



Teatar 18

Specifikacija projekta

Specifikacija projekta

1. Osnovne informacije o sistemu

Naziv teme: Teatar18

Logo:



Naziv tima: Grupa5-Tim18

Nastavna grupa: Grupa 5

Link na repozitorij tima: <https://github.com/OOAD-2023-2024/grupa5-tim18>

Članovi tima:

1. Borovina Lamija, 19356
2. Gutošić Merjem, 19378
3. Jozić Ivona, 19357
4. Kurtović Esma, 19358

Namjena sistema:

Opisati sistem i njegovu namjenu sa maksimalno sedam rečenica. U okviru ovog polja potrebno je objasniti šta sistem treba raditi na apstraktnom nivou, bez detaljnog objašnjavanja pojedinačnih funkcionalnosti i načina razlikovanja aktera sistema (što je predmet daljih poglavlja).

Cilj zamišljenog sistema jeste kreiranje web aplikacije za poslovanje pozorišta. Servis će pružati jednostavnu i intuitivnu uslugu kako korisnicima, tako i uposlenicima. Omogućavat će rezervaciju i kupovinu karata uz mogućnost odabira željenih sjedišta, pregled repertoara i aktuelnih informacija, mogućnost postavljanja pitanja i direktnu komunikaciju s ustanovom. Planirano je da korisnici posjeduju korisnički račun, uvezan s njihovom mail adresom.

2. Funkcionalnosti (poslovni procesi) sistema

Opisati 6 do 8 najznačajnijih funkcionalnosti sistema (u zavisnosti od broja članova u timu). Funkcionalnosti sistema predstavljaju usluge koje sistem pruža korisnicima. Sve funkcionalnosti pripadaju nekoj od različitih vrsta:

- Usluga sistema - u svrhu ostvarivanja krajnje usluge sistema,
- Perzistencija podataka (CRUD operacije)
- Asinhrona operacija - operacije koje koriste principe asinhronne obrade zahtjeva
- Operacija sa specifičnim algoritmom obrade - operacije koje koriste specifične algoritme obrade podataka,
- Korištenje vanjskog uređaja - operacije u kojima se vrši korištenje vanjskih uređaja.

Neophodno je navesti barem po jednu funkcionalnost svake od različitih vrsta.

1) **Naziv funkcionalnosti:** Kreiranje korisničkog računa

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

Za korištenje funkcionalnosti sistema koje se tiču pojedinca, poput rezervacije ili kupovine karata, potrebno je da korisnik kreira korisnički račun. Račun je vezan za korisnikovu email adresu. Kreirani račun mora biti verifikovan od strane administratora sistema.

2) **Naziv funkcionalnosti:** Rezervacija i kupovina karata

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

Ova funkcionalnost omogućava rezervaciju sjedišta za željenu predstavu. Nakon rezervacije, korisniku se daje mogućnost kupovine istih. U slučaju odluke plaćanja rezervisanih karata na licu mjesta, preuzimanje je potrebno izvršiti do dva dana prije predstave, u suprotnom se rezervacija automatski otkazuje. Rezervacija i kupovina su validne nakon potvrde od strane zaposlenika ustanove.

3) **Naziv funkcionalnosti:** Otkazivanje rezervacije karata

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

Pomoću svog korisničkog računa i podataka o rezervaciji, moguće je i otkazati rezervaciju karata, čime one postaju dostupne drugima. Zaposlenik potvrđuje i otkazivanje rezervacije.

4) **Naziv funkcionalnosti:** Ocjena predstave

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

Nakon završene predstave, na osnovu podataka o kupovini karata za istu, vezanih za korisnički račun, je moguće ocijeniti pogledanu predstavu.

5) **Naziv funkcionalnosti:** Preporuka predstava

Vrsta funkcionalnosti: Operacija sa specifičnim algoritmom obrade

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

Zahvaljujući prethodno navedenoj funkcionalnosti, shodno ocjenama različitih korisnika se u aplikaciji daju preporuke za dobro ocijenjene predstave.

6) **Naziv funkcionalnosti:** Repertoar

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

U okviru ove aplikacije je moguće pregledati aktuelni repertoar predstava. Repertoar se ažurira mjesečno.

7) **Naziv funkcionalnosti:** Newsletter

Vrsta funkcionalnosti: Korištenje vanjskog uređaja

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

Zahvaljujući ovoj funkcionalnosti, korisnici sistema će preko mail adrese biti obaviještavani u slučaju nepredviđenih okolnosti kao što je otkazivanje predstave, kao i o drugim aktuelnim informacijama.

8) **Naziv funkcionalnosti:** Pitanja

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

Opisati način ostvarivanja funkcionalnosti sa maksimalno pet rečenica.

Korisnicima sistema će biti omogućeno postavljanje pitanja kroz formular prema ustanovi.

3. Akteri sistema

Potrebno je navesti najmanje tri aktera sistema.

Vrste aktera:

- Korisnik sistema
- Zaposlenik sistema
- Administrator

Neophodno je navesti barem po jednog aktera za svaku od različitih vrsta.

Korisnici usluga sistema

a) **Naziv aktera: Korisnik sistema**

Vrsta aktera: Korisnik usluge

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
1) Kreiranje korisničkog računa	Mogućnost uređivanja
2) Rezervacija i kupovina karata	Mogućnost uređivanja
3) Otkazivanje rezervacije karata	Mogućnost uređivanja
4) Ocjena predstave	Mogućnost uređivanja
6) Repertoar	Mogućnost pregleda
7) Newsletter	Mogućnost pregleda
8) Pitanja	Mogućnost uređivanja

b) **Naziv aktera: Zaposlenik**

Vrsta aktera: Zaposlenik sistema

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
2) Rezervacija i kupovina karata	Mogućnost uređivanja
3) Otkazivanje rezervacije karata	Mogućnost uređivanja
6) Repertoar	Mogućnost uređivanja
7) Newsletter	Mogućnost uređivanja
8) Pitanja	Mogućnost uređivanja

c) **Naziv aktera: Administrator**

Vrsta aktera: Administrator

Funkcionalnosti u kojima akter učestvuje:

Način učešća:

- *Mogućnost pregleda*
- *Mogućnost uređivanja*

Funkcionalnost sistema	Način učešća
1) Kreiranje korisničkog računa	Mogućnost uređivanja
2) Rezervacija i kupovina karata	Mogućnost uređivanja
3) Otkazivanje rezervacije karata	Mogućnost uređivanja
4) Ocjena predstave	Mogućnost uređivanja
5) Preporuka predstava	Mogućnost uređivanja
6) Repertoar	Mogućnost uređivanja



7) Newsletter	Mogućnost uređivanja
8) Pitanja	Mogućnost uređivanja

4. Nefunkcionalni zahtjevi sistema

Opisati najmanje tri najznačajnija nefunkcionalna zahtjeva sistema. Nefunkcionalni zahtjevi predstavljaju ograničenja koja sistem mora zadovoljiti kako bi mogao ispravno obavljati svoje funkcionalnosti. Validacije polja za unos vrijednosti ne predstavljaju nefunkcionalne zahtjeve.

1) **Naziv nefunkcionalnog zahtjeva:** Verifikacija korisničkog računa

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

Potvrda o verifikaciji računa treba stići na mail adresu korisnika u roku od 24h.

2) **Naziv nefunkcionalnog zahtjeva:** Izgled početne stranice

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

Na početnoj stranici se nalazi izbornik funkcionalnosti. Ukoliko je za određenu funkcionalnost neophodno posjedovanje korisničkog računa, tražit će se prijava na sistem.

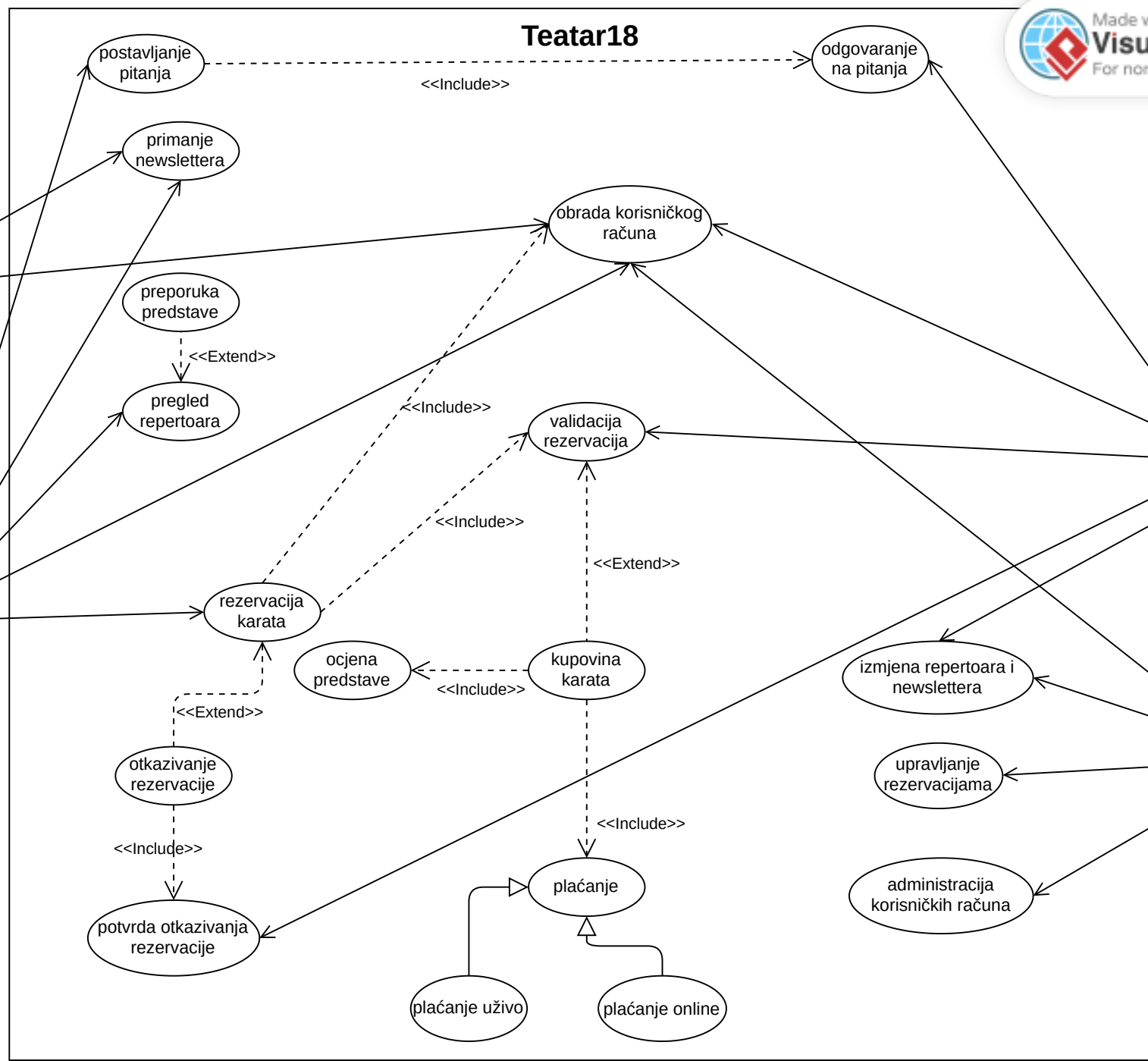
3) **Naziv nefunkcionalnog zahtjeva:** Upravljanje greškama

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

U slučaju internih grešaka sistema, bit će ispisana poruka o grešci.

*Dijagram i
scenariji slučajeva
upotrebe*



e-mail
servis

Korisnik
sistema

Zaposlenik

Administrator

Teatar18 – Slučajevi upotrebe i scenariji

Naziv slučaja upotrebe	Kreiranje korisničkog računa
Opis slučaja upotrebe	Ovaj servis omogućava korisnicima kreiranje računa pomoću kojeg dalje imaju mogućnosti rezervacije/kupovine karata, ocjenjivanja predstava itd.
Vezani zahtjevi	/
Preduslovi	Račun nije ranije kreiran
Posljedice – uspješan završetak	Uspješno kreiranje računa
Posljedice – neuspješan završetak	U slučaju postojanja navedenog računa ili neispravne e-mail adrese, ispisuje se greška o neuspjelom kreiranju
Primarni akteri	Korisnik
Ostali akteri	Administrator, e-mail servis
Glavni tok	Korisnik na stranici najprije bira opciju kreiranja računa ili mu se ista nudi prilikom pokušaja izvršavanja operacije koja zahtijeva njegovo posjedovanje. Kroz formular unosi osnovne podatke o sebi nakon čega se račun uspješno kreira ili se ispisuje poruka o neuspjelom kreiranju ukoliko je došlo do pogreške.
Alternative/proširenja	/

Tok događaja 1 – Uspješan završetak

Korisnik	Administrator	E-mail servis
1 Odabire opciju kreiranja računa		
2 Unosi podatke o sebi		
	3 Potvrđuje uspješno kreiranje računa	

Tok događaja 2 – Neuspješan završetak

Korisnik	Administrator	E-mail servis
1 Odabire opciju kreiranja računa		
2 Unosi podatke o sebi		
	3 Prijavljuje grešku ukoliko račun već postoji	3 Prijavljuje grešku ukoliko je e-mail adresa neispravna
	4 Obavještava korisnika o neuspjelom kreiranju korisničkog računa	

Naziv slučaja upotrebe:	Pregled repertoara
Opis slučaja upotrebe:	Korisnik sistema ima mogućnost pregleda repertoara prije same kupovine i rezervacije karata
Vezani zahtjevi:	/
Preduslovi:	Pristup sistemu
Posljedice - uspješan završetak:	/
Posljedice – neuspješan završetak:	/
Primarni akteri:	Korisnik sistema
Ostali akteri:	/
Glavni tok:	/
Alternative proširenja:	/

Naziv slučaja upotrebe	Preporuka predstava
Opis slučaja upotrebe	Korisnik sistema prilikom pristupa aplikaciji uz pregled repertoara ima i mogućnost da vidi koje su najbolje ocijenjene predstave od strane gledalaca.
Vezani zahtjevi	/
Preduslovi	Pristup sistemu
Posljedice – uspješan završetak	/
Posljedice – neuspješan završetak	/
Primarni akteri	Korisnik sistema
Ostali akteri	/
Glavni tok	/
Alternative/proširenja	/

Naziv slučaja upotrebe	Kreiranje rezervacije
Opis slučaja upotrebe	Korisnicima koji posjeduju korisnički račun je omogućeno kreiranje rezervacija za predstave.
Vezani zahtjevi	/
Preduslovi	Posjedovanje korisničkog računa
Posljedice – uspješan završetak	Uspješno kreiranje rezervacije
Posljedice – neuspješan završetak	U slučaju greške u sistemu ili sa bazom podataka, rezervacija neće biti kreirana o čemu će korisnik biti obaviješten
Primarni akteri	Korisnik
Ostali akteri	Administrator
Glavni tok	Korisnik nakon prijave sa svojim korisničkom računom ima mogućnost kreiranja rezervacije za željenu predstavu. Nakon odabira predstave, nudi mu se mogućnost odabira sjedišta. Pri uspješnom kreiranju rezervacije, dobiva potvrdu na e-mail adresu. Potom može odabrati opciju kupovine. Ukoliko je došlo do interne pogreške i rezervacija nije kreirana, ispisuje se prikladna poruka.
Alternative/proširenja	/

Tok događaja 1 – Uspješan završetak

Korisnik	Administrator
1 Odabire opciju kreiranja rezervacije	
2 Odabire predstavu	
3 Odabire sjedišta	
	4 Potvrđuje rezervaciju i šalje potvrdu korisniku

Tok događaja 2 – Neuspješan završetak

Korisnik	Administrator
1 Odabire opciju kreiranja rezervacije	
2 Odabire predstavu	
3 Odabire sjedišta	

	4 Obavještava korisnika o neuspjelom kreiranju rezervacije zbog pojave greške u sistemu
--	---

Naziv slučaja upotrebe	Kupovina i plaćanje karata
Opis slučaja upotrebe	Nakon rezervacije, korisniku se daje mogućnost kupovine karata, pri čemu se plaćanje može vršiti uživo ili online.
Vezani zahtjevi	/
Preduslovi	Posjedovanje korisničkog računa. Prethodna rezervacija karata te validacija rezervacije od strane uposlenika.
Posljedice – uspješan završetak	Na mail korisnika stiže potvrda o uspješnoj kupovini, kao i plaćanju u slučaju online plaćanja. Također, na njegovom računu ostaje zapamćena ta kupovina.
Posljedice – neuspješan završetak	Ispis poruke o neuspješnoj kupovini. Karte ostaju u rezervaciji. Mogućnost ponovnog pokušaja, ili odabira plaćanja uživo.
Primarni akteri	Korisnik sistema
Ostali akteri	/
Glavni tok	Korisnik sistema nakon rezervacije ima opciju kupovine karata. Sistem nudi opciju plaćanja uživo i online plaćanja. U slučaju uspješnog završetka, korisniku na mail stiže potvrda o kupovini. Ukoliko je odabrano i uspješno izvršeno plaćanje online, to je navedeno u potvrdi. S druge strane, ako je odabrano plaćanje uživo, navodi se rok do kada je isto potrebno izvršiti. Na korisničkom računu ostaje zapamćena ta kupovina. U slučaju neuspješnog ishoda online plaćanja, ispisuje se poruka o neuspjeloj kupovini. Karte tada ostaju u dijelu računa vezanom za rezervacije. Korisniku se nudi mogućnost ponovnog pokušaja, ili odabira drugog vida plaćanja.
Alternative/proširenja	/

Tok događaja 1.1 – Uspješan završetak

Korisnik	Sistem
	1 Korisniku se daje mogućnost kupovine prethodno rezervisanih karata.
2 Odabir opcije kupovine.	
	3 Nude se dvije opcije plaćanja - uživo i online.
4 Odabir opcije plaćanja uživo.	
	5 Slanje potvrde o uspješnoj kupovini na mail, koja uključuje i rok do kada je potrebno platiti i preuzeti karte. Ažuriranje korisničkog računa u skladu s tim.

Tok događaja 1.2 – Uspješan završetak

Korisnik	Sistem
	1 Korisniku se daje mogućnost kupovine prethodno rezervisanih karata.
2 Odabir opcije kupovine.	
	3 Nude se dvije opcije plaćanja - uživo i online.
4 Odabir opcije plaćanja online.	
	5 Vršiti se online plaćanje i verifikacija potrebnih podataka.
	6 Slanje potvrde o uspješnoj kupovini i plaćanju na mail. Ažuriranje korisničkog računa u skladu s tim.

Tok događaja 2 – Neuspješan završetak

Korisnik	Sistem
	1 Korisniku se daje mogućnost kupovine prethodno rezervisanih karata.
2 Odabir opcije kupovine.	

	3 Nude se dvije opcije plaćanja - uživo i online.
4 Odabir opcije online plaćanja.	
	5 Neuspješno plaćanje.
	6 Ispisuje se poruka o neuspješnoj kupovini. Karte ostaju rezervisane. Daje se mogućnost ponovnog pokušaja plaćanja.

Naziv slučaja upotrebe	Otkazivanje rezervacije
Opis slučaja upotrebe	U slučaju da korisnik želi otkazati rezervaciju za izabranu predstavu, sistem mu to omogućava nakon čega se broj slobodnih mjesta za predstavu povećava.
Vezani zahtjevi	/
Preduslovi	Posjedovanje korisničkog računa Izvršena rezervacija za predstavu
Posljedice – uspješan završetak	Potvrda o uspješnom otkazivanju. Broj slobodnih mjesta za predstavu u sistemu se povećava
Posljedice – neuspješan završetak	Korisnik dobiva informaciju o grešci i uputu da pokuša ponovno.
Primarni akteri	Korisnik sistema
Ostali akteri	Zaposlenik sistema, administrator sistema
Glavni tok	Korisnik želi otkazati rezervaciju i šalje zahtjev koji treba potvrditi uposlenik sistema. Ukoliko je zahtjev odobren, korisnik dobiva potvrdu o tome, dok u slučaju neuspješnog otkazivanja može pokušati ponovno ili jednostavno ne preuzeti karte na vrijeme nakon čega će rezervacija biti automatski poništena.
Alternative/proširenja	/

Tok događaja 1 – Uspješan završetak

Korisnik	Zaposlenik sistema	Administrator sistema
1 Korisnik podnio zahtjev za otkazivanje rezervacije		
	2 Zaposlenik odobrava zahtjev	
		3 Broj slobodnih mjesta za predstavu se povećava
4 Korisnik dobiva potvrdu o uspješno otkazanoj rezervaciji		

Tok događaja 2 – Neuspješan završetak

Korisnik	Zaposlenik sistema	Administrator sistema
1 Korisnik podnio zahtjev za otkazivanje rezervacije		
	2 Zaposlenik nije odobrio zahtjev ili je došlo do greške	
		3 Korisniku se šalje obavijest i preporuka novog pokušaja.
4 Korisnik zaprimio informaciju o neuspješnom otkazivanju		
5.1 Šalje zahtjev ponovno 5.2 Odustaje od slanja zahtjeva		
		6 Rezervacija se automatski poništava ako se karte ne preuzmu do dva dana prije predstave

Naziv slučaja upotrebe	Ocjenjivanje predstava
Opis slučaja upotrebe	Nakon odgledane predstave, korisnik sistema ima mogućnost ocijeniti istu, ali nije obavezan.
Vezani zahtjevi	/
Preduslovi	Kupljena karta za predstavu
Posljedice – uspješan završetak	/
Posljedice – neuspješan završetak	/
Primarni akteri	Korisnik sistema
Ostali akteri	Administrator sistema
Glavni tok	Nakon odgledane predstave, korisnik ima mogućnost dati joj ocjenu u skladu s ličnim preferencama. Data ocjena se pohranjuje u sistem, a zatim pozadinski algoritam vrši obradu podataka i eventualno uređuje listu s preporučenim predstavama. Kako korisnik nije obavezan ocijeniti odgledanu predstavu, samim tim ovaj slučaj upotrebe nema negativnih posljedica.
Alternative/proširenja	/

Tok događaja 1 – Uspješan završetak

Korisnik	Administrator sistema
1 Korisnik ocijenio predstavu	
	2 Kroz pozadinski algoritam se vrši obrada podataka i ažurira lista preporučenih predstava

Tok događaja 2 – Neuspješan završetak

Korisnik	Administrator sistema
1 Korisnik nije ocijenio predstavu	
	2 Pozadinskom algoritmu se ne šalju nikakvi podaci, pa nema potrebu za ažuriranjem liste preporučenih predstava

Naziv slučaja upotrebe	Postavljanje pitanja
Opis slučaja upotrebe	Pomoću ovog servisa, korisnik sistema može postaviti pitanje prema ustanovi.
Vezani zahtjevi	/
Preduslovi	Prijava putem korisničkog računa ili ostavljanje povratne mail adrese.
Posljedice – uspješan završetak	Potvrda o uspješno poslanom pitanju.
Posljedice – neuspješan završetak	Ispis poruke o grešci.
Primarni akteri	Korisnik sistema
Ostali akteri	/
Glavni tok	Korisnik sistema postavlja pitanje putem formulara. U slučaju da ima korisnički račun, može se prijaviti na njega, čime se pitanje veže za podatke s tog računa. U suprotnom, potrebno je ostaviti podatke poput mail adrese radi povratnih informacija. Konačno, u ovisnosti od uspješnosti akcije, ispisuje se poruka od strane sistema.
Alternative/proširenja	/

Tok događaja 1 – Uspješan završetak

Korisnik	Sistem
1 Pristup i popunjavanje formulara za pitanja. Eventualna prijava na korisnički račun.	
	2 Ispis poruke o uspješno izvršenoj akciji.

Tok događaja 2 – Neuspješan završetak

Korisnik	Sistem
1 Pristup i popunjavanje formulara za pitanja. Eventualna prijava na korisnički račun.	
	2 Ispis poruke o grešci.

Naziv slučaja upotrebe:	Odgovaranje na pitanja
Opis slučaja upotrebe:	Nakon što korisnik sistema postavi pitanje zaposlenik sistema na to pitanje odgovara.
Vezani zahtjevi:	/
Preduslovi:	Imati ovlasti zaposlenika
Posljedice - uspješan završetak:	Korisnik sistema dobija odgovor
Posljedice – neuspješan završetak:	Korisnik sistema ne dobija odgovor
Primarni akteri:	Zaposlenik sistema
Ostali akteri:	Korisnik sistema
Glavni tok:	Korisnici imaju priliku postaviti pitanje kroz formular. Nakon postavljanja pitanja zaposlenik sistema daje odgovor u vidu povratnog mail-a.
Alternative proširenja:	/

Tok događaja 1 – Uspješan završetak

Korisnik	Sistem
1 Postavljanje pitanja kroz formular.	
	2 Nakon što pitanje dođe do zaposlenika, zaposlenik daje odgovor.
3 Korisnik dobiva odgovor na mail	

Tok događaja 2 – Neuspješan završetak

Korisnik	Sistem
1 Postavljanje pitanja kroz formular	
	2 Pitanje nije došlo do sistema radi greške, ili zaposlenik nije odgovorio na postavljeno pitanje.

Naziv slučaja upotrebe	Primanje newslettera
Opis slučaja upotrebe	Korisnik ima mogućnost da dobiva nove informacije o repertoaru i predstavama na mail.

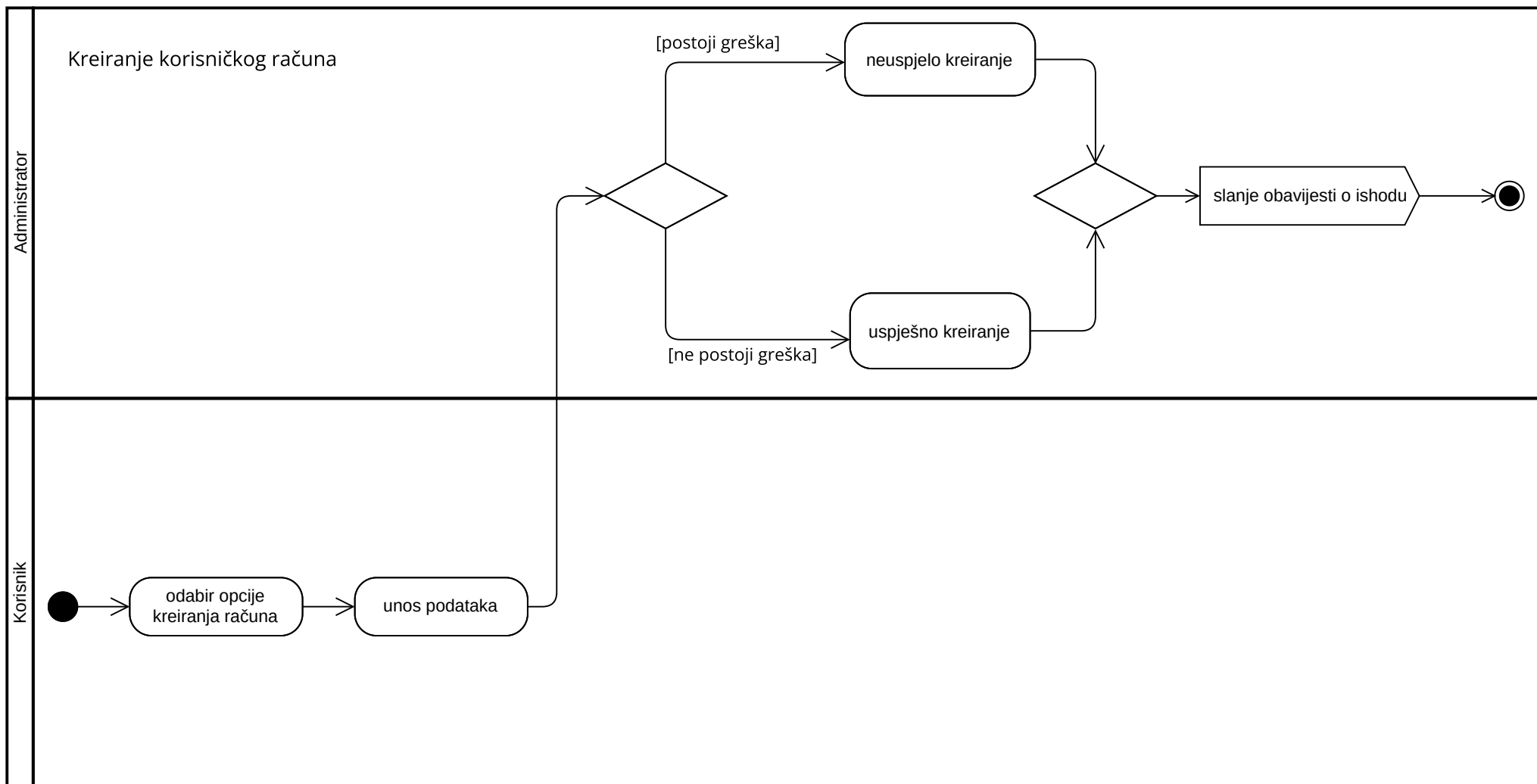
Vezani zahtjevi	/
Preduslovi	Posjedovanje korisničkog računa
Posljedice – uspješan završetak	/
Posljedice – neuspješan završetak	/
Primarni akteri	Korisnik sistema
Ostali akteri	Sistem
Glavni tok	Nakon kreiranja korisničkog računa koristeći svoju mail adresu, korisnik ima mogućnost da putem mail-a dobija nove informacije od interesa, vezane za izmjenu repertoara, otkazivanje predstava i slično.
Alternative/proširenja	/

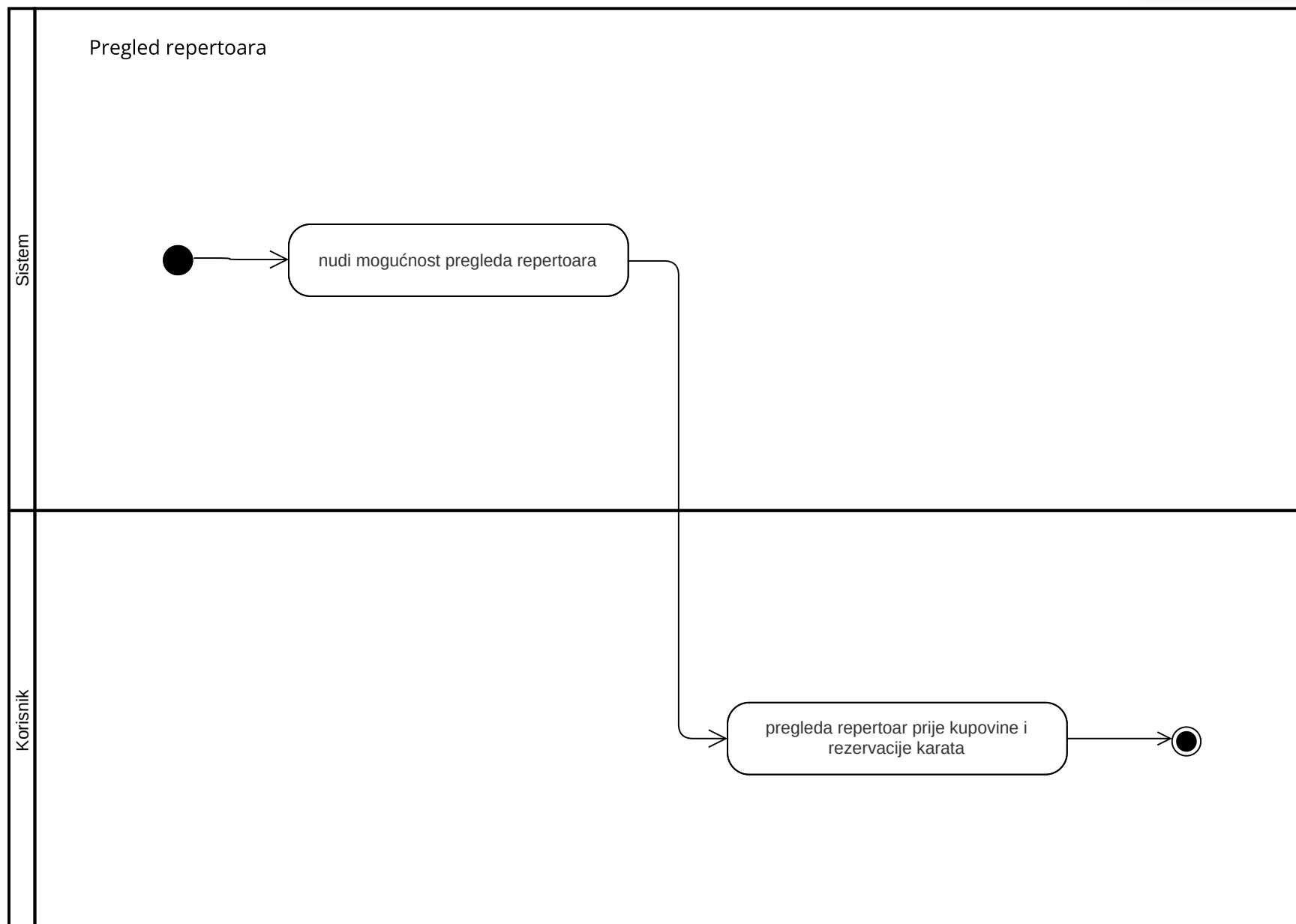
Tok događaja 1 – Uspješan završetak

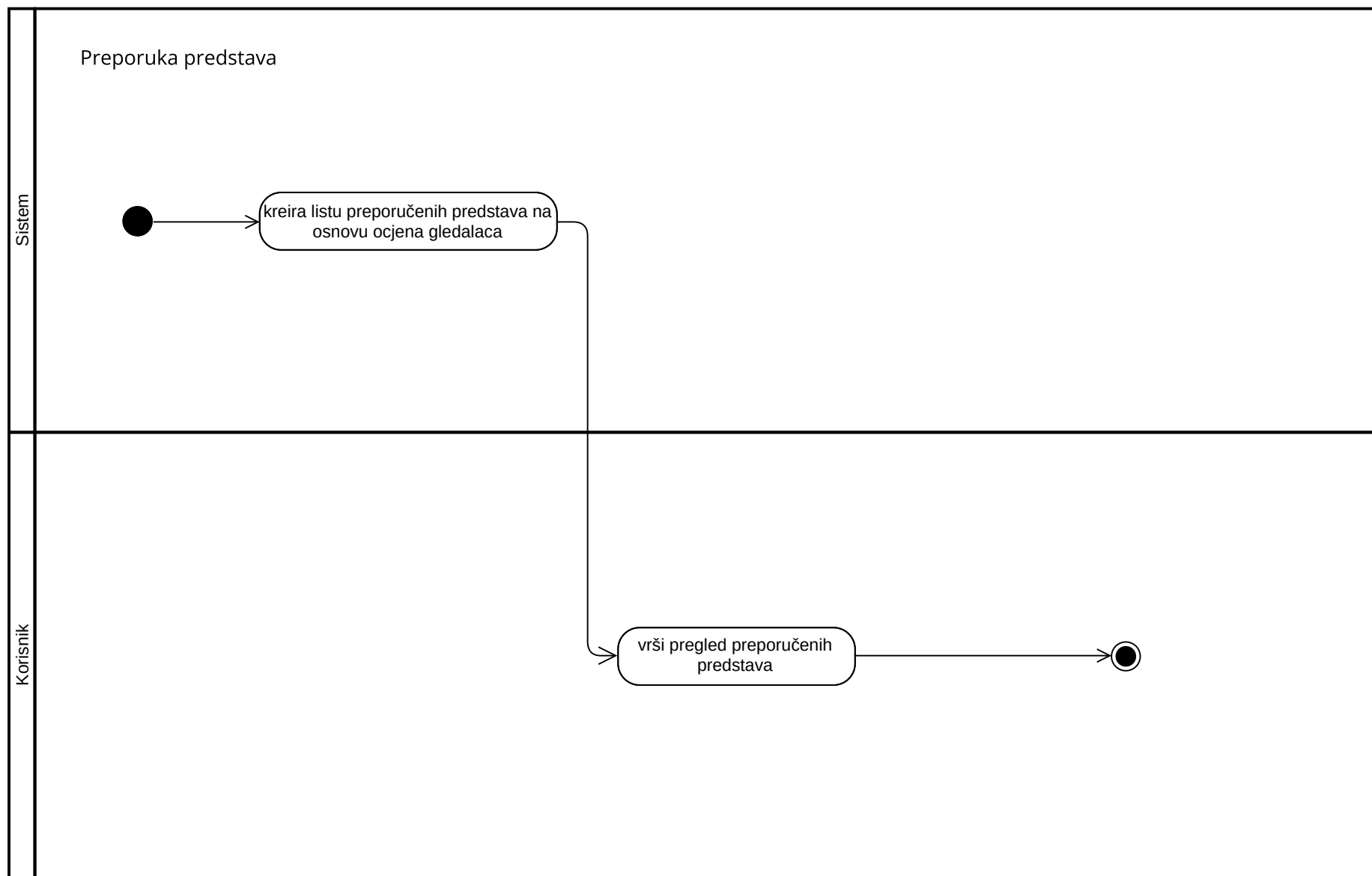
Korisnik	Sistem
1 Formiranje korisničkog računa	
	2 Slanje Newsletter-a
3 Dobivanje novih informacija na mail	

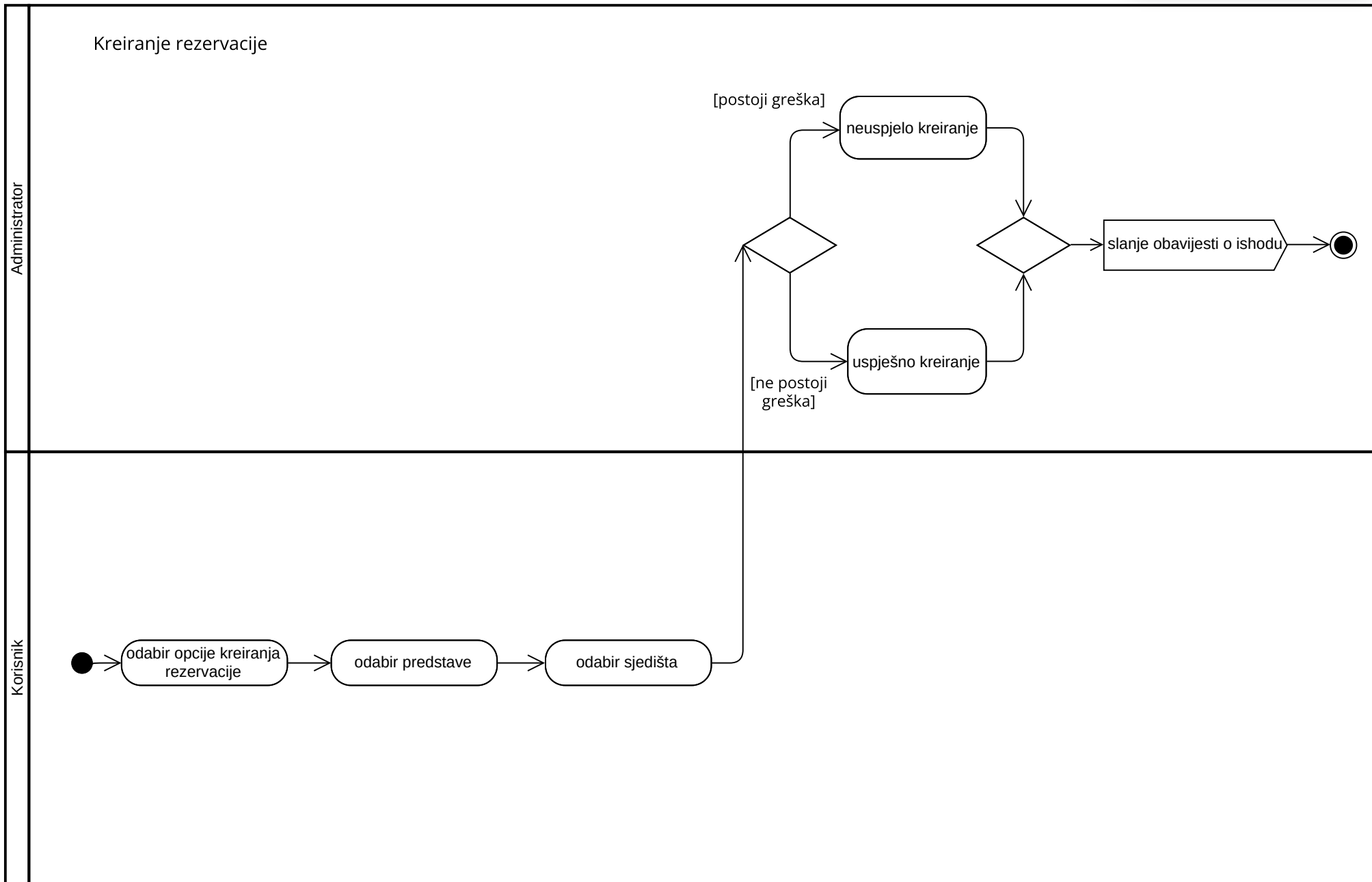
Naziv slučaja upotrebe:	Izmjena repertoara
Opis slučaja upotrebe:	Administrator i zaposlenik sistema bave se izmjenom i ažuriranjem repertoara.
Vezani zahtjevi:	/
Preduslovi:	Imati ovlasti zaposlenika ili administratora sistema
Posljedice - uspješan završetak:	Uspješno izmijenjen repertoar
Posljedice – neuspješan završetak:	Neuspješno izmijenjen repertoar
Primarni akteri:	Administrator, zaposlenik sistema
Ostali akteri:	/
Glavni tok:	Administrator ima pristup i mogućnost ažuriranja i izmjene repertoara, koje redovno obavlja u skladu sa informacijama koje dobija od zaposlenika sistema.
Alternative proširenja:	/

Dijagrami aktivnosti





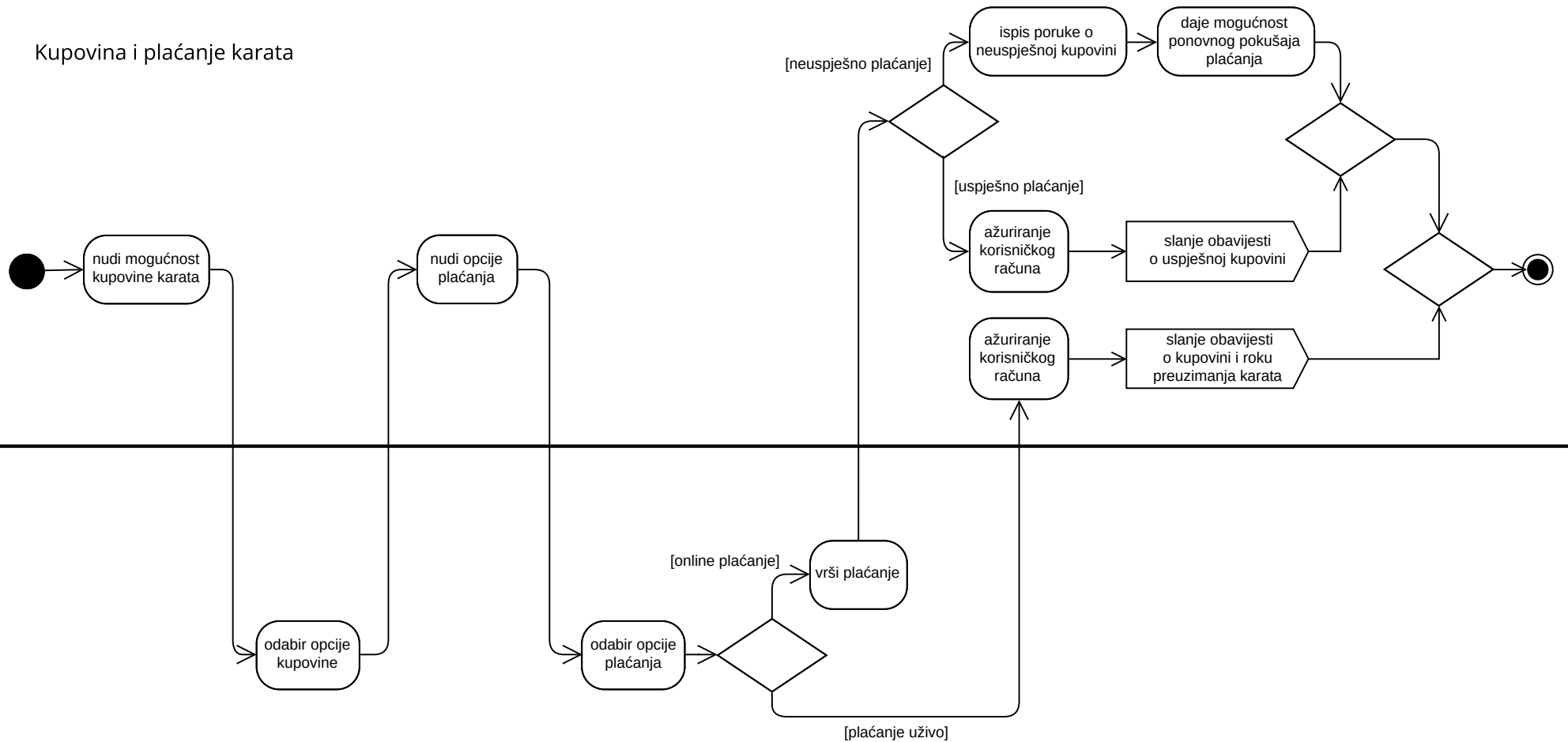


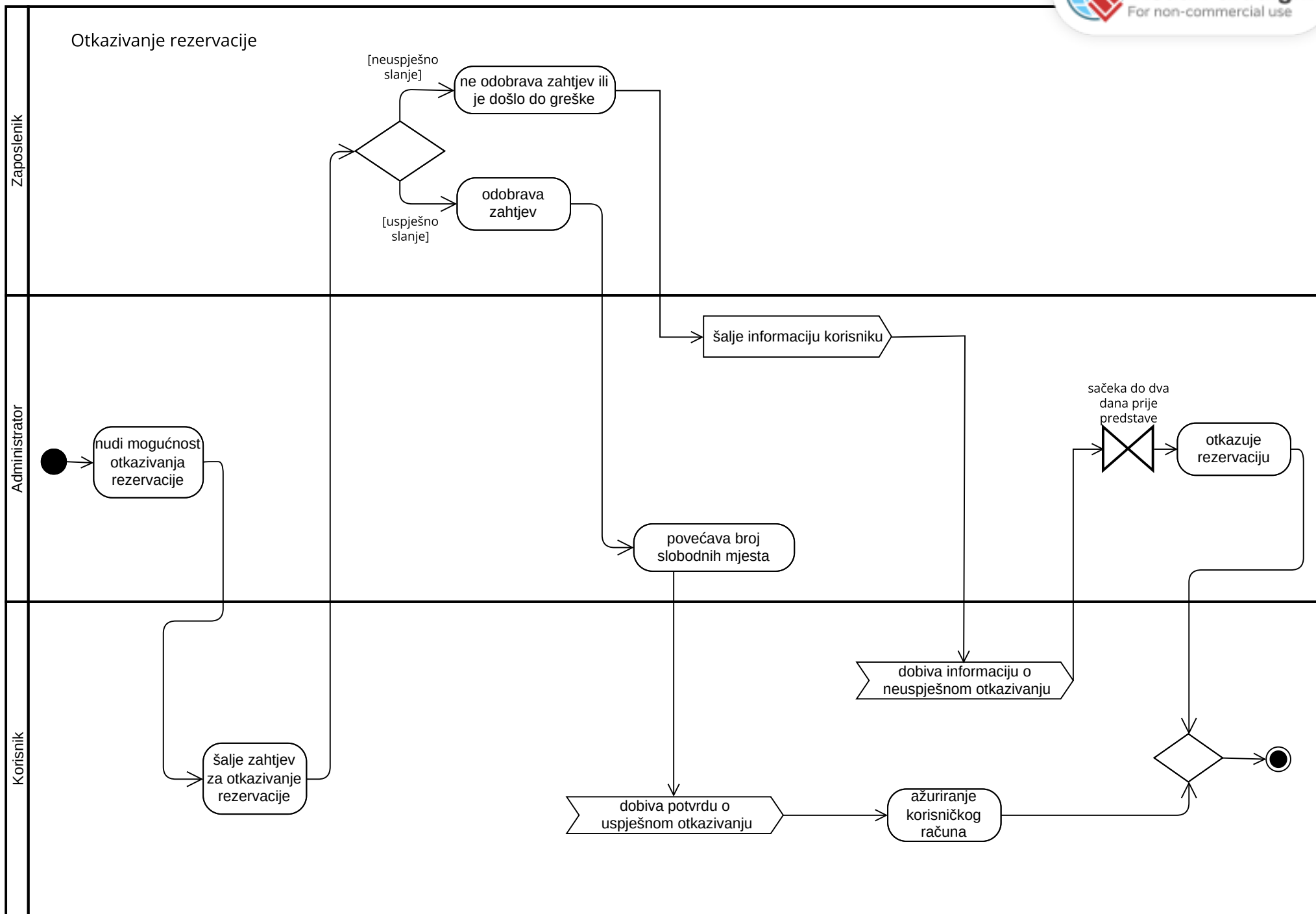


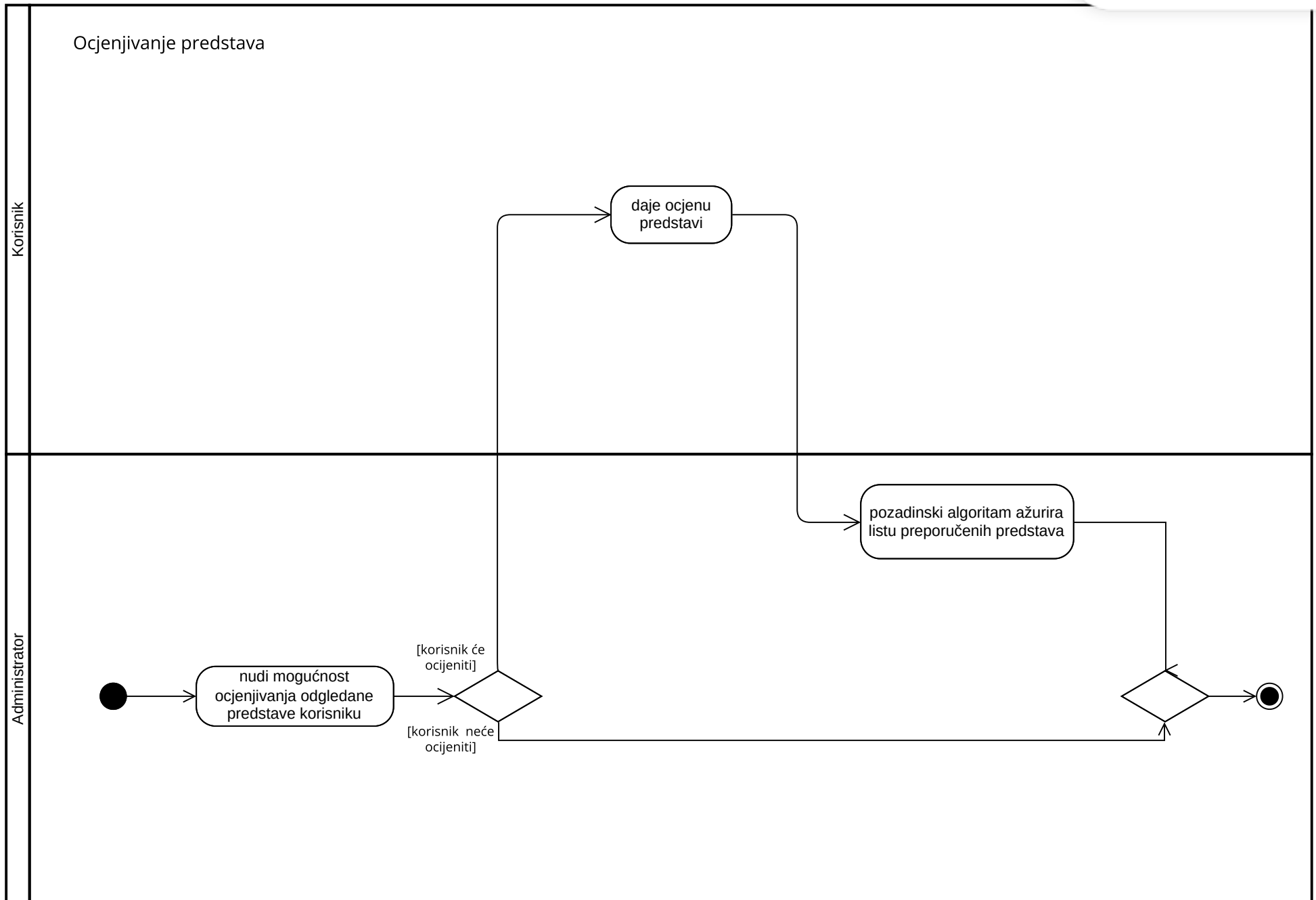
Kupovina i plaćanje karata

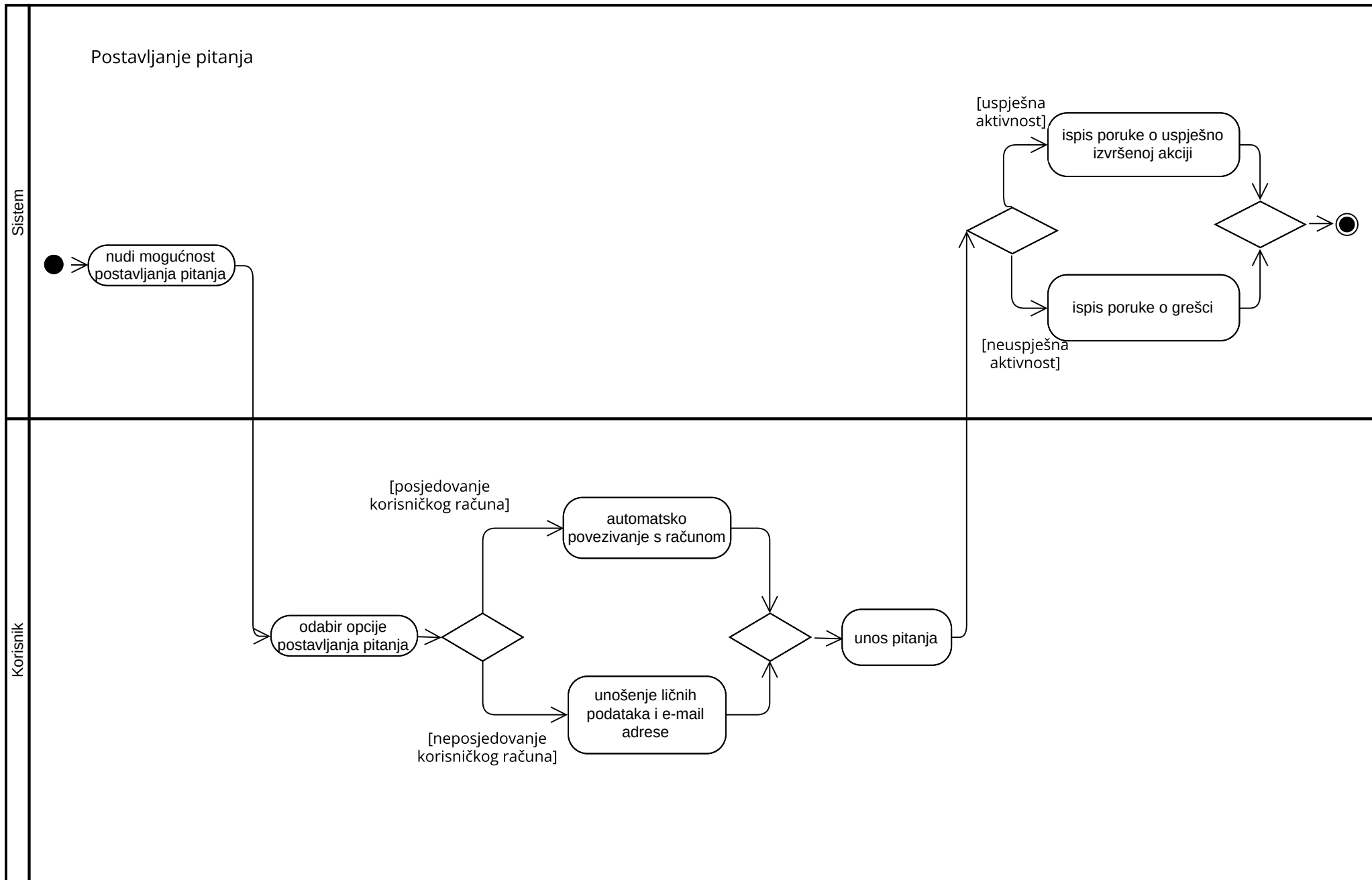
Sistem

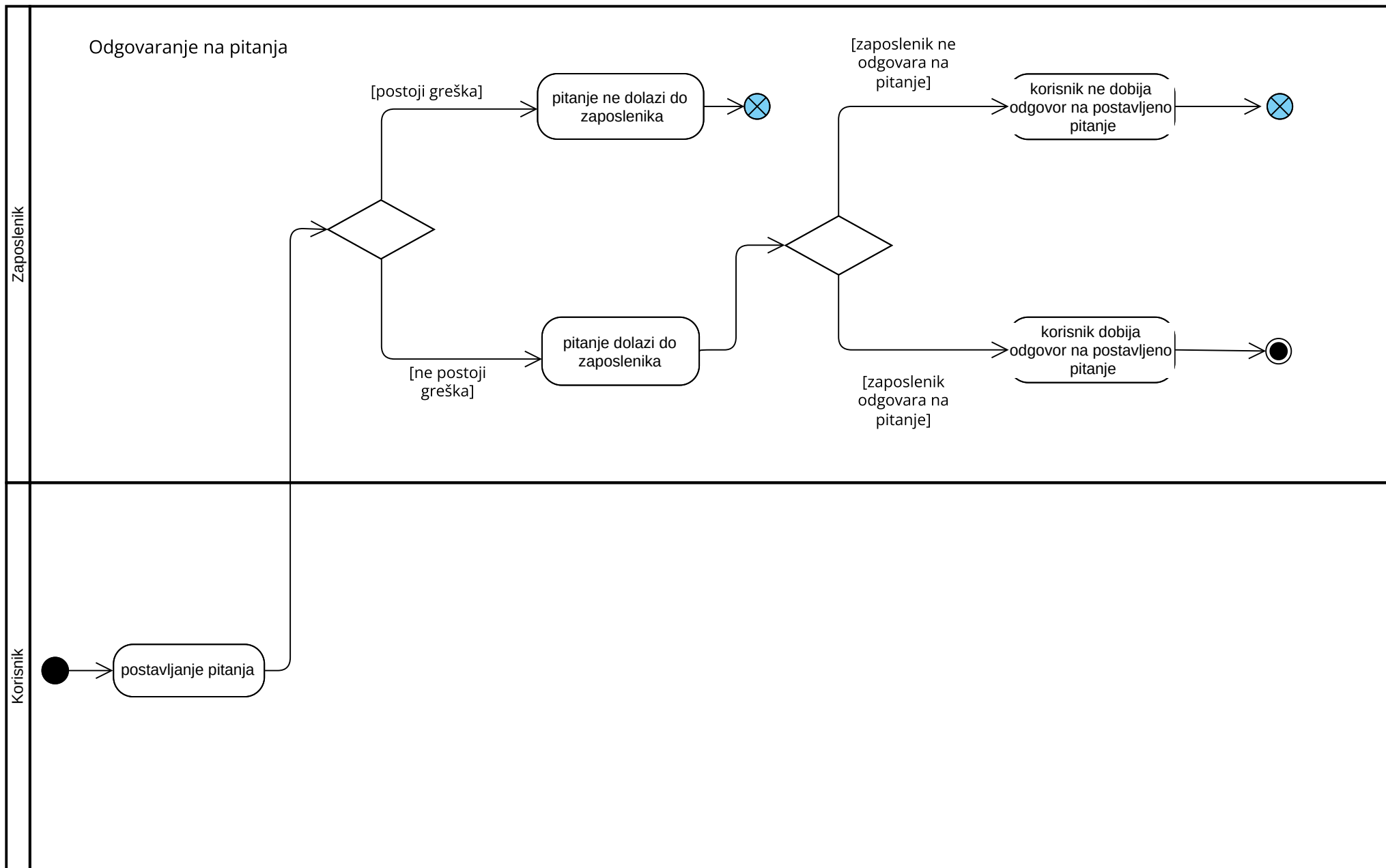
Korisnik

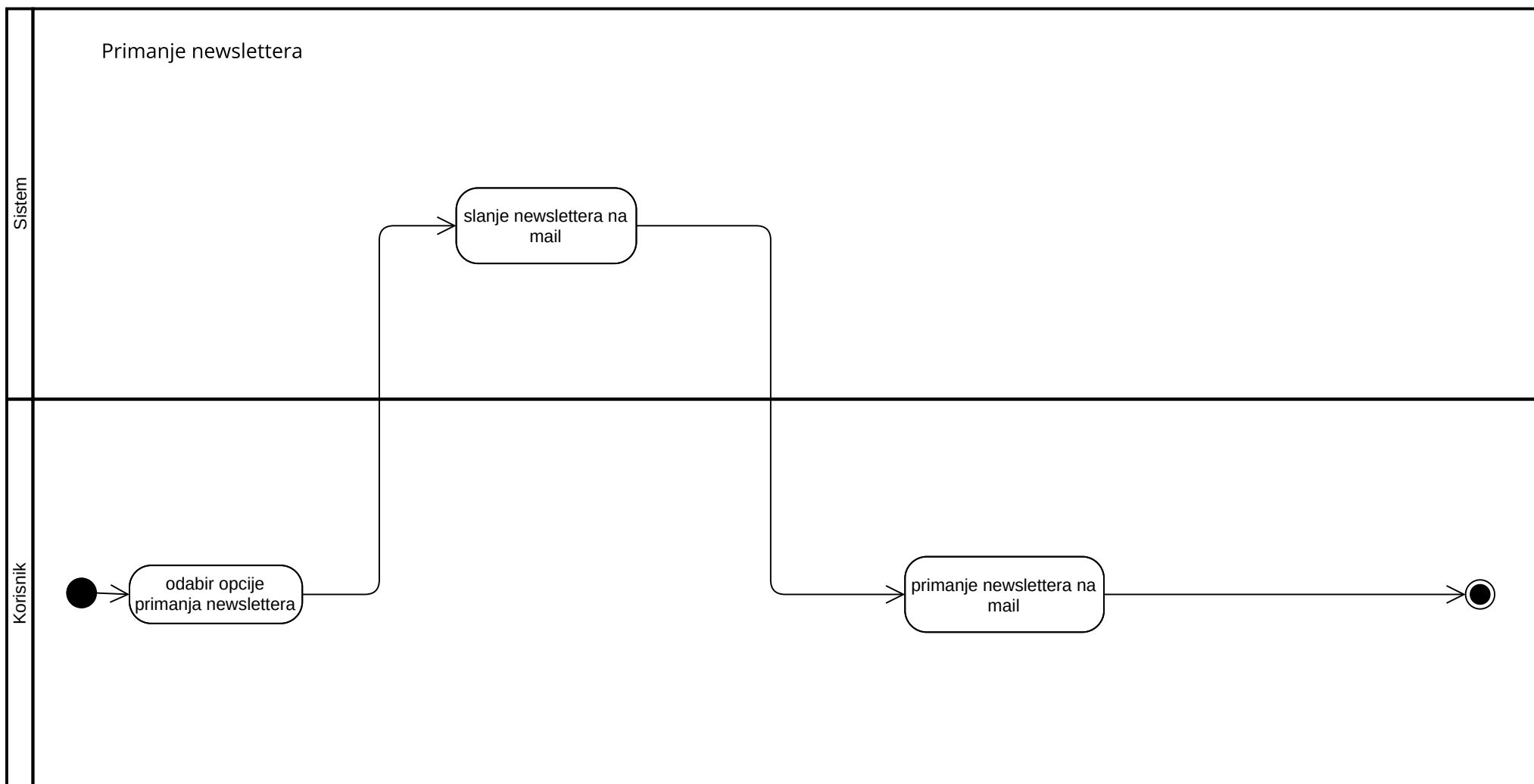




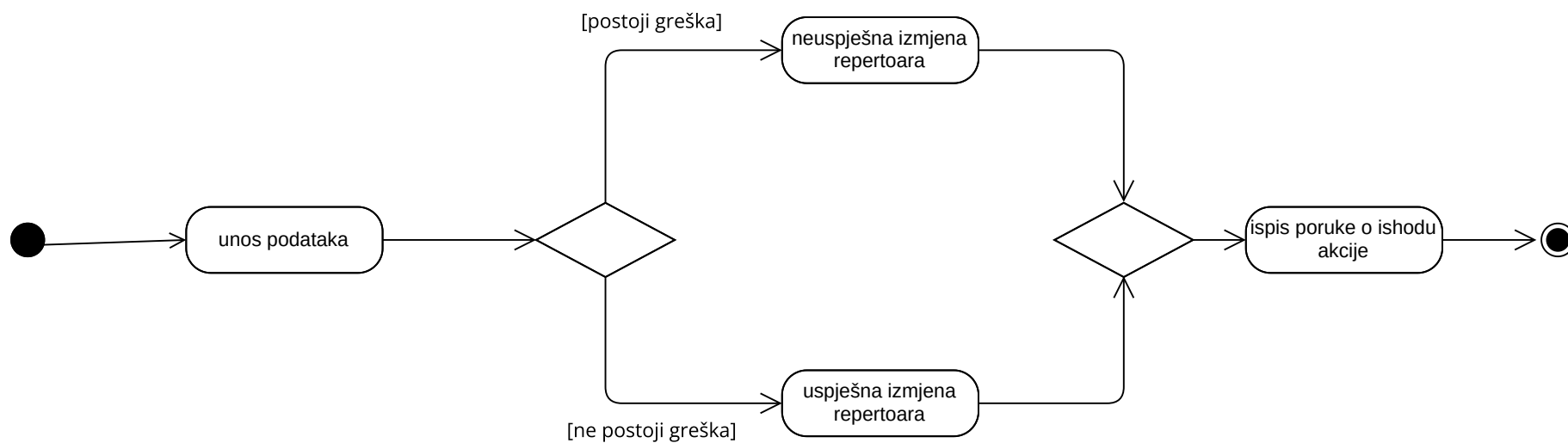








Izmjena repertoara



Prototipi korisnickih interfejsa



Teatar18

Početna

O nama

Pitanja

Prijava

Registracija



Ježeva kuća



Seljačka opera




Mišolovka



Marlena

Repertoar



Teatar18

Početna


O nama

Pitanja


Prijava

Registracija

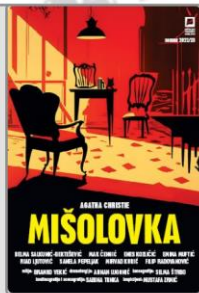
Repertoar




Ježeva kuća
Opera
7. 5. 2024. u 12h



Seljačka opera
Drama
11. 6. 2024. u 19:30h



Mišolovka
Drama
10. 6. 2024. u 19:30h



Marlena
Drama
4. 5. 2024. u 19:30h

Mišolovka

Drama

10.6.2024. u 17:30

130 min

Ocjena: 4.7

Ocijeni

Režiser: Branko Vekić

Scenarist: Adnan Lugonić

Scenograf: Sabina Trnka

Glumci: Belma Salkunić-Bektašević,
Mak Čengić, Enes Kozličić, Emina
Minka Muftić, Riad Ljutović, Sanela
pepeljak, Mirvad Kurić, Filip
Radanović

Opis

Mladi bračni par zablokiran je mećavom na svom imanju s još četiri gosta, i slučajnim prolaznikom koji se dovezao automobilom. Narednik Trotter doskija do njih i govori im da je ubojica na slobodi i da dolazi. Kada se ubojstvo zbilja dogodi svi posumnjaju na Cristophera Wrena, no na kraju zaključuje da ubojica može biti baš bilo tko. Na kraju Trotter namješta zamku za jednog od ukućana.



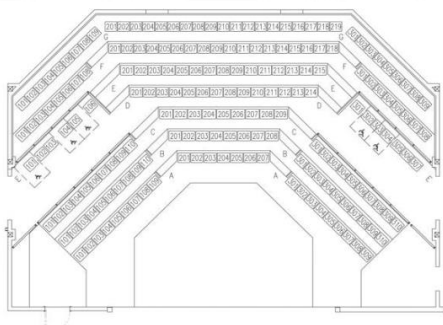
Rezerviži

Rezervacija karata

Datum izvedbe

Broj sjedišta

Odabir sjedišta

Plaćanje 

Pošalji rezervaciju

ORIGINAL

EST.

T · I · C ·

The

2 1 1 8 1 6 0 5 2 3

ACTOR PLAY

ORIGINAL

2009

K · E · T

atre

2 1 1 8 1 6 0 5 2 3

ACTOR PLAY

Username:

username

Password:

Prijava

ORIGINAL

EST.

T · I · C ·

The

2 1 1 8 1

ACTOR PLAY

ORIGINAL

2009

K · E · T

atre

6 0 5 2 3

ACTOR PLAY

Ime i prezime:

ime prezime

Email adresa:

email@example.com

Password:

Registracija



Teatar18

[Početna](#)[O nama](#)[Repertoar](#)[Prijava](#)[Registracija](#)**Username:****eMail adresa:****Aktivne rezervacije:**

ACTOR PLAY

ACTOR PLAY



Teatar18

[Početna](#)[O nama](#)[Pitanja](#)[Prijava](#)[Registracija](#)**eMail adresa za odgovor:****Unesite vaše pitanje:**

ACTOR PLAY

ACTOR PLAY



Početna

O nama

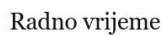
Pitanja

Prijava

Registracija

O nama

Historija

[illegible]

Ponedeljak – Petak : 15:00 - 22:00

Subota: 15:00 – 23:00

Nedjelja i praznici - neradni

Kontakt

E: info@example.ba

T: +387 33 530 174

F: +387 33 226 803

BLAGAJNA: +387 33 530 776

Tehničke informacije

[illegible]

Lokacija

Zmaja od Bosne 8B

71000 Sarajevo, BiH

Analiza i dizajn sistema

Analiza i dizajn sistema

U nastavku je potrebno definisati sve potencijalne klase koje će se koristiti u sistemu. Za određivanje klasa koje će biti neophodne za rad sistema potrebno je koristiti specifikaciju sistema i prethodno kreirane dijagrame.

Template za jednu klasu potrebno je iskopirati onoliko puta koliko je neophodno da bi se definisale sve klase u sistemu.

Definicija klasa u sistemu

Naziv klase: Uloga (Enum)

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Kreiranje korisničkog računa

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
----------------	---------------	------------------

Naziv klase: Korisnik

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Kreiranje korisničkog računa

FZ br. 2: Pregled repertoara

FZ br. 3: Kreiranje rezervacije

FZ br. 4: Kupovina i plaćanje karata

FZ br. 5: Otkazivanje rezervacije

FZ br. 6: Ocjenjivanje predstave

FZ br. 7: Postavljanje i odgovaranje na pitanja

FZ br. 8: Slanje i primanje newslettera

FZ br. 9: Izmjena repertoara

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Ime i prezime	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Username	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Password	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Uloga	Enum	<input type="checkbox"/> Atribut je statički <input checked="" type="checkbox"/> Atribut je <i>enumeration</i>

eMail adresa	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Broj kupljenih karata	int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Žanr (Enum)

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Repertoar

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
----------------	---------------	------------------

Naziv klase: Predstava

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Rezervacija i kupovina karata

FZ br. 2: Ocjena predstave

FZ br. 3: Preporuka predstava

FZ br. 4: Repertoar

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Naziv	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Glumci	List<string>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Scenarist	List<string>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Režiser	List<string>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Scenografi	List<string>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Žanr	Enum	<input type="checkbox"/> Atribut je statički <input checked="" type="checkbox"/> Atribut je <i>enumeration</i>
Opis	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Poster	Uri	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Trajanje	int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Ocjene	List<double>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Slobodne karte	List<int>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Karta

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Rezervacija karata
FZ br. 2: Otkazivanje rezervacije
FZ br. 3: Potvrda otkazivanja rezervacije
FZ br. 4: Validacija rezervacije
FZ br. 5: Kupovina karata
FZ br. 6: Plaćanje
FZ br. 7: Plaćanje uživo
FZ br. 8: Plaćanje online
FZ br. 9: Ocjena predstave

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Redni broj karte	int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Predstava	Predstava	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Datum i vrijeme održavanja	DateTime	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Sjedište	int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Cijena	double	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Plaćena	bool	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Rezervacija

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Rezervacija karata
FZ br. 2: Otkazivanje rezervacije
FZ br. 3: Potvrda otkazivanja rezervacije
FZ br. 4: Validacija rezervacije

FZ br. 5: Kupovina karata
FZ br. 6: Plaćanje
FZ br. 7: Plaćanje uživo
FZ br. 8: Plaćanje online
FZ br. 9: Ocjena predstave

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
ID rezervacije	int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Korisnik	Korisnik	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Predstava	Predstava	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Redni broj karata	List <int>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Kupovina	bool	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Broj rezervisanih karata	int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Obračunavanje popusta	bool	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Repertoar (singleton)

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Prikaz repertoara
FZ br. 2: Mijenjanje repertoara

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Predstave i izvedbe	List <Pair <Predstava, List <DateTime>>>>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Preporuke	List <Predstava>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Instanca repertoara	Repertoar	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Newsletter

Funkcionalni zahtjevi u kojima klasa učestvuje:

FZ br. 1: Slanje newslettera

FZ br. 2: Primanje newslettera

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Informacije	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Redni broj	int	<input checked="" type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Korisnici	List <Korisnik>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Datum slanja	DateTime	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Pitanje

Funkcionalni zahtjevi u kojima klasa učestvuje:

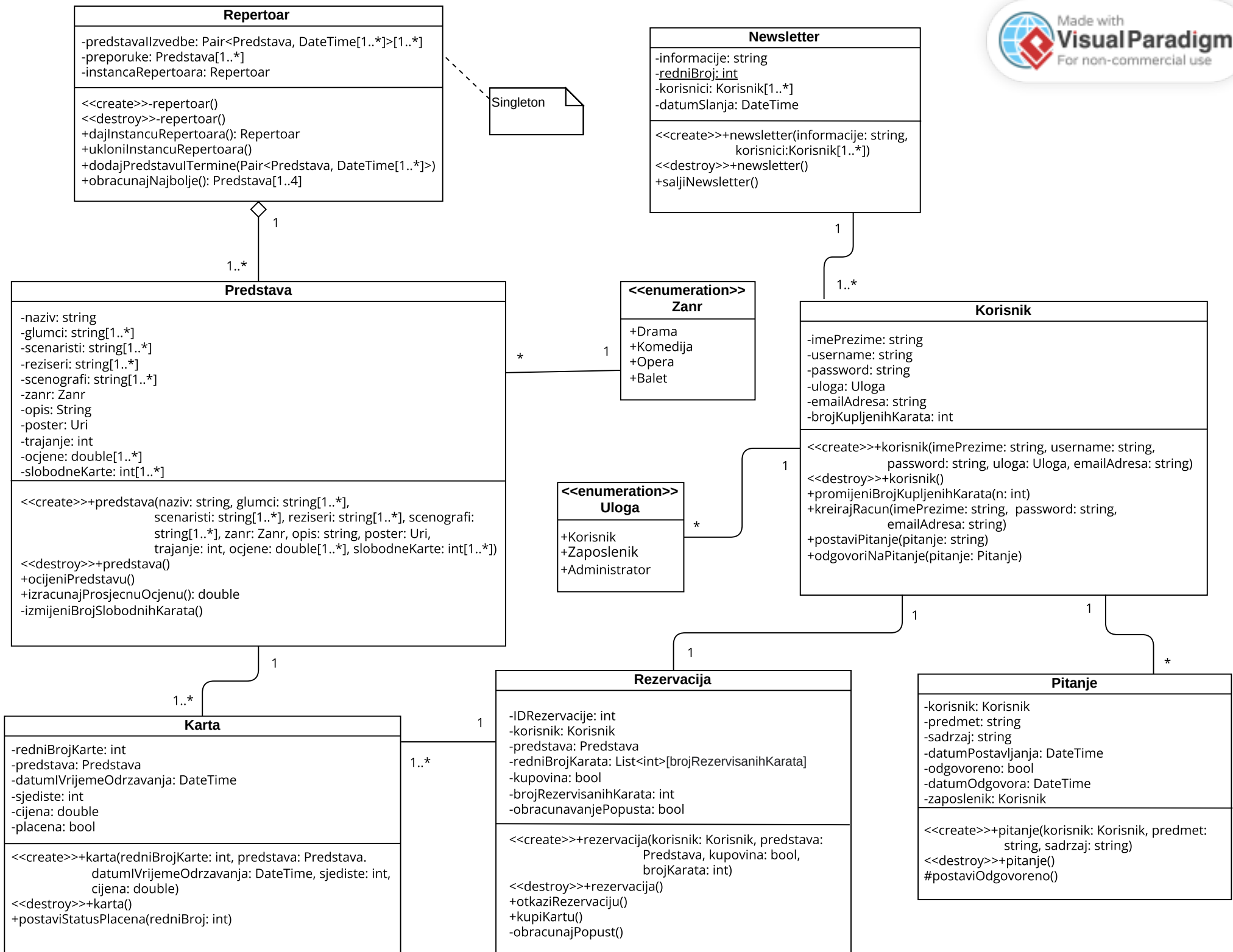
FZ br. 1: Postavljanje pitanja

FZ br. 2: Odgovaranje na pitanja

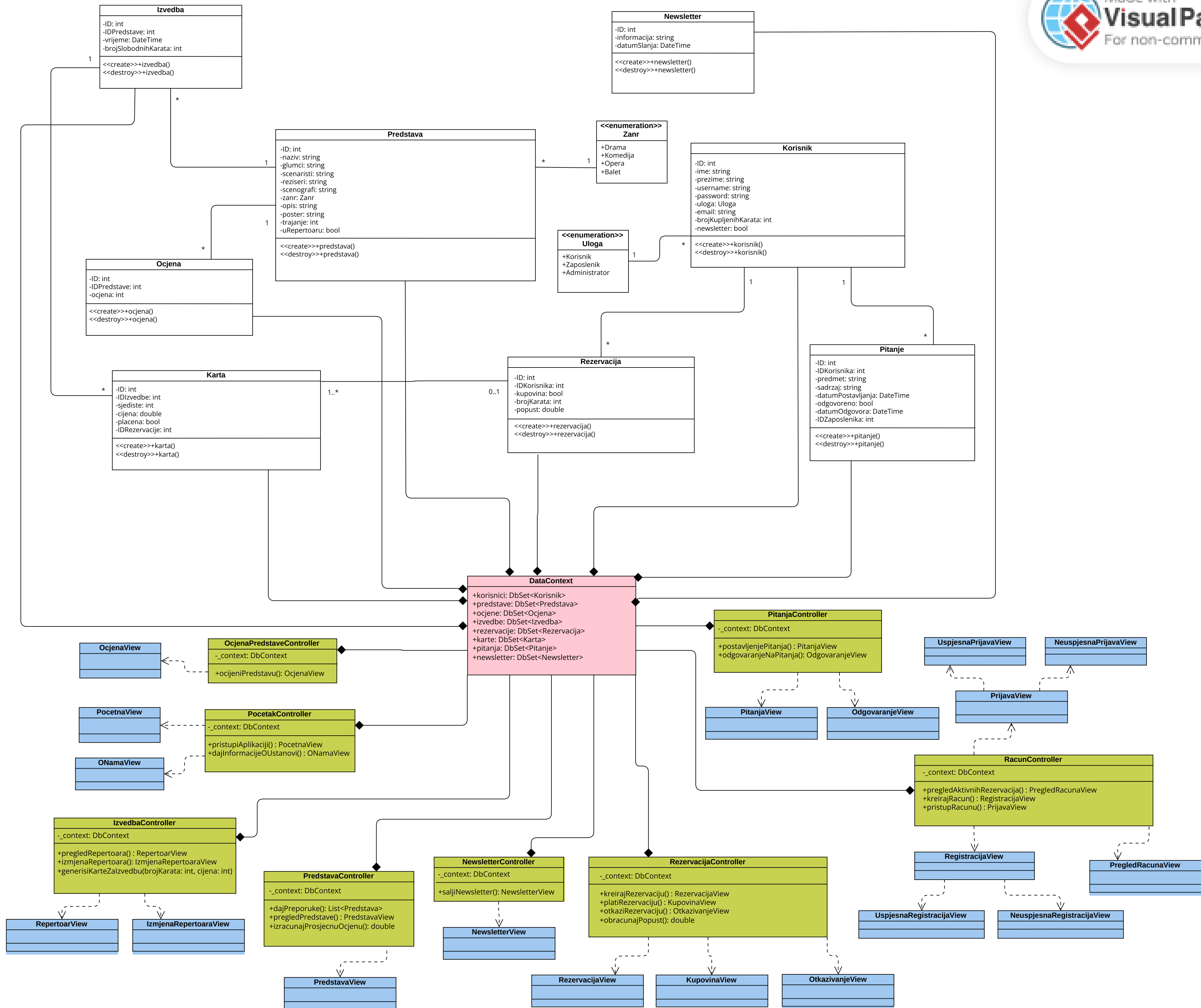
Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Korisnik	Korisnik	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Predmet	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Sadržaj	string	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Datum postavljanja	DateTime	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Odgovoreno	bool	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Datum odgovora	DateTime	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Zaposlenik	Korisnik	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Dijagram klasa

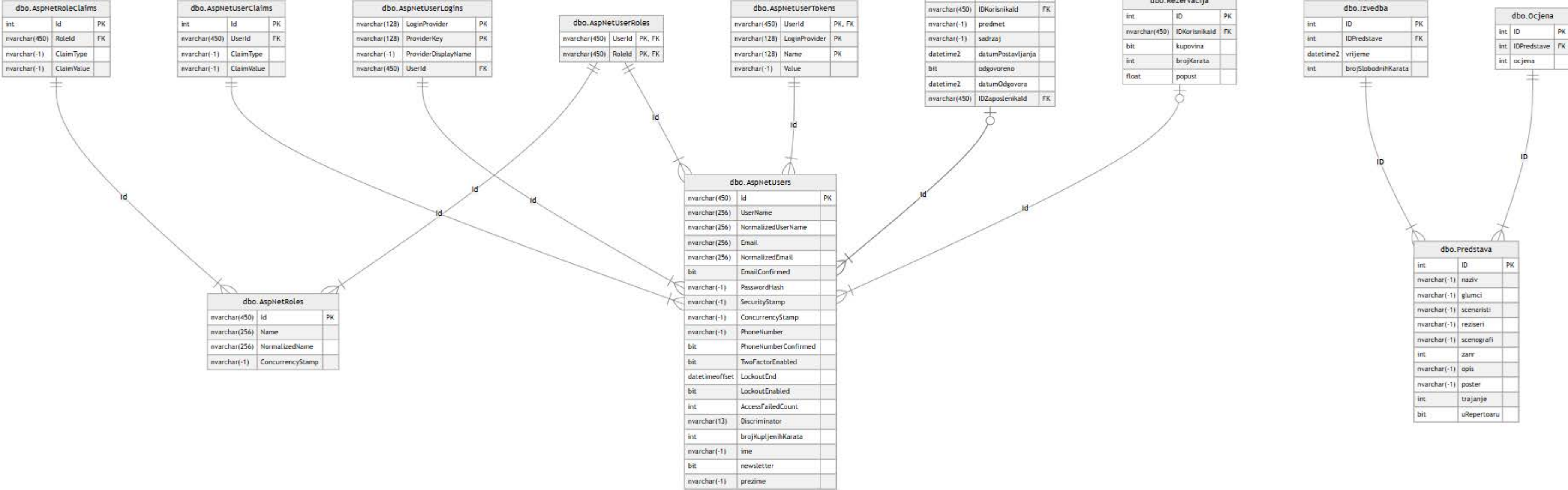


MVC diagram
klasa

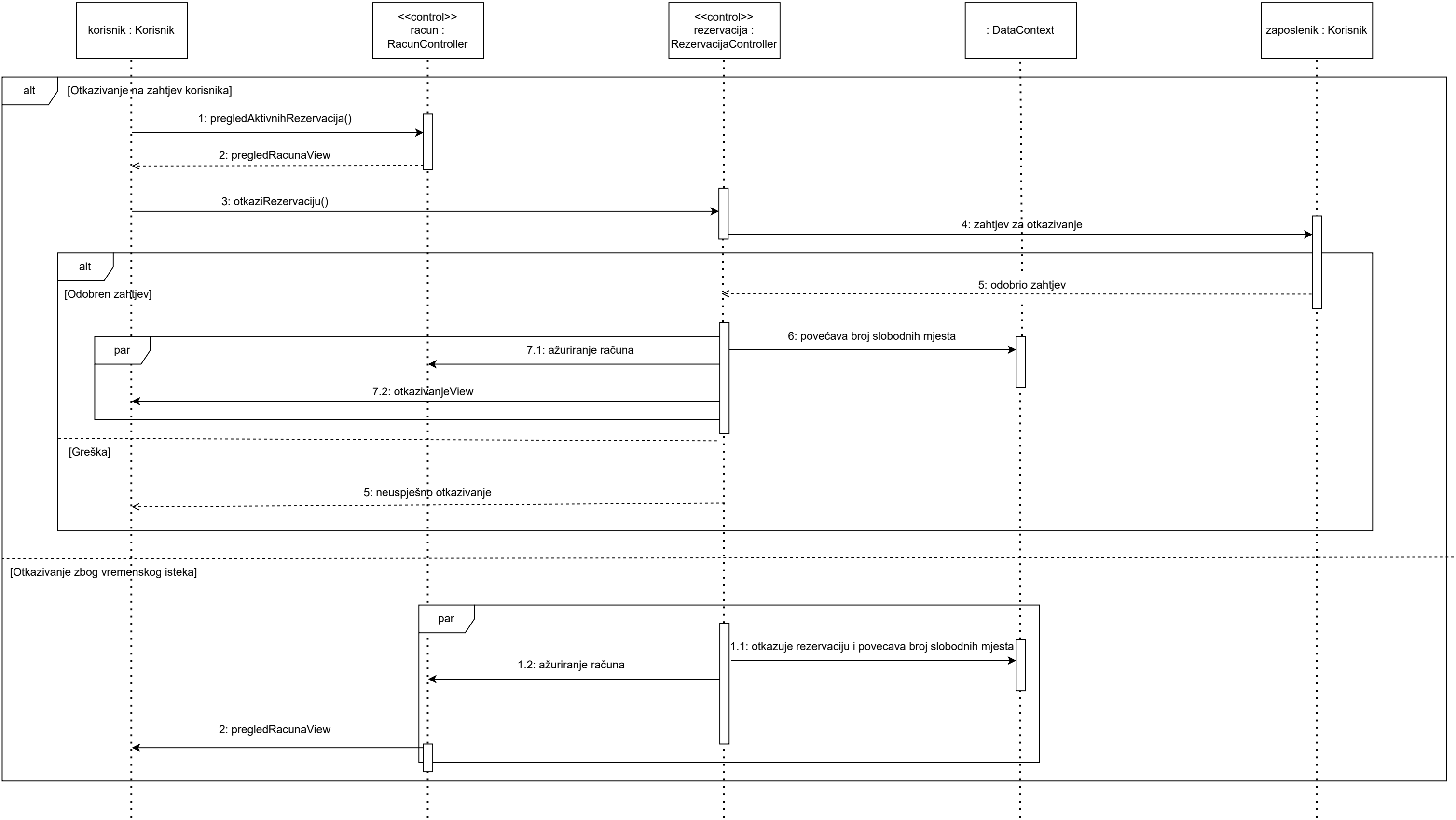


ERD

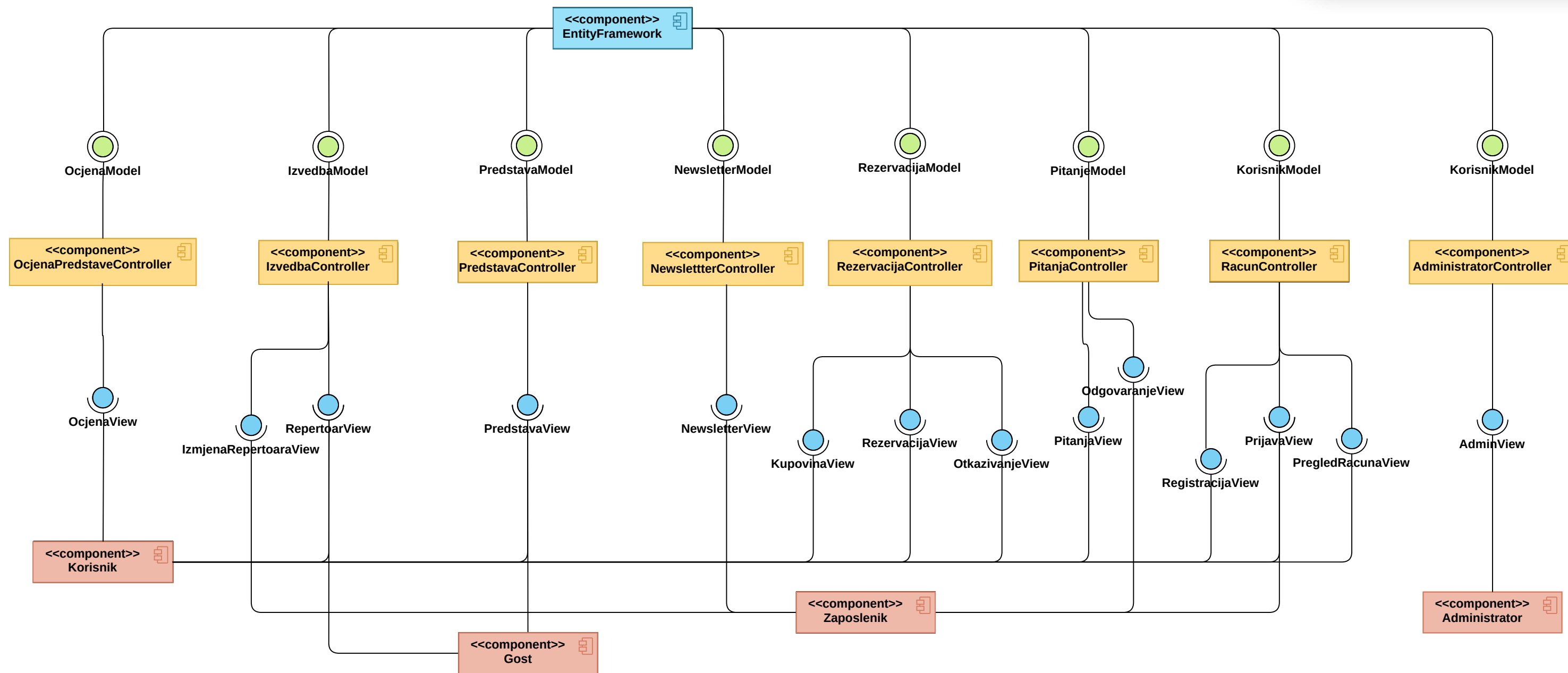
*(entity relationship
diagram)*



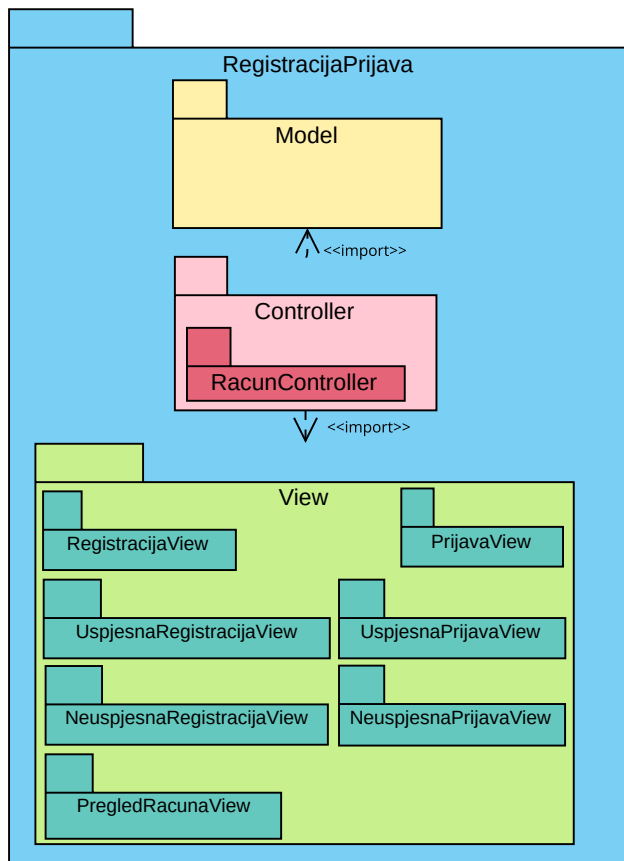
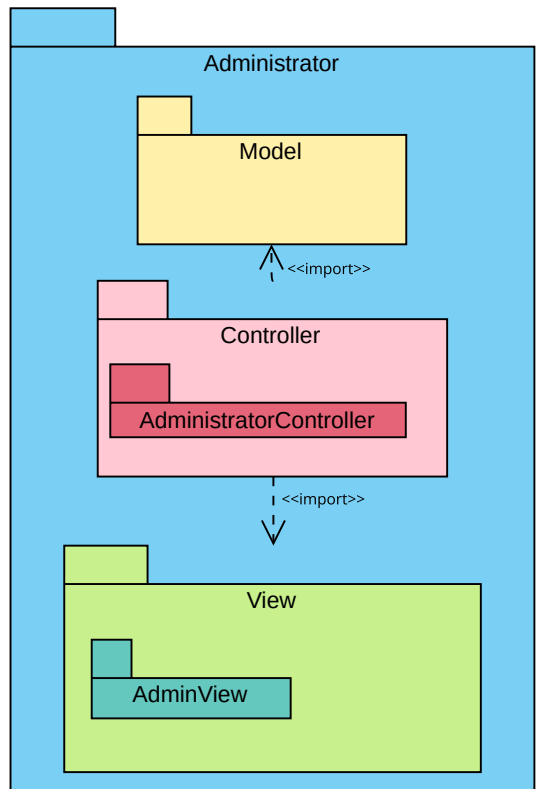
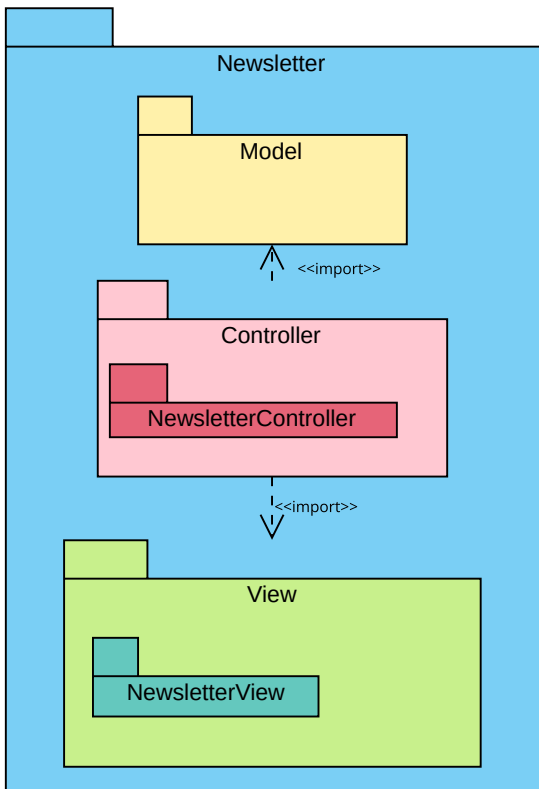
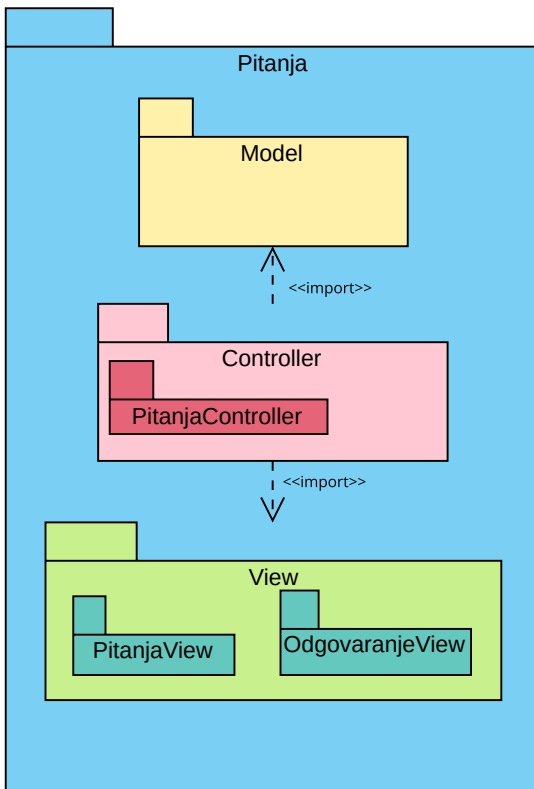
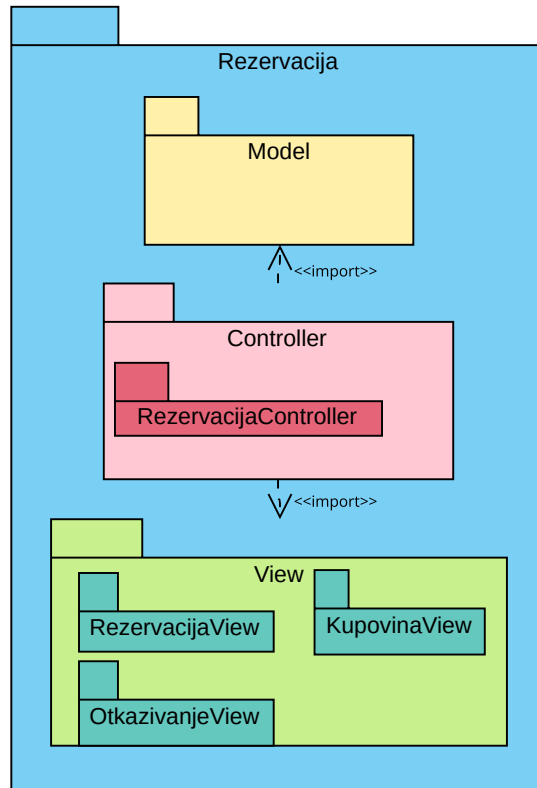
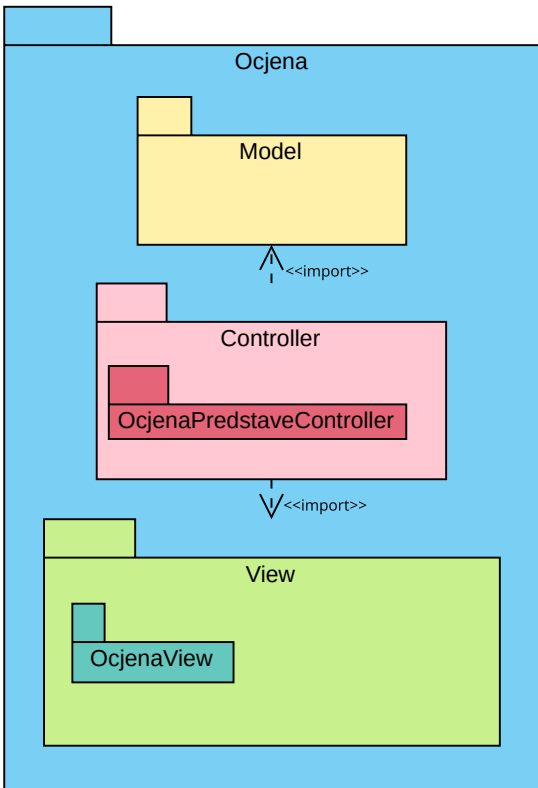
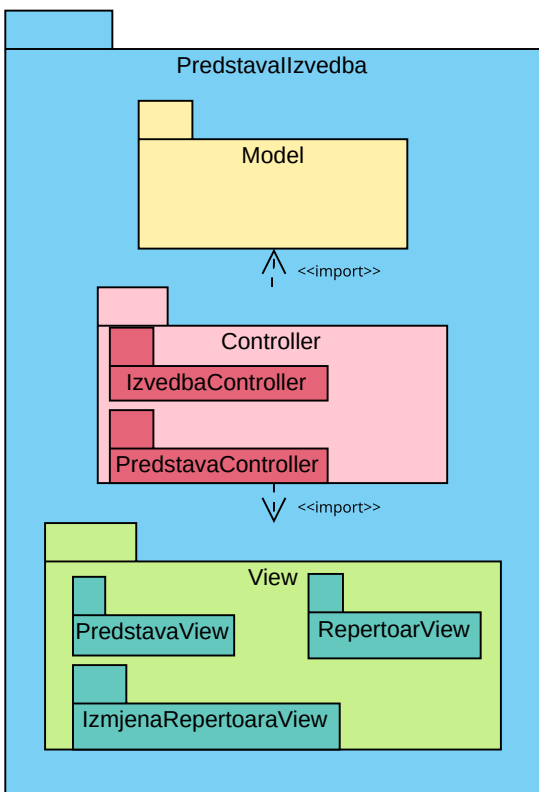
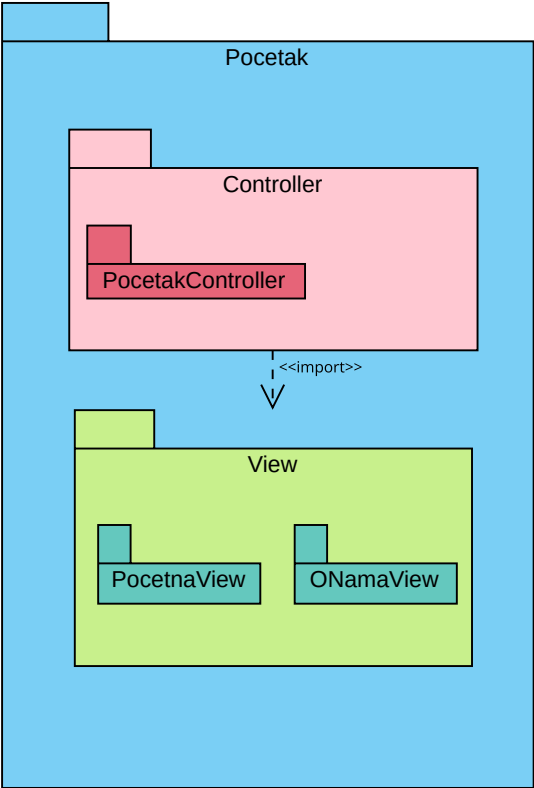
Dijagrami interakcije



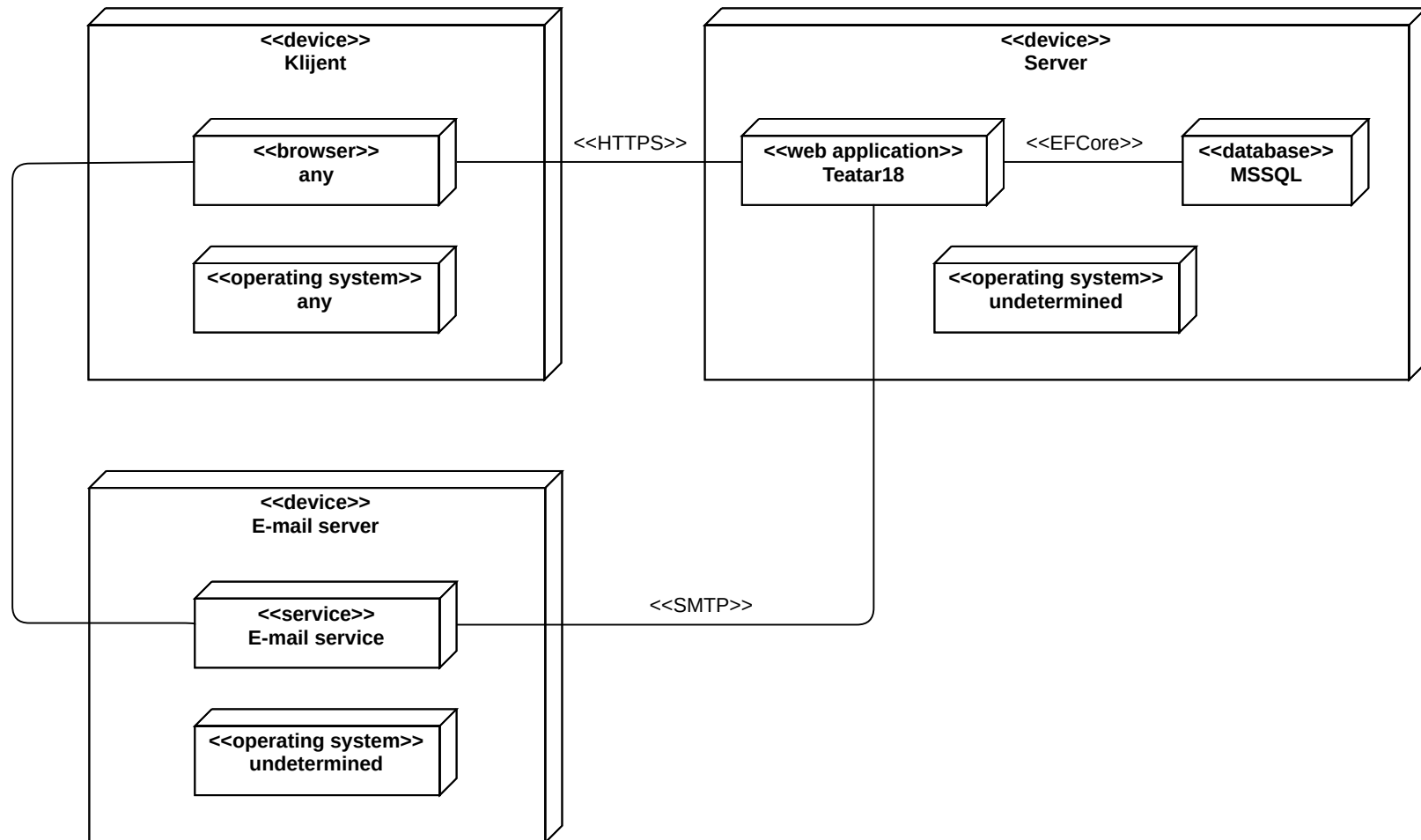
Dijagram komponenti



Dijagram paketa



*Dijagram
rasporedjivanja*



SOLLID

principi

SOLID principi

1) Single Responsibility Principle

Princip glasi: “Klasa bi trebala imati samo jedan razlog za promjenu”. U našem dijagramu klasa, ovaj princip je zadovoljen jer svaka klasa upravlja svojim atributima, koji su privatni. Funkcionalnosti sistema su razdvojene tako da svaka klasa ima pojedinačnu odgovornost. Npr. klasa Rezervacija služi za upravljanje rezervacijom, te je razdvojena od Korisnika i Karte. Također, iako su Korisnici ti koji postavljaju i odgovaraju na pitanja, ove funkcionalnosti smo odvojili u zasebnu klasu Pitanje, budući da to nije primarna uloga Korisnika.

2) Open Closed Principle

Princip glasi: “Entiteti softvera (klase, moduli, metode) trebali bi biti otvoreni za nadogradnju, ali zatvoreni za modifikacije”. Obzirom da su klase povezane vezama asocijacije i agregacije, te da kao attribute često imamo objekte drugih klasa, promjena u jednoj klasi neće značiti promjenu u ostalim klasama. Npr., promjenom objekata klasa Korisnik i Karta, korištenih unutar klase Rezervacija, ne dolazi do stvarne promjene nad klasom Rezervacija, koja nastavlja vršiti svoje funkcionalnosti.

3) Liskov Substitution Principle

Princip glasi: “Podtipovi moraju zamjenjivi njihovim osnovnim tipovima”. Unutar našeg dijagrama klasa nemamo apstraktnih klasa, te samim tim je princip zadovoljen. Razlikovanje vrsta korisnika sistema je implementirano preko enuma Uloga (Korisnik, Zaposlenik i Administrator).

4) Interface Segregation Principle

Princip glasi: “Klijenti ne treba da ovise o metodama koje neće upotrebljavati”. U našem sistemu je ovaj princip zadovoljen, budući da se funkcionalnosti razdvajaju u zavisnosti od uloge korisnika, definisane kao atribut klase Korisnik.

5) Dependency Inversion Principle

Princip glasi: “Moduli visokog nivoa ne bi trebali ovisiti od modula niskog nivoa” / “Moduli ne bi trebali ovisiti od detalja”. Kao i na primjeru principa 3, kako nemamo apstraktnih klasa, ovaj princip je time zadovoljen.

Strukturalni paterni

STRUKTURALNI PATERNI

Adapter patern [implementacija]

U slučaju da nam je potreban drugačiji interface već postojeće klase možemo iskoristiti adapter patern za kreiranje adapter klase kao posrednika između originalne klase i željenog interface-a. Ovim se sprječava izmjena postojeće klase.

U našoj aplikaciji adapter patern se može implementirati za prilagođavanje podataka o zauzetim sjedištima iz baze s ciljem tabelarnog prikaza na korisničkom interface-u. Ovaj adapter bi formatirao podatke na način da budu prilagođeni za tabelarni prikaz kroz html.

Facade patern [implementacija]

Ovaj patern omogućava stapanje više složenijih podsistema u jedan kako bi interakcija korisnika sa sistemom bila jednostavnija i intuitivnija. Istovremeno patern osigurava da korisniku sistema nisu dostupne informacije o samoj implementaciji sistema čime se ispunjava jedan od osnovnih principa objektno orijentisanog programiranja.

Kroz našu aplikaciju ovo smo implementirali na način da su funkcionalnosti rezervacije i kupovine karata kao i odabira željenih sjedišta grupisane u jedinstven korisnički interface, iako su interno implementirani kao odvojeni sistemi.

Decorater patern

Namjena ovog paterna je da omogući dinamičko dodavanje novih elemenata i funkcionalnosti u postojećim objektima pri čemu objekat ne zna da je urađena dekoracija.

Ovaj patern smo iskoristili tako što korisnik ima mogućnost korištenja popusta prilikom plaćanja karte nakon ostvarivanja određenog broja odgledanih predstava. Ovim se proširuje funkcionalnost i usluga sistema za rezervaciju i kupovinu karata.

U slučaju dodavanja posebne vrste karata (npr. VIP) moguće je iskoristiti ovaj patern na sličan način kao i u slučaju popusta.

Bridge patern

Ovaj patern omogućava klasi posjedovanje više različitih apstrakcija i implementaciju za svaku apstrakciju kombinujući dvije ili više ortogonalnih hijerarhija klasa. Koristeći ovaj patern zadovoljava se i open-closed princip.

Ukoliko bismo se odlučili na proširenje načina online plaćanja (npr. dodavanja opcije paypal pored opcije kartičnog plaćanja) ove dvije opcije bismo mogli objediniti u okviru jedne klase

online plaćanja (apstrakcija). Zatim bi bilo potrebno kreirati interface koji bi implementirao metode za plaćanje.

Proxy patern

Proxy patern omogućava klijentu da pristupa objektu putem posrednika a ne direktno. Ovaj posrednik može obavljati dodatne zadatke i vršiti kontrolu pristupa stvarnim objektima. Ovim se osigurava open-closed princip.

Budući da se u našoj bazi podataka čuvaju informacije o korisnicima sistema moguće je iskoristiti proxy patern za autentifikaciju pri pokušaju pristupa stvarnom objektu korisničkog računa. Ovim se osigurava da samo zaposlenici imaju pristup svim korisničkim podacima.

Composite patern

Ovaj patern omogućava formiranje strukture drveta pomoću klasa tako da se jednako tretiraju individualni objekti kao i njihova kompozicija.

U našoj aplikaciji ovaj patern bi se implementirao uvođenjem interface-a kojeg implementiraju predstava i repertoar koji sadrži kolekciju predstava. Ovaj interface bi implementirao ponašanje zajedničko za pojedinačnu komponentu i njihovu kompoziciju kao što su prikaz, izmjena predstave i sl.

Flyweight patern

Ovaj patern omogućava brži pristup objektima na zahtjev. Kako objekti jedne klase mogu vrlo često posjedovati iste vrijednosti atributa ovo svojstvo možemo iskoristiti pri bržem kreiranju novih objekata, pristupu postojećim a posljedično i smanjenom memorijskom zauzeću.

Pri implementaciji naše aplikacije patern se može zadovoljiti prilikom prikazivanja pojedinačnih predstava korisniku na način da bi se uvijek koristito isti layout u koji se ubacuju konkretni podaci za predstavu. Također, ovaj patern bi se mogao primijeniti i prilikom kreiranja novog korisnika na način da bi defaultna uloga korisnika bila korisnik sistema i inicijalni broj kupljenih karata bi bio 0. Potom bi se postavljali podaci za individualnog korisnika.

Kreacijski paterni

KREACIJSKI PATERNI

Singleton [implementacija]

Singleton patern podrazumijeva instanciranje objekta klase samo jednom, te je nakon toga objektu omogućen globalni pristup. Koristi se kada instanciranje objekata klase više od jednom može prouzrokovati probleme kao što su nekorektno ponašanje programa, neadekvatno korištenje resursa ili nekonzistentni rezultati.

U našem sistemu Singleton patern se odnosi na instanciranje klase Repertoar, budući da će u sistemu postojati samo jedan repertoar čiji će se atributi mijenjati jednom mjesečno.

Prototype [implementacija]

Prototype patern omogućava kreiranje objekata bez poznavanja detalja o njegovoj klasi. Omogućava konstrukciju objekta kloniranjem i modifikacijom ranije kreiranog objekta. Koristi se kada je kreiranje novog objekta resursno zahtjevno.

U našem sistemu patern je moguće implementirati na klasi Karta, s obzirom da se u instancama klase Karta za jednu predstavu razlikuju samo njihovi ID-evi. Na ovaj način ćemo izbjeći kreiranje više sličnih objekata sa istim atributima jer će svi atributi biti kopirani, a ID će biti modificiran.

Factory Method

Svrha ovog kreacijskog patern je da omogući kreiranje objekata tako da podklase odluče koja od njih će biti instancirana. Koristi se kada klijent ne mora ili ne treba znati koji tip objekta će biti instanciran, nego mu je samo bitno da može izvršiti određene funkcionalnosti. Factory podklasa ovim ima punu kontrolu nad kreacijskim procesom, a izbjegnuta je bilo kakva veza između klijenta i konkretnih produkata.

Kako ćemo u sklopu našeg sistema pratiti broj odgledanih predstava za korisnike, te omogućiti određeni popust nakon izvjesnog broja kupljenih karata, ovaj patern bi se mogao u budućoj verziji sistema iskoristiti na način da korisnik dobiva samo obavijest o rezervaciji bez da zna da li je rezervacija sa popustom ili nije.

Abstract Factory

Ovaj patern omogućava kreiranje porodica povezanih objekata bez specificiranja konkretnih klasa. Na osnovu apstraktne porodice produkata kreiraju se konkretne fabrike produkata različitih tipova i kombinacija. Klijent radi samo sa aspraktnim klasama i ne zna za stvarne objekte koji su kreirani fabrikama.

Kako Abstract Factory često predstavlja nadogradnju Factory Method paterna za kompleksnije sisteme, a naš sistem nije dovoljno kompleksan, nije moguća implementacija Abstract Factory-a.

Builder patern

Builder patern omogućava kreiranje različitih tipova objekata koristeći isti konstrukcijski kod. Koristi se kada se objekti mogu podijeliti u skupove jednostavnijih objekata koji se razlikuju samo po permutaciji njihovih dijelova. Njegovom upotrebom se zadovoljava Single Responsibility princip i izbjegava ponavljanje koda.

Kako se Builder patern često koristi u aplikacijama koje kreiraju kompleksne strukture koje nisu prisutne u našem sistemu, nije moguće pronaći njegovu primjenu.

Paterni ponašanja

PATERNI PONAŠANJA

Strategy patern

U slučaju kada želimo izdvojiti algoritme u posebne klase, koristimo Strategy patern. Pogodno ga je koristiti ukoliko za isti problem postoje različiti algoritmi koji vode ka rješenju, a klijent je onaj koji bira koji od algoritama želi koristiti. U ovom paternu je, kao i u ostalim paternima, implementacija algoritma u potpunosti neovisna od klijenta.

U nekim budućim nadogradnjama sistema, ovaj patern bi mogao biti primijenjen prilikom rezervacije i kupovine karata. Konkretna implementacija bi se sastojala u tome da se zaposleniku sistema, ukoliko je na sistem prijavljen sa tom ulogom, omogući određen popust prilikom rezervacije i kupovine karata.

U trenutnoj postavci sistema je ovaj patern moguće primijeniti za odabir načina plaćanja, tj. da će plaćanje biti različito obrađeno u zavisnosti od toga da li korisnik nakon izvršene rezervacije odluči da li će kartu platiti online platnom karticom ili pak na licu mjesta pouzećem.

State patern [implementacija]

State patern je karakterističan po tome što omogućava promjenu ponašanja objekta, u zavisnosti od njegovog trenutnog stanja. Pogodan je ukoliko objekat može imati više stanja i ukoliko često dolazi do promjene trenutnog stanja. Patern zadovoljava Single Responsibility i Open / Closed princip.

U našem sistemu ovaj patern možemo iskoristiti za obrađivanje karata i sjedišta, te im postavljati određeni status. Na primjer, status karte može biti rezervirana, kupljena ili slobodna, dok status sjedišta može biti slobodno, zauzeto ili blokirano (ukoliko se koristi kao dio scenografije ili je u nekom slučaju pokvareno).

Template Method patern

Patern je pogodan za upotrebu kada postoji potreba za izdvajanjem određenih koraka algoritma u odvojene podklase, što ne utječe na strukturu samog algoritma, ali omogućava implementaciju pojedinačnih dijelova na različite načine.

Ovaj patern se može primijeniti prilikom prijave na sistem za različite korisnike. Ukoliko se na sistem prijavljuje zaposlenik, bit će zatraženi različiti podaci u odnosu na one kada se na sistem prijavljuje obični korisnik.

Observer patern

Uloga ovog patern je da uspostavi relaciju između objekata tako da prilikom promjene jednog objekta, drugi povezani objekti o tome budu obaviješteni.

U našem sistemu bi ovaj patern mogao biti implementiran sa Newsletter-om. Kako se korisnici prijavljuju za primanje Newslettera, ovaj patern bi mogao biti iskorišten za slanje podataka o ažuriranju repertoara svim zainteresiranim korisnicima. Također, mogao bi se implementirati i za obrade situacija kada za neku predstavu više nema slobodnih karata koje se mogu rezervirati, pa se korisnik može prijaviti da bude obaviješten ukoliko se neka od rezervacija otkaže.

Iterator patern [implementacija]

Iterator patern omogućava kretanje kroz kolekciju podataka bez da nužno poznaje detalje o strukturi kolekcije. Iterator može implementirati različite algoritme koji mu omogućavaju kretanje kroz kolekcije. Kretanje je omogućeno korištenjem zajedničkog interfejsa, a istovremeno se kroz kolekciju može kretati više objekata tipa Iterator.

U našem sistemu Iterator patern bi mogao biti implementiran za potrebe efikasnog prolaženja kroz postojeće predstave. Kako je pregled najbolje ocijenjenih predstava jedna od funkcionalnosti koja je omogućena na našoj stranici, potrebno je ažurirati ovu listu svaki put kada se dodaju nove ocjene u bazu. U tom slučaju, potrebno je iterirati kroz sve aktuelne predstave i generisati nove preporuke. Implementacijom Iterator patern, ovaj pregled bi se značajno unaprijedio i ubrzao.

Command patern

Command patern pretvara zahtjev u samostalni objekt koji se koristi za enkapsuliranje svih informacija potrebnih za izvršenje radnje ili pokretanje događaja u kasnijem trenutku. Ove informacije uključuju ime metode, objekt koji posjeduje metodu i vrijednosti za parametre metode. Ova transformacija omogućava prosljeđivanje zahtjeva kao argumenata metode, odlaganje ili stavljanje u red za izvršenje zahtjeva i podržavanje operacija koje se inače ne bi mogle izvršiti.

Kako u našem sistemu imamo dva načina plaćanja, online ili na licu mjesta, gdje opet može platiti karticom ili kešem, ovaj patern se može iskoristi za kreiranje jedinstvenog zahtjeva koji predstavlja plaćanje, a informacije o načinu plaćanja se popunjavaju u zavisnosti od odabranog načina.