

Cilj softverskog projekta

Cilj softverskog projekta je izrada skalabilne aplikacije koja omogućava korisnicima da u realnom vremenu igraju igru baziranu na popularnom TV kvizu "TV Slagalice".

Zahtevi

Arhitekturno specifični zahtevi

Arhitekturno specifični zahtevi ovog projekta su:

- Performanse - mogućnost real-time igranja uz minimalnu latenciju
- Skalabilnost - podrška za load-balancing i dodavanje servera
- Dostupnost - serveri moraju da budu dostupni igračima u svakom trenutku
- Perzistencija - podaci o igračima se čuvaju trajno a podaci o trenutnoj sesiji privremeno, što omogućava oporavak od gubitka veze

Funkcionalni zahtevi

Glavni funkcionalni zahtevi ovog projekta su:

- korisnici mogu da kreiraju naloge i loguju se na njih
- prijavljeni korisnici (igrači) mogu da započinju igre i ulaze u "sobe"
- igrači tokom igre mogu da igraju dozvoljene poteze ili izađu iz igre

Nefunkcionalni zahtevi

Nefunkcionalni zahtevi ili atributi kvaliteta ovog projekta su:

- mogućnost oporavka od gubitka veze
- mali odziv prilikom promena
- provera validnosti poteza i podataka
- mogućnost jednostavnog dodavanja novih igara
- veći broj parova igrača (soba) istovremeno

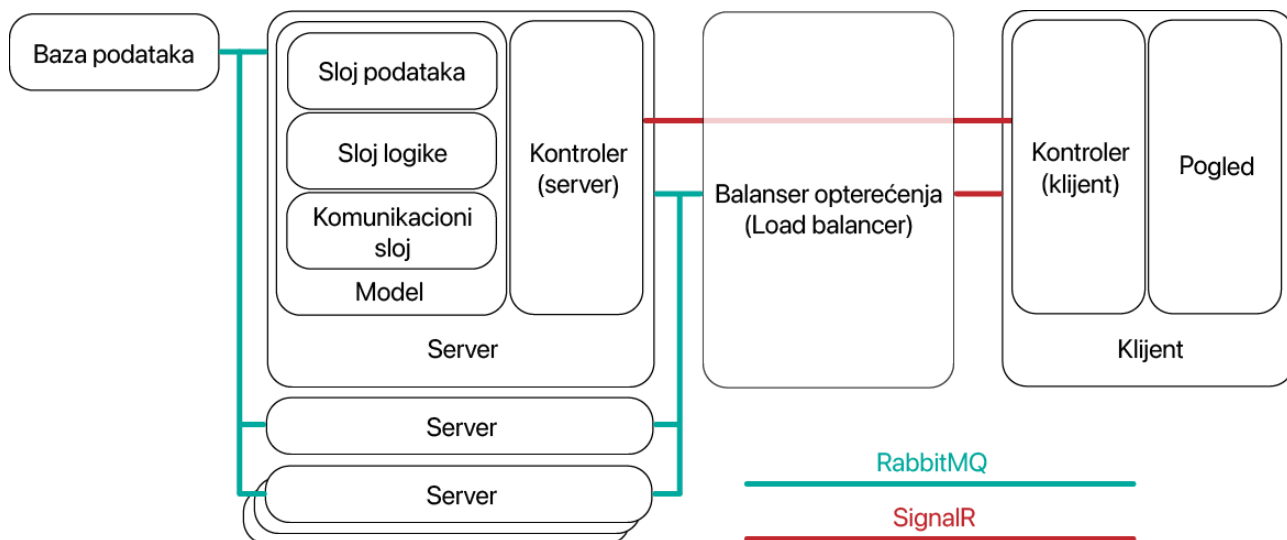
Arhitekturni dizajn

Arhitekturni obrasci

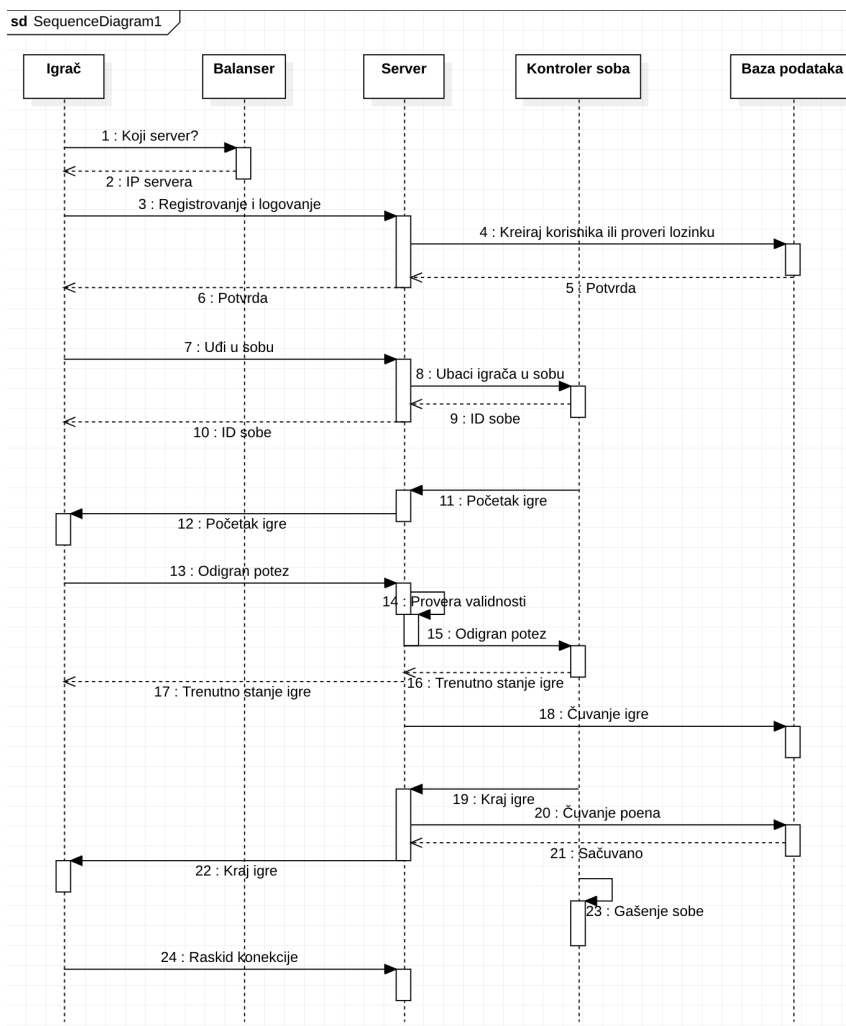
Arhitekturni obrasci koji će se naći u projektu su:

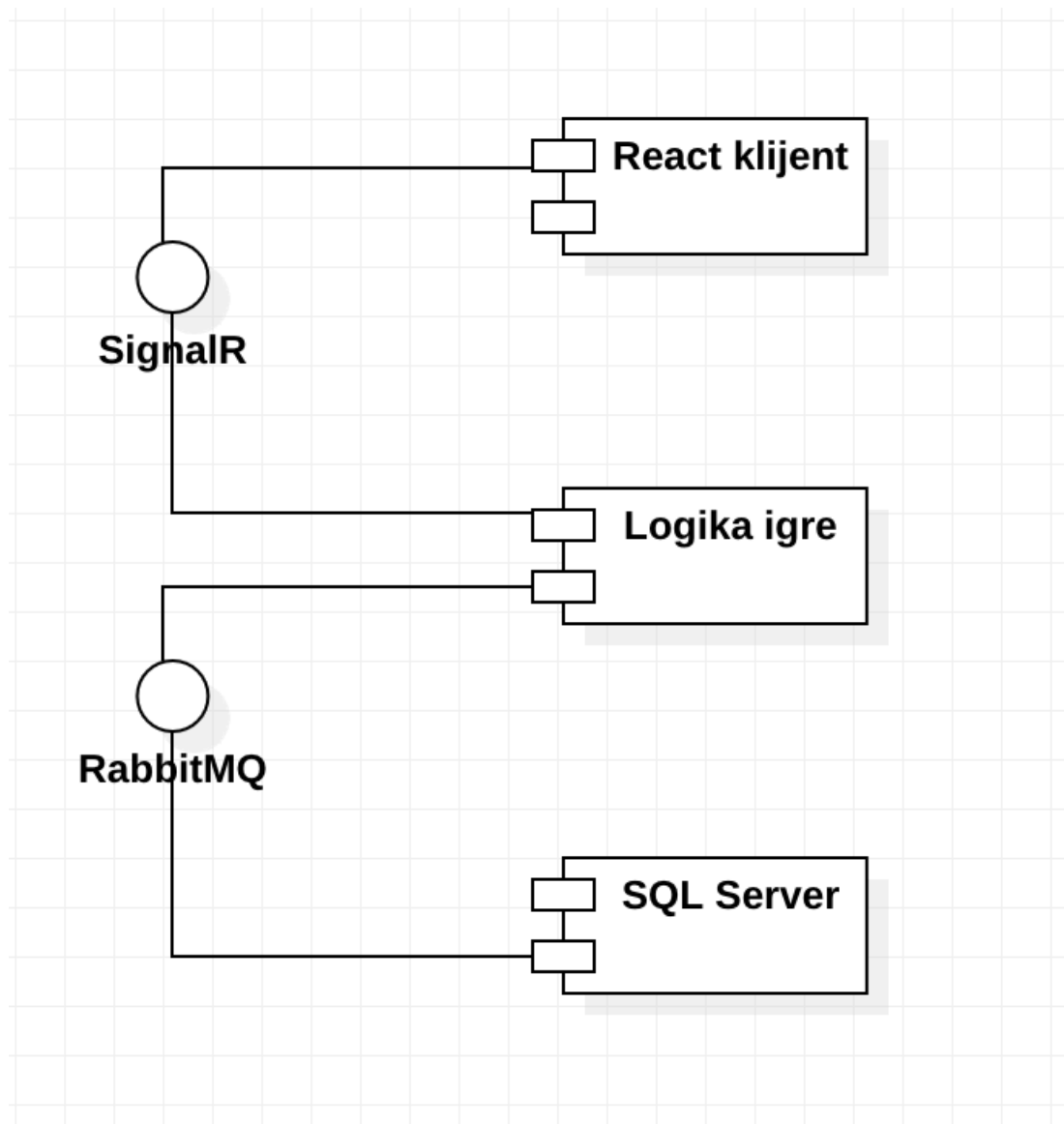
- MVC (Model-View-Controller) - model se nalazi na serverskom delu i obavlja funkcije iz domena aplikacione logike kao što su održavanje i kreiranje soba za igranje, konekcija sa igračima i komunikaciju sa bazom podataka; view je na klijentskom delu i bavi se korisničkim interfejsom; kontroler kao komponenta postoji i na klijentskom i na serverskom delu i posreduje između model i view komponenti.
- Layered - model deo MVC arhitekture je podeljen na slojeve odvojenih odgovornosti: data layer za pristup sistemu baze podataka, business logic layer za logiku igrice, communication layer koji čuva sobe i konekcije sa igračima.
- Broker - RabbitMQ kao broker ima ulogu posrednika između komponenti serverskih delova i omogućava odvajanje (decoupling).
- Client-Server - klijent se nalazi na uređaju igrača i komunicira sa serverom koji reguliše komunikaciju između klijenata, logiku igre i validaciju podataka.
- Publisher-Subscriber - SignalR hub na serveru ima ulogu publishera, dok su klijenti sa otvorenom konekcijom na hub-u subscriberi.

Generalna arhitektura



Strukturni i bihevioralni dijagrami





Alati i biblioteke

Serverski deo će biti implementiran u ASP.NET framework-u sa MVC template-om sa load balancerom realizovanim kao ASP.NET Web API i bazom podataka u formatu Microsoft SQL Servera. Klijentski deo će biti realizovan korišćenjem React biblioteke za JavaScript. Komunikacija između komponenti na serveru će se odvijati preko brokera RabbitMQ, a između klijenta i servera preko SignalR biblioteke. Kao ORM će biti korišćen .NET Core Entity Framework.