

Kreacijski patterni

*boldirane pattern-e ćemo sigurno primijeniti

1. Singleton pattern

Singleton pattern koristimo u slučaju kada nam je potrebno da se klasa/objekat može instancirati samo jednom i kada nam je potrebna jedinstvena kontrola pristupa. U našem sistemu, mogle bismo primijeniti navedeni pattern ukoliko bismo imale nekoliko zaposlenika među kojima bi jedan bio administrator/admin koji bi imao određene privilegije u odnosu na ostale zaposlenike. Na primjer, admin može pogledati i preuzeti izvještaj, dok ostale zaposlenike izvještaj ne zanima. Trenutno, radi jednostavnosti, mi imamo jednog zaposlenika koji naravno ima pristup izvještaju, ali u slučaju potrebe proširenja sistema, mogle bismo na ovaj način iskoristiti Singleton pattern.

2. Prototype pattern

Odlučile smo implementirati prototype pattern zbog njegove vrlo široke primjene, te u našem sistemu ima više mjesta gdje bi se mogao iskoristiti. Obzirom da su naočale primarni objekti s kojima će naš sistem raditi zaključile smo da bi od toga najviše imali koristi. Dakle možemo pattern primijeniti na kontroler ProizvodController kako bi se modifikovale naočale. Prototype dizajn pattern se koristi kada je kreiranje novih objekata skupo ili kompleksno, te umjesto kreiranja objekata iz početka, možemo klonirati postojeće objekte i modifikovati ih po potrebi. Definišimo interfejs za prototip, nazovimo ga IPrototip. Neophodno je da ovaj interfejs ima metodu kloniraj(). Imamo klasu Proizvod koja će služiti kao konkretan prototip.

3. Builder pattern

Jedan od dizajn patterna koji se često koristi u razvoju softvera radi olakšavanja kreiranja kompleksnih objekata. Ovaj pattern je vrlo koristan u implementaciji naše "Naručivanje" klase, s obzirom da se istoimena klasa sastoji od različitih atributa kao što su identifikator narudžbe (id), informacija o zalihi proizvoda, te o ukupnom računu za plaćanje. Također, veza između narudžbe i kupca je ostvarena pomoću foreign key-a (kupaId). Za kreiranje objekata klase "Naručivanje" možemo iskoristiti Builder pattern. Korištenje Builder patterna olakšava konstrukciju objekta korak po korak, što je posebno korisno kada imamo kompleksne objekte s mnogo atributa.

4. Factory method pattern

Dizajn pattern koji se koristi za kreiranje objekata bez eksplicitnog specificiranja njihovih klasa. Ovaj pattern pruža centraliziran način kreiranja objekata, što omogućava fleksibilnost i izbjegava neposrednu ovisnost o konkretnim klasama. U našem projektu, Factory pattern bi mogao biti koristan za kreiranje objekata između klasa "Korisnik" i "Naručivanje". Ovisno o različitim situacijama, svaka od ovih klasa može zahtijevati instancu objekta druge klase, a za to će nam služiti KorisnikFactory i NaručivanjeFactory respektivno.

5. Abstract Factory pattern

U našem sistemu Optika Lentium, odgovarajuće mjesto za primjenu Abstract Factory patterna bi bilo pri kreiranju različitih vrsta naočala. Abstract Factory pattern pruža interfejs za kreiranje familija povezanih ili zavisnih objekata, bez eksplicitnog navođenja njihovih konkretnih klasa. U kontekstu našeg sistema, ova šema može se primijeniti pri kreiranju različitih vrsta naočala koje se nude u online optici. Na primjer, možemo imati apstraktnu fabriku "NaočaleFabrika" koja definiše metode za kreiranje različitih vrsta naočala, kao što su "kreirajsunčane", "kreirajdnevne" itd. Svaka konkretna fabrika bi implementirala ove metode i vraćala odgovarajuće objekte naočala. Na taj način, kada korisnik aplikacije odabere određeno naočale, aplikacija može koristiti apstraktnu fabriku da bi dinamički kreirala odgovarajući objekt naočala, bez da eksplicitno zna koja je konkretna klasa u pitanju.

Strukturalni paterni

*paterni koji će biti implementirani sigurno su boldirani

1. Adapter pattern

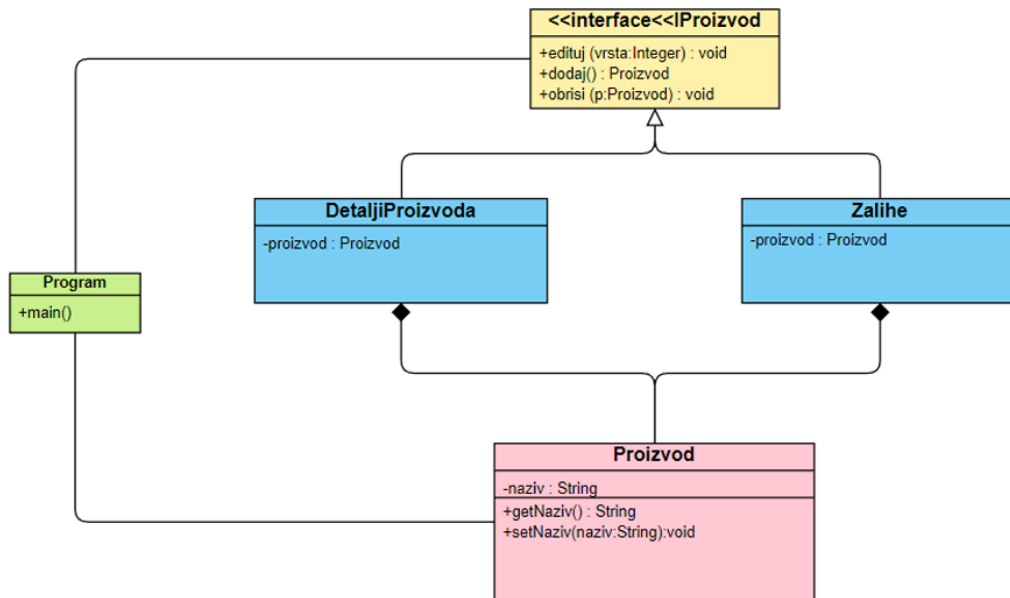
Ovaj pattern se može primijeniti prilikom pretraživanja proizvoda. Pretraga se vrši prema selectu tipa, cijene i brenda proizvoda. Rezultati pretrage se sortiraju prema poretku koji korisnik odabere. Koristimo navedeni pattern zato što nam je potreban drugačiji interface postojeće klase Proizvod, a ne želimo mijenjati klasu pa kreiramo Adapter klasu koja će služiti kao posrednik kako bismo implementirali funkcionalnost pretraživanja bez izmjene postojeće klase.

2. Facade pattern

Facade design pattern - facade pattern olakšava klijentima (kupcima, radincima, vlasniku) da koriste funkcionalnosti aplikacije bez potrebe za poznavanjem unutrašnjeg složenog sistema. Ako bi naša aplikacija imala složene backend sisteme, kao što su upravljanje inventarom, obrada narudžbi, dostava i druge operacije, Facade pattern može ograničiti izlaganje tih kompleksnosti korisnicima. Koristeći Facade, korisnici mogu pristupiti samo pojednostavljenim metodama koje su im potrebne, a kompleksne interne operacije se rješavaju unutar fasade.

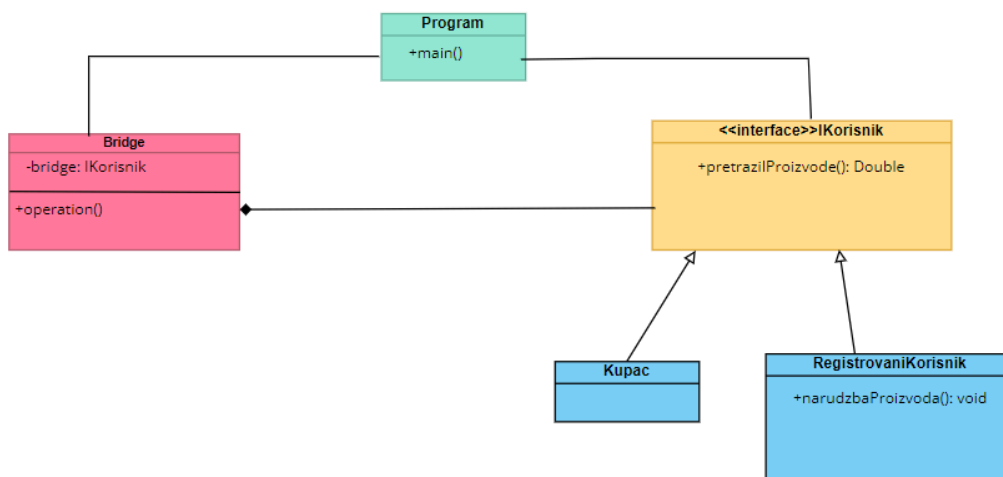
3. Decorator pattern

Radnik ima mogućnost promjene detalja o proizvodu (naziva, cijene, kategorije,...). Sistem na osnovu toga da li se konkretni proizvod nalazi u nekoj narudžbi povećavati broj prodanih primjeraka i smanjivati zalihe, da bismo imali uvid u to da li je proizvod na stanju. Promjene nad proizvodima možemo podijeliti u dvije skupine: DetaljiProizvoda (naziv, cijena, kategorija, podkategorija, opis) i Zalihe (listaNarudžbi, historijaProdaje, zaliha). Uvodimo Decorator klasu koja je povezana agregacijom s IProizvod interface-om i instancira jedan ili više IProizvod objekata.



4. Bridge pattern

Korisnik koji nije prijavljen može kao i registrovani korisnika vršiti pregled proizvoda i njegovih detalja. Ova aktivnost spaja dvije vrste korisnika i ona se sigurno izvršava. Međutim, da bi kupac odabrao opciju *Dodaj u korpu* i izvršio kupovinu, on mora biti registrovan. Ukoliko to nije slučaj, korisniku se nudi mogućnost Log in ili Sign Up kako bi mogao izvršiti kupovinu.



5. Composite pattern

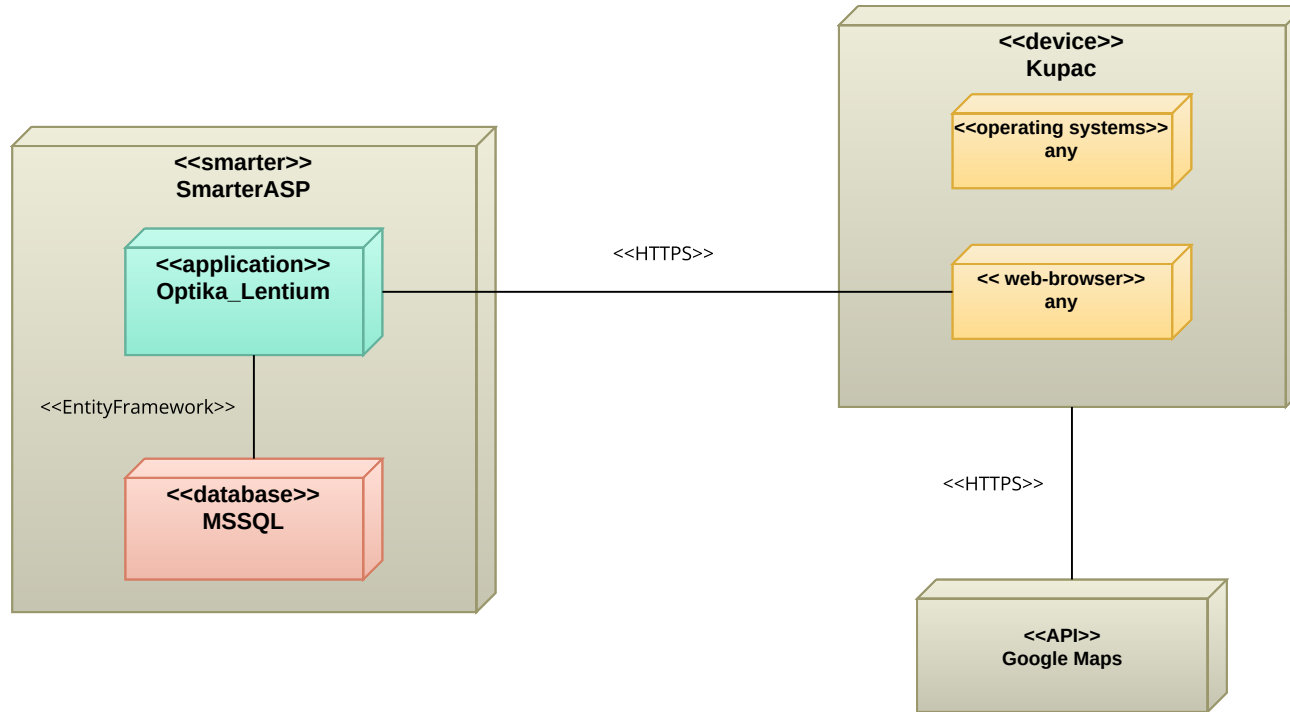
Ovaj pattern može naći primjenu u računanju ukupne cijene koju korisnik plaća za obavljanje narudžbu. Računanje bi se obavljalo, naravno, pri svakom plaćanju. Međutim, ukoliko kupac ostvari popust, cijena se koriguje. Klase će imati interface za istu svrhu.

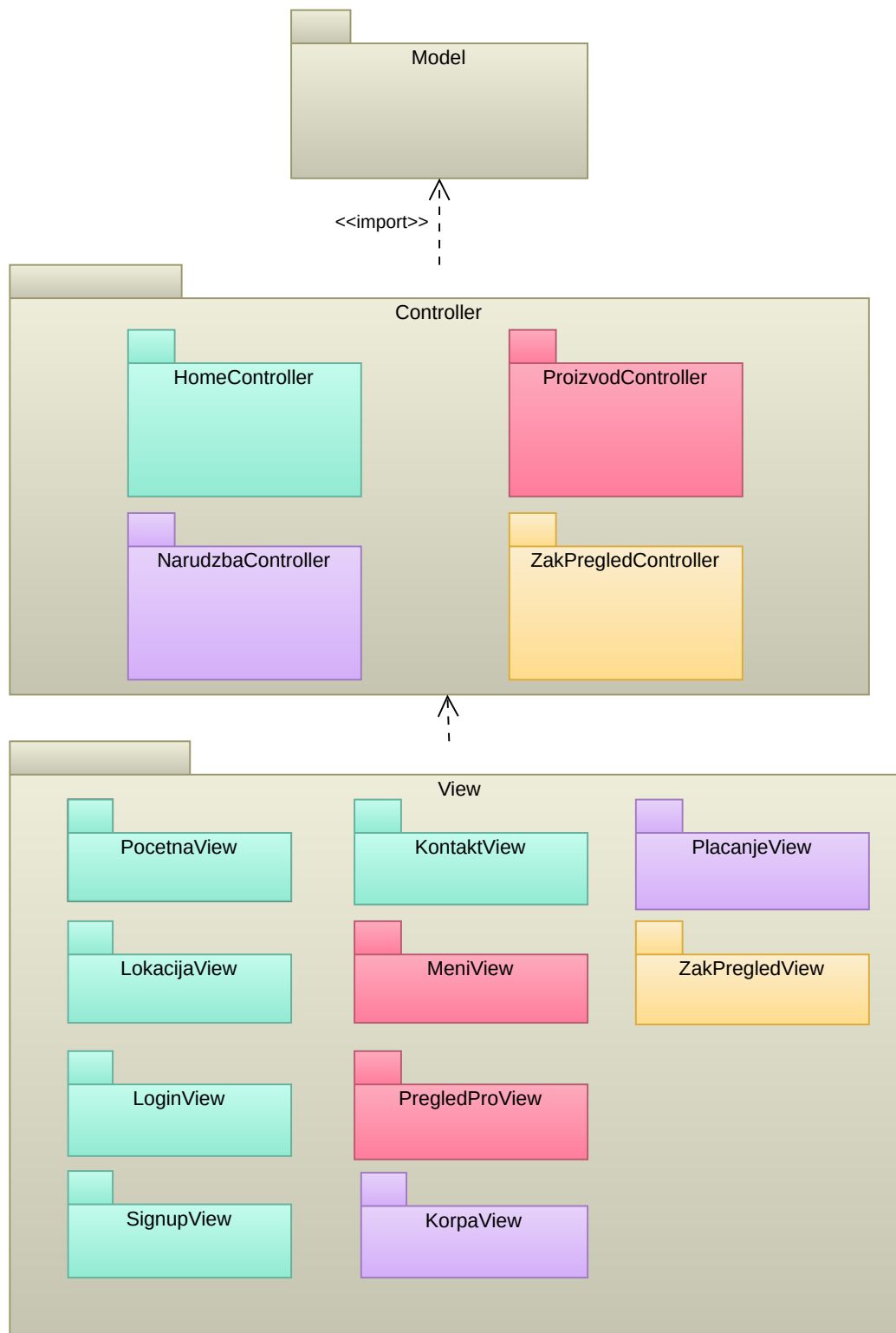
6. Proxy pattern

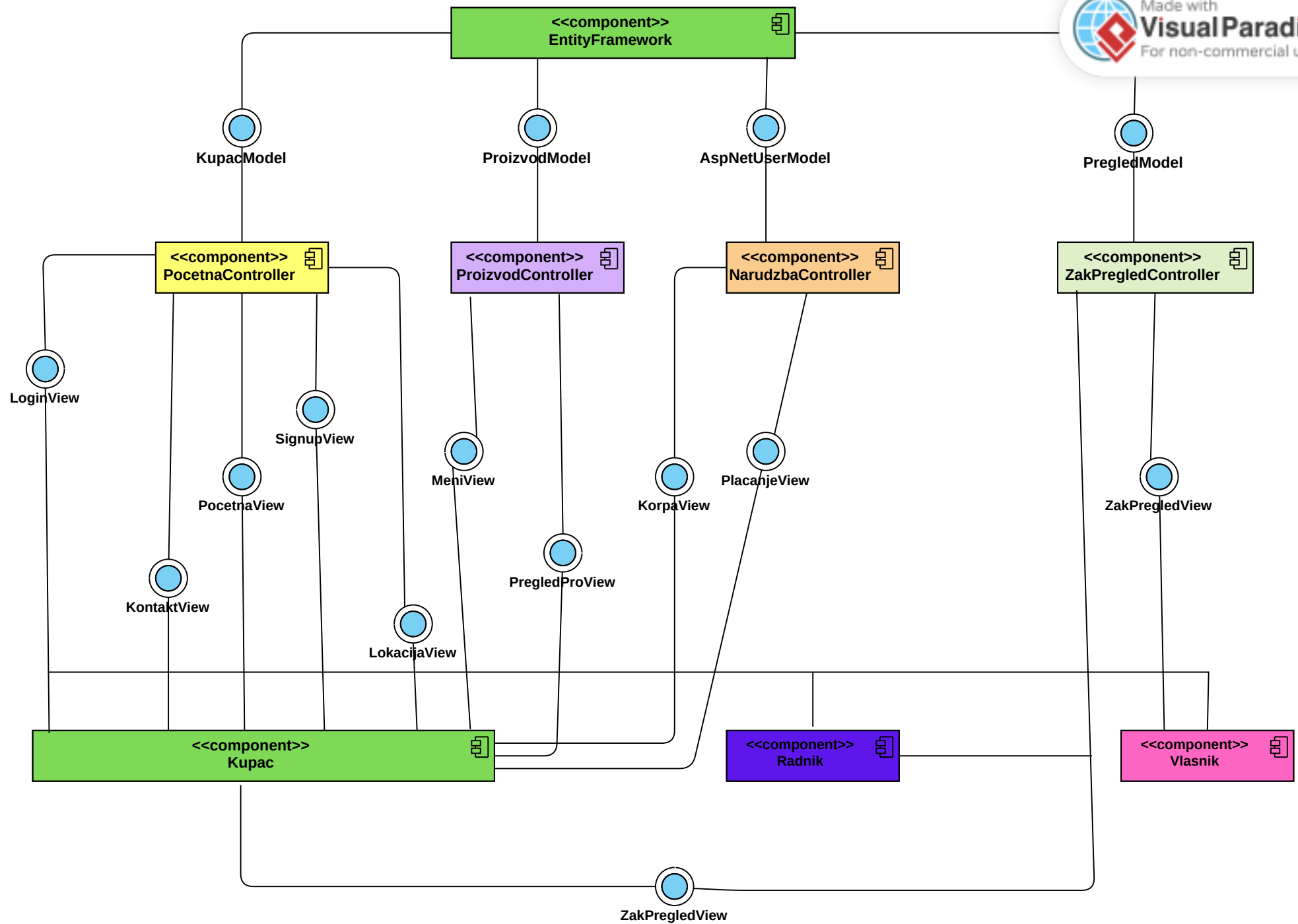
Proxy pattern - koristili bi proxy objekt za autentifikaciju i autorizaciju radnika za obavljanje upravljačkih zadataka poput upravljanja cijenama, pregled finansijskih izvještaja, prihvatanje ili odbijanje narudžbe te uvid u statistiku. Ovim ćemo postići da će proxy objekt omogućiti različitim korisnicima različit nivo pristupa i funkcionalnosti, štiteći sigurnost aplikacije. Proxy će posredovati između klijenta (korisnika) i stvarne implementacije objekta, pružajući dodatnu kontrolu pristupa i osiguravajući da se samo ovlašteni korisnici mogu baviti određenim operacijama.

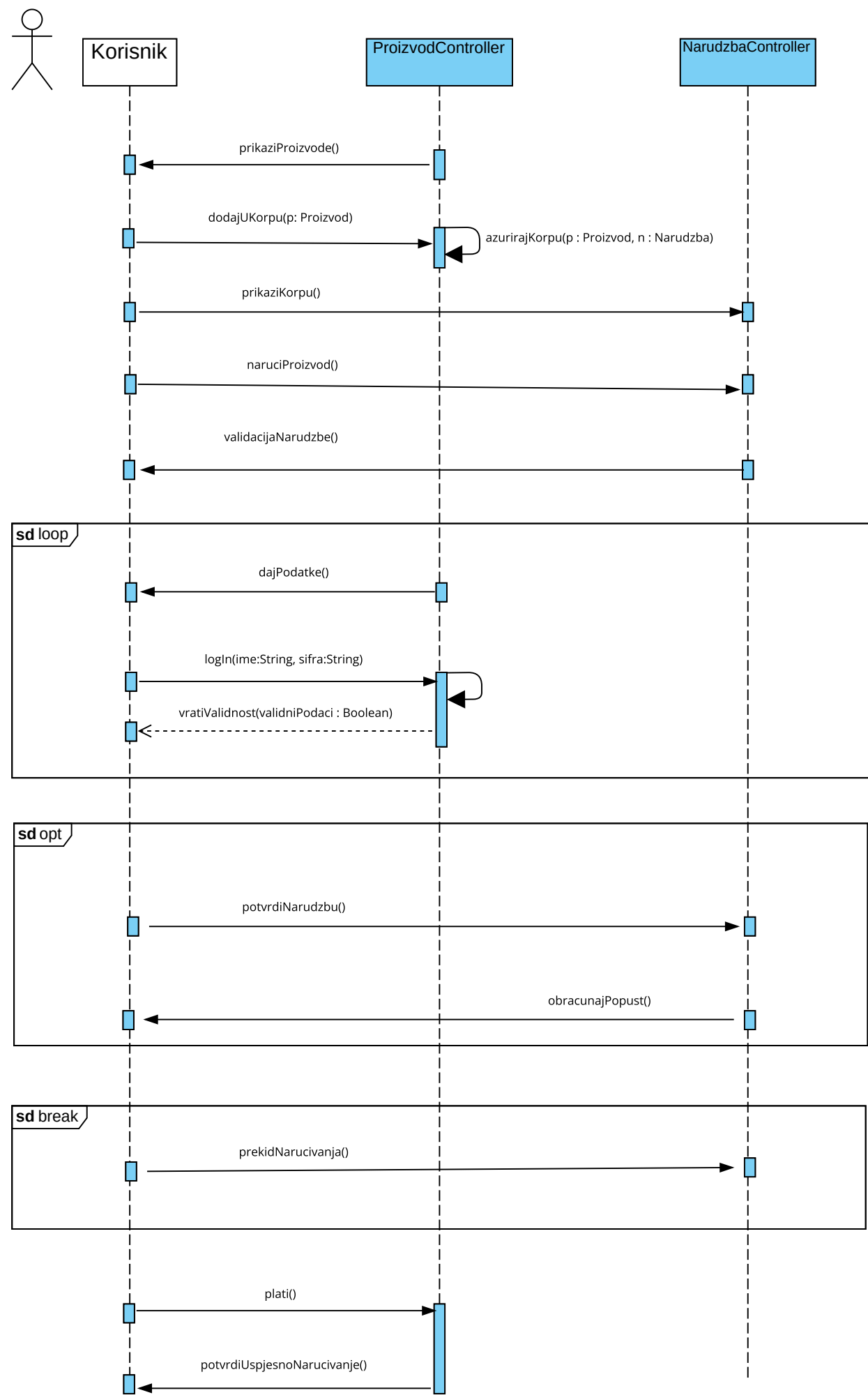
7. Flyweight pattern

Može nam pomoći optimizirati memoriju koja se koristi za pohranu informacija o naočalama. Dijeljenjem zajedničkih podataka možemo smanjiti potrebnu memoriju za pohranu istih informacija za svaku instancu.









SOLID principi

1. S – Single Responsibility Principle (SRP)

Napravili smo klase koje obavljaju samo jednu funkciju. Klasa Proizvod pokazuje sva svojstva proizvoda u ponudi. Klasa Korisnik ima samo informacije o imenu, prezimenu, šifri i emailu osobe.

2. O – Open – Closed Principle (OCP)

Klase Korisnik i Proizvod su direktno povezane. Ukoliko nakon nekog vremena bude potrebno da se doda novi produkt ili nešto slično morali bi mijenjati kod, tj. klasu Korisnik. Ovo možemo riješiti dodavanjem KorisnikInterface interfejsa.

3. L – Liskov Substitution Principle (LSP)

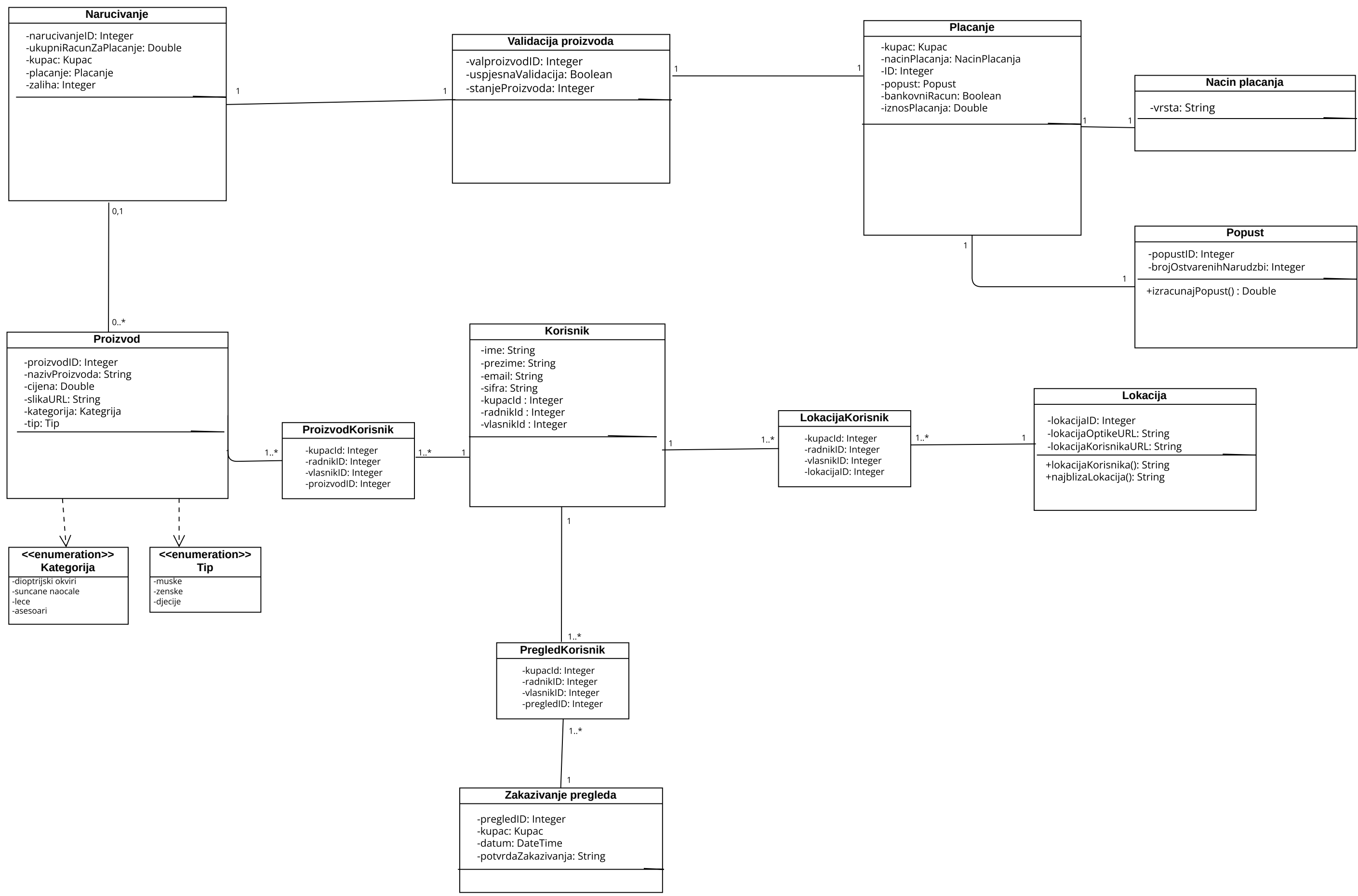
Kada bi imali klasu Osoba i iz te klase naslijedili tri klase Kupac, Radnik i Vlasnik, LSP bi bio prekršen.

4. I – Interface – Segregation Principle (ISP)

Napravili smo klase tako da one nemaju metode koje neće biti iskorištene i nismo imali potrebe za izdvajanjem interfejsa. Razlog je taj što su ovo samo podatkovne klase, a sve radnje će se odvijati u kontroleru.

5. D – Dependency – Inversion Principle (DIP)

U našem primjeru neće doći do kršenja toga da moduli visokog nivoa ovise o modulima niskog nivoa ili da moduli ovise o detaljima.



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: Korisnik

Funkcionalni zahtjevi u kojima klasa učestvuje:

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>
email	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
sifra	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
kupacID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
radnikID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
vlasnikID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Plaćanje

Funkcionalni zahtjevi u kojima klasa učestvuje:

4) Plaćanje

7) Popust za stalne korisnike

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
kupac	Kupac	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
nacinPlacanja	NacinPlacanja	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
ID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
popust	Popust	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
bankovniRacun	Boolean	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
iznosPlacanja	Double	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: NacinPlacanja

Funkcionalni zahtjevi u kojima klasa učestvuje:

4) Plaćanje

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
vrsta	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Popust

Funkcionalni zahtjevi u kojima klasa učestvuje:

2) Naručivanje

4) Plaćanje

6) Validacija

7) Ostvarivanje popusta

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
popustID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
brojOstvarenihNarudzbi	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Proizvod

Funkcionalni zahtjevi u kojima klasa učestvuje:

- 1) Pregled proizvoda
- 2) Naručivanje
- 4) Plaćanje
- 6) Validacija narudžbe

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
proizvodID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
nazivProizvoda	String	<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>
kategorijaProizvoda	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
tipProizvoda	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
slikaURL	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Narucivanje

Funkcionalni zahtjevi u kojima klasa učestvuje:

- 2) Naručivanje
- 4) Plaćanje

- 6) Validacija narudzbe
- 7) Popust za stalne korisnike

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
narucivanjeID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
ukupniRacunZaPlacanje	Double	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
kupac	Kupac	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
placanje	Placanje	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
zaliha	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Validacija proizvoda

Funkcionalni zahtjevi u kojima klasa učestvuje:

- 1) Pregled proizvoda
- 2) Naručivanje
- 4) Plaćanje
- 6) Validacija narudzbe

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
valproizvodID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
uspjesnaValidacija	Boolean	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
stanjeProizvoda	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

Naziv klase: Zakazivanje pregleda

Funkcionalni zahtjevi u kojima klasa učestvuje:

- 3) Log in-registracija
- 4) Plaćanje
- 5) Zakazivanje pregleda

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
pregledID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
kupac	Kupac	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
datum	DateTime	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
potvrdaZakazivanja	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>

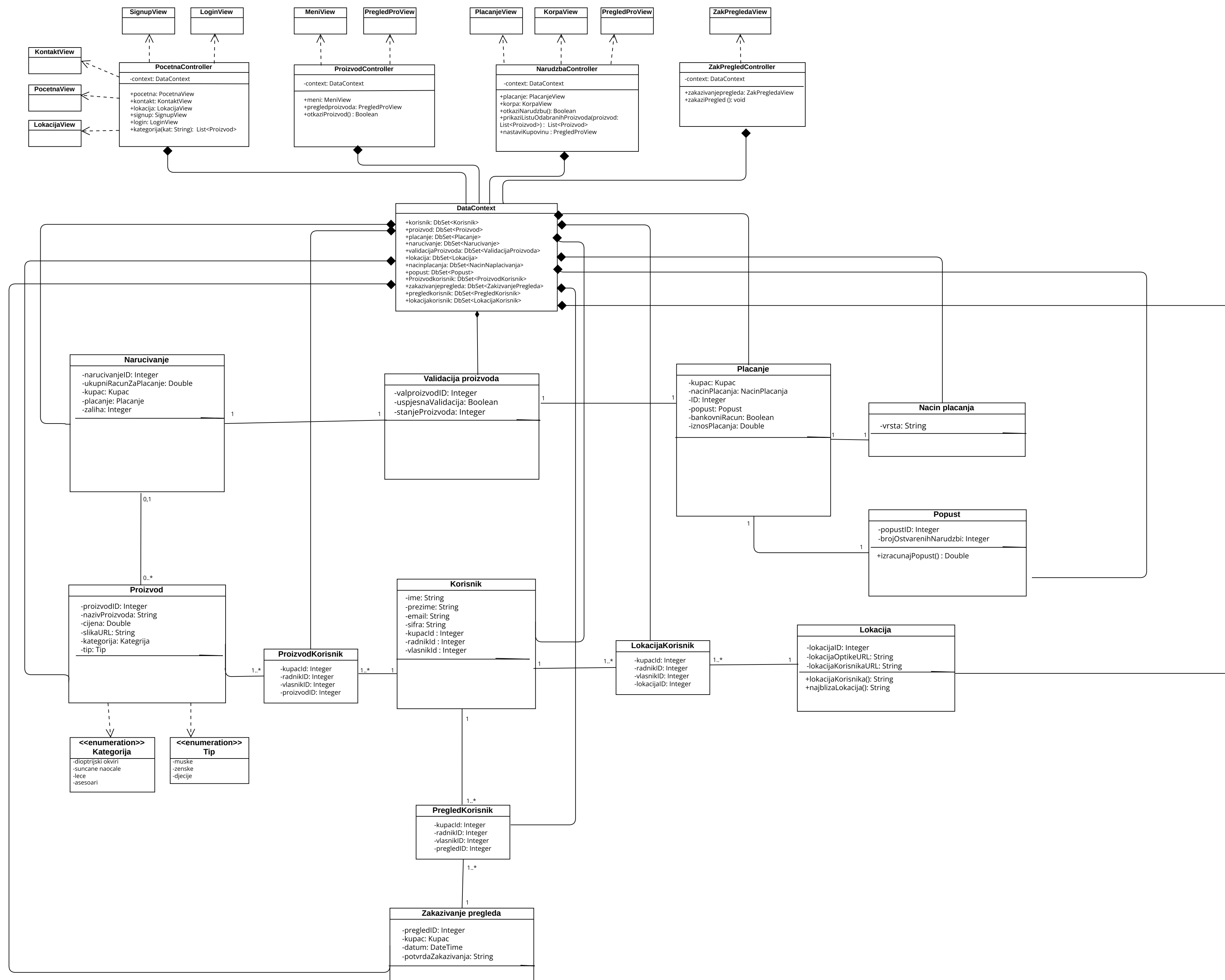
Naziv klase: Lokacija

Funkcionalni zahtjevi u kojima klasa učestvuje:

- 8) Lokacija najbliže poslovnice

Atributi koje klasa posjeduje:

Naziv atributa	Tip varijable	Dodatne napomene
lokacijaID	Integer	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
lokacijaOptikeURL	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>
lokacijaKorisnikaURL	String	<input type="checkbox"/> Atribut je statički <input type="checkbox"/> Atribut je <i>enumeration</i>



[POČETNA](#)[MENI](#)[SIGN UP](#)[LOG IN](#)

Optika Lentium

[Saznajte najbližu lokaciju](#)[Kontaktirajte nas](#)



POČETNA

MENI

SIGN UP

LOG IN

Dioptrijski okviri

Sunčane naočale

Leće i otopine

Asesoari

Zakažite pregled

Meni ponuda



Optika Lentium

Saznajte najbližu lokaciju

Kontaktirajte nas

[POČETNA](#)[MENI](#)[SIGN UP](#)[LOG IN](#)

Dioptrijski okviri

Kategorije:

- > Muški
- > Ženski
- > Dječiji
- > Clip - on

Cijena

KM 0.00

KM 2.000,00

PRIMJENI

Brend

- > Armani
- > Clark
- > Dolce&Gabbana
- > Esprit
- ...



Dolce&Gabbana
DG3241 501
570,00 KM



Dolce&Gabbana
DG3287 502
640,00 KM



Esprit
ET33484 557
215,00 KM



Armani Exchange
AX1052
155,00 KM



Esprit
ET33477 507
215,00 KM



Clark
C1292 C1
229,00 KM

[POČETNA](#)[MENI](#)[SIGN UP](#)[LOG IN](#)

Dolce&Gabbana DG3241 501

570,00 KM

Brend Dolce&Gabbana

Tip: Dioptrijski okviri

Boja: Crna Veličina: 54-19-140

1 na zalihi

NASTAVITE SA KUPOVINOM



[POČETNA](#)[MENI](#)[SIGN UP](#)[LOG IN](#)

VAŠA KORPA

PROIZVOD

CIJENA

KOLIČINA

UKUPNO:

UKUPNO:

570,00 KM



Dolce&Gabbana
DG3241 501

570,00 KM

- 1 +

570,00 KM X

NASTAVITE NA PLAĆANJE

NASTAVITE SA KUPOVINOM

[POČETNA](#)[MENI](#)[SIGN UP](#)[LOG IN](#)

PLAĆANJE KARTICOM

Kartica

Broj kartice

Cvv

Datum isteka kartice

mjesec



godina



POTVRDITE KUPOVINU



[POČETNA](#)[MENI](#)[SIGN UP](#)[LOG IN](#)

Zakazivanje pregleda

Podaci o pacijentu

Ime i prezime

Email

Broj telefona

Spol

Datum rođenja

Datum zakazivanja

Da li ste prije zakazivali termin u našim poslovnicama?

- ☐ Da
- ☐ Ne

POTVRDI



POČETNA

MENI

SIGN UP

LOG IN

Sign Up Here

Username

Email

Password

Sign Up

[Already have an account? Log In](#)

Login Here

Username

Password

Log In

[Don't have an account? Sign Up](#)

[POČETNA](#)[MENI](#)[SIGN UP](#)[LOG IN](#)

Unesite svoju lokaciju



Specifikacija projekta

1. Osnovne informacije o sistemu

Naziv teme: Optika Lentium

Logo:



Naziv tima: Opticari

Nastavna grupa: Grupa5-tim34

Link na repozitorij tima: <https://github.com/OOAD-2023-2024/Optika-Lentium.git>

Članovi tima:

1. Edna Džinić, 19185
2. Elvedina Halilović, 19394
3. Emina Šahbaz, 19076
4. Nađa Kovačević, 19182

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 optike je, prije svega, omogućiti online kupovinu naočala. Također sistem nudi zakazivanje pregleda. Osim toga sistem olakšava radnicima da lakše prate stanje robe, kao i kupcima da detaljnije pregledaju sve artikle i njihove pojedinosti.

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: Pregled proizvoda

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Funkcionalnost "Pregled proizvoda" omogućava korisnicima pretraživanje i detaljan prikaz svakog proizvoda, opciju filtriranje i dodavanje u košaricu.

2) Naziv funkcionalnosti: Naručivanje

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Proces naručivanja treba da se odvija putem interfejsa. Nakon dodavanja u korpu treba da se provjeri da li na stanju ima taj artikal, te da se se potvrdi narudžba. Zatim se vrši obračun narudže, te računa popust ako kupac ostvaruje pravo na to. Nakon plaćanja kupac dobija obavijest o izvršenoj narudžbi i e-račun narudžbe.

3) Naziv funkcionalnosti: Log in-registracija

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Log in sistem treba da omogući ulazak u sistem na različite načine, tj. u zavisnosti da li sistemu pristupa administrator, radnik ili kupac. Također, ovaj sistem treba da obavi validaciju podataka, te ispiše poruku ako su neispravni mail ili šifra.

4) Naziv funkcionalnosti: Plaćanje

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Korisniku je kroz ovu funkcionalnost omogućeno da vrši plaćanje odabranih proizvoda. Od korisnika će biti traženo da unese određene podatke koje zahtjeva sama aplikacija, kako bi se moglo izvršiti uspješno plaćanje.

5) **Naziv funkcionalnosti:** Zakazivanje pregleda

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

U sistemu će biti uneseno vrijeme kada dolazi doktor u optiku. Taj period će biti podijeljen na manje vremenske dijelove. Kupac će biti u mogućnosti da rezerviše svoj odgovarajući termin klikom na odgovarajuće vrijeme. Nakon rezervacije termina kupac dobija obavijest o uspješnoj rezervaciji.

6) **Naziv funkcionalnosti:** Validacija narudžbe

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Kada korisnik dovrši odabir proizvoda, sistem provjerava dostupnost tih proizvoda. U slučaju nedostatka proizvoda, korisniku se pruža obavijest o nedostatku i mogućnost izvršavanja nove narudžbe ili mogućnost napuštanja aplikacije. Nakon uspješne potvrde narudžbe, korisnik može završiti narudžbu plaćanjem.

7) **Naziv funkcionalnosti:** Popust za stalne korisnike

Vrsta funkcionalnosti: Operacija sa specifičnim algoritmom obrade

Opis funkcionalnosti:

Sistem prati koliko je puta isti korisnik naručio nešto sa stranice. Kada isti korisnik naručuje svaki treći put ostvaruje popust od 20%.

8) **Naziv funkcionalnosti:** Lokacija najbliže poslovnice

Vrsta funkcionalnosti: Korištenje vanjskog uređaja

Opis funkcionalnosti:

Sistem je povezan sa google maps. Prilikom pretrage optike, osim prikaza svih poslovnica, sistem pokazuje koja je najbliža vašoj lokaciji.

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

Vrsta aktera: Korisnik sistema

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
1.Pregled proizvoda	Mogućnost pregleda
3.Log in	Mogućnost uređivanja
4.Plaćanje	Mogućnost uređivanja
5.Zakazivanje pregleda	Mogućnost uređivanja
8.Lokacija najbliže poslovnice	Mogućnost pregleda
2.Naručivanje	Mogućnost uređivanja

b) Naziv aktera: Radnik

Vrsta aktera: Zaposlenik sistema

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
2.Naručivanje	Mogućnost pregleda

3. Log in	Mogućnost uređivanja
5. Zakazivanje pregleda	Mogućnost uređivanja
6. Validacija narudžbe	Mogućnost uređivanja

c) **Naziv aktera:** Vlasnik optike

Vrsta aktera: Administrator

Funkcionalnosti u kojima akter učestvuje:

Način učešća:

- *Mogućnost pregleda*
- *Mogućnost uređivanja*

Funkcionalnost sistema	Način učešća
3. Log in	Mogućnost uređivanja
1. Pregled proizvoda	Mogućnost uređivanja
2. Naručivanje	Mogućnost uređivanja
4. Plaćanje	Mogućnost uređivanja
5. Zakazivanje pregleda	Mogućnost uređivanja
6. Validacija narudžbe	Mogućnost uređivanja
7. Popust za stalne korisnike	Mogućnost uređivanja
8. Lokacija najbliže poslovnice	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:** Ograničavanje prilikom plaćanja

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

Ukoliko korisnik izvrši plaćanje, potvrda o plaćanju će se prikazati nakon pet sekundi.

2) **Naziv nefunkcionalnog zahtjeva:** Dostupnost proizvoda

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

Ukoliko odabranog proizvoda nema na stanju, prikazat će se određena poruka nakon tri sekunde.

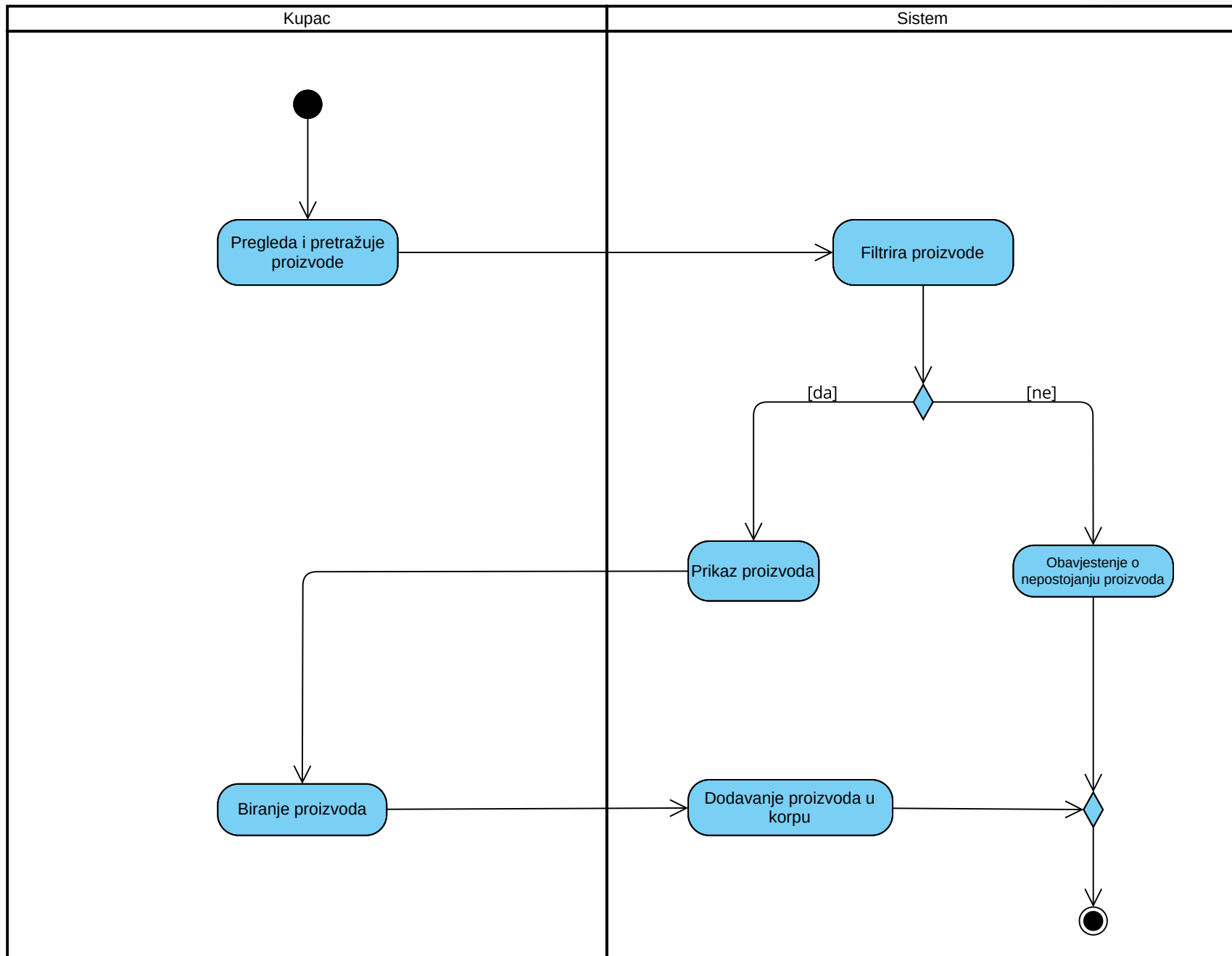
3) **Naziv nefunkcionalnog zahtjeva:** Brzina učitavanja podataka

Opis:

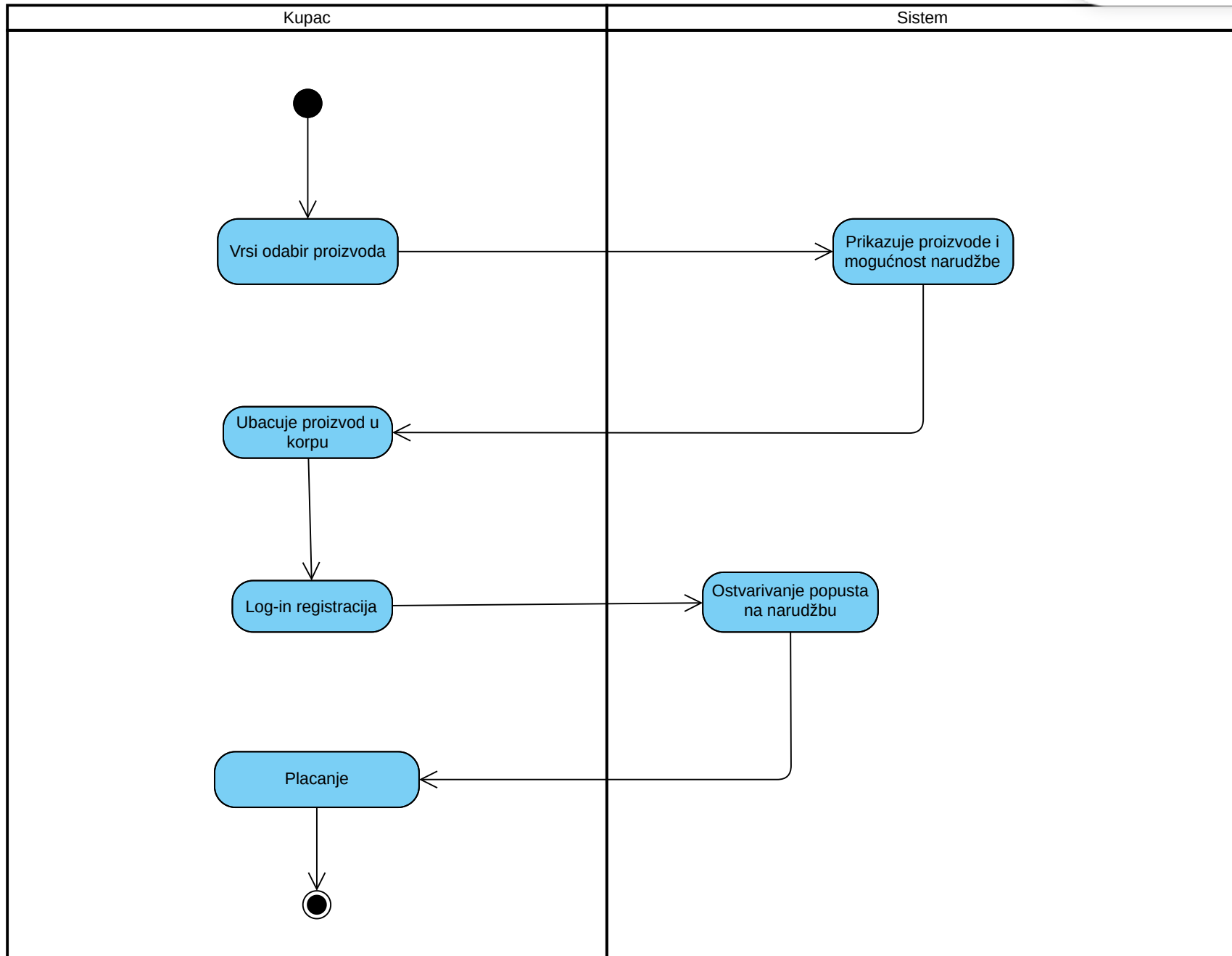
Opisati ograničenje sistema i način na koje se ono ispoljava.

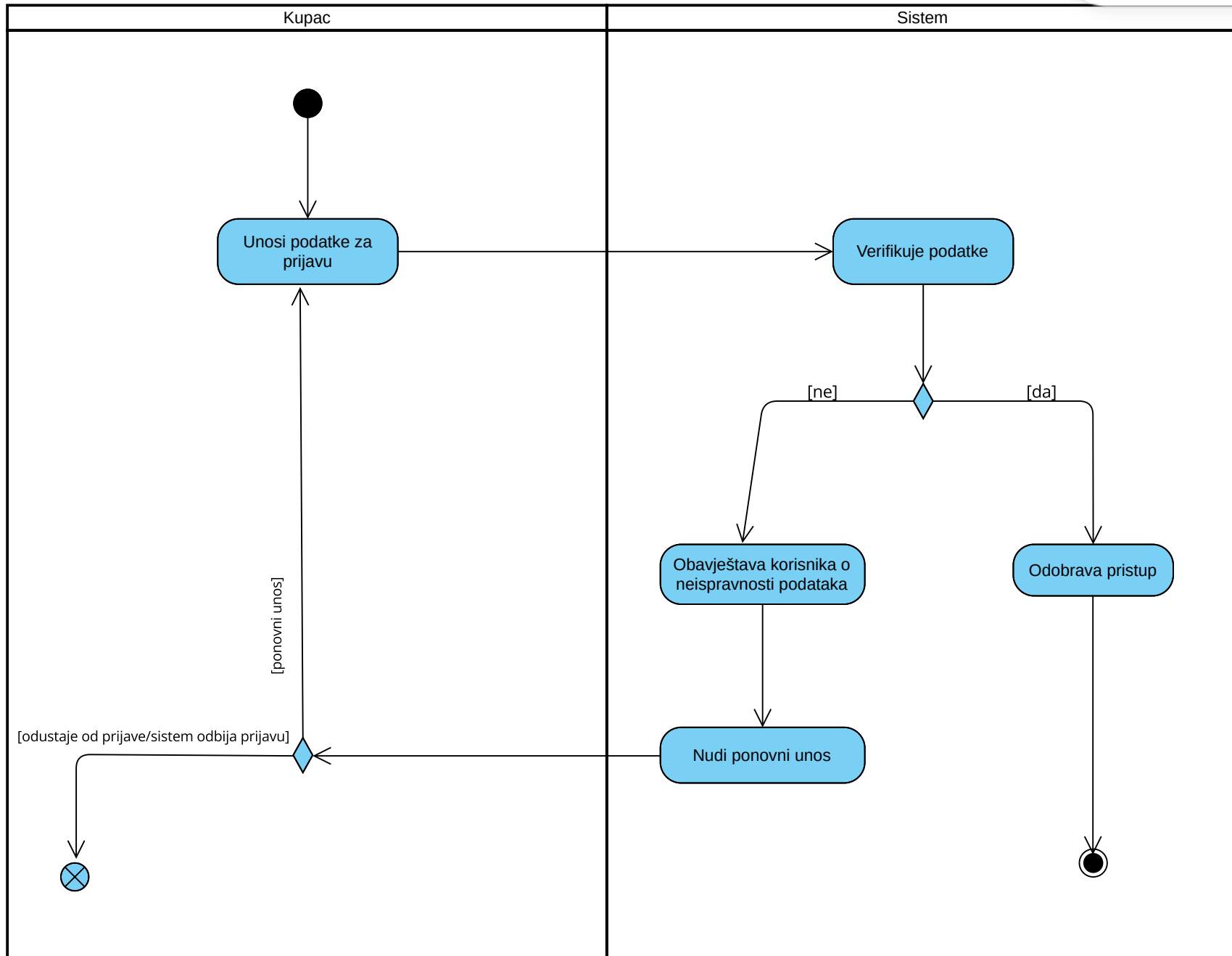
Aplikacija osigurava brzo učitavanje podataka.

Dijagram aktivnosti za pregled proizvoda

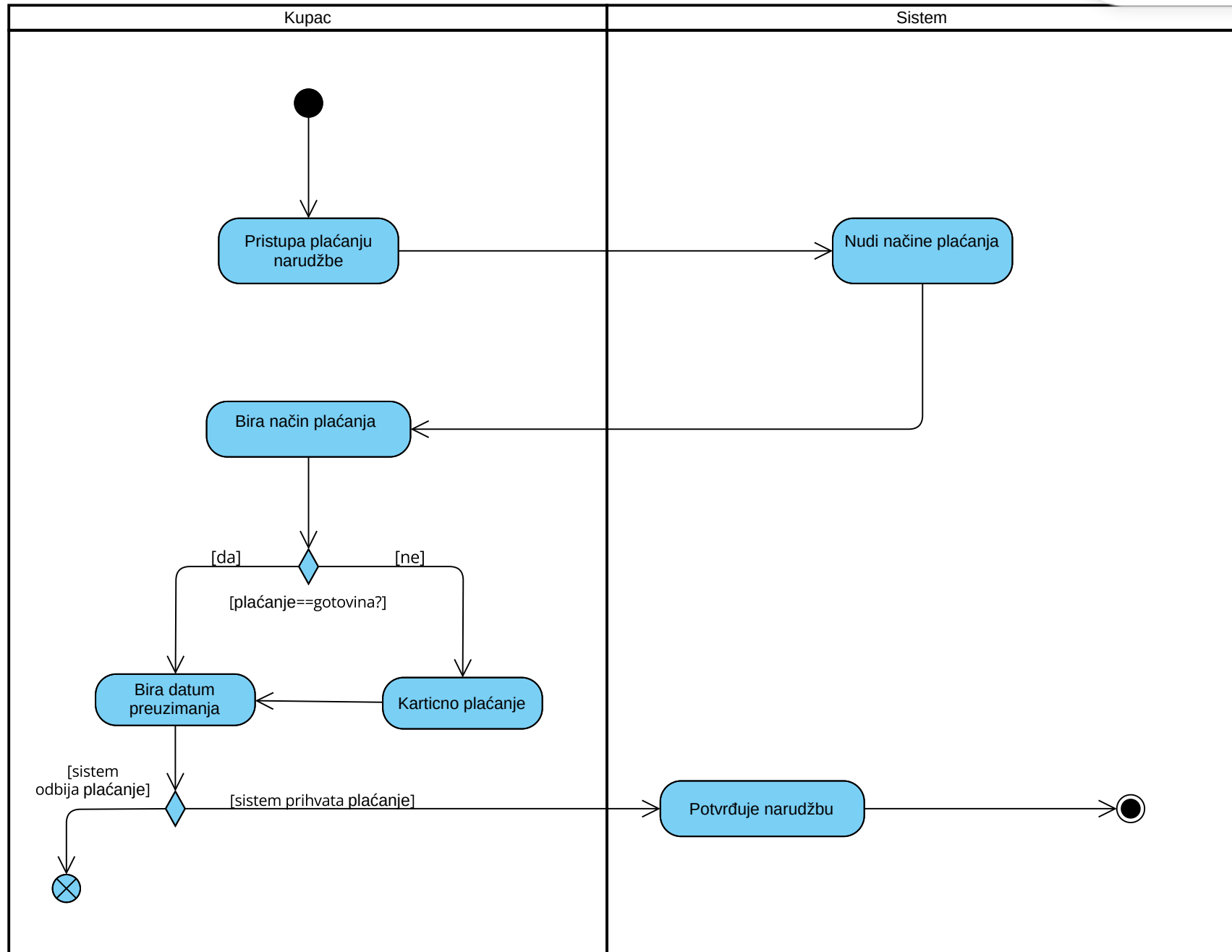


Dijagram aktivnosti za naručivanje

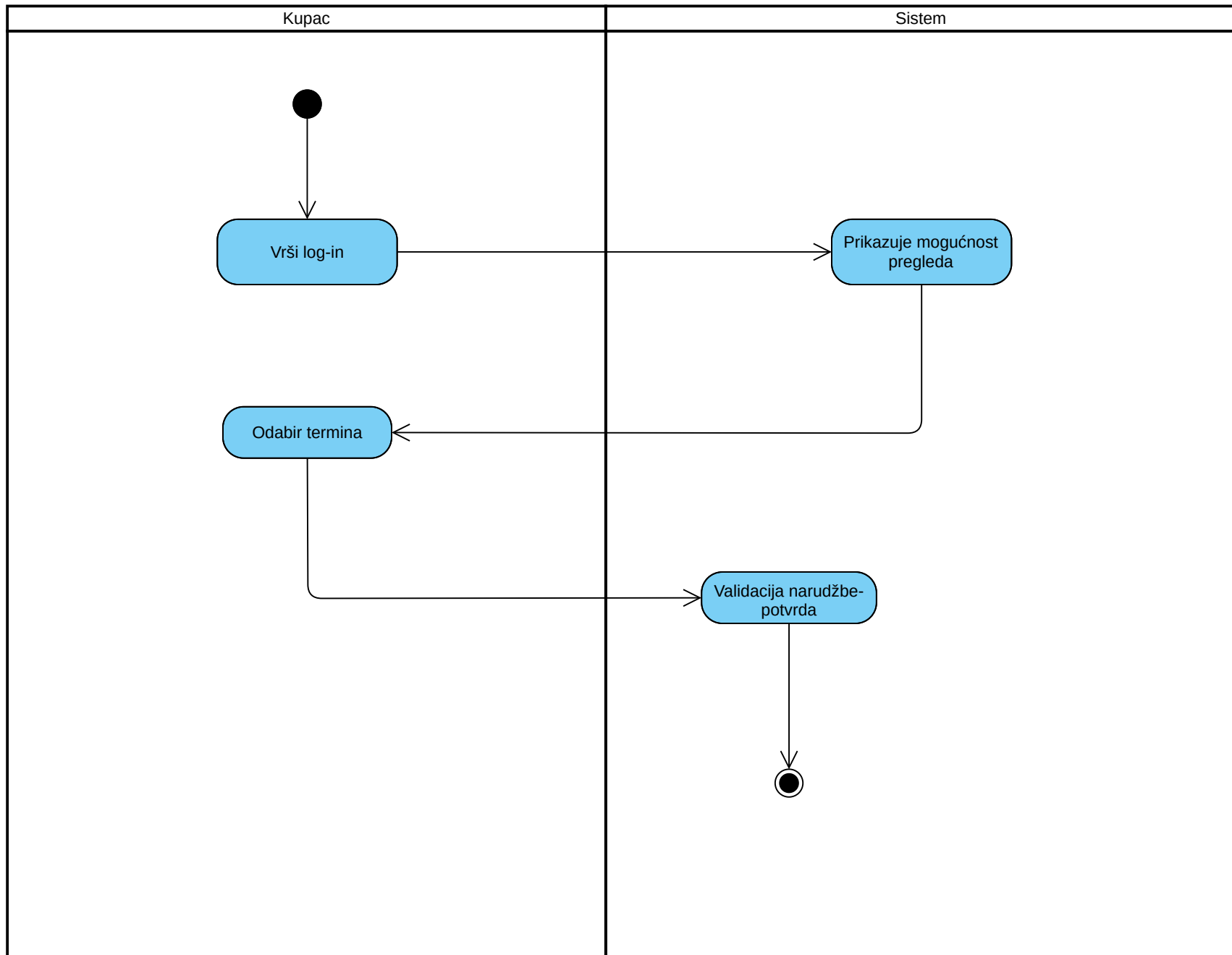




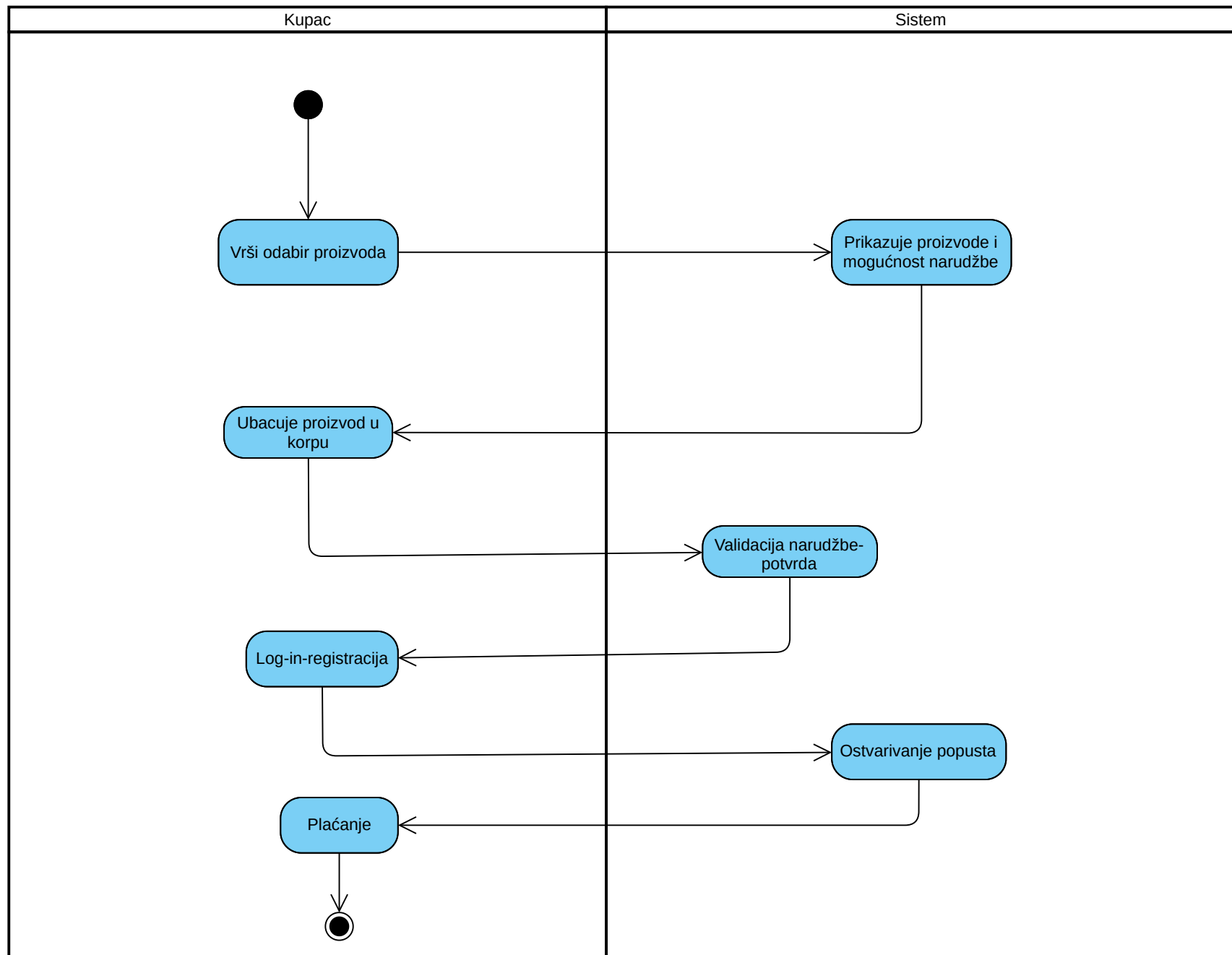
Dijagram aktivnosti za plaćanje



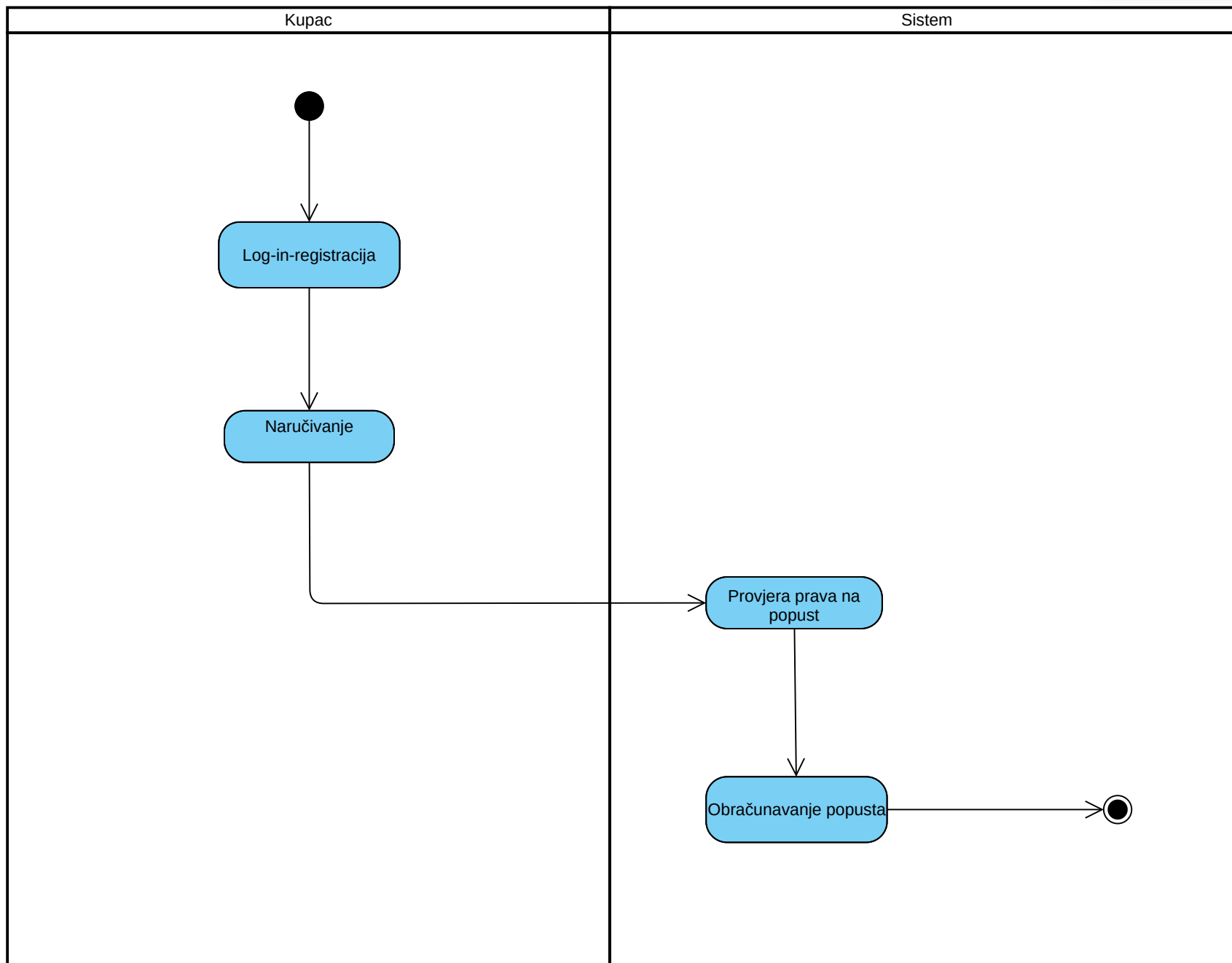
Dijagram aktivnosti za zakazivanje pregleda



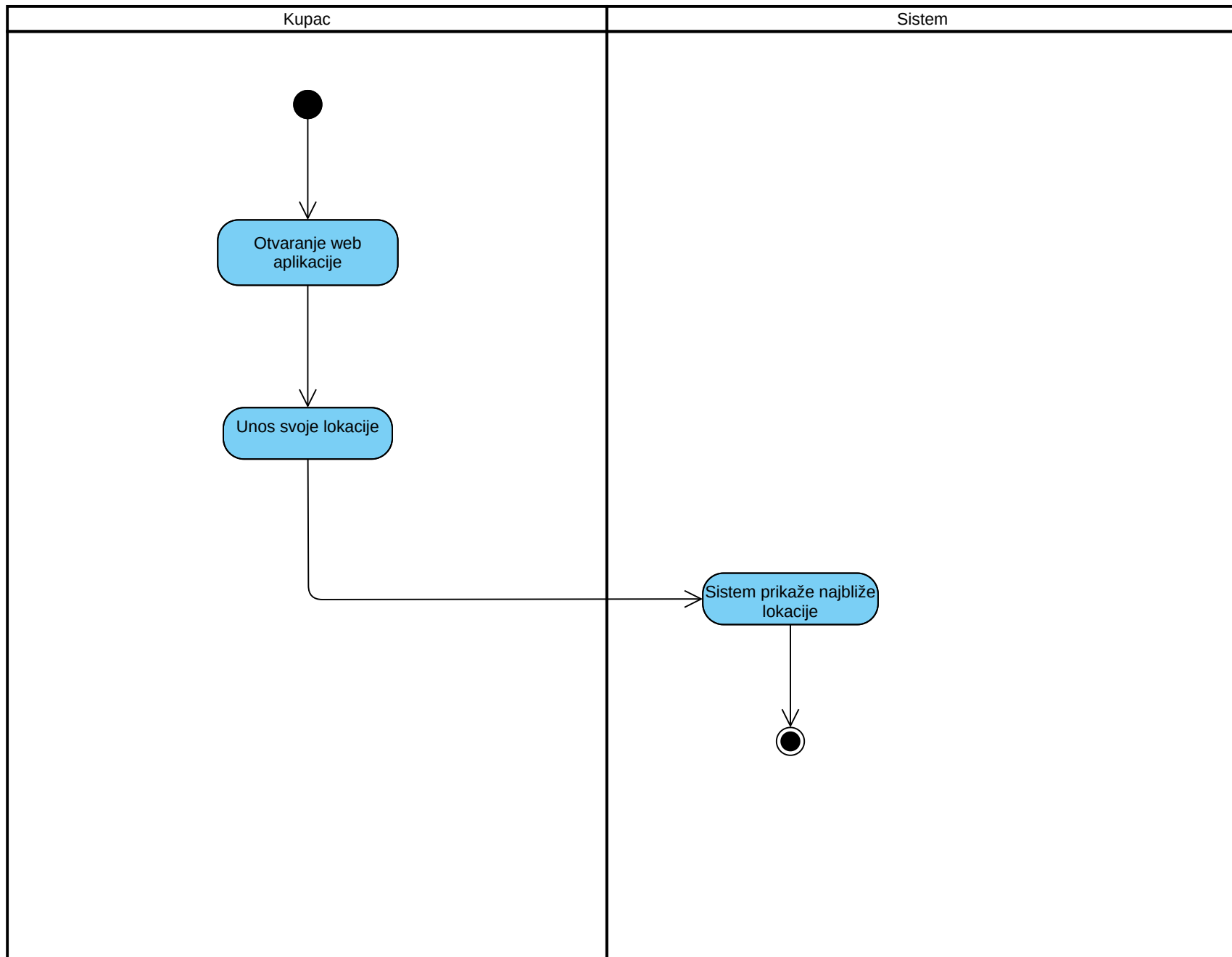
Dijagram aktivnosti za validaciju narudžbe

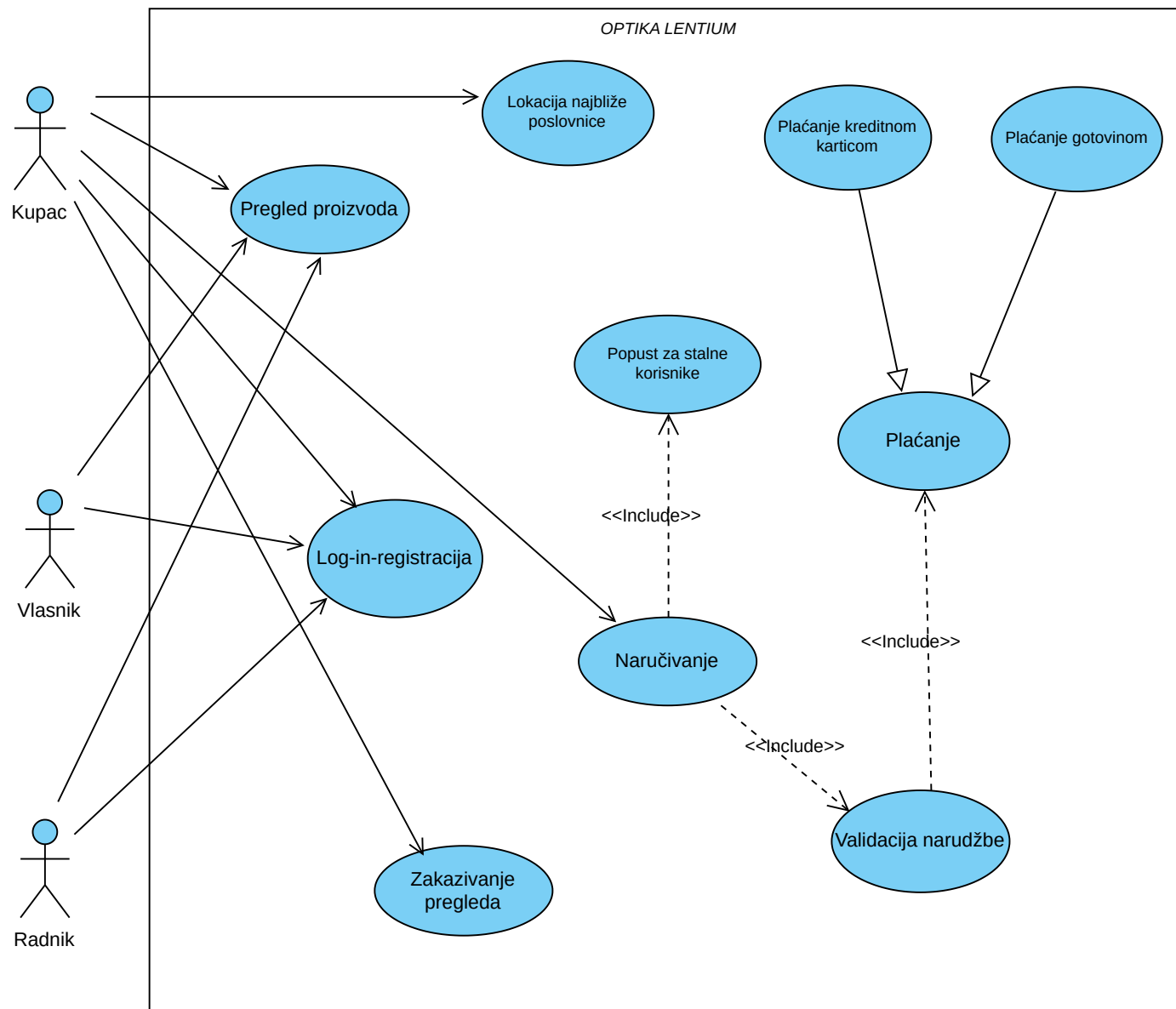


Dijagram aktivnosti za popust za stalne korisnike



Dijagram aktivnosti za lokacija najbliže poslovnice





Dijagram slučajeva upotrebe

1. Pregled proizvoda

Naziv slučaja upotrebe	Pregled proizvoda
Opis slučaja upotrebe	Omogućava korisnicima pretraživanje i detaljan prikaz svakog proizvoda, opciju filtriranje i dodavanje u košaricu.
Vezani zahtjevi	-
Preduslovi	-
Posljedice-uspjšan završetak	Uspješan pregled proizvoda
Posljedice-neuspješan završetak	Proizvod ne postoji
Primarni akteri	Kupac
Ostali akteri	Vlasnik, radnik
Glavni tok	Nakon pristupa stranici (bez potrebe za prijavljivanje), korisnik dobija uvid u sve proizvode i mogućnost da pretražuje proizvode
Alternative/proširenja	-

1.1 Tok događaja: Upješan završetak - Proizvod postoji

<i>Kupac</i>	<i>Sistem</i>
1. Pregleda i pretražuje proizvode po željenim specifikacijama	
	2.Filtrira proizvode
	3. Prikazuje proizvode koji odgovaraju unesenim zahtjevima
4. Bira proizvod čiji detaljni opis želi vidjeti	
	5. Prikazuje detalje i opciju "Add to cart"

1.2 Tok događaja: Neuspješan završetak - Proizvod ne postoji

<i>Kupac</i>	<i>Sistem</i>
1. Pregleda i pretražuje proizvode po željenim specifikacijama	
	2.Filtrira proizvode
	3. Obavještava korisnika o nepostojanju proizvoda

2. Naručivanje

Naziv slučaja upotrebe	<i>Naručivanje</i>
Opis slučaja upotrebe	Kupac popunjava zahtjev za narudžbu proizvoda.
Vezani zahtjevi	Plaćanje
Preduslovi	1. Log-in-registracija
Posljedice – uspješan završetak	Uspješno naručivanje proizvoda
Posljedice – neuspješan završetak	-
Primarni akteri	Kupac
Ostali akteri	Radnik, vlasnik
Glavni tok	Kupac iz opisa proizvoda bira veličinu i ostale ponuđene karakteristike po želji kao i mogućnost napomena, način plaćanja.
Alternative/proširenja	-

2.1. Tok događaja: Uspješan završetak - izvršena narudžba proizvoda

<i>Kupac</i>	<i>Sistem</i>
1.Vrši odabir proizvoda	
	2. Prikazuje detalje proizvoda i mogućnost narudžbe proizvoda
3.Ubacuje proizvod u korpu	
	4. Ostvarivanje popusta na narudžbu
5.Log-in-registracija	
6. Plaćanje	

3. Log in-registracija

Naziv slučaja upotrebe	<i>Log in-registracija</i>
Opis slučaja upotrebe	Korisnik može izvršiti prijavu te dobiti mogućnost narudžbe proizvoda I zakazivanje pregleda.
Vezani zahtjevi	-
Preduslovi	-
Posljedice – uspješan završetak	Uspješno prijavljivanje
Posljedice – neuspješan završetak	Neuspješno prijavljivanje
Primarni akteri	Kupac
Ostali akteri	Radnik, vlasnik
Glavni tok	Ukoliko kupac želi naručiti proizvod iz naše ponude, korisnik unosi svoje korisničke podatke i prijavljuje se. Ukoliko radnik treba da izvrši pregled zakaznih termina, prijavljuje se na sistem.
Alternative/proširenja	-

3.1. Tok događaja: Uspješan završetak - izvršena prijava

<i>Kupac</i>	<i>Sistem</i>
1. Unosi potrebne podatke za prijavu	
	2. Verifikuje podatake
	3. Odobrava pristup

3.2 Tok događaja: Neuspješan završetak - pogrešni podaci

<i>Kupac</i>	<i>Sistem</i>
1. Unosi potrebne podatke za prijavu	
	2. Verifikuje podatke
	3. Obavještava korisnika o neispravnosti podataka
	4. Nudi ponovni unos
5. Ponovo unosi podatke ili odustaje od prijave	
	6. Odbija pokušaj prijave

4. Plaćanje

Naziv slučaja upotrebe	<i>Plaćanje</i>
Opis slučaja upotrebe	Kupac bira vrstu plaćanja: pouzećem ili kartično plaćanje
Vezani zahtjevi	Validacija narudžbe
Preduslovi	Log-in-registracija
Posljedice – uspješan završetak	Uspješno završena narudžba proizvoda plaćanjem
Posljedice – neuspješan završetak	Neuspješno završeno plaćanje ili neodobrena narudžba
Primarni akteri	Kupac
Ostali akteri	Zaposlenik
Glavni tok	Nakon uspješnog dodavanja proizvoda u korpu i potvrde narudžbe, kupac može odabrati mogućnost plaćanja karticom ili pouzećem
Alternative/proširenja	Ostvarivanje popusta na narudžbu

4.1. Tok događaja: Uspješan završetak

<i>Kupac</i>	<i>Sistem</i>
1. Pristupa plaćanju narudžbe	
	2. Nudi načine plaćanja
3. Bira način plaćanja i datum preuzimanja	
	4. Potvrđuje narudžbu

4.2. Tok događaja: Neuspješan završetak

<i>Kupac</i>	<i>Sistem</i>
1. Pristupa plaćanju narudžbe	
	2. Nudi načine plaćanja
3. Bira način plaćanja i datum preuzimanja	
	4. Odbija plaćanje

5. Zakazivanje pregleda

Naziv slučaja upotrebe	<i>Zakazivanje pregleda</i>
Opis slučaja upotrebe	Kupac će biti u mogućnosti da rezerviše svoj odgovarajući termin klikom na odgovarajuće vrijeme.
Vezani zahtjevi	-
Preduslovi	Log-in-registracija
Posljedice – uspješan završetak	Uspješno zakazivanje pregleda
Posljedice – neuspješan završetak	-
Primarni akteri	Kupac
Ostali akteri	Radnik, vlasnik
Glavni tok	Nakon log-in-a na stranicu, kupac dobija mogućnost zakazivanja pregleda.
Alternative/proširenja	-

5.1. Tok događaja: Uspješan završetak – izvršeno zakazivanje pregleda

<i>Kupac</i>	<i>Sistem</i>
1.Vrši log-in	
	2. Prikazuje detalje proizvoda i mogućnost narudžbe proizvoda i zakazivanja pregleda
3.Odabir termina	
	4. Potvrda termina

6. Validacija narudžbe

Naziv slučaja upotrebe	<i>Validacija narudžbe</i>
Opis slučaja upotrebe	Sistem provjerava da li odabranih proizvoda ima na stanju.
Vezani zahtjevi	Plaćanje
Preduslovi	Log-in-registracija Naručivanje
Posljedice – uspješan završetak	Potvrda narudžbe
Posljedice – neuspješan završetak	Odbijanje narudžbe
Primarni akteri	Kupac
Ostali akteri	Radnik, vlasnik
Glavni tok	Ukoliko na stanju nema odabranih proizvoda, kupac dobija odgovarajuće obavještenje te mogućnost odabira novih proizvoda ili odustanka od narudžbe.
Alternative/proširenja	-

6.1. Tok događaja: Uspješan završetak – potvrda narudžbe

<i>Kupac</i>	<i>Sistem</i>
1. Vršiti odabir proizvoda	
	2. Prikazuje detalje proizvoda i mogućnost narudžbe proizvoda
3. Ubacuje proizvod u korpu	
	4. Validacija narudžbe- potvrda
5. Log-in-registracija	
	6. Ostvarivanje popusta
7. Plaćanje	

6.2. Tok događaja: Neuspješan završetak – odbijanje narudžbe

<i>Kupac</i>	Sistem
1. Vršiti odabir proizvoda	
	2. Prikazuje detalje proizvoda i mogućnost narudžbe proizvoda
3. Ubacuje proizvod u korpu	
	4. Validacija narudžbe-odbijanje
	5. Ispisane poruke i vraćanje na pregled proizvoda

7. Popust za stalne korisnike

Naziv slučaja upotrebe	<i>Popust za stalne korisnike</i>
Opis slučaja upotrebe	Kupac može ostvariti popust na cjelokupnu narudžbu
Vezani zahtjevi	-
Preduslovi	Log-in-registracija Naručivanje
Posljedice – uspješan završetak	Uspješno ostvaren popust na narudžbu
Posljedice – neuspješan završetak	-
Primarni akteri	Kupac
Ostali akteri	-
Glavni tok	Nakon naručivanja proizvoda, sistem provjerava da li je izvršena treća narudžba po redu, i ako jeste, obračunava popust.
Alternative/proširenja	-

7.1. Tok događaja: Uspješan završetak - ostvaren popust

<i>Kupac</i>	<i>Sistem</i>
1. Log-in-registracija	
2. Naručivanje	
	3. <i>Provjera prava na popust</i>
	4. <i>Obračunavanje popusta</i>

8. Lokacija najbliže poslovnice

Naziv slučaja upotrebe	<i>Lokacija najbliže poslovnice</i>
Opis slučaja upotrebe	Sistem je povezan sa google maps. Prilikom pretrage optike, osim prikaza svih poslovnica, sistem pokazuje koja je najbliža vašoj lokaciji.
Vezani zahtjevi	-
Preduslovi	-
Posljedice – uspješan završetak	Uspješan pregled poslovnica
Posljedice – neuspješan završetak	-
Primarni akteri	Kupac
Ostali akteri	Vlasnik
Glavni tok	Nakon otvaranja stranice, kupac ima mogućnost da pogleda najbližu lokaciju poslovnica.
Alternative/proširenja	-

8.1. Tok događaja: Uspješan završetak – uspješan pregled poslovnica

<i>Kupac</i>	<i>Sistem</i>
1. Otvaranje web aplikacije	
2. Unos svoje lokacije	
	3. Sistem prikaže najbliže lokacije

Specifikacija projekta

1. Osnovne informacije o sistemu

Naziv teme: Optika Lentium

Logo:



Naziv tima: Opticari

Nastavna grupa: Grupa5-tim34

Link na repozitorij tima: <https://github.com/OOAD-2023-2024/Optika-Lentium.git>

Članovi tima:

1. Edna Džinić, 19185
2. Elvedina Halilović, 19394
3. Emina Šahbaz, 19076
4. Nađa Kovačević, 19182

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 optike je, prije svega, omogućiti online kupovinu naočala. Također sistem nudi zakazivanje pregleda. Osim toga sistem olakšava radnicima da lakše prate stanje robe, kao i kupcima da detaljnije pregledaju sve artikle i njihove pojedinosti.

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: Pregled proizvoda

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Funkcionalnost "Pregled proizvoda" omogućava korisnicima pretraživanje i detaljan prikaz svakog proizvoda, opciju filtriranje i dodavanje u košaricu.

2) Naziv funkcionalnosti: Naručivanje

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Proces naručivanja treba da se odvija putem interfejsa. Nakon dodavanja u korpu treba da se provjeri da li na stanju ima taj artikal, te da se se potvrdi narudžba. Zatim se vrši obračun narudže, te računa popust ako kupac ostvaruje pravo na to. Nakon plaćanja kupac dobija obavijest o izvršenoj narudžbi i e-račun narudžbe.

3) Naziv funkcionalnosti: Log in-registracija

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Log in sistem treba da omogući ulazak u sistem na različite načine, tj. u zavisnosti da li sistemu pristupa administrator, radnik ili kupac. Također, ovaj sistem treba da obavi validaciju podataka, te ispiše poruku ako su neispravni mail ili šifra.

4) Naziv funkcionalnosti: Plaćanje

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Korisniku je kroz ovu funkcionalnost omogućeno da vrši plaćanje odabranih proizvoda. Od korisnika će biti traženo da unese određene podatke koje zahtjeva sama aplikacija, kako bi se moglo izvršiti uspješno plaćanje.

5) **Naziv funkcionalnosti:** Zakazivanje pregleda

Vrsta funkcionalnosti: Usluga sistema

Opis funkcionalnosti:

U sistemu će biti uneseno vrijeme kada dolazi doktor u optiku. Taj period će biti podijeljen na manje vremenske dijelove. Kupac će biti u mogućnosti da rezerviše svoj odgovarajući termin klikom na odgovarajuće vrijeme. Nakon rezervacije termina kupac dobija obavijest o uspješnoj rezervaciji.

6) **Naziv funkcionalnosti:** Validacija narudžbe

Vrsta funkcionalnosti: Perzistencija podataka (CRUD operacija)

Opis funkcionalnosti:

Kada korisnik dovrši odabir proizvoda, sistem provjerava dostupnost tih proizvoda. U slučaju nedostatka proizvoda, korisniku se pruža obavijest o nedostatku i mogućnost izvršavanja nove narudžbe ili mogućnost napuštanja aplikacije. Nakon uspješne potvrde narudžbe, korisnik može završiti narudžbu plaćanjem.

7) **Naziv funkcionalnosti:** Popust za stalne korisnike

Vrsta funkcionalnosti: Operacija sa specifičnim algoritmom obrade

Opis funkcionalnosti:

Sistem prati koliko je puta isti korisnik naručio nešto sa stranice. Kada isti korisnik naručuje svaki treći put ostvaruje popust od 20%.

8) **Naziv funkcionalnosti:** Lokacija najbliže poslovnice

Vrsta funkcionalnosti: Korištenje vanjskog uređaja

Opis funkcionalnosti:

Sistem je povezan sa google maps. Prilikom pretrage optike, osim prikaza svih poslovnica, sistem pokazuje koja je najbliža vašoj lokaciji.

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

Vrsta aktera: Korisnik sistema

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
1.Pregled proizvoda	Mogućnost pregleda
3.Log in	Mogućnost uređivanja
4.Plaćanje	Mogućnost uređivanja
5.Zakazivanje pregleda	Mogućnost uređivanja
8.Lokacija najbliže poslovnice	Mogućnost pregleda
2.Naručivanje	Mogućnost uređivanja

b) Naziv aktera: Radnik

Vrsta aktera: Zaposlenik sistema

Funkcionalnosti u kojima akter učestvuje:

Funkcionalnost sistema	Način učešća
2.Naručivanje	Mogućnost pregleda

3. Log in	Mogućnost uređivanja
5. Zakazivanje pregleda	Mogućnost uređivanja
6. Validacija narudžbe	Mogućnost uređivanja

c) **Naziv aktera:** Vlasnik optike

Vrsta aktera: Administrator

Funkcionalnosti u kojima akter učestvuje:

Način učešća:

- *Mogućnost pregleda*
- *Mogućnost uređivanja*

Funkcionalnost sistema	Način učešća
3. Log in	Mogućnost uređivanja
1. Pregled proizvoda	Mogućnost uređivanja
2. Naručivanje	Mogućnost uređivanja
4. Plaćanje	Mogućnost uređivanja
5. Zakazivanje pregleda	Mogućnost uređivanja
6. Validacija narudžbe	Mogućnost uređivanja
7. Popust za stalne korisnike	Mogućnost uređivanja
8. Lokacija najbliže poslovnice	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:** Ograničavanje prilikom plaćanja

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

Ukoliko korisnik izvrši plaćanje, potvrda o plaćanju će se prikazati nakon pet sekundi.

2) **Naziv nefunkcionalnog zahtjeva:** Dostupnost proizvoda

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

Ukoliko odabranog proizvoda nema na stanju, prikazat će se odoređena poruka nakon tri sekunde.

3) **Naziv nefunkcionalnog zahtjeva:** Brzina učitavanja podataka

Opis:

Opisati ograničenje sistema i način na koje se ono ispoljava.

Aplikacija osigurava brzo učitavanje podataka.

Paterni ponašanja

*paterni koji će biti implementirani sigurno su boldirani

1. State pattern

Navedeni pattern mijenja način ponašanja objekata na osnovu trenutnog stanja. Stanje se ne mijenja po želji korisnika, nego automatski prema definiciji sistema. State pattern je dinamička verzija Strategy pattern-a. U našem sistemu, potencijalni kupac koji ima napravljen korisnički račun, ali nije izvršio prijavu (dakle, uradio je register, ali nije log-in) ima isti tretman i mogućnosti kao i korisnik kojeg nazivamo gost (dakle, onaj ko nije izvršio register i ne posjeduje korisnički nalog). Dok se ne prijavi, korisnik ne može izvršiti kupovinu, nego ima samo mogućnost pregleda proizvoda iz ponude i njihovih detalja kao i gost.

2. Template method pattern

Template method pattern je pattern koji ima jednu glavnu klasu koja ima razne metode. Ta klasa se grana na druge klase koje nasljeđuju sve metode iz bazne klase, ali neke metode može promijeniti, te će različite klase na različite načine obavljati neku metodu. Ovaj pattern smo odlučile implementirati na sljedeći način. Dodat ćemo jednu apstraktnu klasu koja će se zvati PrikazMenija. Kontrolerske klase ProizvodController i NarudzbaController će biti povezane s ovom klasom. Abstraktna klasa PrikazMenija ima dvije metode: prikazProizvoda() i azurirajCijenu(Proizod p).

3. Mediator pattern

Mediator pattern služi kao međuobjekt da bi se izbjeglo direktno povezivanje velikog broja objekata. Mediator je zadužen za komunikaciju između objekata, odnosno za proslijeđivanje poruke od jednog objekta do drugog. U našem sistemu, kako je već naglašeno ranije, nema kompleksnih veza ni mnogo povezanih klasa tako da ovaj pattern trenutno nije pogodan za korištenje.

4. Chain of Responsibility pattern

Chain of responsibility pattern namjenjen je da bi se jedan kompleksni proces obrade razdvojio na način da više objekata na različite načine procesiraju podatke. Svoju primjenu ovaj pattern može pronaći u našoj funkcionalnosti izvršenja narudžbe. Cjelokupan proces naručivanja započinje pregledom proizvoda i detalja i odabirom proizvoda. Zatim, proizvod se ubacuje u korpu. Nakon toga, obračunava se popust ukoliko postoji te se eventualno ažurira cijena i vrši se plaćanje. Konačno, korisnik dobiva potvrdu o uspješnoj narudžbi. Ova kompleksna aktivnost podijeljena je na više podaktivnosti i u njoj učestvuje više aktera kao i klasa i kontrolera.

5. Iterator pattern

Ovaj pattern omogućava sekvencijalni pristup elementima kolekcije bez poznavanja kako je kolekcija struktuirana. Nismo implementirale ovaj pattern, ali potencijalna ideja za implementaciju može biti da uvedemo opciju pregleda proizvoda prema veličini ili prema boji. Za to bi nam bile potrebne dvije klase, VelicinaIterator i BojaIterator koje će imati metodu `sljedeciProizvod():Proizvod`.

6. Observer pattern

Uloga Observer pattern-a je da uspostavi relaciju između objekata tako da kada jedan objekat promijeni stanje, drugi zainteresovani objekti se obavještavaju. U našem sistemu, mogle bismo iskoristiti ovaj pattern pri slanju notifikacije “Nije potreban poseban povod” koja će biti poslana korisniku ukoliko nije izvršio kupovinu određenog vremenskog perioda. Također, slična primjena bi bila ukoliko bismo slali obavijest korisnicima o sniženju tj. popustu na određene proizvode ili popustu na cjelokupnu ponudu za specijalne prilike.

7. State pattern

Ovaj pattern se koristi kada imamo različita stanja u koje neka klasa, tj. njen objekat može doći. Stanja se automatski mijenjaju, odnosno mi ih ne možemo svojevremeno promijeniti. Ovaj pattern bi se mogao implementirati u našem programu tako što bi uvele različita stanja narudžbe. Prvo stanje u kojem se narudžba može nalaziti jeste da je tek kreirana. Sljedeće stanje u kojem se može naći jeste u izradi. I posljednje stanje u kojem se narudžba može naći jeste završetak.