*Elektronski fakultet Niš*



**Arhitektura i projektovanje softvera**

**UNO INFINITY WAR**

****

Tim: B2J

David Bosnić 16489

Nikola Begović 16481

Jovana Jović 16639

PRVA FAZA

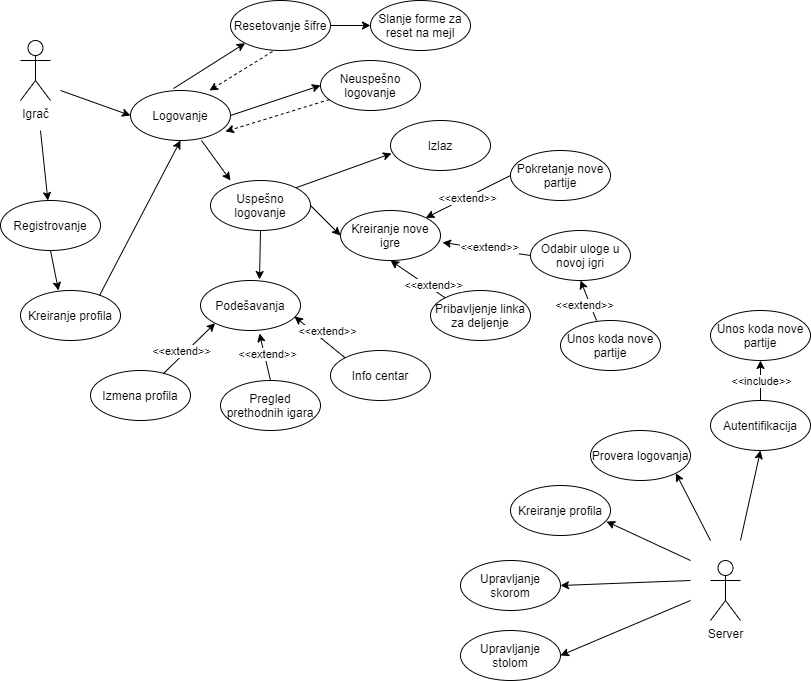
Kontekst i cilj projekta

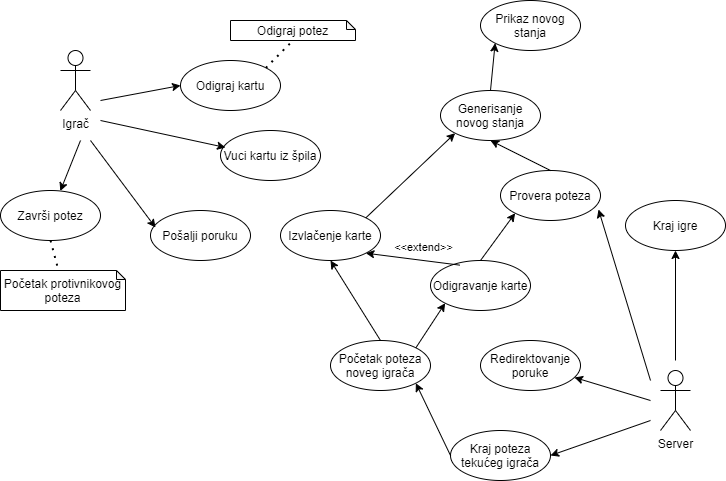
Cilj projekta je izrada aplikacije koja će omogućiti online multiplayer, turn based, kartašku igricu nalik na već postojeću društvenu igru uno. Aplikacija je tipa client-server i mora biti interkativna, sa preglednim i laganim korisničkim interfejsom. Ova igrica predstavlja platformu na kojoj više igrača može igrati za jednim stolom. Broj stolova je neograničen. Kraljni cilj svakog igrača je da ostane bez karata ili da sakupi unapred definisan broj poena. Poeni se sakupljaju odigravanjem poteza koji su u skladu sa standardnim pravilima ove društvene igre. Naravno pri ulazu u igru postoji game menu odakle se rukovodi osvnovnim podesavanjima igre i odakle se pristupa samoj partiji. Projekat se smatra uspesnim ako su svi arhitekturni funkcionalni i nefunkcionalni zahtevi zadovoljeni.

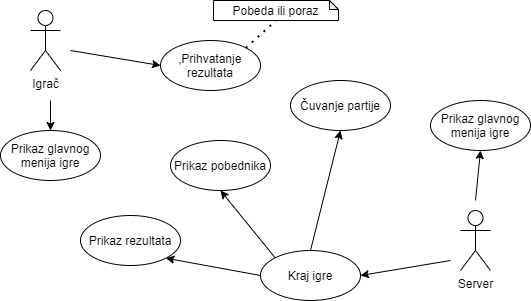
Arhitekturni zahtevi

Arhitekturno značajni slučajevi korišćenja (glavni funkcionalni zahtevi)

1. **Registracija korisničkog naloga** – kreiranje naloga za korisnika koji želi da koristi aplikaciju je neophodno. Na početnoj strani postoji opcija za kreiranje naloga pri čemu je neophodno odabrati email koji nije zauzet i odabrati lozinku.
2. **Prijavljivanje na korisnički nalog** – na početnoj strani pored opcije za kreiranje naloga nalaziće se opcija i za logovanje na već postojeći nalog pomoću email-a i ispravne lozinke.
3. **Meni –** Omoguciti korisniku pristup meniju. Meni treba da sadrzi opcije za kreiranje igre, pridruživanje već kreiranoj igri, pregled prethodnih partija, opcija za odjavljivanje.
4. **Igrač koji počinje igru -**Igrač koji kreira igru igra prvi.
5. **Početak igre –** Na početku igre, potrebno je korisniku prikazati početnu kartu na tabli, početni set karata u njegovoj ruci, tabelu sa imenima svih učesnika I njihovim poenima, kao I identifikator igre na osnovu koje će drugi učesnici moći da se priključe istoj.
6. **Potez –** Nakon početka igre, igrači igraju jedan za drugim. Igrači uvek imaju prikaz ko je na potezu, zadnje odigranu kartu, boju sledeće karte koja se očekuje I prikaz tabele sa trenutnim poenima.
7. **Odigrana karta –** Omogućiti prosleđivanje informacija o odigranoj karti serveru. Na osnovu odigrane karte menja se trenutni pogled igre I table svim učesnicima partije (ukoliko je potez bio validan) a onom ko je odigrao potez menja se I pregled karata u ruci (ukoliko je potez bio validan, a ukoliko nije, prikaže mu se odgovarajuća notifikacija).
8. **Validacija odigrane karte –** Omogućiti validaciju odigrane karte. Svaka karta koju izabere da odigra mora da bude moguća za igru u tom trenutku. U slučaju da potez nije validan prikazaće se odgovarajuća notifikacija onom ko je pokušao da odigra taj potez. Ako je potez validan, pokreće se akcija odigrane karte.
9. **Kraj igre –** kada neko od igrača potroši sve karte ili je skupio unapred definisan broj poena igra se zavrsava. Host partije ima mogućnost da (ukoliko želi) ponovi partiju sa istim učesnicima.
10. **Replay –**Mogućnost da se odgledaju prethodne partije.
11. **Kick-ovanje –** Host ima mogućnost da izbaci nekog igrača iz partije.
12. **Mogucnost za chat-** mogućnost da u toku partije učesnici razmenjuju tekstualne poruke.

**Na sledećim dijagramima su prikazane određene funkcionalnosti aplikacije. Slučajevi koriscenja pre/tokom/nakon jedne igre respektivno su predstavljeni na narednim dijagramima.

**

**

Ne-funkcionalni zahtevi (atributi kvaliteta)

1. Minimalno vreme odziva - Obezbediti odaziv nakon igranja poteza ne vise od 0.5s.
2. Upotrebljivost - Obezbediti intuitivan, pregledan i prost interfejs.
3. Sigurnost - Obezbediti sigurno prenošenje i čuvanje korisničkih podataka, pre svega lozinki drugih osetljivih informacija
4. Skalabilnost - Aplikacija treba da bude u stanju da podrži veliki broj istovremenih korisnika kao i nagle skokove u broju korisnika koji koriste aplikaciju.
5. Obezbediti pouzdanu konekciju sa serverom tokom igre.
6. Obezbediti dostupnost sistema (24x7)

Tehnička i poslovna ograničenja

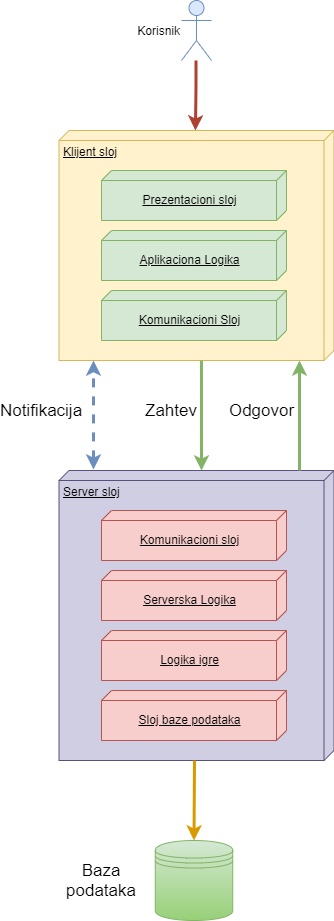
Za klijentsku stranu će se koristiti html, css i javascript. Za realizaciju aplikacionog servera bice koriscen node.JS, za komunikaciju koristiće se socket.IO i za rad sa bazom mongoose. Za perzistenciju podataka o samoj igri koristiće se mongoDB, dok će se za kreiranje i autentifikaciju naloga koristiti firebase.

Arhitekturni dizajn

Nasa aplikacija implementira Layered arhitekturu u kojoj 2 podsloja implementiraju Publish/Subscribe arhitekturni obrazac

1. **Klijentski sloj** – prezentacioni sloj koji pruža interfejs ka korisnicima i interaguje sa klijentskim komunikacionim slojem.
2. **Klijentski komunikacioni sloj** – sloj koji implementira publish/subscribe arhitekturni obrazac i interaguje sa serverskim komunikacionim slojem kao i prezentacionim slojem. Obavestava serverski komunikacioni sloj kada dođe do nekog događaja na klijentu i vrši izmenu interfejsa ukoliko ga serverski komunikacioni sloj obavesti o promeni stanja igre.
3. **Serverski komunikacioni sloj** - sloj koji implementira publish/subscribe arhitekturni obrazac i interaguje sa klijentskim komunikaciom slojem kao i serverom. Prima novonastale događaje od klijenata i prosleđuje ih serveru za obradu. Prosleđuje novonastali događaj svim pretplatnicima.
4. **Serverski sloj** – zadužen je za komunikaciju sa više klijenata, rad sa bazom podataka preko ORM-a i pristigle poteze prosleđuje za obradu sloju logike igre.
5. **Sloj logike igre** – Obrađuje odigran potez i određuje njegovu validnost i efekat poteza
6. **Sloj baze podataka** – čuvaju se podaci o korisnicima i mečevima

Generalna arhitektura (box-line)

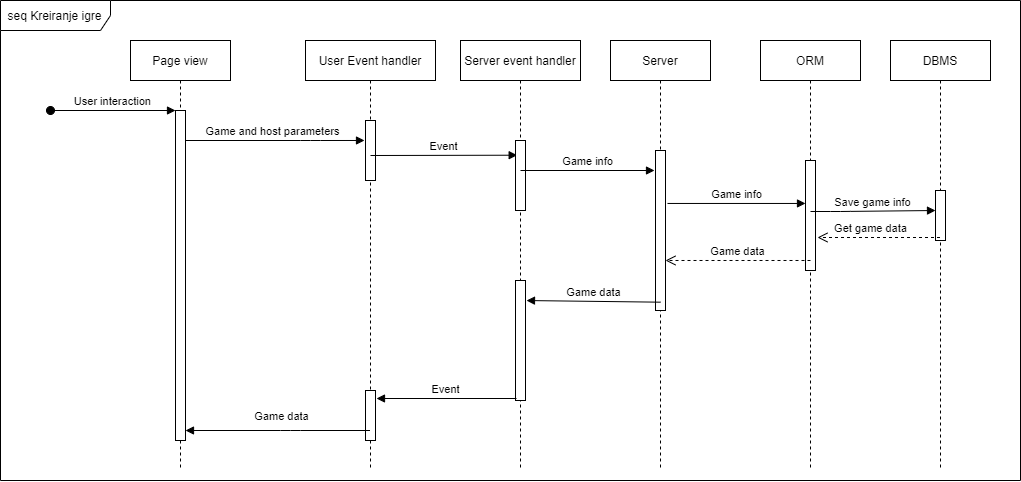


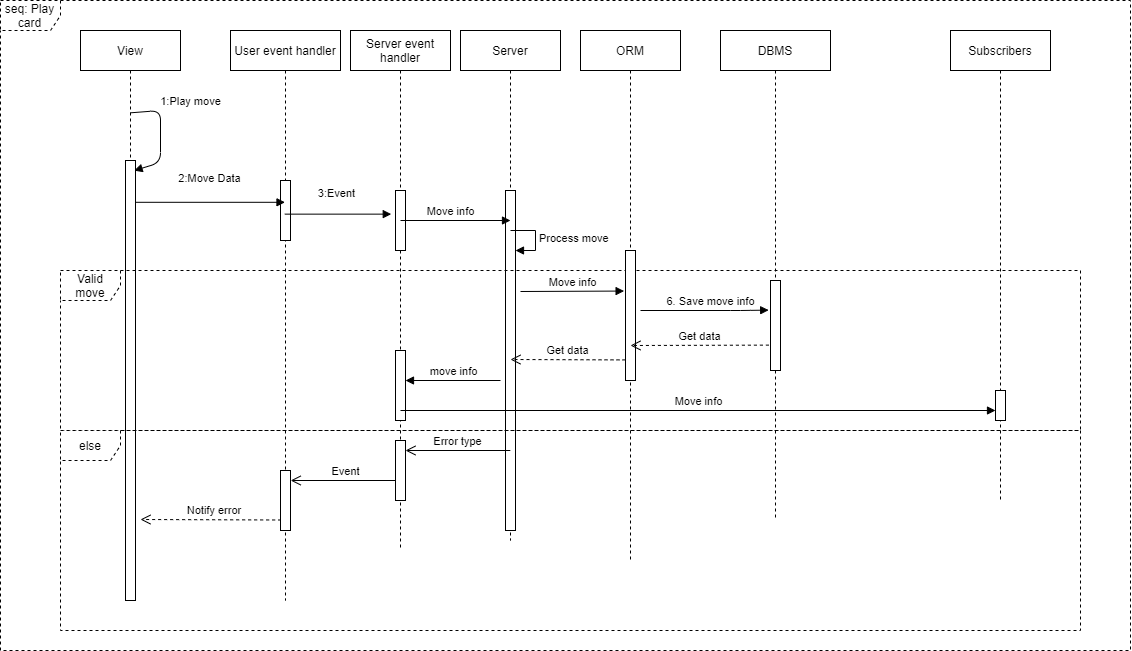
Strukturni pogledi

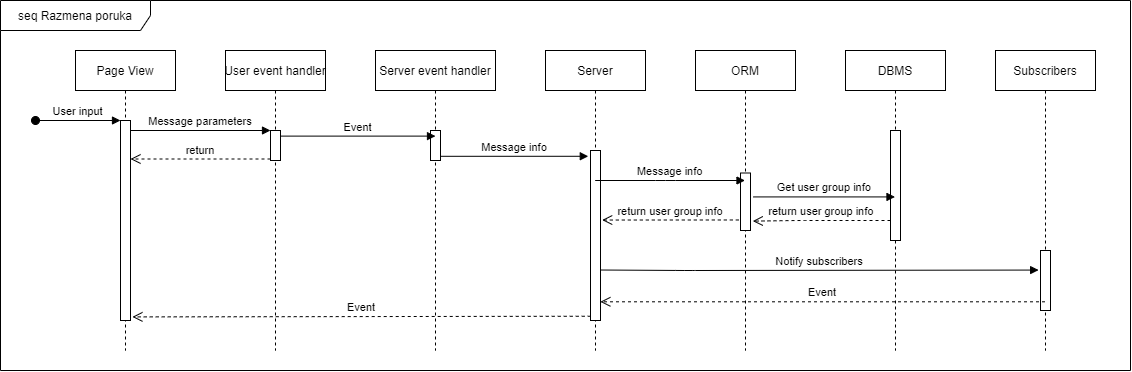
--Implementaciona pitanja - biblioteke, komponente i okviri (framework-i) koji će biti korišćeni za implementaciju

Stukturni pogled/dijagram komponenti/obavezno firebase

Bihevioralni pogled





****