

Oblikovanje programske potpore  
Ak. god. 2019./2020.

Festive  
Dokumentacija, Rev. 2

Grupa: *KulEkipa*  
Voditelj: *Sanja Krznarić*

Datum predaje: 16.1.2020.

Nastavnik: *Hrvoje Nuić*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>5</b>
<b>3 Specifikacija programske potpore</b>	<b>9</b>
3.1 Funkcionalni zahtjevi . . . . .	9
3.1.1 Obrasci uporabe . . . . .	11
3.1.2 Sekvencijski dijagrami . . . . .	23
3.2 Ostali zahtjevi . . . . .	27
<b>4 Arhitektura i dizajn sustava</b>	<b>28</b>
4.1 Baza podataka . . . . .	30
4.1.1 Opis tablica . . . . .	31
4.1.2 Dijagram baze podataka . . . . .	36
4.2 Dijagram razreda . . . . .	37
4.2.1 Dijagram razreda Firebase usluga . . . . .	37
4.2.2 Dijagram razreda paketa <i>authentication</i> . . . . .	40
4.2.3 Dijagram razreda paketa <i>organizator</i> . . . . .	42
4.2.4 Dijagram razreda paketa <i>voditelj</i> . . . . .	43
4.2.5 Dijagram razreda paketa <i>izvođač</i> . . . . .	45
4.2.6 Dijagram razreda paketa <i>common</i> . . . . .	47
4.3 Dijagram stanja . . . . .	50
4.4 Dijagram aktivnosti . . . . .	52
4.5 Dijagram komponenti . . . . .	53
<b>5 Implementacija i korisničko sučelje</b>	<b>55</b>
5.1 Korištene tehnologije i alati . . . . .	55
5.2 Ispitivanje programskog rješenja . . . . .	56
5.2.1 Ispitivanje komponenti . . . . .	56
5.2.2 Ispitivanje sustava . . . . .	62
5.3 Dijagram razmještaja . . . . .	67

5.4 Upute za puštanje u pogon . . . . .	68
<b>6 Zaključak i budući rad</b>	<b>70</b>
<b>Popis literature</b>	<b>72</b>
<b>Indeks slika i dijagrama</b>	<b>74</b>
<b>Dodatak: Prikaz aktivnosti grupe</b>	<b>75</b>

# 1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodataka	Autori	Datum
0.1	Napravljen predložak dokumentacije.	Krznarić	12.10.2019.
0.2	Dodan opis projektnog zadatka.	Krznarić, Kožul	12.10.2019.
0.2.1	Dodani funkcionalni zahtjevi.	Kožul	23.10.2019.
0.2.2	Dodan prvi dio opisa obrazaca uporabe.	Krznarić	24.10.2019.
0.2.3	Dodan drugi dio opisa obrazaca uporabe.	Bartaković	24.10.2019.
0.2.4	Dodan treći dio opisa obrazaca uporabe.	Šimić	25.10.2019.
0.3	Dodani dijagrami obrazaca uporabe, sekvencijski dijagrami i ostali zahtjevi.	Gibanica	26.10.2019.
0.3.1	Ispravljeni opisi obrazaca uporabe, dijagrami obrazaca uporabe i sekvencijski dijagrami.	Gibanica	29.10.2019.
0.3.2	Ispravljeni opisi i dijagrami obrazaca uporabe.	Gibanica	7.11.2019.
0.4	Dodan opis arhitekture baze podataka	Jurišić	6.11.2019.
0.4.1.	Dodan dijagram baze podataka	Jurišić	7.11.2019.
0.4.2.	Proširen opis projekta	Šimić	8.11.2019.
0.4.3.	Dorađeni funkcionalni zahtjevi i obrasci uporabe	Krznarić	11.11.2019.
0.4.4.	Ažurirana tablica aktivnosti	Krznarić, Kožul	12.11.2019.
0.4.5.	Dorađeni opis arhitekture i dijagram baze podataka	Jurišić	14.11.2019.
0.4.6.	Dodana korištена literatura	Krznarić	14.11.2019.
0.5.1.	Dodano korištene tehnologije	Bartaković	07.01.2020.
0.5.2.	Dodan dijagram razmještaja	Bartaković	08.01.2020.
0.5.3.	Dodan dijagram stanja	Šimić	09.01.2020.
0.5.4.	Dodan dijagram aktivnosti	Šimić	10.01.2020.

Rev.	Opis promjene/dodataka	Autori	Datum
0.5.5.	Dodani sastanci i aktivnost, ispravljene neke stavke	Krznarić	11.01.2020.
0.5.6.	Dodan dijagram komponenti	Šimić	13.01.2020.
0.5.7.	Dodano ispitivanje sustava	Krznarić, Šimić	16.01.2020.
0.5.8.	Dodano ispitivanje komponentni	Krznarić, Šimić	16.01.2020.
0.5.9.	Dodani grafovi i osvježene tablice aktivnosti	Krznarić, Šimić	16.01.2020.

## 2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje mobilne aplikacije "Festive" koja će korisniku omogućiti da lako sudjeluje u procesu planiranja, organizacije i održavanja raznih festivala i festivalskih događaja.

Prilikom pokretanja aplikacije, korisniku je omogućena prijava u sustav putem vlastite email adrese i lozinke, odnosno korisniku se omogućuje registracija ako takav račun već ne postoji.

Uspješno organiziranje te funkcioniranje jednog festivala leži u dobroj podjeli zadataka i poslova te koordinaciji ljudi uključenih u njihovo izvršavanje. Potrebno je odlučiti tko vodi festival i smislja njegovu temu, što se sve može raditi na festivalu i tko sve može sudjelovati, kako bi se ostvarila izgradnja objekata potrebnih za uspješno funkcioniranje festivala i što bolje iskustvo ljudi koji su ga došli posjetiti. Ova aplikacija omogućava efikasno organiziranje tako da se svaki dio festivala dijeli na nekoliko događaja.

Događaji mogu biti razni; od držanja štandova do različitih nastupa.

Svaki događaj se dijeli na više poslova koji se mogu odradivati ili uzastopno ili istodobno, a svaki posao se razlaže na jednu ili više djelatnosti koje se moraju napraviti.

Jednostavna za upotrebu, aplikacija Festive omogućava da se, bilo tko sa željom, mogućnostima i sposobnostima za sudjelovanje, prijavi i sudjeluje u ostvarivanju festivala.

Svakom novom, neregistriranom korisniku pruža se mogućnost odabira željene uloge na festivalu. Postoje četiri uloge: osnovne; voditelj, organizator i izvođač te posebna uloga administratora. Svaka uloga podrazumijeva različitu razinu ovlasti i odgovornosti u provedbi festivala. Za uspješno kreiranje računa korisnik pri registraciji treba navesti sljedeće podatke:

- *korisničko ime*
- *lozinka*
- *ime*

- *prezime*
- *email adresa*
- *broj mobitela*
- *slika*

Registrirani korisnik dobiva pristup svom računu i omogućene su mu dodatne funkcionalnosti ovisno o ulozi za koju se prijavio. Svim registriranim korisnicima omogućeno je ispisati propusnicu za festival (ili više njih u slučaju izvođača) koja se sastoji od slike, imena i prezimena korisnika, jedinstvenor QR koda te naziva i loga festivala u kojem sudjeluje.

*Administrator* ima vlastito sučelje pomoću kojeg upravlja bazom podataka te potvrđuje ili odbija novoprijavljene voditelje, organizatore te izvođače.

*Voditelj* u sklopu svog računa može stvarati nove festivale te ih naknadno uređivati ukoliko su još uvijek aktivni. Voditelj prvo mora biti potvrđen od strane administratora kako bi mogao koristiti funkcionalnosti dodijeljene profilu voditelja. Da bi voditelj mogao napraviti novi festival mora popuniti obrazac za izradu festivala. Obrazac je definiran nazivom festivala, kratkim opisom o temi, logotipom i predviđenim trajanjem. Aktivnim festivalima voditelj može dodavati nove događaje te im pridjeljivati organizatore. Događaji se otvaraju obrascem koji je definiran nazivom događaja, opisom, prostorijom održavanja te vremenom početka i završetka događaja.

Voditelj za svaki događaj otvara licitaciju proizvoljnog trajanja kako bi omogućio prijavu organizatorima te na temelju prijava odabire organizatora kojeg smatra najkvalificiranim. Po završetku događaja voditelj može ocijeniti sposobnost organizatora tako da mu ostavi komentar na profil.

*Organizator* se putem licitacije prijavljuje za organiziranje događaja na aktivnim festivalima. Prvo treba biti potvrđen od strane administratora da bi mogao koristiti sve ovlasti.

Organizatore biraju voditelji festivala prema njihovim kvalifikacijama, recenzijama te ocjenom koje mogu naći na profilu organizatora.

Organizator mora za događaje koji su mu dodijeljeni odrediti koje je sve poslove potrebno obaviti za pojedini događaj. Poslove onda treba poredati u redoslijed kojim moraju biti obavljeni. Za svaki posao organizator mora otvoriti licitaciju na

koju se prijavljuju izvođači.

Svaka licitacija sadrži ime festivala za koji se posao obavlja, kratki opis posla te podatke koji ispunjava izvođač koji se prijavljuje za licitaciju.

Licitacija je otvorena točno jedan dan. Pri isteku licitacije, izvođač s najmanjom cijenom automatski se bira. Organizator može produžiti licitaciju u trajanju od jednog dana. Nije dozvoljeno produživanje trajanja licitacije više od jednom.

Organizator može sudjelovati u organiziranju više događaja na festivalu no ne smije biti zadužen za dva događaja koji se odvijaju u isto vrijeme.

Nakon obavljenog posla, organizator može ostaviti komentar na profilu izvođača.

*Izvođač* mora biti prvo potvrđen od strane administratora te nakon toga dobiva pravo na korištenje mnogih mogućnosti koje nudi njegova uloga.

Izvođač na svoj profil treba navesti sve poslove za koje je sposobljen. Također ima pristup pregledu dostupnih licitacija za događaje potrebne za ostvarivanje na aktivnim festivalima. Može se prijaviti za neograničen broj licitacija u ovisnosti o svojim interesima i sposobnostima.

Prilikom prijave na licitaciju izvođač popunjava obrazac u kojem upisuje podatke o predviđenoj cijeni za obavljanje posla, komentar u kojem može, ali i ne mora navesti svoje posebne vještine ili zahtjeve, broj potrebnih osoba za što efikasnije odrađivanje posla i procijenjeno trajanje da bi se cijeli posao u potpunosti napravio.

U svakom trenutku izvođač ima pristup svim otvorenim licitacijama na koje je sposobljen, a još se nije prijavio niti predao svoju ponudu.

Na natječaju za licitaciju pobjeđuje izvođač s ponuđenom najpovoljnijom tj. najnižom cijenom.

Kako bi se olakšala organizacija i spriječio pristup neprijavljenim ljudima za sudjelovanje u festivalu svaka uključena osoba mora sa sobom imati propusnicu kojom dokazuje svoju dozvolu za svojom participacijom.

Propusnice svaki korisnik može isprintati pomoću svog profila u aplikaciji.

Propusnica je standardne veličine 10x7cm te se na njoj nalazi slika korisnika, njegovo ime, prezime te logo festivala. Nadalje, za izvođača propusnica sadrži i vrijeme i prostoriju u kojoj on i njegov tim obavljaju posao.

Iskaznica također sadrži jedinstveni QR kod u kojem je zapisan MD5 hash korisničkog imena i festivala, a u slučaju izvođača, i redni broj posla za koji je zadužen

i redni broj osobe koja je prijavljena kao dio tima za obavljanje posla.

Sve informacije o festivalima, događajima na njima, licitacijama te korisnicima aplikacije se pohranjuju u Firebase bazu podataka.

Svim podacima samo administrator ima pristup dok ostali korisnici mogu vidjeti samo one podatke koji su potrebni za odrađivanje njihovih uloga radi uspješnog organiziranja festivala.

Voditelj može vidjeti profile organizatora i izvođača. Organizator može otvoriti profile izvođača dok izvođač može vidjeti samo vlastitu stranicu.

Radi transparentnosti, voditelj ima pristup svim događajima i licitacijama koji se tiču festivala za koji je zadužen.

Aplikacija ima posebnu komponentu koja će omogućiti korisnicima što kvalitetniju organizaciju-sustav ocjenjivanja.

Pri kraju svakog festivala voditelj može svim organizatorima koji su sudjelovali dodijeliti komentar kojim ocjenjuje njihov rad.

Isto tako organizatori mogu ocijeniti rad i sposobnost izvođača te ostavivši komentar na njihovom profilu. Komentari će se prikazivati na profilu organizatora tj. izvođača, ispod obavljenih licitacija.

Ova funkcionalnost će omogućiti voditeljima, a i izvođačima da odaberu što podobnije i sposobnije ljude kako bi organizacija festivala bila što uspješnija.

Unutar aplikacije postoji još nekoliko funkcionalnosti koje bi mogle olakšati organizaciju festivala poput kalendarja kojim svaki korisnik može vidjeti svoju zauzetost te prema tome rasporediti događaje u kojima će sudjelovati kako mogu što bolje participirati u organizaciji festivala prema svojim mogućnostima.

## 3. Specifikacija programske potpore

### 3.1 Funkcionalni zahtjevi

Dionici:

1. Voditelj
2. Organizator
3. Izvođač
4. Administrator
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
  - (a) registrirati se u sustav svojim korisničkim imenom, lozinkom, imenom, prezimenom, slikom, e-mail adresom i brojem mobitela
  - (b) nakon registracije odabrati za koju se ulogu na festivalu želi prijaviti (dostupne uloge: voditelj, organizator i izvođač)
2. Voditelj (inicijator) može:
  - (a) dodati više festivala (naziv, opis, logotip, trajanje)
  - (b) dodati događaje na festivalu (naziv, opis, prostorija održavanja, vrijeme održavanja)
  - (c) pristupiti profilu organizatora i izvođača
  - (d) pristupiti licitacijama i prijavama unutar njegovog festivala
  - (e) ispisati propusnicu
3. Organizator (inicijator) može:
  - (a) izraditi i promijeniti redoslijed izvršavanja poslova za određeni događaj
  - (b) dodati, i produljiti licitaciju za posao
  - (c) pristupiti profilu izvođača
  - (d) komentirati izvođača kojeg je zaposlio za određeni posao na izvođačevom profilu

(e) ispisati propusnicu

4. Izvođač (inicijator) može:

- (a) odabrati svoju vještinu u kojoj je specijaliziran iz ponuđenih djelatnosti ili napisati svoju
- (b) pristupiti pregledu svih poslova otvorenih za licitaciju za koje je specijaliziran
- (c) prijaviti se na licitaciju (cijena, komentar, broj radnika potrebnih, procjena trajanja) i biti odabran od strane organizatora
- (d) ispisati broj propusnica jednak broju osoba prijavljenih za održivanje posla

5. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o festivalima, događajima, licitacijama, radnicima i poslovima

6. Administrator (inicijator) može:

- (a) prihvati registraciju korisnika u sustavu

### 3.1.1 Obrasci uporabe

#### Opis obrazaca uporabe

##### UC1 - Registracija

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun za pristup aplikaciji
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju registracije
  2. Korisnik bira između uloge voditelja, organizatora i izvođača
  3. Korisnik unosi tražene podatke
  4. Obavijest o čekanju na potvrdu od administratora
- **Opis mogućih odstupanja:**
  - 3.a Upisana e-mail adresa ili korisničko ime je već u uporabi
    1. Sustav javlja da je e-mail adresa ili korisničko ime već zauzeto i ne nastavlja na idući korak registracije dok se ne unese nova adresa ili korisničko ime
  - 3.b Upisana e-mail adresa nije ispravnog formata
    1. Sustav javlja da je e-mail adresa neispravnog formata te ne nastavlja na sljedeći korak registracije dok korisnik ne ispravi e-mail adresu
  - 3.c Administrator ne prihvata prijavu korisnika
    1. Korisnik ne može pristupiti svom profilu

##### UC2 - Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Prijavljanje u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojeći korisnički profil prihvачen od strane administratora
- **Opis osnovnog tijeka:**
  1. Korisnik upisuje e-mail adresu ili korisničko ime te lozinku
  2. Korisnik odabire opciju prijavljivanja u sustav
  3. Sustav preusmjerava korisnika na njegov profil
- **Opis mogućih odstupanja:**
  - 2.a Neispravna (nepostojeća) e-mail adresa ili lozinka

1. Sustav korisniku ispisuje poruku o neispravnim podacima i vraća ga na sučelje za prijavu

### UC3 - Prihvatanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Odobriti registraciju korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran
- **Opis osnovnog tijeka:**
  1. Administrator prihvata registriranog korisnika u sustavu

### UC4 - Dodavanje festivala

- **Glavni sudionik:** Voditelj
- **Cilj:** Stvaranje jednog ili više novih festivala
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije stvaranja novog festivala
  2. Upis traženih podataka vezanih uz festival
  3. Odabir opcije dodavanja festivala u sustav
  4. Preusmjeravanje na stranicu novostvorenog festivala
- **Opis mogućih odstupanja:**
  - 2.a Voditelj nije upisao sve tražene podatke o festivalu
    1. Sustav voditelju ne dozvoljava stvaranje novog festivala i vraća ga na početnu stranicu voditelja

### UC6 - Prihvatanje organizatora

- **Glavni sudionik:** Voditelj
- **Cilj:** Potvrđivanje prijave organizatora za festival
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda prijava za određeni festival
  2. Odabir opcije prihvatanja prijave
  3. Preusmjeravanje na stranicu festivala

### UC7 - Dodavanje događaja

- **Glavni sudiočnik:** Voditelj
- **Cilj:** Dodavanje jednog ili više događaja vezanih uz odabrani festival
- **Sudiočnici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen i nalazi se na stranici odabranog festivala
- **Opis osnovnog tijeka:**
  1. Odabir opcije dodavanja novog događaja
  2. Upis traženih podataka o događaju i odabir jednog od dostupnih organizatora
  3. Povratak na stranicu festivala
- **Opis mogućih odstupanja:**
  - 2.a Voditelj nije upisao sve tražene podatke
    1. Događaj se ne unosi u sustav i voditelja se preusmjerava na stranicu festivala

#### UC9 - Pristup profilu organizatora

- **Glavni sudiočnik:** Voditelj
- **Cilj:** Pregled profila organizatora
- **Sudiočnici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen i nalazi se na stranici odabranog festivala
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda organizatora prijavljenih za festival
  2. Preusmjeranje na stranicu profila organizatora

#### UC10 - Komentiranje organizatora

- **Glavni sudiočnik:** Voditelj
- **Cilj:** Pisanje komentara na profilu organizatora
- **Sudiočnici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen i odabrani organizator je odradio posao za voditelja
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda profila organizatora
  2. Preusmjeranje na stranicu profila organizatora
  3. Odabir opcije dodavanja komentara
  4. Upis komentara
  5. Odabir opcije spremanja komentara
  6. Povratak na stranicu profila organizatora

- **Opis mogućih odstupanja:**

- 3.a Voditelj pokušava ostaviti komentar za organizatora koji još nije radio za njega
  1. Aplikacija onemogućava spremanje komentara
- 3.b Voditelj ne spremi komentar
  1. Aplikacija upozorava organizatora da nije spremio komentar

### **UC11 - Pristup profilu izvođača**

- **Glavni sudionik:** Voditelj
- **Cilj:** Pregled profila izvođača koje je organizator pridijelio događaju
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen i nalazi se na stranici festivala
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda događaja
  2. Odabir opcije pregleda profila izvođača
  3. Preusmjerenje na stranicu profila izvođača

### **UC12 - Pristup licitacijama i prijavama**

- **Glavni sudionik:** Voditelj
- **Cilj:** Pregled licitacija i izvođača koji su se prijavili
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen i nalazi se na stranici festivala
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda događaja
  2. Odabir opcije pregleda stranice otvorene licitacije
  3. Preusmjerenje na stranicu licitacije

### **UC13 - Ispis propusnice**

- **Glavni sudionik:** Voditelj
- **Cilj:** Ispis propusnice za željeni festival
- **Sudionici:** Baza podataka
- **Preduvjet:** Voditelj je prijavljen i nalazi se na stranici festivala
- **Opis osnovnog tijeka:**
  1. Odabir opcije ispisa propusnice
  2. Potvrda dozvole preuzimanja dokumenta

### **UC14 - Prijava za festival**

- **Glavni sudionik:** Organizator
- **Cilj:** Prijava za organizaciju događaja na festivalu
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda festivala
  2. Odabir opcije prijave za organiziranje događaja na festivalu

#### **UC15 - Pristup profilu izvođača**

- **Glavni sudionik:** Organizator
- **Cilj:** Pregled profila izvođača
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda događaja
  2. Odabir opcije pregleda profila izvođača
  3. Preusmjerenje na stranicu profila izvođača

#### **UC16 - Komentiranje izvođača**

- **Glavni sudionik:** Organizator
- **Cilj:** Pisanje komentara na profilu izvođača
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen i odabrani izvođač je odradio posao za organizatora
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda profila izvođača
  2. Preusmjerenje na stranicu profila izvođača
  3. Odabir opcije dodavanja komentara
  4. Upis komentara
  5. Odabir opcije spremanja komentara
  6. Povratak na stranicu profila izvođača
- **Opis mogućih odstupanja:**
  - 3.a Organizator pokušava ostaviti komentar za izvođača koji još nije radio za njega
    1. Aplikacija onemogućava spremanje komentara
  - 3.b Organizator ne spremi komentar

1. Aplikacija upozorava organizatora da nije spremio komentar

### UC17 - Otvaranje licitacije

- **Glavni sudionik:** Organizator
- **Cilj:** Otvoriti licitaciju za određeni posao
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda događaja
  2. Odabir događaja
  3. Odabir opcije dodavanja poslova
  4. Unos novog posla
  5. Odabir opcije spremanja unosa
  6. Povratak na pregled događaja
- **Opis mogućih odstupanja:**
  - 1.a Organizator pokušava dodati posao za završeni događaj
    1. Aplikacija onemogućava odabir događaja
  - 4.a Organizator ne spremi dodani posao
    1. Aplikacija upozorava organizatora da nije spremio unos

### UC18 - Produljivanje licitacije

- **Glavni sudionik:** Organizator
- **Cilj:** Produljivanje aktivne licitacije
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda otvorenih licitacija
  2. Odabir licitacije
  3. Odabir opcije produljenja licitacije
  4. Povratak na pregled aktivnih licitacija
- **Opis mogućih odstupanja:**
  - 2.a Organizator pokušava produljiti licitaciju koja je završena ili je već jednom bila produljena
    1. Aplikacija onemogućava produljivanje

### UC19 - Potvrđivanje prijave

- **Glavni sudionik:** Organizator
- **Cilj:** Potvrđivanje prijave za posao
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda prijava za određeni posao
  2. Odabir opcije prihvatanja prijave
  3. Preusmjeravanje na pregled poslova

#### UC20 - Ispis propusnice

- **Glavni sudionik:** Organizator
- **Cilj:** Ispis propusnice za željeni festival
- **Sudionici:** Baza podataka
- **Preduvjet:** Organizator je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije ispisa propusnice
  2. Potvrda dozvole preuzimanja dokumenta

#### UC21 - Dodavanje djelatnosti

- **Glavni sudionik:** Izvođač
- **Cilj:** Unos djelatnosti za koje je izvođač specijaliziran
- **Sudionici:** Baza podataka
- **Preduvjet:** Izvođač je prijavljen
- **Opis osnovnog tijeka:**
  1. Odabir opcije pregleda djelatnosti
  2. Odabir djelatnosti za koje je izvođač specijaliziran
- **Opis mogućih odstupanja:**
  - 2.a Djelatnost za koju je izvođač specijaliziran nije navedena u izborniku
    1. Izvođač ručno dodaje djelatnost za koju je specijaliziran

#### UC22 - Pristup poslovima

- **Glavni sudionik:** Izvođač
- **Cilj:** Pristupanje svim poslovima otvorenima za licitaciju za koje je specijaliziran
- **Sudionici:** Baza podataka
- **Preduvjet:** Izvođač je prijavljen

- **Opis osnovnog tijeka:**

1. Odabir opcije pregleda otvorenih licitacija
2. Odabir opcije prijave na licitaciju ili izlazak iz izbornika

### UC23 - Prijava na licitaciju

- **Glavni sudionik:** Izvođač

- **Cilj:** Prijavljanje na licitaciju

- **Sudionici:** Baza podataka

- **Preduvjet:** Izvođač je prijavljen

- **Opis osnovnog tijeka:**

1. Odabir opcije pregleda otvorenih licitacija
2. Odabir opcije prijave na licitaciju
3. Unos traženih podataka
4. Povratak na pregled otvorenih licitacija

### UC24 - Ispis propusnica

- **Glavni sudionik:** Izvođač

- **Cilj:** Ispisivanje potrebnog broja propusnica

- **Sudionici:** Baza podataka

- **Preduvjet:** Izvođač je prijavljen

- **Opis osnovnog tijeka:**

1. Odabir opcije ispisa propusnice
2. Odabir festivala i posla koji obavlja
3. Potvrda dozvole preuzimanja dokumenta

- **Opis mogućih odstupanja:**

1. Sustav ispisuje poruku o prekoračenju broja propusnica

### UC25 - Odjava s licitacije

- **Glavni sudionik:** Izvođač

- **Cilj:** Odjavljivanje s prijavljene licitacije

- **Sudionici:** Baza podataka

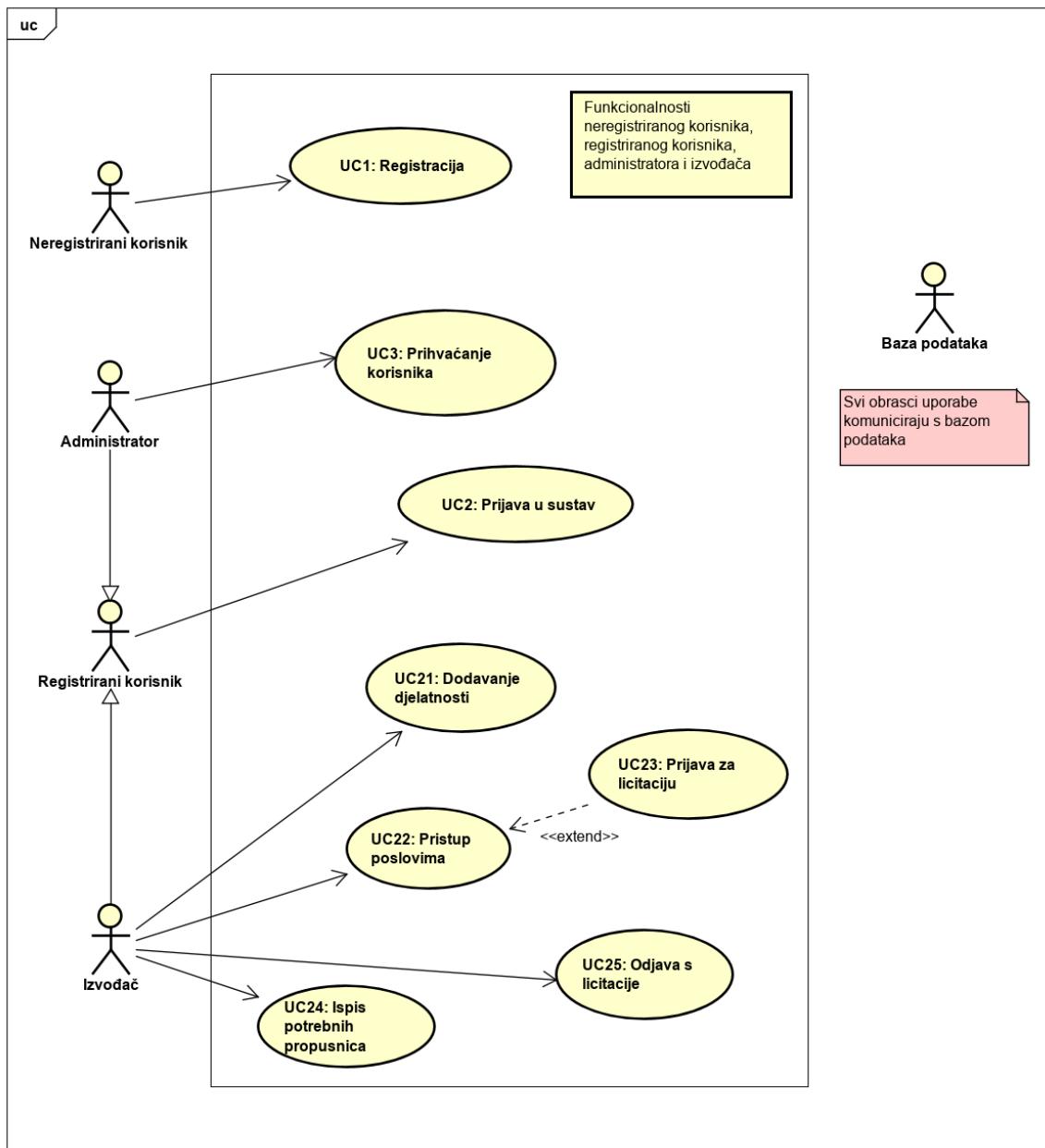
- **Preduvjet:** Izvođač je prijavljen

- **Opis osnovnog tijeka:**

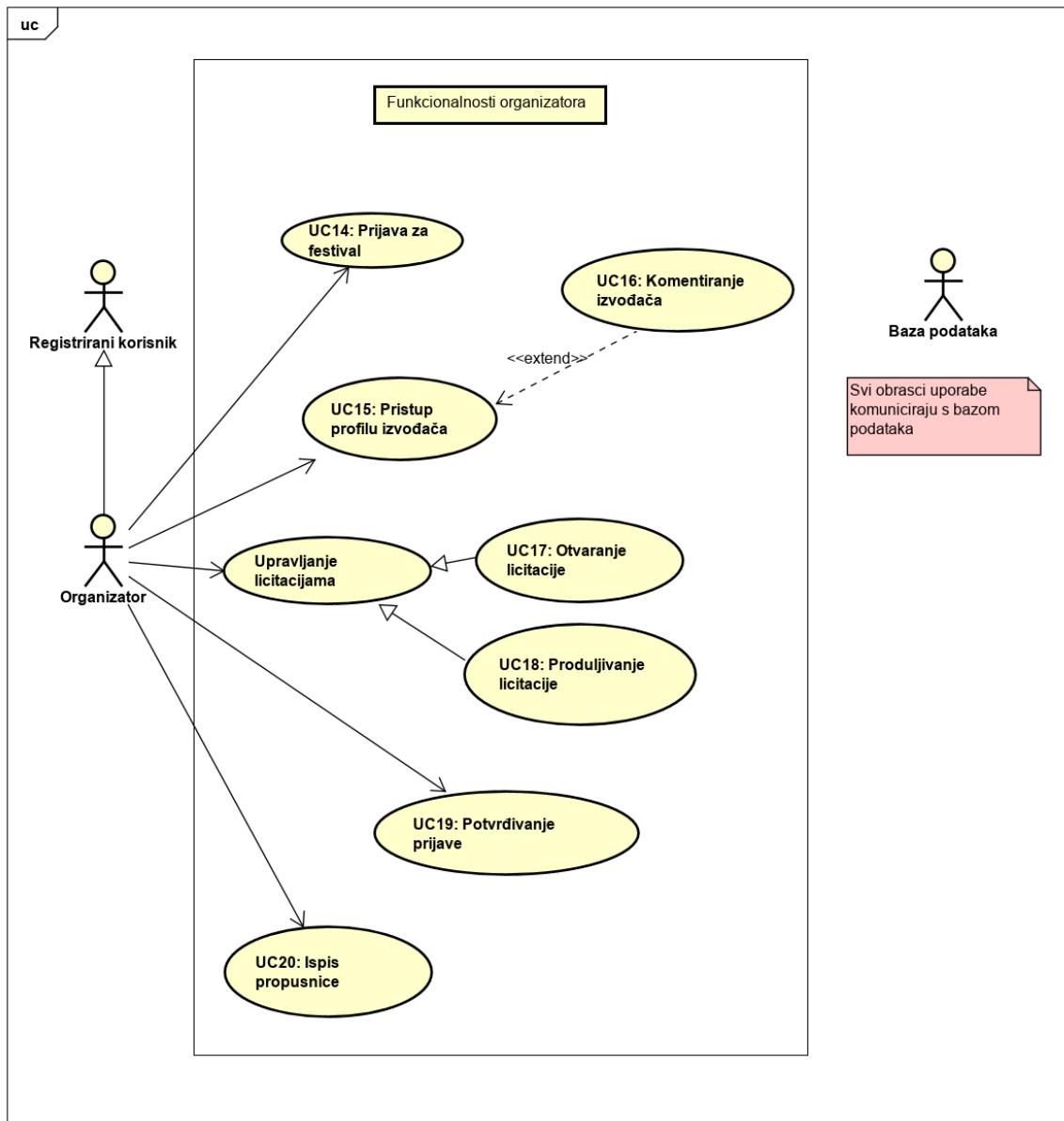
1. Odabir opcije pregleda prijavljenih licitacija
2. Odabir prijavljene licitacije
3. Odabir opcije odjave licitacije

4. Povratak na pregled prijavljenih licitacija
- **Opis mogućih odstupanja:**
  - 3.a Izvođač može slučajno odjaviti licitaciju
    1. Izvođač može ponovno prijaviti ranije odjavljenu licitaciju

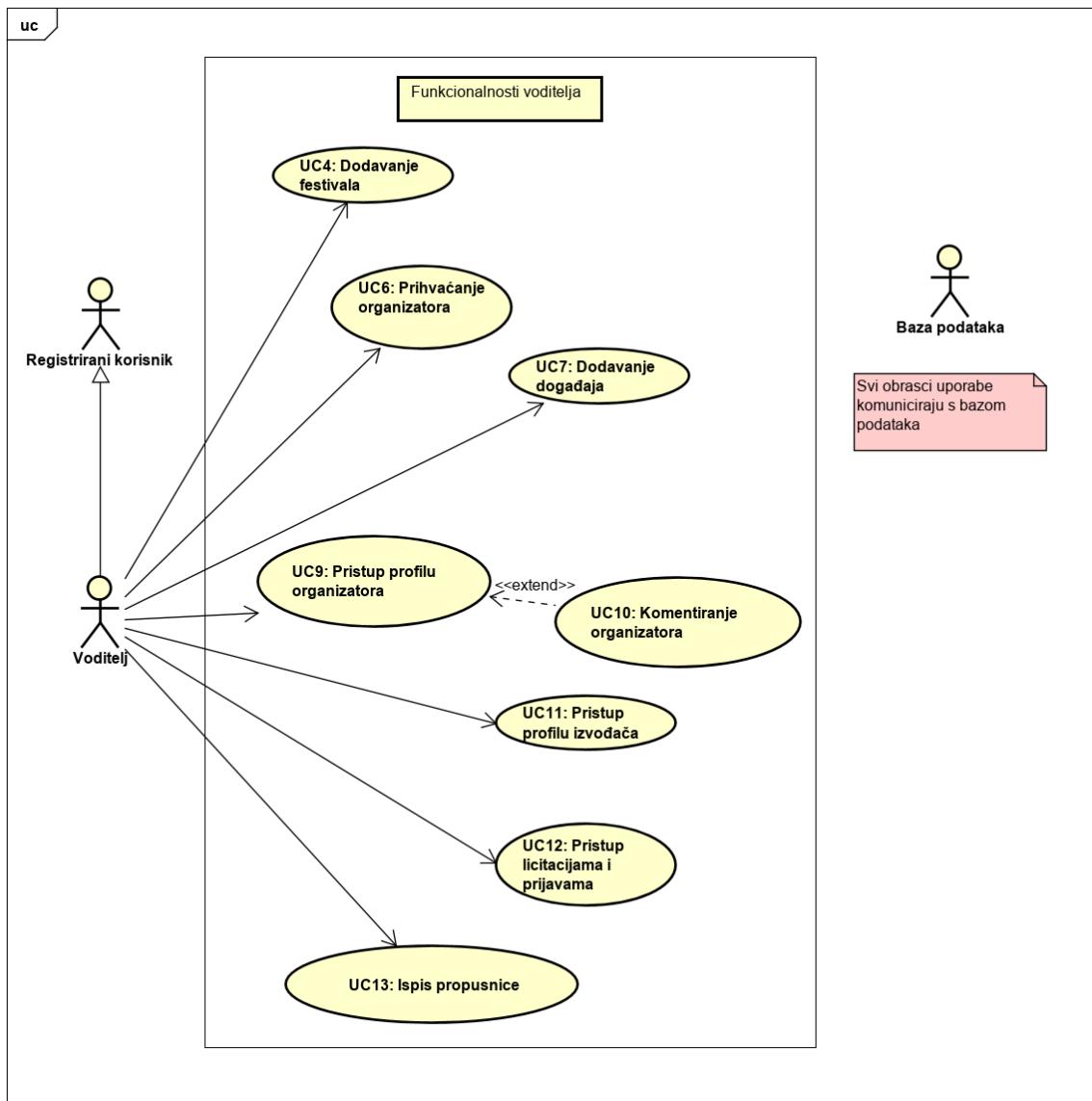
## Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost neregistriranog korisnika, registriranog korisnika, administratora i izvođača



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost organizatora

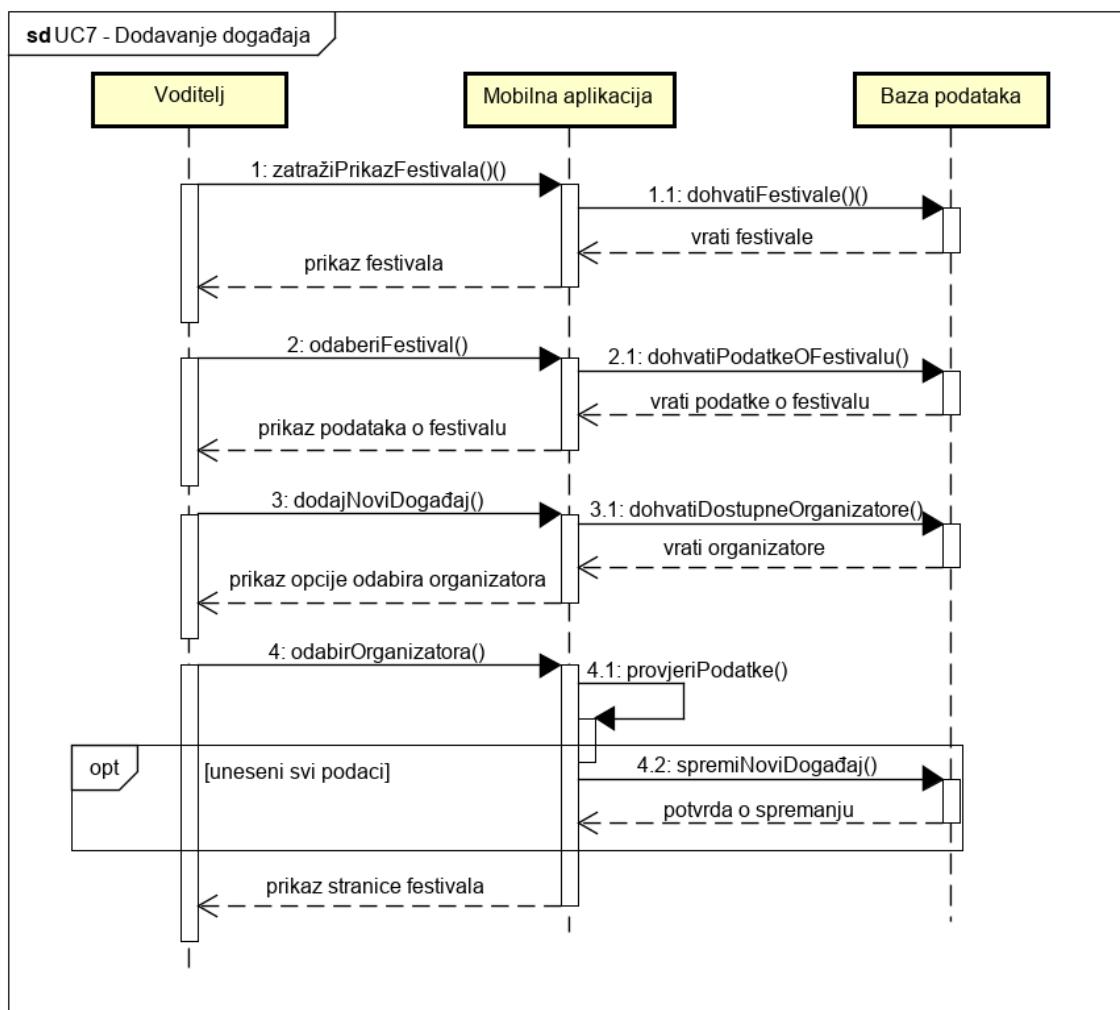


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost voditelja

### 3.1.2 Sekvencijski dijagrami

#### UC7 - Dodavanje događaja

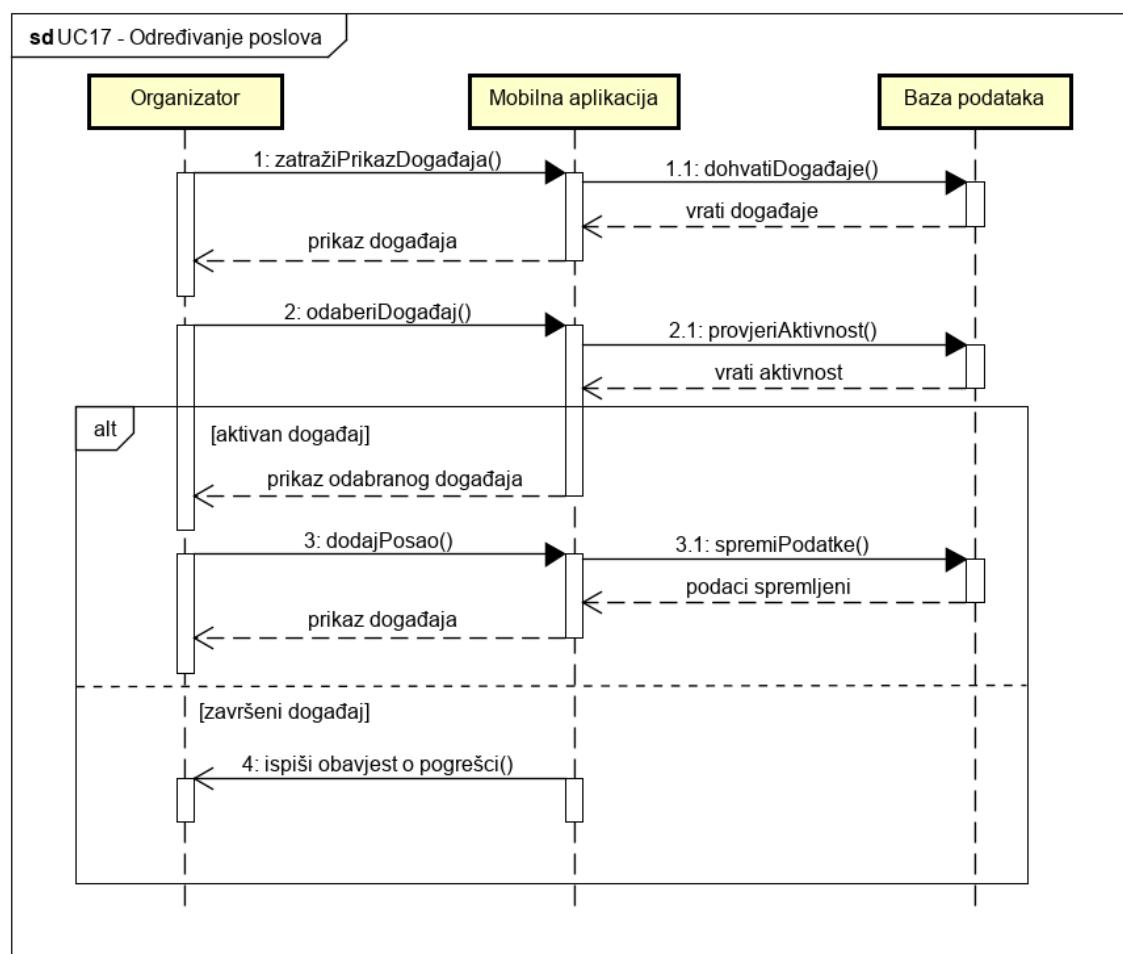
Voditelj šalje zahtjev za pregled vlastitih festivala, poslužitelj dohvaća podatke iz baze podataka i prikazuje ih nakon čega voditelj odabire jedan od festivala kako bi dodao novi događaj za taj festival. Voditelj odabire opciju dodavanja novog događaja na festivalu i upisuje tražene podatke o događaju. Poslužitelj dohvaća trenutno dostupne organizatore iz baze podataka kako bi voditelj mogao odabrati jednog od dostupnih organizatora za događanje. Nakon što su svi podaci uneseni, spremaju se u bazu podataka, a voditelj se preusmjerava na stranicu festivala. Ukoliko svi traženi podaci nisu uneseni, događaj se ne unosi u bazu podataka, a voditelj se preusmjerava natrag na stranicu festivala.



Slika 3.4: Sekvencijski dijagram za UC7

### UC17 - Otvaranje licitacije

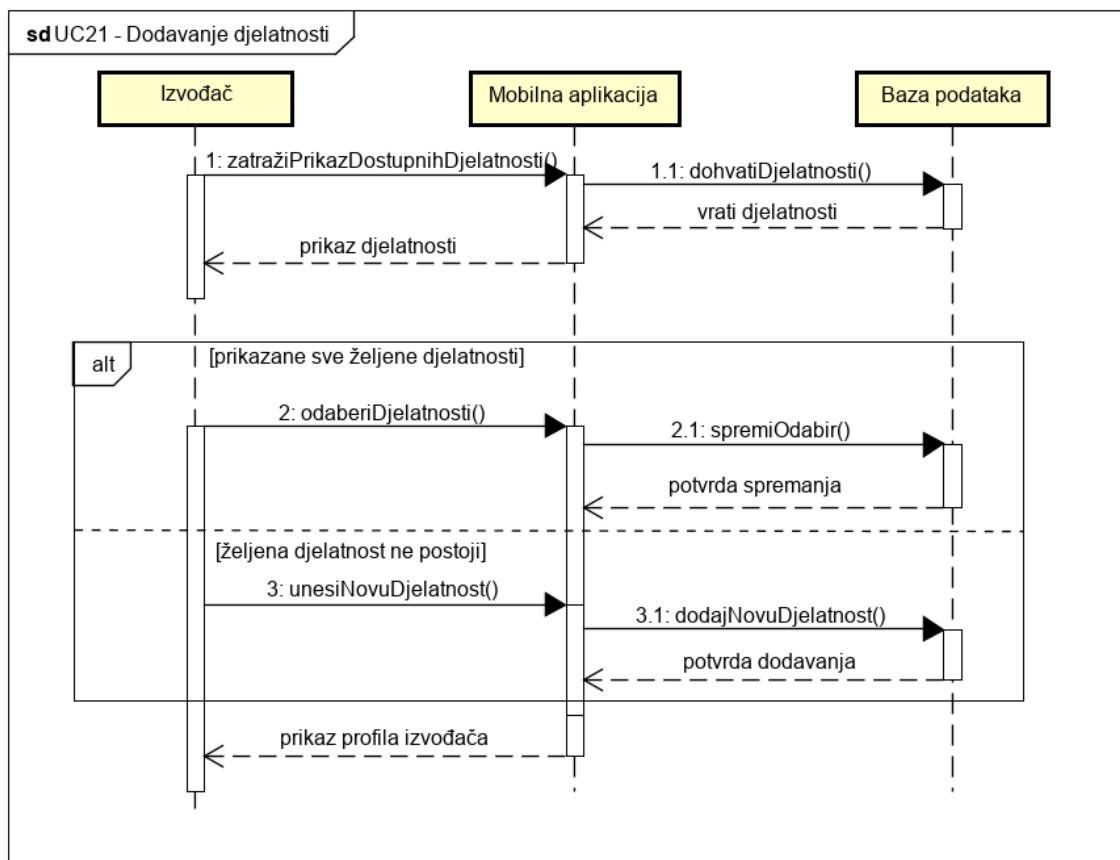
Organizator šalje zahtjev za pregled svojih događaja kako bi mogao dodati novi posao u neki od aktivnih događaja. Poslužitelj iz baze podataka dohvaća podatke o organizatorovim događajima i prikazuje ih. Organizator odabire jedan od ponuđenih događaja, a ukoliko organizator pokuša dodati posao za završeni događaj, aplikacija će mu to onemogućiti uz poruku da odabere neki od aktivnih događaja. Na stranici odabranog događaja odabire opciju dodavanja novog posla nakon čega unosi podatke o poslu koji je potrebno obaviti. Odabirom opcije spremanja uneseni podaci spremaju se u bazu podataka, a u slučaju da organizator ne spremi dodani posao, aplikacija će ga upozoriti da nije ažurirao poslove uz opciju da to ipak učini prije izlaska.



Slika 3.5: Sekvencijski dijagram za UC17

### UC21 - Dodavanje djelatnosti

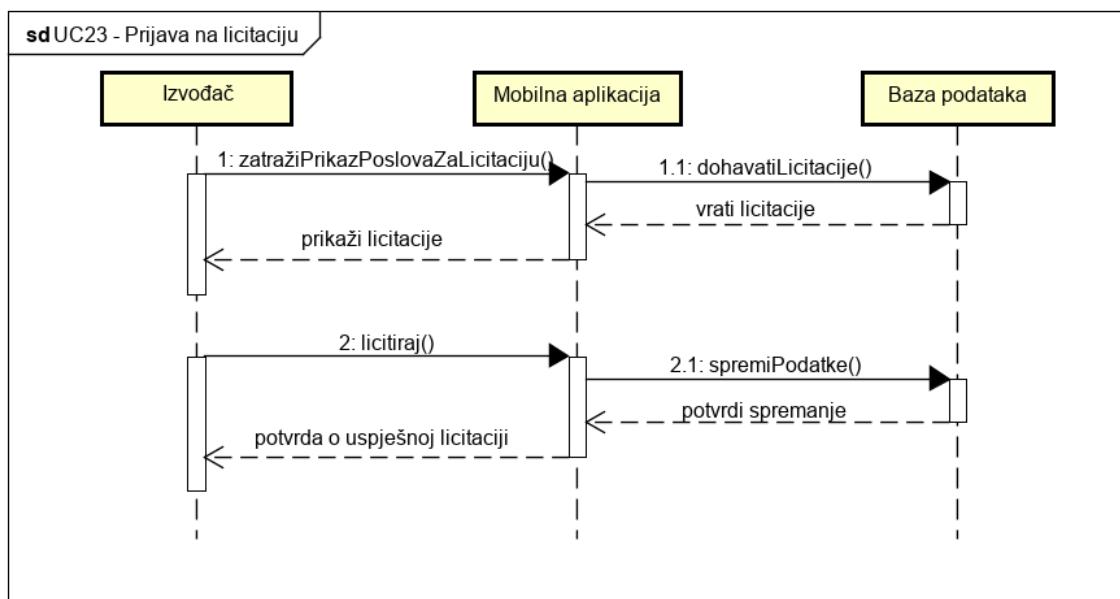
Izvođač šalje zahtjev za pregled popisa mogućih djelatnosti kako bi odabrao djelatnosti za koje je specijaliziran. Poslužitelj dohvaća podatke iz baze podataka i prikazuje ih. Izvođač odabire djelatnosti s popisa, a ukoliko je izvođač specijaliziran za neku djelatnost koja nije na popisu, ima opciju ručno dodati novu djelatnost. Nakon što izvođač odabere sve djelatnosti, podaci se spremaju u bazu podataka, a izvođač je preusmjerjen natrag na pregled svog profila.



Slika 3.6: Sekvencijski dijagram za UC21

**UC23 - Prijava na licitaciju**

Izvođač šalje zahtjev za pregled poslova otvorenih za licitaciju za koje je specijaliziran. Poslužitelj dohvaća podatke iz baze podataka i prikazuje ih. Izvođač odbire željeni posao te upisuje cijenu, komentar, broj osoba potrebnih za odrađivanje posla i procijenjeno trajanje izvođenja posla. Ukoliko su podaci uspješno spremjeni u bazu podataka, izvođač dobiva potvrdu da je prijavljen na licitaciju.



Slika 3.7: Sekvencijski dijagram za UC23

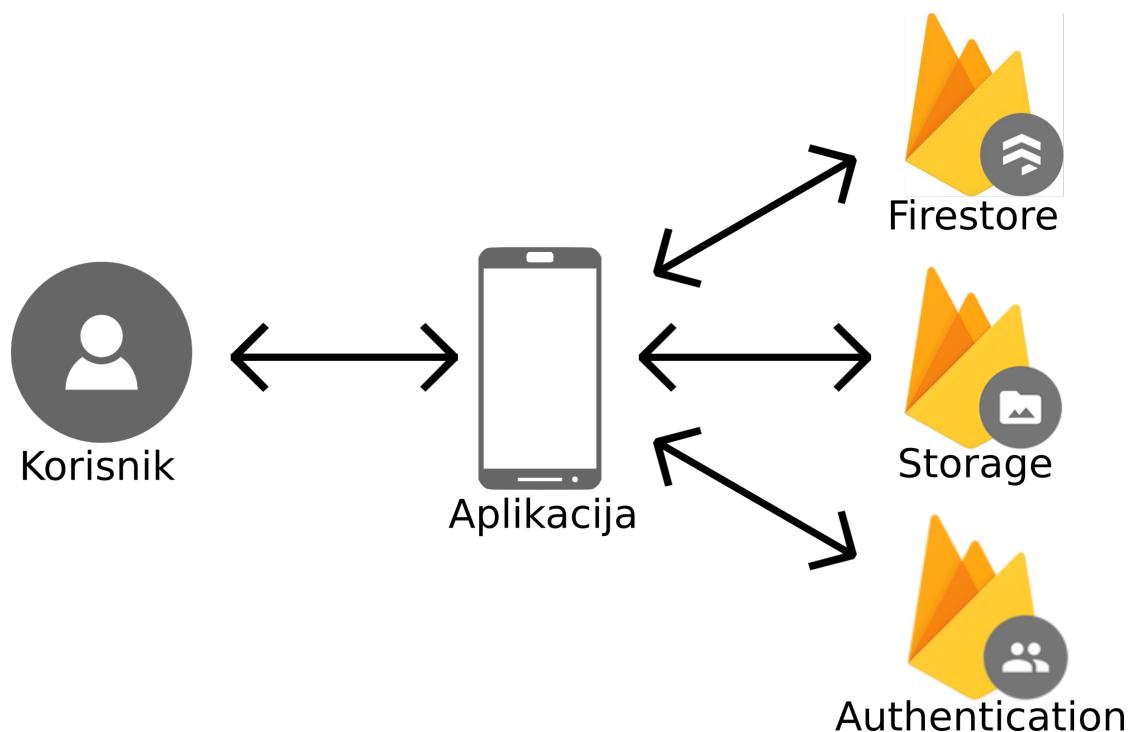
## 3.2 Ostali zahtjevi

- 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 mobilna aplikacija koristeći objektno-orientirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava

## 4. Arhitektura i dizajn sustava

Arhitektura je podijeljena na četiri podsustava:

- Android aplikacija
- Firebase Firestore
- Firebase Storage
- Firebase Authentication



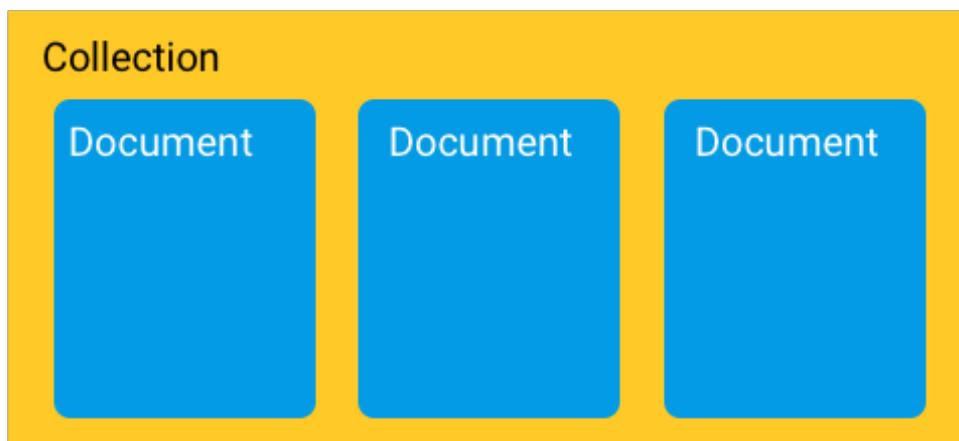
Slika 4.1: Skica sustava

*Android aplikacija* je sučelje koje povezuje korisnika sa sadržajima čiji je kostur definiran unutar različitih aktivnosti. Sadržaj aktivnosti određen je *Firebase* uslugama, čije su komponente objašnjene u idućim odsjećcima. Temeljna poveznica android aplikacije sa specifičnim uslugama sadržana je u JSON datoteci koja se automatski generira pri prvom zahtjevanju korištenja *Firebase* usluge. Komunikacija s

*Firebase* uslugama vrši se instanciranjem prikladnih objekata te korištenjem pri-padnih metoda. Dohvaćanje podataka je asinkrono čime se korisniku omogućuje pristup određenim funkcionalnostima bez čekanja na ostale.

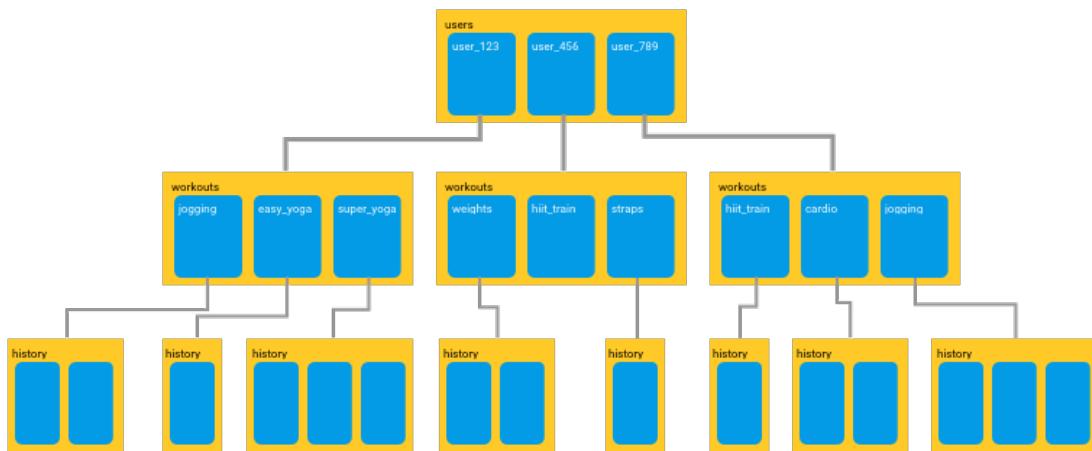
*Firebase Authentication* je usluga koja omogućuje prijavu korisnika u sustav te izradu korisničkog računa. Usluga se temelji na jedinstvenom ključu (*token*) ko-jeg korisnik dobiva nakon uspješne prijave. *Firebase Authentication* omogućuje i podešavanje proizvoljnih parametara za odabrane korisnike (poput, primjerice, različitih uloga). Takvi parametri sadržani su u ključu; Čitanjem njihovih vrijed-nosti omogućeno je upravljanje pristupom.

*Firebase Firestore* je NoSQL baza podataka. Jedna takva baza podataka omogućuje do 1,000,000 istovremenih spajanja, što je za aplikaciju koja povezuje velik broj korisnika prirodan izbor. Umjesto relativno velikog JSON stabla tipičnog za NoSQL baze podataka (koji u slučaju velikog broja podataka zahtjeva denormalizaciju), podaci se pohranjuju u obliku dokumenata (*documents*) i kolekcija (*collections*) kao što je prikazano na slici 4.2. Struktura hipotetske baze podataka neke aplikacije prikazana je na slici 4.3.



Slika 4.2: Struktura *Firebase Firestore* modela

Prednost ovakve strukture je mogućnost dohvaćanja dokumenata bez istovre-menog dohvaćanja podkolekcija, čime se podaci mogu logički strukturirati u hi-jerarhiju bez brige o preuzimanju velike količine suvišnih informacija. Nadalje, umjesto denormalizacije koju zahtjeva većina NoSQL modela, *Firebase Firestore* omogućuje automatsku izgradnju indeksa prilagođenih određenim upitima. Unu-



Slika 4.3: Hipotetska struktura *Firebase Firestore* modela

tar zasebne datoteke, moguće je modificirati sigurnosna pravila za pristup i dijeljenje podataka.

*Firebase Storage* korisnicima omogućuje pohranu multimedijskog sadržaja te sadrži vlastitu datoteku sa sigurnosnim pravilima koju je moguće modificirati na sličan način kao i kod ostalih *Firebase* usluga.

Ovakvu arhitekturu odabiremo radi relativne fleksibilnosti i jednostavnosti korištenja; Modeliranje ne polazi od vrste podataka, već od upita koji se otkrivaju i formiraju tijekom razvoja i evolucije aplikacije.

## 4.1 Baza podataka

### dio 1. revizije

Za potrebe našeg sustava koristimo NoSQL model radi veće fleksibilnosti u odnosu na relacijski model. Ipak, u ovom poglavljju opisujemo ekvivalentan relacijski model. Osnovna jedinica takvog modela je relacija. Relacija je definirana vlastitim imenom i skupom atributa. Baza podataka ove aplikacije koristi sljedeće relacije:

- Korisnik
- Festival

- Događaj
- Posao
- Djelatnost
- Licitacija
- Posao-Djelatnost
- Izvođač-Djelatnost
- Organizator-Izvođač

#### 4.1.1 Opis tablica

**Korisnik** Ovaj entitet sadrži osnovne informacije o korisniku aplikacije. Korisnik je definiran atributima: ime, prezime, email, broj mobitela, slika, uloga i potvrđen. Ovaj entitet je u *One-to-Many* vezi s entitetom Festival preko korisničkog imena. *Many-to-Many* veza s entitetom posao ostvarena je uz pomoć *One-to-Many* veze s entitetom Licitacija preko korisničkog imena. *Many-to-Many* veza s entitetom Djelatnost ostvarena je uz pomoć *One-to-Many* veze s entitetom Izvođač-Djelatnost preko korisničkog imena. *Many-to-Many* refleksivna veza ostvarena je preko entiteta Organizator-Izvođač uz pomoć korisničkog imena organizatora i korisničkog imena izvođača. Za jednog korisnika definirana je jedna uloga, a uloga poprima jednu od sljedećih vrijednosti: *izvođač, organizator, voditelj, administrator*. Samo jedan korisnik je administrator. Broj korisnika s ostalim ulogama nije ograničen.

<b>Korisnik</b>		
ID	VARCHAR	jedinstveno korisničko ime
Ime	VARCHAR	ime korisnika
Prezime	VARCHAR	prezime korisnika
Email	VARCHAR	e-mail adresa korisnika
Broj mobitela	VARCHAR	telefonski broj korisnika
Slika	BLOB	slika korisnika
Uloga	VARCHAR	uloga korisnika
Potvrđen	BIT	status potvrde korisničkog računa od strane administratora

**Festival** Ovaj entitet sadrži osnovne informacije o festivalu. Atributi koji definiraju festival su: identifikator festivala, naziv, opis, slika, datum i vrijeme početka, datum i vrijeme završetka te korisničko ime voditelja. Ovaj entitet je u *One-to-Many* vezi s entitetom Događaj preko atributa identifikator događaja i *Many-to-One* vezi s entitetom Korisnik preko atributa korisničko ime.

Festival		
ID festival	INT	identifikator festivala
Naziv	VARCHAR	ime festivala
Opis	VARCHAR	kratki opis festivala
Slika	BLOB	logotip festivala
Datum vrijeme poč	TIMESTAMP	datum i vrijeme početka festivala
Datum vrijeme kraj	TIMESTAMP	datum i vrijeme završetka festivala
ID voditelj	VARCHAR	korisničko ime voditelja

**Događaj** Ovaj entitet sadrži osnovne informacije o događaju nekog festivala. Atributi koji definiraju događaj su: identifikator događaja, naziv, opis, prostorija, datum i vrijeme početka, datum i vrijeme završetka, identifikator festivala i identifikator voditelja. Ovaj entitet je u *One-to-Many* vezi s entitetom Posao preko identifikatora događaja, identifikatora festivala i korisničkog imena voditelja, *Many-to-One* vezi s entitetom Korisnik preko korisničkog imena voditelja te *Many-to-One* vezi s entitetom Festival preko identifikatora festivala i identifikatora voditelja.

Događaj		
ID događaj	INT	identifikator događaja
Naziv	VARCHAR	naziv događaja
Opis	VARCHAR	opis događaja
Prostorija	VARCHAR	naziv ili oznaka prostorije održavanja događaja
Datum vrijeme poč	TIMESTAMP	datum i vrijeme početka događaja
Datum vrijeme kraj	TIMESTAMP	datum i vrijeme završetka događaja
ID festival	INT	identifikator festivala
ID voditelj	VARCHAR	korisničko ime voditelja

<b>Događaj</b>
----------------

**Posao** Ovaj entitet sadrži osnovne informacije o poslovima koje organizator definira za neki događaj. Atributi koji definiraju posao su: identifikator posla, identifikator festivala, identifikator događaja te korisničko ime voditelja. Ovaj entitet je u *One-to-Many* vezi s entitetom Licitacija preko korisničkog imena izvođača, identifikatora posla, identifikatora festivala, identifikatora događaja te korisničkog imena voditelja, *Many-to-One* vezi s entitetom Događaj preko identifikatora događaja, identifikatora festivala i korisničkog imena voditelja. *Many-to-Many* veza s entitetom Djelatnost ostvarena je uz pomoć Entiteta Posao-Djelatnost preko identifikatora posla, identifikatora djelatnosti, identifikatora festivala, identifikatora događaja te korisničkog imena voditelja.

Posao		
ID posao	INT	identifikator posla
Redni broj	INT	redni broj posla u redoslijedu izvršavanja
Datum vrijeme poč	TIMESTAMP	datum i vrijeme otvaranja licitacije
ID festival	INT	identifikator festivala
ID događaj	INT	identifikator događaja
ID voditelj	VARCHAR	korisničko ime voditelja

**Djelatnost** Ovaj entitet opisuje različite djelatnosti potrebne za obavljanje poslova u kontekstu organizacije festivala. Atributi koji definiraju djelatnost su: identifikator djelatnosti i naziv djelatnosti. *Many-to-Many* veza s entitetom Korisnik ostvarena je uz pomoć entiteta Izvođač-Djelatnost preko identifikatora djelatnosti i identifikatora izvođača. *Many-to-Many* veza s entitetom Posao ostvarena je uz pomoć Entiteta Posao-Djelatnost preko identifikatora posla, identifikatora djelatnosti, identifikatora festivala, identifikatora događaja te korisničkog imena voditelja.

Djelatnost		
ID djelatnost	INT	identifikator djelatnosti
Naziv	VARCHAR	naziv djelatnosti

**Licitacija** Ovaj entitet sadrži elemente potrebne za prijavu izvođača na određeni

posao. Atributi koji definiraju licitaciju su: korisničko ime izvođača, identifikator posla, cijena te komentar. *Many-to-One* veza s entitetom Korisnik ostvarena je preko korisničkog imena izvođača. *Many-to-One* veza s entitetom Posao ostvarena je preko identifikatora posla, identifikatora festivala, identifikatora događaja i korisničkog imena voditelja. Na ovaj način ostvaruje se *Many-to-Many* veza između entiteta Posao i Korisnik uz vlastite atribute cijena i komentar.

Licitacija		
Cijena	INT	cijena ponude od strane izvođača
komentar	VARCHAR	komentar izvođača
ID izvođač	VARCHAR	korisničko ime izvođača
ID posao	INT	identifikator posla

**Posao-Djelatnost** Ovaj entitet sadrži elemente potrebne za prijavu izvođača na određeni posao. Atributi koji definiraju licitaciju su: korisničko ime izvođača, identifikator posla, identifikator festivala, identifikator događaja, identifikator voditelja, cijena te komentar. *Many-to-One* veza s entitetom Korisnik ostvarena je preko korisničkog imena izvođača. *Many-to-One* veza s entitetom Posao ostvarena je preko identifikatora posla, identifikatora festivala, identifikatora događaja i korisničkog imena voditelja. Na ovaj način ostvaruje se *Many-to-Many* veza između entiteta Posao i Korisnik uz vlastite atribute cijena i komentar.

Posao-Djelatnost		
ID posao	INT	identifikator posla
ID djelatnost	VARCHAR	identifikator djelatnosti
ID festival	INT	identifikator festivala
ID događaj	INT	identifikator događaja
ID voditelj	VARCHAR	korisničko ime voditelja

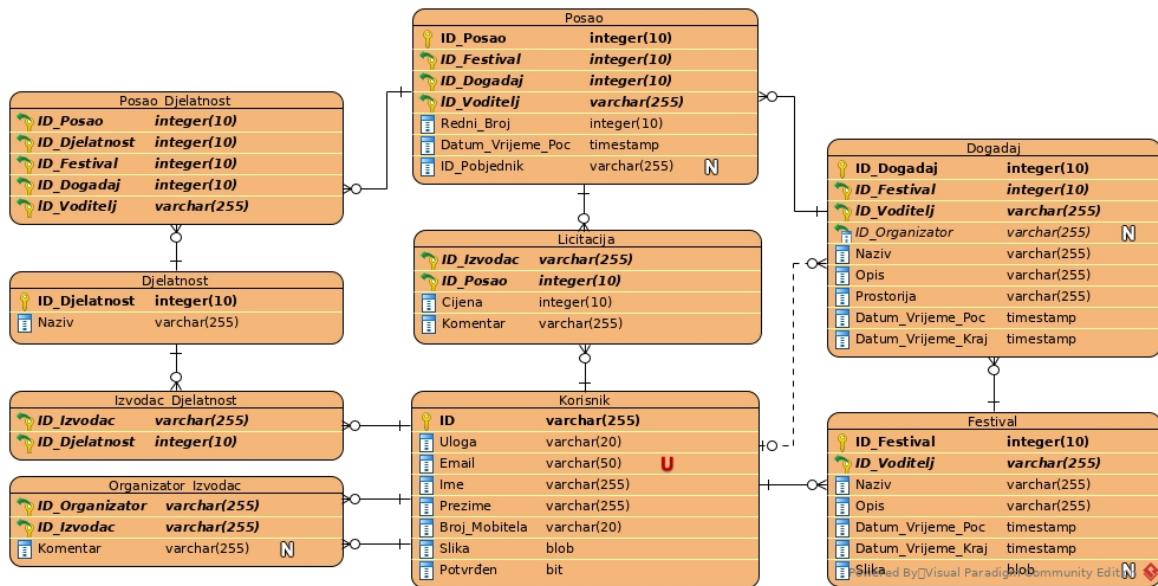
**Izvođač-Djelatnost** Ovaj entitet sadrži atribute potrebne za ostvarenje *Many-to-Many* veze između entiteta Korisnik i Djelatnost. Entitet Izvođač-Djelatnost određen je atributima: korisničko ime izvođača (pomoću kojeg se ostvaruje *Many-to-One* veza s entitetom Korisnik) i identifikator djelatnosti (pomoću kojeg se ostvaruje *Many-to-One* veza s entitetom Djelatnost).

<b>Izvođač-Djelatnost</b>		
ID izvodac	VARCHAR	korisničko ime izvođača
ID djelatnost	INT	identifikator djelatnosti

**Organizator-Izvođač** Ovaj entitet sadrži elemente potrebne za ostvarenje *Many-to-Many* refleksivne veze nad relacijom Korisnik. Entitet Organizator-Izvođač određen je atributima: korisničko ime organizatora, korisničko ime izvođača te komentar. Entitet služi za pohranu komentara koji pojedini organizator, ukoliko smatra da je potrebno, ostavlja na profilu izvođača s kojim je surađivao.

<b>Organizator-Izvođač</b>		
Komentar	INT	komentar organizatora
ID organizator	VARCHAR	korisničko ime organizatora
ID izvođač	VARCHAR	korisničko ime izvođača

### 4.1.2 Dijagram baze podataka



Slika 4.4: ER dijagram baze podataka

## 4.2 Dijagram razreda

Temeljni razred svake Android aplikacije je *Activity*. *Activity* predstavlja prozor s korisničkim sučeljem. Radi dodatnih pogodnosti u kontekstu dizajna, koristimo *AppCompatActivity* razred.

Razredi koje nasljeđuju *AppCompatActivity* predstavljaju prilagođeno korisničko sučelje ovisno o ulozi i namjeni. Točan raspored elemenata pojedinog prozora definiran je u pripadnoj XML datoteci, a pojednom elementu pristupa se nakon stvaranja objekta. U svrhu jednostavnosti, instance razreda *AppCompatActivity* u ostaku teksta nazivamo *aktivnost*. Osim razreda *AppCompatActivity*, često je upotrebljavan razred *Fragment* čije instance u ostaku teksta nazivamo *fragment*. Temeljne funkcionalnosti razreda *Fragment* identične su onima koje nudi razred *AppCompatActivity*. Fragment je, iz perspektive korisnika, dio prozora unutar prozora aktivnosti. Fragmenti omogućavaju prijelaz između različitih prozora uz zadržavanje temeljnih elemenata pozivajuće aktivnosti poput, primjerice, trake izbornika.

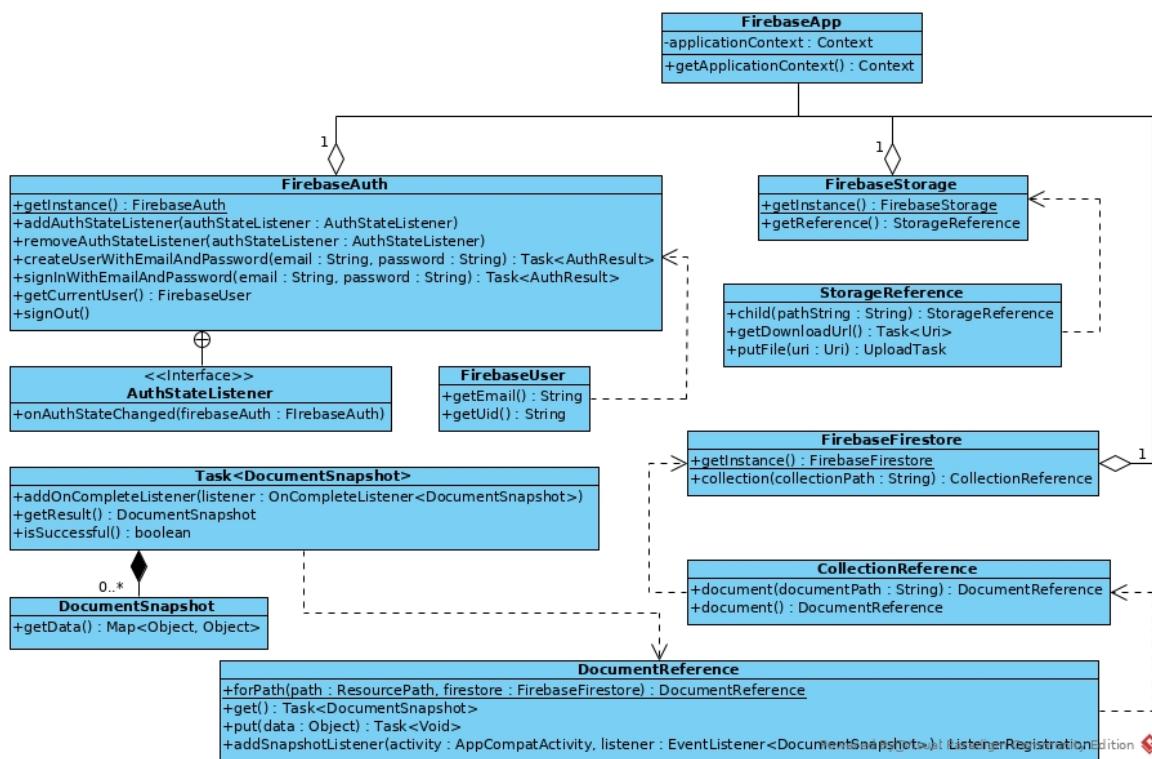
Radi preglednosti, dijagrami će biti prikazani ovisno o smještaju u paketu gdje jedan paket objedinjuje razrede sa sličnom namjenom. Bitno je napomenuti da svaki razred osim *MainActivity* ovisi o stanju instance razreda *FirebaseAuth* čime se onemogućuje neautorizirani pristup bilo kojoj funkcionalnosti koja nije namijenjena za trenutnog korisnika aplikacije.

### 4.2.1 Dijagram razreda Firebase usluga

Na idućoj slici prikazan je dijagram razreda koji opisuje odnos razreda i sučelja u sklopu Firebase usluga. *FirebaseApp* je ulazna točka za sve ostale Firebase razvojne alate. U našoj aplikaciji, kao i u većini aplikacija koje koriste Firebase razvojne alate, nije ostvarena direktna interakcija s ovim razredom - ona se ostvaruje prilikom pozivanja metode *getInstance*. Ovisno o referentnom razredu, pozivom ove metode stvaraju se instance razreda *FirebaseAuth*, *FirebaseStorage* i *FirebaseFirestore*.

*FirebaseAuth* sadrži metode potrebne za prijavu korisnika te registraciju e-poštom i lozinkom. Promjene stanja autorizacije, poput uspješne prijave, moguće je pratiti pozivanjem metode *addAuthStateListener*. Argument ove metode zahtjeva implementaciju sučelja *AuthStateListener*. Ovo sučelje ugniježđeno je unutar razreda *FirebaseAuth* te definira metodu *onAuthStateChanged*. Nadjačavanjem ove metode, definiramo željeno ponašanje ukoliko dođe do prethodno spomenute promjene.

*FirebaseFirestore* omogućuje komunikaciju s Firebase bazom podataka. Dohvaćanje



Slika 4.5: Dijagram razreda Firebase usluga

podataka započinje dohvaćanjem pokazivača na kolekciju - `CollectionReference`. Pokazivač na kolekciju rezultat je uspješnog izvršavanja metode `collection`.

Razred `CollectionReference` nudi mogućnost dohvaćanja pokazivača na pripadni dokument. Pokazivač na dokument - instanca razreda `DocumentReference` - rezultat je uspješnog izvršavanja metode `document`.

Konačno, razred `DocumentReference` pruža metode za manipulaciju podacima definiranim unutar referentnog dokumenta. Pohrana podataka moguća je pozivanjem metode `get`. Sadržaju dokumenta moguće je pristupiti na nekoliko načina; Ukoliko je potrebno osluškivati promjene određenog parametra unutar dokumenta, nad prethodno dobivenom referencom pozivamo metodu `addSnapshotListener`. Ako osluškivanje nije potrebno, dovoljno je koristiti metodu `get`. Ova metoda vraća instancu razreda `Task<DocumentSnapshot>` što znači da, ukoliko je dohvaćanje uspješno, možemo pristupiti instanci razreda `DocumentSnapshot`.

U oba slučaja, razred `DocumentSnapshot` predstavlja stanje referentnog dokumenta. Metoda `getData` pruža mogućnost preuzimanja pripadnog sadržaja u obliku mape.

U svrhu pohranjivanja i dohvaćanja slike koristimo razred `FirebaseStorage`. Po-

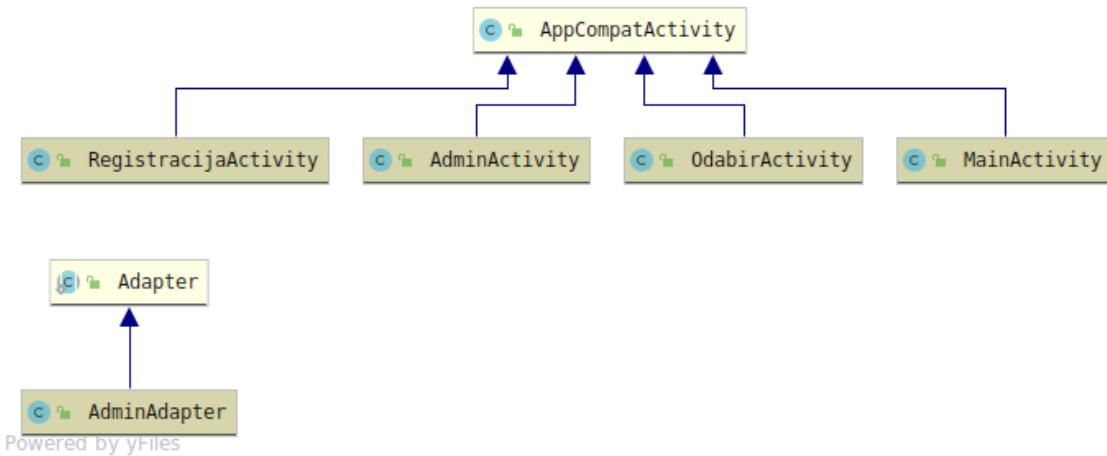
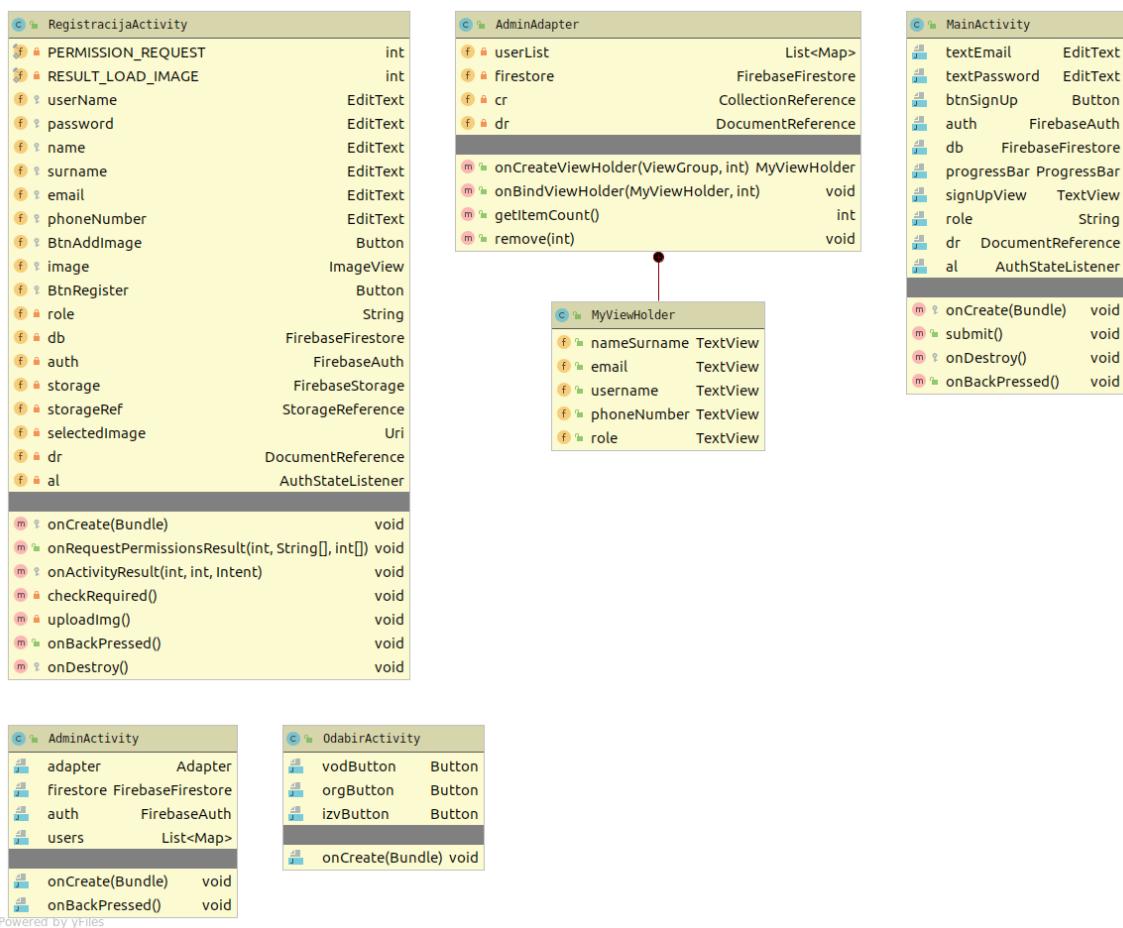
kazivač na spremnik - *StorageReference* - dohvaćamo uz pomoć metode *getReference*. Razred *StorageReference*, osim metoda potrebnih za pohranu i dohvaćanje slika, uz pomoć metode *child* nudi mogućnost pretraživanja različitih pod-direktorija.

#### 4.2.2 Dijagram razreda paketa *authentication*

Na slici 4.7 prikazan je dijagram razreda čija je uloga omogućavanje stvaranja korisničkog računa ili prijave s postojećim. Pojednostavljen dijagram prikazan je na slici 4.6. *MainActivity* predstavlja prozor koji se javlja nakon prvog pokretanja aplikacije. Korisnik ima mogućnost prijave s postojećim korisničkim računom te registracije. Instanca razreda FirebaseAuth služi za pokušaj prijave te funkcionira na prethodno objašnjen način. Ukoliko korisnik odabere registraciju, stvaranjem objekta *Intent* te pozivanjem metode *startActivity* stvara se objekt razreda *OdabirActivity*.

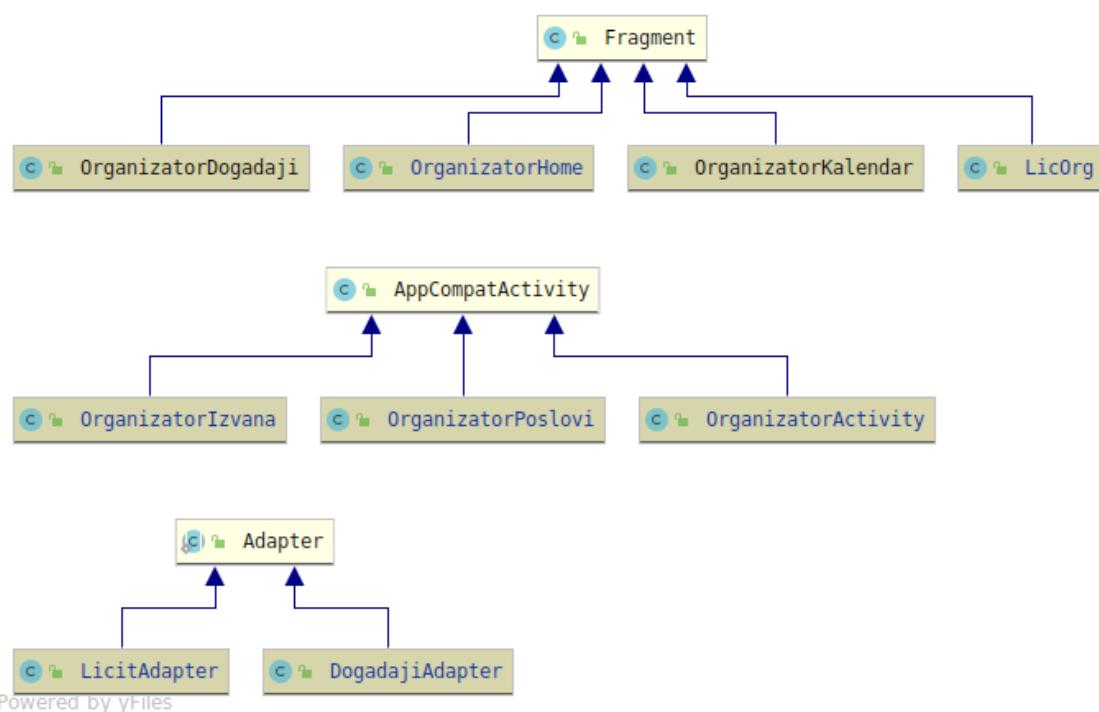
Razred *OdabirActivity* omogućuje odabir jedne od tri ponuđene uloge pomoću tri različita gumba definirana razredima *Button*. Nakon odabira uloge, stvara se objekt *RegistracijaActivity* koja korisniku omogućuje unos podataka potrebnih za stvaranje korisničkog računa. Tekstualni podaci predstavljeni su razredima *Edit-Text*. Moguće je, ali ne nužno, odabrati sliku pritiskom na gumb *BtnAddImage*. Slika je predstavljena razredom *ImageView* gdje je omogućen pregled iste nakon odabira. *BtnRegister* predstavlja gumb koji omogućuje pokušaj registracije. Ako registracija nije uspješna iz bilo kojeg razloga, korisniku će se ispisati odgovarajuća poruka. Ako je registracija uspješna, podaci se spremaju u obliku mape i uz pomoć usluga *FirebaseFirestore* te *FirebaseStorage* pohranjuju u bazu podataka.

*AdminActivity* definira prozor administratora. Administrator pomoću *Firebase* usluga, definiranim u vlastitim atributima, pristupa podacima o svim nepotvrđenim korisnicima. Nakon potvrde korisnika, izmjena u bazi podataka omogućuje istom stvaranje objekta instance pripadne uloge.

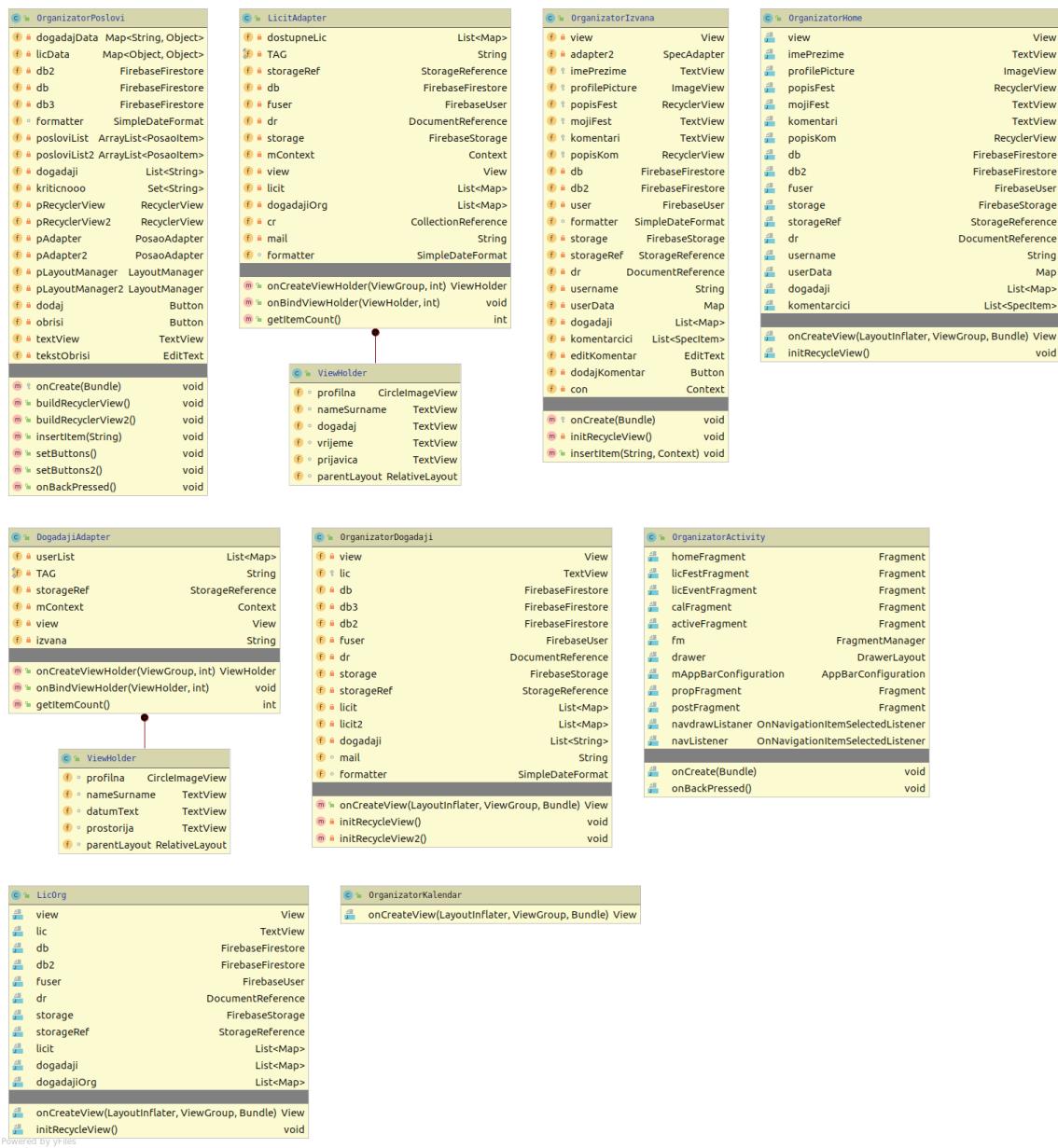
Slika 4.6: Dijagram razreda sadržanih u paketu *authentication*Slika 4.7: Dijagram razreda sadržanih u paketu *authentication*

#### 4.2.3 Dijagram razreda paketa *organizator*

Dijagram razreda čije funkcionalnosti pripadaju ulozi organizatora prikazan je na slici 4.9. Pojednostavljen dijagram prikazan je na slici 4.8. Razred *OrganizerActivity* pruža mogućnosti odabira funkcionalnosti ponuđenih na izbornoj traci čime se, ovisno o odabiru i slijedu odabira, stvaraju instance razreda koje nasljeđuju razred *Fragment* gdje jedan fragment predstavlja jedan prozor s definiranim funkcionalnostima. Razredi koji nasljeđuju razred *Adapter* omogućuju prikaz dohvaćenih stavki u obliku liste što su, u ovom slučaju, informacije o dodijeljenim događajima i licitacijama.



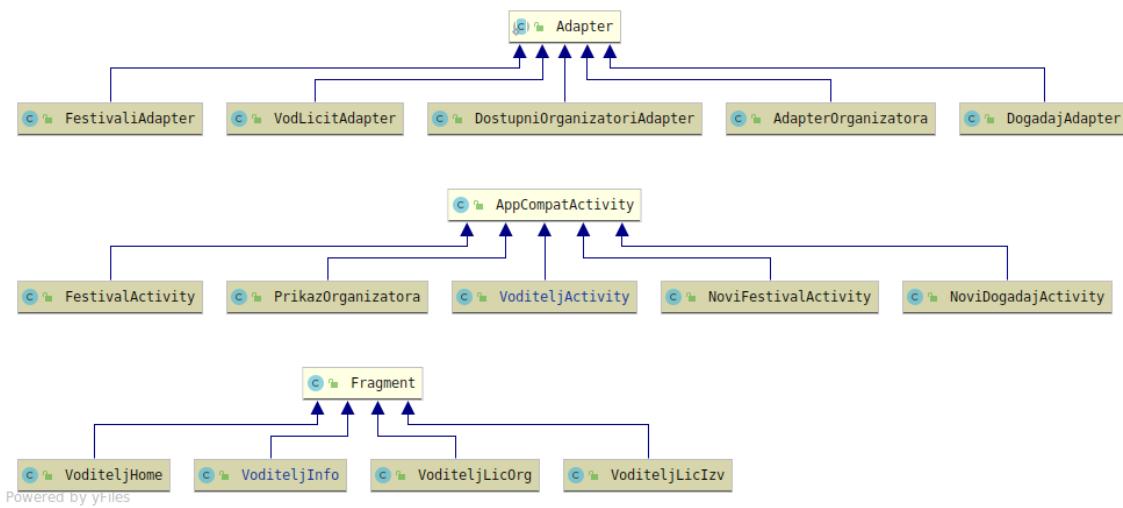
Slika 4.8: Dijagram razreda sadržanih u paketu *organizator*

Slika 4.9: Dijagram razreda sadržanih u paketu *organizator*

#### 4.2.4 Dijagram razreda paketa *voditelj*

Dijagram razreda čije funkcionalnosti pripadaju ulozi voditelja prikazan je na slići 4.11. Pojednostavljen dijagram prikazan je na slići 4.10. Razred `VoditeljActivity` omogućava korisniku pristup funkcionalnostima voditelja. Funkcionalnostima, koje su predstavljene instancama razreda koje nasljeđuju klasu `Fragment`, korisnik može pristupiti na izbornoj traci. Sve liste koje su dio funkcionalnosti profila voditelja ostvarene su razredima koji nasljeđuju klasu `Adapter`, a u ovom slučaju su

to festivali, događaji te aktivne i završene licitacije za organizatora i izvođača.

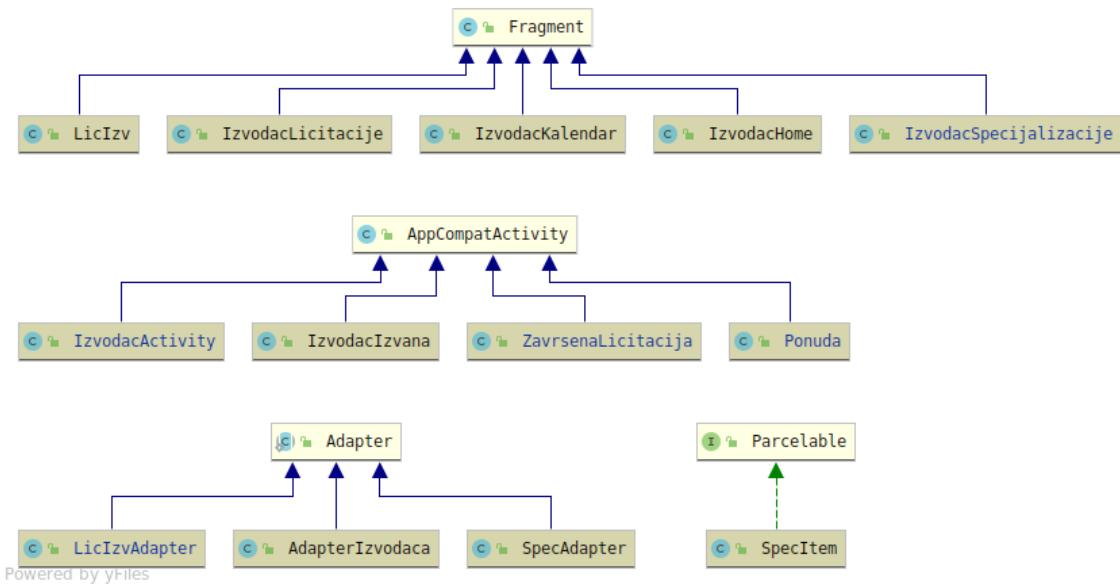


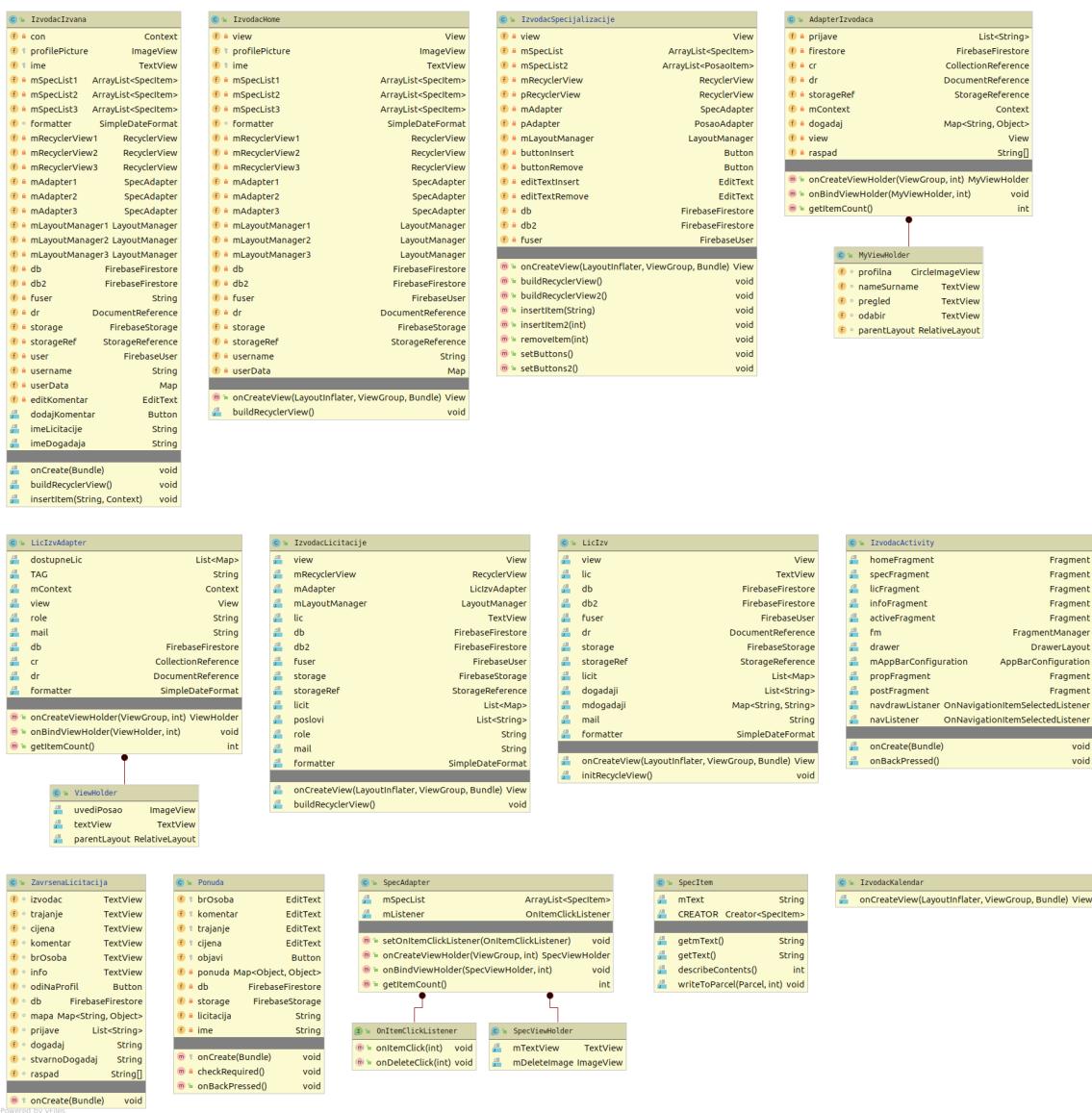
Slika 4.10: Dijagram razreda sadržanih u paketu *voditelj*

Slika 4.11: Dijagram razreda sadržanih u paketu `voditelj`

#### 4.2.5 Dijagram razreda paketa `izvodač`

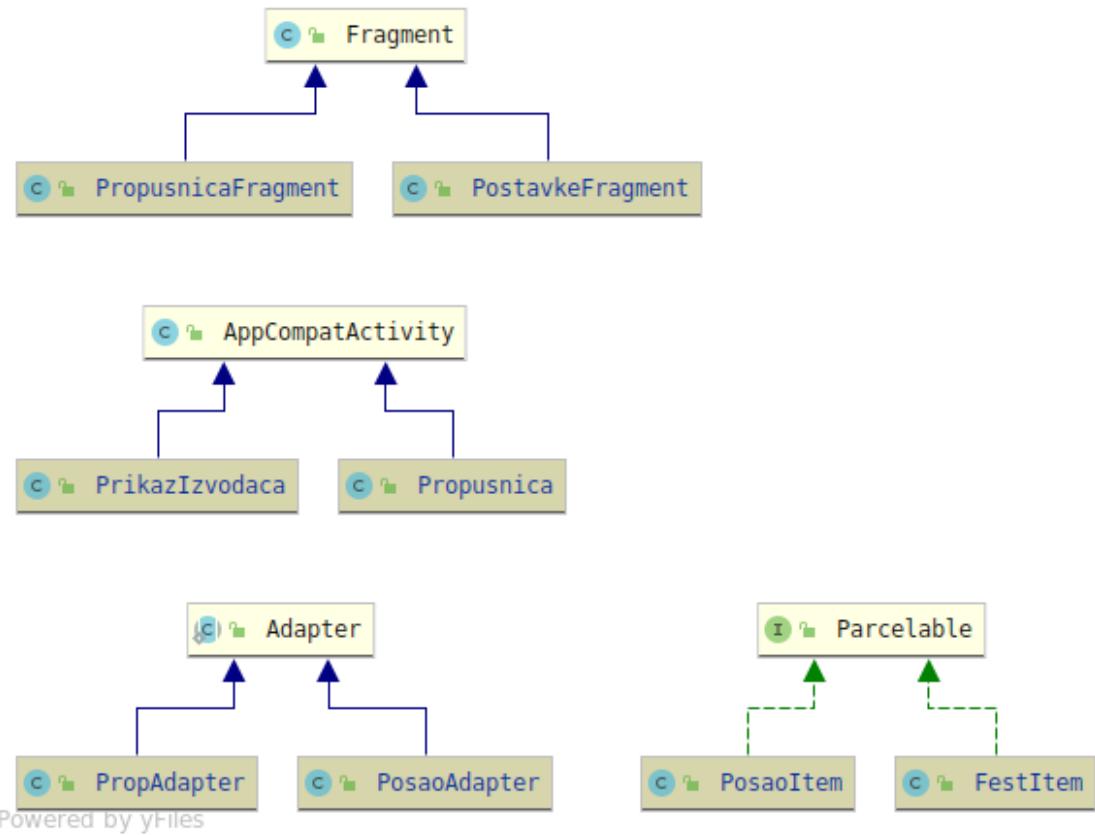
Dijagram razreda čije funkcionalnosti pripadaju ulozi izvođača prikazan je na slići 4.11. Pojednostavljen dijagram prikazan je na slići 4.10. Razredi koji nasljeđuju razrede `Fragment` i `Adapter` funkcionišu na prethodno opisan način. Izvođač ima liste dostupnih licitacija, specijalizacija i komentara.

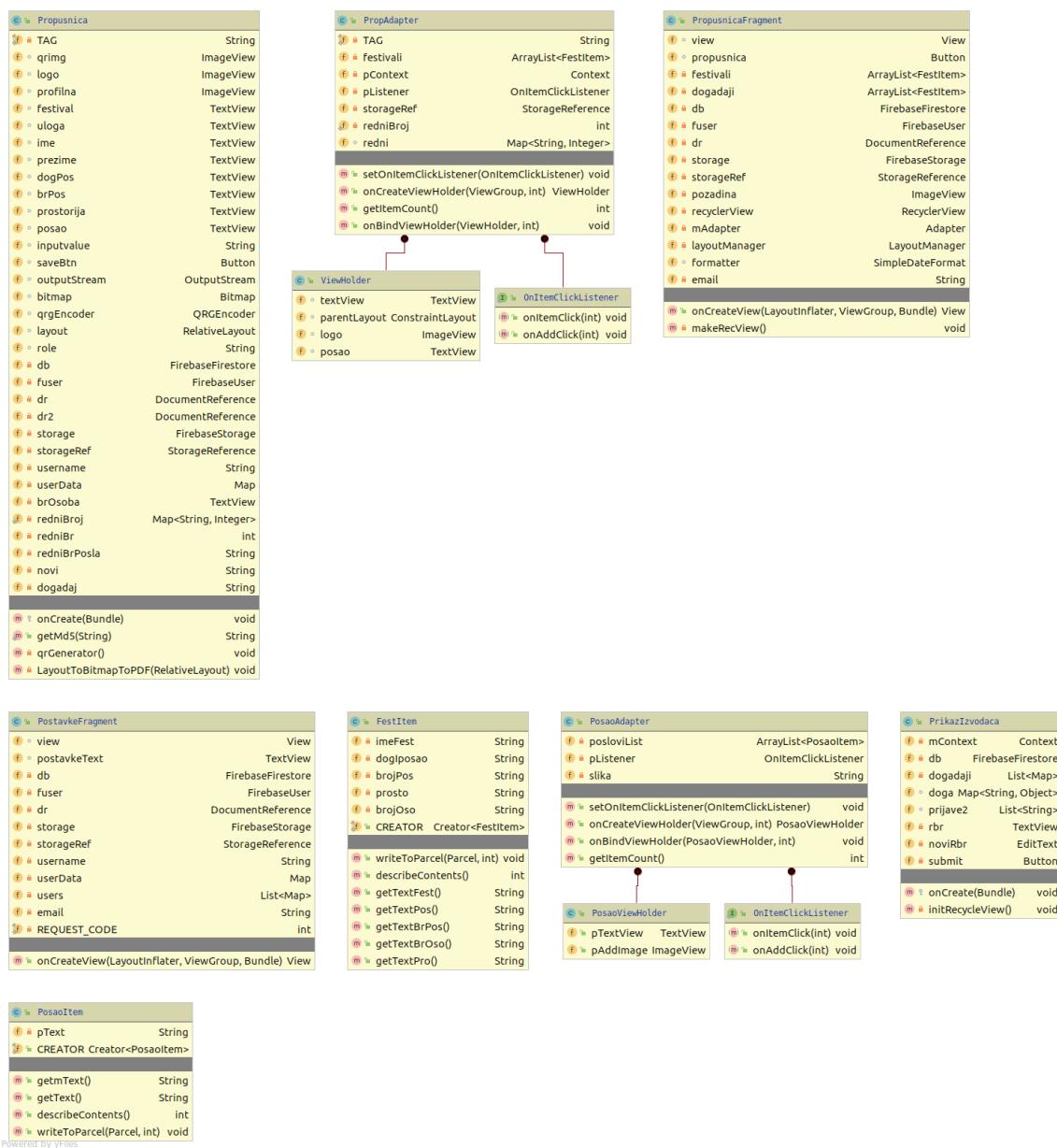
Slika 4.12: Dijagram razreda sadržanih u paketu *izvodac*

Slika 4.13: Dijagram razreda sadržanih u paketu *izvodac*

#### 4.2.6 Dijagram razreda paketa *common*

Razredi koji pripadaju paketu *common* korišteni su od strane dvije ili više uloga. Razredi koji nasljeđuju razrede *Fragment* i *Adapter* funkcioniraju na prethodno opisan način. Naziv razreda prati definiciju njihove upotrebe opisanu u prethodnim poglavljima.

Slika 4.14: Dijagram razreda sadržanih u paketu *common*

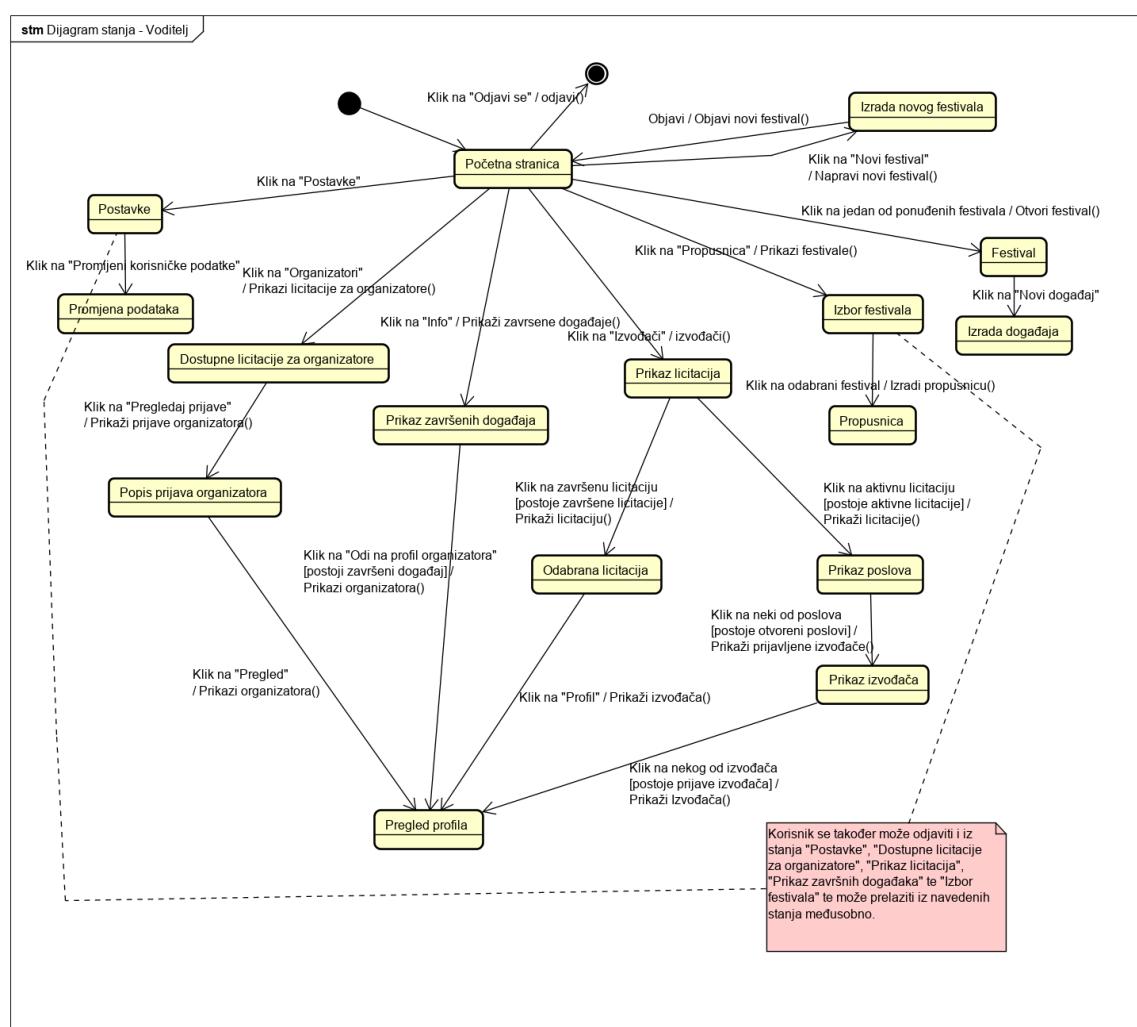
Slika 4.15: Dijagram razreda sadržanih u paketu *common*

### 4.3 Dijagram stanja

Dijagram stanja prikazuje stanje određenog objekta te sve prijelaze u iduća stanja. Na slici 4.7 je prikaz dijagrama stanja za voditelja.

Nakon prijavljivanja, voditelj pristupa svom profilu na kojem ima prikaz vlastitih festivala. Odabirom želenog festivala, voditelj može vidjeti sve događaje koje kroje odabrani festival te uz to može dodati nove događaje odabirom mogućnosti "Novi događaj". Pored popisa festivala postoji opcija "Dodaj festival" kojim voditelj može otvoriti novi festival.

Voditelj, odabirom opcije "Organizatori", ima mogućnost vidjeti licitacije pomoću kojih bira organizatora za određeni događaj.



Slika 4.16: Dijagram stanja voditelja

Odabirom događaja prikazuju mu se svi prijavljeni organizatori te može pristupiti profilu svakom od njih.

Dodatno, voditelj, odabirom opcije "Izvođači", može pristupiti svim otvorenim i zatvorenim licitacijama za poslove vezane uz događaje festivala. Odabirom otvorene licitacije, voditelj može vidjeti sve prijavljene izvođače, priхватiti željenog izvođača te pregledati profil svakog od izvođača. Otvaranjem zatvorene licitacije, voditelj ima pristup pojedinostima izabrane ponude te može pristupiti profilu odbranog izvođača.

Za svaki festival, voditelj može isprintati propusnicu za odabrani festival tako da u klizećem prikazu odabere mogućnost "Propusnica".

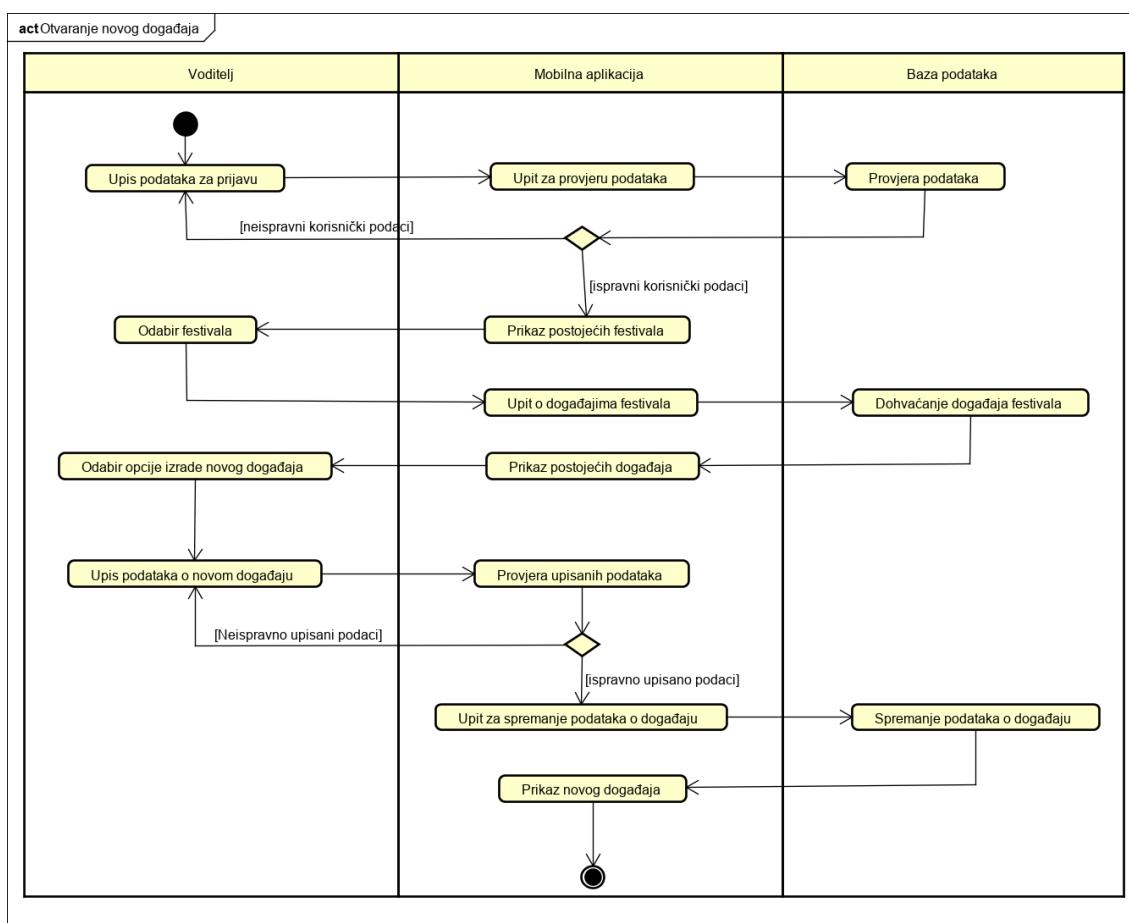
Nakon što se festival završi, voditelj odabirom "Info" može vidjeti sve završene događaje, pristupiti profilu organizatora zaduženog za pojedini događaj te ostaviti komentar. Voditelj također odabirom opcije "Postavke" može promijeniti korisničke podatke.

## 4.4 Dijagram aktivnosti

Dijagram aktivnosti služi za opis načina upravljanja podataka te prikaz toka podataka. Očekuje se što veća jednostavnost te se svaki korak mora završiti prije nego li se prijeđe na idući.

Slika 4.8 prikazuje stvaranje događaja vezanog uz već postojeći festival. Voditelj se prijavljuje u aplikaciju te mu se pri uspješnoj prijavi, na njegovoј početnoj stranici, prikažu svi njegovi otvoreni festivali. Odabirom jednog od festivala, voditelj pristupa pripadajućim događajima.

Voditelj otvara novi događaj biranjem te opcije. Prilikom izrade, voditelj upisuje podatke vezane uz taj događaj. Ako su svi podaci ispravni, događaj se upisuje u bazu podataka te prikazuje među aktivnim događajima.



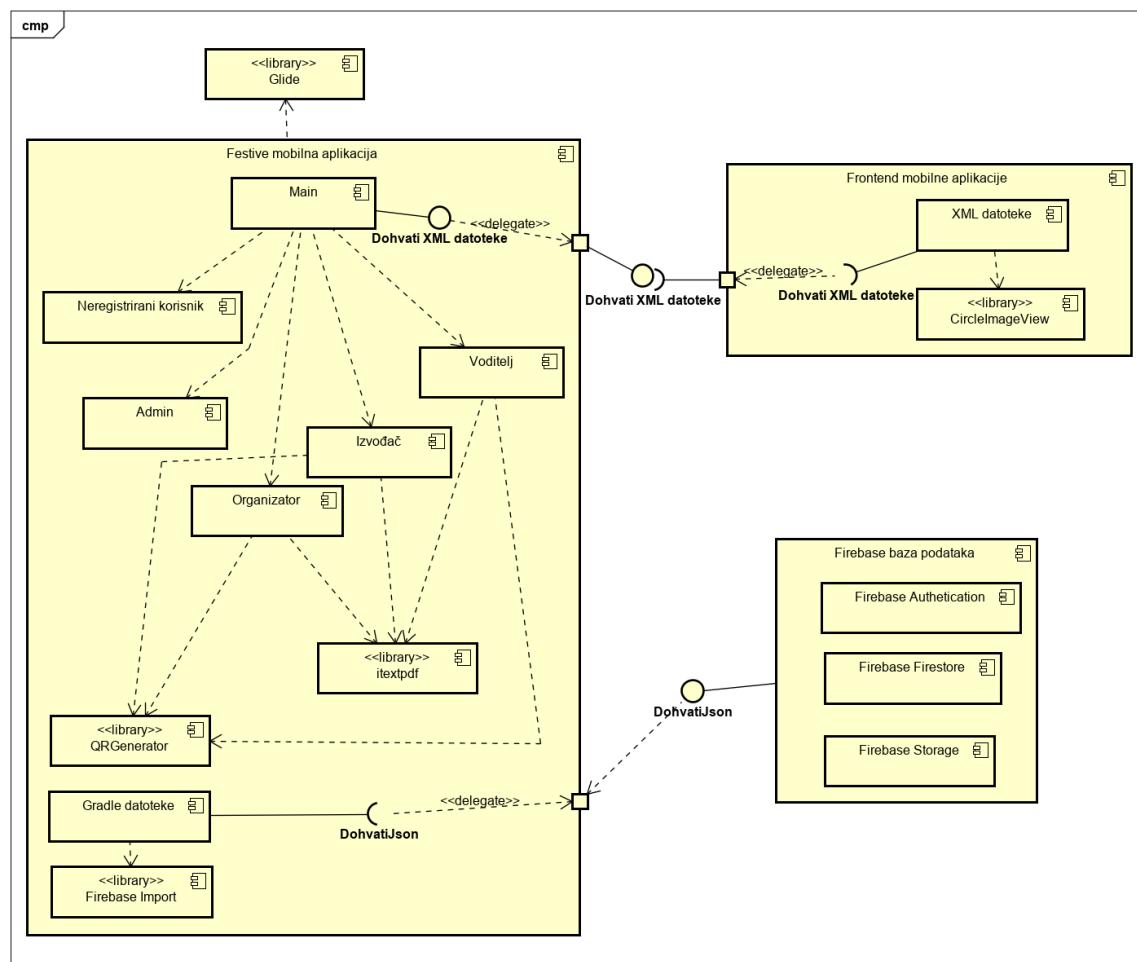
Slika 4.17: Dijagram aktivnosti otvaranja novog događaja

## 4.5 Dijagram komponenti

Dijagram komponenti prikazuje strukturno modeliranje s naglaskom na implementaciju sustava. Opisuje organizaciju i međuovisnost komponenti te interne strukture i njihov odnos prema okolini.

Na slici 4.9 nalazi se dijagram komponenti za aplikaciju Festive.

Aplikacija pomoću Json datoteke dohvaća iz baze podataka (Firebase) sve potrebne informacije. Gradle datoteke zadužene su za sinkronizaciju svih komponenata aplikacije te uz pomoć *Firebase Import* knjižnice učitavaju Json datoteke. Pomoću dobivenih informacija te sinkronizacije, aplikacija prikazuje jednu od 5 uloga; ne-registrirani korisnik, voditelj, organizator, izvođač te admin. Za uloge voditelj, organizator i izvođač potrebna je izrada propusnica koje su ostvarene koristeći



Slika 4.18: Dijagram komponenti

knjižnice *QRGenerator* i *itextpdf*. Knjižnica *Glide* funkcionira na razini cijele aplikacije i služi za učitavanje potrebnih slika za profile i festivale.

Jezik korišten za ostvarivanje funkcionalnosti uloga i aplikacije je Java. Aplikacija pomoću XML datoteka te knjižnice *CircleImageView*, koja služi za prikazivanje slika u okruglom obliku, ostvaruje svoj frontend koji komunicira s korisnikom.

# 5. Implementacija i korisničko sučelje

## 5.1 Korištene tehnologije i alati

Tim je komunicirao pomoću aplikacije WhatsApp. Za izradu UML dijagrama korišten je alat Astah Professional.

Sustav za upravljanje izvornim kodom je Git, dok je repozitorij aplikacije i projektne dokumentacije bio spremlijen na web platformi GitLab.

Korišteni programski jezik je Java, a za aplikaciju je korišteno razvojno okruženje Android Studio. Android Studio je integrirano razvojno okruženje za Googleov operativni sustav Android. Izgrađen na JetBrains-ovom IntelliJ IDEA softveru. Očekivana verzija androida za koju je aplikacija rađena je verzija 10, dok je kao minimalna verzija Androida koju će aplikacija podržavati postavljena verzija 7.

Za bazu podataka odabrana je Firebase baza podataka koja je temeljena na NoSQL principu. Baza podataka se nalazi na poslužitelju u oblaku Google Cloud-a.

<http://www.WhatsApp.com>  
<http://astah.net/editions/professional>  
<http://www.gitlab.com>  
<http://www.java.com>  
<http://developer.android.com>  
<https://firebase.google.com/>

## 5.2 Ispitivanje programskog rješenja

### 5.2.1 Ispitivanje komponenti

#### Funkcionalnost 1: pokretanje registracije u sustav

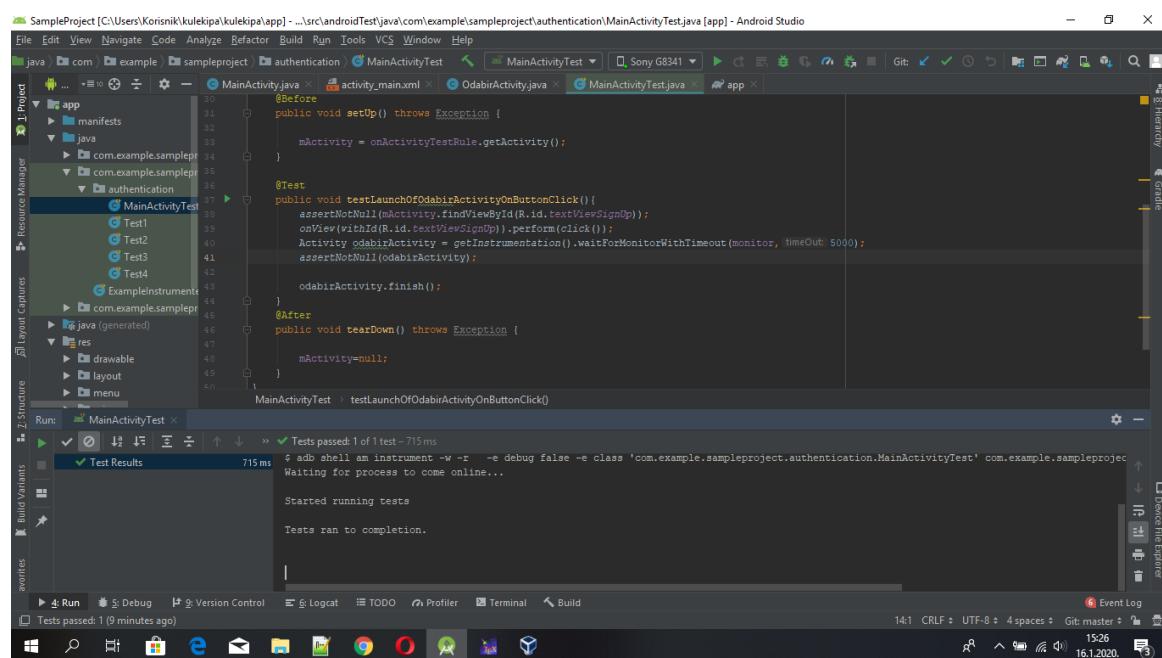
Neregistrirani korisnik odabire mogućnost registracije na početnoj stranici.

Ulaz:

1. Pritisak na gumb "REGISTRIRAJ SE"
2. Otvaranje stranice odabira željene uloge

Očekivani izlaz:

1. Otvaranje stranice odabira uloge



```
SampleProject [C:\Users\Korisnik\kulEkipa\kulEkipa\app] - ...\\src\\androidTest\\java\\com\\example\\sampleproject\\authentication\\MainActivityTest.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools WCS Window Help
File Edit View Navigate Code Analyze Refactor Build Run Tools WCS Window Help
MainActivity.java activity_main.xml authentication MainActivityTest OdabirActivity.java MainActivityTest.java app
@Before
public void setUp() throws Exception {
    mActivity = mActivityTestRule.getActivity();
}
@Test
public void testLaunchOfOdabirActivityOnButtonClick() {
    assertNotNull(mActivity.findViewById(R.id.textViewSignUp));
    onView(withId(R.id.textViewSignUp)).perform(click());
    Activity odabirActivity = getInstrumentation().waitForMonitorWithTimeout(monitor, timeOut: 5000);
    assertNotNull(odabirActivity);
}
@After
public void tearDown() throws Exception {
    mActivity.finish();
}
mActivity=null;
}
MainActivityTest > testLaunchOfOdabirActivityOnButtonClick()
Run: MainActivityTest
Test Results 715 ms
Tests passed: 1 of 1 test - 715 ms
adb shell am instrument -w -f -e debug false -e class 'com.example.sampleproject.authentication.MainActivityTest' com.example.sampleprojec
Waiting for process to come online...
Started running tests
Tests ran to completion.
Event Log
14:1 CRLF: UTF-8: 4 spaces: Git: master: 15:26 16.1.2020.
Run Debug Version Control Logcat TODO Profiler Terminal Build
Tests passed: 1 (9 minutes ago)
Windows Start Task View Taskbar Event Viewer Device File Explorer
14:1 CRLF: UTF-8: 4 spaces: Git: master: 15:26 16.1.2020.
```

Slika 5.1: Unit Test 1

Rezultat: Test prolazi. Funkcionalnost je ispravna.

## Funkcionalnost 2: Parsiranje datuma

Provjerava se ispravnost formatiranja datuma za pohranu u bazu podataka.

Ulaz:

1. Ulaz je Javin kalendar s parametrima (2020,0,1) tj datum 01.01.2020.

Očekivani izlaz:

1. 01/01/20

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "SampleProject". It contains an "app" module with Java and AndroidManifest.xml files. A "test" folder contains a "java" folder with a package "com.example.sampleproject" and a class "ExampleUnitTest".
- Code Editor:** The code for "ExampleUnitTest.java" is displayed. It contains a test method "parsiranjeIsCorrect" which creates a Calendar object for January 1, 2020, and then uses a SimpleDateFormat to format it according to the pattern "MM/dd/yy". An assertion checks if the formatted string equals "01/01/20".
- Run Tab:** The "Run" tab shows the test has passed: "Tests passed: 1 of 1 test - 2 ms". The specific test "parsiranjeIsCorrect" is highlighted.
- Event Log:** The log shows "Cannot Run Git" due to a file not found: "git.exe".
- Bottom Status Bar:** The status bar shows the date and time: "29.29 CRLF+ UTF-8+ 4 spaces+ Git: master 17:38 16.1.2020".

Slika 5.2: Unit Test 2

Rezultat: Test prolazi. Funkcionalnost je ispravna.

### Funkcionalnost 3: Usporedba datuma

Provjerava se ispravnost uspoređivanja parsiranog datuma s neparsiranim radi implementiranja restrikcija vezano uz vremensku domenu rada aplikacije.

Ulaz:

1. Ulazi su instanca Javinog kalendarja i parsirani datum oblika prilagođenog potrebama baze

Očekivani izlaz:

1. 1 (trenutni datum dobiven iz Javinog kalendarja je veći od parsiranog)

The screenshot shows the Android Studio interface with the project 'SampleProject' open. The code editor displays the file 'UTest3.java' which contains Java code for testing date parsing. The test method 'parsiranjeIsCorrect' compares the current calendar time with a parsed date ('01/01/20'). The test passes with a duration of 3 ms. The bottom status bar shows the date as 16.1.2020.

```
public class UTest3 {
    SimpleDateFormat formatter = new SimpleDateFormat("dd/MM/yyyy");
    @Test
    public void parsiranjeIsCorrect() {
        Calendar c = Calendar.getInstance();
        int provjera = 0;
        try {
            provjera = c.getTime().compareTo(formatter.parse("01/01/20"));
        } catch (ParseException e) {
            e.printStackTrace();
        }
        assertEquals(1, provjera);
    }
}
```

Slika 5.3: Unit Test 3

Rezultat: Test prolazi. Funkcionalnost je ispravna.

#### Funkcionalnost 4: Zauzetost organizatora

Provjerava se podudara li se trajanje organizatorovog već postojećeg događaja s onime na koji se želi prijaviti.

Ulaz:

1. Ulazi su datumi oba događaja.

Očekivani izlaz:

1. Može

The screenshot shows the Android Studio interface with the project 'SampleProject' open. The code editor displays the UTest4.java file, which contains a unit test for checking if two times are equal. The test passes, as indicated by the green checkmark in the 'Run' tab. The bottom status bar shows the date and time as 16.1.2020, 18:14.

```
import static org.junit.Assert.assertEquals;  
  
public class UTest4 {  
  
    @Test  
    public void zauzetostOrganizatora() {  
        String organizator = "ne može";  
        if ("18:30".compareTo("19:30") < 0 || "16:30".compareTo("20:30") > 0) organizator="može";  
  
        assertEquals("može", organizator);  
    }  
}
```

Slika 5.4: Unit Test 4

Rezultat: Test prolazi. Funkcionalnost je ispravna.

### Funkcionalnost 5: Neimplementirana funkcionalnost zauzetosti organizatora

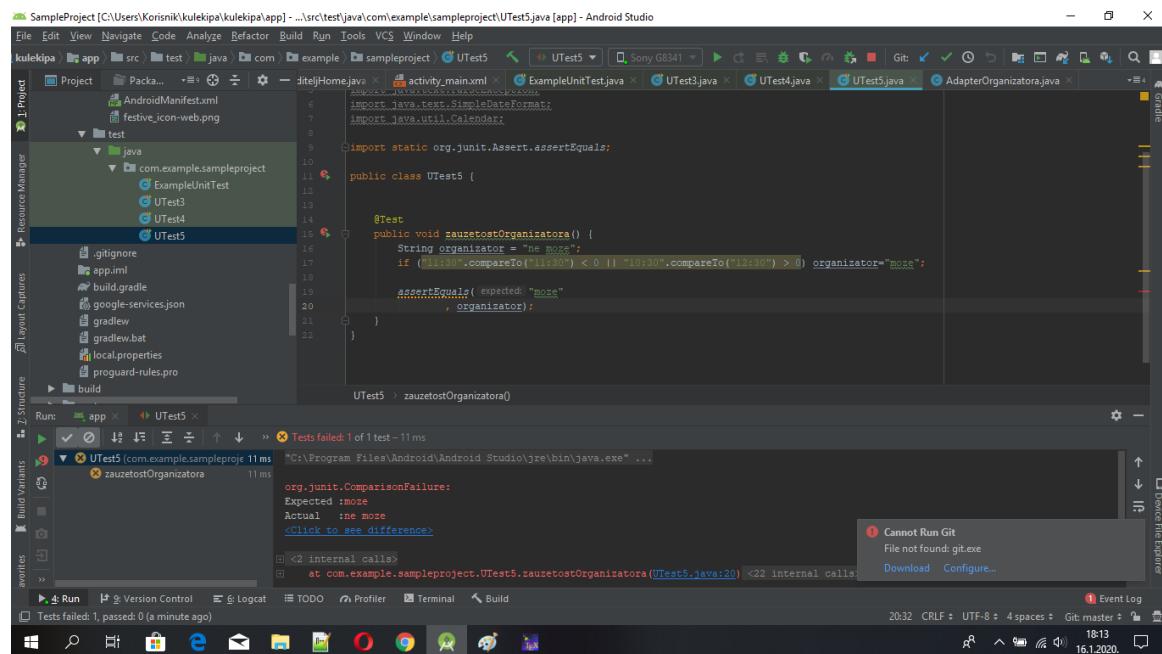
Provjerava se hoće li se organizator moći prijaviti na događaj koji počinje u isto vrijeme u koje njegov već postojeći završava.

Ulaz:

1. Ulazi su datumi oba događaja.

Očekivani izlaz:

1. Može



```

SampleProject [C:\Users\Korisnik\kulekipa\kulekipa\app] - ...src\test\java\com\example\sampleproject\UTest5.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VC$ Window Help
kulekipa app src test java com example sampleproject UTest5 UTest UTest3 UTest4 UTest5
Project Packag... + test + java com.example.sampleproject
Resource Manager
  .gitignore
  app.aiml
  build.gradle
  google-services.json
  gradlew
  gradlew.bat
  local.properties
  proguard-rules.pro
  build
Run: app x UTest5 x
Layout Captures
Build Variants
  Tests failed: 1 of 1 test - 11 ms
  UTest5 (com.example.sampleprojec 11 ms
    "C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
      org.junit.ComparisonFailure:
      Expected :može
      Actual   :ne može
      <click to see difference>
      <2 internal calls>
      at com.example.sampleproject.UTest5.zauzetostOrganizatora(UTest5.java:20) <22 internal calls>
  )
  Cannot Run Git
  File not found: git.exe
  Download Configure...
  Event Log
  20:32 CRLF+ UTF-8+ 4 spaces+ Git master+
  18:13 16.1.2020.
  Run Version Control Profiler Terminal Build
  1 Tests failed: 1, passed: 0 (a minute ago)
  Windows Taskbar

```

Slika 5.5: Unit Test 5

Rezultat: Test pada, ali to ne ometa rad aplikacije jer aplikacija uzima u obzir da organizator nije vremenski putnik te ne može biti na dva mesta u isto vrijeme.

**Funkcionalnost 6: Zauzetost organizatora - rubni slučaj**

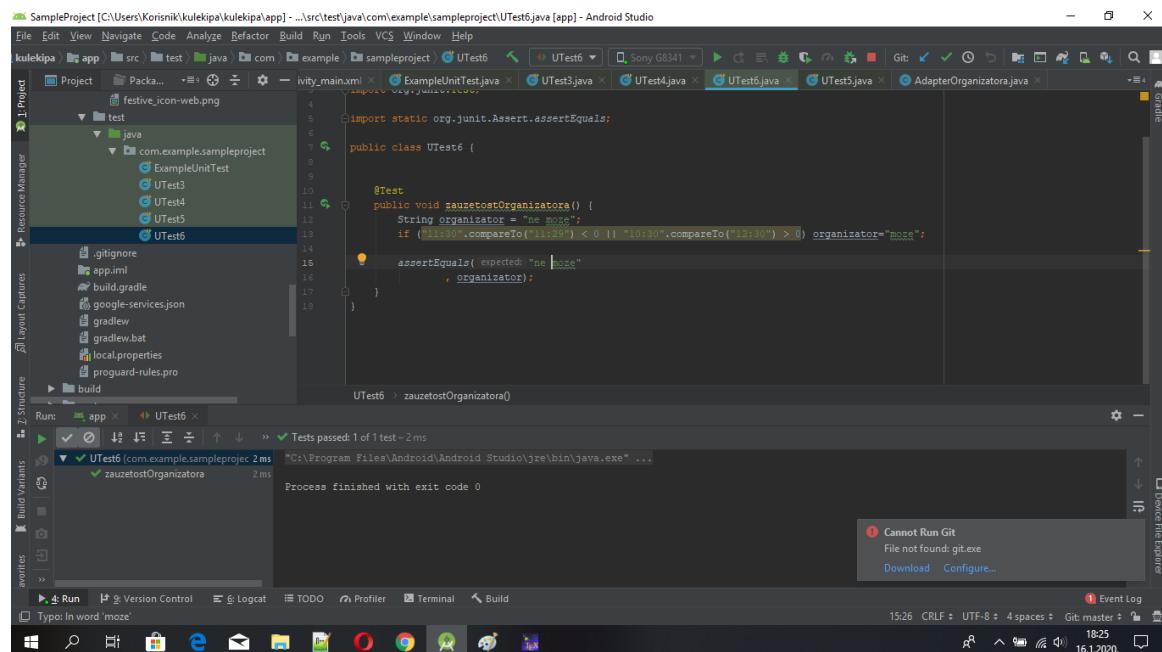
Provjerava se podudara li se trajanje organizatorovog već postojećeg događaja s onime na koji se želi prijaviti ako je početak drugog događaja minutu prije kraja prvog.

Ulaz:

1. Ulazi su datumi oba događaja.

Očekivani izlaz:

1. Ne može



```
SampleProject [C:\Users\Korisnik\kulekipa\kulekipa\app] - ...\\src\\test\\java\\com\\example\\sampleproject\\UTest6.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VC$ Window Help
kulekipa app src test java com example sampleproject UTest6 UTest UTest3 UTest4 UTest5 UTest6 UTest7 AdapterOrganizator.java
Project : I: Project
  - test
    - festive_icon-web.png
      - java
        - com.example.sampleproject
          - ExampleUnitTest
          - UTest3
          - UTest4
          - UTest5
          - UTest6
Resource Manager
  - .gitignore
  - app.iml
  - build.gradle
  - google-services.json
  - gradlew
  - gradlew.bat
  - local.properties
  - proguard-rules.pro
  - build
  - Run: app x UTest6 x
  - Build Variants
  - Favorites
  - Run Version Control Log TODO Profiler Terminal Build
  - Type: In word 'može'
  - Windows Taskbar: Run Task View Start Mail Internet Explorer Google Chrome File Explorer Task Manager

  import static org.junit.Assert.assertEquals;
public class UTest6 {
    @Test
    public void zauzetostOrganizatora() {
        String organizator = "ne može";
        if ("11:30".compareTo("12:30") < 0 || "10:30".compareTo("12:30") > 0) organizator="može";
        assertEquals("ne može", organizator);
    }
}
UTest6 > zauzetostOrganizatora()
Run: app x UTest6 x
  Tests passed: 1 of 1 test - 2 ms
  UTest6 (com.example.sampleprojec 2 ms
    zauzetostOrganizatora 2 ms
      "C:\Program Files\Android\Android Studio\jre\bin\java.exe" ...
      Process finished with exit code 0
  Cannot Run Git
  File not found: git.exe
  Download Configure...
Event Log
15:26 CRLF + UTF-8 + 4 spaces + Git master + 18:25
16.1.2020.
```

Slika 5.6: Unit Test 6

Rezultat: Test prolazi. Funkcionalnost je ispravna.

## 5.2.2 Ispitivanje sustava

### Funkcionalnost 1: dodavanje djelatnosti u profilu izvođača

Korisnik prijavljen na svoj profil u ulozi izvođača dodaje djelatnost koja ne postoji već u bazi.

Ulaz:

1. Pritisak na gumb Specijalizacije
2. Otvaranje stranice specijalizacija
3. Unos teksta "The Job"
4. Pritisak na gumb "Dodaj"

Očekivani izlaz:

1. Otvaranje stranice specijalizacija
2. Prikaz djelatnosti "The Job" među postojećim djelatnostima izvođača

Rezultat: Zadovoljeni svi očekivani rezultati. Aplikacija prolazi test.

```

    package com.example.sampleproject.authentication;
    import androidx.test.ext.junit.runners.AndroidJUnit4;
    import androidx.test.filters.LargeTest;
    import androidx.test.rule.ActivityTestRule;
    import androidx.test.uiautomator.By;
    import androidx.test.uiautomator.UiDevice;
    import androidx.test.uiautomator.UiObject;
    import androidx.test.uiautomator.UiObjectNotFoundException;
    import androidx.test.uiautomator.UiSelector;
    import org.junit.Before;
    import org.junit.Test;
    import org.junit.runner.RunWith;

    @RunWith(AndroidJUnit4.class)
    @LargeTest
    public class Test1 {
        private ActivityTestRule<MainActivity> mActivityRule;
        private UiDevice mDevice;

        @Before
        public void setup() {
            mActivityRule = new ActivityTestRule<MainActivity>(MainActivity.class);
            mDevice = UiDevice.getInstance(mActivityRule.getActivity().getApplicationContext());
        }

        @Test
        public void test1() throws UiObjectNotFoundException {
            // Open specialization page
            mDevice.findObject(By.text("Specijalizacija")).click();
            mDevice.findObject(By.text("Specijalizacija")).click();

            // Enter job title
            mDevice.findObject(By.text("The Job")).click();
            mDevice.findObject(By.text("The Job")).click();

            // Click add button
            mDevice.findObject(By.text("Dodaj")).click();
            mDevice.findObject(By.text("Dodaj")).click();

            // Verify job added
            mDevice.findObject(By.text("The Job")).click();
            mDevice.findObject(By.text("The Job")).click();
        }
    }

```

Slika 5.7: Test 1

## Funkcionalnost 2: pregled profila organizatora

Korisnik prijavljen na svoj profil u ulozi voditelja (sk002@vod.com) pregledava profil organizatora nakon završenog događaja.

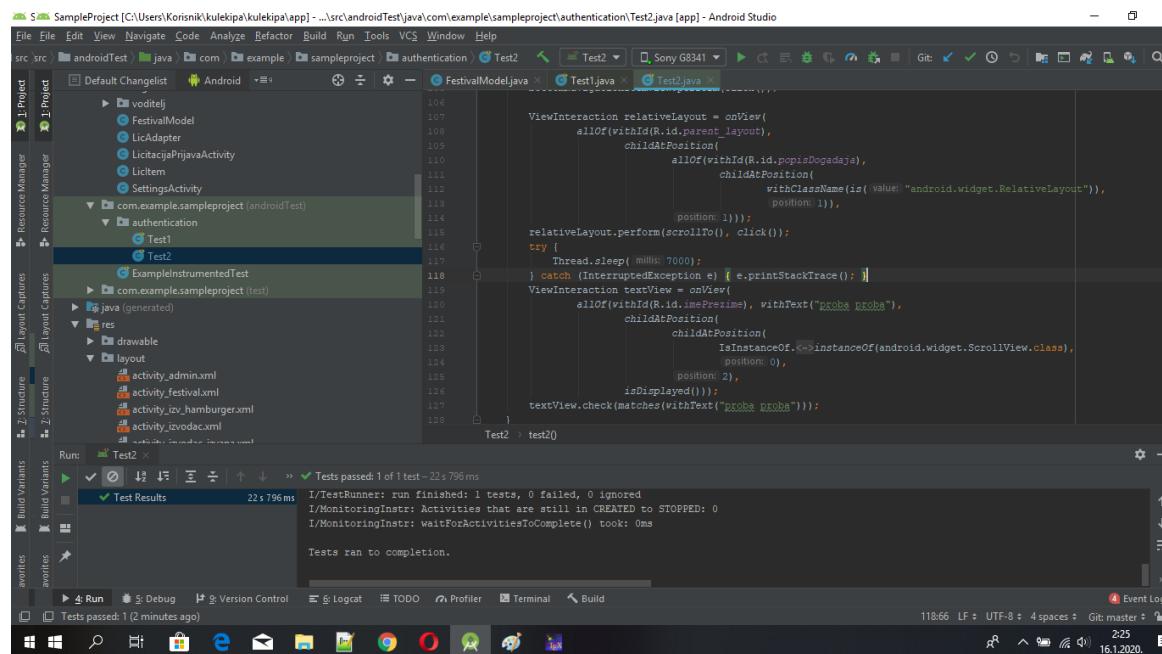
Ulaz:

1. Pritisak na gumb "Info"
2. Otvaranje stranice s popisom završenih događaja
3. Klik na gumb "Pregledaj profil organizatora" kraj događaja "Kušanje čokolade"

Očekivani izlaz:

1. Otvaranje stranice organizatora s imenom "proba proba" uz mogućnost komentiranja

Rezultat: Zadovoljeni svi očekivani rezultati. Aplikacija prolazi test.



```

SampleProject [C:\Users\Koriscnik\kulekipa\kulekipa\app] - ...\src\androidTest\java\com\example\sampleproject\authentication\Test2.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
src src androidTest java com example sampleproject authentication Test2 Test2 Test2.java Test2.java
1 Project 1 Project
Default Changelist Android
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
FestivalModel.java Test2.java Test2.java
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
ViewInteraction relativeLayout = onView(
    allOf(withId(R.id.parent_layout),
        childAtPosition(
            allOf(withId(R.id.popisDogađaja),
                childAtPosition(
                    withClassName(is("android.widget.RelativeLayout")),
                    position: 1))),
    relativeLayout.perform(scrollTo(), click());
try {
    Thread.sleep( millis: 7000 );
} catch (InterruptedException e) { e.printStackTrace(); }
ViewInteraction textView = onView(
    allOf(withId(R.id.textView), withText("proba proba"),
        childAtPosition(
            childAtPosition(
                isInstanceOf(<-- instanceof(android.widget.ScrollView.class),
                position: 0),
                isDisplayed())));
    textView.check(matches(withText("proba proba")));
}
Test2 > test20

```

Run: Test2

Build Variants: avontur avontur

Build Results: Tests passed: 1 of 1 test - 22s 796ms

I/TestRunner: run finished: 1 tests, 0 failed, 0 ignored

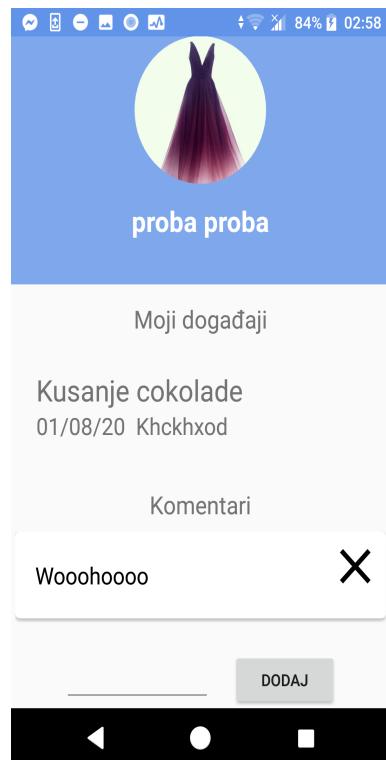
I/MonitoringInstr: Activities that are still in CREATED to STOPPED: 0

I/MonitoringInstr: waitForActivitiesToComplete() took: 0ms

Tests ran to completion.

118:66 LF: UTF-8 4 spaces Git: master 2:25 16.1.2020 Event Log

Slika 5.8: Test 2.1



Slika 5.9: Test 2.2

### Funkcionalnost 3: pregled profila izvođača - rubni slučaj

Korisnik prijavljen na svoj profil u ulozi organizatora pregledava profil izvođača nakon završene licitacije, ali ne i događaja.

Ulas:

1. Pritisak na gumb "Moji događaji"
2. Otvaranje stranice s popisom aktivnih i završenih licitacija
3. Klik na završenu licitaciju "2.radionica ozvučenje"
4. Otvaranje stranice s opisom pobjedničke ponude
5. Klik na gumb "Odi na profil"

Očekivani izlaz:

1. Otvaranje stranice izvođača s imenom "Toni Žalac" uz onemogućeno komentiranje

Rezultat: Zadovoljeni svi očekivani rezultati. Aplikacija prolazi test.

```

106
107     ViewInteraction relativeLayout = onView(
108         allOf(withId(R.id.parent_layout),
109             childAtPosition(
110                 allOf(withId(R.id.popisDogadjaja),
111                     childAtPosition(
112                         withClassName(is("android.widget.RelativeLayout"))),
113                     withId(R.id.recycler_view))));
114
115     onView(withId(R.id.ozvucenje)).check(matches(isDisplayed()));
116
117     // Added a sleep statement to match the app's execution delay.
118     // The recommended way to handle such scenarios is to use Espresso idling resources:
119     // https://google.github.io/android-testing-support-library/docs/espresso/idling-resource/index.html
120     try {
121         Thread.sleep( millis: 7000 );
122     } catch (InterruptedException e) {
123         e.printStackTrace();
124     }
125
126     ViewInteraction textView = onView(
127         allOf(withId(R.id.imPresime), withText("Toni Željez")),
128             childAtPosition(
129                 childAtPosition(
130                     instanceOf(android.widget.ScrollView.class),
131                     position: 0),
132                     position: 0),
133                     isDisplayed())));
134
135     textView.check(matches(withText("Toni Željez")));
136
137     Test3 appCompatButton2.perform(click());
138
139     // The recommended way to handle such scenarios is to use Espresso idling resources:
140     // https://google.github.io/android-testing-support-library/docs/espresso/idling-resource/index.html
141     try {
142         Thread.sleep( millis: 7000 );
143     } catch (InterruptedException e) {
144         e.printStackTrace();
145     }
146
147     ViewInteraction textView = onView(
148         allOf(withId(R.id.imPresime), withText("Toni Željez")),
149             childAtPosition(
150                 childAtPosition(
151                     instanceOf(android.widget.ScrollView.class),
152                     position: 0),
153                     position: 0),
154                     isDisplayed())));
155
156     textView.check(matches(withText("Toni Željez")));
157
158 }

```

Slika 5.10: Test 3.1



Slika 5.11: Test 3.2

#### Funkcionalnost 4: primjer neimplementirane funkcionalnosti

Korisnik prijavljen na svoj profil u ulozi voditelja pregledava profil organizatora

nakon što mu je već ostavio komentar u kontekstu događaja u kojem ga pregledava.

Ulez:

1. Pritisak na gumb "Info"
2. Otvaranje stranice s popisom završenih događaja
3. Klik na gumb "Pregledaj profil organizatora" kraj događaja "Kušanje čokolade"

Očekivani izlaz:

1. Otvaranje stranice organizatora s imenom "proba proba" uz onemogućeno komentiranje

Rezultat: Test pada. Komentiranje je omogućeno.

Objašnjenje: Ova greška ne smeta radu aplikacije jer se tretira kao da se voditelj predomislio i odlučio ostaviti drukčiji komentar organizatoru u kontekstu istog događaja te se pamti samo najnoviji komentar. Ovu grešku stoga zanemaruјemo.

```

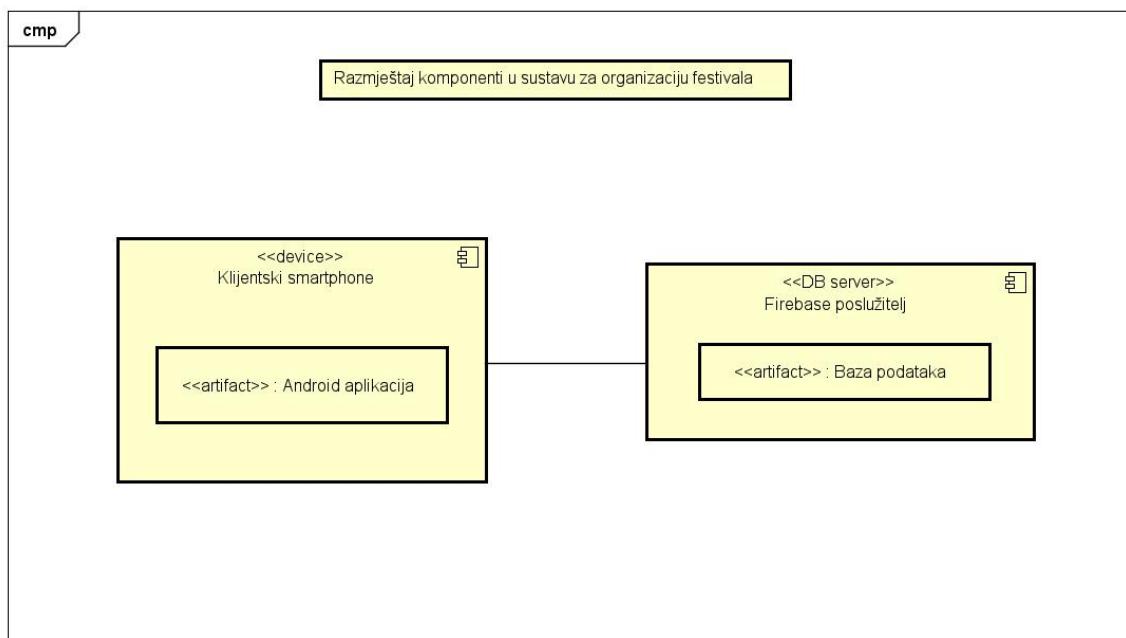
SampleProject [C:\Users\Korisnik\kulekipa\kulekipa\app] - ...src\androidTest\java\com\example\sampleproject\authentication\Test2.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
src src androidTest java com example sampleproject authentication Test2 Test2 Test2.java Sony G8341
Default Changelog Project LicAdapter LicitacijaPrijavaActivity Licitem SettingActivity
FestivalModel.java Test1.java Test2.java Test3.java Test4.java
106
107
108
109
110
111
112
ViewInteraction relativeLayout = onView(
    allOf(withId(R.id.parent_layout),
        childAtPosition(
            allOf(withId(R.id.popisDogađaja),
                childAtPosition(
                    withClassName(is("value: " + android.widget.RelativeLayout)),
                    position: 1))
    )
);
113
114
115
116
117
118
119
120
121
122
123
124
try {
    Thread.sleep( millis: 7000 );
} catch (InterruptedException e) {
    e.printStackTrace();
}
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
ViewInteraction button = onView(
    allOf(withId(R.id.dodataKomentar),
        childAtPosition(
            childAtPosition(
                isInstanceOf(<> instanceof android.widget.ScrollView.class),
                position: 0),
                position: 0));
    isDisplayed()));
button.check(doesNotExist());
private static Matcher<View> childAtPosition(
    final Matcher<View> parentMatcher, final int position) {
}

```

Slika 5.12: Test 4

### 5.3 Dijagram razmještaja

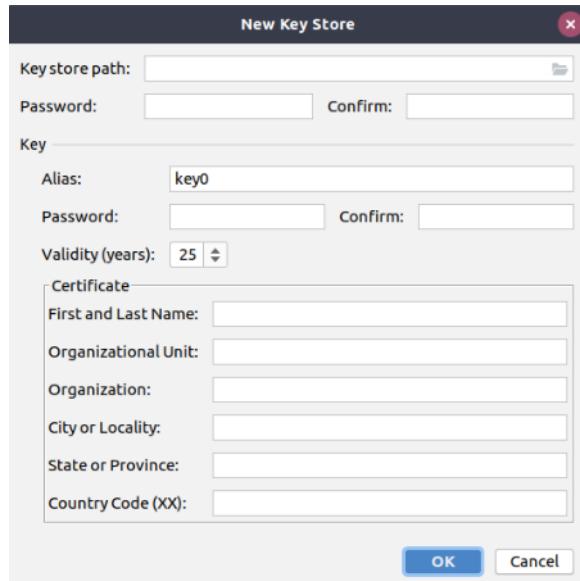
Na pametnom mobitelu klijenta je instalirana android aplikacija pomoću koje klijent upravlja svim mogućnostima sustava. Na udaljenom poslužitelju se nalazi baza podataka koja sadrži sve potrebne informacije koje su korisniku i aplikaciji potrebne, dok se komunikacija same aplikacije i baze podataka odvija se preko IP mreže.



Slika 5.13: Dijagram razmještaja

## 5.4 Upute za puštanje u pogon

Za izgradnju aplikacije koristimo *IntelliJ IDEA* razvojno sučelje. Moguće je korištenje *Android Studio* razvojnog sučelja te je postupak u oba slučaja identičan. Nakon odabira opcije *Build* na glavnoj izbornoj traci, moguće su dvije varijante: *debug* i *release*. Odabiremo *release* verziju s obzirom na pogodnosti u kontekstu optimizacije koda. Za format izgrađene aplikacije odabiremo *APK*. Nadalje, odabiremo izgradnju potpisane (eng. *signed*) aplikacije. Potpisana aplikacija sadrži certifikat koji se ne mijenja za vrijeme životnog ciklusa aplikacije; Za svako ažuriranje aplikacije, dolazi do usporedbe potpisa: potpis aplikacije koja se ažurira mora odgovarati potpisu nove verzije koja se preuzima. Dakle, razvojni tim mora posjedovati identičan privatni ključ koji je generiran za vrijeme izgradnje. Privatni ključ, ako je moguće, ne posjeduje nitko osim razvojnog tima. Obrazac za generiranje ključa prikazan je na slici 5.1. Nakon ispune obrasca, datoteka s ključem spremna se na mjesto gdje može biti ponovno dohvaćena po potrebi.



Slika 5.14: Generiranje ključa za potpis aplikacije

Izgrađenu aplikaciju moguće je objaviti na platformama poput *Google Play* ili ponuditi u obliku datoteke za samostalno preuzimanje. Aplikacija je dostupna na *Aptoide* platformi s obzirom na jednostavnost stvaranja korisničkog računa i besplatno korištenje, ali se radi prethodno spomenutog certifikata može objaviti na bilo kojoj drugoj platformi s osnovnim sigurnosnim zahtjevima (uključujući i *Google Play*). Izgradnja i objava *APK* datoteke na prethodno opisani način dovoljna

je za uspješno puštanje aplikacije u pogon.

## 6. Zaključak i budući rad

Početkom semestra dobili smo zadatak razvijanja aplikacije za organiziranje festivala kako bismo omogućili voditeljima, organizatorima te izvođačima uspješno i efikasno koordiniranje u provođenju festivala. Tijekom ovih 17 tjedana uspješno smo pretvorili naš cilj u djelo.

Unatoč limitiranom znanju te oskudnom iskustvu članova našeg tima, uspješno smo se organizirali i podijelili poslove kako bismo ostvarili dobiven zadatak. Izgradnju aplikacije smo proveli u nekoliko faza.

U prvoj fazi, fokusirali smo se na stvaranje osnovnih funkcija svake aplikacije kao što su registracija, prijavljivanje te povezivanje profila korisnika s bazom podataka. Uz pravilnu podjelu poslova, gdje je svaki član uspješno obavio dobiven posao, uspjeli smo ostvariti sve ciljeve prve faze, jedan od kojih je bilo dokumentiranje te izrada obrazaca uporabe i dijagrama vezanih uz njih.

Druga faza je sadržavala ostvarivanje funkcionalnih zahtjeva naše aplikacije poput omogućavanja otvaranja novih festivala i događaja voditelju, ostvarivanje licitacija za organizatora i izvođača te omogućavanje ostalih opcija vezanih uz profile. Svaki naš korak u izradi aplikacije bio je popraćen odgovarajućim dokumentiranjem.

Treća i posljednja faza je bilo ostvarivanje dodatnih mogućnosti te ispravljanje svih naših propusta. Svaki član tima je savjesno obavljao svoju ulogu te smo zbog toga uspjeli sve završiti na vrijeme.

Naši budući planovi uključuju poboljšavanje izgleda i funkcioniranja aplikacije te dodavanja novih mogućnosti poput kalendara koji bi prikazivao zauzetost korisnika, te poboljšanog sustava vrednovanja gdje bi voditelj (ili organizator) mogao dodijeliti ocjenu od 1 do 5 organizatoru (ili izvođaču) nakon izvršavanja njihovih dužnosti.

Sve u svemu, ovaj projekt je bio jako korisno iskustvo za sve članove tima te smo zadovoljni onim što smo napravili te pogotovo onim što smo naučili.



# Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. The Unified Modeling Language, <https://www.uml-diagrams.org/>
3. Developer Guides — Android Developers, <https://developer.android.com/guide>
4. Firebase Tutorial for Android : Getting Started — raywenderlich.com, <https://www.raywenderlich.com/5114-firebase-tutorial-for-android-getting-started>

# Indeks slika i dijagrama

3.1	Dijagram obrasca uporabe, funkcionalnost neregistriranog korisnika, registriranog korisnika, administratora i izvođača . . . . .	20
3.2	Dijagram obrasca uporabe, funkcionalnost organizatora . . . . .	21
3.3	Dijagram obrasca uporabe, funkcionalnost voditelja . . . . .	22
3.4	Sekvencijski dijagram za UC7 . . . . .	23
3.5	Sekvencijski dijagram za UC17 . . . . .	24
3.6	Sekvencijski dijagram za UC21 . . . . .	25
3.7	Sekvencijski dijagram za UC23 . . . . .	26
4.1	Skica sustava . . . . .	28
4.2	Struktura <i>Firebase Firestore</i> modela . . . . .	29
4.3	Hipotetska struktura <i>Firebase Firestore</i> modela . . . . .	30
4.4	ER dijagram baze podataka . . . . .	36
4.5	Dijagram razreda <i>Firebase usluga</i> . . . . .	38
4.6	Dijagram razreda sadržanih u paketu <i>authentication</i> . . . . .	41
4.7	Dijagram razreda sadržanih u paketu <i>authentication</i> . . . . .	41
4.8	Dijagram razreda sadržanih u paketu <i>organizator</i> . . . . .	42
4.9	Dijagram razreda sadržanih u paketu <i>organizator</i> . . . . .	43
4.10	Dijagram razreda sadržanih u paketu <i>voditelj</i> . . . . .	44
4.11	Dijagram razreda sadržanih u paketu <i>voditelj</i> . . . . .	45
4.12	Dijagram razreda sadržanih u paketu <i>izvođač</i> . . . . .	46
4.13	Dijagram razreda sadržanih u paketu <i>izvođač</i> . . . . .	47
4.14	Dijagram razreda sadržanih u paketu <i>common</i> . . . . .	48
4.15	Dijagram razreda sadržanih u paketu <i>common</i> . . . . .	49
4.16	Dijagram stanja voditelja . . . . .	50
4.17	Dijagram aktivnosti otvaranja novog događaja . . . . .	52
4.18	Dijagram komponenti . . . . .	53
5.1	Unit Test 1 . . . . .	56
5.2	Unit Test 2 . . . . .	57
5.3	Unit Test 3 . . . . .	58

5.4	Unit Test 4	59
5.5	Unit Test 5	60
5.6	Unit Test 6	61
5.7	Test 1	62
5.8	Test 2.1	63
5.9	Test 2.2	64
5.10	Test 3.1	65
5.11	Test 3.2	65
5.12	Test 4	66
5.13	Dijagram razmještaja	67
5.14	Generiranje ključa za potpis aplikacije	68
6.1	Ukupna aktivnost	79
6.2	Aktivnost po članovima 1	79
6.3	Aktivnost po članovima 2	79
6.4	Aktivnost po članovima 3	80
6.5	Aktivnost po članovima 4	80

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### 1. sastanak

- Datum: 12.10.2019.
- Prisustvovali: S.Krznarić, M.Matokanović, T.Jurišić, M.Bartaković, N.Gibanica, I.Šimić, Š.Kožul
- Teme sastanka:
  - alati za izradu programa
  - podjela početnih zadataka

### 2. sastanak

- Datum: 22.10.2019.
- Prisustvovali: S.Krznarić, T.Jurišić, M.Bartaković, N.Gibanica, I.Šimić, Š.Kožul
- Teme sastanka:
  - podjela zadataka vezanih uz dokumentaciju
  - podjela zadataka vezanih uz spajanje na bazu podataka

### 3. sastanak

- Datum: 29.10.2019.
- Prisustvovali: S. Krznarić, T. Jurišić, M. Bartaković, M. Matokanović, I. Šimić, Š. Kožul, N. Gibanica
- Teme sastanka:
  - podjela zadataka vezanih uz izradu korisničkih profila
  - podjela zadataka vezanih uz dokumentaciju

### 4. sastanak

- Datum: 8.11.2019.
- Prisustvovali: S. Krznarić, T. Jurišić, M. Matokanović, I. Šimić, Š. Kožul, N. Gibanica
- Teme sastanka:
  - dogovor za daljnje razvijanje aplikacije

- manje promjene u podjeli poslova

5. sastanak

- Datum: 5.12.2019.
- Prisustvovali: S.Krznarić, M.Matokanović, T.Jurišić, M.Bartaković, N.Gibanica, I.Šimić, Š.Kožul
- Teme sastanka:
  - podjela zadataka vezanih uz implementiranje licitacija organizatora i izvođača
  - podjela zadataka vezanih uz dokumentaciju

6. sastanak

- Datum: 9.1.2020.
- Prisustvovali: S.Krznarić, M.Matokanović, T.Jurišić, M.Bartaković, N.Gibanica, I.Šimić, Š.Kožul
- Teme sastanka:
  - podjela zadataka vezanih uz ispitivanje programskog rješenja i puštanje aplikacije u pogon

7. sastanak

- Datum: 14.1.2020.
- Prisustvovali: S.Krznarić, M.Matokanović, T.Jurišić, M.Bartaković, N.Gibanica, I.Šimić, Š.Kožul
- Teme sastanka:
  - finalni zadaci vezani uz javnu objavu aplikacije i uređivanje dokumentacije

## Tablica aktivnosti

### Kontinuirano osvježavanje

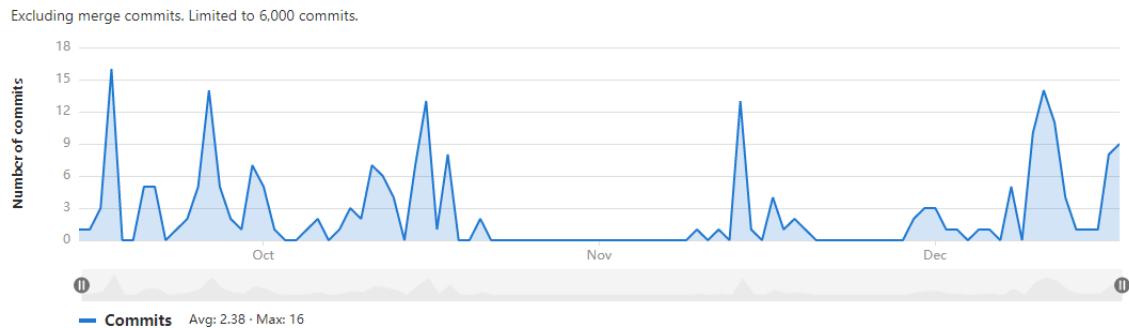
Napomena: Doprinose u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

	Sanja Krznarić	Šime Kožul	Izabela Šimić	Nika Gibanica	Monika Matokanović	Tesa Jurišić	Marko Bartaković
Upravljanje projektom	32	0	0	0	0	0	0
Opis projektnog zadatka	2	2	5	0	0	0	0
Funkcionalni zahtjevi	1	3	1	0	0	0	0
Opis pojedinih obrazaca	2	0	2	1	0	0	2
Dijagram obrazaca	0	0	0	3	0	0	0
Sekvencijski dijagrami	0	0	0	3	0	0	0
Opis ostalih zahtjeva	0	0	0	2	0	0	0
Arhitektura i dizajn sustava	0	0	0	0	0	7	0
Baza podataka	5	4	3	0	2	5	0
Dijagram razreda	0	0	0	0	0	7	0
Dijagram stanja	1	0	5	0	0	0	0
Dijagram aktivnosti	0	0	3	0	0	0	0
Dijagram komponenti	2	0	4	0	0	0	0
Korištene tehnologije i alati	0	0	0	0	0	0	4
Ispitivanje komponenti	7	0	7	2	0	0	0
Ispitivanje sustava	7	0	7	4	0	0	0
Dijagram razmještaja	0	0	0	0	0	0	2
Upute za puštanje u pogon	0	0	0	0	0	7	0
Dnevnik sastajanja	1	0	1	0	0	0	0
Zaključak i budući rad	0	0	3	0	0	0	0
Popis literature	1	0	0	0	0	0	0

Dodatne stavke	Sanja Krznarić	Šime Kožul	Izabela Šimić	Nika Gibanica	Monika Matokanović	Tesa Jurišić	Marko Bartaković
izrada početnih stranica (log in i registracija)	2	0	4	4	0	4	0
izrada baze podataka	0	0	0	0	0	7	0
spajanje s bazom podataka	7	3	5	2	3	7	2
back end	14	9	9	7	8	10	2
izrada profilnih stranica po ulogama	4	2	3	4	6	4	3
propusnice	3	8	3	0	0	0	0
licitacije	8	2	4	0	3	0	0

## Dijagrami pregleda promjena

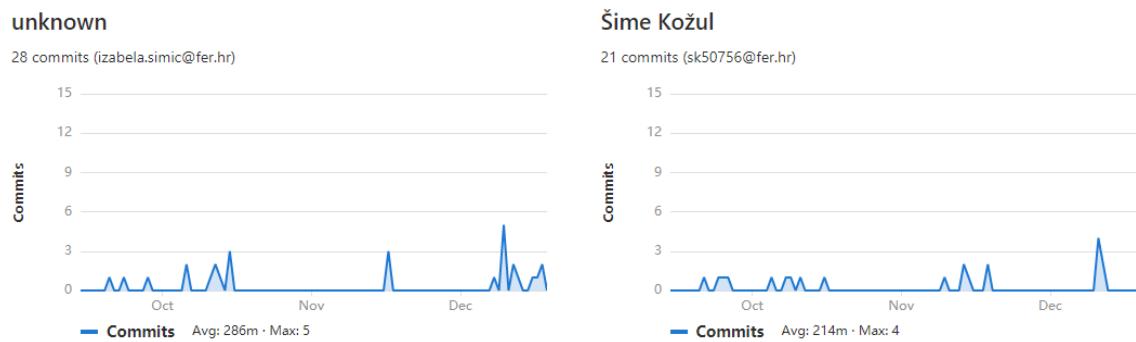
### dio 2. revizije



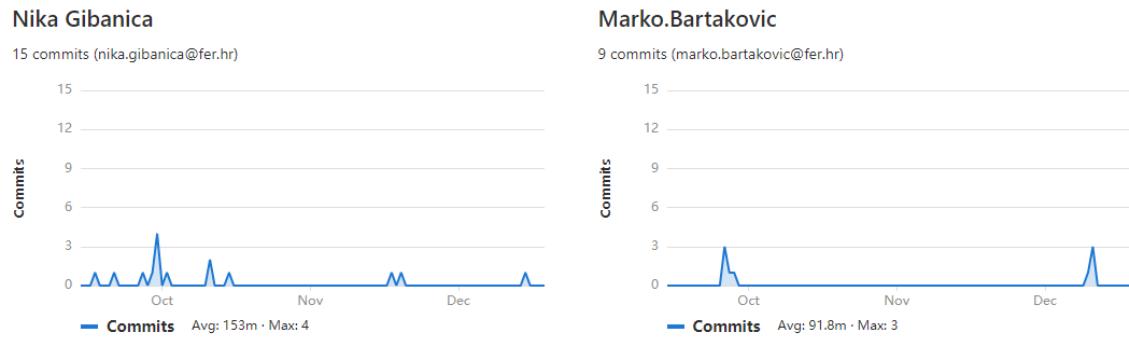
Slika 6.1: Ukupna aktivnost



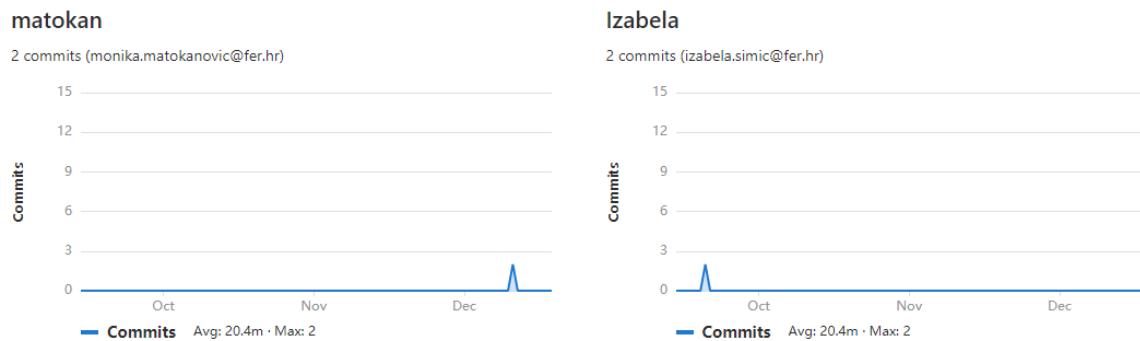
Slika 6.2: Aktivnost po članovima 1



Slika 6.3: Aktivnost po članovima 2



Slika 6.4: Aktivnost po članovima 3



Slika 6.5: Aktivnost po članovima 4

*Napomena: prikazi aktivnosti članova nisu u potpunosti ispravno generirani; nekim članovima nije priznat sav doprinos, a nekim članovima je aktivnost generirana u više grafova.*