*Elektronski fakultet Niš*



**Arhitektura i projektovanje softvera**

**UNO INFINITY WAR**

****

DRUGA FAZA  
Model podataka i perzistencije

Naš projekat UNO – Infinity War koristi mongodb kao primarnu bazu podataka, u kojoj se čuvaju osnovne informacije o korisnicima, kao i o igrama koje se trenutno igraju ili su završene. Za komunikaciju sa bazom i modeliranje objekata koristimo mongoose. Za čuvanje informacija koje su potrebne za logovanje korisnika (kao što su email i šifra) koristimo google-ov servis za autentifikaciju „firebase“.  
Kroz mongoose smo definisali sledeće šeme objekata : Igra, Korisnik, Igrač, Karta, OdigranaKarta i Chat.

* Igra – pamti podatke vezane za igru
  + \_id – jedinstveni identifikator tipa String
  + igraci – niz igrača koji trenutno učestvuju u partiji
  + igracNaPotezu – indeks igrača koji treba sledeći da odigra potez
  + trenutnaKarta – sadrži atribute koji opisuju trenutnu kartu na tabli
  + trenutnaBoja – trenutna važeća boja na tabli
  + obrnutRedosled – opisuje u kom smeru trenutno ide tok igre, tipa Boolean
  + pocelaIgra – opisuje da li je igra startovana
  + zavrsenaIgra – opisuje da li je igra završena
  + brojIgraca – trenutni broj igrača u partiji
  + chat – sadrži poruke prosleđene tokom igre
  + karte – sadrži atribute vezane za sve dosadašnje odigrane karte
  + datum – datum kad je partija započeta
* Korisnik – pamti podatke vezane za korisnika
  + \_id – jedinstveni identifikator tipa String, dodeljuje se nakon logovanja korisnika, pri čemu se vrednost poklapa sa id-jem iz firebase profila datog korisnika
  + prethodne igre - niz identifikatora svih prethodnih partija u kojem je korisnik bio igrač
* Igrac – pamti podatke od značaja vezane za korisnika koji igraju neku partiju
  + karte – pamti atribute vezane za trenutne karte koje se nalaze „u ruci“ igrača
  + ime – nickname koji je igrač odabrao pre pridruživanja partiji
  + indeks – broj koji identifikuje igrača u toku partije
  + idIgraca – identifikator koji ukazuje na \_id korisnika
  + izvucenihKarata – broj koji ukazuje koliko je trenutno karata izvukao igrač (da bi ga ograničili kasnije na neki broj koji odredimo)
  + mozeDaZavrsi- Boolean kojim određujemo da li igrač može da završi potez (npr, ako nije izvukao ni jednu kartu a želi da završi potez, to nije dozvoljeno)
  + socketId – identifikator socketa preko koga se vrši komunikacija
  + poeni – trenutni poeni igrača
* Karta – pamti podatke od značaja vezane za kartu koja je odigrana
  + vrednost – označava brojčanu vrednost karte (0-9 za obične karte, više od toga su specijalne)
  + boja – boja odigrane karte tipa String
  + specijalna – Boolean promenljiva koja ukazuje da li karta ima neki specijalni efekat (promena smera toka igre, naredni igrač da kupi +2 karte itd…)
* OdigranaKarta – pamti podatke koji ukazuju na to koji igrač je odigrao koju kartu
  + karta – ima struktuju kao šema koju smo iznad nazvali „Karta“
  + korisnik – ima strukturu kao šema koju smo iznad nazvali „Korisnik“