## **KNIGHTS&DIAMONDS**

# ARHITEKTURNI DIZAJN SOFTVERSKOG SISTEMA

## Dušan Spasić 16868

## Danica Antić 16473

## Kontekst i cilj projekta

Projekat predstavlja projektovanje i implementaciju multiplayer turn-based web aplikacije Knights&Diamonds koja prezentuje borbu između dva igrača korišćenjem više tipova karata. Svaki od igrača sastavlja špil karata koji se sastoji od karata stvorenja (creature card), magičnih karata (spell card) i karti zamki (trap card). Soba (borba) se kreira tako što jedan igrač izaziva drugog igrača. Svaki igrač na početku ima životne bodove (life points) i izvlači 5 karata iz svog špila. Potezi se igraju kombinacijom karata, ali korisnik može i da preskoči svoj potez. Pobednik je onaj igrač koji prvi uništi životne bodove protivnika.

## Arhitekturni zahtevi

## Glavni funkcionalni zahtevi

* **Prijava korisnika** –prijava postojećeg korisnika (username i password)
* **Registracija korisnika** – potrebno je omogućiti novim korisnicima registraciju u sistem
* **Kreiranje špila** –korisnik sastavlja špil karata od onih koje su mu dostupne
* **Izazivanje željenog protivnika** – svaki korisnik ima mogućnost da izazove na borbu protivnika koji je trenutno dostupan
* **Prihvatanje izazova** – korisnik koji je izazvan može prihvatiti ili odbiti borbu
* **Odigravanje poteza** – na početku svakog poteza izvlači se nova karta, zatim se bira jedna od dostupnih mogućnosti u zavisnosti od trenutnog stanja igre
* **Kraj igre** – pobeđuje igrač koji smanji protivnikove poene (life points) na 0
* **Komunikacija između igrača**

## Ne-funkcionalni zahtevi (atributi kvaliteta)

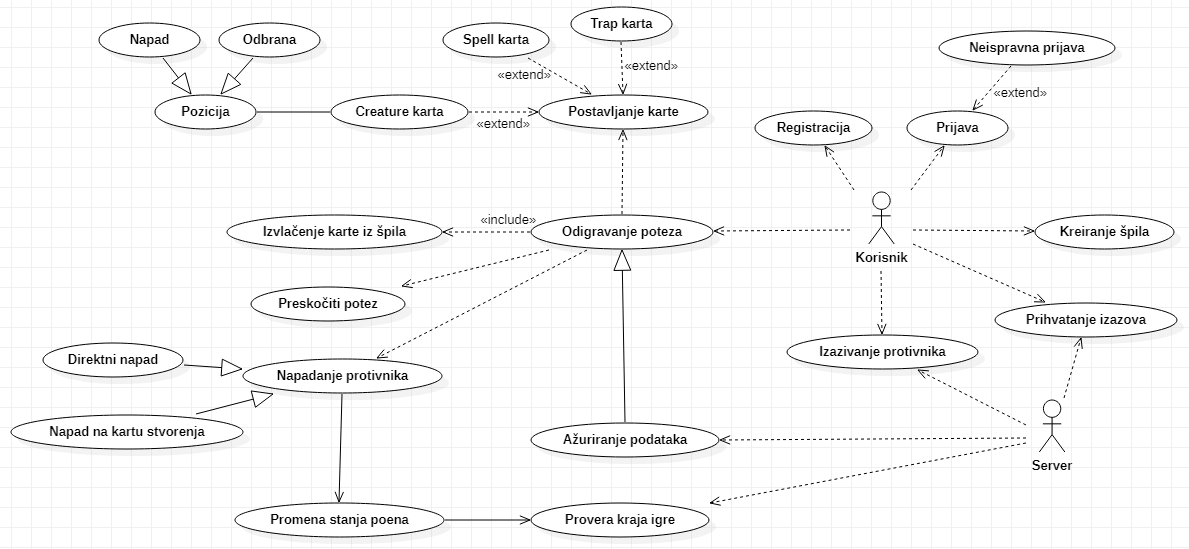
* **Performanse** – potrebno je da propusna moć bude što veća, a vreme odziva što kraće
* **Pouzdanost** – nivo performansi treba da ostane isti bez obzira na okruženje korisnika, potrebno je obezbediti toleranciju na greške i konzistentnost podataka
* **Skalabilnost** –sa povećanjem broja zahteva, propusna moć ostaje konstantna
* **Modifikabilnost** – prilikom promene i evoluiranja aplikacije, funkcionalni zahtevi ostaju zadovoljeni
* **Upotrebljivost** – potrebno je da aplikacija bude intuitivna i jednostavna za korišćenje
* **Sigurnost** – osigurati autentifikaciju i autorizaciju korisnika

## Tehnička i poslovna ograničenja

Tehnička ograničenja:

* Optimizacija za web čitače – Google Chrome, Mozilla Firefox, Microsoft Edge, Opera
* Notifikacija događaja – obaveštenje kada korisnik preduzme specifičnu akciju, npr. korisnik želi da odigra trap kartu
* Apstrakcija podataka – važno je da interna organizacija baze podataka (šema) bude sakrivena od strane API-a

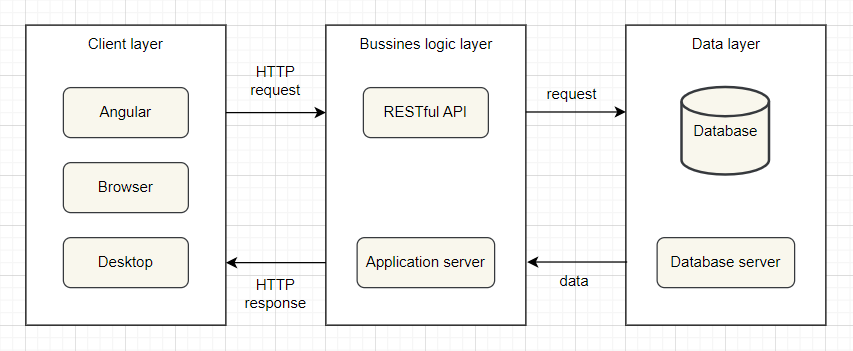
## Arhitekturni dizajn



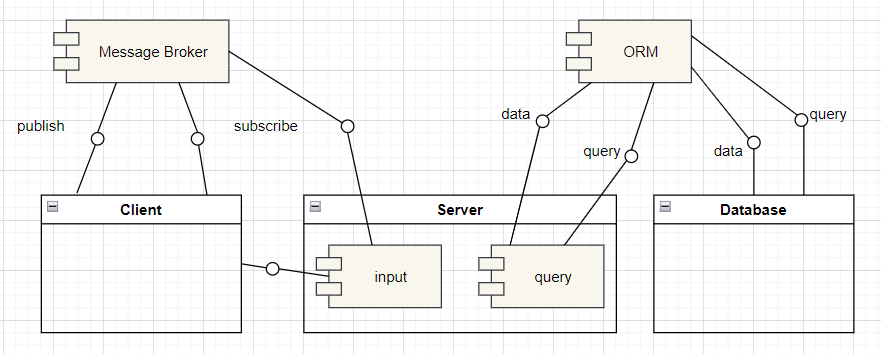
## Arhitekturni obrasci

* **Layerd arhitektura** – troslojna (klijent, server, baza podataka)
* **MVC** – **V**iew komponentu će predstavljati Angular framework. REST API na back end-u će predstavljati **M**odel i **C**ontroller i biće zadužen za obezbeđivanje podataka za View komponente
* **Publish-Subscribe** – sloj poslovne logike poseduje publish-subscribe mogućnosti kako bi omogućio distribuirano izvršavanje aplikacije; biće iskorišćen za prosleđivanje događaja tokom igre koji uključuju odigravanje poteza
* **Repository** – služiće za kreiranje, upravljanje i održavanje kompleksnog i centralizovanog skladišta podataka

## Generalna arhitektura

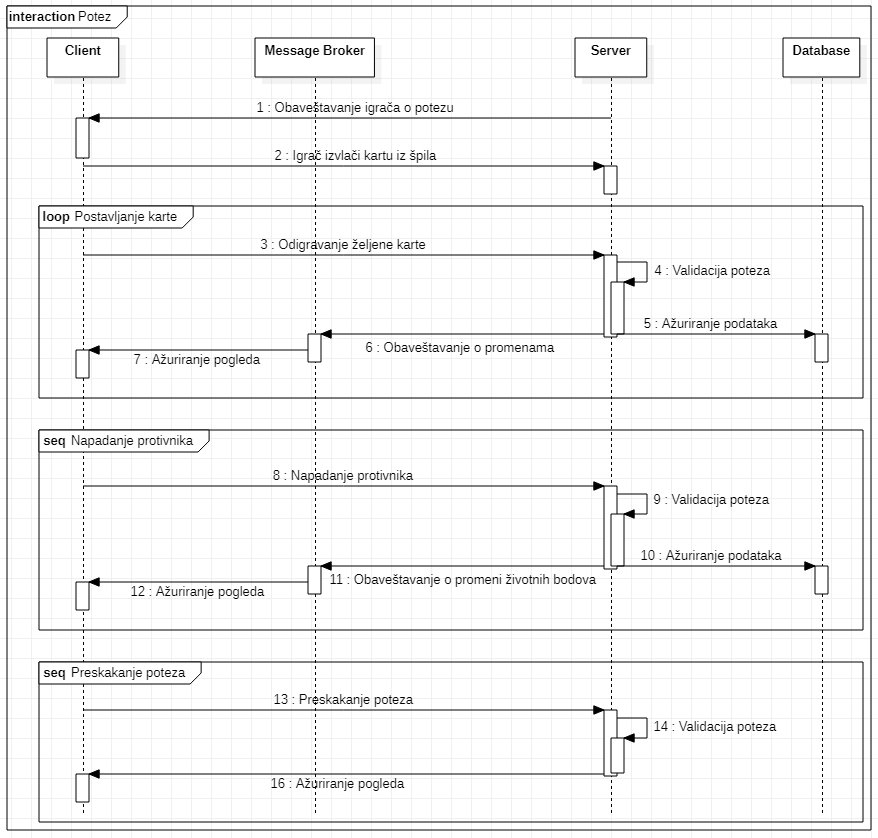


## Strukturni pogledi

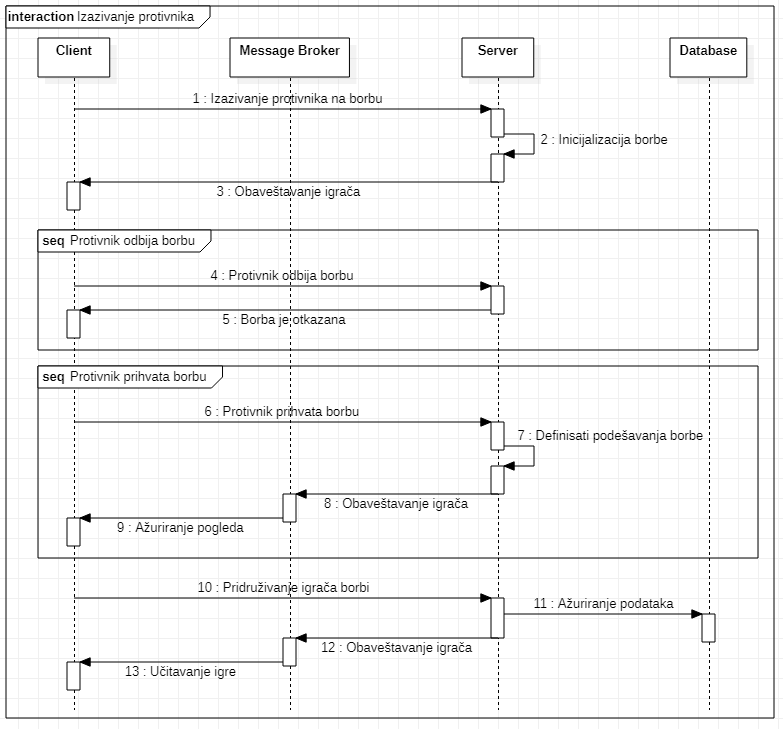


## Bihevioralni pogledi

* Sekvencijalni dijagram – Potez



* Sekvencijalni dijagram – Izazivanje protivnika



## Implementaciona pitanja

* Angular – JavaScript frontend framework
* ASP .NET Core Framework – Serverska aplikacija
* Microsoft EntityFramework Core – ORM
* MS SQL – Baza podataka
* SignalR – Message Broker

## Analiza arhitekture

## Potencijalni rizici u implementaciji i strategije prevazilaženja

Potencijalni rizik u implementaciji Knights&Diamonds sistema je problem kapaciteta servera za veliki broj korisnika. Strategija za prevazilaženje ovog rizika može biti testiranje performansi i opterećenje servera.