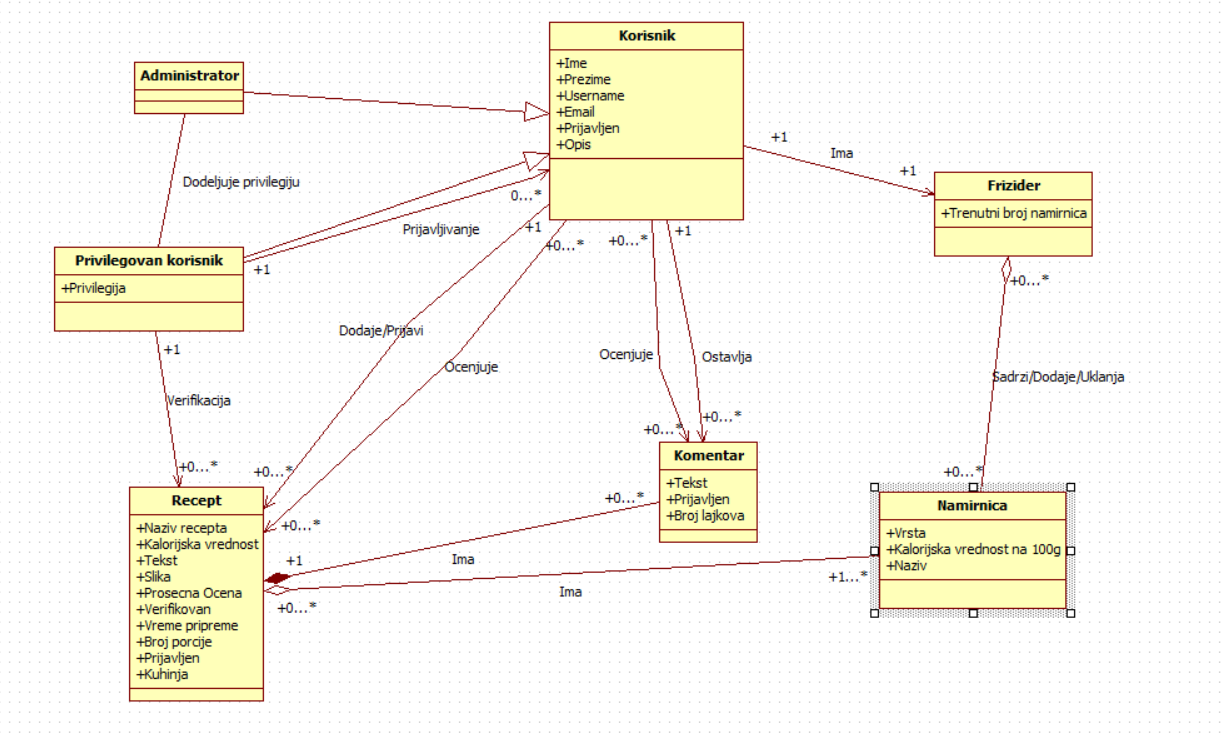
# Pogled na implementaciju sistema

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

## Model domena

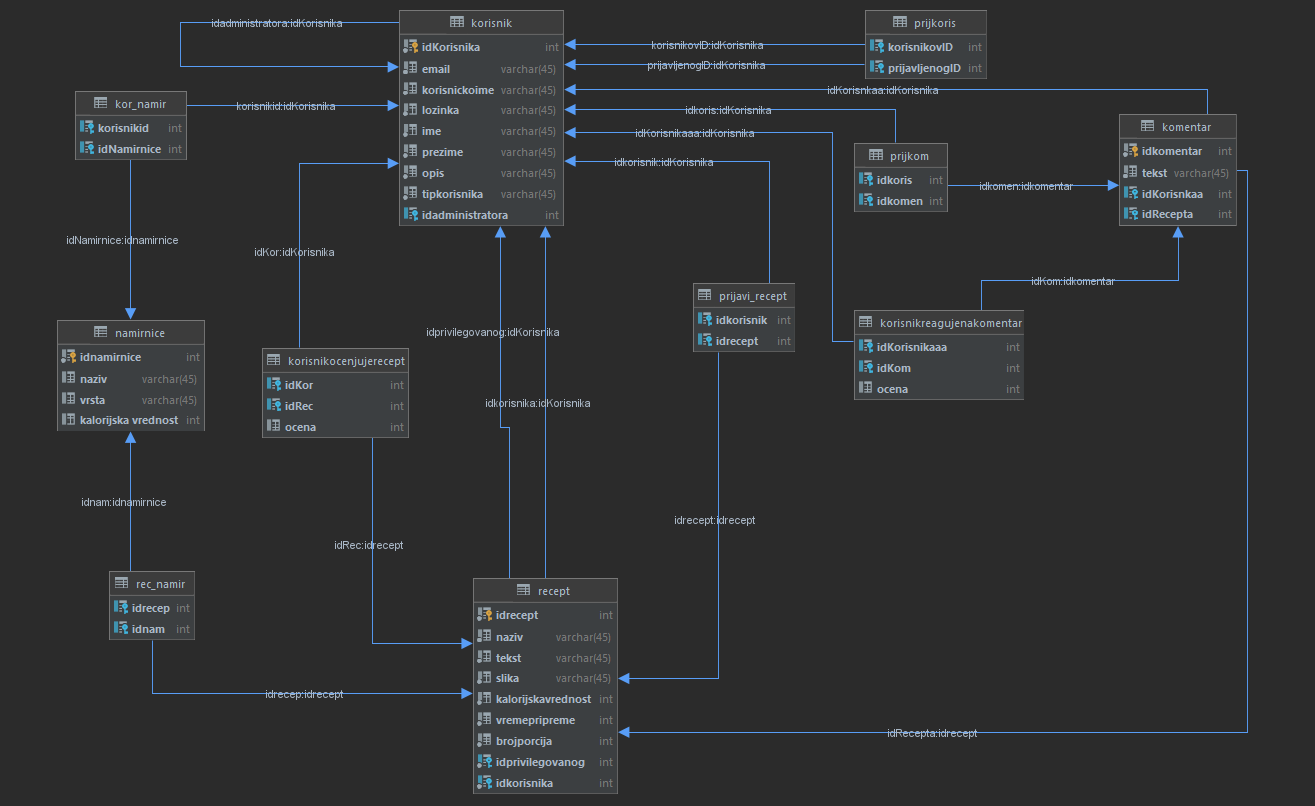
Model domena za koji se ReceByte aplikacija projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

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



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu. Baza podataka i dijagram su kreirani korišćenjem *DataGrip*-a, dok je migracija na MySQL obavljena pomoću *EF Core-a.*

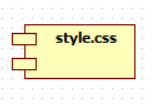


## Komponente sistema

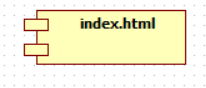
U folderu aplikacija nalazi se folder ReceByte u kome se nalaze fajlovi koji su zaduženi za generisanje statičkog sadržaja ove aplikacije. Ti fajlovi su html fajlovi (u kojima se nalazi implementacija web stranica zajedno sa bootstrap temom). U ovom folderu se nalazi folder assets u kome se nalazi folder img u kome su smeštene slike korišćene prilikom izrade aplikacije kao i folder css u kome se nalazi fajl stzle.css koji sadrži stilove koje aplikacija koristi, i folder js u kome se nalazi fajl main.js koji sadrži skripte koje su uključene u stranicu zajedno sa angular aplikacijama.

### Komponente korisničkog interfejsa

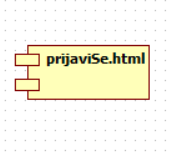
Dizajn korisničkog interfejsa za sve html stranice obuhvaćen je fajlom style.css



Komponenta index.html implementira početnu stranicu aplikacije koja je ista za sve tipove korisnika (posetioce, prijavljene korisnike, administratore i privilegovane korisnike). Ova stranica može voditi do ostalih stranica: recepti.html, prijaviSe.html (u slučaju posetioca), kontakt.html i user.html (u slučaju prijavljenog korisnika, administratora i privilegovanog korisnika).



Komponenta prijaviSe.html implementira stranicu preko koje se može kreirati profil ili se može prijaviti na već ostojeći profil. Do ove stranice se može doći preko index.html stranice, stranice recepti.html ili kontakti.html ukoliko se aplikaciji pristupa kao posetilac.



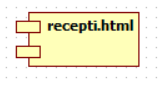
Komponenta user.html implementira početnu stranicu korisnika sa profilom. Do ove stranice se može doći preko index.html stranice, stranice recepti.html ili kontakti.html ukoliko se aplikaciji pristupa kao administrator, prijavljeni korisnik ili privilegovani korisnik. Preko ove stranice se može doći do stranice dodajRecept.html, stranica sadrži sekciju sa obaveštenjima u slučaju privilegovanog korisnika ili administratora.



Komponenta dodajRecept.html implementira stranicu za dodavanje novog recepta aplikaciji. Do ove stranice se može doći preko user.html stranice.



Komponenta recepti.html implementira stranicu na kojoj se nalaze svi recepti iz baze podataka aplikacije. Svaki od recepata na ovoj stranici može dovesti do stranice recept.html.



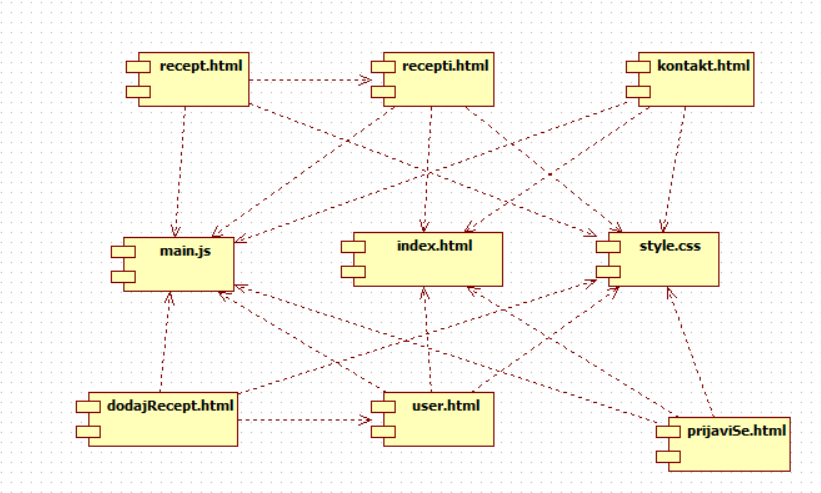
Komponenta recept.html implementira stranicu na kojoj se nalaze karakteristike recepta odabranog na stranici recepti.html. Ova stranica se razlikuje u zavisnosti od odabranog recepta. Do ove stranice se može doći preko index.html stranice, kontakt.html, stranice prijaviSe.html (u slučaju posetioca) ili user.html ukoliko se aplikaciji pristupa kao administrator, prijavljeni korisnik ili privilegovani korisnik.



Komponenta kontakt.html implementira stranicu na kojoj se nalaze informacije za kontaktiranje SANTima. Do ove stranice se može doći preko index.html stranice, recepti.html, stranice prijaviSe.html (u slučaju posetioca) ili user.html ukoliko se aplikaciji pristupa kao administrator, prijavljeni korisnik ili privilegovani korisnik.

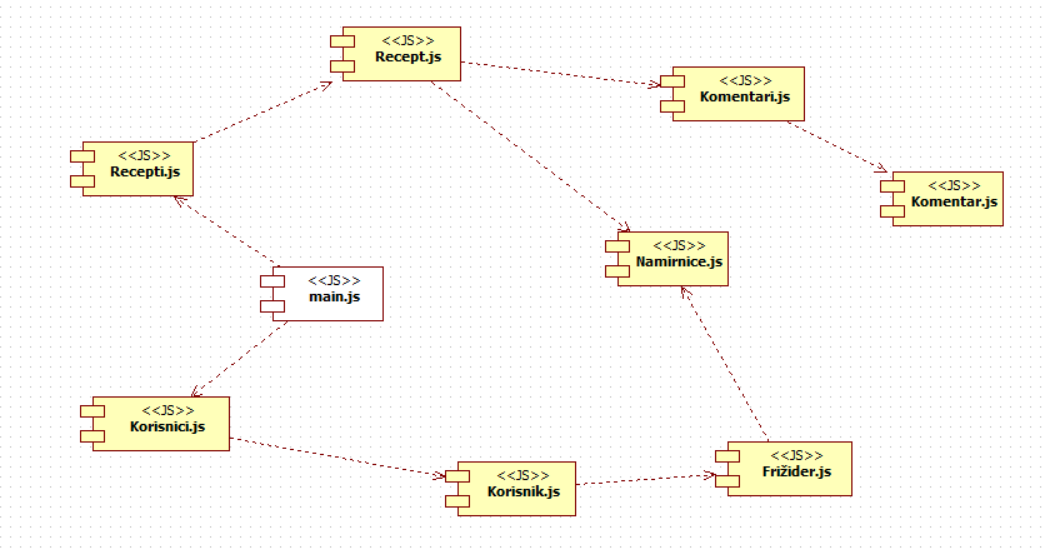


Na sledećoj slici prikazana je povezanost između komponenti.



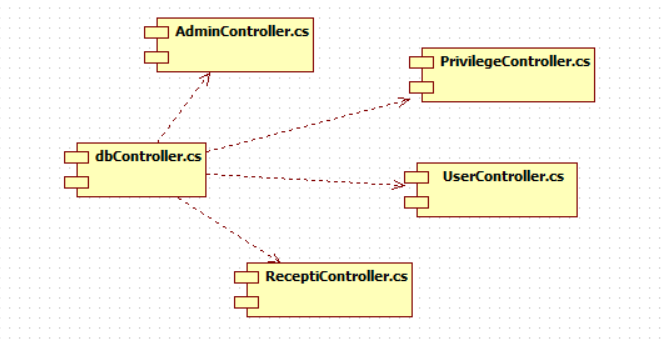
### Komponente aplikacione logike

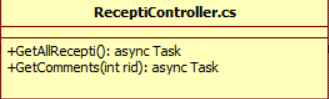
Komponente koje realizuju domen problema se uključuju isključivo preko **main.js** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



### Komponente za pristup podacima

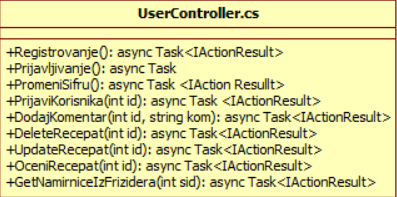
Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru **dbController.cs**.





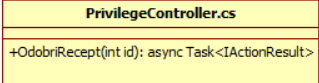
Navedene funkcije obavljaju sledeće zadatke:

* GetAllRecepti() - vraca sve recepte iz baze podataka
* GetComments(int rid) – vraca komentare za zadati recept



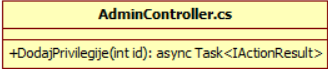
Navedene funkcije obavljaju sledeće zadatke:

* Registrovanje() – upisuje u bazu podataka iz forme registracija
* Prijavljivanje() – proverava da li korisnik postoji u bazi
* PromeniSifru() – menja sifru korisniku
* PrijaviKorisnika(int sid) – kreira zalbu sa unetim podacima i upisuje u bazu
* DodajKomentar(int rid, string kom) – upisuje u bazu uneti komentar
* DeleteRecepat(int rid) – brise recepat sa datim id-om
* UpdateRecepat(int rid) – ažurira recepat
* OceniRecepat(int rid) – upisuje ocenu za dati recepat
* GetNamirniceIzFrizidera(int sid) – vraća namirnice koje korisnik poseduje u frižideru



Navedene funkcije obavljaju sledeće zadatke:

* OdobriRecepat(int rid) – odobrava recepat koji je poslat na reviziju



Navedene funkcije obavljaju sledeće zadatke:

* DodajPrivilegije(int id) – dodaje privilegije prijavljenom korisniku

# 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 web aplikaciji.
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. ReceByte aplikacija će biti dostupana 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupana ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.