Vinyl

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 24.04.2020. | 1.0 | Inicijalna verzija | Luka, Dijana, Teodora, Bogdan |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

[1. Cilj dokumenta 5](#_Toc38754493)

[2. Opseg dokumenta 5](#_Toc38754494)

[3. Reference 5](#_Toc38754495)

[4. Predstavljanje arhitekture 5](#_Toc38754496)

[5. Ciljevi i ograničenja arhitekture 5](#_Toc38754497)

[6. Pogled na slučajeve korišćenja 5](#_Toc38754498)

[6.1 Dijagrami slučajeva korišćenja 6](#_Toc38754499)

[6.2 Kratak opis slučajeva korišćenja 8](#_Toc38754500)

[6.2.1 Logovanje na sistem 8](#_Toc38754501)

[6.2.2 Pregled informacija 8](#_Toc38754502)

[6.2.3 Pregled osnovnih podataka o gramofonskim pločama 8](#_Toc38754503)

[6.2.4 Izmena informacija o plocama 8](#_Toc38754504)

[6.2.5 Ostavljanje komentara 8](#_Toc38754505)

[6.2.6 Ocenjivanje ploce 8](#_Toc38754506)

[6.2.7 Pregled podataka o dogadjajima 8](#_Toc38754507)

[6.2.8 Kreiranje dogadjaja. 9](#_Toc38754508)

[6.2.9 Izmena događaja 9](#_Toc38754509)

[6.2.10 Pregled podataka o bendovima 9](#_Toc38754510)

[6.2.11 Pregled podataka o korisnicima 9](#_Toc38754511)

[6.2.12 Pregled spiska clanova 9](#_Toc38754512)

[6.2.13 Pregled podataka o odredjenom clanu 9](#_Toc38754513)

[6.2.14 Kreiranje novog clana 9](#_Toc38754514)

[6.2.15 Brisanje postojeceg clana 9](#_Toc38754515)

[6.2.16 Izmena informacija o clanu 9](#_Toc38754516)

[6.2.17 Prijavljivanje na sistem 9](#_Toc38754517)

[6.2.18 Logovanje 9](#_Toc38754518)

[6.2.19 Registracija 9](#_Toc38754519)

[6.2.20 Unos novih podataka 9](#_Toc38754520)

[6.2.21 Interakcija sa drugim korisnicima 10](#_Toc38754521)

[6.2.22 Slanje zahteva za prijateljstvo **.** 10](#_Toc38754522)

[7. Pogled na logičku arhitekturu sistema 10](#_Toc38754523)

[7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 10](#_Toc38754524)

[7.1.1 Korisnički interfejs 11](#_Toc38754525)

[7.1.2 Aplikaciona logika 11](#_Toc38754526)

[7.1.3 Pristup podacima 11](#_Toc38754527)

[7.1.4 Vue 11](#_Toc38754528)

[7.1.5 ASP.NET Core i EF Core 11](#_Toc38754529)

[7.1.6 SQL LocalDB 11](#_Toc38754530)

[8. Pogled na procese 11](#_Toc38754531)

[8.1 Procesi 12](#_Toc38754532)

[8.1.1 Web čitač 12](#_Toc38754533)

[8.1.2 Web server 12](#_Toc38754534)

[8.1.3 ASP.NET Core-EF Core 12](#_Toc38754535)

[8.1.4 SQLLocalDB 12](#_Toc38754536)

[9. Pogled na raspoređivanje sistema 12](#_Toc38754537)

[9.1 Klijent 12](#_Toc38754538)

[9.2 Web server 13](#_Toc38754539)

[9.3 DBMS server 13](#_Toc38754540)

[10. Pogled na implementaciju sistema 13](#_Toc38754541)

[10.1 Model sistema 13](#_Toc38754542)

[10.2 Šema baze podataka 14](#_Toc38754543)

[10.3 Komponente sistema 15](#_Toc38754544)

[10.3.1 Komponente korisničkog interfejsa 15](#_Toc38754545)

[10.3.2 Komponente aplikacione logike 15](#_Toc38754546)

[10.3.3 Komponente za pristup podacima 17](#_Toc38754547)

[11. Performanse 17](#_Toc38754548)

[12. Kvalitet 17](#_Toc38754549)

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture Vinyl portala.

# Opseg dokumenta

Dokument se odnosi na web aplikaciju - Vinyl, koja će biti razvijena od strane VLEteam-a. Vinyl je sajt namenjen gramofonskim pločama . Namena web aplikacije je interakcija ljubitelja gramofonskih ploča, druženje kao i razmena iskustva, preko domena interneta.

# Reference

Spisak korišćene literature:

1. Vinyl – Predlog projekta, SWE-Vinyl-01, V1.0, 2020, VLEteam.
2. Vinyl – Planirani raspored aktivnosti na projektu, V1.0, 2020, VLEteam.
3. Vinyl – Plan realizacije projekta, V1.0, 2020, VLEteam.
4. Vinyl – Vizija sistema, V1.0, 2020, VLEteam.
5. Vinyl – Specifikacija zahteva, V1.0, 2020, VLEteam.

# 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. Vinyl projekat će biti implementiran kao Web aplikacija zasnovana na Vue framework-u i SQL LocalDB bazi podataka.
2. Klijentski deo Vinyl projekta će biti optimizovan za sledeće Web čitače: Internet Explorer 6.0 i noviji, Opera 8.0 i noviji, kao i Firefox (Mozilla).
3. Svi zahtevi u pogledu performansi, dati u specifikaciji zahteva, moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

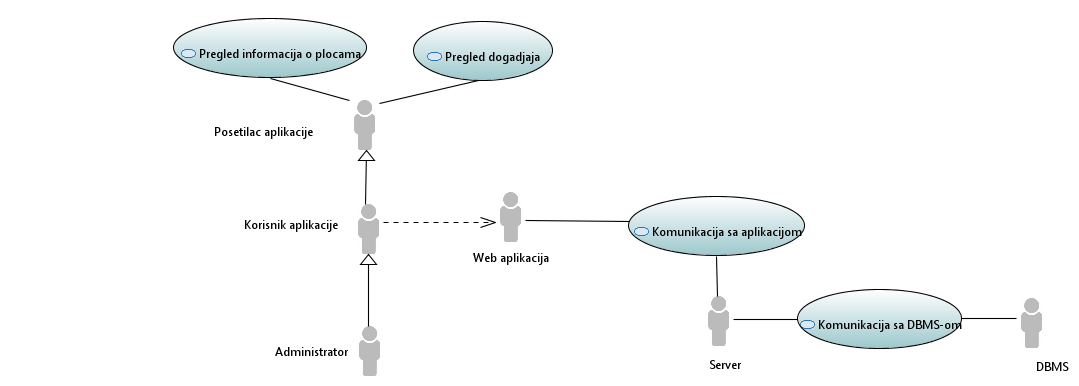
U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva.

Slučajevi korišćenja Vinyl aplikacije su:

* *Pregled informacija*
  + *Pregled osnovnih podataka o plocama*
    - Izmena informacija o plocama
    - Ostavljanje komentara
    - Ocenjivanje ploce
  + *Pregled podataka o dogadjajima*
    - Kreiranje dogadjaja
    - Izmena dogadjaja
  + *Pregled podataka o bendovima*
  + *Pregled podataka o korisnicima*
    - Pregled spiska korisnika
    - Pregled podataka o određenom clanu
      * Kreiranje novog clana
      * Brisanje postojeceg clana
      * Izmena informacija
* *Prijavljivanje na sistem*
  + *Logovanje*
  + *Registracija*
    - Unos novih podataka
* *Interakcija sa drugim korisnicima*
  + *Slanje zahteva za prijateljstvo*

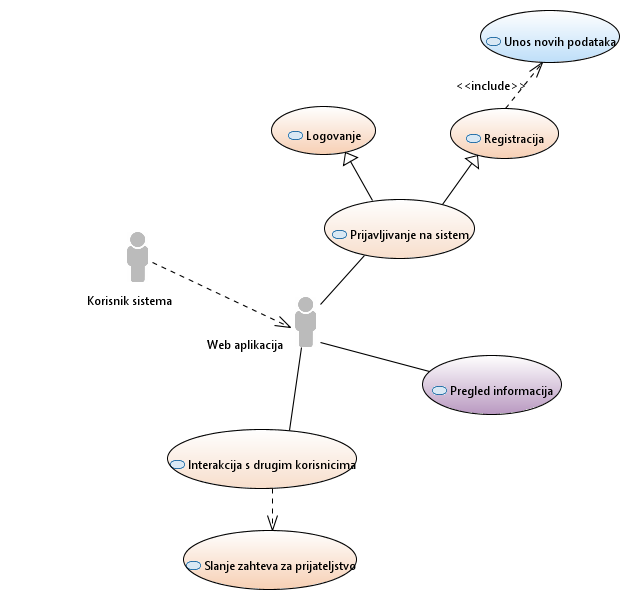
Ove slučajevi korišćenja mogu da iniciraju posetilac aplikacije, korisnik aplikacije ili administrator.

## Dijagrami slučajeva korišćenja

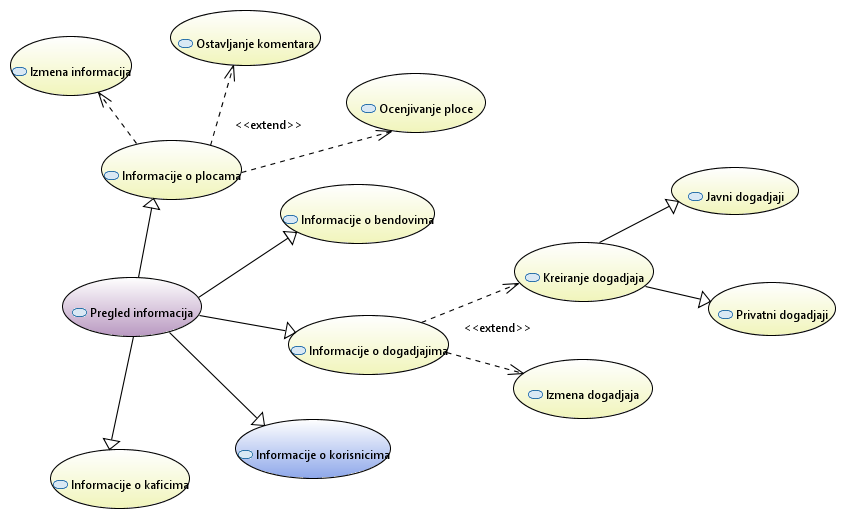
Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja Vinyl portala prikazan je na sledećoj slici:

Slučaj korišćenja je komunikacijasa Web aplikacijom, obuhvata složenije radnje i može se dalje razložiti na pojedinačne slučajeve korišćenja.

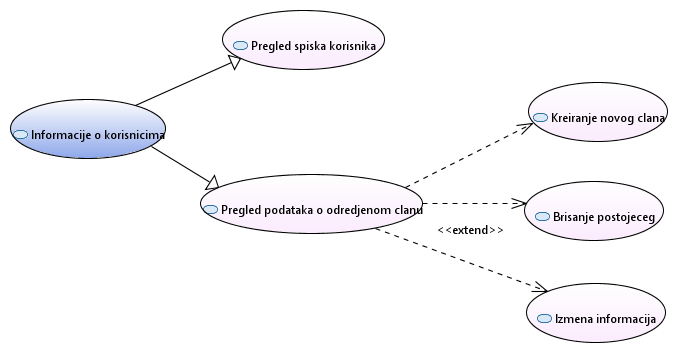
Detaljni dijagram za slučaj komunikacije sa Web aplikacijom prikazan je na sledećoj slici :



U zavisnosti od toga koji tip infomacija želi korisnik da pregleda, dobijamo detaljniji dijagram:



Tip informacija o korisniku na sledećem dijagramu:



## Kratak opis slučajeva korišćenja

### Logovanje na sistem

Kratak opis: Korisnik se loguje na sistem svojim korisničkim nalogom, a u cilju korišćenja Vinyl-a. Akteri koji iniciraju ovaj slucaj koriscenja su Korisnik sistema i Administracija sa preduslovom da nisu trenutno prijavljeni.

### Pregled informacija

Kratak opis: Prikaz stranice aplikacije sa osnovnim podacima o njoj. Informacije o plocama, bendovima, dogadjajima.  
Akteri su Korisnik sistema, Posetilac i Adminisracija

### Pregled osnovnih podataka o gramofonskim pločama

Kratak opis: Prikaz stranice portala sa osnovnim podacima o albumima i izvođačima.  
Akteri su Posetilac portala, Korisnik sistema i Administracija.

### Izmena informacija o plocama

Kratak opis: Izmena slike ploce, teksta ili demo verzije audio snimka.

Akter je Administracija.

### Ostavljanje komentara

Kratak opis: Ispod ploce je deo za diskusije. Moze se ostaviti komentar vezan za datu plocu.

Akter je Korisnik sistema.

### Ocenjivanje ploce

Kratak opis: Na svakoj strani sa plocom postoji deo za ocenjivanje ploce, tj.da li nam se ta ploca dopada ili ne, ukoliko smo je slusali.

Akter je Korisnik sistema.

### Pregled podataka o dogadjajima

Kratak opis: Kada se dogadjaj odrzava, gde, koja se muzika tamo slusa i spisak ljudi koji dolaze

Akteri su Korisnik sistema, Posetilac i Administracija

### Kreiranje dogadjaja.

Kratak opis: Dodavanje novog dogadjaja u kalendar aktivnosti.  
Akteri su Korisnik sistema i Administracija.

### Izmena događaja

Kratak opis: Izmena informacija/brisanje postojećeg događaja od strane korisnika koji je taj događaj kreirao.  
Akteri su Korisnik sistema i Administracija.

### Pregled podataka o bendovima

Kratak opis: Na svakoj strani sa plocom se nalazi i kratak opis izvodjaca, tj.benda.

Akteri su Korisnik sistema, Posetilac i Administracija

### Pregled podataka o korisnicima

Kratak opis: Prikaz informacija o svim korisnicima.

Akter koji inicira slučaj korišćenja je Administrator.

### Pregled spiska clanova

Kratak opis: Prikaz stranice sa spiskom članova .

Akter koji inicira slučaj korišćenja je Administrator.

### Pregled podataka o odredjenom clanu

Kratak opis: Prikaz stranice sa informacijama o određenom članu..

Akteri su Korisnik sistema i Administrator.

### Kreiranje novog clana

Kratak opis: Kreiranje korisničkog naloga za novog člana.

Akter je Administrator.

### Brisanje postojeceg clana

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg člana.

Akteri su Administrator i Korisnik sistema

### Izmena informacija o clanu

Kratak opis: Izmena korisničkog naloga i podataka za postojećeg člana.

Akteri su Administrator i Korisnik sistema

### Prijavljivanje na sistem

Kratak opis: Podrazumeva logovanje ili registraciju, u zavisnosti od toga da li korisnik vec ima nalog.

Akter je Korisnik sistema.

### Logovanje

Kratak opis: Prijavljivanje na sistem pomocu vec kreiranog naloga.

Akter je Korisnik sistema.

### Registracija

Kratak opis: Kreiranje naloga u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akter je Korisnik sistema.

### Unos novih podataka

Kratak opis: Korisnik, prilikom registracije, unosi validne podatke o sebi, u cilju kreiranja sopstbenog naloga.

Akter je Korisnik sistema.

### Interakcija sa drugim korisnicima

Kratak opis: Podrazumeva slanje zahteva drugim korisnicima i pozivanje na javne ili privatne dogadjaje.

Akter je Korisnik sistema.

### Slanje zahteva za prijateljstvo **.**

Kratak opis: Član dodaje željene osobe za prijatelje u sistemu, koji imaju pristup njihovim podacima i spisku pozvanih korisnika za dogadjaj kreiran od strane clana.

Akteri su Korisnik sistema i Administracija.

.

# 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.

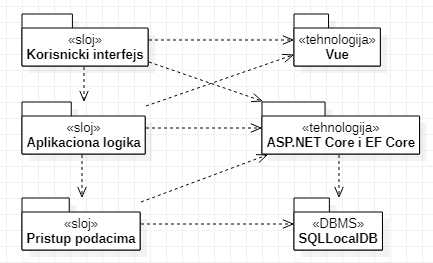
Logički pogled na Vinyl sistem obuhvata 3 glavna paketa: Klijentske aplikacije, Serverska aplikacija i Pristup podacima.

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

Veza između podsistema se ostvaruje korišćenjem paketa *Aplikaciona logika.*Ona predstavlja srednji sloj sistema koji sadrži Vue biblioteku namenjenu izradi same Web stranice prethodnog sloj i skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži klase koje predstavljaju interfejs za siguran pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

* Sloj

Grafički korisnički interfejs je softversko okruženje koje omogućava korisniku adekvatnu komunikaciju sa kompjuterom koristeći pojedine funkcije. Realizovan je koriscenjem Vue i Bootstrap framework-a kao i ASP.NET Core metoda za razmenu podataka. U njemu su sadrzane sve HTML stranice, multimedijalni sadržaji, objekti (route) sa informacijama , vue prodavnice i komponente.

Zavisi od sloja aplikacione logike, kao i framework-ova Vue i ASP.NET Core.

### Aplikaciona logika

* Sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi Vinyl portala. Sadrži Vue komponente koje realizuju funkcionalnost karakterističnu za domen primene portala i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima, zaštita od ilegalnog pristupa podacima se vrši korišćenjem ASP.NET Core-a primenom JSON Web tokena. Komunikacija izmedju klijentske aplikacije i DBMS-a realizuje se ASP.NET projektom, uz koriscenje Entitiy framework-a.

Ovaj sloj zavisi od sloja za pristup podacima i ASP.NET Core i EF Core paketa.

### Pristup podacima

* Sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture zadužen za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u SQLLocalDB bazi podataka,a pribavljaju preko entity framework-a.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od paketa ASP.NET Core i EF i SQL LocalDB baze.

### Vue

* Tehnologija

Vue tehnologijaili progresivni frameworkza izradu korisničkih interfejsa ima skup opcionalnih alata i glavnu biblioteku fokusiranu na sloj prikaza.

Tehnologija Vue definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### ASP.NET Core i EF Core

* Tehnologija

Tehnologija ASP.NET je framework otvorenog koda što znači da je njegov izvorni kod dostupan svima na uvid. To je konzolna aplikacija koja kreira web poslužitelj u Main metodi i obezbeđuje se mehanizam za pozivanje api-a preko URL-a. Vinyl aplikacija je kreirana kao code-first aplikacija i za mapiranje koda na bazu podataka koristi se EF Core.

### SQL LocalDB

DBMS

Kopira minimalan skup datoteka, potreban za pokretanje SQL servera, za upravljanje bazama podataka koji će se koristiti za realizaciju Vinyl portala.

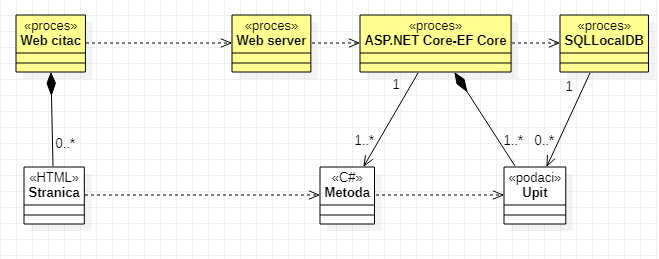
# Pogled na procese

U ovom odeljku je sadržan pogled na procesnu arhitekturu Vinyl 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 Vinyl portala kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju Vinyl portala.



### 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.

### ASP.NET Core-EF Core

ASP.NET Core proces obavlja posao obrade zadate preko upita i C# metode, koja generiše odgovarajući sadržaj koji Web server šalje Web čitaču. Za izvršenje može da zahteva usluge SQL LocalDB servera-a. Komunikacija izmedju klijentske aplikacije i DBMS-a realizuje se uz pomoć ASP.NET Core i Entitiy framework-a.

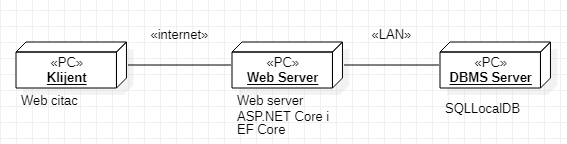
### SQLLocalDB

SQL LocalDB Server je proces koji izvršava funkcionalnost MySQL sistema za upravljanje bazama podataka. Ovaj proces prihvata zahteve, manipuliše podacima, i izvršava zahteve nad bazom podataka i vraća dobijene rezultate prethodnom procesu.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju igre. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa i niti.

Na sledećoj slici dat je UML dijagram raspoređivanja Vinyl sistema.



## Klijent

Pristup Vinyl portalu se obavlja preko klijentskih računara na kojima se izvršava Web čitač. Jedini način za pristup DBMS-u je preko Web Server čvora koji se obraća DBMS-u. Za povezivanje između Klijenta i Web Servera, čvorova, koristi se Internet tako da nema ograničenja u pogledu lokacije Web Server čvora.

## 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 . Web server može da bude na istom računaru kao i DBMS ili može da bude povezan sa njim lokalno.

## DBMS server

DBMS je čvor na kome se izvršava SQL server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. DBMS može, ali ne mora da se nalazi na istom računaru kao i Web server ali zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

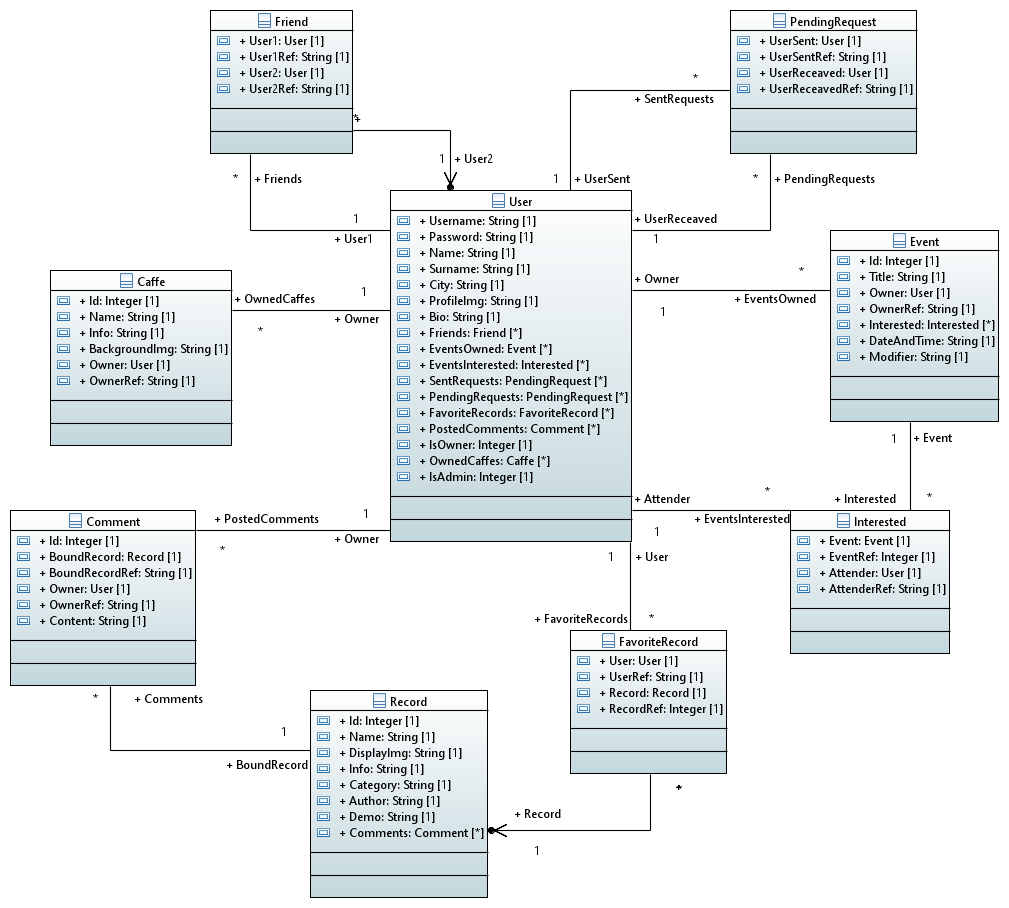
# Pogled na implementaciju sistema

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

## Model sistema

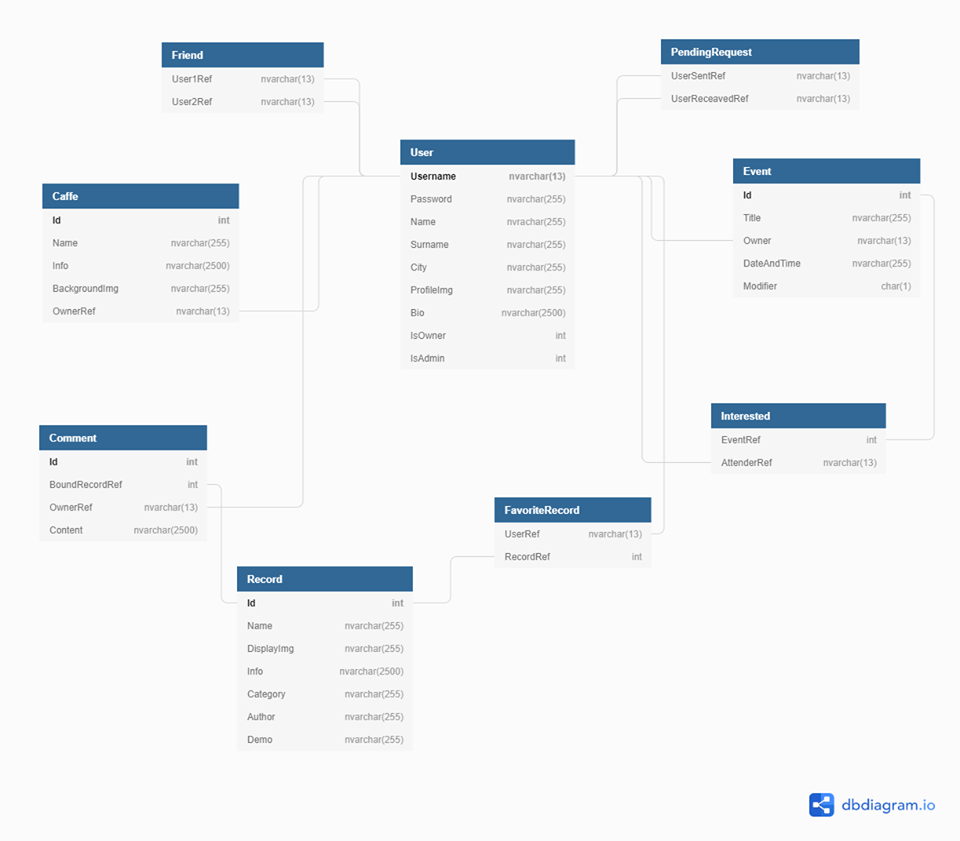
Model sistema za koji se Vinyl projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane sistemske komponente, kao i veze koje se mogu identifikovati između njih.

Model sistema 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 je nastala mapiranjem C# koda. Ovu funkcionalnost obezbeđuje EF Core.



## Komponente sistema

Komponente sistema Vinyl portala su Vue komponente čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:

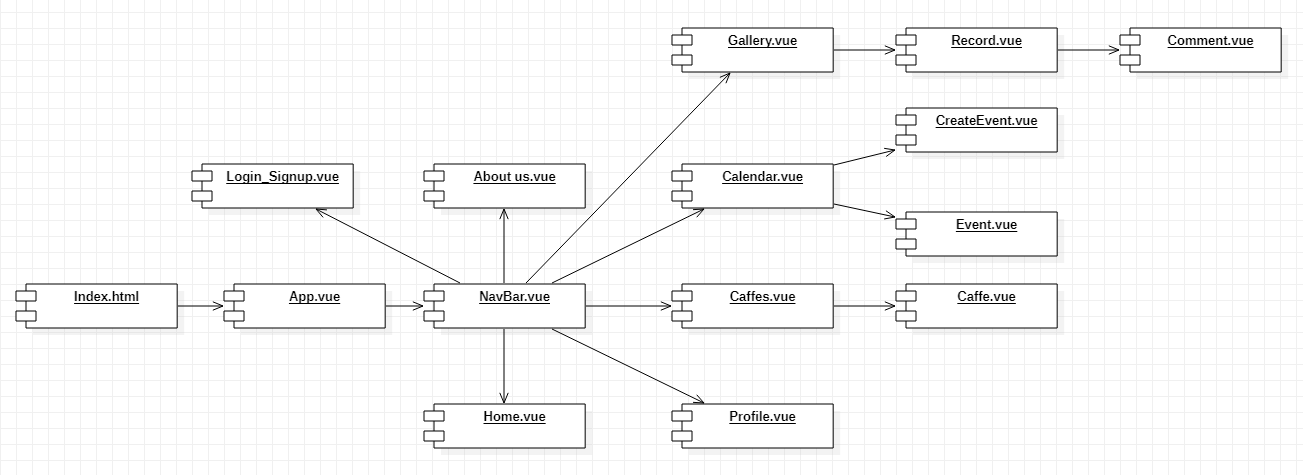


Komponenta **index.html** implementira I prikazuje komponentu **App.vue.**

Komponenta **App.vue** predstavlja kontejner čiji se sadržaj menja u odnosu na stranice koje korisnik izabere za prikaz.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.html i App.vue** komponenti korisničkog interfejsa. Na taj način zadržavaju sva definisana podešavanja stila. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



10.3.2.1 **NavBar.vue** komponentna predstavlja navigaciju koja sadrži linkove ka ostalim stranicama aplikacije.

10.3.2.2 **LogIn/SignUp.vue** prestavlja formu za registrovanje i/ili prijavljivanje korisnika na sistem.

10.3.2.3 **Home.vue** prestavlja početnu stranu aplikacije.

10.3.2.4 **About\_us.vue** predstavlja stranicu koja sadrži kratak opis o našoj aplikaciji.

10.3.2.5 **Gallery.vue** predstavlja stranicu galerije naše aplikacije u kojoj se nalaze informacije o pločama, bendovima kao i korisnicki komentari o plocama.

10.3.2.6 **Record.vue** komponenta je gradivni elemnt **Gallery.vue** stranice. **Record.vue** sadrži informacije o ploči, izvođačkom bendu, kratak opis ploče, ocenu, korisnicke komentare o ploči i kratak demo.

10.3.2.7 **Comment.vue** komponenta predstavlja deo **Record.vue** komponente, koja sluzi za dodavanje korisničkih komentara o ploči.

10.3.2.8 **Calendar.vue** predstavlja stranicu na kojoj se nalaze svi trenutno aktivni događaji. Korisnici mogu videti informacije o svim javnim događajima.

10.3.2.9 **CreateEvent.vue** komponenta daje korisniku mogućnost da kreira i pošalje zahtev za kreiranje događaja u određenom kafiću.

10.3.2.10 **Event.vue** komponenta predstavlja gradivni deo **Calendar.vue** stranice. Sadrži informacije o tome kog datuma se održava događaj, u koje vreme, u kom kafiću, ko je organizator događaja kao i listu korisnika koji dolaze na događaj i listu korisnika koji su zainteresovani.

10.3.2.11 **Cafes.vue** predstavlja stranicu koja prikazuje kafiće.

10.3.2.12 **Cafe.vue** komponenta predstavlja gradivni element stranice **Cafes.vue**. **Cafe.vue** sadrži informacije o kafiću kao što su: naziv, slika, lokacija, kratak opis i vlasnik kafića.

10.3.2.13 **Profile.vue** predstavlja stranicu profila korisnika, na kojoj korisnik moze da vidi sve svoje informacije, i listu svojih prijatelja.

### Komponente za pristup podacima

Pristup bazi podataka se vrsi kroz nekoliko komponenti.

10.3.3.1 **LogIn/SignUp.vue** komponenta. Pri registraciji ili prijavljivanju na sistem korisnik unosi svoje podatke. Ako je u pitanju prijavljivanje proverava se da li unešeni podaci postoje u bazi i u zavisnosti od rezultata izvršava se određena akcija(ako su podaci ispravni, vrši se prijavljivanje korisnika na sistem, u suportnom se vraća poruka sa greškom). Ako je u pitanju registrovanje korisnika, prvo se vrši validacija podataka u samoj formi, a zatim se pristupa bazi i provera da li postoji korisnik sa istim korisničkim imenom. Ako postoji, korisniku se vraća poruka da je korisničko ime zauzeto, u suportnom, vrši se kreiranje naloga korisnika, ažurira se baza podataka, a zatim i prijavljivanje korisnika na sistem.

10.3.3.2 **Comment.vue** komponenta. Pri dodavanju korisničkog komentara određenoj ploči, vrši se pristup bazi, zatim se unešeni komentar unosi u bazi i vezuje za određenu ploču, baza se ažurira, a zatim se korisniku prikazuje ažurirana web strana.

10.3.3.3 **CreateEvent.vue** komponenta. Korisnik popunjava formu za kreiranje događaja i zatim se zahtev šalje vlasniku odabranog kafica. Ako vlasnik odobri zahtev, kreira se novi događaj, unosi se u bazu, baza se ažurira i korisniku se prikazuje ažurirana web strana

# 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:

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.

# Kvalitet

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

1. Vinyl 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.