Programsko inženjerstvo ak. god 2024./2025.

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Quiz Finder

Tim: <TG08.4>

Magna Creare

Nastavnik: Vlado Sruk

Asistent: Miljenko Krhen

Demonstrator: Petar Jakuš

Članovi tima:

- Nikolas Jarić (nikolas.jaric@fer.unizg.hr)
- Ema Skoko (ema.skoko@fer.unizg.hr)
- Matija Križević (matija.krizevic@fer.unizg.hr)
- Ante Perić (ante.peric2@fer.unizg.hr)
- Petar Krtalić (petar.krtalic@fer.unizg.hr)
- Vice Sladoljev (vice.sladoljev@fer.unizg.hr)

Detaljan opis korisničkih zahtjeva za aplikaciju "Quiz Finder"

Generalno

Aplikacija *Quiz Finder* zamišljena je kao inovativno digitalno rješenje koje omogućava ljubiteljima pub kvizova da jednostavno pronađu kvizove u svom okruženju, prijave se za sudjelovanje, i sudjeluju u događajima prema vlastitim interesima. S druge strane, aplikacija pruža organizatorima moćan alat za organizaciju kvizova, uključujući upravljanje prijavama, promociju kvizova i praćenje povratnih informacija od sudionika.

Primarni fokus aplikacije je na jednostavnosti korištenja, prilagođavanju korisničkog iskustva te osiguravanju visoke razine sigurnosti. Korisničko sučelje je responzivno i prilagođeno kako mobilnim uređajima tako i stolnim računalima, čime se omogućuje jednostavan pristup kvizovima u bilo kojem trenutku i s bilo kojeg uređaja. Kroz intuitivne opcije filtriranja i pretrage, korisnici mogu brzo pronaći kvizove koji odgovaraju njihovim interesima i rasporedu, birajući između različitih kriterija kao što su težina kviza, lokacija, kategorija i cijena kotizacije.

Aplikacija također služi kao platforma za povezivanje sudionika i organizatora, čime se stvara zajednica ljubitelja kvizova. Osim prijave na kvizove, korisnicima se omogućuje praćenje omiljenih organizatora i lokacija (kafića i pubova) te ocjenjivanje kvizova u kojima su sudjelovali. To pridonosi stvaranju povratne veze između organizatora i sudionika, omogućujući organizatorima da kontinuirano unapređuju kvalitetu svojih događanja.

Aplikacija je također zamišljena kao fleksibilna platforma koja može rasti i prilagođavati se potrebama korisnika. Organizatori mogu jednostavno stvarati nove kvizove i uređivati postojeće, dok administratori mogu nadgledati rad cijelog sustava, upravljati korisnicima i intervenirati u slučaju bilo kakvih nepravilnosti.

Cilj projektnog zadatka

Cilj ovog projekta je razvoj aplikacije koja omogućuje korisnicima jednostavan pristup informacijama o pub kvizovima, pretragu dostupnih kvizova na temelju vlastitih preferencija te jednostavnu prijavu. Za organizatore, cilj je stvoriti platformu koja centralizira sve aspekte organizacije kvizova, od unosa podataka o kvizovima do praćenja prijava i povratnih informacija sudionika. Aplikacija također uključuje administrativne funkcionalnosti koje omogućuju nadzor i kontrolu cjelokupnog sustava kako bi se osigurala sigurnost i visoka kvaliteta korisničkog iskustva.

U širem kontekstu, Quiz Finder nastoji riješiti problem disperziranih i neuređenih informacija o pub kvizovima koje trenutno organizatori i sudionici razmjenjuju preko raznih kanala (društvenih mreža, web stranica, tiskanih oglasa). Razvijanjem centralizirane platforme, očekuje se poboljšanje vidljivosti i dostupnosti kvizova te povećanje interakcije između zajednice natjecatelja i organizatora.

Problematika zadatka

Aplikacija adresira nekoliko ključnih problema s kojima se suočavaju sudionici i organizatori pub kvizova:
• **Teškoća u pronalaženju relevantnih kvizova**: Zbog raznih dostupnih platformi i društvenih mreža, korisnici teško nalaze sveobuhvatne i centralizirane informacije o kvizovima. Kroz aplikaciju Quiz Finder, korisnici će moći filtrirati kvizove prema kriterijima kao što su težina kviza, tema, lokacija i cijena, što olakšava pregled i

prijavu.
• **Nedostatak standardiziranog sustava prijave**: Trenutni sustavi prijave variraju od organizatora do organizatora, što može biti zbunjujuće za sudionike. Ovaj projekt uključuje ujednačen sustav za prijavu i registraciju za kvizove, pružajući korisnicima dosljedno iskustvo.
• **Kompleksnost upravljanja kvizovima**: Organizatorima kvizova može biti izazovno koordinirati prijave, upravljati povratnim informacijama i osigurati nesmetano odvijanje kviza. Aplikacija će organizatorima pružiti centralizirano rješenje koje olakšava organizaciju događanja, praćenje broja prijavljenih ekipa i post-kviz evaluaciju.

Potencijalna korist ovog projekta

Prednosti Quiz Findera proizlaze iz njegove sposobnosti da olakša pristup kvizovima za sve uključene korisnike:
• **Natjecatelji** će imati jednostavan i personaliziran način pretraživanja kvizova na temelju interesa, ocjena kvizova i prijašnjih sudjelovanja, što omogućava poboljšano korisničko iskustvo i povećava vjerojatnost sudjelovanja.
• **Organizatori kvizova** dobivaju platformu za upravljanje svojim događanjima, što uključuje evidenciju prijava, praćenje timova te dobivanje povratnih informacija. Sve informacije o kvizu se jednostavno unose i ažuriraju kroz jedno sučelje, što smanjuje potrebu za korištenjem dodatnih alata.
• **Administratori** aplikacije imat će pristup alatima za nadzor i kontrolu sustava, uključujući moderiranje sadržaja, što doprinosi sigurnosti i održavanju kvalitete aplikacije.
Aplikacija također koristi popularne sigurnosne protokole (OAuth2) kako bi zaštitila korisničke podatke tijekom procesa autentifikacije, što povećava sigurnost i povjerenje korisnika u sustav.

Postojeća slična rješenja

Na tržištu postoje i druga rješenja koja nude određene funkcionalnosti povezane s kvizovima, ali nijedno od njih ne nudi sveobuhvatnu integraciju za pretragu, organizaciju i praćenje lokalnih pub kvizova kao Quiz Finder. Neki primjeri sličnih rješenja su:
• **QuizUp**: Aplikacija koja nudi veliki broj kvizova u raznim kategorijama, ali se uglavnom fokusira na online kvizove i ne pruža opcije za organizaciju fizičkih pub kvizova.
• **Meetup**: Omogućava organizaciju događaja uživo, ali nije specifična za kvizove, te ne uključuje specijalizirane alate za kvizove poput filtriranja prema težini ili kategorijama kviza.
U odnosu na postojeća rješenja, Quiz Finder nudi jedinstvenu kombinaciju prilagođenih alata za organizatore kvizova i natjecatelje, uključujući praćenje organizatora i pretragu kvizova na temelju tematskih i geografskih kriterija.

Skup korisnika

Korisnici aplikacije uključuju:
• **Neregistrirane korisnike**: Mogu pregledavati dostupne kvizove bez potrebe za prijavom, čime im je omogućeno jednostavno istraživanje mogućnosti koje aplikacija nudi, ali bez mogućnosti prijave na kvizove, praćenja organizatora te ocjenjivanja kvizova.
• **Natjecatelje**: Osobe koje redovito posjećuju kvizove, entuzijaste i grupe prijatelja. Ova skupina korisnika

koristi aplikaciju za pretragu i prijavu na kvizove, praćenje omiljenih organizatora te sudjelovanje u kvizovima po vlastitom izboru.
• **Organizatore**: Kafići, pubovi, restorani ili profesionalni organizatori kvizova koji koriste aplikaciju kako bi promovirali događanja, upravljali prijavama i prikupljali povratne informacije nakon kvizova.
• **Administratore**: Osobe koje nadziru sigurnost i pravilno funkcioniranje aplikacije, uključujući moderiranje sadržaja i podršku korisnicima.

Mogućnost prilagodbe rješenja

Quiz Finder je osmišljen kao fleksibilna aplikacija koja se može prilagoditi i proširiti na različite načine:
• **Dodavanje novih kategorija kvizova**: Aplikacija može uključiti specijalizirane kategorije kvizova, poput kvizova za djecu, tematskih kvizova za blagdane ili specijalnih događanja.
• **Integracija s društvenim mrežama**: Korisnici bi mogli dijeliti informacije o kvizovima na društvenim mrežama, što bi povećalo vidljivost i potaklo veće sudjelovanje.
• **Multijezičnost**: Dodavanje podrške za više jezika omogućilo bi proširenje aplikacije na druge zemlje i povećalo dostupnost.

Opseg projektnog zadatka

Opseg projekta obuhvaća razvoj frontend i backend komponenata, integraciju API servisa i postavljanje sigurnosnih protokola:
• **Frontend**: Aplikacija će biti razvijena u React.js, osiguravajući responzivno korisničko sučelje za mobilne uređaje i računala.
• **Backend**: Razvoj backend sustava pomoću Django frameworka omogućit će upravljanje logikom aplikacije i interakciju s bazom podataka PostgreSQL. Kroz Django će se implementirati API-ji za povezivanje s frontend dijelom.
• **Sigurnost**: Implementacija autentifikacije putem OAuth2 omogućit će sigurnu pohranu i zaštitu korisničkih podataka tijekom registracije i prijave.

Moguće nadogradnje projektnog zadatka

Nadogradnje koje bi mogle poboljšati aplikaciju uključuju:
• **Gamifikacija korisničkog iskustva**: Uvođenjem bodovnih sustava i postignuća, korisnici bi mogli ostvariti određene nagrade ili značke, čime bi se potaknuo angažman i povećalo zadovoljstvo korisnika.
• **Mobilna aplikacija**: Razvoj aplikacije za iOS i Android omogućio bi korisnicima bolji pristup aplikaciji dok su u pokretu.
• **Analitički alati za organizatore**: Organizatorima bi mogla biti dodana mogućnost pregleda statističkih podataka o sudjelovanju i popularnosti kvizova, čime bi se optimiziralo iskustvo organiziranja.

Zaključak

Aplikacija Quiz Finder predstavlja cjelovito rješenje za ljubitelje pub kvizova i organizatore kvizova, olakšavajući pristup informacijama, pojednostavljujući prijavu i organizaciju događanja. Integracija Google Maps API-a i sigurnosnih protokola poput OAuth2 pruža dodatnu vrijednost i sigurnost korisnicima.

Fleksibilnost u razvoju omogućava skalabilnost aplikacije i prilagodbu potrebama korisnika, čineći je dugoročno održivom i korisnom.

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
F-001	Organizator može kreirati kviz unoseći naziv, kategoriju, težinu, datum, lokaciju itd.	Visok	Organizator	Kviz kreiran s unesenim detaljima.
F-002	Organizator može pregledati povratne informacije i ocjene kvizova.	Srednji	Organizator	Povratne informacije i ocjene prikazane.
F-003	Organizator može postaviti kviz kao dio lige.	Srednji	Organizator	Kviz uspješno dodan u ligu.
F-004	Natjecatelji mogu pretraživati kvizove prema različitim kriterijima.	Visok	Natjecatelj	Rezultati pretrage prikazani.
F-005	Natjecatelji mogu pregledavati kvizove na karti s Google Maps API-jem.	Visok	Natjecatelj	Kvizovi prikazani na karti.
F-006	Natjecatelji mogu prijaviti svoje timove za odabrane kvizove.	Visok	Natjecatelj	Prijava uspješno zabilježena.
F-007	Natjecatelji mogu pregledati profile organizatora.	Srednji	Natjecatelj	Profili organizatora prikazani.
F-008	Natjecatelji mogu ocjenjivati kvizove nakon sudjelovanja.	Srednji	Natjecatelj	Ocjena kviza uspješno dodana.
F-009	Administrator može upravljati korisničkim računima.	Visok	Administrator	Računi pregledani i upravljani.
F-010	Administrator može pregledati i upravljati prijavljenim ekipama.	Visok	Administrator	Prijave prikazane i ažurirane.
F-011	Administrator može pregledati i upravljati kvizovima i recenzijama.	Visok	Administrator	Recenzije prikazane i ažurirane.

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
F-012	Administrator može intervenirati u kvizovima.	Srednji	Administrator	Kvizovi uređeni ili izbrisani.
F-013	Neregistrirani korisnici mogu pregledavati kvizove.	Nizak	Neregistrirani	Kvizovi pregledani bez prijave.
F-014	Neregistrirani korisnici mogu pregledavati kvizove na karti.	Nizak	Neregistrirani	Kvizovi prikazani na karti bez prijave.
F-015	Aplikacija omogućava sigurno prijavljivanje/registraciju putem OAuth2.	Visok	Svi korisnici	Korisnici se mogu prijaviti putem OAuth2.
F-016	Aplikacija omogućava filtriranje kvizova po težini, cijeni, kategoriji i ligi.	Srednji	Svi korisnici	Rezultati filtriranja prikazani.
F-017	Aplikacija omogućava pretraživanje kvizova po imenu kviza i imenu organizatora.	Srednji	Svi korisnici	Rezultati filtriranja prikazani.

Ostali zahtjevi

Nefunkcionalni zahtjevi

Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju kako se sustav treba ponašati i koja ograničenja treba poštivati, uključujući performanse, korisničko iskustvo, pouzdanost, sigurnost i druge tehničke standarde.

- **Podržani jezici korisničkog sučelja:** Aplikacija mora podržavati hrvatski jezik kao glavni jezik korisničkog sučelja, uz opciju dodatnih jezika ako se to naknadno zahtijeva. Podrška za više jezika omogućava pristup aplikaciji širem krugu korisnika i poboljšava internacionalnu upotrebu.
- **Vrijeme odziva:** Vrijeme odziva aplikacije ne smije biti duže od 2-3 sekunde na prosječnoj brzini interneta za učitavanje stranica i feedova. Brzo vrijeme učitavanja doprinosi boljim korisničkim iskustvima i zadržavanju korisnika u aplikaciji.
- **Podržani broj korisnika:** Sustav mora biti skalabilan i sposoban podržati stotine korisnika istovremeno bez značajnog smanjenja performansi, kako bi mogao podnijeti veliki broj prijava i interakcija na kvizovima. To uključuje podršku za velike količine podataka, kao što su korisnički računi i kvizovi.

- **Podržane platforme:** Aplikacija mora biti kompatibilna s modernim web preglednicima (Chrome, Firefox, Safari, Edge) te optimizirana za mobilne uređaje (iOS, Android) putem responzivnog dizajna. Aplikacija mora pružati besprijekorno korisničko iskustvo na svim uređajima, omogućujući pristup sa svih platformi bez gubitka funkcionalnosti.
- **Razina zaštite:** Komunikacija između korisnika i aplikacije mora biti šifrirana pomoću SSL/TLS protokola, a svi podaci korisnika i kvizova trebaju biti zaštićeni odgovarajućim sigurnosnim mjerama, uključujući enkripciju baze podataka i kontrolu pristupa na temelju uloga korisnika. Ovo osigurava sigurnost osjetljivih informacija, poput osobnih podataka i financijskih transakcija.
- **Visoka dostupnost:** Aplikacija mora biti dostupna najmanje 99,9% vremena, uz minimalne zastoje koji se odnose samo na održavanje ili planirane nadogradnje sustava. Visoka dostupnost je ključna za očuvanje korisničkog povjerenja i stalni rad aplikacije, bez prekida u funkcionalnosti.
- **Pouzdanost sustava:** Sustav mora biti pouzdan, s minimalnim brojem grešaka i zastoja, te mora osigurati pravovremenu obradu podataka i odgovora korisnicima. Pouzdanost je važna za održavanje dosljednog korisničkog iskustva i smanjenje broja tehničkih problema koji mogu ometati rad aplikacije.
- **Modularnost:** Sustav mora biti izgrađen na modularnoj arhitekturi koja omogućava jednostavno dodavanje novih funkcionalnosti i lakše održavanje. Modularni dizajn omogućuje lakšu nadogradnju sustava, bez potrebe za velikim promjenama u postojećem kodu.
- **Skalabilnost:** Aplikacija mora biti dizajnirana za skaliranje kako bi mogla podržati povećanje broja korisnika, kvizova i interakcija, uz minimalne promjene u infrastrukturi. To osigurava dugoročnu održivost aplikacije, bez potrebe za čestim velikim investicijama u hardver ili infrastrukturu.
- **Učinkovitost:** Aplikacija mora omogućiti brzo pretraživanje kvizova i interakciju s kartama, filtrima i drugim alatima, bez opterećenja sustava. Učinkovitost sustava smanjuje vrijeme potrebno za obavljanje radnji, čime se povećava zadovoljstvo korisnika i omogućuje veća produktivnost.

Zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NF-3.1	Sustav treba biti oblikovan tako da omogućuje jednostavno održavanje.	Visok
NF-3.1.7	Sustav treba imati dovoljnu dokumentaciju.	Visok
NF-3.1.7.1	Kôd sustava treba biti dokumentiran prema "Code Conventions for the Python Programming Language" dostupnim na službenoj Python dokumentaciji.	Visok
NF-3.1.7.2	Sustav treba biti opisan putem dokumenta oblikovanja (SRS - Software Requirements Specification).	Visok
NF-3.1.7.3	Sustav treba biti popraćen "Priručnikom za rad" koji opisuje pravilnu upotrebu sustava, uključujući korisničke upute za organizatore, natjecatelje, administratore i neregistrirane korisnike.	Visok

ID zahtjeva	Opis	Prioritet
NF-3.1.7.4	Sustav treba imati "Plan implementacije" za pravilno postavljanje sustava, uključujući sve potrebne korake za instalaciju, konfiguraciju i pokretanje aplikacije na serveru.	Visok

Dionici

Dionici i aktori sustava

Dionici su sve osobe, organizacije ili sustavi koji imaju interes u sustavu ili su odgovorni za njegov rad. To uključuje korisnike, administratore sustava, naručitelje, te razvojni tim. Akteri su entiteti koji izravno komuniciraju sa sustavom, bilo da ga koriste (inicijatori ili sudionici) ili komuniciraju s njim u nekoj drugoj ulozi.

Dionici:

- **Korisnici** (organizatori, natjecatelji, administratori, neregistrirani korisnici)
- **Administrator sustava** (osoba odgovorna za održavanje sustava i upravljanje korisnicima)
- **Razvojni tim** (tim koji razvija i održava sustav)
- **Naručitelj** (osoba ili organizacija koja je naručila razvoj sustava)

Aktor 1: Organizator (inicijator)

Organizator je osoba koja inicira procese u sustavu kao što su kreiranje, uređivanje i upravljanje kvizovima.

- **Funkcionalnosti koje organizator može obavljati:**
  - **Kreirati kvizove**: Organizator može unijeti sve potrebne podatke za kviz (naziv, kategorija, datum, lokacija, broj ekipa).
    - **F-001:** Kreiranje kviza
  - **Dodavanje kviz u ligu**: Organizator može dodati kviz u ligu.
    - **F-003:** Dodavanje kviz u ligu
  - **Pratiti povratne informacije**: Organizator može pregledati ocjene i povratne informacije sudionika.
    - **F-002:** Pregled povratnih informacija

Aktor 2: Natjecatelj (sudionik)

Natjecatelj je osoba koja sudjeluje u kvizovima, pretražuje dostupne kvizove i prijavljuje svoj tim.

- **Funkcionalnosti koje natjecatelj može obavljati:**
  - **Pretraživanje kvizova**: Natjecatelj može pretraživati kvizove prema različitim kriterijima (kategorija, lokacija, cijena).
    - **F-004:** Pretraga kvizova
  - **Prijava na kviz**: Natjecatelj može prijaviti svoj tim za odabrani kviz.
    - **F-006:** Prijava na kviz
  - **Pregled kvizova na karti**: Natjecatelj može pregledati kvizove na karti integriranoj s Google Maps API.

        - **F-005:** Prikaz kvizova na karti
    - **Ocjenjivanje kvizova**: Nakon završetka kviza, natjecatelj može ocijeniti kviz.
        - **F-008:** Ocjenjivanje kvizova

Aktor 3: Administrator sustava (inicijator)

Administrator je osoba odgovorna za upravljanje korisnicima, održavanje sustava i intervenciju u slučaju problema.

- **Funkcionalnosti koje administrator može obavljati:**
    - **Upravljanje korisnicima**: Administrator može pregledavati i brisati korisničke račune.
        - **F-009:** Upravljanje korisnicima
    - **Upravljanje kvizovima**: Administrator može intervenirati u kvizovima.
        - **F-011:** Upravljanje kvizovima

Aktor 4: Neregistrirani korisnik (sudionik)

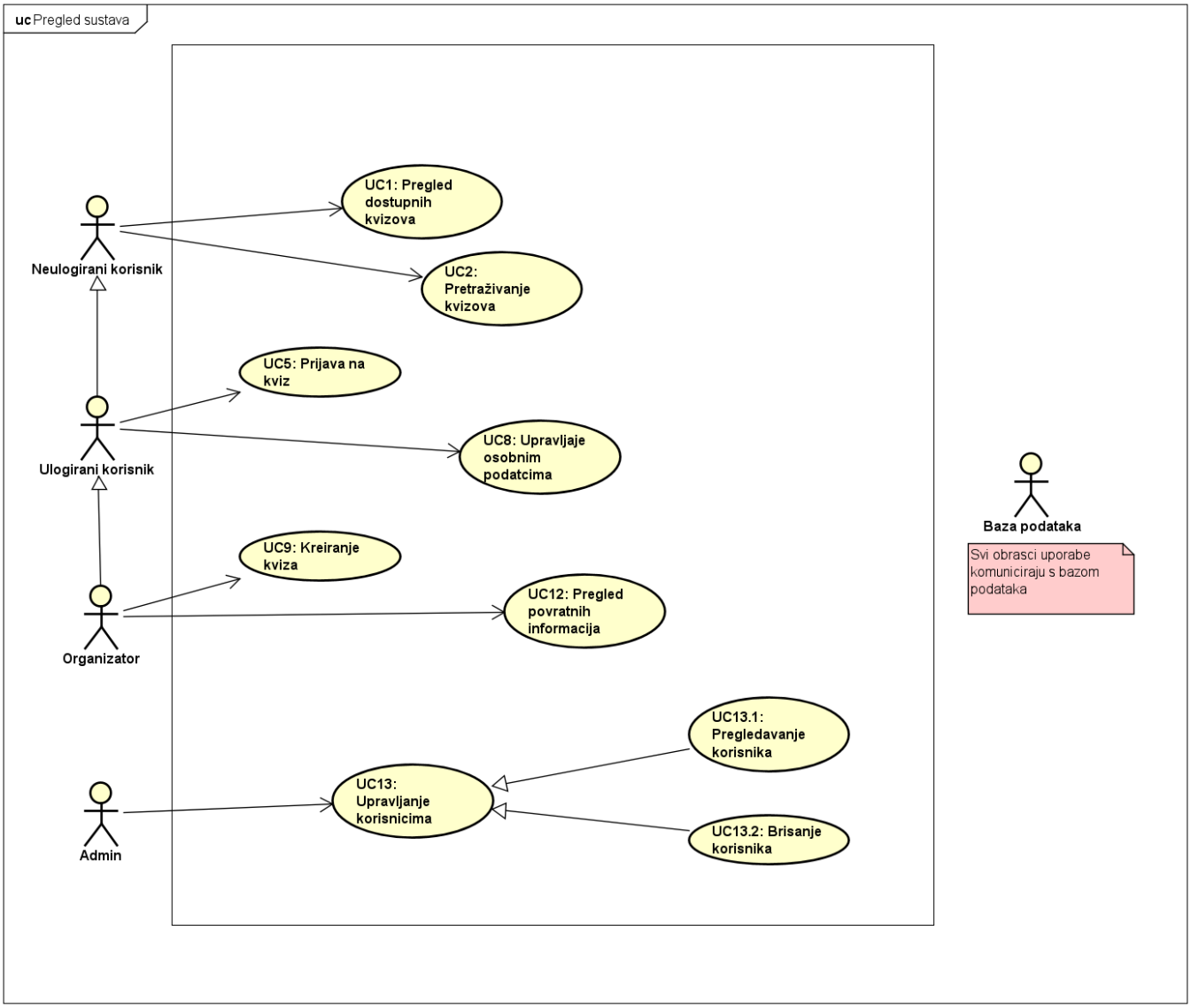
Neregistrirani korisnik je osoba koja samo pretražuje kvizove bez mogućnosti prijave ili sudjelovanja.

- **Funkcionalnosti koje neregistrirani korisnik može obavljati:**
    - **Pretraživanje kvizova**: Neregistrirani korisnik može pretraživati kvizove prema kategorijama, lokaciji, cijeni i broju ekipa.
        - **F-017:** Pretraga kvizova (neregistrirani)
    - **Pregled kvizova na karti**: Neregistrirani korisnik može pregledavati kvizove na karti.
        - **F-014:** Prikaz kvizova na karti (neregistrirani)

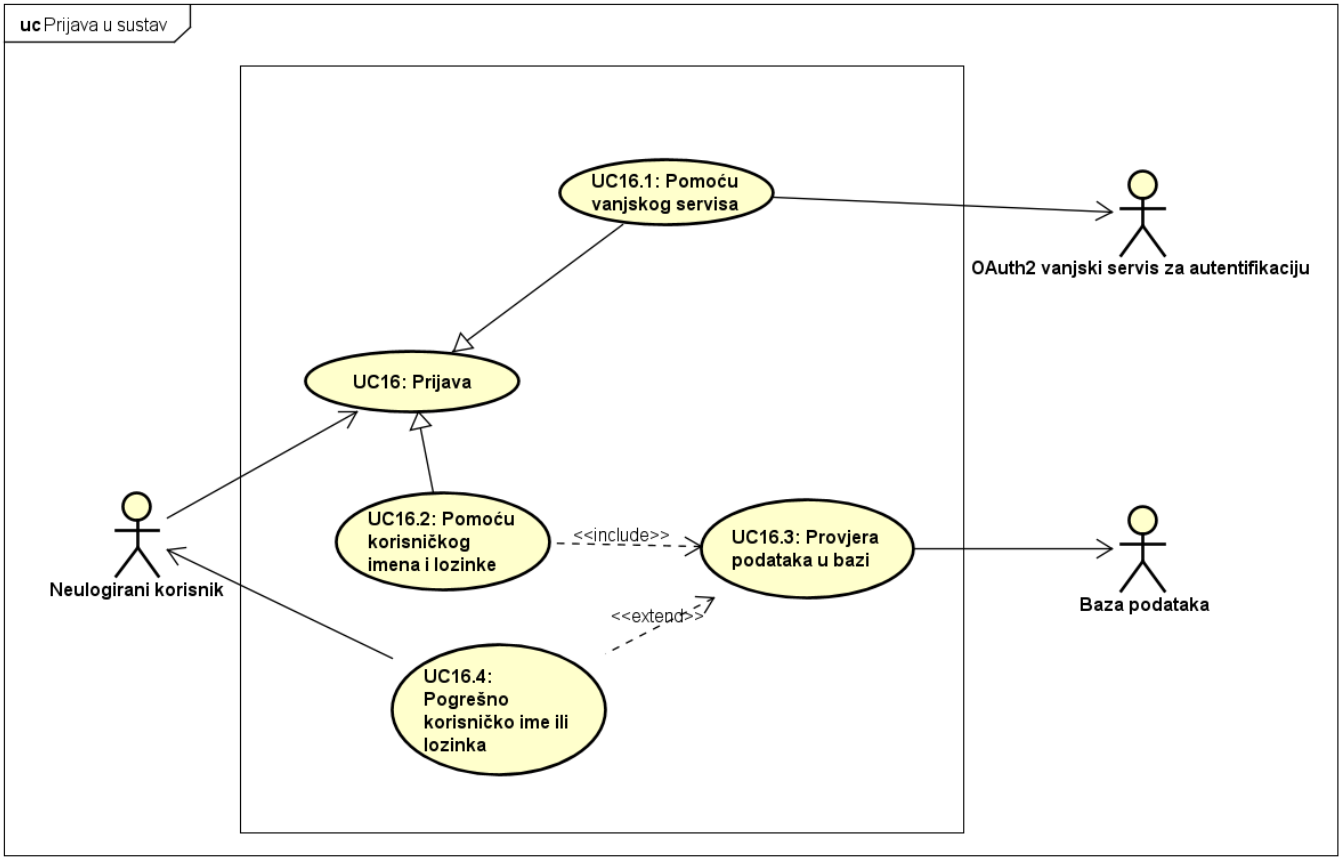
Obrasci uporabe

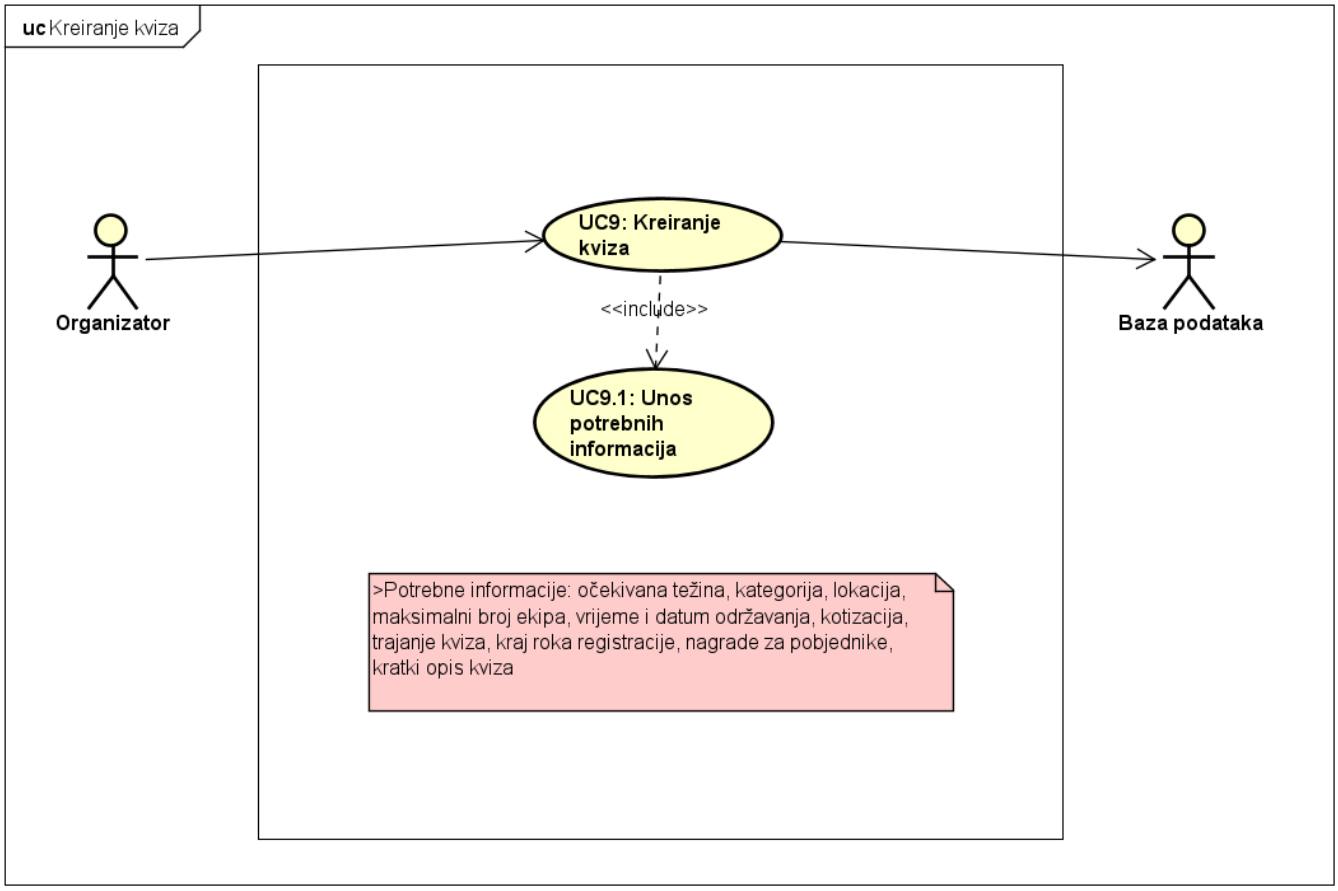
Dijagrami obrazaca uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

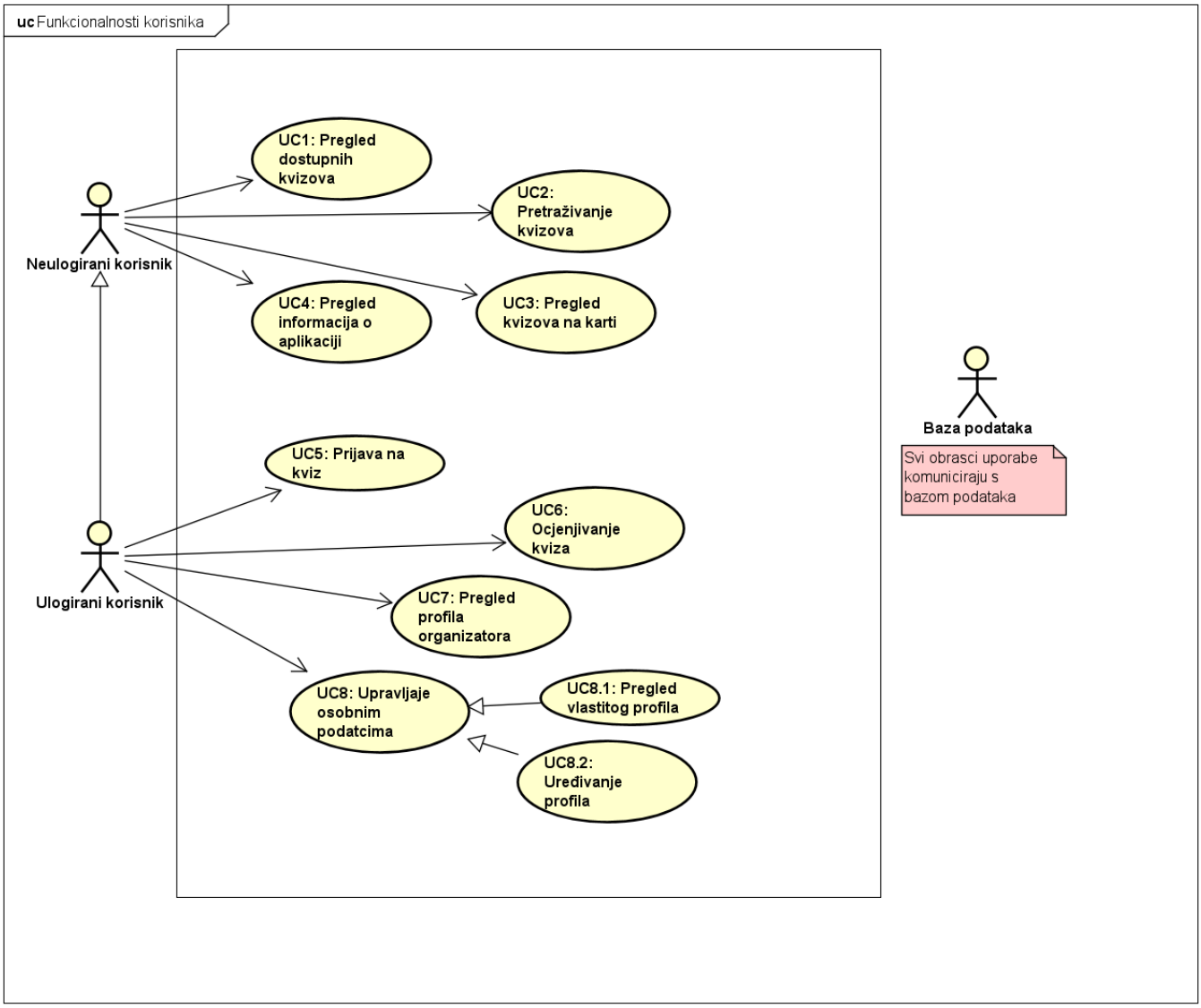


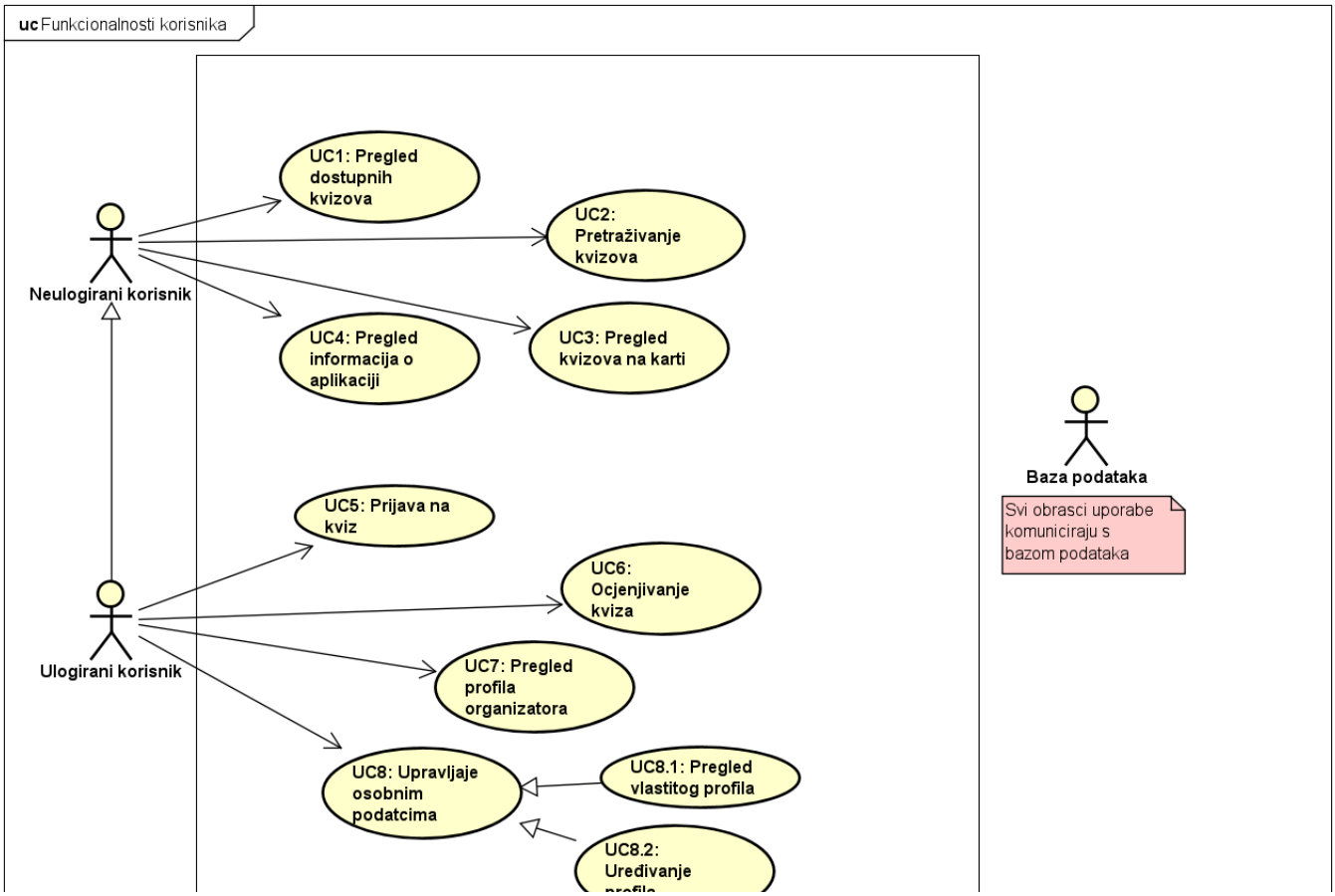
2. Dijagram obrazaca uporabe za ključne značajke

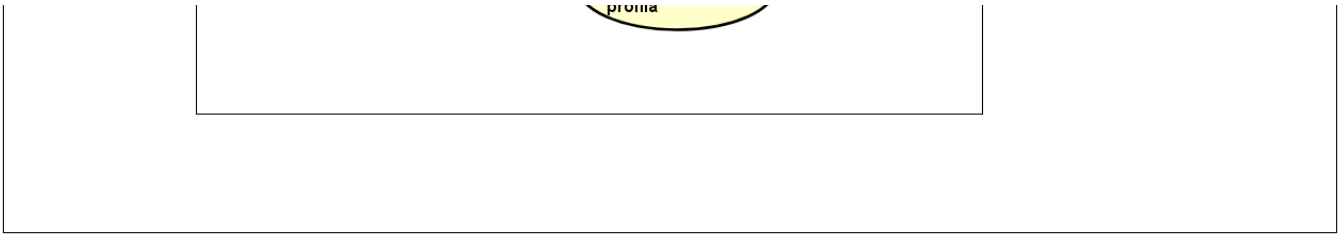


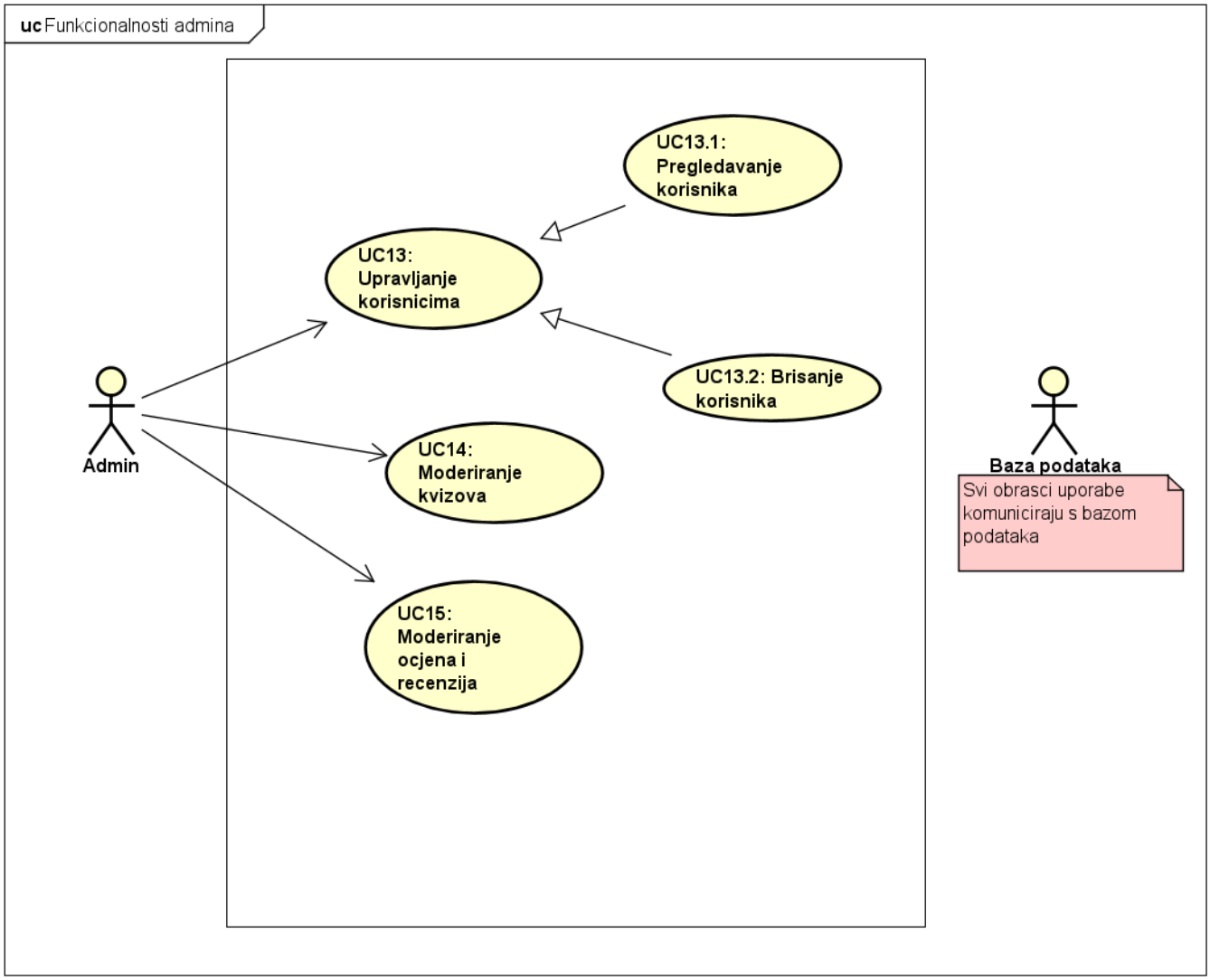


3. Dijagram obrazaca uporabe za korisničke uloge

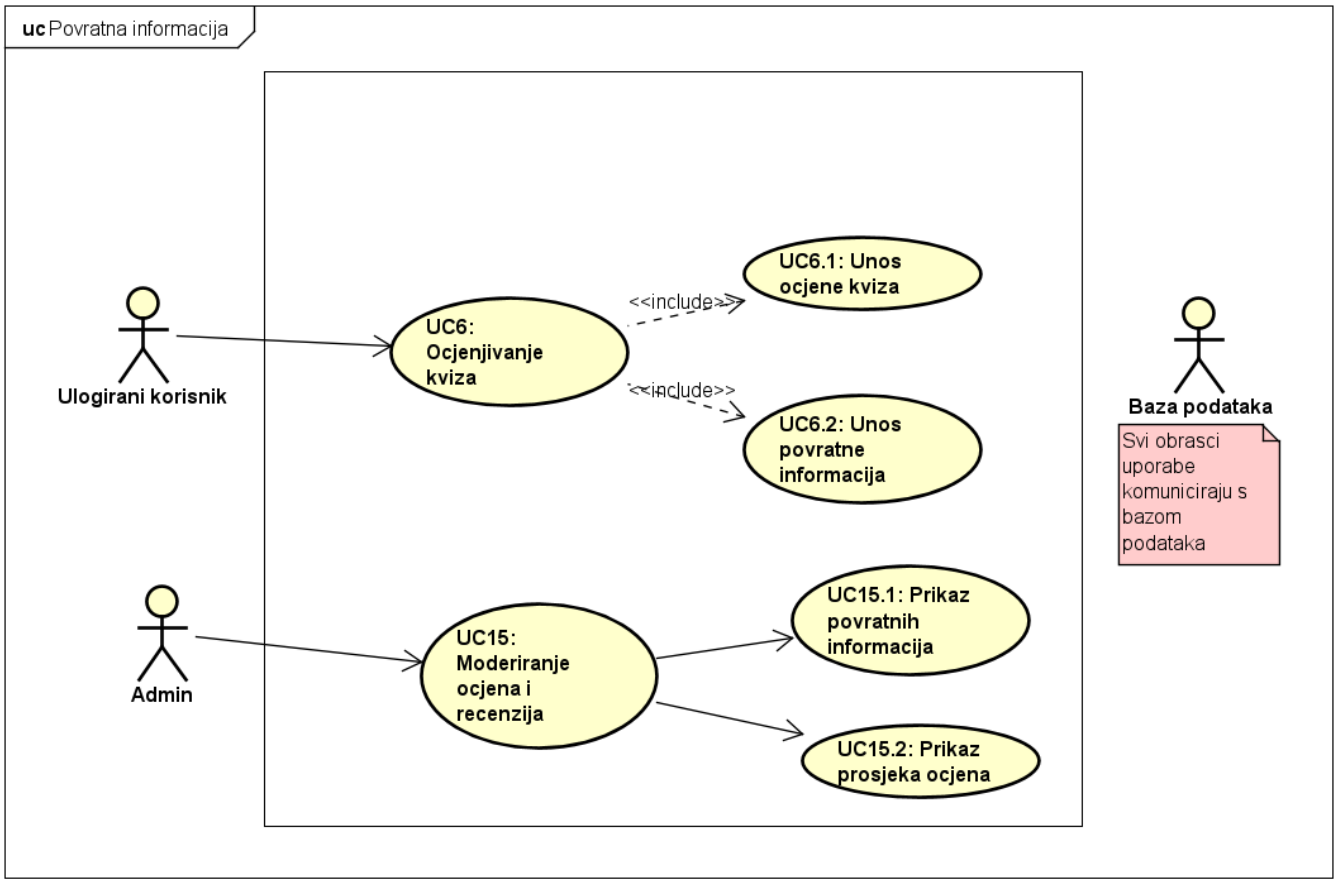

uc Funkcionalnosti korisnika
UC1: Pregled dostupnih kvizova
UC2: Pretraživanje kvizova
UC3: Pregled kvizova na karti
UC4: Pregled informacija o aplikaciji
Neulogirani korisnik
UC5: Prijava na kviz
UC6: Ocjenjivanje kviza
UC7: Pregled profila organizatora
UC8: Upravljaje osobnim podatcima
UC8.1: Pregled vlastitog profila
UC8.2: Uređivanje profila
Ulogirani korisnik
Baza podataka
Svi obrasci uporabe komuniciraju s bazom podataka


uc Funkcionalnosti korisnika
UC1: Pregled dostupnih kvizova
UC2: Pretraživanje kvizova
UC3: Pregled kvizova na karti
UC4: Pregled informacija o aplikaciji
Neulogirani korisnik
UC5: Prijava na kviz
UC6: Ocjenjivanje kviza
UC7: Pregled profila organizatora
UC8: Upravljaje osobnim podatcima
UC8.1: Pregled vlastitog profila
UC8.2: Uređivanje
Ulogirani korisnik
Baza podataka
Svi obrasci uporabe komuniciraju s bazom podataka

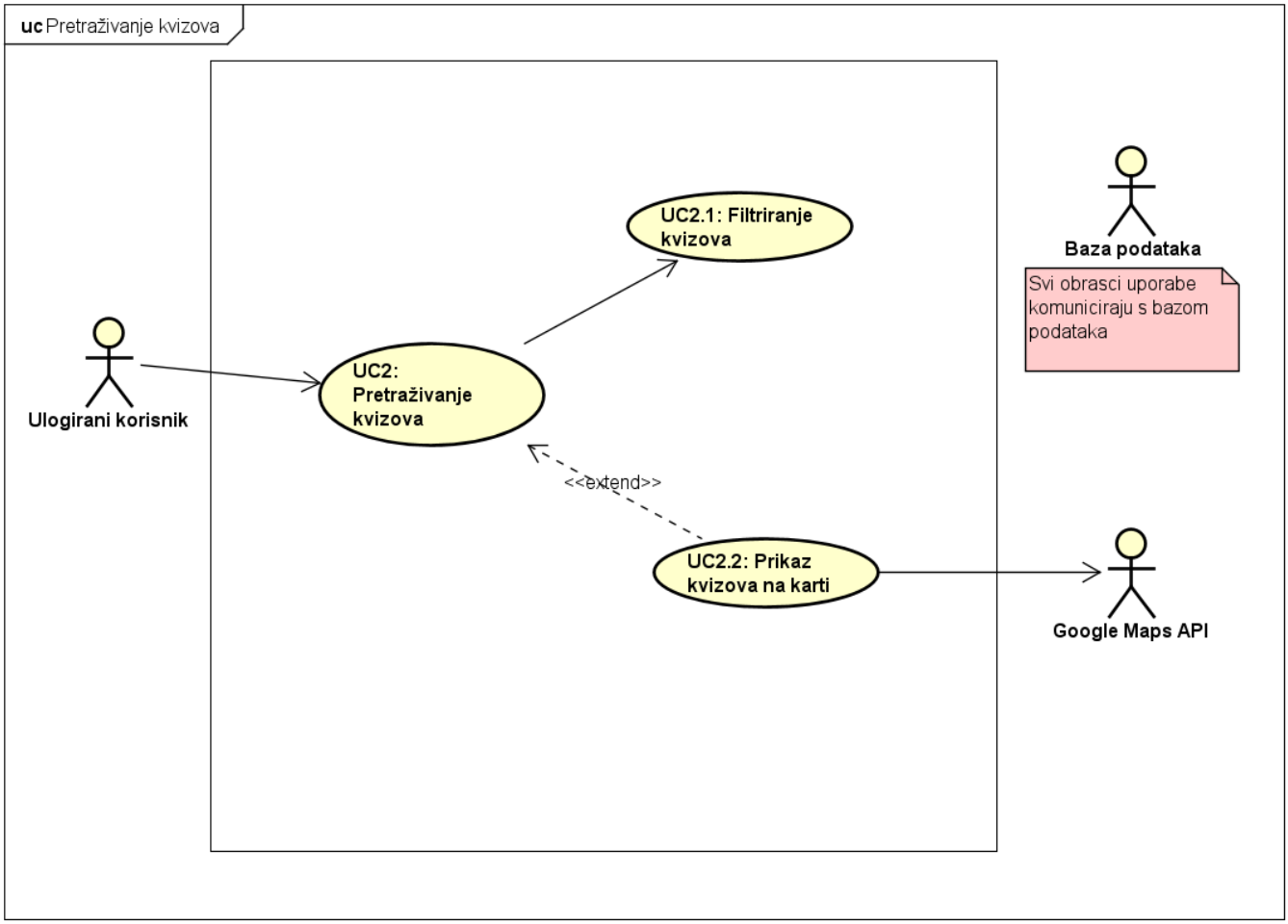




4. Dijagram obrazaca uporabe za osnovne poslovne procese



5. Dijagram obrazaca uporabe za kritične sustave i integracije



Opis obrazaca uporabe

UC1 - Pregled dostupnih kvizova

- **Glavni sudionik:** Neulogirani korisnik
- **Cilj:** Omogućiti korisniku pregled svih dostupnih kvizova.
- **Sudionici:** Neulogirani korisnik, Baza podataka
- **Preduvjet:** Korisnik nije ulogiran.
- **Opis osnovnog tijeka:**
    1. Korisnik odabire opciju za pregled dostupnih kvizova.
    2. Aplikacija šalje zahtjev bazi podataka za sve dostupne kvizove.
    3. Baza podataka vraća popis kvizova.
    4. Aplikacija prikazuje popis kvizova korisniku.
- **Opis mogućih odstupanja:**
    - **Odstupanje:** Ako baza podataka ne odgovara na zahtjev, aplikacija prikazuje poruku o grešci i nudi opciju za ponovno pokušavanje.

UC2 - Pretraživanje kvizova

- **Glavni sudionik:** Neulogirani korisnik
- **Cilj:** Omogućiti korisniku pretraživanje kvizova prema određenim kriterijima.
- **Sudionici:** Neulogirani korisnik, Baza podataka
- **Preduvjet:** Korisnik nije ulogiran.
- **Opis osnovnog tijeka:**
    1. Korisnik unosi kriterij pretraživanja i potvrđuje pretragu.
    2. Aplikacija šalje kriterije pretraživanja bazi podataka.
    3. Baza podataka vraća kvizove koji zadovoljavaju kriterij pretraživanja.
    4. Aplikacija prikazuje rezultate pretraživanja korisniku.
- **Opis mogućih odstupanja:**
    - **Odstupanje:** Ako nema rezultata pretraživanja, aplikacija prikazuje poruku "Nema pronađenih rezultata."

UC3 - Pregled kvizova na karti

- **Glavni sudionik:** Neulogirani korisnik
- **Cilj:** Omogućiti korisniku pregled kvizova na karti.
- **Sudionici:** Neulogirani korisnik, Baza podataka

- **Preduvjet:** Korisnik nije ulogiran.
- **Opis osnovnog tijeka:**
    1. Korisnik obabire opciju za prikaz karte.
    2. Aplikacija šalje zahtjev za dohvat kvizova bazi podataka.
    3. Baza podataka vraća podatke.
    4. Aplikacija prikazuje kvizove na karti korisniku.
- **Opis mogućih odstupanja:**
    - **Odstupanje:** Ako podaci o kvizovima nisu dostupni, korisnik može pokušati osvježiti stranicu.

UC4 - Pregled informacija o aplikaciji

- **Glavni sudionik:** Neulogirani korisnik
- **Cilj:** Informirati korisnika o svrsi i funkcionalnostima aplikacije.
- **Sudionici:** Neulogirani korisnik
- **Preduvjet:** Korisnik nije ulogiran.
- **Opis osnovnog tijeka:**
    1. Korisnik odabire opciju za pregled informacija o aplikaciji.
    2. Aplikacija prikazuje informacije o aplikaciji korisniku.
- **Opis mogućih odstupanja:**
    - Nema mogućih odstupanja.

UC5 - Prijava na kviz

- **Glavni sudionik:** Ulogirani korisnik
- **Cilj:** Omogućiti korisniku prijavu na kviz.
- **Sudionici:** Ulogirani korisnik, Baza podataka
- **Preduvjet:** Korisnik je ulogiran i pregledava dostupne kvizove.
- **Opis osnovnog tijeka:**
    1. Korisnik odabire kviz na koji se želi prijaviti.
    2. Aplikacija provjerava dostupnost kviza u bazi podataka.
    3. Ako je kviz dostupan, aplikacija dodaje korisnika na listu prijavljenih za taj kviz.
    4. Baza podataka potvrđuje uspješnu prijavu.
    5. Aplikacija prikazuje poruku o uspješnoj prijavi korisniku.
- **Opis mogućih odstupanja:**
    1. **Odstupanje:** Ako kviz nije dostupan, aplikacija prikazuje poruku "Kviz više nije dostupan" i vraća korisnika na listu dostupnih kvizova.

2. **Odstupanje:** Ako prijava ne uspije zbog tehničkog problema, aplikacija prikazuje poruku o grešci i nudi opciju za ponovno pokušavanje.

UC6 - Ocjenjivanje kviza

- **Glavni sudionik:** Ulogirani korisnik
- **Cilj:** Omogućiti korisniku da ocijeni kviz koji je završio.
- **Sudionici:** Ulogirani korisnik, Baza podataka
- **Preduvjet:** Korisnik je ulogiran i završio je kviz.
- **Opis osnovnog tijeka:**
  1. Korisnik pristupa opciji za ocjenjivanje kviza.
  2. Korisnik unosi ocjenu i/ili komentar na kviz.
  3. Aplikacija šalje ocjenu i komentar bazi podataka.
  4. Baza podataka pohranjuje ocjenu i komentar.
  5. Aplikacija potvrđuje korisniku da je ocjenjivanje uspješno završeno.
- **Opis mogućih odstupanja:**
  1. **Odstupanje:** Ako unos ocjene ne uspije zbog greške u vezi s bazom podataka, aplikacija prikazuje poruku o grešci i nudi mogućnost ponovnog slanja.
  2. **Odstupanje:** Ako korisnik pokuša ocijeniti kviz koji nije završio, aplikacija prikazuje poruku "Ocjenjivanje je moguće samo nakon završetka kviza."

UC7 - Pregled profila organizatora

- **Glavni sudionik:** Ulogirani korisnik
- **Cilj:** Omogućiti korisniku da pregleda profil organizatora.
- **Sudionici:** Ulogirani korisnik, Baza podataka
- **Preduvjet:** Korisnik je ulogiran i pregledava informacije o organizatoru.
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za pregled podataka organizatora.
  2. Aplikacija prikazuje podatke poput prosječne ocjene organizatorovih ocjena.
- **Opis mogućih odstupanja:**
  - **Odstupanje:** Ako podaci nisu dostupni, korisnik može pokušati osvježiti stranicu.

UC8 - Upravljanje osobnim podacima

- **Glavni sudionik:** Ulogirani korisnik
- **Cilj:** Omogućiti korisniku pregled, uređivanje ili brisanje svojih osobnih podataka.
- **Sudionici:** Ulogirani korisnik, Baza podataka

- **Preduvjet:** Korisnik je ulogiran.
- **Opis osnovnog tijeka:**
    1. Korisnik odabire opciju za upravljanje osobnim podacima.
    2. Aplikacija korisniku prikazuje njegov profil.
    3. Korisnik odabire željenu opciju i aplikacija izvršava odgovarajuću radnju.
- **Opis mogućih odstupanja:**
    1. **Odstupanje:** Ako baza podataka nije dostupna, aplikacija prikazuje poruku o grešci i nudi opciju za ponovno pokušavanje.
    2. **Odstupanje:** Ako korisnik unese nevažeće podatke, aplikacija prikazuje poruku o grešci i traži unos ispravnih podataka.
    3. **Odstupanje:** Ako korisnik pokuša obrisati račun bez potvrde, aplikacija traži potvrdu prije nastavka s brisanjem.

UC9 - Kreiranje kviza

- **Glavni sudionik:** Organizator
- **Cilj:** Omogućiti organizatoru kreiranje novog kviza.
- **Sudionici:** Organizator, Baza podataka
- **Preduvjet:** Organizator je ulogiran u sustav.
- **Opis osnovnog tijeka:**
    1. Organizator odabire opciju za kreiranje novog kviza.
    2. Aplikacija prikazuje obrazac za unos podataka o kvizu.
    3. Organizator unosi potrebne podatke za kviz i potvrđuje kreiranje.
    4. Aplikacija šalje podatke o kvizu u bazu podataka za pohranu.
    5. Baza podataka potvrđuje uspješno pohranjivanje kviza, a aplikacija prikazuje poruku o uspješnom kreiranju kviza.
- **Opis mogućih odstupanja:**
    1. **Odstupanje:** Ako unos podataka nije ispravan (npr. prazna polja), aplikacija prikazuje poruku o grešci i traži ispravljanje podataka.
    2. **Odstupanje:** Ako baza podataka ne odgovara, aplikacija prikazuje poruku o grešci i nudi opciju za spremanje podataka lokalno dok se ne uspostavi veza s bazom podataka.

UC10 - Postavljanje kvizova kao dio lige

- **Glavni sudionik:** Organizator
- **Cilj:** Omogućiti organizatoru postavljanje kvizova kao dio lige.
- **Sudionici:** Organizator, Baza podataka
- **Preduvjet:** Organizator je ulogiran u sustav i ima kreirane kvizove.

- **Opis osnovnog tijeka:**
    1. Organizator odabire kviz koji želi uključiti u ligu.
    2. Aplikacija daodaje kviz u ligu i podaci se spremaju u bazu podataka.
- **Opis mogućih odstupanja:**
    - **Odstupanje:** Ako podaci nisu dostupni, korisnik može pokušati osvježiti stranicu.

UC11 - Pregled ocjena kvizova

- **Glavni sudionik:** Organizator
- **Cilj:** Omogućiti organizatoru pregled ocjena kvizova.
- **Sudionici:** Organizator, Baza podataka
- **Preduvjet:** Organizator je ulogiran i ima kvizove.
- **Opis osnovnog tijeka:**
    1. Organizator odabire kviz za koji želi vidjeti ocjenu.
    2. Aplikacija prikazuje ocjenu.
- **Opis mogućih odstupanja:**
    - Nema mogućih odstupanja.

UC12 - Pregled povratnih informacija

- **Glavni sudionik:** Organizator
- **Cilj:** Omogućiti organizatoru pregled povratnih informacija koje su korisnici ostavili za njegove kvizove.
- **Sudionici:** Organizator, Baza podataka
- **Preduvjet:** Organizator je ulogiran i ima barem jedan kviz s prikupljenim povratnim informacijama.
- **Opis osnovnog tijeka:**
    1. Organizator odabire kviz za koji želi vidjeti povratne informacije.
    2. Aplikacija šalje zahtjev bazi podataka za povratne informacije korisnika za odabrani kviz.
    3. Baza podataka vraća povratne informacije o kvizu.
    4. Aplikacija prikazuje povratne informacije organizatoru.
- **Opis mogućih odstupanja:**
    1. **Odstupanje:** Ako nema povratnih informacija za odabrani kviz, aplikacija prikazuje poruku "Nema povratnih informacija za ovaj kviz."
    2. **Odstupanje:** Ako baza podataka nije dostupna, aplikacija prikazuje poruku o grešci i nudi opciju za ponovno pokušavanje kasnije.

UC13 - Upravljanje korisnicima

- **Glavni sudionik:** Admin
- **Cilj:** Omogućiti adminu pregled, uređivanje i brisanje korisničkih računa.
- **Sudionici:** Admin, Baza podataka
- **Preduvjet:** Admin je prijavljen u sustav i ima odgovarajuće ovlasti.
- **Opis osnovnog tijeka:**
    1. Admin odabire opciju za upravljanje korisnicima.
    2. Aplikacija prikazuje popis svih korisnika.
    3. Admin može odabrati korisnika za pregledavanje podataka, uređivanje informacija ili brisanje računa.
    4. Aplikacija prikazuje odabranu funkcionalnost (pregled, uređivanje ili brisanje) i traži potvrdu od admina.
    5. Baza podataka ažurira promjene i vraća potvrdu uspješnosti operacije.
- **Opis mogućih odstupanja:**
    1. **Odstupanje:** Ako baza podataka nije dostupna, aplikacija prikazuje poruku o grešci i nudi mogućnost ponovnog pokušaja.
    2. **Odstupanje:** Ako admin pokuša izbrisati korisnika koji ima aktivne kvizove, aplikacija prikazuje upozorenje i nudi mogućnost prijenosa kvizova na drugog korisnika prije brisanja.

UC14 - Moderiranje kvizova

- **Glavni sudionik:** Admin
- **Cilj:** Osigurati da kvizovi na platformi budu u skladu s pravilima.
- **Sudionici:** Admin, Baza podataka
- **Preduvjet:** Admin je prijavljen i ima pristup listi kvizova.
- **Opis osnovnog tijeka:**
    1. Admin otvara popis svih kvizova na platformi.
    2. Admin odabire kviz za moderiranje.
    3. Aplikacija prikazuje sve relevantne informacije o odabranom kvizu.
    4. Admin može urediti ili ukloniti kviz prema potrebi.
    5. Aplikacija ažurira promjene u bazi podataka.
- **Opis mogućih odstupanja:**
    1. **Odstupanje:** Ako baza podataka ne odgovara, aplikacija prikazuje poruku o grešci i nudi ponovno pokušavanje kasnije.

UC15 - Moderiranje ocjena i recenzija

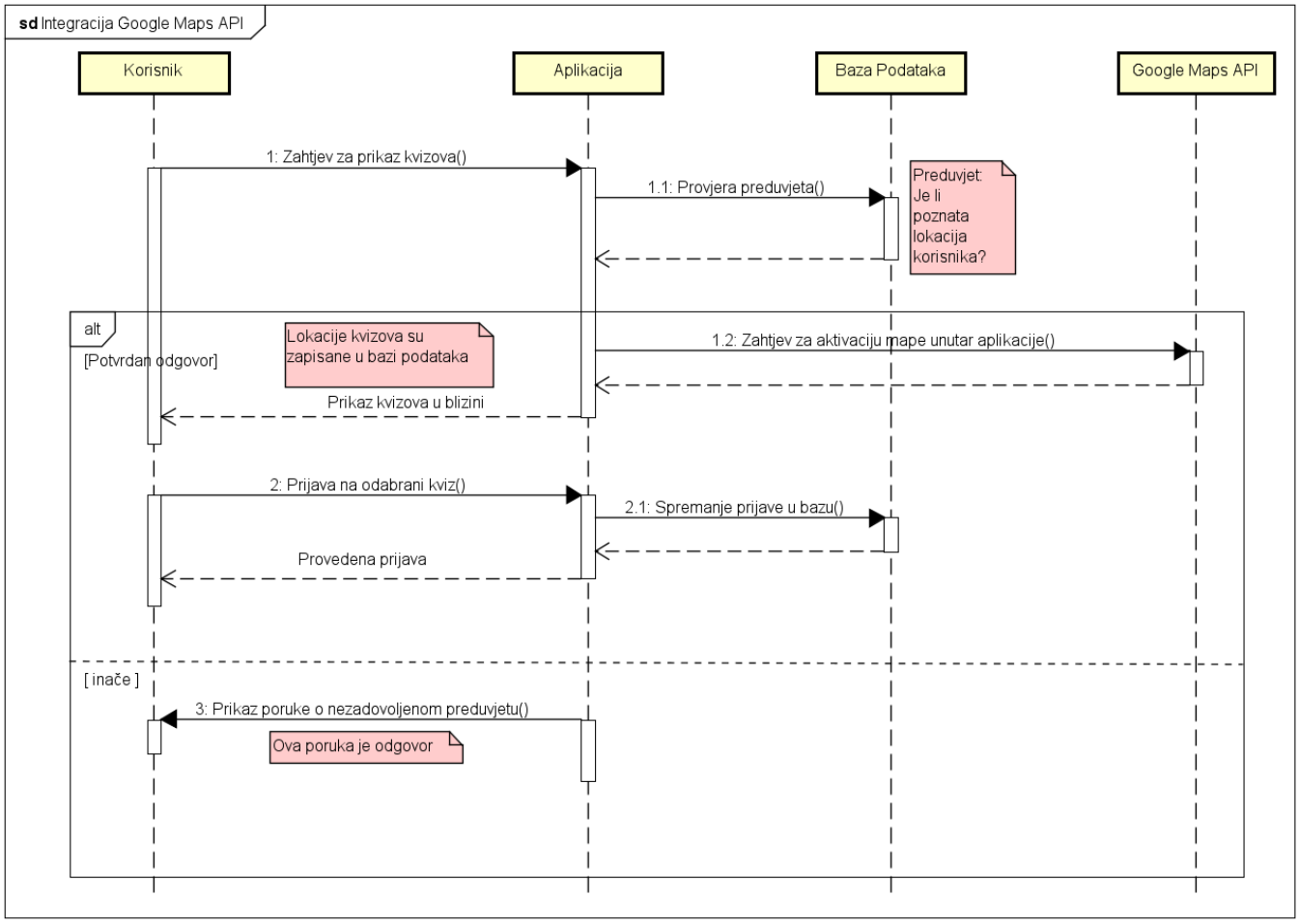
- **Glavni sudionik:** Admin
- **Cilj:** Izbrisati neprimjerene rezencije.

- **Sudionici:** Admin, Baza podataka
- **Preduvjet:** Admin je prijavljen u sustav i ima pristup podacima.
- **Opis osnovnog tijeka:**
    1. Admin odabire opciju za brisanje rezencije.
    2. Aplikacija briše odabranu recenziju i ocjenu i ta promjena se sprema u bazi podataka.
- **Opis mogućih odstupanja:**
    - Nema mogućih odstupanja.

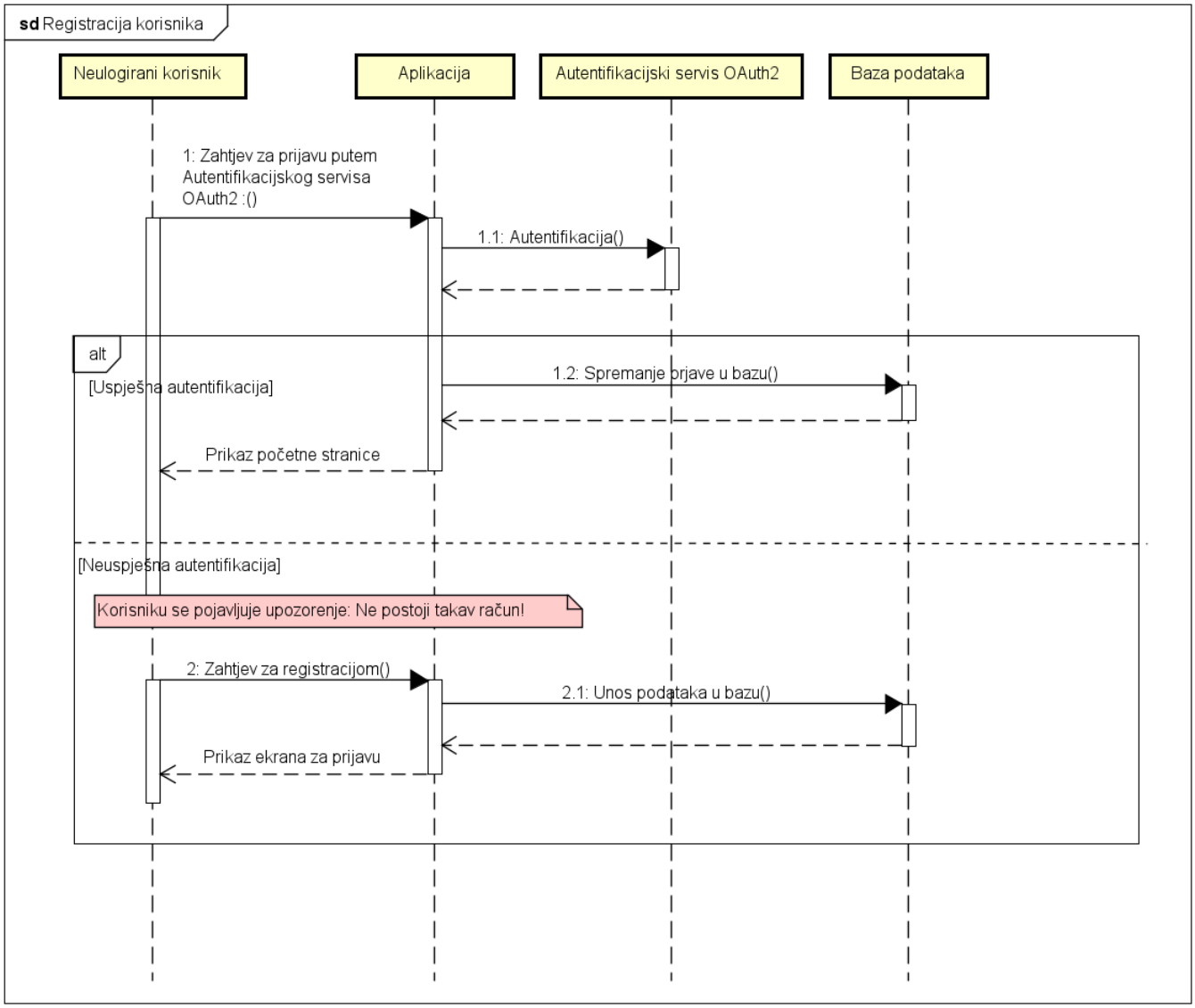
UC16 – Prijava u sustav

- **Glavni sudionik:** Neulogirani korisnik
- **Cilj:** Omogućiti korisniku prijavu u sustav.
- **Sudionici:** Neulogirani korisnik, Baza podataka, Vanjski servis za autentifikaciju
- **Preduvjet:** Korisnik nije ulogiran.
- **Opis osnovnog tijeka:**
    1. Korisnik pokreće proces prijave u sustav.
    2. Sustav prikazuje opcije prijave.
    3. Korisnik bira način prijave: putem vanjskog servisa ili putem korisničkog imena i lozinke.
    4. Prijava završava uspješno ili neuspješno, ovisno o provjeri podataka.
- **Opis mogućih odstupanja:**
    1. **Odstupanje:** Ako korisnik unese pogrešne podatke, dobiva obavijest o pogrešci i ima priliku ponoviti prijavu.

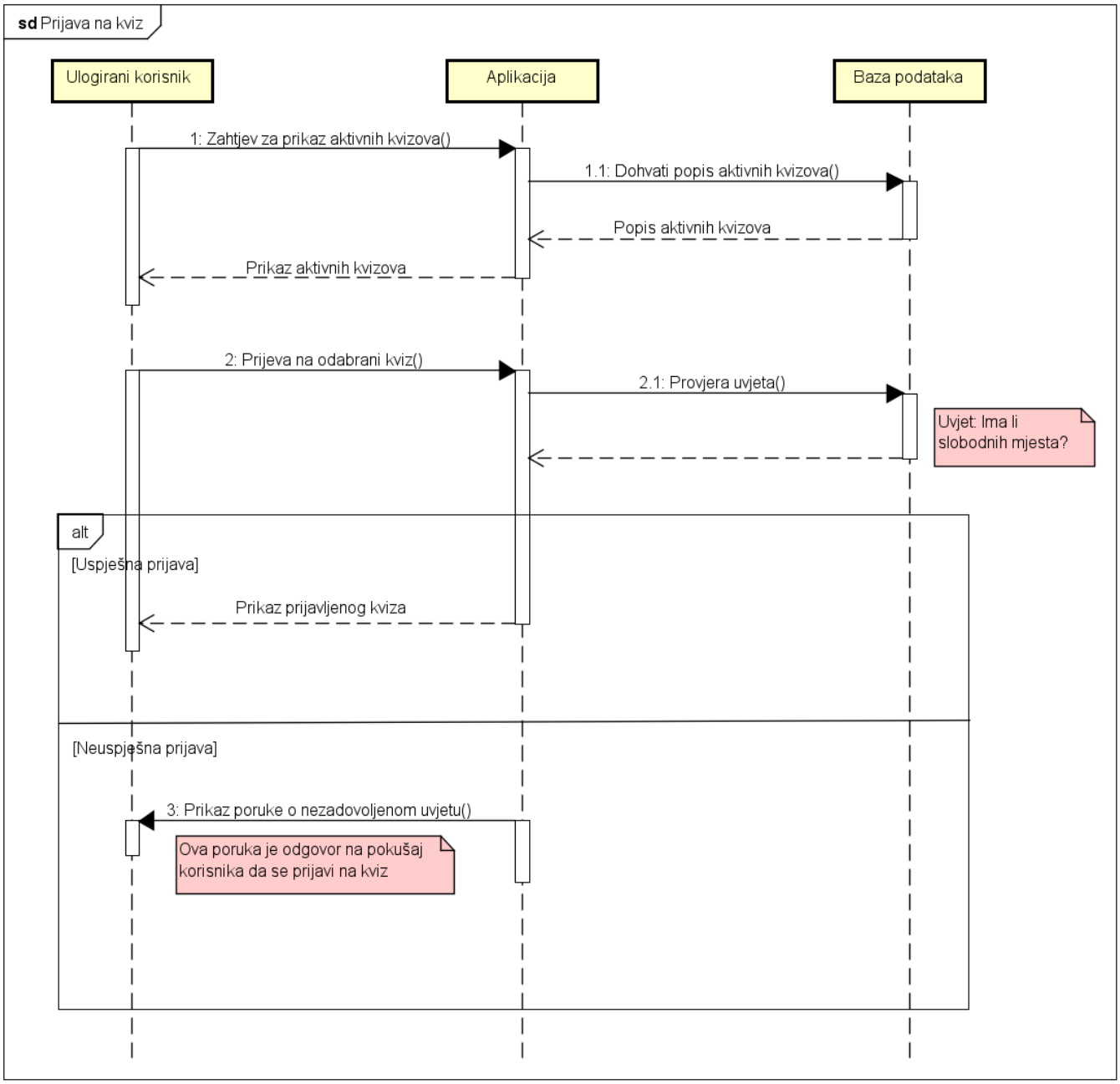
Sekvencijski dijagrami



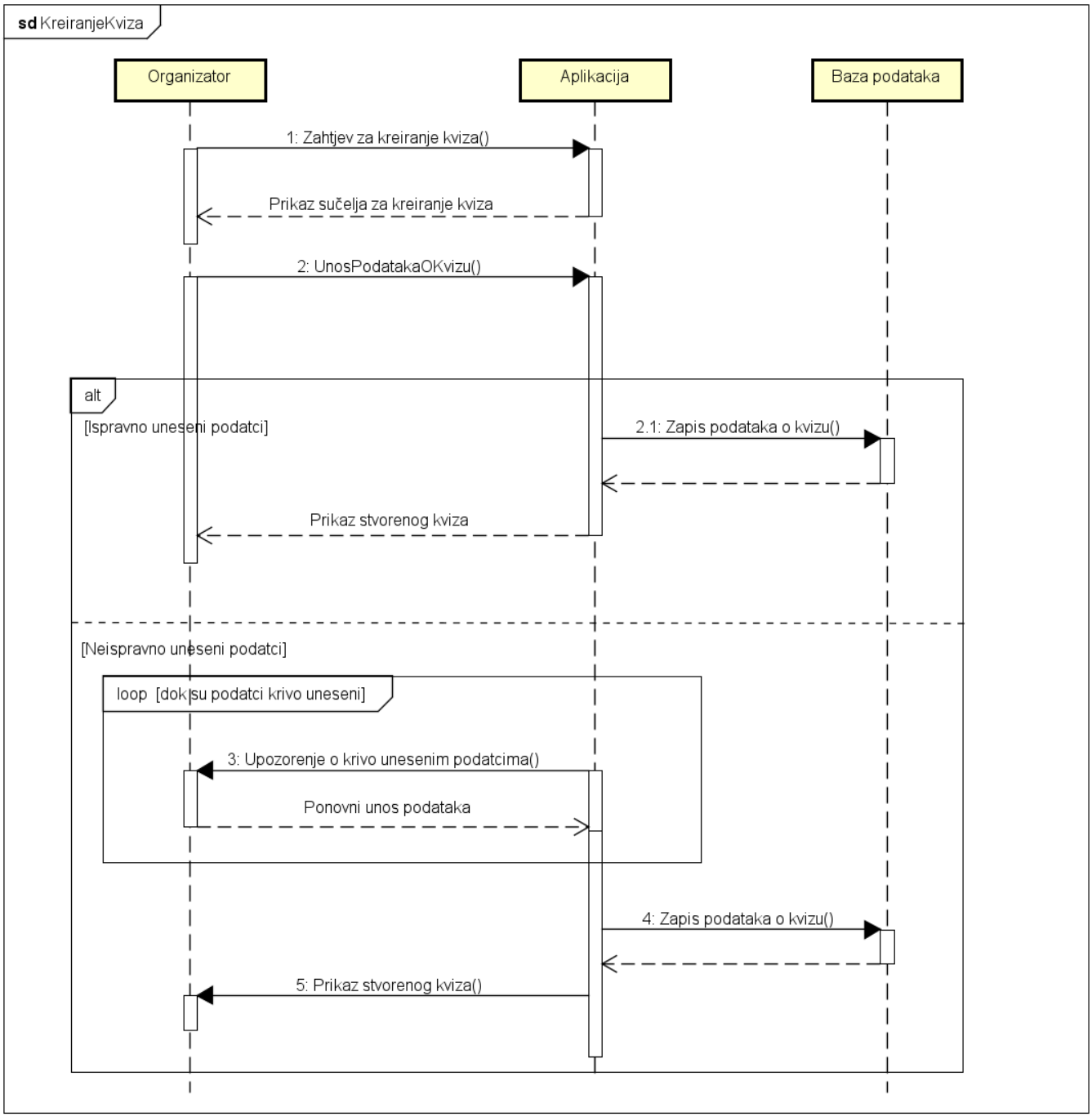
Korisnik putem aplikacije šalje zahtjev za prikaz dostupnih kvizova u svojoj blizini.
Aplikacija provjerava zadovoljava li korisnik preduvjete za prikaz kvizova, ova provjera uključuje slanje zahtjeva bazi podataka kako bi se utvrdilo je li poznata lokacija korisnika.
Baza podataka obrađuje zahtjev i odgovara aplikaciji informirajući je li poznata lokacija korisnika.
Ako je preduvjet zadovoljen (tj. poznata je lokacija korisnika): aplikacija prikazuje mapu koristeći Google Maps API s lokacijama kvizova koji su zabilježeni u bazi podataka.
Korisnik ima mogućnost odabira jednog od prikazanih kvizova te prijave na njega, a aplikacija bilježi prijavu i šalje povratnu informaciju korisniku da je prijava uspješno provedena.
Ako preduvjet nije zadovoljen (tj. nije poznata lokacija korisnika): aplikacija prikazuje poruku korisniku o nemogućnosti prikaza kvizova u blizini jer nije poznata njegova lokacija.



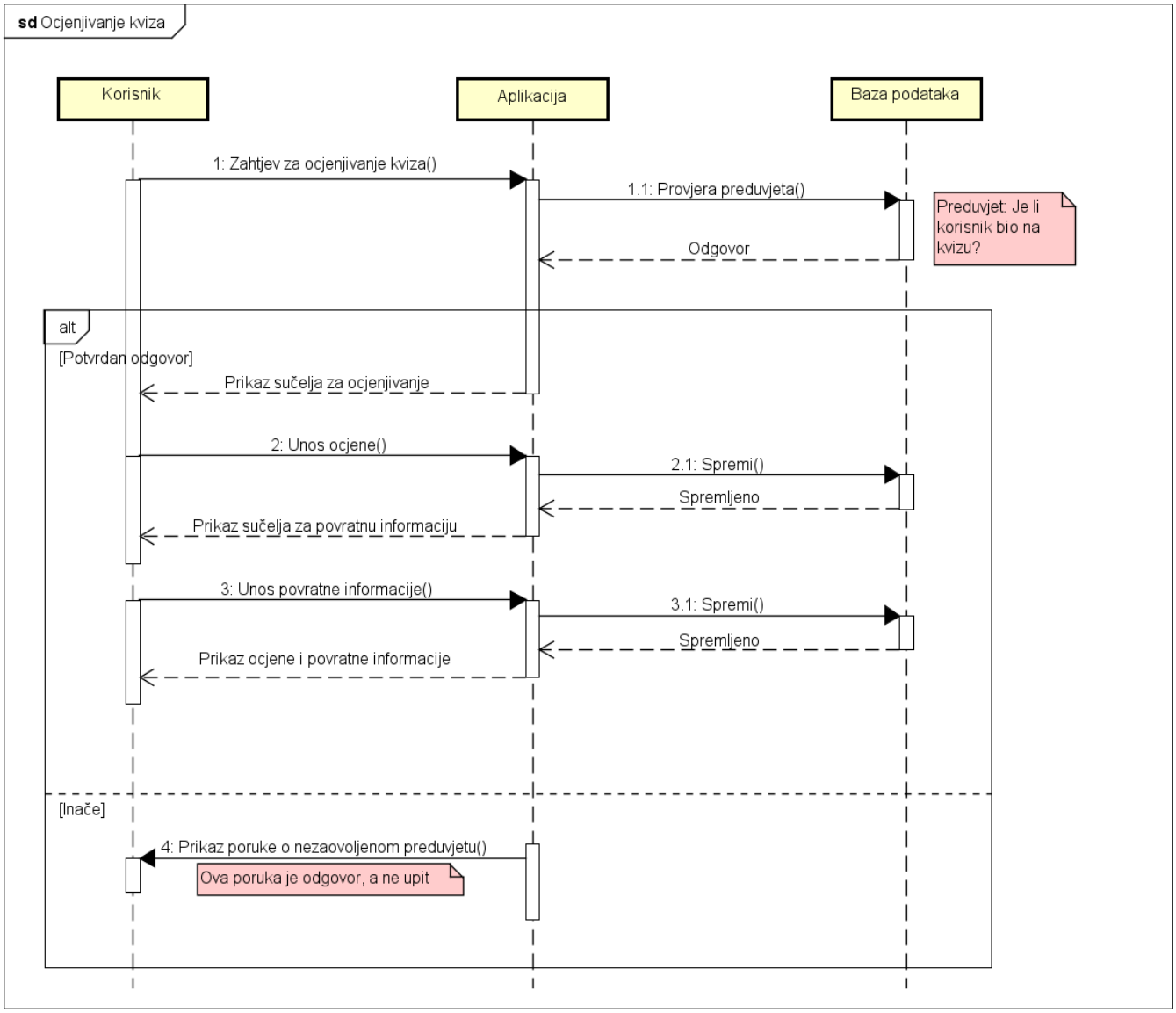
Korisnik pokušava pristupiti aplikaciji, šaljući zahtjev za prijavu pomoću autentifikacijskog servisa OAuth2. Aplikacija preusmjerava zahtjev autentifikacijskom servisu OAuth2 radi provjere korisničkog računa. Servis provodi autentifikaciju i šalje odgovor aplikaciji.
Ako je autentifikacija uspješna, aplikacija prikazuje početnu stranicu korisniku. Proces prijave je završen, a korisnik sada ima pristup aplikaciji.
Ako autentifikacija nije uspješna (korisnički račun ne postoji), korisniku se prikazuje upozorenje s porukom: "Ne postoji takav račun!"
Nakon upozorenja, korisnik ima mogućnost započeti postupak registracije.
Korisnik šalje zahtjev za registracijom kako bi kreirao novi račun. Aplikacija prima zahtjev za registracijom i unosi korisničke podatke u bazu podataka.
Nakon uspješnog unosa podataka, aplikacija prikazuje ekran za prijavu.



Ulogirani korisnik šalje zahtjev za prikaz aktivnih kvizova prema aplikaciji.
Aplikacija prima zahtjev korisnika i šalje zahtjev bazi podataka kako bi dohvatila popis aktivnih kvizova, aplikacija potom prikazuje taj popis korisniku.
Nakon što korisnik odabere kviz iz prikazanog popisa, šalje zahtjev za prijavu na taj odabrani kviz.
Aplikacija prima korisnikov zahtjev za prijavu i šalje zahtjev bazi podataka za provjeru slobodnih mjesta.
Ako postoji dovoljno slobodnih mjesta, prijava je uspješna, a aplikacija korisniku prikazuje potvrdu o prijavi na kviz.
Ako nema dovoljno slobodnih mjesta, aplikacija korisniku prikazuje poruku o neuspješnoj prijavi.



Organizator šalje zahtjev za kreiranje kviza. Aplikacija prikazuje sučelja za unos podataka potrebnih za kreiranje kviza. Organizator unosi podatke o kvizu.
Kada su podaci ispravno uneseni, aplikacija šalje podatke o kvizu na zapisivanje u bazu podataka, zatim prikazuje organizatoru potvrdu o uspješnom kreiranju kviza.
Ako su podaci pogrešno uneseni, aplikacija šalje upozorenje organizatoru o greškama u unesenim podacima, organizator potom ima mogućnost ispraviti podatke i ponovno ih unijeti. Aplikacija će slati upozorenje sve dok organizator ne unese podatke ispravno.
Nakon što su ispravljeni podaci uneseni, zapisuju se u bazu podataka.



Korisnik pokreće proces ocjenjivanja kviza slanjem zahtjeva aplikaciji. Aplikacija prima ovaj zahtjev i pokreće provjeru preduvjeta da bi ustanovila ispunjava li korisnik uvjete za ocjenjivanje kviza.
Aplikacija šalje upit bazi podataka, baza provjerava je li korisnik sudjelovao u kvizu kako bi mogao ostaviti ocjenu.
Ako je korisnik ispunio preduvjet, aplikacija prikazuje korisniku sučelje za ocjenjivanje kviza. Korisnik unosi ocjenu kviza, potom aplikacija korisniku prikazuje sučelje za unos povratne informacije. Korisnik unosi povratnu informaciju o kvizu, a aplikacija prikazuje korisniku sažetak unesenih ocjena i povratnih informacija.
Ako korisnik nije ispunio preduvjet (nije bio na kvizu), aplikacija prikazuje korisniku poruku o nezadovoljenom preduvjetu, ova poruka objašnjava da korisnik nije bio na kvizu i stoga ne može unijeti ocjenu ili povratne informacije.

Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Obrasci uporabe	Funkcionalni zahtjevi
UC1: Pregled dostupnih kvizova	F-013, F-016

Obrasci uporabe	Funkcionalni zahtjevi
UC2: Pretraživanje kvizova	F-004, F-017
UC3: Pregled kvizova na karti	F-005, F-014
UC4: Pregled informacija o aplikaciji	-
UC5: Prijava na kviz	F-006
UC6: Ocjenjivanje kviza	F-008
UC7: Pregled profila organizatora	F-007
UC8: Upravljanje osobnim podatcima	-
UC9: Kreiranje kviza	F-001
UC10: Postavljanje kviza kao dio lige	F-003
UC11: Pregled ocjena kvizova	F-002
UC12: Pregled povratnih informacija	-
UC13: Upravljanje korisnicima	F-010
UC14: Moderiranje kvizova	F-012
UC15: Moderiranje ocjena i recenzija	F-011
UC16: Prijava	F-015

Arhitektura sustava

Odabrana arhitektura omogućava visoku modularnost i skalabilnost sustava, čineći ga pogodnim za daljnje proširivanje i integraciju novih funkcionalnosti, dok održava jasnu podjelu odgovornosti između različitih slojeva aplikacije.

Opis arhitekture

Arhitektura sustava Quiz Finder temelji se na klijent-poslužitelj modelu, gdje su odgovornosti sustava jasno podijeljene između frontend i backend komponenti. Sustav je osmišljen kao web aplikacija koja koristi moderne tehnologije za razvoj skalabilnog, fleksibilnog i održivog sustava za upravljanje pub kvizovima.

Sustav je podijeljen u tri glavne komponente:

Frontend:
Tehnologija: React.js
Uloga: Frontend aplikacija je korisničko sučelje sustava, koje omogućava korisnicima interakciju s kvizovima. Korištenjem React.js omogućavamo dinamičko i responzivno sučelje koje se brzo učitava, a također omogućava stvaranje jedne stranice (Single Page Application, SPA). Komunikacija s backendom: Frontend aplikacija komunicira s backendom putem RESTful API-ja (koji je implementiran u backendu). Ovaj API omogućuje dohvat podataka o kvizovima, sudionicima, organizatorima, kao i prijavu na kvizove ili pregled rezultata. Responsivnost: Frontend je dizajniran tako da bude responzivan i prilagodljiv različitim uređajima, od desktop računala do mobilnih telefona.

Backend:
Tehnologija: Python Django
Uloga: Backend aplikacija je odgovorna za poslovnu logiku sustava, upravljanje podacima i autentifikaciju korisnika. Backend pruža RESTful API koji omogućuje komunikaciju između frontenda i baze podataka, kao i integraciju s vanjskim uslugama (kao što je Google Maps API ili vanjski sustavi za autentifikaciju korisnika poput Googlea i Facebooka). Autentifikacija i autorizacija: Autentifikacija korisnika se realizira putem OAuth2 protokola, što omogućava korisnicima prijavu putem vanjskih servisa poput Googlea ili Facebooka. OAuth2 omogućava siguran pristup podacima korisnika bez potrebe za pohranjivanjem korisničkih lozinki u sustavu. Poslovna logika: Backend upravlja svim poslovnim procesima kao što su kreacija i ažuriranje kvizova, upravljanje korisnicima (organizatorima i sudionicima), praćenje rezultata kvizova te slanje notifikacija korisnicima. Sigurnost: Korištenje standarda poput OAuth2 omogućava siguran pristup podacima, dok se svi podaci prenose putem HTTPS protokola kako bi se osigurala privatnost i sigurnost korisničkih podataka.

Baza podataka:
Tehnologija: PostgreSQL (relacijska baza podataka)
Uloga: PostgreSQL je odabrana kao relacijska baza podataka koja omogućava pouzdano i efikasno pohranjivanje podataka o korisnicima, kvizovima, prijavama i rezultatima. Relacijska baza je pogodna jer omogućava složene upite, transakcije i osigurava integritet podataka, što je ključno za praćenje povijesti kvizova i korisničkih aktivnosti. Dizajn baze: Baza podataka uključuje tablice za korisnike (organizatore, sudionike), kvizove (s pitanjima i odgovorima), rezultate (s podacima o ocjenama i vremenu), kao i tablice za upravljanje prijavama korisnika na kvizove. Relacijski model baze omogućava složene relacije među podacima (npr. povezivanje korisnika s kvizovima, sudjelovanje na više kvizova, praćenje rezultata). Vanjske integracije:

Google Maps API: Ovaj API koristi se za prikazivanje lokacija pubova ili kvizova na karti, omogućujući korisnicima da lakše pronađu mjesta na kojima se održavaju kvizovi. OAuth2 za autentifikaciju: Korisnici se mogu prijaviti putem vanjskih servisa kao što su Google ili Facebook, što im omogućuje jednostavan pristup bez potrebe za pamćenjem dodatnih lozinki. E-mail i push notifikacije: Backend može slati notifikacije korisnicima putem e-maila ili push notifikacija o nadolazećim kvizovima, rezultatima i novostima vezanim za aplikaciju.

Obrazloženje odabira arhitekture

Arhitektura sustava temelji se na klijent-poslužitelj modelu, koji je odabran zbog svoje jednostavnosti, fleksibilnosti i skalabilnosti. Klijent-poslužitelj model omogućuje jasnu podjelu odgovornosti između frontend i backend komponenata, što olakšava održavanje i proširivanje sustava.

Razlog zašto je ovaj model odabran uključuje sljedeće prednosti:

Jednostavnost implementacije: Klijent-poslužitelj model je vrlo jednostavan za implementaciju i omogućuje brzi razvoj aplikacije. Frontend i backend komponente mogu se razvijati paralelno, što ubrzava cijeli proces. **Fleksibilnost**: Ovaj model omogućuje daljnje proširivanje sustava. Na primjer, backend može biti proširen dodavanjem novih funkcionalnosti, kao što je integracija s novim API-jima, dok frontend može biti unaprijeđen s novim korisničkim sučeljem bez značajnih promjena na backendu. Odvajanje odgovornosti: Jasno odvajanje odgovornosti između frontend i backend sloja omogućava bolju organizaciju koda, smanjenje složenosti i lakše održavanje aplikacije.

Skalabilnost: Klijent-poslužitelj model omogućava skaliranje sustava. Na primjer, ako se poveća broj korisnika ili zahtjeva prema backendu, backend se može skalirati neovisno, bez potrebe za promjenom frontend komponente. Alternativa koja je razmatrana bila je mikrouslužna arhitektura, no zbog složenosti i potrebe za višestrukim timovima, odlučeno je da za sada koristi klijent-poslužitelj model, jer pruža optimalan balans između fleksibilnosti, jednostavnosti i potreba sustava.

Organizacija sustava na visokoj razini

Sustav Quiz Finder organiziran je na visokoj razini u tri ključne komponente:

Frontend (React.js) – Komponenta koja omogućava korisnicima interakciju s aplikacijom. Korištenjem React.js, omogućavamo brzu i responzivnu aplikaciju koja se može lako prilagoditi različitim uređajima i veličinama zaslona.

Backend (Django) – Poslovna logika sustava. Backend upravlja svim procesima vezanim za korisnike, kvizove i rezultate. Osigurava RESTful API za komunikaciju s frontendom, a podaci se pohranjuju u relacijsku bazu podataka (PostgreSQL).

Baza podataka (PostgreSQL) – Pohrana podataka o korisnicima, kvizovima, prijavama i rezultatima. Relacijska baza omogućava efikasnu pohranu i upitavanje podataka, osiguravajući integritet podataka i omogućujući napredne upite.

Vanjski servisi – Integracija s vanjskim servisima poput Google Maps API za prikazivanje lokacija i OAuth2 za autentifikaciju.

Komunikacija između frontenda i backenda ostvaruje se putem REST API-ja koji omogućava razmjenu podataka i sinhronizaciju između sustava.

Organizacija aplikacije

Aplikacija je podijeljena u dva sloja:

Frontend sloj:
Korisničko sučelje: Svi podaci koji se prikazuju korisnicima dolaze s backend servera putem REST API-ja. Frontend aplikacija omogućava korisnicima da pretražuju kvizove, prijavljuju se na njih, pregledavaju rezultate i interagiraju s drugim korisnicima. Responsivnost: Koristi React.js, koji omogućava aplikaciji da bude responzivna i prilagodljiva različitim uređajima.

Backend sloj:
Poslovna logika: Backend aplikacija upravlja svim ključnim poslovnim procesima, uključujući autentifikaciju, autorizaciju, upravljanje kvizovima i rezultatima, kao i interakciju s bazom podataka. **REST API**: Backend nudi REST API koji omogućava frontend aplikaciji pristup podacima o korisnicima, kvizovima i rezultatima.

Baza podataka

Vrsta i implementacija baze podataka
Za pohranu podataka u sustavu "Quiz Finder", odabrali smo relacijsku bazu podataka temeljenju na PostgreSQL. PostgreSQL je odabran zbog svoje pouzdanosti, skalabilnosti i sposobnosti obrade velikih količina podataka, što je ključno za naš sustav koji će upravljati podacima o korisnicima, kvizovima, rezultatima, te korisničkim interakcijama.

Zašto PostgreSQL?
Integritet podataka: PostgreSQL omogućuje upotrebu ACID principa (Atomicity, Consistency, Isolation, Durability), što osigurava visok stupanj pouzdanosti pri radu s transakcijama.
Podrška za složene upite: Relacijska priroda baze omogućava jednostavno postavljanje upita koji kombiniraju podatke iz više tablica (npr. korisnici, kvizovi, rezultati).
Fleksibilnost i skalabilnost: PostgreSQL podržava velike baze podataka, a skalabilnost omogućuje lakše proširenje sustava kako rastu zahtjevi.
Sigurnost: PostgreSQL pruža visoku razinu sigurnosti, uključujući enkripciju podataka i robustne opcije kontrole pristupa. Glavne komponente baze podataka

Baza podataka u ovom sustavu sastoji se od nekoliko ključnih komponenata, među kojima su **tablice** i **relacije**.

Opis tablice

- **User**: Tablica koja sadrži informacije o korisnicima, uključujući ime, email, te podatke za autentifikaciju putem OAuth usluga. Svaki korisnik je povezan s određenom ulogom preko atributa `role_id`.
- **Role**: Sadrži različite uloge korisnika (npr. organizator, sudionik, administrator), definirajući prava pristupa i privilegije unutar aplikacije.
- **FavoriteOrganizer**: Omogućuje korisnicima da označe organizatore kao favorite, čime se stvara brži pristup omiljenim organizatorima.
- **Quiz**: Sadrži osnovne informacije o kvizovima kao što su naziv, opis, kategorija, mjesto održavanja, i detalji poput maksimalnog broja timova i naknade za sudjelovanje.

- **Team**: Predstavlja timove koji sudjeluju u kvizovima, s informacijama kao što su naziv tima i korisnik koji ga je registrirao.
- **Review**: Omogućuje korisnicima da ocijene i komentiraju kvizove, povezujući svaku recenziju s određenim kvizom i korisnikom.
- **Notification**: Služi za pohranu obavijesti za korisnike, uključujući poruke obavijesti i status o tome je li obavijest pročitana.

User Table

Atribut	Tip podatka	Opis varijable
id	int	Primarni ključ, jedinstveni identifikator za korisnika
name	varchar(100)	Ime korisnika
email	varchar(255)	Email adresa korisnika
oauth_provider	varchar(50)	Pružatelj usluge za autentifikaciju
oauth_token	varchar(255)	OAuth token za autentifikaciju
role_id	int	Strani ključ koji povezuje ulogu korisnika
created_at	datetime	Datum i vrijeme kreiranja korisnika

Role Table

Atribut	Tip podatka	Opis varijable
id	int	Primarni ključ, jedinstveni identifikator za ulogu
role_name	varchar(50)	Naziv uloge

FavoriteOrganizer Table

Atribut	Tip podatka	Opis varijable
id	int	Primarni ključ, jedinstveni identifikator zapisa favorita
user_id	int	Strani ključ koji povezuje korisnika
organizer_id	int	Strani ključ koji povezuje organizatora
created_at	datetime	Datum i vrijeme dodavanja favorita

Quiz Table

Atribut	Tip podatka	Opis varijable
id	int	Primarni ključ, jedinstveni identifikator za kviz
title	varchar(100)	Naslov kviza

Atribut	Tip podatka	Opis varijable
description	text	Detaljan opis kviza
category	varchar(50)	Kategorija kviza
difficulty	varchar(20)	Razina težine kviza
location	varchar(255)	Lokacija održavanja kviza
max_teams	int	Maksimalan broj timova
registration_deadline	datetime	Rok za prijavu na kviz
fee	decimal(10,2)	Naknada za sudjelovanje
duration	int	Trajanje kviza
organizer_id	int	Strani ključ koji povezuje organizatora
is_league	boolean	Označava je li kviz ligaškog tipa
prizes	text	Opis nagrada
created_at	datetime	Datum i vrijeme kreiranja kviza
start_time	datetime	Vrijeme početka kviza

Team Table

Atribut	Tip podatka	Opis varijable
id	int	Primarni ključ, jedinstveni identifikator za tim
name	varchar(100)	Naziv tima
quiz_id	int	Strani ključ koji povezuje kviz
registered_by	int	ID korisnika koji je registrirao tim
created_at	datetime	Datum i vrijeme kreiranja tima

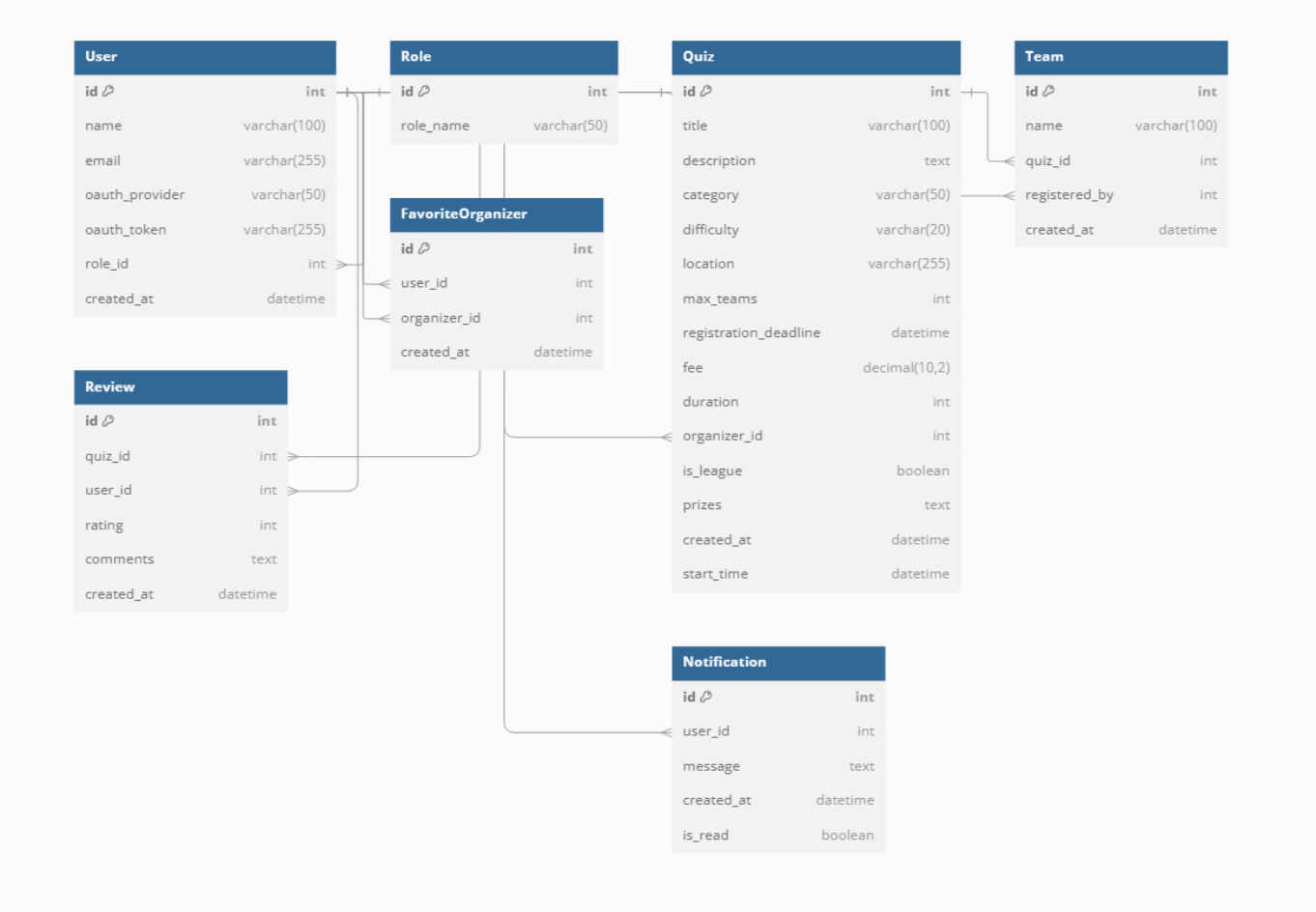
Review Table

Atribut	Tip podatka	Opis varijable
id	int	Primarni ključ, jedinstveni identifikator za recenziju
quiz_id	int	Strani ključ koji povezuje kviz
user_id	int	Strani ključ koji povezuje korisnika
rating	int	Ocjena kviza
comments	text	Komentari korisnika
created_at	datetime	Datum i vrijeme kreiranja recenzije

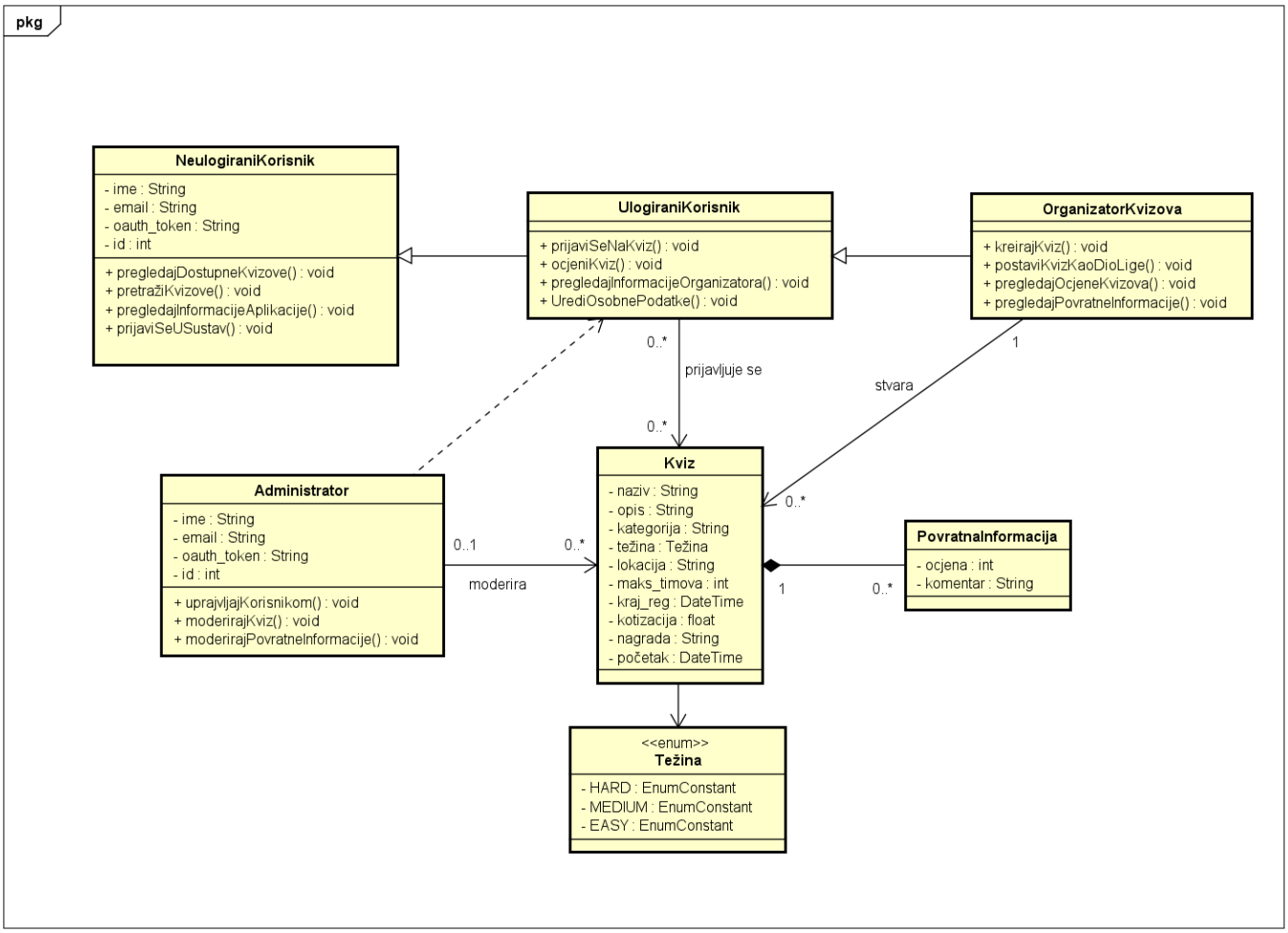
Notification Table

Atribut	Tip podatka	Opis varijable
id	int	Primarni ključ, jedinstveni identifikator za obavijest
user_id	int	Strani ključ koji povezuje korisnika
message	text	Poruka obavijesti
created_at	datetime	Datum i vrijeme kreiranja obavijesti
is_read	boolean	Status je li obavijest pročitana

Dijagram baze podataka



Dijagram razreda



Opis dijagrama razreda

Dijagram prikazuje strukturu sustava za organizaciju kvizova s različitim tipovima korisnika, funkcionalnostima i povratnim informacijama. Dijagram sadrži sljedeće klase:

Klase i njihovi atributi i metode:

1. NeulogiraniKorisnik

- **Atributi**:
  - ime: String
  - email: String
  - oauth_token: String
  - id: int
- **Metode**:
  - pregledajDostupneKvizove() : void – Pregled dostupnih kvizova na home page-u.
  - pretražiKvizove() : void – Pretraga kvizova po određenim kriterijima.
  - pregledajInformacijeAplikacije() : void – Pregled osnovnih informacija aplikacije.
  - prijaviSeUSustav() : void – Omogućava prijavu u sustav.

2. UlogiraniKorisnik

- **Metode**:
  - prijaviSeNaKviz() : void – Prijava na određeni kviz.
  - ocijeniKviz() : void – Ocjenjivanje kviza na kojem je korisnik sudjelovao.
  - pregledajInformacijeOrganizatora() : void – Pregled informacija organizatora kvizova.

    - urediOsobnePodatke() : void – Uređivanje vlastitih podataka tj. pregled, izmjena i brisanje korisničkog računa.

3. OrganizatorKvizova

- **Metode**:
    - kreirajKviz() : void – Kreiranje novog kviza.
    - postaviKvizKaoDioLige() : void – Postavljanje kviza kao dio lige.
    - pregledajOcjeneKvizova() : void – Pregled ocjene kvizova.
    - pregledajPovratneInformacije() : void – Pregled povratnih informacija o kvizovima.

4. Administrator

- **Atributi**:
    - ime: String
    - email: String
    - oauth_token: String
    - id: int
- **Metode**:
    - upravljajKorisnikom() : void – Upravljanje korisnicima u sistemu tj. izmjena i brisanje bilo kakvog neprimjerenog sadržaja.
    - moderirajKviz() : void – Moderiranje kvizova.
    - moderirajPovratneInformacije() : void – Pregled povratnih informacija.

5. Kviz

- **Atributi**:
    - naziv: String – Naziv kviza.
    - opis: String – Kratak opis kviza.
    - kategorija: String – Kategorija kviza.
    - težina: Težina – Težina kviza, predstavlja enumeraciju.
    - lokacija: String – Lokacija održavanja kviza.
    - maks_timova: int – Maksimalan broj timova.
    - kraj_reg: DateTime – Datum i vrijeme završetka registracije.
    - kotizacija: float – Iznos kotizacije.
    - nagrada: String – Vrsta ili opis nagrade.
    - početak: DateTime – Početno vrijeme kviza.
- **Metode**:
    - pobjednikKviza() : void – Određivanje pobjednika kviza.

6. PovratnaInformacija

- **Atributi**:
    - ocjena: int – Ocjena kviza od strane korisnika.
    - komentar: String – Komentar koji prati ocjenu.

7. Težina (Enumeracija)

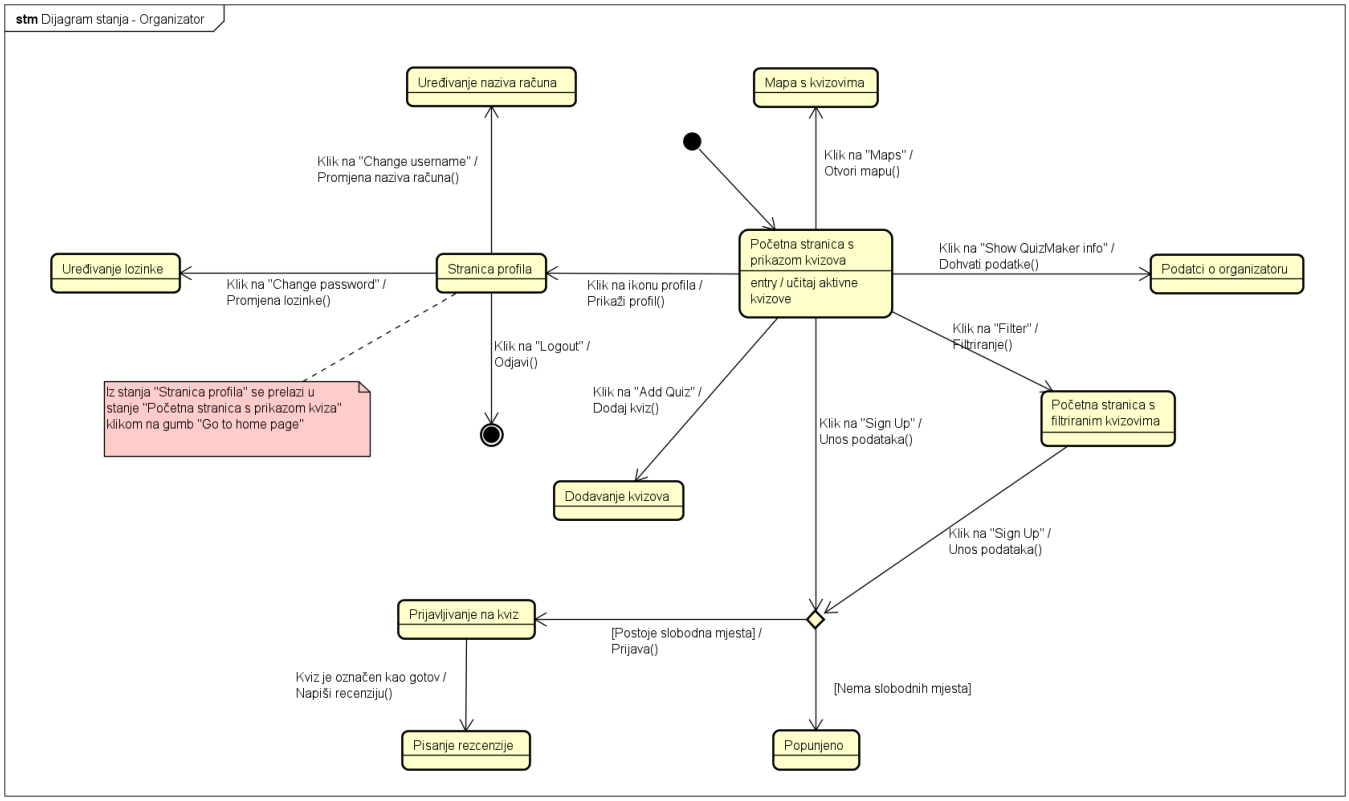
- Sadrži konstante koje označavaju težinu kviza.

Veze među klasama:

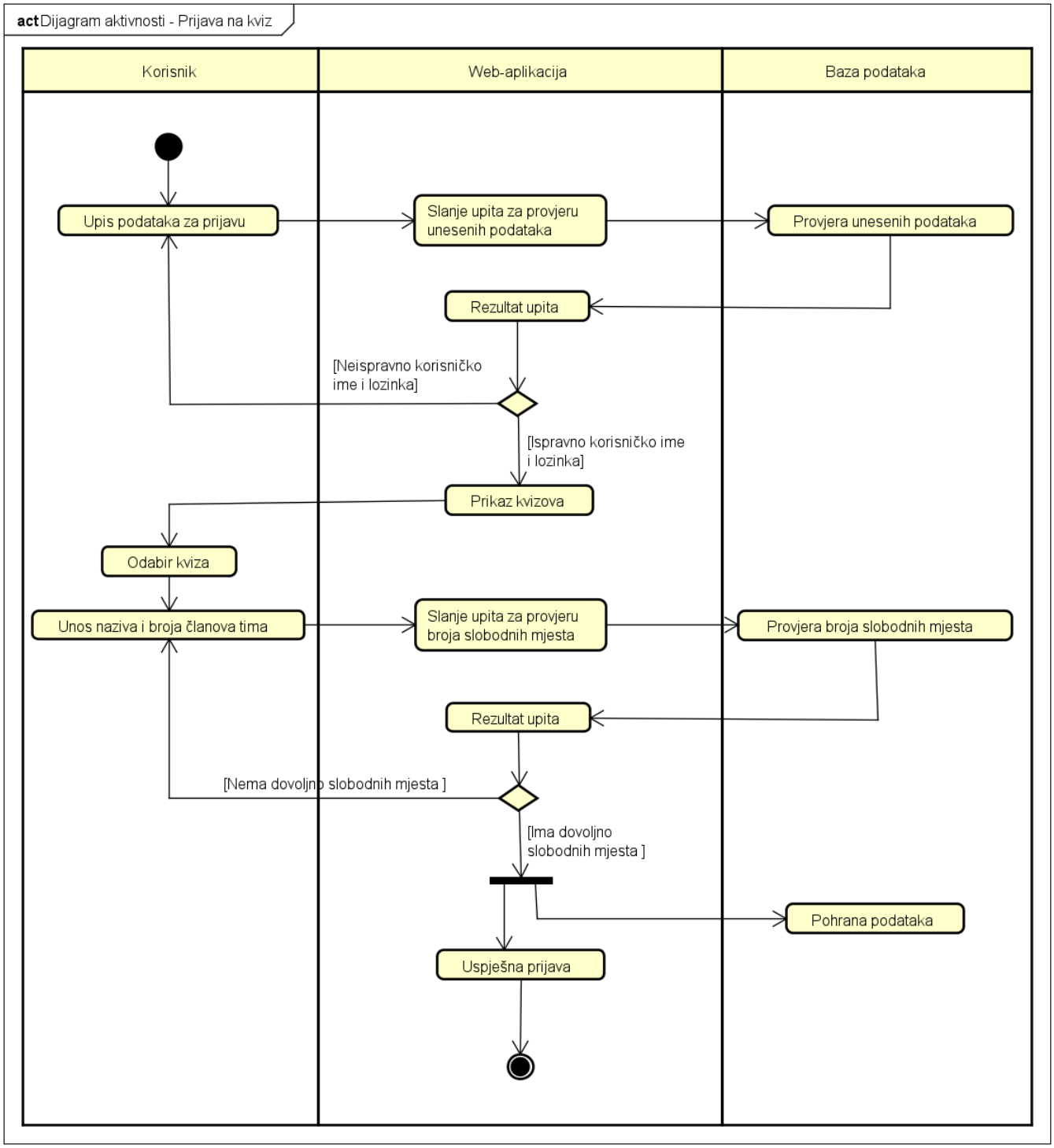
- Klasa UlogiraniKorisnik nasljeđuje klasu NeulogiraniKorisnik, što znači da sadrži sve njene atribute i metode.
- Klasa OrganizatorKvizova nasljeđuje klasu UlogiraniKorisnik, što znači da sadrži sve njene atribute i metode.
- Veza UlogiraniKorisnik – Kviz: korisnik se može prijaviti na više kvizova i na jedan kviz se može prijaviti više korisnika.
- Veza OrganizatorKvizova – Kviz: jedan organizator može kreirati više kvizova, ali jedan kviz ima samo jednog organizatora.
- Veza Administrator – Kviz: administrator može moderirati nula kvizova ili više, ali kviz može imati najviše jednog administratora.
- Veza Administrator – UlogiraniKorisnik: administrator može upravljati podatcima korisnika tj. može ih izmijeniti ili izbrisati ukoliko krše pravila.
- Veza Kviz – PovratnaInformacija: svaki kviz može imati više povratnih informacija, ali jedna povratna informacija se odnosi na jedan kviz.
- Klasa Težina predstavlja enumeraciju koja se koristi u klasi Kviz za opis težine kviza.

Dinamičko ponašanje aplikacije

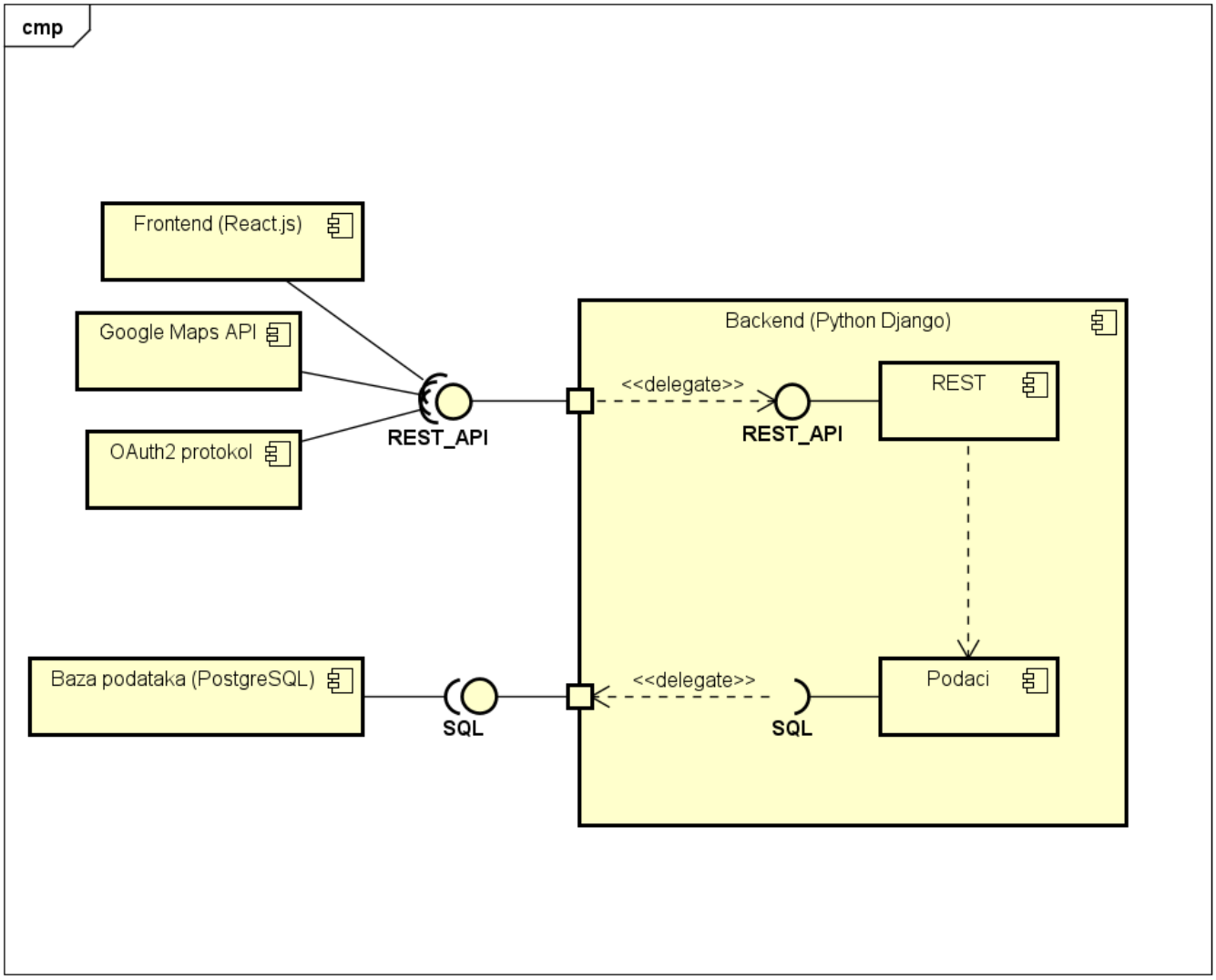
UML dijagram stanja



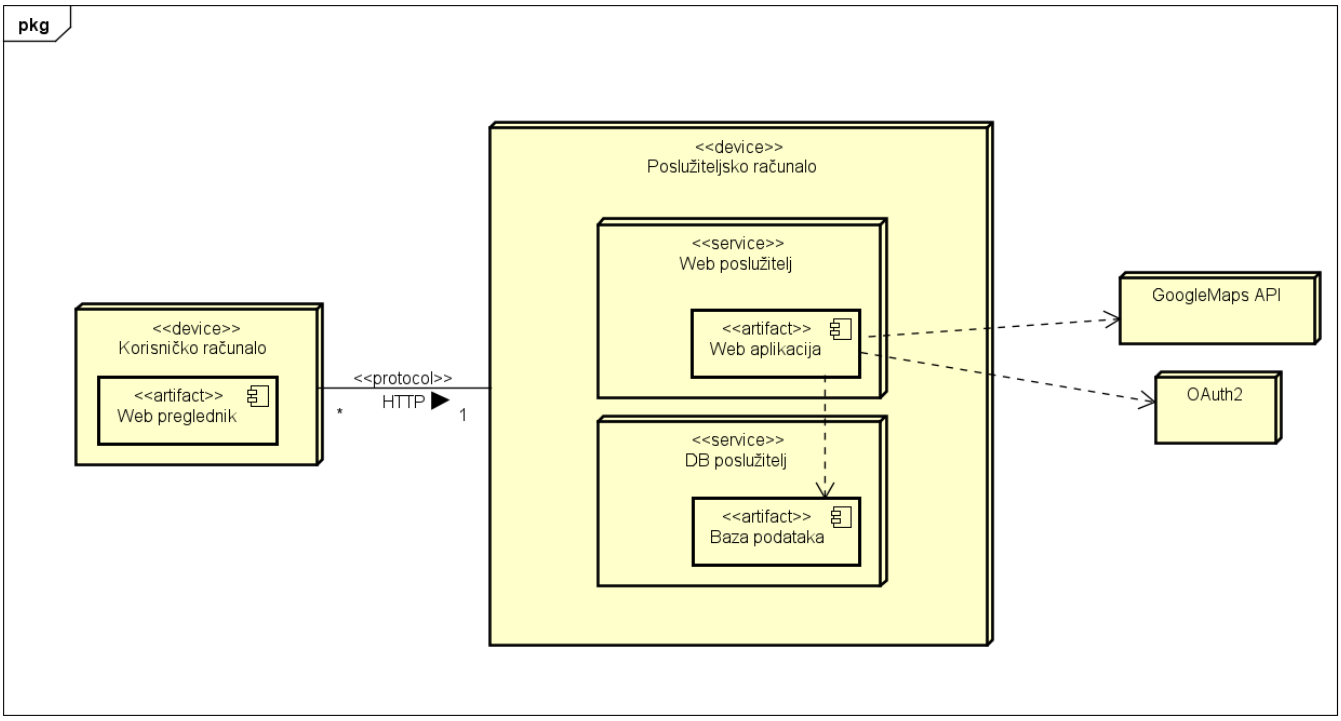
UML dijagram aktivnosti


act Dijagram aktivnosti - Prijava na kviz
Korisnik
Web-aplikacija
Baza podataka
Upis podataka za prijavu
Slanje upita za provjeru unesenih podataka
Provjera unesenih podataka
Rezultat upita
[Neispravno korisničko ime i lozinka]
[Ispravno korisničko ime i lozinka]
Prikaz kvizova
Odabir kviza
Unos naziva i broja članova tima
Slanje upita za provjeru broja slobodnih mjesta
Provjera broja slobodnih mjesta
Rezultat upita
[Nema dovoljno slobodnih mjesta]
[Ima dovoljno slobodnih mjesta]
Pohrana podataka
Uspješna prijava

Dijagram komponenata



Dijagram razmještaja

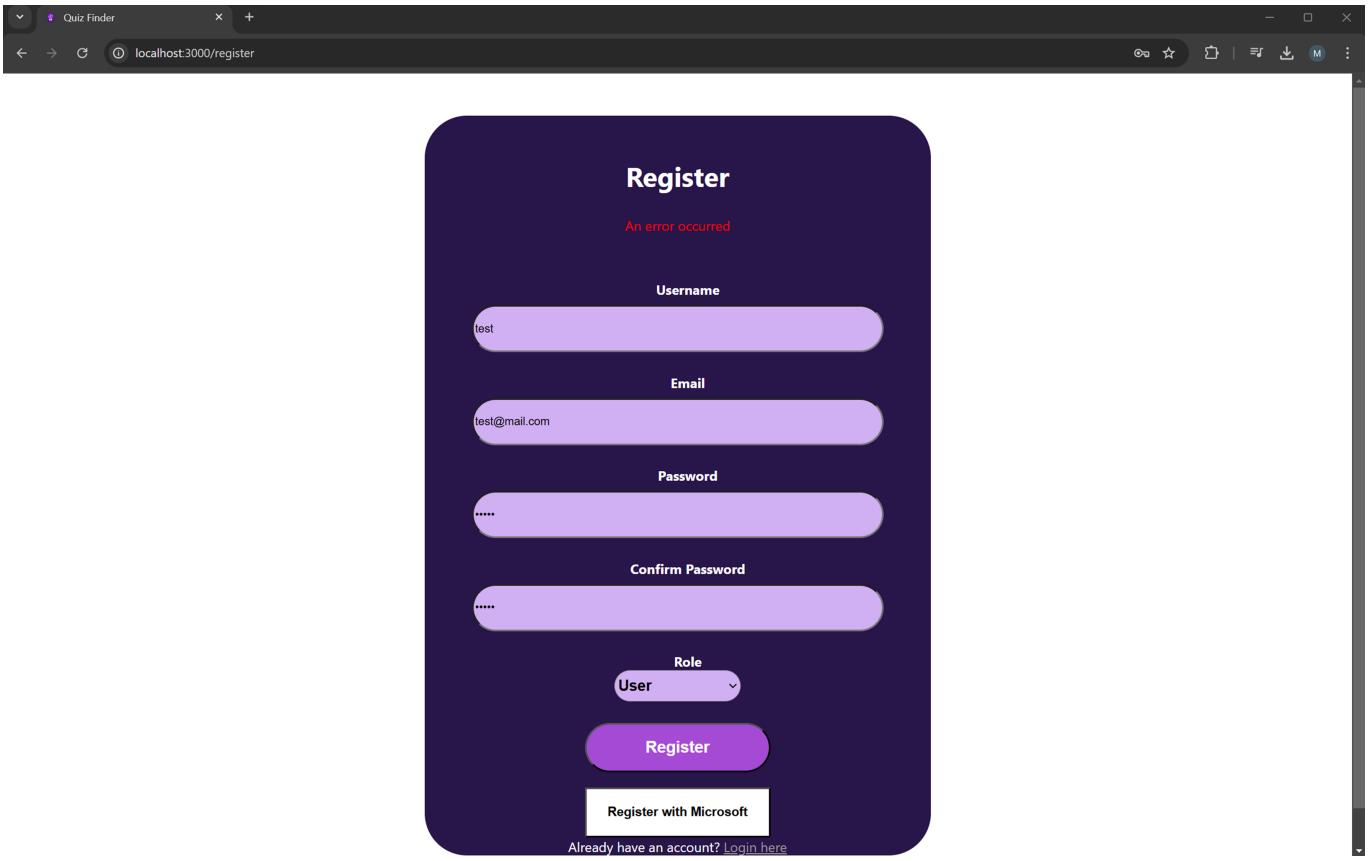

pkg
<<device>>
Poslužiteljsko računalo
<<service>>
Web poslužitelj
<<artifact>>
Web aplikacija
<<service>>
DB poslužitelj
<<artifact>>
Baza podataka
<<device>>
Korisničko računalo
<<artifact>>
Web preglednik
<<protocol>>
HTTP
*
1
GoogleMaps API
OAuth2

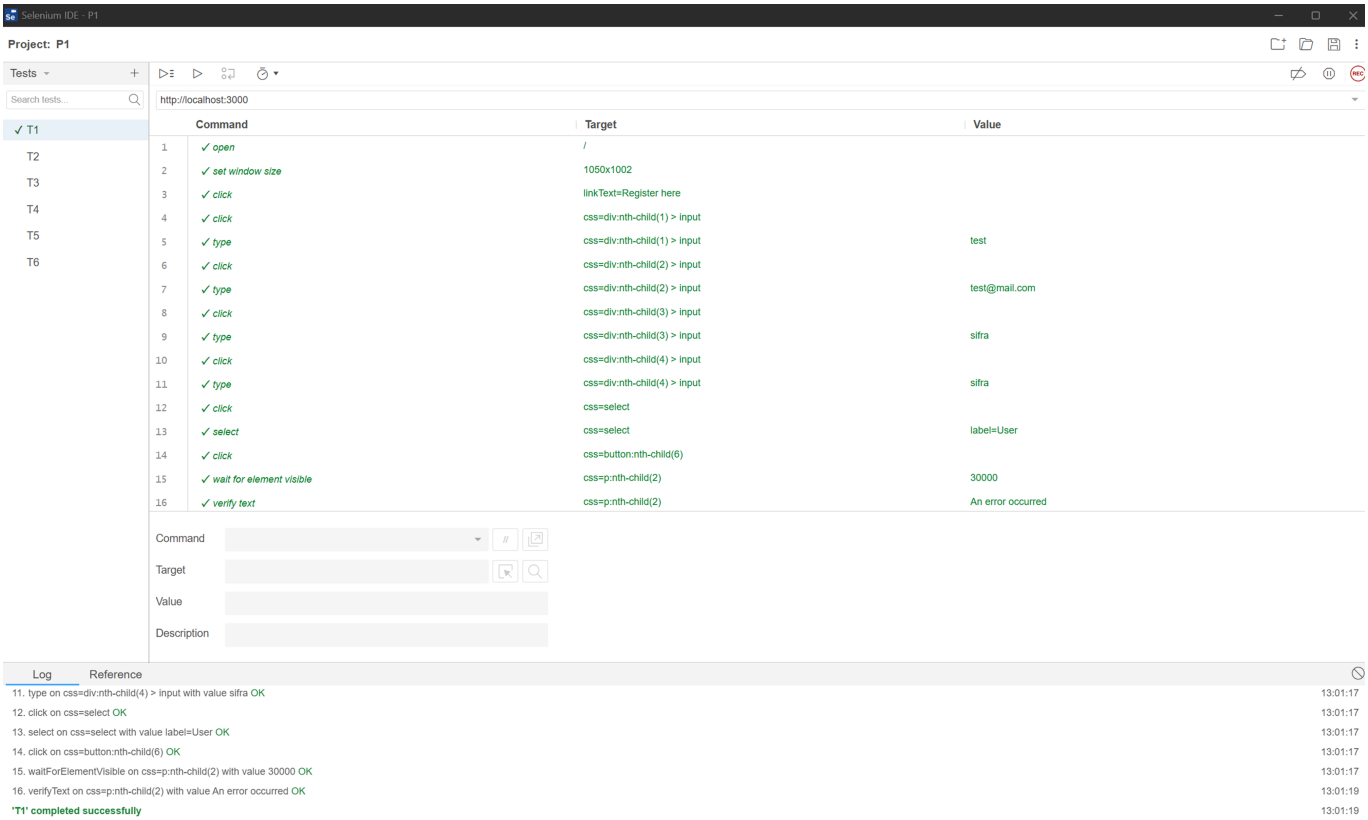
- u testu 7 se koristi uloga admin (username: admin, password: admin), u testu 10 se koristi uloga organizatora (username: test1, password: sifra1), a u ostalima se koristi uloga korisnika (username: test, password: sifra)

Ispitivanje komponenti

Ispitni slučaj 1

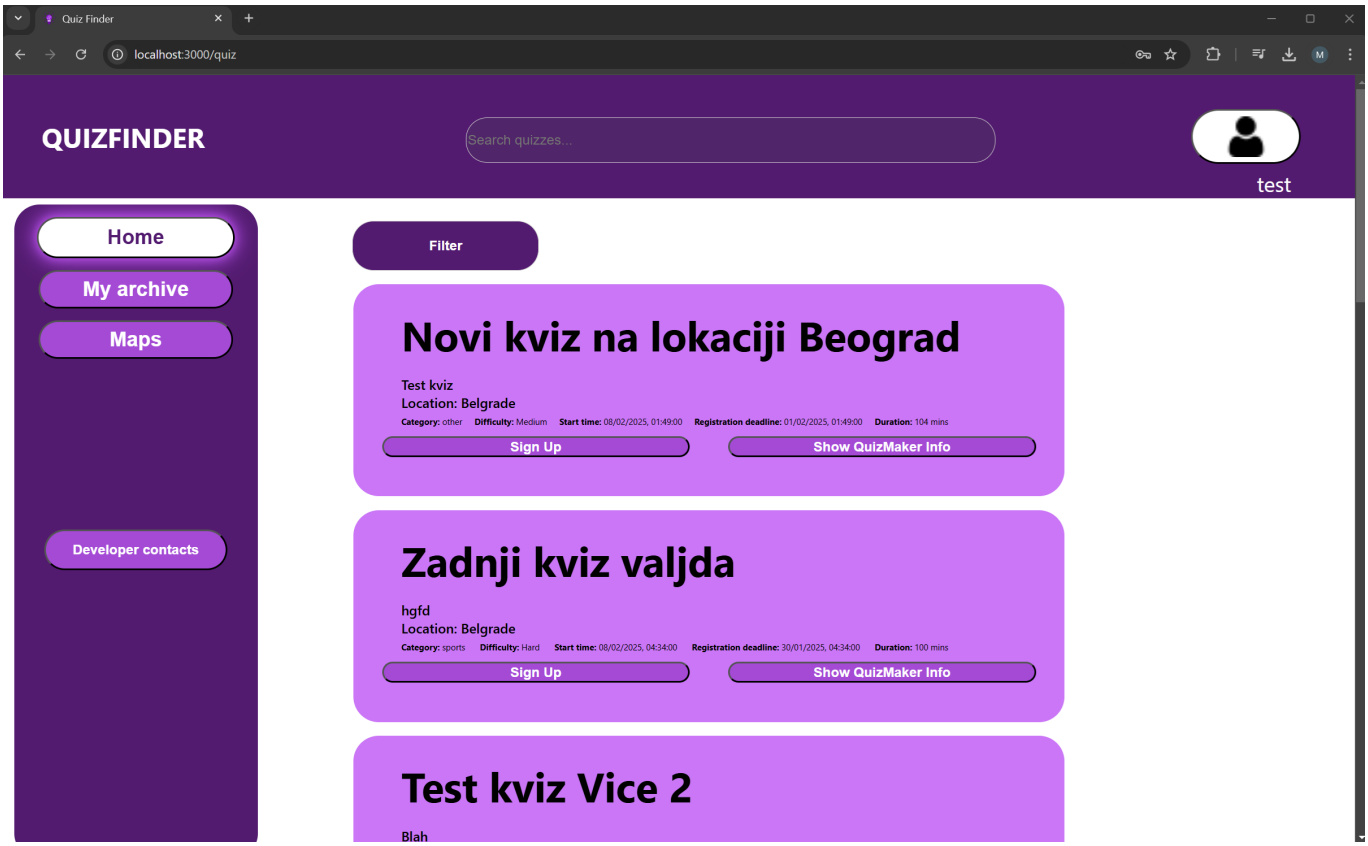
1. Test: Registracija već postojećeg korisnika
2. Ulazni podatci: Username = „test“, Email = „test@mail.com“, Password = „sifra“, Confirm Password = „sifra“, Role = User Očekivani rezultati: pojava poruke „An error occurred“ Dobiveni rezultat: prolaz ispitivanja
3. Postupak provođenja ispitivanja: otvoriti aplikaciju i stisnuti na gumb „Register here“, unijeti ulazne podatke i stisnuti na gumb „Register“

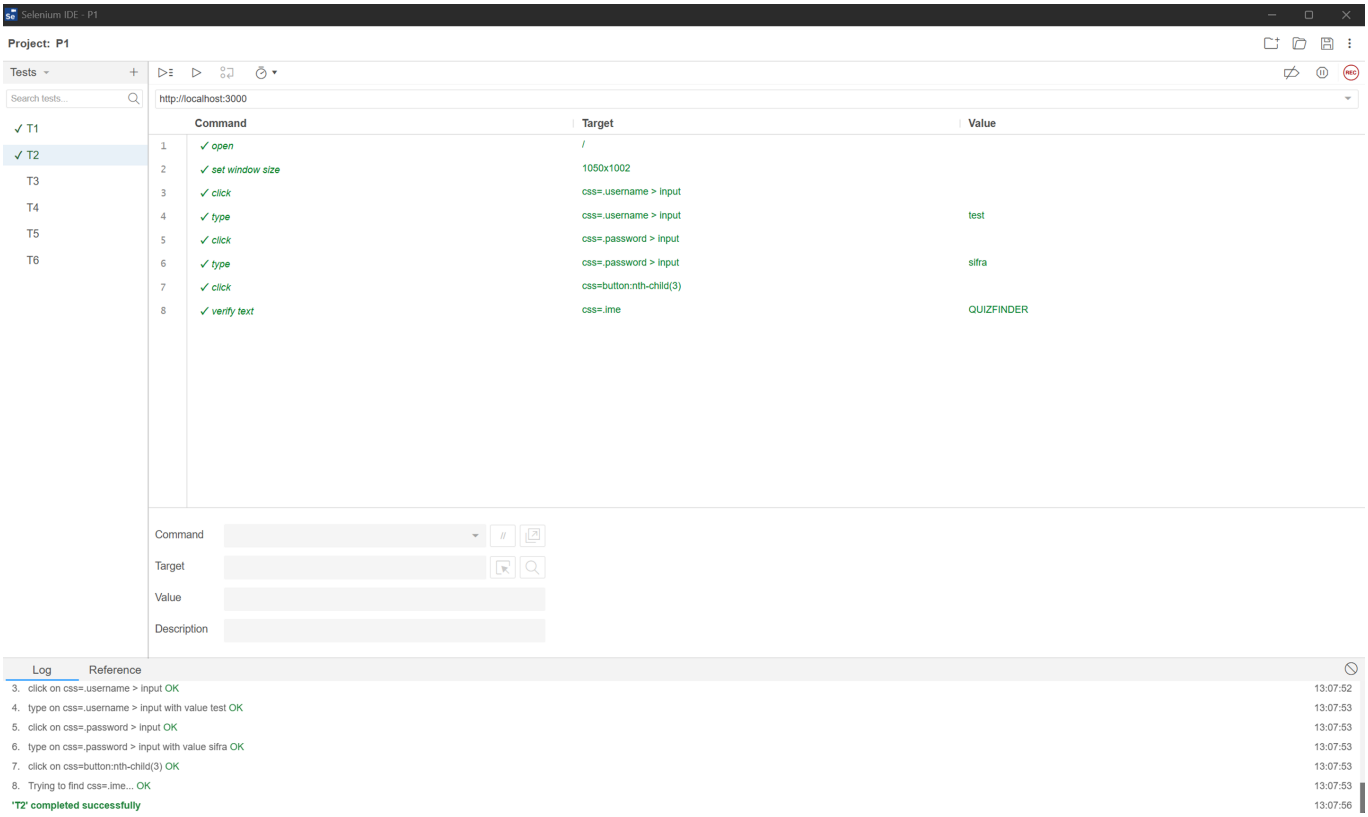




Ispitni slučaj 2

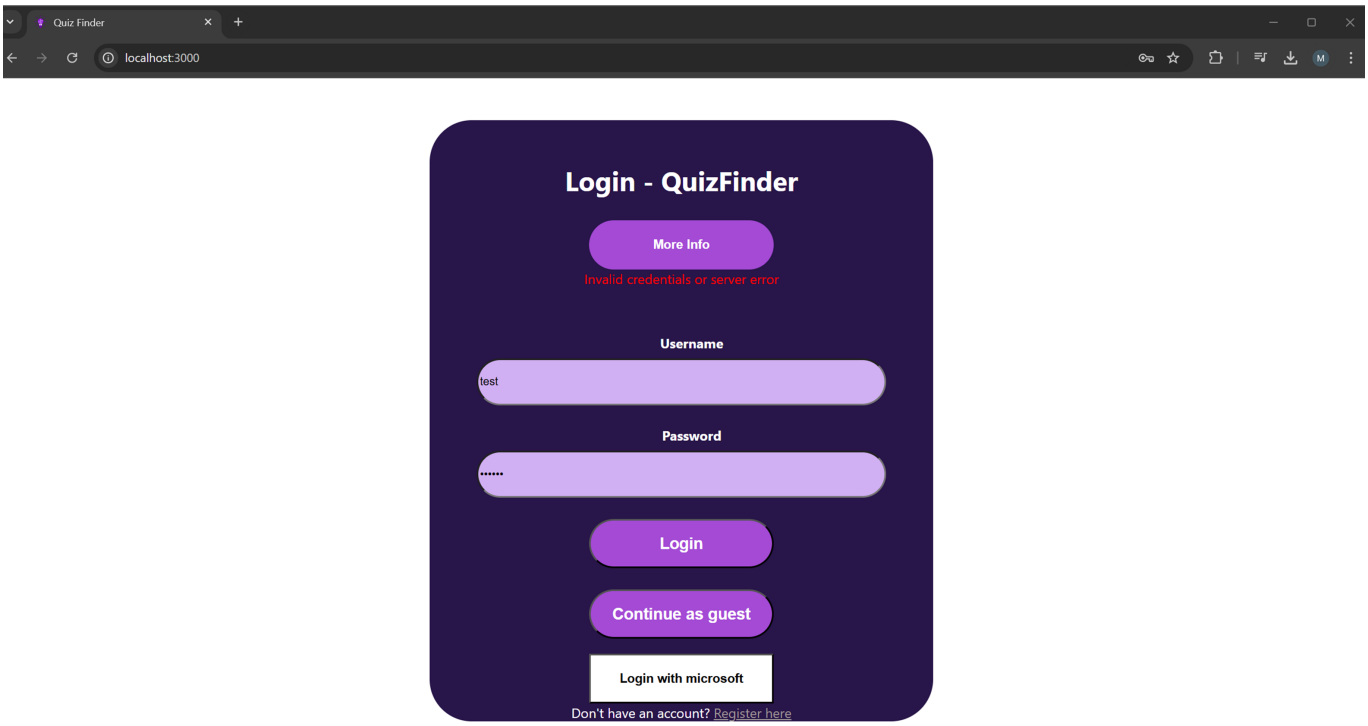
1. Test: Prijava u sustav s ispravnim podatcima
2. Ulazni podatci: Username = „test“, Password = „sifra“ Očekivani rezultati: korisnik je preusmjeren na početnu stranicu Dobiveni rezultat: prolaz ispitivanja
3. Postupak provođenja ispitivanja: otvoriti aplikaciju i unijeti ulazne podatke i stisnuti na gumb „Login“

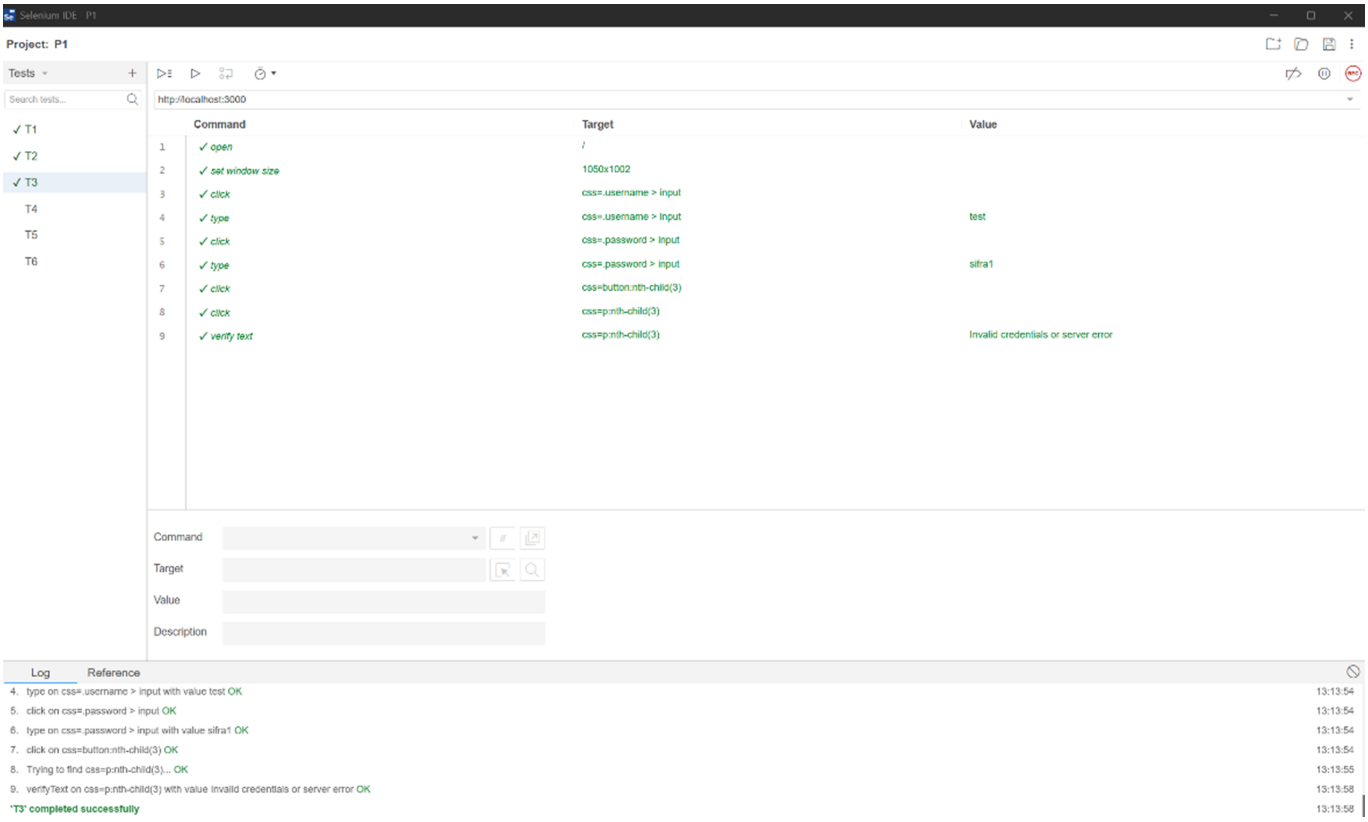




Ispitni slučaj 3

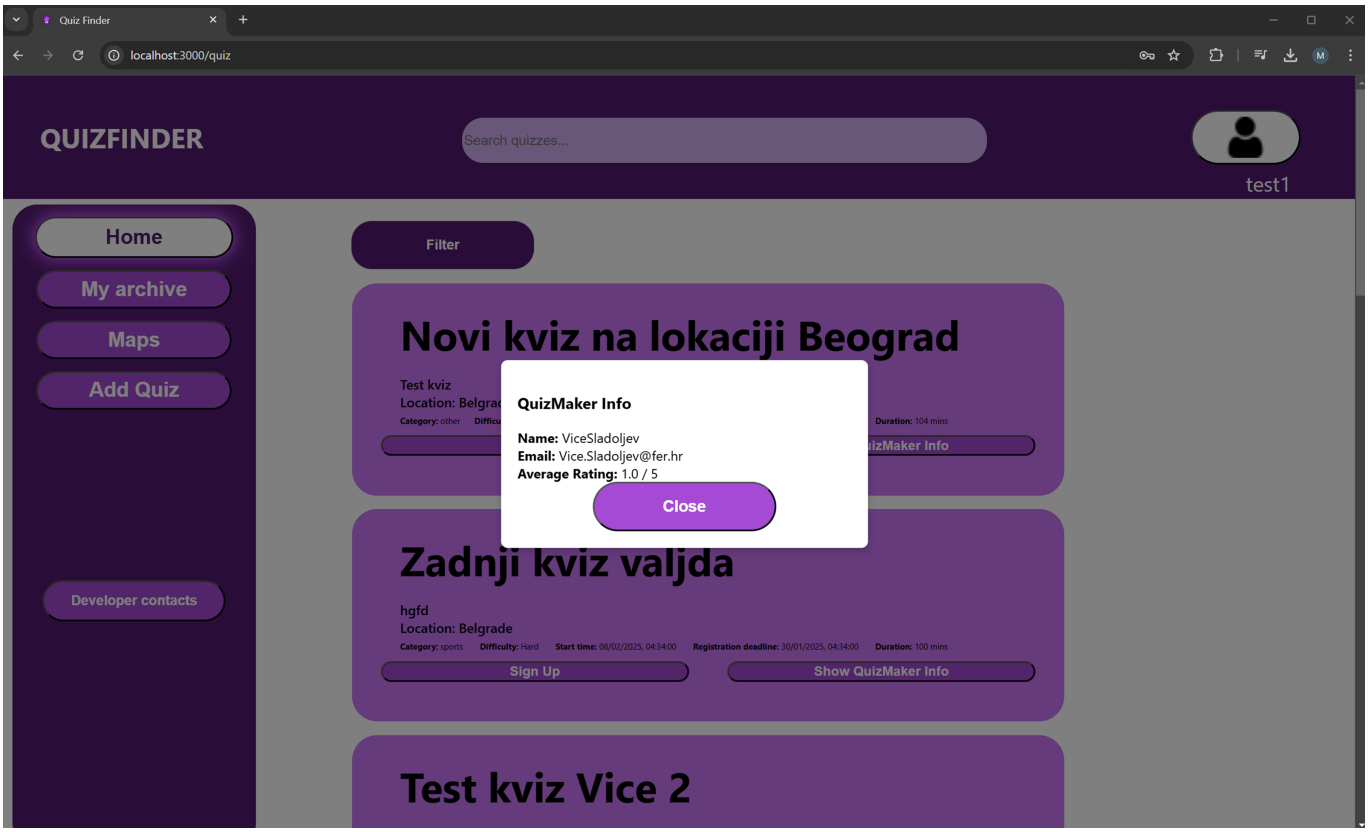
1. Test: Prijava u sustav s neispravnim podatcima
2. Ulazni podatci: Username = „test“, Password = „sifra1“ Očekivani rezultati: korisnik neće biti preusmjeren na početnu stranicu, nego će se pojaviti poruka „Invalid credentials or server error“ Dobiveni rezultat: prolaz ispitivanja
3. Postupak provođenja ispitivanja: otvoriti aplikaciju i unijeti ulazne podatke i stisnuti na gumb „Login“

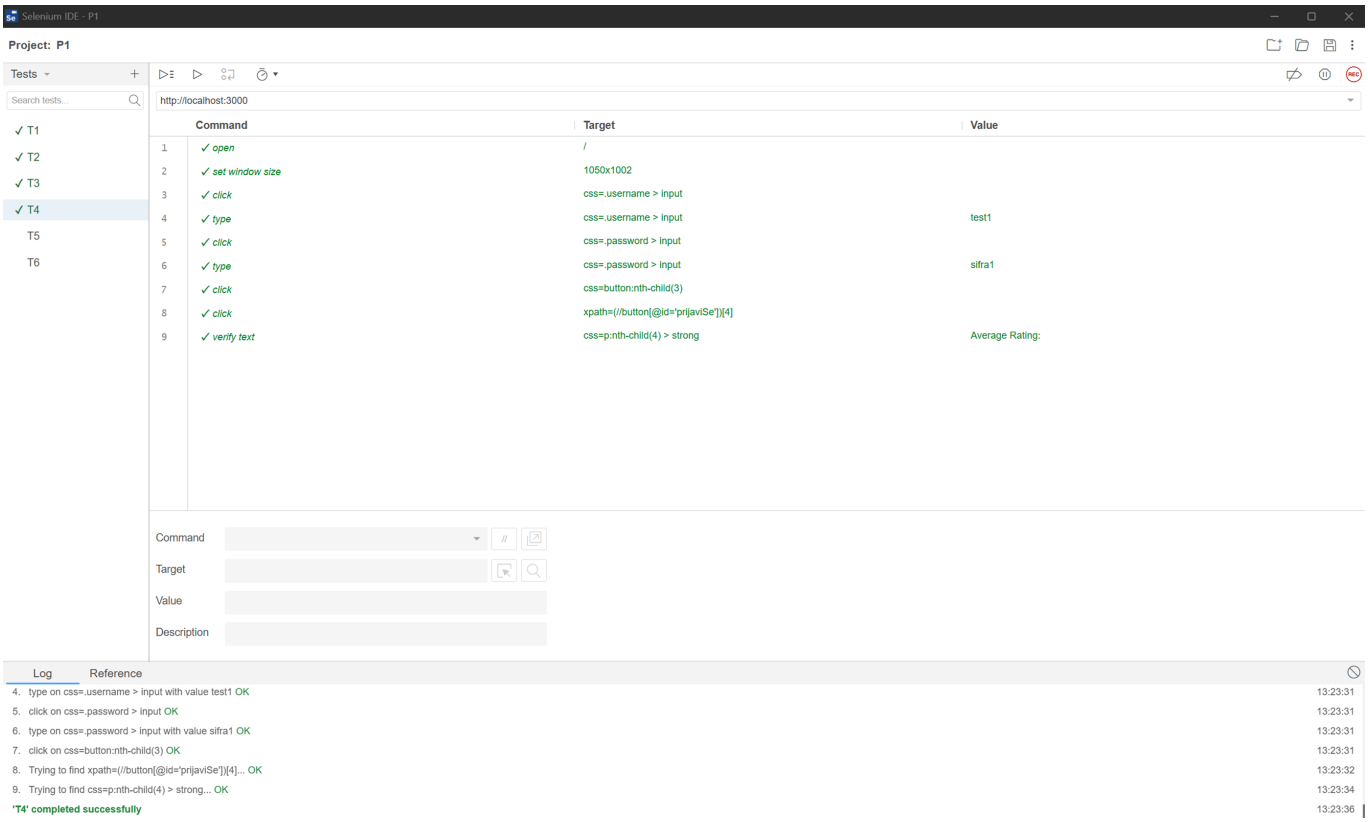




Ispitni slučaj 4

1. Test: Prikaz podataka o organizatoru kviza
2. Ulazni podatci: Username = „test“, Password = „sifra“ Očekivani rezultati: pregled informacija o organizatoru kviza Dobiveni rezultat: prolaz ispitivanja
3. Postupak provođenja ispitivanja: otvoriti aplikaciju, unijeti ulazne podatke i stisnuti na gumb „Login“, nakon uspješne prijave odabrati kviz i stisnuti gumb „Show QuizMaker Info“ na tom kvizu

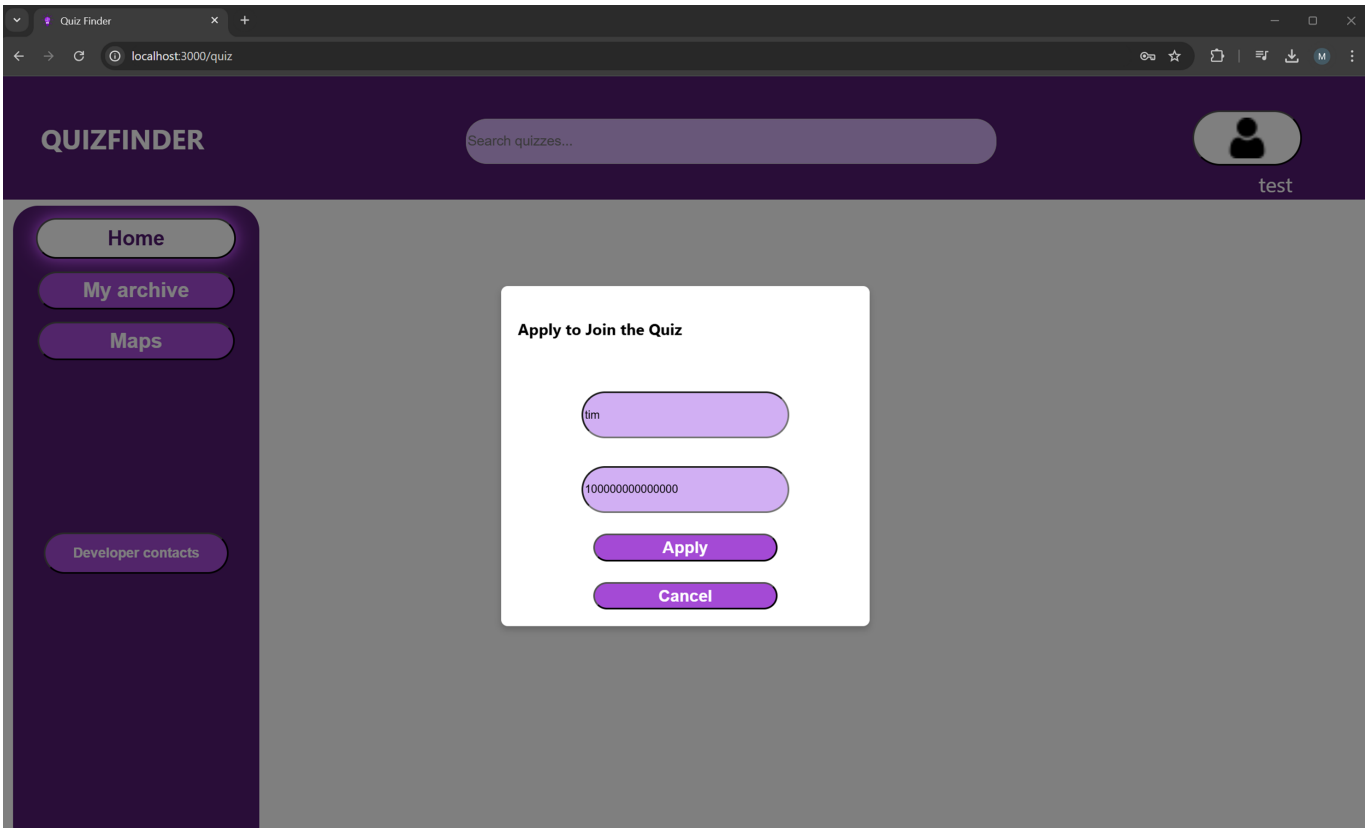


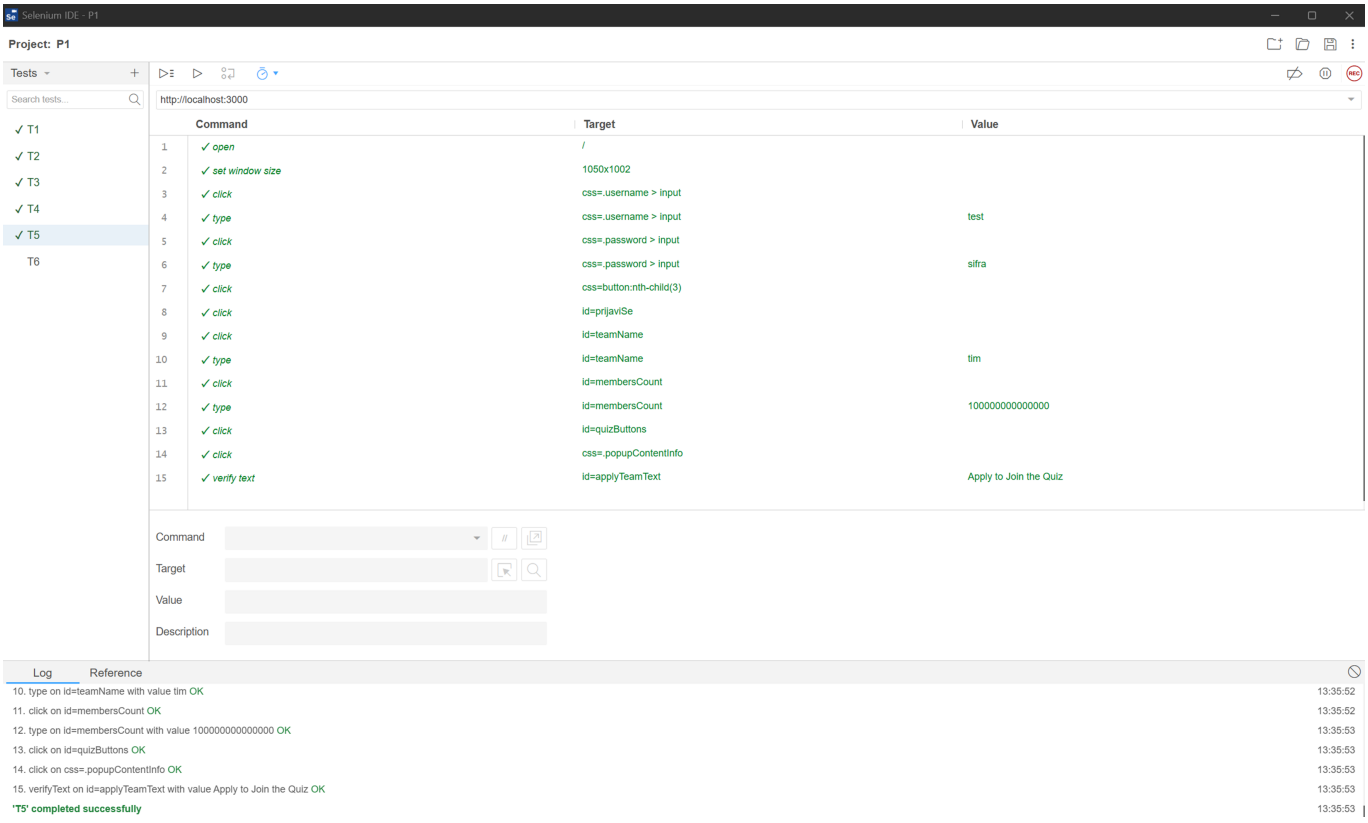


Ispitni slučaj 5

1. Test: Pokušaj prijave na kviz s brojem članova tima koji je veći nego broj ljudi koji može sudjelovati na kvizu
2. Ulazni podatci: Username = „test“, Password = „sifra“, Team Name = „tim“, Members Count = „100000000000000“ Očekivani rezultati: neće se ništa dogoditi tj. prijava na kviz će biti neuspješna (prozor za prijavu na kviz će i dalje ostati otvoren) Dobiveni rezultat: prolaz ispitivanja

3. Postupak provođenja ispitivanja: otvoriti aplikaciju, unijeti korisničko ime i lozinku i stisnuti na gumb „Login“, nakon uspješne prijave odabrati kviz i stisnuti gumb „Sign Up“ na tom kvizu, u otvorenom prozoru unijeti ime tima i broj članova i kliknuti na „Apply“

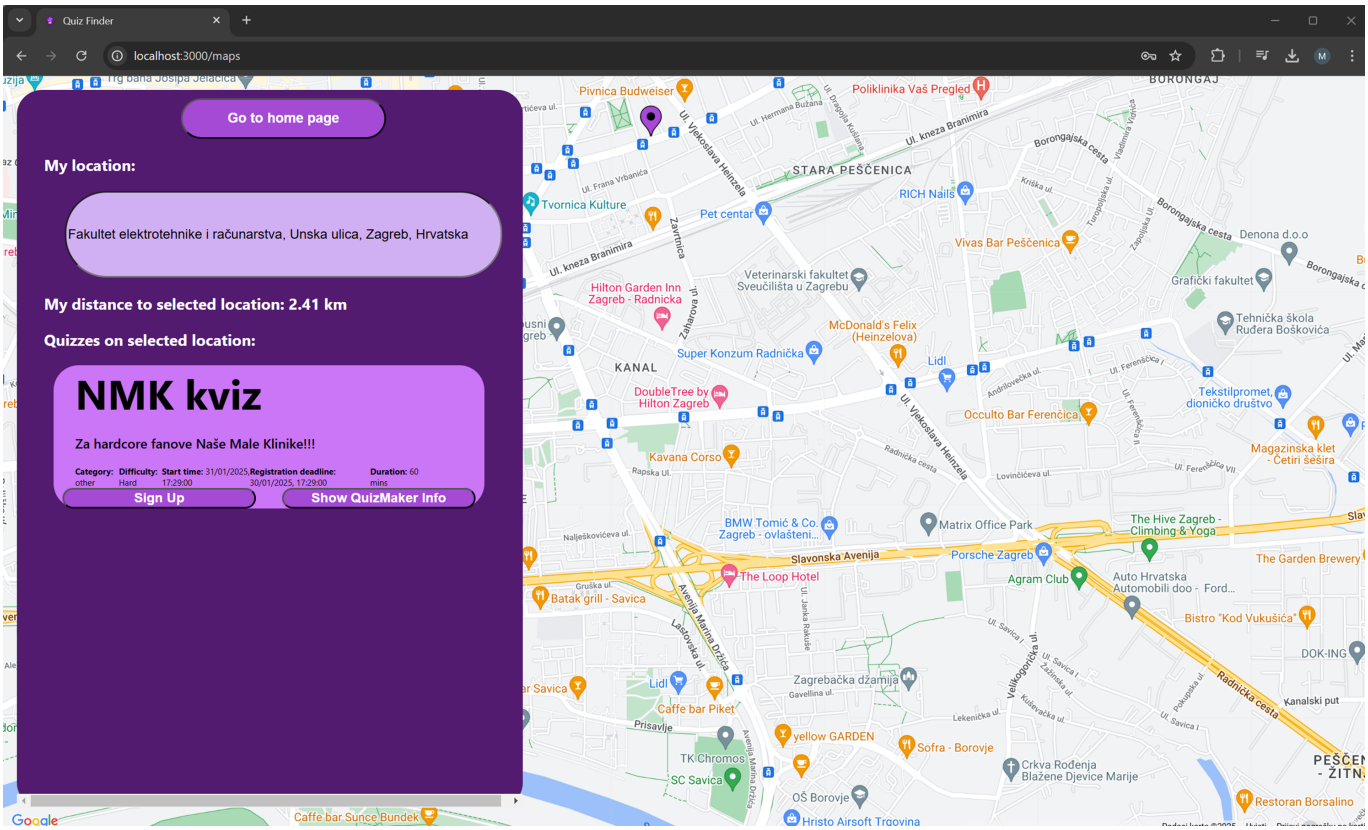


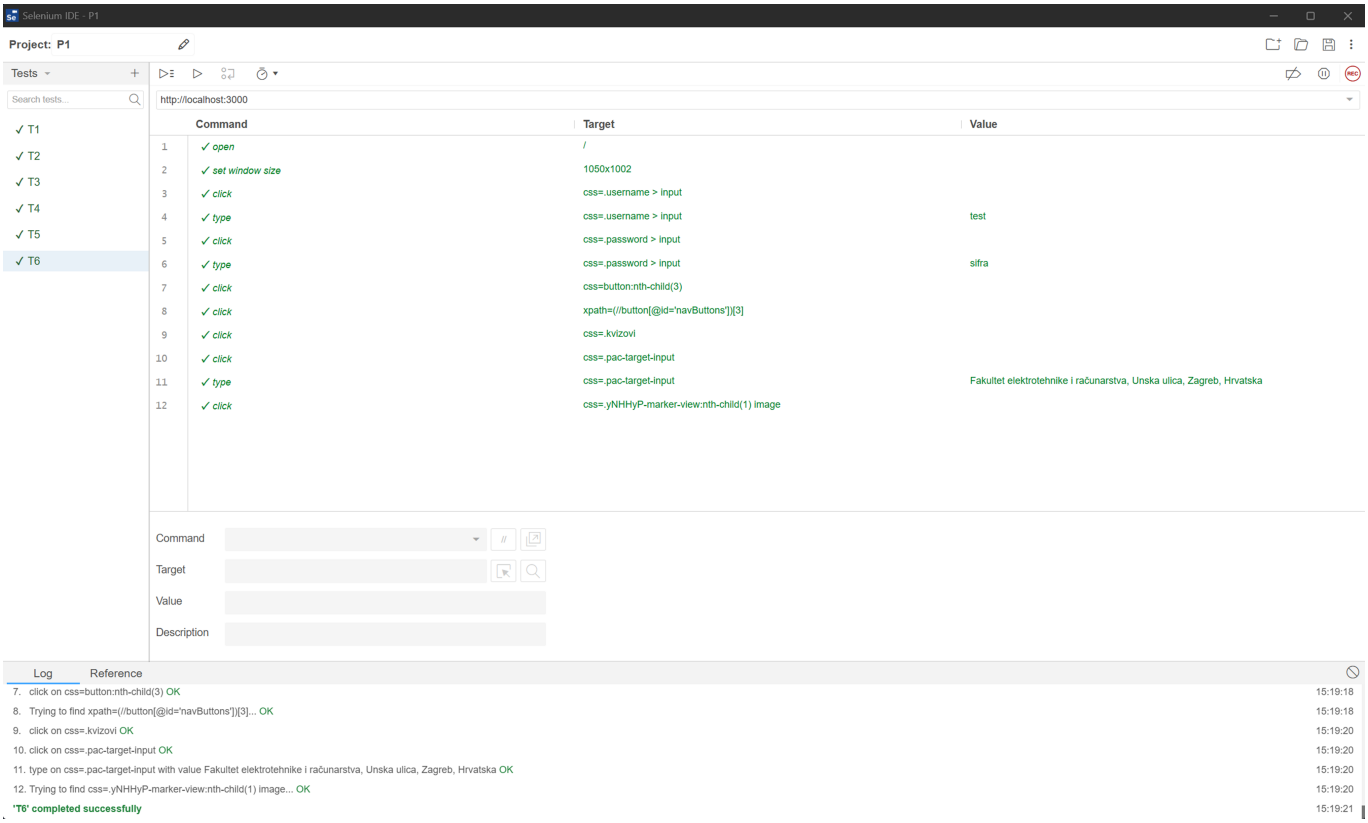


Ispitni slučaj 6

1. Test: Pregled kvizova na karti

2. Ulazni podatci: Username = „test“, Password = „sifra“, Unesite lokaciju = „Fakultet elektrotehnike i računarstva, Unska ulica, Zagreb, Hrvatska“ Očekivani rezultati: prikaz aktivnih kvizova na odabranoj lokaciji i udaljenosti unesene lokacije od lokacije kviza Dobiveni rezultat: prolaz ispitivanja
3. Postupak provođenja ispitivanja: otvoriti aplikaciju, unijeti korisničko ime i lozinku i stisnuti na gumb „Login“, nakon uspješne prijave kliknuti na gumb „Maps“, na mapi unijeti svoju lokaciju i na karti stisnuti odabranu lokaciju kviza

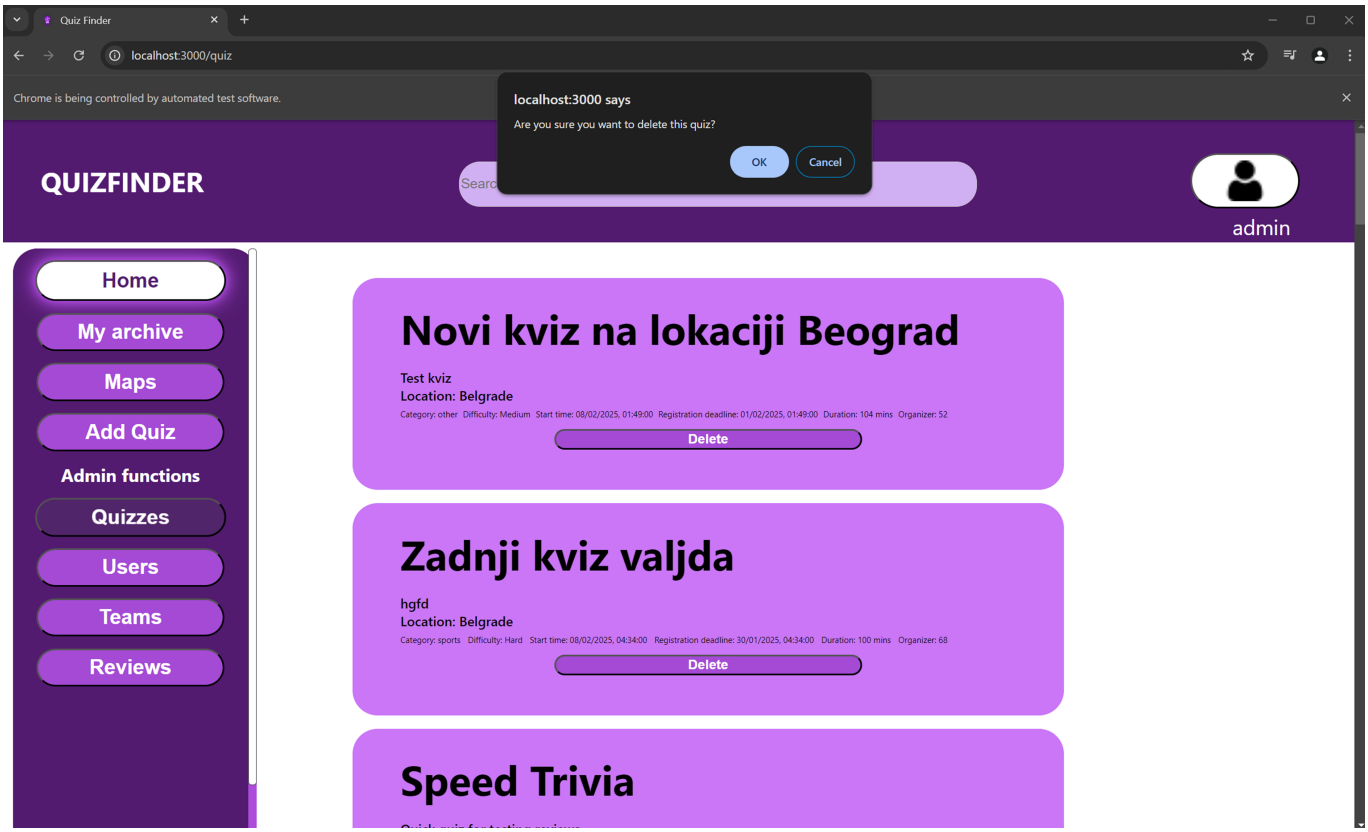


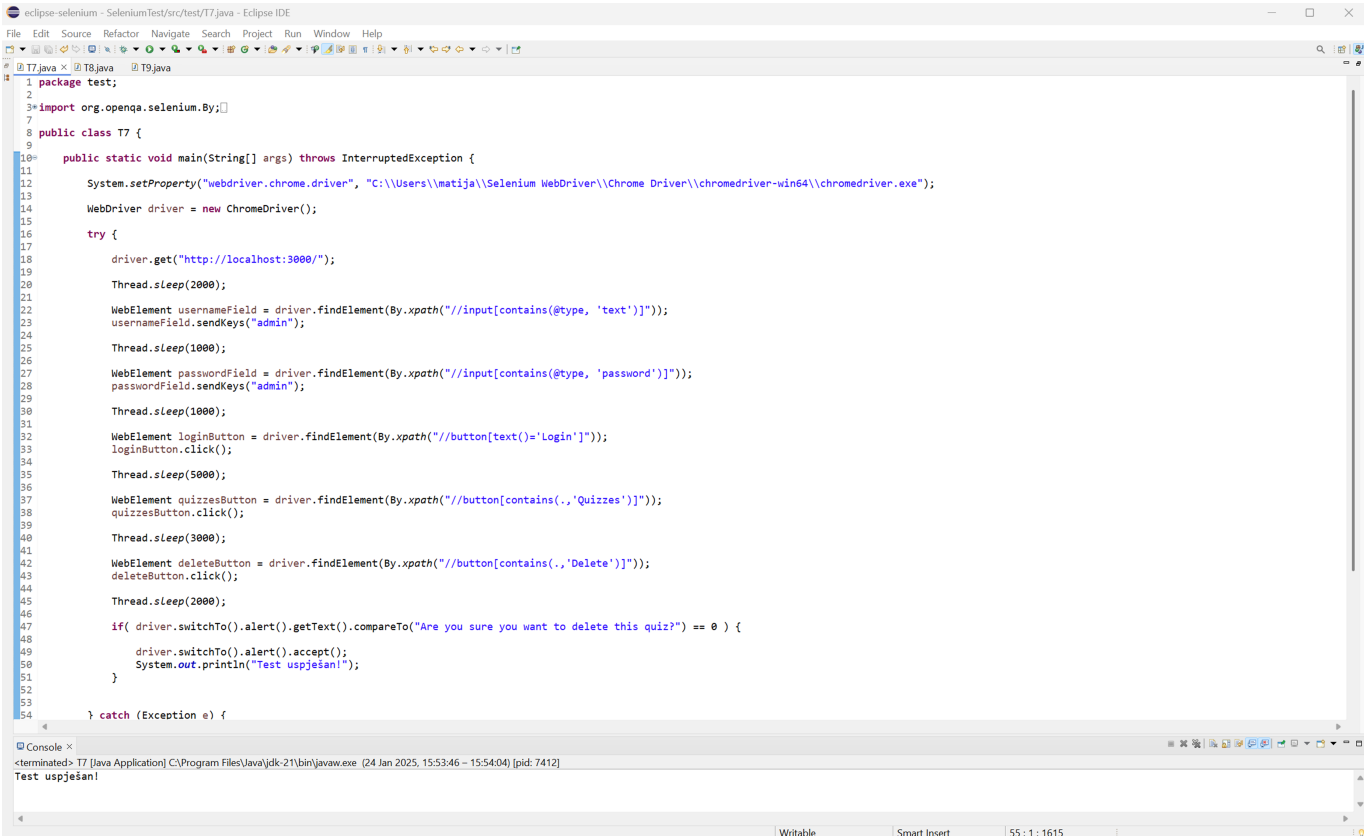


Ispitivanje sustava

Ispitni slučaj 7 – brisanje kvizova(funkcionalnost admina)

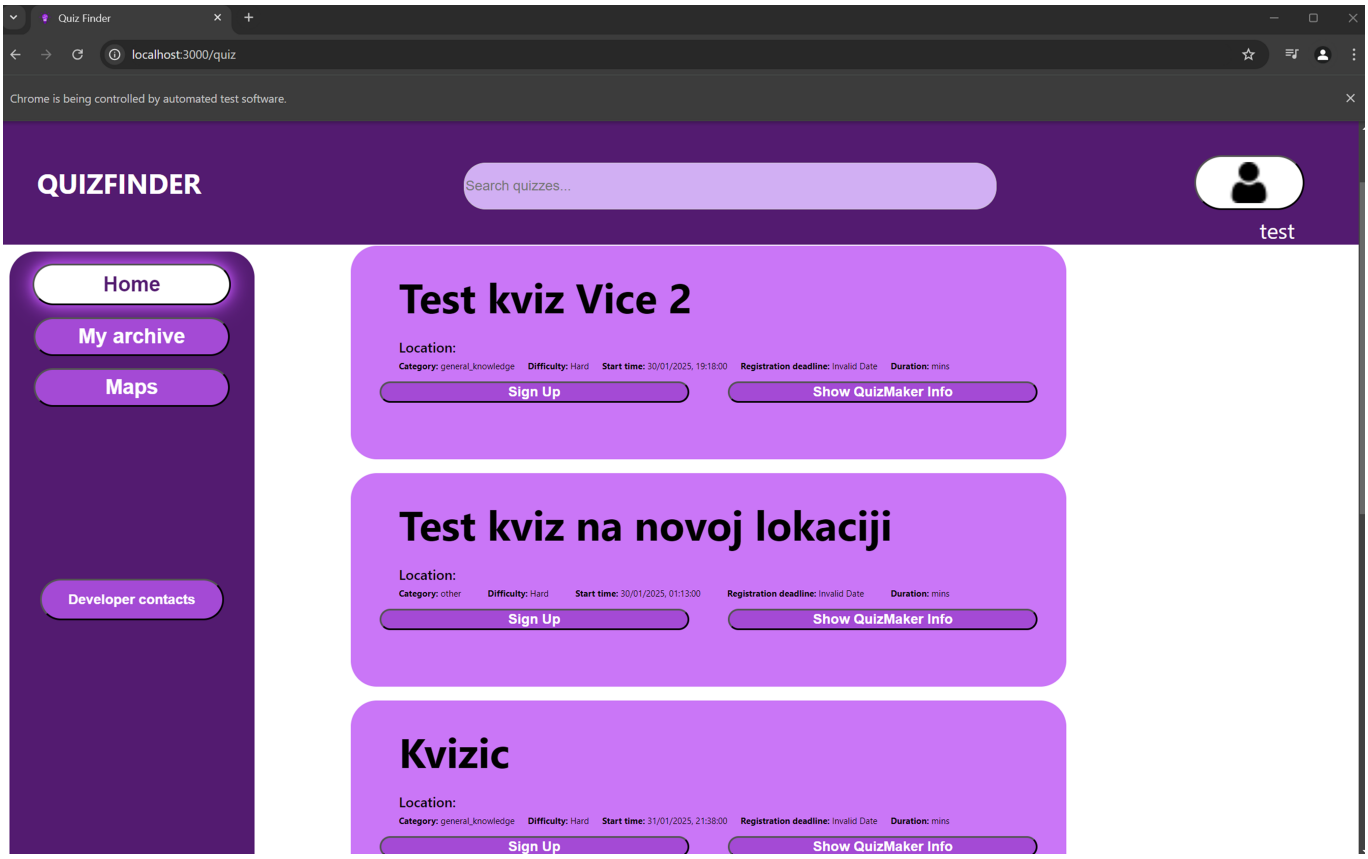
1. Ulazi: Username = „admin“, Password = „admin“
2. Koraci ispitivanja: otvoriti aplikaciju, unijeti korisničko ime i lozinku i stisnuti na gumb „Login“, nakon uspješne prijave kliknuti na gumb „Quizzes“ nakon čega će se otvoriti popis kvizova, na odabranom kvizu kliknuti gumb „Delete“, kada se pojavi upozorenje "Are you sure you want to delete this quiz?" kliknuti „OK“
3. Očekivani izlaz: uspješno obrisan kviz
4. Dobiveni izlaz: uspješno obrisan kviz

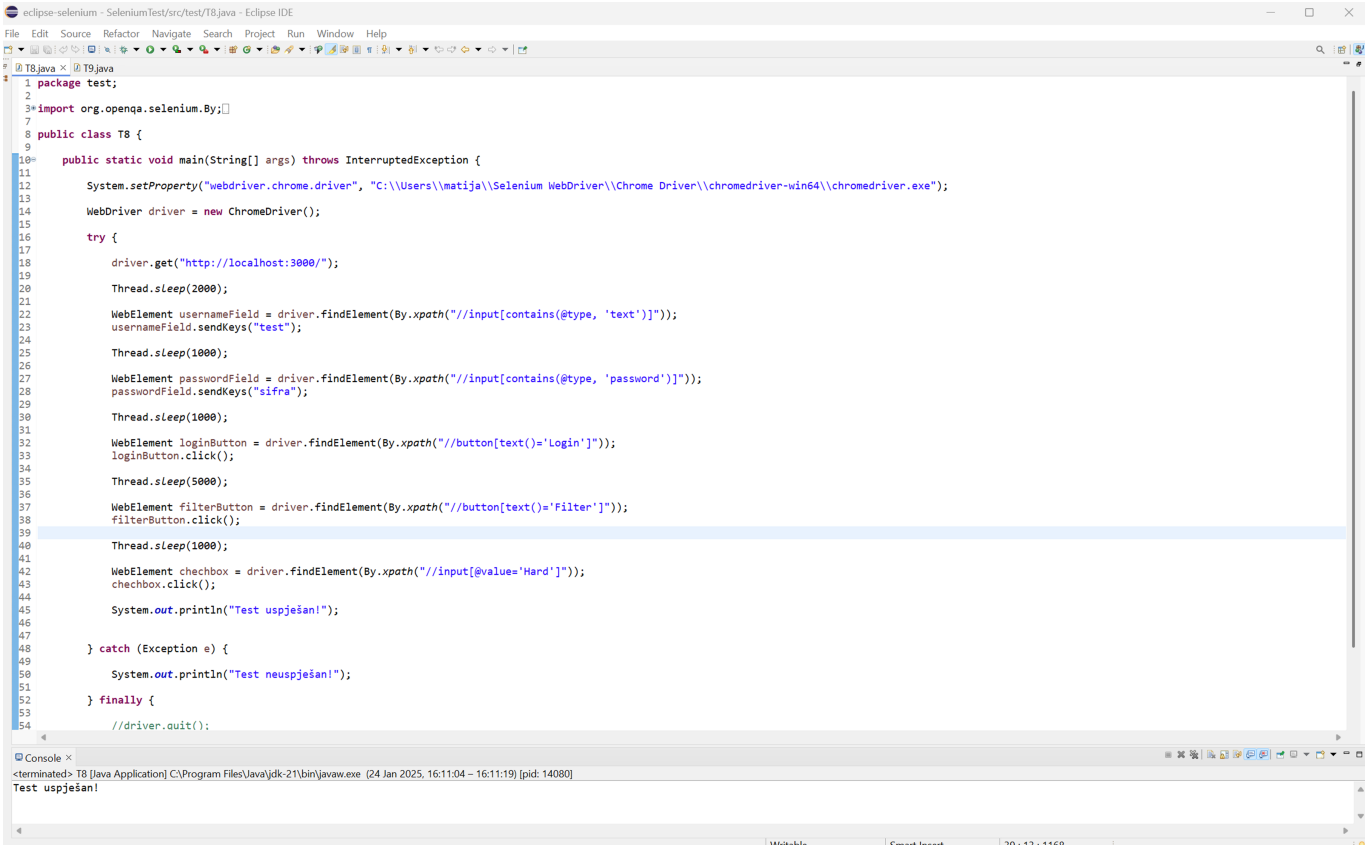




Ispitni slučaj 8 – filtriranje kvizova

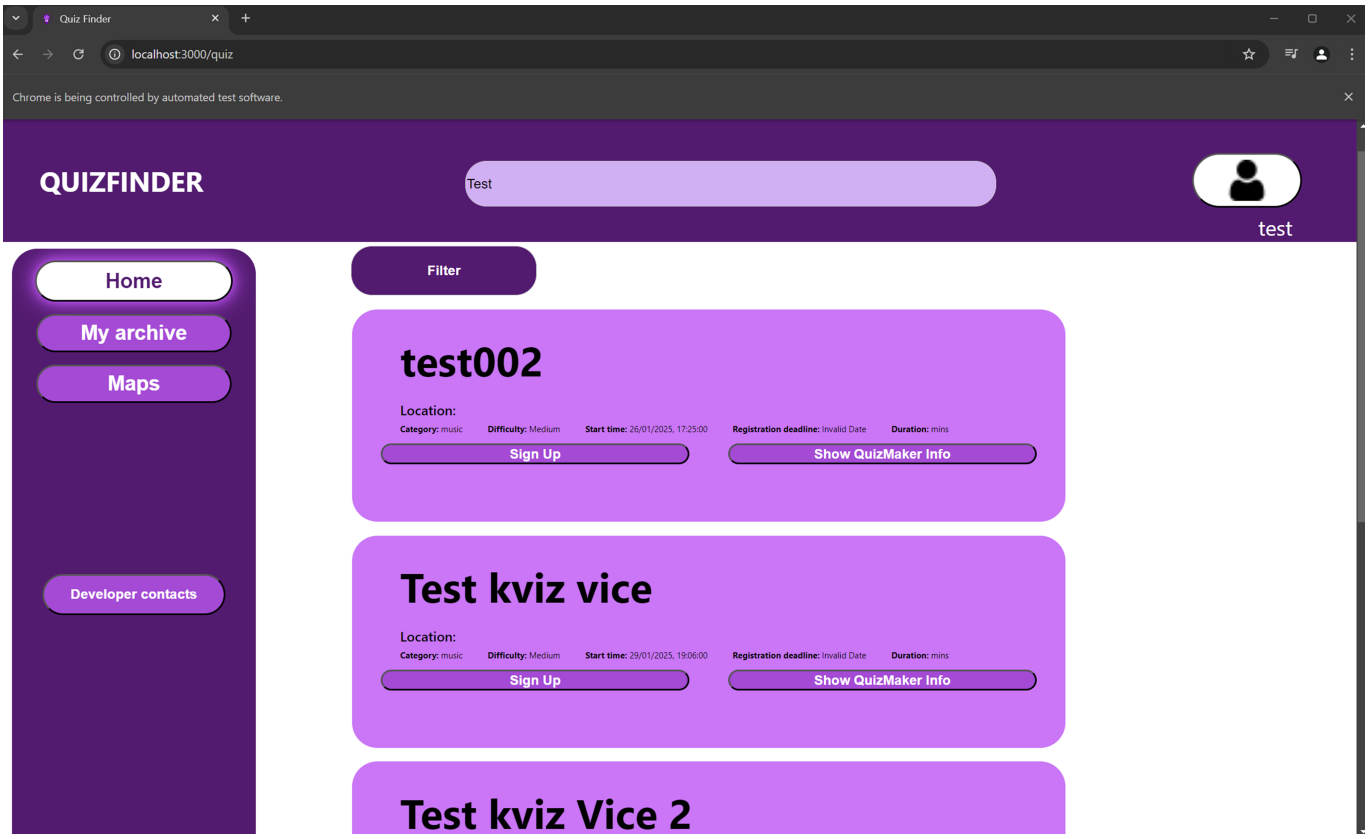
1. Ulazi: Username = „test“, Password = „sifra“
2. Koraci ispitivanja: otvoriti aplikaciju, unijeti korisničko ime i lozinku i stisnuti na gumb „Login“, nakon uspješne prijave kliknuti „Filter“ i označiti kučiću za težinu kviza „Hard“
3. Očekivani izlaz: uspješno filtrirani kvizovi
4. Dobiveni izlaz: uspješno filtrirani kvizovi

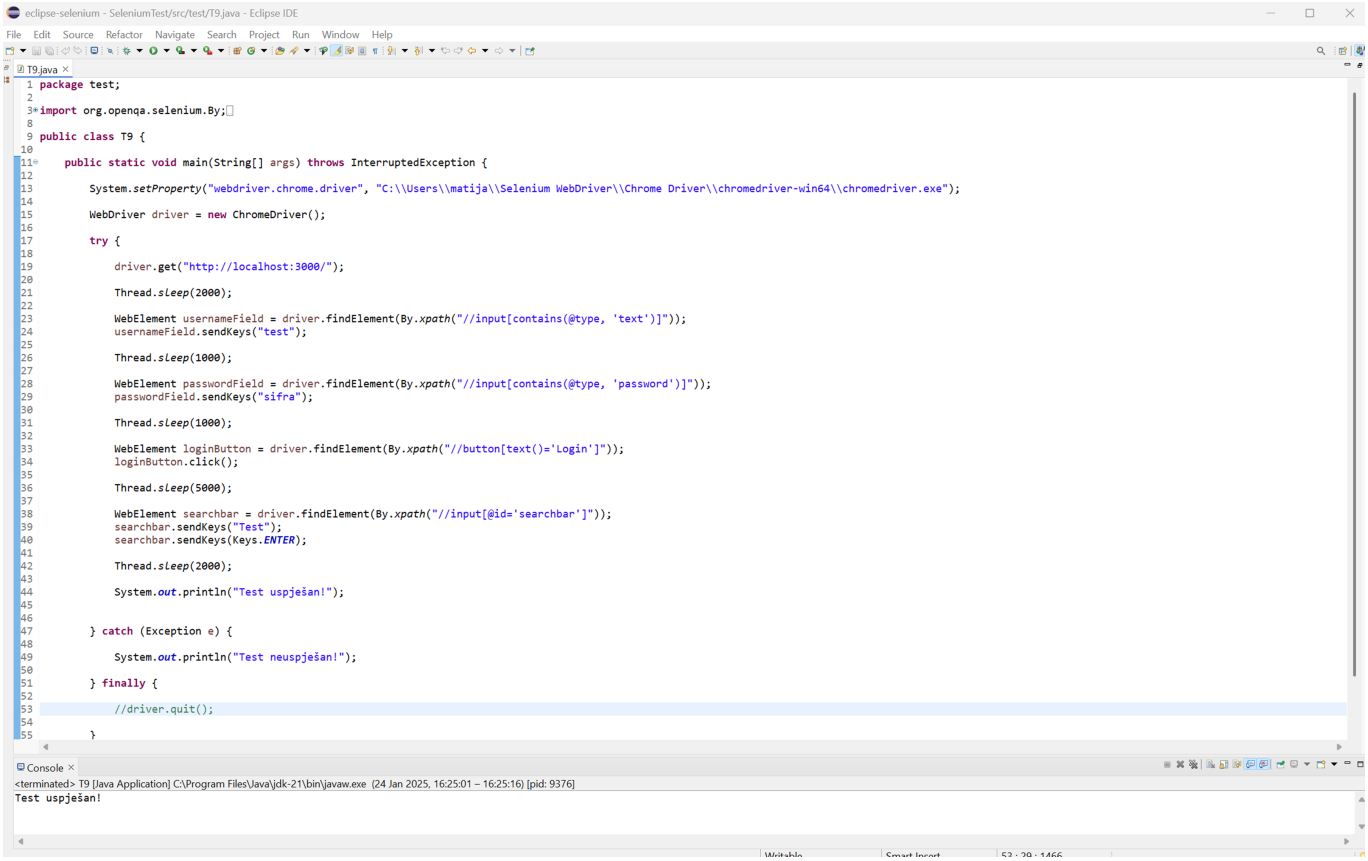




Ispitni slučaj 9 – pretraživanje kvizova

1. Ulazi: Username = „test“, Password = „sifra“, searchbar = „Test“
2. Koraci ispitivanja: otvoriti aplikaciju, unijeti korisničko ime i lozinku i stisnuti na gumb „Login“, nakon uspješne prijave kliknuti na searchbar i unijeti „Test“ i kliknuti ENTER na tipkovnici
3. Očekivani izlaz: uspješno pretraženi kvizovi
4. Dobiveni izlaz: uspješno pretraženi kvizovi

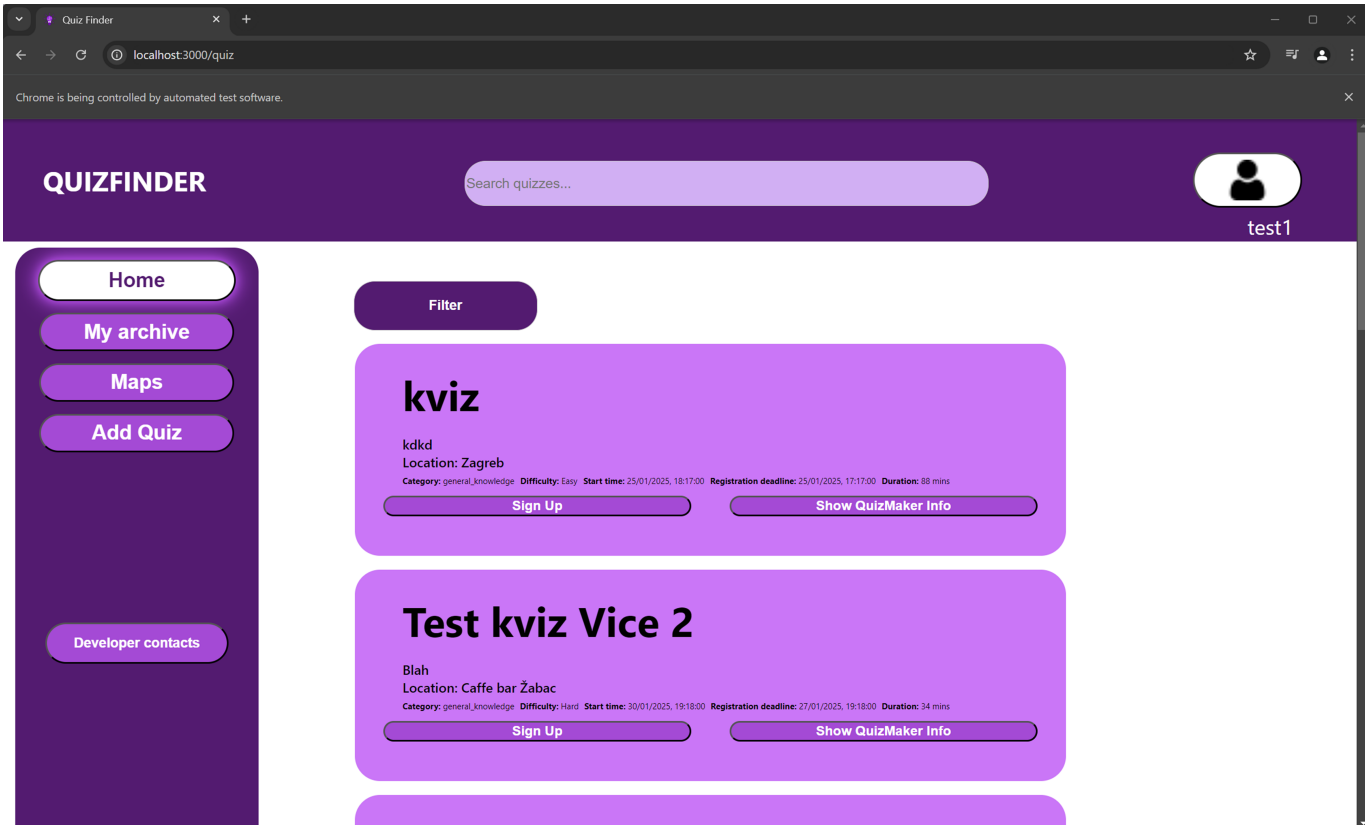


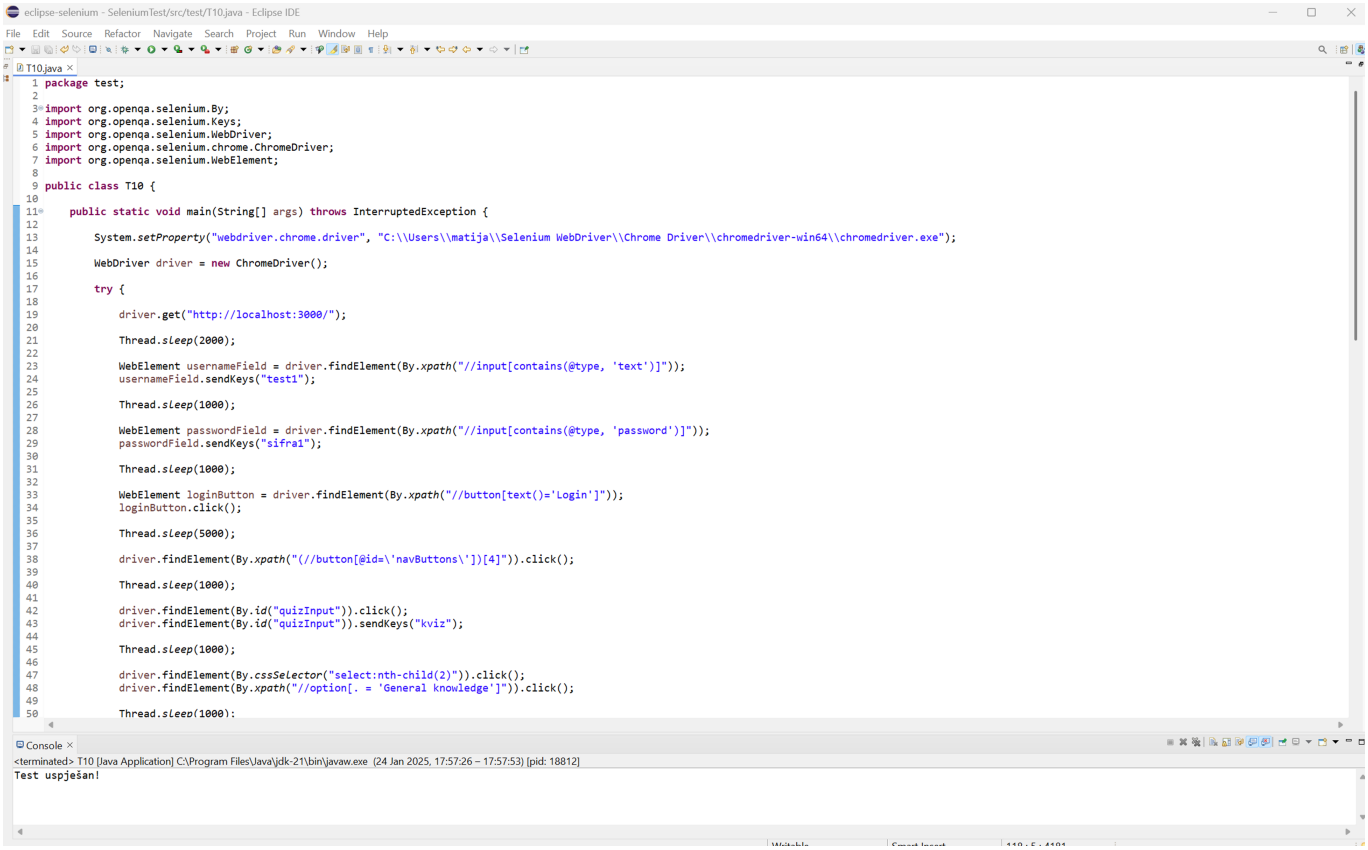


Ispitni slučaj 10 – stvaranje kvizova

1. Ulazi: Username = „test1“, Password = „sifra1“, Quiz Title = „kviz“, Category = „General knowledge“, Location = „Zagreb“, Max Teams= „10“, Fee = „5“, Duration = „88“, Difficulty = „Easy“, Description = „kdkd“, Prizes = „nagrada“, Quiz start time = „25-01-2025 18:17“, Registration deadline= „25-01-2025 17:17“

2. Koraci ispitivanja: otvoriti aplikaciju, unijeti korisničko ime i lozinku i stisnuti na gumb „Login“, nakon uspješne prijave kliknuti na gumb „Add Quiz“, unijeti ulazne podatke i kliknuti „Submit Quiz“
3. Očekivani izlaz: uspješno stvoreni kviz
4. Dobiveni izlaz: uspješno stvoreni kviz





Korištene tehnologije i alati

1. Programski jezici:

- **JavaScript** (verzija 16.13): Glavni programski jezik za frontend aplikacije, omogućujući dinamičko ponašanje na web stranici. JavaScript je osnova za React aplikaciju.
- **Python** (verzija 3.9): Korišten za backend aplikacije, omogućujući razvoj u Django okruženju. Python je izabran zbog svoje jednostavnosti, učinkovitosti u razvoju web aplikacija i velikog broja dostupnih biblioteka.

2. Radni okviri i biblioteke:

- **React** (verzija 18): Popularna JavaScript biblioteka koja omogućuje izgradnju dinamičnih i interaktivnih korisničkih sučelja. React omogućava modularni pristup razvoju aplikacija kroz ponovnu upotrebu komponenti, što ubrzava razvoj i olakšava održavanje.
- **Django** (verzija 4.1): Python framework za backend aplikacije. Django je odabran zbog svoje sigurnosti, brzine razvoja, i lakoće za implementaciju RESTful API-ja i povezivanje s bazama podataka.
- **Vite** (verzija 4.0): Alat za izgradnju frontend aplikacija, koji omogućuje brzu izradu i pokretanje React aplikacija, kao i optimizaciju za produkcijsko okruženje.
- **Supabase**: Alatza upravljanje bazama podataka. Supabase je odabran zbog svoje jednostavnosti u postavljanju i upravljanju PostgreSQL bazama podataka u oblaku.

3. Baza podataka:

- **PostgreSQL** (verzija 13): Relacijska baza podataka koja se koristi za pohranu podataka aplikacije, uključujući korisničke račune, kvizove, recenzije, itd. PostgreSQL je odabrana zbog svoje stabilnosti, skalabilnosti i moćnih značajki kao što su transakcije i podrška za složene upite.
- **Supabase**: Supabase je platforma koja omogućuje jednostavno povezivanje s PostgreSQL bazama podataka u oblaku, pružajući API, autentifikaciju i praćenje podataka bez potrebe za vlastitim poslužiteljima.

4. Razvojni alati:

- **Visual Studio Code** (verzija 1.72): Glavni IDE za razvoj frontend i backend dijela aplikacije. VS Code je odabran zbog svoje fleksibilnosti, velike podrške za ekstenzije i integraciju s Git-om.
- **Git** (verzija 2.34): Alat za verzioniranje koda koji omogućuje suradnju u timu i praćenje promjena u kodu. Git je korišten za upravljanje verzijama izvornog koda i zajednički rad na projektu.

5. Alati za ispitivanje:

- **Selenium** (verzija 4.0): Alat za UI ispitivanje koji omogućuje automatizaciju interakcija s web preglednicima. Selenium je korišten za testiranje korisničkog sučelja aplikacije, osiguravajući da svi elementi funkcioniraju ispravno na različitim uređajima i preglednicima. Alat je odabran zbog svoje fleksibilnosti i podrške za više jezika, uključujući Python.

6. Alati za razmještaj:

7. Cloud platforma:

- **Render**: Platforma korištena za hostanje backend aplikacije. Render omogućuje jednostavnu i pouzdanu implementaciju aplikacija, podržava povezivanje s PostgreSQL bazama podataka, te osigurava skalabilnost i dostupnost aplikacije u produkcijskom okruženju.

Obrazloženje izbora tehnologija:

1. **React** je odabran za frontend jer omogućuje brzi razvoj interaktivnih korisničkih sučelja uz jednostavnu organizaciju koda kroz komponente.
2. **Django** je odabran za backend zbog svoje moćne administracije, integracije s bazama podataka, i jednostavne implementacije RESTful API-ja.
3. **PostgreSQL** je robusna baza podataka koja nudi bogate mogućnosti za pohranu i upravljanje podacima, uključujući složene upite i podršku za transakcije.
4. **Supabase** je odabran kao cloud rješenje jer omogućuje jednostavno upravljanje bazama podataka i autentifikaciju s minimalnim naporom.

Opis konfiguracije:

Za razvoj frontend dijela, korišten je **Vite** za optimizirani razvoj i brzo pokretanje aplikacije. **React** je postavljen s najnovijim verzijama, a za backend se koristi **Django** koji je konfiguriran za povezivanje s Supabase-om.

1. Instalacija

Preduvjeti:

Za pokretanje aplikacije potrebno je imati instalirane sljedeće komponente:

- **Node.js** (najmanje verzija 16) za frontend
- **npm** (za upravljanje paketima)
- **Python 3.x** (najnovija verzija)
- **Django** (za backend)
- Bazu podataka po želji (Supabase online BP)

Preuzimanje izvornog koda:

Za preuzimanje izvornog koda, kloniraj Git repozitorij aplikacije:

```
git clone https://github.com/nikolasjaricfer/magna-creare.git
cd quizfinder
```

Instalacija ovisnosti

```
cd frontend
npm install
```

```
cd backend
pip install -r requirements.txt
```

2. Postavke

Konfiguracijske datoteke:

Konfiguracija aplikacije pohranjena je u `.env` datoteci, koja se nalazi u korijenskom direktoriju aplikacije. Ova datoteka sadrži sve ključne postavke koje trebate prilagoditi, uključujući postavke za bazu podataka, API ključeve i druge važne varijable.

Primjer `.env` datoteke:

```
DATABASE_URL=postgres://user:password@supabase-url:5432/dbname
API_KEY=your_api_key
SUPABASE_URL=https://your-supabase-url.supabase.co
SUPABASE_ANON_KEY=your_supabase_anon_key
```

```
```

- **DATABASE_URL**: Ovdje unosite URL za vašu bazu podataka, uključujući korisničko ime, lozinku, adresu poslužitelja, port i naziv baze podataka. Ako koristite Supabase, zamijenite `localhost` s URL-om Supabase poslužitelja i odgovarajuće postavke.
- **API_KEY**: Ovdje unesite svoj API ključ koji će aplikacija koristiti za povezivanje s vanjskim servisima.
- **SUPABASE_URL** i **SUPABASE_ANON_KEY**: Ove varijable postavite ako koristite Supabase za pohranu podataka. Pronaći ćete ove postavke u postavkama vašeg Supabase projekta.

Postavke baze podataka:

Kako koristiš Supabase, ne morate lokalno postavljati bazu podataka, ali ćete morati izvršiti migracije i postaviti početne podatke.

Inicijalizacija baze podataka i migracije:

Za inicijalizaciju baze podataka i izvršenje svih potrebnih migracija, pokrenite sljedeće naredbe:

```
npm run db:migrate
```

Postavljanje inicijalnih podataka: Ako je potrebno, možete postaviti inicijalne podatke u bazu pomoću naredbe za seeding (postavljanje početnih podataka):

```
npm run db:seed
```

Ove naredbe osiguravaju da vaša baza podataka bude ispravno postavljena i da su svi potrebni podaci (npr. korisnici, uloge, postavke) prisutni u bazi prije pokretanja aplikacije.

Ako koristite Supabase, migracije i seeding možete postaviti unutar Supabase sučelja, no može biti korisno imati ove opcije u aplikaciji za lakšu automatizaciju.

Ovaj odjeljak pokriva konfiguraciju aplikacije kroz `.env` datoteku i kako postaviti bazu podataka. Ako imaš specifične zahtjeve za migracije ili druge postavke, slobodno mi javi, pa mogu prilagoditi upute.

3. Pokretanje aplikacije

- Ući u radni direktorij

```
cd magna-creare
```

- Otvoriti terminal te ući u backend i pokrenuti sljedeće komande

```
```

```
cd backend
python ...\manage.py runserver
```

- Otvoriti drugi terminal te ući u frontend i pokrenuti sljedeće komande

```
cd frontend
npm run dev
```

4. Upute za administratore

- Pristup administratorskom sučelju:
  - prijava s admin role-om
  - početni podatci za prijavu admina mogu se dobiti na upit developerima
- Redovito održavanje
  - Čišćenje spama
  - Brisanje duplih koristnika
  - Brisanje vulgarnih komentara
  - Pregled logova koristnika
  - Ažuriranje aplikacije
- Rješavanje problema
  - Komunikacija s developerima
  - Error logs

5. Primjer za Render platformu (Cloud Deploy)

Opis pristupa aplikaciji na javnom poslužitelju

Pristup putem internetskog preglednika:

Aplikaciju možete pristupiti putem internetskog preglednika na sljedećem URL-u: QuizFinder

Koraci za pristup:

1. Otvorite internetski preglednik (npr. Chrome, Firefox, Edge).
2. U adresnu traku unesite URL aplikacije: `https://yourdomain.com`.
3. Pritisnite Enter za učitavanje aplikacije.

Ograničenja pristupa:

- **Sigurnost**: Aplikacija koristi HTTPS za sigurnu komunikaciju, osiguravajući enkripciju podataka koji se šalju i primaju između poslužitelja i klijenta.
- **Autentifikacija**: Neregistrirani korisnici mogu pregledavati dostupne kvizove, ali moraju se prijaviti (ili registrirati) kako bi sudjelovali u kvizovima ili pristupili dodatnim funkcijama.
- **Ovisnost o internetu**: Aplikacija zahtijeva stabilnu internetsku vezu za pristup svim funkcijama i podacima na javnom poslužitelju.

Pristup administratorskom sučelju:

Administratori aplikacije mogu pristupiti administratorskom sučelju na sljedećem URL-u:

Na ovoj stranici možete upravljati svim funkcijama aplikacije, uključujući upravljanje korisnicima, kvizovima, recenzijama i ostalim postavkama aplikacije.

Koraci za pristup administratorskom sučelju:

1. Prvo se prijavite kao administrator s vašim korisničkim računom.
2. Posjetite URL za administratorsko sučelje: `https://yourdomain.com/admin`.
3. Unesite svoje administrativne podatke za prijavu.

Ograničenja pristupa administratorskom sučelju:

- Samo korisnici s administratorskom ulogom mogu pristupiti administratorskom sučelju.
- Preporučuje se redovito ažuriranje administratorskih lozinki za veću sigurnost.
- Ako koristite vanjski servis poput Supabase, provjerite postavke pristupa korisnicima kako biste osigurali da samo ovlašteni korisnici imaju administratorski pristup.

Zaključak projekta

Projekt "Quiz Finder" uspješno smo razvili kao full-stack web aplikaciju koja omogućava korisnicima jednostavno pretraživanje i sudjelovanje u pub kvizovima. Tijekom rada na projektu uspjeli smo integrirati sve ključne funkcionalnosti, poput različitih korisničkih uloga, Google Maps API-ja, vanjskih servisa za autentifikaciju (OAuth2), i implementirati bazu podataka koristeći Supabase s PostgreSQL-om. Projekt je pružio praktično iskustvo rada u timskom okruženju, kao i rješavanje izazova koji se pojavljuju u procesu razvoja kompleksnih aplikacija.

Utjecaj na budući rad i karijeru

Rad na ovom projektu značajno je unaprijedio naše tehničke vještine, poput razvoja s modernim tehnologijama (Django, React, Vite), integracije vanjskih servisa te rada s bazama podataka. Osim tehničkih aspekata, stekli smo iskustvo u timskom planiranju, dokumentiranju projekata i komunikaciji, što će biti neprocjenjivo u budućim profesionalnim angažmanima.

Ovaj projekt nam otvara vrata prema karijerama u razvoju softvera, pogotovo u domenama web developmenta i full-stack razvoja. Također, implementacija složenih funkcionalnosti, poput autentifikacije i integracija s API-jevima, služi kao dobar primjer naše sposobnosti da se nosimo s izazovnim zahtjevima u stvarnim projektima. Dugoročno gledano, iskustvo stečeno ovim projektom bit će nam temelj za buduće uspješne projekte i priliku za daljnje usavršavanje.

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak	Nikolas Jarić	20.10.2024.
0.2	Dodan opis projektnog zadatka	Nikolas Jarić	22.10.2024.
0.3	Dodani funkcionalni i nefunkcionalni zahtjevi	Nikolas Jarić	23.10.2024.
0.4	Opisani svi dionici	Matija Križević	24.10.2024.
0.5	Napisana arhitektura web aplikacije	Nikolas Jarić	30.10.2024.
0.6	Stavljeni UC i sekvencijski dijagrami	Matija Križević	1.11.2024.
0.7	Stavljen opis baze	Matija Križević	3.11.2024.
0.8	Stavljen dijagram razreda	Matija Križević	6.11.2024.
0.9	Opiseni korištene tehnologije i alati	Nikolas Jarić	7.11.2024.
1.0	Napravljen popis literature	Nikolas Jarić	9.11.2024.
1.1	Popravljeni dijagrami po naputku asistenta	Matija Križević	12.12.2024.
1.2	Zapisani sastanci u Wiki	Matija Križević	21.1.2025.
1.3	Dijagram razreda	Matija Križević	10.1.2025.
1.4	Dijagram stanja	Matija Križević	11.1.2025.
1.5	Dijagram aktivnosti	Matija Križević	12.1.2025.
1.6	Dijagram komponenti	Matija Križević	13.1.2025.
1.7	Korištene tehnologije i alati	Matija Križević	14.1.2025.
1.8	Dijagram razmještaja	Matija Križević	15.1.2025.
1.9	Upute za puštanje u pogon	Matija Križević	16.1.2025.
2.0	Dnevnik sastajanja	Matija Križević	17.1.2025.
2.1	Dokumentacija	Matija Križević	18.1.2025.
2.2	Zaključak i budući rad	Matija Križević	19.1.2025.
2.3	Popis literature	Matija Križević	22.1.2025.

1. Programsko inženjerstvo, FER ZEMRIS, http://www.fer.hr/predmet/proinz
2. Astah Community, http://astah.net/editions/uml-new
3. Stack Overflow Community https://stackoverflow.com
4. Geeks For Geeks Community https://www.geeksforgeeks.org

<u>Dnevnik sastanaka</u>:

1. sastanak

- **Datum**: 15. listopada 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
    - upoznavanje članova tima međusobno
    - biranje imena
    - raspravljanje o radnom iskustvu, željama, idejama i znanju

2. sastanak

- **Datum**: 16. listopada 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
    - nakon prvog labosa
    - odluka da radimo svoju temu

3. sastanak

- **Datum**: 17. listopada 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
    - članovi iznose svoje teme te se glasa koja tema je najbolja
    - Quiz Finder tema je pobjedila
    - Detaljno razrađivanje tehničkog opisa web aplikacije Quiz Finder te slanje profesoru na odobrenje

4. sastanak

- **Datum**: 20. listopada 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
    - Napravljen Git na GitHubu
    - Svi članovi dodani kao collaborators
    - Prvi testni pullovi, addovi, commitovi i pushevi

5. sastanak

- **Datum**: 22. listopada 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
    - Raspravljanje o funkcionalnim i nefunkcionalnim zahtjevima
    - Pripremanje za sljedeći labos

6. sastanak

- **Datum**: 26. listopada 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev

- **Tema sastanka**:
  - Raspravljanje o arhitekturi web aplikacije
  - Raspravljanje te izrada sekvencijskih i use case dijagrama
  - Team leader je svima podijelio zadatke koje moraju napraviti tj. koje tehnologije naučiti i napraviti mali samostalni projekt
  - Ema Skoko uzima ulogu dizajnera

7. sastanak

- **Datum**: 3. studenog 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Raspravljanje o danim zadatcima
  - Pokušaj dogovora o ulogama (frontend, backend, full-stack, dokumentacija)
  - Postavljanje inicijalnog Python Django i React projekta
  - Ema Skoko je osmislila i napravila dizajn koji su svi članovi prihvatili

8. sastanak

- **Datum**: 5. studenog 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Dogovor uloga
  - Ema Skoko i Petar Krtalić uzimaju ulogu frontenda
  - Ante Perić i Vice Sladovljev uzimaju ulogu backenda
  - Matija Križević uzima ulogu full-stack developera (frontend i backend featuri) uz pomoć pri dokumentaciji (primarno dijagrami)
  - Nikolas Jarić, već izabrani team leader, uzima također ulogu glavnog za dokumentaciju te full stack (frontend i backend featuri)

9. sastanak

- **Datum**: 9. studenog 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Raspravljanje o problemima do kojih su članovi tima došli u svojim samostalnim projektima
  - Raspravljanje o prvom backend i frontend kodu napisanom za web aplikaciju Quiz Finder

10. sastanak

- **Datum**: 12. studenog 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Međusobno pomaganje oko bugova u kodu
  - Spajanje frontenda i backenda u cjelinu

11. sastanak

- **Datum**: 13. studenog 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev

- **Tema sastanka**:
  - Odabir jos jednog feature za napraviti
  - Raspodjela posla

12. sastanak

- **Datum**: 15. studenog 2024.
- **Prisustvovali:** Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Rasprava o kodu
  - Zadnja provjera dokumentacije
  - Zajedničko rješavanje bugova u kodu
  - Deployanje aplikacije

13. sastanak

- **Datum**: 20. studenog 2024.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Dodavanje Guest uloge.
  - Dodavanje mogućnosti brisanja kvizova i recenzija za admina.
  - Implementacija funkcije dodavanja recenzija.

14. sastanak

- **Datum**: 29. studenog 2024.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Implementacija "Moje arhive" za prikaz prošlih prijava na kvizove.
  - Dodavanje mogućnosti admina za brisanje korisnika.
  - Početak implementacije funkcionalnosti prijavljivanja na kviz.

15. sastanak

- **Datum**: 9. prosinca 2024.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Dovršavanje funkcionalnosti prijavljivanja na kviz.
  - Razrada pretraživanja ("Search") unutar aplikacije.
  - Početna implementacija osnovnih filtera.

16. sastanak

- **Datum**: 11. prosinca 2024.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Proširivanje filtera za pretraživanje kvizova.
  - Dodavanje prikaza informacija o quizmakerima.
  - Priprema za proširenje funkcionalnosti s lokacijama i mapama.

17. sastanak

- **Datum**: 18. prosinca 2024.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Proširivanje funkcionalnosti lokacija za povezivanje s Google Maps API-jem.
  - Integracija funkcionalnosti za računanje udaljenosti od kviza.
  - Testiranje funkcionalnosti vezanih uz lokacije.

18. sastanak

- **Datum**: 27. prosinca 2024.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Pisanje i izvršavanje ispitnih slučajeva za osnovne funkcionalnosti.
  - Debugging i ispravljanje grešaka u integraciji lokacija.
  - Početak pripreme projektne dokumentacije.

19. sastanak

- **Datum**: 3. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Provedba osnovnih "Admin mogućnosti".
  - Dokumentiranje ispitnih slučajeva i dijagrama.
  - Završna testiranja Google Maps funkcionalnosti.
  - Rad na dizajnu prezentacije za projekt.

20. sastanak

- **Datum**: 6. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Dovršavanje projektne dokumentacije.
  - Finalizacija prezentacije.
  - Priprema za prezentaciju aplikacije.

21. sastanak

- **Datum**: 8. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Proširenje filtera za pretraživanje prema quizmakerima i lokacijama.
  - Testiranje svih funkcionalnosti aplikacije.
  - Završna priprema za deploy.

22. sastanak

- **Datum**: 10. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev

- **Tema sastanka**:
  - Završna provjera administracijskih funkcionalnosti.
  - Optimizacija pretraživanja i filtera.
  - Završno testiranje prijavljivanja na kviz.

23. sastanak

- **Datum**: 11. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Debugging funkcionalnosti vezanih za udaljenost i mape.
  - Provjera svih korisničkih rola (guest, admin).
  - Dokumentiranje svih bugova i rješenja.

24. sastanak

- **Datum**: 14. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Detaljno testiranje funkcionalnosti admin panela.
  - Završna testiranja korisničkog sučelja.
  - Priprema za finalni deploy na Render.

25. sastanak

- **Datum**: 15. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Prva provjera aplikacije na javnom poslužitelju.
  - Prikupljanje povratnih informacija od tima.

26. sastanak

- **Datum**: 17. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Rješavanje prijavljenih problema nakon prvog deploya.
  - Finalna provjera rada aplikacije na produkciji.
  - Priprema za prezentaciju aplikacije vanjskim korisnicima.

27. sastanak

- **Datum**: 21. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Zadnja provjera svih funkcionalnosti.
  - Optimizacija performansi aplikacije na poslužitelju.
  - Završna priprema prezentacijskih materijala.

28. sastanak

- **Datum**: 22. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Testiranje opterećenja aplikacije.
  - Analiza logova nakon testiranja na produkciji.
  - Dovršavanje zadnjih detalja za korisničko sučelje.

29. sastanak

- **Datum**: 24. siječnja 2025.
- **Prisustvovali**: Nikolas Jarić, Ema Skoko, Ante Perić, Matija Križević, Petar Krtalić, Vice Sladoljev
- **Tema sastanka**:
  - Evaluacija projekta i tima.
  - Deploy aplikacije na Render.
  - Završna revizija dokumentacije.
  - Prezentacija finalnog proizvoda unutar tima.

Plan rada

Prvi ciklus

Aktivnost	Tjedan 1	Tjedan 2	Tjedan 3	Tjedan 4	Tjedan 5	Tjedan 6	Tjedan 7
Upoznavanje tima		✔					
Smišljanje teme i setup Gita			✔				
Razrada funkcionalnih i nefunkcionalnih zahtjeva				✔			
Razrada arhitekture web aplikacije te stvaranje use case i sekvencijskih dijagrama					✔		
Inicijalizacija projekta						✔	
Izrada register i login pagea						✔	
Feature mijenjanja lozinke korisnika i dodavanje kviza							✔
Push na main branch, deployment te završetak dokumentacije za 1. ciklus							✔

Početak drugog ciklusa

Aktivnost	Tjedan 8	Tjedan 9
Početak implementacije "guest user" funkcionalnosti	✔	

Aktivnost	Tjedan 8	Tjedan 9
Integracija Google Maps API-ja		✔
Izrada "Contact Page"	✔	
Osnovna implementacija administracijskih funkcija		✔
Testiranje implementiranih funkcionalnosti		✔

Zimski praznici

Aktivnost	Praznici tjedan 1	Praznici tjedan 2
Planiranje naprednih funkcionalnosti	✔	
Priprema za implementaciju filtera i pretrage		✔
Revizija dosadašnjih funkcionalnosti	✔	
Detaljna dokumentacija		✔

Nastavak drugog ciklusa

Aktivnost	Tjedan 10	Tjedan 11	Tjedan 12	Tjedan 13
Dodavanje naprednih filtera	✔			
Funkcionalnost prijave na kviz		✔		
Implementacija "Admin" opcija za upravljanje kvizovima			✔	
Dodavanje mogućnosti brisanja recenzija			✔	
Završetak projektne dokumentacije			✔	
Testiranje aplikacije i rješavanje bugova			✔	
Izrada prezentacije				✔

Legenda:

- ✔ označava ispunjen zadatak

Tablica aktivnosti

Feature	Sloj	Tehnologija	Opis	Osoba	Uloženo vrijeme
Upravljanje projektom	Organizacijski	/	Koordiniranje rada na projektu	Nikolas Jarić	16h

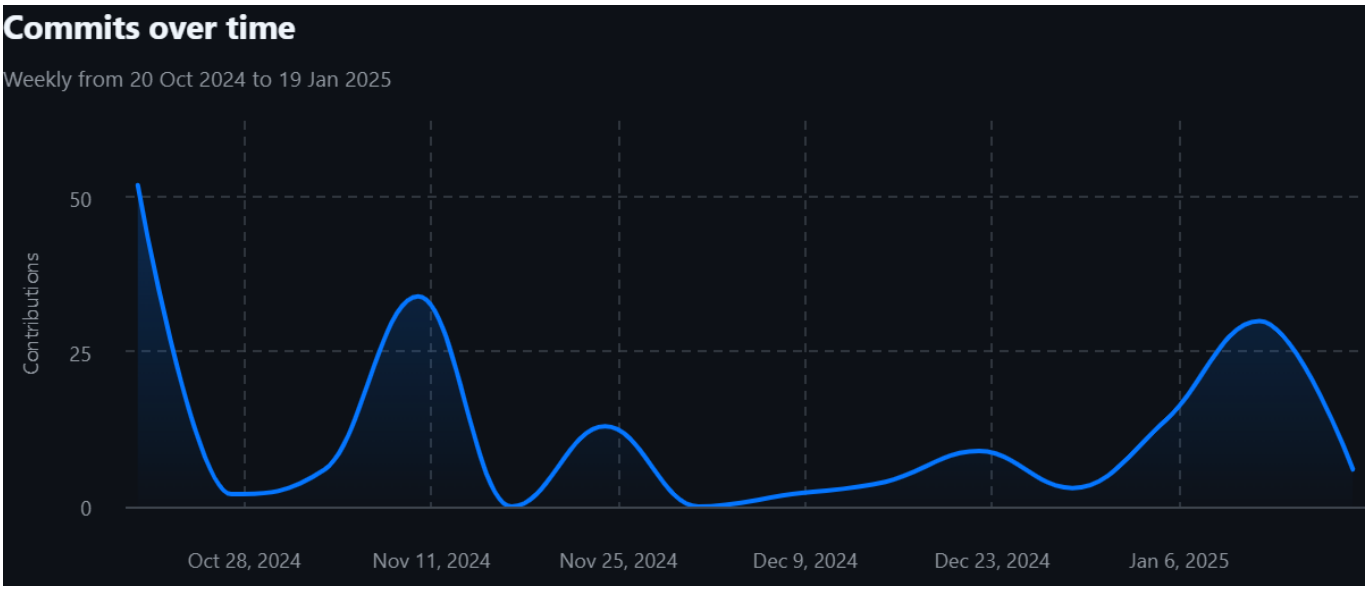
Feature	Sloj	Tehnologija	Opis	Osoba	Uloženo vrijeme
Opis projektnog zadatka	Dokumentacija	Markdown	Napravljen detaljan tehnički opis web aplikacije	Ante Perić	2h
Funkcionalni zahtjevi	Dokumentacija	Markdown	Razrada funkcionalnih zahtjeva aplikacije	Petar Krtalić	2h
Opis pojedinih obrazaca	Dokumentacija	Markdown	Opis svakog traženog obrasca	Matija Križević	7h
Dijagram obrazaca	Dokumentacija	Astah	Izrada svakog dijagrama obrasca u Astahu	Matija Križević	6h
Sekvencijski dijagrami	Dokumentacija	Astah	Izrada svakog sekvencijskog dijagrama u Astahu	Matija Križević	4h
Opis ostalih zahtjeva	Dokumentacija	Markdown	Opis ostalih potrebnih zahtjeva	Matija Križević	1h
Arhitektura i dizajn sustava	Dokumentacija i dizajn	Markdown i Figma	Izrada dizajna web aplikacije i opis arhitekture	Ema Skoko	5h
Baza podataka	Dokumentacija i baza podataka	Markdown i Astah	Osmišljena struktura baze podataka i nacrtana	Ante Perić	2h
Dijagram razreda	Dokumentacija	Astah	Crtanje dijagrama razreda	Matija Križević	3h
Dijagram stanja	Dokumentacija	Astah	Crtanje dijagrama stanja	Matija Križević	2h
Dijagram aktivnosti	Dokumentacija	Astah	Crtanje dijagrama aktivnosti	Matija Križević	1,5h
Dijagram komponenti	Dokumentacija	Astah	Crtanje dijagrama komponenti	Matija Križević	3h
Korištene tehnologije i alati	Dokumentacija	Markdown	Zapisivanje korištenih tehnologija i alata	Matija Križević	0,5h
Ispitivanje programskog rješenja	Dokumentacija	Selenium	Rađenje automatsih skripti	Matija Križević	10h
Dijagram razmještaja	Dokumentacija	Astah	Crtanje dijagrama razmještaja	Matija Križević	2h

Feature	Sloj	Tehnologija	Opis	Osoba	Uloženo vrijeme
Upute za puštanje u pogon	Dokumentacija	Markdown	Pisanje uputa za puštanje u pogon	Matija Križević	1h
Dnevnik sastajanja	Dokumentacija	Markdown	Bilježenje događaje svakog sastanka	Nikolas Jarić	5h
Zaključak i budući rad	Dokumentacija	Markdown	Zapis zaključka i budućeg rada	Matija Križević	0,5h
Popis literature	Dokumentacija	Markdown	Bilježenje korištene literature	Nikolas Jarić	0,5h
Izrada baze podataka	Baza podataka	PostgreSQL	Izrada baze podataka u Supabase servisu	Nikolas Jarić	0,5h
Spajanje s bazom podataka	Backend i baza podataka	Python Django	Spajanje backenda s bazom podataka	Ante Perić	3h
Izrada prezentacije	Dokumentacija	Google Slides i MS Powerpoint	Izrađivanje prezentcije	Matija Križević	5h
Izrada LoginPage	Frontend	React	Izrada frontenda Login Page-a	Ema Skoko	10h
Izrada backenda Login-a	Backend	Python Django	Izrada backenda za login	Ante Perić	7h
Izrada kontakt stranice	Frontend	React	Izrada stranice s kontaktima developera	Matija Križević	2h
Izrada backenda mijenjanja lozinke korisnika	Backend	Python DJango	Izrada backenda za mijenjanje lozinke korisnika	Vice Sladoljev	3h
Mijenjanje lozinke	Frontend	React	Stvaranje sučelja za mijenjanje lozinke korisnika	Petar Krtalić	4h
Integracija OAuth2 registracije	Backend	Python Django	Stvaranje funkcionalnosti OAauth2 servisa	Ante Perić	8h
Stvaranje frontenda za OAuth2	Frontend	React	Stvaranje frontenda za OAuth2	Ema Skoko	5h
Funkcionalnost dodavanja kvizova	Backend	Python Django	Dodano dodavanje kvizova u bazu sa svim atributima	Ante Perić	4h

Feature	Sloj	Tehnologija	Opis	Osoba	Uloženo vrijeme
Dodavanje kvizova	Frontend	React	Forms sučelje za dodavanja kviza	Petar Krtalić	5h
Deployanje aplikacije	Svi slojevi	Render docs	Deployanje aplikacije	Nikolas Jarić	2h
Početak implementacije "guest user"	Backend	Django	Implementacija osnovne funkcionalnosti za goste	Nikolas Jarić	8h
Integracija Google Maps API-ja	Frontend	React	Ugrađen Google Maps API u aplikaciju	Ema Skoko	6h
Osnovna implementacija administracijskih funkcija	Backend	Django	Kreirane osnovne admin funkcionalnosti	Petar Krtalić	7h
Testiranje implementiranih funkcionalnosti	Testiranje	Selenium	Testiranje svih novih funkcionalnosti aplikacije	Vice Sladoljev	4h
Planiranje naprednih funkcionalnosti	Organizacijski	/	Planiranje novih funkcionalnosti i poboljšanja	Nikolas Jarić	3h
Priprema za implementaciju filtera i pretrage	Dokumentacija	Markdown	Razrada detalja za implementaciju filtera i pretrage	Ante Perić	6h
Revizija dosadašnjih funkcionalnosti	Testiranje	Manual	Provjera ispravnosti postojećih funkcionalnosti	Ema Skoko	4h
Dodavanje naprednih filtera	Backend	Django	Implementirani dodatni filteri za pretragu kvizova	Petar Krtalić	9h
Funkcionalnost prijave na kviz	Frontend	React	Kreirana mogućnost prijave korisnika na kvizove	Matija Križević	8h
Implementacija "Admin" opcija	Backend	Django	Dodane opcije za administraciju kvizova	Vice Sladoljev	7h
Dodavanje mogućnosti brisanja recenzija	Backend	Django	Omogućeno brisanje recenzija putem admin panela	Ema Skoko	5h

Feature	Sloj	Tehnologija	Opis	Osoba	Uloženo vrijeme
Završetak projektne dokumentacije	Dokumentacija	Markdown	Dovršena i pregledana tehnička dokumentacija	Ante Perić	10h
Testiranje aplikacije i rješavanje bugova	Testiranje	Selenium, Manual	Testirani svi dijelovi aplikacije i ispravljeni bugovi	Matija Križević	11h
Izrada prezentacije	Dokumentacija	PowerPoint	Pripremljena prezentacija za završno izlaganje	Vice Sladoljev	6h
Filtriranje na frontendu	Frontend	React	Dodane opcije za filtriranje kvizova na korisničkom sučelju	Ema Skoko	7h
Pretraživanje na frontendu	Frontend	React	Implementirana opcija pretraživanja kvizova	Ante Perić	5h
Admin panel na frontendu	Frontend	React	Izrađeno korisničko sučelje za administraciju kvizova	Petar Krtalić	9h

Dijagram pregleda promjena



Ključni izazovi i rješenja

<u>Izazovi</u>:

- Nitko u timu nema nikakvo pravo radno iskustvo

- Prvi susret s React.js-om
- Prvi susret s Pythong Djangom
- Prvi put primijenjeno korištenje baze podataka
- Spajanje frontenda i backenda
- Tehnička komunikacija
- Crtanje različitih dijagrama

Rješenja:

- Sposobnost učenja
- Uloženi mnogi sati
- Stalna komunikacija
- Česti sastanci
- Mnogo samostalnog istraživanja i eksperimentiranja