**Arhitekturni dizajn softverskog sistema**

**4InRoW**

Aleksandra Stanojević 15386

Jovana Stamenković 15372

4InRow je interaktivna igra između dva igrača. Data je tabla sa mestima za kuglice. Svaki igrač ima svoju boju i vertikalno ubacuje kuglicu te boje na najniže mesto u određenom redu. Pobednik je prvi igrač koji poređa 4 kuglice u jednom redu.

**Funkcionalni zahtevi**

1. Povezivanje klijenta i servera.
2. Klijentu je omogućeno odigravanje nove partije.
3. Server spaja dva igrača i omogućava odigravanje nove igre.
4. Server vodi računa o potezima igrača i rezultatu.
5. Igrač/klijent je obavešten o tome koji je rezultat i ko je na potezu.
6. Logika za implementaciju igre.
7. Vođenje evidencije i skladištenje podataka.

**Arhitekturni zahtevi**

1. Potrebno je omogućiti pamćenje velike količine podataka.
2. Šema baze podataka je skrivena.
3. Sistem će biti dostupan 7 dana nedeljno, 24 sata dnevno.

**Nefunkcionalni zahtevi**

Pouzdanost

Performanse – potrebno je smanjiti vreme odziva i prilagoditi propusnu moć sistemu koji se realizuje.

Upotrebljivost

Dostupnost

Modifikabilnost – laka promena sistema

Skalabilnost – kako će se aplikacija ponašati kada se poveća broj zahteva.

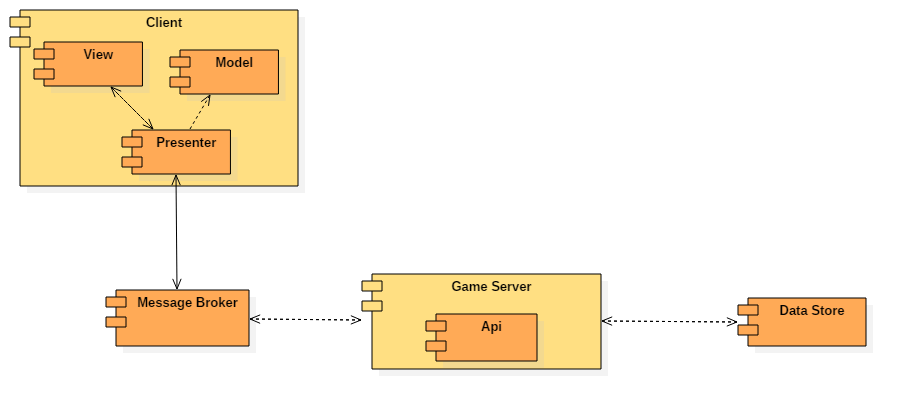
**Arhitekturni dizajn**

**Strukturni pogled**

Prikazuje strukturu sistema navodeći komponente sistema.

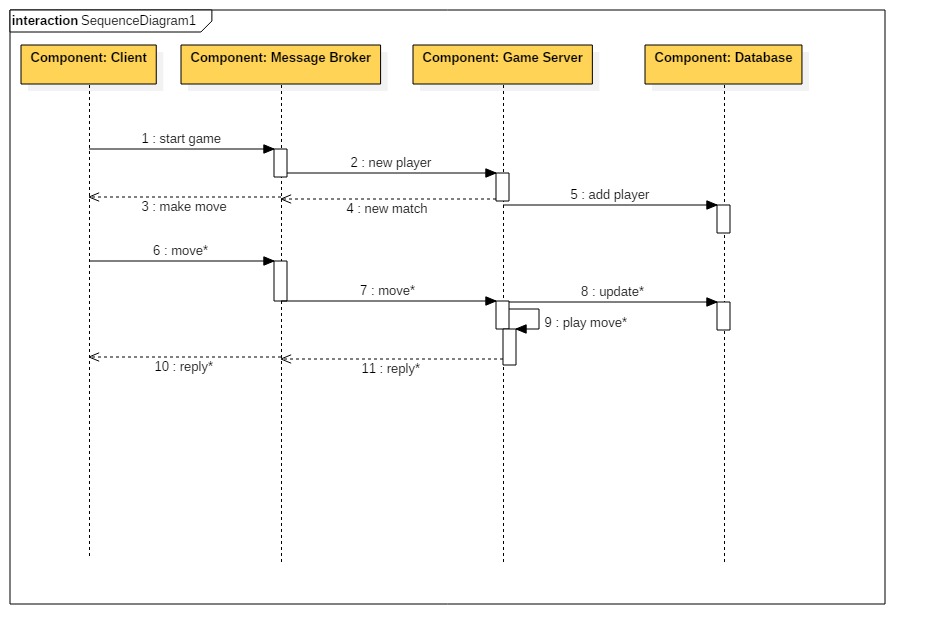
Klijent ima unutrašnju stukturu zasnovanu na MVP projaktnom obrascu.

Klijent komicira sa serverom preko Presenter-a koristeći Message Broker. Na game serveru se izvršava Api koji komunicira sa bazom podataka.

****

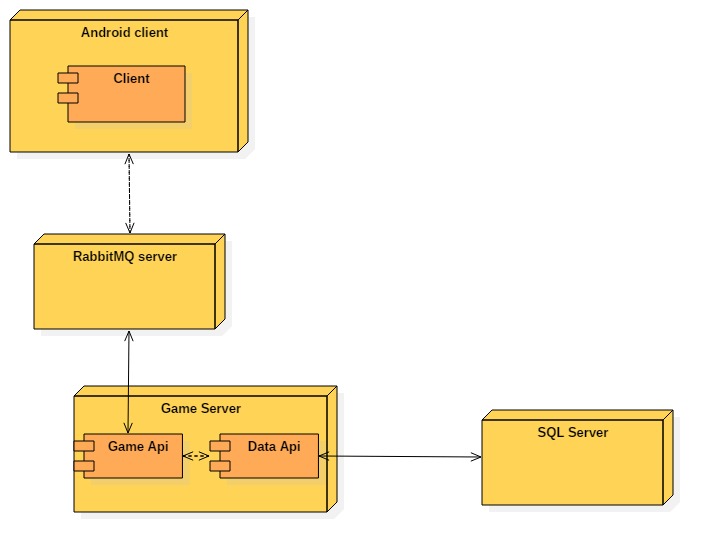
**Bihevioralni pogled**

Bihevioralni pogled opisuje interakciju između komponenti sistema prilikom povezivanja novog igrača na server. Prvo se povezuju dva igrača u jedan meč, zatim se repetitvno odigrava niz poteza tako što se dodelje privilegija poteza naizmenično igračima.

****

**Alokacioni pogled**

Alokacioni pogled prikazuje raspored komponenti na čvorovima tj. na stvarnim računarima. Client se nalazi na android uređaju. Za komunikaciju sa serverom se koristi RabbitMQ server.

****

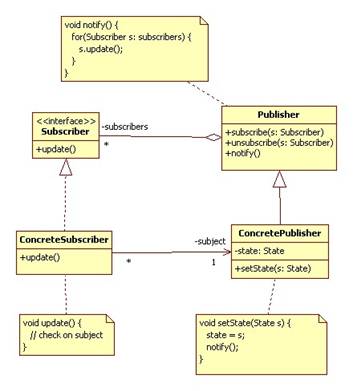
**Arhitekturni obrasci**

**1.MVP(Model-View- Preseter)** - modifikacija MVC gde ne komuniciraju direktno View i Model (u android aplikaciji ).



**2.Publish-Subscriber** - za komunikaciju izmedju klijenta i server; ovaj obrazac je sadržan u Message Broker-u.

Modifikacija u odnosu na standardni obrazac: funkcija Notify ne obaveštava sve klijente, već samo one koji učestvuju u određenoj partiji (2 igrača).



**3.Singleton -** kreiranje konekcije.

**4. MVC** – struktura Web Api –ja .

**Specifikacija biblioteka i programskih okvira**

**Android studio framework** -izrada klijentske aplikacije.

**ASP.net MVC framework** - framework za izradu WEB API-ja.

**RabbitMQ.NET** - biblioteka za Message Broker.

**Nhybernate** -ORM mapiranje.

**Java AMPQ library** -biblioteka za povezivanje na Message Broker, kristi se u android aplikaciji.

**RabbitMQ** -message broker server.

**Oracle DBMS** - server baze podataka.