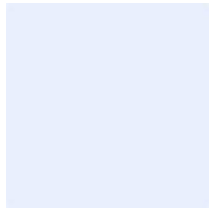


Specifikacija projekta

1. Osnovne informacije o sistemu

Naziv teme: Online rezervacija/prodaja karata

Logo:



Naziv tima: Tim 37

Nastavna grupa: Grupa 7

Link na repozitorij tima:

<https://github.com/OOAD-2023-2024/OnlineRezervacija-ProdajaKarata-Tim37>

Članovi tima:

1. (Denan Poturak, 19045)
2. (Hamza Hrnjic, 19085)
3. (Zejd Krek, 17530)

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).

Dati sistem u osnovi ima za cilj olaksanje kupovine i prodaje karata za manifestacije svih žanrova. Karte će biti u vidu QR koda, koji će korisniku biti prikazan prilikom završetka kupovine iste. Glavna funkcionalnost sistema je prikaz rasporeda slobodnih/zauzetih mjesta u sali izabrane manifestacije, te omogućavanje odabira željenog mjesta. Da bi korisnik mogao izvršiti kupovinu karte mora imati registrovan korisnički nalog, a isti može samostalno kreirati. Vodit će se briga o lojalnosti aktivnih kupaca na način da poslije kupovine 5 karata dobijaju popust od 30% na sljedeću kartu.

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 i pristupanje korisničkim računom

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

Svaki korisnik može napraviti korisnički račun i može pristupati sajtu te kupovati karte. Posjedovanje korisničkog naloga će biti uslov za kupovinu karte. U zavisnosti od računa imaće cemo administratorsku, korisnički i zaposleničku formu.

2) **Naziv funkcionalnosti:** Kupovina ulaznica

Vrsta funkcionalnosti: Korištenje vanjskog uređaja

Opis funkcionalnosti:

Pri završetku kupovine korisnik dobija jedinstveni QR kod koji će biti skeniran prilikom ulaska na manifestaciju. Jedinstvenost QR kodova garantuje da neće dolaziti do zabune.

3) **Naziv funkcionalnosti:** Prikaz slobodnih mjesta unutar dvorane

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

Poslije odabira manifestacije kojoj želi prisustvovati korisnik, bit će prosljeđen na odabir mjesta u dvorani/sali i omogućen mu je prikaz svih slobodnih i zauzetih mjesta, te pomoću drop menija može izabrati red i kolonu.

4) **Naziv funkcionalnosti:** Upravljanje manifestacijom i kartama

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Administrator može kreirati nove manifestacije te dodavati broj karata broj sjedista za određenu manifestaciju. Također administratoru će biti omogućeno brisanje te izmjena parametara datog događaja.

5) **Naziv funkcionalnosti:** Popust na broj kupljenih karata

Vrsta funkcionalnosti: Operacija sa specifičnim algoritmom obrade

Opis funkcionalnosti:

Ukoliko korisnik kupi 5 karata, sljedeća karta koju bude kupovao dobija 25% popusta na cijenu. Vjerujemo da bi ovom funkcionalnošću postigli veću kupovinu karata kao i korištenje sajta, te povećanje lojalnosti.

6) **Naziv funkcionalnosti:** Pregled kupaca karata

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

Za svaki događaj će uposlenik moći vidjeti sve posjetioce koji su kupili kartu, sa nekim osnovnim informacijama koje su bitne.

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:** Kupac karata

Vrsta aktera: Korisnik usluge

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
(1) Kreiranje i pristupanje korisničkom racunu)	Mogućnost uređivanja
(2) Kupovina ulaznice	Mogućnost pregleda
(3) Prikaz Slobodnih mjesta	Mogućnost pregleda
(5) Popust na broj kupljenih karata	Mogućnost pregleda

b) **Naziv aktera:** Administrator

Vrsta aktera: Administrator

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
(4) Upravljanje manifestacija i kartama	Mogućnost uređivanja
(6) Pregled kupaca karata	Mogućnost uređivanja
(1) Kreiranje i pristupanje korisničkim racunom	Mogućnost uređivanja
(2) Kupovina ulaznice	Mogućnost pregleda
(3) Prikaz slobodnih mjesta	Mogućnost uređivanja
(5) Popust na broj kupljenih karata	Mogućnost uređivanja

c) **Naziv aktera:** Zaposlenik

Vrsta aktera: Zaposlenik sistema

Funkcionalnosti u kojima akter učestvuje:

Način učešća:

- *Mogućnost pregleda*
- *Mogućnost uređivanja*

Funkcionalnost sistema	Način učešća
(6) Pregled kupaca karata	Mogućnost pregleda
(4) Upravljanje manifestacijama i kartama	Mogućnost uređivanja
(2) Kupovina ulaznice	Mogućnost pregleda
(3) Prikaz slobodnih mjesta	Mogućnost pregleda
(5) Popust na broj kupljenih karata	Mogućnost pregleda

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: Dostupnost sistema

Opis:

Sistem mora obezbijediti dostupnost u svakom trenutku, osim u slučaju problema koji se ne tiču istog.

2) Naziv nefunkcionalnog zahtjeva: Brzina aplikacije/Performanse

Opis:

Aplikacija mora u roku od 10 sekundi izvršiti izmjenu bazi prilikom kupovine karte.

3) Naziv nefunkcionalnog zahtjeva: Sigurnost sistema

Opis:

Sistem podatke mora razmjenivati sa određenim stepenom sigurnosti, tj vrsiti enkripciju isti. Na taj nacin ćemo svakom korisniku garantirati zastitu podataka.

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: User

Funkcionalni zahtjevi u kojima klasa učestvuje:

(FZ br. 1: Kreiranje i pristupanje korisničkim racunom)

(FZ br. 2: Kupovina ulaznica)

(FZ br. 00: ime zahtjeva)

(FZ br. 00: ime zahtjeva)

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Ime	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Prezime	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
JMBG	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>
Datum rođenja	Date	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Email	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Manifestacija

Funkcionalni zahtjevi u kojima klasa učestvuje:

(FZ br. 1: Pregled svih manifestacija)

(FZ br. 00: ime zahtjeva)

(FZ br. 00: ime zahtjeva)

(FZ br. 00: ime zahtjeva)

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Naziv manifestacije	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Datum i vrijeme manifestacije	Date	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Kategorija manifestacije	String	<input type="checkbox"/> Atribut je statički <input checked="" type="checkbox"/> Atribut je <i>enumeration</i>
Mjesto održavanja	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Trajanje manifestacije	Int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Opis manifestacije	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Cijena karte	Double	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Slika	Image	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
ID manifestacije	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: Pregled kupljenih karata)

(FZ br. 2: Otkazivanje karte)

(FZ br. 3: Pregled kupaca)

(FZ br. 00: ime zahtjeva)

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Datum kupovine	Date	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Kod karte	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Kolicina	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
JMB kupca	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
ID manifestacije	Int	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Mjesta	Array<Mjesto>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Ukupna cijena	Double	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Mjesto

Funkcionalni zahtjevi u kojima klasa učestvuje:

(FZ br. 1: Pregled slobodnih mjesta)

(FZ br. 2: Kupovina karte)

(FZ br. 00: ime zahtjeva)

(FZ br. 00: ime zahtjeva)

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Broj reda	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Broj kolone	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Status	Boolean	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Slobodna mjesta

Funkcionalni zahtjevi u kojima klasa učestvuje:

(FZ br. 00: Pregled slobodnih mjesta)

(FZ br. 00: ime zahtjeva)

(FZ br. 00: ime zahtjeva)

(FZ br. 00: ime zahtjeva)

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Datum i vrijeme događaja	Date	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
Slobodna mjesta	Array<Mjesto>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Historija karata

Funkcionalni zahtjevi u kojima klasa učestvuje:

(FZ br. 1: Pregled slobodnih mjesta)

(FZ br. 2: Kupovina karte)

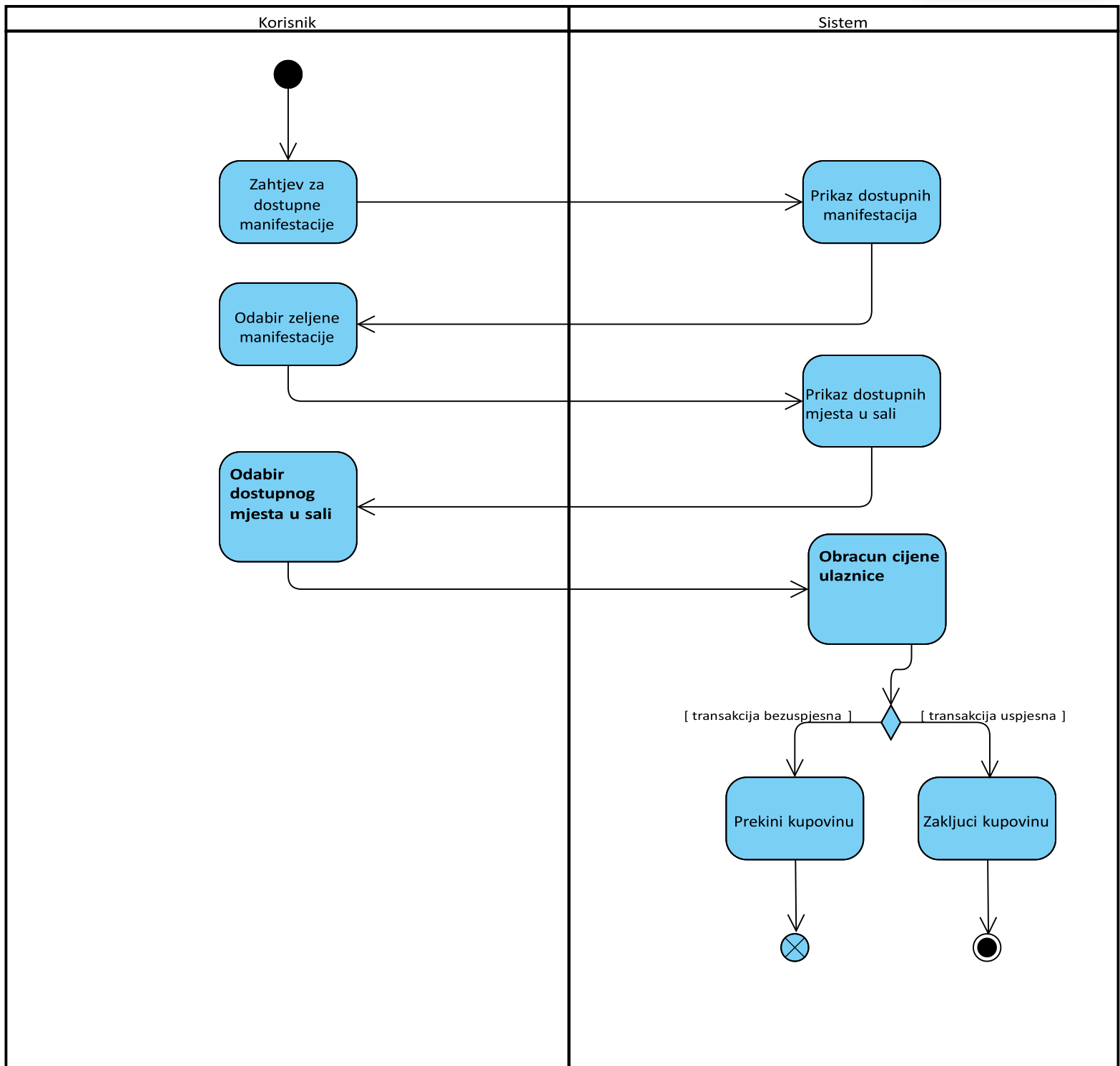
(FZ br. 00: ime zahtjeva)

(FZ br. 00: ime zahtjeva)

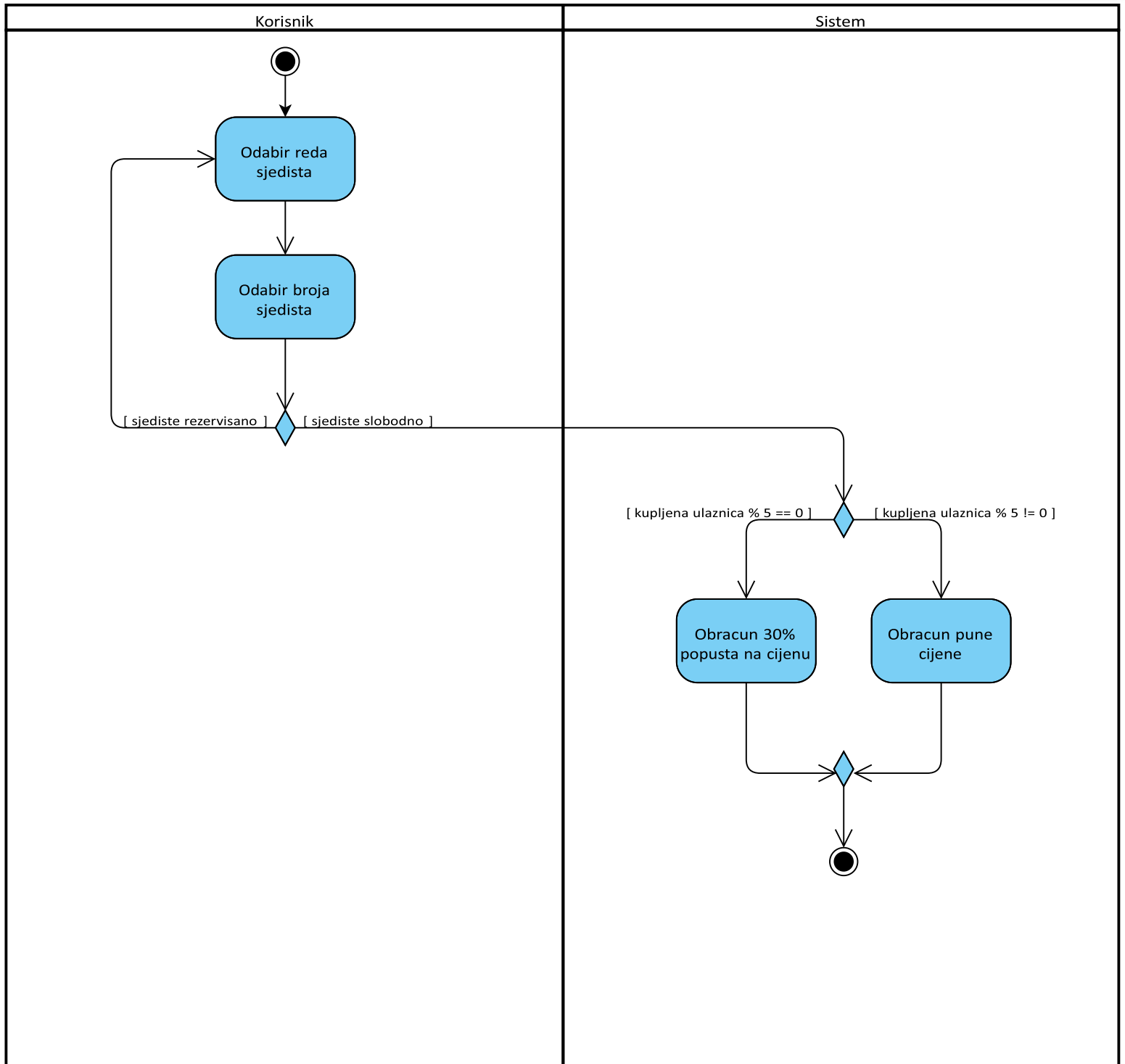
Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
Kupljene karte	Array<Karta>	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Odabir i kupovina nove karte

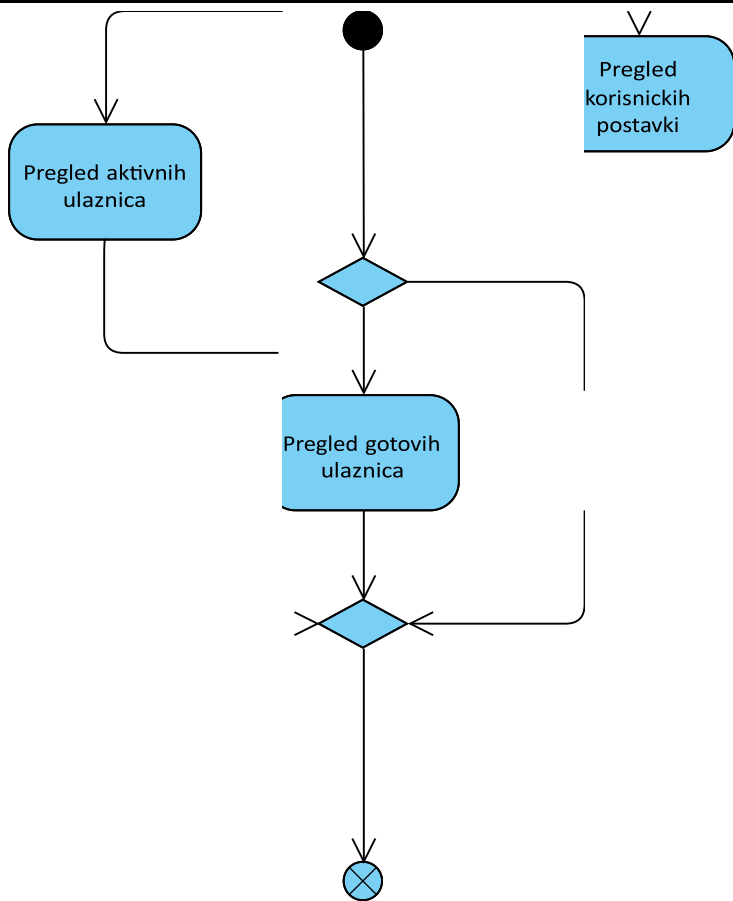


Odabir dostupnog mjesta u sali i obracun cijene ulaznice

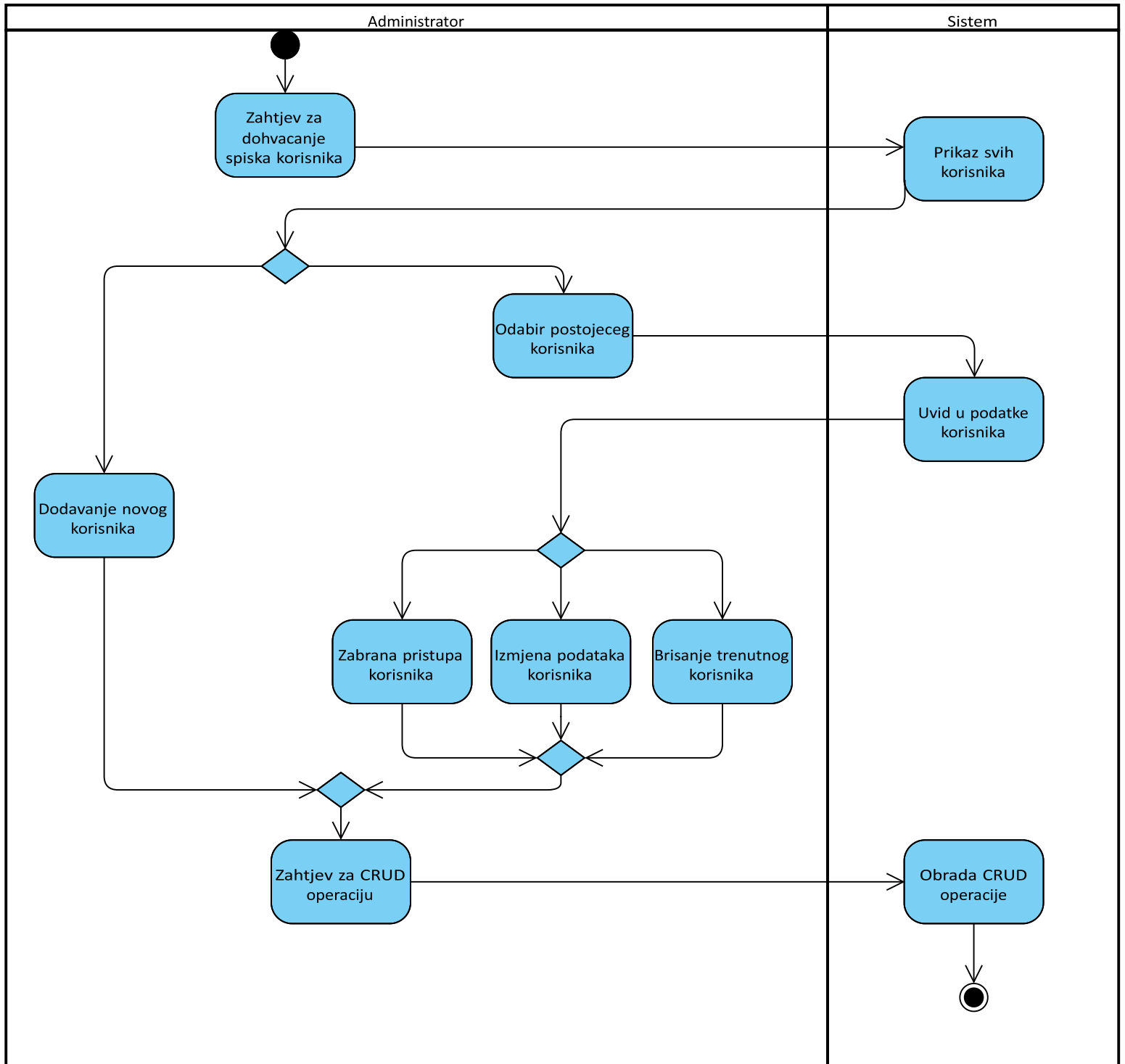


Pregled korisnickog racuna

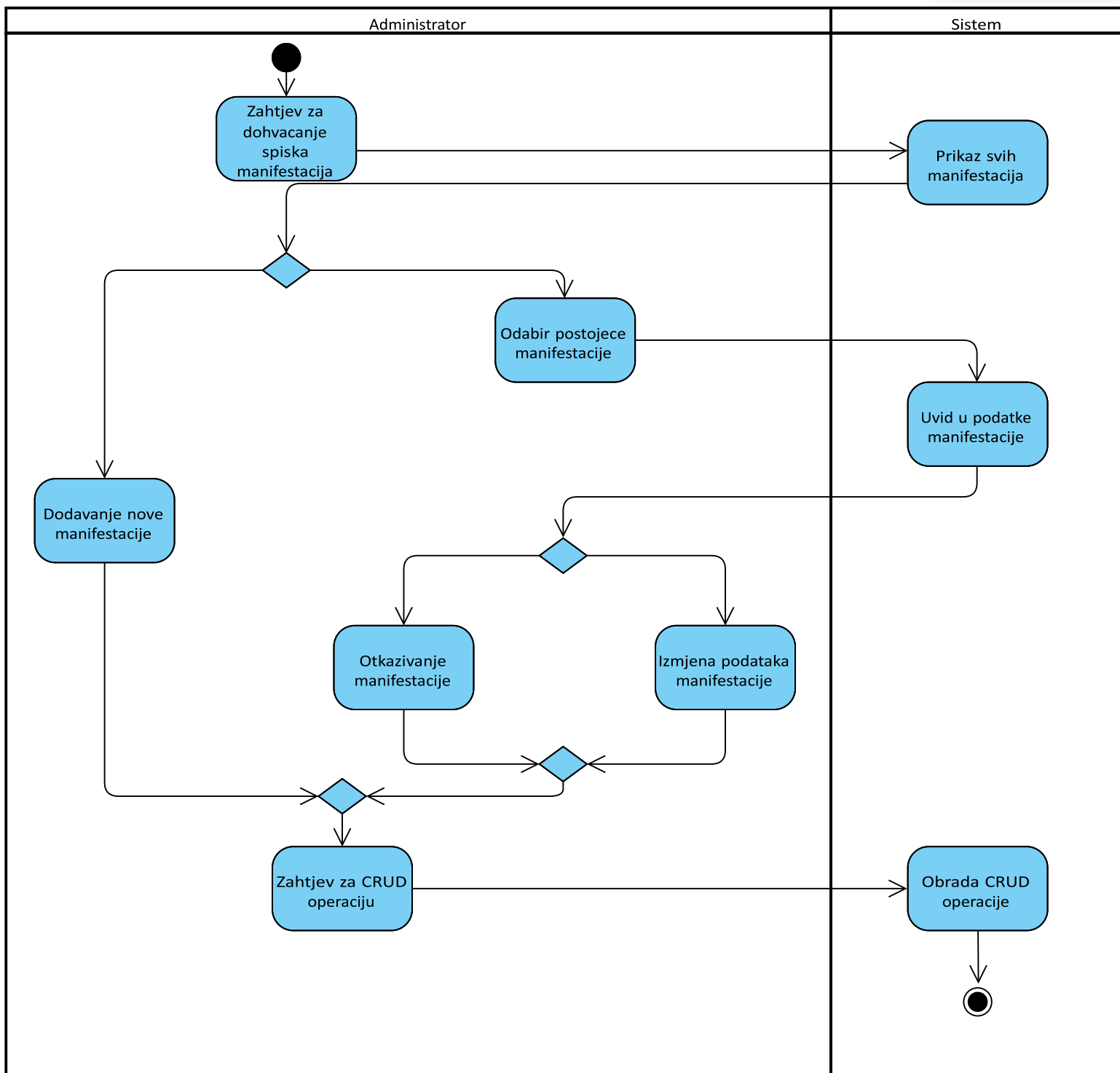
Korisnik	Sistem
----------	--------



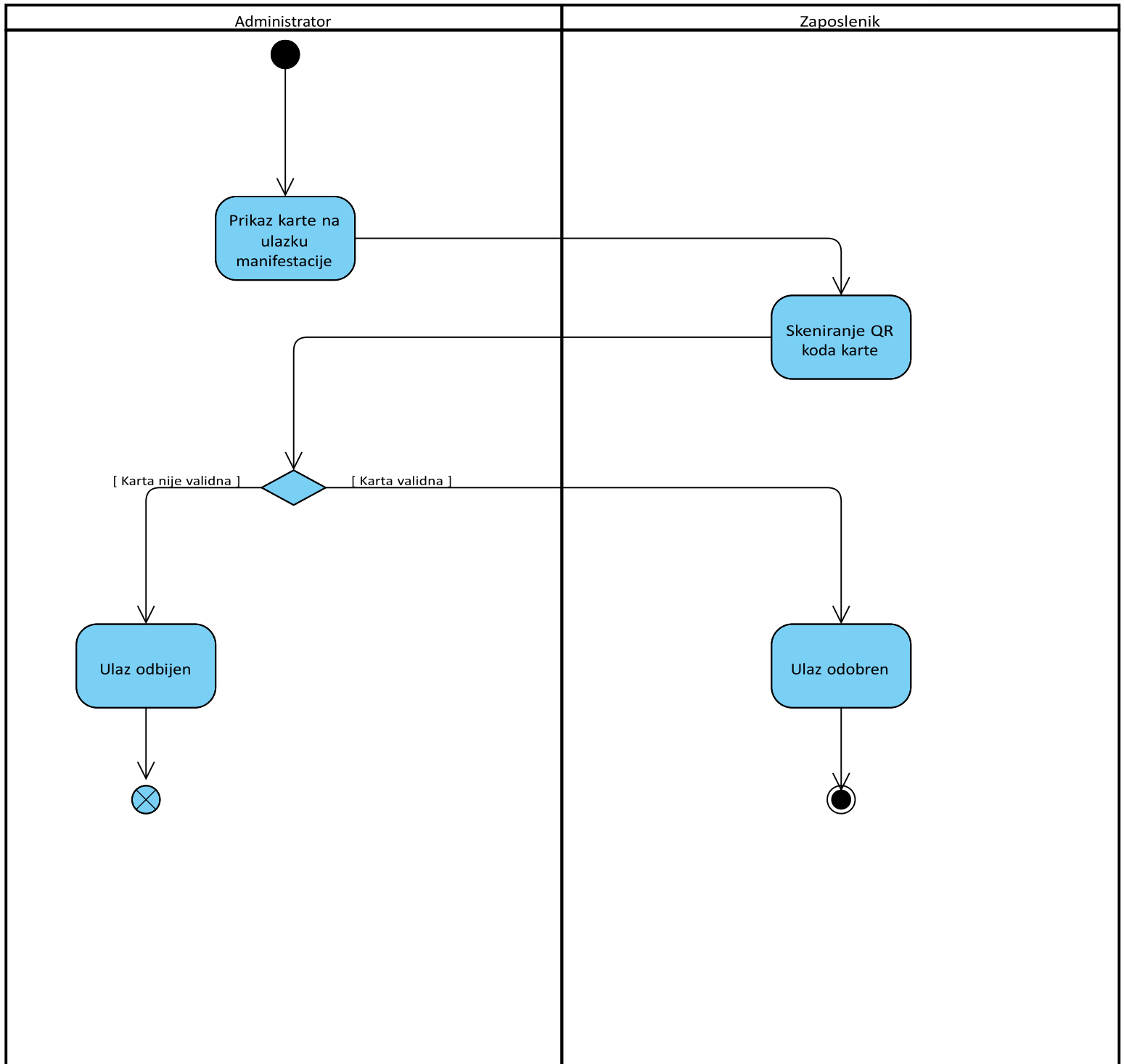
Uvid u korisnike



Kontrola manifestacija

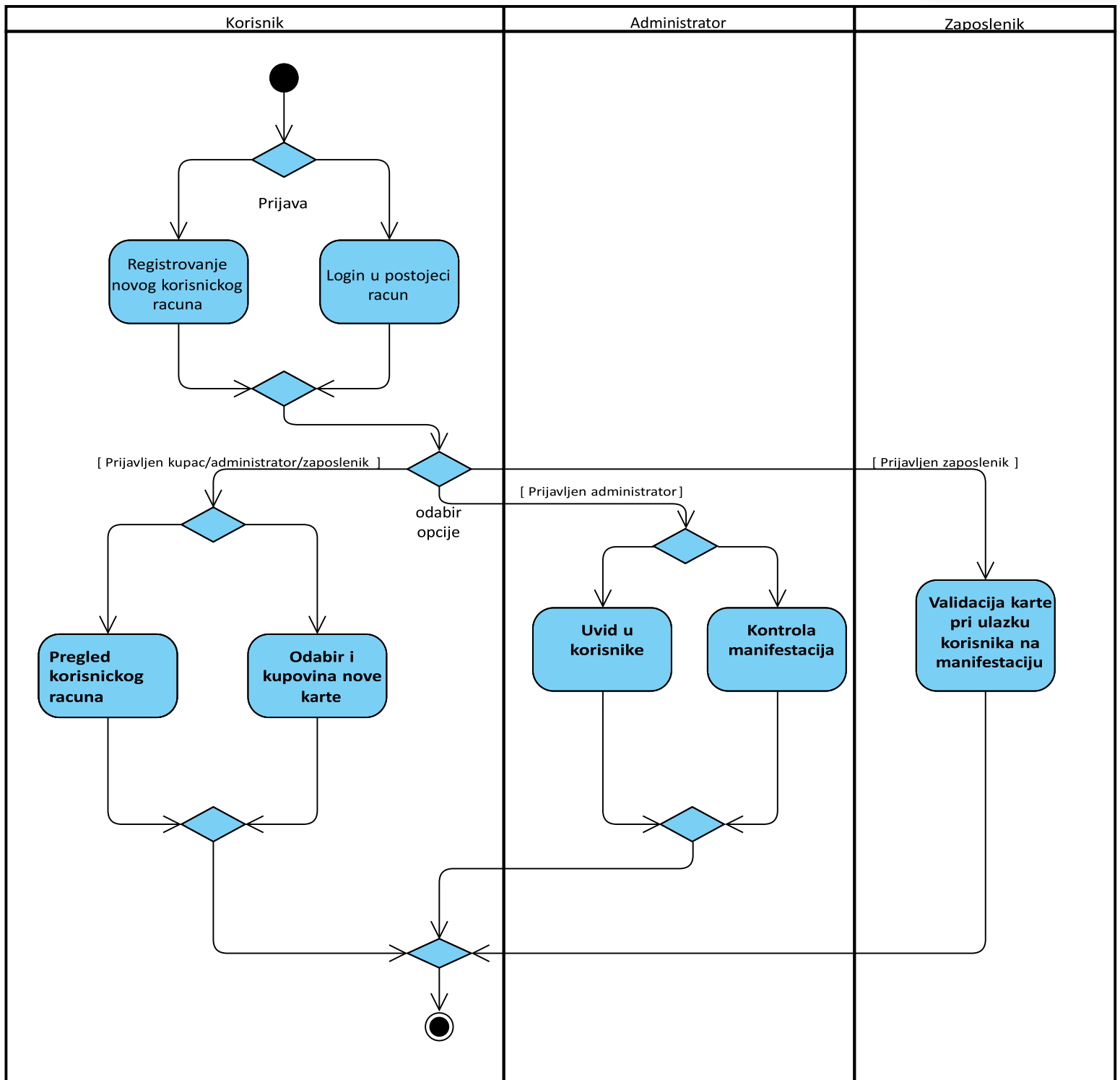


Validacija karte pri ulasku korisnika na manifestaciju



Kontrola korisnickog racuna





DIJAGRAM KLASA – POSTIVANJE SOLID PRINCIPA

1. User (Korisnik)

SRP: Klasa je zadužena samo za pohranu i upravljanje informacijama o korisniku. Ne bavi se logikom poslovnih procesa ili interakcijom s bazom podataka, što je u skladu s principom jedne odgovornosti.

OCP: Može se proširiti bez modifikacije, na primjer dodavanjem novih atributa korisnika, što može biti korisno za nove zahtjeve sistema bez promjene postojećeg koda.

LSP, ISP, DIP: Ovi principi nisu direktno primjenjivi jer User djeluje kao osnovna klasa podataka bez nasljeđivanja ili komplikovanih zavisnosti.

2. Manifestacija

SRP: Fokusrana je isključivo na pohranu informacija o događajima/manifestacijama. OCP: Klasa se može proširiti dodavanjem novih atributa ili funkcija bez izmjene postojećeg koda.

LSP, ISP, DIP: Ovi principi se ne primjenjuju jer klasa ne koristi interfejse ili nasljeđivanje.

3. Karta

SRP: Klasa služi za definiranje informacija o kupljenim kartama, i to je njena jedina odgovornost.

OCP: Klasa može lako podržati dodavanje novih funkcija kao što su različite vrste popusta za karte, bez mijenjanja postojeće strukture.

LSP, ISP, DIP: Nije relevantno jer klasa ne nasljeđuje ni ne implementira interfejse.

4. Mjesto

SRP: Ova klasa ima jasnu odgovornost, pružanje informacija o lokaciji unutar nekog događaja.

OCP: Struktura klase omogućava dodavanje novih atributa, poput specifičnih oznaka mjesta (npr. VIP sekcija), bez promjene postojeće implementacije.

LSP, ISP, DIP: Ovi principi nisu direktno primjenjivi jer nema interakcije s nasljeđivanjem ili komplikovanim zavisnostima.

5. Slobodna mjesta

SRP: Održava listu slobodnih mjesta za manifestaciju. Njegova jedina odgovornost je održavanje i upravljanje dostupnim mjestima.

OCP: Može se lako prilagoditi za različite vrste manifestacija bez potrebe za izmjenom postojeće logike.

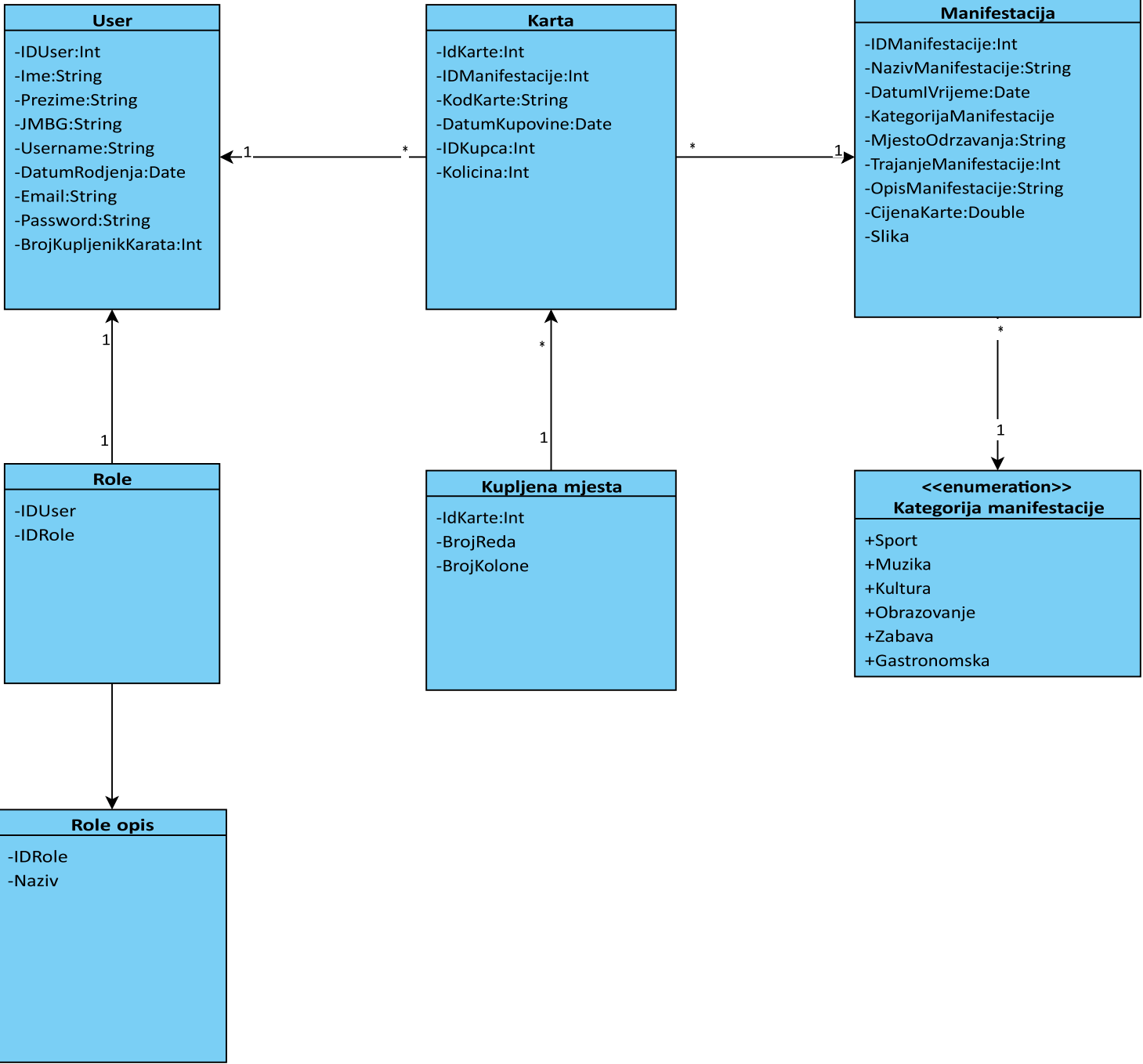
LSP, ISP, DIP: Kao i prethodne klase, ovi principi nisu direktno primjenjivi.

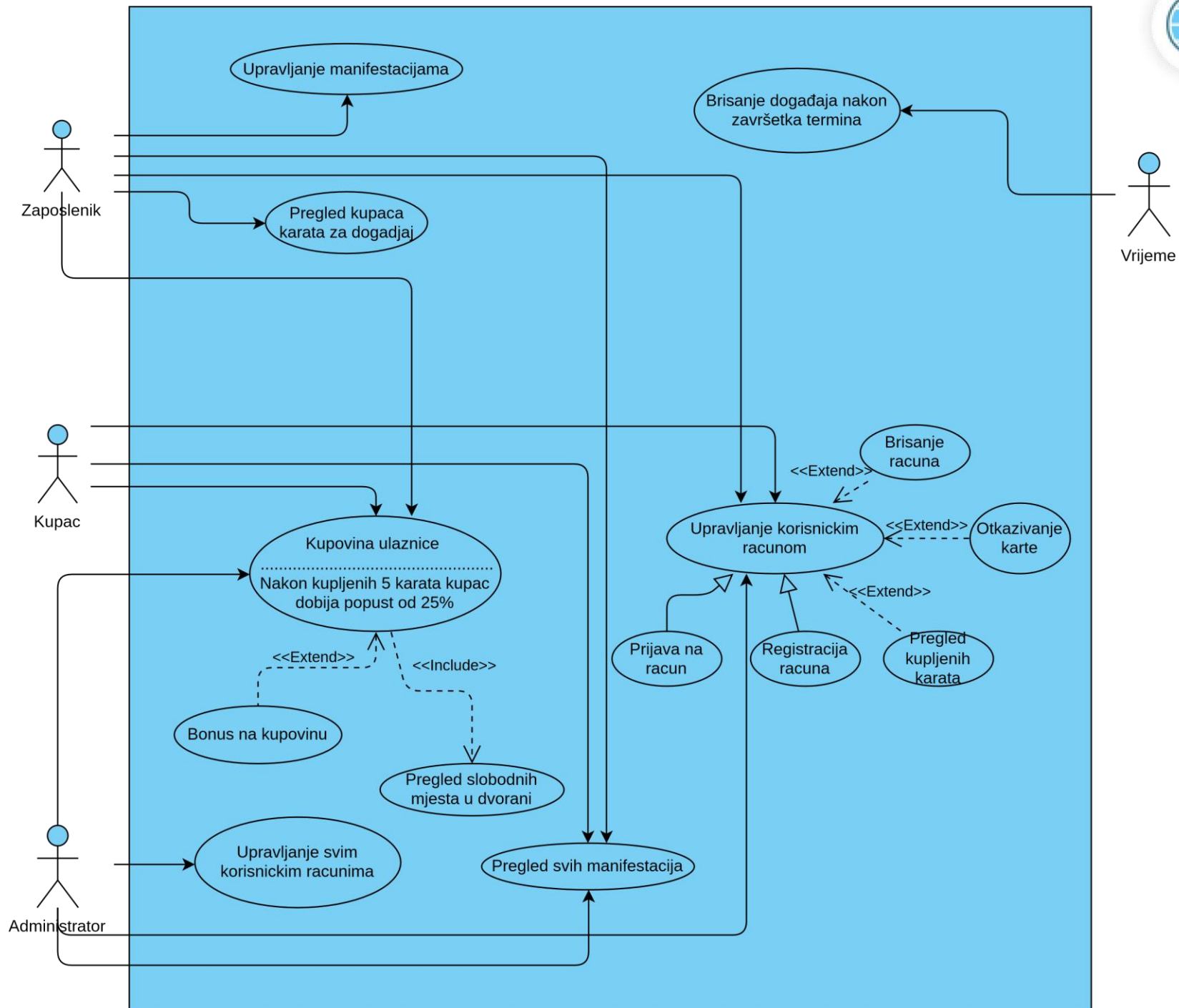
6. Historija karata

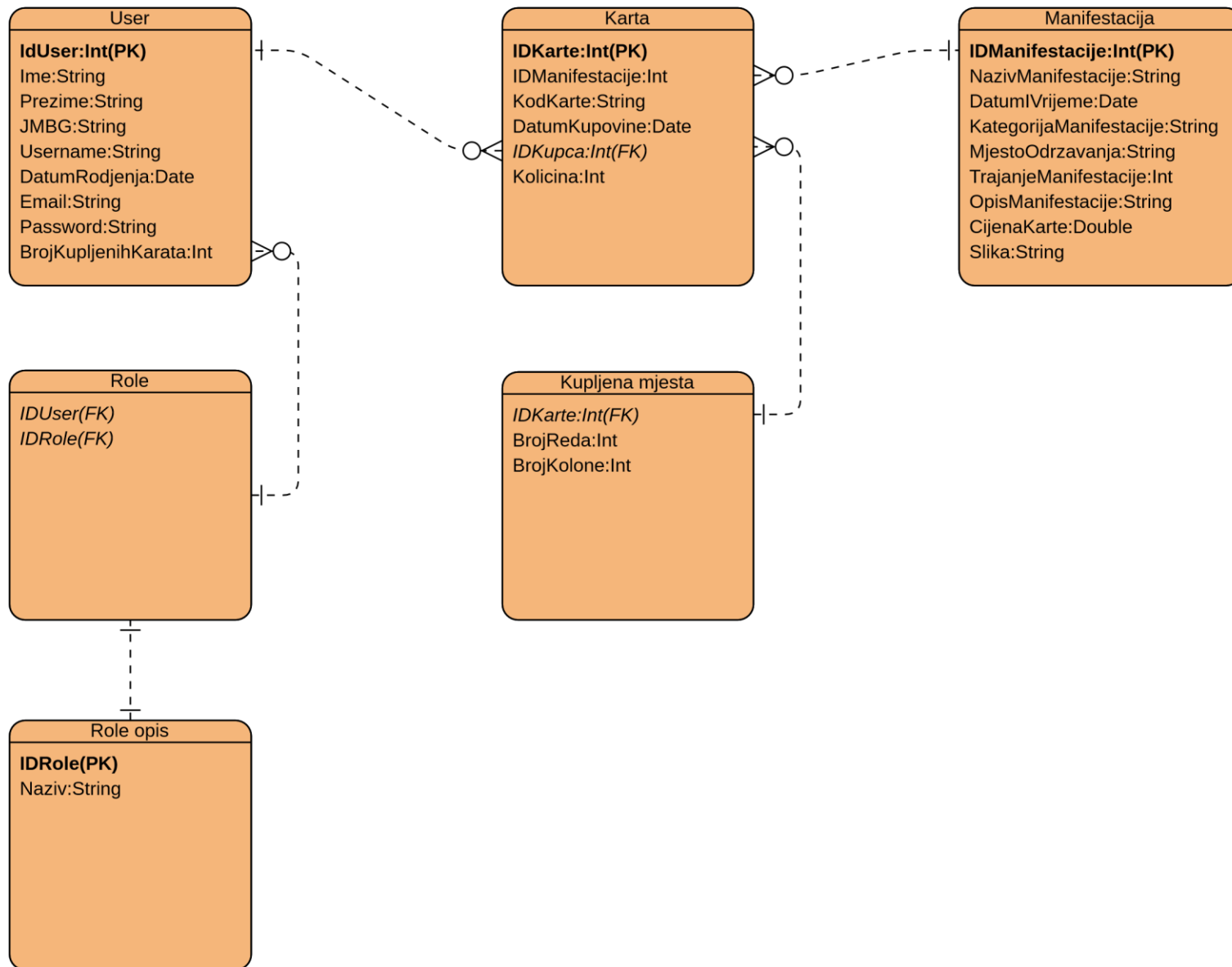
SRP: Upravlja istorijom kupovina karata od strane korisnika, što je njen osnovni zadatak.

OCP: Može se proširiti za različite vrste izvještavanja ili praćenja historije kupovine bez uticaja na postojeći kod.

LSP, ISP, DIP: Ovi principi nisu direktno primjenjivi.







PATERNI PONAŠANJA

Observer Pattern

Observer patern omogućava da objekti budu obaviješteni o promjenama stanja drugog objekta. Ovaj patern je koristan kada postoji potreba za obavještavanjem više objekata o promjenama koje su se dogodile u subjektu.

Primjer: U našem sistemu možemo koristiti Observer patern za obavještavanje korisnika o promjenama statusa njihove rezervacije karata.

Strategy Pattern

Strategy patern omogućava definiranje porodica algoritama, enkapsuliranje svakog od njih i njihovu zamjenu prema potrebi. Ovo je korisno kada postoji potreba za mijenjanjem algoritama u toku izvršavanja programa.

Primjer: U našem sistemu možemo koristiti Strategy patern za različite načine obračuna popusta na karte.

Command Pattern

Command patern omogućava enkapsulaciju zahtjeva kao objekta, omogućavajući parametrijizaciju klijenata sa različitim zahtjevima, stavljanje zahtjeva u redove ili zapisivanje operacija.

Primjena: Može se koristiti za upravljanje različitim akcijama u korisničkom interfejsu (npr. poništavanje i ponavljanje akcija rezervacije karata).

Chain of Responsibility Pattern

Chain of Responsibility patern omogućava da se zahtjev prosljeđuje kroz lanac obrađivača dok ga jedan od njih ne obradi.

Primjena: Može se koristiti za obradu različitih zahtjeva korisnika, kao što su zahtjevi za povrat novca ili reklamacije.

Mediator Pattern

Mediator patern omogućava da objekti komuniciraju preko posrednika, smanjujući time međusobne zavisnosti između njih.

Primjena: Može se koristiti za koordinaciju komunikacije između različitih modula sistema, kao što su moduli za rezervaciju karata, plaćanje, i korisničku podršku.

State Pattern

State pattern omogućava objektima da mijenjaju svoje ponašanje kada im se promijeni stanje.

Primjena: Može se koristiti za upravljanje različitim stanjima rezervacije karata (npr. kreirana, potvrđena, otkazana).

Template Method Pattern

Template Method pattern definira skeleton algoritma u metodi, ostavljajući implementaciju koraka podklasama.

Primjena: Može se koristiti za definisanje opšteg procesa obrade narudžbi, gdje specifični koraci kao što su validacija plaćanja ili slanje potvrde mogu biti definirani u podklasama.

Visitor Pattern

Visitor pattern omogućava definiranje nove operacije bez mijenjanja klasa objekata nad kojima se operacija izvršava.

Primjena: Može se koristiti za izvođenje različitih operacija nad kolekcijom objekata, kao što su generisanje izvještaja o prodaji karata.

Zaključak

Dodavanje paterna ponašanja u postojeći sistem za online rezervaciju i prodaju karata donosi mnoge prednosti, uključujući povećanu fleksibilnost, lakšu održivost i proširivost sistema.

Primjena Observer i Strategy paterna pomaže u rješavanju specifičnih problema vezanih za obavještanje korisnika i fleksibilnost algoritama popusta. Ostali paterni ponašanja mogu se primijeniti u različitim dijelovima sistema kako bi se dodatno poboljšala njegova funkcionalnost i efikasnost. Na taj način, sistem postaje modularniji, lakši za proširenje i održavanje.

KREACIJSKI PATERNI

Kreacijski paterni u dizajnu softvera su ključni za rješavanje problema stvaranja objekata na fleksibilan i efikasan način. Oni omogućavaju kontrolu nad stvaranjem objekata, olakšavaju promjene u načinu kreiranja objekata, i pomažu u upravljanju kompleksnošću sistema. U ovom dokumentu ćemo objasniti potrebu za dodavanjem kreacijskih paterna u postojeći sistem za online rezervaciju i prodaju karata. Fokusirat ćemo se na primjenu Factory Method i Singleton paterna, te ćemo objasniti gdje i kako bi se ostali kreacijski paterni mogli primijeniti.

Primjena Kreacijskih Paterna

Factory Method Pattern

Factory Method pattern omogućava stvaranje objekata pomoću metoda koji vraćaju instance različitih klasa, čuvajući pritom apstrakciju i izbjegavajući direktno korištenje konkretnih klasa.

Primjer:

U našem sistemu možemo koristiti Factory Method patern za kreiranje različitih tipova karata (VIP, standardna, studentska, itd.) bez direktnog instanciranja tih klasa.

Singleton Pattern

Singleton patern osigurava da postoji samo jedna instanca određene klase i pruža globalnu tačku pristupa toj instanci. Ovo je korisno za upravljanje resursima kao što su konfiguracije, konekcije prema bazama podataka, ili upravljanje sesijama korisnika.

Primjer:

U našem sistemu možemo koristiti Singleton patern za upravljanje konekcijama prema bazi podataka.

Ostali Kreacijski Paterni

Builder Pattern

Builder patern omogućava postepenu izgradnju kompleksnih objekata. Umjesto kreiranja objekata kroz kompleksne konstruktore, koristi se niz koraka koji grade krajnji objekat.

Primjena:

Može se koristiti za kreiranje kompleksnih entiteta kao što su Manifestacija sa mnogim opcionalnim atributima (npr. naziv, datum, lokacija, lista izvođača).

Prototype Pattern

Prototype patern omogućava stvaranje novih objekata kopiranjem postojećih, umjesto kreiranja novih instanci. Ovo je korisno kada je kreiranje objekta skupo ili kompleksno.

Primjena:

Može se koristiti za kreiranje duplikata Karte sa sličnim atributima.

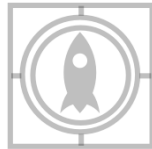
Abstract Factory Pattern

Abstract Factory patern omogućava kreiranje familije povezanih objekata bez specifikacije njihovih konkretnih klasa. Koristi se kada postoji potreba za kreiranjem različitih setova objekata.

Primjena:

Može se koristiti za kreiranje različitih Tipova Karata i Plaćanja.

Dodavanje kreacijskih paterna u postojeći sistem za online rezervaciju i prodaju karata donosi mnoge prednosti, uključujući fleksibilnost, ponovnu upotrebljivost koda, i lakšu održivost sistema. Primjena Factory Method i Singleton paterna pomaže u rješavanju specifičnih problema vezanih za kreiranje objekata u našem sistemu, dok se ostali kreacijski paterni mogu koristiti za rješavanje drugih složenih situacija. Na taj način, sistem postaje modularniji, lakši za proširenje i održavanje.



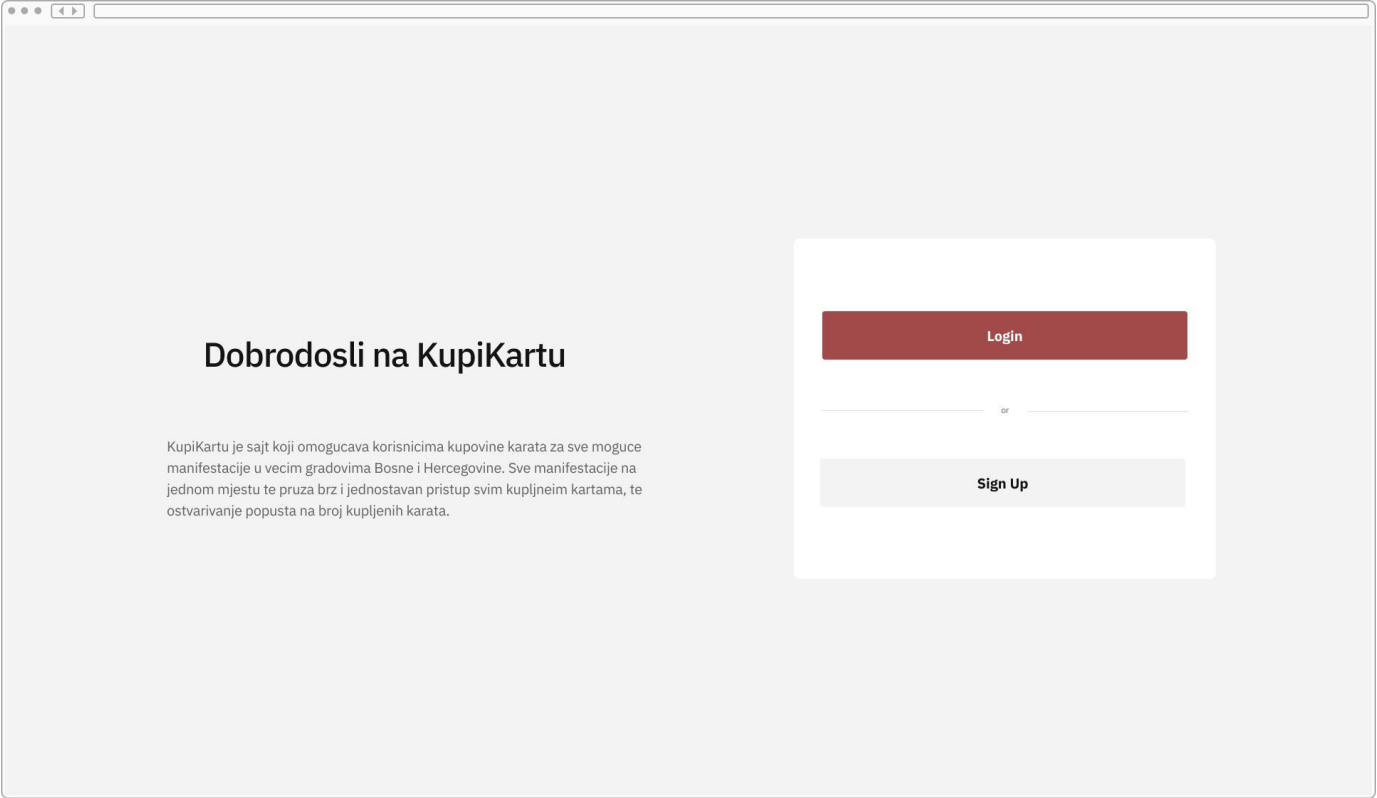
KupiKartu

Hamza Hrnjić



10 screens

Screen 1



Login

Username

Password

Login

Go Back

Sign Up

Ime

Prezime

JMBG

Email

Username


Password

Datum rođenja


Sunday 21th April 2024 20:56

Sign Up

Go Back




Hamza Hrnjic




30 Apr 2024

MAJA BEROVIC




30 Apr 2024

LEPA BRENA




30 Apr 2024

HARIS DZINOVIC




07. april u 18h

EMINA JAHOVIC



DATE : 14. 10. PETAK

JALA BRAT
BUBA CORELLI



17/11/2023

ULAZ OD 17.30

KONCERT OD 20.00

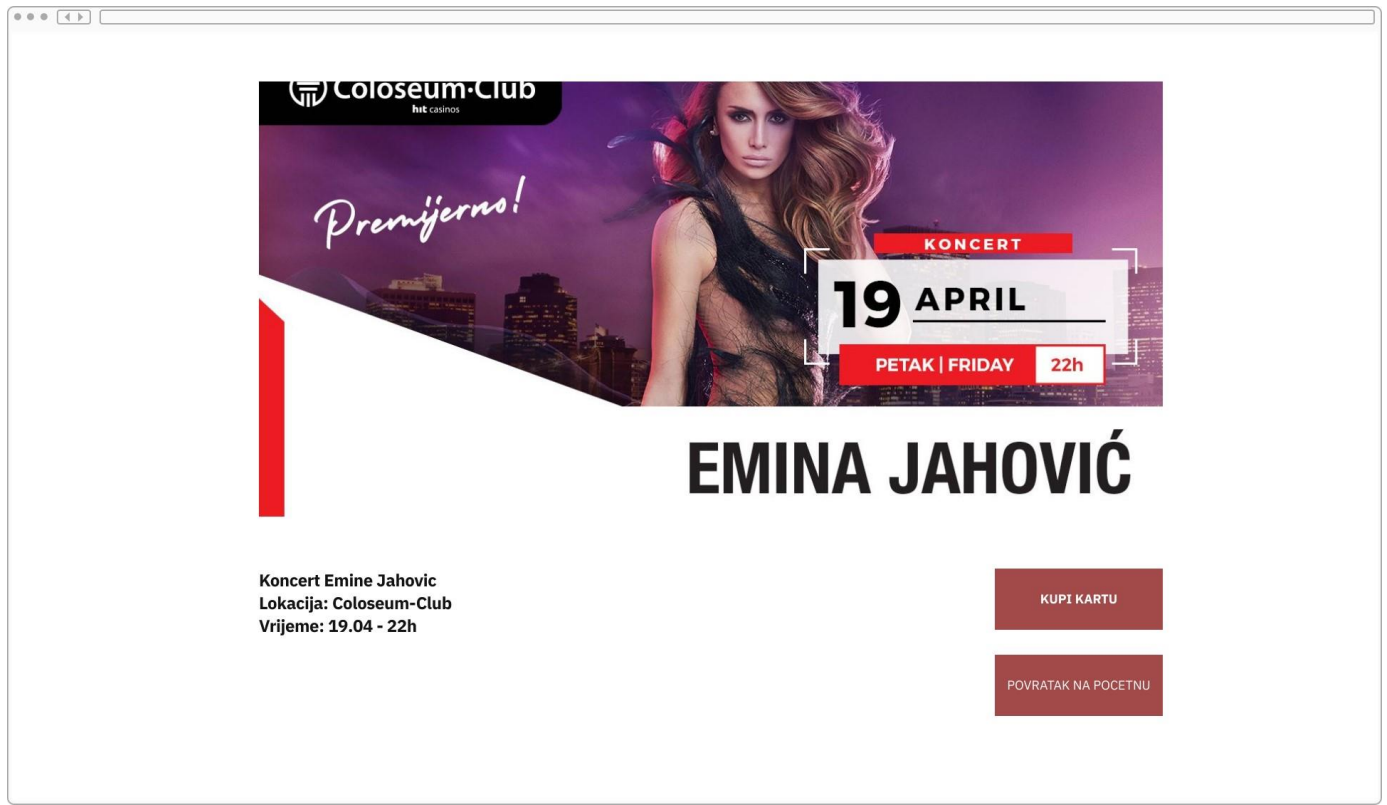
DINO MERLIN

BEOGRAD Štark Arena

made with proto.io

exported on 01/05/2024, 00:37

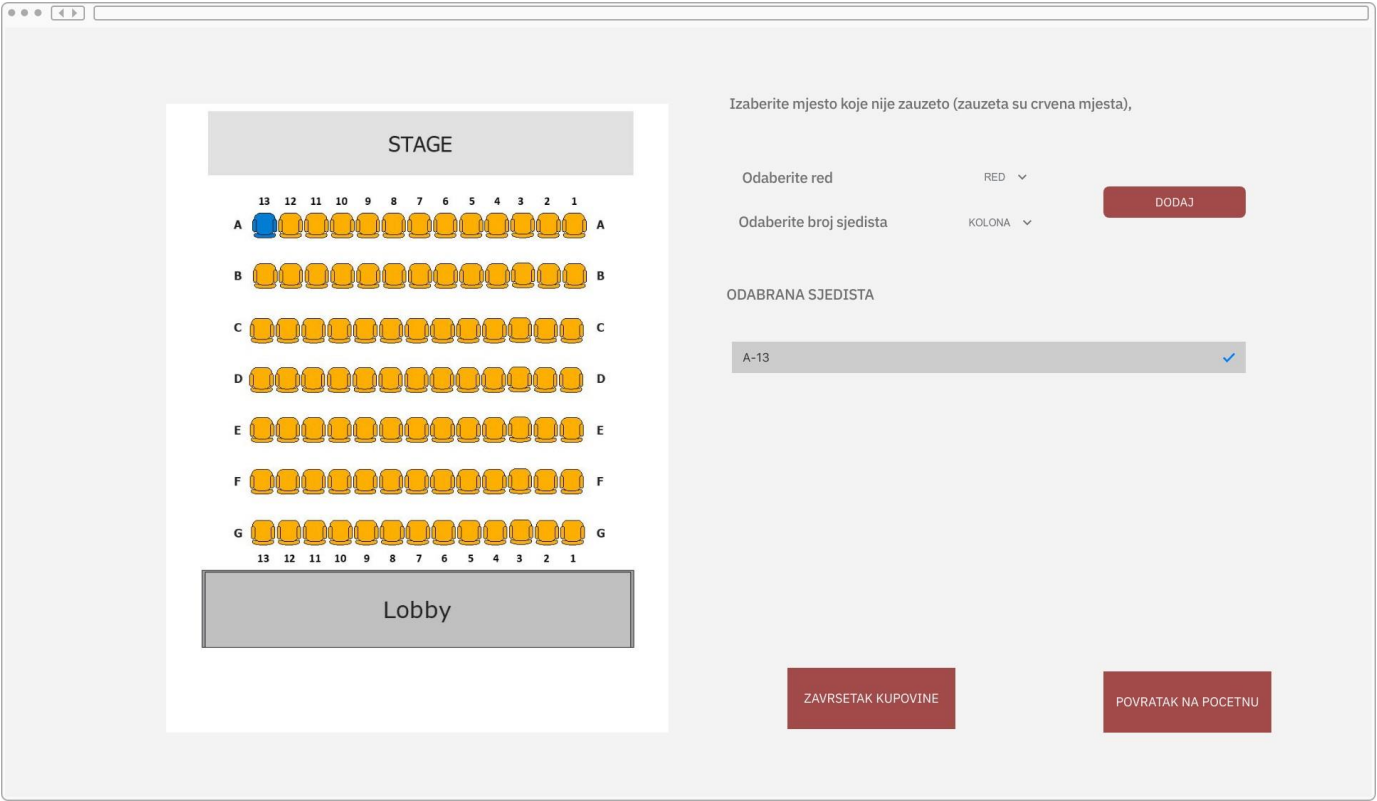
5/11



Koncert Emine Jahovic
Lokacija: Colosseum-Club
Vrijeme: 19.04 - 22h

KUPI KARTU


POVRATAK NA POCETNU



KupiKartu

Pregled Karata


Nadolazeci dogadjaji



Koncert Emine Jahovic

19.04.2024


[vрати kartu](#)



Koncert Lepe Brene

20.04.2024


[vрати kartu](#)



Koncert Maja Berovic

07.05.2024

[vрати kartu](#)




Koncert Jala Brat Buba Coreli

08.08.2024

[vрати kartu](#)


Prethodni dogadjaji



Predstava Enisa Beslagica

21.01.2024

[vрати kartu](#)



Koncert Dino Merlin

23.02.2024

[vрати kartu](#)

Odabir datuma dogadjaj

<

JAN 18

>

MO	TU	WE	TH	FR	SA	SU
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

POVRATAK NA POCETNU

made with proto.io

exported on 01/05/2024, 00:37

9/11

Promjena lozinke


Stara lozinka

Nova lozinka

Ponovi novu lozinku

PROMJENI

NAZAD



Colosseum-Club
hit casinos

Premijerno!

KONCERT

19 APRIL

PETAK | FRIDAY **22h**

EMINA JAHOVIĆ

Vrijeme do kojeg se može otkazati: 17.04.2024

Moguće je otkazivanje do naznačenog vremena,
nakon toga sve karte se mogu a i ne moraju iskoristiti

OTKAZI

NAZAD

STRUKTURALNI PATTERNI ZA KUPIKARTU

1) ADAPTER PATTERN

Adapter patern omogućava rad dviju nespojivih interfejsa. U našem slučaju, možemo ga koristiti za integraciju različitih API-ja za plaćanje ili različitih baza podataka.

Npr. `PlaćanjeAdapter`: omogućava integraciju različitih metoda plaćanja (PayPal, Stripe, bankovni transferi) koristeći jedinstveni interfejs za plaćanje

2) FACADE PATTERN

Facade patern pruža jednostavniji interfejs za kompleksan sistem. Može se koristiti za pojednostavljenje interakcija s podsistemima kao što su upravljanje korisnicima, upravljanje manifestacijama, i sl.

Npr. `RezervacijskiSistem`: Fasada koja objedinjuje funkcionalnosti upravljanja korisnicima, manifestacijama i kartama.

3) DECORATOR PATTERN

Decorator patern se koristi za dodavanje dodatnih funkcionalnosti objektima dinamički, bez promjene njihovog koda. Ovaj patern možete koristiti za proširenje funkcionalnosti karata.

Npr. `Karta`: Može se dekorirati dodatnim funkcionalnostima poput popusta, dodatnih pogodnosti, ili VIP statusa.

4) BRIDGE PATTERN

Bridge patern razdvaja apstrakciju od njene implementacije, omogućavajući nezavisnu promjenu obje strane. Ovo je korisno kada trebamo proširiti obje strane hijerarhije (apstrakciju i implementaciju) bez promjene postojeće logike.

Npr. `Karta`: Može imati različite vrste (VIP, standardna) koje se mogu implementirati pomoću različitih platformi za isporuku (fizička karta, digitalna karta).

5) PROXY PATTERN

Proxy patern koristi se za kontrolu pristupa objektu. Može se koristiti za implementaciju sigurnosnih provjera ili keširanja.

Npr. `KartaProxy`: Kontrolirane pristup stvarnoj klasi `Karta`, omogućavajući provjeru valjanosti ili autorizacije prije izvršenja operacija.

6) COMPOSITE PATTERN

Composite patern omogućava tretiranje pojedinačnih objekata i njihovih kolekcija uniformno. Koristan je za prikaz složenih hijerarhija podataka.

Npr. Sjediste: Može se organizirati kao kompozitna struktura koja omogućava tretiranje pojedinačnih sjedišta i sekcija sa sjedištima na isti način.

7) FLYWEIGHT PATTERN

Flyweight patern smanjuje korištenje memorije dijeljenjem što je moguće više stanja između objekata. Korisno je kada imamo veliki broj sličnih objekata.

Npr. Mjesto: Možemo koristiti Flyweight patern za upravljanje sjedištima u dvorani, gdje svako sjedište može dijeliti zajedničke podatke poput reda i kolone.

SCENARIJI ZA DIJAGRAM SLUČAJA UPOTREBE

Primjer scenarija: **Popust na broj kupljenih karata**

Naziv slučaja upotrebe	Mogućnost ostvarivanja popusta
Opis slučaja upotrebe	Svaki kupac karte će imati mogućnost ostvarivanja popusta na kupovinu karata
Vezani zahtjevi	/
Preduslovi	• Kupovina ulaznica
Posljedice - uspješan završetak	Ukoliko korisnik je prethodno kupio 5 karata na 6-toj ostvaruje popust (25%)
Posljedice - neuspješan završetak	Korisnik nije prethodno kupio 5 karata tako da plaća po regularnoj cijeni
Primarni akteri	Kupac karte
Ostali akteri	/
Glavni tok	Kupac karte se logira na svoj račun nakon registracije, kupuje kartu, te se automatski racuna koliko je kupio karata. Ukoliko je ostvario popust sljedeći put se racuna popust na cijenu.

Tok događaja - Uspješan završetak

Korisnik	Sistem
1. Korisnik uspješno kupuje kartu	
	2. Sistem zna da na sljedećoj karti obračunava popust
3. Korisnik ponovno kupuje neku drugu kartu i prikazuje mu se popust	
	4. Sistem ponovno počinje brojanje kupljenih karti

Tok događaja - Neuspješan završetak

Korisnik	Sistem
1. Korisnik kupuje kartu	
	2. Sistem još uvijek vidi da korisnik nije kupio više od 5 karti

<ul style="list-style-type: none"> Korisnik ponovno kupuje neku drugu kartu i ne prikazuje mu se popust 	
	<ul style="list-style-type: none"> Sistem nastavlja brojanje sve do navedenog broja karti

Primjer scenarija: **Prikaz slobodnih mjesta unutar dvorane**

Naziv slučaja upotrebe	Prikaz slobodnih mjesta unutar dvorane, te broj raspoloživih karata
Opis slučaja upotrebe	Korisnik ima mogućnost pregleda mjesta unutar dvorane, te broj karata koje se mogu prodati
Vezani zahtjevi	/
Preduslovi	<ul style="list-style-type: none"> Prikaz svih manifestacija koje se održavaju te njihovo vrijeme
Posljedice - uspješan završetak	Korisnik uspješno bira mjesto koje želi i manifestaciju
Posljedice - neuspješan završetak	Korisnik nije stigao na vrijeme izabrati mjesto na manifestaciji koje je želio
Primarni akteri	Kupac karte
Ostali akteri	/
Glavni tok	Kupac karte pregleda manifestacije koje su dostupne, bira manifestaciju koju želi posjetiti. Kada izabere manifestaciju, pojave mu se dvorana i slobodna sjedišta, te ih on bira

Tok događaja - Uspješan završetak

Korisnik	Sistem
1. Kupac karte pronalazi manifestaciju koju želi posjetiti	
	2. Pokazuje mu broj karata i slobodna mjesta u dvorani
3. Kupac bira mjesto gdje će sjediti, bira ga po svojoj želji, ukoliko je ono slobodno	

Tok događaja - Neuspješan završetak

Korisnik	Sistem
-----------------	---------------

1. Kupac karte pronalazi manifestaciju koju želi posjetiti	
	2. Pokazuje mu broj karata i slobodna mjesta u dvorani
3. Kupac bira mjesto gdje će sjediti, ali ga ne bira po svojoj želji, već samo ono dostupno	

Primjer scenarija: **Prijava/login**

Naziv slučaja upotrebe	Upravljanje korisničkim računom
Opis slučaja upotrebe	Svaka osoba će imati mogućnost prijave i registracije svog računa. U zavisnosti od toga da li je osoba kupac karte, zaposlenik ili administrator, imat će različit interfejs početnog prozora i funkcionalnosti.
Vezani zahtjevi	/
Preduslovi	/
Posljedice - uspješan završetak	Korisnik pristupa sistemu i dobija određene mogućnosti sistema
Posljedice - neuspješan završetak	/
Primarni akteri	Kupac karte, administrator, zaposlenik
Ostali akteri	/
Glavni tok	Korisniku se nakon prijave pojavljuje prozor sa određenim funkcionalnostima. Kod kupca karte se pojavljuje mogućnost odabira događaja, kupnje karte. Kod zaposlenika mogućnost dodavanja predstava i admin ima sve mogućnosti

Tok događaja - Uspješan završetak

Korisnik	Sistem
1. Kupac karte, administrator i zaposlenik popunjavaju podatke pri kreiranju računa	
	2. Korisniku pokazuje da se uspješno registrovao i da je račun kreiran

Tok događaja - Uspješan završetak

Korisnik	Sistem
1. Korisnik se loginuje na račun.	
	2. Korisniku pokazuje da se uspješno logirao i dozvoljava mu određene
	funkcije

Tok događaja - Neuspješan završetak

[Nema neuspješnog završetka registracije](#)

Tok događaja - Neuspješan završetak

Korisnik	Sistem
1. Korisnik se loginuje na račun.	
	2. Korisniku pokazuje da nije uspješno loginovao na račun zbog pogrešnih podataka

Primjer scenarija: **Prikaz manifestacije i njihovo vrijeme**

Naziv slučaja upotrebe	Prikaz svih manifestacija koje se održavaju te njihovo vrijeme
Opis slučaja upotrebe	Svaki korisnik će imati mogućnost pregleda manifestacija koje se održavaju te i njihovo vrijeme održavanja
Vezani zahtjevi	/
Preduslovi	• Upravljanje korisničkim računom
Posljedice - uspješan završetak	Korisnik je kreirao ili se ulogirao na svoj račun te dobio opciju pregleda manifestacija koje se održavaju a i njihovog vremena održavanja
Posljedice - neuspješan završetak	Korisnik se ne može login-at na site
Primarni akteri	Kupac karte
Ostali akteri	/
Glavni tok	Korisnik se registrira ili ulogira na svoj račun. Sa tom radnjom on postaje validan u pregledu manifestacija i njihovog vremena, te na osnovu toga bira na koju će manifestaciju ići i kad mu odgovara..

Tok događaja - Uspješan završetak

k	m
1. Korisnik kreira ili ulogira na svoj račun	
	2. Sistem omogućava prikaz manifestacija i vremena događanja manifestacija

Tok događaja - Neuspješan završetak

Nema neuspešnog završetka zbog toga što je pregled lokacije održavanja događaja osnovna funkcija sistema

Primjer scenarija: **Upravljanje kartama**

Naziv slučaja upotrebe	Upravljanje kartama
-------------------------------	---------------------

Korisni	Siste
----------------	--------------

Opis slučaja upotrebe	Omogućava da administrator i zaposlenik dodaje, briše, uređuje, pregleda karte.
Vezani zahtjevi	/
Preduslovi	/
Posljedice - uspješan završetak	Administrator ili zaposlenik želi da obriše karte sa prodajnih karata ili da doda moguću kartu koja je višak sistem mu to omogući i karte se nađu u prodajnim kartama ili izbrisane budu
Posljedice - neuspješan završetak	Administrator ili zaposlenik želi da doda kartu u prodaju ali sistem mu prikaže previse karata je u prodaji
Primarni akteri	Administrator, zaposlenik
Ostali akteri	/
Glavni tok	Administrator u svojoj želji (ili zaposlenik ali u konsultacijama sa adminom) dodaje, briše, uređuje i gleda moguće karte.

Tok događaja - Uspješan završetak

Korisnik	Sistem
1. Administrator ili zaposlenik dodaje kartu unutar prodaje	
	<ul style="list-style-type: none"> Sistem mu omogućava dodaju karte jer nema granice za mjesta
<ul style="list-style-type: none"> Administrator ili zaposlenik se predomislio i briše kartu 	
	<ul style="list-style-type: none"> Sistem mu omogućava brisanje karte

Tok događaja - Neuspješan završetak

Korisnik	Sistem
1. Administrator ili zaposlenik dodaje kartu unutar prodaje	
	2. Sistem mu nije omogućio tu radnju zbog ograničenja mjesta

Primjer scenarija: **Kupovina karte**

Naziv slučaja upotrebe	Kupovina ulaznica
Opis slučaja upotrebe	Korisnik kupuje ulaznicu za željenu manifestaciju
Vezani zahtjevi	/
Preduslovi	/
Posljedice - uspješan završetak	Korisnik pregleda moguće manifestacije te kupuje kartu za željenu manifestaciju
Posljedice - neuspješan završetak	Korisnik pregleda moguće manifestacije te kupuje kartu za željenu manifestaciju ali sistem mu javlja da su sve karte rasprodane
Primarni akteri	Kupac karte
Ostali akteri	/
Glavni tok	Kupac karte nakon login ili registracije na profil prikazuju mu se sve manifestacije te odabira željenu i kupuje ulaznicu za željenu manifestaciju

Tok događaja - Uspješan završetak

Korisnik	Sistem
1. Kupac karte se loguje i registrira na račun i na osnovu prikazanih manifestacija kupuje kartu.	
	<ul style="list-style-type: none">• Sistem omogućava kupovinu karte.

Tok događaja - Neuspješan završetak

Korisnik	Sistem
1. Kupac kartese loguje i registrira na račun i na osnovu prikazanih manifestacija kupuje kartu.	
	2. Sistem provjera broj slobodnih karata za prodaju, posto su sve rasprodane odbija kupovinu karte uz poruku da su sve prodane

Primjer scenarija: **Povratak kupljene karte**

Naziv slučaja upotrebe	Povrat karte
Opis slučaja upotrebe	Mogućnost povrata karte u određenom roku
Vezani zahtjevi	/
Preduslovi	1) Upravljanje korisničkim računom 2) Kupovina ulaznica
Posljedice - uspješan završetak	Korisnik se predomislio sa kupovinom karata i želi da je vrati. Pri kupovini mu piše rok do kad se karta može vratiti. Korisnik pritišće opciju vrati kartu. Sistem prihvata i karte se refundira.
Posljedice - neuspješan završetak	Korisnik se predomislio sa kupovinom karte i želi da je vrati ali nije gledao rok do kad treba vratiti te pritišće opciju vrati kartu u vrijeme kad je rok prosao. Sistem mu ne prihvata povrat karte i karte se ne refundira.
Primarni akteri	Korisnik
Ostali akteri	Administrator, zaposlenik
Glavni tok	Korisnik kupovinom karte ima mogućnost povrata te iste karte u određenom roku zavisno od organizatora predstave. Ako korisnik kartu refundira u roku zadanom za povrat njegov novac se vraća njemu
	i karta će biti oduzeta od njega. Ako korisnik ne refundira kartu u roku onda mu neće doći do povrata novac i karta ostaje kod njega

Tok događaja - Uspješan završetak

Korisnik	Sistem
1. Korisnik kupuje kartu i želi je refundirati	
	<ul style="list-style-type: none"> Sistem provjerava da li se refundacija radi u dozvoljenom roku ako da onda refundira i novac je vraćen

Tok događaja - Neuspješan završetak

Korisnik	Sistem
1. Korisnik kupuje kartu i želi je refundirati	
	2. Sistem provjerava da li se refundacija radi u dozvoljenom roku ako je rok za povrat karte prosao novac se ne vraca i akcija nije uspjesna

