

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Programsko inženjerstvo ak.god 2024./2025  
QuizWiz

Nastavnik:

Vlado Sruk

Tim: <TG09.1>

Ime tima: Wizionari

## Sadržaj

1. Opis projektnog zadatka .....	4
1.1 Uvod i cilj projekta .....	4
1.2 Opis problema i motivacija .....	4
1.3 Potencijalna korist projekta .....	4
1.4 Postojeća slična rješenja .....	5
1.5 Skup korisnika.....	6
1.5.1 Uloge korisnika .....	6
1.6 Opseg i mogućnost prilagodbe rješenja.....	6
1.7 Moguće nadogradnje .....	7
2. Analiza zahtjeva.....	7
Funkcionalni zahtjevi.....	7
Ostali (nefunkcionalni) zahtjevi.....	10
Zahtjevi održivosti .....	10
Zahtjevi skalabilnosti.....	10
Zahtjevi dostupnosti.....	11
Sigurnosni zahtjevi .....	12
Zahtjevi performansi .....	12
Zahtjevi ograničenja .....	13
Zahtjevi interoperabilnosti.....	13
Dionici .....	13
Aktori.....	14
3. Specifikacija zahtjeva sustava.....	15
Obrasci uporabe.....	15
UC1 - Registracija korisnika/tima .....	15
UC2 - Prijava u sustav .....	15
UC3 - Odjava iz sustava .....	16
UC4 - Uređivanje profila.....	16
UC5 - Brisanje korisničkog računa .....	17
UC6 - Pretraživanje kvizova .....	17
UC7 - Prijava tima na kviz.....	18
UC8 - Povlačenje prijave.....	18
UC9 - Pregled statusa prijave .....	19
UC10 - Pregled rezultata i arhive.....	19
UC11 - Ocjenjivanje kvizova i organizatora .....	20
UC12 - Kreiranje kviza .....	20

UC13 - Uređivanje/brisanje kviza .....	21
UC14 - Upravljanje prijavama .....	21
UC15 - Unos rezultata i rang liste .....	22
UC16 - Odobravanje organizatora i upravljanje korisnicima .....	22
UC17 - Slanje obavijesti .....	22
UC18 - Brisanje recenzija .....	23
Dijagrami obrazaca uporabe .....	24
1. UC dijagram za korisničke račune .....	24
2. UC dijagram funkcionalnosti sudionika/tima .....	24
3. UC dijagram funkcionalnosti organizatora kviza .....	25
4. UC dijagram funkcionalnosti administratora i sustava .....	25
Sekvencijski dijagrami .....	26
UC1, UC2 Neprijavljeni korisnik - registracija i prijava korisnika u sustav .....	26
UC3, UC4, UC4 Prijavljeni korisnik - odjava, uređivanje i brisanje računa .....	26
UC6, UC7, UC8, UC9, UC10, UC11 Sudionik - pretraživanje kvizeva, prijava na kviz, upravljanje svojim prijavama, pregled arhive završenih kvizeva, ocjenjivanje kviza/organizatora .....	28
UC12, UC13, UC14, UC15, UC17 Organizator - upravljanje kvizevima i prijavama, automatsko slanje obavijesti .....	29
4. Arhitektura i dizajn sustava .....	31
Arhitektura sustava .....	31
4.1 Opis arhitekture .....	31
4.2 Obrazloženje odabira arhitekture .....	35
4.3 Organizacija sustava na visokoj razini .....	35
4.4 Organizacija aplikacije .....	36
Baza podataka .....	37
4.5 Opis tablica .....	37
4.6 Dijagram baze podataka .....	40
Dijagram razreda .....	41

# 1. Opis projektnog zadatka

## 1.1 Uvod i cilj projekta

Projekt **QuizWiz** predstavlja web-aplikaciju namijenjenu digitalizaciji i modernizaciji organizacije pub kvizova, omogućujući jednostavnu komunikaciju i koordinaciju između organizatora i sudionika. Cilj projekta je razviti intuitivno, centralizirano rješenje koje omogućuje upravljanje kvizovima, prijave timova, evidenciju sudionika te obavještanje o promjenama u stvarnom vremenu.

Glavna svrha aplikacije je poboljšati transparentnost i učinkovitost organizacije pub kvizova, koji su danas često vođeni ručno ili putem društvenih mreža i chat grupa. Sustav **QuizWiz** omogućuje organizatorima jednostavno dodavanje događaja, odobravanje prijave i komunikaciju s timovima, dok korisnicima pruža mogućnost pretraživanja kvizova, prijave i informiranja o promjenama.

## 1.2 Opis problema i motivacija

U praksi, većina pub kvizova organizira se neformalno – putem objava na Facebook grupama ili privatnih poruka. Takav pristup dovodi do niza poteškoća:

- nedostatka centraliziranih informacija o dostupnim kvizovima,
- nemogućnosti praćenja broja prijavljenih timova i kapaciteta,
- nejasne komunikacije između organizatora i sudionika,
- otežanog informiranja o promjenama termina ili otkazivanju događaja.

Aplikacija **QuizWiz** adresira ove probleme pružajući objedinjenu platformu za sve dionike – administratore sustava, organizatore kvizova i igrače. Sustav uvodi automatizirane prijave, ograničavanje kapaciteta, notifikacije e-mailom i komentare ispod kvizova za brzu komunikaciju. Time se eliminira potreba za neformalnim kanalima i postiže bolja kontrola, preglednost i pouzdanost organizacije kvizova.

## 1.3 Potencijalna korist projekta

Projekt **QuizWiz** donosi višestruke koristi različitim skupinama korisnika:

- **Organizatorima** omogućuje jednostavno planiranje, objavu i administraciju događaja uz automatske obavijesti i praćenje prijave.
- **Sudionicima** nudi preglednu tražilicu kvizova, transparentne prijave, brzu komunikaciju s organizatorima i pravovremene obavijesti o promjenama.
- **Administratorima** sustava osigurava nadzor nad pravilnim radom sustava, dodavanje novih organizatora i održavanje baze mjesta kvizova.

- Vlasnicima ugostiteljskih objekata indirektno koristi povećanjem posjećenosti i promocijom događaja.

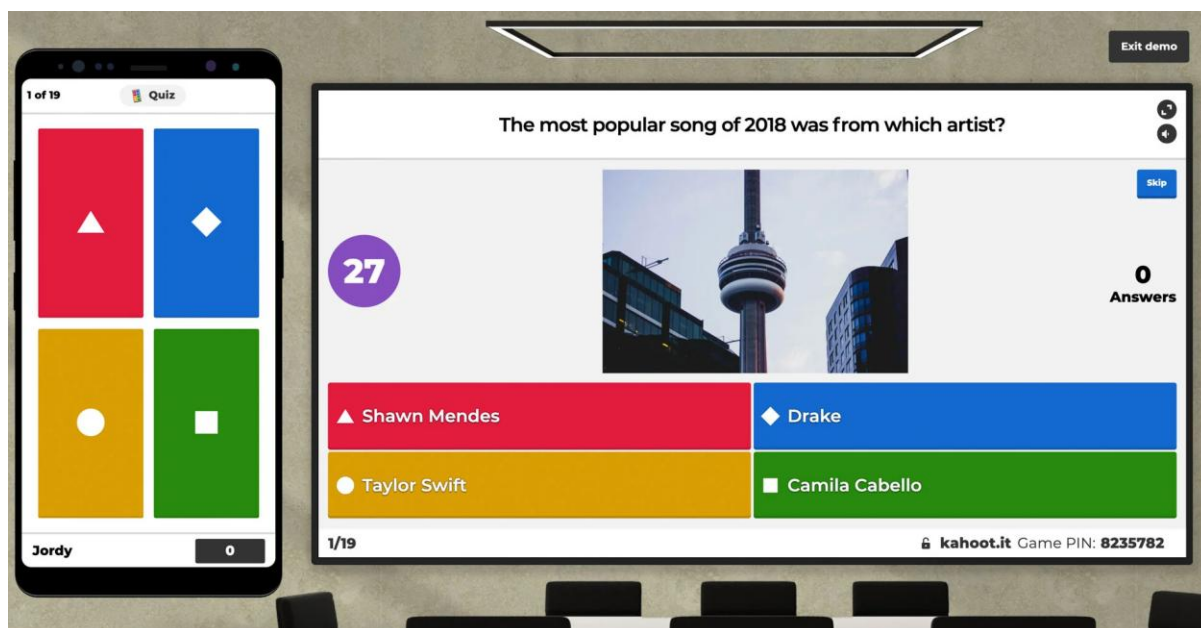
Dodatno, platforma može doprinijeti stvaranju zajednice kvizaša, potaknuti međusobno natjecanje i promovirati kulturu znanja i timskog duha.

## 1.4 Postojeća slična rješenja

Na tržištu postoji nekoliko platformi koje djelomično nude funkcionalnosti slične QuizWizu, no nijedna nije posebno prilagođena lokalnoj organizaciji pub kvizova uživo.

Kahoot!

Kahoot je globalno popularna aplikacija za interaktivne kvizove, najčešće korištena u obrazovanju. Njezina osnovna svrha je vođenje kvizova u stvarnom vremenu gdje korisnici odgovaraju putem mobilnih uređaja. Međutim, Kahoot nije namijenjen za prijave timova, upravljanje događajima ni komunikaciju između organizatora i sudionika.



Slika 1 Primjer Kahoot aplikacije

QuizPlanet

QuizPlanet je popularna mobilna aplikacija za online natjecanja u kvizovima između pojedinaca, no fokus joj je na natjecanju jedan-na-jedan, bez podrške za stvarne događaje ili timove. Projekt QuizWiz razlikuje se po tome što omogućuje fizičko sudjelovanje u lokalnim pub kvizovima i fokusira se na organizacijsku komponentu.

MyQuiz

MyQuiz je platforma za internetske kvizove i evente, često korištena za online timska natjecanja. Iako nudi upravljanje kvizovima i prijavama, fokusirana je na online okruženje, dok QuizWiz kombinira digitalnu prijavu i fizičko održavanje događaja u ugostiteljskim prostorima.

## 1.5 Skup korisnika

Sustav QuizWiz primarno je namijenjen trima skupinama - organizatorima kvizova i korisnicima (igračima) te administratorima sustava. Sekundarni korisnici mogu biti vlasnici ugostiteljskih objekata koji organiziraju kvizove te posjetitelji koji traže zanimljive događaje.

### 1.5.1 Uloge korisnika

1. Administrator sustava Administrator sustava zadužen je za održavanje ispravnosti rada aplikacije. On kreira i upravlja korisničkim računima organizatora, odobrava nove organizatore te po potrebi ažurira ili uklanja podatke o mjestima održavanja kvizova. Također ima uvid u sve kreirane događaje i korisničke aktivnosti radi osiguravanja pravilnog rada sustava.
2. Organizator kviza
  - kreira novi kviz s podacima: naslov, opis, datum, vrijeme, lokacija (Google Maps integracija), maksimalan broj timova, vrsta prijava (otvorene / uz odobrenje), oznake (npr. "opće znanje", "glazbeni", "tematski"),
  - odobrava ili odbija prijave za kvizove koji zahtijevaju verifikaciju,
  - otkazuje ili odgađa kviz uz automatsku e-mail obavijest svim prijavljenima,
  - prati broj prijavljenih timova i automatski prelazak kviza u stanje Popunjen kad se dosegne kapacitet.
3. Korisnik (igrač / tim)
  - pregledava i pretražuje kvizove po oznakama i datumu,
  - prijavljuje svoj tim na kviz (odmah ili uz odobrenje organizatora, ovisno o postavci),
  - prima obavijesti o odobrenju, otkazivanju ili odgodi kviza,
  - otkazuje prijavu tima prije održavanja kviza.
  -

## 1.6 Opseg i mogućnost prilagodbe rješenja

Projekt u prvoj fazi uključuje osnovne funkcionalnosti: upravljanje kvizovima, prijave, komunikaciju i obavijesti. Sustav je modularan i može se proširivati bez izmjena postojeće arhitekture.

Mogućnosti prilagodbe uključuju:

- dodavanje višezjezične podrške,
- prilagodbu dizajna različitim vrstama kvizova (glazbeni, tematski, opći),
- mogućnost integracije s društvenim mrežama radi promocije događaja,

- prilagodbu za mobilne uređaje s responzivnim dizajnom.

Sustav je zamišljen kao web-aplikacija s mogućnošću kasnijeg razvoja mobilne verzije.

## 1.7 Moguće nadogradnje

Potencijalne nadogradnje projekta uključuju:

- dodavanje sustava ocjenjivanja i povratnih informacija nakon održanih kvizova,
- sustav bodovanja i rang lista za timove i pojedince,
- automatizirane kvizaške lige koje povezuju više događaja istog tipa,
- analitički modul za organizatore (statistika prijava, posjećenosti i popularnosti),
- mogućnost integracije s platnim sustavima za plaćene kvizove ili posebne evente.

## 2. Analiza zahtjeva

### Funkcionalni zahtjevi

Tablica 1 Tablica Funkcionalnih zahtjeva

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-001	Sustav omogućuje registraciju	Visok	Zahtjev dionika	Ispunjavanje osnovnih informacija za kreiranje računa.
F-001.1	Registracija korisnika	Visok	Zahtjev dionika	Korisnici se mogu registrirati putem obrasca s osnovnim podacima (korisničko ime, lozinka, e-mail, naziv tima, kontakt podaci) i dobiti potvrdu o uspješnoj registraciji. Korisnik također može odabrati opciju registracije tima, dodaje naziv i članove tima.
F-001.2	Registracija organizatora kvizova	Visok	Zahtjev dionika	Organizatori mogu kreirati profile.

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-002	Sustav omogućuje prijavu	Visok	Zahtjev dionika	Ispunjavanje potrebnih informacija za log-in u sustav.
F-002.1	Prijava korisnika	Visok	Zahtjev dionika	Korisnici se mogu prijaviti pomoću korisničkog imena ili e-maila i lozinke.
F-002.2	Prijava organizatora kvizova	Visok	Zahtjev dionika	Prijava organizatora kviza pomoću e-mail adrese i lozinke.
F-003	Uređivanje profila	Srednji	Zahtjev dionika	Omogućuje se uređivanje profila korisnika i/ili organizatora.
F-003.1	Uređivanje profila korisnika	Srednji	Zahtjev dionika	Korisnici mogu ažurirati korisničko ime i kontakt podatke.
F-003.2	Uređivanje profila tima	Srednji	Zahtjev dionika	Timovi mogu ažurirati naziv, članove i kontakt informacije.
F-003.3	Uređivanje profila organizatora	Srednji	Zahtjev dionika	Organizatori mogu ažurirati podatke (naziv, kontakt, opis, lokacija).
F-004	Kreiranje, uređivanje i brisanje postojećih kvizova	Visok	Zahtjev dionika	Sustav omogućuje kreiranje, uređivanje i brisanje postojećih kvizova te upravljanje prijavama timova na kvizove.
F-004.1	Kreiranje kvizova	Visok	Zahtjev dionika	Organizatori mogu uspješno kreirati i objavljivati kvizove s detaljima (naziv, opis, datum i vrijeme održavanja, lokacija, maksimalan broj timova, status prijava).
F-004.2	Uređivanje i brisanje postojećih kvizova	Visok	Zahtjev dionika	Organizatori mogu promijeniti ili obrisati kviz.
F-004.3	Upravljanje prijavama timova i obavijestima	Visok	Zahtjev dionika	Organizatori mogu ograničiti broj prijave, zatvoriti prijave i slati obavijesti o promjenama termina ili lokacije.



ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvaćanja
F-005	Pretraživanje kvizova prema datumu, lokaciji ili nazivu	Visok	Povratne informacije korisnika	Korisnici mogu filtrirati i pronaći kviz po datumu, lokaciji ili nazivu.
F-006	Prijava timova na kviz	Visok	Zahtjev dionika	Timovi se mogu prijaviti na kviz sve dok broj prijava ne premaši limit.
F-006.1	Pregled i praćenje prijave na kvizove	Visok	Povratne informacije korisnika	Timovi mogu vidjeti status prijave i primati obavijesti o promjenama.
F-006.2	Povlačenje prijave	Srednji	Zahtjev dionika	Sudionici mogu povući prijavu prije određenog roka.
F-007	Unos rezultata i rang lista	Srednji	Dokument zahtjeva	Organizatori mogu unijeti rezultate i rang liste (informacije su dostupne timovima) nakon završetka kviza.
F-007.1	Pregled rezultata i rang liste	Srednji	Povratne informacije korisnika	Korisnici imaju pristup informacijama o rezultatu i rang listi kviza nakon završetka.
F-008	Ocjenjivanje kvizova i organizatora	Srednji	Povratne informacije korisnika	Sudionici mogu ostaviti ocjenu i recenziju za kvizove i organizatore (prosječna ocjena se prikazuje za poboljšanje kvalitete).
F-008.1	Odgovor na recenzije i davanje povratnih informacija	Nizak	Zahtjev dionika	Organizatori mogu odgovarati na recenzije i davati povratne informacije.
F-009	Pregled povijesti sudjelovanja, rezultata i poretka	Srednji	Povratne informacije korisnika	Sudionici mogu pregledavati svoje prošle kvizove i rezultate (sustav prikazuje osobnu statistiku).
F-009.1	Prikaz arhive održanih kvizova i rezultata	Nizak	Dokument zahtjeva	Organizatori i sudionici mogu pristupiti arhivi i pregledavati rezultate kvizova.

## Ostali (nefunkcionalni) zahtjevi

### Zahtjevi održivosti

Tablica 2 Tablica zahtjeva održivosti

ID zahtjeva	Opis	Prioritet
NO-1	Sustav treba biti popraćen "Priručnikom za rad" koji opisuje pravilnu upotrebu aplikacije.	Visok
NO-2	Sustav treba imati dovoljnu dokumentaciju koda i arhitekture.	Srednji
NO-3	Aplikacija treba imati ugrađen alat za generiranje izvještaja o korištenju i prijavama.	Srednji
NO-4	Aplikacija treba biti jednostavna i intuitivna za korištenje.	Visok
NO-5	Sustav treba omogućiti jednostavno izvođenje nadogradnji bez prekida rada aplikacije.	Visok
NO-6	Aplikacija treba biti modularna, kako bi se lakše nadograđivala i prilagođavala novim zahtjevima ili integracijama s vanjskim sustavima.	Srednji
NO-7	Sustav treba imati automatsko logiranje grešaka i događaja, radi lakšeg otklanjanja problema.	Visok

### Zahtjevi skalabilnosti

Tablica 3 Tablica zahtjeva skalabilnosti

ID zahtjeva	Opis	Prioritet
NS-1	Sustav treba podržavati veliki broj istovremenih korisnika, posebno tijekom objave popularnih kvizova.	Visok
NS-2	Sustav treba podržavati minimalno 500 istovremenih korisnika bez smanjenja performansi.	Visok
NS-3	Sustav treba moći obraditi prijavu tima (odobriti/odbiti) u roku od nekoliko sekundi te poslati povratnu informaciju korisniku.	Srednji

ID zahtjeva	Opis	Prioritet
NS-4	Omogućiti asinkronu obradu složenijih operacija u pozadini (slanje obavijesti, generiranje statistike), čime se smanjuje vrijeme čekanja korisnika.	Visok
NS-5	Baza podataka treba podržavati horizontalnu skalabilnost, kako bi se smanjilo opterećenje na jednom serveru i omogućio paralelni rad više instanci baze.	Visok
NS-6	Sustav treba imati mehanizam automatskog balansiranja opterećenja među serverima.	Srednji
NS-7	Sustav treba omogućiti lako dodavanje novih servera ili instanci baze podataka radi rasta korisnika.	Visok

## Zahtjevi dostupnosti

Tablica 4 Tablica zahtjeva dostupnosti

ID zahtjeva	Opis	Prioritet
NPP-1	Sustav treba biti oblikovan tako da podržava jednostavno održavanje.	Visok
NPP-2	Aplikacija mora biti responzivna i usklađena s WCAG standardima pristupačnosti, kako bi bila dostupna svim korisnicima.	Visok
NPP-3	Implementirati strategije za visoku dostupnost: redundancija podataka i failover mehanizmi koji omogućuju nastavak rada u slučaju kvara.	Visok
NPP-4	Aplikacija treba imati ugrađeni centar za podršku i mogućnost slanja povratnih informacija putem e-maila.	Srednji

## Sigurnosni zahtjevi

Tablica 5 Tablica sigurnosnih zahtjeva

ID zahtjeva	Opis	Prioritet
NSG-1	Sustav mora implementirati višefaktorsku autentifikaciju za organizatore kvizova.	Visok
NSG-2	Osjetljivi podaci korisnika (lozinka, e-mail adrese) moraju biti šifrirani pri pohrani i prijenosu.	Visok
NSG-3	Sustav mora imati zaštitu od napada i drugih uobičajenih sigurnosnih prijetnji.	Srednji
NSG-4	Sustav mora omogućiti upravljanje pravima pristupa i razlikovanje uloga korisnika.	Srednji
NSG-5	Aplikacija treba imati ugrađeni centar za podršku i mogućnost slanja povratnih informacija putem e-maila.	Srednji

## Zahtjevi performansi

Tablica 6 Tablica zahtjeva performansi

ID zahtjeva	Opis	Prioritet
NP-1	Sustav treba omogućiti brzo učitavanje stranica kviza za sve korisnike.	Visok
NP-2	Aplikacija treba prikazati rezultate korisničkih akcija (prijava na kviz, pregled rezultata,...) u roku 1-2 sekunde.	Visok
NP-3	Aplikacija treba biti optimizirana kako bi učinkovito koristila resurse poslužitelja i smanjila opterećenje procesora i memorije, osiguravajući stabilan rad čak i uz veliko opterećenje.	Srednji
NP-4	Sustav treba automatski optimizirati slike, lokacije i medijske sadržaje kviza radi smanjenja vremena učitavanja.	Srednji
NP-5	Sustav treba podržavati generiranje izvještaja i statistike bez primjetnog usporavanja.	Srednji

## Zahtjevi ograničenja

Tablica 7 Tablica zahtjeva ograničenja

ID zahtjeva	Opis	Prioritet
NOG-1	Omogućiti prijavu samo timovima iz regije ili države u kojoj aplikacija.	Srednji
NOG-2	Postaviti ograničenje broja prijava koje jedan tim može podnijeti u kratkom vremenskom razdoblju (npr. jednu prijavu svakih 30 minuta).	Nizak

## Zahtjevi interoperabilnosti

Tablica 8 Tablica zahtjeva interoperabilnosti

ID zahtjeva	Opis	Prioritet
NKU-1	Sustav treba biti kompatibilan sa svim modernim preglednicima.	Visok
NKU-2	Sustav treba nuditi obavijesti putem e-maila i push notifikacija.	Srednji
NKU-3	Sustav treba omogućiti jednostavno pretraživanje kvizova po oznakama, lokaciji i datumu.	Srednji

## Dionici

1. sudionici/timovi (igrači) - korisnici koji sudjeluju u kvizovima, pregledavaju rezultate i primaju obavijesti
2. organizatori kvizova - korisnici koji kreiraju i uređuju kvizove, upravljaju prijavama i rezultatima te šalju obavijesti
3. administrator sustava - zadužen za održavanje sustava, upravljanje korisnicima i nadzor događaja
4. naručitelj - Vlado Sruk
5. razvojni tim – Wizionari

## Aktori

- A1: sudionici/timovi
  - registracija ili kreiranje tima - F-001.1
  - prijava u sustav - F-002.1
  - pretraživanje kvizova po datumu, lokaciji ili oznakama - F-005, NKU-3
  - prijavljivanje tima na kviz (odmah ili uz odobrenje organizatora) - F-006
  - pregled i praćenje statusa prijava - F-006.1
  - povlačenje prijava prije roka - F-006.2
  - ocjenjivanje kvizova i organizatora - F-008
  - primanje obavijesti o statusu prijave, odgodi ili otkazu kviza - NKU-2
  - pregled arhive održanih kvizova i osobne statistike - F-009, F-009.1
- A2: organizator kviza
  - registracija kao organizator - F-001.2
  - prijava u sustav kao organizator - F-002.2
  - uređivanje profila organizatora - F-003.3
  - kreiranje, uređivanje i brisanje kvizova - F-004, F-004.1, F-004.2
  - upravljanje prijavama tima i slanje obavijesti - F-004.3
  - unošenje rezultata i rang liste - F-007
  - odgovaranje na recenzije i davanje povratnih informacija - F-008.1
  - pristup arhivi održanih kvizova i rezultata - F-009.1
- A3: administrator sustava
  - upravljanje korisničkim računima organizatora i odobravanje novih organizatora - F-001.2, NSG-4
  - ažuriranje ili uklanjanje podataka o mjestima održavanja kvizova - F-004, NO-5
  - praćenje svih kreiranih događaja i aktivnosti korisnika - NO-7, NS-4
  - održavanje sigurnosti, dostupnosti i kompatibilnosti sustava - NPP-3, NSG-1, NSG-2
- A4: razvojni tim/naručitelj
  - nadziranje razvoja novih funkcionalnosti i modularnosti sustava - NO-6
  - optimiziranje performansi i responzivnosti - NP-1, NP-2, NP-3, NP-4

- implementacija skalabilnosti i asinkrone obrade zahtjeva - NS-5, NS-6, NS-7
- planiranje nove verzije sustava i integracije - F-004.1, NKU-1, NKU-2

### 3. Specifikacija zahtjeva sustava

#### Obrasci uporabe

##### UC1 - Registracija korisnika/tima

- Glavni sudionik: Sudionik/tim
- Cilj: Registracija novog korisnika ili tima u sustav
- Sudionici: Baza podataka
- Preduvjet: —
- Referenca: *F-001, F-001.1, F-001.2*

##### Opis osnovnog tijeka:

1. Sudionik/tim otvara obrazac za registraciju.
2. Unosi potrebne podatke (korisničko ime, e-mail, lozinka, naziv tima, članovi, kontakt podaci).
3. Odabire opciju "Registriraj me".
4. Sustav provjerava ispravnost i jedinstvenost podataka.
5. Sustav sprema podatke u bazu i šalje obavijest o uspješnoj registraciji putem e-maila.

##### Opis mogućih odstupanja:

- a. Korisnik je ispunio neki podatak nepravilno.
  - i. Sustav javlja grešku i označava nepravilno polje.
- b. E-mail ili korisničko ime već postoji.
  - i. Sustav javlja korisniku da su podaci zauzeti i nudi prijavu.

##### UC2 - Prijava u sustav

- Glavni sudionik: Sudionik/tim
- Cilj: Prijava korisnika u sustav
- Sudionici: Baza podataka
- Preduvjet: Korisnik mora biti registriran
- Referenca: *F-002, F-002.1, F-002.2*

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara obrazac za prijavu.
2. Unosi korisničko ime ili e-mail i lozinku.
3. Odabire opciju "Prijavi me".
4. Sustav provjerava ispravnost unesenih podataka.
5. Sustav učitava početnu stranicu sustava s prikazom korisničkog profila.

**Opis mogućih odstupanja**

- a. Podaci su pogrešno uneseni.
  - i. Sustav javlja pogrešku i označava problematično polje.
- b. Račun nije aktiviran.
  - i. Sustav obavještava korisnika da mora potvrditi e-mail.

**UC3 - Odjava iz sustava**

- Glavni sudionik: Sudionik/tim
- Cilj: Odjava prijavljenog korisnika iz sustava
- Sudionici: —
- Preduvjet: Korisnik mora biti prijavljen
- Referenca: —

**Opis osnovnog tijeka:**

1. Sudionik/tim odabire opciju "Odjava".
2. Sustav prekida korisničku sesiju.
3. Korisniku se učitava početna stranica aplikacije.
4. Sustav prikazuje poruku "Odjava uspješna".

**Opis mogućih odstupanja:**

- a. Tijekom odjave nastane tehnička pogreška.
  - i. Sustav obavještava korisnika da odjava nije dovršena.

**UC4 - Uređivanje profila**

- Glavni sudionik: Sudionik/tim, Organizator kviza, Administrator sustava
- Cilj: Ažurirati korisničke podatke profila
- Sudionici: Baza podataka
- Preduvjet: Korisnik mora biti prijavljen
- Referenca: F-003, F-003.1, F-003.2, F-003.3

**Opis osnovnog tijeka:**



1. Korisnik otvara svoj profil.
2. Odabire opciju "Uredi profil".
3. Ažurira željene podatke (npr. korisničko ime, kontakt informacije, opis, lokaciju).
4. Odabire "Spremi promjene".
5. Sustav sprema promjene u bazu i prikazuje poruku o uspješnom ažuriranju.

**Opis mogućih odstupanja:**

- a. Uneseni podaci nisu valjani.
  - i. Sustav označava pogrešna polja i traži ispravak.
- b. Došlo je do greške pri spremanju.
  - i. Sustav prikazuje poruku o neuspjelom ažuriranju profila.

## UC5 - Brisanje korisničkog računa

- Glavni sudionik: Sudionik/tim
- Cilj: Brisanje vlastitog korisničkog računa iz sustava
- Sudionici: Baza podataka
- Preduvjet: Korisnik mora biti prijavljen
- Referenca: —

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara postavke računa.
2. Odabire opciju "Obriši račun".
3. Sustav traži potvrdu brisanja.
4. Nakon potvrde, sustav briše korisničke podatke iz baze i odjavljuje korisnika.
5. Korisnik prima potvrdu o brisanju računa putem e-maila.

**Opis mogućih odstupanja:**

- a. Korisnik odustane od brisanja.
  - i. Sustav prekida proces i vraća korisnika na profil.
- b. Sustav ne može obrisati račun zbog tehničke greške.
  - i. Sustav obavještava korisnika o neuspjelom brisanju.

## UC6 - Pretraživanje kvizova

- Glavni sudionik: Sudionik/tim
- Cilj: Pretraživanje kvizova prema kriterijima
- Sudionici: Baza podataka
- Preduvjet: —
- Referenca: F-005, NKU-3

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara stranicu s kvizovima.
2. Unosi kriterije pretrage (naziv, lokacija, datum).
3. Sustav prikazuje popis kvizova koji odgovaraju upitu.
4. Korisnik može otvoriti detalje pojedinog kviza.

**Opis mogućih odstupanja:**

- a. Nema kvizova koji odgovaraju kriterijima.
  - i. Sustav prikazuje poruku "Nema pronađenih rezultata".

### UC7 - Prijava tima na kviz

- Glavni sudionik: Sudionik/tim
- Cilj: Prijava tima na kviz
- Sudionici: Baza podataka
- Preduvjet: Korisnik mora biti prijavljen
- Referenca: *F-006, F-006.1*

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara stranicu kviza.
2. Odabire opciju "Prijavi tim".
3. Sustav provjerava dostupnost mjesta i status prijava.
4. Sustav sprema prijavu i prikazuje poruku o uspjehu.

**Opis mogućih odstupanja:**

- a. Kviz je popunjen.
  - i. Sustav onemogućuje prijavu i prikazuje poruku.
- b. Kviz je otkazan.
  - i. Sustav prikazuje obavijest i ne dopušta prijavu.

### UC8 - Povlačenje prijave

- Glavni sudionik: Sudionik/tim
- Cilj: Povlačenje prethodno poslane prijave na kviz
- Sudionici: Baza podataka
- Preduvjet: Korisnik mora imati aktivnu prijavu
- Referenca: *F-006.2*

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara listu prijava.

2. Odabire kviz i opciju "Povuci prijavu".
3. Sustav traži potvrdu radnje.
4. Sustav ažurira status prijave na "Povučen".
5. Sustav prikazuje poruku o uspješnom povlačenju.

**Opis mogućih odstupanja:**

- a. Rok za povlačenje je istekao.
  - i. Sustav onemogućuje povlačenje i prikazuje upozorenje.

## UC9 - Pregled statusa prijave

- Glavni sudionik: Sudionik/tim
- Cilj: Pregled statusa prijavljenih kvizova
- Sudionici: Baza podataka
- Preduvjet: Korisnik mora biti prijavljen
- Referenca: *F-006.1*

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara stranicu "Moje prijave".
2. Sustav prikazuje popis svih prijavljenih kvizova.
3. Statusi prijave (potvrđeno, odbijeno, na čekanju) su vidljivi korisniku.

**Opis mogućih odstupanja:**

- a. Korisnik nema prijavu.
  - i. Sustav prikazuje poruku "Još niste prijavljeni ni na jedan kviz."

## UC10 - Pregled rezultata i arhive

- Glavni sudionik: Sudionik/tim
- Cilj: Pregled završenih kvizova i rezultata
- Sudionici: Baza podataka
- Preduvjet: Rezultati moraju biti uneseni
- Referenca: *F-007, F-009*

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara sekciju "Rezultati".
2. Odabire kviz iz arhive.
3. Sustav prikazuje rang listu i rezultate.

**Opis mogućih odstupanja:**

a. Još nema rezultata za odabrani kviz.

*i.* Sustav prikazuje poruku "Rezultati još nisu objavljeni."

## UC11 - Ocjenjivanje kvizova i organizatora

- Glavni sudionik: Sudionik/tim
- Cilj: Ostaviti ocjenu i recenziju kviza ili organizatora
- Sudionici: Baza podataka
- Preduvjet: Korisnik mora biti sudjelovao na kvizu
- Referenca: *F-008, F-008.1*

**Opis osnovnog tijeka:**

1. Sudionik/tim otvara kviz u kojem je sudjelovao.
2. Odabire opciju "Ocijeni kviz" ili "Ocijeni organizatora".
3. Unosi ocjenu i opcionalni komentar.
4. Sustav sprema ocjenu i ažurira prosjek ocjena.

**Opis mogućih odstupanja:**

a. Korisnik nije sudjelovao na kvizu.

*i.* Sustav ne dopušta unos ocjene.

b. Recenzija sadrži neprimjeren sadržaj.

*i.* Sustav odbija recenziju i prikazuje poruku o kršenju pravila.

## UC12 - Kreiranje kviza

- Glavni sudionik: Organizator kviza
- Cilj: Kreirati i objaviti novi kviz
- Sudionici: Baza podataka
- Preduvjet: Organizator mora biti prijavljen
- Referenca: *F-004, F-004.1*

**Opis osnovnog tijeka:**

1. Organizator otvara obrazac za novi kviz.
2. Unosi podatke o kvizu (naziv, opis, datum, vrijeme, lokaciju, broj timova).
3. Odabire opciju "Objavi".
4. Sustav sprema podatke i prikazuje poruku o uspješnom kreiranju kviza.

**Opis mogućih odstupanja:**

a. Nisu uneseni svi potrebni podaci.

*i.* Sustav označava obavezna polja i traži ispravak.

### UC13 - Uređivanje/brisanje kviza

- Glavni sudionik: Organizator kviza
- Cilj: Uređivanje ili brisanje postojećeg kviza
- Sudionici: Baza podataka
- Preduvjet: Organizator mora biti vlasnik kviza
- Referenca: *F-004.2*

#### Opis osnovnog tijeka:

1. Organizator otvara stranicu svog kviza.
2. Odabire opciju "Uredi" ili "Obriši kviz".
3. Sustav provjerava prava pristupa.
4. Sustav sprema izmjene ili briše kviz i prikazuje potvrdu.

#### Opis mogućih odstupanja:

- a. Kviz ima aktivne prijave.
  - i. Sustav onemogućuje brisanje i nudi opciju "Odgoditi kviz".

### UC14 - Upravljanje prijavama

- Glavni sudionik: Organizator kviza
- Cilj: Upravljanje prijavama timova na kviz
- Sudionici: Baza podataka
- Preduvjet: Organizator mora imati objavljen kviz
- Referenca: *F-004.3*

#### Opis osnovnog tijeka:

1. Organizator otvara listu prijava za svoj kviz.
2. Pregledava pristigle prijave i podatke o timovima.
3. Odobrava ili odbija prijave.
4. Sustav ažurira statute i šalje obavijesti prijavljenima.

#### Opis mogućih odstupanja:

- a. Broj prijava premašuje maksimalni limit.
  - i. Sustav automatski zatvara prijave i prikazuje obavijest.

## UC15 - Unos rezultata i rang liste

- Glavni sudionik: Organizator kviza
- Cilj: Unos rezultata kviza i prikaz rang liste
- Sudionici: Baza podataka
- Preduvjet: Kviz mora biti završen
- Referenca: *F-007*

### Opis osnovnog tijeka:

1. Organizator otvara završeni kviz.
2. Unosi rezultate i bodove po timovima.
3. Sustav automatski generira rang listu.
4. Rezultati postaju vidljivi sudionicima.

### Opis mogućih odstupanja:

a. Nisu uneseni svi rezultati.

*i.* Sustav upozorava korisnika i omogućava djelomično spremanje.

## UC16 - Odobravanje organizatora i upravljanje korisnicima

- Glavni sudionik: Administrator sustava
- Cilj: Upravljanje korisnicima i odobriti nove organizatore
- Sudionici: Baza podataka
- Preduvjet: Administrator mora biti prijavljen
- Referenca: *NSG-4*

### Opis osnovnog tijeka:

1. Administrator otvara konzolu za upravljanje korisnicima.
2. Pregledava zahtjeve za ulogu organizatora.
3. Odobrava ili odbija zahtjeve.
4. Može uređivati ili deaktivirati korisnike.
5. Sustav sprema promjene i šalje obavijesti korisnicima.

### Opis mogućih odstupanja:

a. Administrator nema potrebne ovlasti.

*i.* Sustav blokira akciju i prikazuje upozorenje.

## UC17 - Slanje obavijesti

- Glavni sudionik: Sustav (automatizirano)
- Cilj: Obavješćavanje korisnika o promjenama ili događajima

- Sudionici: Servis za e-mail i push notifikacije
- Preduvjet: Postoji događaj koji zahtijeva obavijest
- Referenca: *NKU-2*

**Opis osnovnog tijeka:**

1. Sustav prepoznaje događaj (promjena termina, nova prijava, rezultat).
2. Generira sadržaj obavijesti.
3. Šalje obavijest korisnicima putem e-maila ili push notifikacije.
4. Bilježi status isporuke.

**Opis mogućih odstupanja:**

- a. Servis za slanje obavijesti nije dostupan.
  - i. Sustav sprema obavijest u red čekanja i pokušava ponovno kasnije.

## UC18 - Brisanje recenzija

- Glavni sudionik: Administrator sustava
- Cilj: Ukloniti neprimjerene recenzije ili komentare
- Sudionici: Baza podataka
- Preduvjet: Administrator mora biti prijavljen
- Referenca: *F-008.1*

**Opis osnovnog tijeka:**

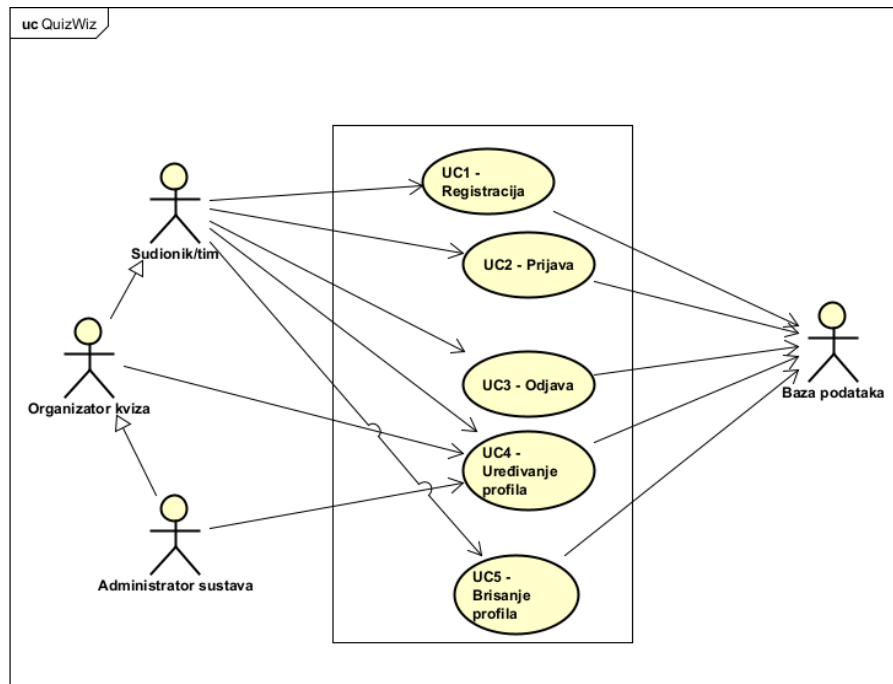
1. Administrator otvara sekciju recenzija.
2. Pregledava sve prijavljene ili označene recenzije.
3. Odabire opciju "Obriši recenziju".
4. Sustav uklanja recenziju iz baze i obavještava korisnika o brisanju.

**Opis mogućih odstupanja:**

- a. Recenzija je već uklonjena.
  - i. Sustav prikazuje poruku "Recenzija ne postoji."
- b. Administrator nema dovoljne ovlasti.
  - i. Sustav onemogućuje brisanje i prikazuje upozorenje.

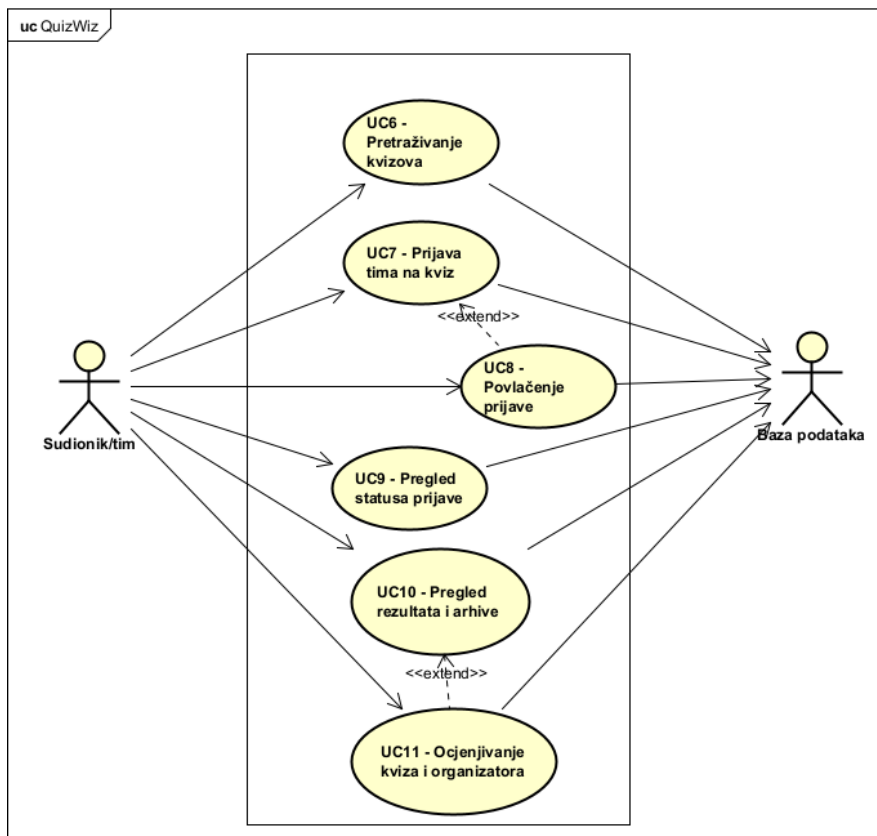
# Dijagrami obrazaca uporabe

## 1. UC dijagram za korisničke račune



Slika 2 UC dijagram za korisničke račune

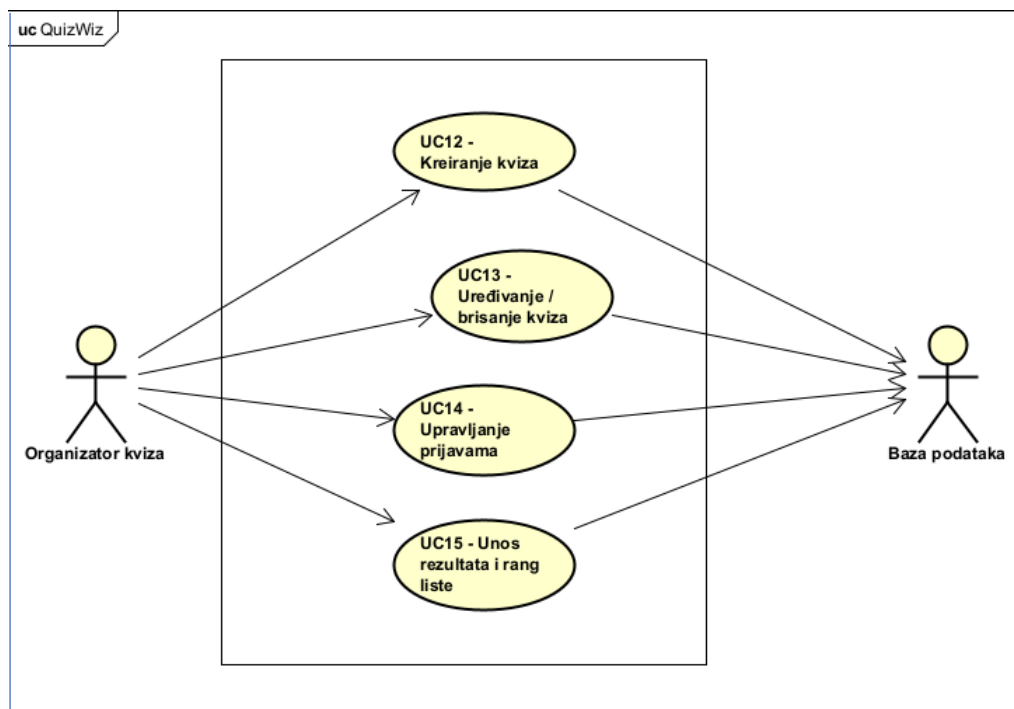
## 2. UC dijagram funkcionalnosti sudionika/tima



Slika 3 UC dijagram funkcionalnosti sudionika/tima

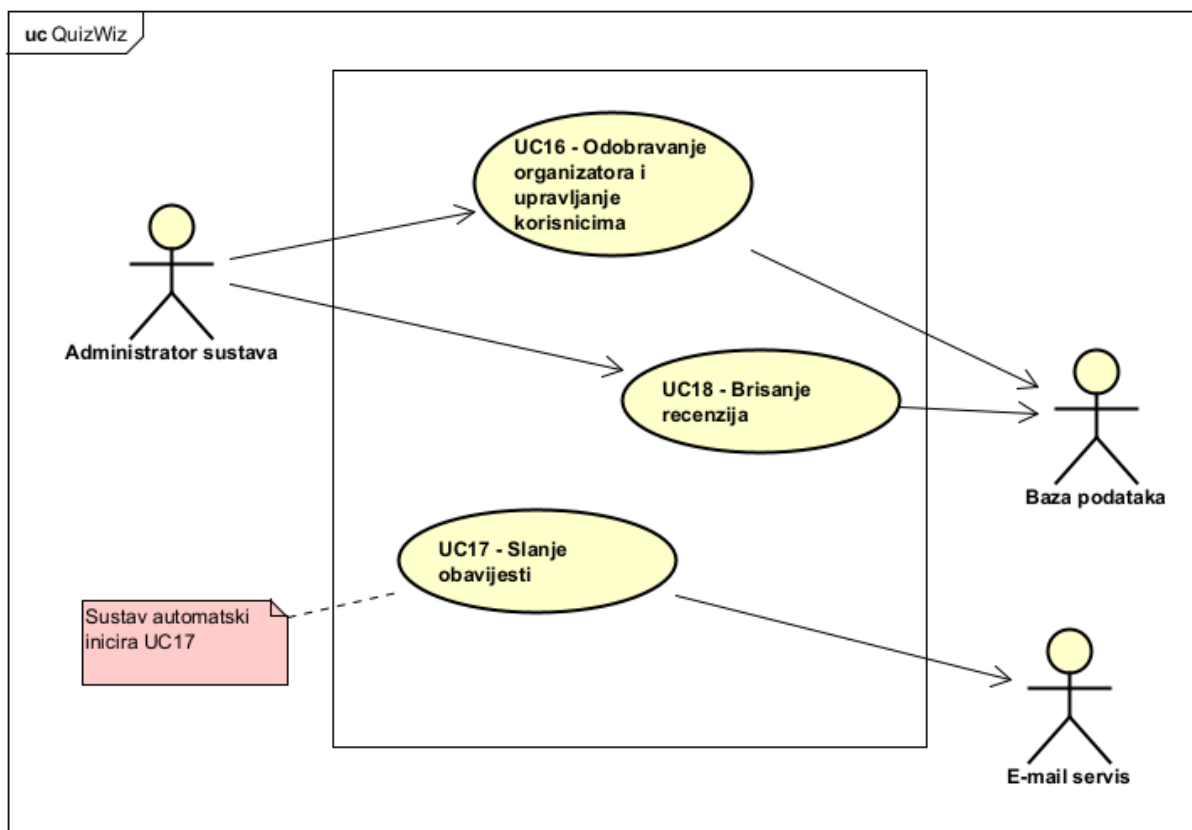


### 3. UC dijagram funkcionalnosti organizatora kviza



Slika 4 UC dijagram funkcionalnosti organizatora kviza

### 4. UC dijagram funkcionalnosti administratora i sustava



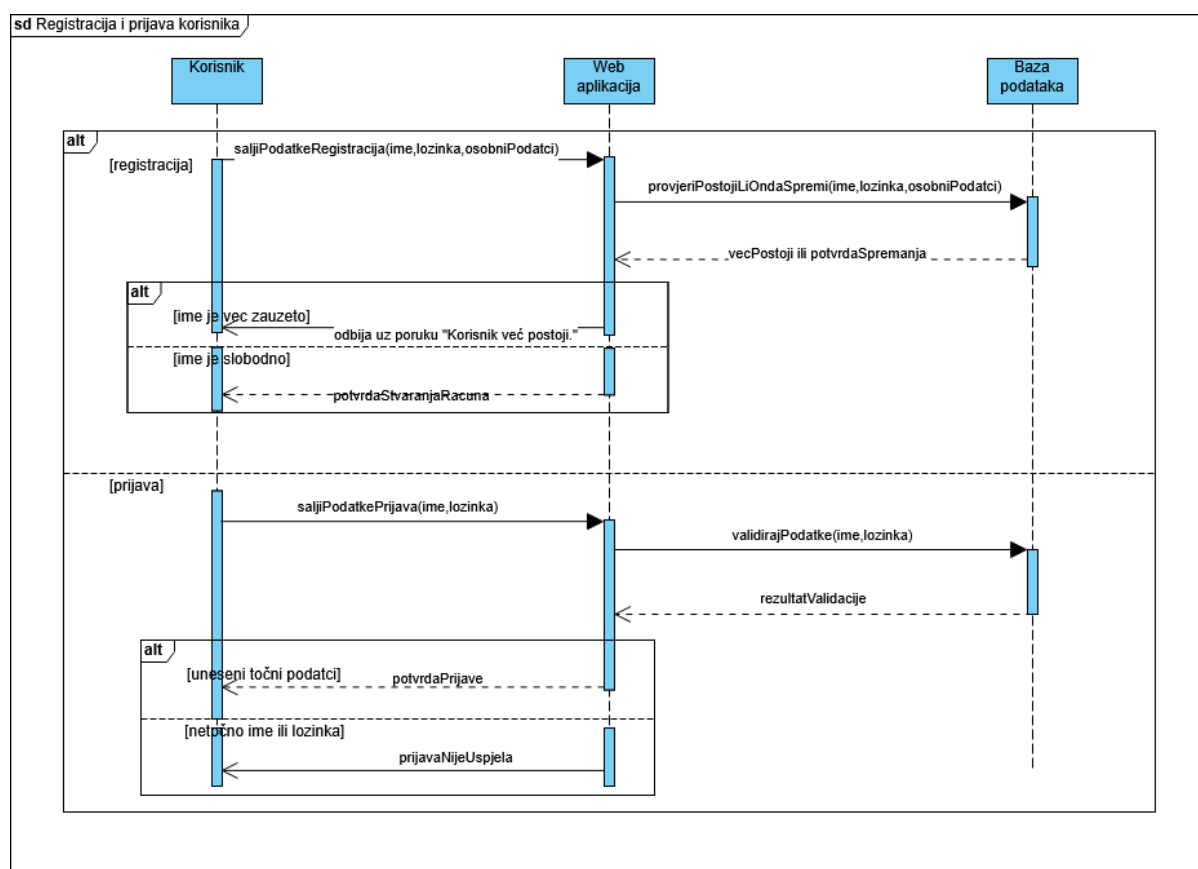
Slika 5 UC dijagram funkcionalnosti administratora i sustava

## Sekvencijski dijagrami

### UC1, UC2 Neprijavljeni korisnik - registracija i prijava korisnika u sustav

Ako je odabrana registracija, neprijavljeni korisnik tada unosi željeno ime, lozinku i odgovarajuće osobne podatke. Aplikacija proslijeđuje podatke prema bazi podataka, gdje se provjerava postoji li već korisnik s istim imenom. Ukoliko uneseno ime ne postoji, u bazu podataka spremaju se podatci o korisniku, a korisniku se javlja da je registracija uspješna. Ukoliko već postoji ime u bazi podataka, javlja se korisniku da registracija nije uspješna.

Ako je odabrana prijava, neprijavljeni korisnik unosi svoje ime i lozinku, koji se proslijeđuju i validiraju. U slučaju da lozinka odgovara korisničkom računu, javlja se korisniku da je prijava uspješna, a u suprotnom da je uneseno krivo ime ili lozinka te da je prijava neuspješna.



Slika 6 Sekvencijski dijagram - registracija i prijava korisnika u sustav

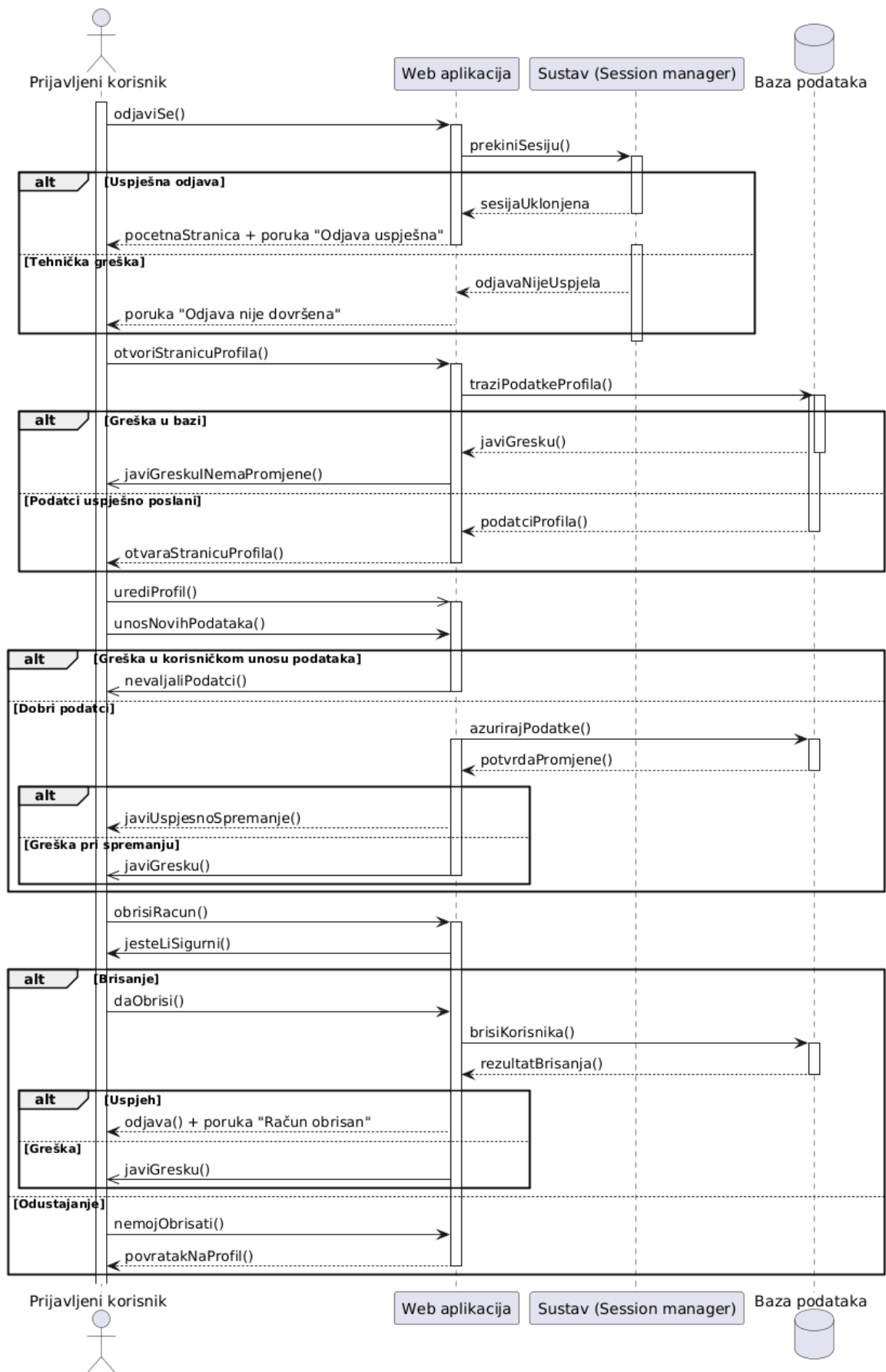
### UC3, UC4, UC4 Prijavljeni korisnik - odjava, uređivanje i brisanje računa

Dijagram prikazuje osnovne radnje prijavljenog korisnika: odjavu, uređivanje i brisanje računa.

Korisnik se može odjaviti iz sustava, pri čemu aplikacija prekida sesiju i vraća poruku o uspjehu ili grešci. Na profilnoj stranici aplikacija dohvaća podatke iz baze; ako je došlo do pogreške, prikazuje se poruka o neuspjehu, a ako je uspješno, otvara se profil.

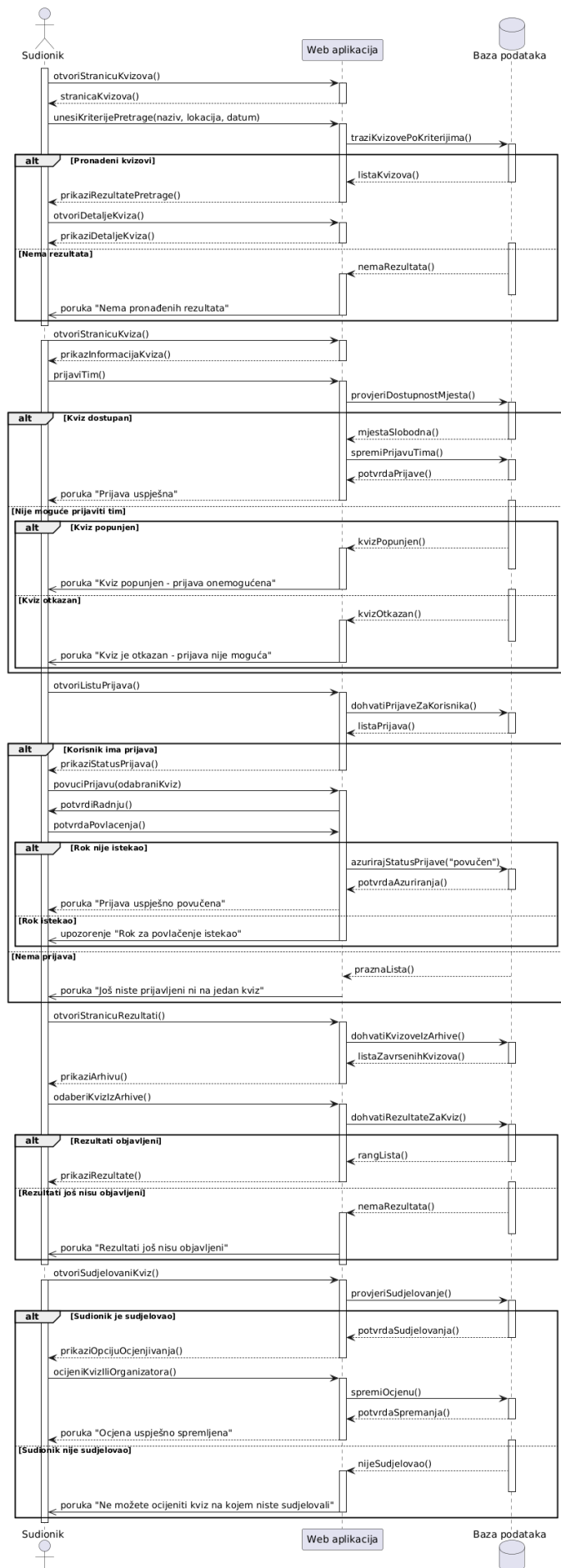
Pri uređivanju profila korisnik unosi nove podatke. Ako su podaci neispravni, sustav javlja pogrešku, a ako su ispravni, ažurira ih u bazi i prikazuje poruku o uspješnom spremanju.

Korisnik može i obrisati račun. Sustav traži potvrdu, a ako korisnik pristane, briše račun iz baze i odjavljuje ga uz poruku o uspjehu; u suprotnom, vraća ga na profil bez promjena.



Slika 7 Sekvencijski dijagram - odjava, uređivanje i brisanje računa

## UC6, UC7, UC8, UC9, UC10, UC11 Sudionik - pretraživanje kvizeva, prijava na kviz, upravljanje svojim prijavama, pregled arhive završenih kvizeva, ocjenjivanje kviza/organizatora



Dijagram prikazuje procese koje korisnik, primarno sudionik, može izvršavati putem web aplikacije vezane uz pretragu i prijavu na kviz, upravljanje vlastitim prijavama, te pregled rezultata završenih kvizeva i ocjenjivanje.

Slika 8 Sekvencijski dijagram - pretraživanje kvizeva, prijava na kviz, upravljanje svojim prijavama, pregled arhive završenih kvizeva, ocjenjivanje kviza/organizatora

## UC12, UC13, UC14, UC15, UC17 Organizator - upravljanje kvizovima i prijavama, automatsko slanje obavijesti

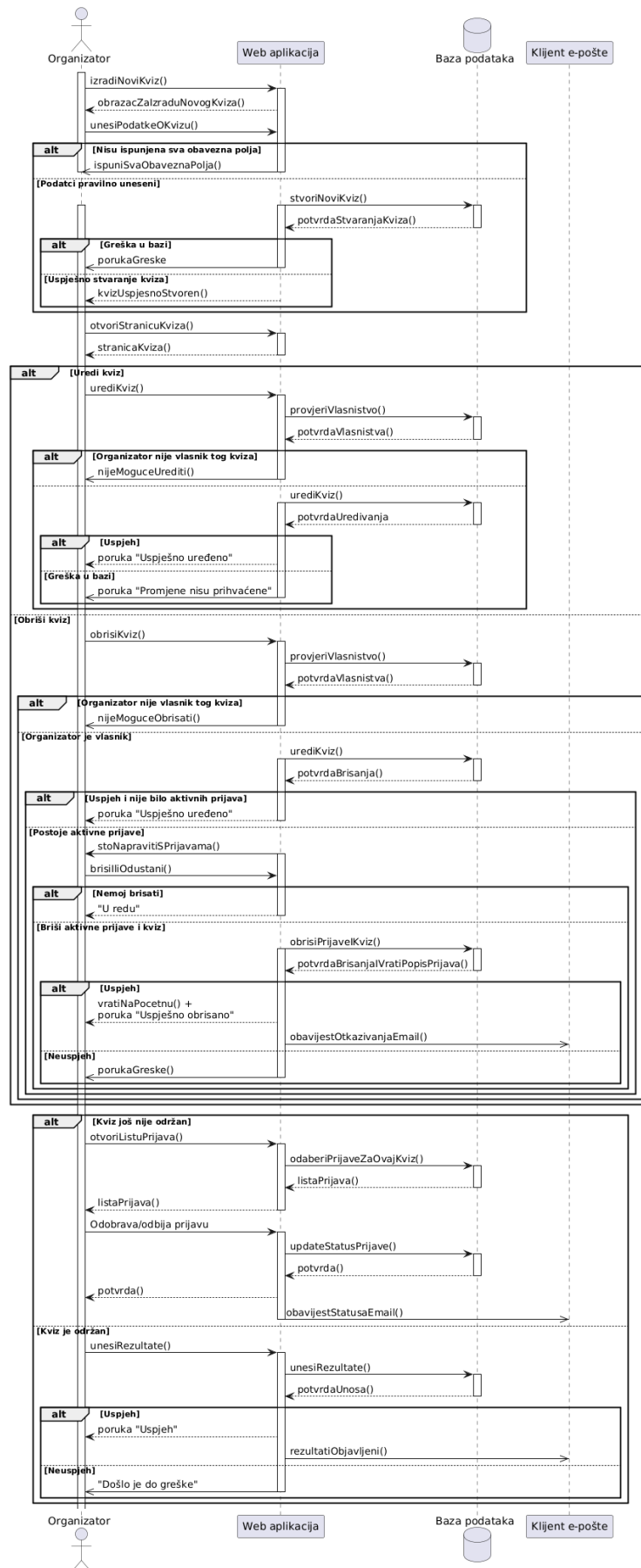
Dijagram prikazuje procese koje organizator može izvršavati putem web aplikacije u vezi s upravljanjem kvizovima. Organizator ima mogućnost stvaranja novog kviza, uređivanja i brisanja postojećih kvizova koje je sam kreirao, kao i upravljanja prijavama te unos rezultata nakon održanog kviza.

Proces započinje kada organizator pošalje zahtjev za izradu novog kviza. Aplikacija vraća obrazac za unos podataka, nakon čega organizator unosi potrebne informacije o kvizu. Ako nisu popunjena sva obavezna polja, sustav obavještava organizatora da ih mora ispuniti. U slučaju da su svi podaci ispravno uneseni, aplikacija šalje zahtjev bazi podataka za stvaranje novog kviza. Baza vraća potvrdu o uspješnom stvaranju kviza ili poruku o pogrešci ako dođe do problema.

Nakon stvaranja kviza, organizator može otvoriti stranicu kviza te odabrati želi li ga urediti ili obrisati. Ako odabere uređivanje, aplikacija prvo provjerava u bazi podataka je li organizator stvarni vlasnik kviza. Ako nije, sustav javlja da uređivanje nije moguće. Ako jest, baza se ažurira s novim podacima i, ovisno o ishodu, organizator dobiva poruku o uspješnom uređivanju ili o pogrešci.

Ako organizator odluči obrisati kviz, također se provjerava vlasništvo. Ako je potvrđeno da je on vlasnik, sustav provjerava postoje li aktivne prijave. Ako ih nema, kviz se briše i prikazuje poruka o uspješnom brisanju. Ako postoje aktivne prijave, sustav traži od organizatora odluku želi li odustati ili obrisati i prijave i kviz. Ako odabere brisanje, baza podataka briše sve povezane prijave i kviz, vraća potvrdu te sustav šalje obavijest putem e-pošte prijavljenim korisnicima o otkazivanju kviza.

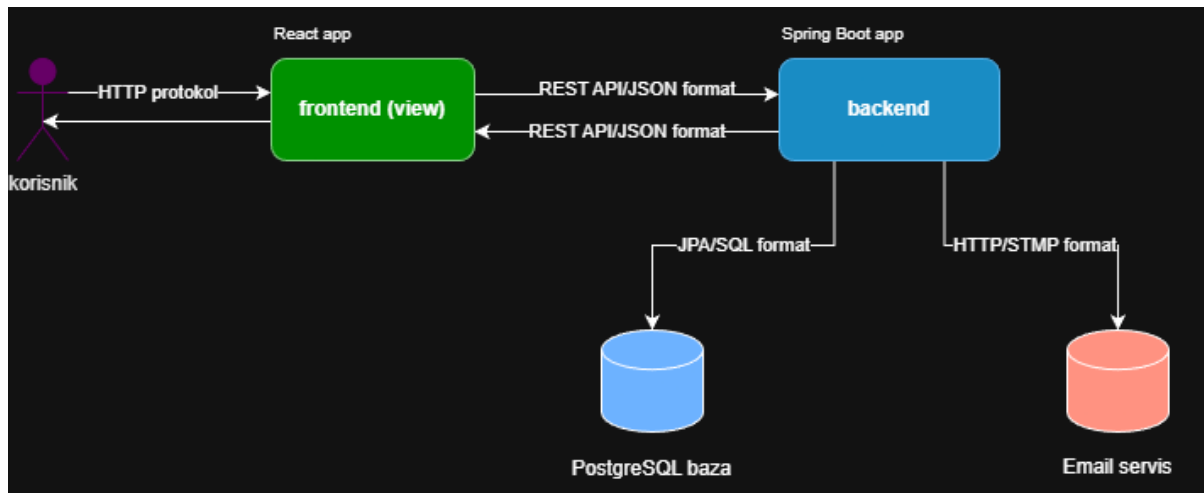
U dijelu koji se odnosi na prijave i rezultate, razlikuju se dvije situacije. Ako kviz još nije održan, organizator može pregledati listu prijava, odobriti ili odbiti sudionike te nakon potvrde izmjena sustav šalje e-mail obavijesti prijavljenima o statusu njihove prijave. Ako je kviz već održan, organizator unosi rezultate. Nakon što baza potvrdi unos, sustav prikazuje poruku o uspješnom unosu i šalje e-mail obavijest sudionicima o objavljenim rezultatima. U slučaju greške, organizator prima odgovarajuću poruku o neuspjelom unosu.



Slika 9 Sekvencijski dijagram - upravljanje kvizovima i prijavama, automatsko slanje obavijesti

## 4. Arhitektura i dizajn sustava

### Arhitektura sustava



Slika 10 Prikaz arhitekture sustava

### 4.1 Opis arhitekture

#### 4.1.1 Pregled arhitekture na visokoj razini

Sustav QuizWiz temelji se na višeslojnoj web arhitekturi klijent-poslužiteljskog tipa, implementiranoj kroz modularno organizirani sustav s jasnim odvajanjima slojeva (frontend, backend, baza podataka). Ovaj pristup omogućuje jednostavniju implementaciju i održavanje tijekom početne faze razvoja, uz mogućnost kasnije transformacije u arhitekturu mikroserversa ako se pojavi potreba za većom skalabilnošću.

#### 4.1.2 Stil arhitekture

Odabrani arhitektonski stil je klijent-poslužitelj s MVC (Model-View-Controller) obrascem unutar backend sustava. Glavni razlozi za odabir ovog pristupa su:

- jasno odvajanje odgovornosti između slojeva: korisničko sučelje, poslovna logika i podaci
- jednostavna održivost i testiranje - promjene u logici ili prikazu mogu se provoditi neovisno
- skalabilnost - backend se može proširiti u više servisa u budućnosti
- kompatibilnost s web tehnologijama - arhitektura prirodno podržava REST API-je za komunikaciju s frontend slojem

Ovaj stil ima visoku koheziju i nisku povezanost, čime se postiže veća fleksibilnost, ponovna upotrebljivost i stabilnost sustava.

### 4.1.3 Podsustavi i njihove odgovornosti

Tablica 9 Tablica podsustava i njihovih odgovornosti

podstav	opis i odgovornosti
autentifikacijski podstav	Upravlja registracijom, prijavom i autorizacijom korisnika. Koristi JWT (JSON Web Token) za autentifikaciju zahtjeva.
podstav upravljanja kvizovima	Omogućuje kreiranje, uređivanje, otkazivanje i pregled kvizova, uključujući unos informacija o datumu, mjestu, kapacitetu i oznakama (vrsta kviza).
podstav prijava timova	Omogućuje prijave timova na kvizove, odobravanje prijave od strane organizatora i automatsku kontrolu popunjenosti kapaciteta.
podstav notifikacija	Automatski šalje e-mail obavijesti o odobrenju, otkazivanju ili promjeni termina kviza.
administratorski podstav	Omogućuje administratoru pregled i upravljanje organizatorima, kvizovima i lokacijama.

### 4.1.4 Preslikavanje na radnu platformu

Sustav QuizWiz bit će implementiran na hibridnoj radnoj platformi, kombinirajući cloud infrastrukturu i mogućnost lokalnog razvoja. Odabrana platforma omogućuje fleksibilnost, skalabilnost i jednostavno održavanje sustava.

Predviđena implementacija:

- frontend (React) - frontend aplikacija bit će deploy-ana na cloud platformu, što omogućuje dostupnost aplikacije korisnicima putem interneta
  - React aplikacija se razvija lokalno gdje se također provjerava rad aplikacije prije deploja u cloud (Vercel)
- backend (Spring Boot) - backend aplikacija će biti deploy-ana na cloud server (Railway)
  - cloud deployment osigurava visoku dostupnost i otpornost na greške
- baza podataka (PostgreSQL) - PostgreSQL baza će biti hostana u cloudu (Supabase)
  - cloud hostanje omogućuje automatsko skaliranje, sigurnosne kopije i monitoring performansi

Prednosti hibridnog pristupa:



- fleksibilnost - razvojni tim može raditi lokalno, dok radna verzija radi u cloud okruženju
- pouzdanost - cloud hostanje ima automatizirane backup-ove i monitoring, što smanjuje rizik od gubitka podataka i prekida rada sustava
- razvoj i testiranje - lokalno okruženje omogućuje brze iteracije i debugging, a radna verzija za to vrijeme ostaje stabilna

#### *4.1.5 Spremišta podataka*

Sustav koristi relacijsku bazu podataka - PostgreSQL, zbog:

- pogodnosti za strukturirane podatke poput korisnika, kvizova, timova i rezultata
- podrške za složene odnose
- integriteta podataka putem stranih ključeva
- mogućnosti skaliranja i sigurnosti u cloud okruženju

#### *4.1.6 Mrežni protokoli*

- HTTP/HTTPS - glavni protokol za komunikaciju između klijenta i poslužitelja putem REST API poziva
- WebSocket (u kasnijim fazama) - za real-time notifikacije o promjenama kvizova i statusima prijava
- SMTP - za slanje e-mail obavijesti putem vanjskih servera

#### *4.1.7 Globalni upravljački tok*

1. Korisnik (organizator ili igrač) pristupa web aplikaciji putem preglednika (frontend).
2. Frontend šalje zahtjev REST API-ju (backend) putem HTTP/HTTPS protokola.
3. Backend obrađuje zahtjev odgovarajućim kontrolerom i komunicira s bazom podataka.
4. Nakon obrade, backend vraća odgovor u JSON formatu.
5. Frontend prikazuje rezultat korisniku i, prema potrebi, pokreće notifikacijski podsustav.
6. U slučaju izmjena (npr. otkazivanja kviza), backend automatski šalje obavijesti putem e-maila svim prijavljenim korisnicima.

### 4.1.8 Sklopovskoprogramski zahtjevi

Tablica 10 Tablica sklopovskoprogramskih zahtjeva

komponenta	minimalni zahtjev
operativni sustav (OS)	Linux, Windows ili macOS
procesor	dvojezgreni CPU (2.0GHz+)
RAM memorija	4GB (razvojno), 8GB (poslužiteljsko okruženje)
preglednik	Chrome, Edge, Firefox, Brave, Safari...
okruženja	Node.js (v18+), PostgreSQL (v15+)

### 4.1.9 Upravljanje autentifikacijom i sesijama putem JWT-a

U svrhu sigurne autentifikacije korisnika i zaštite API poziva, sustav koristi **JWT (JSON Web Token)** mehanizam za upravljanje sesijama. Ovaj pristup omogućuje **stateless autentifikaciju**, eliminirajući potrebu za pohranom sesijskih podataka na strani poslužitelja, što pridonosi skalabilnosti i sigurnosti sustava.

#### Mehanizam rada:

- Prilikom prijave, poslužitelj (Spring Boot backend) generira JWT koji uključuje identitet korisnika, ulogu i vrijeme isteka.
- Token se digitalno potpisuje tajnim ključem te vraća klijentskoj aplikaciji (frontend – React).
- Frontend pohranjuje token (npr. u localStorage) te ga šalje u svakom sljedećem zahtjevu prema backendu u HTTP zaglavlju:
- Backend validira token prije obrade svakog zahtjeva i, u slučaju valjanosti, omogućuje pristup zaštićenim resursima.

#### Prednosti:

- **Skalabilnost** – stateless pristup omogućuje jednostavno horizontalno skaliranje backend aplikacije.
- **Sigurnost** – digitalno potpisani tokeni štite od manipulacije i smanjuju napade poput CSRF-a.
- **Jednostavno upravljanje sesijama** – frontend jasno razlikuje autentificirano stanje bez potrebe za server-side sesijama.

### Sigurnosne napomene:

- Tokeni imaju definirani rok trajanja (exp claim) i automatski istječu nakon određenog vremena.
- Planira se podrška za **refresh token** mehanizam radi produljenja trajanja sesije bez ponovne prijave.
- Komunikacija između frontend-a i backend-a odvija se isključivo putem **HTTPS protokola**.

JWT pristup u potpunosti je usklađen s višeslojnom web arhitekturom i pridonosi boljoj modularnosti, sigurnosti i neovisnom razvoju frontend i backend komponenti.

## 4.2 Obrazloženje odabira arhitekture

### 4.2.1 Izbor arhitekture temeljen na principima oblikovanja

Glavni principi korišteni pri dizajnu:

- visoka kohezija - komponente imaju jasno definirane funkcije
- niska povezanost - frontend i backend su neovisni i komuniciraju isključivo putem REST API-ja
- fleksibilnost - modularna struktura koda omogućuje dodavanje novih funkcionalnosti bez većih promjena
- sigurnost - autentifikacija i autorizacija putem JWT tokena te HTTPS komunikacija
- skalabilnost - mogućnost prelaska s monolitne na mikrouslužnu arhitekturu u budućnosti

### 4.2.2 Razmatrane alternative

Mikrouslužna arhitektura je bila razmatrana, ali je odbačena zbog nepotrebne složenosti u početnoj fazi projekta. NoSQL baza je razmotrena, ali je odbačena jer aplikacija koristi čvrste relacijske veze i zahtijeva integritet podataka. Serverless pristup je također razmatran, no trenutno nije potreban zbog malog broja istovremenih korisnika.

## 4.3 Organizacija sustava na visokoj razini

- Klijent-poslužitelj: Frontend (React) i backend (Spring Boot) komuniciraju putem REST API-ja. Klijent se spaja na backend preko HTTPS protokola i dohvaća podatke u JSON formatu.
- Baza podataka: Relacijska baza PostgreSQL pohranjuje sve korisničke, kvizne i prijavne podatke, uz definirane odnose i indeksiranje za brze pretrage.
- Datotečni sustav: Korisnički avatari i slike kvizova pohranjuju se u cloud spremište.

- Grafičko sučelje: Frontend koristi React za responsivno, intuitivno web sučelje. Komunikacija s backendom odvija se putem REST API poziva.

## 4.4 Organizacija aplikacije

### 4.4.1 Frontend i Backend slojevi

Sustav QuizWiz organiziran je u dva glavna dijela: frontend i backend, koji zajedno čine višeslojnu web arhitekturu prema MVC obrascu. Komunikacija između slojeva odvija se putem REST API-ja, a podaci se razmjenjuju u JSON formatu.

Frontend, implementiran u React.js, čini prezentacijski (view) sloj sustava. On omogućuje korisnicima interakciju s aplikacijom te vizualni prikaz podataka dohvaćenih s backend-a.

Glavne značajke:

- Struktura se temelji na modularnim i višekratno upotrebljivim komponentama (QuizList, LoginForm).
- Komunikacija s backendom odvija se putem HTTP zahtjeva (GET, POST...).
- Obrada i prikaz odgovora je u JSON formatu.
- Upravljanje stanjem i navigacijom obavlja se pomoću React hookova i routera.

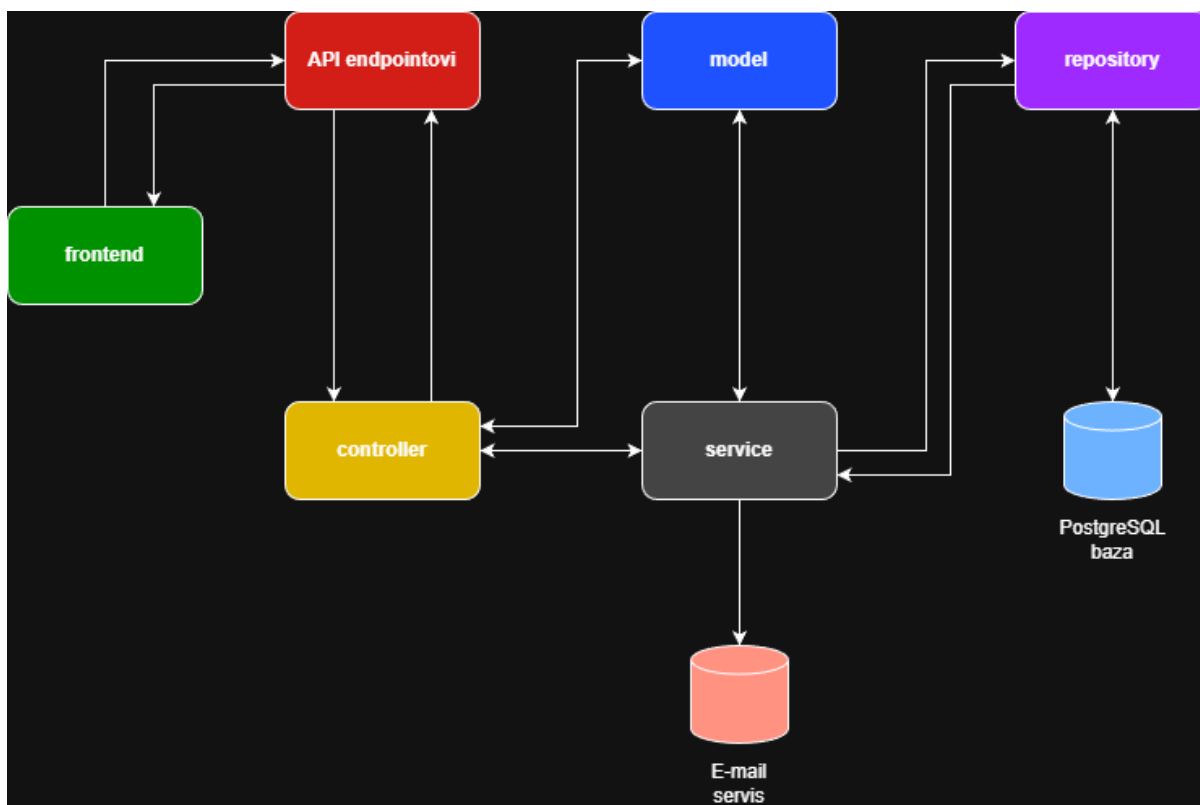
Backend, implementiran u Spring Boot frameworku, čini poslužiteljski (server) dio sustava i implementira MVC (Model-View-Controller) arhitekturu. Njegova glavna zadaća je obrada zahtjeva primljenih od frontenda, izvršavanje poslovne logike, pristup bazi podataka te vraćanje odgovora u JSON formatu.

### 4.4.2 MVC arhitektura

Organizacija slojeva unutar backend-a:

Tablica 11 Tablica organizacije slojeva unutar backend-a

sloj	Opis i odgovornosti
Routes/API endpointovi	Definiraju REST API putanje koje primaju zahtjeve s frontenda.
Controller	Prima i obrađuje HTTP zahtjeve, poziva odgovarajuće servise te vraća JSON odgovore. Odgovoran je za povezivanje frontend-a s poslovnom logikom.
Service	Implementira poslovnu logiku - autentifikaciju i autorizaciju korisnika, slanje e-mail notifikacija, upravljanje kvizovima, timovima i korisnicima.
Repository	Komunicira s bazom podataka koristeći JPA.
Model	Predstavlja podatkovni sloj, odnosno Java entitete koji preslikavaju tablice u PostgreSQL bazi. Svaki entitet definira strukturu podataka.



Slika 11 Prikaz slojeva

## Baza podataka

### 4.5 Opis tablica

#### 4.5.1 KORISNIK

Tablica 12 Tablica korisnika

Atribut	Tip podatka	Opis varijable
Korisnik_id	INT	Jedinstveni identifikator korisnika
Username	VARCHAR (50)	Korisničko ime
Email	VARCHAR (100)	Adresa e-pošte korisnika
Password	VARCHAR (255)	Lozinka korisnika
Kontakt_broj	VARCHAR (20)	Telefonski broj korisnika
Uloga	ENUM('IGRAČ', 'ORGANIZATOR')	Uloga korisnika u sustavu

### 4.5.2 IGRAČ

Tablica 13 Tablica igrača

Atribut	Tip podatka	Opis varijable
Korisnik_id (FK)	INT	Referenca na tablicu KORISNIK

### 4.5.3 ORGANIZATOR

Tablica 14 Tablica organizatora

Atribut	Tip podatka	Opis varijable
Organizator_id	INT	Jedinstveni identifikator organizatora
Korisnik_id (FK)	INT	Referenca na tablicu KORISNIK

### 4.5.4 TIM

Tablica 15 Tablica Tima

Atribut	Tip podatka	Opis varijable
Team_id	INT	Jedinstveni identifikator tima
Ime_tima	VARCHAR (100)	Naziv tima
Broj_clanova	INT	Broj članova tima
Status_prijave	ENUM ('ČEKA', 'ODOBRENO', 'ODBIJENO')	Status prijave tima
BrBodova	INT	Broj osvojenih bodova
Rang	INT	Trenutni rang tima
sent_at	DATETIME	Datum i vrijeme slanja prijave
odobreno_at	DATETIME	Datum i vrijeme odobrenja prijave
Kviz_id (FK)	INT	Referenca na tablicu KVIZ

#### 4.5.5 KVIZ

Tablica 16 Tablica kviza

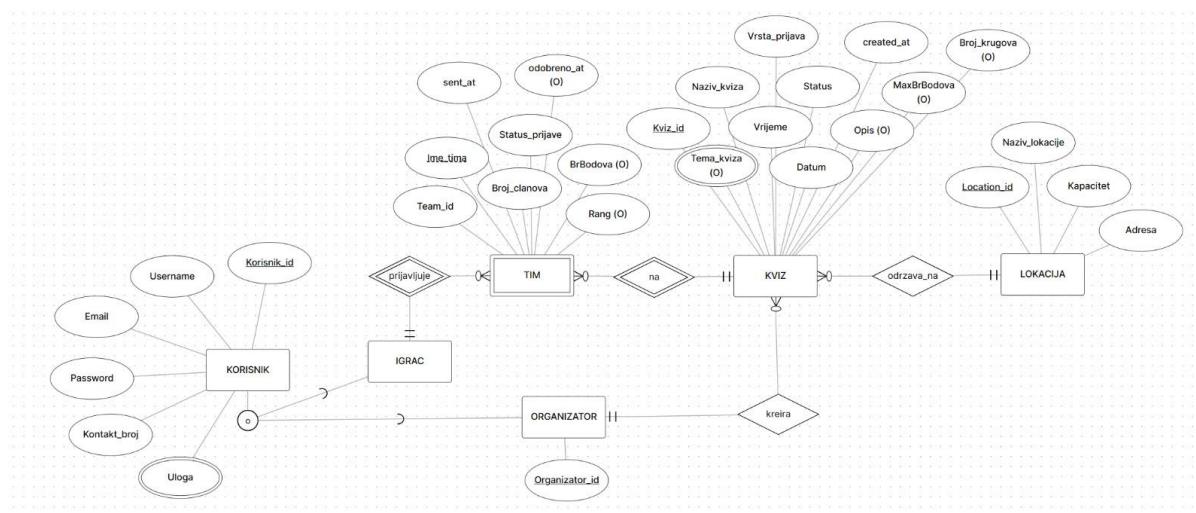
Atribut	Tip podatka	Opis varijable
Kviz_id	INT	Jedinstveni identifikator kviza
Naziv_kviza	VARCHAR (100)	Naziv kviza
Tema_kviza	VARCHAR (100)	Tema kviza
Datum	DATE	Datum održavanja kviza
Vrijeme	TIME	Vrijeme održavanja kviza
Vrsta_prijava	ENUM ('TIMSKA','POJEDINAČNA')	Tip prijave
Status	ENUM ('AKTIVAN', 'ZAVRŠEN', 'OTKAZAN')	Status kviza
Opis	TEXT	Kratki opis kviza
created_at	DATETIME	Datum i vrijeme kreiranja kviza
MaxBrBodova	INT	Maksimalni broj bodova
Broj_krugova	INT	Ukupan broj krugova u kvizu
Location_id (FK)	INT	Referenca na tablicu LOKACIJA
Organizator_id (FK)	INT	Referenca na tablicu ORGANIZATOR

## 4.5.6 LOKACIJA

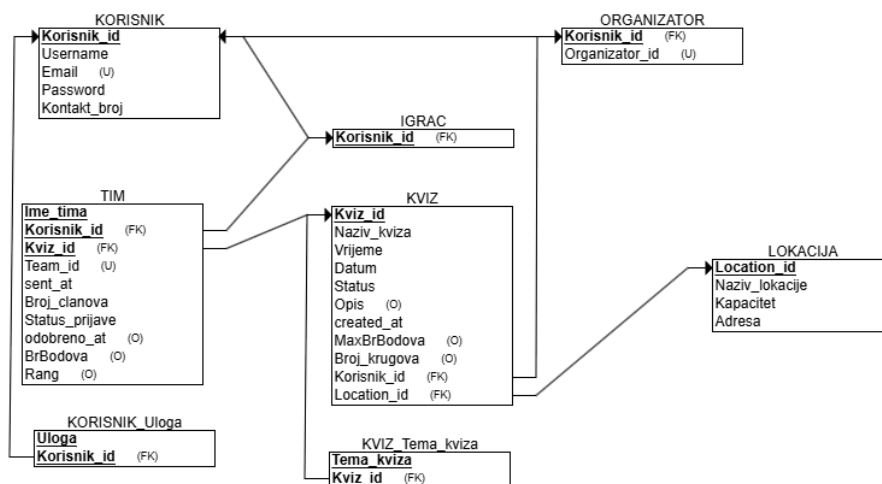
Tablica 17 Tablica lokacija

Atribut	Tip podatka	Opis varijable
Location_id	INT	Jedinstveni identifikator lokacije
Naziv_lokacije	VARCHAR (100)	Naziv lokacije kviza
Kapacitet	INT	Maksimalan broj sudionika
Adresa	VARCHAR (255)	Adresa održavanja kviza

## 4.6 Dijagram baze podataka



Slika 12 prikaz dijagrama baze podataka



Slika 13 Shema baze podataka (ERD)



## Dijagram razreda

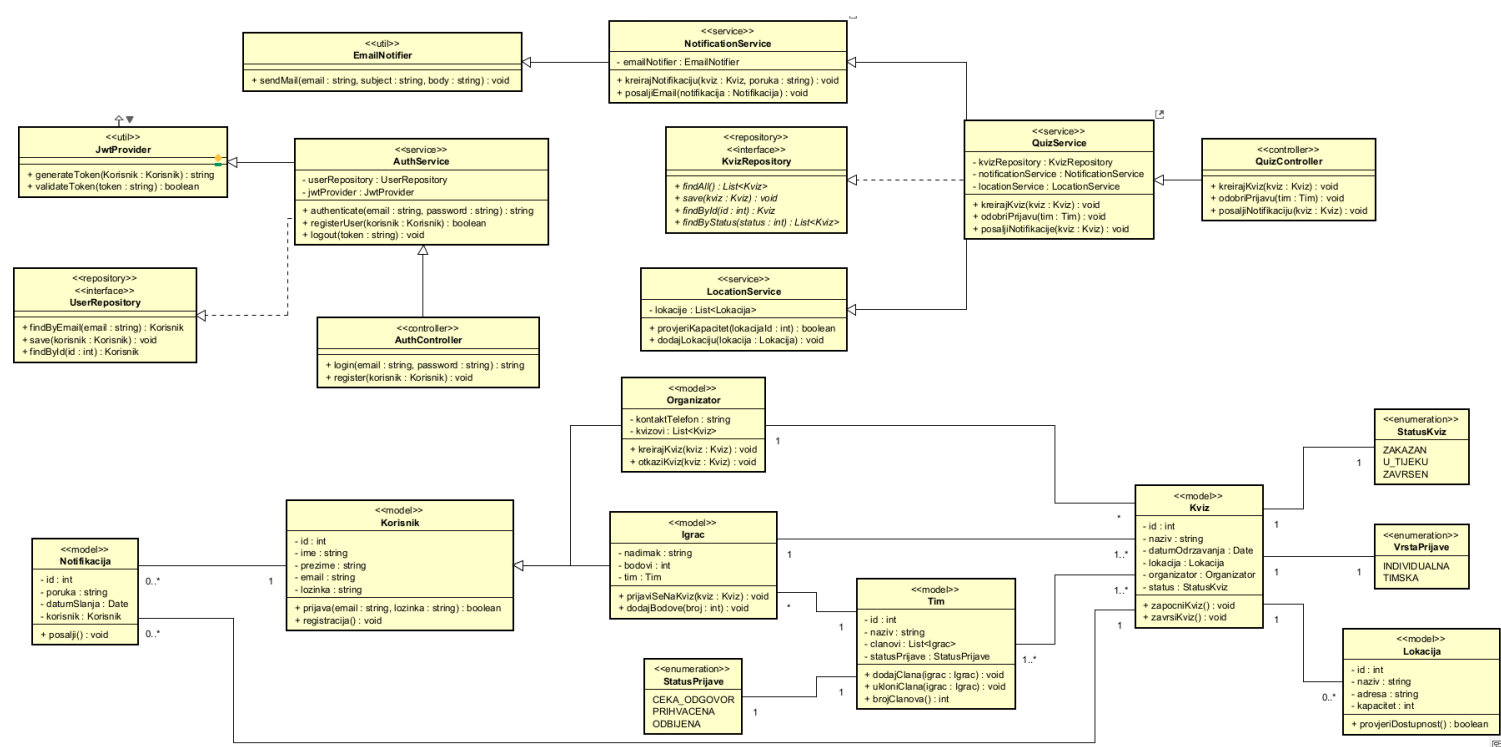
Ovaj dijagram prikazuje strukturu i odnose između razreda uključenih u sustav 'QuizWiz', koji omogućuje prijavu korisnika, organizaciju kvizova i slanje obavijesti sudionicima.

Korisnici su predstavljeni klasom 'Korisnik', iz koje nasljeđuju razredi 'Igrač' i 'Organizator', čime se omogućuje razlikovanje uloga u sustavu. Klasa 'AuthController' upravlja autentifikacijom i registracijom korisnika putem 'AuthService', koji koristi 'UserRepository' za pristup podacima o korisnicima te 'JwtProvider' za generiranje sigurnosnih tokena.

Organizatori mogu stvarati i upravljati kvizovima preko klase 'QuizController', koja se oslanja na 'QuizService'. Taj servis koristi 'KvizRepository' za rad s podacima o kvizovima, 'LocationService' za provjeru dostupnosti lokacija i 'NotificationService' za slanje obavijesti. Klasa 'NotificationService' koristi pomoćnu klasu 'EmailNotifier' za slanje poruka korisnicima.

U modelu su definirani razredi 'Kviz', 'Lokacija', 'Tim' i 'Notifikacija', koji međusobno povezuju korisnike, lokacije i statuse kvizova. Enum razredi 'StatusPrijave', 'StatusKviz' i 'VrstaPrijave' određuju stanja i vrste prijava unutar sustava.

Ovaj dijagram jasno prikazuje slojevitú arhitekturu sustava, s odvojenim kontrolerima, servisima, repozitorijima, modelima i pomoćnim klasama, što doprinosi preglednosti i modularnosti aplikacije.



Slika 14 UML dijagram razreda