

Programsko inženjerstvo

Ak. god. 2022./2023.

GlobeRunner

Dokumentacija, Rev. 2

Grupa: *CodeBreathers*

Voditelj: *Lovro Kovačić*

Datum predaje: 13. 1. 2023.

Nastavnik: *Hrvoje Nuić, mag. ing.*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
2.1 Uvod	5
2.2 Cilj i očekivane funkcionalnosti	6
3 Specifikacija programske potpore	10
3.1 Funkcionalni zahtjevi	10
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	28
3.2 Ostali zahtjevi	31
4 Arhitektura i dizajn sustava	32
4.1 Baza podataka	34
4.1.1 Opis tablica	34
4.1.2 Dijagram baze podataka	39
4.2 Dijagram razreda	40
4.3 Dijagram stanja	43
4.4 Dijagram aktivnosti	44
4.5 Dijagram komponenti	46
5 Implementacija i korisničko sučelje	48
5.1 Korištene tehnologije i alati	48
5.2 Ispitivanje programskog rješenja	50
5.3 Dijagram razmještaja	51
5.4 Upute za puštanje u pogon	52
6 Zaključak i budući rad	58
Popis literature	60
Indeks slika i dijagrama	61

Dodatak: Prikaz aktivnosti grupe

62

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Popunjen predložak za naslovnicu.	V. Pezo	29.10.2022.
0.2	Napisana prva verzija opisa projekta.	V. Pezo	30.10.2022.
0.3	Funkcionalni zahtjevi, prvi dio obrazaca uporabe, ostali zahtjevi.	M. Šola	02.11.2022.
0.4	Raspisani obrasci uporabe do kraja	V. Pezo	2.11.2022.
0.5	Dodani dijagrami obrazaca upotrebe.	L. Raguž	3.11.2022.
0.6	Dodani sekvencijski dijagrami	E. Kumer	3.11.2022.
0.7	Ispravci 2. i 3. poglavlja, dodavanje fotografije.	V. Pezo	16.11.2022.
0.8	Napisan opis arhitekture sustava, opis baze podataka	M. Šola	17.11.2022.
0.9	Dodan dijagram baze podataka	S.Rivic Carevic	17.11.2022.
0.10	Izmijenjeni sekvencijski dijagrami, dodan opis sekvencijskih dijagrama i zapisnici sastanaka	E. Kumer	17.11.2022.
0.11	Dodan opis dijagrama razreda i dijagram za Models	V. Pezo	18.11.2022.
0.12	Dodani dijagrami Controllers i Data transfer object u dijelu s dijagramima razreda	L. Raguž	18.11.2022.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

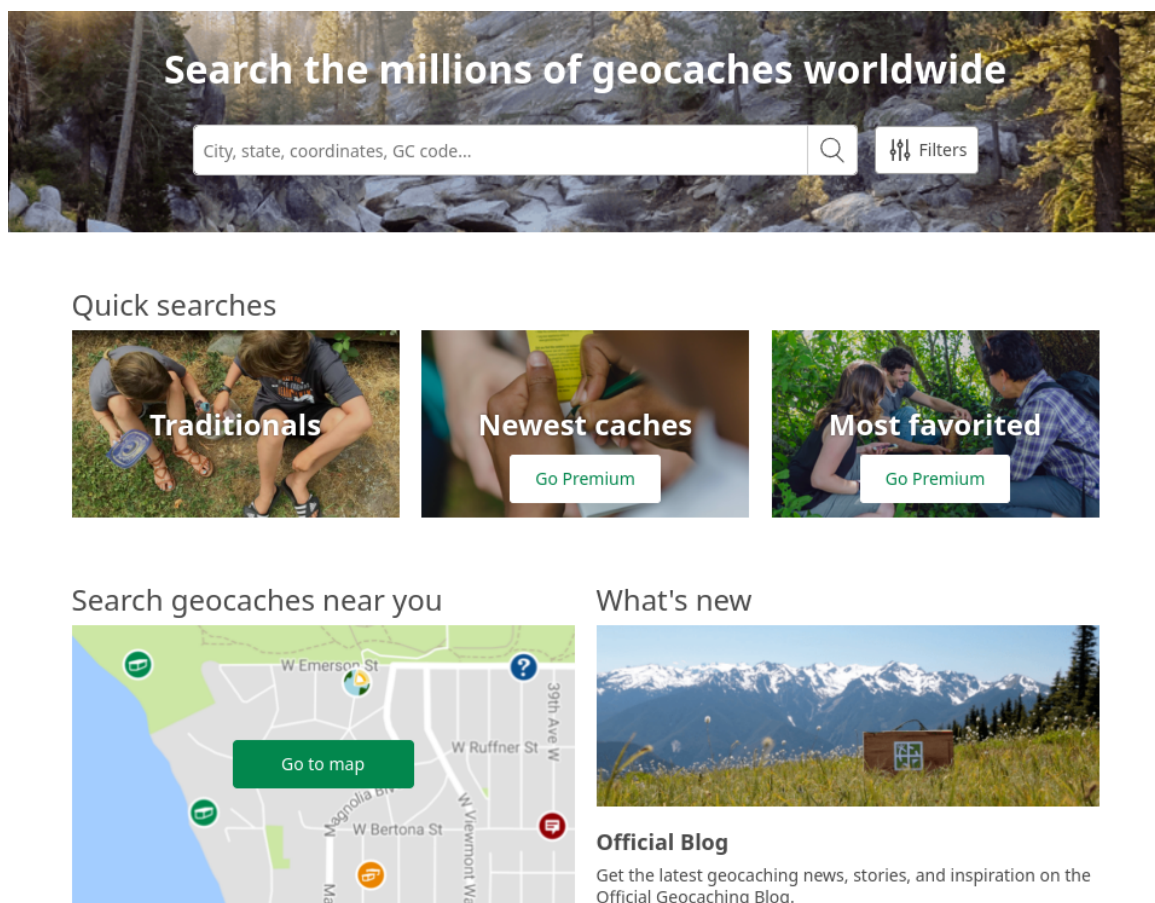
Rev.	Opis promjene/dodatka	Autori	Datum
1.0	Konačni ispravci, prijelomi stranica, aktivnost	V. Pezo	18.11.2022.
1.1	Ispravci 4.1 poglavlja, izmijenjen dijagram baze podataka	S.Rivic Carevic	10.01.2023.
1.2	Dodan dijagram razmještaja i njegov opis	S.Rivic Carevic	11.01.2023.
1.3	Dodan dijagram aktivnosti i njegov opis	S.Rivic Carevic	12.01.2023.
1.4	Dodan dijagram stanja i njegov opis	S.Rivic Carevic	13.01.2023.
1.5	Ažuriran dijagram razreda i dodan dijagram komponenti s opisom	V. Pezo	13.01.2023.
1.6	Ažurirani obrasci uporabe i arhitektura sustava, dodan dio o korištenim tehnologijama i alatima	M. Šola	13.01.2023.
1.7	Upute za puštanje u pogon	E. Kumer	13.01.2023.
1.8	Dodan zaključak	V. Pezo	13.01.2023.
2.0	Dodani zapisnici sastanaka, aktivnost	M. Šola	13.01.2023.

2. Opis projektnog zadatka

2.1 Uvod

Igra *GlobeRunner* zamišljena je kao način da se ljude potakne na istraživanje lokalnih povijesnih i prirodnih znamenitosti. Pri posjetima znamenitostima igrači će se kretati na otvorenom, što je svakako bolja alternativa provođenju vremena u zatvorenom prostoru. Kroz same bitke, odnosno odabir kartica za njih, igrači će bolje upoznati bitne attribute lokalnih znamenitosti kraj kojih često prolaze bez razmišljanja jer će im upravo pažljiv odabir znamenitosti donositi pobjede u bitkama protiv drugih igrača.

Platforme sa sličnom vizijom već postoje, jedna od njih je *Geocaching*, Slika 2.1. Radi se o igri koja se zasniva na skrivenim spremnicima (eng. *geocaches*) koji sadrže mali dnevnik u koji se posjetitelji zapisuju kada ga pronađu. Igrači u samoj aplikaciji biraju koji spremnik žele pronaći i zatim se upute u potragu. Prilikom pronalaska u spremniku mogu ostaviti i neku sitnicu, ili pak uzeti nešto iz njega, a nakon pronalaska svoje dojmove mogu objaviti na platformi. Razlika u odnosu na igru *GlobeRunner* jest u tome da postoje fizički artefakti koji se skrivaju na proizvoljna mjesta i zatim pronalaze, dok je cilj ove igre sam posjet lokaciji koja treba biti na neki način značajna, povijesno ili pak geografski. Po pitanju nepostojanja fizičkih artefakata, *GlobeRunner* je sličniji igri *Pokémon GO* u kojoj se uz mjesta lokalnih znamenitosti vežu zamišljena bića iz animirane japanske franšize *Pokémon*, zbog čega se pak razlikuje po značenju samih virtualnih artefakata. Sam tijek igre, kroz bitke u kojima se koriste kartice, podsjeća na kolekcionarsku igru *Yu-Gi-Oh!*, no cijela premisa igre u potpunosti je drugačija.

Slika 2.1: Web sjedište platforme *Geocaching*

Upravo bi iz navedenih razloga *GlobeRunner* mogao privući igrače koji žele na zabavan način upoznati područja u kojima borave i kojima se kreću, kojeg god uzrasta bili, dok god paze na svoju sigurnost, posebno pri posjeti izazovnijim lokacijama poput planinskih vrhova. Ova aplikacija otvara svoja vrata i onim najznatijeljnijima koji žele poboljšati igru, odnosno predložiti nove lokacije i tako istaknuti njihov značaj.

2.2 Cilj i očekivane funkcionalnosti

Cilj je ovog projekta razviti programsku potporu za web aplikaciju, odnosno igru *GlobeRunner*, koja će igračima omogućiti prikupljanje kartica koje su vezane uz neke geografske ili povijesne znamenitosti tako što će fizički doći na lokaciju same znamenitosti. Svaka lokacija ima minimalno naziv, opis i fotografiju. Nakon što skupi karticu, igrač ju može koristiti u bitkama protiv drugih igrača u svojoj

blizini. Izazov koji očekujemo nakon puštanja aplikacije u pogon jest populacija baze podataka jer je postojanje kartica za skupljanje premisa za privlačenje igrača. Stoga je cilj razvoja programske potpore osigurati i postojanje mogućnosti da igrači doprinesu igri predlaganjem novih kartica, ali i da ti doprinosi budu pregledani i odobreni od strane za to ovlaštenih korisnika - kartografa.

Prilikom pokretanja sustava prikazuje se stranica dobrodošlice za neautentificirane korisnike na kojoj se ukratko predstavlja igra i nude gumbi *Sign in*, *Register as player* i *Register as cartographer*. Prijava postojećeg korisnika vrši se email adresom ili korisničkim imenom i lozinkom, dok su za registraciju novog igrača (*Player*) potrebni sljedeći podatci:

- korisničko ime
- fotografija
- lozinka
- email adresa

Za registraciju kartografa (*Cartographer*) potrebni su i sljedeći podatci:

- IBAN računa (za uplatu naknade)
- fotografija osobne iskaznice

Nakon ispunjavanja obrasca za registraciju, novi igrač treba potvrditi svoju e-mail adresu klikom na dostavljenu poveznicu. Tako potvrđen igrač može krenuti s upotrebom web aplikacije, dok registraciju kartografa naknadno mora odobriti administrator.

Igrač (*Player*) nakon registracije započinje igru s predefiniranim početnim brojem bodova u ELO sustavu (npr. u šahu je 1500). Nakon prijave igrač vidi početnu stranicu koja prikazuje kartu na kojoj su označene lokacije s karticama koje još nije prikupio. Kada se igrač nađe u blizini neskupljenih kartica, ispod karte se pojavi popis kartica u neposrednoj blizini, njih onda može pokušati pokupiti pritiskom na gumb "Pokupi" koji će to dopustiti samo ako je igrač jako blizu kartici. Na početnoj stranici, kao i na ostalima, igrač vidi i izbornik koji mu omogućuje navigaciju kroz web aplikaciju. Sadržaj izbornika je sljedeći:

- Vlastiti profil (*My profile*)
- Globalna statistika (*Global stats*)
- Igrači u blizini (*Nearby players*)

Kroz izbornik igrač može pristupiti vlastitom profilu na kojem vidi statistiku svojih borbi i sve karte koje je skupio. U izborniku se također nalazi i poveznica na stranicu "Globalna statistika" koja prikazuje ljestvicu svih igrača sortiranu po ELO bodovnom sustavu, a uz to prikazuje i statistiku svih odigranih bitki i skupljenih kartica u igri. Do stranice koja prikazuje popis drugih igrača u blizini igrač također dolazi preko spomenutog izbornika. Na tom popisu igrač odabire koga će izazvati, a nakon što drugi igrač prihvati izazov, svaki odabire karte s kojima će ići u bitku. Iznad spomenutog popisa igraču se pojave zahtjevi za bitke kada ih drugi igrači izazovu, a on ih onda može prihvatiti ili odbiti. Bitka se provodi usporedbom površina koje razapinju tri kartice koje pojedini igrač odabere, uz napomenu da se površina skalira prosjekom jačina tih triju kartica. Jačina kartice smanjuje se nakon svakog korištenja i to smanjenje vrijedi samo za korisnika koji ju koristi. Nakon svake bitke igračima se ažurira njihov broj bodova shodno ELO sustavu. Ako igrač pređe određeni prag bodova, dodjeljuje mu se status naprednog igrača.

Napredni igrač (*Advanced player*), uz sve spomenute funkcionalnosti običnog igrača, u svom izborniku ima i opciju "Dodaj lokaciju" koja ga vodi na stranicu za predlaganje nove lokacije, odnosno kartice. Kako bi dodao neku lokaciju, napredni igrač mora biti u njezinoj blizini. U obrazac za dodavanje lokacije mora unijeti sve podatke koje kartica treba imati - naziv, opis i fotografiju. Nakon što predloži novu lokaciju, ona mu postaje vidljiva na njegovoj karti za dodavanje, ali uz naznaku da još nije odobrena.

Kartograf (*Cartographer*) je poseban korisnik aplikacije koji ima ulogu pregledavanja, dopunjavanja i odobravanja predloženih lokacija za popunjavanje baze podataka. Prilikom prijave u sustav kartograf na početnoj stranici vidi kartu na kojoj mu se, ovisno o odabranim filtrima, prikazuju prijedlozi za nove kartice i/ili već odobrene kartice. Zadatak je kartografa da odbaci prijedlog ako već postoji lokacija koja se pokušava dodati. Ako lokacija već ne postoji u bazi, onda kartograf pregledava i uređuje opis, naziv i fotografiju nakon čega karticu može odobriti ili pak označiti da ju treba provjeriti na terenu, pri čemu ju dodaje u listu kartica koje treba provjeriti. Putem izbornika kartograf može pristupiti sljedećim funkcionalnostima:

- Vlastiti profil (*My profile*)
- Provjera na terenu (*On-site approval*)

Na stranici vlastitog profila kartograf ima uvid u kartice koje je već odobrio. Stranica za provjeru na terenu daje kartografu priliku da iz liste kartica koje treba

provjeriti na terenu odabere one koje će on provjeriti. Tako dopunjava svoju listu kartica za običi. Kada označi željeni broj lokacija za provjeru, kartograf klikom na gumb *Generate route* zadaje sustavu problem trgovačkog putnika koji sustav rješava koristeći vanjski servis OSRM te kartografu prikazuje rutu koju treba obići da bi na terenu provjerio sve kartice koje je odabrao.

Administrator (*Admin*) ima ulogu nadgledanja korisnika, lokacija i kartografa, a u tu svrhu u svom izborniku ima pristup sljedećim stranicama:

- Svi korisnici (*All users*)
- Sve kartice (*All cards*)
- Odobravanje kartografa (*Cartograph approvals*)

Na stranici "Svi korisnici" administratoru se prikazuje popis svih korisnika. Za svakog korisnika nudi mu se opcija uređivanja osobnih podataka i opcija isključenja iz igre koje ga vode na pripadne stranice s tim funkcionalnostima. Na stranici "Sve kartice" administratoru je prikazana karta sa svim karticama u igri, a klikom na bilo koju od njih administrator dobiva opciju uređivanja kartice izmjenom bilo kojeg atributa same kartice. "Odobravanje kartografa" je stranica putem koje administrator dobiva uvid u registracije kartografa koje još nisu odobrene. Za svaku od njih nudi mu se opcija da ju odobri.

Uz očekivane funkcionalnosti, identificirali smo i nadogradnje sustava koje bi poboljšale sigurnost, performanse i iskustvo korisnika, a čiji razvoj nije dio ovog projekta. Neke od njih su:

- Složeniji proces borbe
- Funkcija obnavljanja snage kartice ponovnim skupljanjem
- Korištenje *hash* funkcija za spremanje i razmjenu zaporki
- Integracija drugih načina prijave

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Igrači
 - (a) Igrač početnik
 - (b) Napredni igrač
2. Kartograf
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) vidjeti stranicu dobrodošlice na kojoj se nalaze upute za korištenje aplikacije
 - (b) pregledati dostupne kartice na karti
 - (c) vidjeti globalnu statistiku odigranih borbi i sakupljenih lokacija te po-redak svih igrača
 - (d) se registrirati u sustav, stvoriti novi korisnički račun za koji su mu po-trebni korisničko ime, fotografija, lozinka i e-mail adresa
2. Igrač početnik (inicijator) može:
 - (a) sakupljati lokacije, tj. kartice
 - (b) vidjeti popis ostalih aktivnih igrača koji se nalaze unutar 50 km i s njima ući u borbu
 - i. odabir skupa karata s kojima će se boriti
 - (c) pristupiti pregledu profila igrača
 - i. prikaz svih karata koje je igrač skupio
 - ii. prikaz ranga na globalnoj ljestvici

iii. prikaz statistike vezane uz zadnjih 10 borbi s drugim igračima

3. Napredni igrač (inicijator) može:

- (a) sakupljati lokacije, tj. kartice
- (b) vidjeti popis ostalih aktivnih igrača koji se nalaze unutar 50 km i s njima ući u borbu
 - i. odabir skupa karata s kojima će se boriti
- (c) pristupiti pregledu profila igrača
 - i. prikaz svih karata koje je igrač skupio
 - ii. prikaz ranga na globalnoj ljestvici
 - iii. prikaz statistike vezane uz zadnjih 10 borbi s drugim igračima
- (d) prijaviti novu lokaciju (s potrebnim podacima) u čijoj se neposrednoj blizini nalazi

4. Kartograf (inicijator) može:

- (a) se registrirati za što su mu potrebni IBAN račun za uplatu i fotografija osobne iskaznice
- (b) nadopuniti bazu podatka s lokacijama koje su igrači prijavili
- (c) vidjeti prijave na karti
 - i. odbijanje, potvrđivanje i uređivanje prijave
 - ii. oznaka za potrebnu potvrdu s terena
- (d) označiti koje će prijave osobno provjeriti

5. Administrator (inicijator) može:

- (a) vidjeti i uređivati popis svih registriranih korisnika i njihovih osobnih podataka
- (b) uređivati postojeće lokacije u igri
- (c) dodijeliti igračima privremeno isključenje iz igre

6. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o lokacijama
- (c) pohranjuje sve podatke o provedenim borbama i izazovima

3.1.1 Obrasci uporabe

UC1 - Registracija kao igrač

- **Glavni sudionik:** Gost
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Gost odabire opciju za registraciju igrača
 2. Gost unosi potrebne korisničke podatke
 3. Gost dobiva e-mail kojim potvrđuje svoj račun
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnog e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC2 - Registracija kao kartograf

- **Glavni sudionik:** Gost
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju kartografa
 2. Korisnik unosi potrebne korisničke podatke
 3. Korisnik dobiva e-mail kojim potvrđuje svoju prijavu
 4. Korisnika administratori potvrđuju kao kartografa
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu, pružanje neispravnog e-maila ili unos neispravnog IBAN-a
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC3 - Prijava u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke
 2. Potvrda o ispravnosti unesenih podataka
 3. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime/e-mail/lozinka
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju

UC4 - Pregled osobnih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati osobne podatke i statistike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju *My Profile*
 2. Aplikacija prikazuje osobne podatke korisnika i njegovu statistiku

UC5 - Pregled profila drugih igrača

- **Glavni sudionik:** Igrač
- **Cilj:** Pregledati profil drugog igrača
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik se nalazi na bilo kojoj stranici koja mu prikazuje korisničko ime drugog igrača
 2. Korisnik klikom na korisničko ime igrača zahtijeva pregled njegovog profila

3. Korisniku se prikazuje javni profil odabranog korisnika

UC6 - Uređivanje vlastitih podataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Urediti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za promjenu podataka
 2. Korisnik mijenja svoje osobne podatke
 3. Korisnik sprema promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Korisnik promijeni svoje osobne podatke, ali ne odabere opciju *Save changes*
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

UC7 - Brisanje korisničkog računa

- **Glavni sudionik:** Korisnik
- **Cilj:** Izbrisati svoj račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pregledava osobne podatke
 2. Otvara se stranica s osobnim podacima korisnika
 3. Korisnik briše račun
 4. Korisnički račun se izbriše iz baze podataka
 5. Otvara se stranica za registraciju

UC8 - Skupljanje kartice

- **Glavni sudionik:** Igrač
- **Cilj:** Pokupiti karticu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i nalazi se u neposrednoj blizini lokacije na kojoj je kartica

- **Opis osnovnog tijeka:**
 1. Korisnik odabire karticu na karti
 2. Korisnik odabire opciju *Collect*
 3. Kartica se nalazi na stranici *My Profile* -> *Inventory*
- **Opis mogućih odstupanja:**
 - 2.a Odabir kartice koju nije moguće pokupiti s korisnikove trenutne lokacije
 1. Sustav obavještava korisnika o neuspjelom sakupljanju kartice

UC9 - Pregled globalne statistike

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati globalnu statistiku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju *Global stats* koja ga vodi na stranicu gdje dobiva uvid u globalnu statistiku

UC10 - Pregled igrača u blizini

- **Glavni sudionik:** Igrač
- **Cilj:** Pregledati igrače u blizini
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju prikaz liste najbližih igrača

UC11 - Izazivanje igrača

- **Glavni sudionik:** Igrač
- **Cilj:** Izazvati drugog igrača
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen i nalazi se unutar 50 km udaljenosti od drugog igrača
- **Opis osnovnog tijeka:**
 1. Igrač odabire igrača kojeg želi izazvati
 2. Igrač šalje drugom igraču zahtjev za borbu

UC12 - Odgovor na izazov igrača

- **Glavni sudionik:** Igrač
- **Cilj:** Odgovoriti na izazov drugog igrača
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen
- **Opis osnovnog tijeka:**
 1. Igrač dobiva obavijest o pristiglom izazovu
 2. Igrač označava *Accept* ako prihvata izazov ili *Deny* ako odbija

UC13 - Izbor karata i borba

- **Glavni sudionik:** Igrač
- **Cilj:** Izabrati karte i boriti se s drugim igračem
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen i prihvatio je izazov ili je njegov izazov prihvaćen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire tri kartice među onima koje je prikupio
 2. Korisnik klikom na gumb *Fight!* ulazi u borbu
 3. Korisniku se prikazuje ishod borbe i ažurirani broj njegovih bodova

UC14 - Prijava nove lokacije

- **Glavni sudionik:** Napredni igrač
- **Cilj:** Prijaviti novu lokaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen i ima dovoljno visok *ELO score*
- **Opis osnovnog tijeka:**
 1. Igrač odabire opciju *Add location*
 2. Igrač unosi naziv i opis te dodaje fotografiju
- **Opis mogućih odstupanja:**
 - 1.a Igrač pokušava dodati lokaciju koja je previše udaljena od njegove trenutne lokacije
 1. Sustav obavještava igrača o neuspješnoj prijavi nove lokacije
 - 2.a Nepotpun i/ili neispravan unos obaveznih podataka
 1. Sustav obavještava igrača o neuspješnom unosu obaveznih podataka i vraća ga na stranicu za prijavu nove lokacije

UC15 - Pregled lokacija koje čekaju pregled

- **Glavni sudionik:** Igrač
- **Cilj:** Pregledati vlastite predložene lokacije koje čekaju pregled kartografa.
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen
- **Opis osnovnog tijeka:**
 1. Igrač se nalazi na stranici za predlaganje lokacija
 2. Igrač odabire filter neodobrenih lokacija
 3. Igraču se na karti prikazuju vlastite predložene lokacije koje čekaju pregled.

UC16 - Pregled odobrenih lokacija

- **Glavni sudionik:** Igrač
- **Cilj:** Pregledati vlastite predložene lokacije koje su odobrene.
- **Sudionici:** Baza podataka
- **Preduvjet:** Igrač je prijavljen
- **Opis osnovnog tijeka:**
 1. Igrač se nalazi na stranici za predlaganje lokacija
 2. Igrač odabire filter odobrenih lokacija
 3. Igraču se na karti prikazuju vlastite predložene lokacije koje su prošle provjeru

UC17 - Pregled prijedloga lokacija

- **Glavni sudionik:** Kartograf
- **Cilj:** Pregledati predložene lokacije koje nisu odobrene.
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na početnoj stranici
 2. Kartograf odabire filter neodobrenih lokacija
 3. Kartografu se na karti prikazuju predložene lokacije koje još nisu provjerene

UC18 - Uređivanje prijedloga lokacija

- **Glavni sudionik:** Kartograf
- **Cilj:** Urediti predloženu lokaciju koja nije odobrena.
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na početnoj stranici
 2. Kartograf odabire filter neodobrenih lokacija
 3. Kartograf odabire lokaciju koju želi urediti
 4. Kartograf uređuje podatke u prijedlogu lokacije
 5. Kartograf sprema promjene
 6. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Kartograf promijeni podatke o lokaciji, ali ne odabere opciju *Save changes*
 1. Sustav obavještava kartografa da nije spremio podatke prije izlaska iz prozora

UC19 - Odobravanje prijedloga lokacija

- **Glavni sudionik:** Kartograf
- **Cilj:** Odobriti predloženu lokaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na početnoj stranici
 2. Kartograf odabire filter neodobrenih lokacija
 3. Kartografu se na karti prikazuju predložene lokacije koje još nisu odobrene
 4. Kartograf odabire lokaciju koju želi odobriti
 5. Kartograf odobrava lokaciju klikom na gumb *Approve card*
 6. Baza podataka se ažurira

UC20 - Označavanje lokacije za provjeru

- **Glavni sudionik:** Kartograf
- **Cilj:** Označiti da je potrebna terenska provjera za predloženu lokaciju.
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na početnoj stranici
 2. Kartograf odabire filter neodobrenih lokacija
 3. Kartografu se na karti prikazuju predložene lokacije koje još nisu odobrene
 4. Kartograf odabire lokaciju koju želi označiti za terensku provjeru
 5. Kartograf označava da je potrebna terenska provjera lokacije klikom na gumb *Request on-site check*
 6. Baza podataka se ažurira

UC21 - Preuzimanje kartica za provjeru

- **Glavni sudionik:** Kartograf
- **Cilj:** Pregledati predložene lokacije koje trebaju terensku provjeru
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na stranici *On-site approval*
 2. Kartograf odabire filter koji će prikazati lokacije koje trebaju terensku provjeru
 3. Kartografu se na karti prikazuju predložene lokacije koje trebaju provjeru na terenu
 4. Kartograf klikom odabire lokaciju koju želi osobno obići, zatim klikne na gumb *Add to my route*
 5. Baza podataka se ažurira

UC22 - Uklanjanje lokacija

- **Glavni sudionik:** Kartograf
- **Cilj:** Ukloniti lokaciju s vlastite liste za provjeru na terenu
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen

- **Opis osnovnog tijeka:**

1. Kartograf se nalazi na stranici *On-site approval*
2. Kartograf odabire filter koji će prikazati lokacije koje je dodao na vlastitu listu za provjeru na terenu
3. Kartografu se na karti prikazuju njegove lokacije koje trebaju provjeru na terenu
4. Kartograf klikom odabire lokaciju koju želi ukloniti iz rute, a uklanja ju klikom na gumb *Remove*
5. Baza podataka se ažurira

UC23 - Generiranje rute

- **Glavni sudionik:** Kartograf
- **Cilj:** Generirati najkraću rutu za obilazak odabranih lokacija UC
- **Sudionici:** Baza podataka, OSRM servis
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na stranici *On-site approval*
 2. Kartografu se na karti prikazuju predložene lokacije koje trebaju provjeru na terenu
 3. Kartograf odabire filtre po želji - može prikazati lokacije koje je dodao na vlastitu listu za provjeru na terenu i/ili lokacije koje trebaju terensku provjeru, ali ih nitko nije zadužio
 4. Kartograf izražava zahtjev za generiranjem rute koja će obići sve odabrane lokacije klikom na gumb *Generate route*
 5. Sustav rješava problem trgovačkog putnika putem vanjskog servisa i prikazuje kartografu generiranu rutu

UC24 - Odobravanje lokacije

- **Glavni sudionik:** Kartograf
- **Cilj:** Odobriti lokaciju s vlastite liste za terensku provjeru
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na stranici *On-site approval*
 2. Kartografu se ispod karte prikazuje lista lokacija koje je dodao na vlastitu listu za provjeru na terenu

3. Kartograf pojedinu lokaciju odobrava klikom na gumb *Approve*
4. Baza podataka se ažurira

UC25 - Pregled osobno odobrenih lokacija

- **Glavni sudionik:** Kartograf
- **Cilj:** Kartograf želi pregledati popis lokacija koje je do sada odobrio
- **Sudionici:** Baza podataka
- **Preduvjet:** Kartograf je prijavljen
- **Opis osnovnog tijeka:**
 1. Kartograf se nalazi na stranici *My profile*
 2. Kartografu se na njegovom profilu prikazuje lista svih lokacija koje je odobrio

UC26 - Pregled podataka svih korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Dobiti uvid u osobne podatke svih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator se nalazi na stranici *All users*
 2. Administrator vidi popis svih korisnika i klikom na korisničko ime pojedinog korisnika biva odveden na stranicu gdje dobiva uvid u njegove osobne podatke

UC27 - Uređivanje podataka korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Urediti osobne podatke nekog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator se nalazi na stranici *All users*
 2. Administrator vidi popis svih korisnika i klikom na gumb za uređivanje pored korisničkog imena pojedinog korisnika biva odveden na stranicu gdje dobiva mogućnost izmjene njegovih osobnih podataka

- **Opis mogućih odstupanja:**

2.a Administrator na gumb iste funkcionalnosti može kliknuti na stranici gdje pregledava osobne podatke nekog korisnika (UC26)

1. Sustav odvodi administratora na stranicu za izmjenu osobnih podataka

UC28 - Privremeno isključenje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Privremeno onemogućiti pristup aplikaciji za nekog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator se nalazi na stranici *All users*
 2. Administrator vidi popis svih korisnika i klikom na gumb za isključenje isključuje korisnika na konačan vremenski period

UC29 - Pregled svih lokacija

- **Glavni sudionik:** Administrator
- **Cilj:** Dobiti uvid u podatke o svim lokacijama
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator se nalazi na stranici *All cards*
 2. Administrator vidi popis svih kartica i klikom na pojedinu biva odveden na stranicu gdje dobiva uvid u podatke o njoj

UC30 - Uređivanje podataka lokacije

- **Glavni sudionik:** Administrator
- **Cilj:** Urediti podatke neke lokacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator se nalazi na stranici *All cards*
 2. Administrator vidi popis svih lokacija i klikom na gumb za uređivanje pored pojedine kartice biva odveden na stranicu gdje dobiva mogućnost izmjene podataka o lokaciji

- **Opis mogućih odstupanja:**

2.a Administrator na gumb iste funkcionalnosti može kliknuti na stranici gdje pregledava podatke o lokaciji (UC29)

1. Sustav odvodi administratora na stranicu za izmjenu podataka o lokaciji

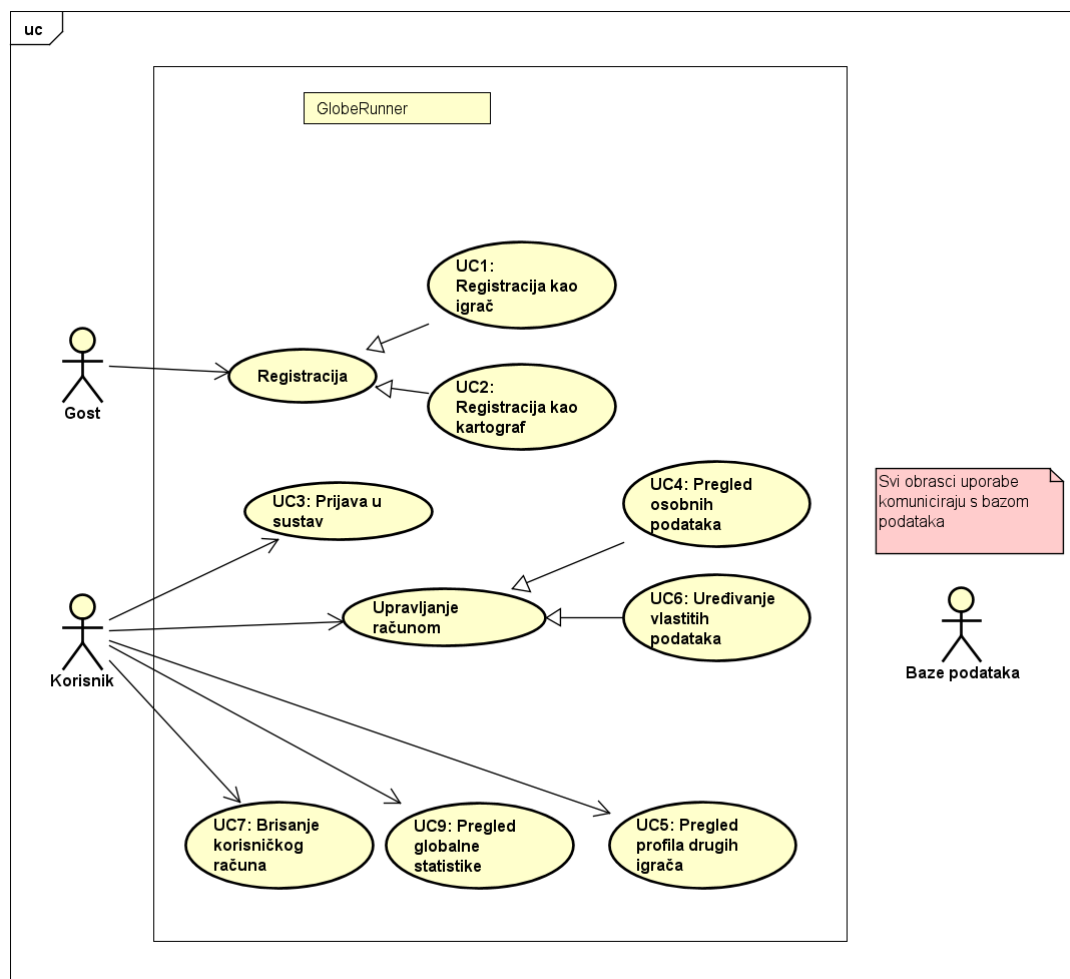
UC31 - Pregled neodobrenih kartografa

- **Glavni sudionik:** Administrator
- **Cilj:** Dobiti uvid u neodobrene registracije kartografa
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator se nalazi na stranici *Cartograph approvals*
 2. Administrator vidi popis svih neodobrenih registracija kartografa i klikom na pojedinu biva odveden na stranicu gdje dobiva uvid u podatke koje je kartograf predao u prijavi

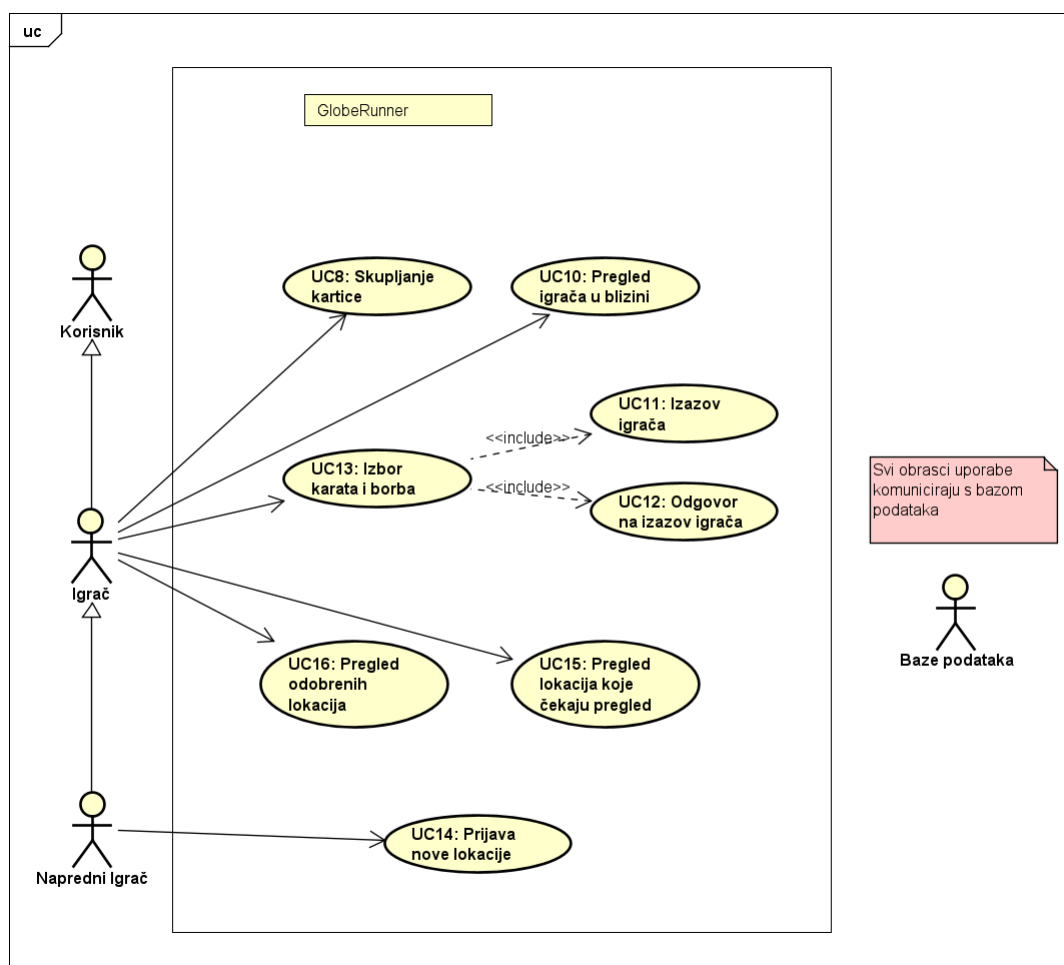
UC32 - Odobravanje kartografa

- **Glavni sudionik:** Administrator
- **Cilj:** Odobriti registraciju kartografa
- **Sudionici:** Baza podataka
- **Preduvjet:** Administrator je prijavljen
- **Opis osnovnog tijeka:**
 1. Administrator se nalazi na na kojoj ima uvid u podatke koje je kartograf predao u prijavi (UC31)
 2. Klikom na gumb *Approve* administrator odobrava registraciju kartografa

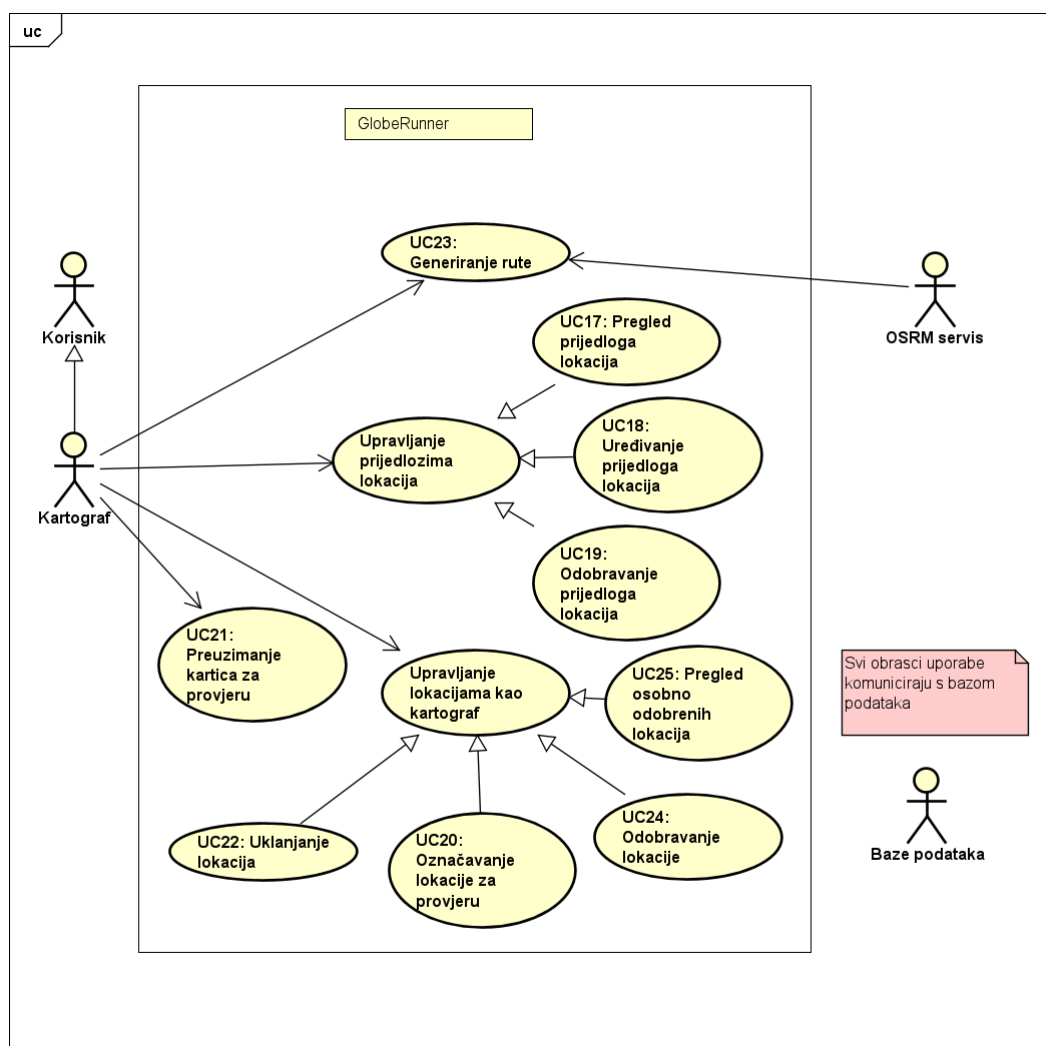
Dijagrami obrazaca uporabe



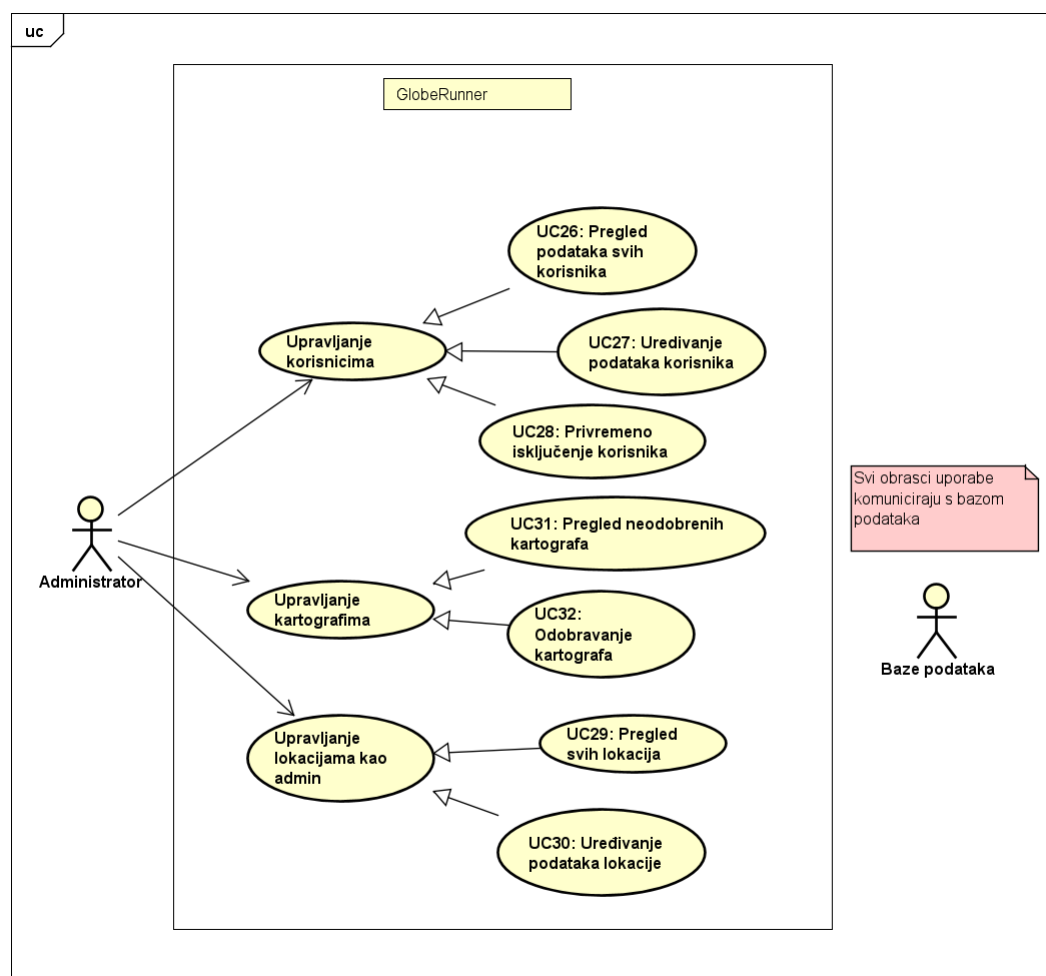
Slika 3.1: Dijagram obrasca uporabe, funkcionalnost gosta i korisnika



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost igrača i naprednog igrača



Slika 3.3: Dijagram obrasca uporabe, funkcionalnost kartografa

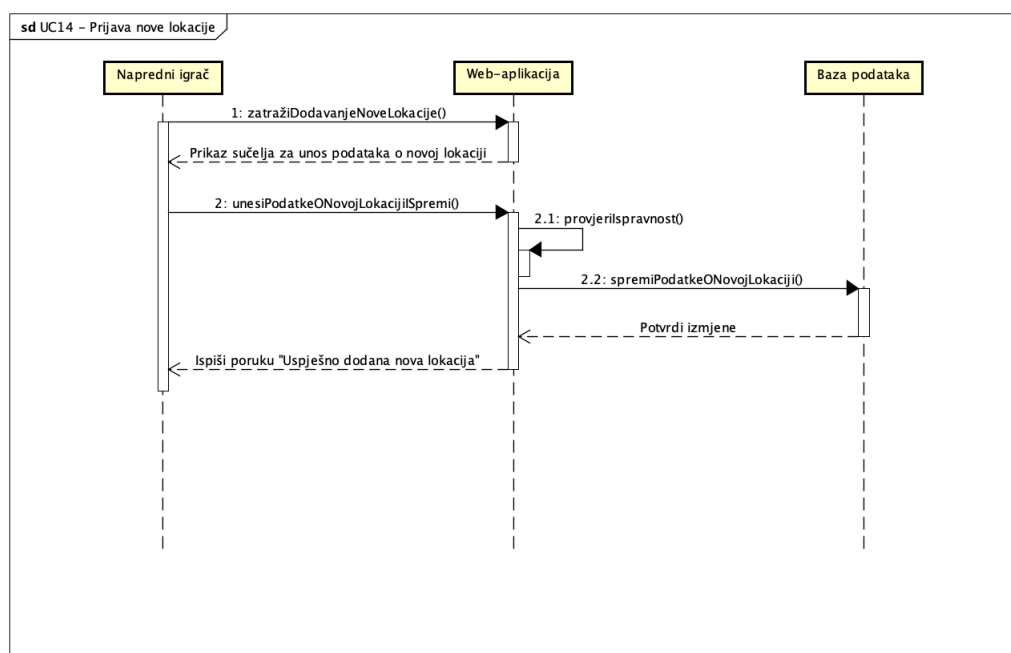


Slika 3.4: Dijagram obrasca uporabe, funkcionalnost admina

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC14 - Prijava nove lokacije

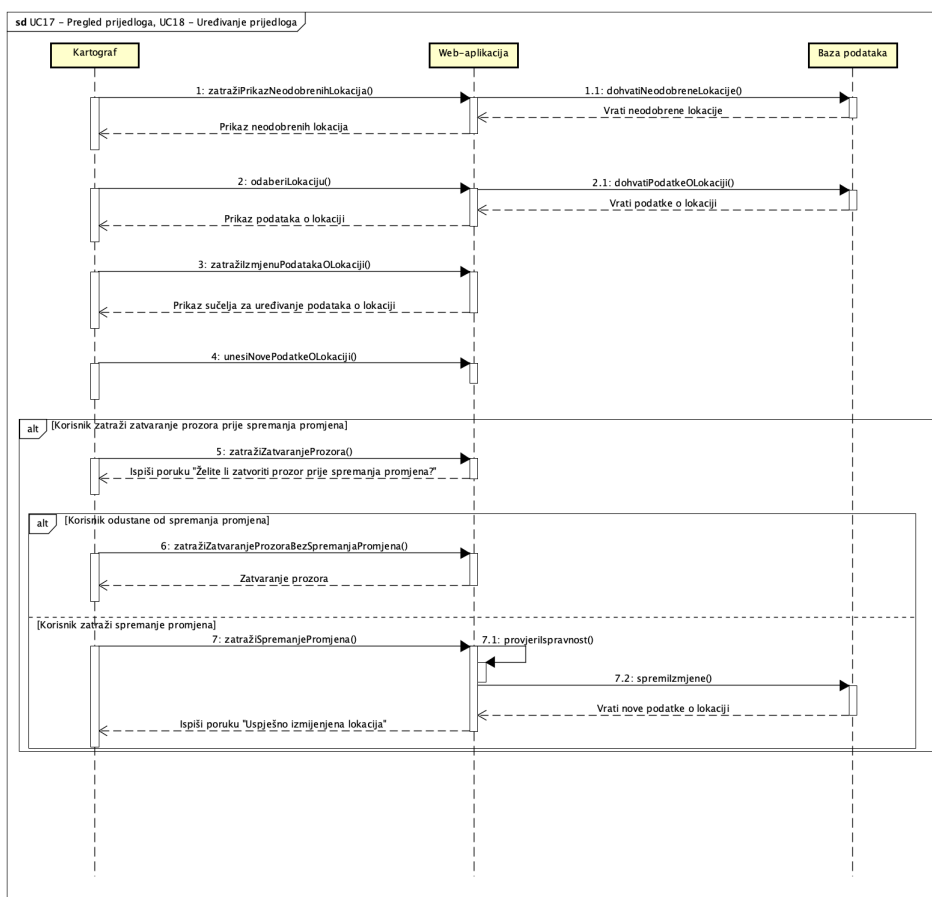
Napredni igrač šalje zahtjev za dodavanje kartice te mu poslužitelj prikazuje sučelje za unos podataka o novoj kartici/lokaciji. Napredni igrač unosi sve podatke o kartici koju želi dodati te sprema provjere. Poslužitelj tada provjerava ispravnost unesenih podataka te ih, u slučaju uspješne validacije, sprema u bazu. Nakon toga, poslužitelj naprednom igraču ispisuje poruku o uspješnom dodavanju nove kartice/lokacije.



Slika 3.5: Sekvencijski dijagram za UC14

Obrasci uporabe UC17 - Pregled prijedloga i UC18 - Uređivanje prijedloga

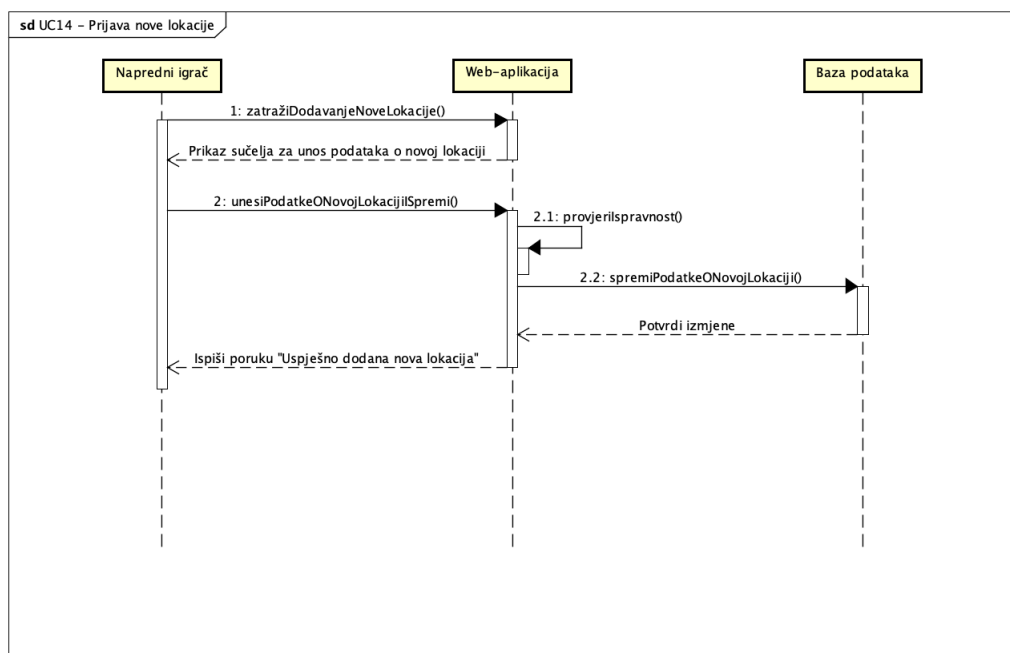
Kartograf šalje zahtjev za prikaz neodobrenih lokacija. Poslužitelj dohvaća neodobrene lokacije iz baze te ih prikazuje kartografu. Kartograf šalje zahtjev za izmjenu podataka o odabranoj lokaciji/kartici. Poslužitelj prikazuje kartografu sučelje za uređivanje postojećih podataka o neodobrenoj lokaciji. Kartograf unosi izmjene u prikazano sučelje. Ako kartograf pokuša zatvoriti prozor prije spremanja unesenih izmjena, poslužitelj ga upozorava da promjene nisu spremljene. Tada kartograf može odabrati svejedno zatvoriti prozor ili spremiti promjene. Ako odabere zatvoriti prozor prije spremanja, unesene izmjene neće biti spremljene u bazu. Ako odabere spremiti promjene, poslužitelj unesene podatke o lokaciji sprema u bazu te ispisuje poruku o uspješnoj izmjeni podataka.



Slika 3.6: Sekvencijski dijagram za UC17, UC18

Obrasci uporabe UC21 - Preuzimanje kartice za provjeru i UC23 - Generiranje rute

Kartograf šalje zahtjev za prikaz kartica koje trebaju terensku provjeru. Poslužitelj dohvaća kartice koje trebaju terensku provjeru iz baze te ih prikazuje kartografu. Kartograf šalje zahtjev za preuzimanjem kartice za provjeru. Poslužitelj u bazu podataka bilježi da je kartica preuzeta za provjeru te bilježi koji kartograf ju je preuzeo. Kartograf prije generiranja rute još jednom šalje zahtjev za prikaz neodobrenih lokacija. Poslužitelj ih dohvaća iz baze podataka te prikazuje kartografu. Kartograf šalje zahtjev za generiranje rute. Poslužitelj iz baze dohvaća kartice koje je taj kartograf preuzeo za provjeru te ih šalje OSRM servisu i zahtjeva generiranje rute. OSRM servis poslužitelju vraća generiranu rutu, a poslužitelj ju prikazuje kartografu.



Slika 3.7: Sekvencijski dijagram za UC21, UC23

3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Sustav kao valutu koristi EURO
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS
- Rukovanje osobnim podacima treba biti usklađeno s važećim direktivama
- Sustav treba koristiti GPS za pozicioniranje igrača i kartica na karti
- Sustav treba koristiti OSRM servis za rješavanje problema trgovačkog putnika

4. Arhitektura i dizajn sustava

Korištena arhitektura se može podijeliti na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka

Web preglednik program je koji omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Svaki internetski preglednik ima i ulogu prevoditelja, odnosno on interpretira kod kojim je stranica napisana i tako realizira grafičko sučelje i neke jednostavnije funkcionalnosti. Korisnik putem web preglednika šalje zahtjeve web poslužitelju.

Web poslužitelj centralna je komponenta za rad web aplikacije. Njegova je glavna zadaća komunikacija klijenta s aplikacijom. Komunikacija se odvija preko HTTP (engl. *Hyper Text Transfer Protocol*) protokola koji se koristi u prijenosu informacija na webu. Poslužitelj je onaj koji pokreće web aplikaciju, zaprima korisničke zahtjeve i prosljeđuje ih aplikaciji na obradu. Web poslužitelj koji smo odabrali za našu web aplikaciju je DigitalOcean. Na njega smo postavili naše *frontend* i *backend* komponente, a kao poslužitelj baze podataka odlučili smo koristiti Render.

Korisnik koristi web aplikaciju za obrađivanje željenih zahtijeva. Web aplikacija obrađuje zahtjev te ovisno o zahtjevu, pristupa bazi podataka nakon čega preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku.

Programski jezik kojeg smo odabrali za izradu naše web aplikacije je Python zajedno s Flask radnim okvirom te programski jezik TypeScript s React radnim okvirom u sklopu kojeg smo koristili MUI library. Odabrano razvojno okruženje je Visual Studio Code. Arhitektura sustava temeljit će se na MVC (Model-View-Controller) konceptu. MVC koncept podržan je od strane Flask radnog okvira i kao takav ima gotove predloške koji nam olakšavaju razvoj web aplikacije.

Karakteristika MVC koncepta je nezavisan razvoj pojedinih dijelova aplikacije što za posljedicu ima jednostavnije ispitivanje kao i jednostavno razvijanje i dodavanje novih svojstava u sustav. MVC koncept sastoji se od:

- **Model** – središnja komponenta sustava. Predstavlja dinamičke strukture podataka, neovisno o korisničkom sučelju. Izravno upravlja podacima, logikom i pravilima aplikacije te prima ulazne podatke od Controllera
- **View** – bilo kakav prikaz podataka, poput grafa. Mogući su različiti prikazi iste informacije poput grafičkog ili tabličnog prikaza podataka.
- **Controller** – prima ulaze i prilagođava ih za prosljeđivanje Modelu ili Viewu. Upravlja korisničkim zahtjevima i temeljem njih izvodi daljnju interakciju s ostalim elementima sustava.

4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olakšava modeliranje stvarnog svijeta. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- User
- Cartographer
- Player
- Admin
- Card
- Inventory
- Fight
- Challenge

4.1.1 Opis tablica

User Ovaj entitet sadržava sve važne informacije o korisniku aplikacije, ali je apstraktan. Sadrži attribute: userID, username, name, email, profilePhoto, password, confirmed, signedIn, banned. Ovaj entitet je povezan s entitetima Player i Cartographer koji su specijalizacije entiteta User.

User	Type	Description
userID	VARCHAR	jedinstveni identifikator korisnika
username	VARCHAR	korisničko ime
name	VARCHAR	ime i prezime korisnika
email	VARCHAR	email korisnika
profilePhoto	VARCHAR	fotografija korisnika
password	VARCHAR	lozinka korisnika
confirmed	BOOLEAN	je li korisnik potvrđen putem emaila
signedIn	BOOLEAN	je li korisnik ulogiran

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

User	Type	Description
banned	BOOLEAN	je li korisniku zabranjen pristup

Cartographer Ovaj entitet sadržava sve važne informacije o korisniku aplikacije koji je kartograf. Sadrži attribute: userID, IBAN, document i verified. Ovaj entitet je specijalizacija entiteta User te je u vezi *One-to-Many* s entitetom Card preko atributa cartographerID.

Cartographer	Type	Description
userID	VARCHAR	jedinstveni identifikator korisnika (user.userID)
IBAN	VARCHAR	IBAN kartografa
document	VARCHAR	fotografija osobne iskaznice kartografa
verified	BOOLEAN	je li administrator odobrio kartografa

Player Ovaj entitet sadržava sve važne informacije o korisniku aplikacije koji je igrač. Sadrži attribute: userID, advanced, eloScore, playerLocation i challengeable. Ovaj entitet je specijalizacija entiteta User te je u vezi *One-to-Many* s entitetom Inventory preko atributa userID te je u vezi *One-to-Many* s entitetom Card preko atributa authorUserID. Također je u vezi *One-to-Many* s entitetom Challenge preko atributa challengerUserID, tj. victimUserID kao što je i u *One-to-Many* vezi s entitetom Fight preko atributa player1UserID, tj. player2UserID.

Player	Type	Description
userID	VARCHAR	jedinstveni identifikator korisnika (user.userID)
advanced	BOOLEAN	razina ovlasti igrača može biti ordinary ili advanced
eloScore	INT	ELO score igrača
playerLocation	VARCHAR	aktivna lokacija korisnika
challengeable	BOOLEAN	je li korisnika moguće izazvati na borbu

Admin Ovaj entitet sadržava sve važne informacije o adminu. Sadrži attribute: adminID, username, name, email, password.

Admin	Type	Description
adminID	INT	jedinstveni identifikator admina
username	VARCHAR	korisničko ime
name	VARCHAR	ime i prezime admina
email	VARCHAR	email admina
password	VARCHAR	lozinka admina

Card Ovaj entitet sadržava sve važne informacije o kartici. Sadrži attribute: cardID, cardLocation, locationPhoto, title, description, cardStatus, authorUserID, cartographerID. Ovaj entitet je u vezi *One-to-Many* s entitetom Inventory preko atributa cardID te je u *One-to-Many* vezi s entitetom Fight preko atributa cardIDxy gdje je x element 1, 2 i y je element 1, 2, 3. Card je također u *Many-to-One* vezi s entitetom Player preko atributa authorUserID te je u *Many-to-One* vezi s entitetom Cartographer preko atributa cartographerID.

Card	Type	Description
cardID	INT	šifra kartice
cardLocation	VARCHAR	lokacija kartice
locationPhoto	VARCHAR	fotografija kartice
title	VARCHAR	naziv kartice
description	VARCHAR	opis kartice
cardStatus	ENUM	status kartice može biti: submitted, unclaimed, claimed, verified
authorUserID	VARCHAR	koji igrač je predložio dodavanje nove kartice u igru (user.userID)
cartographerID	VARCHAR	koji kartograf je odobrio zahtjev za izradu nove kartice (user.userID)

Inventory Ovaj entitet sadržava sve važne informacije o karticama koje pojedini igrači posjeduju. Sadrži attribute: userID, cardID i strength. Ovaj entitet je u *Many-to-One* vezi s entitetom Player preko atributa userID te u *Many-to-One* vezi s entitetom Card preko atributa cardID.

Inventory	Type	Description
userID	VARCHAR	identifikacijski broj korisnika koji posjeduje karticu (user.userID)
cardID	INT	šifra kartice
strength	INT	jačina kartice

Fight Ovaj entitet sadržava sve važne informacije o borbi između dva igrača. Sadrži attribute: fightID, player1UserID, player2UserID, player1Ready, player2Ready, cardID11, cardID12, cardID13, cardID21, cardID22, cardID23, points1, points2, fightTimestamp. Ovaj entitet je u *Many-to-One* vezi s entitetom Player preko atributa userID te je i u *Many-to-One* vezi s entitetom Card preko atributa cardID.

Fight	Type	Description
fightID	INT	šifra borbe
player1UserID	VARCHAR	pobjednik borbe (user.userID)
player2UserID	VARCHAR	gubitnik borbe (user.userID)
player1Ready	BOOLEAN	je li prvi igrač spreman za borbu
player2Ready	BOOLEAN	je li drugi igrač spreman za borbu
cardID11	INT	prva kartica igrana od strane prvog igrača (card.cardID)
cardID12	INT	druga kartica igrana od strane prvog igrača (card.cardID)
cardID13	INT	treća kartica igrana od strane prvog igrača (card.cardID)
cardID21	INT	prva kartica igrana od strane drugog igrača (card.cardID)

Nastavljeno na idućoj stranici

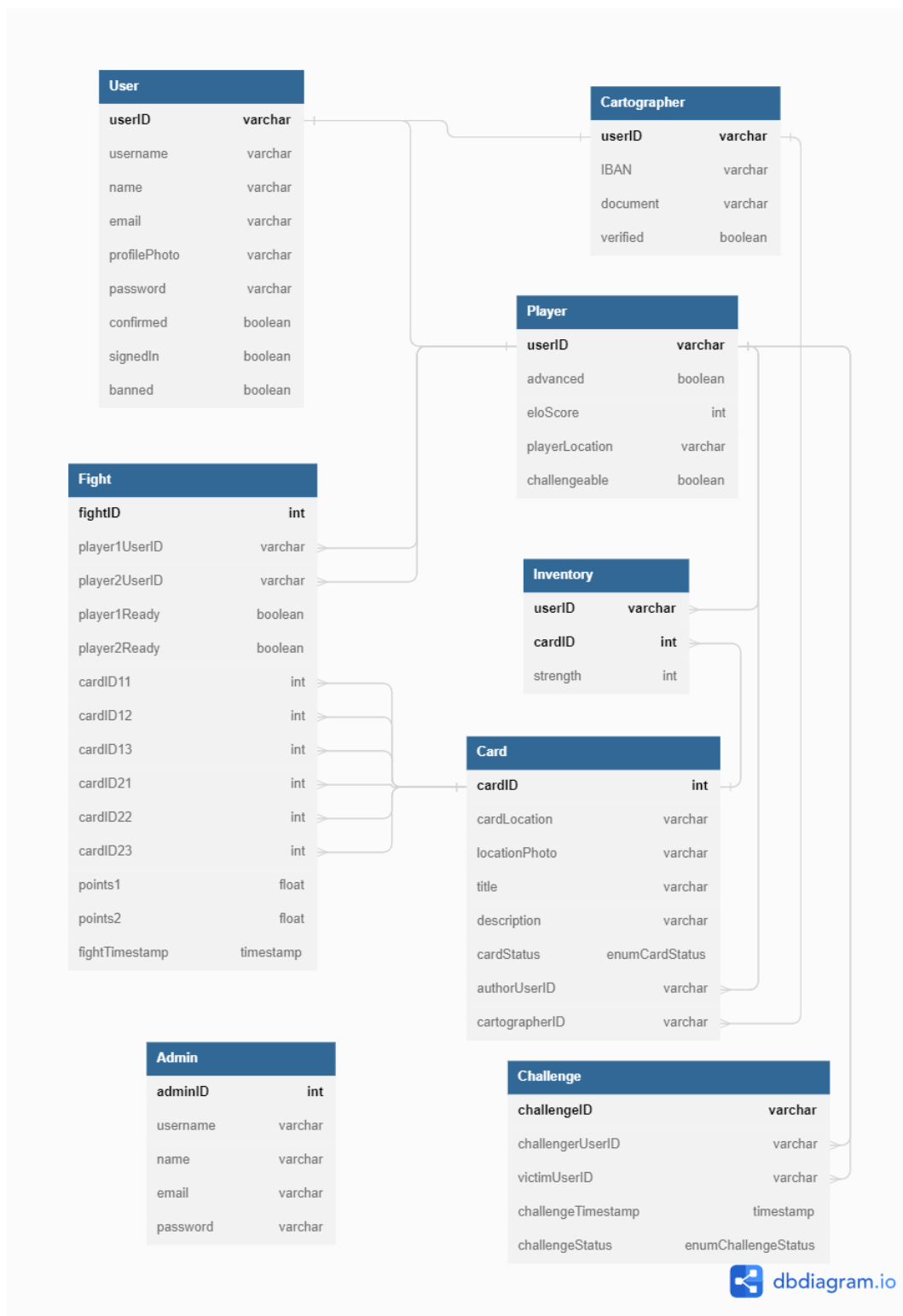
Nastavljeno od prethodne stranice

Fight	Type	Description
cardID22	INT	druga kartica igrana od strane drugog igrača (card.cardID)
cardID23	INT	treća kartica igrana od strane drugog igrača (card.cardID)
points1	FLOAT	broj bodova prvog igrača
points2	FLOAT	broj bodova drugog igrača
fightTimestamp	TIMESTAMP	trenutak u kojem je provedena borba

Challenge Ovaj entitet sadržava sve važne informacije o izazovu između dva igrača. Sadrži attribute: challengeID, challengerUserID, victimUserID, challengeTimestamp, challengeStatus. Ovaj entitet je u *Many-to-One* s entitetom Player preko atributa challengerUserID, tj. victimUserID.

Challenge	Type	Description
challengeID	VARCHAR	šifra izazova
challengerUserID	VARCHAR	igrač koji šalje zahtjev za borbu (user.userID)
victimUserID	VARCHAR	igrač koji je izazvan na borbu (user.userID)
challengeTimestamp	TIMESTAMP	trenutak u kojem je nastao izazov
challengeStatus	ENUM	izazov može biti: pending, accepted, declined, went too far

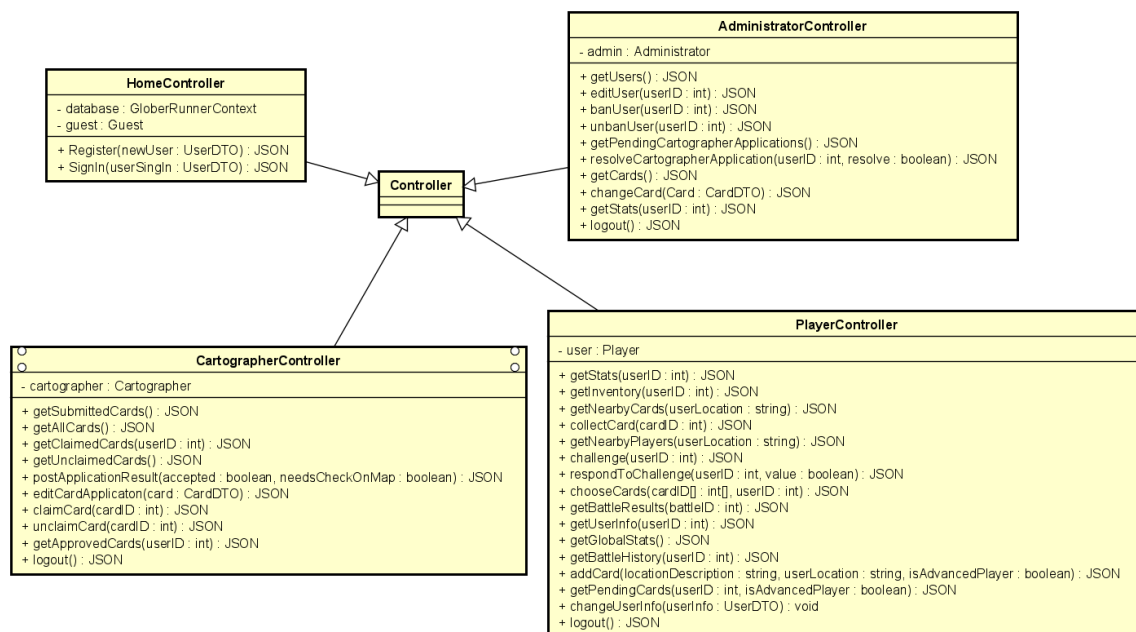
4.1.2 Dijagram baze podataka



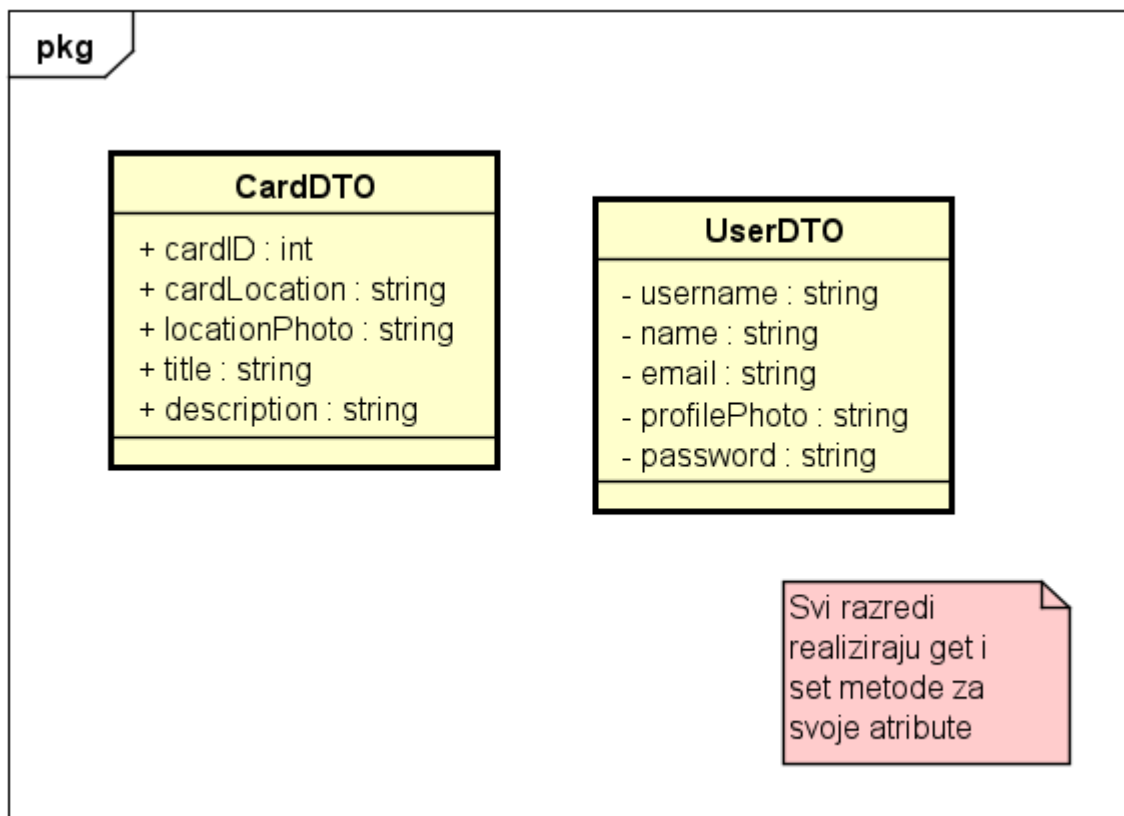
Slika 4.1: Dijagram baze podataka

4.2 Dijagram razreda

Slike 4.2, 4.3 i 4.4 prikazuju razrede u *backend* dijelu implementirane MVC arhitekture. S ulogom obrade dolaznih HTTP zahtjeva implementirane su metode u razredima koji nasljeđuju klasu *Controllers*, prikazani na slici 4.2. Pri posluživanju sadržaja i isporuci podataka razredi sa slike 4.2 koriste i izmjenjuju *Data transfer objects*, čiji su razredi prikazani na slici 4.3, koje naposljetku isporučuju u JSON formatu. Za dohvaćanje i implementiranje interakcija među entitetima poput iza-zova i borbi služe razredi sa slike 4.4.

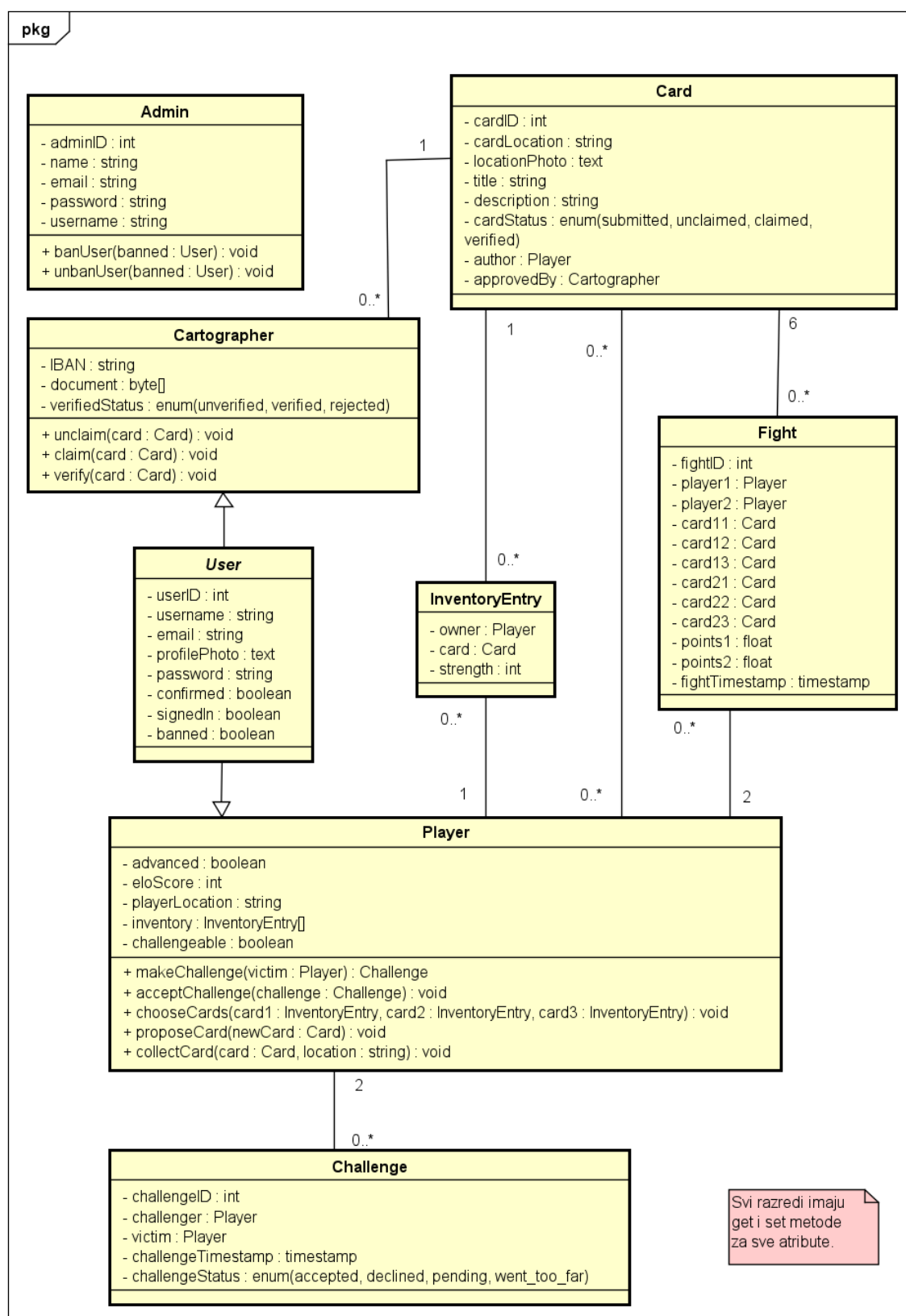


Slika 4.2: Dijagram razreda - dio Controllers



Slika 4.3: Dijagram razreda - dio Data transfer objects

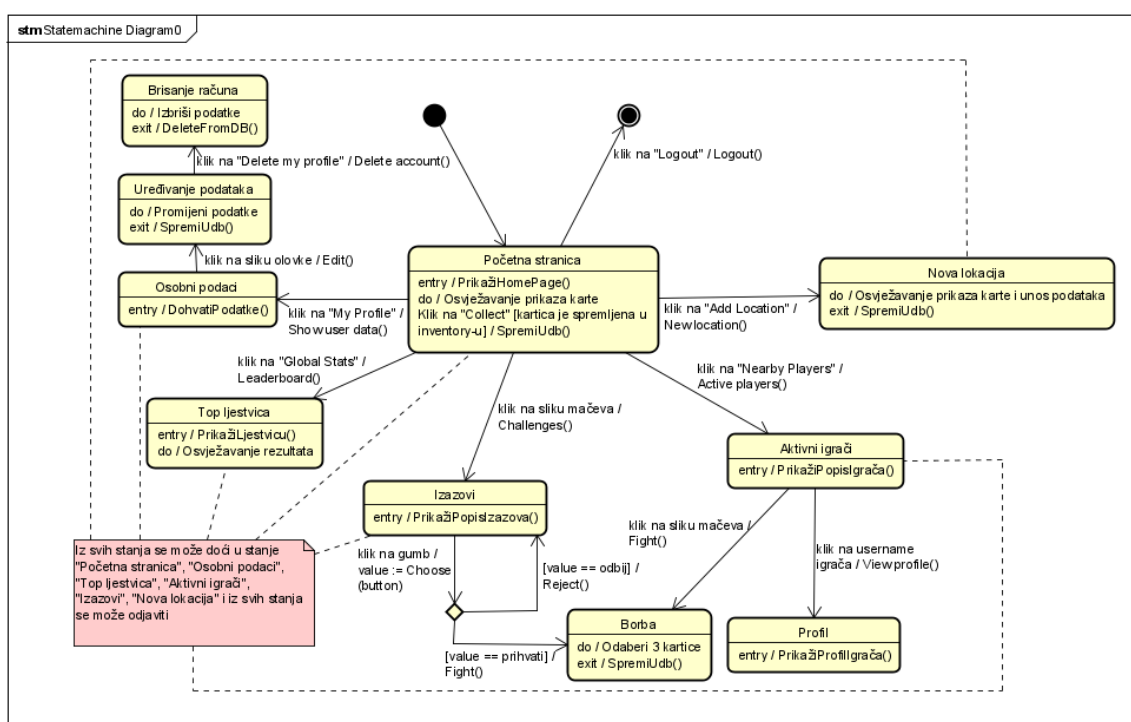
Dijagram koji prikazuje modele sadrži klase koje preslikavaju stvarne entitete u objektnu paradigmu, a uz to omogućuju i komunikaciju s bazom podataka. Razred *Admin* predstavlja administratora koji ima najveće ovlasti i mogućnosti izmjene podataka. *User* je pak apstraktna klasa koja modelira korisnika nastalog registracijom, konkretno to mogu biti *Cartographer* i *Player*, a oni su klase koje nasljeđuju klasu *User* i pružaju specifične funkcionalnosti i komunikaciju s pripadnim relacijama u bazi podataka. Razred *Card* pruža vezu između pripadne relacije u bazi podataka i objektnog modela, a sadrži podatke bitne za prikaz i smještanje kartice u prostoru. Razred *InventoryEntry* povezuje karticu s igračem dodajući joj pritom snagu koja skalira učinak kartice u borbi. Razredom *Challenge* grupiraju se podatci vezani uz izazov koji je jedan igrač uputio drugom, a koji može rezultirati borbom. Klasa *Fight* predstavlja skup podataka koji je potreban za razrješenje pobjednika neke borbe.



Slika 4.4: Dijagram razreda - dio Models

4.3 Dijagram stanja

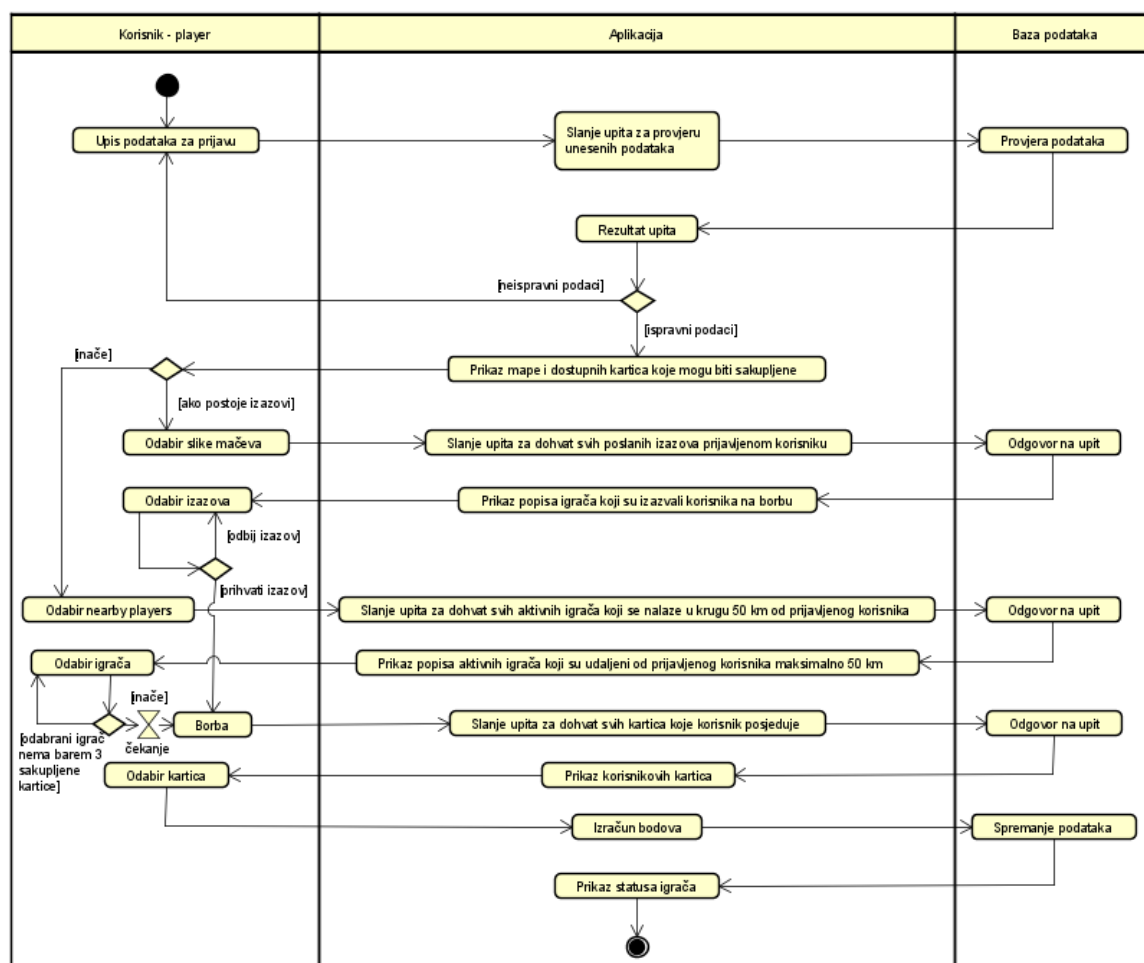
Dijagram stanja sa slike 4.5 prikazuje sva moguća stanja u kojima se napredni igrač može zateći zajedno s njihovim prijelazima i akcijama. Poslije prijave, našem korisniku se otvora početna stranica na kojoj može pretraživati kartu i birati kartice. Odabirom slike mačeva u zaglavlju aplikacije dolazi na stranicu s popisom izazova koje može prihvatiti ili odbiti. Također, igrač ima opciju padajućeg izbornika sa sljedećim ponuđenim stanjima: "Osobni podaci", "Top ljestvica", "Aktivni igrači" i "Nova lokacija". Klikom na gumb "My Profile" učita mu se stranica s njegovim osobnim podacima, košarica prikupljenih kartica i statistički graf koji prikazuje odnos pobjeda i poraza u borbama. Svoje osobne podatke igrač može promijeniti ili u stanju uređivanja odabrati opciju brisanja računa. Klikom na gumb "Global Stats" prikazuje se rank naprednog igrača i ljestvica najuspješnijih po bodovima. Zatim, klikom na "Nearby Players" korisnik dobiva popis aktivnih igrača u svojoj blizini kojima može vidjeti profil ili poslati zahtjev za borbu. Samo napredni igrač ima mogućnost predlaganja novih kartica u igri klikom na "Add Location" gdje se ukazuje forma za ispunjavanje i karta.



Slika 4.5: Dijagram stanja

4.4 Dijagram aktivnosti

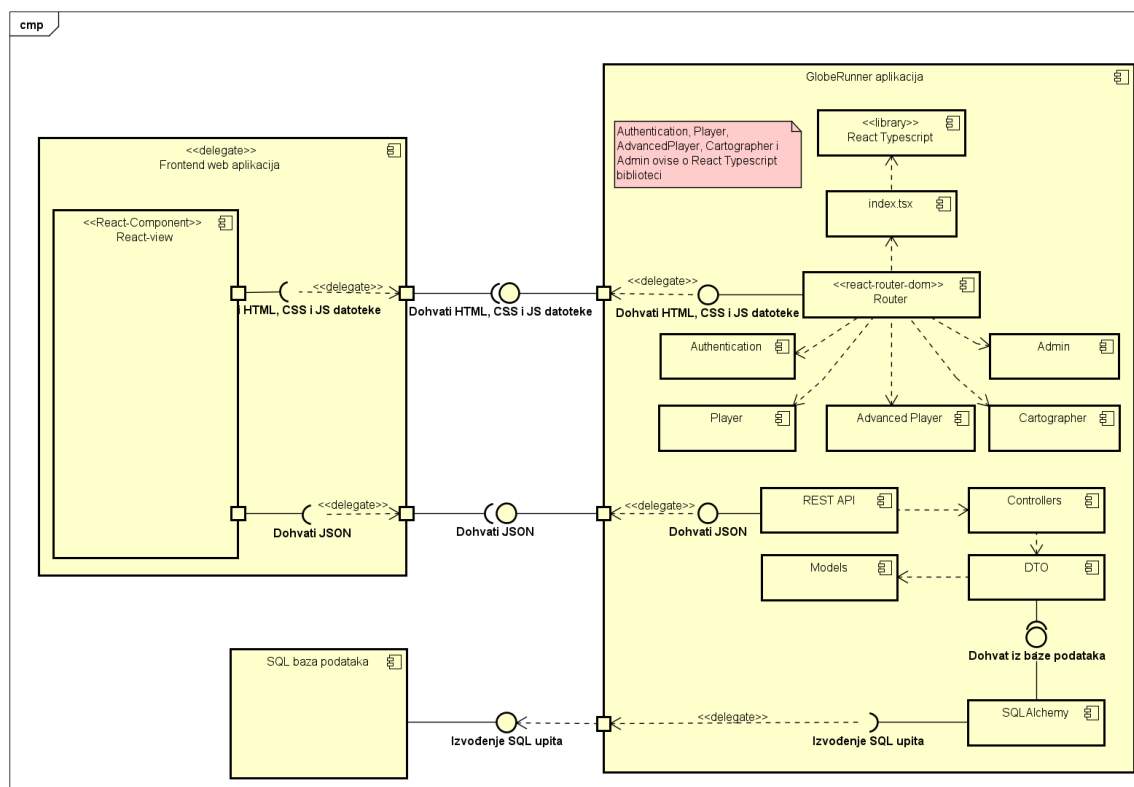
Slika 4.6 prikazuje dijagram aktivnosti koji svojim nizom akcija pobliže objašnjava značajni dio našeg sustava. Glavna akcija u našoj aplikaciji je borba pa smo modelirali sva ponašanja koja vode do kreiranja i odvijanja same borbe. Korisnik se najprije mora prijaviti u sustav te ako su njegovi podaci ispravni, prikaže mu se početna stranica s mapom i karticama koje može sakupiti. Ako njegovi podaci nisu ispravni, korisnika se preusmjerava nazad na login stranicu. Nakon uspješne prijave, pretpostavimo li da je korisnik već sakupio minimalno 3 kartice, on odabire želi li izazvati aktivne igrače u blizini ili prihvatiti izazov drugih igrača, ukoliko oni postoje. Za prvu opciju korisnik bira polje "Nearby players" u padajućem izborniku "Menu" i dobiva popis dostupnih igrača, dok za drugu opciju, odabire ikonu prekriženih mačeva i vidi popis izazova. Da bi korisnik pokrenuo borbu mora ili prihvatiti neki od izazova ili odabrati nekog igrača koji ima uvjete za mogućnost igranja borbe te pričekati da izazvani igrač prihvati izazov. Kada je postupak borbe pokrenut, igrači biraju kartice kojima se žele boriti i aplikacija računa iznos njihovih bodova. Na kraju, svi podaci se spremaju u bazu radi zapisivanja statistike i aplikacija vraća status, odnosno je li korisnik pobijedio ili izgubio u borbi.



Slika 4.6: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti koji se nalazi na slici 4.7 prikazuje strukturu cijele aplikacije. *Frontend* komponenti pristupa se putem sučelja kojim se dohvaćaju HTML, CSS i JS datoteke koje realiziraju korisničko sučelje za korištenje aplikacijom. To je postignuto nizom TypeScript datoteka koje od kojih svaka čini pojedinu React komponentu koju smo mi realizirali koristeći razne gotove React i Material UI komponente. Prije isporuke aplikacije korisniku, TypeScript programski kod prevodi se u JavaScript kod. Sve TypeScript datoteke ovise o React biblioteci. Kako bi korisniku pružila funkcionalnost, React aplikacija koja je isporučena korisniku koristi sučelje za dohvat JSON podataka kako bi pristupila GlobeRunner aplikaciji koju prema REST načelu realizira Flask razvojno okruženje u korijenskoj ruti flask aplikacije. Od tamo, a na temelju navedenih ruta, preusmjeravaju se zahtjevi koje aplikacija zaprimi kako bi ih obradila konkretna podkomponenta. Flask uz alat SQLAlchemy ostvaruje *Object Relational Mapping* koristeći implementaciju klasa u komponenti Models pomoću kojih su stvorene tablice u bazi podataka. SQLAlchemy ostvaruje komunikaciju s bazom i dohvaća potrebne podatke izvršavanjem SQL upita.



Slika 4.7: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Tim je za potrebe komunikacije koristio WhatsApp¹ aplikaciju i Discord² platformu. WhatsApp je služio za osnovnu komunikaciju poput dogovora sastanaka i prenošenje najbitnijih informacija o napretku projekta. Za online sastanke se koristio Discord jer je omogućavao jednostavno spajanje u glasovni poziv i dijeljenje ekrana drugim članovima tima, a uz to nudi i mogućnost organiziranja različitih kanala kako bi komunikacija među podtimovima (*frontend*, *backend*) bila organizirana. Korišteni sustav za upravljanje izvornim kodom je Git³, a udaljeni repozitorij je postavljen na web platformu GitLab⁴.

Za pisanje dokumentacije je bio korišten Overleaf⁵ što je cloud-based LaTeX editor zahvaljujući kojem je dokumentacija s aktualnim promjenama bila dostupna svakom članu bez dodatnih instalacija. Kao provjeru pravopisa koristili smo online alat ispravi.me⁶. UML dijagrame izrađivali smo u alatu Astah UML⁷, a dijagram baze podataka izrađen je u alatu dbdiagram.io⁸.

Odabrano razvojno okruženje za našu aplikaciju bio je Visual Studio Code⁹ koji ima integrirani terminal i podršku za Git te uključuje podršku za debugiranje. VS Code je odabran zbog toga što je dostupan za sve operacijske sustave koje smo koristili za rad na projektu - Windows, Linux i macOS.

U pisanju aplikacije korišten je Flask¹⁰ kao radni okvir za programski jezik Python¹¹. Uz to je korišten i alat pipenv¹² koji pomaže pri stvaranju virtualnog

¹<https://www.whatsapp.com/>

²<https://discord.com/>

³<https://git-scm.com/>

⁴<https://about.gitlab.com/>

⁵<https://www.overleaf.com/>

⁶<https://ispravi.me/>

⁷<https://astah.net/products/astah-uml/>

⁸<https://dbdiagram.io/home>

⁹<https://code.visualstudio.com/>

¹⁰<https://flask.palletsprojects.com/en/2.2.x/>

¹¹<https://www.python.org/>

¹²<https://pipenv.pypa.io/en/latest/>

okruženja za *backend* dio aplikacije te alat SQLAlchemy¹³ koji olakšava korištenje baze podataka i SQL-a u *backendu*. Za izradu *frotenda* koristio se programski jezik TypeScript¹⁴ koji je zapravo omotač za programski jezik JavaScript¹⁵ te ima dodatne funkcionalnosti u odnosu na JavaScript. Također, koristio se React¹⁶ koji je JavaScript library, ali ga se često smatra i radnim okvirom, iako to tehnički nije. Material UI¹⁷, tj. MUI je jedan od libraryja koji se uvelike koristi u projektu zbog svoje jednostavnosti i povezanosti s Reactom. On nudi i omogućava korištenje već gotovih komponenti te olakšava stvaranje novih.

Za prikazivanje mapa u aplikaciji koristio se Leaflet¹⁸, JavaScript library za interaktivne mape, a za generiranje ruta kartografa koristio se OSRM¹⁹, open source alat koji generira najkraću rutu za obilazak željenih točaka.

Baza podataka nalazi se na cloud poslužitelju Render²⁰, a *frotend* i *backend* postavljeni su na web poslužitelju DigitalOcean²¹.

¹³<https://www.sqlalchemy.org/>

¹⁴<https://www.typescriptlang.org/>

¹⁵<https://www.javascript.com/>

¹⁶<https://reactjs.org/>

¹⁷<https://mui.com/>

¹⁸<https://leafletjs.com/>

¹⁹<https://project-osrm.org/>

²⁰<https://render.com/>

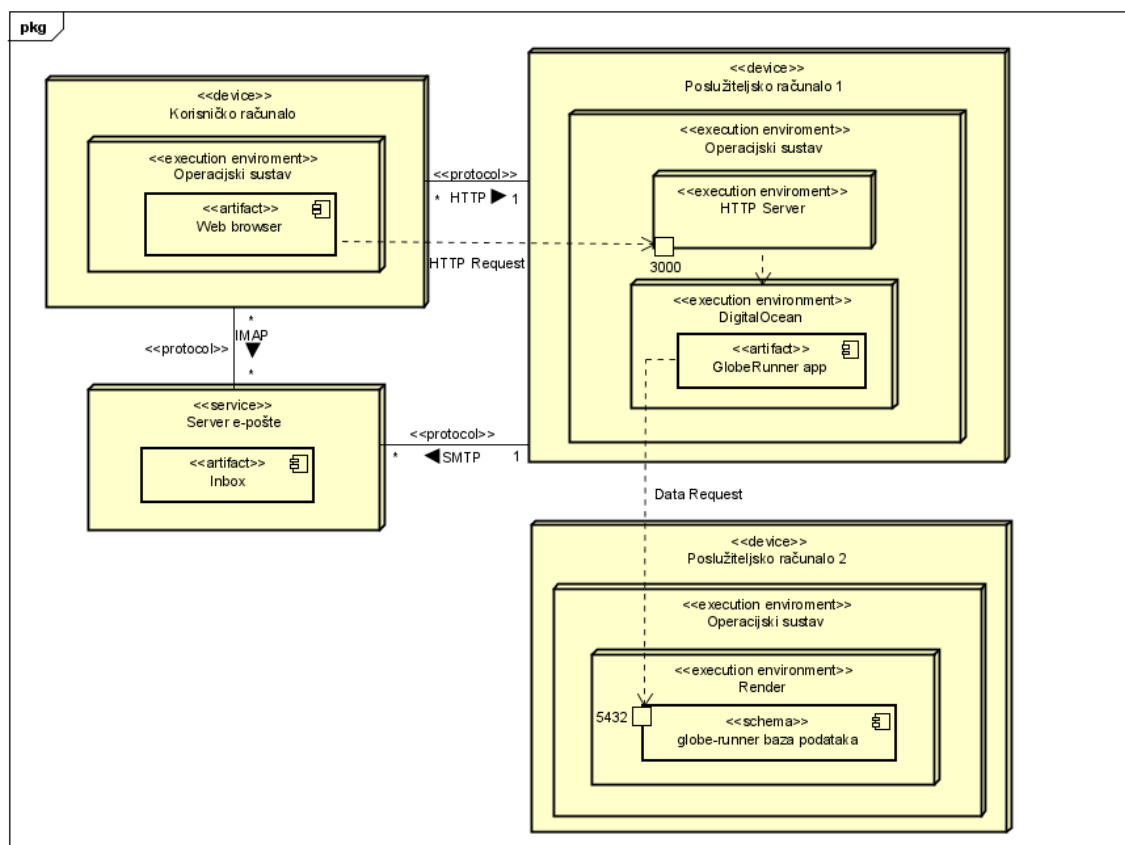
²¹<https://www.digitalocean.com/>

5.2 Ispitivanje programskog rješenja

Sve komponente i funkcionalnosti testirali smo putem korisničkog sučelja i Postmana.

5.3 Dijagram razmještaja

Na slici 5.1 prikazan je specifikacijski dijagram razmještaja. Iz njega lako možemo iščitati topologiju našeg sustava te odnos između sklopovskih i programskih komponenti. Kao što vidimo, korisnici (administratori, igrači i kartografi) internet-skim preglednikom pristupaju web-aplikaciji. Komunikacija između korisničkog i poslužiteljskog računala odvija se preko HTTP veze. Web browser šalje HTTP serveru *HTTP request* na port 3000. HTTP server ovisan je o web-poslužitelju aplikacije, u našem slučaju DigitalOcean, na kojem se nalazi GlobeRunner. Svi potrebni podaci za ispravan rad aplikacije spremaju se u globe-runner bazu podataka koja je "deployana" na Render poslužitelju. Dohvaćanje tih podataka odvija se tako da aplikacija šalje *Data request* na port 5432. Također, u slučaju registracije novog korisnika (igrača ili kartografa) aplikacija SMTP protokolom šalje email s linkom za potvrdu kreiranja profila. Pretpostavljamo da tada korisnici taj isti mail primaju IMAP protokolom dok se spajaju na poslužitelja e-pošte.



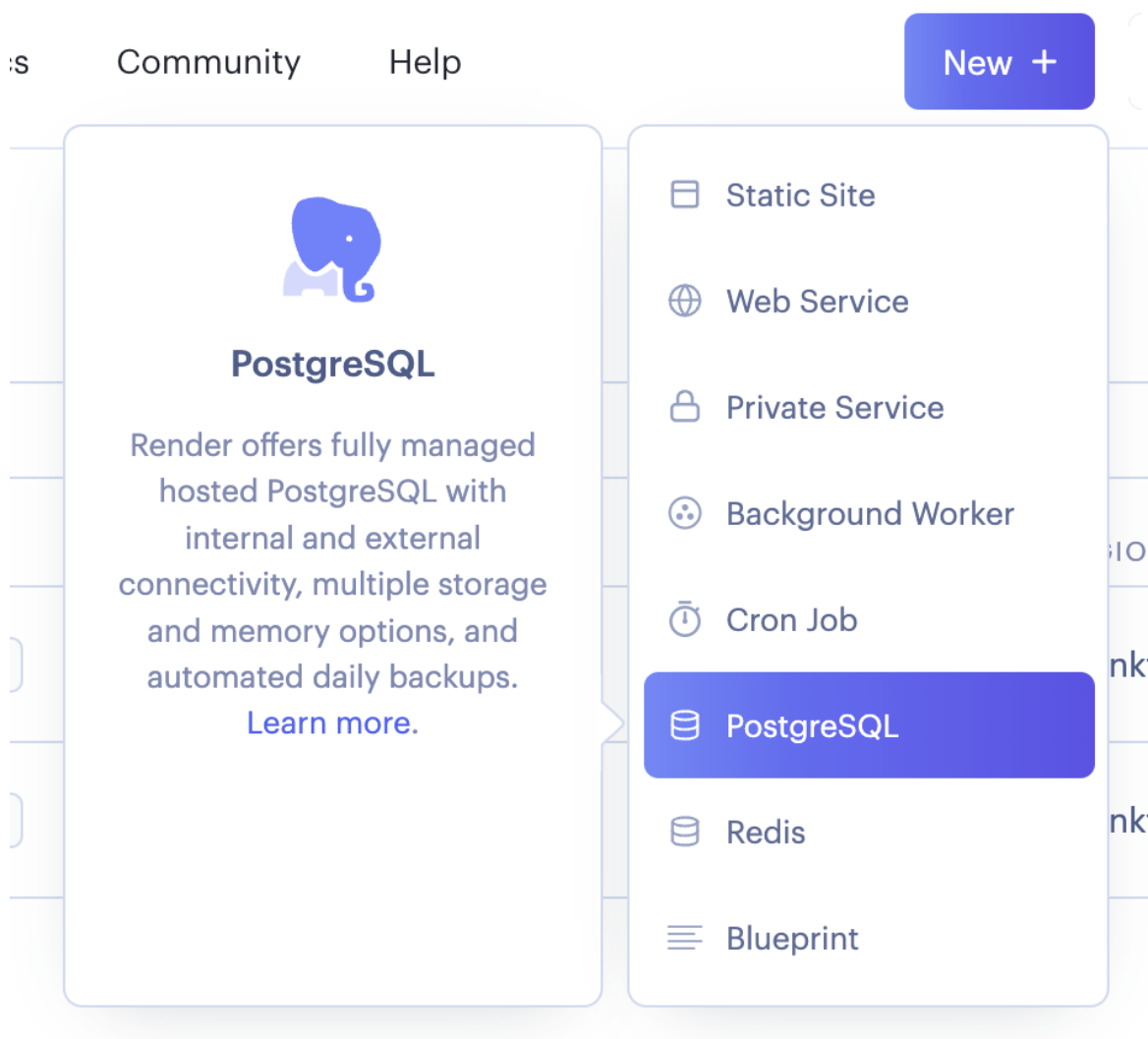
Slika 5.1: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Pokretanje aplikacije na vlastitom računalu zahtjeva postavljanje PostgreSQL baze za podatke aplikacije te Redis baze za session-e. Ovdje će postupak za to biti opisan koristeći usluge Render servisa s obzirom da su sve usluge potrebne za ovo besplatne. Cijelo pokretanje projekta radimo preko Docker-a.

Postavljanje baze podataka na Render servisu

1. Otvoriti render.com i registrirati se.
2. Napraviti novu PostgreSQL bazu. Ime odabrati proizvoljno.



New PostgreSQL

Name	bazapodataka
Database	randomly generated unless specified
User	randomly generated unless specified
Region	Frankfurt (EU Central)
PostgreSQL Version	15
Datadog API Key	

Please [enter your payment information](#) to select an instance type with higher limits.

Instance Type	RAM	CPU	Storage	Price
<input checked="" type="radio"/> Free	256 MB	Shared	1 GB	\$0 / month

- Otvoriti Dashboard i kliknuti na upravo kreiranu bazu.
- Podaci na dnu u Info prozoru pod Connections će nam trebati za spajanje aplikacije s bazom.

Info

Metrics

Backups

Logs

Connections

Hostname

hostname

Port

port

Database

database

Username

username

Password

Internal Database URL

External Database URL

PSQL Command

- Pod Access Control treba dodati vlastitu IP računala na kojem će se vrtiti aplikacija.

Info

Metrics

Backups

Logs

Access Control

3 IP ranges are allowed from outside of your private network.

Sources are specified [CIDR block notation](#).

Source	Description
IP adresa	opis

Test an IP address ⓘ

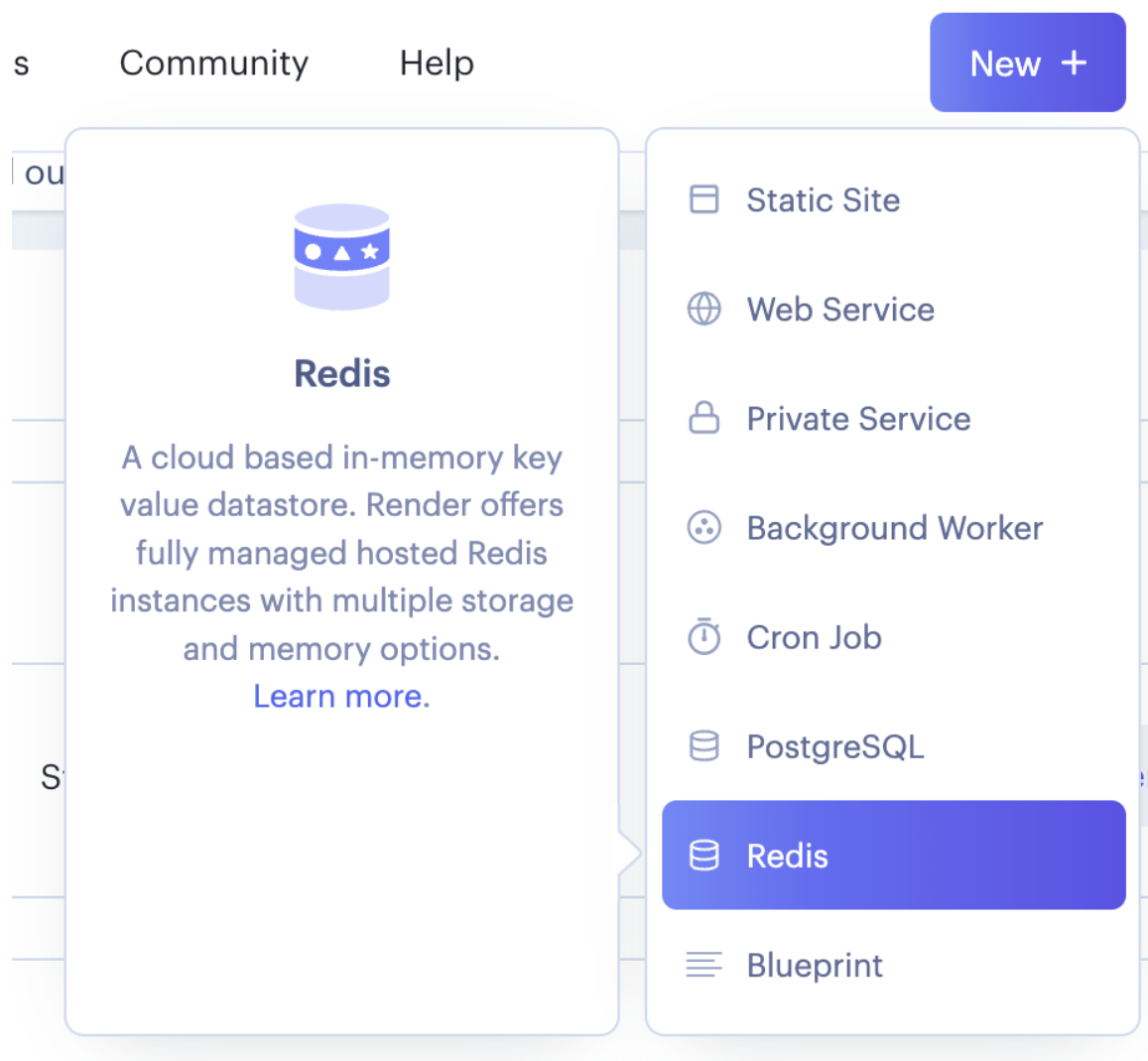
+ Add source

Cancel

Save

Postavljanje Redis baze na Render servisu

1. Otvoriti render.com i registrirati se.
2. Napraviti novu Redis bazu. Ime odabrati proizvoljno.



New Redis Instance

[Read the docs](#)

Name

redis-baza

Region

The region where your Redis runs. Services must be in the same region to communicate privately and you currently have services running in Frankfurt.

Frankfurt (EU Central)

Maxmemory Policy

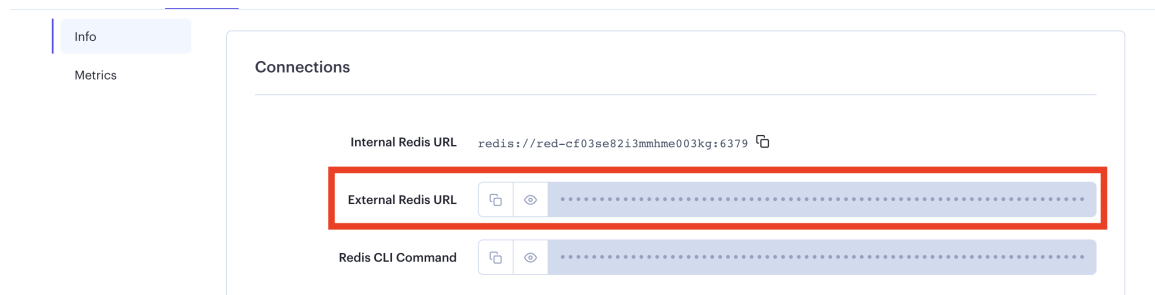
The eviction policy for when Redis' memory is full.

allkeys-lru (recommended for caches)

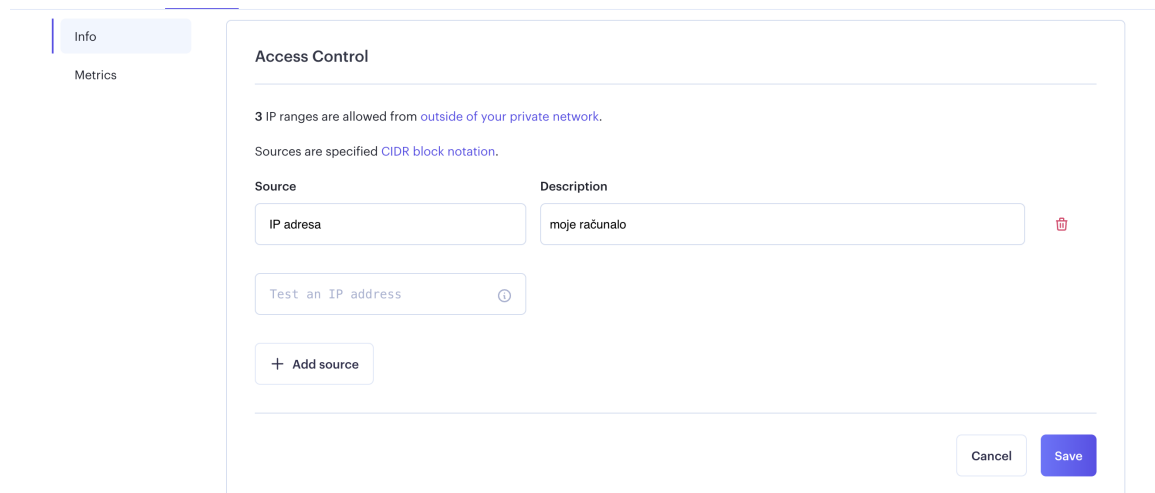
Please enter your payment information to select an instance type with higher limits.

Instance Type	RAM	Connection Limit	Persistence	Price
<input checked="" type="radio"/> Free	25 MB	50	×	\$0 / month

3. Otvoriti Dashboard i kliknuti na upravo kreiranu bazu.
4. External Redis URL pod Connections će nam trebati za spajanje aplikacije s bazom.



5. Pod Access Control treba dodati vlastitu IP računala na kojem će se vrtiti aplikacija.



Pokretanje aplikacije

1. Instalirati Docker.
2. Za slanje mailova treba napraviti gmail account te za njega postaviti lozinku za aplikacije prema sljedećim uputama.
3. Klonirati git repozitorij aplikacije na računalo s kojeg će biti pokrenuta aplikacija.
4. U IzvorniKod repozitoriju stvoriti .env datoteku čiji sadržaj treba popuniti s podacima o PostgreSQL i Redis bazi te e-mailu.

Sadržaj .env datoteke:

POSTGRES_PASSWORD = <Password iz Connections podataka za PostgreSQL bazu>

POSTGRES_HOST = <Hostname iz Connections podataka

za PostgreSQL bazu>.frankfurt-postgres.render.com

POSTGRES_DB = <Database iz Connections podataka za PostgreSQL bazu>

POSTGRES_USER = <Username iz Connections podataka za PostgreSQL bazu>

REDIS_URL = <External Redis URL>

DEFAULT_FROM_EMAIL = <vlastiti e-mail>

DEFAULT_FROM_EMAIL_PASSWORD = <App password za gore navedeni e-mail>

SECRET_KEY = <proizvoljni niz znakova>

SECURITY_PASSWORD_SALT = <proizvoljni niz znakova>

5. Iz IzvorniKod repozitorija pokrenuti naredbu docker-compose up.
6. Otvoriti <http://localhost:3000> u browseru.

6. Zaključak i budući rad

Naša je grupa imala zadatak razviti web aplikaciju koja funkcionira kao online igra u kojoj igrači prikupljaju kartice i s njima ulaze u bitke s obližnjim igračima, dok najbolji među njima mogu predlagati nove lokacije koje kartografi pak pregledavaju, uređuju i odobravaju, a administratori sve to nadziru uz potpunu kontrolu nad korisnicima i lokacijama. Na ovom smo projektu radili 10 tjedana i uspješno smo implementirali tražene funkcionalnosti.

Tijekom prvog nastavnog ciklusa oformili smo tim i postavili neke organizacijske premise, a zatim se i podijelili u *frontend* i *backend* podtimove, uz dio članova posebno fokusiran na izradu dokumentacije projekta. Značajnu smo količinu truda i vremena uložili u dokumentiranje zahtjeva i pripremu kostura projekta, koji su nam naposljetku bili uvelike od koristi.

U drugom nastavnom ciklusu intenzivno smo radili na implementaciji dokumentiranih funkcionalnosti i ključni pristup koji nam je pomogao u radu bilo je timsko uhođavanje u tehnologije koje je praćeno i samostalnim radom članova. Isprva smo značajnu količinu vremena trošili na organizaciju i podjelu posla, no ubrzo smo tome doskočili organizacijskom tablicom i direktnom komunikacijom članova backend i frontend timova. Problem na koji smo naišli na kraju jest nemogućnost korištenja aplikacije na udaljenom servisu jer korištenje lokacije zahtijeva komunikaciju HTTPS protokolom, no to nismo uspjeli postići dodavanjem certifikata na korištenom servisu.

Komunikaciju unutar tima ostvarili smo na nekoliko razina - WhatsApp grupu koristili smo za bitne obavijesti i dogovore oko općih sastanaka, vlastiti Discord server koristili smo za odvajanje pisane i glasovne komunikacije unutar tima na teme general, backend, frontend i dokumentacija. Ti su nam kanali služili za svakodnevnu suradnju i smanjenje spam sadržaja. Uz navedeno, koristili smo i zajednički Google disk gdje smo vodili tjedne bilješke sastanaka koje su nam služile za isticanje tekuće problematike i evidentiranje pomaka i resursa, kao i ideja koje smo svi istovremeno bilježili tijekom sastanaka.

Rad na ovom projektu na razne je načine bio novi za sve nas, posebno zbog zahtijevane koordinacije i balansiranja zadataka među članovima. Nailazili smo na

manje poteškoće u komunikaciji i međusobnim očekivanjima no uspješno smo prebrodili i te izazove. U realizaciji projekta mnogo bismo toga promijenili, no zbog drugih opterećenja tijekom semestra trudili smo se postići osnovnu funkcionalnost aplikacije i u tome smo uspjeli te smo zato zadovoljni ukupnom provedbom projekta.

Popis literature

Kontinuirano osvježavanje

Popisati sve reference i literaturu koja je pomogla pri ostvarivanju projekta.

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

2.1	Web sjedište platforme <i>Geocaching</i>	6
3.1	Dijagram obrasca uporabe, funkcionalnost gosta i korisnika	24
3.2	Dijagram obrasca uporabe, funkcionalnost igrača i naprednog igrača	25
3.3	Dijagram obrasca uporabe, funkcionalnost kartografa	26
3.4	Dijagram obrasca uporabe, funkcionalnost admina	27
3.5	Sekvencijski dijagram za UC14	28
3.6	Sekvencijski dijagram za UC17, UC18	29
3.7	Sekvencijski dijagram za UC21, UC23	30
4.1	Dijagram baze podataka	39
4.2	Dijagram razreda - dio Controllers	40
4.3	Dijagram razreda - dio Data transfer objects	41
4.4	Dijagram razreda - dio Models	42
4.5	Dijagram stanja	43
4.6	Dijagram aktivnosti	45
4.7	Dijagram komponenti	47
5.1	Dijagram razmještaja	51
6.1	Prikaz aktivnosti na repozitoriju	68
6.2	Prikaz aktivnosti na repozitoriju	69

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 17. listopada 2022.
- Prisustvovali: Lovro Kovačić, Petar Hotovec, Vladimir Pezo, Martina Šola, Sara Rivić Carević, Lovro Raguž, Ela Kumer
- Teme sastanka:
 - prolazak projektnog zadatka i definiranje pitanja za sastanak s asistentom

2. sastanak

- Datum: 27. listopada 2022.
- Prisustvovali: Lovro Kovačić, Petar Hotovec, Vladimir Pezo, Martina Šola, Sara Rivić Carević, Lovro Raguž, Ela Kumer
- Teme sastanka:
 - postavljanje GitLab-a
 - podjela u grupe na projektu (backend, frontend, dokumentacija, dizajn)
 - definiranje tehnologija koje ćemo koristiti
 - dogovor oko početka pisanja dokumentacije

3. sastanak

- Datum: 1. studenoga 2022.
- Prisustvovali: Lovro Kovačić, Petar Hotovec, Vladimir Pezo, Martina Šola, Lovro Raguž, Ela Kumer
- Teme sastanka:
 - dokumentacija, definiranje UC-ova
 - podjela pisanja dokumentacije
 - lokalno postavljanje kostura projekta na vlastitim računalima

4. sastanak

- Datum: 2. studenoga 2022.

- Prisustvovali: Lovro Kovačić, Petar Hotovec, Vladimir Pezo, Martina Šola, Lovro Raguž, Ela Kumer, Sara Rivić Carević
- Teme sastanka:
 - dokumentacija, podjela poslova za izradu sekvencijskih dijagrama i dijagrama obrazaca uporabe
 - prva verzija login funkcionalnosti s frontend strane

5. sastanak

- Datum: 3. studenoga 2022.
- Prisustvovali: Lovro Kovačić, Vladimir Pezo, Martina Šola, Lovro Raguž, Ela Kumer
- Teme sastanka:
 - pregled UC-ova
 - pregled napravljenih dijagrama obrazaca uporabe
 - pregled napravljenih sekvencijskih dijagrama

6. sastanak

- Datum: 10. studenoga 2022.
- Prisustvovali: Lovro Kovačić, Petar Hotovec, Sara Rivić Carević, Vladimir Pezo, Lovro Raguž, Ela Kumer
- Teme sastanka:
 - povezivanje frontend-a i backend-a
 - razrada borbe
 - podjela zadataka za prvu predaju

7. sastanak

- Datum: 15. studenoga 2022.
- Prisustvovali: Lovro Kovačić, Sara Rivić Carević, Vladimir Pezo, Ela Kumer, Martina Šola
- Teme sastanka:
 - ispravci i dopune dokumentacije
 - ispravci baze
 - spremanje slika u bazu

8. sastanak

- Datum: 16. studenoga 2022.
- Prisustvovali: Lovro Kovačić, Sara Rivić Carević, Vladimir Pezo, Ela Kumer, Martina Šola, Lovro Raguž, Petar Hotovec
- Teme sastanka:

- definicija uloge administratora
- dijagram baze
- priprema dokumentacije za predaju
- dijagrami razreda

9. sastanak

- Datum: 17. studenoga 2022.
- Prisustvovali: Lovro Kovačić, Sara Rivić Carević, Vladimir Pezo, Ela Kumer, Martina Šola, Lovro Raguž, Petar Hotovec
- Teme sastanka:
 - demonstracija generičke funkcionalnosti asistentu
 - specifikacija arhitekture projekta
 - DTO dijagrami
 - definiranje uloge korisnika i uloge gosta

10. sastanak

- Datum: 8. prosinca 2022.
- Prisustvovali: Lovro Kovačić, Sara Rivić Carević, Vladimir Pezo, Ela Kumer, Martina Šola, Lovro Raguž, Petar Hotovec
- Teme sastanka:
 - priprema za kolokviranje projekta

11. sastanak

- Datum: 15. prosinca 2022.
- Prisustvovali: Lovro Kovačić, Sara Rivić Carević, Vladimir Pezo, Martina Šola, Lovro Raguž, Petar Hotovec
- Teme sastanka:
 - rezimiranje dosadašnjeg napretka
 - dogovor oko planova za daljnji rad
 - podjela ljudi u timove

12. sastanak

- Datum: 22. prosinca 2022.
- Prisustvovali: Lovro Kovačić, Vladimir Pezo, Martina Šola, Lovro Raguž, Petar Hotovec
- Teme sastanka:
 - predaja alfa inačice

13. sastanak

- Datum: 31. prosinca 2022.

- Prisustvovali: Lovro Kovačić, Vladimir Pezo, Ela Kumer, Martina Šola, Petar Hotovec
- Teme sastanka:
 - bannanje usera
 - errori za krivi tip usera
 - odobravanje kartica (kartograf)
 - generiranje rute
 - flowchart borbe
 - globalno konfiguriranje axiosa
 - pregled profila
 - daljnja raspodjela poslova i određivanje milestone-a

14. sastanak

- Datum: 5. siječnja 2023.
- Prisustvovali: Lovro Kovačić, Sara Rivić Carević, Vladimir Pezo, Ela Kumer, Martina Šola, Petar Hotovec
- Teme sastanka:
 - podjela poslova do kraja na frontendu
 - završne izmjene u backendu
 - plan završetka projekta

15. sastanak

- Datum: 14. siječnja 2023.
- Prisustvovali: Sara Rivić Carević, Vladimir Pezo, Martina Šola
- Teme sastanka:
 - podjela ostatka dokumentacije

Tablica aktivnosti

	Lovro Kovačić	Ela Kumer	Martina Šola	Sara Rivić Carević	Lovro Raguž	Vladimir Pezo	Petar Hotovec
Upravljanje projektom	4	0	0	0	0	0	0
Opis projektnog zadatka	0	2	3	1	1	6	0
Funkcionalni zahtjevi	2	3	5	2	2	2	2
Opis pojedinih obrazaca	2	0	4	0	1	8	0
Dijagram obrazaca	0	0	0	0	6	2	0
Sekvencijski dijagrami	0	3	0	0	1	2	0
Opis ostalih zahtjeva	0	5	2	0	0	1	0
Arhitektura i dizajn sustava	2	0	3	0	0	1	0
Baza podataka	2	5	4	12	2	3	0
Dijagram razreda	4	4	2	0	4	5	2
Dijagram stanja	0	0	0	5	0	0	0
Dijagram aktivnosti	0	0	0	4	0	0	0
Dijagram komponenti	0	0	0	0	0	4	0
Korištene tehnologije i alati	0	0	4	0	0	0	0
Ispitivanje programskog rješenja	0	0	0	0	0	0	0
Dijagram razmještaja	0	0	0	3	0	0	0
Upute za puštanje u pogon	0	2	0	0	0	0	0
Dnevnik sastajanja	0	1	1	0	0	0	0
Zaključak i budući rad	0	0	0	0	0	1	0

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Lovro Kovačić	Ela Kumer	Martina Šola	Sara Rivić Carević	Lovro Raguž	Vladimir Pezo	Petar Hotovec
Popis literature	0	0	0	0	0	0	0
Upoznavanje s tehnologijama	7	5	6	6	6	6	8
Izrada kostura projekta	6	7	2	2	1	2	6
Izrada login/register screna	14	0	0	0	0	0	8
Izrada home screena	1	0	0	0	0	0	5
Povezivanje frontenda i backenda	8	3	0	5	3	0	6
Izrada baze podataka i modela	7	3	0	9	3	0	1
Izrada login/register screena (backend)	1	7	0	1	4	0	0
Deployment	0	24	0	0	5	0	0
Izrada i popunjavanje organizacijskog lista za HTTP zahtjeve	0	4	3	0	3	2	5
Izrada funkcionalnosti za igrača	10	12	6	0	3	17	17
Izrada funkcionalnosti za administratora	12	10	0	0	3	6	10
Izrada funkcionalnosti za kartografa	6	10	0	0	2	21	17
Bug fixes	10	12	0	0	17	5	13

Dijagrami pregleda promjena

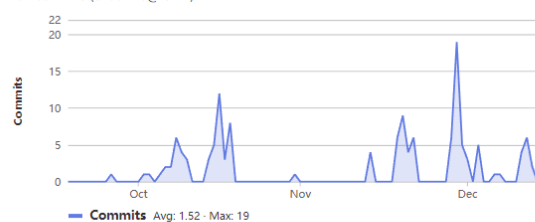
Commits to main

Excluding merge commits. Limited to 6,000 commits.



Ela

134 commits (ek53440@fer.hr)



kazokid

107 commits (lovro.kovacic7@gmail.com)



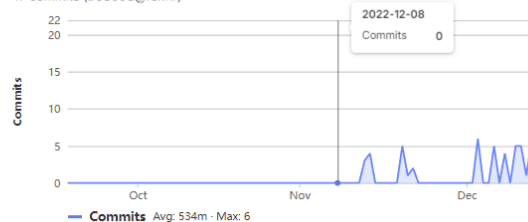
photovec

99 commits (petar.hotovec44@gmail.com)



Lovro Raguž

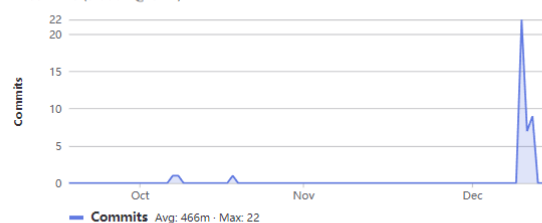
47 commits (lr53603@fer.hr)



Slika 6.1: Prikaz aktivnosti na repozitoriju

Sara Rivić Carević

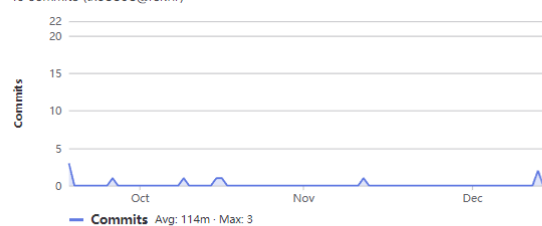
41 commits (sr53612@fer.hr)

**atom-cluster**

30 commits (vladimir.pezo@fer.hr)

**Lovro Kovačić**

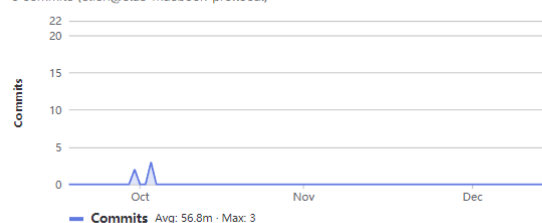
10 commits (lk53393@fer.hr)

**martina.sola**

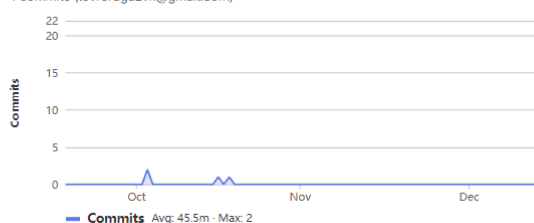
8 commits (martina.sola@fer.hr)

**Ela Kumer**

5 commits (elich@elas-macbook-pro.local)

**Lovro Raguz**

4 commits (lovroraguzvk@gmail.com)



Slika 6.2: Prikaz aktivnosti na repozitoriju