

2020/2021.

Night algorithm

DOKUMENTACIJA

TIM: Adna Nuspahić

Emin Šarak

Amna Trčalo

SADRŽAJ

Specifikacija	4
Use-case dijagram	7
Scenariji i dijagrami aktivnosti:	
▪ Kreiranje događaja	8
▪ Otkazivanje rezervacije za događaj	11
▪ Poništavanje događaja	14
▪ Prijava na događaj	17
▪ Registracija korisnika	21
▪ Registracija vlasnika objekta	24
▪ Validacija dolaska	27
Korisnički interfejs:	
▪ Neregistrovani korisnik	30
▪ Registracija korisnika	35
▪ Prijava korisnika	38
▪ Lista prijatelja	39
▪ Prijava na događaj	40
▪ Profil korisnika	43
▪ Profil prijatelja	44
▪ Profil registrovanog korisnika	45
▪ Profil vlasnika objekta	46
▪ Registracija događaja	47
▪ Recenzija događaja	49
▪ Uvid u aktuelni događaj	50
▪ Uvid u završeni događaj	51
Dijagram klasa	52
Dijagram objekata	53
MVC dijagram	54
SOLID	55
ERD	57
Paterni:	
▪ Strukturalni paterni	58
▪ Kreacijski paterni	60
▪ Paterni ponašanja	61

Implementacija paterna 63

Dijagrami sekvenci:

- Ažuriranje liste prijatelja 64
- Kreiranje događaja 65
- Poništavanje događaja 66

Dijagrami složene strukture:

- Dijagram komponenti 67
- Dijagram paketa 68
- Dijagram raspoređivanja 69

SPECIFIKACIJA

OPIS TEME:

Aplikacija za rezervaciju aktuelnih događaja u ugostiteljskom objektu. Omogućava kako zakazivanje događaja tako i obavještavanje korisnika o istim i njihovu rezervaciju.

Olakšava organizaciju događaja, njihovo promovisanje i komunikaciju između organizatora i klijenta.

PROCESI:

- Registracija korisnika:

Pri upotrebi aplikacije korisnik se treba registrovati unošenjem svojih ličnih podataka nakon čega dobiva mogućnost korištenja aplikacije ili odabire opciju da poveže svoj račun sa nekom od postojećih društvenih mreža (google, facebook) čime se preskače proces unošenja ličnih podataka. Pri kreiranju lozinke minimalan broj znakova je 8, a također mora sadržavati i mala i velika slova. Također, korisničko ime je jedinstveno što znači da ako korisnik unese već postojeće ime dolazi mu obavijest da je ime već zauzeto i da mora unijeti novo. Korisnik je obavezan navesti starosnu dob i spol. Korisnik također ima opciju upload-a profilne fotografije koju može i naknadno promijeniti nakon čega će doći notifikacija ostalim korisnicima s njegove liste prijatelja da je to učinio. Nakon toga, navodi se lista interesovanja u vidu hashtag-a (koja se naknadno može i modifikovati) na osnovu kojih se filtriraju događaji koji se prikazuju.

Korisnik bez registracije ima samo uvid u aktuelne događaje, bez mogućnosti rezervisanja stola ili uvida koje osobe idu na taj događaj.

- Interakcija s drugim korisnicima:

Mogućnost dodavanja drugih korisnika aplikacije na listu prijatelja pomoću njihovog username-a/e-mail čime se dobiva mogućnost pozivanja prijatelja s liste na događaje te pregled događaja na koji su prijatelji prijavljeni. Korisnik također dobiva notifikaciju kada se prijatelj s njegove liste prijavi na neki događaj. Korisnik ima mogućnost da prijavi (report) druge korisnike u slučaju zloupotrebe podataka ili neprimjerenog sadržaja. Korisnik također može obrisati prijatelja s njegove liste.

- Registracija vlasnika objekta:

Vlasnik objekta dobiva posebne login podatke od strane administratora čime postaje ovlašten za kreiranje događaja i njihovo brisanje. Vlasnik objekta ima uvid u broj prijavljenih osoba i rezervacije stolova za svaki događaj. Nakon završetka događaja, vlasnik ima uvid u recenzije koje su korisnici ostavili za događaj.

- Registracija događaja / Otkazivanje događaja:

Registraciju i otkazivanje događaja ima pravo izvršiti samo vlasnik objekta. Potrebno je popuniti podatke o događaju (naziv, tip događaja, vrijeme održavanja, detaljan opis događaja), posebne napomene ako ih ima, te optionalno postaviti dobro ograničenje korisnika koji mogu prisustvovati događaju. Kreiranje novog događaja podrazumijeva navođenje specifikatora na osnovu kojih se odlučuje kojim korisnicima će događaj biti prikazan na dijelu za preporučene događaje. Specifikatori podrazumijevaju hashtag-e, dob korisnika i spol. Na osnovu ovoga korisnici dobivaju notifikacije o nadolazećim događajima iz njihove grupe interesovanja. Notifikacije se isključivo mogu slati do 1h prije početka događaja. Vlasnik objekta može otkazati događaj pri čemu dolazi notifikacija o otkazivanju svim korisnicima koji su rezervisali taj događaj.

- Prijava na događaj / Otkazivanje rezervacije za događaj:

Korisnik se prijavljuje na događaj na način da odabere događaj koji želi, a zatim odabere za koliko osoba rezerviše događaj te optionalno navede neku napomenu ako je ima. Korisnik također može vidjeti i sve njegove prijatelje koji idu na isti događaj. Zatim se prikazuje tlocrt prostora gdje su crvenom bojom označeni već zauzeti stolovi, a zelenom bojom označeni slobodni. Korisnik odabire sto koji želi. Ukoliko korisnik ne ispunjava uslov dobnog ograničenja, dobit će upozorenje, te mu rezervacija neće biti prihvaćena. Nakon uspješne rezervacije korisnik dobiva notifikaciju da je rezervacija prihvaćena i QR kod koji se skenira drugim uređajem prilikom dolaska na događaj te se šalje potvrda dolaska na server što predstavlja validaciju njegovog dolaska. Također korisnik u bilo kojem trenutku može otkazati rezervaciju čime se šalju notifikacije njegovim prijateljima.

- Recenzija događaja:

Korisniku nakon završetka događaja dolazi notifikacija da izvrši recenziju događaja. Ocjenom od 1 do 5 ocjenjuje događaj te optionalno ostavi detljaniji komentar o događaju.

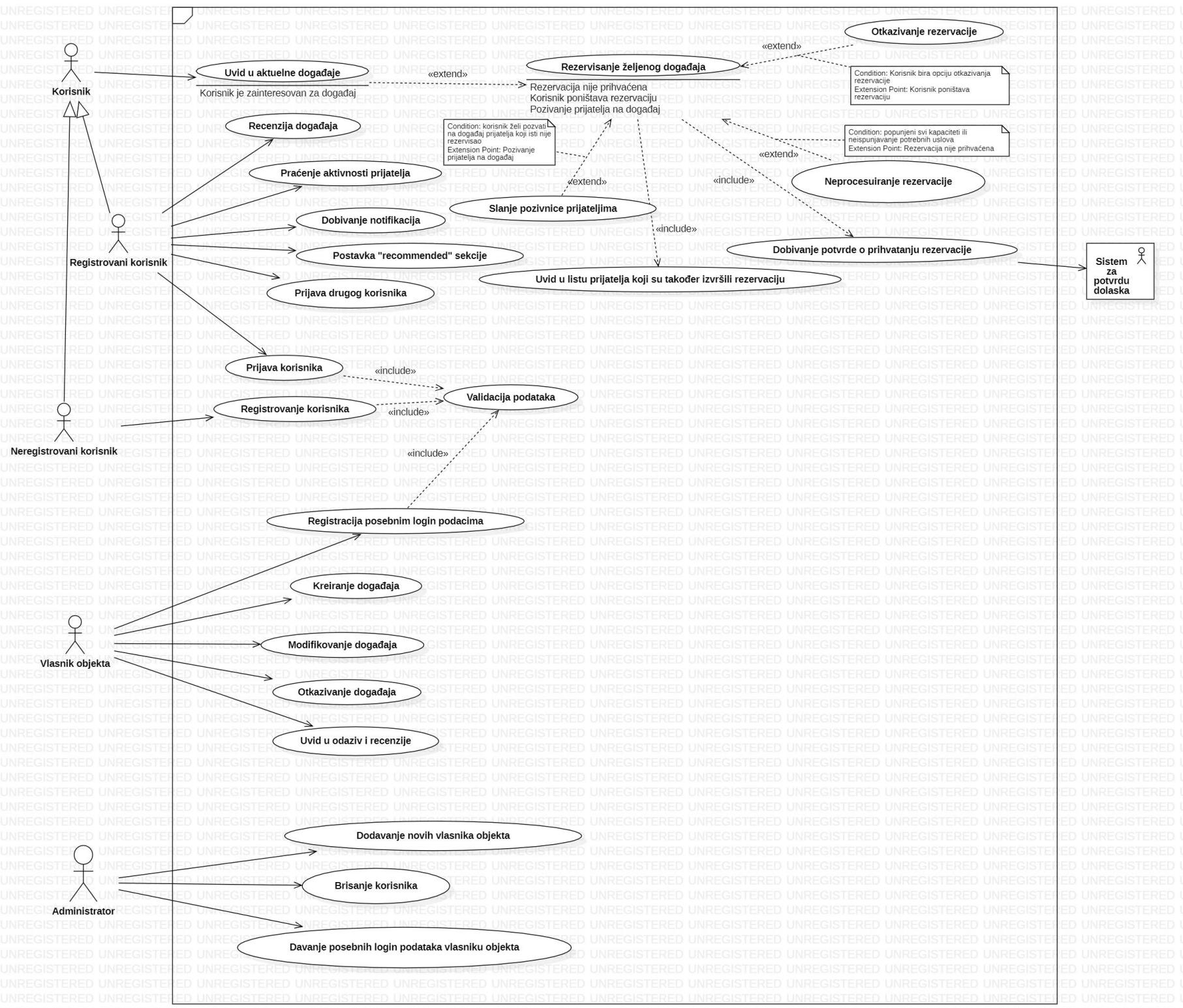
FUNKCIONALNOSTI:

- Mogućnost prijave na aplikaciju sa različitim ulogama
- Lista prijatelja: korisnik može dodati ostale korisnike na svoju listu pomoću njihovog username-a; dodani korisnici mogu prihvati ili odbiti taj zahtjev; prihvatanjem zahtjeva međusobno postaju prijatelji; moguće obrisati prijatelja s liste
- Korisnik dobiva "recommended" dio koji se filtrira na osnovu hashtag-a (korisnikovih interesovanja)
- Korisnik na profilu prijatelja vidi sve događaje na koje je taj prijatelj prijavljen
- Upload profilne fotografije korisnika
- Dobivanje notifikacija: korisnik dobiva obavještenje kada ga neko doda na listu prijatelja; obavještenje o skorom početku događaja na koji se prijavio; obavještenje za događaje na osnovu njegovih interesovanja; obavještenje kada prijatelj promijeni profilnu fotografiju; obavještenje kada se prijatelj prijavi na neki događaj; nakon završetka događaja dolazi obavještenje da korisnik izvrši recenziju
- Ograničenje o dolasku notifikacije do 1h prije početka događaja

- Uvid u slobodne stolove prilikom rezervacije
- Ne prihvatanje rezervacije korisnika ukoliko se prijavio na događaj za koji ne ispunjava uslove dobnog ograničenja
- Registracija na događaj
- CRUD mogućnost nad korisnikom i vlasnikom objekta
- Registracija događaja ako ste vlasnik objekta
- Skener koji potvrđuje dolazak korisnika (mušterije)
- Filtriranje aktivnosti po hashtag-u
- Ažuriranje/brisanje događaja kao vlasnik objekta
- Uvid u interesovanje prijatelja za događaj (koji prijatelji idu na koji događaj)
- Neregistrovani korisnik ima mogućnost uvida u aktuelne događaje
- Ograničenje za neregistrovane korisnike da ne mogu rezervisati stol ili vidjeti koliko i koje osobe idu na neki događaj
- Recenzija događaja od 1 do 5 nakon završetka
- Ograničenje na korisnikovu lozinku: minimalan broj znakova je 8, a također moraju biti i velika i mala slova
- Ograničenje da je korisničko ime jedinstveno
- Vlasnik objekta ima uvid u recenzije događaja nakon njegovog završetka
- Korisnik ima mogućnost prijave (report) drugog korisnika zbog neprimijerenog sadržaja, zloupotrebe podataka...
- Administrator dodaje nove vlasnike objekta i daje im posebne login podatke
- Administrator briše korisnikov račun ako dobije više od 3 prijave

AKTERI:

- Vlasnik objekta - osoba koja ima mogućnost kreiranja eventa, njegovog modifikovanja, promocije i otkazivanja. Ima uvid u broj registrovanih korisnika na aplikaciji, broj rezervacija, broj otkazanih rezervacija i broj osoba koji je prisustvovao događaju.
- Korisnik - ima uvid u nadolazeće događaje, postavlja kriterije na osnovu kojih se događaji filtriraju i prikazuju u recommended sekciji, dobiva notifikacije koje mu omogućavaju lakše praćenje događaja, uvid u nerezervisana mjesta, te mogućnost rezervacije mjesta po želji, kao i otkazivanje rezervacije u bilo kojem trenutku, mogućnost praćenja aktivnosti drugog korisnika njegovim dodavanjem na friend-listu.
- Neregistrovani korisnik - ima uvid u aktuelne događaje u objektu.
- Administrator - dodaje nove vlasnike objekta, te po potrebi briše druge korisnike.
- Uredaj za skeniranje - skenira QR kod prilikom dolaska korisnika (mušterije) na događaj, a zatim šalje odgovarajuću informaciju bazi.



Scenarij: Kreiranje događaja

Opis	Vlasnik objekta kreira novi događaj, popunjava podatke o događaju i šalje notifikaciju svim korisnicima.
Vezani zahtjevi	Vlasnik objekta kreira novi događaj.
Preduvjeti	Kreator događaja je vlasnik objekta.
Posljedice - uspješan završetak	Vlasnik objekta je uspješno kreirao novi događaj i korisnici koji ga prate su notificirane.
Posljedice - neuspješan završetak	Vlasnik objekta nije kreirao novi događaj.
Primarni akter	Vlasnik objekta
Ostali akteri	Korisnik, Sistem za notifikacije, Baza podataka
Glavni tok	Završava kreiranim novim događajem.
Proširenja / alternative	Neuspješna prijava na aplikaciju kao vlasnik objekta

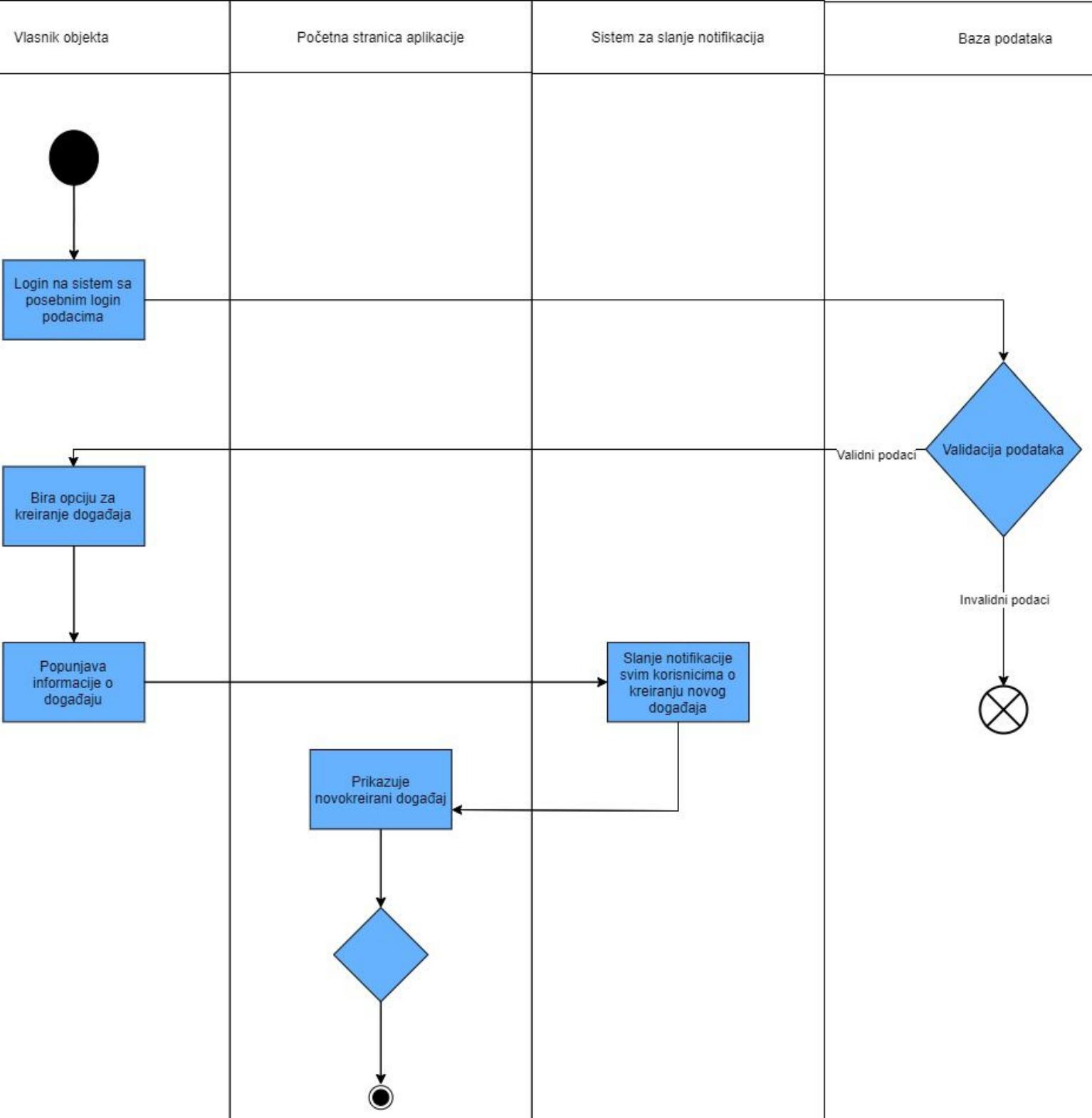
Tok događaja - uspješan završetak

Vlasnik objekta	Korisnik	Sistem za notifikacije	Baza podataka
1. Prijavljuje se na sistem sa posebnim login podacima			2. Validacija unesenih podataka
3. Bira opciju za kreiranje događaja			
4. Popunjava informacije o događaju			
5. Dodaje specifikatore na osnovu kojih se odlučuje kojim će korisnicima događaj biti prikazan			

6. Objavljuje događaj			
7. Prima informaciju o uspješno kreiranom događaju			
	9. Dobija informaciju o novokreiranom događaju	8. Prima zahtjev za slanje notifikacija svim korisnicima	

Alternativni tok - neuspješna prijava na aplikaciju kao vlasnik objekta

Vlasnik objekta	Korisnik	Sistem za notifikacije	Baza podataka
1. Prijavljuje se na sistem sa posebnim login podacima			2. Validacija unesenih podataka
			3. Vraćanje informacije da su neispravni login podaci
4. Dobivanje upozorenja o neispravnosti podataka			
5. Nastavak korištenja aplikacije			



Scenarij: Otkazivanje rezervacije za događaj

Opis	Korisnik otkazuje rezervaciju za događaj tako što odabere željeni događaj na koji je već prijavljen i ispunil određene podatke. Korisnik u bilo kojem trenutku može otkazati rezervaciju. Svi korisnici s liste prijatelja dobivaju notifikaciju.
Vezani zahtjevi	<ul style="list-style-type: none">– Mogućnost otkazivanja rezervacije.– Slanje odgovarajućih notifikacija.
Preduvjeti	<ul style="list-style-type: none">– Osoba koja otkazuje rezervaciju mora biti registrovan kao korisnik.– Korisnik je već prijavljen na događaj.– Događaj mora postojati.– Za slanje notifikacija o otkazivanju korisnik mora imati listu prijatelja.
Posljedice - uspješan završetak	Korisnik je otkazao rezervaciju.
Posljedice - neuspješan završetak	Korisnik nije uspio otkazati rezervaciju.
Primarni akteri	Korisnik
Ostali akteri	Baza podataka, sistem za notifikacije
Glavni tok	Završava uspješnim otkazivanjem rezervacije.
Proširenja/Alternative	Neuspješna prijava na aplikaciju ili korisnik nije prijavljen na događaj.

Tok događaja – uspješno otkazivanje rezervacije

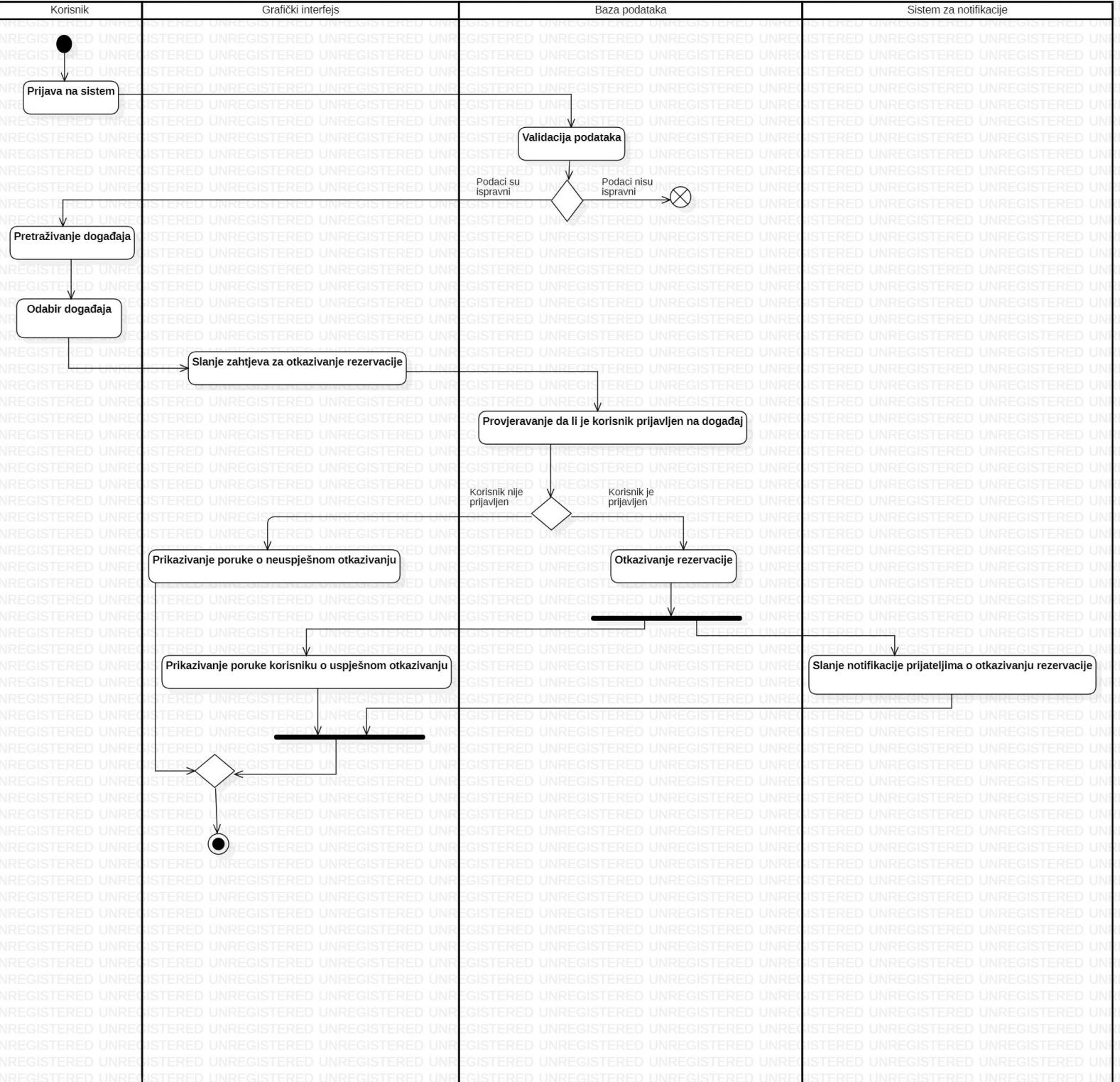
Korisnik	Baza podataka	Sistem za notifikacije
1. Prijava na aplikaciju	2. Validacija podataka	
3. Odabir događaja	4. Provjeravanje da li je korisnik prijavljen na događaj	
5. Otkazivanje rezervacije		
	6. Unošenje u bazu	7. Primanje informacija o događaju i korisniku
		8. Slanje notifikacija svim prijateljima s liste

Alternativni tok – neuspješno otkazivanje rezervacije zbog neuspješne prijave na aplikaciju

Korisnik	Baza podataka	Sistem za notifikacije
1. Prijava na aplikaciju	2. Validacija podataka	
	3. Vraćanje informacije da su neispravni login podaci	
4. Dobivanje upozorenja o neispravnosti podataka		
5. Nastavak korištenja aplikacije		

Alternativni tok – neuspješno otkazivanje rezervacije jer korisnik nije prijavljen na događaj

Korisnik	Baza podataka	Sistem za notifikacije
1. Prijava na aplikaciju	2. Validacija podataka	
3. Odabir događaja	4. Provjera da li je korisnik prijavljen na događaj	
	5. Vraćanje informacije da korisnik nije prijavljen na događaj	
6. Dobivanje upozorenja o nemogućnosti otkazivanja rezervacije		
7. Nastavak korištenja aplikacije		



Scenarij: Poništavanje događaja

Opis	Vlasnik objekta poništava ranije kreirani događaj o čemu se notificiraju svi korisnici koji su rezervisali isti
Vezani zahtjevi	Poništavanje događaja Slanje notifikacije korisnicima
Preduslov	Vlasnik objekta ima aktivan događaj koji želi poništiti
Posljedice - Uspješan završetak	Vlasnik objekta je poništio događaj
Primarni akter	Vlasnik objekta
Ostali akteri	Sistem za notifikacije, korisnik, baza podataka
Glavni tok	Završava poništenim događajem
Proširenja/Alternative	Neuspješna prijava

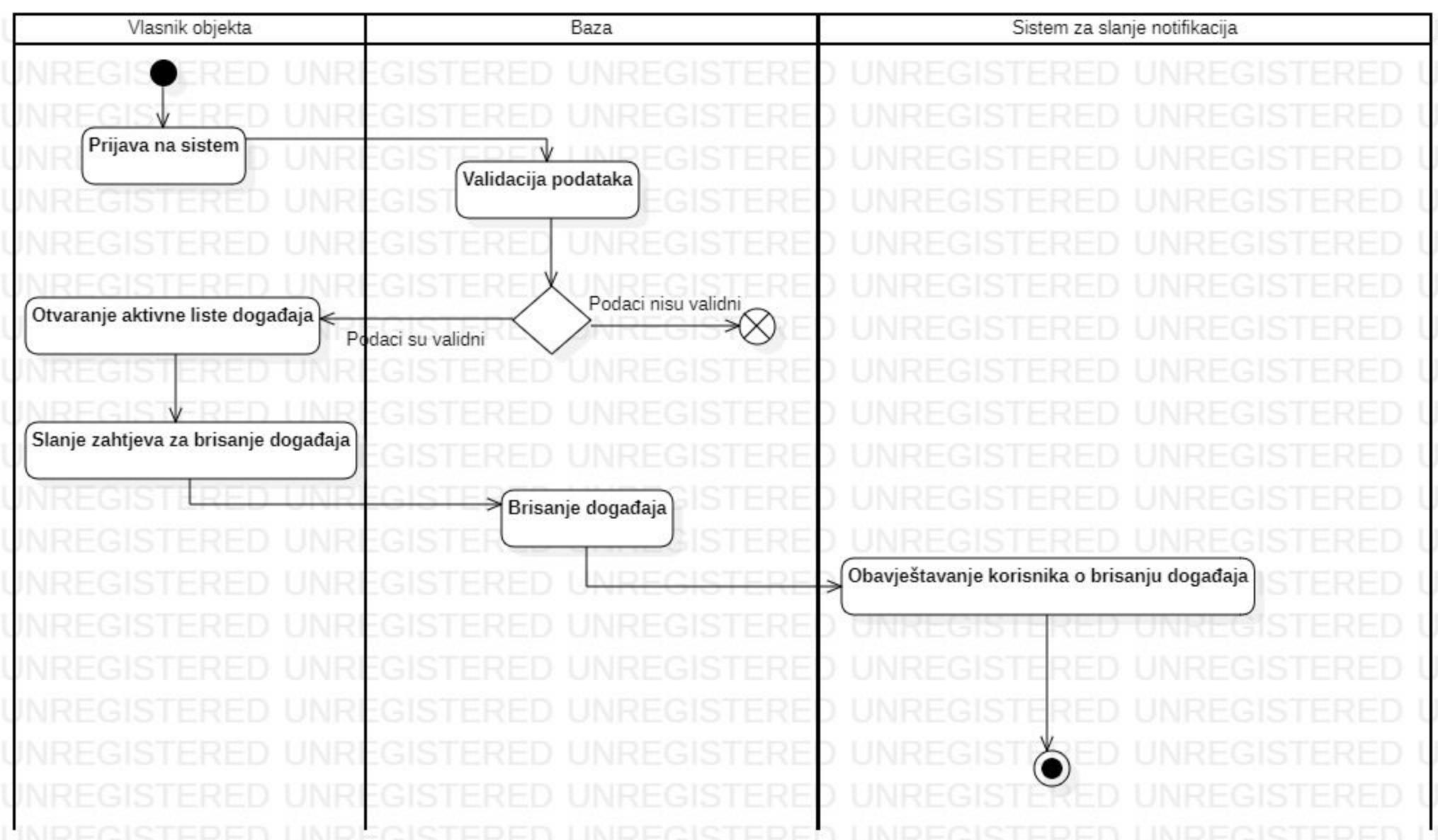
Tok događaja - uspješno poništen događaj

Vlasnik objekta	Korisnik	Sistem za slanje notifikacija	Baza podataka
1.Loguje se na sistem			2.Validacija podataka
3.Otvara listu aktivnih događaja			4.Dobija zahtjev za listu svih aktivnih događaja vlasnika objekta
6.Dobija listu aktivnih događaja			5.Vraća aktivne događaje
7.Pronalazi odgovarajući događaj			
8.Bira opciju za brisanje i šalje zahtjev bazi			9.Prima zahtjev o brisanju događaja i briše isti
9.Šalje zahtjev za slanje notifikacija		10.Dobiva zahtjev za slanje notifikacija	
	12. Dobiva notifikaciju o otkazivanju	11.Šalje notifikaciju o otkazivanju	

	poništavanju događaja	događaja svim korisnicima koji su se prijavili na isti	
--	--------------------------	--	--

Alternativni tok - neuspješno poništavanje događaja zbog neuspješne prijave

Vlasnik objekta	Korisnik	Sistem za slanje notifikacija	Baza podataka
1.Loguje se na sistem			2.Validacija podataka
			3.Vraća informaciju da su neispravni login podaci
4.Dobivanje upozorenja o neispravnosti podataka			
5.Nastavak korištenja aplikacije			



Scenarij: Prijava na događaj

Opis	Korisnik se prijavljuje na događaj na način da odabere događaj koji želi, a zatim odabere za koliko osoba rezerviše događaj te na interaktivnoj mapi tlocrta objekta odabire koji sto želi, a zatim optionalno navede neku napomenu ako je ima. Nakon uspješne prijave dobiva notifikaciju da je prijava prihvaćena, a prijateljima s njegove liste se također šalje notifikacija da se prijavio na događaj.
Vezani zahtjevi	<ul style="list-style-type: none">– Mogućnost prijave na događaj.– Slanje odgovarajućih notifikacija.
Preduvjeti	<ul style="list-style-type: none">– Osoba koja se prijavljuje na događaj mora biti registrovan kao korisnik.– Događaj mora postojati.– Za slanje notifikacija o prijavi korisnik mora imati listu prijatelja.– Mora biti slobodnih mjesta.– Mora ispunjavati dobro ograničenje.
Posljedice - uspješan završetak	Korisnik se uspješno prijavio na događaj.
Posljedice - neuspješan završetak	Korisnik se nije uspio prijaviti na događaj.
Primarni akteri	Korisnik
Ostali akteri	Baza podataka, sistem za notifikacije, interaktivna mapa tlocrta objekta
Glavni tok	Završava uspješnom prijavom.
Proširenja/Alternative	Nema slobodnog mjesta; korisnik ne ispunjava dobro ograničenje; neuspješna prijava na aplikaciju

Tok događaja – uspješna prijava na događaj

Korisnik	Baza podataka	Sistem za notifikacije	Interaktivna mapa tlocrta objekta
1. Prijava na aplikaciju	2. Validacija podataka		
3. Odabir događaja	4. Provjera slobodnih mesta		
	5. Provjera dobnog ograničenja		
			6. Odabir željenog stola
7. Prijava na događaj	8. Unošenje u bazu	9. Primanje informacija o događaju i korisniku	
		10. Slanje notifikacija svim prijateljima s liste	

Alternativni tok - neuspješna rezervacija zbog neuspješne prijave na aplikaciju

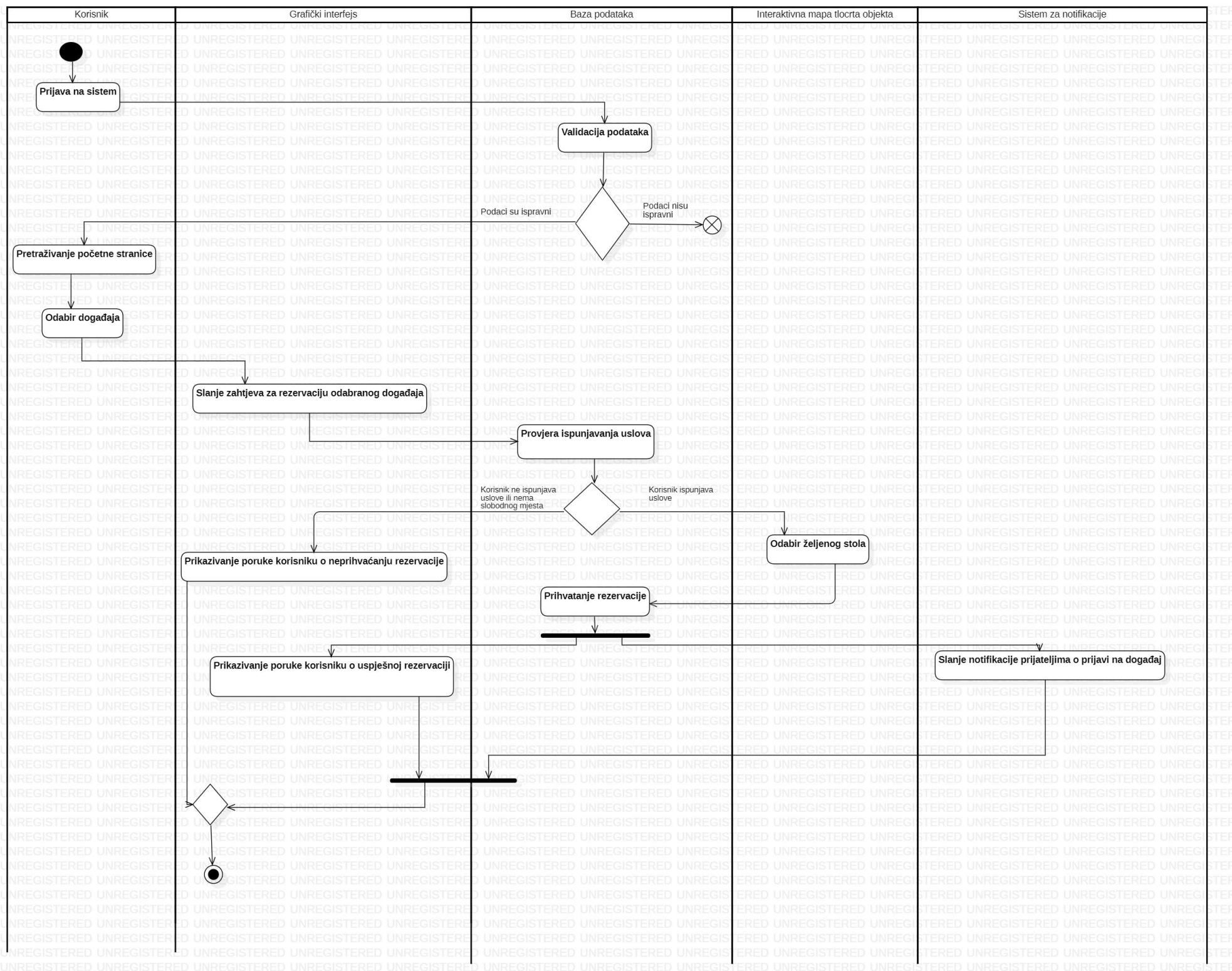
Korisnik	Baza podataka	Sistem za notifikacije	Interaktivna mapa tlocrta objekta
1. Prijava na aplikaciju	2. Validacija podataka		
	3. Vraćanje informacije da su neispravni login podaci		
4. Dobivanje upozorenja o neispravnosti podataka			
5. Nastavak korištenja aplikacije			

Alternativni tok - neuspješna rezervacija jer nema slobodnog mesta

Korisnik	Baza podataka	Sistem za notifikacije	Interaktivna mapa tlocrta objekta
1. Prijava na aplikaciju	2. Validacija podataka		
3. Odabir događaja	4. Provjera slobodnih mesta		
	5. Vraćanje informacije da je kapacitet popunjen		
6. Dobivanje upozorenja o popunjenoći kapaciteta događaja			
7. Nastavak korištenja aplikacije			

Alternativni tok - neuspješna rezervacija zbog neispunjavanja dobnog ograničenja

Korisnik	Baza podataka	Sistem za notifikacije	Interaktivna mapa tlocrta objekta
1. Prijava na aplikaciju	2. Validacija podataka		
3. Odabir događaja	4. Provjera slobodnih mesta		
	5. Provjera dobnog ograničenja		
	6. Vraćanje informacije o neispunjavanju dobnog ograničenja		
8. Dobivanje upozorenje o dobnom ograničenju i neispunjavanju istog			
9. Nastavak korištenja aplikacije			



Scenarij: Registracija korisnika

Opis	Korisnik popunjava lične podatke nakon čega dobiva mogućnost korištenja aplikacije ili odabire opciju da poveže svoj račun sa nekom od postojećih društvenih mreža čime se preskače proces unošenja ličnih podataka. Navodi se lista interesovanja u vidu hashtag-a na osnovu kojih se filtriraju događaji koji se prikazuju.
Vezani zahtjevi	
Preduvjeti	Korisnik je uspješno unesen u bazu podataka.
Posljedice - uspješan završetak	Korisnik ima pristup aplikaciji.
Posljedice - neuspješan završetak	Korisnik nema pristup aplikaciji.
Primarni akteri	Korisnik
Ostali akteri	Baza podataka
Glavni tok	Završava uspješno registrovanim novim korisnikom
Proširenja / alternative	Neuspješna registracija korisnika Već postoji osoba sa istim log in podacima

Tok događaja

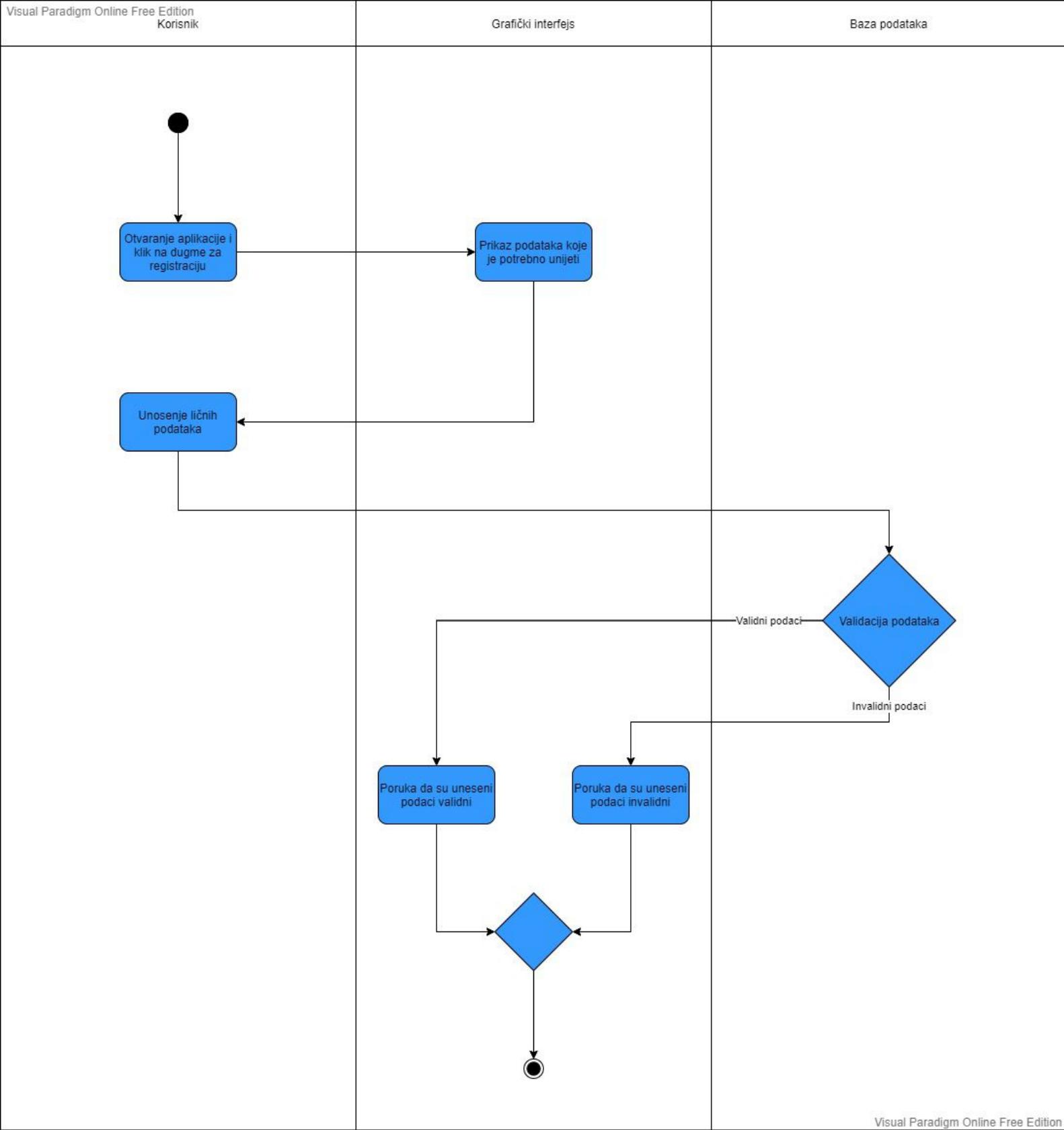
Korisnik	Baza podataka
1. Unošenje ličnih podataka	
	2. Provjera da li već postoji korisnik sa tim podacima
	3. Ispituje validnost podataka
	4. Potvrđuje validnost podataka
	5. Unošenje korisnika u bazu
6. Dobija potvrdu o uspješno kreiranom računu	
7. Nastavak korištenja aplikacije	

Alternativni tok - Neuspješna registracija korisnika

Korisnik	Baza podataka
1. Unošenje ličnih podataka	
	2. Provjera da li već postoji korisnik sa tim podacima
	3. Ispituje validnost podataka
	4. Vraća informaciju da podaci nisu validni
5. Dobija upozorenje da uneseni podaci nisu validni	
6. Nastavak korištenja aplikacije	

Alternativni tok - Već postoji osoba sa istim log in podacima

Korisnik	Baza podataka
1. Unošenje ličnih podataka	
	2. Provjera da li već postoji korisnik sa tim podacima
	3. Vraća informaciju da već postoji korisnik sa istim log in podacima
4. Dobija upozorenje da već postoji korisnik sa istim log in podacima	
5. Nastavak korištenja aplikacije	



Scenarij: Registracija vlasnika objekta

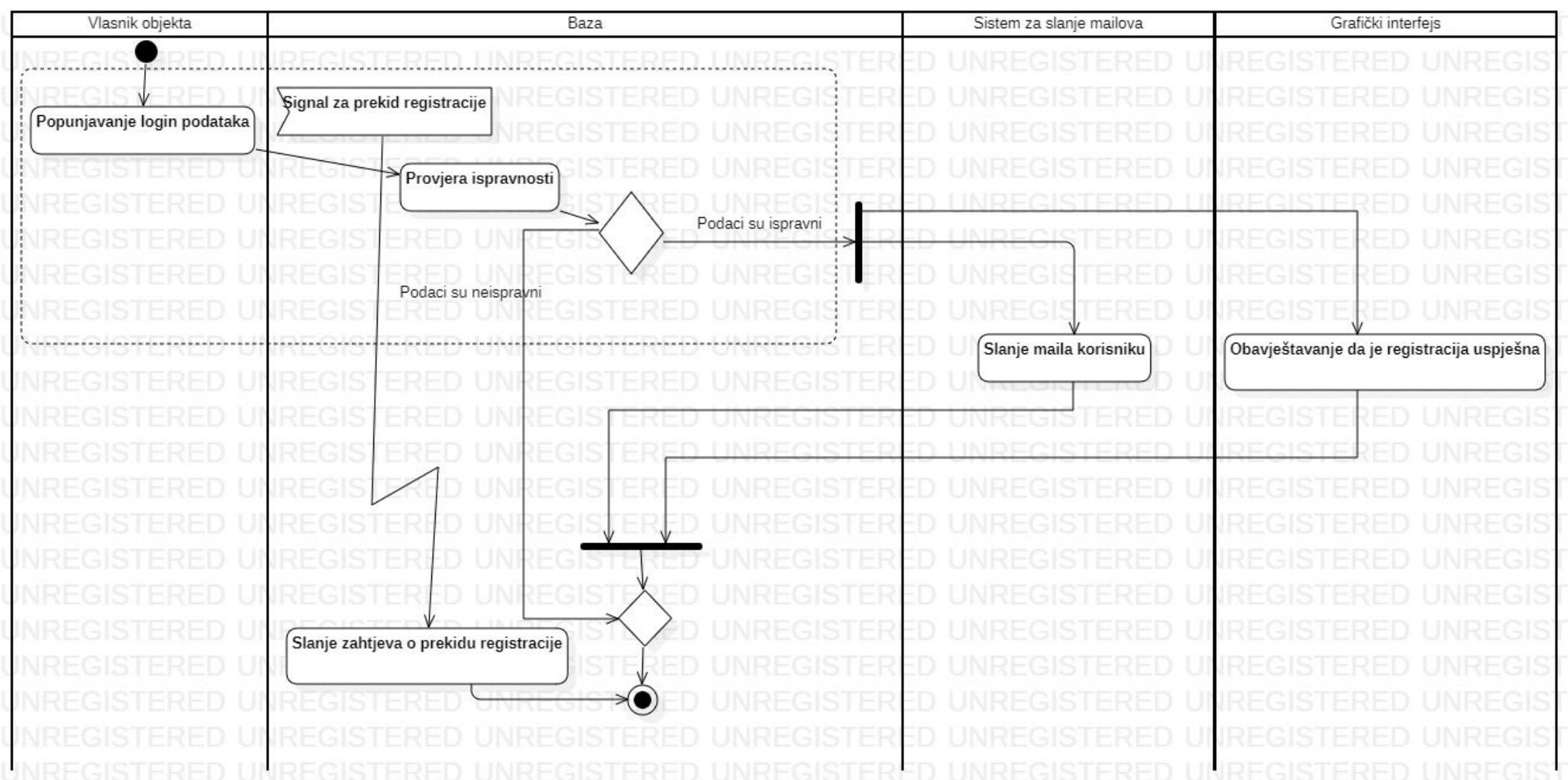
Opis	Vlasnik objekta kreira svoj račun za aplikaciju na osnovu dobijenih login informacija od admina. Podaci moraju proći kroz verifikaciju baze.
Vezani zahtjevi	
Preduvjeti	Osoba je vlasnik objekta i dobila je posebne login podatke od admina
Posljedice - uspješan završetak	Vlasnik objekta je uspješno napravio račun i dobio potvrdu na mail
Posljedice - neuspješan završetak	Vlasnik objekta nije uspio kreirati račun
Primarni akter	Vlasnik objekta
Ostali akteri	Sistem za verifikaciju, baza podataka
Glavni tok	Završava kreiranim računom
Proširenja / Alternative	Uneseni su neispravni login podaci

Tok događaja - Uspješno kreiranje računa

Korisnik	Baza podataka
1. Pristup login stranici	
2. Odabir opcije za registraciju	
3. Odabir tipa prijave za vlasnike objekta	
4. Popunjavanje login podataka	5. Primanje login podataka
	6. Provjera ispravnosti
	7. Slanje potvrde o ispravnosti
8. Popunjavanje podataka o objektu	
9. Primanje potvrde o uspješno kreiranom računu	

Alternativni tok - Neuspjeh pri validaciji login podataka

Korisnik	Baza podataka
	7. Provjera ispravnost
9. Primanje informacije o neispravnosti login podataka	8. Neispravni login podaci ili pronalazak poklapanja
10. Ponovno popunjavanje novih login podataka	
	11. Nastavak procesa u tački 5 glavnog toka



Scenarij: Validacija dolaska

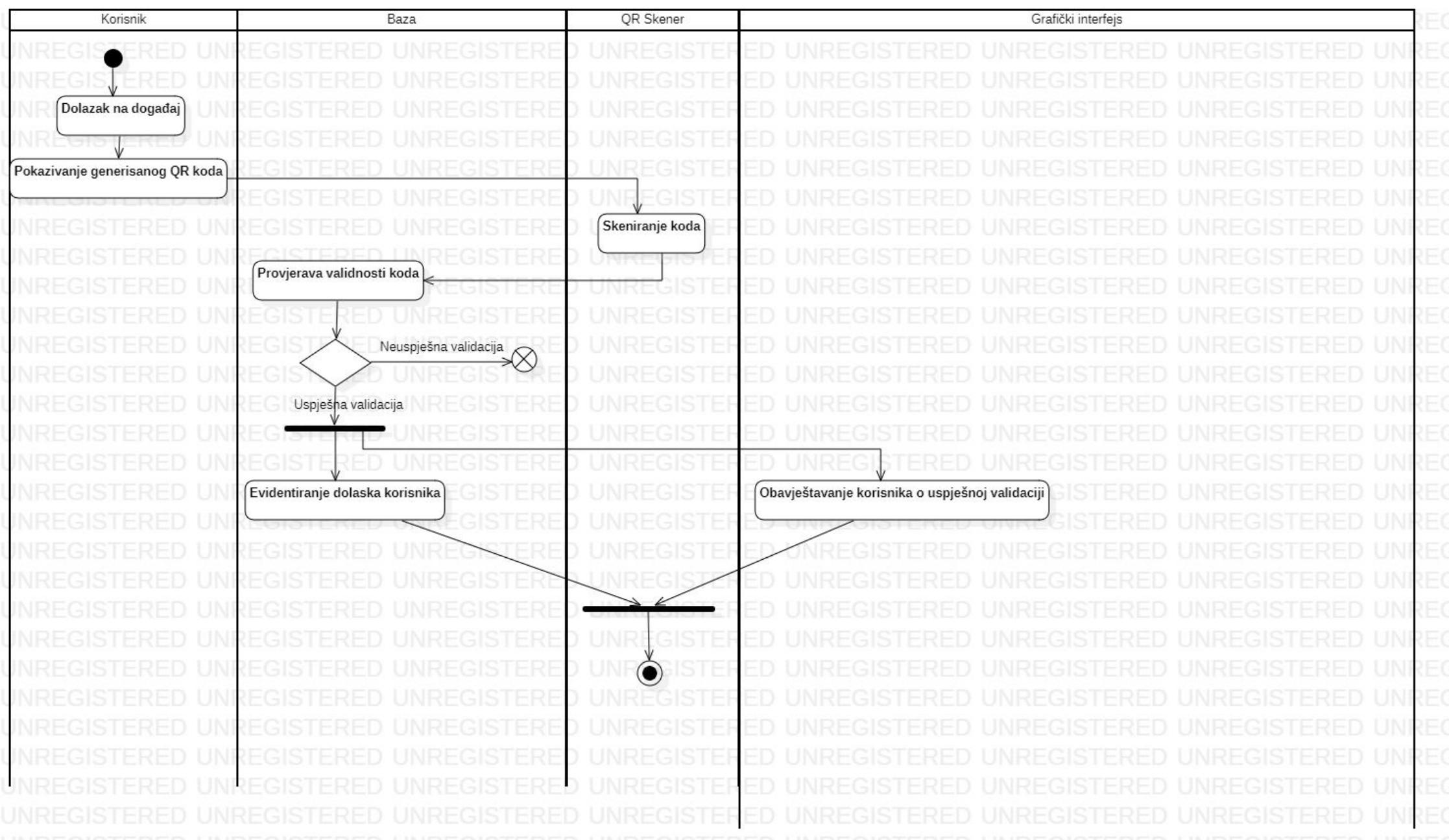
Opis	Korisnik koji se prijavio na događaj potvrđuje svoj dolazak na isti skeniranjem QR koda.
Vezani zahtjevi	Korisnik je ranije rezervisao događaj
Preduslovi	Korisnik je dobio potvrdu o prihvatanju rezervacije
Posljedice - uspješan završetak	Korisnik je uspješno registrovao svoj dolazak na događaj
Posljedice - neuspješan završetak	Korisnik nije uspio registrovati svoj dolazak
Primarni akter	Korisnik
Ostali akteri	QR skener, baza podataka
Glavni tok	Završava validiranim dolaskom
Proširenja/Alternative	Korisnik je dao pogrešan QR kod ili je ranije otkazao rezervaciju

Tok događaja - Uspješna validacija dolaska

Korisnik	Baza podataka	QR skener
1.Dolazi na događaj		
2.Pokazuje QR kod skeneru		3.Skenira QR kod
	5. Provjera ispravnosti proslijedene rezervacije	4.Šalje upit bazi za provjeru ispravnosti
	6. Šalje odgovor skeneru o pronašlasku rezervacije	7. Dobija informaciju iz baze
9. Prima informaciju o uspjehu validacije	7. Zapisuje dolazak korisnika	8. Obavještava korisnika o uspješnoj validaciji

Alternativni tok - neuspješna validacija dolaska zbog pogrešnog QR koda ili ranijeg otkazivanja rezervacije

Korisnik	Baza podataka	QR skener
1.Dolazi na događaj		
2.Pokazuje QR kod skeneru		3.Skenira QR kod
	5. Ne pronađe proslijeđenu rezervaciju i šalje odgovor skeneru	4.Šalje upit bazi za provjeru ispravnosti
		6. Dobija informaciju iz baze
8. Prima informaciju o neuspješnoj validaciji		7. Obavještava korisnika o neuspješnoj validaciji





[Početna](#) | [Dogadaji](#) | [Galerija](#) | [Kontakt](#)

[Prijava](#)

[Registracija](#)

NIGHT ALGORITHM

Nadolazeći događaji

Događaj 1



Događaj 2



[Pogledaj sve događaje](#)

Night Algorithm



Lokacija objekta



E-mail objekta





[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

[Prijava](#)

[Registracija](#)

Nadolazeći događaji

Događaj 1

Text o događaju

Vidi više



Događaj 2

Text o događaju

Vidi više



Za rezervaciju događaja potrebna je prijava ili registracija





[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

[Prijava](#)

[Registracija](#)

Galerija

Događaj 1



Događaj 2



[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

Prijava

Registracija

Događaj 1 - galerija





Kontaktirajte nas

Za više informacija o nadolazećim događajima i našoj ponudi
kontaktirajte nas putem kontakt forme

Night Algorithm d.o.o.
Adresa objekta
88000 Mostar
Bosna i Hercegovina

Mobitel: +387 xx xxx xxx



Night Algorithm

<https://nighthalgorithm.com/registracija>

Početna | Dogadaji | Galerija | Kontakt Prijava Registracija

Registracija

 Facebook prijava  Google prijava

ili se registrujte preko forme

Korisničko ime

E-mail

Lozinka (minimalno 8 znakova, uključujući velika i mala slova)

Ponovite lozinku

Dalje...

Sva prava pridržana © 2021 NightAlgorithm

Korisničko ime zauzeto!

Unesite ponovo

E-mail ne postoji!

Unesite ponovo

Lozinka mora sadržavati minimalno 8 znakova, velika i mala slova!

Unesite ponovo

Lozinke se ne podudaraju!

Unesite ponovo



[Početna](#) | [Dogadaji](#) | [Galerija](#) | [Kontakt](#)

Prijava

Registracija

Registracija



Profilna fotografija

Spol ▾

[Promijeni sliku profila](#)

Unesite datum rođenja:

3

Mjesec rođenja ▾

Godina rođenja ▾

Dalje...





[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

Prijava

Registracija

Registracija

Napravite svoju listu interesovanja!

Odaberite 3 ili više ponuđenih opcija:

TAG 1

TAG 2

TAG 3

TAG 4

TAG 5

TAG 6

TAG 7

TAG 8

TAG 9

TAG 10

TAG 11

TAG 12

TAG 13

TAG 14

TAG 15

TAG 16

TAG 17

TAG 18

TAG 19

TAG 20

Registrujte se



Uspješno ste registrovani!

[Nazad na početnu](#)

Night Algorithm

[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

[Prijava](#) [Registracija](#)

Looking for party? Sign in first

[Facebook prijava](#) [Google prijava](#)

ili se prijavite preko forme

E-mail

Lozinka

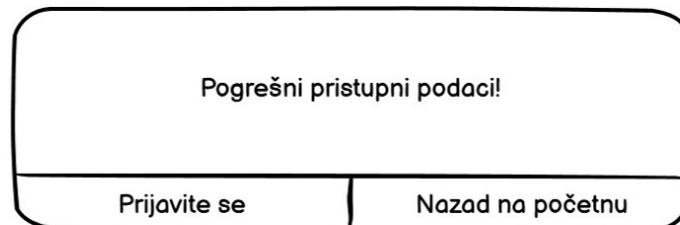
[Zaboravili ste lozinku?](#)

Zapamti me

Prijavite se

Nemate račun? [Registrirajte se](#)

Sva prava pridržana © 2021 NightAlgorithm

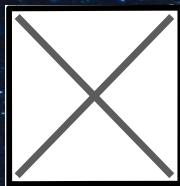


[Početna](#) | [Dogadaji](#) | [Galerija](#) | [Kontakt](#)[Prijava](#)[Registracija](#)

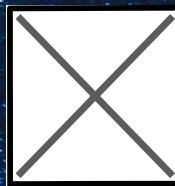
Lista prijatelja

 Pretraži

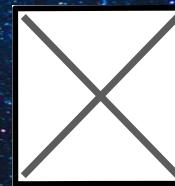
! Zahtjevi za prijateljstvo

[Dodaj prijatelja](#)

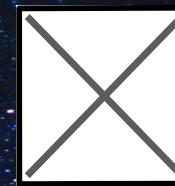
KorisničkoIme



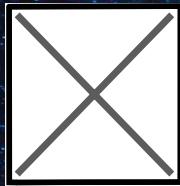
KorisničkoIme



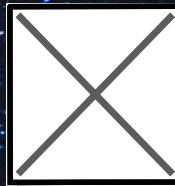
KorisničkoIme



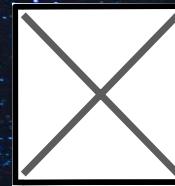
KorisničkoIme



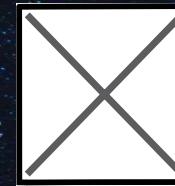
KorisničkoIme



KorisničkoIme



KorisničkoIme



KorisničkoIme

Sva prava pridržana © 2021 NightAlgorithm



KorisničkoIme je promijenio/la
profilnu fotografiju!

[Otiđi na profil](#)[Nazad](#)

KorisničkoIme se prijavio/la na
ImeDogađaja

[Pogledaj događaj](#)[Nazad](#)



[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

[Prijava](#)

[Registracija](#)

Prijava na događaj 1

Odaberite broj osoba za koliko rezervišete:

1

Da li imate neku posebnu napomenu?

NE



DA

Napomena

[Dalje...](#)





[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

[Prijava](#)

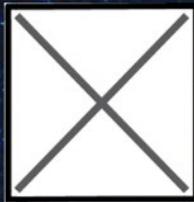
[Registracija](#)

Prijava na događaj 1

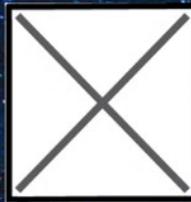
Pozovite prijatelja na ovaj događaj:

< Pozovi

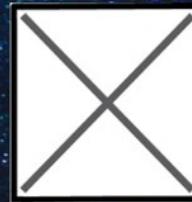
Vaši prijatelji koji su već prijavljeni na događaj 1:



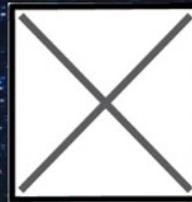
KorisničkoIme



KorisničkoIme



KorisničkoIme



KorisničkoIme

Dalje...



Night Algorithm

<https://nightalgorithm.com/prijavanadogadjaj>

Početna | Događaji | Galerija | Kontakt | Prijava | Registracija

Prijava na događaj 1

Odaberite gdje želite da bude Vaš sto:

💡 Crvenom bojom su označeni zauzeti, a zelenom slobodni stolovi

Rezerviši

Sva prava pridržana © 2021 NightAlgorithm

Neuspješna rezervacija
Ne ispunjavate dobro ograničene za ovaj događaj!

[Nazad na početnu](#)

Uspješno ste rezervisali sto!

[Dalje](#)

Sačuvajte ovaj QR kod i pokažete ga na ulazu!

[Spremi QR kod](#)

[Početna](#) | [Dogadjaji](#) | [Galerija](#) | [Kontakt](#)[Prijava](#)[Registracija](#)

Ime korisnika



E-mail



Datum rođenja

[Pošalji zahtjev za prijateljstvo](#)[Prijavi korisnika](#)

KorisničkoIme Vam je poslao zahtjev za prijateljstvo

Prihvati?

[Da](#)[Ne](#)Da li ste sigurni da želite prijaviti
KorisničkoIme?[Da](#)[Ne](#)

[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)[Prijava](#)[Registracija](#)

Profilna fotografija



Ime korisnika



E-mail



Datum rođenja

Prijavljeni događaji:



Događaj 1



Događaj 2



Događaj 3

[Izbriši prijatelja](#)[Prijavi korisnika](#)Da li ste sigurni da želite obrisati
KorisničkoIme sa liste prijatelja?[Da](#)[Ne](#)

[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)[Prijava](#)[Registracija](#)[Promijeni profilnu fotografiju](#)

Ime korisnika



E-mail



Datum rođenja

[< Lista prijatelja](#)

Preporučeni događaji:



Događaj 1.



Događaj 2.



Događaj 3.

Prijavljeni događaji:



Događaj 1.



Događaj 2.



Događaj 3.

[< Otkaži rezervaciju za događaj](#)

Da li ste sigurni da želite otkazati rezervaciju za
ImeDogađaja?

[Da](#)[Ne](#)

Uspješno ste otkazali rezervaciju!

[Nazad na početnu](#)


[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)
[Prijava](#)[Registracija](#)

Profilna fotografija



Ime vlasnika objekta



E-mail



Datum rođenja

[Promijeni profilnu fotografiju](#)

Registrirani događaji



Događaj 1



Događaj 2



Događaj 3

Arhiva:



Događaj 1



Događaj 2



Događaj 3

[Registruj događaj](#)[Otkaži registrirani događaj](#)

Sva prava pridržana © 2021 NightAlgorithm



Da li ste sigurni da želite otkažati ImeDogađaja?

[Da](#)[Ne](#)

Uspješno ste otkažali događaj!

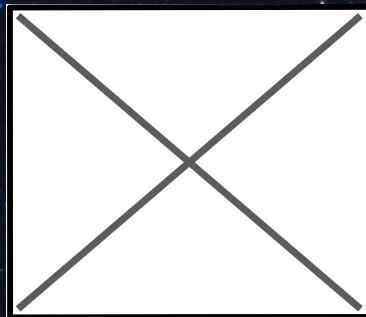
[Nazad na početnu](#)

Nažalost, ImeDogađaja je otkažan!

[Pogledajte druge događaje](#)[Nazad na početnu](#)Notifikacija korisniku o
otkaživanju događaja na koji je
prijavljen

[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)[Prijava](#)[Registracija](#)

Registracija - Događaj 1

[Upload fotografiju](#)**Naziv događaja:****Tip događaja** ▾**Datum održavanja:** / /**Vrijeme održavanja:** : **Opis događaja:****Posebne napomene:**
Napomena
Dobno ograničenje:

NE



DA

 18 **Dalje...**

[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)[Prijava](#)[Registracija](#)

Registracija - Događaj 1

Odaberite 3 ili više specifikatora za ovaj događaj:

 TAG 1 TAG 2 TAG 3 TAG 4 TAG 5 TAG 6 TAG 7 TAG 8 TAG 9 TAG 10 TAG 11 TAG 12 TAG 13 TAG 14 TAG 15 TAG 16 TAG 17 TAG 18 TAG 19 TAG 20

Registrujte događaj

Sva prava pridržana © 2021 NightAlgorithm



Uspješno ste registrovali događaj!

[Nazad na početnu](#)

Registrirani događaj bi Vas mogao zanimati!

[Pogledaj događaj](#)[Nazad na početnu](#)

Notifikacija korisniku o
događaju na osnovu njegove
liste interesovanja

Molimo Vas izvršite recenziju Događaj 1

Dalje

Nazad na početnu

Koliko Vam se dopao događaj?



[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)[Prijava](#)[Registracija](#)

Događaj 1

Naziv događaja:

Datum održavanja:

 / /

Vrijeme održavanja:

 : 

Događaj 1

Broj prijavljenih osoba:

< Uvid u popunjenošt



Klikom na dugme prikaže se
tlocrt objekta i
slobodni/rezervisani stolovi





[Početna](#) | [Događaji](#) | [Galerija](#) | [Kontakt](#)

Prijava

Registracija

Događaj 1



Događaj 1

Naziv događaja:

Datum održavanja:

 / /

Vrijeme održavanja:

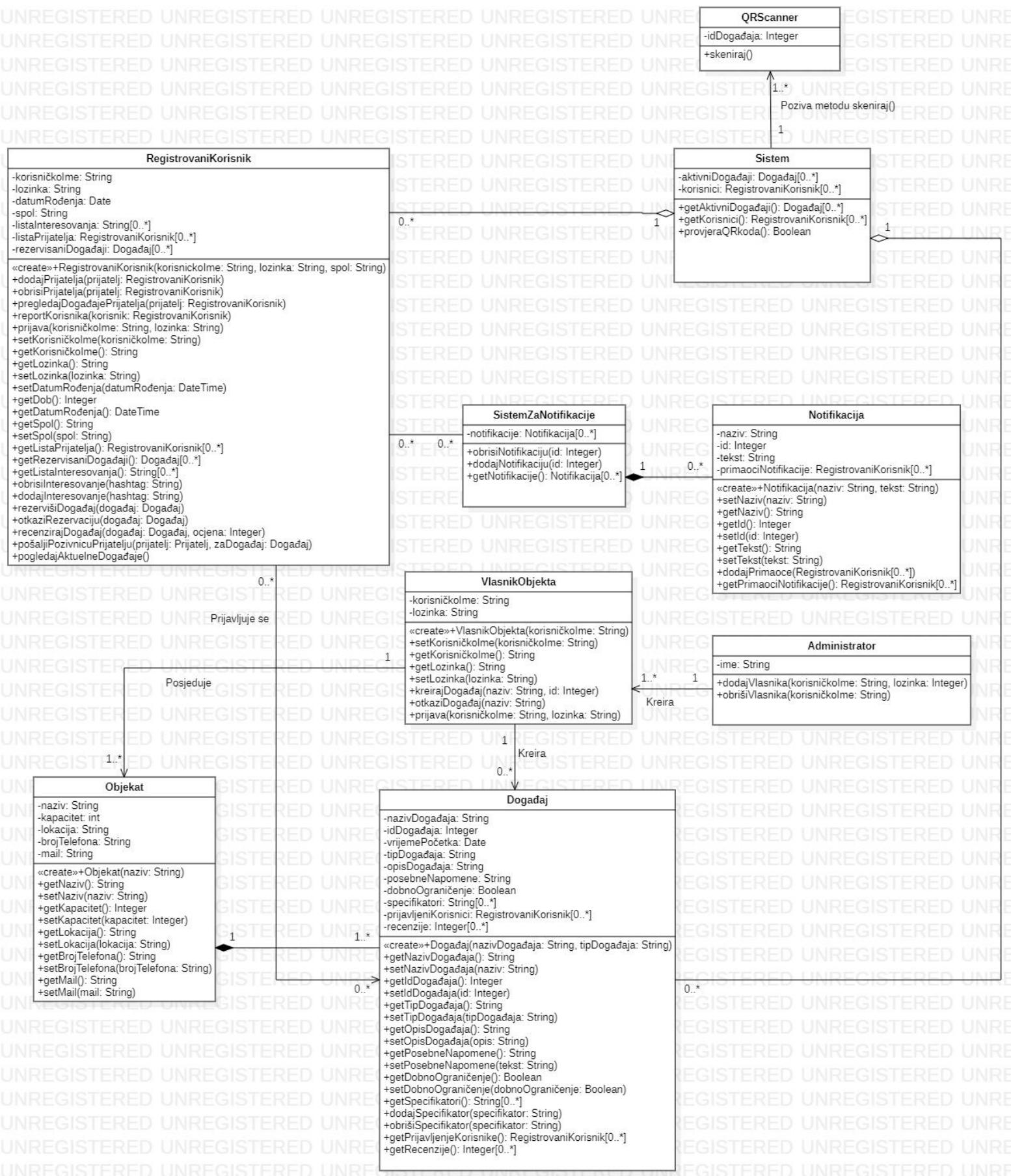
 : 

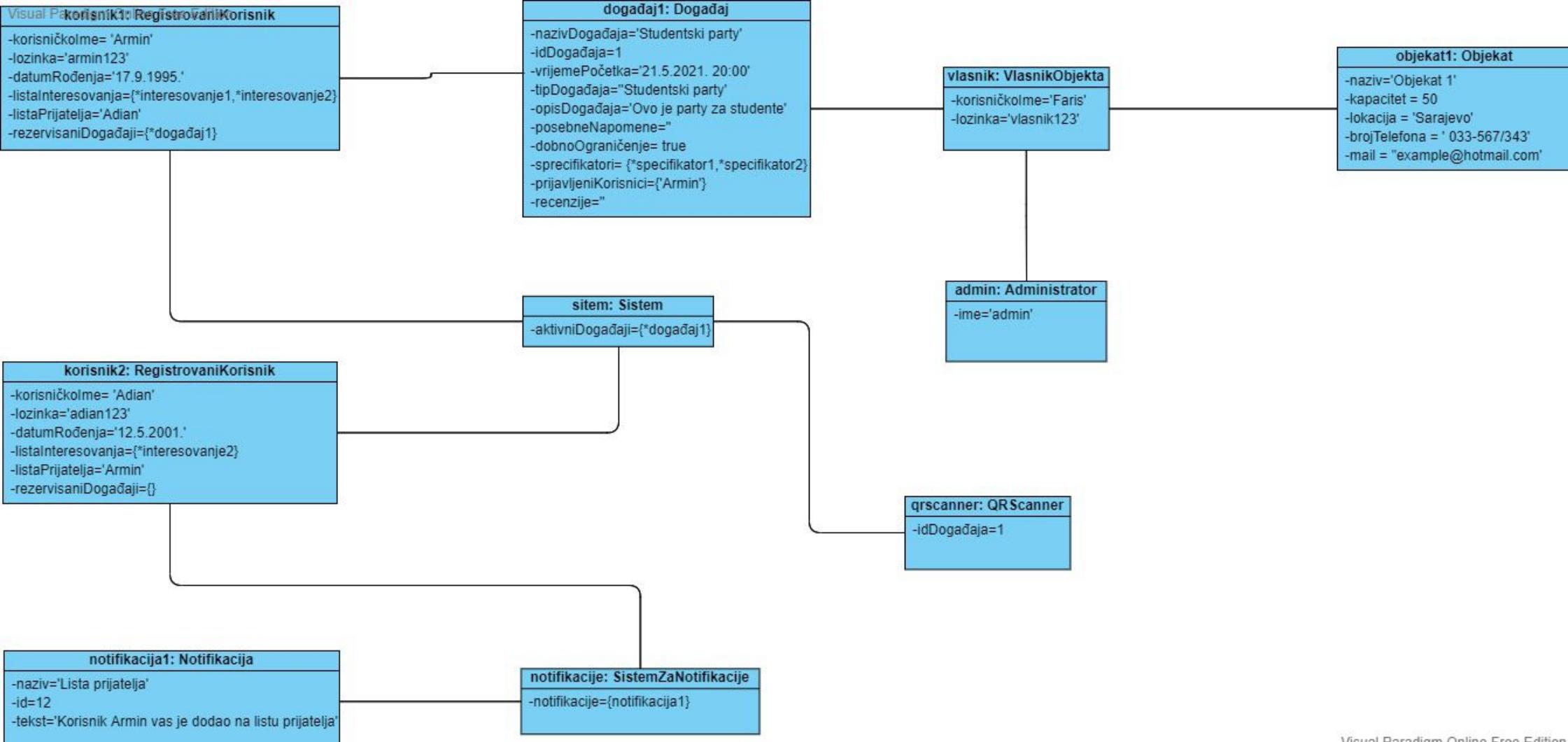
Broj prijavljenih osoba:

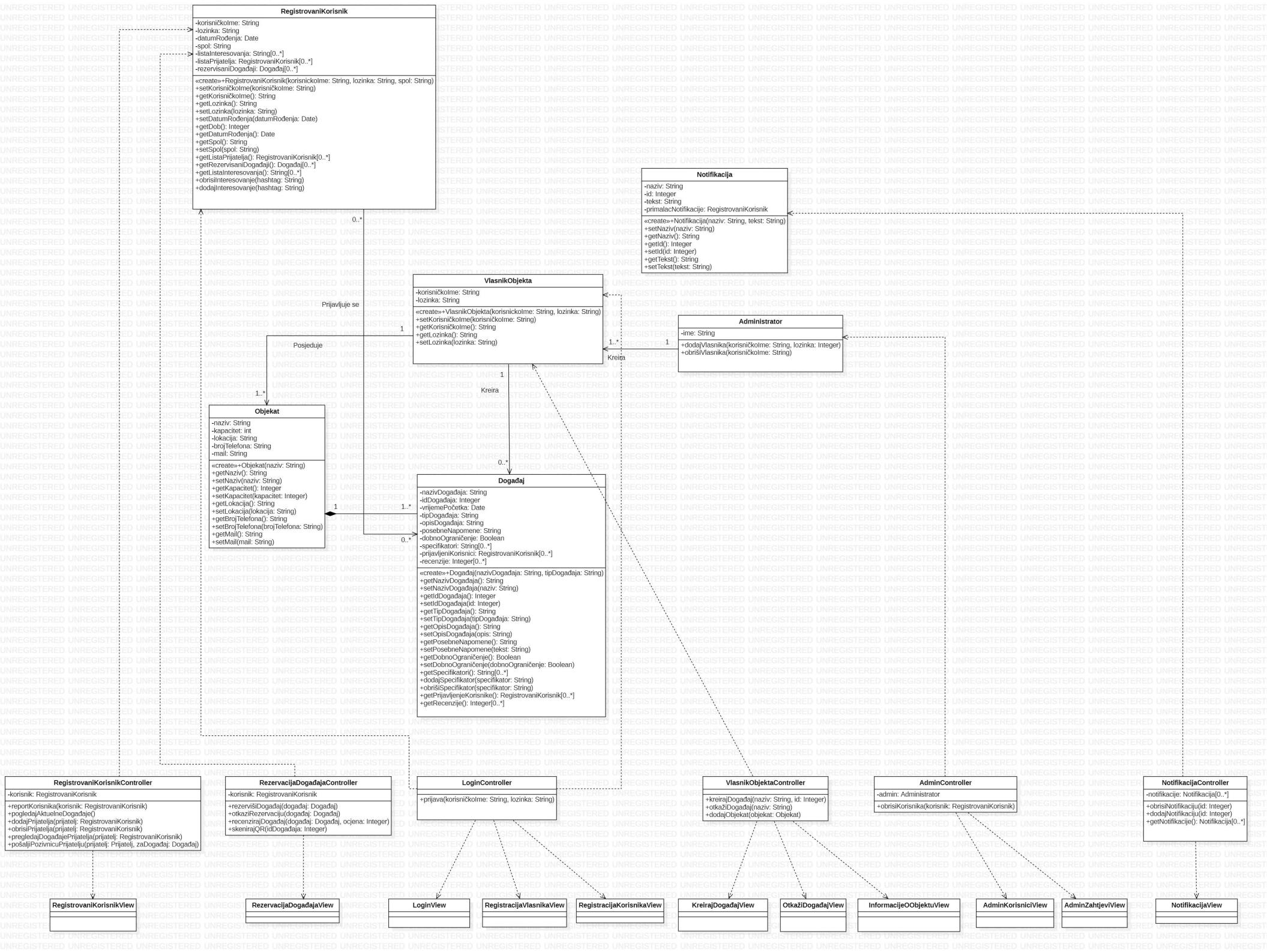


Uvid u recenzije









Single-responsibility principle

Prilikom dizajniranja našeg sistema trudili smo se da ispoštujemo Single responsibility principle. Prema ovome principu, svaka klasa bi trebala imati pojedinačnu odgovornost, te bi trebao postojati samo jedan razlog za promjenu klase. S obzirom da se svaka klasa koju smo definirali brine samo za ono za što je namijenjena, možemo reći da klase imaju pojedinačnu odgovornost, te da je ovaj princip ispoštovan.

Open-closed principle

Ovaj princip nalaže da kod treba biti otvoren za nadogradnju, a zatvoren za modifikacije, odnosno trebalo bi biti moguće što jednostavnije dodavanje novih funkcionalnosti bez potrebe da se modifikuju postojeće klase, njihovi atributi i funkcije. Jedan od načina da se to najlakše postigne jeste korištenje nasljeđivanja i polimorfizma. S obzirom da ne koristimo apstraktne klase, ne možemo implementirati polimorfizam, te također niti jedna klasa nije izvedena iz neke druge klase, tako da ovaj princip nije baš u potpunosti ispoštovan. Bez obzira na to, prilikom implementacije ćemo obratiti pažnju na to da napisani kod bude takav da je potrebno vršiti što manje modifikacija postojećeg koda prilikom dodavanja novih funkcionalnosti, kako bi sistem bio što otvoreniji za nadogradnju, te kako bi ovaj princip bio ispoštovan u određenoj mjeri.

Liskov substitution principle

Liskov princip zamjene kaže da podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima. Pošto u okviru našeg sistema ne postoje nikakvi podtipovi određenih tipova, samim time nema mnogo smisla komentarisati da li je ovaj princip zadovoljen.

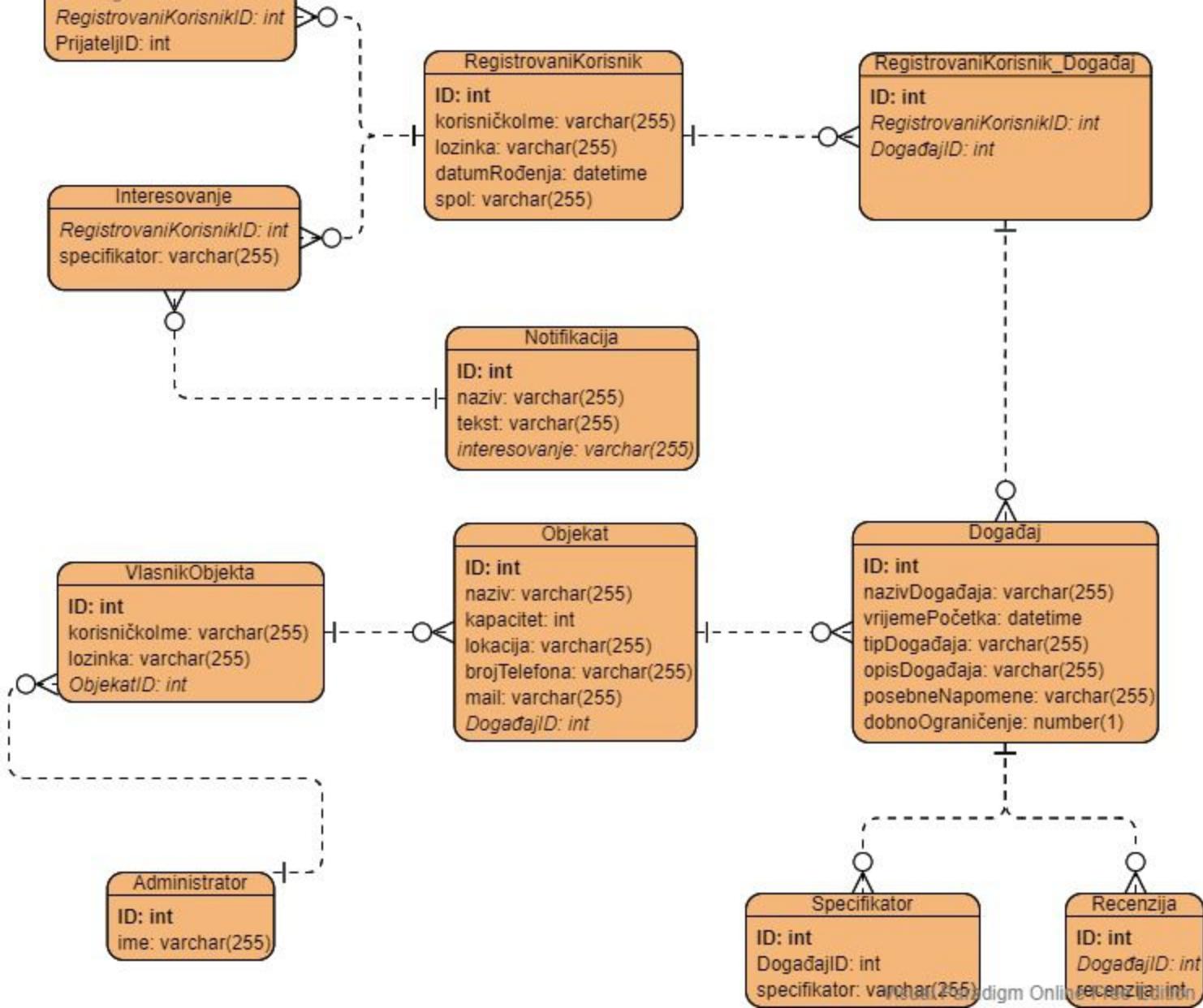
Interface segregation principle

Princip izoliranja interfejsa se odnosi na to da klijenti ne bi trebali da ovise od metoda koje neće upotrebljavati. Veoma jednostavno se može primijetiti da je ovaj princip zadovoljen, te da su metode koje su potrebne korisniku dostupne, te ih korisnik može koristiti, dok ostale metode koje korisniku nisu potrebne nisu niti dostupne, te će one biti označene kao private metode koje korisnik ne može direktno pozivati.

Dependency inversion principle

Princip inverzije ovisnosti glasi da moduli visokog nivoa ne bi trebali da ovise od modula niskog nivoa, odnosno oba bi trebala da ovise od apstrakcija. To znači da ne bi trebalo ovisiti od konkretnih klasa, nego bi prilikom nasljeđivanja trebalo razmotriti slučaj da je osnovna klasa

apstraktna. Pošto niti jedna od klasa koju koristimo nije apstraktna, može se zaključiti da princip inverzije ovisnosti nije ispoštovan.



STRUKTURALNI PATERNI

1. ADAPTER PATERN

U klasi RegistrovaniKorisnik imamo metodu getListaPrijatelja() kojom korisnik može pregledati sve svoje prijatelje. Mogli bismo napraviti adapter kojim ćemo vraćati sortiranu listu prijatelja po broju zajedničkih događaja, tj. prvo mjesto da bude prijatelj koji je prisustvovao najviše događaja kao i korisnik. Patern bismo implementirali tako što ćemo naslijediti adapter iz klase RegistrovaniKorisnik i implementirati interfejs IPrijatelji.

2. FACADE PATERN

Facade paternom ćemo objediniti metode getRezervisaniDogađaji() iz klase RegistrovaniKorisnik i getPrijavljeniKorisnike() iz klase Događaj. Nova metoda u facade klasi će omogućiti korisniku da ima uvid u sve prijavljene korisnike na odabrani događaj bez obzira jesu li mu prijatelji ili ne.

3. DECORATOR PATERN

Ovaj patern bi se mogao primijeniti ako bismo krenuli u pravcu proširivanja broja tipova događaja, ako bi se dodavale mogućnosti da događaji mogu biti VIP ili ne, događaji na otvorenom ili zatvorenom, sa promocijama i bez itd. Dekorator patern bi tu poslužio jer bi se tada moglo kombinovati sve te kategorije pri kreiranju događaja. To bi se moglo implementovati na način da se kreira interfejs IDogađaj koji sadrži sve metode klase Događaj i onda se kreiraju dekorator klase koje implementiraju taj interfejs (klase VipDogađajDekorator, PromocijaDogađajDekorator...). Bilo bi moguće napraviti kombinacije raznih tipova događaja pozivima new VipDogađajDekorator(new PromocijaDogađajDekorator(... new Događaj() ...)).

4. BRIDGE PATERN

Ovaj patern bi se mogao primjeniti na tipove različitog slanja notifikacija. Korisnik može primiti notifikaciju putem aplikacije vezano o događajima, prijateljima i slično, a preko e-maila će dobiti notifikaciju o potvrdi registracije.

5. PROXY PATERN

Dobra upotreba ovog paterna bi bila u metodi recenzirajDogađaj(događaj: Događaj, ocjena: Integer) u klasi RegistrovaniKorisnik gdje bi se onemogućilo recenziranje događaja od strane korisnika koji nije prisustvovao istom. Da bismo to postigli dodat ćemo interfejs IRecenziraj i klasu ProxyRecenziraj koja kroz metodu autentifikacija(događaj: Događaj, korisnik: RegistrovaniKorisnik) provjerava da li je dati korisnik prisustvovao događaju.

6. COMPOSITE PATERN

Pošto postoje različiti načini prijave na sistem za korisnika i vlasnika objekta, Composite patern bismo iskoristili tako što bismo napravili interfejs IPrijavi, te metodu prijava(korisničkolme:String, lozinka:String) koja bi se različito implementirala u klasi RegistrovaniKorisnik i VlasnikObjekta.

7. FLYWEIGHT PATERN

- Pošto korisnik ima mogućnost postavljanja profilne slike, u slučaju da je ne želi postaviti, koristit ćemo unaprijed zadalu sliku sa našeg sistema, tj. avatar. Na taj način će se koristiti ista slika za sve korisnike koji nisu postavili vlastitu, čime smanjujemo korištenje memorijskih resursa. Ovo ćemo postići kreiranjem interfejsa ISlika koji vraća slike svih korisnika. Postavljena slika korisnika će biti u klasi Slika, a avatar u klasi Avatar. Klase Slika i Avatar implementiraju interfejs ISlika.
- Ovaj patern može pomoći u slučaju kreiranja događaja koji ima iste ili slične osobine kao ranije kreiran događaj, npr. ako novi događaj ima isto dobro ograničenje, napomene, opis i tip, a drugačiji termin, možemo iskoristiti flyweight i na taj način smanjiti potrebu za memorijom.

KREACIJSKI PATERNI

1. SINGLETON PATERN

- U našem sistemu login je potreban da bi se koristila većina funkcionalnosti. Ovaj patern se koristi kada se odvija više logiranja, pa možemo dodati novu klasu LogSingleton koja će vršiti logiranje u sistemu na najefikasniji mogući način. Ona će sadržavati privatni staticki atribut, privatni konstruktor i gettere.
- Također, može se iskoristiti klasa ErrorSingleton kao error manager koja će upravljati svim errorima umjesto da kreiramo više error manager-a.

2. PROTOTYPE PATERN

Prepostavljajući da će neki atributi kao što su tipDogađaja, opisDogađaja, posebneNapomene i dobroOgraničenje biti isti za većinu događaja, pri kreiranju novog događaja trebali bismo svaki put prolaziti kroz sve te atribute i kopirati njihove vrijednosti u novi objekat. Zbog toga možemo napraviti interfejs IDogađaj koji bi nam omogućio da kloniramo instance Dogadaj klase.

3. FACTORY METHOD PATERN

U slučaju da imamo više tipova notifikacija, patern bi mogao biti primjenjen na način da postoji FactoryKlase koja sadrži metodu Factory() koja na osnovu proslijeđenih podataka o događaju kreira odgovarajuće notifikacije.

4. ABSTRACT FACTORY PATERN

Ovaj patern omogućava da se kreiraju familije povezanih objekata. Mi ne vidimo konkretnu primjenu ovog paterna u našem sistemu jer se nigdje ne radi sa većim brojem srodnih objekata/klasa.

5. BUILDER PATERN

Ovaj patern možemo iskoristiti pri kreiranju događaja pri čemu bismo u par koraka razdvojili informacije o događaju tako da bismo ih ponovo mogli koristiti za kreiranje nekog novog događaja. Za implementaciju ovog paterna moramo imati Klijent klasu koja će pozivati Direktor klasu u kojoj su neophodne sekvene za stvaranje objekta. Zatim bismo imali interfejs IKreatorDogađaja koja definiše specifične metode potrebne za kreiranje događaja. Zatim KreatorDogađaja klasu koju poziva Direktor i koja stvarno kreira događaj, dok nam Produkt klasu čini klasa Događaj,

PATERNI PONAŠANJA

1. **STRATEGY PATERN**

Koristit ćemo ovaj patern u RezervacijaDogađajaController-u u kojem ćemo implementirati metodu Sortiranje(tipSortiranja: int). Implementirat ćemo interfejs ISortiranje kojeg će nasljeđivati klase SortirajPoOgraničenju(), SortirajPoDatumu(), SortirajPoTipu().

2. **STATE PATERN**

Ovaj patern možemo iskoristiti prilikom različitih načina prikazivanja događaja, zavisno od toga da li je korisnik prijavljen na sistem ili nije, tj. neprijavljeni korisnik može pregledati događaje, ali ne rezervisati ih, dok prijavljeni ima mogućnost rezervacije. Implementirat ćemo interfejs IStanje koji će prikazivati odgovarajući pogled zavisno od toga da li je korisnik prijavljen ili nije. Zbog toga implementirali smo još dva pogleda PregledPrijavljenogView() i PregledNeprijavljenogView().

3. **TEMPLATE METHOD PATERN**

Ovaj patern služi za omogućavanje izmjene ponašanja u jednom ili više dijelova. Najčešće se primjenjuje kada se za neki kompleksni algoritam uvijek trebaju izvršiti isti koraci, ali pojedinačne korake moguće je izvršiti na različite načine. Naš sistem je relativno jednostavan pa nismo vidjeli konkretnu primjenu ovog paterna.

4. **ITERATOR PATERN**

Iterator patern možemo iskoristiti prilikom korisnikovog pregleda događaja. Korisnik može izabrati na koji način će se prikazati događaji, npr. slučajnim rasporedom, prema datumu održavanja itd.

5. **OBSERVER PATERN**

Upotreba ovog paterna bi bila da korisnik dobije notifikaciju o nadolazećim događajima iz njegove grupe interesovanja.

Uloge su sljedeće:

- Subject - vlasnik objekta
- IObservable – korisnik
- Update – prijem notifikacije
- Notify – notifikacija koju dobiju svi korisnici iz iste grupe interesovanja
- State – događaj

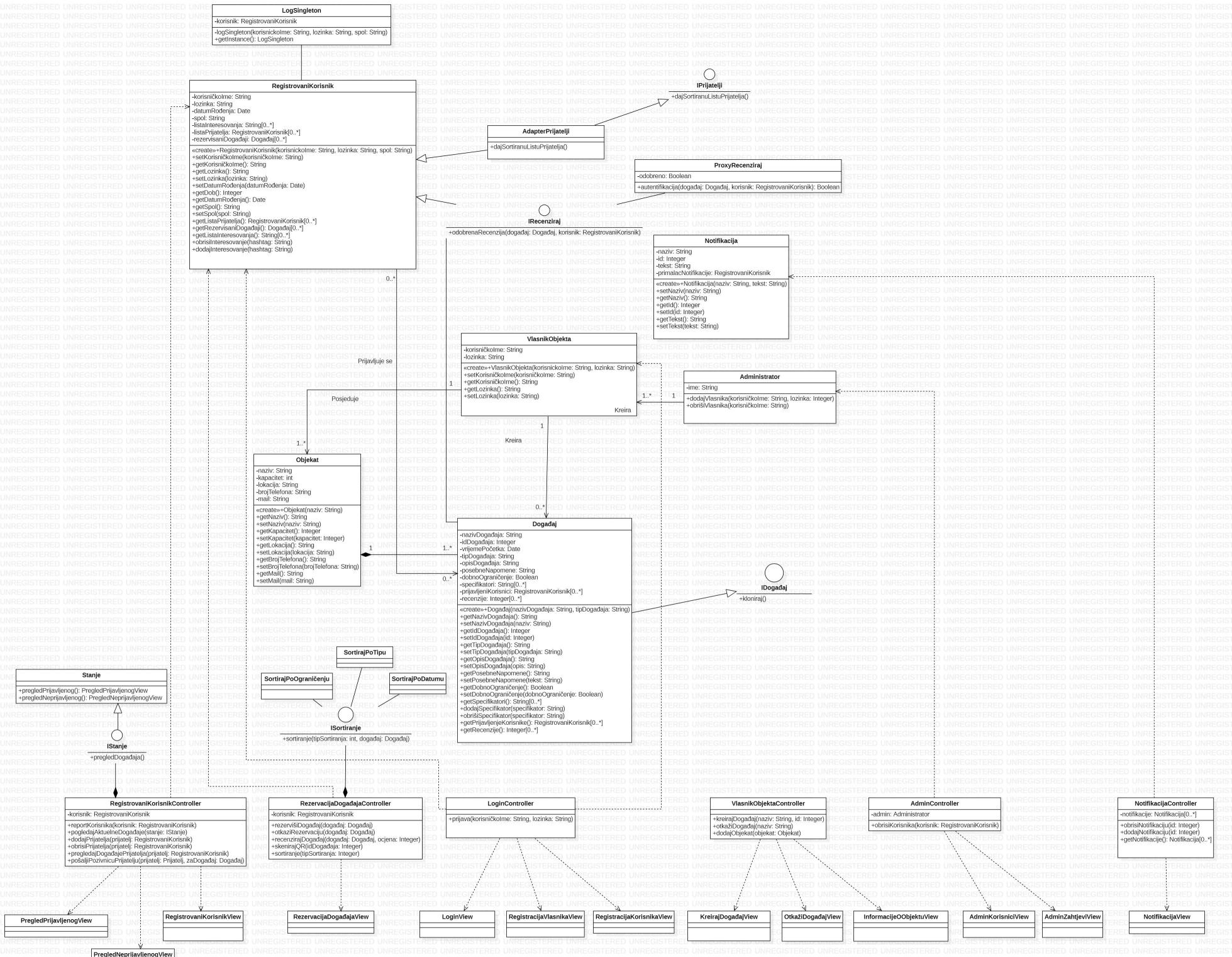
6. **CHAIN OF RESPONSIBILITY PATERN**

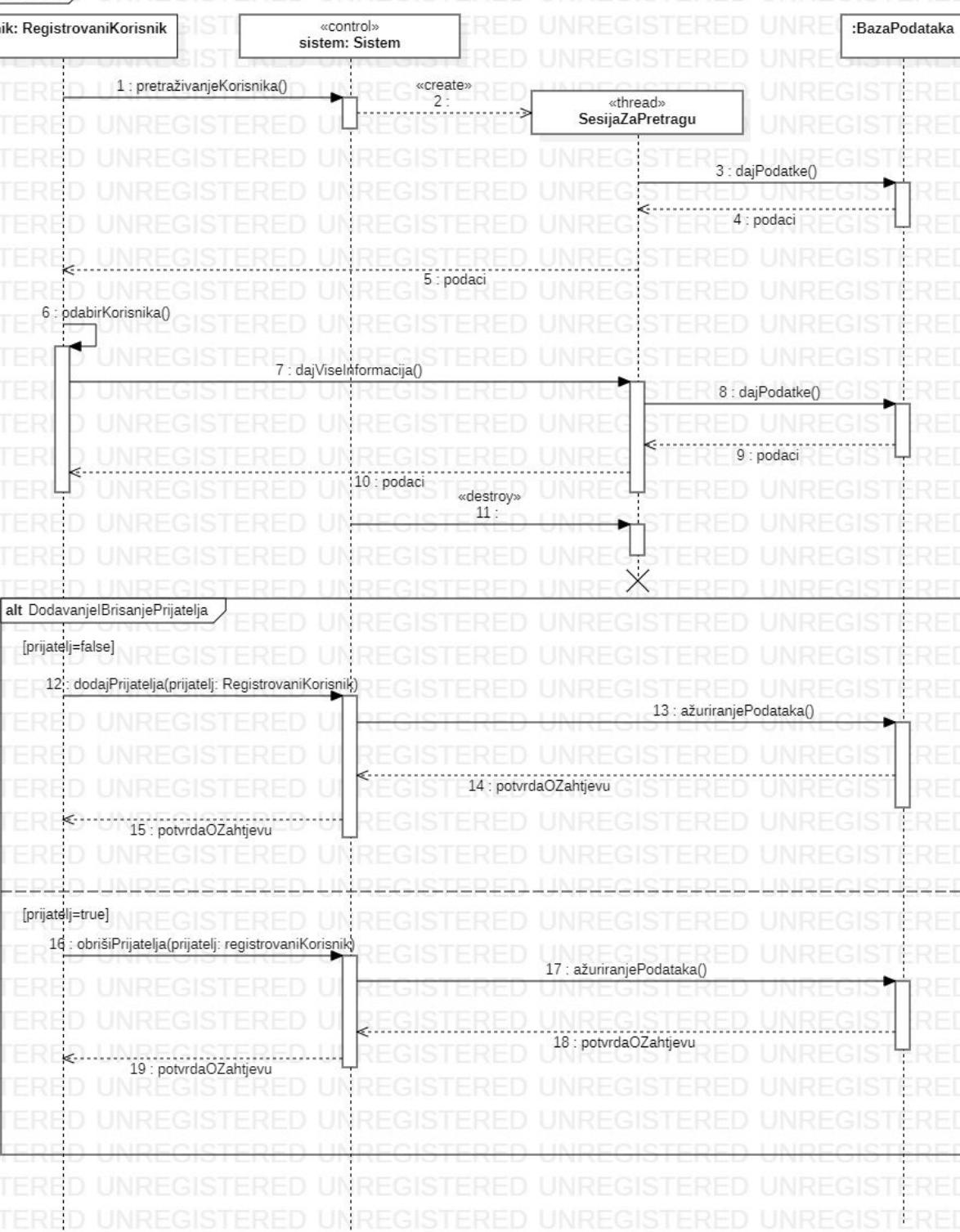
Chain of responsibility patern možemo iskoristiti pri rezervaciji događaja kako bismo bili sigurni da rezervaciju vrši registrovani korisnik te da ispunjava sve preduslove za

prisustvovanje događaju. U našem primjeru handleri vrše procesiranje podataka i odlučuju da li će zahtjev biti proslijeden dalje. Pretpostavljajući da će korisnik ispunjavati sve uslove, handleri će uspješno završiti procesiranje te će proces završiti uspješno rezervisanim događajem.

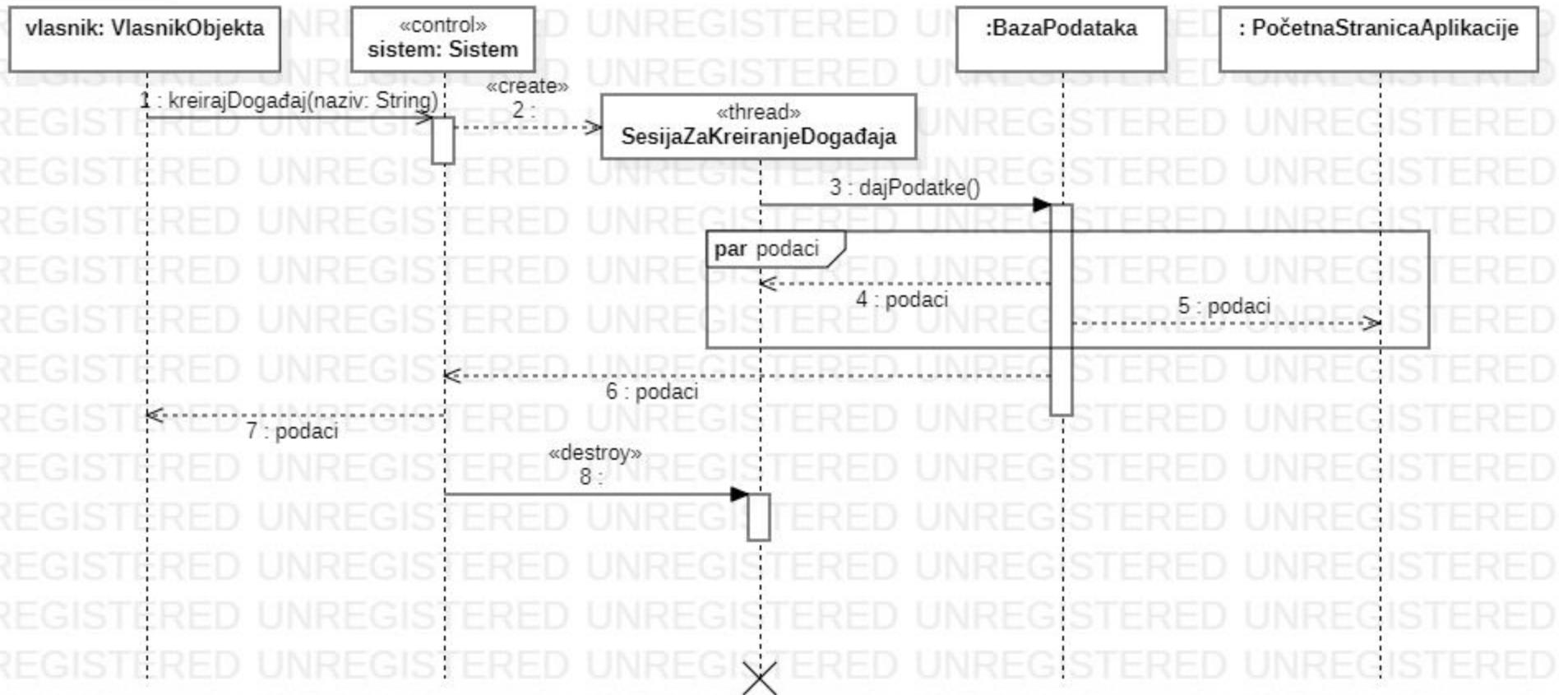
7. MEDIATOR PATERN

Mediator patern možemo iskoristiti pri recenziji događaja pri čemu ćemo implementirati interfejs IMedijator koji će provjeravati da li je korisnik prisustvovao događaju putem metode Provjeri(). Pošto klasa RegistrovaniKorisnik može ostavljati recenzije njoj ćemo dodati atribut tipa IMedijator.





sd KreiranjeDogađaja



sd PoništavanjeDogadaja

