Projekat – Faza 1 – Arhitektura softvera

Nikola Vasic, 18100

Luka Zdravkovic, 18167

1. **Kontekst i cilj softverskog projekta** :

Cilj ovog softverskog projekta jeste projektovati i implementirati klon multiplejer arkadne igre pod nazivom Agar.io, pružajući korisnicima slično iskustvo u onlajn multiplejer okruženju.

Sa jednostavnom i prepoznatljivom igračkom mehanikom, ova aplikacija želi pružiti korisnicima jedno zabavno iskustvo.

1. **Arhitekturno specifični zahtevi** :

Skalabilnost – S obzirom na dimanicnu prirodu igre i varijabilan broj igrača, arhitektura mora biti sposobna rukovati s promenjivim opterećenjem

Mrežna latencija – Igra zahteva nisku mrežnu latenciju kako bi osigurala glatko kretanje igrača i brzu reakciju na njihove akcije

Dostpunost servera – igrači mogu da udju u igru u bilo kom trenutku

Perzistencija podataka – obezbediti trajno memorisanje podataka svakog igrača

1. **Funkcionalni Zahtevi** :

- Registracija naloga i povezivanje na server igre

- Leaderboard, postojanje top liste gde igrač može u svakom trenutku videti svoju poziciju

- Mogućnost kretanja po polju na osnovu definisanih kontrola igre

- Izmena, tj azuriranje podataka

1. **Nefunkcionalni zahtevi** :

- Niska latencija – omogućiti brz odziv na akcije igrača, takodje redukcija vremena cekanja za ulazak u igru

- Jednostavan pristup - preko web pregledača

- Provera validnosti akcija prilikom igranja

- Mogućnost brzog oporavka veze prilikom njenog gubitka

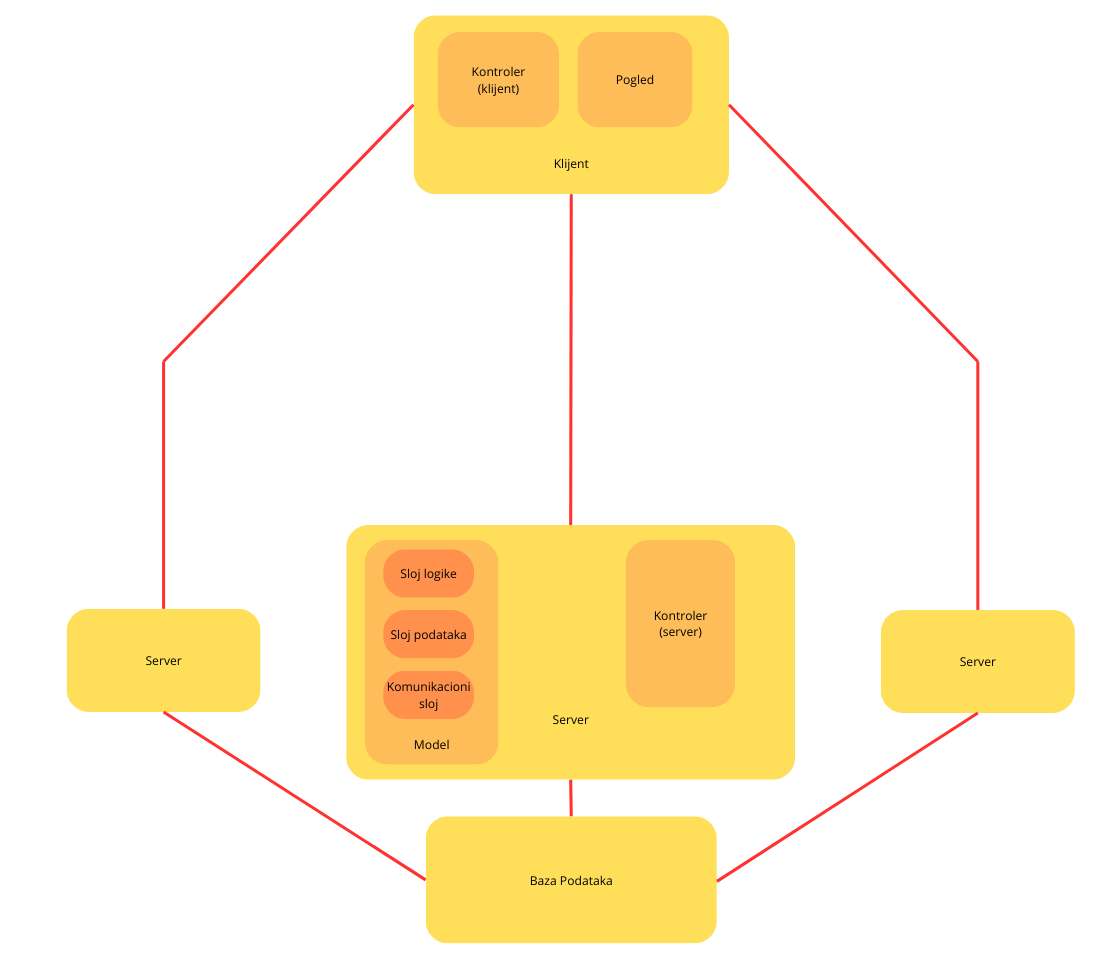
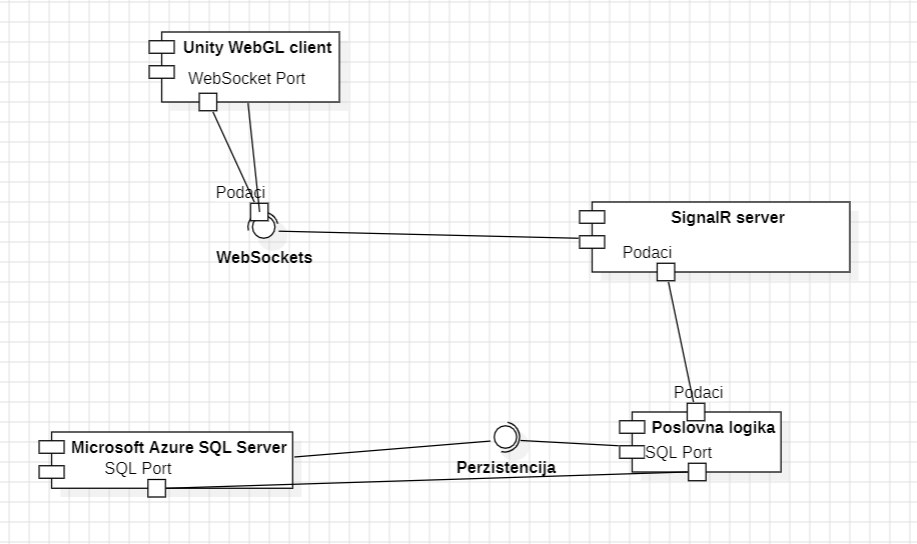
1. **Arhitekturni Dizajn** :

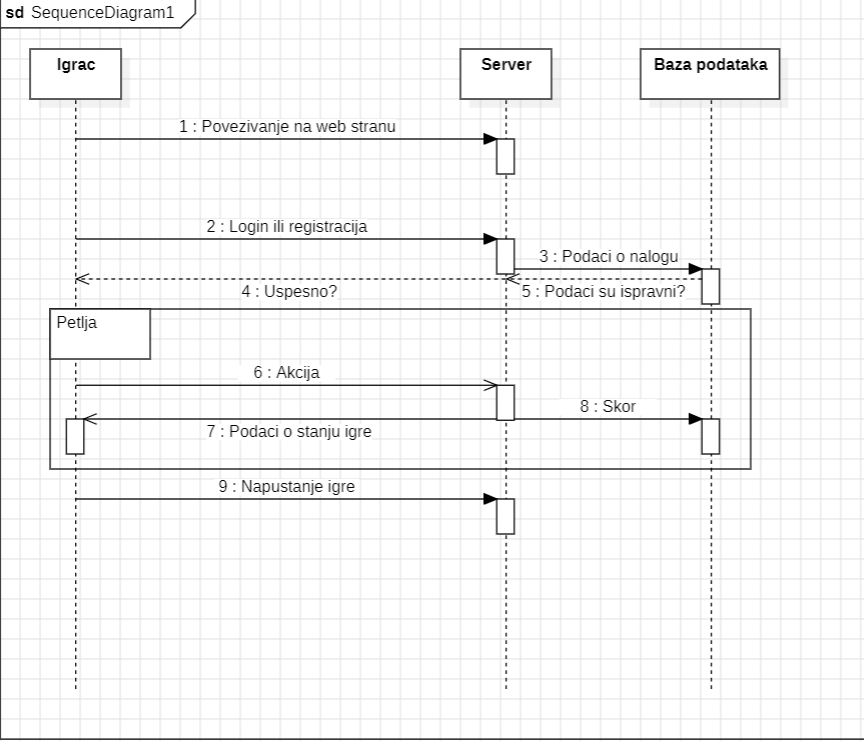
* Arhitekturni obrasci **:**

- Pub/Sub (Publisher/Subscriber) – server će imati SignalR hub koji ima ulogu publishera, dok su igrači subscriberi

- Client – Server – komunikacija u kojoj klijenta predstavljaju aktivni igrači u samoj igri. Sva komunikacija, uključujući interakcije medju igračima, logike igre i procesa validacije, centralizovana je na serverskoj strani aplikacije.

- Model-View-Controller(MVC) – ovaj obrazac biće implementiran kako bi struktuirao slojeve unutar same aplikacije. Model će sadržavati ključnu logiku aplikacije, pogled (view) će se odnositi za frontend (klijenta), dok će kontroler delovati kao posrednik izmedju modela i pogleda.

* **Box-Line dijagram** :
* Strukturni dijagram (komponenti)

* Bihevioralni dijagram (sekvencijalni)

6**. Aplikacioni okviri/Biblioteke** :

Baza podataka biće implementirana korišćenjem Redis baze podataka. Serverski deo će se realizovati pomoću .NET frameworka, kompanije Microsoft, dok će klijentska strana aplikacije biti realizovana pomoću Unity-ja, višeplatformskog game engine-a. Komunikacija izmedju servera i klijenta će se odvijati pomoću SignalR konekcije (WebSocket). Za komunikaciju izmedju serverskih komponenti biće implementiran RabbitMQ.