Arhitektura i projektovanje softvera

**Ommi**

- Web aplikacija za kolaborativno stvaranje muzike -

**Arhitekturni dizajn**

Antonije Petrović 16279

Darko Radivojević 16301

Sadržaj

[1. Osnovne funkcionalnosti softverskog sistema 3](#_Toc26290650)

[1.1 Sistem korisničkih naloga 3](#_Toc26290651)

[1.2 Sistem soba 3](#_Toc26290652)

[1.3 Sviranje 3](#_Toc26290653)

# 1. Osnovne funkcionalnosti softverskog sistema

## 1.1 Sistem korisničkih naloga

### 1.1.1 Registracija

Registracija novog korisnika. Vrši se zadavanjem email adrese i lozinke. Takođe, bira se i korisničko ime. Vrši se provera da li je uneti email zauzet, a ista provera se radi i za uneto korisničko ime.

### 1.1.2 Prijavljivanje

Prijavljivanje na aplikaciju pomoću email-a/korisničkog imena i lozinke. Vrši se provera unetih podataka i eventualno obaveštavanje korisnika ukoliko su uneti podaci pogrešni.

## 1.2 Sistem soba

### 1.2.1 Kreiranje nove sobe

Kreiranje nove sobe (novog projekta) kome mogu pristupati drugi korisnici. Bira se da li će soba biti javna ili će pristup biti omogućen unošenjem lozinke. Bira se ime sobe, i maksimalan broj učesnika koje soba može da primi.

### 1.2.2 Ulazak u postojeću sobu

Iz liste soba na početnoj stranici, bira se neka javna soba i moguće je pristupiti joj ukoliko ima praznih mesta. Takođe, privatnim sobama je moguće pristupiti ako se unese ispravna lozinka.

### 1.2.3 Pretraga i filtriranje soba

U listi soba je moguće pretraživati trenutno dostupne sobe po imenu, a moguće je i filtrirati ih po vrsti – javne ili privatne.

### 1.2.4 Invite u sobu pomoću linka

Moguće je generisati link za pristup sobi koji korisnik može poslati drugim korisnicima kako bi se oni pridružili sobi koju je napravio.

## 1.3 Sviranje

### 1.3.1 Dodavanje instrumenta

U okviru sobe, tj. kreiranog projekta, moguće je dodati novi instrument. Instrument se bira iz liste trenutno podržanih instrumenata i dodaje se u projekat, tako da svi trenutni učesnici vide tu promenu.

### 1.3.2 Dodavanje note

Biranjem odgovarajućeg trenutka u vremenu, za odabrani instrument je moguće dodati “notu”, tj. moguće je označiti da u datom trenutku treba odsvirati zvuk tog instrumenta.

### 1.3.3 Pokretanje, pauziranje, zaustavljanje loop-a

Korisnik može pokrenuti, pauzirati ili zaustaviti loop, pri čemu se ova promena prenosi do svih učesnika. Loop je glavna petlja, podeljena na 16 delova, koja se vrti u krug dok god je uključena. Jedan prolazak kroz loop predstavlja jedan takt u muzičkom smislu.

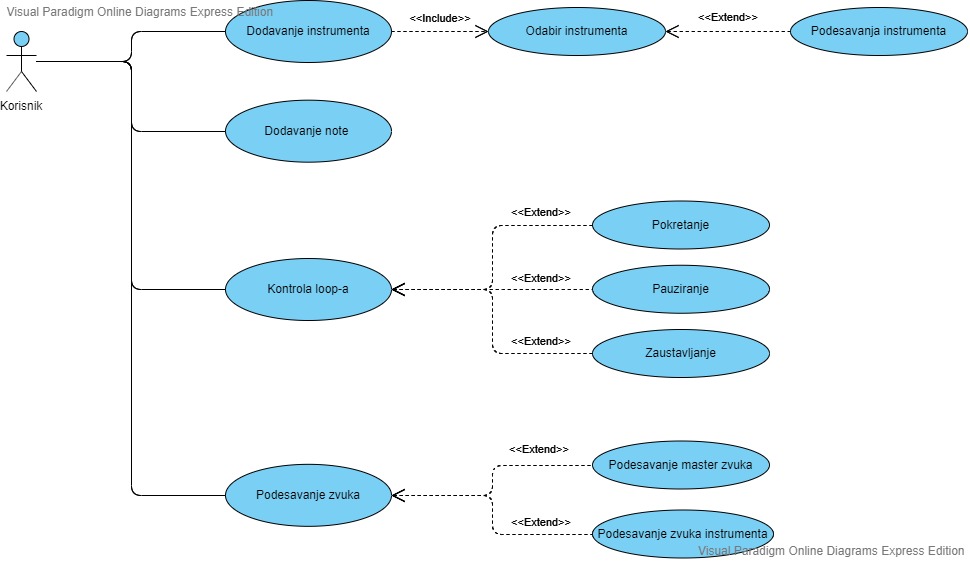
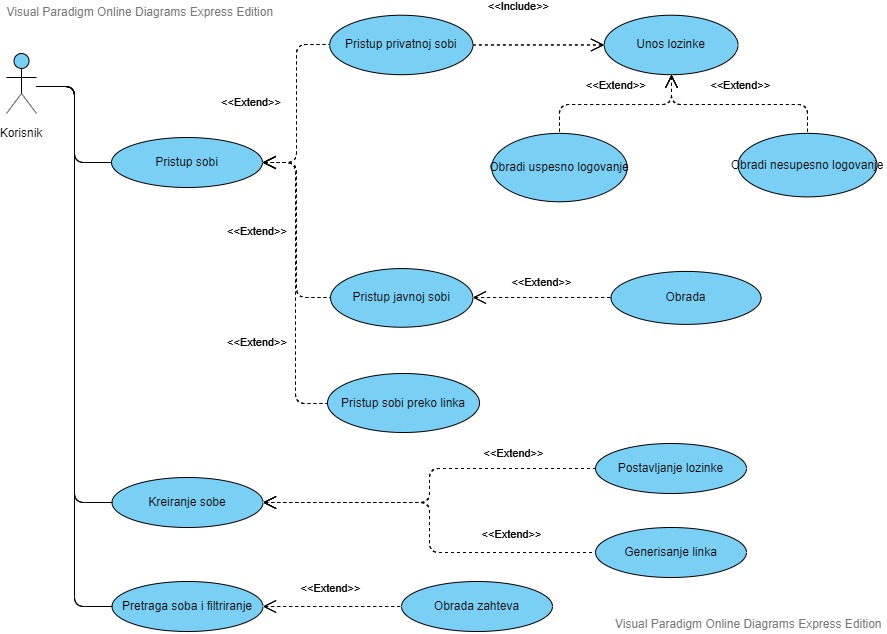
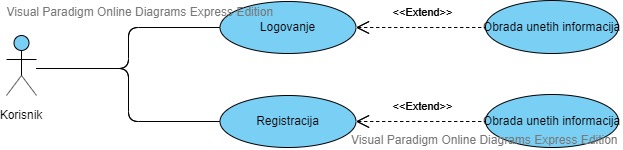
### 1.3.4 Podešavanje zvuka instrumenta

Moguće je podešavati zvuk pojedinačnog instrumenta. Ova podešavanja se odnose na podešavanje jačine zvuka instrumenta, mutiranje zvuka, podešavanja boje, itd.

### 1.3.5 Master podešavanje zvuka

Master podešavanja zvuka se odnose na ukupnu jačinu izlazne melodije, definisanje tempa u jedinicama BPM (*eng. Beats Per Minute*), itd.

# 2. Slučajevi korišćenja



# 3. Nefunkcionalni zahtevi

## 3.1 Minimalno vreme odziva

Zbog muzičke prirode aplikacije, od velike je važnosti da vreme odziva bude što manje. Obrada dolazećih promena i brzo obaveštavanje svih zainteresovanih strana o nastalim promenama predstavljaju bitan zahtev. Potrebno je da 90% zahteva bude obrađeno i na njih bude odgovoreno u roku od najviše 2s.

## 3.2 Upotrebljivost

Aplikaciju treba realizovati tako da je jednostavna za korišćenje i pruža dobro korisničko iskustvo.

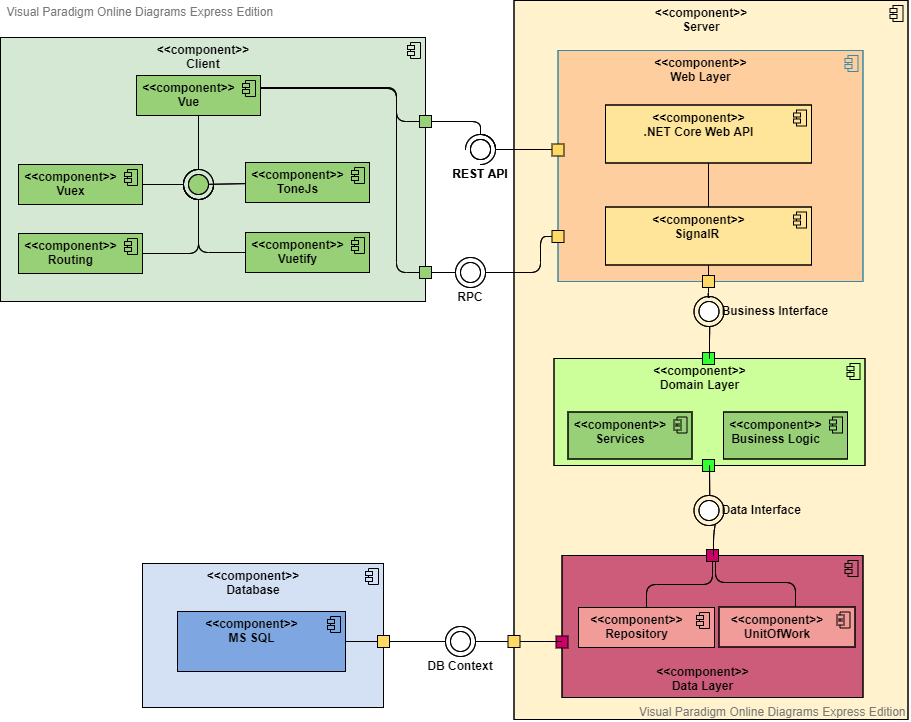
## 3.3 Skalabilnost

Aplikacija treba da bude u stanju da lako podrži veliki broj istovremenih korisnika, kao i nagle, kratkoročne skokove u broju korisnika koji koriste aplikaciju. Podaci koji će se prenositi ne predstavljaju velike podatke, pa to neće praviti probleme.

## 3.4 Sigurnost

Potrebno je obezbediti sigurno prenošenje i čuvanje korisničih podataka, pre svega lozinki i drugih osetljivih informacija.

# 4. Arhitekurni dizajn



## 4.1 Klijent

Na klijentskoj strani, koristićemo Vue.js aplikacioni okvir (framework). Vuex će omogućiti jedinstveno upravljanje svim deljenim resursima, dok će Vue Routing omogućiti upravljanje rutama na klijentskoj aplikaciji. Vuetify je Vue UI biblioteka koja olakšava rad sa izgledom aplikacije prateći principe Material Design-a. S obzirom da je aplikacija namenjena radu sa zvukom, koristićemo i neku audio JavaScript biblioteku poput – Tone.js ili Howler.js.

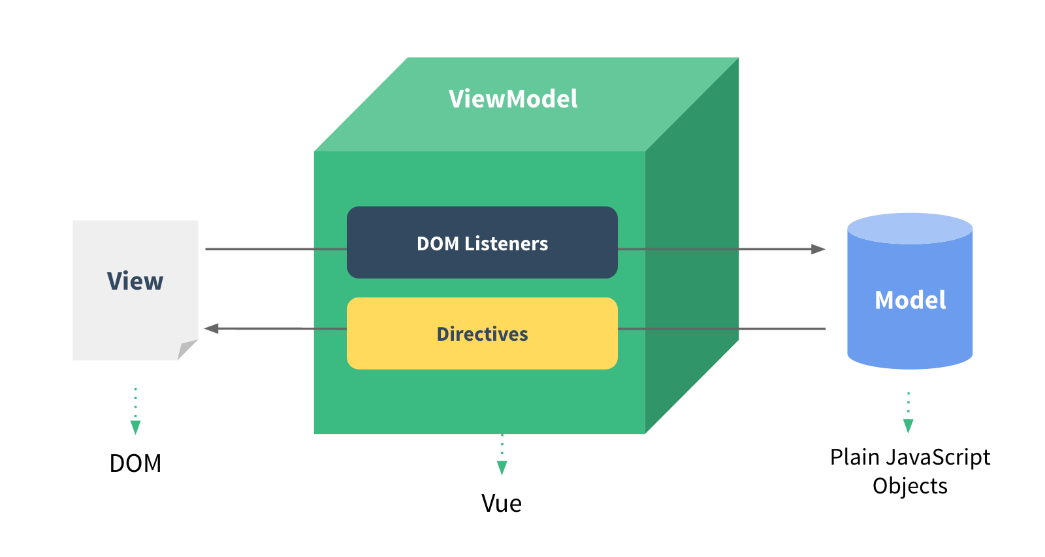
### Zadatak: Razmotriti kako klijentski aplikacioni okvir implementira MVC obrazac

Za Vue.js se može reći da ne implementira MVC obrazac, već MVVM (Model-View-ViewModel). Zapravo, Vue sam po sebi predstavlja deo ovog obrasca – ViewModel. Ovaj framework dvosmerno povezuje View i Model.

**ViewModel** je objekat koji sinhronizuje Model i View. Svaka Vue instanca je jedan ViewModel.

var vm = new Vue({ /\* options \*/ })

**View** je DOM element kojim upravljaju Vue instance. Svaka Vue instanca je povezana sa određenim DOM elementom.   
**Model** je objekat koji predstavlja običan JavaScript objekat. Kada se jednom nađe unutar neke Vue instance, on postaje reaktivan i kada dođe do promene njegovih atributa, sve Vue instance koje su zainteresovane će biti obaveštene o nastalim promenama.



## 4.2 Server

Klijent će sa serverom komunicirati preko RESTful API-ja koji će biti implementiran kao .NET Core WebAPI. Za komunikaciju koju inicira server, koristićemo SignalR – biblioteku .NET-a koja omogućava asinhrono obaveštavanje klijenata sa strane servera u realnom vremenu. SignalR koristi WebSocket protokol kao i neke prednosti HTML5 standarda. Takođe, SignalR omogućava i serverski RPC, pri čemu se mogu pozivati JavaScript funkcije u klijentskom pretraživaču, iz serverskog .NET koda. Obe komponente, i WebAPI i SignalR, koristiće zajedničku domensku logiku aplikacije koja će biti smeštena u Domain sloju. Ovaj sloj će, da bi čuvao i koristio podatke aplikacije, komunicirati sa slojem podataka. Na sloju podataka koristićemo EntityFramework Core kao ORM alat za MS SQL relacionu bazu. Ovaj framework implementira UnitOfWork obrazac u obliku DbContext klase, pri čemu se on sastoji od više DbSet-ova koji predstavljaju tabele u bazi podataka i čine Repository obrazac.

## 4.3 Baza podataka

Koristićemo Microsoft SQL Server kao DBMS za bazu podataka.