

Arhitektura i projektovanje softvera



DrawOut

Faza I - Arhitekturni projekat

Pavle Antonijević 18077

Dušan Petrović 18329

Sadržaj

Sadržaj.....	1
Kontekst i cilj projekta.....	2
Kontekst projekta:.....	2
Ciljevi projekta:.....	2
Arhitekturni zahtevi.....	3
Arhitekturno specifični zahtevi.....	3
Ne-funkcionalni zahtevi.....	4
Tehnička i poslovna ograničenja.....	5
Arhitekturni dizajn.....	6
Arhitekturni obrasci.....	7
Generalna arhitektura.....	8
Strukturni pogled.....	9
Bihevioralni pogled.....	10
Implementacija.....	11

Kontekst i cilj projekta

Kontekst projekta:

Projekat "DrawOut" je realtime višekorisnička web igra dizajnirana kao interaktivna platforma za zabavu i druženje. Ova igra omogućava korisnicima da u timovima učestvuju u crtanju i pogađanju reči/pojmova, nudeći jedinstveno iskustvo koje kombinuje kreativnost, brzo razmišljanje i timski rad.

Aplikacija funkcioniše tako što jedan korisnik može da:

1. se priključi već postojećoj sobi (preko liste soba, linka ili nasumično) ili
2. kreira svoju sobu (i time dobija ulogu - administrator sobe/room admin)

Nakon što korisnik uspešno napravi sobu ili se pridruži nekoj od već postojećih, njegov prvi korak je izbor tima - crvenog ili plavog. Ovaj izbor je moguć pod uslovom da odabrani tim ima slobodnih mesta. Korisnik može preuzeti ulogu vođe tima ili, ako već ima tu poziciju, može prepustiti vođstvo nekom drugom članu tima.

Administrator sobe ima mogućnost da pokrene partiju. Svaka partija uključuje ukupno osam rundi, a trajanje svake runde određuje administrator sobe birajući jednu od ponuđenih opcija.

Timovi smenjuju naizmenično po rundama, pri čemu plavi tim uvek počinje prvi.

Redosled plavi/crveni tim se održava u svakoj rundi. U svakoj rundi, jedan član tima koji do sada nije crtao biva nasumično izabran da u ograničenom vremenu nacрта zadata reč/pojam. Za to isto vreme, ostali članovi njegovog tima pokušavaju da preko timskog četa pogode šta je nacrtano. Što brže pogode, više poena dobijaju. Međutim, ukoliko ne pogode zadata reč/pojam, protivnički tim ima mogućnost da u određenom vremenu "ukradu" poene. U tom slučaju, samo vođa protivničkog tima može da pogađa reč/pojam, dok mu ostatak tima daje sugestije preko timskog četa.

Ekipa koja nakon osam rundi, odnosno završetka partije, ima više poena proglašava se pobednikom.

Ciljevi projekta:

Interaktivnost i zabava: Glavni cilj je kreiranje platforme koja će korisnicima omogućiti zabavno i angažujuće iskustvo kroz igru crtanja i pogađanja.

Timski rad: Podsticanje korisnika na saradnju i timski rad, gde članovi tima zajedno rade na pogađanju reči na osnovu crteža.

Edukacija i kreativnost: Omogućavanje korisnicima da razvijaju svoje kreativne veštine i uče nove reči ili pojmove kroz igru.

Očekivani rezultati: Razvoj igre koja će privući širok spektar korisnika različitih uzrasta i interesovanja. Stvaranje platforme koja omogućava lako povezivanje i interakciju između igrača.

Potencijal za rast i razvoj: Postepeno dodavanje novih funkcionalnosti i reči/tema za crtanje. Mogućnost integracije sa društvenim mrežama radi deljenja crteža i rezultata. Prilika za organizovanje online takmičenja i događaja unutar igre.

Arhitekturni zahtevi

U ovom odeljku ćemo definisati arhitekturno specifične zahteve, glavne funkcionalne i ne-funkcionalne zahteve koje treba zadovoljiti, kao i tehnička i poslovna ograničenja koja se odnose na realizaciju web aplikacije “DrawOut”.

Arhitekturno specifični zahtevi

Funkcionalni zahtevi web aplikacije “DrawOut”:

- ❖ Upravljanje korisničkim profilom
 - Kreiranje/editovanje korisničkog imena
 - Odabir ikonice iz liste ponuđenih ikonica
- ❖ Kreiranje/podešavanje sobe
 - Zadavanje imena sobe
 - Opcija zaključavanje sobe korisnički generisanom lozinkom
 - Odabir rečnika za generisanje reči iz liste dostupnih rečnika
 - Opcija dodavanja prilagođenih (custom) reči, maksimalno 20
 - Opcija podešavanja trajanja pojedinačne runde
 - Mogućnost deljenja direkt linka sobe
 - Opcija slučajne raspodele igrača po timovima
- ❖ Administracija sobe (uloga - room admin)
 - Mogućnost prekida igre/sesije
 - Mogućnost (od)pauziranja igre
 - Mogućnost uklanjanja igrača iz sobe
 - Mogućnost startovanja igre
- ❖ Pridruživanje sobi i timu
 - Opcija za pridruživanje igri u bilo kom trenutku
 - Odabir tima prilikom pridruživanja, maksimum 4 igrača po timu
 - Mogućnost promene tima
 - Mogućnost prepuštanja uloge “vođa tima” (uloga - team leader)
 - Mogućnost napuštanja sobe
- ❖ Interakcija i upravljanje korisnika tokom igre
 - Nasumičan redosled igrača unutar tima za crtanje
 - Crtanje i odabir alata za crtanje

- Pogađanje reči/pojma
- Osvajanje poena pogađanjem reči/pojma
- Komunikacija preko timskog četa
- Pregled crteža u realnom vremenu (sopstvenog i protivničkog tima)
- Mogućnost “krađe” poena u slučaju da protivnički tim ne uspe da pogodi zadatu reč/pojam (uloga - team leader)

Ne-funkcionalni zahtevi

U procesu projektovanja sistema, fokus se stavlja na ispunjavanje sledećih atributa kvaliteta sistema, odnosno ne-funkcionalnih zahteva:

❖ Performanse:

- Brzo učitavanje i odzivnost igre, minimalno kašnjenje u komunikaciji između klijenta i servera.
- Efikasna obrada podataka u realnom vremenu, posebno tokom crtanja i pogađanja.

❖ Skalabilnost:

- Sposobnost sistema da podrži veliki broj korisnika i soba istovremeno, bez smanjenja performansi.
- Mogućnost proširenja kapaciteta servera kako bi se zadovoljile rastuće potrebe igre.

❖ Pouzdanost:

- Visoka dostupnost igre, sa minimalnim vremenom prekida.
- Robustno upravljanje greškama i očuvanje stanja igre u slučaju prekida.

❖ Sigurnost:

- Zaštita privatnosti korisnika, bezbedna komunikacija i čuvanje korisničkih podataka.
- Zaštita od neautorizovanog pristupa i manipulacije u igri.

❖ Korisničko iskustvo:

- Intuitivan i pristupačan korisnički interfejs.
- Jasno definisane instrukcije i pravila igre.
- Prilagođenost različitim uređajima i ekranima.

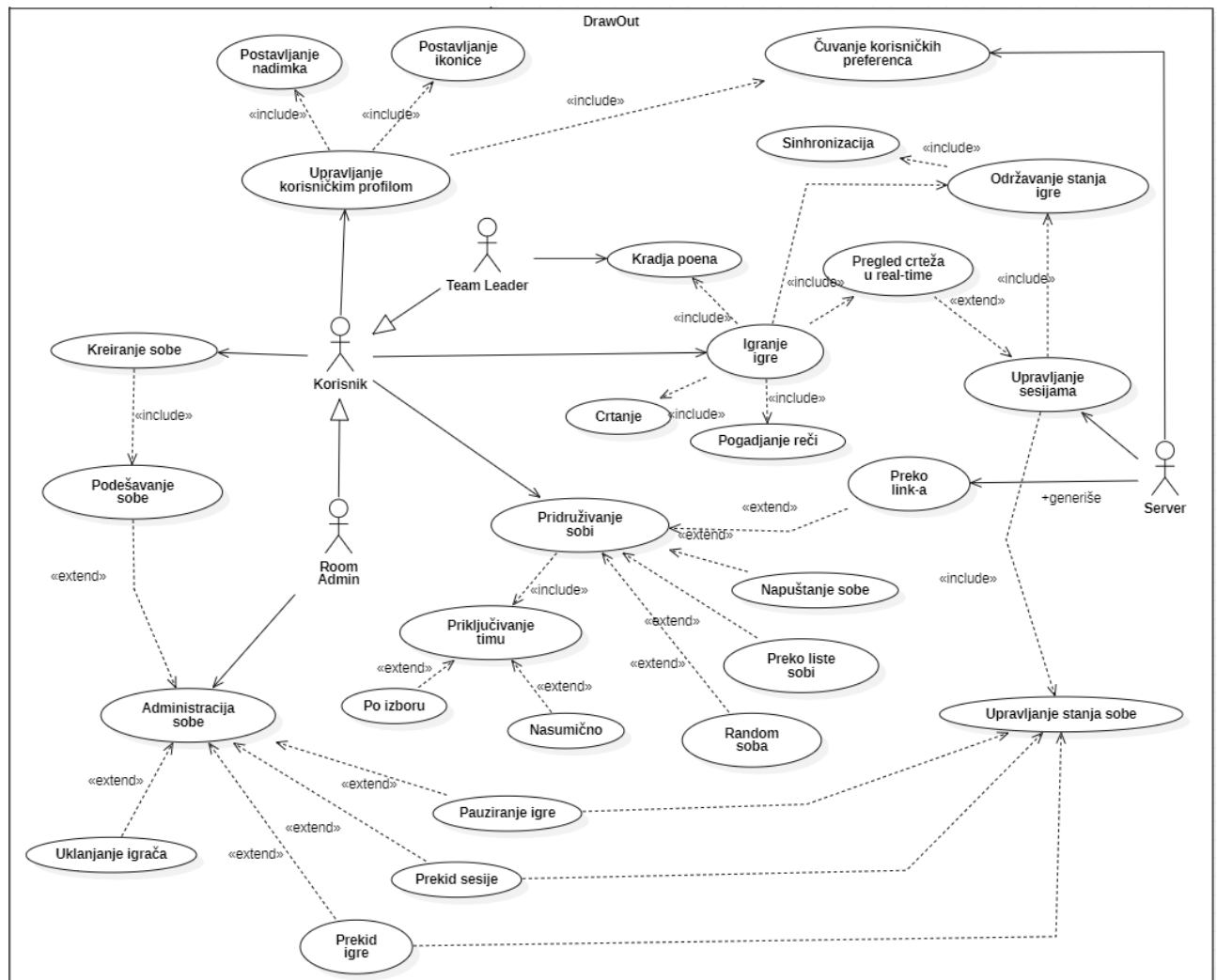
Tehnička i poslovna ograničenja

Tehnička ograničenja u okviru projekta su:

- ❖ **Apstrakcija i prikrivenost baze podataka** - Ključno je da način na koji su podaci organizovani unutar baze podataka ostane sakriven od strane API-a. Ovo osigurava da korisnici imaju pristup samo onim podacima koji su namenjeni za prikaz, dok je unutrašnja struktura i međusobna povezanost tih podataka skrivena
- ❖ **Pristup i optimizacija za web** - Ovo uključuje specifičnu optimizaciju za popularne web pretraživače kao što su Google Chrome, Mozilla Firefox, Microsoft Edge i Opera, omogućavajući korisnicima nesmetan pristup i interakciju sa aplikacijom bez obzira na izabrani web pretraživač

Arhitekturni dizajn

U ovom delu dokumenta ćemo se detaljno posvetiti opisivanju arhitekturnih obrazaca, korišćenih u projektovanju arhitekture web aplikacije “DrawOut”. Takođe, opisaćemo generalnu arhitekturu, kao i strukturni i biheviornalni pogled (dijagram) na aplikaciju.



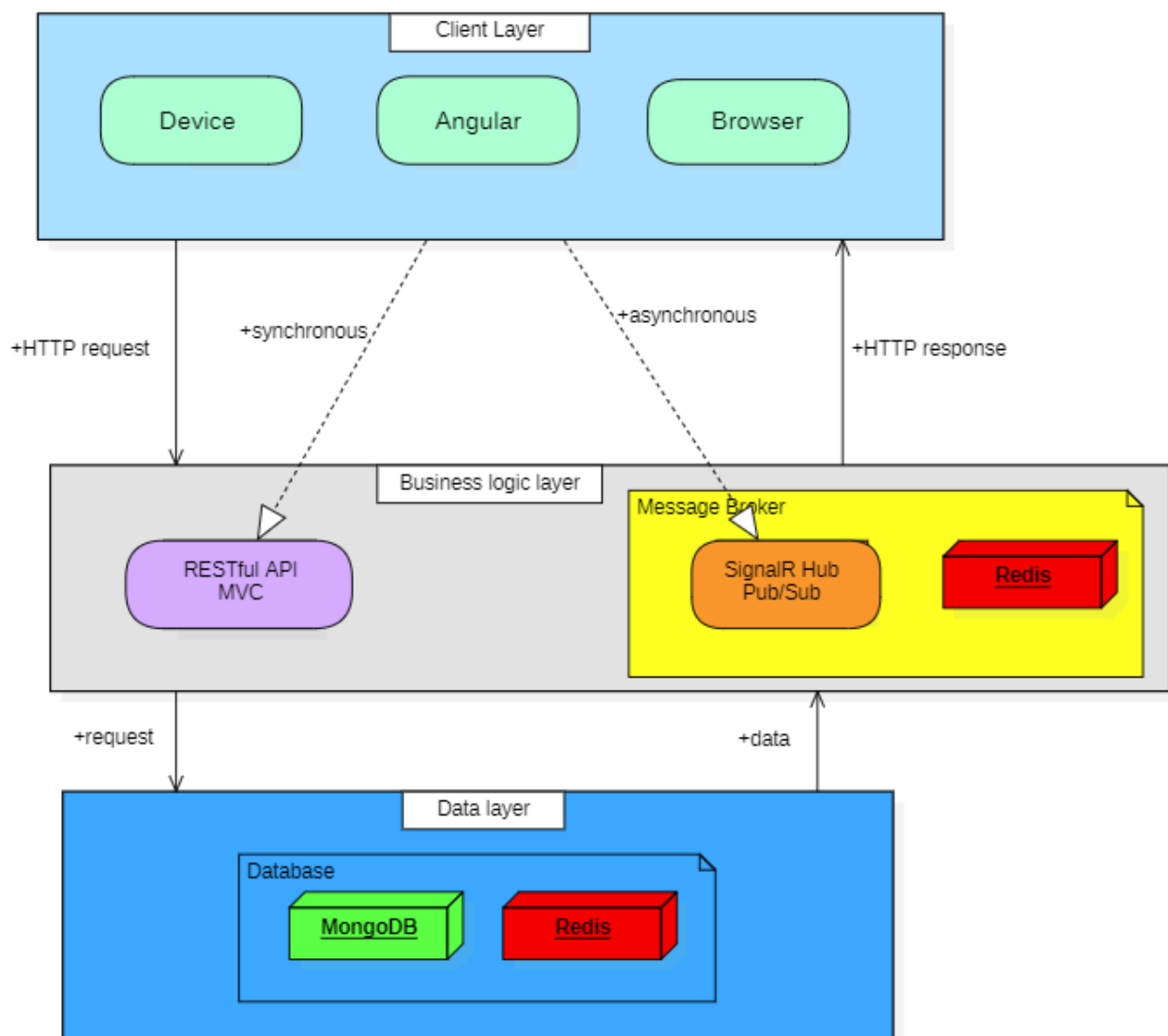
Use Case diagram osnovnih funkcionalnosti

Arhitekturni obrasci

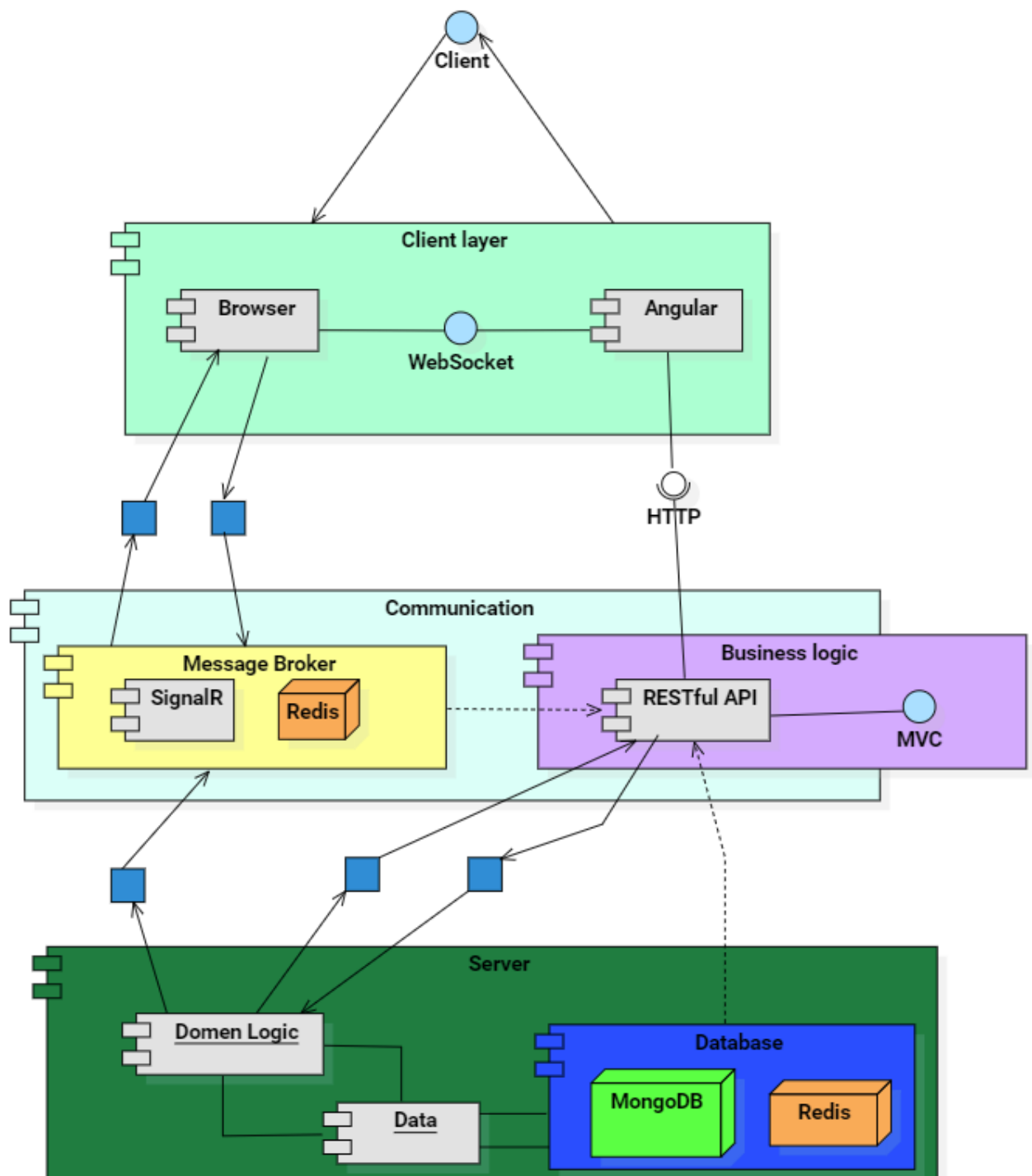
U nastavku će biti predstavljeni arhitekturni obrasci koji će biti korišćeni u toku realizacije "DrawOut" web aplikacije:

- ❖ **Klijent/Server arhitektura:** U projektu "DrawOut", ova arhitektura je ključna za podržavanje interakcije između korisnika (klijenta) i servera koji upravlja logikom igre, stanjem soba i sesijama. Ovaj obrazac omogućava centralizovanu obradu podataka na serveru i distribuiran pristup klijentima. To znači da će sav težak posao, poput upravljanja igrom i održavanja stanja, biti na serveru, dok će klijenti pristupati ovoj logici preko korisničkog interfejsa
- ❖ **Model-View-Controller (MVC):** Ovaj obrazac u projektu "DrawOut" koristi se na klijentskoj strani za upravljanje korisničkim interfejsom, obradu korisničkih interakcija, i ažuriranje prikaza. Ovaj obrazac pomaže u odvajanju korisničkog interfejsa od poslovne logike, čineći aplikaciju lakšom za održavanje i razvoj. Model upravlja podacima i pravilima igre, View prikazuje informacije i korisnički interfejs, a Controller deluje kao posrednik između Modela i View-a, upravljajući korisničkim unosima
- ❖ **Publish/Subscribe (Pub/Sub) model:** Ovaj obrazac se u "DrawOut" koristi za real-time obaveštavanje korisnika o promenama u igri, kao što su ažuriranja stanja igre ili obaveštenja o potezima drugih igrača. Ovo se može postići kroz tehnologije poput WebSocket-a, gde server (publisher) emituje informacije, a klijenti (subscribers) ih primaju u realnom vremenu. Ovo omogućava igračima da dožive fluidnu i interaktivnu igru, bez kašnjenja u komunikaciji

Generalna arhitektura

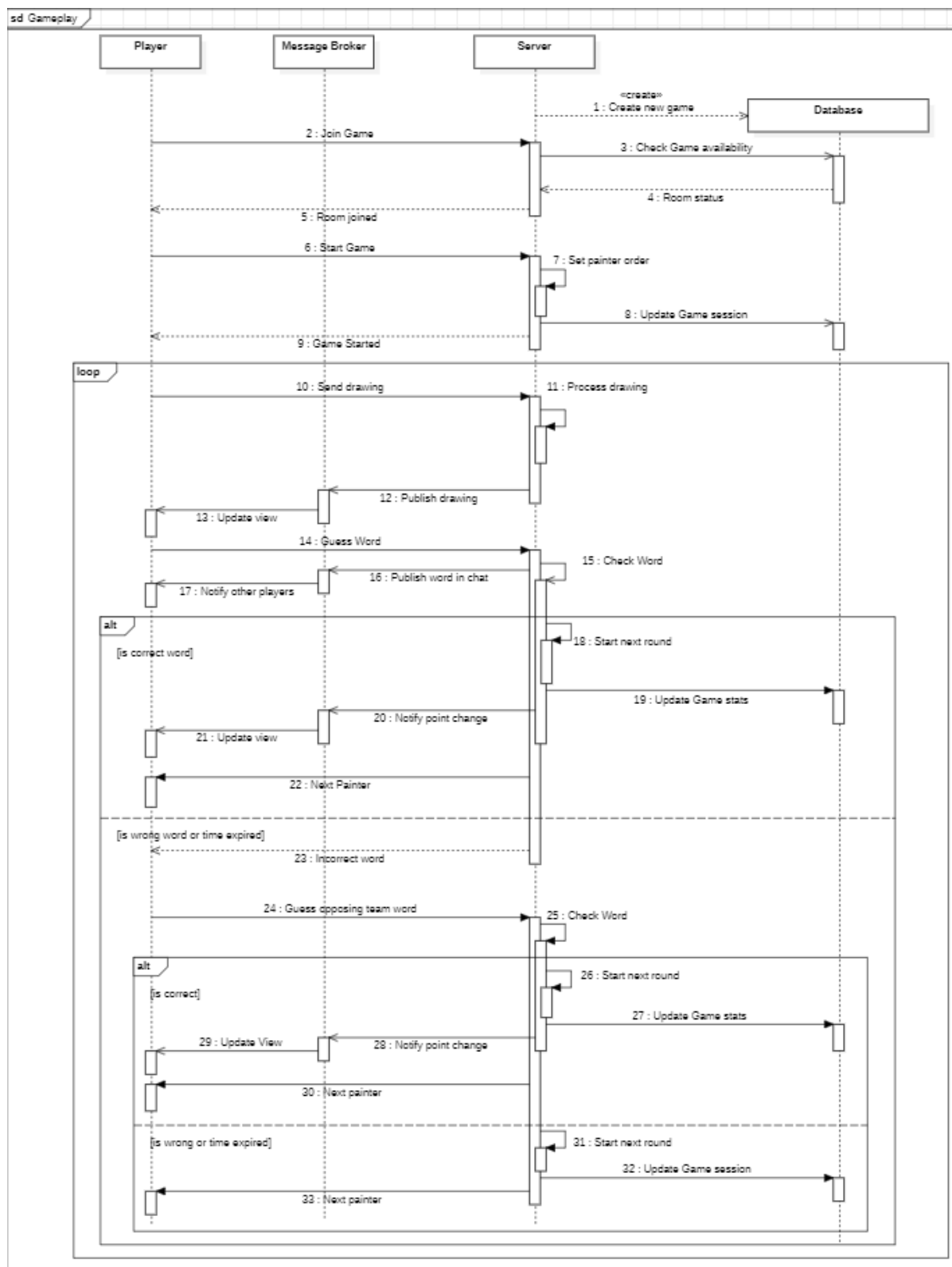


Strukturni pogled



Bihevioralni pogled

Na priloženoj slici je predstavljen primer bihevioralnog prikaza projekta "DrawOut", gde je detaljno ilustrovan proces toka igre. Ovaj prikaz obuhvata različite faze i interakcije unutar igre, ali ne uključuje specifičnu situaciju poslednje runde.



Implementacija

U ovom pasusu ćemo definisati okvire (framework-e) i biblioteke koje će biti korišćene u izradi web aplikacije “DrawOut”.

- ❖ **AngularJS** - JavaScript okvir za pisanje front-end, klijentske strane web aplikacije
- ❖ **Socket.IO/Websocket** - podrška real-time komunikacije između klijenta i servera
- ❖ **[.NET 8.0](#)** - okvir za pisanje back-end, serverske strane web aplikacije
- ❖ **Microsoft Entity Framework** - okvir za objektno-relaciono mapiranje
- ❖ **MongoDB** - NoSQL baza podataka
- ❖ **Redis** - NoSQL baza podataka, za keširanje i upravljanje sesijama/stanjima
- ❖ **SignalR** - Message Broker, biblioteka za uspostavljanje real-time komunikacije
- ❖ **Swagger/OpenAPI** - Za dokumentaciju i testiranje RESTful API-ja