

UNIVERZITET U BEOGRADU

MASTER RAD

---

Idejni projekat informacionog sistema za  
međunarodni kulturni centar

---

*Autor:*  
Ana BAJIĆ

*Mentor:*  
Prof. dr. Saša MALKOV



23. septembar 2020

UNIVERZITET U BEOGRADU

## *Sažetak*

Matematički fakultet

**Idejni projekat informacionog sistema za međunarodni kulturni centar**

Ana BAJIĆ

Međunarodni kulturni centar je kulturna institucija, slična bilo kom opštinskom ili gradskom kulturnom centru, koja ima akcenat na međusobnoj interakciji svojih članova i spajanju ostalih kulturnih centara na međunarodnom nivou. U ovom radu, u svrhu osposobljavanja rada međunarodnog kulturnog centra, pravi se informacioni sistem. Informacioni sistemi predstavljaju jednu od najčešćih i najvažnijih primena informacionih tehnologija. Nijedna poslovna organizacija ne može da funkcioniše u savremenom okruženju bez dobro organizovanih i lako dostupnih informacija. Dobar idejni projekat je jedna od najvažnijih pretpostavki uspešnog razvoja informacionog sistema. S tim na umu, pri izradi projekta prate se smernice analiziranja procesa korišćenja usluga međunarodnog kulturnog centra, prepoznavanja slučajeva upotrebe, modeliranja podataka i izrade interfejsa. Kao rezultat idejnog projekta nastaje implementacija prototipa.

***Mentor:***

prof. dr. Saša Malkov, vanredni profesor  
Univerzitet u Beogradu, Matematički Fakultet

***Komisija:***

prof. dr. Nenad Mitić, redovni profesor  
Univerzitet u Beogradu, Matematički Fakultet

prof. dr. Vladimir Filipović, redovni profesor  
Univerzitet u Beogradu, Matematički Fakultet

# Sadržaj

<b>Sažetak</b>	<b>i</b>
<b>1 Uvod</b>	<b>1</b>
1.1 Uloga međunarodnog kulturnog centra . . . . .	1
1.2 Metodologija analize i projektovanja . . . . .	1
1.3 Korišćeni alati . . . . .	2
1.3.1 Diagrams.net . . . . .	2
1.3.2 IntelliJ IDEA . . . . .	3
1.3.3 Visual Studio Code . . . . .	3
<b>2 Analiza</b>	<b>4</b>
2.1 Analiza procesa korišćenja usluga kulturnog centra . . . . .	4
2.2 Dijagrami toka podataka . . . . .	6
2.3 Neophodni podaci . . . . .	10
<b>3 Projekat informacionog sistema</b>	<b>12</b>
3.1 Slučajevi upotrebe . . . . .	12
3.1.1 Modul korisnički profil . . . . .	12
3.1.2 Modul događaji . . . . .	16
3.1.3 Modul kartica . . . . .	20
3.2 Model podataka . . . . .	24
3.2.1 Modul upravljanje profilom . . . . .	24
3.2.2 Modul upravljanje događajima . . . . .	27
3.2.3 Modul upravljanje karticom . . . . .	28
3.3 Interfejs . . . . .	31
3.3.1 Deo aplikacije zajednički za administratora i predavača . . . . .	31
3.3.2 Deo aplikacije namenjen administratoru . . . . .	34
3.3.3 Deo aplikacije namenjen predavaču . . . . .	35
3.3.4 Deo aplikacije namenjen operateru . . . . .	41
3.3.5 Deo aplikacije namenjen osnovnom informisanju korisnika . . . . .	41
3.3.6 Deo aplikacije namenjen operaterima i administratorima lokalnih ustanova kulture . . . . .	42
<b>4 Implementacija</b>	<b>43</b>
4.1 Arhitektura aplikacije . . . . .	43
4.1.1 Korišćene tehnologije prilikom implementacije prezentacionog sloja . . . . .	43
4.1.2 Korišćene tehnologije prilikom implementacije središnjeg sloja . . . . .	45
4.1.3 Korišćene tehnologije prilikom implementacije sloja podataka . . . . .	46
4.2 Serverski deo aplikacije . . . . .	48
4.2.1 Detalji implementacije . . . . .	48
4.3 Klijentski deo aplikacije . . . . .	53
4.3.1 Detalji implementacije . . . . .	53
<b>5 Zaključak</b>	<b>58</b>
<b>Literatura</b>	<b>60</b>

# Slike

2.1	Dijagram konteksta sistema . . . . .	6
2.2	Dijagram toka podataka nivoa 1 . . . . .	7
2.3	Dijagram toka podataka nivoa 2, modul "Upravljanje profilom" . . . . .	8
2.4	Dijagram toka podataka nivoa 2, modul "Upravljanje događajima" . . . . .	9
2.5	Dijagram toka podataka nivoa 2, modul "Upravljanje karticom" . . . . .	9
3.1	Dijagram slučajeva upotrebe, modul "Upravljanje korisničkim profilom" . . . . .	12
3.2	Dijagram slučajeva upotrebe, modul "Upravljanje događajima" . . . . .	16
3.3	Dijagram slučajeva upotrebe, modul "Upravljanje karticom" . . . . .	20
3.4	Dijagram entiteta i odnosa . . . . .	30
3.5	Početna stranica . . . . .	31
3.6	Stranica za registraciju korisnika . . . . .	32
3.7	Stranica za prijavljivanje korisnika . . . . .	33
3.8	Deo odeljka "Prijava i predlozi" za pregled korisničkih profila u administratorskom delu aplikacije . . . . .	34
3.9	Deo odeljka "Prijava i predlozi" za pregled prijava za izdavanje kartice u administratorskom delu aplikacije . . . . .	34
3.10	Deo odeljka "Prijava i predlozi" za pregled prijava za poziciju predavača u administratorskom delu aplikacije . . . . .	34
3.11	Odeljak "Događaji" u administratorskom delu aplikacije . . . . .	35
3.12	Odeljak "Izveštaji" u administratorskom delu aplikacije . . . . .	35
3.13	Odeljak "Osnovne informacije" i opcija ažuriranja profila . . . . .	36
3.14	Odeljak "Osnovne informacije" i opcija brisanja profila . . . . .	36
3.15	Odeljak "Osnovne informacije" u predavačkom delu aplikacije . . . . .	37
3.16	Odeljak "Događaji" u predavačkom delu aplikacije . . . . .	37
3.17	Odeljak "Događaji" i opcija dodavanja i izmene događaja u predavačkom delu aplikacije . . . . .	38
3.18	Stranica za unos podataka sa platne kartice radi plaćanja . . . . .	38
3.19	Obaveštenje o uspešnoj transakciji . . . . .	39
3.20	Odeljak korisničkog profila za prikaz kartice - slučaj kad predavač nema karticu . . . . .	40
3.21	Odeljak korisničkog profila za prikaz kartice - slučaj kad predavač ima karticu . . . . .	40
3.22	Odeljak korisničkog profila za prikaz mape . . . . .	41
4.1	Arhitektura aplikacije . . . . .	43
4.2	Struktura komponente za prijavljivanje na sistem generisane pomoću <i>Angular CLI</i> -a . . . . .	44
4.3	H2 interfejs - stranica za prijavljivanje . . . . .	47
4.4	H2 interfejs - pregled tabela i izvršavanje upita . . . . .	47
4.5	Interfejs unutar okruženja IntelliJ IDEA za rad sa bazom podataka - pregled tabela i izvršavanje upita . . . . .	48
4.6	Interfejs unutar okruženja IntelliJ IDEA za rad sa bazom podataka - dodavanje i konektovanje na bazu podataka . . . . .	49
4.7	Organizacija serverske aplikacije . . . . .	50

# Programski kodovi

4.1	Povezivanje glavnih fajlova jedne komponente . . . . .	44
4.2	Rutiranje . . . . .	44
4.3	<i>Maven</i> zavisnost za podešavanje REST krajnjih tačaka aplikacije . . . . .	45
4.4	Ulazna tačka <i>Spring Boot</i> aplikacije . . . . .	45
4.5	Konekcija na <i>in-memory</i> bazu podataka . . . . .	46
4.6	Konekcija na lokalnu bazu podataka . . . . .	46
4.7	Omogućavanje korišćenja interfejsa za H2 bazu podataka . . . . .	46
4.8	Omogućavanje povezivanja sa PostgreSQL bazom . . . . .	47
4.9	<i>Maven</i> zavisnost za rad sa PostgreSQL bazom podataka . . . . .	47
4.10	Primer kontrolera za rad sa događajima . . . . .	49
4.11	Klasa za univerzalni odgovor na HTTP zahteve . . . . .	50
4.12	SQL upit . . . . .	51
4.13	Repozitorijum za kartice . . . . .	51
4.14	Repozitorijum za predavače . . . . .	51
4.15	Entitet koji predstavlja događaj . . . . .	52
4.16	Skrivajući meni . . . . .	53
4.17	Osnovna struktura kontrolera komponente za registraciju . . . . .	53
4.18	HTML struktura komponente za registraciju . . . . .	55
4.19	Korišćenje direktive <i>*ngFor</i> radi prikazivanja svih ocena koje je predavač dobio . . . . .	56
4.20	Inicijalizacija dekoratora polja . . . . .	56
4.21	Funkcija za rad sa dijalogom . . . . .	56

# 1. Uvod

Ovaj rad se bavi izradom idejnog projekta informacionog sistema za povezivanje kulturnih centara Balkana, kao i implementacijom njegovih ključnih delova. Glavni cilj sistema je omogućavanje korisnicima da pronađu događaje u svojoj okolini, da se upoznaju sa obližnjim znamenitostima i povežu sa ljudima sličnih interesovanja, ma gde se našli. Izrađen je prototip za administrativni deo informacionog sistema i za deo koji predstavlja korisnički profil organizatora događaja.

## 1.1 Uloga međunarodnog kulturnog centra

Međunarodni kulturni centar o kojem se govori u radu je fiktivan i stvoren samo za svrhe master rada, sva sličnost u imenima, nazivima ili korišćenoj terminologiji je slučajna. Osmišljen je sa ciljem spajanja individualnih kulturnih centara (na nivou opštine, grada ili regiona) više država. Pored toga, saradnja može biti ostvarena i sa ustanovama kulture kao što su muzeji, pozorišta ili galerije. Posebna pažnja bi bila posvećena ostvarivanju saradnje sa manje poznatim ustanovama kulture i lokalitetima kako bi bila povećana posećenost. Nastao je sa idejom promovisanja bogate svetske kulture. Organizovanjem i bavljenjem različitim kulturnim dešavanjima, Centar teži da poboljša međusobno poznavanje i zbližavanje naroda, čime bi se uočile njihove sličnosti i prihvatile razlike. Međunarodni kulturni centar je osmišljen kao institucija u kojoj su dostupne sve informacije u vezi sa svetskim kulturnim dešavanjima, kojima imaju pristup ljudi iz bilo kog kraja sveta. Međunarodni kulturni centar teži ostvarivanju saradnje sa već postojećim kulturnim centrima, kako bi zajedno doprineli boljoj promociji i većoj posećenosti dešavanja.

U slučaju pandemija, kao što je u trenutku pisanja rada bila COVID-19 pandemija, Centar bi organizovao onlajn predavanja radi podizanja svesti ljudi o problemu i svemu što mogu da učine kako bi sebe i ljude u svojoj okolini zaštitili, onlajn kreativne radionice, na primer za izradu zaštitnih maski ili radi upoznavanja ljudi sa hobijima kojima se mogu baviti tokom karantina. Takođe, korisnicima bi na jednom mestu bile dostupne ažurne informacije o radu ustanova kulture, organizovanim manjim ili virtuelnim turama i slično.

## 1.2 Metodologija analize i projektovanja

Kao prvi korak u projektovanju sistema, prikazani su odgovarajući procesi sistema. Za predstavljanje procesa, korišćeni su dijagrami toka podataka (eng. *Data Flow Diagram*). Ovi dijagrami grafički prikazuju tokove kretanja podataka u sistemu. Opisuju procese sistema, njihove međuzavisnosti i na koji način oni razmenjuju podatke. Postoje u više nivoa - dijagram nivoa 0 je kontekstni dijagram čiji je jedini proces sam informacioni sistem, a što se nivoi povećavaju, to se procesi više rastavljaju na pot-procese i prikazuje se više detalja. Glavni elementi ovih dijagrama su procesi, tok podataka, skladišta podataka i krajnji entiteti.

Procesi su delovi sistema koji prenose i obrađuju podatke. Tok podataka predstavlja prenos informacija kroz sistem - tačno opisuje gde nastaje podatak koji se prenosi između povezanih elemenata i gde na kraju završava. Skladišta podataka služe za čuvanje, ažuriranje i čitanje podataka koji se kreću kroz sistem. Skladište može biti baza podataka ali i na primer grupa fizičkih dokumenata. Krajnji entiteti predstavljaju sve učesnike koji nisu unutar samog sistema. To mogu biti korisnici

sistema, spoljašnji sistemi sa kojima sistem saraduje ili delovi istog sistema koji se ne modeluju na dijagramu.

Za opisivanje poslova korišćeni su dijagrami slučajeva upotrebe. Fokusiraju se na same aktere i aktivnosti koje oni obavljaju. Svaki dijagram slučajeva upotrebe prati i opis slučaja upotrebe. Opis sadrži naziv, aktere, kratak opis, preduslove - sve ono što važi pre izvršavanja procesa, postuslove - sve ono što važi nakon izvršavanja procesa, osnovni tok i alternativne tokove - opisi posebnih slučajeva koji odstupaju od osnovnog toka.

Informacioni sistemi rukuju velikim količinama podataka. Podaci i njihovi međusobni odnosi su predstavljeni dijagramom entiteta i odnosa. Na dijagramu su predstavljeni entiteti i njihovi atributi i tipovi atributa, primarni i strani ključevi. Predstavljene su i veze između atributa, koje mogu biti jedan prema jedan (jednom entitetu tipa A odgovara najviše jedan entitet tipa B), jedan prema više (jednom entitetu tipa A odgovara više entiteta tipa B) i više prema više (više entiteta tipa B odgovara većem broju entiteta tipa A). Zahvaljujući ovim vezama mogu nastati asocijativni elementi, koji su teže uočljivi na početku projektovanja sistema.

## 1.3 Korišćeni alati

Sledeći alati su korišćeni za izradu dijagrama, modela baze podataka kao i implementaciju aplikacije:

1. Diagrams.net [3]
2. IntelliJ IDEA [6]
3. Visual Studio Code [17]

### 1.3.1 Diagrams.net

Diagrams.net je alat otvorenog koda [4] za izradu UML dijagrama. Predstavlja dobru besplatnu varijantu softvera za izradu dijagrama kao što su Visual Paradigm [16] ili Microsoft Visio [8]. Dostupan je u okviru veb pregledača a moguće je preuzeti izvršni fajl i instalirati ga na operativnim sistemima Windows, macOS i Linux [10].

Za izradu dijagrama korišćenih prilikom projektovanja informacionog sistema za međunarodni kulturni centar je korišćena onlajn verzija ovog alata i u nastavku će biti opisane njene mogućnosti. Na samom početku rada alat nudi opciju korišćenja već gotovih obrazaca (eng. *template*) za različite tipove dijagrama. Nezavisno od izabranog obrasca, na dijagram je moguće dodavati željene elemente. Elementi koji su najčešće korišćeni su dostupni u traci sa alatkama (eng. *toolbar*) a pronalaženje ostalih elementa se vrši unošenjem ključne reči u polje za pretragu. Tako na primer ako je potrebno napraviti dijagram slučajeva upotrebe, dovoljno je uneti reč „use“ u polje za pretragu i biće prikazani svi potrebni elementi za izradu ovog dijagrama.

Alat nudi i stilizovanje elemenata - moguće je obojiti ih i podešavati njihov gradijent, teksturu i prozirnost. Takođe, moguće je stilizovati i tekst vezan za elemente - moguće je promeniti font, poziciju, boju i pozadinsku boju teksta.

Izrađeni dijagram se može sačuvati u više formata - može se sačuvati kao slika u PNG, JPEG ili SVG formatu (gde svaki tip nudi različite opcije prilikom čuvanja), zatim kao PDF, HTML ili XML dokument kao i u *.drawio* formatu, koji se može učitati u alat radi nastavka rada na dijagramu. Pored toga, alat nudi mogućnost da se izrađeni dijagram umetne u već postojeće HTML stranice i Google ili Microsoft dokumente.



### 1.3.2 IntelliJ IDEA

IntelliJ IDEA je razvojno okruženje napravljeno od strane kompanije JetBrains. Dostupno je na operativnim sistemima Windows, macOS i Linux [11]. Postoje dve verzije - normalna i napredna. Normalna je besplatna dok je za naprednu potrebno platiti licencu. Glavna prednost napredne verzije je veći broj pogodnosti koje pruža. Za razvoj informacionog sistema za međunarodni kulturni centar korišćena je napredna verzija okruženja, uz besplatnu studentsku licencu.

Podržava rad sa više različitih programskih jezika, tehnologija i razvojnih okvira kao što su Java, Spring, Hibernate i JPA (eng. *Java Persistence API*) [13]. Nudi pomoć pri programiranju u vidu autokompletiranja koda, navigacije po bazi koda, refaktorisanja koda i debugovanja. Ima ugrađenu podršku za sisteme za kontrolu projekata i kontrolu verzija (Git, SVN). Moguće je pristupiti bazi podataka i pregledati i menjati tabele direktno iz okruženja.

### 1.3.3 Visual Studio Code

Visual Studio Code (VSC) je razvojno okruženje napravljeno od strane kompanije Microsoft. Dostupno je na operativnim sistemima Windows, macOS i Linux [12]. Alat je otvorenog koda [19].

Podržava rad sa više različitih programskih jezika, tehnologija i razvojnih okvira, pri čemu najveću podršku imaju JavaScript, NodeJS, HTML i CSS. Nudi pomoć pri programiranju u vidu autokompletiranja koda, navigacije po bazi koda, refaktorisanja koda i debugovanja. Ima ugrađenu podršku za sisteme za kontrolu projekata i kontrolu verzija.

Moguće je proširiti podršku VSC-a uz pomoć ekstenzija. Na ovaj način se razvojno okruženje može prilagoditi ličnim potrebama i sklonostima i moguće je dodati ili proširiti podršku za različite jezike [18].

## 2. Analiza

### 2.1 Analiza procesa korišćenja usluga kulturnog centra

U Beogradu postoji veliki broj kulturnih centara - na nivou opštine (kao što je Kulturni centar Čukarica ili Centar za obrazovanje i kulturu “Božidarac 1947” na Vračaru), a i na nivou samog grada (Kulturni centar Beograda). Slična situacija se zatiče i u drugim većim gradovima u Srbiji, na primer, na nivou regiona, Kulturni centar Vojvodine “Miloš Crnjanski” u Novom Sadu.

Svaki od tih kulturnih centara ima svoj veb-sajt sa informacijama i događaje i radionice koje organizuje. Mali broj kulturnih centara omogućava prijavljivanje za iste preko veb-sajta, već je potrebno da korisnik ili dođe do same ustanove ili da se raspita i prijavi telefonskim pozivom.

Glavni problem koji je uočen je slaba funkcionalnost ovih veb-sajtova kao i službi za telefonsku komunikaciju sa klijentima. Delovi veb-sajta ne rade, preglednost nije najbolja i informacije nisu lako dostupne. Prosečnom korisniku koji je zainteresovan da posećuje događaje i radionice nije lako i privlačno da prati desetine veb-sajtova kako bi bio u toku sa aktuelnim dešavanjima. Takođe, brojevi telefona na kojima korisnik može da se detaljnije informiše o dešavanjima često služe i za druge namene, pa se ponekad teško dobijaju ili se službenici teško fokusiraju i komunikacija sa njima nije dovoljno efikasana. Na kraju, korisnik možda i ne zna za kulturne centre na drugim opštinama.

Dodatni problem, koji se može svrstati u kategoriju nedostupnih informacija, se uviđa kod događaja i radionica koje imaju ograničen broj mesta. Na veb-sajtovima kulturnih centara često nije označeno koliko još slobodnih mesta ima, te korisnik koji želi da učestvuje može izgubiti svoje vreme pokušavajući da se prijavi (uživo ili telefonskim pozivom) jer ne zna da mesta za događaj nema.

Kako je ideja iza ovog projekta stvaranje internacionalnog kulturnog centra, potrebno je razmotriti i informisanje o događajima i u inostranstvu. Postoji nekoliko start-up kompanija koje se bave sličnim problemom - da svojim korisnicima omoguće lak pristup informacijama o zemlji u kojoj se nalaze, ipak prepreka na koju se često nailazi je predlaganje znamenitosti korisnicima čiji im je sadržaj već dobro poznat. Dok ovo jeste dobar način da se upozna sa kulturom zemlje, bilo bi dobro da se omogući i malo “intimniji” pogled u život ljudi te zemlje i njihovu kulturu. Kao i Srbija, i druge zemlje često organizuju kulturne događaje i manifestacije za koje turisti ne mogu na lak način da saznaju. Takođe, informacije o njima su često dostupne samo na jeziku države u kojoj se dešavaju, te to može biti odbojno korisniku koji ne poznaje jezik.

Ideja je organizovati i povezati sve te ustanove kulture u jedan internacionalni kulturni centar (eng. *ICC - International Cultural Center*). Dakle, sve te ustanove bi nastavile da postoje, s tim da bi bile članice ICC-a. Ovime bi postojao jedan sistem za informisanje o svim dešavanjima koje te ustanove organizuju. Radi lakšeg pregleda, kako ne bi bili pretrpani informacijama, korisnicima bi bilo omogućeno da pretražuju samo lokalne ustanove kulture. Sve potrebne informacije o ustanovi i događajima koje organizuje bi bile vidno i lako dostupne korisnicima jer informacioni sistem ICC-a zahteva njihovo unošenje pri pravljenju profila ustanove i događaja. Korisnicima bi, pored mogućnosti prijavljivanja uživo u ustanovi ili telefonskim pozivom, bilo omogućeno da se za svaki događaj mogu prijaviti na jednostavan način i putem veb-sajta. Imali bi uvid u broj prijavljenih korisnika kao i u detaljan opis događaja i njegovih organizatora.

Prilikom razmatranja saradnje sa drugim kulturnim centrima, moraju se razmotriti dve opcije - postoje kulturni centri koji već poseduju svoj informacioni sistem i oni koji ga nemaju. Kulturni centri koji nemaju svoj informacioni sistem najčešće imaju samo čisto prezentacioni veb-sajt i osposobljenu telefonsku liniju za informisanje. ICC bi i jednima i drugima nudio saradnju.

Kulturni centri koji imaju svoj informacioni sistem bi dobili mogućnost da dele podatke sa ICC-om. ICC bi tim kulturnim centrima predstavio svoje komponente sa kojima bi povezali svoj sistem radi uspostavljanja saradnje i razmene podataka. Prilikom pravljenja profila preko informacionog sistema lokalnog kulturnog centra, korisniku se nudi opcija da svoj profil podeli sa ICC-om. Korisniku bi bila predstavljena politika privatnosti deljenja podataka i njegovi podaci bi bili podeljeni samo u slučaju da prihvati uslove. Time bi se izbeglo pravljenje i održavanje profila kod više kulturnih centara. Korisnik koji se odluči za ovu opciju bi postao punopravni korisnik ICC-a, ni po čemu različit od korisnika koji je profil napravio preko informacionog sistema ICC-a. Saradnja bi išla i u drugom pravcu - u slučaju da je korisnik već član ICC-a i preko njega pronađe lokalni kulturni centar čija ga ponuda interesuje, ICC bi mu ponudio opciju da svoj profil podeli sa tim kulturnim centrom.

Postojala bi opcija da se svaki događaj koji lokalni kulturni centar napravi zabeleži u informacioni sistem ICC-a. Na ovaj način bi se povećavala baza događaja ICC-a a uz to bi se povećavala i vidljivost napravljenih događaja. U slučaju da lokalni kulturni centar odluči da deli svoje događaje sa ICC-om, mogao bi da deli i prijavljene korisnike - kada korisnik odluči da se preko informacionog sistema lokalnog kulturnog centra prijavi za događaj, podaci o toj prijavi bi bili prosleđeni informacionom sistemu ICC-a. Kako već postojeći lokalni kulturni centri nisu od svog nastanka članovi ICC-a i imaju aktuelne događaje napravljenje pre učlanjivanja, bilo bi potrebno da izlože komponentu svog informacionog sistema koja bi služila za dohvatanje postojećih događaja. Ovo bi bio dodatan način za proširivanje baze događaja ICC-a i za održavanje njene aktuelnosti.

Kako se događaji dele između lokalnih kulturnih centara i ICC-a, korisnicima bi bilo omogućeno da karte za događaje kupe direktno preko informacionog sistema ICC-a. ICC bi imao sve potrebne informacije o kupovini karata za deljene događaje (koje bi dobijao prilikom pravljenja događaja od strane lokalnog kulturnog centra) te bi korisnik mogao da kupi kartu direktno a u zavisnosti od dogovora o realizaciji kupovine između ICC-a i lokalnog kulturnog centra bi dobio potvrdu od jedne ili obe strane o uspešnoj kupovini. Pri tome bi ICC prosledio podatke o kupovini informacionom sistemu lokalnog kulturnog centra. U slučaju da lokalni kulturni centar ne želi ovakav tip saradnje ili je saradnja teža za realizovanje (kupovina karata u pozorištu zahteva odabir sedišta, što bi značilo da bi ICC morao da ima plan sedenja za svako pozorište ili bioskop sa kojim saraduje), ICC bi korisnike prosledio direktno na informacioni sistem lokalnog kulturnog centra radi obavljanja kupovine.

ICC bi lokalnim kulturnim centrima koji nemaju svoj informacioni sistem ponudio saradnju u vidu administratorskog panela. U informacionom sistemu ICC-a bi bili napravljeni profili za administratore podsistema određenog za konkretan lokalni kulturni centar. Ti administratori bi imali slične mogućnosti kao administratori celog sistema - dodavanje događaja i rukovanje njima, dodavanje korisnika, pregled statistike - ali bi bile ograničene samo na podatke vezane za njihov podsistem.

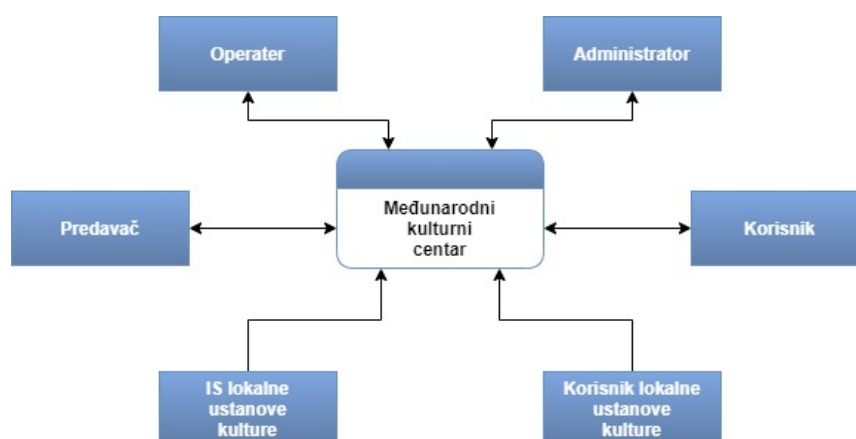
Lokalnim kulturnim centrima koji ostvare saradnju sa ICC-om bi bilo omogućeno slanje obaveštenja korisnicima o dešavanjima u vidu imejlova ili notifikacija preko mobilnih uređaja. Mogli bi da personalizuju imejlve koji se šalju ili umetanjem svojih obrazaca ili samo umetanjem teksta u obrazac koji koristi ICC. ICC bi naravno pokrивao i druge zemlje i njihove ustanove. Pružao bi korisnicima mogućnost, kroz

vweb-sajt i mobilnu aplikaciju, da pretraže lokalne ustanove kulture i prijave se za događaje, gde god se našli. Takođe, sistem bi u početku bio dostupan na srpskom i engleskom jeziku, ali bi se vremenom uvodile podrške i za druge jezike. Ovime bi bilo osigurano da korisnik u svakom trenutku, nezavisno od svog poznavanja jezika, može da dođe do informacija o događaju i da se za isti prijavi.

## 2.2 Dijagrami toka podataka

Dijagram konteksta predstavlja procese na najvišem nivou - jedini proces je sam informacioni sistem. Dodatno, prikazani su svi elementi okruženja u kome IS funkcioniše i sa kojima komunicira - akteri, baze podataka i drugi poslovni sistemi. Svaki sledeći nivo dijagrama toka podataka rastavlja procese predstavljene u prethodnom nivou na potprocesse.

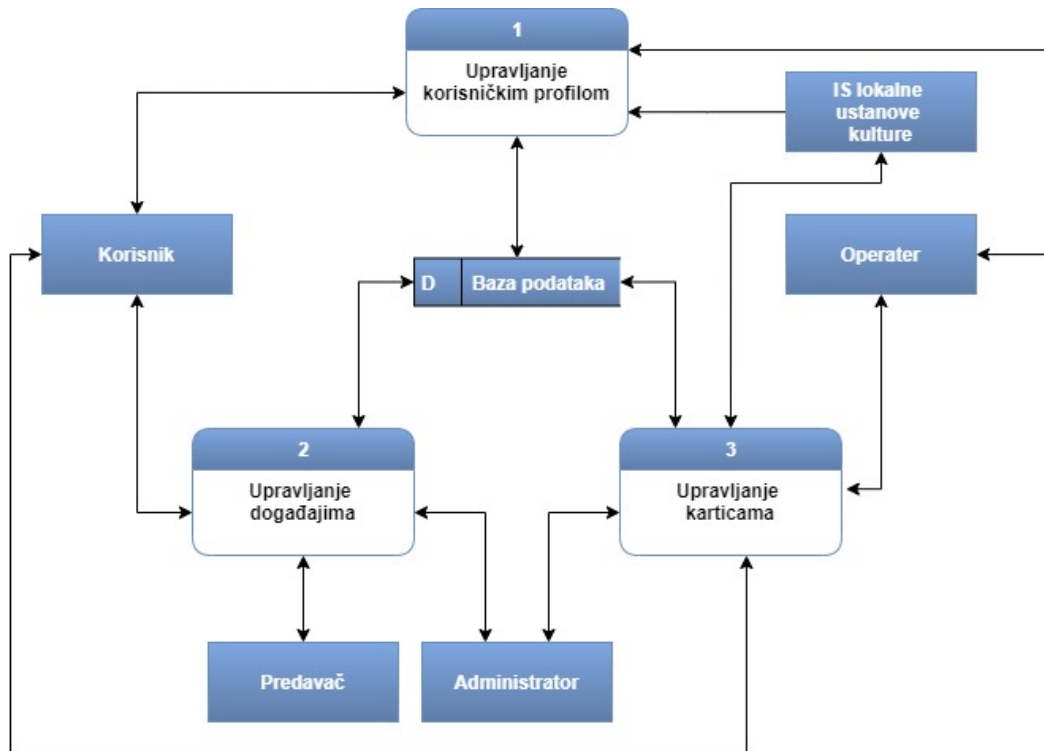
Na dijagramu konteksta, koji je predstavljen na slici 2.1, glavni proces je sam *Međunarodni kulturni centar*. Sistem interaguje sa akterima *Administrator*, *Korisnik*, *Predavač*, *Operater*, *IS lokalne ustanove kulture* i *Korisnik lokalne ustanove kulture*.



SLIKA 2.1: Dijagram konteksta sistema

Dijagram toka podataka prvog nivoa (slika 2.2) glavni proces deli na tri potprocesa:

1. Upravljanje profilom
2. Upravljanje događajima
3. Upravljanje karticom

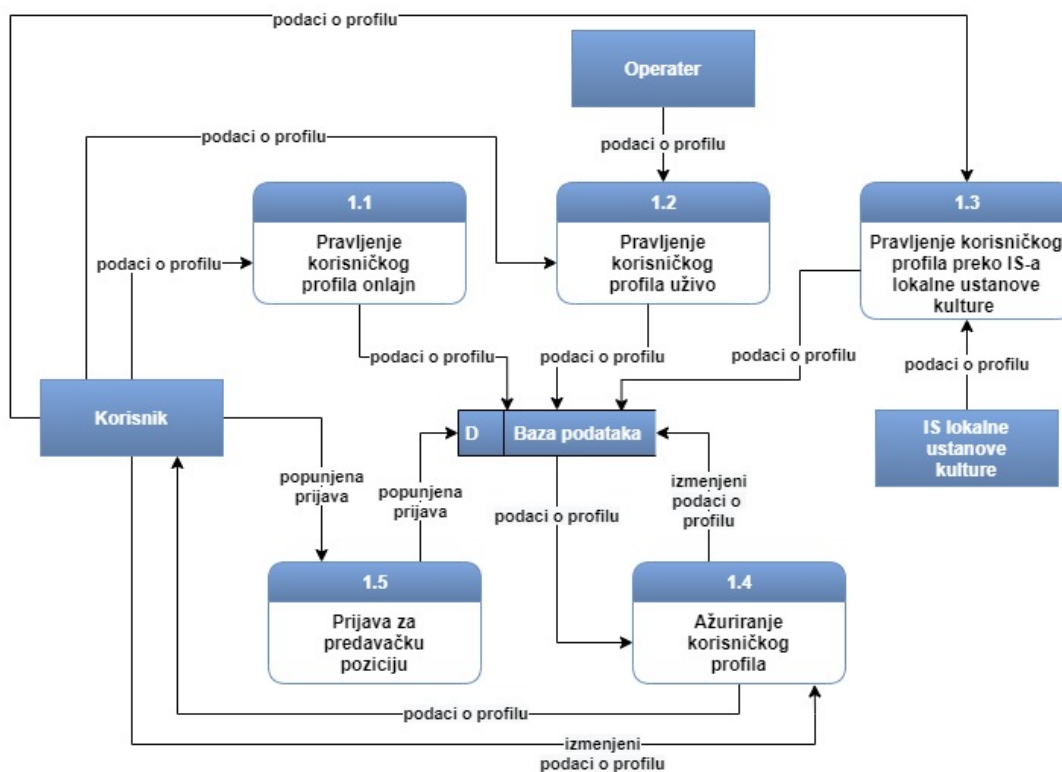


SLIKA 2.2: Dijagram toka podataka nivoa 1

Dijagram toka podataka drugog nivoa detaljno razlaže i opisuje procese nivoa 1. Prikazan je i tok podataka između aktera, procesa i skladišta podataka.

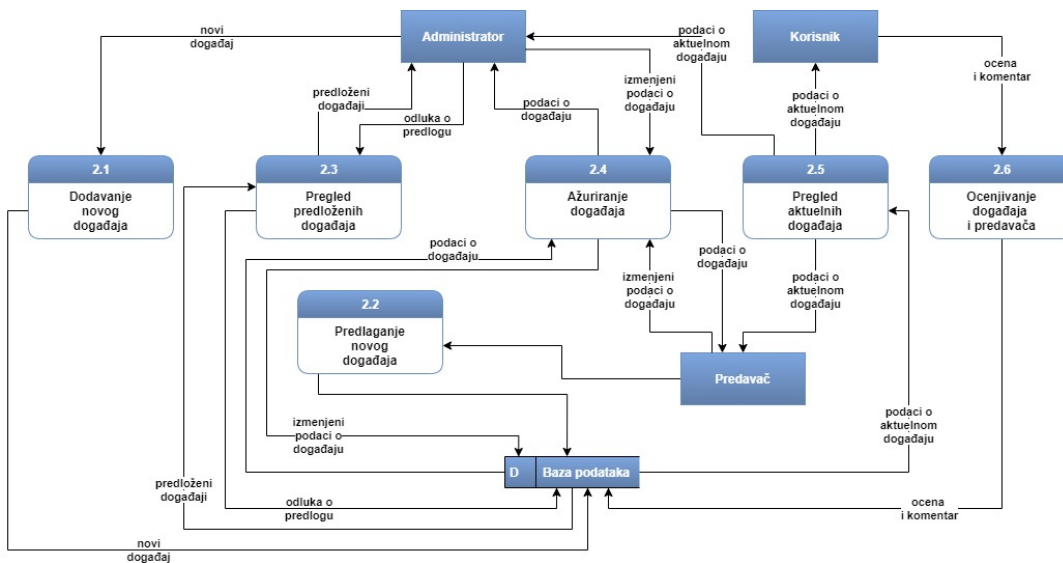
Proces “Upravljanje profilom” je razložen na pet potprocesa: “Pravljenje korisničkog profila onlajn”, “Pravljenje korisničkog profila uživo”, “Pravljenje korisničkog profila onlajn preko IS-a partnerske ustanove kulture”, “Ažuriranje korisničkog profila” i “Prijava za predavačku poziciju” (slika 2.3).

Proces “Upravljanje događajima” je razložen na šest potprocesa: “Dodavanje novog događaja”, “Predlaganje novog događaja”, “Pregled predloženih događaja”, “Ažuriranje događaja”, “Pregled aktuelnih događaja” i “Ocenjivanje događaja i predavača” (slika 2.4).

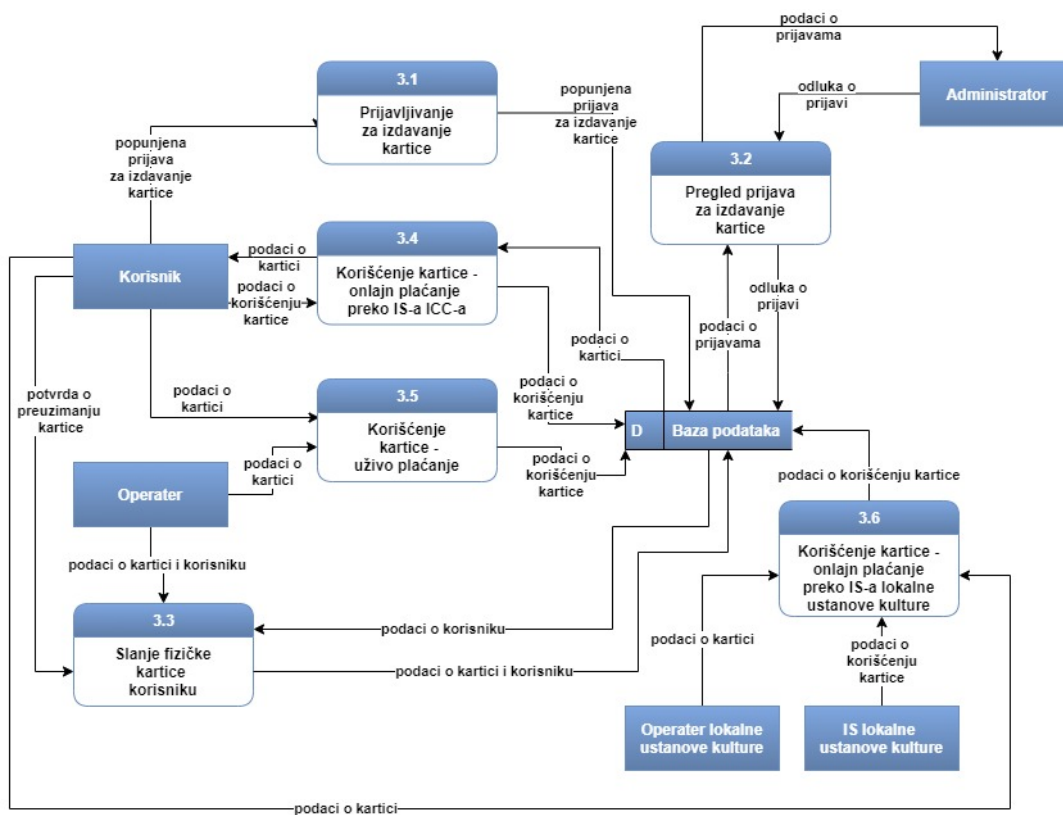


SLIKA 2.3: Dijagram toka podataka nivoa 2, modul "Upravljanje profilom"

Proces "Upravljanje karticom" je razložen na pet potprocesa: "Prijavlivanje za izdavanje kartice", "Pregled prijava za izdavanje kartice", "Korišćenje kartice prilikom kupovine karata - onlajn plaćanje", "Korišćenje kartice prilikom kupovine karata - uživo plaćanje u poslovnicu ICC-a", "Korišćenje kartice prilikom kupovine karata - uživo plaćanje na šalteru partnerske ustanove kulture" (slika 2.5).



SLIKA 2.4: Dijagram toka podataka nivoa 2, modul “Upravljanje događajima”



SLIKA 2.5: Dijagram toka podataka nivoa 2, modul “Upravljanje karticom”

## 2.3 Neophodni podaci

Analizom procesa korišćenja usluga kulturnog centra prepoznati su podaci i njihovi atributi. U fazi projektovanja će na osnovu ovih podataka i razrade slučajeve upotrebe biti izrađen potpuni model podataka.

1. Korisnik - predstavlja jednog korisnika sistema. Opisuje se narednim atributima:
  - Ime
  - Prezime
  - Imejl adresa
  - Tip korisnika (Administrator, Korisnik, Predavač, Operater)
  - Šifra
  - Korisničko ime
  - Zemlja
  - Grad
  - Adresa
  - Broj telefona
  - Vreme pravljenja korisničkog profila
2. Predavač - predstavlja specijalizaciju Korisnika. Dodatno se karakteriše narednim atributom:
  - Specijalizacija (Istorija, kinematografija, muzika, gluma, strani jezici, ples, fotografija, književnost i slično)
  - Aktivan
3. Administrator - predstavlja specijalizaciju entiteta Korisnik. Dodatno se karakteriše narednim atributom:
  - Ustanova kulture čiji je administrator
4. Operater - predstavlja specijalizaciju entiteta Korisnik. Dodatno se karakteriše narednim atributom:
  - Ustanova kulture čiji je operater
5. Lokalna ustanova kulture - predstavlja lokalnu ustanovu kulture koja može a i ne mora da ima saradnju sa ICC-om. Dodatno se karakteriše narednim atributom:
  - Naziv
  - Zemlja
  - Grad
  - Adresa
  - Tip (Muzej, pozorište, kulturni centar i slično)
  - Ostvarena saradnja
6. Događaj - predstavlja jedan događaj. Opisan je narednim atributima:
  - Naziv



- Grad
  - Država
  - Mesto održavanja (ustanova kulture)
  - Datum i vreme
  - Opis
  - Cena karte
  - Potrebna karta
7. Predlog događaja - predstavlja predlog novog događaja nekog predavača. Okarakterisan je narednim atributima:
- Naziv
  - Grad
  - Država
  - Mesto održavanja (ustanova kulture)
  - Datum i vreme
  - Opis
8. Prijava za izdavanje kartice - predstavlja prijavu korisnika za izdavanje kartice. Opisana je narednim atributima:
- Podaci o korisniku koji se prijavio
  - Adresa za dostavu
  - Datum prijave
9. Kartica - predstavlja karticu korisnika. Opisana je narednim atributima:
- Broj kartice
  - Tip kartice (privremeno blokirana, trajno blokirana, regularna, napredna)
  - Status (aktivna, neaktivna)
  - Broj korišćenja
  - Broj poena
  - Datum izdavanja
10. Karta za događaj - predstavlja kupljenu kartu za aktuelni događaj. Opisana je narednim atributima:
- Redni broj karte
  - Podaci o korisniku koji je kupio kartu
  - Datum kupovine
  - ID transakcije
  - Status transakcije (započeto, uspešno, neuspešno)
11. Ocena - predstavlja ocenu koju je korisnik dao događaju i predavaču. Opisana je narednim atributima:
- Komentar
  - Ocena (na skali od 1 do 5)
  - Vreme ocenjivanja

## 3. Projekat informacionog sistema

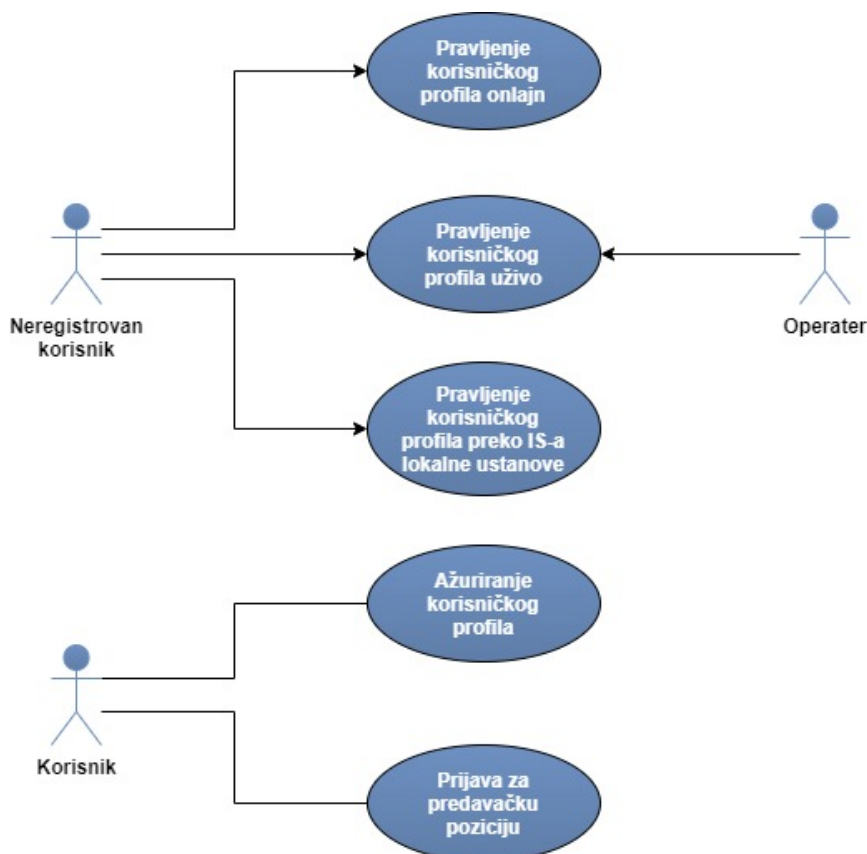
### 3.1 Slučajevi upotrebe

Na osnovu modela procesa, koji je predstavljen u obliku dijagrama toka podataka (odeljak 2.2), prepoznati su naredni slučajevi upotrebe.

#### 3.1.1 Modul korisnički profil

Dijagram slučajeva upotrebe za modul za korisnički profil je prikazan na slici 3.1. U okviru ovog modula mogu se prepoznati sledeći slučajevi upotrebe:

1. Pravljenje korisničkog profila onlajn
2. Pravljenje korisničkog profila uživo
3. Pravljenje korisničkog profila preko IS-a lokalne ustanove kulture
4. Ažuriranje korisničkog profila
5. Prijava za predavačku poziciju



SLIKA 3.1: Dijagram slučajeva upotrebe, modul "Upravljanje korisničkim profilom"

### 3.1.1.1 Pravljenje korisničkog profila onlajn

1. **Kratak opis:** Korisnik vrši registraciju tako što popunjava formular svojim ličnim podacima. Vršiti se validacija podataka i korisnik dobija informaciju o uspešnosti akcije.
2. **Učesnici:** Neregistrovan korisnik
3. **Preduslovi:** /
4. **Postuslovi:** Korisniku je napravljen nalog na sistemu. Baza korisnika je ažurirana.
5. **Osnovni tok:**
  - 1) Korisnik odlazi na deo sistema za registraciju novih korisnika.
  - 2) Korisniku sistem prikazuje opciju za registraciju korišćenjem profila neke druge ustanove kulture čiji je član.
  - 3) Korisnik unosi potrebne podatke i bira opciju "Registruj se".
  - 4) Sistem vrši validaciju podataka.
  - 5) Sistem šalje imejl korisniku sa linkom za potvrdu registracije.
  - 6) Korisnik proverava svoje sanduče i potvrđuje registraciju prateći link za potvrdu.
  - 7) Sistem obaveštava korisnika da je nalog uspešno kreiran.
6. **Alternativni tokovi:**

**Korisnik bira da se registruje postojećim profilom:** 3a - Korisnik umesto opcije "Registruj se" pronalazi na spisku ustanova kulture onu čiji je već korisnik i bira opciju "Poveži profil". Korisniku se prikazuje politika privatnosti deljenja podataka koju mora da prihvati kako bi nastavio dalje.

**Neispravan unos:** 4a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

**Izostanak imejla za potvrdu:** 6a - Ukoliko korisniku nije stigao imejl za potvrdu registracije, može tražiti da mu se pošalje novi odabirom opcije "Nije ti stigao verifikacioni imejl?".

### 3.1.1.2 Pravljenje korisničkog profila uživo

1. **Kratak opis:** Korisnik vrši registraciju tako što operateru daje svoje lične podatke. Operater ih unosi u sistem, vrši se njihova validacija i korisnik dobija informaciju o uspešnosti akcije.
2. **Učesnici:** Neregistrovan korisnik, operater
3. **Preduslovi:** /
4. **Postuslovi:** Korisniku je napravljen nalog na sistemu. Baza korisnika je ažurirana.
5. **Osnovni tok:**
  - 1) Korisnik odlazi do poslovnice ICC-a.
  - 2) Korisnik daje potrebne podatke operateru.

- 3) Operater unosi podatke o korisniku u sistem.
- 4) Sistem vrši validaciju podataka.
- 5) Sistem šalje imejl korisniku sa linkom za potvrdu registracije.
- 6) Operater izveštava korisnika da mu je poslat imejl.
- 7) Korisnik proverava svoje sanduče i potvrđuje registraciju prateći link za potvrdu.
- 8) Sistem obaveštava operatera da je nalog uspešno kreiran.

6. **Alternativni tokovi:**

**Neispravan unos:** 4a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

**Izostanak imejla za potvrdu:** 7a - Ukoliko korisniku nije stigao imejl za potvrdu registracije, operater mu može poslati novi odabirom opcije "Nije ti stigao verifikacioni imejl?".

### 3.1.1.3 Pravljenje korisničkog profila onlajn preko IS-a partnerske ustanove kulture

1. **Kratak opis:** Korisniku se prilikom registracije na IS partnerske ustanove kulture nudi opcija da podeli svoj profil i postane član ICC-a.
2. **Učesnici:** Korisnik lokalne ustanove
3. **Preduslovi:** /
4. **Postuslovi:** Korisniku je napravljen nalog na sistemu ICC-a. Baza korisnika je ažurirana.
5. **Osnovni tok:**
  - 1) Korisnik odlazi na deo IS-a lokalne ustanove za registraciju novih korisnika.
  - 2) Korisniku IS lokalne ustanove prikazuje opciju da svoj profil podeli sa sistemom ICC-a.
  - 3) Korisnik unosi potrebne podatke, bira opciju za deljenje profila i bira opciju za pravljenje profila.
  - 4) IS lokalne ustanove unešene podatke prosleđuje sistemu ICC-a.
  - 5) Sistem ICC-a vrši validaciju podataka.
  - 6) Sistem ICC-a šalje imejl korisniku sa linkom za potvrdu registracije.
  - 7) Korisnik proverava svoje sanduče i potvrđuje registraciju prateći link za potvrdu.
  - 8) Sistem obaveštava korisnika da je nalog uspešno kreiran.

6. **Alternativni tokovi:**

**Neispravan unos:** 5a - Ako sistem ICC-a pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, vraća odgovarajući odgovor IS-u lokalne ustanove koji je potrebno prikazati korisniku.

**Automatska potvrda imejla za potvrdu:** 6a - U slučaju da IS lokalne ustanove već šalje svoj imejl za potvrdu registracije, sistem ICC-a šalje imejl za potvrdu registracije imejl serveru IS-a lokalne ustanove koji automatski potvrđuje registraciju.

**Izostanak imejla za potvrdu:** 7a - Ukoliko korisniku nije stigao imejl za potvrdu registracije, može tražiti da mu se pošalje novi odabirom opcije "Nije ti stigao verifikacioni imejl?".

#### 3.1.1.4 Ažuriranje korisničkog profila

1. **Kratak opis:** Korisnik vrši ažuriranje svog ličnog profila menjanjem podataka u formularu.
2. **Učesnici:** Korisnik
3. **Preduslovi:** Korisnik je prijavljen na sistem.
4. **Postuslovi:** Lični podaci korisnika su uspešno izmenjeni.
5. **Osnovni tok:**
  - 1) Korisnik odlazi na deo stranice za pregled svojih ličnih podataka.
  - 2) Korisnik bira opciju "Ažuriraj podatke".
  - 3) Korisnik po potrebi menja podatke formulara koji mu je prezentovan.
  - 4) Korisnik bira opciju "Sačuvaj izmene".
  - 5) Sistem vrši validaciju podataka.
  - 6) Sistem obaveštava korisnika da su izmene uspešno sačuvane.
6. **Alternativni tokovi:**

**Neispravan unos:** 5a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

#### 3.1.1.5 Prijava za predavačku poziciju

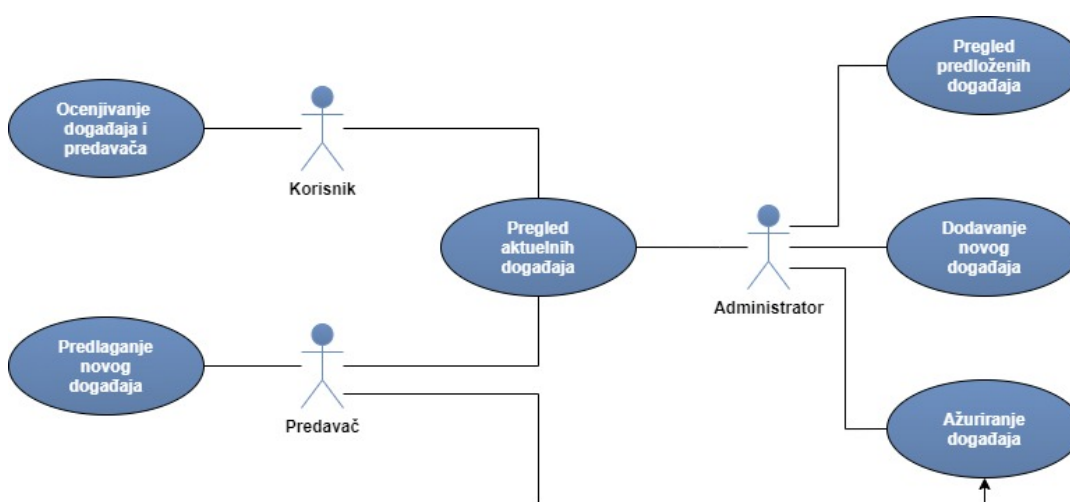
1. **Kratak opis:** Korisnik se prijavljuje za poziciju predavača, čime administrator dobija obaveštenje o prijavi.
2. **Učesnici:** Korisnik
3. **Preduslovi:** Korisnik je prijavljen na sistem.
4. **Postuslovi:** Administrator je dobio obaveštenje o postojanju prijave za predavačku poziciju.
5. **Osnovni tok:**
  - 1) Korisnik odlazi na deo stranice za pregled svojih ličnih podataka.
  - 2) Korisnik bira opciju "Prijavi se za predavačku poziciju".
  - 3) Sistem prikazuje korisniku formular na kome se nalaze polja:
    - Specijalizacija
    - Propratno pismo
  - 4) Korisnik unosi podatke u formular.
  - 5) Korisnik bira opciju "Prijavi se".
  - 6) Sistem vrši validaciju podataka.
  - 7) Sistem obaveštava korisnika da je prijava uspešno primljena.
6. **Alternativni tokovi:**

**Neispravan unos:** 6a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

### 3.1.2 Modul događaji

Dijagram slučajeva upotrebe za modul za upravljanje događajima je prikazan na slici 3.2. U okviru ovog modula mogu se prepoznati sledeći slučajevi upotrebe:

1. Dodavanje novog događaja
2. Predlaganje novog događaja
3. Pregled predloženih događaja
4. Ažuriranje događaja
5. Pregled aktuelnih događaja
6. Ocenjivanje događaja i predavača



SLIKA 3.2: Dijagram slučajeva upotrebe, modul “Upravljanje događajima”

#### 3.1.2.1 Dodavanje novog događaja

1. **Kratak opis:** Administrator dodaje novi događaj.
2. **Učesnici:** Administrator
3. **Preduslovi:** Administrator je prijavljen na sistem.
4. **Postuslovi:** Dodan je novi događaj i svi predavači koji su za njega vezani su obavešteni o tome.
5. **Osnovni tok:**
  - 1) Administrator odlazi na deo stranice za pregled događaja.
  - 2) Administrator bira opciju “Dodaj novi događaj”.
  - 3) Sistem prikazuje formular za dodavanje novog događaja na kome se nalaze polja:
    - Ime događaja
    - Grad

- Država
  - Mesto održavanja
  - Datum i vreme
  - Opis
  - Spisak predavača
- 4) Administrator unosi podatke u formular.
  - 5) Administrator bira opciju "Dodaj".
  - 6) Sistem vrši validaciju podataka.
  - 7) Sistem čuva novi događaj.
  - 8) Sistem obaveštava administratora o uspešnosti akcije.

6. **Alternativni tokovi:**

**Neispravan unos:** 6a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

### 3.1.2.2 Predlaganje novog događaja

1. **Kratak opis:** Predavač pravi predlog događaja.
2. **Učesnici:** Predavač
3. **Preduslovi:** Predavač je prijavljen na sistem.
4. **Postuslovi:** Dodat je novi predlog događaja i administrator je obavešten o tome.
5. **Osnovni tok:**
  - 1) Predavač odlazi na deo stranice za pregled događaja.
  - 2) Predavač bira opciju "Predloži novi događaj".
  - 3) Sistem prikazuje formular za predlaganje novog događaja na kome se nalaze polja:
    - Ime događaja
    - Grad
    - Država
    - Mesto održavanja
    - Datum i vreme
    - Opis
    - Spisak predavača
  - 4) Predavač unosi podatke u formular.
  - 5) Predavač bira opciju "Dodaj".
  - 6) Sistem vrši validaciju podataka.
  - 7) Sistem čuva novi predlog.
  - 8) Sistem obaveštava predavača o uspešnosti akcije.

6. **Alternativni tokovi:**

**Neispravan unos:** 6a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

### 3.1.2.3 Pregled predloženih događaja

1. **Kratak opis:** Administrator ima uvid u predloge novih događaja koje su pravili predavači. Administrator ima mogućnost prihvatanja ili odbijanja predloga.
2. **Učesnici:** Administrator
3. **Preduslovi:** Administrator je prijavljen na sistem. Postoje predlozi događaja.
4. **Postuslovi:** Prihvaćeni predlozi su dodati u spisak svih aktuelnih događaja. Predavači dobijaju obaveštenje ako je njihov predlog odbijen.
5. **Osnovni tok:**
  - 1) Administrator odlazi na deo stranice sa svim prijavama i predlozima.
  - 2) Administrator ima uvid u sve podatke o predloženom događaju i u profil predavača koji je taj predlog napravio, kao i dve opcije: *“Prihvati predlog”* i *“Odbij predlog”*.
    - i. a) Administrator bira opciju *“Prihvati predlog”*.
    - b) Sistem obaveštava administratora o uspešnosti operacije.
    - ii. a) Administrator bira opciju *“Odbij predlog”*.
    - b) Sistem prikazuje formular sa poljem za razlog odbijanja predloga.
    - c) Administrator unosi razlog i bira opciju *“Sačuvaj”*.
    - d) Sistem vrši validaciju podataka.
    - e) Sistem obaveštava administratora o uspešnosti operacije.
6. **Alternativni tokovi:**

**Neispravan unos:** (2-ii-d)a - Ukoliko administrator izostavi razlog odbijanja predloga, sistem ispisuje odgovarajuću poruku.

### 3.1.2.4 Ažuriranje događaja

1. **Kratak opis:** Administrator ažurira događaj. Svi prijavljeni korisnici kao i predavači dobijaju obaveštenje o izmenama.
2. **Učesnici:** Administrator
3. **Preduslovi:** Administrator je ulogovan na sistem. Postoje aktuelni događaji.
4. **Postuslovi:** Podaci o događaju su uspešno izmenjeni i svi prijavljeni korisnici kao i predavači su obavešteni o izmenama.
5. **Osnovni tok:**
  - 1) Administrator odlazi na deo stranice o događajima.
  - 2) Administrator pronalazi događaj koji želi da izmeni.
  - 3) Administrator bira opciju *“Izmeni”*.
  - 4) Sistem prikazuje formular za izmenu događaja.
  - 5) Administrator unosi potrebne podatke.
  - 6) Administrator bira opciju *“Sačuvaj izmene”*.
  - 7) Sistem vrši validaciju podataka.
  - 8) Sistem obaveštava administratora o uspešnosti operacije.



- 9) Sistem obaveštava sve prijavljene korisnike kao i predavače o postojanju izmena.

#### 6. Alternativni tokovi:

**Neispravan unos:** 7a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

### 3.1.2.5 Pregled aktuelnih događaja

1. **Kratak opis:** Korisniku se prikazuje lista aktuelnih događaja.
2. **Učesnici:** Korisnik
3. **Preduslovi:** Korisnik je ulogovan na sistem. Postoje aktuelni događaji.
4. **Postuslovi:** Otvorena je stranica sa listom aktuelnih događaja.
5. **Osnovni tok:**

- 1) Korisnik odlazi na deo stranice za događaje.
- 2) Sistem prikazuje dve liste događaja:
  - a) Aktuelni događaji
  - b) Završeni događaji

Za svaki događaj se prikazuje ime događaja, mesto održavanja, grad i država održavanja, vreme održavanja, spisak predavača, kao i opcije za prijavu za učešće i zainteresovanost za događaj.

#### 6. Alternativni tokovi: /

### 3.1.2.6 Ocenjivanje događaja i predavača

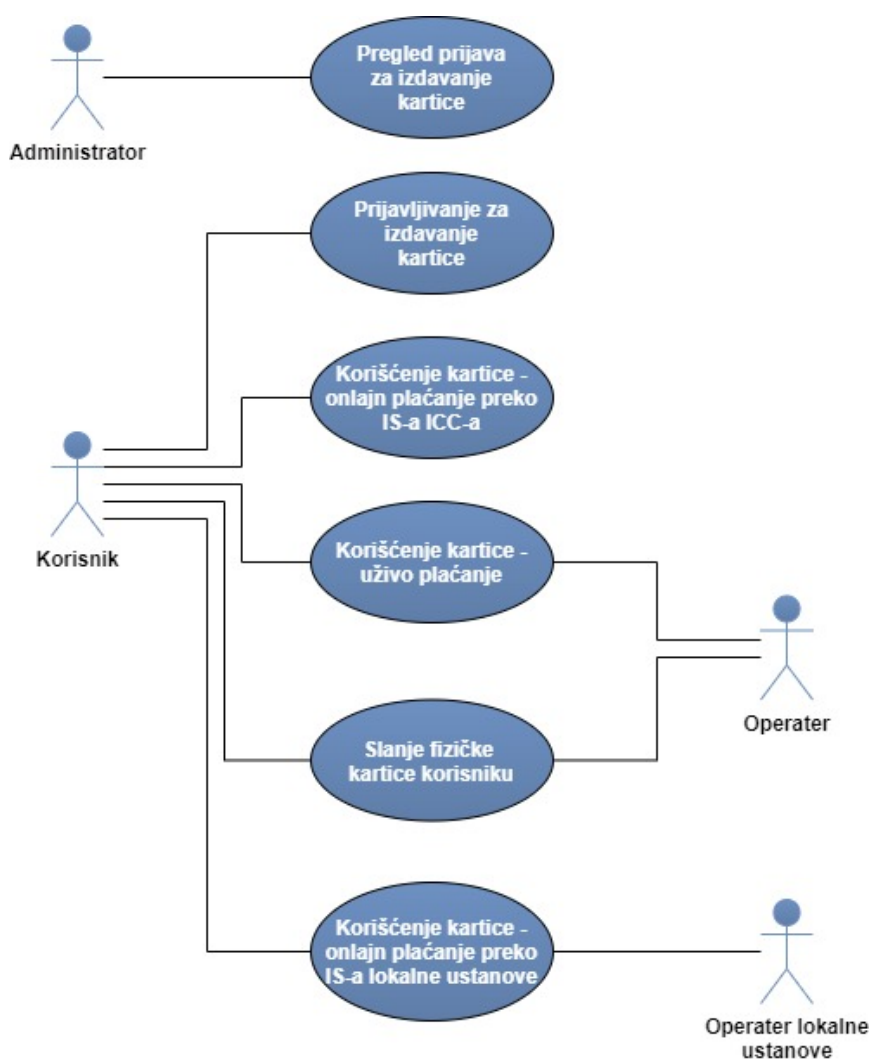
1. **Kratak opis:** Korisnik može da oceni i da komentar na događaj i predavače nakon što se događaj završio.
2. **Učesnici:** Korisnik
3. **Preduslovi:** Korisnik je ulogovan na sistem. Postoje završeni događaji kojima je korisnik prisustvovao.
4. **Postuslovi:** Ažurirane su ocene za događaj i predavače. Predavač može da pročita komentar i ocenu na svom profilu.
5. **Osnovni tok:**

- 1) Korisnik odlazi na deo stranice za događaje.
- 2) Korisnik u listi završenih događaja pronalazi događaj kojem je prisustvovao i bira opciju "Oceni događaj".
- 3) Korisnik bira ocenu koju želi da da i opciono unosi propratni komentar.
- 4) Korisnik bira opciju "Sačuvaj".
- 5) Sistem čuva ocenu koju je korisnik dao, ažurira ukupnu ocenu događaja i predavača.

### 3.1.3 Modul kartica

Dijagram slučajeve upotrebe za modul za upravljanje karticom je prikazan na slici 3.3. U okviru ovog modula mogu se prepoznati sledeći slučajevi upotrebe:

1. Prijavljanje za izdavanje kartice
2. Pregled prijave za izdavanje kartice
3. Slanje fizičke kartice korisniku
4. Korišćenje kartice - onlajn plaćanje preko IS-a ICC-a
5. Korišćenje kartice - uživo plaćanje u poslovnici ICC-a
6. Korišćenje kartice - uživo plaćanje na šalteru partnerske ustanove kulture



SLIKA 3.3: Dijagram slučajeve upotrebe, modul “Upravljanje karticom”

#### 3.1.3.1 Prijavljanje za izdavanje kartice

1. **Kratak opis:** Korisnik se prijavljuje za izdavanje kartice kulturnog centra.

2. **Učesnici:** Korisnik
3. **Preduslovi:** Korisnik je ulogovan na sistem.
4. **Postuslovi:** Prijava je sačuvana u sistemu. Administrator je dobio obaveštenje o novoj prijavi.
5. **Osnovni tok:**
  - 1) Korisnik odlazi na deo stranice o karticama.
  - 2) Korisnik bira opciju "Prijavi se".
  - 3) Sistem prikazuje formular za unos ličnih podataka korisnika.
  - 4) Korisnik popunjava formular.
  - 5) Korisnik bira da li želi da mu kartica bude poslata na unetu adresu ili želi da je lično pokupi.
  - 6) Korisnik bira opciju "Sačuvaj".
  - 7) Sistem vrši validaciju podataka.
  - 8) Sistem obaveštava korisnika o uspešnosti akcije.
6. **Alternativni tokovi:**

**Neispravan unos:** 6a - Ako sistem pri validaciji podataka ustanovi da je unešen neispravan podatak ili da neki podatak nedostaje, ispisuje odgovarajuću poruku.

**Korisnik se već prijavio:** 2a - Ukoliko se korisnik već prijavio za izdavanje kartice, prikazuje mu se odgovarajuća poruka.

**Korisnik već poseduje karticu:** 2b - Ukoliko je korisniku već izdata kartica, prikazuju se sledeći podaci:

  - a) Broj kartice
  - b) Broj korišćenja kartice
  - c) Datum izdavanja

### 3.1.3.2 Pregled prijava za izdavanje kartice

1. **Kratak opis:** Administrator se prikazuje lista prijava za izdavanje kartice.
2. **Učesnici:** Administrator
3. **Preduslovi:** Administrator je ulogovan na sistem. Postoje prijave za izdavanje kartice.
4. **Postuslovi:** Otvorena je stranica sa listom prijava za izdavanje kartice.
5. **Osnovni tok:**
  - 1) Administrator odlazi na deo stranice za pregled prijava i predloga.
  - 2) Administrator ima uvid u sve podatke o prijavi za izdavanje kartice - podatke o korisniku koji se prijavio i datum prijave - kao i dve opcije "Prihvati prijavu" i "Odbij prijavu".
    - i. a) Administrator bira opciju "Prihvati prijavu".
    - b) Sistem obaveštava administratora o uspešnosti operacije.
    - c) Sistem generiše broj kartice, vezuje je za profil korisnika čiju je prijavu administrator prihvatio.

- d) Sistem generiše imejl sa podacima o korisniku i generisanoj kartici i šalje ga operaterima radi izdavanja fizičke kartice. Tok se nastavlja u slučaju upotrebe 3.1.3.3.
- ii. a) Administrator bira opciju "Odbij prijavu".  
b) Sistem obaveštava administratora o uspešnosti operacije.

6. **Alternativni tokovi:** /

### 3.1.3.3 Slanje fizičke kartice korisniku

1. **Kratak opis:** Operater nakon primljenog obaveštenja o izdatoj kartici bira praznu karticu, vezuje podatke korisnika na nju i izdaje je korisniku.
2. **Učesnici:** Operater, Korisnik
3. **Preduslovi:** Operater je ulogovan na sistem. Postoji izdata kartica.
4. **Postuslovi:** Korisnik je primio fizičku karticu.

5. **Osnovni tok:**

- 1) Operater otvara imejl sa obaveštenjem o izdatoj kartici.
- 2) Operater odlazi na deo sistema za vezivanje fizičkih kartica za profil korisnika.
- 3) Operater otvara podatke o korisniku na osnovu imejla.
- 4) Operater unosi podatke o izdatoj fizičkoj kartici.
- 5) Operater bira opciju "Poveži".
- 6) Operater predaje kuriru fizičku karticu i potvrdu o preuzimanju kartice koju korisnik potpisuje pri pouzecu.
- 7) Operater dobija potvrdu od kurira i proverava potpis korisnika.
- 8) Operater u sistemu označava da je korisnik primio karticu.

6. **Alternativni tokovi:**

**Korisnik je izabrao da lično pokupi karticu:** 6a - Korisnik dolazi u poslovnicu ICC-a. Operater mu predaje izdatu fizičku karticu. Slučaj upotrebe se nastavlja na koraku 8.

### 3.1.3.4 Korišćenje kartice prilikom kupovine karata - onlajn plaćanje preko IS-a ICC-a

1. **Kratak opis:** Korisnik koristi karticu prilikom onlajn kupovine karte radi ostvarivanja pogodnosti.
2. **Učesnici:** Korisnik
3. **Preduslovi:** Korisnik ima karticu ICC-a.
4. **Postuslovi:** Korisnik je kupio kartu. Ažuriran je spisak prijavljenih korisnika za događaj za koji je korisnik kupio kartu.

5. **Osnovni tok:**

- 1) Korisnik odlazi na deo stranice sa spiskom aktuelnih događaja.

- 2) Korisnik bira opciju "Kupi kartu" za događaj koji ga interesuje.
- 3) Korisniku se prikazuje popust na kupovinu koji može da ostvari ukoliko iskoristi poene sa svoje kartice, kao i ukupna cena karte.
- 4) Korisnik bira opciju "Plati".
- 5) Korisnik se vodi na spoljašnji sistem za onlajn plaćanje platnom karticom.
- 6) Nakon uspešnog plaćanja sistem korisniku dodaje odgovarajući broj poena na karticu, u zavisnosti od cene karte.
- 7) Sistem korisnika upisuje na spisak prijavljenih korisnika za događaj.
- 8) Sistem korisniku šalje potvrdu na imejl sa podacima o kupljenoj karti.

#### 6. Alternativni tokovi:

**Korisnik koristi poene sa kartice:** 4a - Korisnik je pre plaćanja izabrao i opciju za korišćenje poena radi postizanja popusta.

**Korisnik je birao korišćenje poena radi postizanja popusta:** 6a - Korisniku se skidaju poeni sa kartice na osnovu ostvarenog popusta. Korisnik ne dobija dodatne poene.

#### 3.1.3.5 Korišćenje kartice prilikom kupovine karata - uživo plaćanje u poslovnici ICC-a

1. **Kratak opis:** Korisnik koristi karticu prilikom kupovine karte uživo radi ostvarivanja pogodnosti.
2. **Učesnici:** Korisnik, Operater
3. **Preduslovi:** Korisnik ima karticu ICC-a.
4. **Postuslovi:** Korisnik je kupio kartu. Ažuriran je spisak prijavljenih korisnika za događaj za koji je korisnik kupio kartu.
5. **Osnovni tok:**
  - 1) Korisnik dolazi do poslovnice ICC-a.
  - 2) Korisnik izveštava operatera o događaju za koji želi da kupi kartu.
  - 3) Operater korisnika obaveštava o ceni karte i mogućnosti korišćenja kartice radi ostvarivanja popusta.
  - 4) Korisnik operateru daje svoje podatke i plaća kartu.
  - 5) Operater unosi podatke o korisniku u sistem i prijavljuje ga za učešće na događaju.
  - 6) Operater izdaje kartu korisniku.
  - 7) Sistem korisniku dodeljuje poene na karticu.
  - 8) Sistem korisnika upisuje na spisak prijavljenih korisnika za događaj.
  - 9) Sistem korisniku šalje potvrdu na imejl sa podacima o kupljenoj karti.

#### 6. Alternativni tokovi:

**Korisnik koristi poene sa kartice:** 4a - Korisnik je pre plaćanja izabrao i opciju za korišćenje poena radi postizanja popusta.

**Korisnik je birao korišćenje poena radi postizanja popusta:** 7a - Korisniku se skidaju poeni sa kartice na osnovu ostvarenog popusta. Korisnik ne dobija dodatne poene.

### 3.1.3.6 Korišćenje kartice prilikom kupovine karata - uživo plaćanje na šalteru partnerske ustanove kulture

1. **Kratak opis:** Korisnik koristi karticu prilikom kupovine karte uživo radi ostvarivanja pogodnosti.
2. **Učesnici:** Korisnik, Operater partnerske ustanove kulture
3. **Preduslovi:** Korisnik ima karticu ICC-a. Ustanova kulture ima dogovor sa ICC-om o kupovini karata korišćenjem kartice ICC-a.
4. **Postuslovi:** Korisnik je kupio kartu.
5. **Osnovni tok:**
  - 1) Korisnik dolazi do šaltera ustanove kulture koja organizuje događaj kom želi da prisustvuje.
  - 2) Korisnik izveštava operatera o događaju za koji želi da kupi kartu.
  - 3) Operater korisnika obaveštava o ceni karte i mogućnosti korišćenja kartice radi ostvarivanja popusta.
  - 4) Korisnik operateru daje svoje podatke i plaća kartu.
  - 5) Operater unosi podatke o korisniku u IS lokalne ustanove.
  - 6) IS lokalne ustanove prenosi podatke o korišćenju kartice IS-u ICC-a
  - 7) Operater izdaje kartu korisniku.
  - 8) IS ICC-a korisniku dodeljuje poene na karticu.
6. **Alternativni tokovi:**

**Korisnik koristi poene sa kartice:** 4a - Korisnik je pre plaćanja izabrao i opciju za korišćenje poena radi postizanja popusta.

**Korisnik je birao korišćenje poena radi postizanja popusta:** 8a - Korisniku se skidaju poeni sa kartice na osnovu ostvarenog popusta. Korisnik ne dobija dodatne poene.

## 3.2 Model podataka

Baza podataka se može posmatrati kroz tri glavna modula: upravljanje profilom, upravljanje događajima i upravljanje karticom. Model je napravljen na osnovu prepoznatih potrebnih podataka (odjeljak 2.3) i slučajeva upotrebe (odjeljak 3.1). Svakom entitetu su dodati odgovarajući atributi koji ostvaruju veze sa drugim entitetima. Dijagram entiteta i odnosa je prikazan na slici 3.4.

### 3.2.1 Modul upravljanje profilom

Prepoznati su naredni entiteti i atributi koji ih opisuju:

1. Korisnik (relacija *User*) - predstavlja jednog korisnika sistema. Opisuje se narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator korisnika u sistemu
  - Ime - niska koja predstavlja ime korisnika
  - Prezime - niska koja predstavlja prezime korisnika

- Imejl adresa - niska koja predstavlja imejl adresu kojom se korisnik registrovao
- Tip korisnika - niska koja predstavlja tip korisnika (Korisnik, Predavač, Administrator, Operater)
- Šifra - kriptovana niska koja predstavlja šifru kojom se korisnik prijavljuje na sistem
- Korisničko ime - niska koju korisnik koristi prilikom prijavljivanja na sistem
- Zemlja - niska koja predstavlja zemlju prebivališta korisnika
- Grad - niska koja predstavlja grad prebivališta korisnika
- Adresa - niska koja predstavlja ulicu prebivališta korisnika
- Broj telefona - niska koja predstavlja kontakt broj telefona korisnika
- Status prijave za predavačku poziciju - niska koja predstavlja korak u procesu prijavljivanja za predavačku poziciju (nije poslata prijava, prijava je poslata, prijava je odbijena i prijava je prihvaćena)
- Vreme pravljenja korisničkog profila - vremenska oznaka koja predstavlja datum i vreme kada je korisnik napravio profil
- Status verifikacije imejl adrese - istinitosna vrednost koja predstavlja da li je korisnik verifikovao svoju imejl adresu ili ne

U odnosu na analizu, dodati su atributi *status verifikacije imejl adrese* i *status prijave za predavačku poziciju*. Ovi atributi nisu ključni u opisu jednog Korisnika sistema, ali su nastali kao posledice slučajeva upotrebe. Kada se korisnik prijavi za predavačku poziciju (slučaj upotrebe 3.1.1.5), potrebno je znati da li je njegova prijava uspešna ili ne, radi prikazivanja odgovarajuće poruke. Potrebna je i informacija o tome da li je korisnik verifikovao svoju imejl adresu kako bi mogao da neometano koristi sistem (slučaj upotrebe 3.1.1.2).

2. Predavač (relacija *Presenter*) - predstavlja specijalizaciju entiteta Korisnik. Dodatno se karakteriše narednim atributom:
  - ID korisnika - ceo broj koji predstavlja jedinstveni identifikator korisnika u sistemu
  - Specijalizacija - niska koja predstavlja oblast za koju je predavač specijalizovan
  - Aktivan - istinitosna vrednost koja predstavlja da li je predavač trenutno aktivan ili ne (da li ima pravo da organizuje događaje)
3. Administrator (relacija *Admin*) - predstavlja specijalizaciju entiteta Korisnik. Dodatno se karakteriše narednim atributom:
  - ID korisnika - ceo broj koji predstavlja jedinstveni identifikator korisnika u sistemu
  - ID ustanove kulture - ceo broj koji predstavlja vezu sa ustanovom kulture čiji je korisnik administrator
4. Operater (relacija *Operator*) - predstavlja specijalizaciju entiteta Korisnik. Dodatno se karakteriše narednim atributom:
  - ID korisnika - ceo broj koji predstavlja jedinstveni identifikator korisnika u sistemu

- ID ustanove kulture - ceo broj koji predstavlja vezu sa ustanovom kulture čiji je korisnik operater
5. Lokalna ustanova kulture (relacija *LocalEstablishment*) - predstavlja ustanovu kulture koja može a i ne mora imati saradnju sa ICC-om. Opisuje se narednim atributima:
- ID - ceo broj koji predstavlja jedinstveni identifikator ustanove u sistemu
  - Naziv - niska koja predstavlja naziv ustanove kulture
  - Zemlja - niska koja predstavlja zemlju u kojoj se ustanova kulture nalazi
  - Grad - niska koja predstavlja grad u kom se ustanova kulture nalazi
  - Adresa - niska koja predstavlja ulicu u kojoj se ustanova kulture nalazi
  - Tip - ceo broj koji predstavlja vezu sa tipom ustanove kulture
  - Saradnja - istinitosna vrednost koja predstavlja da li je ostvarena sradnja sa ustanovom ili ne

U odnosu na atribute prepoznate u analizi, atribut *tip* je pretvoren u vezu sa relacijom *Tip ustanove kulture* radi lakšeg održavanja baze podataka.

6. Tip ustanove kulture (relacija *LocalEstablishmentType*) - predstavlja tip ustanove kulture. Opisuje ga naredni atribut:
- ID - ceo broj koji predstavlja jedinstveni identifikator tipa ustanove kulture u sistemu
  - Tip ustanove - niska koja opisuje tip ustanove kulture (muzej, pozorište, kulturni centar i slično)
7. Ocena (relacija *Rating*) - predstavlja ocenu koju je korisnik dodelio događaju i samim tim predavačima. Opisana je narednim atributima:
- ID - ceo broj koji predstavlja jedinstveni identifikator ocene u sistemu
  - Komentar - niska koja sadrži opcioni komentar korisnika o događaju
  - Ocena - ceo broj na skali od 1 do 5
  - Vreme ocenjivanja - vremenska oznaka kada je data ocena
  - ID korisnika - ceo broj koji predstavlja vezu sa korisnikom koji je dao ocenu događaja
  - ID predavača - ceo broj koji predstavlja vezu sa predavačem koji je dobio ocenu
  - ID događaja - ceo broj koji predstavlja vezu sa događajem koji je dobio ocenu
8. Token (relacija *VerificationToken*) - predstavlja token koji se vezuje za verifikaciju imejl adrese korisnika. Opisan je atributom:
- ID - ceo broj koji predstavlja jedinstveni identifikator tokena u sistemu
  - Token - niska koja predstavlja generisani token za verifikaciju konkretne imejl adrese
  - ID korisnika - ceo broj koji predstavlja vezu sa korisnikom kom je poslat verifikacioni imejl sa konkretnim tokenom



9. Prijava za predavačku poziciju (relacija *PresenterApplication*) - predstavlja prijavu korisnika za predavačku poziciju. Opisana je narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator prijave u sistemu
  - Specijalizacija korisnika - niska koja predstavlja oblast za koju je predavač specijalizovan
  - Prorpatno pismo - niska koja predstavlja komentar koji je korisnik ostavio uz prijavu koja detaljnije opisuje njegove veštine i motivaciju za prijavu
  - Vreme prijave - vremenska oznaka kada je prijava poslata
  - ID korisnika - ceo broj koji predstavlja vezu sa korisnikom koji se prijavio za predavačku poziciju
10. Odbijena prijava za predavačku poziciju (relacija *DeclinedPresenterApplication*) - predstavlja odbijenu prijavu za predavačku poziciju korisnika. Kako predstavlja specijalizaciju Prijave za predavačku poziciju, opisana je svim atributima tog entiteta kao i narednim atributom:
  - ID prijave - ceo broj koji predstavlja jedinstveni identifikator prijave u sistemu
  - Razlog odbijanja - niska koja predstavlja objašnjenje koje je administrator ostavio prilikom odbijanja prijave

### 3.2.2 Modul upravljanje događajima

Prepoznati su naredni entiteti i atributi koji ih opisuju:

1. Događaj (relacija *Event*) - predstavlja jedan događaj. Opisan je narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator događaja u sistemu
  - Naziv - niska koja predstavlja naziv događaja
  - Grad - niska koja predstavlja grad u kom se održava događaj
  - Država - niska koja predstavlja grad u kom se održava događaj
  - Mesto održavanja - niska koja predstavlja ustanovu kulture u kojoj se događaj održava
  - Datum i vreme - vremenska oznaka koja predstavlja tačno vreme početka događaja
  - Opis - niska koja opisuje događaj
  - Cena karte - niska koja predstavlja cenu karte za prisustovanje događaju
  - Potrebna karta - istinitosna vrednost koja predstavlja da li je potrebno kupiti kartu za događaj ili ne
2. Predavač na događaju (relacija *EventsPresenter*) - predstavlja vezu između predavača i događaja - na kojim događajima predaju koji predavači. Ovaj entitet je opisan atributima:
  - ID događaja - ceo broj koji predstavlja vezu sa događajem
  - ID predavača - ceo broj koji predstavlja vezu sa predavačem koji događaj drži

3. Učesnik na događaju (relacija *EventsAttendee*) - predstavlja vezu između korisnika i događaja - na koje događaje se koji korisnik prijavio. Opisan je narednim atributima:
  - ID događaja - ceo broj koji predstavlja vezu sa događajem
  - ID korisnika - ceo broj koji predstavlja vezu sa korisnikom koji se prijavio da prisustvuje događaju
  - Status - ceo broj koja predstavlja vezu sa statusom učesnika na događaju
4. Status učesnika na događaju (relacija *AttendeeStatu*) - predstavlja status korisničke prijave na događaj. Opisuje ga naredni atribut:
  - ID - ceo broj koji predstavlja jedinstveni identifikator tipa prijave u sistemu
  - Tip prijave - niska koja opisuje tip prijave (da li je korisnik prijavljen, da li je korisnik zainteresovan, da li je korisnik već kupio kartu)
5. Predlog događaja (relacija *EventSuggestion*) - predstavlja predlog novog događaja nekog predavača. Okarakterisan je narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator predloga u sistemu
  - Naziv - niska koja predstavlja naziv događaja
  - Grad - niska koja predstavlja grad u kom se održava događaj
  - Država - niska koja predstavlja grad u kom se održava događaj
  - Mesto održavanja - niska koja predstavlja ustanovu kulture u kojoj se događaj održava
  - Datum i vreme - vremenska oznaka koja predstavlja tačno vreme početka događaja
  - Opis - niska koja opisuje događaj
  - ID predavača - ceo broj koji predstavlja vezu sa predavačem koji je poslao predlog događaja
6. Odbijeni predlog događaja (relacija *DeclinedEventSuggestion*) - predstavlja odbijen predlog događaja. Kako predstavlja specijalizaciju Predloga događaja, okarakterisan je svim atributima tog entiteta kao i narednim atributom:
  - ID predloga - ceo broj koji predstavlja jedinstveni identifikator predloga u sistemu
  - Razlog odbijanja - niska koja predstavlja objašnjenje koje je administrator ostavio prilikom odbijanja predloga

### 3.2.3 Modul upravljanje karticom

Prepoznati su naredni entiteti i atributi koji ih opisuju:

1. Kartica (relacija *Card*) - predstavlja karticu korisnika. Opisana je narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator kartice u sistemu
  - Broj kartice - ceo broj koji predstavlja jedinstveni broj kartice
  - Tip kartice - ceo broj koji predstavlja vezu sa tipom kartice
  - Status - istinitosna vrednost koja predstavlja da li je kartica aktivna ili ne

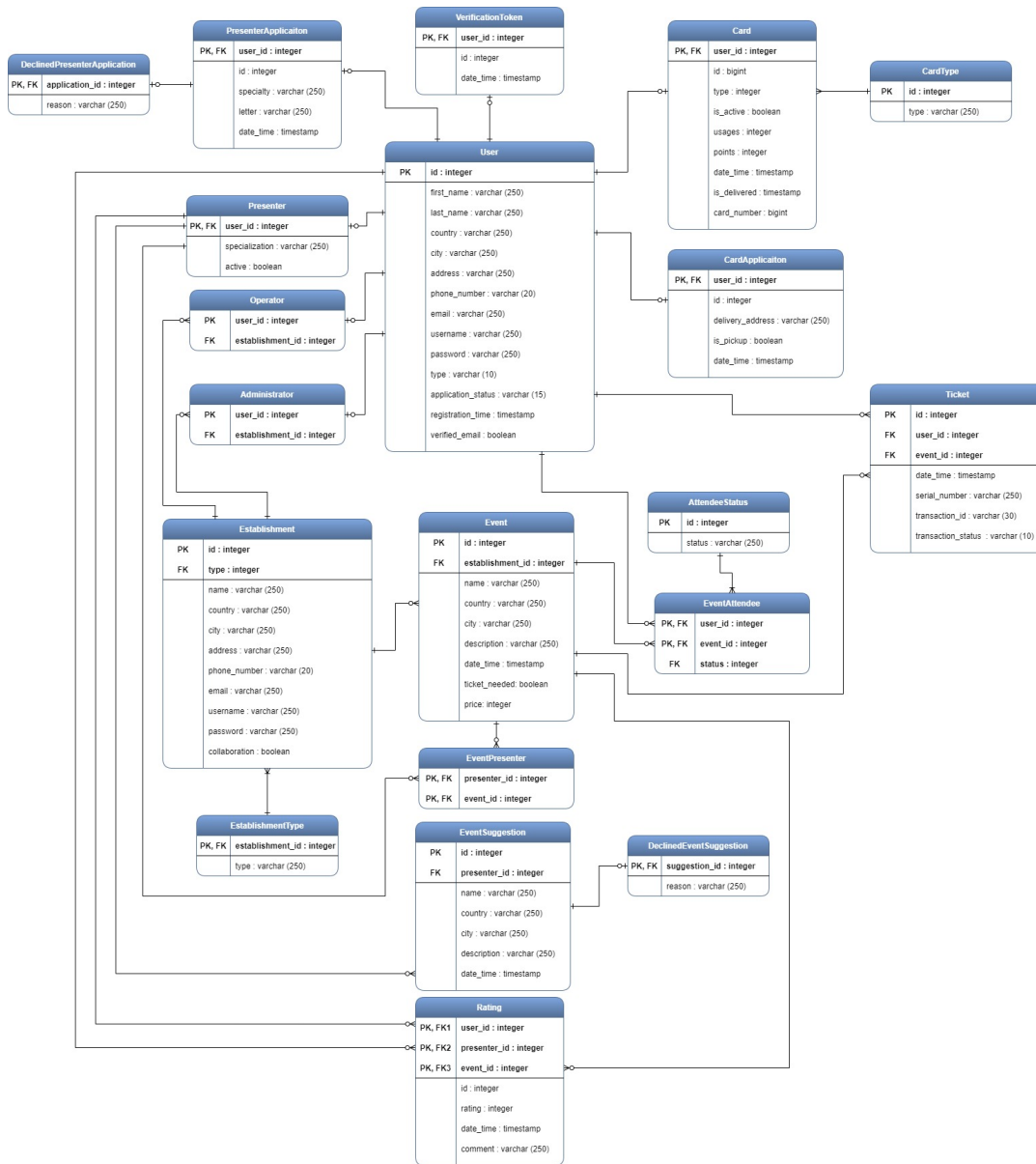
- Broj korišćenja - ceo broj koji predstavlja broj korišćenja kartice prilikom kupovine karata
- Broj poena - ceo broj koji predstavlja trenutni broj poena na kartici koji se mogu koristiti radi ostvarivanja popusta
- Datum izdavanja - vremenska oznaka koja predstavlja vreme izdavanja kartice
- Dostava - istinitosna vrednost koja predstavlja da li je fizička kartica dostavljena korisniku
- ID korisnika - ceo broj koji predstavlja vezu sa korisnikom

Pored atributa prepoznatih u analizi, postoji i atribut *dostava* koji je nastao kao posledica slučaja upotrebe 3.1.3.3. Atribut *tip* je pretvoren u vezu sa relacijom *Tip kartice* radi lakšeg održavanja baze podataka.

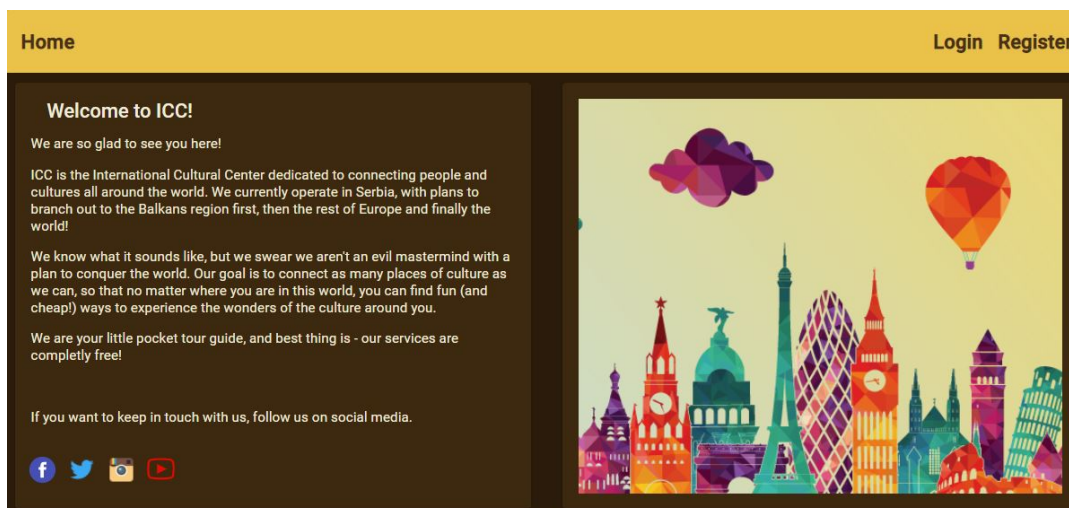
2. Tip kartice (relacija *CardType*) - predstavlja tip kartice korisnika. Opisana je narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator tipa kartice u sistemu
  - Tip kartice - niska koja opisuje tip kartice (da li je u pitanju obična kartica, da li je kartica privremeno blokirana)
3. Prijava za izdavanje kartice (relacija *CardApplication*) - predstavlja prijavu korisnika za izdavanje kartice. Okarakterisana je narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator prijave u sistemu
  - ID korisnika - ceo broj koji predstavlja vezu sa korisnikom koji se prijavio za izdavanje kartice
  - Adresa - opcionalna niska koja predstavlja adresu za dostavu fizičke kartice
  - Lično preuzimanje - istinitosna vrednost koja predstavlja da li korisnik želi lično da preuzme svoju karticu na šalteru ICC-a
  - Datum - vremenska oznaka koja predstavlja vreme prijave za izdavanje kartice

Atribut *ID korisnika* predstavlja sve podatke o korisniku koji su navedeni u analizi. Kao posledica slučaja upotrebe 3.1.3.1, dodat je i atribut *lično preuzimanje*.

4. Karta za događaj (relacija *Ticket*) - predstavlja kupljenu kartu za događaj. Okarakterisana je narednim atributima:
  - ID - ceo broj koji predstavlja jedinstveni identifikator karte u sistemu
  - ID korisnika - ceo broj koji predstavlja vezu sa korisnikom koji je kupio kartu
  - Redni broj - ceo broj koji predstavlja redni broj karte (karte su često numerisane van njihovih identifikatora u sistemu)
  - ID događaja - ceo broj koji predstavlja vezu sa događajem za koji je kupljena karta
  - Datum - vremenska oznaka koja predstavlja vreme kupovine karte
  - ID transakcije - niska koja označava jedinstveni identifikator kupovine karte
  - Status transakcije - niska koja označava status kupovine (na početku ima vrednost "započeto" a nakon završetka kupovine može imati vrednosti "uspešno" ili "neuspešno")



SLIKA 3.4: Dijagram entiteta i odnosa



SLIKA 3.5: Početna stranica

### 3.3 Interfejs

Interfejs aplikacije informacionog sistema predstavlja sponu između korisnika i informacionog sistema. Zamišljen je da bude intuitivan i lak za korišćenje. Korisnicima je potrebno prikazati veliku količinu informacija, a opet ih njima ne preopteretiti, tako da se pri projektovanju interfejsa vodilo računa o balansiranju ta dva aspekta.

Svim korisnicima, i registrovanim i neregistrovanim, su dostupne stranice kao što su “O nama”, “Aktuelni događaji”, “Zemlje sa kojima se saradjuje”, “Kontakt” i služe isključivo za dobijanje informacija. Takođe, neophodni su interfejsi za rad operatera iz poslovnica ICC-a i za rad operatera i administratora lokalnih ustanova kulture sa kojima se saraduje. U prototipu je implementiran deo namenjen samo administratoru, deo namenjen samo predavaču kao i deo aplikacije koji je zajednički za administratora i predavača. U prototipu nije implementiran interfejs koji predstavlja osnovno informisanje o radu i ponudama međunarodnog kulturnog centra kao ni interfejsi za operatere iz poslovnica i lokalnih ustanova. Ilustracije u nastavku ovog poglavlja su uzete iz implementacije prototipa aplikacije.

#### 3.3.1 Deo aplikacije zajednički za administratora i predavača

Na slici 3.5 je prikazana početna stranica koja se otvara po ulasku na veb sajt. Sadrži kratak opis glavne ideje iza ICC-a kao i veze ka društvenim mrežama preko kojih ljudi mogu doći do dodatnih informacija.

Da bi bilo moguće pristupiti ključnim mogućnostima informacionog sistema, potrebno je posedovati korisnički profil na njemu. Neregistrovani korisnik ima pristup samo dvema stranicama - “Registracija” i “Prijavljivanje”. Stranica “Registracija” pruža mogućnost pravljenja korisničkog profila (slika 3.6). Kako bi neregistrovani korisnik napravio profil, potrebno je da unese svoju imejl adresu, željeno korisničko ime, svoje ime i prezime i željenu lozinku. Odabirom opcije “Registruj se” na sistemu se pravi profil i šalje se imejl za verifikaciju na imejl adresu koju je korisnik uneo.

Nakon registracije, korisniku se prikazuje stranica za prijavljivanje na sistem, na kojoj unose svoje korisničko ime i loziku (slika 3.7). U slučaju da mu nije stigao imejl za potvrdu registracije, može odavratiti opciju ponovnog slanja imejla. Otvara se formular za unošenje korisničkog imena. Kada korisnik unese svoje korisničko ime i odabere opciju “Pošalji”, ponovo će mu se poslati verifikacioni imejl.

Home Login Register

## Register

Email

Username

First name

Last name

Password

[Already have an account? Go ahead and login!](#)

SLIKA 3.6: Stranica za registraciju korisnika

Home Login Register

## Login

Username

Password

Login

Didn't receive the confirmation email?

Don't have an account? Go ahead and register!

SLIKA 3.7: Stranica za prijavljivanje korisnika

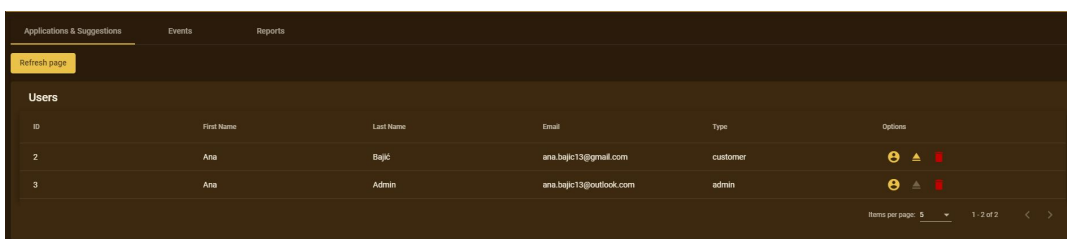
### 3.3.2 Deo aplikacije namenjen administratoru

Kada se na sistem prijavi administrator, prikazuje mu njegova početna stranica. Na njoj može pristupiti jednom od tri odeljka - “Prijave i predlozi”, “Događaji” i “Izveštaji”. Podrazumevano je otvoren odeljak “Prijave i predlozi”.

#### 3.3.2.1 Prijave i predlozi

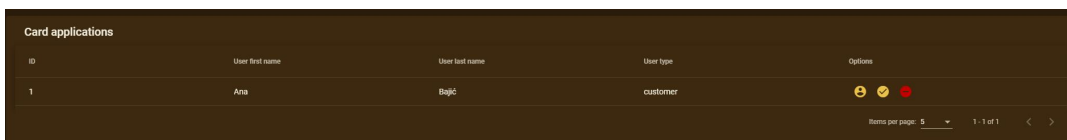
Odeljak “Prijave i predlozi” administratoru pruža uvid u spisak svih korisnika sistema, spisak prijava za izdavanje kartice, spisak prijava za poziciju predavača i spisak svih predloga novih događaja.

Administrator ima mogućnost pregleda korisničkih profila kao i brisanje korisničkih profila. U slučaju brisanja profila, potrebno je da administrator popuni formu sa razlogom brisanja (slika 3.8).



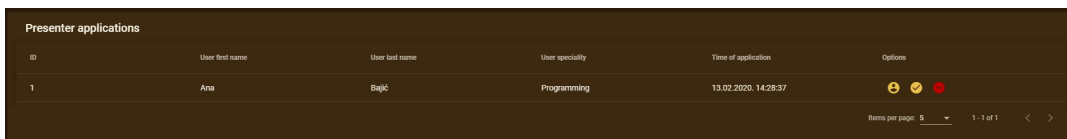
SLIKA 3.8: Deo odeljka “Prijave i predlozi” za pregled korisničkih profila u administratorskom delu aplikacije

Administrator može prihvatiti ili odbiti prijavu za izdavanje kartice kulturnog centra. Pre nego što odluči, može pregledati korisnički profil korisnika koji se prijavio za izdavanje. U slučaju da izabere da odbije prijavu, mora ostaviti i razlog odbijanja (slika 3.9).



SLIKA 3.9: Deo odeljka “Prijave i predlozi” za pregled prijava za izdavanje kartice u administratorskom delu aplikacije

Administrator može da pregleda i prijave za predavačku poziciju. Osim što ima uvid u korisnički profil korisnika koji se prijavio za poziciju, može videti i kvalifikaciju korisnika kao i prpratno pismo koje je ostavio prilikom prijave. Administrator može prihvatiti i odbiti prijavu. U slučaju da izabere da odbije prijavu, mora ostaviti i razlog odbijanja (slika 3.10).



SLIKA 3.10: Deo odeljka “Prijave i predlozi” za pregled prijava za poziciju predavača u administratorskom delu aplikacije



Administrator može pregledati i prihvatiti ili odbiti sve predložene događaje. Za svaki predlog, ima opciju pregleda korisničkog profila predavača koji je događaj predložio kao i dve opcije - prihvatanje i odbijanje predloga. U slučaju da izabere da odbije predlog, mora ostaviti i razlog odbijanja.

### 3.3.2.2 Događaji

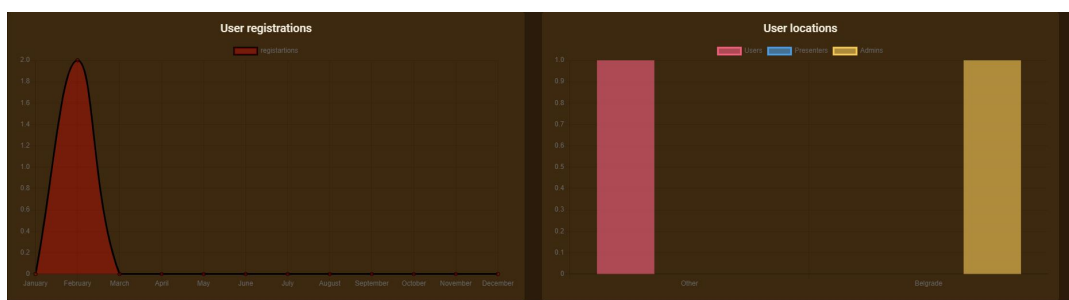
Odeljak “Događaji” pruža uvid u spiskove svih aktuelnih i završenih događaja. Dodatno, omogućava administratoru da dodaje nove događaje kao i da aktuelne događaje promeni (slika 3.11).

Applications & Suggestions									
Events									
Reports									
Refresh page Add new event									
Current events									
ID	Name	City	Country	Venue	Time of event	Description	Presenters	Options	
1	Movie night - Star Wars	Belgrade	Serbia	KC Čukarica	21.02.2020. 18:00:00				
Items per page: 5 1 - 1 of 1									
Past events									
ID	Name	City	Country	Venue	Time of event	Description	Presenters	Options	
2	Look at the date! Look at it!	Belgrade	Serbia	KC Čukarica	02.02.2020. 20:20:00				
Items per page: 5 1 - 1 of 1									

SLIKA 3.11: Odeljak “Događaji” u administratorskom delu aplikacije

### 3.3.2.3 Izveštaji

Odeljak “Izveštaji” pruža statistički uvid u informacioni sistem - ukupan broj događaja po gradu, ukupan broj događaja po gradu trenutnog dana, ukupan broj kreiranja korisničkih profila po mesecima, broj korisnika po državama i slično (slika 3.12).



SLIKA 3.12: Odeljak “Izveštaji” u administratorskom delu aplikacije

## 3.3.3 Deo aplikacije namenjen predavaču

Kada se na sistem prijavi predavač, prikazuje mu se njegov korisnički profil, sa četiri odeljka - “Osnovne informacije”, “Događaji”, “Kartica” i “Mapa”. Podrazumevano je otvoren odeljak “Osnovne informacije”.

### 3.3.3.1 Osnovne informacije

Odeljak “Osnovne informacije” (slika 3.15) pruža predavaču uvid u svoje lične podatke - imejl adresu, ime, prezime, adresu, grad, državu i broj telefona. Pored

toga, ima mogućnost ažuriranja profila, pri čemu pored ovih podataka može izmeniti i svoju lozinku (slika 3.13). Predavač može i izbrisati svoj profil, pri čemu je potrebno da unese svoju lozinku kako bi potvrdio svoju odluku (slika 3.14).

SLIKA 3.13: Odeljak “Osnovne informacije” i opcija ažuriranja profila

SLIKA 3.14: Odeljak “Osnovne informacije” i opcija brisanja profila

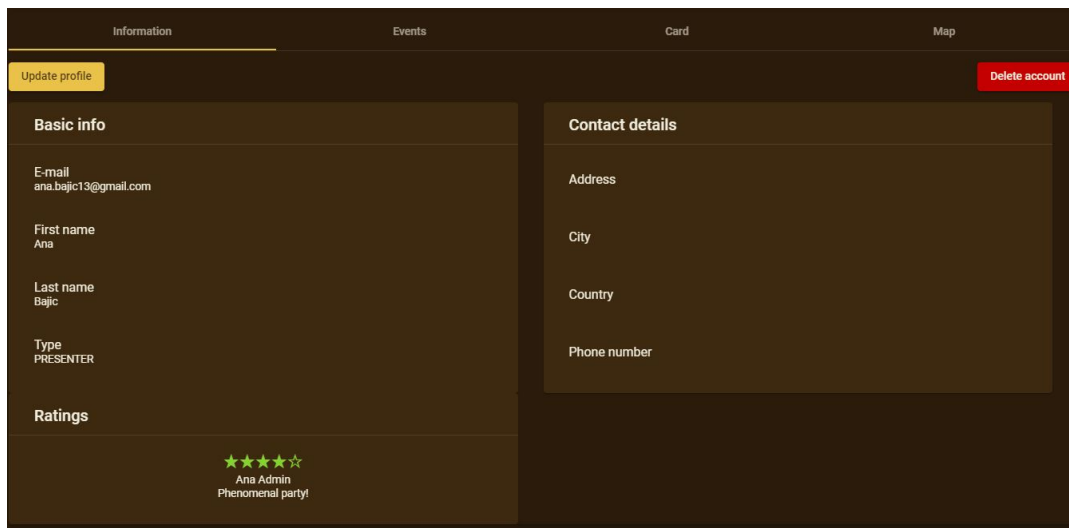
Predavač može pregledati i svoje ocene. Prikazana mu je ukupna ocena (prosečna ocena svih datih ocena) kao i sve pojedinačne ocene i komentari koje su drugi korisnici ostavili.

### 3.3.3.2 Događaji

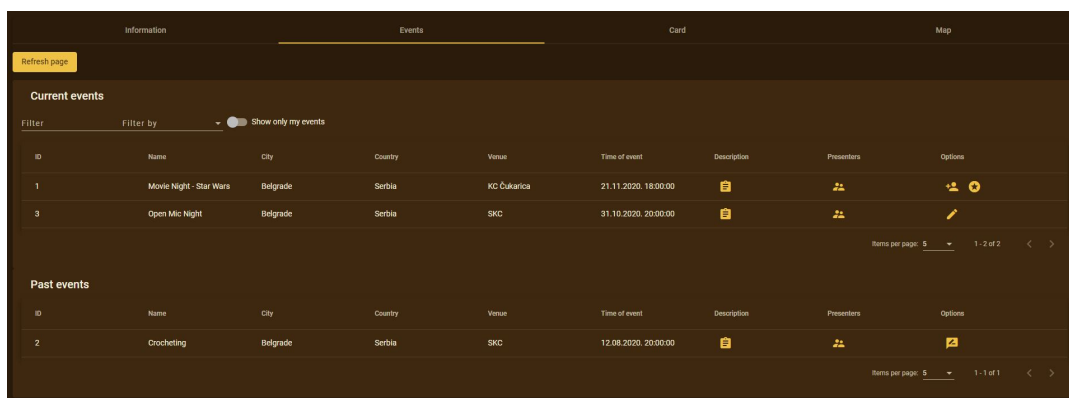
Odeljak “Događaji” (slika 3.16) pruža predavaču uvid u aktuelne događaje, kao i događaje koji su se završili. Predavač može da izabere opciju da se prikažu samo njegovi događaji - oni koje je on organizovao kao i oni na kojima učestvuje ili kao posmatrač ili kao predavač. Tabelu aktuelnih događaja može i filtrirati po gradu, državi i mestu održavanja.

Uz svaki događaj postoji nekoliko opcija - prijava za učešće na događaju, prijava zainteresovanosti za događaj a ako je u pitanju događaj koji je predavač organizovao onda i opciju ažuriranja događaja. Prilikom odabira opcije za ažuriranje događaja, otvara se popunjeni formular sa slike 3.17. Predavač može menjati sve podatke, s tim da ne može datum i vreme održavanja podesiti tako da događaj bude u prošlosti. U slučaju da predavač pokuša to da uradi, sistem će mu ispisati odgovarajuću poruku.

Predavač takođe ima i mogućnost dodavanja predloga novog događaja odabirom opcije “Dodaj novi predlog događaja” (slika 3.17). Predavač mora uneti sve podatke u formular. U slučaju da neki ne unese, sistem će ispisati odgovarajuću poruku. Jedino



SLIKA 3.15: Odeljak “Osnovne informacije” u predavačkom delu aplikacije



SLIKA 3.16: Odeljak “Događaji” u predavačkom delu aplikacije

polje koje nije obavezno je polje za dodavanje predavača. U slučaju da predavač to polje ostavi prazno, podrazumevano će on biti sačuvan kao predavač na tom događaju.

**Edit event**

Event name  
Crochet - Intro class! Make those loops!

City  
Belgrade

Country  
Serbia

Venue  
KC Cukarica

start date  
20-Feb-2020 17:30

Description  
Come one, come all! We will learn the basics of crochet in the 2 hours the event lasts. I will teach you - yes, every single one of you - following skills: making a chain, single crochet, double crochet, increases and decreases and the magic of the magic circle.

Add presenters  
Ana Bajic

Finish Cancel

SLIKA 3.17: Odeljak “Dogadaji” i opcija dodavanja i izmene događaja u predavačkom delu aplikacije

Prilikom odabira opcije za prijavljivanje za učešće na događaju, u slučaju da je za taj događaj potrebna ulaznica, sistem predavača preusmerava na stranicu eksternog sistema za plaćanje (slika 3.18). Na toj stranici, predavač unosi podatke o svojoj platnoj kartici.

**PAYMENT SERVICE**

Name on Card

Card Number

CVC  
ex. 311

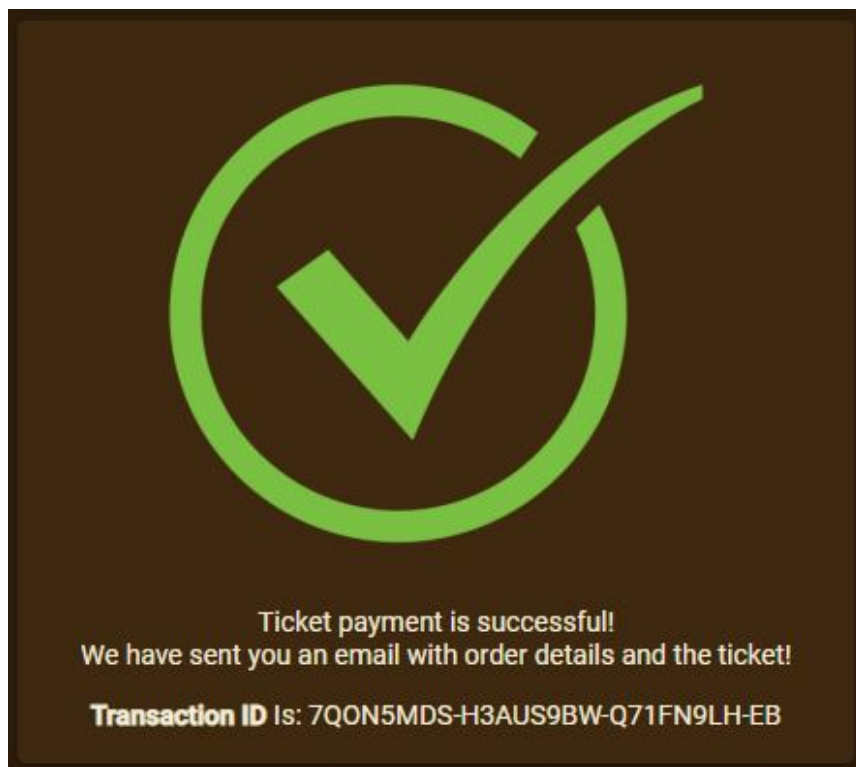
Expiration  
MM YYYY

Total: 200 RSD

Pay >

SLIKA 3.18: Stranica za unos podataka sa platne kartice radi plaćanja

Nakon što izabere opciju da izvrši plaćanje, preusmerava se na stranicu sistema na kojoj mu se prikazuju podaci o uspešnoj kupovini (slika 3.19).



SLIKA 3.19: Obaveštenje o uspešnoj transakciji

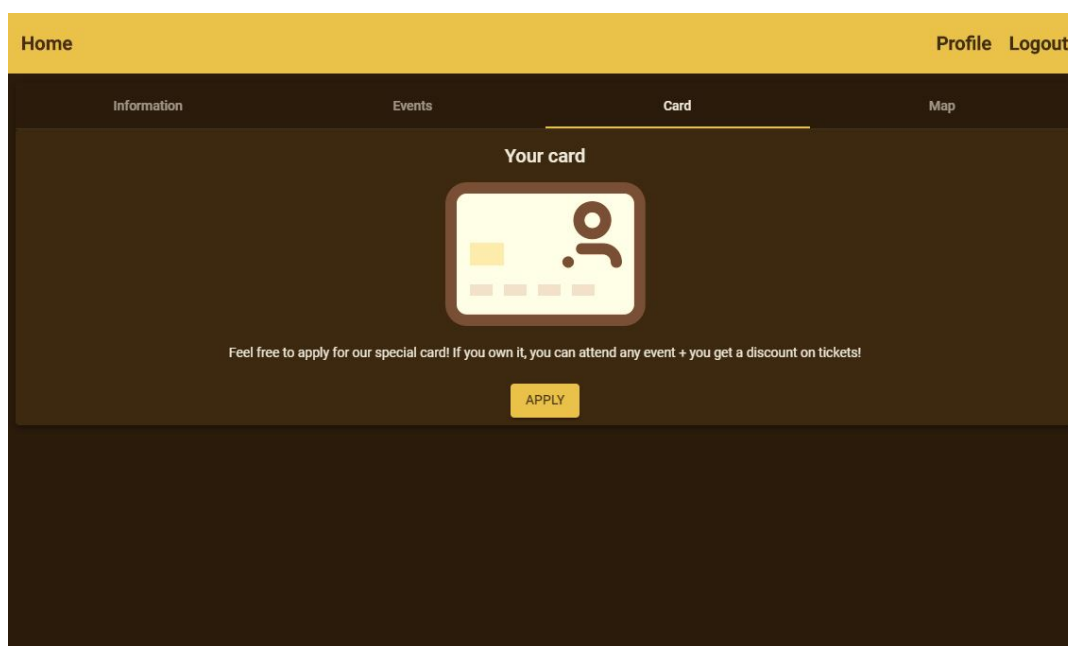
### 3.3.3.3 Kartica

Odeljak "Kartica" pruža predavaču mogućnost da se prijavi za karticu, a ako je poseduje da vidi podatke o njoj (slike 3.20, 3.21). U slučaju da mu je prijava za izdavanje kartice odbijena, na ovom mestu će postojati obaveštenje sa razlogom odbijanja i predavač će opet imati mogućnost prijave za izdavanje kartice.

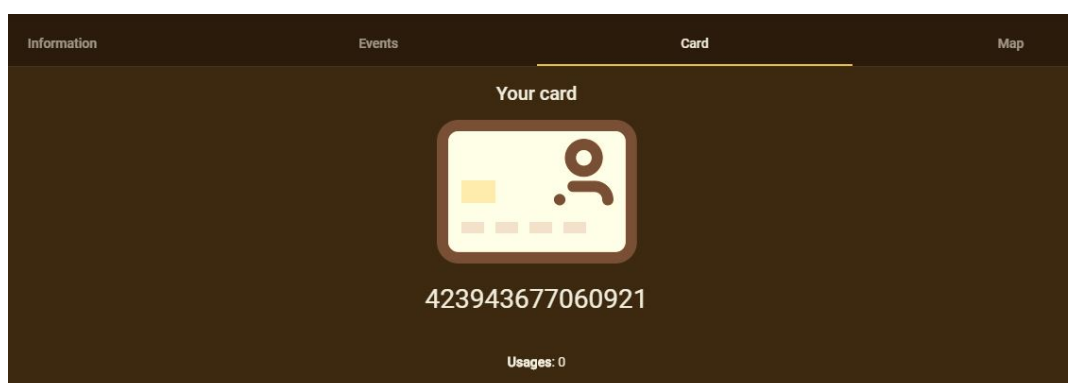
Predavaču koji poseduje karticu se prikazuju podaci kao što su broj kartice (koji koristi radi dobijanja popusta pri kupovini ulaznica za događaje), broj upotreba, datum izdavanja i broj osvojenih poena.

U slučaju da je prijava za izdavanje kartice odbijena, predavaču se prikazuje obaveštenje sa razlogom odbijanja.

U budućim iteracijama implementacije biće omogućeno korišćenje osvojenih poena. Predavač će moći da izabere neku od "nagrada" i potroši svoje poene. U slučaju da je izabrao neku od "fizičkih" nagrada a nema zabeleženu adresu stanovanja, sistem će ga obavestiti da mora da ažurira svoje podatke pre nego što tu nagradu odabere. Takođe, u slučaju da se predavač prijavio da prisustvuje nekom događaju za koji je potrebno kupiti kartu, na ovoj stranici će mu biti prikazana veza sa informacionim sistemom ustanove u kojoj se taj događaj organizuje ili sa informacionim sistemom organizatora događaja koja vodi ka stranici za kupovinu ulaznice, na kojoj može da iskoristi svoju karticu.



SLIKA 3.20: Odeljak korisničkog profila za prikaz kartice - slučaj kad predavač nema karticu

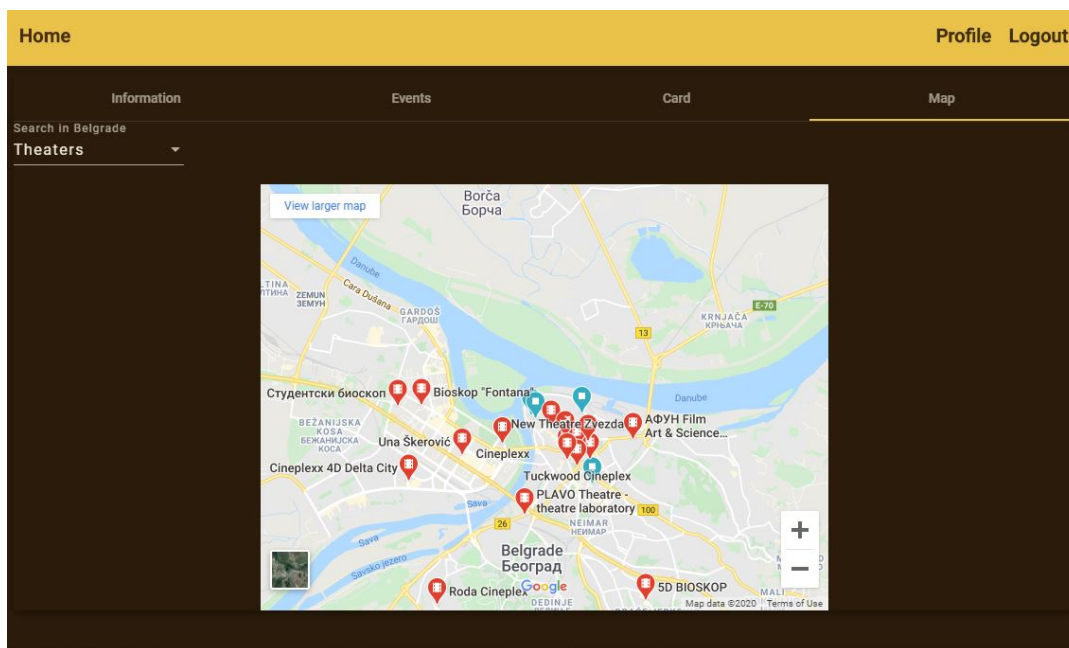


SLIKA 3.21: Odeljak korisničkog profila za prikaz kartice - slučaj kad predavač ima karticu

### 3.3.3.4 Mapa

Odeljak "Mapa" pruža predavaču mogućnost da pregleda centre kulture u svojoj blizini - muzeje, pozorišta, stadione, parkove, spomenike, crkve i hramove, palate. Takođe ima i opciju da odabere pregled svih centara kulture u svojoj blizini (slika 3.22).

U budućim iteracijama implementacije biće dodata mogućnost pregleda i samih događaja u blizini korisnika, na mapi će posebno biti naznačeni centri sa kojima međunarodni kulturni centar blisko saraduje a postojaće i globalni pregled događaja i centara. Dodatno, korisnik će imati mogućnost da napravi i spisak mesta koje želi da poseti i biće mu generisana optimalna putanja obliska.



SLIKA 3.22: Odeljak korisničkog profila za prikaz mape

### 3.3.4 Deo aplikacije namenjen operateru

Operater bi se, nakon prijave na sistem, prikazao interfejs pomoću kog bi mogao da vrši sve svoje poslove prepoznate analizom sistema. Operateru se prikazuje početna stranica sa odeljcima *“Kreiranje korisničkog profila”*, *“Pregled kartica”* i *“Plaćanje”*. U prototipu nije implementiran ovaj deo aplikacije ali je dat kratak pregled mogućnosti koje bi pružao.

Odeljak *“Kreiranje korisničkog profila”* služi za pravljenje profila za korisnika koji lično dođe u poslovnici. Operateru se prikazuje forma u koju je potrebno da unese podatke koje mu korisnik da, kao što su njegovo ime, prezime, kontak telefon, imejl adresa i izabrano korisničko ime. Sistem bi korisniku generisao proizvoljnu šifru koja mu se šalje na imejl adresu, a operater bi korisnika uputio da šifru promeni.

Odeljak *“Pregled kartica”* operater daje uvid u izdate kartice. Za svaku karticu operater može da vidi da li ju je potrebno poslati poštom ili ne. Operater svaku izdatu karticu koja nije dostavljena korisniku povezuje sa fizičkom karticom. Deo stranice služi za označavanje da li je korisnik primio karticu.

Odeljak *“Plaćanje”* omogućava operateru da asistira korisnike prilikom kupovine karata za prisustvovanje događaju. Operater ima opciju da unese naziv događaja za koji korisnik želi da kupi kartu, odabere opciju za kupovinu i unese podatke o kartici korisnika.

### 3.3.5 Deo aplikacije namenjen osnovnom informisanju korisnika

Svakom korisniku sistema se prilikom otvaranja glavne stranice aplikacije prikazuju odeljci *“O nama”*, *“Aktuelni događaji”*, *“Zemlje sa kojima se saraduje”* i *“Kontakt”*. U prototipu nije implementiran ovaj deo aplikacije ali je dat kratak pregled mogućnosti koje bi pružao.

Odeljak *“O nama”* pruža osnovne informacije o ICC-u. Dostupne informacije uključuju vreme osnivanja, glavnu ideju iza projekta, kratke biografije osnivača i politiku poslovanja.

Odeljak "Aktuelni događaji" pruža svim korisnicima pregled osnovnih informacija o aktuelnim događajima kao što su vreme i mesto održavanja i opis. Pored svakog događaja postoji opcija "Više" koja trenutno neprijavljene korisnike vodi na stranicu za prijavljivanje na sistem a prijavljene korisnike vodi na detaljan prikaz događaja opisan u odeljku 3.3.3.2.

Odeljak "Zemlje sa kojima se saraduje" pruža pregled zemalja u kojima ICC ima svoje ogranke. Na ovoj stranici korisnici mogu da dobiju informaciju o zemljama čija se kultura promovise.

Odeljak "Kontakt" korisnicima pruža ažurne informacije o načinu na koji mogu kontaktirati ICC. Prikazana je imejl adresa i kontakt telefoni, pri čemu korisnik može da odabere svoju zemlju kako bi mu se prikazao telefon lokalne poslovnice ICC-a. Pored toga, na stranici postoji polje koje omogućava svakom korisniku, koji poseduje validnu imejl adresu, da pošalje poruku ICC-u.

### 3.3.6 Deo aplikacije namenjen operaterima i administratorima lokalnih ustanova kulture

U zavisnosti od tipa i nivoa ostvarene saradnje između ICC-a i lokalne ustanove kulture, operaterima i administratorima tih ustanova bi bio prikazan drugačiji interfejs. Postoji stranica slična onoj koju koristi administrator sistema ICC-a, koja je opisana u odeljku 3.3.2, s tim da bi bila modifikovana potrebama lokalnih ustanova. Postoje odeljci "Pregled korisnika", "Pregled događaja", "Pregled plaćanja" i "Statistika". U prototipu nije implementiran ovaj deo aplikacije ali je dat kratak pregled mogućnosti koje bi pružao.

Odeljak "Pregled korisnika" operaterima i administratorima pruža uvid u sve korisnike koje dele sa sistemom ICC-a. Za svakog korisnika postoji mogućnost brisanja njegovog profila iz sistema ICC-a u slučaju da korisnik odluči da više ne želi da bude član. Dostupno je i polje za dodavanje novog korisnika, u okviru kog postoji opcija za deljenje njegovog profila sa sistemom ICC-a.

Odeljak "Pregled događaja" pruža uvid u sve događaje koji su vezani za konkretnu ustanovu kulture. Na ovoj stranici postoji opcija za dodavanje novog događaja a postojeće događaje je moguće obrisati i izmeniti. Događaji se mogu podeliti sa sistemom ICC-a.

Odeljak "Pregled plaćanja" pruža mogućnost pregleda svih kupovina karata za događaje koje organizuje konkretna ustanova kulture. Postoji i opcija za kupovinu karte za korisnika koji to želi uživo da završi.

Odeljak "Statistika" pruža pregled statističkih podataka vezanih isključivo za poslovanje konkretne ustanove kulture. Administratori u svakom trenutku mogu izabrati vrstu podataka koju žele da vide. U ponudi je statistika o aktivnosti korisnika, kupljenim kartama, prijavama za učešće na događajima i slično.



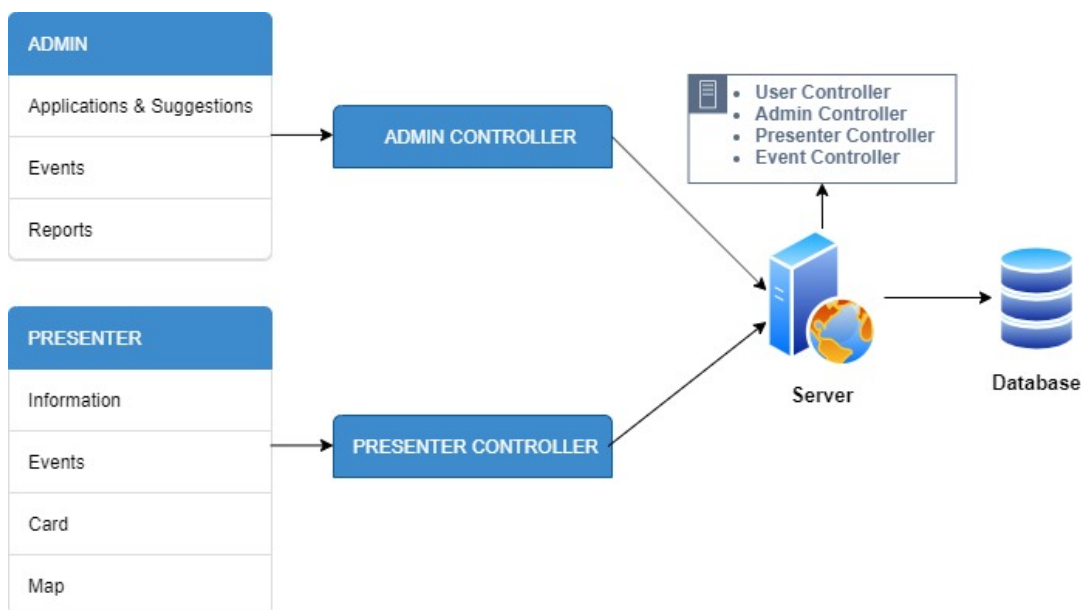
## 4. Implementacija

U ovoj glavi je opisan prototip, detalji implementacije i serverskog i klijentskog dela, ali je pre svega objašnjena arhitektura same aplikacije. Pri opisivanju arhitekture posebno je posvećena pažnja korišćenim tehnologijama.

### 4.1 Arhitektura aplikacije

Arhitektura aplikacije je predviđena da bude troslojna. Sastoji se iz prezentacionog sloja, središnjeg sloja i sloja podataka (slika 4.1).

1. Prezentacioni sloj čini korisnički interfejs izgrađen u veb okviru *Angular 8*.
2. Središnji sloj, izgrađen u veb okviru *Spring Boot*, predstavlja sponu između prezentacionog sloja i sloja podataka. Veza se ostvaruje preko klijentskih i serverskih kontrolera.
3. Sloj podataka, koju čini H2 baza podataka, predstavlja skladište podataka potrebnih za funkcionisanje informacionog sistema.



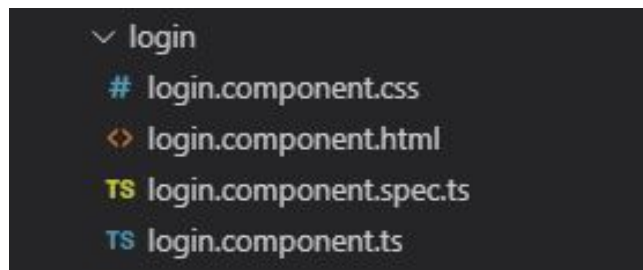
SLIKA 4.1: Arhitektura aplikacije

#### 4.1.1 Korišćene tehnologije prilikom implementacije prezentacionog sloja

Za implementaciju klijentskog dela aplikacije korišćen je programski jezik *TypeScript* [15] (koji je sintaksni nadskup programskog jezika *JavaScript* koji razvija i održava Majkrosoft) i veb okvir (eng. *web framework*) baziran na njemu - *Angular 8* [1]. Za dizajn stranica je korišćena je biblioteka *Angular Material* [2] koja pruža komponente za izgradnju interfejsa veb aplikacije (eng. *UI components*).

Angular se koristi za izradu takozvanih aplikacija sa jednom stranicom (eng. *single page applications*). Ovakve aplikacije učitaju jednu glavnu HTML stranicu i njen

sadržaj menjaju dinamički u zavisnosti od interakcija korisnika. Glavna HTML stranica je *index.html* a njen sadržaj se menja učitavanjem različitih komponenti. Svaka komponenta se sastoji iz četiri fajla: CSS fajla koji HTML kodu komponente dodaje stil, HTML fajla koji predstavlja glavni sadržaj komponente, fajla za *unit* testiranje komponente (*.spec.ts* fajl) i *TypeScript* fajla koji sadrži glavnu funkcionalnost komponente (slika 4.2).



SLIKA 4.2: Struktura komponente za prijavljivanje na sistem generisane pomoću *Angular CLI*-a

Da bi se tri glavna fajla (CSS, HTML i TS) međusobno povezali, dovoljno je u *TypeScript* fajlu naglasiti odakle komponenta dobija stilove i gde se nalazi njena glavna struktura (programski kod 4.1).

```
1 @Component({
2   selector: 'app-login',
3   templateUrl: './login.component.html',
4   styleUrls: ['./login.component.css']
5 })
```

PROGRAMSKI KOD 4.1: Povezivanje glavnih fajlova jedne komponente

Odabir prave komponente koju je potrebno učitati nakon akcije korisnika *Angular* rešava pomoću rutiranja. Za HTML komponente koje služe za izvršavanje akcija kao što su dugme ili link se vezuje ruta na koju treba otići. Potrebno je svaku komponentu povezati sa putanjom u URI-u kao što je prikazano u kodu 4.2.

```
1 const routes: Routes = [
2   { path: 'login', component: LoginComponent },
3   { path: 'register', component: RegisterComponent },
4   {
5     path: 'dashboard',
6     component: DashboardComponent,
7     canActivate: [RoleGuardService],
8     data: {
9       expectedRole: 'admin'
10    }
11  },
12  {
13    path: 'profile',
14    component: ProfileComponent,
15    canActivate: [AuthGuardService]
16  }
17 ];
```

PROGRAMSKI KOD 4.2: Rutiranje

Učitavanje nekih komponenti je potrebno onemogućiti određenim korisnicima. Na primer, običan korisnik ne sme da učita glavnu stranicu administratora ili neprijavljen korisnik ne sme pristupiti svom profilu. Za to se koriste "čuvari"(eng. *guards*), kao što

se vidi u kodu 4.2. Postoje dve glavne vrste čuvara: čuvari po ulozi (eng. *role guards*) i čuvari po autorizaciji (eng. *authorization guards*). Čuvari po ulozi brane korisnicima sa određenim ulogama pristup nekim komponentama a čuvari po autorizaciji brane neprijavljenim korisnicima pristup nekim komponentama.

Funkcionalnosti koje deli veći broj komponenti se organizuju u servise (eng. *service*). Primer ovakvog servisa je HTTP servis koji pruža interfejs za komunikaciju sa serverskim delom aplikacije. Servis je potrebno deklarirati kao "provajdera" (eng. *provider*) što omogućava njegovo umetanje u komponente (eng. *dependency injection*).

#### 4.1.2 Korišćene tehnologije prilikom implementacije središnjeg sloja

Za implementaciju serverskog dela aplikacije korišćen je programski jezik Java i veb okvir (eng. *web framework*) *Spring Boot* [14]. Korišćena je baza podataka H2 o kojoj će biti više reči u odeljku 4.1.3.

*Spring Boot* je veb okvir koji omogućava lako pravljenje *Spring* aplikacija i njihovo puštanje u rad. Sva podešavanja koja bi bilo potrebno ručno uraditi pri izgradnji *Spring* aplikacije, *Spring Boot* pravi automatski na osnovu zavisnosti (eng. *dependencies*) koje su dodate u projekat. *Spring Boot* podržava i grupe zavisnosti koje se često koriste u praksi, radi lakšeg održavanja koda. Na primer, programski kod 4.3 korišćen pri implementaciji aplikacije prikazuje zavisnost koja se koristi radi pisanja *REST* krajnjih tačaka (eng. *endpoints*).

```

1 <dependency>
2   <groupId>org.springframework.boot</groupId>
3   <artifactId>spring-boot-starter-web</artifactId>
4 </dependency>

```

PROGRAMSKI KOD 4.3: *Maven* zavisnost za podešavanje *REST* krajnjih tačaka aplikacije

Ulazna tačka *Spring Boot* aplikacije je klasa koja sadrži anotaciju *SpringBootApplication* i u kojoj se nalazi *main* metod. Ova klasa je prikazana programskim kodom 4.4. Anotacija *EnableAsync* nije ključna u ulaznoj tački *Spring Boot* aplikacije, već je dodata radi izvršavanja asinhronih metoda.

```

1 import org.springframework.boot.SpringApplication;
2 import org.springframework.boot.autoconfigure.SpringBootApplication;
3 import org.springframework.scheduling.annotation.EnableAsync;
4
5 @SpringBootApplication
6 @EnableAsync
7 public class CultureApplication {
8     public static void main(String[] args) {
9         SpringApplication.run(CultureApplication.class, args);
10    }
11 }

```

PROGRAMSKI KOD 4.4: Ulazna tačka *Spring Boot* aplikacije

Za dodatna podešavanja kao što su veza sa bazom, port na kom aplikacija sluša, host za slanje imejl-ova kao i čuvanje nekih opštih vrednosti koristi se fajl *application.properties*.

*Spring Initializr* je onlajn alat koji omogućava jednostavno generisanje praznog *Spring Boot* projekta - moguće je izabrati da li će projekat biti zasnovan na *Maven* ili *Gradle* automatizacionom alatu, programski jezik u kom će projekat biti pisan, potrebne zavisnosti kao i metapodatke projekta.

### 4.1.3 Korišćene tehnologije prilikom implementacije sloja podataka

Kao skladište podataka korišćena u toku razvoja aplikacije je baza podataka H2 [5]. H2 je sistem za upravljanje relacionim bazama podataka (eng. *RDBMS - relational database management system*) pisan u Javi. Omogućava pravljenje *in-memory* tabela - tabela koje se čuvaju u RAM memoriji računara. Ova osobina znači da je pristup podacima veoma brz. Sa druge strane, u slučaju da mašina na kojoj se baza podataka nalazi padne ili se restartuje svi podaci se gube.

Pomoću H2 je moguće čuvati podatke i na disku. Može raditi u više modova: lokalnom (eng. *embedded*) modu, serverskom modu i mešanom modu. Lokalni mod podrazumeva povezivanje na lokalnu bazu podataka korišćenjem JDBC-a. Serverski mod podrazumeva povezivanje sa udaljenom bazom podataka korišćenjem JDBC-a preko TCP/IP. Mešani mod podrazumeva i lokalne i udaljene konekcije u isto vreme.

Na primer, ukoliko je potrebno povezati *Spring Boot* aplikaciju sa *in-memory* bazom podataka, dovoljno je liniju 4.5 upisati u *application.properties* fajl.

```
1 spring.datasource.url = jdbc:h2:mem:culture
```

PROGRAMSKI KOD 4.5: Konekcija na *in-memory* bazu podataka

Ukoliko je potrebno povezati se na lokalnu bazu podataka, dovoljno je izmeniti liniju 4.5 linijom 4.6, pri čemu putanja do baze mora biti apsolutna.

```
1 spring.datasource.url = jdbc:h2:file:/data/culture
```

PROGRAMSKI KOD 4.6: Konekcija na lokalnu bazu podataka

*Spring Boot* pruža interfejs za pregled baze podataka kao i za izvršavanje upita. Podrazumevano nije omogućeno koristiti ovaj interfejs, te je potrebno u *application.properties* fajl uneti liniju 4.7. Nakon pokretanja aplikacije, dovoljno je u pretraživaču otvoriti <http://localhost:8080/h2-console>. Biće predstavljena stranica za prijavljivanje na kojoj je potrebno uneti korisničko ime i šifru koji su postavljeni u *application.properties* fajlu (slika 4.3).

```
1 spring.h2.console.enabled = true
```

PROGRAMSKI KOD 4.7: Omogućavanje korišćenja interfejsa za H2 bazu podataka

Nakon prijavljivanja, prikazuje se interfejs sa slike 4.4. Prikazane su sve postojeće tabele kao i opisi njihovih kolona, polje za izvršavanje upita nad bazom podataka i polje za ispis rezultata upita. Nakon izvršavanja upita i dobijanja rezultata, odabirom opcije "Edit" moguće je promeniti redove tabele (umesto pokretanja upita za ažuriranje tabele).

Nakon završetka implementacije informacionog sistema, kao sistem za upravljanje stalnim skladištem podataka korišćen je sistem PostgreSQL [9]. Za upravljanje podacima i pregled podataka korišćena je ugrađena podrška za rad sa bazom podataka okruženja za razvoj *IntelliJ IDEA*.

U okviru okruženja *IntelliJ IDEA* je moguće izabrati opciju da se pored strukture projekta i koda prikazuje i okruženje za rad sa bazom podataka. Odabirom te opcije se prikazuje interfejs sa slike 4.5. Sa leve strane se nalazi pregled baze podataka sa detaljnijim opisima tabela, kao što su spisak kolona i njihovi tipovi, i sekvence koje se koriste za automatsko generisanje identifikatora tabela. U okviru editora je moguće izvršavati potrebne PostgreSQL upite.

Moguće je dodati proizvoljan broj konekcija prema raznovrsnim aktuelnim bazama podataka, kao što su Azure, DB2, H2, MySQL i PostgreSQL. Prozor za dodavanje konekcije je prikazan na slici 4.6. U slučaju da fale drajveri za povezivanje sa izabranom bazom, okruženje nudi opciju za njihovo automatsko ažuriranje.

SLIKA 4.3: H2 interfejs - stranica za prijavljivanje

SLIKA 4.4: H2 interfejs - pregled tabela i izvršavanje upita

Potrebno je izmeniti *application.properties* fajl kako bi *SpringBoot* znao na koju se bazu podataka povezuje. Programski kod 4.8 prikazuje sve neophodne podatke za ostvarivanje veze.

```

1 spring.datasource.platform=postgres
2 spring.datasource.url=jdbc:postgresql://localhost:5433/postgres
3 spring.datasource.username=postgres
4 spring.datasource.password=1234

```

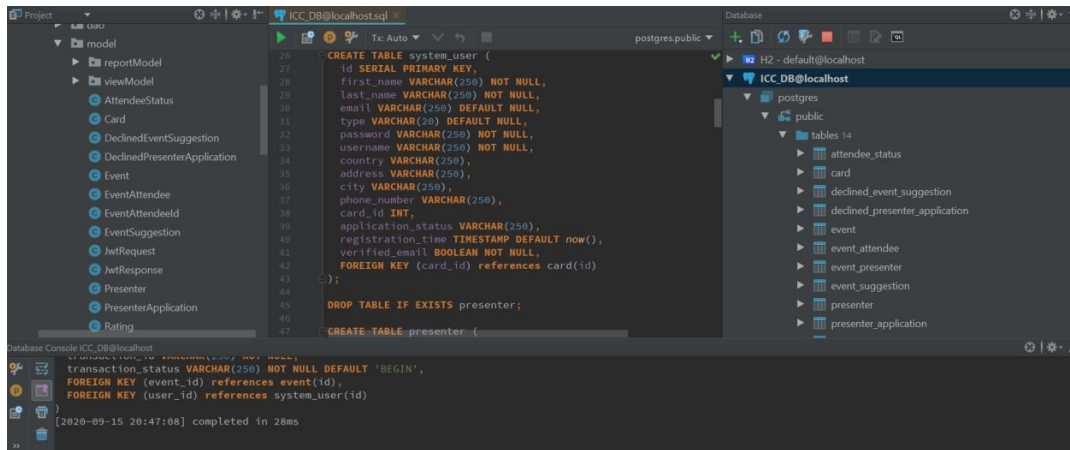
PROGRAMSKI KOD 4.8: Omogućavanje povezivanja sa PostgreSQL bazom

Neophodno je dodati i odgovarajuću zavisnost za rad sa PostgreSQL bazom podataka (programski kod 4.9).

```

1 <dependency>
2   <groupId>org.postgresql</groupId>
3   <artifactId>postgresql</artifactId>
4   <scope>runtime</scope>

```



SLIKA 4.5: Interfejs unutar okruženja IntelliJ IDEA za rad sa bazom podataka - pregled tabela i izvršavanje upita

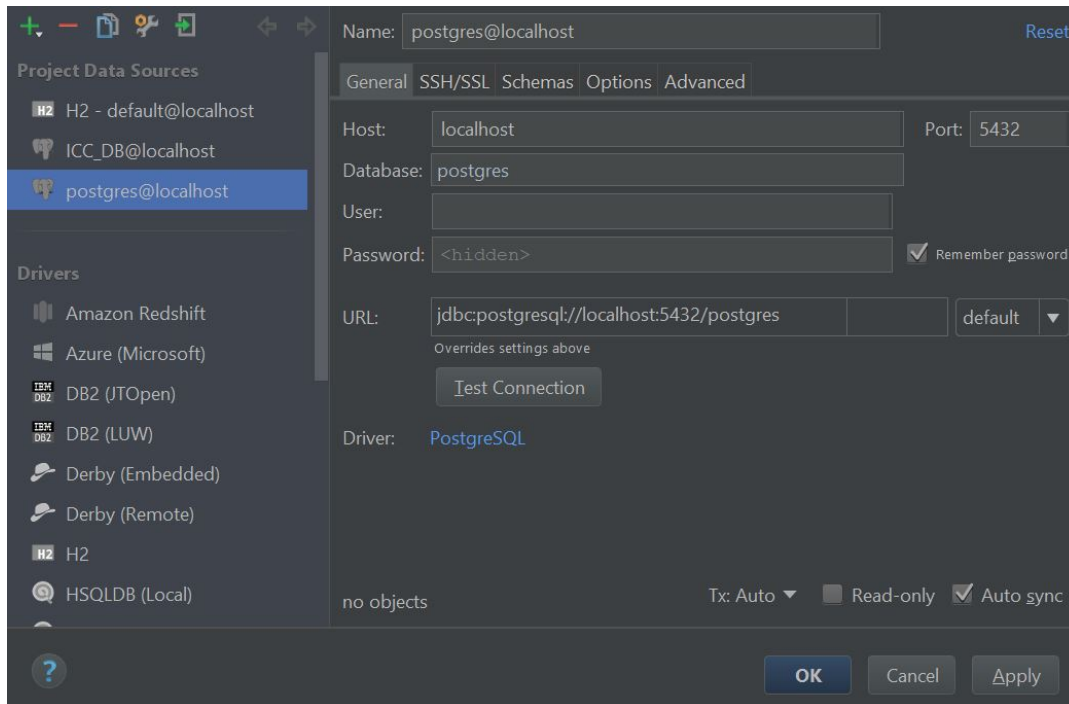
5 </dependency>

PROGRAMSKI KOD 4.9: *Maven* zavisnost za rad sa PostgreSQL bazom podataka

## 4.2 Serverski deo aplikacije

### 4.2.1 Detalji implementacije

Serverski deo aplikacije je izgrađen kao skup kontrolera, servisa, modela i repozitorijuma, uz dodatne konfiguracione i pomoćne fajlove (slika 4.7).



SLIKA 4.6: Interfejs unutar okruženja IntelliJ IDEA za rad sa bazom podataka - dodavanje i konektovanje na bazu podataka

Kontroleri primaju zahteve od klijenta i pozivaju odgovarajuće servise da taj zahtev obrade (programski kod 4.10). Da bi klasa bila prepoznata od strane *Spring Boot*-a, mora biti označena anotacijom *@RestController*. Anotacija *@CrossOrigin* dozvoljava da veb aplikacija pokrenute na jednom domenu dobije pristup podacima aplikacije pokrenute na drugom domenu.

Anotacija *@Autowired* predstavlja jednostavan način uvođenja zavisnosti (eng. *dependency injection*). Nije potrebno pozivati konstruktor klase koju je potrebno koristiti već *Spring Boot* to u pozadini uradi sam.

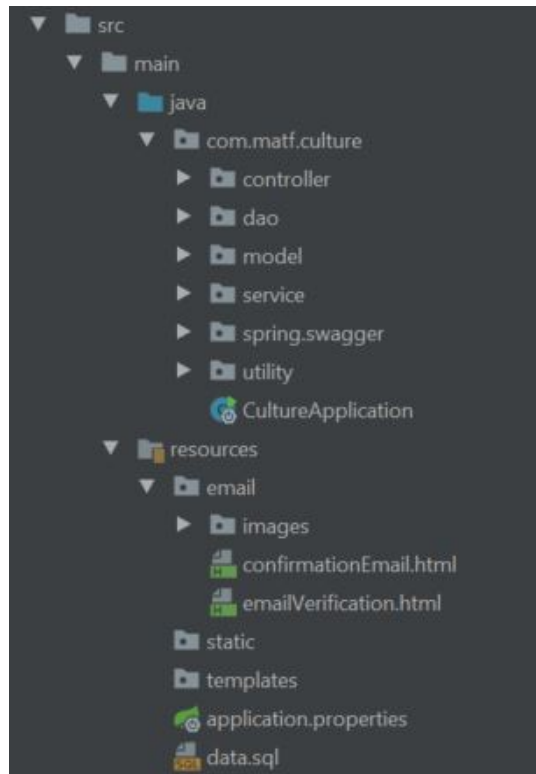
*Spring Boot* podržava osnovne tipove HTTP zahteva - *GET* za dohvaćanje podataka, *POST* za čuvanje (kreiranje) novih podataka, *PUT* za menjanje postojećih podataka i *DELETE* za brisanje podataka. Krajnjim tačkama klijent pristupa menjanjem putanja u URI-u. U okviru putanje je moguće dodati i promenljive (eng. *path variable*), koje se iz putanje čitaju pomoću anotacije *@PathVariable*. Telo zahteva u JSON formatu se čita pomoću anotacije *@RequestBody* i pritom je potrebno napraviti klasu u okviru modela pogleda (eng. *view model*) koja će odgovarati JSON-u koji stiže od klijenta. Programski kod 4.10 prikazuje navedene odlike kontrolera na primeru kontrolera za rad sa događajima.

```

1 @RestController
2 @CrossOrigin
3 public class EventController {
4     @Autowired
5     private EventService eventService;
6
7     @GetMapping("/getAllEvents")
8     public ReturnJson getAllEvents() {
9         return eventService.getAllEvents();
10    }
11
12    @PostMapping("/createEvent")

```





SLIKA 4.7: Organizacija serverske aplikacije

```

13     public ReturnJson createEvent(@RequestBody EventModel event) {
14         return eventService.createEvent(event);
15     }
16
17
18     @PutMapping("/updateEvent/{eventId}")
19     public ReturnJson updateEvent(@PathVariable Long eventId,
20     @RequestBody EventModel event) {
21         return eventService.updateEvent(eventId, event);
22     }
23
24     @DeleteMapping("/cancelEvent/{eventId}")
25     public ReturnJson cancelEvent(@PathVariable Long eventId) {
26         return eventService.cancelEvent(eventId);
27     }

```

PROGRAMSKI KOD 4.10: Primer kontrolera za rad sa događajima

Formu prikazanu programskim kodom 4.10 prate svi implementirani kontroleri. Nakon obrade podataka, klijentu se vraća objekat tipa *ReturnJson*. *ReturnJson* je klasa posebno implementirana tako da sadrži podatak o uspešnosti zahteva, rezultujući objekat, poruku o grešci i kod greške, ako greška postoji (programski kod 4.11). *ReturnJson* je univerzalni odgovor svih implementiranih kontrolera. Pri vraćanju odgovora iz kontrolera, objekat se serijalizuje u JSON format.

```

1 public class ReturnJson {
2     private Object object;
3     private Boolean result;
4     private String message;
5     private String code;

```



```

6
7     public ReturnJson(Object object, Boolean result, String message,
8     String code) {
9         this.object = object;
10        this.result = result;
11        this.message = message;
12        this.code = code;
13    }
14    //getters & setters
15 }

```

PROGRAMSKI KOD 4.11: Klasa za univerzalni odgovor na HTTP zahteve

Kako bi komunikacija sa bazom podataka bila moguća, *Spring Boot* zahteva postojanje repozitorijuma i entiteta. Repozitorijum je interfejs, što znači da njegove metode neće biti eksplicitno implementirane. Svaki repozitorijum je obeležen anotacijom `@Repository` i nasljeđuje *Spring Boot*-ov generički interfejs `JpaRepository<T,G>`. U imenu ovog interfejsa, *T* predstavlja entitet čiji je interfejs repozitorijum a *G* je tip ID-a entiteta. *JPA* je skraćenica za *Java Persistence API* [7]. To je API koji pruža skup neophodnih metoda za rad sa bazom podataka i skup pravila za pisanje proizvoljnih metoda. Metodi prate princip imenovanja oblika `findByColumnName(ColumnType value)`. Ekvivalent ovome je SQL upit prikazan programskim kodom 4.12.

```

1 SELECT *
2 FROM table_name
3 WHERE column_name = value

```

PROGRAMSKI KOD 4.12: SQL upit

U imenu metode se mogu nadovezivati pretrage po više kolona korišćenjem *AND* i *OR* ključnih reči, u slučaju pretrage po intervalu može se koristiti ključna reč *BETWEEN* i slično. Primer jednog repozitorijuma je prikazan programskim kodom 4.13.

```

1 @Repository
2 public interface CardRepository extends JpaRepository<Card, Long> {
3     Card findByCardNumber(Long cardNumber);
4
5     List<Card> findByIsActive(boolean isActive);
6 }

```

PROGRAMSKI KOD 4.13: Repozitorijum za kartice

Nije uvek moguće na ovaj način izraziti složenije upite. Za te potrebe, moguće je koristiti anotaciju `@Query` uz metodu repozitorijuma. Ova anotacija omogućava pisanje proizvoljnih SQL upita nad bazom podataka. Primer jednog upisa u bazu se može videti u programskom kodu 4.14.

```

1 @Repository
2 public interface PresenterRepository extends JpaRepository<Presenter,
3     Long> {
4
5     @Modifying
6     @Query(value="insert into presenters (user_id, speciality) values (:
7     userId, :speciality)", nativeQuery = true)
8     void createNewPresenter(@Param("userId") Long userId, @Param("
9     speciality") String speciality);
10 }

```

PROGRAMSKI KOD 4.14: Repozitorijum za predavače

Prepoznate entitete baze podataka je potrebno predstaviti i u Java kodu. Jedan entitet je predstavljen jednom klasom, koja je označena anotacijama `@Entity` i `@Table`, koji vezuje klasu za ime tabele u bazi. Svaki entitet ima svoj identifikator koji se označava anotacijom `@Id` a uz nju se može dodati anotacija `@GeneratedValue` koja *Spring Boot*-u govori na koji način da automatski generiše identifikator entiteta. Ostala polja, atributi, entiteta mogu ali ne moraju biti označena anotacijom `@Column`. Svaki entitet mora imati prazan konstruktor, kao i metode za postavljanje i menjanje atributa. Primer jednog entiteta prikazan je programskim kodom 4.15.

```

1 @Entity
2 @Table(name = "events")
3 public class Event {
4     @Id
5     @GeneratedValue(strategy = GenerationType.IDENTITY)
6     private Long id;
7
8     private String name;
9     private String city;
10    private String country;
11    private String venue;
12    private Timestamp dateTime;
13    private String description;
14
15    @ManyToMany(fetch = FetchType.EAGER)
16    @JoinTable(
17        name = "events_presenters",
18        joinColumns = @JoinColumn(name = "event_id"),
19        inverseJoinColumns = @JoinColumn(name = "presenter_id"))
20    @JsonIgnoreProperties({"events", "eventSuggestions", "
declinedEventSuggestions"})
21    private Set<Presenter> presenters = new HashSet<>();
22
23    @OneToMany(
24        mappedBy = "event",
25        cascade = CascadeType.ALL
26    )
27    @JsonIgnoreProperties({"event", "user"})
28    private List<EventAttendee> attendeeList = new ArrayList<>();
29
30    @OneToMany(
31        mappedBy = "event",
32        cascade = CascadeType.ALL
33    )
34    @JsonIgnoreProperties({"event", "presenter"})
35    private List<Rating> ratings = new ArrayList<>();
36
37    public Event() {}
38
39    // getters & setters

```

PROGRAMSKI KOD 4.15: Entitet koji predstavlja događaj

Dodatne anotacije koje se mogu primetiti u programskom kodu 4.15 su:

- `@OneToMany` koja predstavlja vezu sa drugim entitetom “jedan prema više”. U slučaju entiteta Događaj, jedan događaj može imati više prijavljenih korisnika dok jedan prijavljeni korisnik može biti prijavljen samo za jedan događaj.
- `@ManyToMany` koja predstavlja vezu “više prema više”. U ovom slučaju, jedan događaj može biti organizovan od strane više predavača a jedan predavač može organizovati mnogo događaja.

- `@JoinTable` koja predstavlja veznu tabelu koja nastaje u slučaju odnosa “više prema više”.
- `@JsonIgnoreProperties` koja prilikom serijalizacije objekta u JSON format skriva obeleženo polje. Ova anotacija je korišćena kako ne bi došlo do rekurzivnog ispisa podataka u rezultujućem JSON-u.

## 4.3 Klijentski deo aplikacije

### 4.3.1 Detalji implementacije

Glavna stranica svake *Angular* aplikacije je *index.html*. Ona ima formu standardne HTML stranice: sadrži tagove *html*, *head*, *body*. U telu ove html stranice se nalazi tag *app-root*. Ovaj tag nije tag propisan HTML standardom, već je namenski tag - selektor (eng. *selector*). Kao što se vidi sa programskog koda 4.1, svaka komponenta sadrži i selektor. Ovaj kratak string omogućava da se bilo gde u kodu može iskoristiti tag tog imena i na tom mestu umetnuti HTML komponenta kao i njene funkcionalnosti.

Tag *app-root* odgovara komponenti *app.component.html* (programski kod 4.16). Na vrhu se nalazi tag *app-layout* koji služi za postavljanje mreže (eng. *grid*) za komponente stranice. Tag *mat-sidenav-container* je komponenta *Angular Material*-a koja služi za izgradnju prilagodljivog (eng. *responsive*) menija. U slučaju da je ekran na kome je otvorena aplikacija dovoljno mali, meni će se prikazati sa leve strane. Lista linkova koja se prikazuje u tom meniju se umeće iz komponente *sidenav-list*. U slučaju da je ekran na kome je otvorena aplikacija dovoljno veliki, prikazuje se standardni interfejs i meni se umeće iz komponente *header*. Tag *router-outlet* je *Angular*-ov tag umesto kog se umeću komponente kojima se pristupa pomoću rutera.

```

1 <app-layout>
2   <mat-sidenav-container>
3     <mat-sidenav #sidenav role="navigation">
4       <app-sidenav-list (sidenavClose)="sidenav.close()"></app-sidenav-
      list>
5     </mat-sidenav>
6     <mat-sidenav-content>
7       <app-header (sidenavToggle)="sidenav.toggle()"></app-header>
8       <main>
9         <router-outlet></router-outlet>
10      </main>
11     </mat-sidenav-content>
12   </mat-sidenav-container>
13 </app-layout>

```

PROGRAMSKI KOD 4.16: Skrivaјуći meni

Ruta koja se podrazumevano otvara prilikom pristupa aplikaciji je */register*. Programski kod 4.17 prikazuje kontroler ove komponente a programski kod 4.18 prikazuje HTML strukturu komponente. Na vrhu svakog kontrolera se nalaze uvozi (eng. *imports*) klasa i funkcija koje se nalaze u eksternim fajlovima. Svaki kontroler može imati klasne promenljive (kako je sam klasa), mora imati konstruktor (preko koga se vrši umetanje (eng. *dependency injection*)) kao i klasne funkcije. Konstruktor se poziva pri instancijaciji klase i poziva ga JavaScript izvršilac (eng. *engine*). Funkciju *ngOnInit* poziva sam *Angular* kada završi sa inicijalizacijom komponente, dakle nakon konstruktora.

```

1 import { Component, OnInit } from '@angular/core';
2 import { ReturnJson } from '../models/return-json.model';
3 import { passwordValidator } from '../validators/password-validator';

```

```

4 import { FormGroup, FormControl, Validators } from '@angular/forms';
5 import { HttpService } from '../services/http-service.service';
6 import { Router } from "@angular/router";
7
8 @Component({
9   selector: 'app-register',
10  templateUrl: './register.component.html',
11  styleUrls: ['./register.component.css']
12 })
13 export class RegisterComponent implements OnInit {
14
15   public loading: boolean;
16   public form: FormGroup;
17   public error: string | null;
18
19   constructor(private httpService: HttpService,
20               private router: Router) {
21     this.loading = false;
22
23     this.form = this.createFormGroup();
24   }
25
26   ngOnInit() {
27   }
28
29   .....
30 }

```

PROGRAMSKI KOD 4.17: Osnovna struktura kontrolera komponente za registraciju

Glavna karakteristika *Angular*-a je dvosmerno povezivanje podataka (eng. *two-way data binding*). Dvosmerno povezivanje podataka omogućava da se podatak iz kontrolera veže za HTML kod. Postoje tri moguća načina vezivanja podataka:

1. Prikazivanje vezanog podatka u HTML kodu korišćenjem vitičastih zagrada. U kodu 4.18 se može videti da se vrednost promenljive *error* ispisuje na ovaj način (linija 37).
2. Prosleđivanje vezanih podataka kao atributa HTML tagova korišćenjem uglastih zagrada. U kodu 4.18 se mogu uočiti mnoge primene ovog vezivanja. Na primer, dugme za registraciju je onesposobljeno u zavisnosti od toga da li je forma *form* ispravno popunjena (linija 42). Ispravnost popunjenosti forme se proverava u kontroleru.
3. Vezivanje događaja korišćenjem običnih zagrada. Ovo vezivanje se koristi ukoliko je potrebno reagovati na akcije korisnika. U kodu 4.18 primer ovog vezivanja bi bilo dugme za registraciju (linija 42). Kod *(click)="submit()"* označava da će se funkcija kontrolera pod imenom *submit()* pozvati kada korisnik pritisne dugme za registraciju.

Dakle, dvosmerno vezivanje omogućava praćenje izmena podataka i od strane korisnika i od strane kontrolera kao i reagovanje na akcije korisnika.

*Angular* pruža mogućnost korišćenja strukturnih direktiva. Strukturne direktive manipulišu DOM strukturom tako što dodaju ili uklanjaju elemente. Karakteriše ih *\** ispred imena. Primer direktive u kodu 4.18 je direktiva *\*ngIf* koja element prikazuje ili uklanja (dakle, element ili postoji ili ne postoji, nije samo skriven ili prikazan) u zavisnosti od uslova.

```

1 <mat-card>
2   <mat-card-title>Register</mat-card-title>
3   <mat-card-content>
4     <form [formGroup]="form">
5       <p>
6         <mat-form-field>
7           <input type="text" matInput placeholder="Email"
8             FormControlName="email">
9         </mat-form-field>
10      </p>
11     <p>
12       <mat-form-field>
13         <input type="text" matInput placeholder="Username"
14           FormControlName="username">
15       </mat-form-field>
16     </p>
17     <p>
18       <mat-form-field>
19         <input type="text" matInput placeholder="First name"
20           FormControlName="firstName">
21       </mat-form-field>
22     </p>
23     <p>
24       <mat-form-field>
25         <input type="text" matInput placeholder="Last name"
26           FormControlName="lastName">
27       </mat-form-field>
28     </p>
29     <p>
30       <mat-form-field>
31         <input type="password" matInput placeholder="Password"
32           FormControlName="password">
33       </mat-form-field>
34     </p>
35     <p *ngIf="(email.dirty || email.touched) && email.invalid &&
36       email.errors.email" class="error">
37       Please enter a valid email address
38     </p>
39     <p *ngIf="(password.dirty || password.touched) && password.
40       invalid && password.errors.invalidPassword" class="error">
41       Password is too weak! You need at least one uppercase and
42       lowercase letter and one number, with a length over 8 characters.
43     </p>
44     <p *ngIf="error" class="error">
45       {{ error }}
46     </p>
47   </form>
48 </mat-card-content>
49 <mat-card-actions class="button">
50   <button mat-raised-button (click)="submit()" color="primary" [
51     disabled]="form.invalid">Register</button>
52 </mat-card-actions>
53 <mat-card-content>
54   <button mat-button color="primary" routerLink="/login">Already
55   have an account? Go ahead and login!</button>
56 </mat-card-content>
57 </mat-card>
58 <div class="progress-loader" [hidden]="!loading">
59 <mat-progress-spinner [mode]=" 'indeterminate ' "></mat-progress-spinner>
60 </div>

```

PROGRAMSKI KOD 4.18: HTML struktura komponente za registraciju

Još jedna često korišćena direktiva je *\*ngFor* koja prolazi kroz kolekciju koja postoji u kontroleru i ispisuje njene elemente. Primer upotrebe direktive *\*ngFor* je dat u kodu 4.19. Direktiva se koristi radi prikazivanja svih ocena koje je predavač dobio tako što prolazi kroz listu dobijenih ocena.

```

1 <mat-card ...>
2   <mat-card-header>
3     <mat-card-title>Ratings</mat-card-title>
4   </mat-card-header>
5   <mat-card-content>
6     <mat-divider></mat-divider>
7     <mat-list>
8       <mat-list-item *ngFor="let rating of user.receivedRatings" ...>
9         <p matLine>
10          <star-rating [starType]=' 'icon' " [rating]="rating.rating" [
showHalfStars]="true"></star-rating>
11          </p>
12          <h3 matLine class="list-item"> {{rating.user.firstName + ' ' +
rating.user.lastName}} </h3>
13          <p matLine>
14            <span class="list-item"> {{rating.comment}} </span>
15          </p>
16        </mat-list-item>
17      </mat-list>
18    </mat-card-content>
19 </mat-card>

```

PROGRAMSKI KOD 4.19: Korišćenje direktive *\*ngFor* radi prikazivanja svih ocena koje je predavač dobio

Svi kontroleri su implementirani tako da sadrže formu prikazanu u programskom kodu 4.17. Još neke dodatne odlike kontrolera koje ne implementiraju svi kontroleri su:

1. Funkcija *ngAfterViewInit* koja inicijalizuje vrednosti promenljivih koje postaju validne tek nakon inicijalizacije HTML koda. Na primer, inicijalizuju se dekoratori polja (eng. *property decorators*) kao što je *@ViewChild* koji omogućava rad sa DOM elementima (programski kod 4.20 opisuje inicijalizaciju straničenja (eng. *pagination*) i sortiranja tabele).
2. Funckije za otvaranje dijaloga. Dijalog je pop-ap prozor u kom se učitava sadržaj neke komponente, kome se mogu proslediti podaci i kontrolera komponente koja ga otvara i koji, po zatvaranju, može vratiti podatke kontroleru komponente koja ga je otvorila (programski kod 4.21).

```

1 @ViewChild(MatPaginator) paginator: QueryList<MatPaginator>;
2 @ViewChild(MatSort) sort: QueryList<MatSort>;
3
4 .....
5
6 ngAfterViewInit() {
7   this.currentEventData.paginator = this.paginator.toArray()[0];
8   this.currentEventData.sort = this.sort.toArray()[0];
9   this.pastEventData.paginator = this.paginator.toArray()[1];
10  this.pastEventData.sort = this.sort.toArray()[1];
11 }

```

PROGRAMSKI KOD 4.20: Inicijalizacija dekoratora polja

```

1 openEditEvent(event: EventModel) {
2   const dialogRef = this.dialog.open(

```

```
3     EventEditDialogComponent ,
4     {data: {event: event},
5     disableClose: false ,
6     autoFocus: true})
7   .afterClosed()
8   .subscribe(result => {
9     if (result.data) {
10      if (event === null)
11        this.createEvent(result.data);
12      else
13        this.updateEvent(result.data, event.id);
14    }
15  });
16 }
```

PROGRAMSKI KOD 4.21: Funkcija za rad sa dijalogom

## 5. Zaključak

U radu su opisani idejni projekat i implementacija dela informacionog sistema *ICC* koji bi trebalo da omogući povezivanje kulturnih centara sveta radi lakšeg upoznavanja sa znamenitostima svih povezanih zemalja kao i aktuelnim događajima u njima.

Sistem omogućava korisnicima da pronađu informacije o događajima i ustanovama kulture u zemlji u kojoj se nalaze, dođu do ulaznica po povoljnim cenama a i da ostvaruju bonuse i dobijaju poklone za svoje interesovanje.

Predavači imaju sve opcije koje ima i običan korisnik, s tim da dodatno mogu da organizuju svoje događaje i prate prijavljene korisnike. Korisnici koji su prisustvovali događaju mogu da ocene predavača.

Administrator ima mogućnost da prati sve što se dešava na sistemu. Ima opciju da organizuje događaj, s tim da se to posmatra kao događaj kulturnog centra, i da odredi predavače za njega. Ima uvid u sve prijave za poziciju predavača, prijave za izdavanje članske kartice i predloge događaja a uz to ima i opcije da ih prihvati ili odbije.

Izrađena implementacija nije potpuna, već predstavlja prototip. Ključna stvar koja nedostaje je mogućnost korišćenja kartice. Prvi korak bi bio omogućavanje korišćenja kartice unutar kulturnog centra - dakle za kupovinu ulaznica za događaje koje centar i registrovani predavači organizuju. Zatim bi bilo potrebno implementirati mogućnost kupovine ulaznica za događaje koje organizuju ustanove kulture sa kojima kulturni centar saraduje. Ovo bi moglo biti izvedeno odvođenjem korisnika na stranicu ustanove kulture za kupovinu ulaznica, pri čemu bi oni omogućili unošenje broja kartice korisnika kulturnog centra kako bi on ostvario popust ili poene.

Svako korišćenje kartice bi korisniku donosilo poene, kako bi se ljudi motivisali da prisustvuju kulturnim događajima. Sistem poena bi bio organizovan tako da korisnik može da iskoristi određen broj poena kako bi ostvario popust ili čak besplatno dobio ulaznicu. Možda bi mogle da se osmisle i neke "fizičke" nagrade.

Korišćenje kartice je ključno za rad sistema, te je razmatrano odvojeno od ostalih proširenja. Sistem bi vremenom rastao i proširenja koja bi bila razmatrana u budućnosti su sledeća:

- Uvođenje hobi odeljka. Lokalni kulturni centri često organizuju razne radionice u oblastima slikanja, rada sa vunicom (šivenje, štrikanje, heklanje), muzike i mnogim drugim. Hobi odeljak informacionog sistema bi omogućio ljudima da dele svoje tvorevine i uputstva da se iste naprave, da pružaju pomoć i konstruktivne kritike drugima.
- Podržavanje festivala kao tipa događaja. Ono što bi karakterisalo festival je to da na njemu mogu aktivno učestvovati korisnici. Korisnicima bi bilo omogućeno da se prijavljuju za svoje mesto ili štand na festivalu, pri čemu bi mogli da naprave malu reklamu za sebe i to što na festivalu izlažu. Postojala bi mogućnost ostavljanja komentara izlagačima kako bi zainteresovani korisnici mogli dodatno da se raspitaju.
- Podržavanje programa rekreacije kao tipa događaja. Kao i u slučaju običnih događaja, korisnici bi mogli da se prijave za događaj, ali bi dodatno imali uvid u rasporede treninga, pomoćne materijale i mogućnost direktnog kontakta sa trenerom zarad savetovanja.



Upotreba pametnih mobilnih uređaja (eng. *smartphone*) raste iz godine u godinu, te je obavezni sledeći korak implementacija *Android* i *iOS* aplikacija koje bi nudile udobniji pristup svim mogućnostima informacionog centra kao i veb sajt. Kako bi aplikacije bile još privlačnije korisnicima, mogle bi pružati dodatne mogućnosti, kao na primer:

- Korišćenje kartice uživo, skeniranjem njenog koda sa telefona korisnika, čime bi korisnicima bilo omogućeno da ne moraju da podižu ulaznice za događaj na blagajni ustanove gde se događaj održava, već bi interno bilo provereno da li je korisnik - vlasnik kartice - načinio kupovinu ulaznice za događaj na koji pokušava da uđe.
- Postojanje interaktivne mape sa opcijom pregleda trenutnog stanja u saobraćaju kao i sa prikazom najbržeg puta do određene destinacije na više načina - peške, automobilom ili gradskim prevozom.
- Slanje notifikacija (eng. *push notifications*) korisniku kada se u njegovoj blizini pojavi novi događaj ili nova ustanova kulture.

Rastom i širenjem internacionalnog kulturnog centra rašće i usluge koje je potrebno da podržava, čime će se informacioni sistem proširivati. U ovom radu je prikazana osnova informacionog sistema - ono što je neophodno za njegovo funkcionisanje. Osnova je projektovana i implementirana što opštije, kako bi proširivanje sistema u budućnosti bilo lakše. Vodilo se računa o tome da projekat bude detaljno opisan i dokumentovan, a da sam kod bude dobro organizovan i pregledan.

# Literatura

- [1] *Angular*. <https://angular.io/>. [Online; accessed 21-October-2019].
- [2] *Angular Material*. <https://material.angular.io/>. [Online; accessed 21-October-2019].
- [3] *Diagrams.net*. <https://diagrams.net/>. [Online; accessed 31-October-2019].
- [4] *GitHub projekat za alat diagrams.net*. <https://github.com/jgraph/drawio>. [Online; accessed 24-October-2019].
- [5] *H2 Database Engine*. <https://www.h2database.com/html/main.html>. [Online; accessed 24-October-2019].
- [6] *IntelliJ IDEA*. <https://www.jetbrains.com/idea/>. [Online; accessed 24-September-2019].
- [7] *Java Persistence API*. <https://docs.spring.io/spring-data/jpa/docs/current/reference/html>. [Online; accessed 24-October-2019].
- [8] *Microsoft Visio*. <https://www.microsoft.com/en/microsoft-365/visio/flowchart-software>. [Online; accessed 24-October-2019].
- [9] *PostgreSQL*. <https://www.postgresql.org/>. [Online; accessed 24-September-2019].
- [10] *Preuzimanje izvršnih fajlova za alat diagrams.net*. <https://github.com/jgraph/drawio-desktop/releases/tag/v13.2.2>. [Online; accessed 24-October-2019].
- [11] *Preuzimanje izvršnog fajla za razvojno okruženje IntelliJ IDEA*. <https://www.jetbrains.com/idea/download/>. [Online; accessed 24-September-2019].
- [12] *Preuzimanje izvršnog fajla za razvojno okruženje Visual Studio Code*. <https://code.visualstudio.com/download>. [Online; accessed 24-September-2019].
- [13] *Programski jezici i tehnologije koji su podržani u IntelliJ IDEA*. [https://www.jetbrains.com/idea/features/editions\\_comparison\\_matrix.html](https://www.jetbrains.com/idea/features/editions_comparison_matrix.html). [Online; accessed 24-September-2019].
- [14] *Spring Boot*. <https://spring.io/projects/spring-boot>. [Online; accessed 24-October-2019].
- [15] *TypeScript - JavaScript that scales*. <https://www.typescriptlang.org/>. [Online; accessed 21-October-2019].
- [16] *Visual Paradigm*. <https://www.visual-paradigm.com/>. [Online; accessed 24-October-2019].
- [17] *Visual Studio Code*. <https://code.visualstudio.com/>. [Online; accessed 24-September-2019].
- [18] *Visual Studio Code ekstenzije*. <https://marketplace.visualstudio.com/VSCode>. [Online; accessed 24-September-2019].
- [19] *Visual Studio Code GitHub repozitorijum*. <https://github.com/Microsoft/vscode/>. [Online; accessed 24-September-2019].