

Programsko inženjerstvo

Ak. god. 2021./2022.

CodeShark

Dokumentacija, Rev. 1.0

Grupa: DomeFanClub
Voditelj: Marko Damjanić

Datum predaje: 26. 10. 2021.

Nastavnik: Hrvoje Nuić

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	23
3.2 Ostali zahtjevi	26
4 Arhitektura i dizajn sustava	27
4.1 Baza podataka	28
4.1.1 Opis Tablica	28
4.1.2 Dijagram baze podataka	34
4.2 Dijagram razreda	35
4.3 Dijagram stanja	39
4.4 Dijagram aktivnosti	40
4.5 Dijagram komponenti	41
5 Implementacija i korisničko sučelje	42
5.1 Korištene tehnologije i alati	42
5.2 Ispitivanje programskog rješenja	43
5.2.1 Ispitivanje komponenti	43
5.2.2 Ispitivanje sustava	43
5.3 Dijagram razmještaja	44
5.4 Upute za puštanje u pogon	45
6 Zaključak i budući rad	46
Popis literature	47
Indeks slika i dijagrama	48

Dodatak: Prikaz aktivnosti grupe

49

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Marko Damjanić	26.10.2021.
0.2	Napravljen opis projekta	Marko Damjanić	7.11.2021.
0.3	Dodani dionici i aktori	Fran Jelavić	16.11.2021.
0.4	Dodan prvi dio obrazaca uporabe	Luka Maros	16.11.2021.
0.4.1	Dodan drugi dio obrazaca uporabe	Antonio Griparić	16.11.2021.
0.5	Dodana arhitektura i dizajn sustava	Fran Jelavić	16.11.2021.
0.6	Uspostavljena Baza podataka	Marko Damjanić	16.11.2021.
0.7	Dodani Sekvencijski dijagrami	Luka Maros	18.11.2021.
0.7.1	Dodani ostali zahtjevi	Marko Damjanić	18.11.2021
0.7.2	Popravljen baza podataka	Marko Damjanić	18.11.2021
0.7.3	Dodani dijagrami obrazaca uporabe	Antonio Griparić	19.11.2021

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.8	Dodani dijagrami razreda	Roko Mihalić	19.11.2021.
0.9	Korigiranje teksta	Marko Damjanić	19.11.2021.
1.0	Provjera dokumentacije	Marko Damjanić	19.11.2021.
1.1	Dodan dijagram stanja	Marko Damjanić	13.1.2022.

2. Opis projektnog zadatka

Rješavanjem programskih zadataka i kompetitivnim programiranjem današnji programeri razvijaju svoje logičke vještine, ubrzavaju brzinu razvoja svoga koda, pišu kvalitetniji kod i lakše briljiraju kod firmi koje cijene visoke kompetitivnosti programera.

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "CodeShark" koja će korisniku omogućiti da rješava programske zadatke prik ladne njegovim interesima i razini znanja, te da se natječe na natjecanjima organiziranih od strane voditelja. Zadatak voditeljima je stvarati zadatke korisnicima te organizirati natjecanja koja će najbolje natjecatelje nagraditi trofejima prikazanim na profilu korisnika.

Korisnici na stranici mogu pregledavati i rješavati zadatke, natjecati se u natjecanjima koja su dostupna u kalendaru, te pratiti profile ostalih natjecatelja i voditelja natjecanja. Neregistriranim korisnicima se prilikom pokretanja zadataka ili tijekom pokušaja prijave na natjecanje otvara stranica za log-in pomoću već registriranog profila (potrebno upisati korisničko ime i lozinku) te ako nemaju već registriran profil imaju opciju registracije. Za registraciju novog profila potrebni su sljedeći podaci:

- korisničko ime
- fotografija
- lozinka
- ime
- prezime
- email adresa
- željena uloga na koju se prijavljuje (voditelj natjecanja ili natjecatelj)

Registracija se završava potvrdom preko email adrese. Automatski se korisniku dodjeljuju prava natjecatelja a za ulogu i prava voditelja mora dodatno potvrditi i administrator. Registrirani korisnik može pregledavati, mijenjati osobne podatke i izbrisati svoj korisnički račun.

Natjecatelj može pristupiti stranici s zadacima za vježbanje, na kojoj je prikazan popis svih postojećih zadataka koji su već bili iskorišteni na jednom od natjecanja (javni zadatci) te ostalih zadataka koje su voditelji dodali. Zadaci se mogu pretraživati po korisničkom imenu autora, sortirati po težini, statusu riješenosti te raznim drugim kategorijama podjele. Pritiskom na zadatak otvara se stranica tog zadatka gdje mu je omogućeno učitavanje programskog rješenja u aplikaciju. Učitani program se ovisno o potrebi prevodi u računalni kod, te potom provjerava na temelju zadanih primjera. Izlaz programa mora biti jednak rješenju i izvesti se unutar zadanog vremena. Broj bodova koje je natjecatelj ostvario se dodjeljuje proporcionalno broju točno riješenih primjera.

Natjecatelj također može pristupiti stranici s natjecanjima. Na njoj je prikazan interaktivni kalendar s svim budućim, trenutnim i prošlim natjecanjima. Također sadrži i uređenu listu natjecanja koja se može sortirati po autoru, državi za koju je namijenjeno te klasi natjecanja (za početnike, veterane i otvoren upad). Pritiskom na natjecanje otvara se stranica tog natjecanja s svim bitnim detaljima. Ako se to natjecanje još nije održalo, natjecatelj se može na njega prijaviti a kada bude vrijeme početka natjecanje se može pridružiti.

U trenutku početka natjecanja svi zadatci postaju vidljivi aktivnim natjecateljima. Za svaki zadatak, natjecatelj može poslati datoteku s programskim kodom. Završetkom natjecanja objavljuje se rang lista svih natjecatelja po ostvarenom broju bodova. Natjecateljima se na temelju postignuća za prva tri mjesta dodjeljuje pehar. Za izračun osvojenih bodova na natjecanju potrebno je uzeti u obzir provedeno vrijeme za rješavanje zadatka i postotak točnih primjera.

Natjecatelj nakon natjecanja može vidjeti i popis svih učitanih rješenja od nekog drugog natjecatelja. Slično, svaki zadatak ima popis svih natjecatelja koji su učitali neko rješenje za taj zadatak, broj točnih primjera po najboljem učitavanju od natjecatelja, prosječnom vremenu izvršavanja po primjeru i gumb za dohvat učitano rješenja koji se aktivira samo onim natjecateljima koji su već potpuno točno riješili zadatak.

Ako se natjecanje već održalo natjecatelj može sebi pokrenuti virtualno natjecanje. Virtualno natjecanje je aktivno samo za natjecatelja koji ga je pokrenuo. Pri isteku ograničenog vremena ili po želji natjecatelja virtualno natjecanje završava. Pri završetku se natjecatelja rangira u usporedbi s službenim rezultatima korisnika koji su prisustvovali tom natjecanju.

Natjecatelj isto tako može napraviti virtualno natjecanje gdje aplikacija nasumično odabere zadatke, ali tako da ravnomjerno rasporedi težine zadataka prema količini bodova pojedinog zadatka.

Postoje još 2 vrste korisnika, oboje s većim pravima od natjecatelja, a to su:

- voditelj
- administrator

Voditelj ima mogućnosti učitati nove zadatke u aplikaciju i organizirati natjecanja. Za postaviti novi zadatak potrebno je:

- naziv zadatka
- broj bodova (težina 1-5)
- vremensko ograničenje izvršavanja programa
- tekst zadatka
- primjeri za evaluaciju (ulaz i izlaz programa)
- privatnost zadatka

Privatni zadatak automatski postaje javan završetkom natjecanja.

Voditelj može izraditi natjecanje i ono svima postaje vidljivo u kalendaru natjecanja. Za organizaciju natjecanja su mu potrebni:

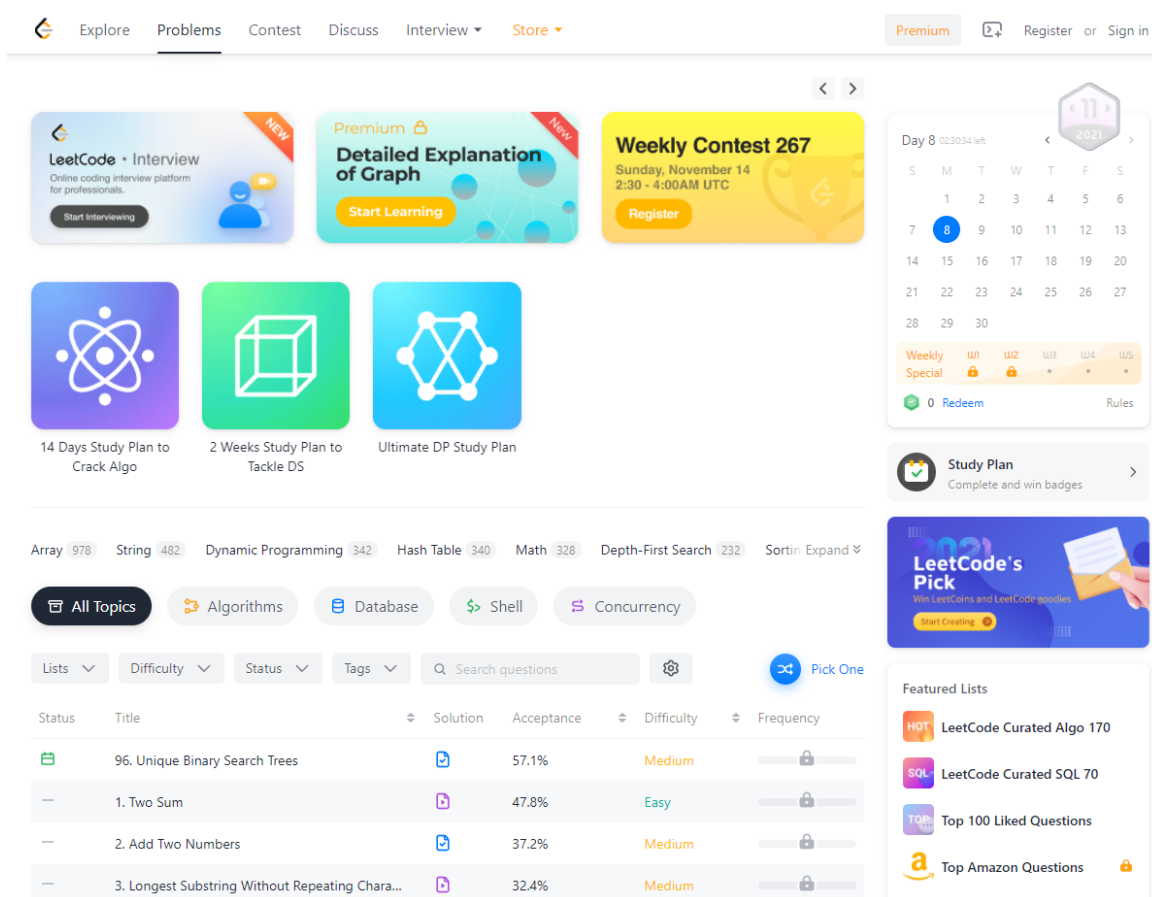
- naziv natjecanja
- tekst natjecanja
- vrijeme početka i završetka natjecanja (max 48 sati)
- broj zadataka
- odabrani privatni zadaci koje je voditelj učitao
- sličica pehara
- država natjecanja

Voditelj može uređivati vlastito objavljene zadatke i natjecanje. Uređivanjem zadatka se ne mijenjaju prethodno ostvareni rezultati na tom zadatku.

Administrator sustava ima najveće ovlasti. On može vidjeti popis svih registriranih korisnika i njihovih osobnih podataka te im mijenjati dodijeljena prava i osobne podatke. Ima potpun pristup bazi zadataka i natjecanja te ih ima pravo obrisati.

Uz vlastiti profil, u web aplikaciji postoji stranica s listom profila preko koje se mogu pogledati statistike i trofeji drugih korisnika. Na profilu natjecatelja su ispisane statistike o broju točno riješenih i broju isprobanih zadataka. Na profilima je također kalendar natjecanje na kojem je korisnik prisustvovao, a za voditelje i ona natjecanja koja su organizirali. Profili voditelja također sadrže popis njihovih učitanih zadataka s mogućnošću sortiranja.

Od sličnih rješenja istaknut je <https://leetcode.com/>. Za razliku od leetcode-a CodeShark ima mogućnosti virtualnih natjecanja koja pridonose boljoj pripremi korisnika. CodeShark se također više fokusira na svojom jednostavnosti i interaktivnosti.



Slika 2.1: <https://leetcode.com/problemset/all/>

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Natjecatelj
2. Voditelj
3. Administrator
4. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) pretraživati natjecanja i pristupiti popisu zadataka s natjecanja i sudionika natjecanja
 - (b) pretraživati i čitati zadatke s natjecanja
 - (c) pristupiti informacijama tuđeg korisničkog profila:
 - i. korisničko ime, ime i prezime, i osvojene nagrade
 - ii. statistike u što se ubraja broj točno riješenih i broj isprobanih zadataka
 - iii. popis natjecanja na kojima je korisnik prisustvovao ili natjecanja koja je održavao
 - (d) se registrirati u sustav, stvoriti novi korisnički račun za koji:
 - i. su mu potrebni korisničko ime, lozinka, ime i prezime te e-mail adresa
 - ii. mora izabrati razinu pristupa "natjecatelj" ili "voditelj"
2. Natjecatelj (inicijator) može:
 - (a) pregledavati i mijenjati osobne podatke
 - (b) izbrisati svoj korisnički račun
 - (c) pristupiti zadatcima za vježbu i rješavati ih
 - (d) prijaviti i odjaviti se na natjecanja i pristupiti natjecanjima

- (e) osvojiti nagrade
- (f) pokrenuti virtualno natjecanje

3. Voditelj (inicijator) može:

- (a) pregledavati i mijenjati osobne podatke
- (b) izbrisati svoj korisnički račun
- (c) pristupiti zadatcima za vježbu i rješavati ih
- (d) prijaviti i odjaviti se na natjecanja i pristupiti natjecanjima
- (e) osvojiti nagrade
- (f) pokrenuti virtualno natjecanje
- (g) pokrenuti i voditi natjecanje
- (h) kreirati zadatke
- (i) mijenjati i brisati vlastite zadatke

4. Administrator (inicijator) može:

- (a) pristupiti zadatcima za vježbu i rješavati ih
- (b) prijaviti i odjaviti se na natjecanja i pristupiti natjecanjima
- (c) osvojiti nagrade
- (d) pokrenuti virtualno natjecanje
- (e) pokrenuti i voditi natjecanje
- (f) kreirati zadatke
- (g) mijenjati vlastite zadatke
- (h) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (i) korisnike brisati i mijenjati im razinu pristupa aplikaciji (natjecatelj, voditelj, administrator)
- (j) pristupiti statistici

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Registracija u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne korisnicke podatke
 3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjehom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC2 - Prijava

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

UC3 - Pregled mogeg profila

- **Glavni sudionik:** Korisnik

- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Moj Profil"
 2. Aplikacija prikazuje osobne podatke korisnika

UC4 - Promjena podataka na mojem profilu

- **Glavni sudionik:** Korisnik
- **Cilj:** Promjeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Uredi profil"
 2. Korisnik mijenja svoje osobne podatke
 3. Korisnik sprema promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Korisnik promjeni svoje osobne podatke, ali ne odabire opciju "Spremi"
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaza iz prozora.

UC5 - Odjava iz sustava

- **Glavni sudionik:** Korisnik
- **Cilj:** Odjava iz sustava
- **Sudionici:** Korisnik
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Odjava"
 2. Sustav vraća korisnika na početnu stranicu

UC6 - Brisanje mog profila

- **Glavni sudionik:** Korisnik
- **Cilj:** Brisanje korisničkog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen

- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju "Izbriši Moj Profil"
2. Sustav odjavljuje korisnika i vraća ga na početnu stranicu
3. Baza Podataka se ažurira

UC7 - Pregled zadataka

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati sve dostupne zadatke za vježbu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Zadaci"
 2. Korisniku se prikažu svi dostupni zadaci za vježbu

UC8 - Rješavanje zadataka za vježbu

- **Glavni sudionik:** Korisnik
- **Cilj:** Pokretanje i rješavanje zadataka za vježbu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire jedan od zadataka za vježbu
 2. Sustav prebacuje korisnika na stranicu za rješavanje zadataka

UC9 - Dodavanje zadataka

- **Glavni sudionik:** Voditelj
- **Cilj:** Dodati novi zadatak u bazu zadataka za vježbu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora imati ovlasti "Voditelj"
- **Opis osnovnog tijeka:**
 1. Voditelj odabire opciju "Dodaj novi zadatak"
 2. Voditelj upisuje tekst, težinu, rješenje i vremensko ograničenje za zadatak
 3. Voditelj odabire opciju "Završi"
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Voditelj unosi podatke u nedozvoljenom formatu
 1. Sustav obavještava voditelja o neuspjelom upisu podataka

2. Voditelj popravljá unose

UC10 - Pregled sudionika koji su rješavali određeni zadatak

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Pregledati sudionike koji su rješavali određeni zadatak
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire jedan od zadataka
 2. Natjecatelju se otvara lista svih koji su ga rješavali

UC11 - Brisanje zadataka

- **Glavni sudionik:** Voditelj
- **Cilj:** Izbrisati zadatak iz baze zadataka za vježbu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora imati ovlasti "Voditelj"
- **Opis osnovnog tijeka:**
 1. Voditelj odabire jedan od zadataka za vježbu
 2. Voditelj odabire opciju "Izbriši zadatak"
 3. Baza podataka se ažurira

UC12 - Uređivanje zadataka za vježbu

- **Glavni sudionik:** Voditelj
- **Cilj:** Urediti zadatak iz baze zadataka za vježbu
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora imati ovlasti "Voditelj"
- **Opis osnovnog tijeka:**
 1. Voditelj odabire jedan od zadataka za vježbu
 2. Voditelj odabire opciju "Uredi zadatak"
 3. Voditelj upisuje tekst, težinu, rješenje i vremensko ograničenje za zadatak
 4. Voditelj odabire opciju "Završi"
 5. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Voditelj unosi podatke u nedozvoljenom formatu
 1. Sustav obavještava voditelja o neuspjelom upisu podataka
 2. Voditelj popravljá unose

UC13 - Pregled natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Pregledati listu svih natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Natjecanja"
 2. Sustav prenosi korisnika na stranicu s kalendarom i listom natjecanja

UC14 - Pokretanje virtualnog natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Vježba putem virtualnih natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Voditelj odabire opciju "Natjecanja"
 2. Korisnik odabire "Novo virtualno natjecanje"
 3. Korisnik započinje rješavanje natjecanja

UC15 - Dodavanje natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Dodati natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj odabire karticu "Natjecanja"
 2. Voditelj dodaje novo natjecanje
 3. Voditelj unosi potrebne podatke o natjecanju
 4. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 3.a Voditelj unosi podatke u nedozvoljenom formatu
 1. Sustav obavještava voditelja o neuspjelom upisu podataka
 2. Voditelj popravljajući neispravne unose i stvara natjecanje ili odustaje od stvaranja natjecanja

UC16 - Pokretanje natjecanja

- **Glavni sudionik:** Voditelj
- **Cilj:** Pokrenuti natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu voditelja
- **Opis osnovnog tijeka:**
 1. Voditelj odabire karticu "Natjecanja"
 2. Voditelj odabire jedno od svojih natjecanja
 3. Voditelj pokreće natjecanje

UC17 - Prijava na natjecanje

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Prijaviti se na nadolazeće natjecanje
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od nadolazećih natjecanja
 3. Natjecatelj se prijavljuje na odabrano natjecanje

UC18 - Pristupanje natjecanju

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Pristupiti natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav i na natjecanje koje je trenutno aktivno
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od trenutno aktivnih natjecanja na koje je prijavljen
 3. Natjecatelj pristupa natjecanju

UC19 - Odjava s natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Odjaviti se s natjecanja
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen u sustav i na natjecanje

- **Opis osnovnog tijeka:**

1. Natjecatelj odabire karticu "Natjecanja"
2. Natjecatelj odabire jedno od nadolazećih natjecanja na koje je prijavljen
3. Natjecatelj se odjavljuje s odabranog natjecanja

UC20 - Uređivanje natjecanja

- **Glavni sudionik:** Voditelj

- **Cilj:** Urediti natjecanje

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu voditelja

- **Opis osnovnog tijeka:**

1. Voditelj odabire karticu "Natjecanja"
2. Voditelj odabire jedno od svojih natjecanja
3. Voditelj uređuje pojedinosti svog natjecanja
4. Promjene se upisuju u bazu podataka

- **Opis mogućih odstupanja:**

- 3.a Voditelj unosi podatke u nedozvoljenom formatu

1. Sustav obavještava voditelja o neuspjelom upisu podataka
2. Voditelj popravljajući neispravne unose i uspješno uređuje natjecanje ili odustaje od uređivanja natjecanja

UC21 - Pregled sudionika na natjecanju

- **Glavni sudionik:** Korisnik

- **Cilj:** Pregledati natjecatelje koji su sudjelovali na natjecanju

- **Sudionici:** Baza podataka

- **Preduvjet:** -

- **Opis osnovnog tijeka:**

1. Korisnik odabire karticu "Natjecanja"
2. Korisnik odabire jedno od prošlih natjecanja
3. Prikazuje se lista natjecatelja koji su sudjelovali na odabranom natjecanju

UC22 - Brisanje natjecanja

- **Glavni sudionik:** Administrator

- **Cilj:** Obrisati postojeće natjecanje

- **Sudionici:** Baza podataka

- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Natjecanja"
 2. Administrator odabire jedno od natjecanja
 3. Administrator briše odabrano natjecanje
 4. Natjecanje se briše iz baze podataka

UC23 - Dohvat učitano g rješenja zadatka s natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Dohvatiti učitano rješenje zadatka nekog od natjecatelja koji je sudjelovao na natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati potpuno točno riješen konkretan zadatak
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od prošlih natjecanja
 3. Natjecatelj odabire jedan od zadataka s odabranog natjecanja
 4. Natjecatelj dohvaća rješenje zadatka

UC24 - Završetak natjecanja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Završiti s natjecanjem
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i morao je pristupiti natjecanju
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od trenutno aktivnih natjecanja na koje je prijavljen
 3. Natjecatelj pristupa natjecanju
 4. Nakon rješavanja zadataka, a prije isteka vremena, natjecatelj završava natjecanje

UC25 - Pregled ostalih korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled profila ostalih korisnika

- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire karticu "Korisnici"
 2. Prikazuje se lista svih registriranih natjecatelja i voditelja

UC26 - Promjena ovlasti korisniku

- **Glavni sudionik:** Administrator
- **Cilj:** Promjeniti ovlasti korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Korisnici"
 2. Administrator odabire korisnika
 3. Administrator mijenja odabranom korisniku ovlasti
 4. Promjene se upisuju u bazu podataka

UC27 - Pregled statistike pojedinog korisnika

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregledati statistiku pojedinog korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire karticu "Korisnici"
 2. Korisnik odabire određenog korisnika s liste
 3. Prikazuje se statistika odabranog korisnika

UC28 - Pregled svih učitanih rješenja nekog natjecatelja

- **Glavni sudionik:** Natjecatelj
- **Cilj:** Pregledati sva učitana rješenja natjecatelja koji je sudjelovao na natjecanju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen
- **Opis osnovnog tijeka:**
 1. Natjecatelj odabire karticu "Natjecanja"
 2. Natjecatelj odabire jedno od prošlih natjecanja

3. Natjecatelj odabire jednog od natjecatelja koji je sudjelovao na natjecanju
4. Prikazuju se sva učitana rješenja odabranog natjecatelja

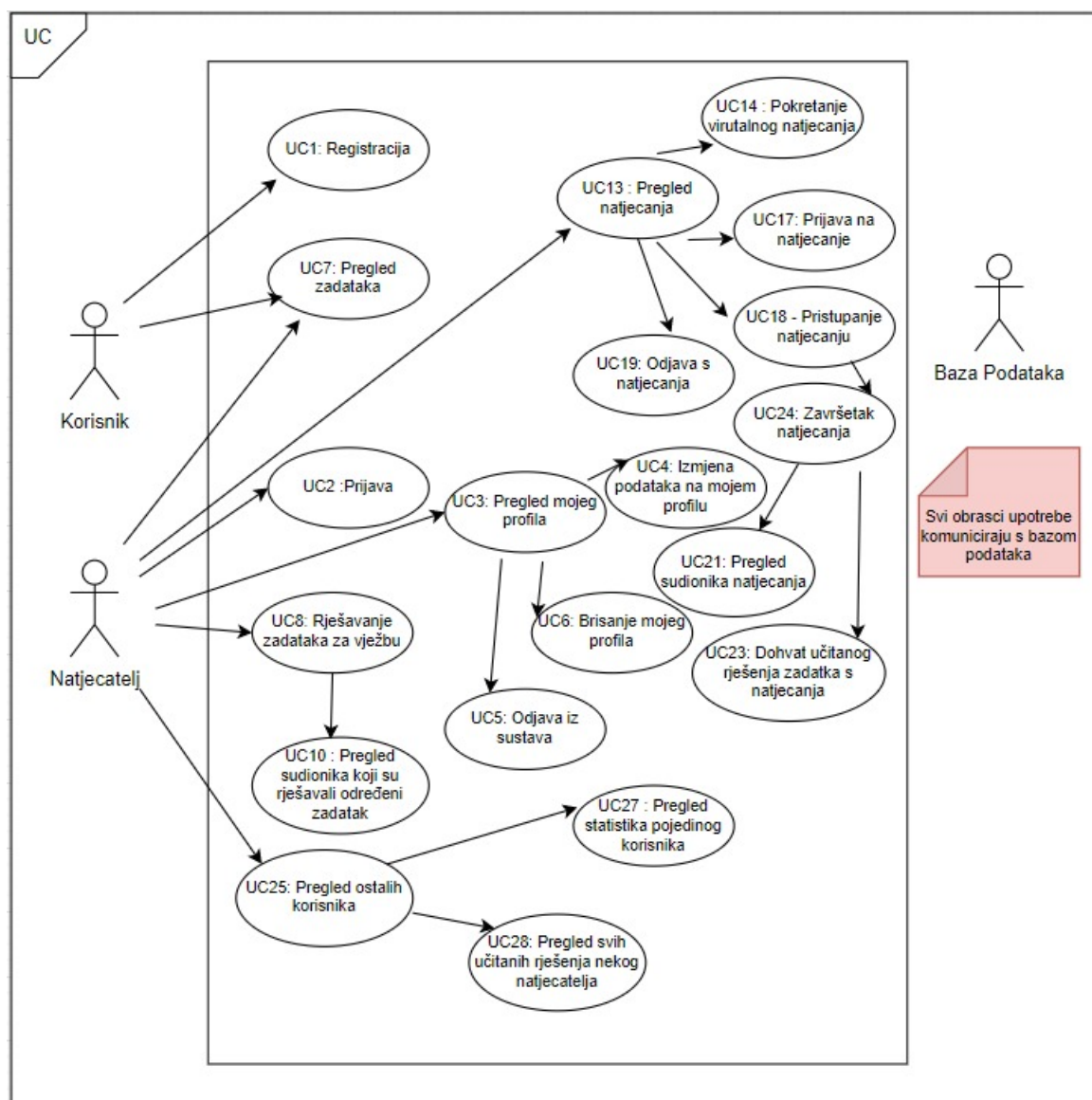
UC29 - Promjena osobnih podataka korisniku

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti osobne podatke korisniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Korisnici"
 2. Administrator odabire određenog korisnika
 3. Administrator mijenja željene osobne podatke korisniku
 4. Promjene se upisuju u bazu podataka
- **Opis mogućih odstupanja:**
 - 3.a Voditelj unosi podatke u nedozvoljenom formatu
 1. Sustav obavještava administratora o neuspjelom upisu podataka
 2. Administrator popravljajući neispravne unose i uspješno mijenja osobne podatke odabranog korisnika ili odustaje od promjena

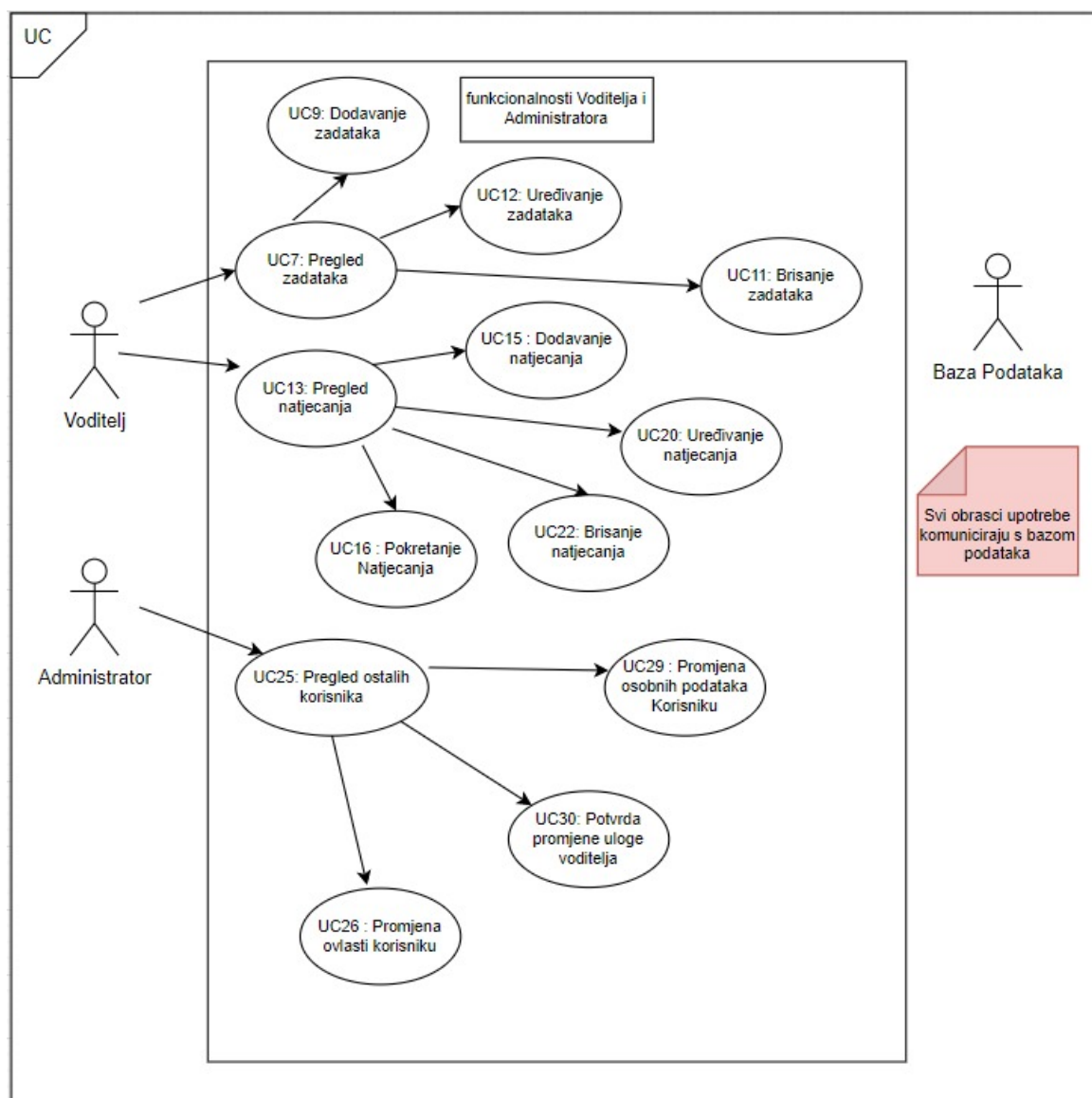
UC30 - Potvrda promjene uloge voditelja

- **Glavni sudionik:** Administrator
- **Cilj:** Potvrditi promjenu uloge voditelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik mora biti prijavljen i imati ulogu administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire karticu "Korisnici"
 2. Administrator odabire određenog korisnika
 3. Administrator potvrđuje korisniku promjenu uloge u voditelja natjecanja
 4. Promjene se upisuju u bazu podataka

Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost korisnika i natjecatelja

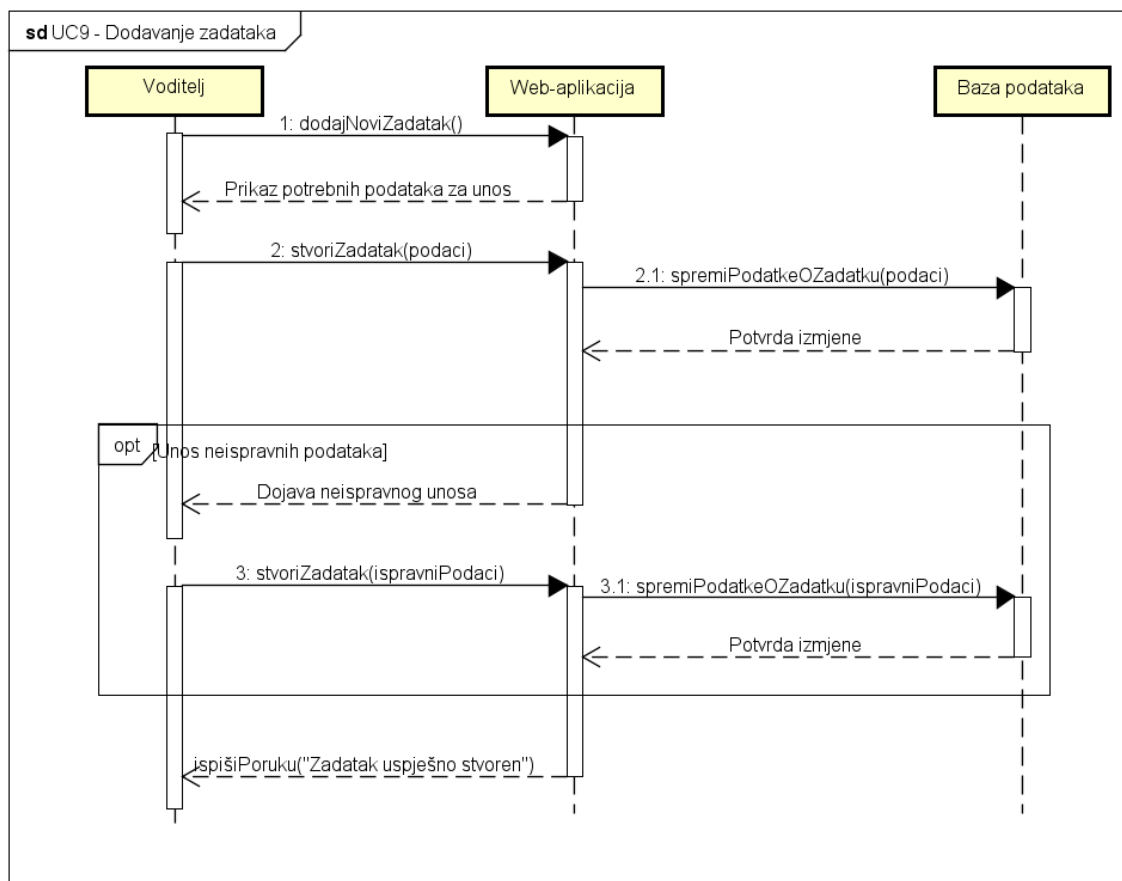


Slika 3.2: Dijagram obrasca uporabe, funkcionalnost voditelja i administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC9 - Dodavanje zadatka

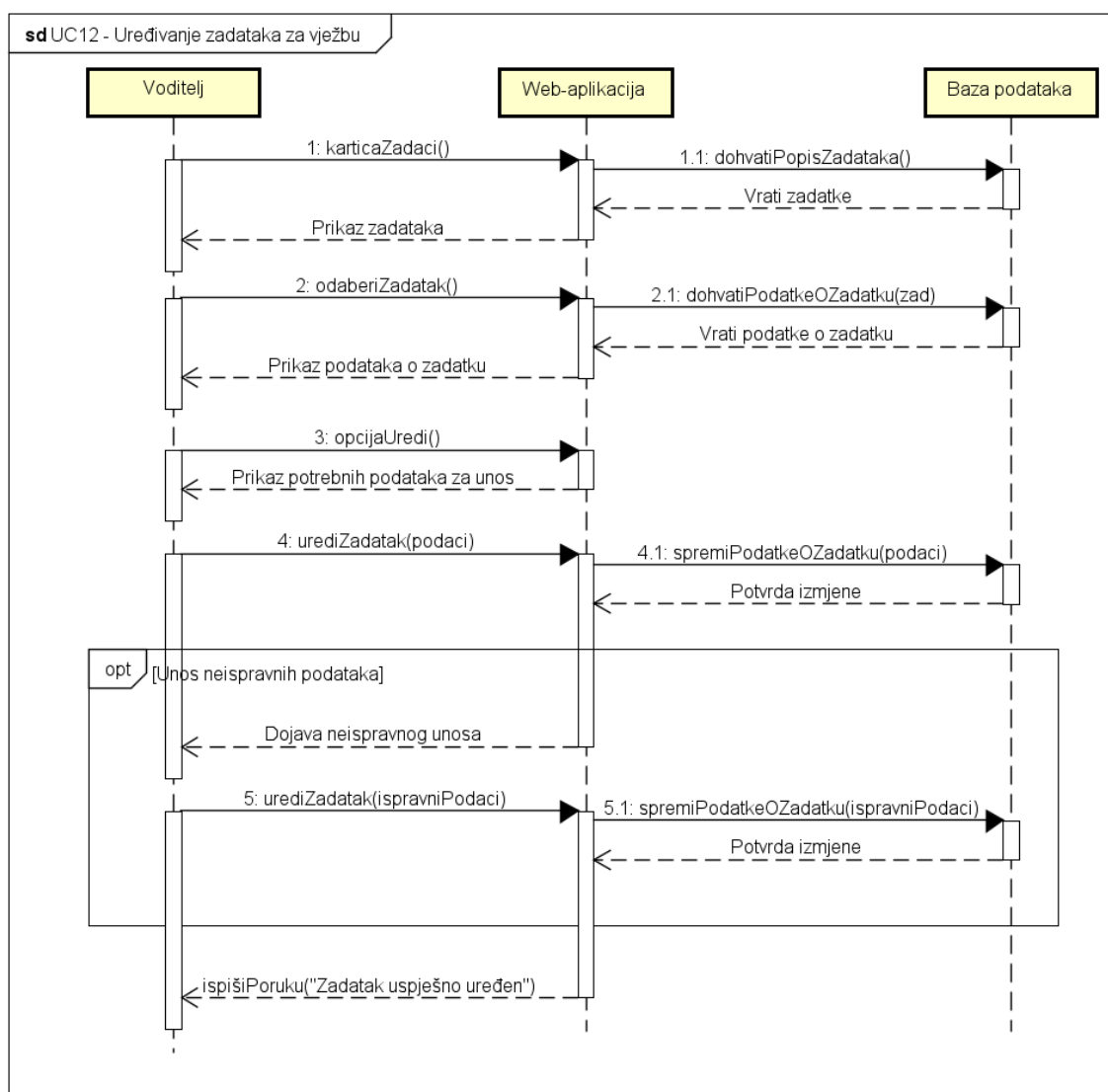
Voditelj šalje zahtjev za dodavanjem novog zadatka te mu se prikazuje stranica s poljima koja treba popuniti. Ta polja su naziv zadatka, broj bodova (tj. težina 1-5), vremensko ograničenje izvršavanja programa, tekst zadatka, primjeri za evaluaciju (ulaz i očekivani izlaz programa) i privatnost zadatka. Voditelj upisuje tražene podatke i sprema promjene. Ukoliko su svi upisani podaci ispravno napisani, voditelj dobiva potvrdu u obliku poruke da je uspješno stvorio zadatak, a ako su upisani podaci na neki način neispravni, voditelju se dojavljuje greška i nudi da ispravi podatke.



Slika 3.3: Sekvencijski dijagram za UC9

Obrazac uporabe UC12 - Uređivanje zadataka za vježbu

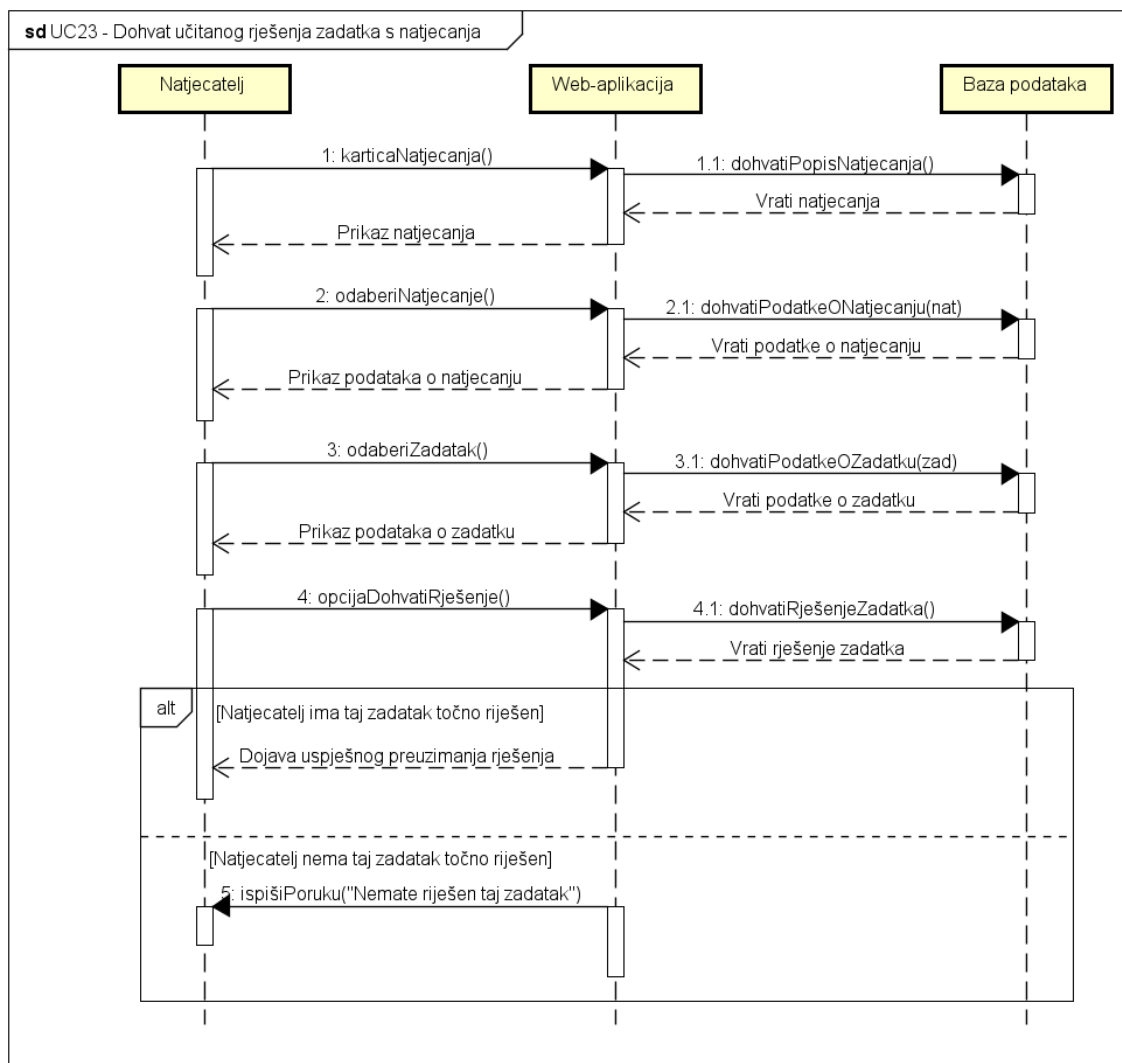
Voditelj najprije odabire karticu sa zadacima te mu se prikazuje popis zadataka. Zatim odabire određeni zadatak s popisa i dobiva sve informacije o njemu. Zatim odabire opciju za uređivanje zadataka i, slično kao i u prethodnom dijagramu, prikazuje mu se stranica s istim poljima koja treba popuniti. Voditelj uređuje željene podatke, sprema promjene i, ako su uređeni podaci ispravni, dobiva potvrdu o uspješnom uređivanju zadatka, inače dobiva poruku greške i nudi mu se da ispravi podatke.



Slika 3.4: Sekvencijski dijagram za UC12

Obrazac uporabe UC23 - Dohvat učitano g rješenja zadatka s natjecanja

Natjecatelj najprije odabire karticu s natjecanjima i dobiva popis natjecanja. Nakon toga bira željeno natjecanje i dobiva sve informacije o njemu. Zatim odabire zadatak čije rješenje želi dohvatiti i prikazuju mu se podaci o zadatku. Na ovoj stranici odabire opciju za dohvat rješenja i prima potvrdu o njegovom uspješnom preuzimanju samo ako je natjecatelj već riješio taj zadatak, i to potpuno točno, a inače dobiva poruku da ne može preuzeti rješenje tog zadatka.



Slika 3.5: Sekvencijski dijagram za UC23

3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Sustav mora validirati podatke na klijentskoj i poslužiteljskoj strani
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS.
- Sustav je responzivan na mobilnim uređajima

4. Arhitektura i dizajn sustava

Arhitekturu je moguće podijeliti na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijskog sadržaja vezanog uz njih. Korisnik putem web preglednika šalje zahtjev web poslužitelju te mu web preglednik kao interpreter prevodi web-stranicu i njezin sadržaj u format koji je korisniku razumljiv.

Web poslužitelj temelj je rada web aplikacije te je njegov glavni zadatak omogućiti komunikaciju klijenta s aplikacijom. Komunikacija se ostvaruje preko protokola HTTP (engl. *Hyper Text Transfer Protocol*), koji služi za prijenos informacija na webu. Web aplikacija se pokreće preko poslužitelja koji joj prosljeđuje zahtjev od strane korisnika.

Web aplikacija služi za obradu korisničkih zahtjeva. Ovisno o zahtjevu, web aplikacija tijekom obrade zahtjeva pristupa bazi podataka te korisniku vraća odgovor u obliku HTML (engl. *HyperText Markup Language*) dokumenta koji se prikazuje preko web preglednika.

Codeshark web aplikacija bazirana je na programskom jeziku Python-u za razvoj *backend*-a, zajedno s JavaScript-om uz biblioteku React za razvoj *frontend*-a. Kao razvojno okruženje koristio se Microsoft Visual Studio.

Arhitektura sustava temelji se na konceptu MVC-a (Model-View-Controller). Karakteristika MVC koncepta je nezavisan razvoj pojedinih dijelova aplikacije što za posljedicu ima jednostavnije ispitivanje kao i jednostavno razvijanje i dodavanje novih svojstava u sustav.

MVC koncept sastoji se od:

- **Model** - Središnja komponenta sustava. Predstavlja dinamičke strukture podataka, neovisne o korisničkom sučelju. Izravno upravlja podacima, logikom i pravilima aplikacije. Ujedno i prima ulazne podatke od Controller-a.

- **View** - Bilo kakav prikaz podataka, poput grafa. Mogući su različiti prikazi iste informacije poput grafičkog ili tabličnog prikaza podataka.
- **Controller** - Prima ulaze i prilagođava ih za prosljeđivanje Model-u ili View-u. Upravlja korisničkim zahtjevima i temeljem njih izvodi daljnju interakciju s ostalim elementima sustava.

4.1 Baza podataka

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

- Korisnik
- Natjecanje
- Zadatak
- Test-Primjeri
- Upload-Rjesenja
- Virtualno-Natjecanje
- Trofej
- Sudjeluje-Na
- Je-Osvojio
- Klasa-Natjecanja

4.1.1 Opis Tablica

Korisnik Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: Korisnikov ID, Korisničko ime, lozinku, ime, prezime, sliku profila, email, titulu i korisnikovu razinu ovlasti. Ovaj entitet u vezi je One-to-Many s entitetom Natjecanje preko ID-a korisnika (AutorID), u vezi One-to-Many s entitetom Zadatak preko ID-a korisnika (AutorID), u vezi Many-to-Many s Je-Osvojio preko ID-a korisnika, u vezi Many-to-Many s Sudjeluje-na preko ID-a korisnika, u vezi One-to-Many s entitetom Upload-Rjesenja preko ID-a korisnika te u vezi One-to-Many s entitetom VirtNatjecanje preko ID-a korisnika.

Korisnik		
Korisnik ID	INT	jedinstveni indikator korisnika
KorisnickoIme	VARCHAR	identificirajuće ime korisnika
Lozinka	VARCHAR	hash lozinke
SlikaProfila	VARCHAR	slika profila korisnika
Ime	VARCHAR	ime korisnika
Prezime	VARCHAR	prezime korisnika
Email	VARCHAR	email korisnika
Titula	VARCHAR	prilagođena titula korisnika
NivouPrava	INT	razina ovlasti korisnika
Token	VARCHAR	token napravljen za korisnika
Token Generiran	TIMESTAMP	vrijeme kada je token generiran
Aktivan	Boolean	stanje verifikacije korisnika

Natjecanje Ovaj entitet sadržava sve važne informacije o održavanju natjecanja. Sadrži attribute: ID Natjecanja, ime natjecanja, tekst natjecanja, vrijeme kraja natjecanja, vrijeme početka natjecanja, sliku trofeja, broj zadataka, ID autora natjecanja, ID klase natjecanja, ID trofeja. Ovaj entitet u vezi je Many-to-One s entitetom Korisnik preko ID-a korisnika (AutorID), u vezi One-to-Many s entitetom Zadatak preko ID-a natjecanja, u vezi One-to-Many s VirtNatjecanje preko ID-a natjecanja, u vezi One-to-One s entitetom Trofej preko ID-a trofeja te u vezi Many-to-One s entitetom IDKlasaNatjecanja preko ID-a klase natjecanja.

Natjecanje		
NatjecanjeID	INT	jedinstveni indikator natjecanja
ImeNatjecanja	VARCHAR	identificirajuće ime natjecanja
Tekst Natjecanja	VARCHAR	sadržaj teksta natjecanja

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Natjecanje		
VrijemeKraj	TIMESTAMP	vrijeme završetka natjecanja
VrijemePoc	TIMESTAMP	vrijeme početka natjecanje
SlikaTrofeja	VARCHAR	sličica trofeja
BrojZadataka	INT	broj zadataka u natjecanju
AutorID	INT	autorov Korisnik ID (korisnik.KorisnikID)
ID Klase Natjecanja	INT	ID Klase Natjecanja (klasanatjecanja.IDKlaseNatjecanja)
TrofejID	INT	Trofej ID (trofej.TrofejID)

Zadatak Ovaj entitet sadržava sve važne informacije o zadacima. Sadrži attribute: Zadatak ID, ime zadatka, tekst zadatka, bodovi zadatka, maksimalno vrijeme izvršavanja zadatka, privatnost zadatka, slag naziv zadatka, id autora , id natjecanja(opcionalno). Ovaj entitet u vezi je Many-to-One s entitetom Korisnik preko ID-a korisnika (AutorID), u vezi Many-to-One s entitetom Natjecanje preko ID-a natjecanja , u vezi One-to-Many s TestPrimjeri preko ID-a zadatka, u vezi One-to-Many s entitetom UploadRjesenja preko ID-a zadatka te u 4 veze One-to-Many s entitetom VirtNatjecanje preko ID-a zadataka.

Zadatak		
ZadatakID	INT	jedinstveni indikator zadatka
ImeZadatka	VARCHAR	identificirajući naziv zadatka
Bodovi	INT	broj bodova(težina 1-5)
MaxVrijeme Izvrs	NUMERIC	maksimalno vrijeme izvršavanja programa
TekstZadatka	VARCHAR	sadržaj teksta zadatka
Privatnost	Boolean	stanje privatnosti zadatka
Slag	VARCHAR	slag verzija naziva zadatka
AutorID	INT	autorov Korisnik ID (korisnik.KorisnikID)

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Zadatak		
NatjecanjeID	INT	(opcionalno) Natjecanje ID (natjecanje.NatjecanjeID)

Test Primjeri Ovaj entitet sadržava sve važne informacije o testnim primjerima za zasebne zadatke. Sadrži attribute: Zadatak ID, ulaz testa, izlaz testa. Ovaj entitet u vezi je Many-to-One s entitetom Zadatak preko ID-a zadatka.

Test Primjeri		
Ulaz	VARCHAR	ulaz testa
Zadatak ID	INT	Zadatak ID (zadatak.ZadatakID)
Izlaz	VARCHAR	željeni izlaz testa

Upload Rjesenja Ovaj entitet sadržava sve informacije koje su bitne oko uploada rješenja na zadatak. Sadrži attribute: Zadatak ID, korisnikov id, vrijeme predaje rješenja, predano rješenje, prolaznost, prosječno vrijeme izvršavanja, aktivnost natjecanja. Ovaj entitet u vezi je Many-to-One s entitetom Korisnik preko ID-a korisnika te u vezi Many-to-One s entitetom Zadatak preko ID-a zadatka.

Upload Rjesenja		
Vrijeme Predaje	TIMESTAMP	vrijeme predaje rješenja
Korisnik ID	INT	Korisnik ID (korisnik.KorisnikID)
Zadatak ID	INT	Zadatak ID (zadatak.ZadatakID)
Predano Rjesenje	VARCHAR	datoteka koju je predao korisnik
Prolaznost	NUMERIC	posto riješenosti primjera zadataka
ProsjVrijeme Izvrs	NUMERIC	prosječno vrijeme izvršavanja po primjeru

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Upload Rjesenja		
NatjecanjeTraje	BOOLEAN	zadatkova veza s natjecanjima(1-natjecanje trenutno traje,0-natjecanje trenutno ne traje)

Virtualno Natjecanje Ovaj entitet sadržava sve važne informacije o Virtualnim natjecanjima. Sadrži attribute: Korisnikov ID, ID natjecanja, ID random zadatka težine 2, ID random zadatka težine 3, ID random zadatka težine 4 te ID random zadatka težine 5. Ovaj entitet u vezi je Many-to-One s entitetom Korisnik preko ID-a korisnika, u vezi Many-to-One s entitetom Natjecanje preko ID-a natjecanja te u 4 veze Many-to-One s entitetom Zadatak preko ID-ova zadataka.

Virt Natjecanje		
Korisnik ID	INT	Korisnik ID (korisnik.KorisnikID)
Natjecanje ID	INT	(Opcionalno) Natjecanje ID (natjecanje.NatjecanjeID)
RandZadTez2	INT	(Opcionalno) slučajno odabrani zadatak težine 2 (zadatak.ZadatakID)
RandZadTez3	INT	(Opcionalno) slučajno odabrani zadatak težine 3 (zadatak.ZadatakID)
RandZadTez4	INT	(Opcionalno) slučajno odabrani zadatak težine 4 (zadatak.ZadatakID)
RandZadTez5	INT	(Opcionalno) slučajno odabrani zadatak težine 5 (zadatak.ZadatakID)

Trofej Ovaj entitet sadržava sve važne informacije o trofejima. Sadrži attribute: Trofej ID, ime trofeja, slika trofeja. Ovaj entitet u vezi je Many-to-One s entitetom Korisnik preko ID-a korisnika, u vezi One-to-Many s entitetom Natjecanje preko ID-a trofeja te u vezi Many-to-Many s Je-Osvojio preko ID-a trofeja.

Trofej		
TrofejID	INT	jedinstveni indikator trofeja
ImeTrofeja	VARCHAR	ime trofeja
SlikaTrofeja	VARCHAR	slika trofeja

Sudjeluje Na Ovaj entitet sadržavi listu svih sudjelovanja korisnika na natjecanjima. Sadrži attribute: Korisnik ID, Natjecanje ID.

Sudjeluje Na		
Korisnik ID	INT	Korisnik ID (korisnik.KorisnikID)
Natjecanje ID	INT	Natjecanje ID (natjecanje.NatjecanjeID)

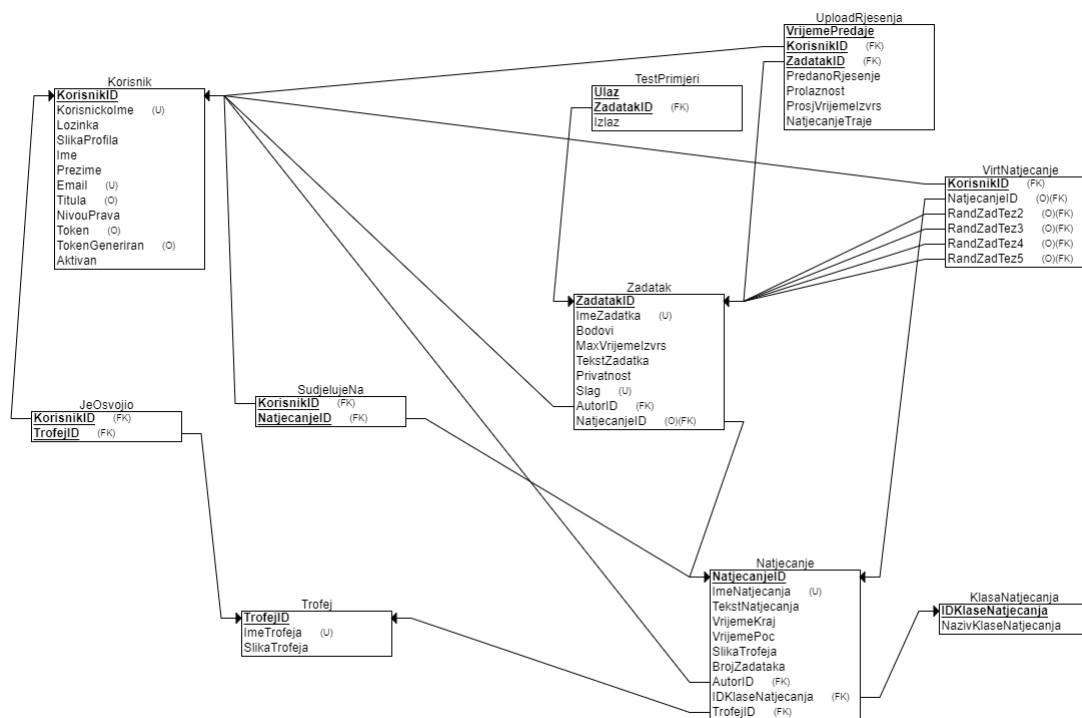
Je Osvojio Ovaj entitet sadržava listu svih korisnika s osvojenim peharima. Sadrži attribute: Korisnik ID, Trofej ID. Ovaj entitet u vezi je One-To-Many s entitetom Natjecanje preko ID-a klase natjecanja.

Je Osvojio		
Korisnik ID	INT	Korisnik ID (korisnik.KorisnikID)
Trofej ID	INT	Trofej ID (trofej.TrofejID)

Klasa Natjecanja Ovaj entitet sadržava sve informacije o klasi natjecanja. Sadrži attribute: ID klase natjecanja, naziv klase natjecanja.

Klasa Natjecanja		
ID Klase Natjecanje	INT	jedinstveni indikator klase natjecanje
Naziv Klase Natjecanja	VARCHAR	naziv klase natjecanja

4.1.2 Dijagram baze podataka



Slika 4.1: E-R dijagram baze podataka

4.2 Dijagram razreda

Na slici 4.2. vidimo osnovni model UML dijagrama koji prikazuje osnovne klase koje koristimo na backendu.

Trenutni dijagram razreda opisuje izgled klasa i njihovih metoda koje će nam biti potrebne u projektu. Trenutno pošto imamo registraciju, ulogiravanje i validaciju, koristimo samo klasu Korisnik te njene metode. Imamo privatnu metodu koja se zove `__get_id()` preko koje dobivamo korisnikid iz baze podataka, zato što je to metoda koja dohvaća privatni ključ te ne želimo da korisnik ikad ima pristup tome. Tu metodu trenutno ne koristimo previše već će biti jako korisna kasnije u konekcijama s instancama ostalih klasa prilikom dobivanja informacija iz ostalih tablica baze podataka. Kao što se vidi, svaka klasa ima konstruktor `__init__()` te primaju informacije koje su identične onima iz njihovih modela iz baze podataka.



Slika 4.2: Dijagram razreda

Budući da radimo u programskom jeziku Python, ne možemo lako prikazati Data Transfer objects. Međutim, prilikom registracije, sve informacije koje dobivamo s frontenda prosjeđujemo u konstruktor `Korisnik`. Kasnije, prilikom login-a, preko korisničkog imena dobivamo podatke iz baze te izrađujemo instancu `Korisnik`. S tim `Korisnikom` možemo baratati pomoću njegovih podataka. Tražimo iz baze podataka je li aktiviran te odgovara li hash unesene lozinke hashu lozinke koja je spremljena u bazu podataka, tj. u instanci `Korisnik`.

Naša klasa `Korisnik` se zapravo referira i na voditelja i na administratora. Oni svi imaju identične podatke te duplikacija koda nije potrebna. Velika razlika je samo u njihovom podatku `NivouPrava` o kojem će ovisiti kakve ovlasti na stranici ima. To će također biti funkcija koju će biti privatna. Voditelj će za razliku od

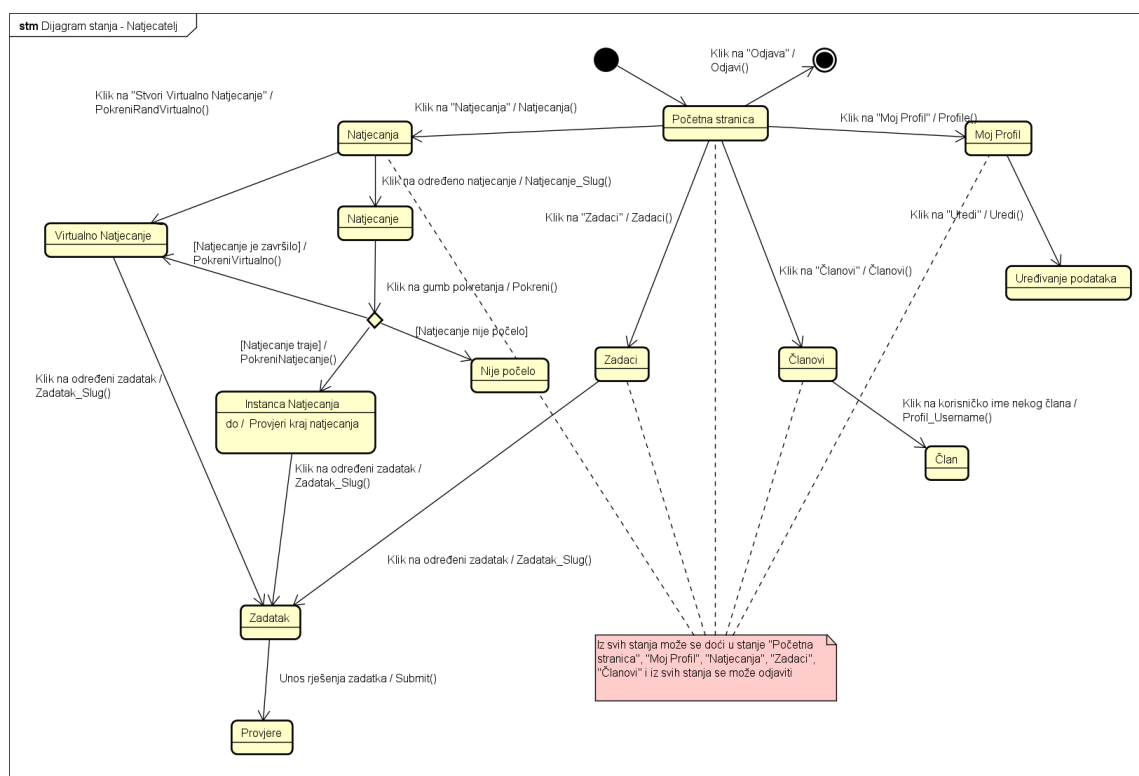
natjecatelja imati pristup stranicama za kreiranje zadataka te stvaranju natjecanja. Administrator će uz to sve moći uređivati ovlasti svih korisnika te će pomoću toga imati mogućnost potvrditi voditelja što će biti na posebnoj stranici.

dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

4.3 Dijagram stanja

Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.3 prikazan je dijagram stanja za registriranog natjecatelja. Nakon prijave, natjecatelju se prikazuje početna stranica na kojoj može otići na stranicu zadataka. Klikom na odabrani zadatak on unosi svoj programski kod koji se šalje na provjeru te dobiva rezultate. Također, može otići na stranicu natjecanja gdje mi se nudi sva bivša, trenutna i buduća natjecanja. Klikom na natjecanje ga vodi na stranicu tog natjecanje gdje ga ima opciju pokrenuti. Ovisno o statusu trajanja natjecanje, pokreće se različita instanca natjecanja. Na toj instanci se prikazuju zadaci koje je moguće riješiti u ograničenom vremenu. Uz stranice zadataka i natjecanja, natjecatelj uvijek može pristupiti stranici vlastitog profila na kojoj može uređivati svoje podatke, te stranici ostalih članova s koje može gledati profile ostalih korisnika.



Slika 4.3: Dijagram stanja

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

5.2 Ispitivanje programskog rješenja

dio 2. revizije

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.

5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.

5.4 Upute za puštanje u pogon

dio 2. revizije

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.

6. Zaključak i budući rad

dio 2. revizije

U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.

Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.

Popis literature

Kontinuirano osvježavanje

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

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

Indeks slika i dijagrama

2.1	https://leetcode.com/problemset/all/	8
3.1	Dijagram obrasca uporabe, funkcionalnost korisnika i natjecatelja .	21
3.2	Dijagram obrasca uporabe, funkcionalnost voditelja i administratora	22
3.3	Sekvencijski dijagram za UC9	23
3.4	Sekvencijski dijagram za UC12	24
3.5	Sekvencijski dijagram za UC23	25
4.1	E-R dijagram baze podataka	34
4.2	Dijagram razreda	36
4.3	Dijagram stanja	39

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.

1. sastanak

- Datum: u ovom formatu: 13. siječnja 2022.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 13. siječnja 2022.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

Tablica aktivnosti

Kontinuirano osvježavanje

Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

Dijagrami pregleda promjena

dio 2. revizije

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s gitlaba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s gitlab.com stranice, u izborniku Repository, pritiskom na stavku Contributors.