

Programsko inženjerstvo

Ak. god. 2020./2021.

Kamp Mlade Nade

Dokumentacija, Rev. 1

Grupa: *MajstoriSFera*

Voditelj: *Ivan Sabolić*

Datum predaje: *13. 11. 2020.*

Nastavnik: *Eugen Vušak*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	4
2.1 Uvodno	4
2.2 Prijave na kamp	4
2.3 Registracija korisničkog računa	4
2.4 Grupe	4
2.5 Aktivnosti	5
2.6 Početna stranica	5
3 Specifikacija programske potpore	7
3.1 Funkcionalni zahtjevi	7
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	17
3.2 Ostali zahtjevi	21
4 Arhitektura i dizajn sustava	22
4.1 Baza podataka	23
4.1.1 Opis tablica	23
4.1.2 Dijagram baze podataka	27
4.2 Dijagram razreda	28
4.3 Dijagram stanja	31
4.4 Dijagram aktivnosti	32
4.5 Dijagram komponenti	33
5 Implementacija i korisničko sučelje	34
5.1 Korištene tehnologije i alati	34
5.2 Ispitivanje programskog rješenja	35
5.2.1 Ispitivanje komponenti	35
5.2.2 Ispitivanje sustava	39
5.3 Dijagram razmještaja	45
5.4 Upute za puštanje u pogon	46
5.4.1 Instalacija Git-a	46

5.4.2	Instalacija Heroku CLI-a	46
5.4.3	Kreiranje Heroku udaljenog repozitorija	47
5.4.4	Postavljanje baze podataka	47
5.4.5	Postavljanje backenda aplikacije na Heroku poslužitelj	47
5.4.6	Postavljanje frontenda aplikacije na Heroku poslužitelj	48
6	Zaključak i budući rad	49
	Popis literature	50
	Indeks slika i dijagrama	51
7	Dodatak: Prikaz aktivnosti grupe	52
	Dodatak: Prikaz aktivnosti grupe	52
7.1	Dnevnik sastajanja	52
7.2	Tablica aktivnosti	54
7.3	Dijagrami pregleda promjena	56

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen template	Ivan	14/10/2020
0.2	Dodan UML dijagram	Ivan i Ivana	21/10/2020
0.3	Dodana baza podataka	Maja i Katarina	09/11/2020
0.4	Dodani funkcionalni zahtjevi	Maja i Katarina	10/11/2020
0.5.1	Obrasce i sekvencijski dijagrami	Ivan, Zvonimir i Martin	11/11/2020
0.5.2	Dodani obrasce i sekvencijski dijagrami	Ivana	11/11/2020
0.5.3	Dodani obrasce i sekvencijski dijagrami	Filip	11/11/2020
1.0	Uređena dokumentacija za reviziju	Ivan	12/11/2020
1.1	Dodani dijagram stanja, aktivnosti i komponenti i njihovi opisi	Katarina i Maja	07/01/2021
1.2	Dijagram komponenti i ispitivanje programskog rješenja	Ivana	11/01/2021
1.3	Upute za puštanje u pogon i dnevnik sastajanja	Ivana	12/01/2021
1.4	Dodatak	Zvonimir	13/01/2021

2. Opis projektnog zadatka

2.1 Uvodno

Cilj ovog projekta je razviti online platformu kojom bi se olakšala organizacija kampa računarstva „Mlade nade“.

2.2 Prijave na kamp

U opsegu ove platforme omogućene su prijave za sudjelovanje na kampu, ali i prijave za animatore. Prijave su omogućene samo za vrijeme trajanja prijava koje se zadaje od strane organizatora. Prijave za animatore i sudionike su u istim vremenskim intervalima. I animatori i sudionici moraju prilikom prijave upisati svoje puno ime, e-mail adresu, broj telefona, datum i godinu rođenja te kratko motivacijsko pismo. Sudionici mlađi od 18 godina moraju dodatno unijeti i broj telefona odgovorne osobe. Organizatori moraju moći vidjeti popis prijava te jednostavno odbiti ili prihvatiti prijavu.

2.3 Registracija korisničkog računa

Nakon što se nečija prijava prihvati, automatski se stvara račun za tu osobu te mu se pridodaje korisničko ime generirano tako da se uzme prvo slovo imena i prezime bez diakritičkih znakova (npr. „Ana Anić” - „aanic”). Ako je nečija prijava prihvaćena, tu osobu se o tome obavještava mailom, te mu se šalju podatci potrebni za registraciju (npr. korisničko ime i link za registraciju). Ako prijava nije bila prihvaćena, osoba se o tome isto obavještava e-mail adresom. Korisnici zatim prilikom registracije za dobiveno korisničko ime upisuju lozinku.

2.4 Grupe

Nakon završetka odabira prijava, organizatori određuju broj grupa u koje će sudionici biti razvrstani. Razvrstavanje u grupe izvodi se slučajnim odabirom, no mora postojati mogućnost razmještanja sudionika. Nakon što su formirane grupe potrebno je raspored popuniti s aktivnostima te njima pridružiti grupe.

2.5 Aktivnosti

Organizatori, koji su evidentirani direktno u bazi, definiraju aktivnosti. Svaka aktivnost ima svoje ime, kratki opis i trajanje, a postoji nekoliko tipova aktivnosti:

- aktivnosti na kojima sudjeluje samo jedna grupa,
- aktivnosti na kojima sudjeluje točno N grupa,
- aktivnosti na kojima sudjeluje maksimalno N grupa,
- aktivnosti na kojima nužno sudjeluju sve grupe.

Aktivnosti se mogu izvršavati više puta u raznim vremenima, ali jedna aktivnost uvijek traje jednako te se svaka aktivnost mora izvršavati maksimalno jednom u danom trenutku. Prilikom stvaranja instance aktivnosti potrebno je provjeriti i upozoriti na kršenje sljedećih uvjeta:

- aktivnost se neće preklapati s aktivnošću istog tipa,
- pridružen je minimalno jedan animator,
- pridružen je odgovarajući broj grupa,
- nijedna od pridruženih grupa neće imati konflikte s drugim aktivnostima koje su već navedene,
- nijedna od pridruženih grupa nije već pridružena jednakoj aktivnosti,
- pridruženi animatori neće imati konflikte s drugim aktivnostima na koje su pridruženi.

Prilikom završetka dodavanja aktivnosti treba dodatno provjeriti jesu li sve grupe sudjelovale na svakoj aktivnosti točno jednom. Svaki dan postoje tri aktivnosti koje su nepomične a one su: doručak u 8h, ručak u 12h i večera u 18h. Na tim aktivnostima sudjeluju sve grupe i svi animatori te one traju 1h.

2.6 Početna stranica

Na početnoj stranici prezentirane su osnovne informacije o kampu: vrijeme održavanja, trajanje i aktivnosti. Sudionici i animatori nakon prijave u sustav, prije početka kampa vide samo odbrojavanje do početka kampa i imaju mogućnost kontaktiranja organizatora. Nakon početka prijava (za oba tipa korisnika) vodi na stranicu koja pokazuje njihov raspored ili agendu (odabrati jedno). Animatori moraju vidjeti popis svih grupa, njihovih članova i drugih animatora kao i njihove kontakt podatke, dok sudionici vide iste podatke samo za svoju grupu i animatore. Dodatno, i sudionici i animatori moraju vidjeti popis aktivnosti na kojima su sudjelovali te moraju imati opciju ocjenjivanja aktivnosti (1-10) te ostavljanja kratkog opisa njihovog dojma aktivnosti. Nakon

što je kamp završio, sudionicima i animatorima je potrebno omogućiti ocjenjivanje i ostavljanje vlastitog dojma za cjelokupno iskustvo. Organizatori trebaju imati popis svih povratnih ocjena po aktivnostima te ih moraju moći pretraživati prema sljedećim atributima: korisnik, grupa i/ili aktivnost.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Organizator kampa (naručitelj)
2. Sudionici kampa
3. Animatori
4. Administrator
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) vidjeti osnovne informacije o kampu
 - (b) vidjeti vrijeme održavanja, trajanja i popis aktivnosti
 - (c) prijaviti se na prijavu za sudjelovanje u kampu ili na prijavu za animatora gdje upisuju svoje puno ime, e-mail adresu, broj telefona, datum i godinu rođenja i motivacijsko pismo
 - (d) sudionici koji imaju manje od 18 godina moraju unijeti i broj telefona odgovorne osobe
 - (e) biti obaviješten o tome je li njegova prijava prihvaćena putem e-maila
2. Sudionik (inicijator) može:

Nakon prijave u sustav prije početka kampa:

 - (a) prilikom registracije upisati lozinku za dobiveno korisničko ime
 - (b) vidjeti odbrojanje do početka kampa
 - (c) kontaktirati organizatora

Nakon početka kampa:

- (a) vidjeti stranicu s rasporedom ili agendom
- (b) vidjeti popis članova svoje grupe i animatore i njihove kontakt podatke
- (c) vidjeti popis aktivnosti na kojima su sudjelovali i ocijeniti aktivnost (1-10) i ostaviti kratak opis dojma

Nakon završetka kampa:

- (a) ocijeniti cjelokupno iskustvo i ostaviti vlastiti dojam

3. Animator (inicijator) može:

Nakon prijave u sustav prije početka kampa:

- (a) prilikom registracije upisati lozinku za dobiveno korisničko ime
- (b) vidjeti odbrojanje do početka kampa
- (c) kontaktirati organizatora

Nakon početka kampa:

- (a) vidjeti stranicu s rasporedom ili agendom
- (b) vidjeti popis svih grupa, njihovih članova i popis drugih animatora te njihove kontakt podatke
- (c) vidjeti popis aktivnosti na kojima su animirali i ocijeniti aktivnost (1-10) i ostaviti kratak opis dojma

Nakon završetka kampa:

- (a) ocijeniti cjelokupno iskustvo i ostaviti vlastiti dojam

4. Organizator (inicijator) može:

- (a) zadati početne informacije o kampu (vrijeme održavanja, trajanje, aktivnosti)
- (b) jednostavno definirati aktivnost (ime, kratki opis, trajanje)
- (c) omogućiti prijave za kamp i odrediti vrijeme trajanja prijavi
- (d) vidjeti popis prijavi te ih odbiti ili prihvatiti

- (e) nakon završetka odabira prijava određivati broj grupa u koje će sudionici biti raspoređeni
- (f) popuniti raspored s aktivnostima uz određene uvjete i aktivnostima pridružiti grupe
- (g) vidjeti popis svih povratnih ocjena po aktivnostima
- (h) pretraživati povratne ocjene po atributima korisnik, grupa i/ili aktivnost

5. Administrator (inicijator) može:

- (a) vidjeti popis svih registriranih korisnika
- (b) brisati i dodavati korisnike i davati im ovlaštenja i mijenjati im razinu pristupa (organizator, sudionik, animator)
- (c) slučajnim odabirom rasporediti sudionike u grupe
- (d) premještat i sudionika iz grupe u grupu
- (e) nakon prihvaćene prijave stvoriti korisnički račun i poslati e-mail korisniku
- (f) ako je odbijena prijava obavijestiti korisnika e-mailom
- (g) micati dojmove čije riječi nisu sukladne pravilima korištenja stranice

6. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o aktivnostima i grupama
- (c) pohranjuje krajnje dojmove

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Organizacija kampa

- **Glavni sudionik:** Organizator
- **Cilj:** Postaviti osnovne informacije o kampu
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator mora biti evidentiran u bazi
- **Opis osnovnog tijeka:**
 1. Organizator se prijavljuje kroz web aplikaciju
 2. Organizator kroz sučelje za izradu kampa unosi podatke
 3. Organizator odabire gumb kreacije kampa

UC2 - Kreiranje aktivnosti

- **Glavni sudionik:** Organizator
- **Cilj:** Napraviti novu aktivnost
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Organizator odabire sučelje za kreaciju nove aktivnosti
 2. Organizator odabire detalje nove aktivnosti
 3. Detalji nove aktivnosti se prikazuju organizatoru koji ih može potvrditi te tako završiti proces kreacije

UC3 - Zadavanje trajanja prijave

- **Glavni sudionik:** Organizator
- **Cilj:** Odrediti vrijeme prijave za sudionike i animatore
- **Sudionici:** Baza podataka

- **Preduvjet:** Organizator mora biti evidentiran u bazi
- **Opis osnovnog tijeka:**
 1. Organizator odabire sučelje za određivanje trajanja prijave
 2. Organizator odabire vrijeme trajanja prijave i za koga je prijava namijenjena
 3. Otvara se mogućnost prijave u zadanom terminu

UC4 - Prijava za sudjelovanje

- **Glavni sudionik:** Potencijalni sudionik (neregistrirani korisnik)
- **Cilj:** Prijava za sudjelovanje na kampu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire sučelje za prijavu za sudionika na kamp
 2. Korisnik upisuje tražene podatke
 3. Prijava se prikazuje organizatoru koji dalje može prihvatiti ili odbiti prijavu

UC5 - Potvrda prijave sudionika

- **Glavni sudionik:** Organizator
- **Cilj:** Potvrditi prijavu sudionika
- **Sudionici:** Baza podataka
- **Preduvjet:** Sudionik mora poslati prijavu
- **Opis osnovnog tijeka:**
 1. Organizator prihvaća prijavu
 2. Organizator šalje prihvaćenom sudioniku link za registraciju na e-mail naveden u prijavi

UC6 - Odbijanje prijave sudionika

- **Glavni sudionik:** Organizator
- **Cilj:** Odbiti prijavu sudionika
- **Sudionici:** Baza podataka
- **Preduvjet:** Sudionik mora poslati prijavu
- **Opis osnovnog tijeka:**
 1. Organizator odbija prijavu
 2. Organizator šalje sudioniku obavijest o odbijanju na e-mail naveden u prijavi

UC7 - Prijava za animatore

- **Glavni sudionik:** Potencijalni animator (neregistrirani korisnik)
- **Cilj:** Prijava za posao animatora na kampu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire sučelje za prijavu za animatora na kamp
 2. Korisnik upisuje tražene podatke
 3. Prijava se prikazuje organizatoru koji dalje može prihvatiti ili odbiti prijavu

UC8 - Registracija sudionika

- **Glavni sudionik:** Sudionik
- **Cilj:** Registracija sudionika
- **Sudionici:** Baza podataka
- **Preduvjet:** Prihvaćena prijava od strane organizatora
- **Opis osnovnog tijeka:**
 1. Korisnik dobiva mail sa podacima za registraciju

2. Korisnik upisuje proizvoljnu lozinku na sučelju za registraciju
3. Registracija se evidentira u bazi podataka

UC9 - Razmještaj sudionika po grupama

- **Glavni sudionik:** Organizator
- **Cilj:** Razmjestiti sudionike po grupama
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je završio odabir prijava
- **Opis osnovnog tijeka:**
 1. Organizator određuje broj grupa za sudionike
 2. Grupa svakog sudionika se evidentira u bazi podataka

UC10 - Zahtjev za premještanje

- **Glavni sudionik:** Sudionik
- **Cilj:** Zatražiti premještanje u drugu grupu
- **Sudionici:** Baza podataka
- **Preduvjet:** Sudionici su raspoređeni po grupama
- **Opis osnovnog tijeka:**
 1. Sudionik odabire opciju zahtjeva za premještanje
 2. Sudioniku se izlistaju ostale grupe
 3. Sudionik odabire željenu grupu

UC11 - Premještanje pojedinog sudionika u drugu grupu

- **Glavni sudionik:** Organizator
- **Cilj:** Premjestiti sudionika u drugu grupu
- **Sudionici:** Baza podataka
- **Preduvjet:** Zahtjev za premještanje

- **Opis osnovnog tijeka:**

1. Organizator zaprima zahtjev za premještaj sudionika u drugu grupu
2. Premještaj se evidentira u bazi podataka

UC12 - Punjenje rasporeda i provjera aktivnosti

- **Glavni sudionik:** Organizator
- **Cilj:** Popuniti raspored s aktivnostima i provjeriti kršenje uvjeta aktivnosti
- **Sudionici:** Baza podataka
- **Preduvjet:** Formirane su grupe
- **Opis osnovnog tijeka:**
 1. Aktivnost se neće preklapati s aktivnošću istog tipa
 2. Pridružen je minimalno jedan animator
 3. Pridružen je odgovarajući broj grupa
 4. Nijedna od pridruženih grupa neće imati konflikte s drugim aktivnostima koje su već navedene
 5. Nijedna od pridruženih grupa nije već pridružena jednakoj aktivnosti
 6. Pridruženi animatori neće imati konflikte s drugim aktivnostima na koje su pridruženi

UC13 - Ocjenjivanje aktivnosti

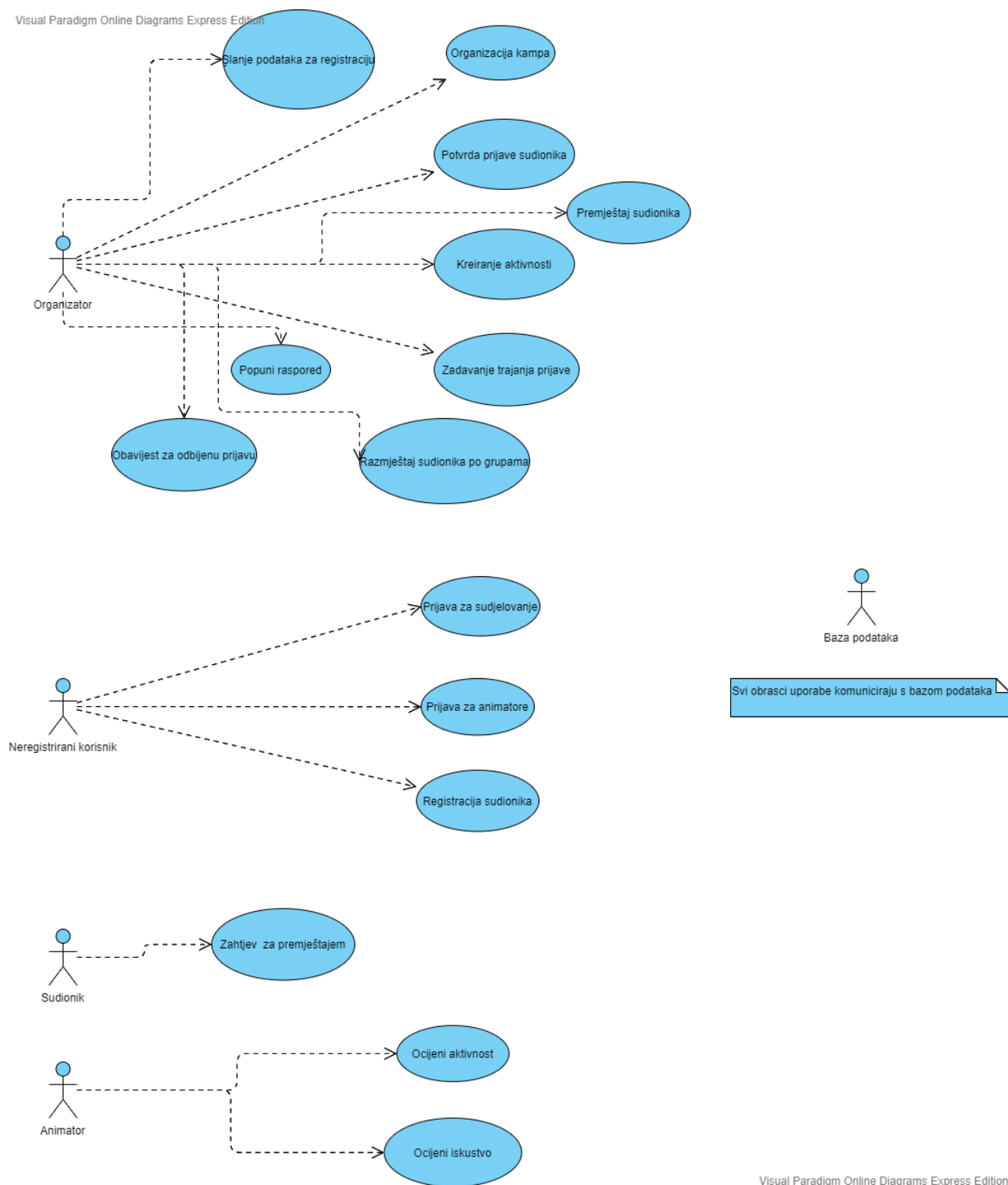
- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Dati povratnu ocjenu za aktivnosti
- **Sudionici:** Baza podataka
- **Preduvjet:** Aktivnost je završila
- **Opis osnovnog tijeka:**
 1. Sudionici i organizatori ocjenjuju aktivnost
 2. Ocjene se evidentiraju u bazi podataka

3. Organizator može pretraživati ocjene

UC14 - Ocjenjivanje cjelokupnog iskustva

- **Glavni sudionik:** Sudionici i animatori
- **Cilj:** Dati povratnu ocjenu za cjelokupno iskustvo
- **Sudionici:** Baza podataka
- **Preduvjet:** Kamp je završio
- **Opis osnovnog tijeka:**
 1. Sudionici i organizatori ocjenjuju cjelokupno iskustvo
 2. Ocjene se evidentiraju u bazi podataka
 3. Organizator može pretraživati ocjene

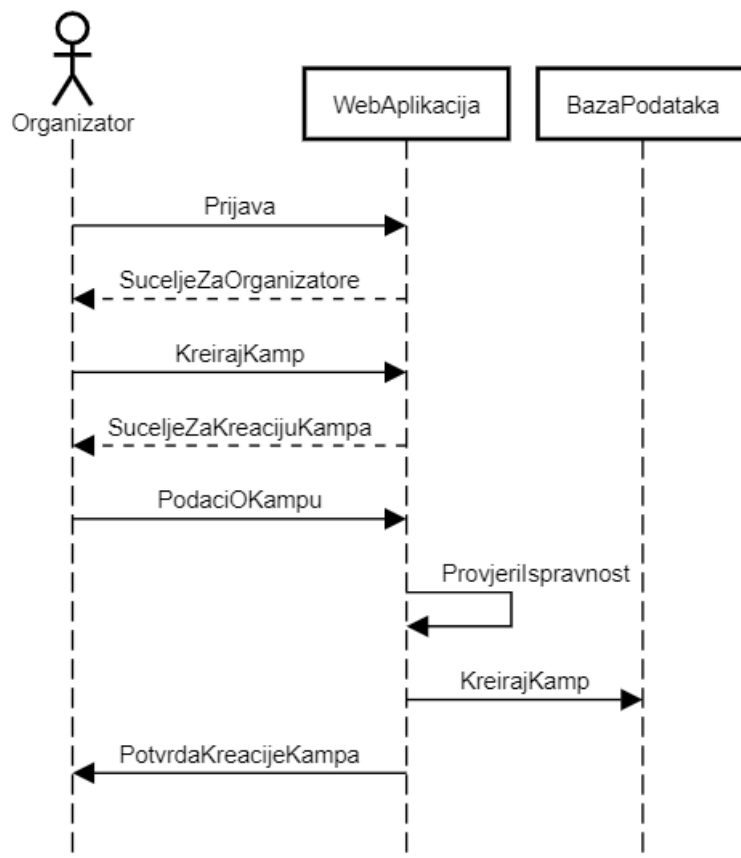
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrazaca uporabe

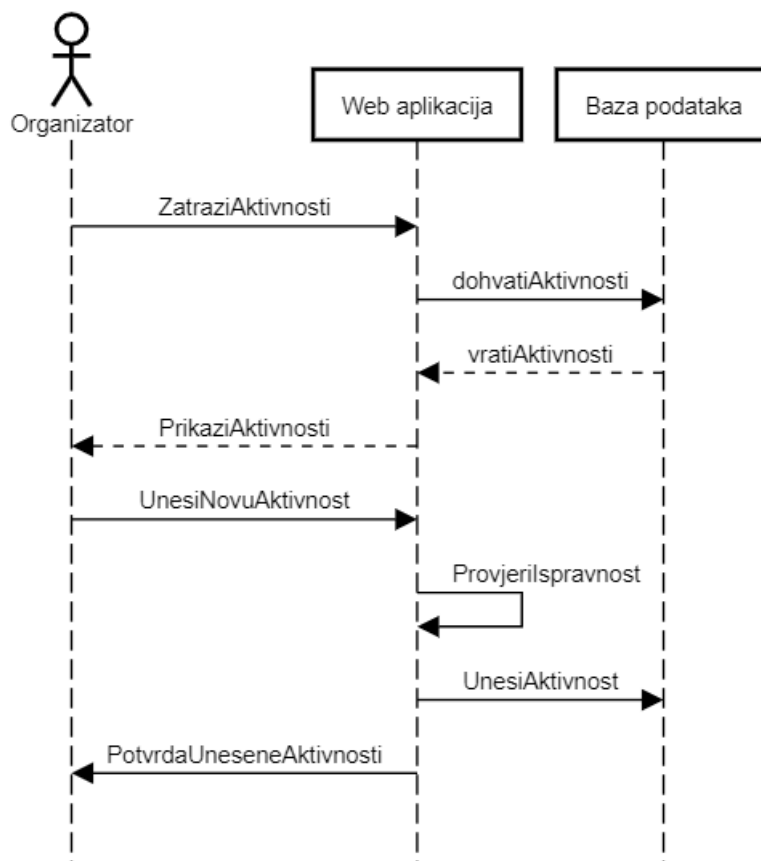
3.1.2 Sekvencijski dijagrami

Obrazac upotrebe UC1 - Organizacija kampa



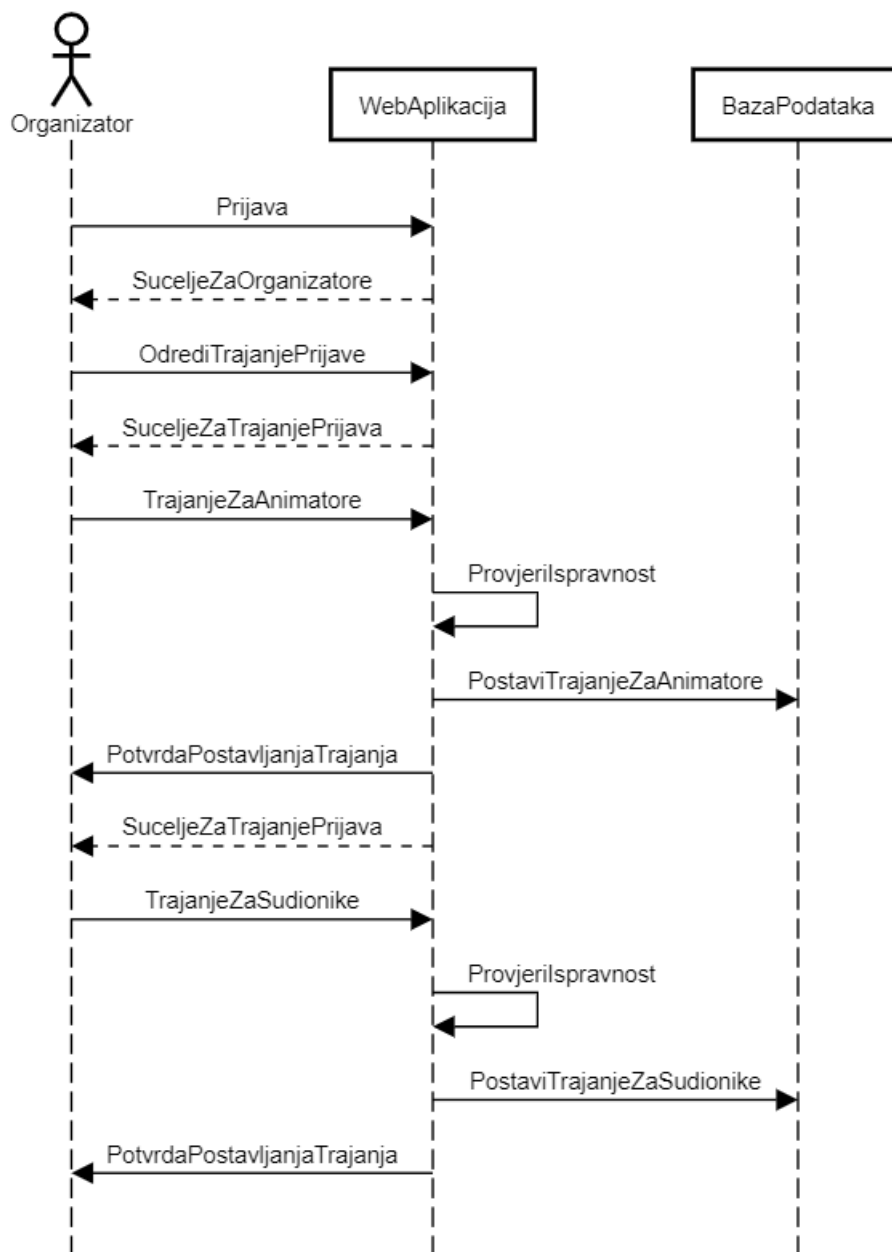
Slika 3.2: Sekvencijski dijagram za UC1

Obrazac upotrebe UC2 - Kreiranje aktivnosti



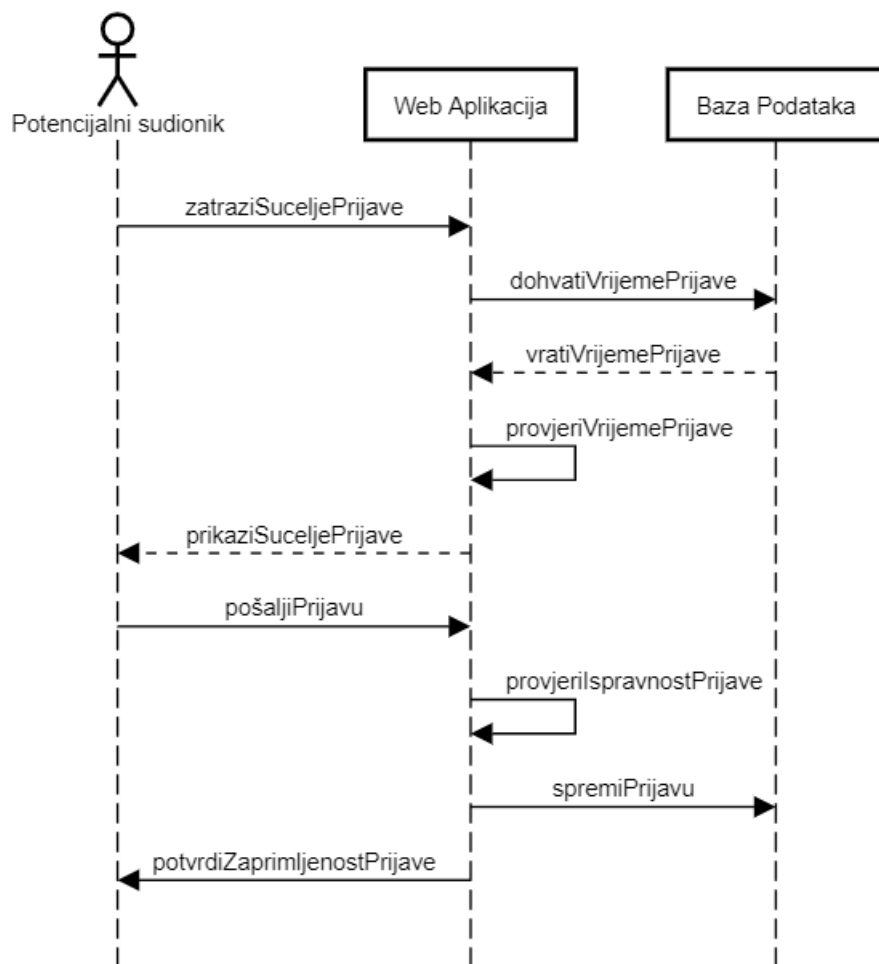
Slika 3.3: Sekvencijski dijagram za UC2

Obrazac upotrebe UC3 - Zadavanje trajanja prijave



Slika 3.4: Sekvencijski dijagram za UC3

Obrazac upotrebe UC4 - Prijava za sudjelovanje



Slika 3.5: Sekvencijski dijagram za UC4

3.2 Ostali zahtjevi

- sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- 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 treba omogućiti rad više korisnika u stvarnom vremenu
- izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- pristup sustavu mora biti omogućen iz javne mreže

4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na tri podsustava:

- web poslužitelj
- web aplikacija
- baza podataka

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Svaka stranica pisana je u kodu, a web preglednik je pretvara u ono što mi vidimo. Dakle, svaki internetski preglednik je prevoditelj. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj osnova je rada web aplikacije. On šalje i prima podatke od mnogostrukih klijenata. Komunikacija između njega i korisnika se odvija preko HTTP (engl. Hyper Text Transfer Protocol) protokola, što je protokol u prijenosu informacija na webu. Poslužitelj je onaj koji pokreće web aplikaciju te joj prosljeđuje zahtjev.

Za obrađivanje željenih zahtjeva koristi se web aplikacija. Ako je potrebno pristupa bazi podataka te preko poslužitelja korisniku vraća odgovor u obliku HTML dokumenta vidljivog u web pregledniku.

Programski jezik koji smo odabrali za izradu naše web aplikacije je JavaScript zajedno sa Bootstrap radnim okvirom te HTML i CSS programske jezike za oblikovanje. Oda-brano razvojno okruženje programske potpore je Visual studio code.

Temelj arhitekture sustava ležat će na MVC (Model-View-Controller) konceptu. Naime, taj koncept ima već napravljene predloške koji nam pomažu u izradi web aplikacije te je podržan od radnog okvira. Velika prednost MVC koncepta je da omogućuje programeru da razvija komponente aplikacije nezavisno jedne o drugima, što olakšava testiranje, traženje grešaka i dodavanje novih funkcionalnosti.

MVC koncept sastoji se od 3 dijela:

- Model – centralni dio sustava koji direktno upravlja podacima, logikom i pravilima sustava. Isto tako prima ulazne podatke Controllera
- View – glavna uloga mu je da prikazuje podatke. Ista informacija može se prikazati na nekoliko različitih načina, poput grafova, tablica i sl.
- Controller – bavi se prilagodbom ulaza koje prosljeđuje Modelu i Viewu te upravlja zahtjevima korisnika i pomoću njih djeluje na ostale sustave.

4.1 Baza podataka

Za potrebe našeg sustava za organizaciju kampa računarstva „Mlade nade” koristimo relacijsku bazu podataka. Relacijska baza sastoji se od relacija, tj. tablica koje sadrže naziv i skup atributa. Ovakva baza nam omogućuje brzu i jednostavnu pohranu i izmjenu podataka te dohvat podataka za daljnju obradu. Dijagram baze olakšava razumijevanje namjene podataka i njihove povezanosti.

Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Osoba
 - Animator
 - Sudionik
 - Organizator
- Aktivnost
 - Aktivnost1
 - AktivnostSve
 - AktivnostMaxN
 - AktivnostN
- Grupa
- Sudjeluje
- Račun
- Dojam
- Prijava
- Kamp

4.1.1 Opis tablica

osoba - ovaj entitet sadrži podatke o osobama koje na bilo koji način sudjeluju u kampu. Sadrži attribute: puno ime osobe, ID osobe, motivacijsko pismo, datum rođenja, broj telefona odgovorne osobe (za sudionike mlađe od 18 godina), e-mail i broj telefona. Generalizacija je entiteta organizator, animator i sudionik. U vezi je one-to-one s entitetom dojam preko atributa Idosobe.

osoba		
Idosobe	INT	identifikacijski broj osobe
punoIme	VARCHAR	ime i prezime osobe
motPismo	VARCHAR	motivacijsko pismo
datumRod	DATE	datum rođenja osobe
brojTelefonaOdgOsobe	VARCHAR	broj telefona odgovorne osobe
Email	VARCHAR	e-mail osobe
brojTel	VARCHAR	broj telefona osobe

animator - ovaj entitet sadrži podatke o animatorima koji sudjeluju u raznim aktivnostima. Specijalizacija je entiteta osoba. Uz attribute entiteta osoba sadrži još i entitet naziv aktivnosti. U vezi je many-to-one s entitetom aktivnost preko atributa nazivAkt.

animator		
Idosobe	INT	identifikacijski broj osobe
nazivAkt	VARCHAR	naziv aktivnosti u kojoj sudjeluje
punoIme	VARCHAR	ime i prezime osobe
motPismo	VARCHAR	motivacijsko pismo
datumRod	DATE	datum rođenja osobe
brojTelefonaOdgOsobe	VARCHAR	broj telefona odgovorne osobe
Email	VARCHAR	e-mail osobe
brojTel	VARCHAR	broj telefona osobe

sudionik - ovaj entitet sadrži podatke o sudionicima kampa. Specijalizacija je entiteta osoba. Uz attribute entiteta osoba sadrži još i entitet naziv grupe. U vezi je many-to-one s entitetom grupa preko atributa nazivGrupa.

Sudionik		
Idosobe	INT	identifikacijski broj osobe
nazivGrupa	VARCHAR	naziv grupe čiji je član
punoIme	VARCHAR	ime i prezime osobe
motPismo	VARCHAR	motivacijsko pismo
datumRod	DATE	datum rođenja osobe
brojTelefonaOdgOsobe	VARCHAR	broj telefona odgovorne osobe
Email	VARCHAR	e-mail osobe
brojTel	VARCHAR	broj telefona osobe

organizator - ovaj entitet sadrži informacije o osobi koja organizira kamp. Specijalizacija je entiteta osoba. Uz attribute entiteta osoba sadrži još i entitet nazivGrupa.

organizator		
Idorganizatora	INT	identifikacijski broj organizatora
punoIme	VARCHAR	ime i prezime osobe
motPismo	VARCHAR	motivacijsko pismo
datumRod	DATE	datum rođenja osobe
brojTelefonaOdgOsobe	VARCHAR	broj telefona odgovorne osobe
Email	VARCHAR	e-mail osobe
brojTel	VARCHAR	broj telefona osobe

aktivnost - ovaj entitet sadrži podatke o aktivnostima koje se odvijaju u kampu. Generalizacija je entiteta aktivnost1, aktivnostSve, aktivnostMaxN i aktivnostN. Sadrži attribute naziv aktivnosti, opis i trajanje aktivnosti. U vezi je many-to-one s entitetom sudjeluje preko atributa nazivAkt.

aktivnost		
nazivAkt	VARCHAR	Naziv aktivnosti
opis	VARCHAR	Kratak opis aktivnosti
trajanje	INTERVAL	Trajanje aktivnosti

aktivnost1 - ovaj entitet sadrži podatke o aktivnostima u kojima sudjeluje samo jedna grupa sudionika. Specijalizacija je entiteta aktivnost. Sadrži sve attribute entiteta aktivnost tj. naziv aktivnosti, opis i trajanje.

aktivnost1		
nazivAkt	VARCHAR	naziv aktivnosti
opis	VARCHAR	kratak opis aktivnosti
trajanje	INTERVAL	trajanje aktivnosti

aktivnostSve - ovaj entitet sadrži podatke o aktivnostima u kojima sudjeluju sve postojeće grupe sudionika. Specijalizacija je entiteta aktivnost. Sadrži sve attribute entiteta aktivnost tj. naziv aktivnosti opis i trajanje.

aktivnostSve		
nazivAkt	VARCHAR	naziv aktivnosti
opis	VARCHAR	kratak opis aktivnosti
trajanje	INTERVAL	trajanje aktivnosti

aktivnostMaxN - ovaj entitet sadrži podatke o aktivnostima u kojima sudjeluje maksimalno N grupa sudionika. Specijalizacija je entiteta aktivnost. Sadrži sve attribute entiteta aktivnost tj. naziv aktivnosti, opis i trajanje.

aktivnostMaxN		
nazivAkt	VARCHAR	naziv aktivnosti
opis	VARCHAR	kratak opis aktivnosti
trajanje	INTERVAL	trajanje aktivnosti

aktivnostN - ovaj entitet sadrži podatke o aktivnostima u kojima sudjeluje točno N grupa sudionika. Specijalizacija je entiteta aktivnost. Sadrži sve attribute entiteta aktivnost tj. naziv aktivnosti, opis i trajanje.

aktivnostN		
nazivAkt	VARCHAR	naziv aktivnosti
opis	VARCHAR	kratak opis aktivnosti
trajanje	INTERVAL	trajanje aktivnosti

grupa - ovaj entitet opisuje grupe sudionika koji sudjeluju u aktivnostima. Sadrži atribut naziv grupe. U vezi je many-to-one s entitetom sudjeluje preko atributa naziv-Grupa.

grupa		
nazivGrupa	VARCHAR	naziv grupe sudionika

sudjeluje - ovaj entitet sadrži sve informacije o odnosu grupa i aktivnosti, odnosno daje informacije o tome koja grupa sudjeluje u kojoj od aktivnosti. Sadrži atribut naziv grupe te je preko tog atributa u vezi one-to-many s entitetom sudionik.

sudjeluje		
nazivGrupa	VARCHAR	naziv grupe sudionika
nazivAkt	VARCHAR	naziv aktivnosti

račun - ovaj entitet sadrži podatke o korisničkom računu koji je se izrađuje na temelju prihvaćene prijave za svakog sudionika i animatora. Sadrži attribute korisnickoIme, lozinka, poveznica i Idosobe. U vezi je one-to-one s entitetom osoba preko atributa Idosobe.

račun		
korisnickoIme	VARCHAR	korisničko ime osobe
Idosobe	INT	identifikacijski broj osobe
lozinka	VARCHAR	lozinka korisničkog računa
poveznica	VARCHAR	poveznica za registraciju

dojam - ovaj entitet sadrži informacije o osvrtu i ocjeni koju sudionici i animatori ostavljaju nakon završetka kampa. Sadrži entitete ID osobe, ocjena, komentar. U vezi je one-to-one s entitetom osoba preko atributa Idosobe.

dojam		
Idosobe	INT	identifikacijski broj osobe
ocjena	INT	ocjena kampa
komentar	VARCHAR	kratki osvrt

prijava - ovaj entitet sadrži podatke o prijavama za sudjelovanje u kampu. Sadrži attribute prijavaZa (određuje prijavljuje li se osoba za sudionika ili animatora), vrijeme početka prijave i vrijeme trajanja prijave.

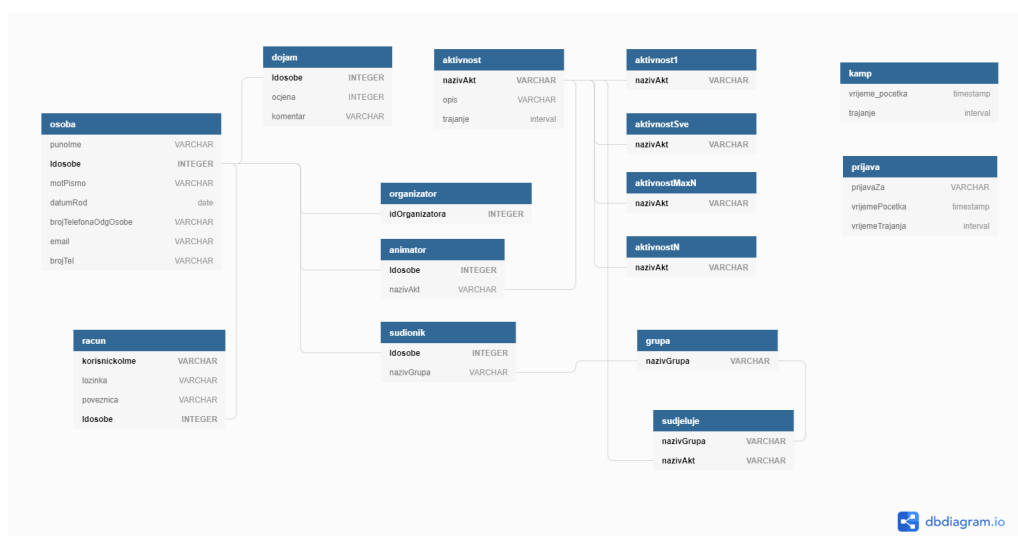
prijava		
prijavaZa	VARCHAR	prijava za sudionika/animatora
vrijemePocetka	TIMESTAMP	vrijeme početka prijave
vrijemeTrajanja	INTERVAL	trajanje prijave

kamp - ovaj entitet sadrži informacije o kampu za računarstvo za koji se izrađuje aplikacija. Sadrži attribute vrijemePocetka i vrijemeTrajanje.

kamp		
vrijemePocetka	TIMESTAMP	vrijeme početka kampa
vrijemeTrajanja	INTERVAL	trajanje kampa

4.1.2 Dijagram baze podataka

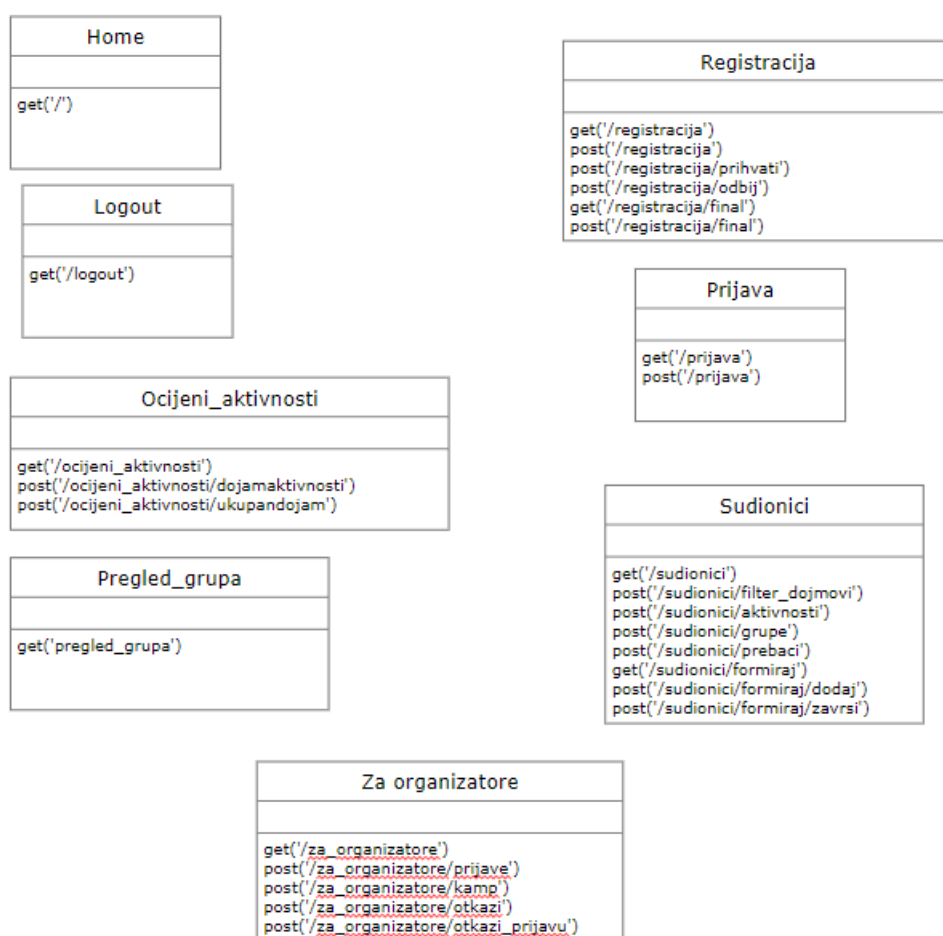
Dijagram baze podataka



Slika 4.1: ER dijagram baze podataka

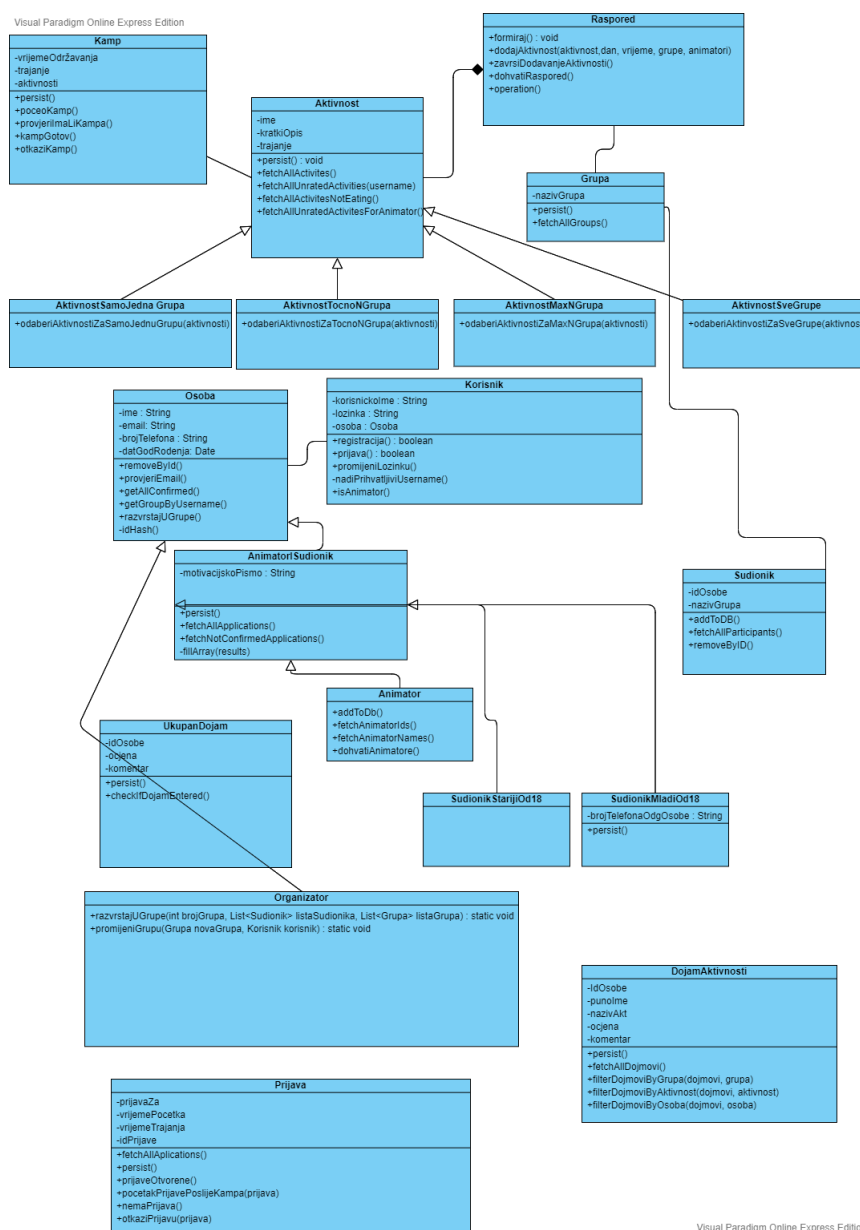
4.2 Dijagram razreda

Na slikama 4.2 i 4.3 prikazani su razredi koji pripadaju backend dijelu MVC arhitekture. Razredi prikazani na slici 4.2 nasljeđuju Controller razred. Metode implementirane u tim razredima manipuliraju s podacima, a oni su dohvaćeni pomoću metoda implementiranih u Model razredima. Metode implementirane u Controller razredima vraćaju JSON datoteke s html status kodom. Prikazane su samo ovisnosti između razreda koji pripadaju istom dijelu dijagrama. Iz naziva i tipova atributa u razredima možemo zaključiti vrstu ovisnosti među različitim razredima.



Slika 4.2: Dijagram razreda - dio Controllers

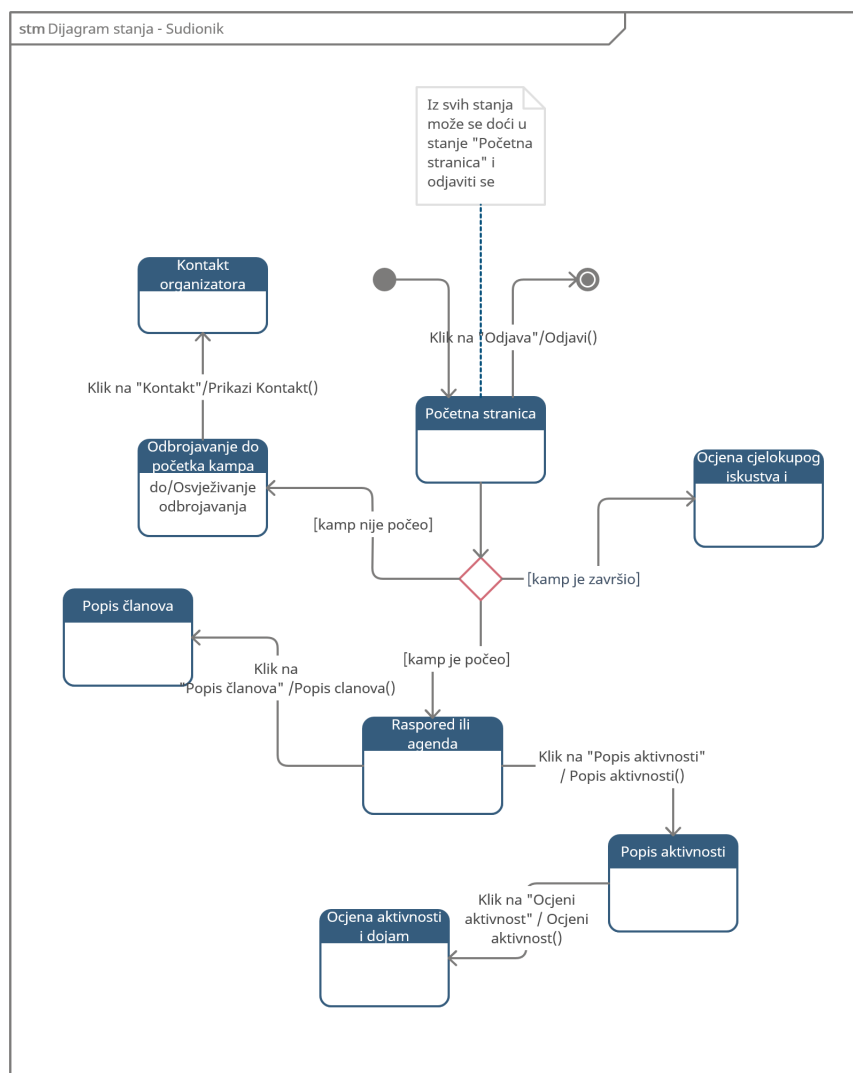
Model razredi preslikavaju strukturu baze podataka u aplikaciji. Metode koje su implementirane direktno komuniciraju s bazom podataka i tako vraćaju tražene podatke. Razred Korisnik predstavlja neregistriranog korisnika koji se može registrirati u sustav unoseći tražene informacije. Također ako je već registriran, može se prijaviti u sustav. Nakon prijave on može biti registriran kao Sudionik ili kao Animator, i ovisno o tome može koristiti drugačije funkcionalnosti stranice. Razred Organizator predstavlja osobu koja može razmještati ostale korisnike po grupama i koja ima najveće ovlasti.



Slika 4.3: Dijagram razreda - dio Models

4.3 Dijagram stanja

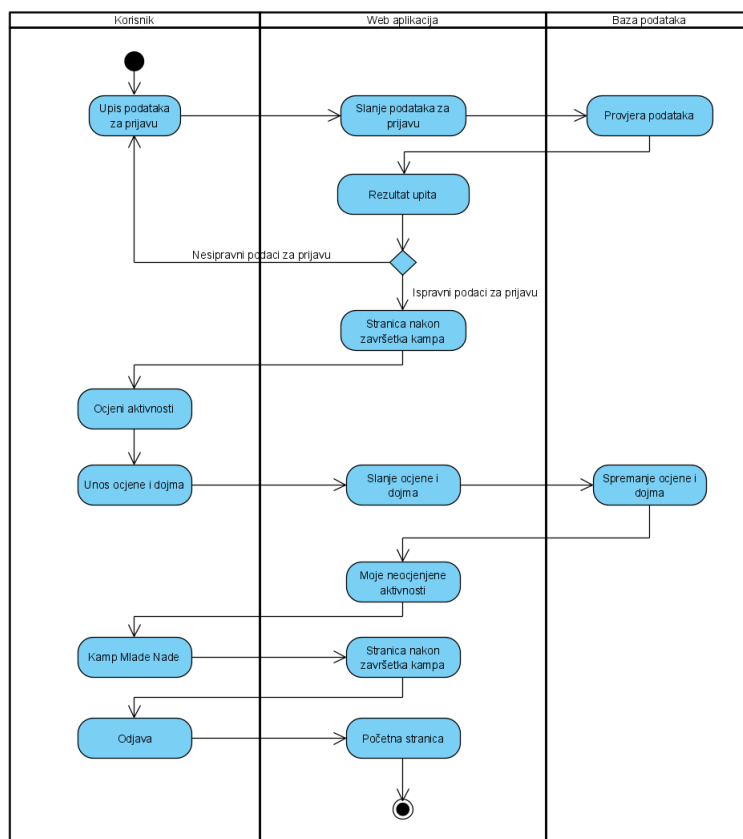
Dijagram stanja je dijagram koji prikazuje određena stanja objekata i prijelaze jednog stanja u drugo temeljene na određenim događajima. Na slici 4.4 prikazan je dijagram stanja za prijavu otprije registriranog korisnika – Sudionika i o prikazu njegovih opcija ovisno o tome je li kamp počeo, traje, ili je završio. Nakon prijave, ako kamp još nije počeo sudioniku se prikazuje odbrojavavanje do početka kampa i klikom može kontaktirati organizatora. Nakon prijave i nakon početka kampa, sudionik vidi svoj osobni raspored ili agendu, te klikom na *Popis članova* može vidjeti popis članova i njihove kontakte. Naposljetku, nakon prijave i nakon završetka kampa, sudionik može ocijeniti svoje cjelokupno iskustvo i ostaviti dojam.



Slika 4.4: Dijagram stanja

4.4 Dijagram aktivnosti

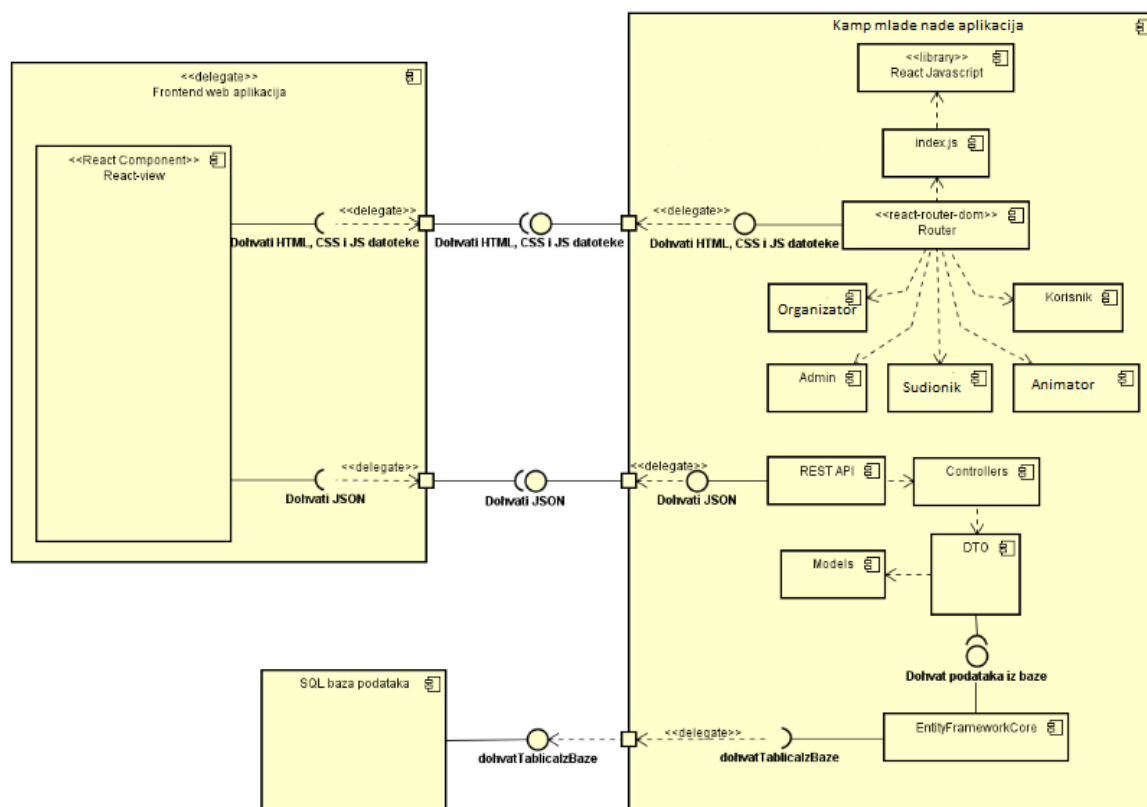
Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili podatkovnog toka te modeliranje poslovnih procesa. To je u osnovi dijagram toka koji predstavlja tok iz jedne u drugu aktivnost. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na slici 4.5 prikazan je proces ostavljanje dojma i ocjene korisnika za određenu aktivnost. Korisnik se prijavi u sustav te s popisa odabere jednu od aktivnosti u kojima sudjeluje. Zatim mu se prikazuje okvir za upis dojma i ocjene. Nakon upisivanja dojma i ocjene, podaci se spremaju u bazu podataka i korisnik se može odjaviti.



Slika 4.5: Dijagram aktivnosti

4.5 Dijagram komponenti

Sustavu se pristupa putem dva sučelja. Preko sučelja za dohvat HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju frontendu. Router je komponenta koja na upit s url-om određuje koja će se datoteka poslužiti na sučelje. Frontend se sastoji od niza JavaScript datoteka koje su raspoređene u simboličke cjeline i ovise o React biblioteci iz koje dohvaćaju gotove komponente (gumbi, forme i sl.). Backend je baziran na arhitekturi *Repository Service Controller*. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju backendu. Razredi koji implementiraju sučelje Service zaduženi su za dohvaćanje podataka iz baze podataka pomoću SQL upita uz pomoć Repositoryja. Podaci koji su pristigli iz baze šalju se MVC arhitekturi u obliku DTO (Data transfer object). React-view komponenta preko dostupnih sučelja komunicira s web aplikacijom te osvježava prikaz i dohvaća nove podatke.



Slika 4.6: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacije MS Teams¹.

Za izradu UML dijagrama korišten je alat Visual Paradigm Online², a kao sustav za upravljanje izvornim kodom Git³, a udaljeni repozitorij projekta je dostupan na web platformi GitLab⁴.

Kao razvojno okruženje korišten je Visual Studio Code⁵, integrirano razvojno okruženje (kratica IDE) tvrtke Microsoft. Koristi se prvenstveno za razvoj računalnih programa za operacijski sustav Windows, kao i za web-stranice, web-aplikacije, mobilne aplikacije i web-usluge. Visual studio za razvoj softvera koristi Microsoftove platforme kao što su Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight.

Baza podataka se nalazi na pgAdmin 4 platformi⁶.

¹<https://www.microsoft.com/hr-hr/microsoft-365/microsoft-teams>

²<https://online.visual-paradigm.com>

³<https://git-scm.com>

⁴<https://gitlab.com/>

⁵<https://code.visualstudio.com/>

⁶<https://www.pgadmin.org/>

5.2 Ispitivanje programskog rješenja

5.2.1 Ispitivanje komponenti

*U okviru paketa **Mocha** i **Chai** proveden je unit testing. Testovi su provedeni nad nekim funkcionalnostima razreda Grupa, Kamp, Sudionik, MailSender i Osoba.*

Testovi kamp klase

```
const assert = require('assert');
const Kamp = require('../models/Kamp');

describe('Kamp Test', () => {
  it('treba vratiti undefined jer nema kampa', () => {
    assert.equal(undefined, Kamp.provjeriImaLiKampa().pocetak);
  });

  it('treba vratiti 2', async () => {
    let kamp = new Kamp("2021-01-14T18:47", 2, []);
    await kamp.persist();
    let result = await Kamp.provjeriImaLiKampa();
    assert.equal(result.trajanje, 2);
    await Kamp.otkaziKamp();
  });
});
```

Testovi grupa

```
const assert = require('assert');
const Grupa = require('../models/Grupa');

describe('Grupa Test', () => {
  it('treba vratiti naziv grupe', async () => {
    let grupa = new Grupa("grupa 1")
    await grupa.persist();
    let result = await Grupa.fetchAllGroups();
    assert.equal(result[0].naziv, "grupa 1");
  });
});
```

Testovi pošiljatelja mailova

```
const assert = require('assert');
const MailSender = require('../models/MailSender');

describe('Mail Sender Test', () => {

  it('treba vratiti poruku prihvaćanja', () => {
    let mailOptions = MailSender.send("korisnik@random.com", "krandom", "accepted");
    assert.equal(mailOptions.subject, 'Čestitamo! Primljeni ste u naš kamp!');
  });

  it('treba vratiti poruku odbijanja', () => {
    let mailOptions = MailSender.send("korisnik2@random.com", "krandom2", "denied");
    assert.equal(mailOptions.subject, 'Vaš zahtjev je obijen');
  });
});
```

Test za random raspored sudionika u grupe

```
const expect = require('chai').expect;
const assert = require('assert');

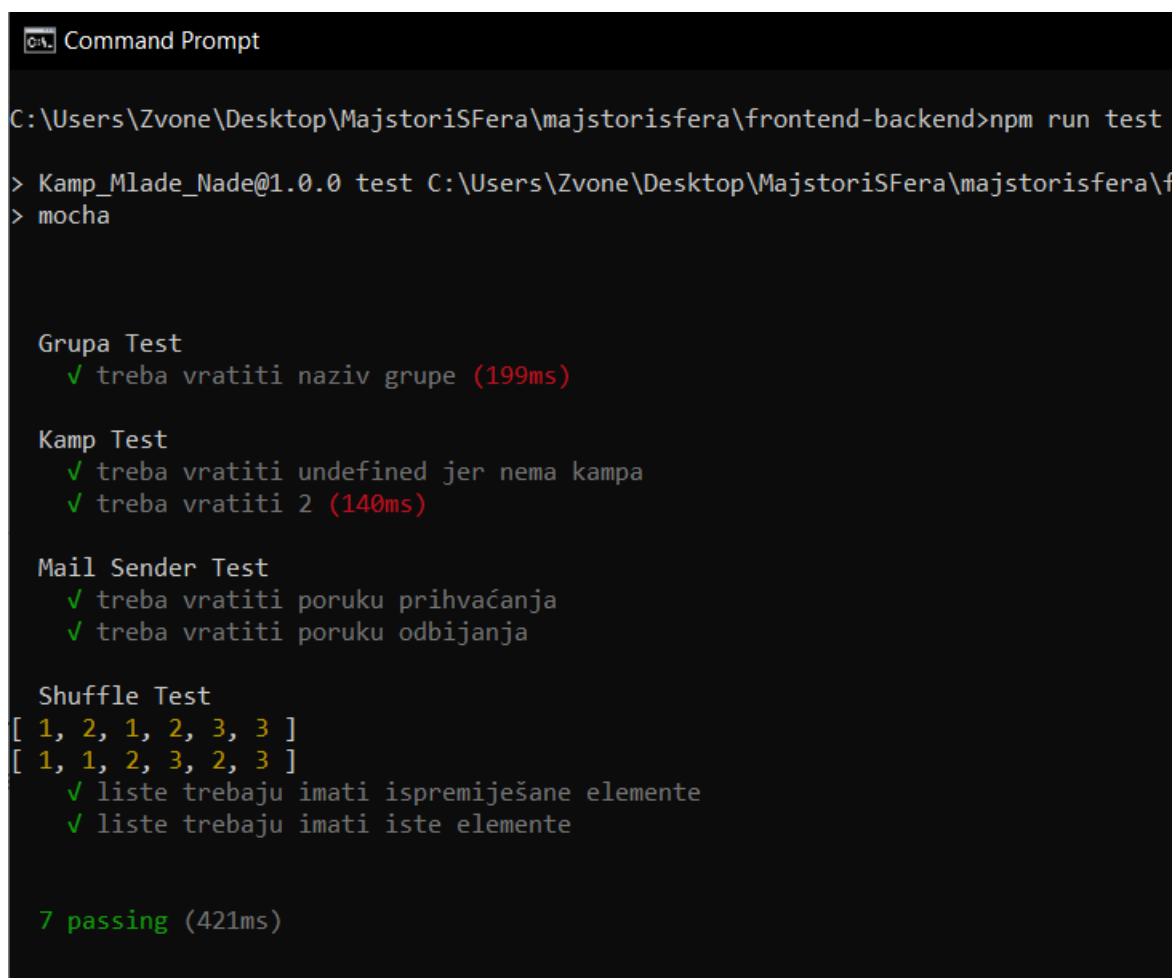
describe('Shuffle Test', () => {

  it('liste trebaju imati ispremiješane elemente', async () => {
    let array = [1, 2, 1, 2, 3, 3];
    let arrayShuffled = [1, 2, 1, 2, 3, 3];
    arrayShuffled = shuffle(arrayShuffled);
    console.log(array);
    console.log(arrayShuffled);
    assert.notStrictEqual(array, arrayShuffled);
  });

  it('liste trebaju imati iste elemente', () => {
    let array = [1, 2, 1, 2, 3, 3];
    let arrayShuffled = [1, 2, 1, 2, 3, 3];
    arrayShuffled = shuffle(arrayShuffled);
    expect(array).to.have.same.members(arrayShuffled);
  });
});

function shuffle(array) {
  for (var i = array.length - 1; i > 0; i--) {
    var j = Math.floor(Math.random() * (i + 1));
    var temp = array[i];
    array[i] = array[j];
    array[j] = temp;
  }
  return array
}
```

Screenshot pokretanja i uspjeha testova



```
Command Prompt

C:\Users\Zvone\Desktop\MajstoriSFera\majstorisfera\frontend-backend>npm run test

> Kamp_Mlade_Nade@1.0.0 test C:\Users\Zvone\Desktop\MajstoriSFera\majstorisfera\frontend-backend
> mocha

Grupa Test
  ✓ treba vratiti naziv grupe (199ms)

Kamp Test
  ✓ treba vratiti undefined jer nema kampa
  ✓ treba vratiti 2 (140ms)

Mail Sender Test
  ✓ treba vratiti poruku prihvatanja
  ✓ treba vratiti poruku odbijanja

Shuffle Test
  [ 1, 2, 1, 2, 3, 3 ]
  [ 1, 1, 2, 3, 2, 3 ]
  ✓ liste trebaju imati ispremiješane elemente
  ✓ liste trebaju imati iste elemente

7 passing (421ms)
```

Slika 5.1: Pokrenuti testovi

5.2.2 Ispitivanje sustava

Pomoću radnog okvira Selenium provedeno je ispitivanje funkcionalnosti organizacije kampa i organizacije prijave za kamp, prijave osobe na kamp, potvrde prijave od strane organizatora te registracije korisnika. Za izradu ispitnih slučajeva korišten je alat Selenium IDE.

Ispitni slučaj 1: Organizacija kampa i prijava za kamp

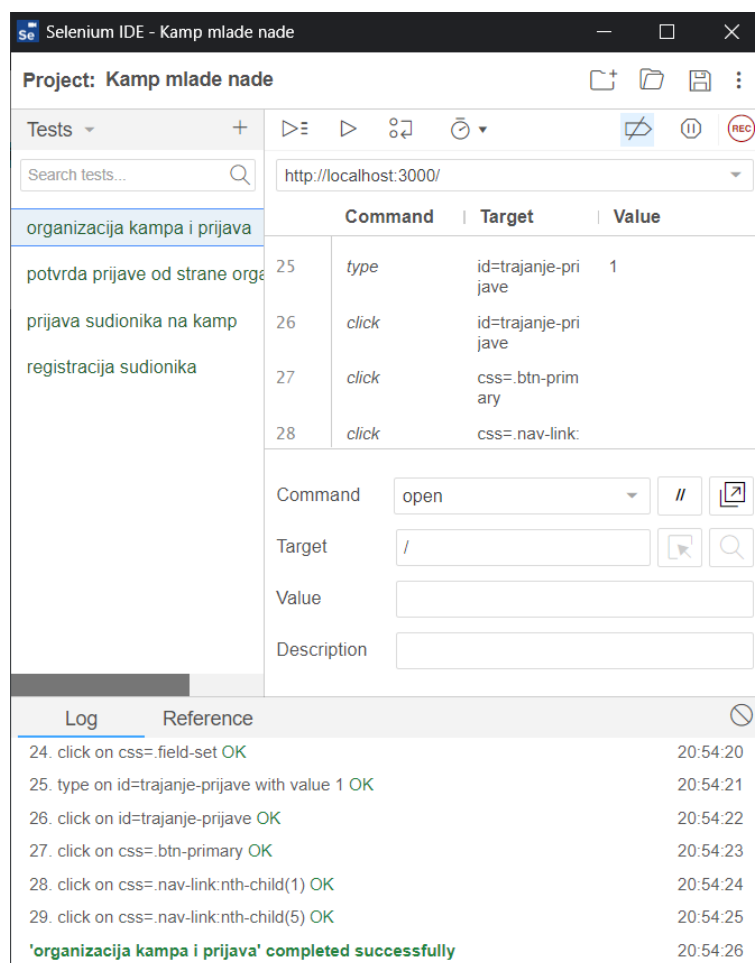
Ulaz:

- *Prijava kao organizator.*
- *Organiziranje kampa u određeno vrijeme.*
- *Organiziranje prijava u određeno vrijeme.*

Očekivani rezultat:

- *Na početnoj stranici ispisano je vrijeme početka i vrijeme trajanja kampa.*

Rezultat: *Očekivani rezultat je zadovoljen. Aplikacija je prošla test.*



Slika 5.2: Organizacija kampa i prijava

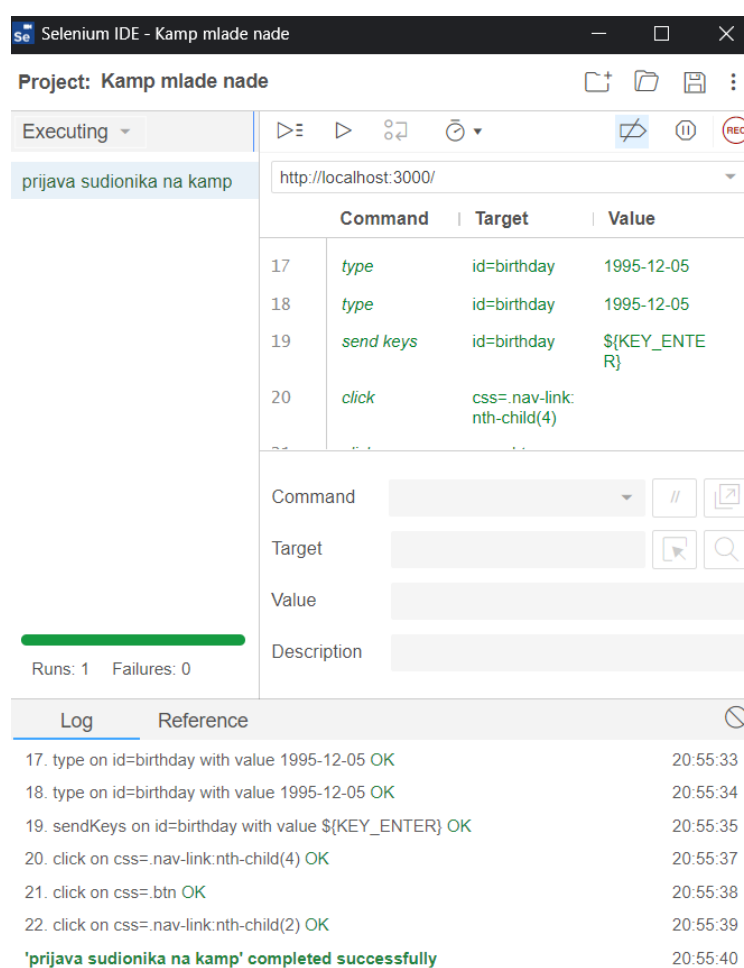
Ispitni slučaj 2: Prijava sudionika na kamp**Ulaz:**

- *Unos podataka korisnika u sklopu prijave za sudionika na kampu.*

Očekivani rezultat:

- *Prijava evidentirana u bazi i samim tim se prikazuje organizatoru na stranici sa prijavama.*

Rezultat: Očekivani rezultat je zadovoljen. Aplikacija je prošla test.



Slika 5.3: Organizacija kampa i prijava

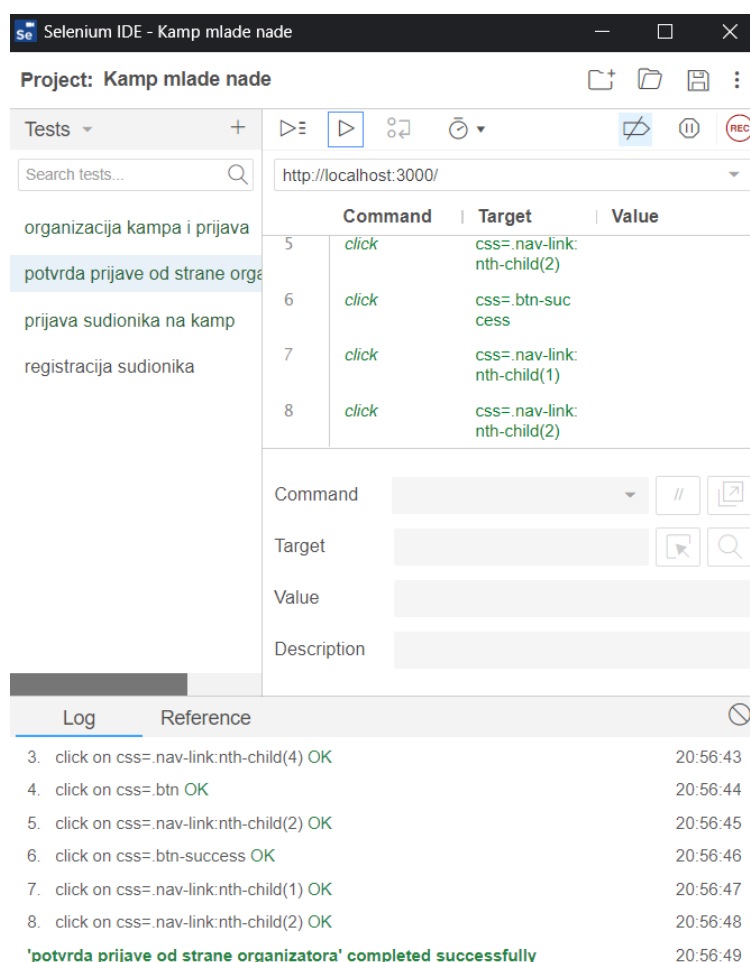
Ispitni slučaj 3: Potvrda prijave od strane organizatora**Ulaz:**

- *Prijava korisnika u kojoj organizator vidi njegove podatke*
- *Odabir hoće li prihvatiti ili odbiti prijavu.*

Očekivani rezultat:

- *Na stranici na kojoj organizator inače može vidjeti prijave više nema prijave koja je potvrđena.*
- *Korisniku je poslan mail sa linkom i korisničkim imenom za registraciju.*

Rezultat: Očekivani rezultat je zadovoljen. Aplikacija je prošla test.



Slika 5.4: Potvrda prijave od strane organizatora

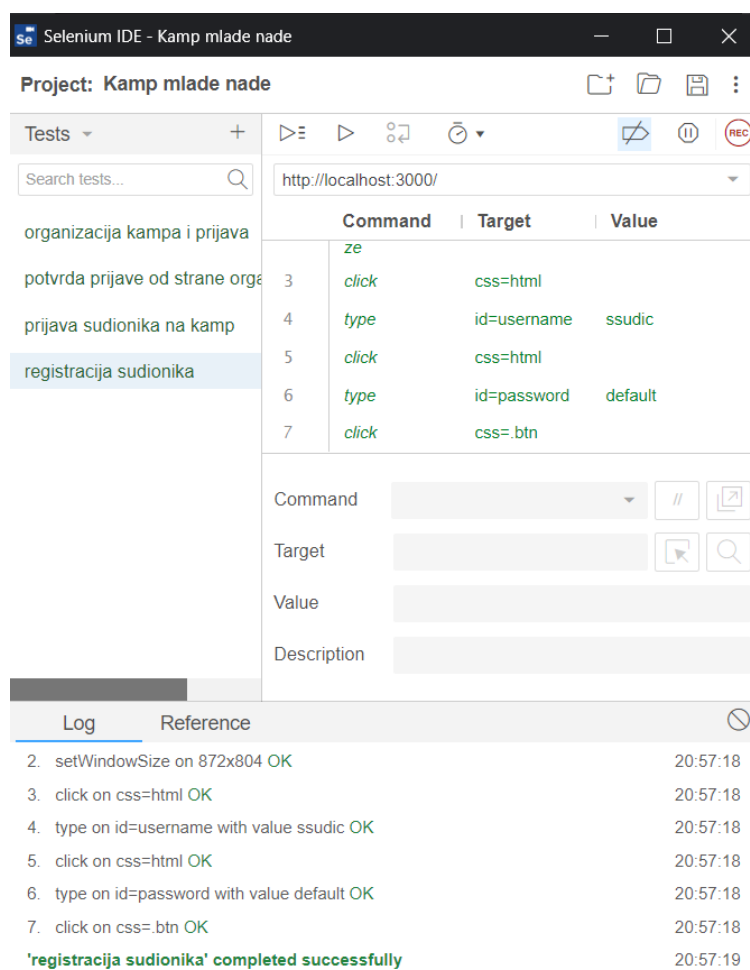
Ispitni slučaj 4: Registracija sudionika**Ulaz:**

- *Web lokacija za kompletiranje registracije.*
- *Korisničko ime za registraciju.*

Očekivani rezultat:

- *Sudionik kampa uspješno se ulogirao.*
- *Sudionik kampa može vidjeti svoje korisničko ime u gornjem desnom kutu.*
- *Sudioniku se otvaraju opcije koje na web stranici ima svaki sudionik.*

Rezultat: Očekivani rezultat je zadovoljen. Aplikacija je prošla test.



Slika 5.5: Registracija sudionika

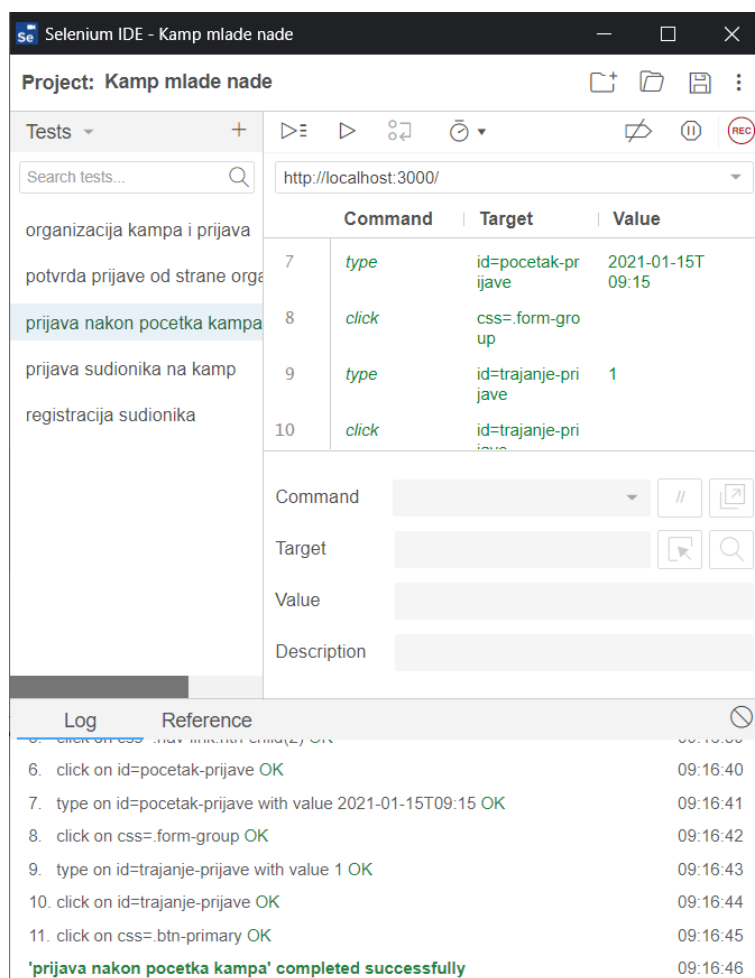
Ispitni slučaj 5: Pokušaj organiziranja prijave nakon početka kampa**Ulaz:**

- Datum i vrijeme početka kampa
- Odabir početka prijava u korisničkom sučelju.

Očekivani rezultat:

- Organiziranje prijave se odbija uz prikladnu poruku.
- Organizator je preusmjeren na početnu stranicu

Rezultat: Očekivani rezultat nije zadovoljen. Pokušajem organiziranja prijave nakon početka kampa događa se *ReferenceError*. Aplikacija nije prošla test.

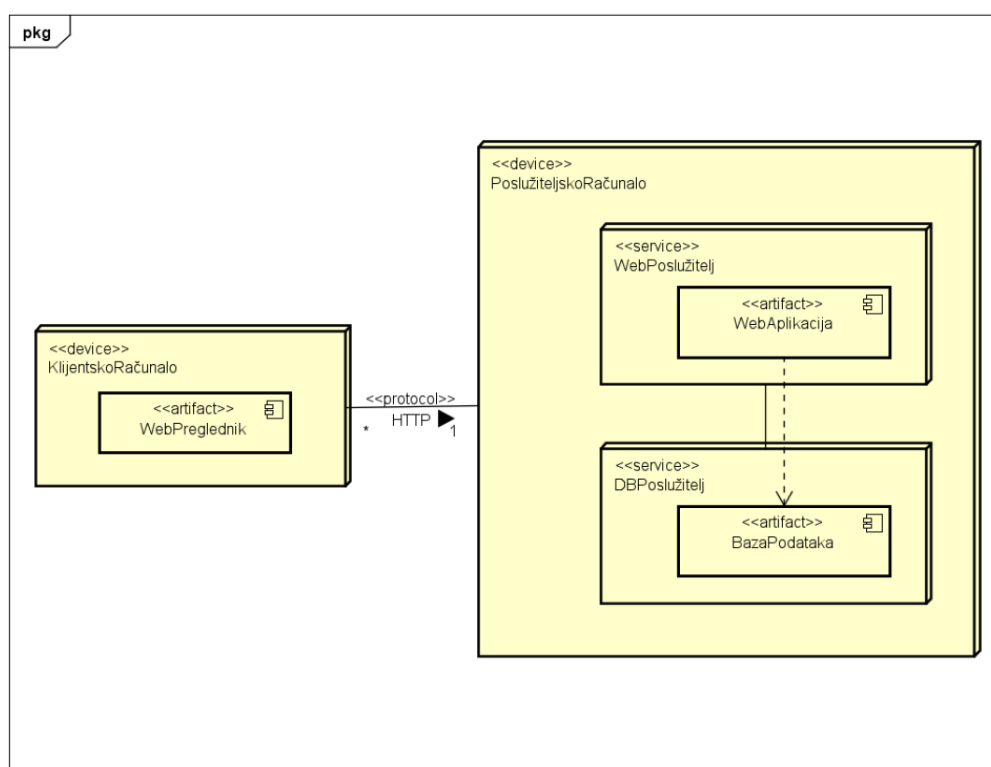


Slika 5.6: Organiziranje prijave nakon početka kampa

5.3 Dijagram razmještaja

Dijagrami razmještaja prikazuju računalne resurse potrebne za ispravno funkcioniranje sustava te njihove međusobne odnose: stvarne uređaje, komponente programske potpore koje se na njima izvršavaju i veze između njih. Dijagram razmještaja je statički strukturni UML-dijagram.

Sustav je temeljen na arhitekturi *klijent-poslužitelj*. Na poslužiteljskom računalu nalaze se web poslužitelj (na kojem se nalazi web aplikacija) i poslužitelj baze podataka (na kojem se nalazi baza podataka). Web poslužitelj i poslužitelj baze podataka su u vezi ovisnosti, točnije promjene na web poslužitelju uzrokuju promjene u bazi podataka. Na klijentskom se računalu nalazi web preglednik putem kojeg se pristupa web poslužitelju, odnosno web aplikaciji. Komunikacija između klijentskog i poslužiteljskog računala odvija se preko HTTP veze.



Slika 5.7: Dijagram razmještaja

5.4 Upute za puštanje u pogon

5.4.1 Instalacija Git-a

Potrebno je instalirati Git korisničko sučelje prema uputama na stranici⁷ i na računalu kreirati novu mapu za aplikaciju. Nakon instalacije potrebno je otvoriti *Git Bash* i pozicionirati se u direktorij namijenjen za datoteke aplikacije. To se može učiniti tako da se desnim klikom na željenu mapu odabere opcija *Git Bash Here*. Sljedeći korak je prijava ili registracija na sustav Gitlab putem internet stranice i postavljanje podataka o korisničkom računu na lokalni Git repozitorij. To se obavlja upisivanjem sljedećih naredbi u Git Bash:

- *git config user.email email*
- *git config user.name username*

U mapu je potrebno klonirati projekt odabirom opcije *Clone* u gitlab repozitoriju te kopirati link ispod oznake *Clone with HTTPS*. U Git Bash je potrebno upisati *git clone link*. Kod projekta je sada praćen u lokalnom Git repozitoriju.

5.4.2 Instalacija Heroku CLI-a

Heroku sučelje naredbenog retka (CLI) olakšava izradu i upravljanje Heroku aplikacijama izravno iz terminala. To je bitan dio upotrebe Herokua. Potrebno je preuzeti instalacijski paket⁸ minimalne verzije 7.0.x., pokrenuti instalacijski paket i slijediti upute na ekranu. Nakon instalacije potrebno je postaviti varijable okruženja sljedećim putem: *Control Panel - System and Security - System - Advanced System Settings*. Otvorit će se prozor *System properties* te je na kartici *Advanced* potrebno pritisnuti gumb *Environment Variables*. U prozoru *Environment Variables* u sekciji *User Variables* potrebno je odabrati *Path* i kliknuti gumb *Edit*. U prozoru *Edit environment variable* odabrati *New* te upisati putanju do bin datoteke u folderu heroku, koja se najčešće nalazi u *C://Program Files*. Ako je instalacija uspješna, u terminalu je moguće pokrenuti *heroku -v* što će rezultirati prikazom verzije.

⁷<https://git-scm.com/book/en/v2/Getting-Started-Installing-Git>

⁸<https://devcenter.heroku.com/articles/heroku-cli>

5.4.3 Kreiranje Heroku udaljenog repozitorija

Prvi je korak registracija i prijava u sustav Heroku. Nakon toga, potrebno je otvoriti terminal i pozicionirati se u git lokalni repozitorij, odnosno mapu kreiranu za projekt. Sljedeći korak je pokretanje naredbe ***heroku login*** i nakon toga ***heroku plugins:install java***. Sljedeća naredba koju je potrebno pokrenuti je ***heroku create*** u naredbenom retku (terminalu). Ta naredba stvara novu, praznu aplikaciju na Heroku poslužitelju povezanu s Git repozitorijem. Poželjno je nakon toga pokrenuti naredbu ***git remote -v*** što će rezultirati potvrdom da je udaljeni repozitorij pod imenom *heroku* postavljen u aplikaciji.

Potrebno je kreirati dvije aplikacije na Herokuu: jednu za backend i jednu za frontend. Jedna aplikacija je jedan poslužitelj.

5.4.4 Postavljanje baze podataka

Za postavljanje baze podataka potrebno je u kartici *Resources* u postavkama aplikacije dodati *Heroku Postgres* dodatak. Nakon ovog koraka na serveru će se konfigurirati prazna baza podataka. U kartici *Settings* na postavkama baze podataka potrebno je odabrati opciju *View Credentials*. Tada dobivamo podatke za spajanje na bazu koji su potrebni za konfiguraciju backenda.

5.4.5 Postavljanje backenda aplikacije na Heroku poslužitelj

Prije postavljanja backenda na poslužitelj potrebno je konfigurirati podatke za spajanje na bazu. U datoteci *application.properties* unutar projekta potrebno je upisati podatke dobivene na sučelju Heroku. Nakon konfiguracije konekcije, potrebno je pokrenuti naredbu ***maven install***. Time će se izgraditi aplikacija. Unutar target direktorija unutar projekta kreirat će se *.war* datoteka. Tu datoteku potrebno je prenijeti na Heroku poslužitelj.

Koristeći naredbeni redak potrebno je postaviti se u direktorij projekta te pokrenuti naredbu ***heroku war:deploy*** ime *war* datoteke ***-app*** naziv aplikacije kreirane na Heroku. Nakon toga će na Heroku poslužitelj biti postavljen backend aplikacije.

5.4.6 Postavljanje frontenda aplikacije na Heroku poslužitelj

Nakon inicijalizacije udaljenog repozitorija na heroku, potrebno je pokrenuti naredbu ***npm install*** za instalaciju svih paketa potrebnih *node.js*-u.

Da bi frontend aplikacije bio postavljen na Heroku poslužitelj, izvede se naredba ***git push heroku master*** koja *pusha* kod iz lokalnog repozitorija grane master na heroku udaljeni repozitorij. Ovu naredbu potrebno je pokrenuti svaki puta kada se želi postaviti novija verzija aplikacije na Heroku.

Make some changes to the code you just cloned and deploy them to Heroku using Git.

```
$ git add .  
$ git commit -am "make it better"  
$ git push heroku master
```

Slika 5.8: Postavljanje nove verzije aplikacije na Heroku

Na kraju, naredba ***heroku open*** služi za pokretanje aplikacije u pretpostavljenom pretraživaču.

6. Zaključak i budući rad

Rad na ovom projektu u sklopu predmeta Programsko inženjerstvo pružio nam je bogato iskustvo, kako u organizaciji rada u timu, tako i u upotrebi brojnih alata koje smo koristili te samoj izradi web aplikacije. Naš je zadatak bio u nekoliko mjeseci napraviti web aplikaciju kampa *Mlade nade* namijenjenu organizatorima, kako bi im olakšali posao organizacije, ali i budućim sudionicima i animatorima, koji na ovaj način puno brže i lakše podnose prijave.

Na samom početku prvog ciklusa bilo je potrebno opisati i razraditi projektni zadatak. Iako smo u ranoj fazi mislili da bi lakše bilo opisivati funkcionalnosti usputno, za vrijeme rada na aplikaciji, pokazalo se da je detaljna razrada funkcionalnosti prije same izrade bila ne samo korisna, nego i nužna za jednostavniju organizaciju i rad. Dijagrami i obrasci uporabe izrađeni u prvom ciklusu (dijagrami obrazaca uporabe, sekvencijski dijagrami, dijagrami razreda i baze podataka) bili su temelj za daljnji rad članova tima zaduženih za backend i frontend. Dobro izrađena dokumentacija pomogla nam je izbjeći nedoumice oko razvoja funkcionalnosti i olakšala raspodjelu posla među članovima tima. U prvoj fazi projekta neki od nas su se prvi puta susreli s korištenim alatima i programskim jezicima odabranima za rad. Savladali smo osnove korištenja sustava Git, LaTeX i izradu dokumentacijskih dijagrama te prvi puta u praksi koristili bazu podataka.

U drugoj fazi projekta naišli smo na veće izazove. Kako većina nas nije imala iskustva u radu s korištenim alatima, bili smo primorani samostalno učiti kako bismo ostvarili zadani cilj. Bilo je potrebno dokumentirati ostale UML dijagrame i napisati popratnu dokumentaciju kako bi korisnici mogli lakše koristiti našu aplikaciju i kako bismo olakšali rad na unapređenju iste.

Naša komunikacija ostvarena je čestim sastancima i zajedničkim radom što se pokazalo vrlo efikasnim. Uvidjeli smo da je za rad u timu sastavljenom od ovako velikog broja članova nužna dobra komunikacija i informiranost svih članova grupe o napretku projekta. Shvatili smo važnost dobre organizacije u smislu raspodjele posla među članovima tima te vremena potrebnog da se zadaci izvrše. Da bi tim funkcionirao kao cjelina, jako je važno da svaki član tima odradi svoj dio posla savjesno i na vrijeme, kako bi se izbjegao nepotrební angažman ostalih članova tima. Sva stečena iskustva i znanja koristit ćemo i nadograđivati u budućim projektima.

Popis literature

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

3.1	Dijagram obrazaca uporabe	16
3.2	Sekvencijski dijagram za UC1	17
3.3	Sekvencijski dijagram za UC2	18
3.4	Sekvencijski dijagram za UC3	19
3.5	Sekvencijski dijagram za UC4	20
4.1	ER dijagram baze podataka	28
4.2	Dijagram razreda - dio Controllers	29
4.3	Dijagram razreda - dio Models	30
4.4	Dijagram stanja	31
4.5	Dijagram aktivnosti	32
4.6	Dijagram komponenti	33
5.1	Pokrenuti testovi	38
5.2	Organizacija kampa i prijava	40
5.3	Organizacija kampa i prijava	41
5.4	Potvrda prijave od strane organizatora	42
5.5	Registracija sudionika	43
5.6	Organiziranje prijave nakon početka kampa	44
5.7	Dijagram razmještaja	45
5.8	Postavljanje nove verzije aplikacije na Heroku	48

7. Dodatak: Prikaz aktivnosti grupe

7.1 Dnevnik sastajanja

1. sastanak

- Datum: 05. listopada 2020.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić
- Teme sastanka:
 - (a) prijedlozi mogućih vlastitih tema projekta
 - (b) rasprava o znanjima i vještinama

2. sastanak

- Datum: 07. listopada 2020.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić
- Teme sastanka:
 - dogovor podjele posla
 - uvod u projekt i organizaciju

3. sastanak

- Datum: 14. listopada 2020.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić, E.Vušak
- Teme sastanka:
 - sastanak s asistentom
 - rasprava o dobivenom zadatku

4. sastanak

- Datum: 28. listopada 2020.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić

- Teme sastanka:
 - definiranje funkcionalnosti aplikacije
 - rasprava o dizajnu aplikacije

5. sastanak

- Datum: 04. studenoga 2020.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić, E.Vušak
- Teme sastanka:
 - revizija napravljene dokumentacije
 - tehnička pitanja

6. sastanak

- Datum: 11. studenoga 2020.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić, E.Vušak
- Teme sastanka:
 - demonstracija funkcionalnosti i dokumentacije

7. sastanak

- Datum: 16. prosinca 2020.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić
- Teme sastanka:
 - podjela novih zadataka

8. sastanak

- Datum: 08. siječnja 2021.
- Prisustvovali: K.Kovačević, M.Magdalenić, M.Palčić, Z.Rezo, I.Sabolić, I.Šokčević, F.Vučić, E.Vušak
- Teme sastanka:
 - demonstracija funkcionalnosti
 - tehnička pitanja

7.2 Tablica aktivnosti

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							
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>							

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

7.3 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.