

# **Arhitekturni dizajn softverskog sistema**

## **ScrumPRO**

**Mihailo Rašić, 15850**

**Aleksandar Trajković, 15944**

**Marko Tomašević, 15940**

## **Opis projekta**

Aplikacija ScrumPRO je Web aplikacija zamišljena da pruži podršku klijentima prilikom kreiranja softverskih proizvoda. Aplikacija će da se koristi prilikom agilnog razvoja softvera vodeći se scrum procesom. ScrumPRO omogućava korisnicima (kako menadžmentu, tako i developerima) da imaju uvid u zadatke koje treba da obave, kao i da znaju u kojoj su fazi oni zadaci koji se trenutno obavljaju.

## **Funkcionalni zahtevi**

- Kreiranje novog naloga i prijava na nalog.
- Kreiranje projekta i pristupanje određenom projektu.
- Mogućnost kreiranja story-ja unutar određenog projekta.
- Mogućnost podele story-ja na task-ove.
- Dodavanje story-ja iz backlog-a u novi sprint.
- Koordinacija story-ja i task-ova u zavisnosti od toga u kojoj se fazi nalaze u okviru sprinta.
- Uređivanje svog profila.

## **Nefunkcionalni zahtevi**

- Sigurnost i bezbednost – kako aplikacija radi sa “osetljivim” podacima. Neophodno je voditi računa o sigurnosti kako aplikacije tako i korisnika (podaci moraju biti dobro čuvani, a korisnici imaju svoj jedinstven nalog).
- Proširljivost – pošto se radi o jednoj širokoj oblasti aplikacija mora biti proširljiva radi kasnije nadogradnje.
- Performanse – potrebno je smanjiti vreme odziva i prilagoditi propusnu moć sistemu koji se realizuje.

## **Arhitekturni obrasci**

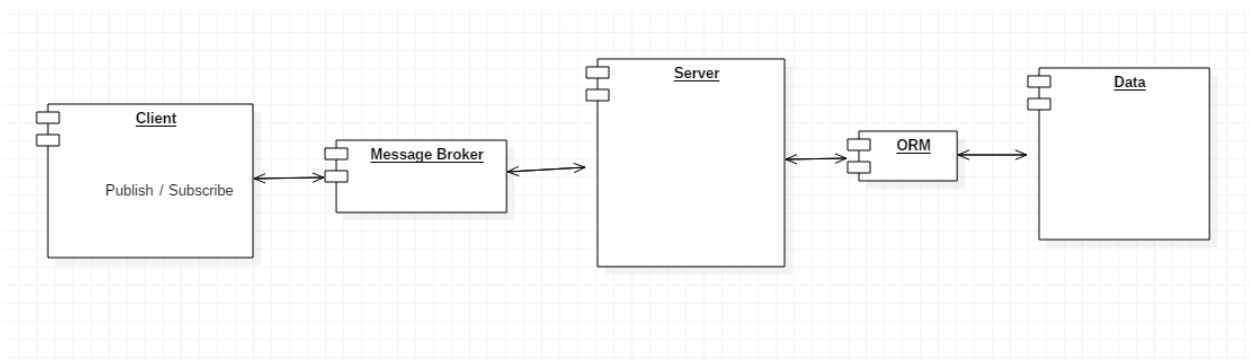
- Layered pattern  
Predstavlja osnovni obrazac i osnovu samog projekta. Aplikacija se sastoji iz tri sloja koje čine klijent, server i baza, između kojih postoje još dva međusloja broker između klijenta i servera i ORM alat između servera i baze.

- **Broker**  
Kako je aplikacija distribuirana pogodno je iskoristiti dati obrazac. Broker obrazac je implementiran u vidu middleware-a preko koga klijenti pozivaju i koriste servise servera sa kojim komuniciraju.
- **Publish-subscribe**  
Kako korisnik može da kreira i manipuliše različitim storijima i taskovima vezanim za projekat neophodno da svi ostali koji pripadaju istom projektu to mogu i da vide. Ovaj obrazac će biti implementiran kroz sam message broker koji predstavlja sponu izmedju klijenta i servera.

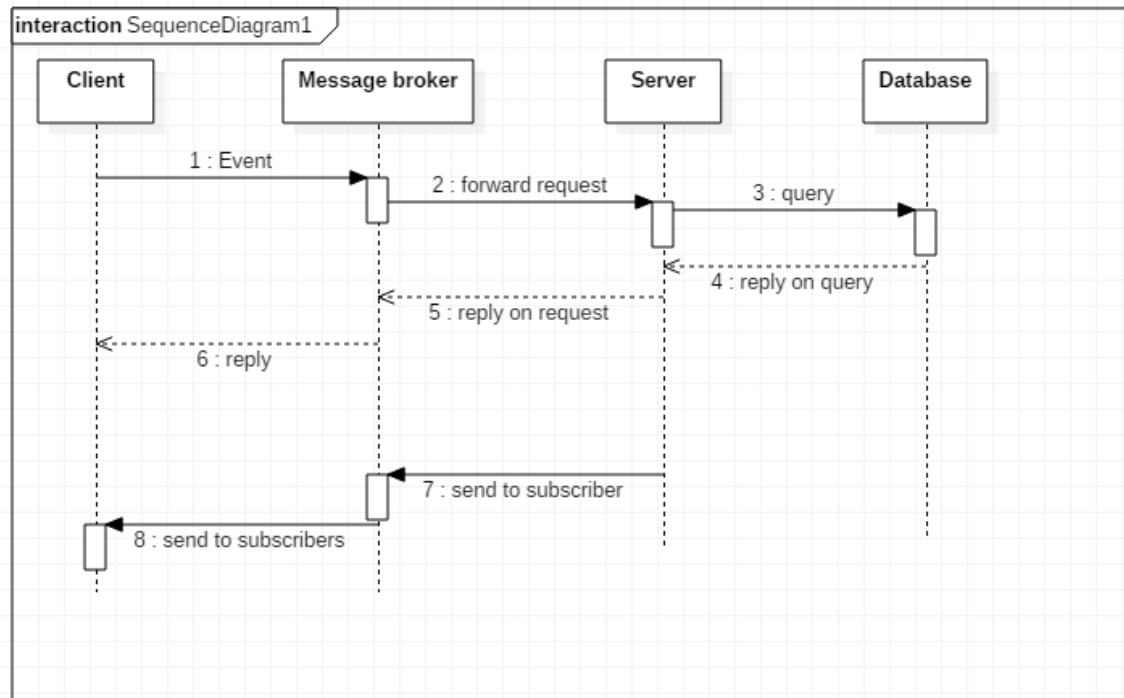
U izradi projekta će se koristiti i MVC .

## Arhitekturni dizajn

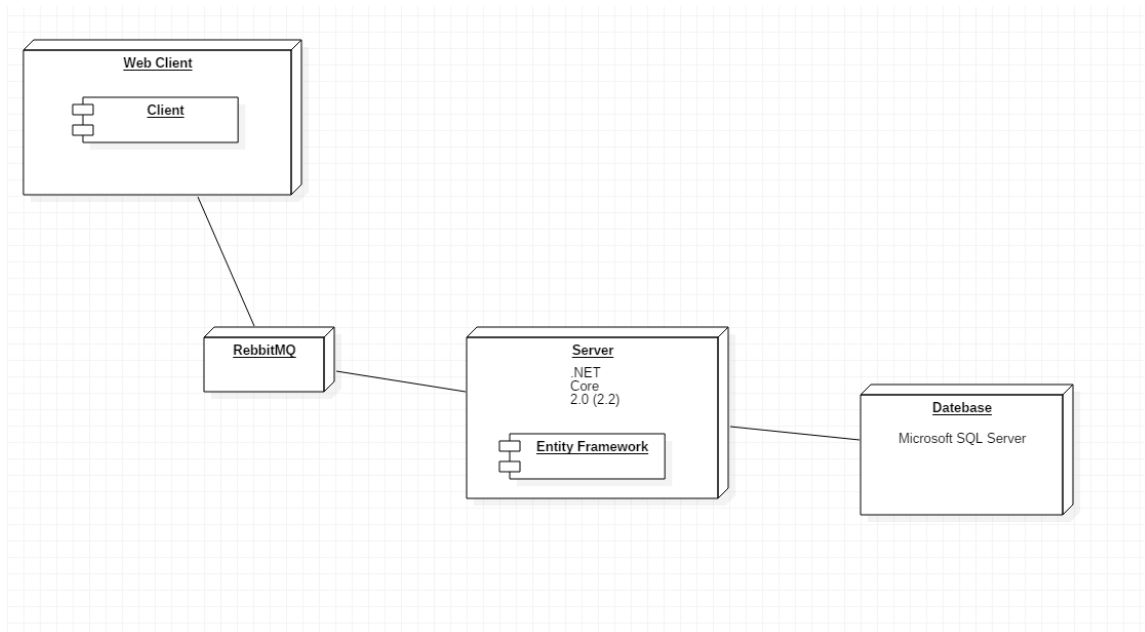
- **Strukturni pogled**



- Bihevioralni pogled



- Alokacioni pogled



- **Strukturni pogled**

Prikazuje strukturu sistema navodeći komponente sistema.

Klijentska strana ima unutrašnju strukturu zasnovanu na MVC projektnom obrascu.

Klijentska strana komunicira sa serverom preko Controller-a koristeći Message Broker.

Server preko ORM-a komunicira sa bazom podataka.

- **Bihevioralni pogled**

Bihevioralni pogled opisuje interakciju između komponenti sistema prilikom nastanka događaja. Prvo se upisuje promena u bazu podataka, a nakon toga se o tome obaveštavaju svi pretplaćeni korisnici (subscribers).

- **Alokacioni pogled**

Alokacioni pogled prikazuje raspored komponenti na čvorovima tj. na stvarnim računarima. Klijentska aplikacija se otvara u Web pretraživaču . Za komunikaciju sa serverom se koristi RabbitMQ server. Komunikacija između servera i baze se odvija pomoću mapiranja objekata putem FluentNHibernate-a.

## **Specifikacija biblioteka i programskih okvira**

- Angular 8 – za razvoj klijentske app
- MS SQL DBMS – za bazu podataka
- .NET Core 2.0 – za razvoj serverske app
- RabbitMQ – message broker
- Entity Framework – kao ORM