

# **Texas Holdem Poker**

*Arhitektura i projektovanje softvera*

# Sadržaj

Opis igre .....	3
Funkcionalni zahtevi .....	4
Nefunkcionalni zahtevi .....	4
Arhitekturni zahtevi .....	4
Arhitekturni dizajn .....	5
Arhitekturni obrasci .....	7
Specifikacija biblioteka i programskih okvira .....	7

# OPIS IGRE

Texas Holdem Poker je igra kartama za od 2 do 9 igrača. Koristi se klasičan špil od 52 karte . Cilj igre je imati najbolje karte ili nadmudriti protivnike sa povišavanjem uloga koji oni ne žele pratiti. Igra ide u smeru kazaljke na satu .Igrači mogu da biraju koji je njihov potez: da li pusta igru dalje, povišuje ulog,odustaje ili prati povišen ulog (ako je povišen). Igra ima 5 faza . Prva faza: Svaki igrač stavlja početni ulog i dobija 2 karte koje samo oni vide. Igrači biraju njihov potez. Druga faza: Delilac stavlja tri karte na sto. Igrači biraju njihov potez. Treća faza: Delilac stavlja jednu karte na sto . Igrači biraju njihov potez.Četvrta faza: Delilac stavlja jednu karte na sto . Igrači biraju njihov potez. Peta faza: Upoređuju se jačine karata najbolje 5 od 7 . Igrač koji ima najbolje karte on je pobednik i uzima ulog. Igra može biti završena i ranije ako je ostao samo jedan igrač. Kada se završi igra karte se mešaju i dele opet. Igra je konačno završena kada ostane samo 1 ili 2 igrača za sto.

Jačina karata je zadata unapred od najbolje do najgore:

Ime jačine - karte koje su potrebne za tu jačinu

Royal flush - 10,A,J,Q,K u istom znaku

Straight flush – 5 karte u nizu u istom znaku

Poker – 4 karte istog broja

Full house – 3 karte jednog broja plus 2 karte drugog broja

Flush – 5 karte u istom znaku

Straight – 5 karte u nizu

Three of a kind – 3 karte istog broja

Two pair – 2 karte jednog broja plus 2 karte drugog broja

One pair - 2 karte jednog broja

High card – Kada se ne ispunjava nijedan uslov gore naveden onda se gleda samo najveća karta

## **Funkcionalni zahtevi**

- Povezivanje klijenta i servera
- Kreiranje korisničkih naloga
- Prijavljivanje uz pomoć već kreiranih naloga
- Server omogućava kreiranje novog stola ako su svi postojeći puni
- Biranje stola na koji korisnik želi da igra
- Server omogućava početak igre
- Server vodi računa o toku igre , potezima igrača i ishodima poteza
- Vođenje evidencije i skladištenje podataka

## **Nefunkcionalni zahtevi**

- Pouzdanost
- Skalabilnost - potrebno je da aplikacija može da podrži veliki broj igrača
- Performanse - potrebno je smanjiti vreme odziva
- Održljivost - potrebno je da aplikacija bude laka za odžavanje
- Modifikabilnost - potrebno je omogućiti laku promenu i nadogradnju sistema
- Dostupnost - potrebno je da aplikacija bude dostupna 24/7
- Lakoća korišćenja - potrebno je da aplikacija bude jednostavna za korišćenje

## **Arhitekturni zahtevi**

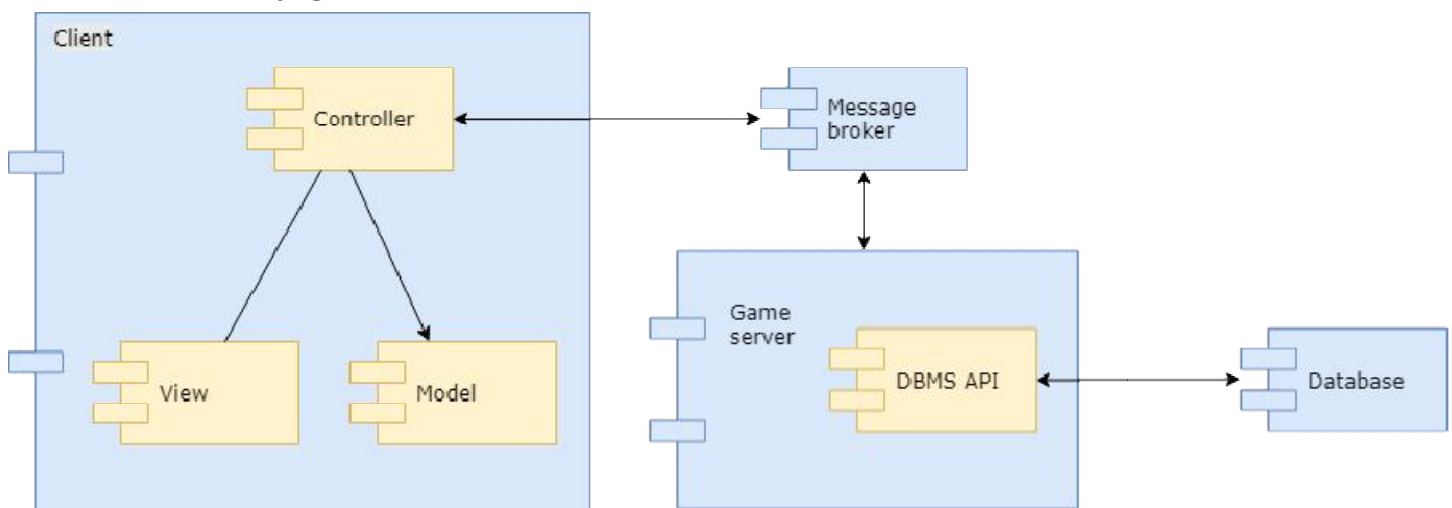
- Korišćenje relacione baze podataka
- Sistem će biti dostupan 24 / 7
- Skrivena šema baze podataka

# Arhitekturni dizajn

## Strukturni pogled

Osnovne komponente sistema su: Klijent (model, view, controller), MessageBroker, Server i Baza podataka. Klijent je baziran na MVC strukturi. Klijent komunicira sa serverom preko controller-a koristeći Message Broker. Na serveru se nalazi API koji komunicira sa MySQL bazom podataka.

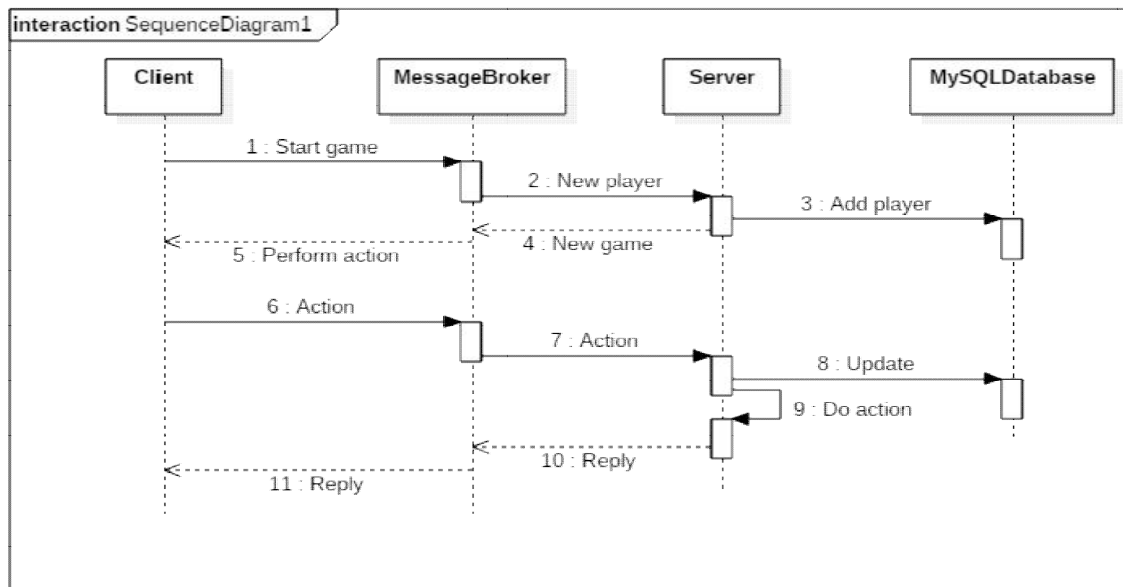
## Strukturni dijagram



## Bihevioralni pogled

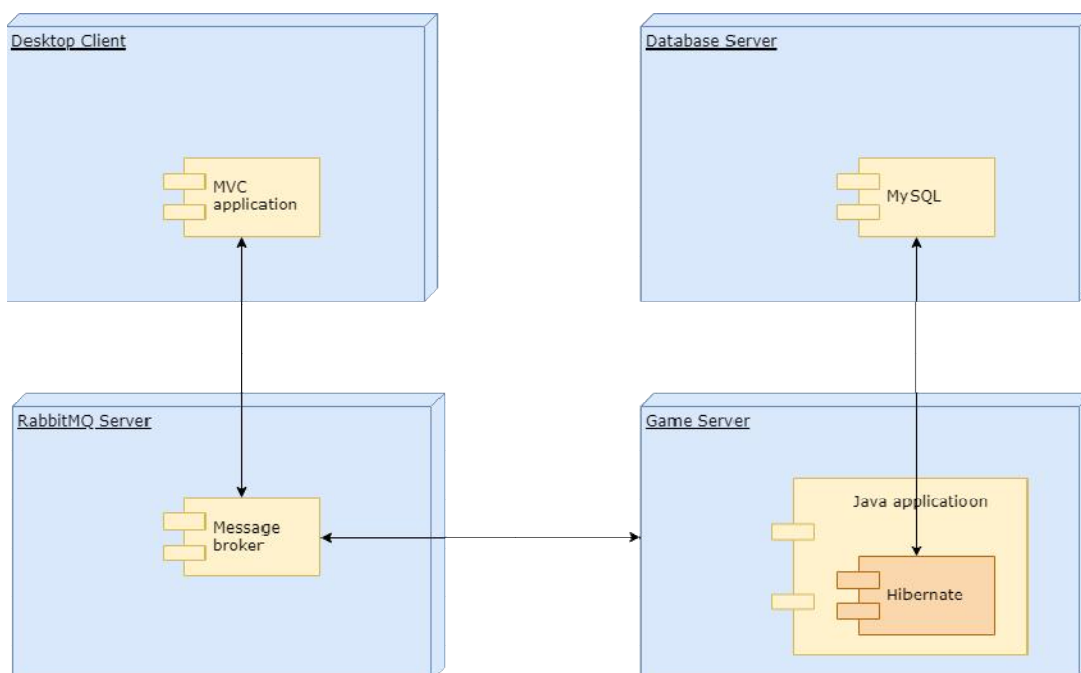
Bihevioralni pogled opisuje interakciju između komponenti sistema prilikom povezivanja novog igrača na server. Igra automatski počinje kada bude minimalno 2 igrača za sto. Igra se odvija kroz niz faza koje se smenjuju ciklično.

## Sekvencijalni dijagram



## Alokacioni pogled

Navedeni dijagram ilustruje alokacioni pogled na raspored opisanih komponenti na računarima. Klijentska aplikacija je implementirana kao MVC Java aplikacija. Serverska aplikacija se realizuje kao Java aplikacija (Spring framework-RESTful Web Service). Kao baza podataka se koristi MySQL. Za komunikaciju između klijenta i servera koristi se RabbitMQ message broker, a kao ORM se koristi Hibernate.



# Arhitekturni obrasci

- **MVC(Model-View-Controller)**
- **Layered-** 3 sloja, prvi klijentska aplikacija, drugi broker i serverska aplikacija a treci baza podataka.
- **Publish-Subscriber**-komunikacija između klijenta i servera, ovaj obrazac je sadržan u Message Brokeru. (Modifikacija) - Funkcija Notify obaveštava samo klijente koji su za istim stolom.
- **Repository**-komunikacija sa bazom

## Specifikacija biblioteka i programskih okvira

Spring framework

RabbitMQ message broker server

Hibernate ORM alat

MySQL DBMS server baze podataka