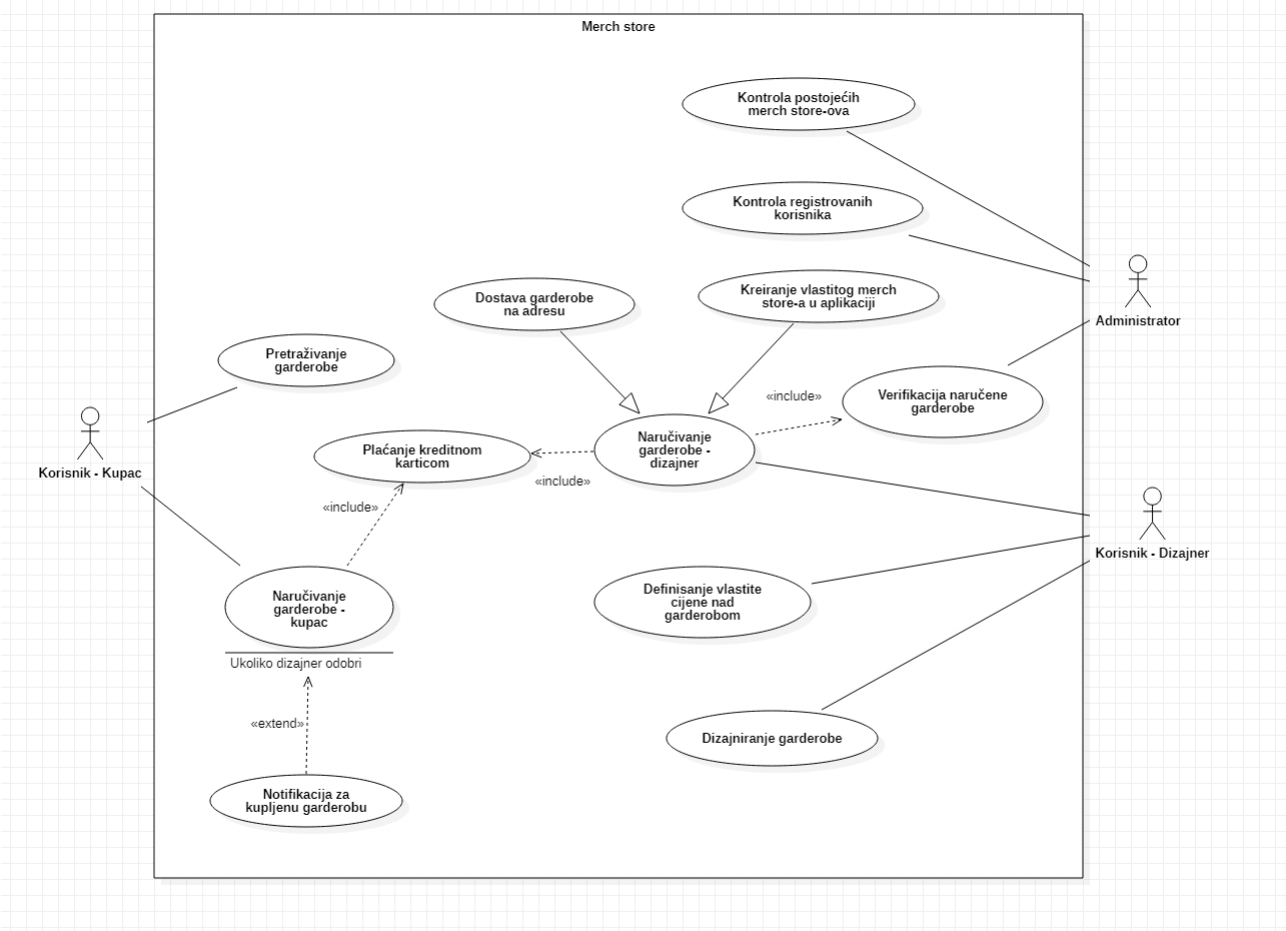


Merch store - Dokumentacija

Dijagram aktivnosti:



Scenarija i dijagrami aktivnosti

Scenario broj 1: Naručivanje garderobe (dizajner)

Naziv Slučaja upotrebe	Naručivanje garderobe (dizajner)
Opis slučaja upotrebe	Korisnik naručuje garderobu koju je prethodno dizajnirao uz validaciju administratora
Vezani zahtjevi	-
Preduslovi	1.1 Dizajniranje garderobe
Posljedice - uspješan završetak	Uspješno naručivanje garderobe
Posljedice - neuspješan završetak	Narudžba odbijena
Primarni akteri	Korisnik
Ostali akteri	Administrator
Glavni tok	Nakon dizajniranja garderobe, korisnik naručuje garderobu. Nakon narudžbe, administrator dobija notifikaciju za validiranje narudžbe. Ukoliko se narudžba invalidira ili ne validira u roku od 72 sata, narudžba se poništava. Nakon validacije korisnik vrši plaćanje preko kreditne kartice. Korisnik može odlučiti da garderobu naruči ili na adresu ili na vlastiti merch store u okviru aplikacije.
Alternative/proširenja	-

Tok događaja 1.1: uspješan završetak

Korisnik	Administrator	Sistem za validaciju kartica
Naručivanje garderobe na adresu		
	Validacija narudžbe	
Unošenje podataka za plaćanje		
		Provjera podataka i plaćanje

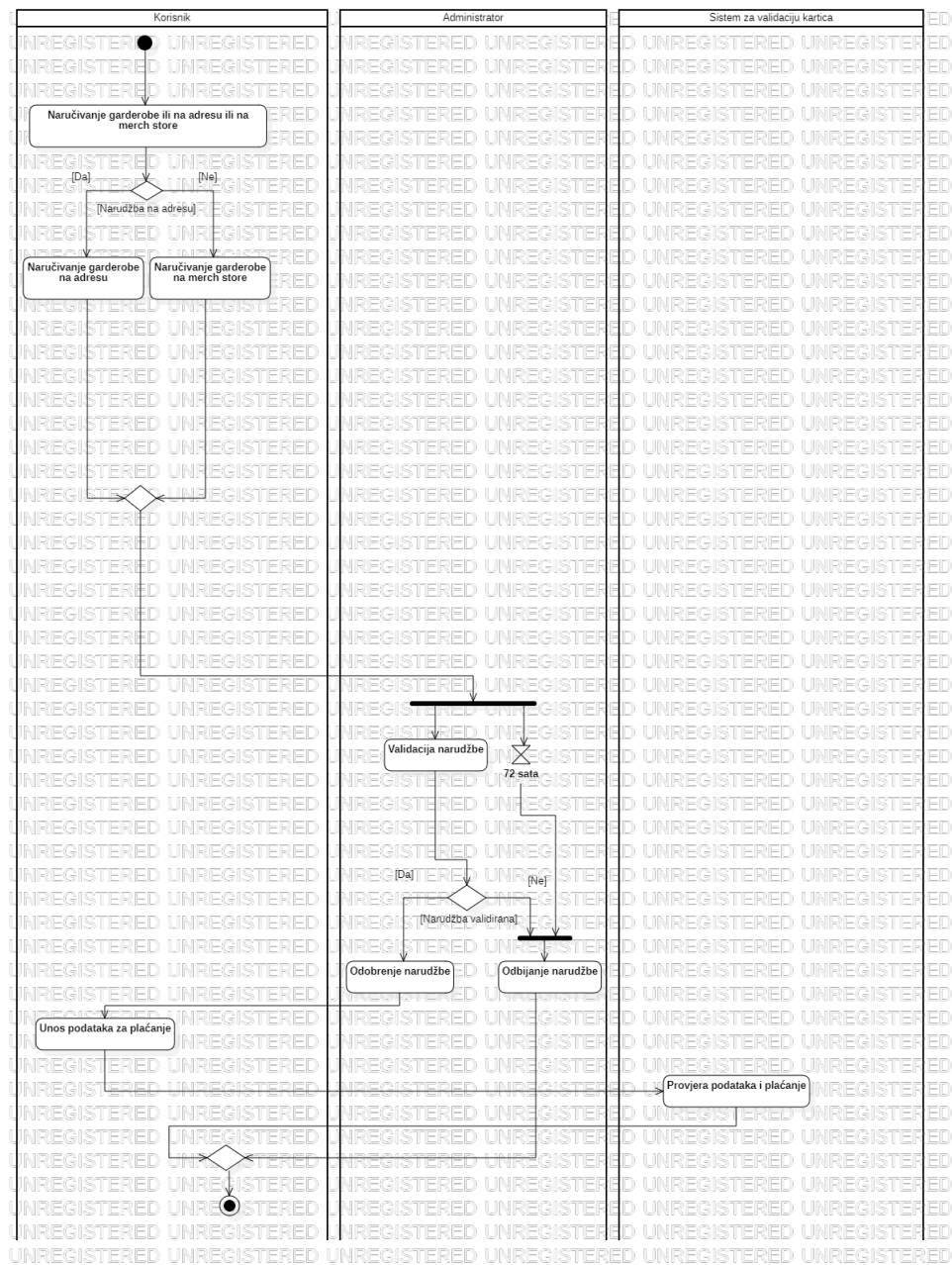
Tok događaja 1.2: uspješan završetak

Korisnik	Administrator	Sistem za validaciju kartica
Naručivanje garderobe na merch store u okviru aplikacije		
	Validacija narudžbe	
Unošenje podataka za plaćanje		
		Provjera podataka i plaćanje

Tok događaja 1.3: neuspješan završetak

Korisnik	Administrator	Sistem za validaciju kartica
Naručivanje garderobe na adresu		
	ponišćavanje narudžbe jer je administrator invalidirao zahtjev ili je proteklo 72 sata bez validacije administratora	

Dijagrami aktivnosti za scenario 1:



Scenario broj 2: Naručivanje garderobe (korisnik)

Naziv Slučaja upotrebe	Naručivanje garderobe (korisnik)
Opis slučaja upotrebe	Korisnik naručuje garderobu preko koda koji mu pošalje dizajner
Vezani zahtjevi	-
Preduslovi	2.1 Pretraživanje garderobe
Posljedice - uspješan završetak	Uspješno naručivanje garderobe
Posljedice - neuspješan završetak	Narudžba odbijena
Primarni akteri	Korisnik
Ostali akteri	Sistem
Glavni tok	Nakon pretraživanja pomoću koda, korisnik naručuje artikal koji želi. Naručivanje uključuje unošenje adrese na koju će artikal biti dostavljen i plaćanje kreditnom karticom. Ukoliko su podaci neispravni, narudžba se odbija. Ukoliko dizajner (vlasnik merch store-a sa kojeg se naručuje) ima uključene notifikacije, dobija notifikaciju na vanjski uređaj (mobilni, e-mail).
Alternative/proširenja	Notifikacija za kupljenu garderobu

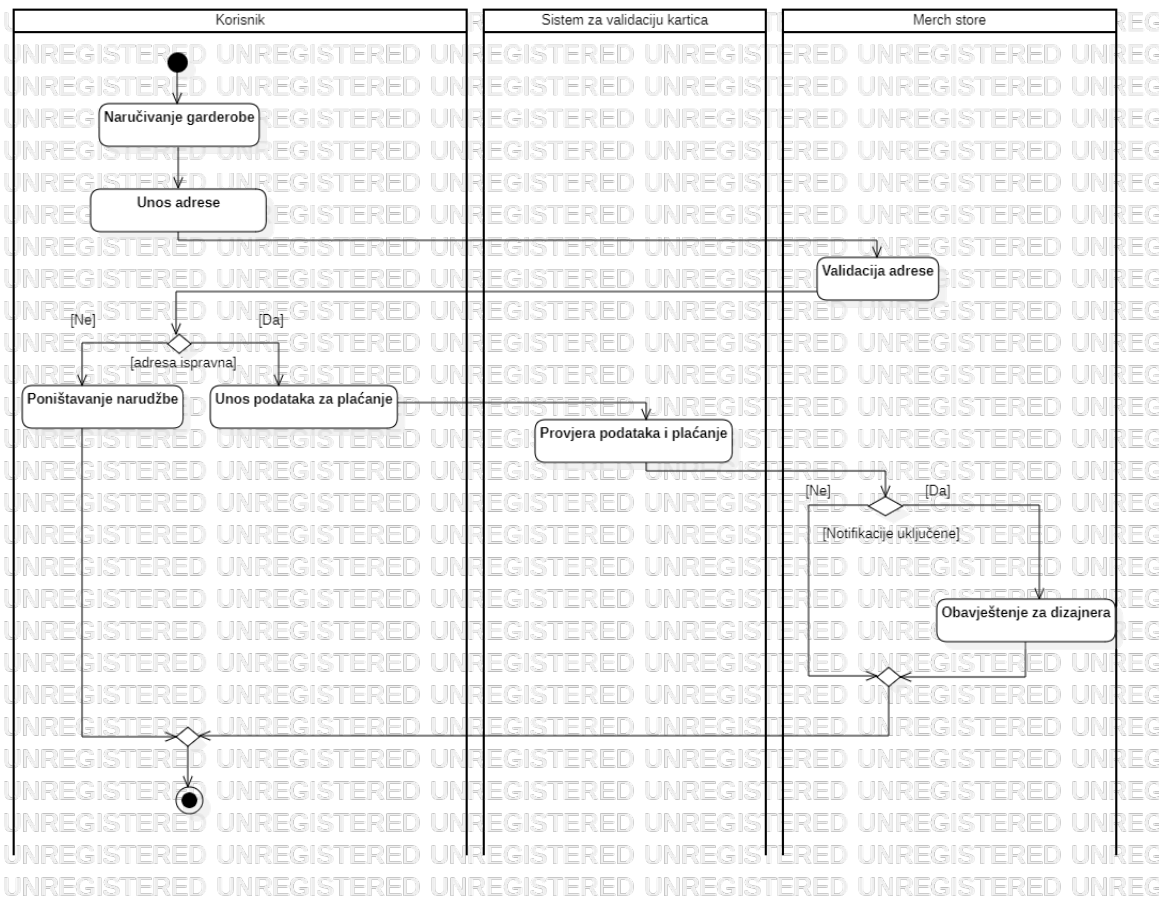
Tok događaja 2.1: uspješan završetak

Korisnik	Sistem za validaciju kartica	Merch store
Naručivanje garderobe		
Unošenje adrese		
		Validacija adrese
Unošenje podataka za plaćanje		
	Provjera podataka i plaćanje	
		Obavještenje za dizajnera

Tok događaja 2.2: neuspješan završetak

Korisnik	Sistem za validaciju kartica	Merch store
Naručivanje garderobe		
Unošenje podataka		
		Odbijanje narudžbe zbog neispravne adrese

Dijagrami aktivnosti za scenario 2:



Scenario broj 3: Naručivanje garderobe na merch store

Naziv Slučaja upotrebe	Naručivanje garderobe na merch store
Opis slučaja upotrebe	Korisnik naručuje garderobu na merch store
Vezani zahtjevi	3.1 Naručivanje garderobe (dizajner)
Preduslovi	3.1 Naručivanje garderobe (dizajner)
Posljedice - uspješan završetak	Uspješno naručivanje garderobe
Posljedice - neuspješan završetak	Narudžba odbijena
Primarni akteri	Korisnik
Ostali akteri	Sistem
Glavni tok	Nakon što je korisnik odlučio da naruči garderobu na vlastiti merch store, provjerava se da li korisnik ima prethodno kreiran merch store. Ukoliko nema mora kreirati. Kreiranje podrazumijeva popunjavanje podataka za adresu korisnika i kreditne kartice na koju će biti uplaćivanje transakcije drugih korisnika koji budu kupovali sa datog merch store-a. Ukoliko su podaci neispravni, narudžba se odbija. Ukoliko su ispravni ili korisnik ima merch store, korisnik postavlja cijenu na naručenu garderobu.
Alternative/proširenja	Kreiranje merch store-a

Tok događaja 3.1: uspješan završetak

Korisnik	Merch store
Naručivanje garderobe na merch store	
Unošenje podataka potrebnih za kreiranje merch store-a	
	Validacija podataka
Postavljanje cijene na garderobu	

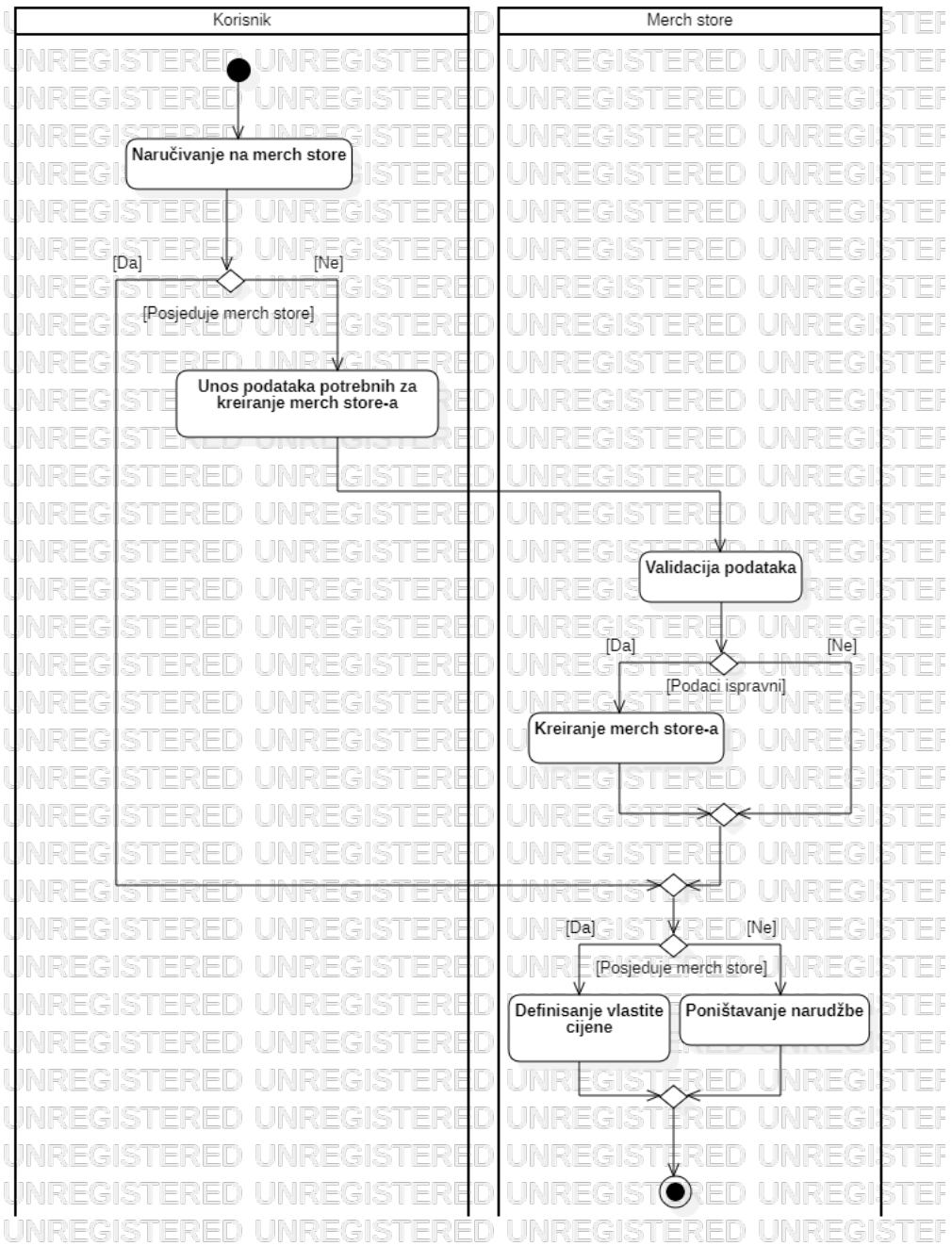
Tok događaja 3.2: neuspješan završetak

Korisnik	Merch store
Naručivanje garderobe na merch store	
Postavljanje cijene na garderobu	

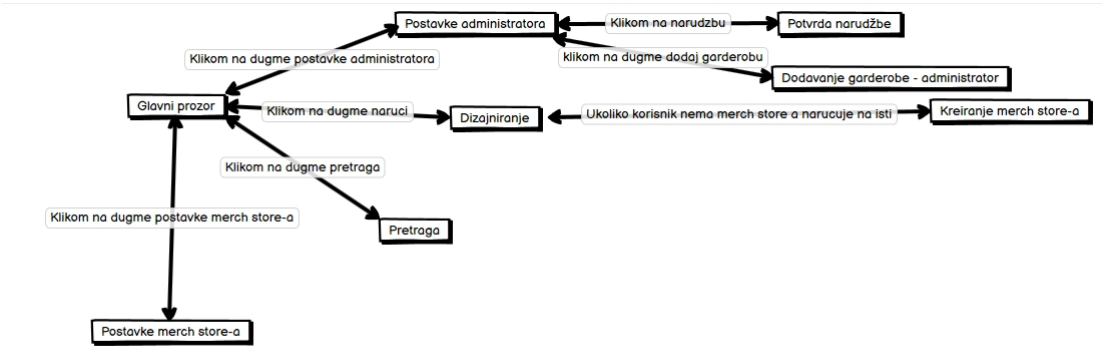
Tok događaja 3.3: neuspješan završetak

Korisnik	Merch store
Naručivanje garderobe na merch store	
Unošenje podataka potrebnih za kreiranje merch store-a	
	Poništavanje narudžbe zbog pogrešnih podataka

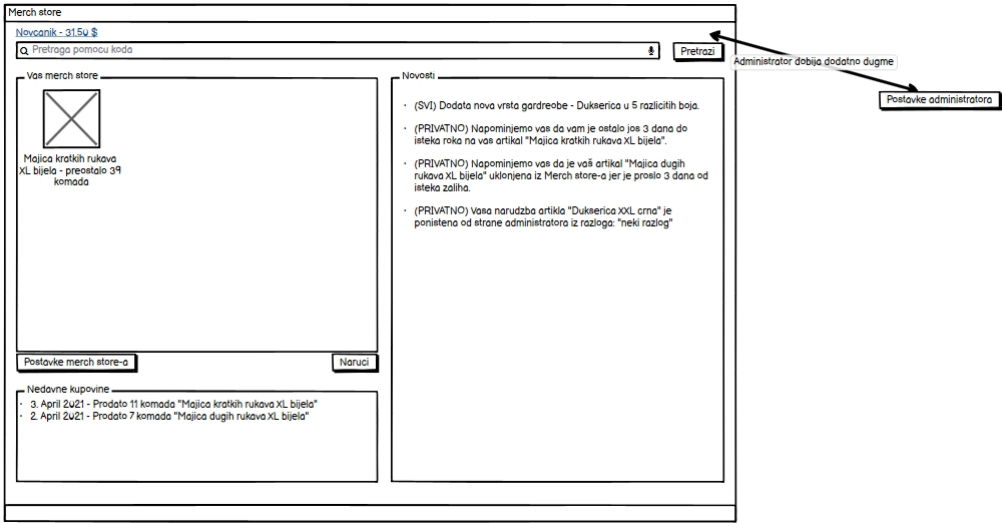
Dijagrami aktivnosti za scenario 3:



Korisnički interfejs
Glavni dijagram toka:



Glavni prozor:



Dizajniranje:

Dizajniranje

Trenutni izgled

Na garderobi se nalaze dugmad koja oznacavaju pozicije na kojima se slika moze nalaziti. Pozicije i velicinu slika odreduje administrator pri ubacivanju garderobe

Torso lijevo

TM

Torso desno

Garderoba na raspolaganju

Majica kratkih rukava

Kosulja

Dukserica

Postavke

Naziv artikla

Boja (bijela)

Velicina (XL)

Kolicina

☐ Dostava na adresu

☐ Dostava na merch store

Adresa

Potvrdi (prelaz na placanje)

Poništi

Greska

Ne posedujete merch store, da li zelite kreirati?

Ne

Da

Ukoliko ne posedujete merch store

Onemogucen unos ukoliko je odabrana opcija dostava na merch store

Dodavanje garderobe - administrator:


Izgled

Torso centar

Torso lijevo

Torso desno

Pri dodavanju garderobe u aplikaciju, administrator mora postaviti sliku date garderobe, te odrediti pozicije na kojima ce dizajneri moći postaviti svoje slike. Nakon postavljanja imena pozicije i pritiska na dugme dodaj poziciju, administrator može raditi drag&drop na mjesto garderobe koje želi



Potvrdi poziciju

Ponisti zadnju poziciju

Ponisti sve

Naziv pozicije

Dodaj novu poziciju

Postavke garderobe

Naziv garderobe

Cijena

Pocetna kolicina

Boja

Velicina

Bijela

Crna

Plava

XL

XXL

M

Obrisi boju

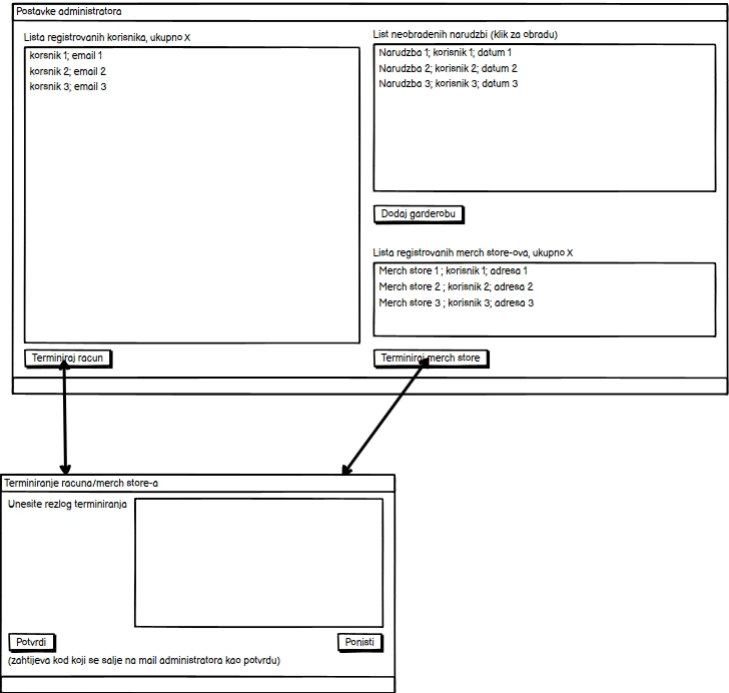
Obrisi Velicinu

Potvrdi

Kreiranje merch store-a:



Postavke administratora:



Postavke merch store-a:

Postavke

☐ Ukljuci notifikacije na email

☐ Ukljuci notifikacije na vanjski uredaj

Povezi/odvezi vanjski uredaj sa merch store-om

Trenutna adresa

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Promjeni

Ovo je adresa na koju ce biti dostavljena preostala garderoba

Postavke artikala

Odaberite artikla

Preostala kolicina : X komada

Preostalo dozvoljeno vrijeme u merch store-u: X dana

Obnovi zalih

Dostavi preostalo na adresu

Povezivanje vanjskog uredaja

Povezivanje

Trenutno podrzavamo samo mobilni uredaj

Broj telefona

XXX-XXX-XXX

Odvezivanje

Odaberite uredaj

Potvrdi

Potvrda narudžbe:

Potvrda narudžbe

Narudžba od korisnika: XXXXXXX

Trenutni izgled

Torso lijevo

TM

Torso desno

Potvrdi

Poništi

Poništi

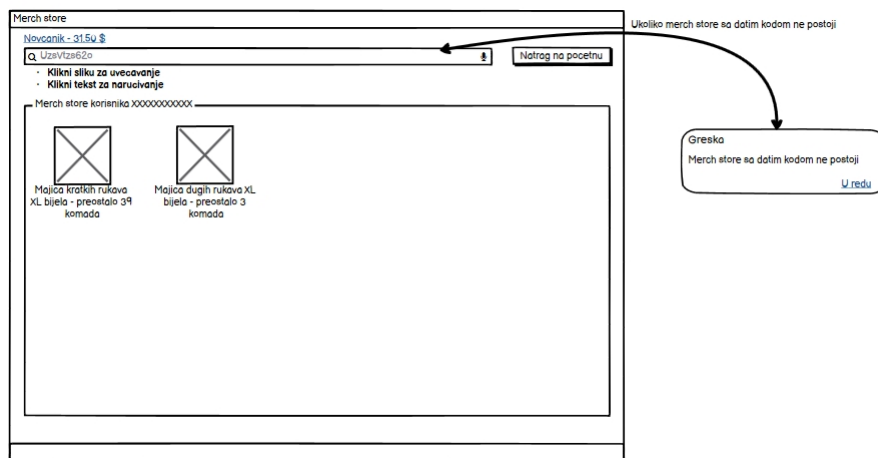
Poništavanje narudžbe

Unesite razlog poništavanja

Potvrdi

Poništi

Pretraga:



SOLID principi:

Single responsibility i open-closed principi:

Klasa Korisnik: Klasa koja opisuje korisnika. Implementira 2 interfejsa: INarucivanje i IPlacanje. Narucivanje kao funkcionalnost je odvojeno od ove klase iz razloga što narucivanje mijenja stanje baze (tj. nije zadatak ove klase), a u budućnosti bi se mogao dodati novi tip korisnika koji ne bi mogao naručivati (npr. guest). Kontrola uplatama nije stvar sa kojom bi se ova klasa trebala baviti, pa je taj dio odvojen u zaseban interfejs.

Klasa Administrator: Klasa koja opisuje administratora. Implementira 4 interfejsa: IObavijest, ITerminiranje, IPotvrdivanje, IBaza. Razlog što su ove funkcionalnosti administratora odvojene od same klase je zbog postojanja mogućnosti nadogradnje sistema npr. mogućnost dodavanja moderatora koji će moći potvrđivati narudžbe ali ne i terminirati korisnike ili mogućnost da jedan korisnik šalje poruku drugom korisniku.

Klasa Garderoba: Klasa koja opisuje garderobu koju korisnik može dizajnirati. Klasa nema specifičnih zahtijeva. Klasa MerchStore: Klasa koja opisuje merch store. Slično kao i klasa Garderoba, implementira interfejs IMerchStoreBaza.

Klasa NarudzbaDizajnera: Klasa koja opisuje stavke narudžbe koje je korisnik postavio pri narudžbi. Ova klasa nema specijalnih zahtijeva.

Jedan dodatni primjer ispunjavanja Single responsibility principa su interfejsi INarucivanje i IDizajnNarucivanje koji zasebno služe za naručivanje od strane dizajnera i od strane kupca.

Liskov substitution princip:

Ovaj princip je primjenjen na 2 načina.

Prvi način je vezan za klase Administrator i Korisnik. Vidimo da su atributi ovih klasa slični te da bi možda mogli postaviti da je Korisnik generalizacija Administratora. Međutim to nije dobra ideja iz razloga što su korisnik i administrator imaju različite prioritete i mogućnosti (npr. administrator nema mogućnost naručivanja ili dizajniranja).

Drugi način je vezan za klase Garderoba i NarudzbaDizajnera. Slično kao i prethodni način, ove klase imaju slične attribute, međutim one se koriste u potpuno drugačijim situacijama, te ih ne treba povezivati generalizacijom.

Interface segregation princip:

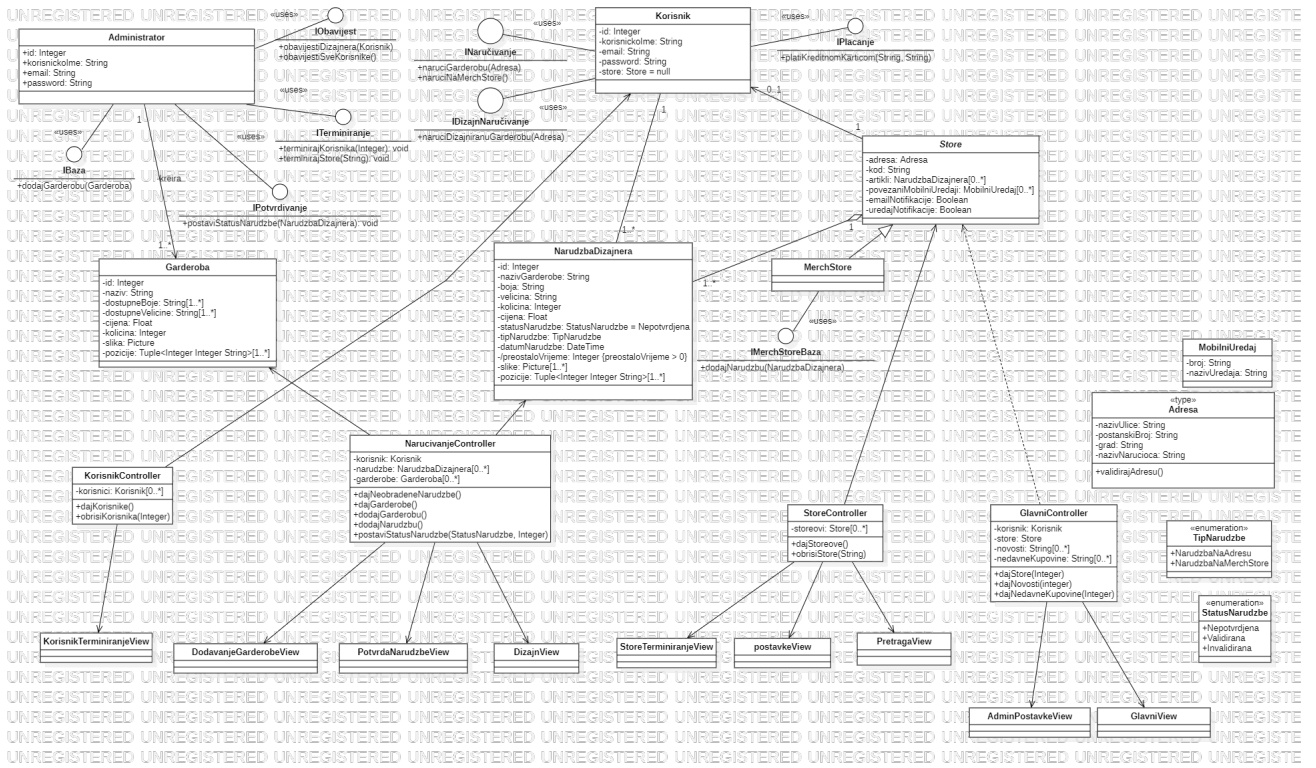
Prva primjena je kod interfejsa koje implementira klasa Administrator. Naime mi smo mogli interfejsse ITerminiranje i IPotvrdivanje spojiti u jedan interfejs (npr. IAdministratorPrivilegije), ali onda ukoliko želimo dodati novi tip administratora (moderator) koji bi samo mogao potvrđivati narudžbe ali ne i terminirati, to ne bismo mogli uraditi.

Druga primjena je kod interfejsa koje implementira klasa Korisnik. Slično kao i prethodna primjena, mi smo mogli ove interfejsse spojiti u jedan interfejs INarucivanje, ali ako npr. želimo dodati mogućnost da neregistrovani korisnici mogu samo naručivati (ali ne i dizajnirati i posjedovati merch store), to ne bi mogli.

Dependency inversion princip:

Primjena kod klase MerchStore. Ukoliko želimo dodati novu klasu npr. VIPMerchStore, tada ukoliko korisnik ima atribut tipa MerchStore, on neće moći posjedovati VIPMerchStore. Stoga smo napravili novu apstraktnu klasu Store, te promijenili atribut kod klase Korisnik u Store.

SOLID dijagram klasa sa MVC paternom:



Strukturalni paterni:

Proxy:

Ovaj pattern se primjenjuje pri pretraživanju store-ova: Kada korisnik pretražuje merch store preko koda, tada on treba da dobije samo artikle koji se nalaze u datom store-u, dok kada administrator pretražuje, pored artikala, treba se ispisati i korisničko ime korisnika koji posjeduje store i adresa na koju je store povezan.

Facade:

Ovaj pattern se primjenjuje kod kreiranja merch store-a. Naime kada korisnik kreira merch store, on samo treba upisati adresu, dok se u pozadini vrši generisanje koda, validiranje adrese i dodavanje store-a u bazu.

Adapter:

Pretpostavimo da želimo dodati API koji će omogućiti validaciju adresa. Međutim taj api se neće moći direktno iskoristiti na objekte tipa Adresa. U toj situaciji bi mogli dodati adapter klasu sa kojom bi mogli omogućiti validiranje adrese pomoću dodanog API-ja.

Dekorater:

Ukoliko pretpostavimo da garderoba, umjesto da se dodaje slika za svaku boju zasebno, mijenja boju algoritamskim putem, onda bi pri dizajniranju garderobe mogli dodati dekorator jer bi se na istu garderobu morale primijeniti dvije promjene: promjena boje i dodavanje slike.

Bridge:

Pretpostavimo da dodamo novu klasu zvanu Moderator, koji će moći mijenjati status narudžbi ali neće moći terminirati korisnike i storeove. Tada možemo primijeniti ovaj pattern jer i administrator i moderator koriste istu metodu.

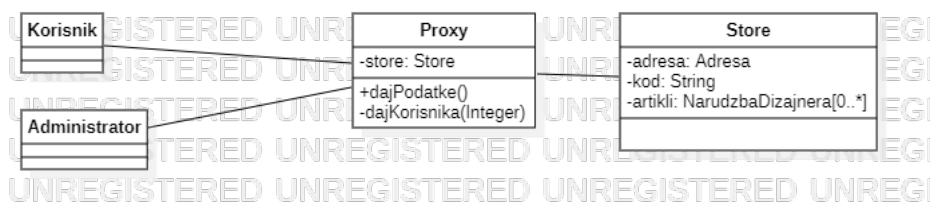
Composite:

Pretpostavimo da smo dodali novu klasu nazvanu VIPStore u kojem se obračunavaju popusti na svaki tip garderobe, te dodali novu mogućnost administratoru koji dozvoljava da se ispiše neto vrijednosti svih store-ova. Tada bi mogli iskoristiti Composite, tako da administrator jednom metodom može izračunati neto vrijednost svakog store-a bez obzira na tip store-a.

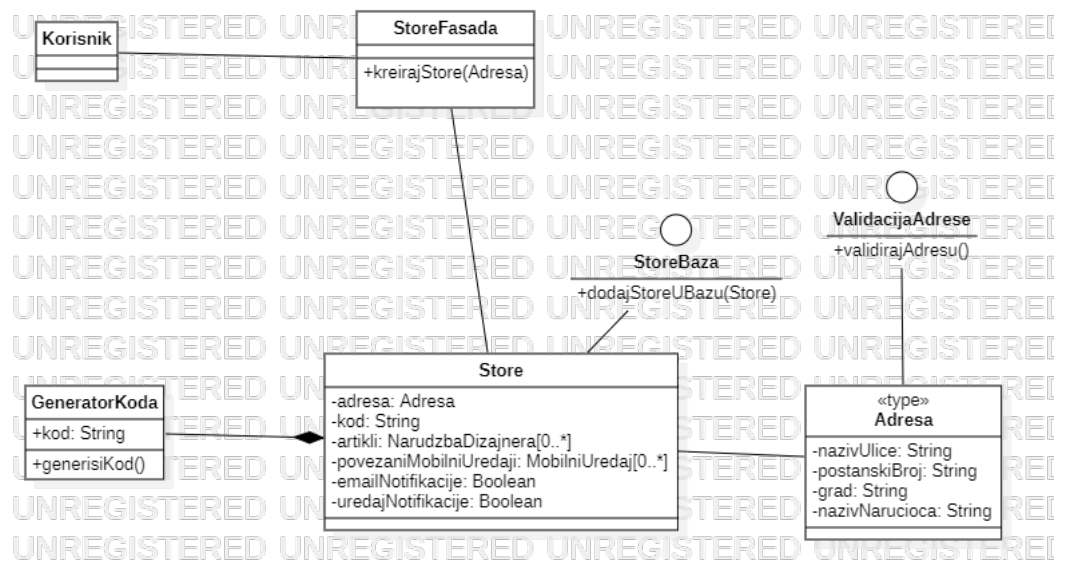
Flyweight:

Primjena kod klase Garderoba. Pri prikazivanju ove klase na korisničkom interfejsu, umjesto da kreiramo novu instancu ove klase za svaku boju garderobe, možemo samo mijenjati slike, a ostale attribute ne mijenjati.

Proxy patern:



Facade patern:



Kreacijski paterni:

Singleton patern:

Možemo primjeniti na obavijest koja se prosljeđuje svim korisnicima, tj. kako svaki korisnik vidi tu obavijest, nije potrebno praviti zasebnu instancu za svakog korisnika.

Prototype patern:

Pošto u sistemu postoji mogućnost da korisnik obnovi rok narudžbe, ovaj patern bi mogli primjeniti u ovom slučaju, tj. umjesto da korisniku ponovno prikazujemo interfejs dizaniranja, možemo jednostavno kopirati prethodni narudžbu i promijeniti preostalo vrijeme.

Factory method patern:

Ukoliko pored klase MerchStore imamo dodatnu klasu PremiumStore koja takođe nasljeđuje klasu Store, tada bi ovaj patern mogli iskoristiti na način da u zavisnosti od toga koji tip Store-a korisnik želi kreirati, taj tip se i kreira pomoću neke druge klase ili interfejsa.

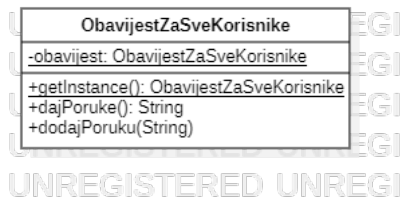
Abstract factory patern:

Ukoliko pored MerchStore imamo i PremiumStore, te pored NarudzbaDizajnera imamo PremiumNarudzba i ove dvije vrste narudzbi nasljeđuju klasu Narudzba, tada možemo iskoristiti ovaj patern jer bi store klase imaju zajedničku metodu dodajNarudzbu, a narudzbe mogu biti različitog tipa.

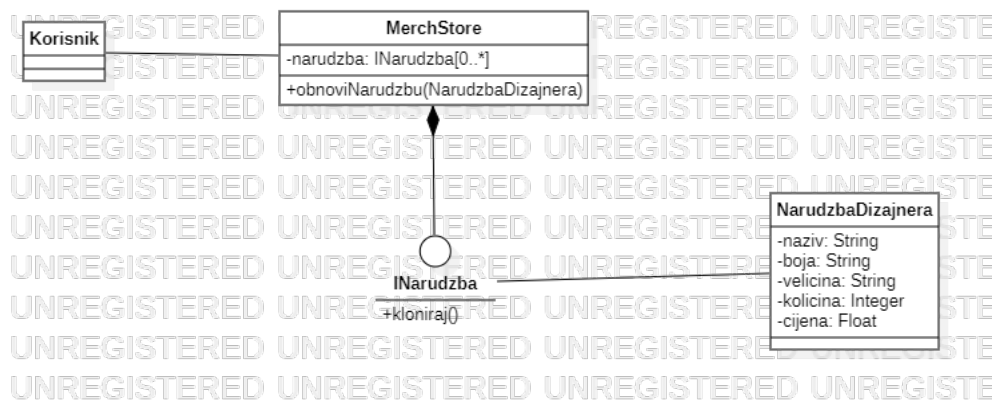
Builder patern:

Builder patern se može primjeniti kod klase Store. Kako je za kreiranje store-a potrebno generisanje koda, možemo korisniku dozvoliti da kreira vlastiti kod ili da se kod generiše automatski.

Singleton patern:



Prototype patern:



Paterni ponašanja:**Strategy patern:**

Ovaj patern možemo primjeniti kod prikazivanja liste korisnika u interfejsu terminiranja korisnika. Možemo odobriti da administrator odluči po kojem kriteriju želi sortirati korisnike (po korisničkom imenu, mailu ili kodu merch store-a).

State patern:

Ukoliko pretpostavimo da imamo još i klasu PremiumStore, tada bi ovaj patern mogli iskoristiti na način da ukoliko korisnik posjeduje premium store, tada u opcijama pored svakog artikla ispisuje se i popust na taj artikal koji korisnik može promijeniti.

Template patern:

Ovaj patern možemo primjeniti pri dizajniranju garderobe, pošto se dizajniranje sastoji iz više koraka. Npr. možemo klasu Garderoba rastaviti na dvije klase: ObojenaGarderoba i NeobojenaGarderoba. Ukoliko dizajner odluči naručiti neobojevu garderobu, tada plaća manji iznos (jer se garderoba ne mora farbat). Na sličan način bi mogli napraviti i za dodavanje slika (ukoliko se želi naručiti bez dodavanja slika plaća se manji iznos jer se garderoba ne more slati na printanje).

Observer patern:

Ovaj patern možemo primjeniti pri validaciji narudžbe jer je tad potrebno obavijestiti dizajnera o tome da li je narudžba validirana ili poništena. Ovaj patern takođe možemo primjeniti i pri kupovini garderobe jer korisnik treba imati informaciju o količini prodane garderobe posljednjih nekoliko dana. Još jedna primjena ovog paternu jer pri dodavanju nove garderobe za dizajniranje, jer tada svi dizajneri trebaju biti obaviješteni da postoji novi tip garderobe na raspolaganju.

Iterator patern:

Ukoliko na glavni interfejs dodamo novu mogućnost koja omogućava korisniku da "srcoll-a" kroz artikle, te prikazuje neke statistike (npr. ukupno zarada na artiklu, ukupno prodato zadnjih 3-7-30 dana i sl.) tada bi mogli primjeniti ovaj patern. Korisnik bi mogao zatražiti da vidi sve detalje u zasebnom interfejsu (dakle ova mogućnost bi bila mini verzija novog intefejsa) i da mijenja kriterij po kojem se "srcoll-a" kroz artikle.

Chain of responsibility patern:

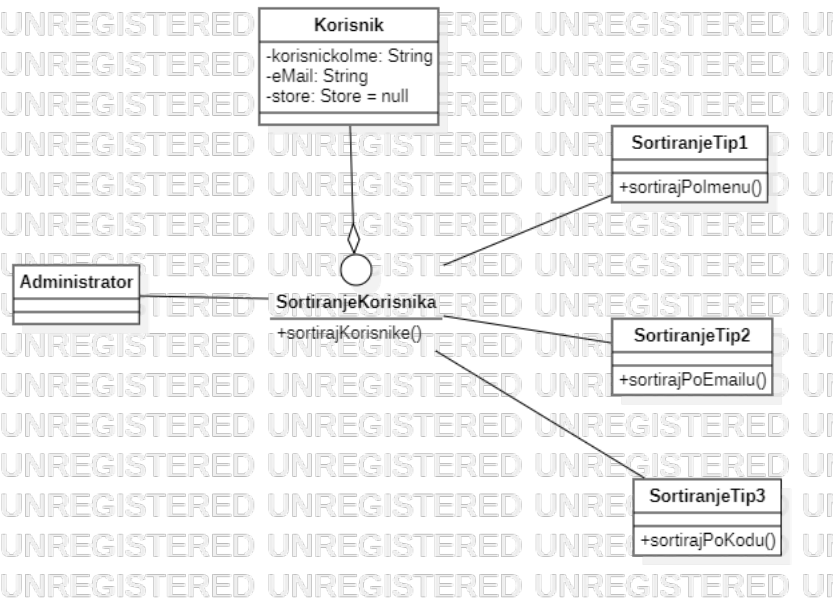
Ukoliko dodamo novu funkcionalnost koja omogućava dizajneru da umjesto čitavog procesa dizajniranja, odabere samo boju, veličinu i slike a na osnovu tih informacija se kreiraju garderobe koje podržavaju date parametre nakon čega dizajner bira jednu ili više od dizajniranih garderoba za naručiti, tada bi mogli primjeniti ovaj

patern.

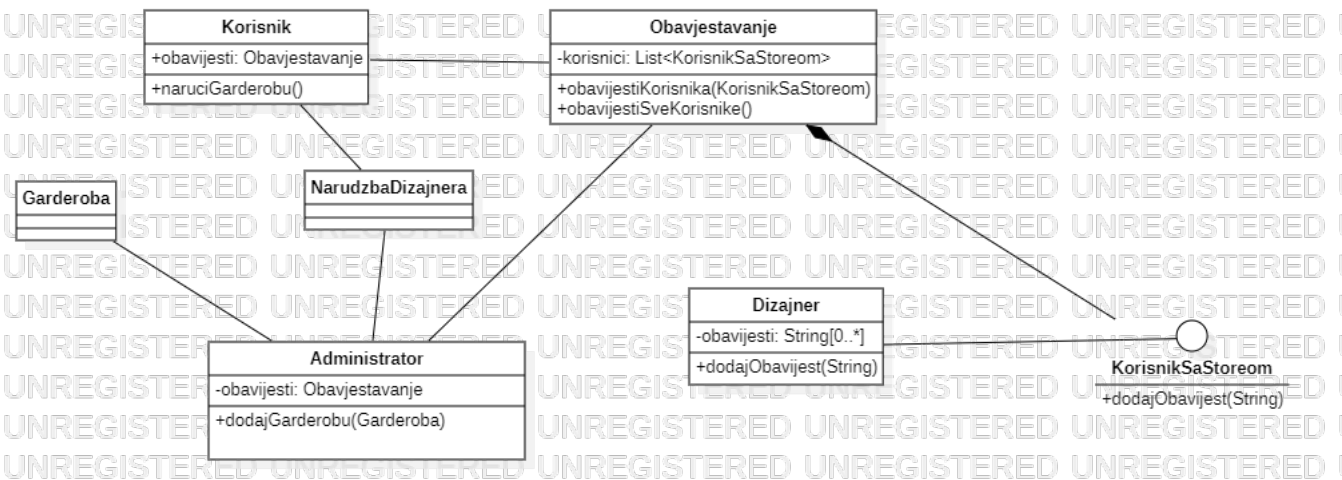
Mediator patern:

Ukoliko pretpostavimo da imamo još i klasu PremiumStore, tada bi ovaj patern mogli primjeniti na način se provjerava ma koji tip store-a ide narudžba, te ako se ustanovi da narudžba ide u premium store, tada se ona automatski validira, u suprotnom se proslijeđuje administratoru na validaciju.

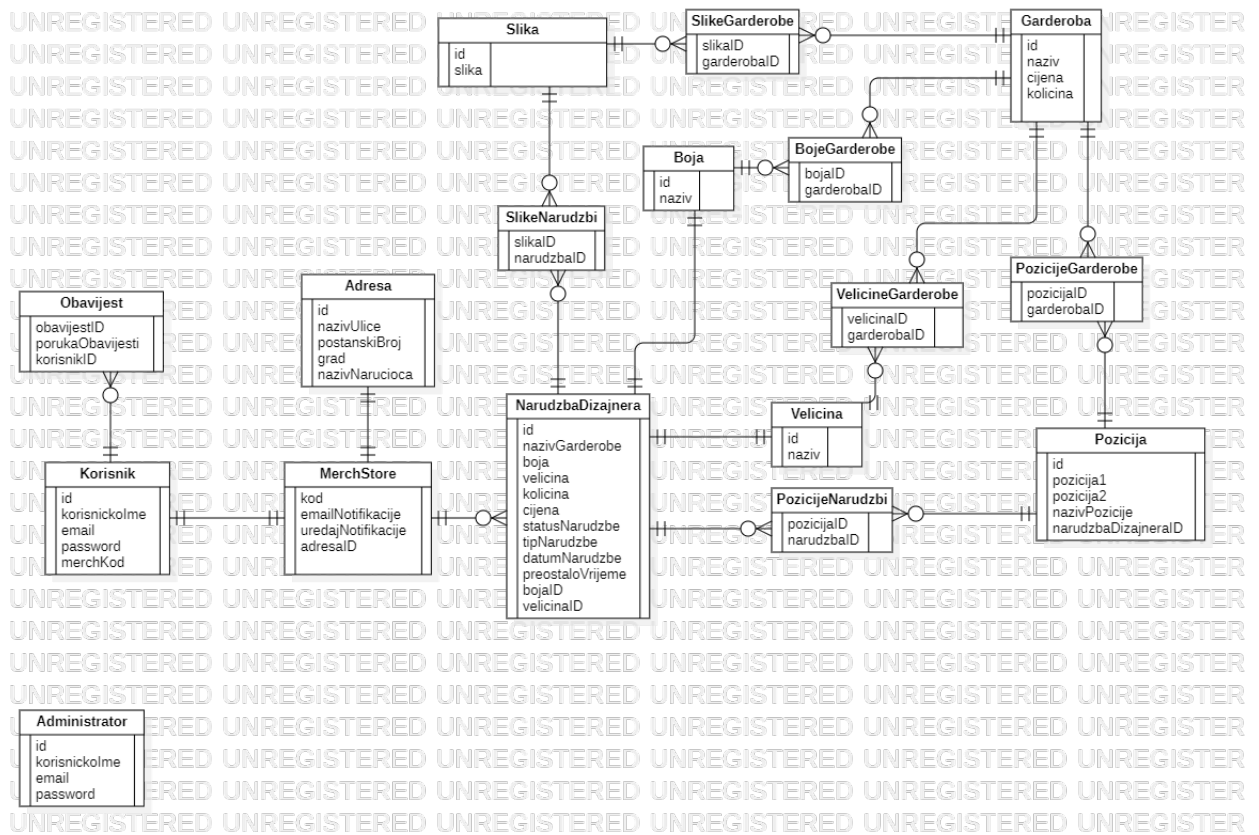
Strategy pattern:



Observer pattern:

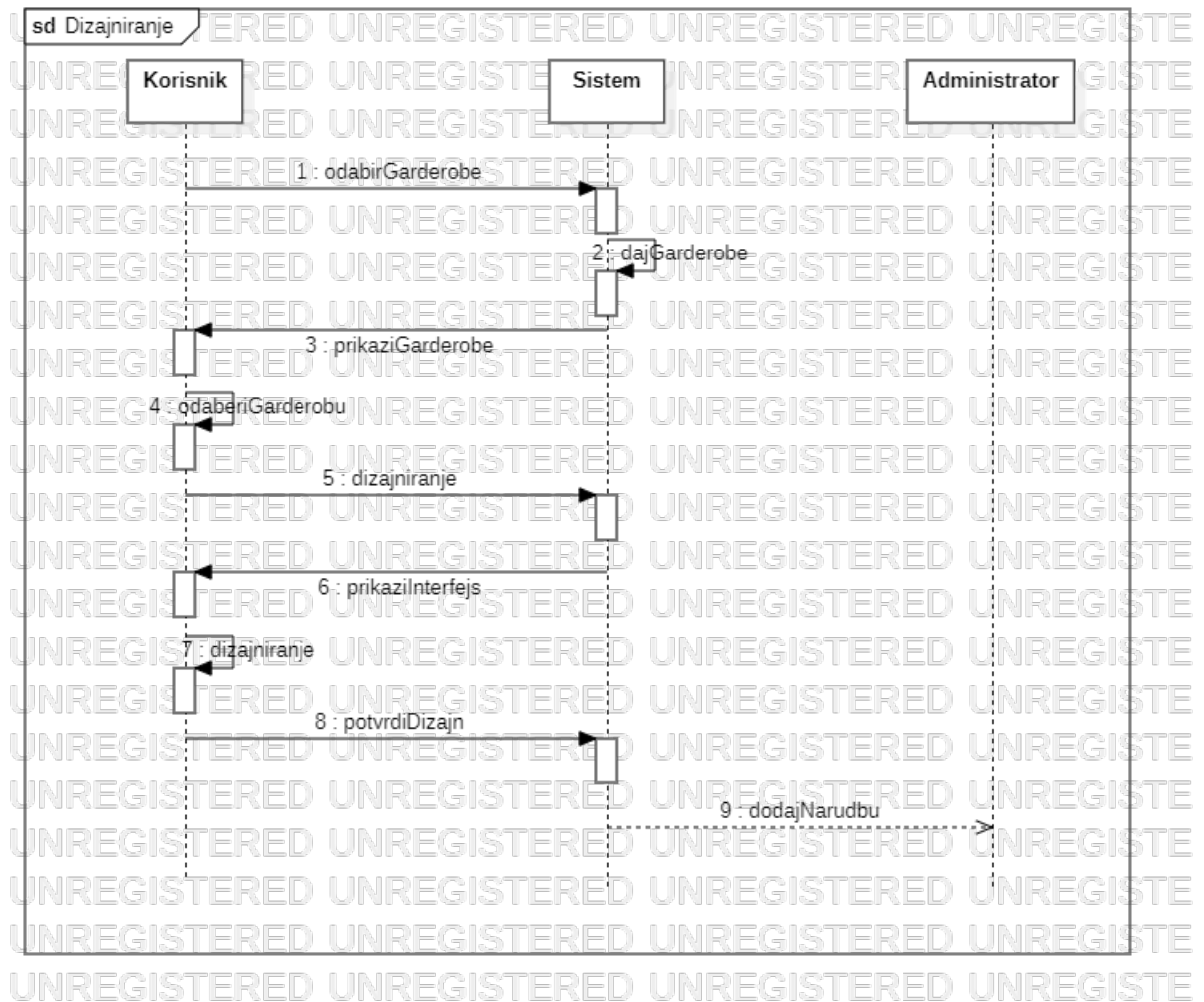


ERD dijagram

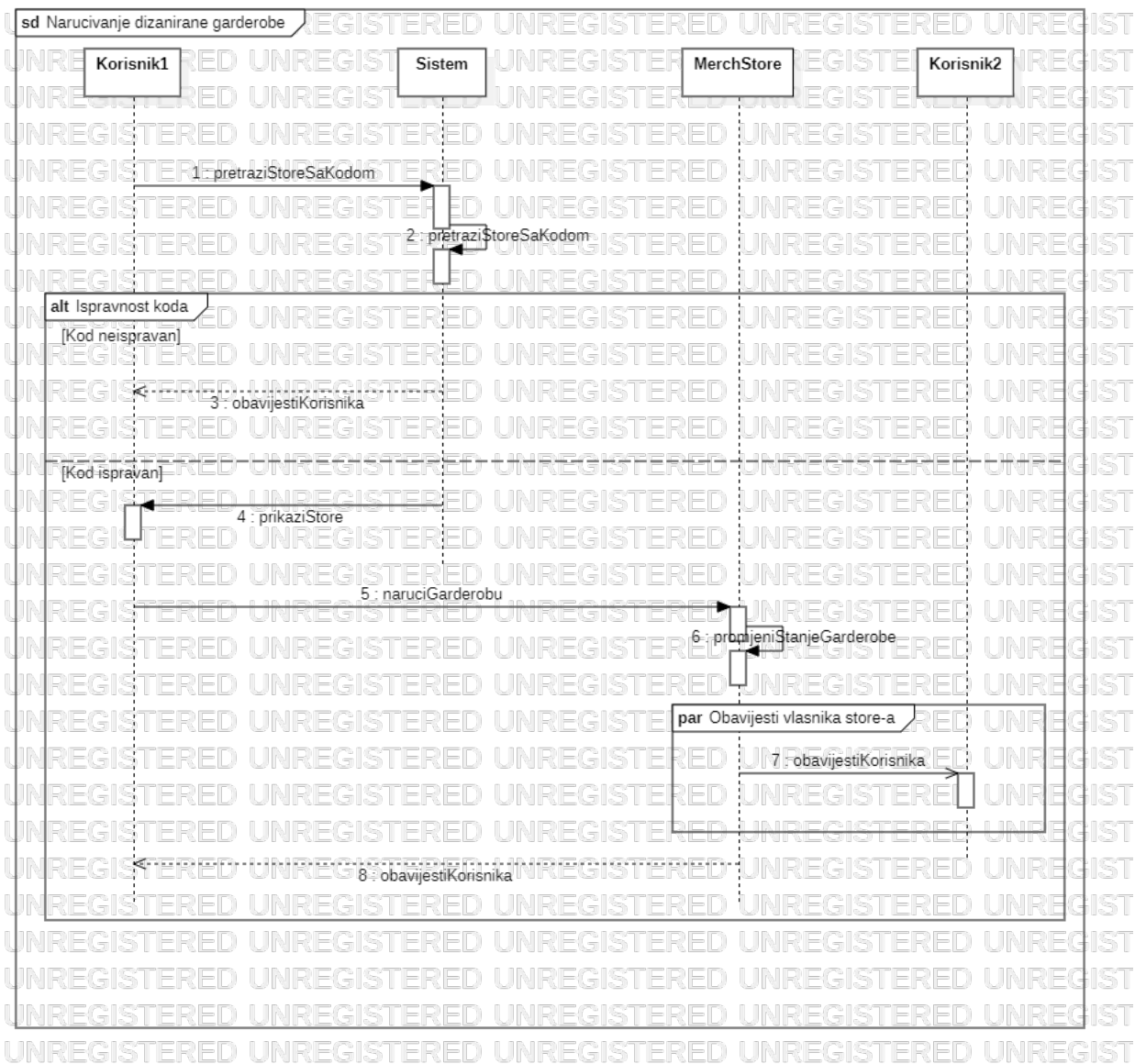


Dijagrami sekvenci

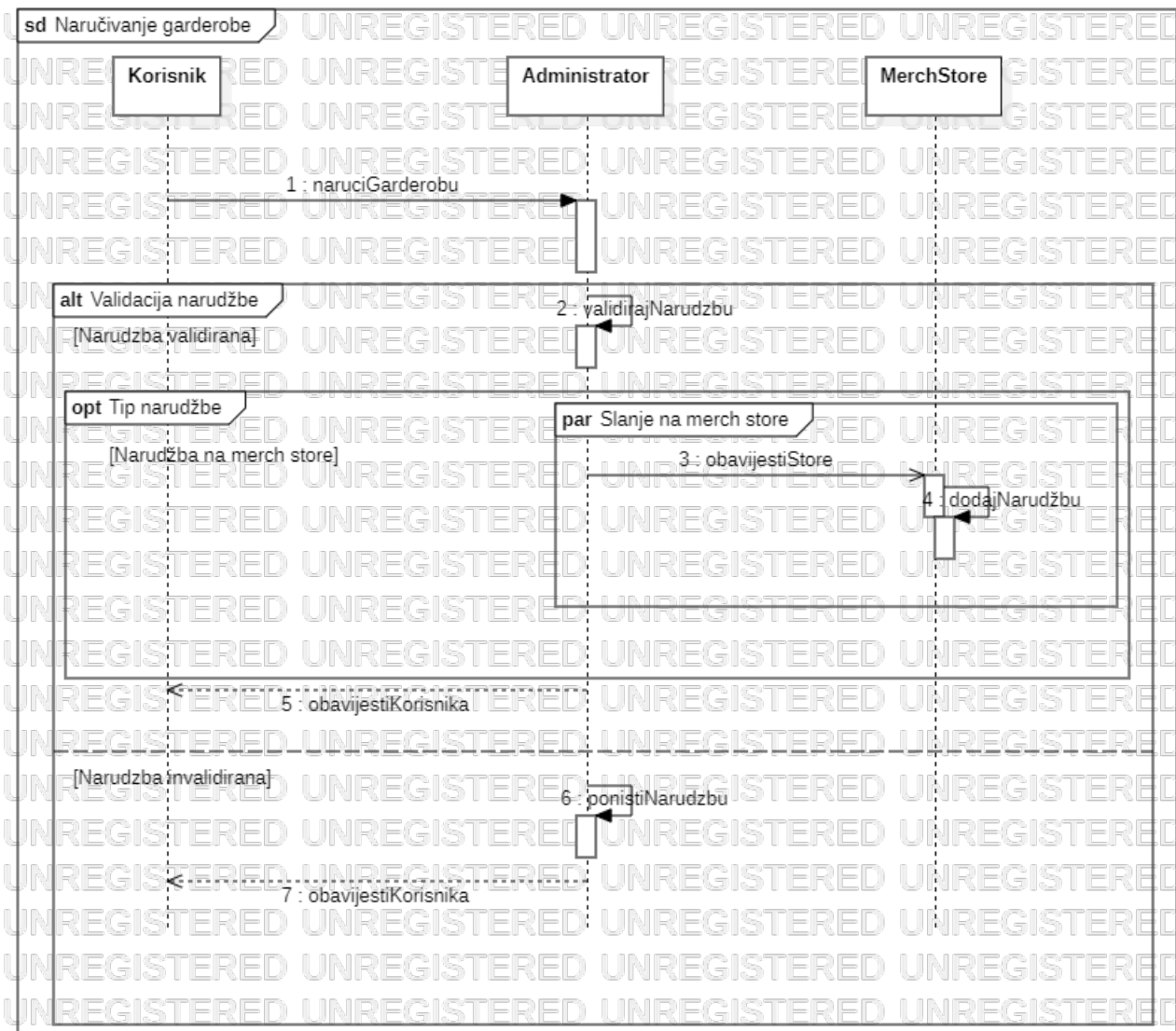
Dizajniranje:



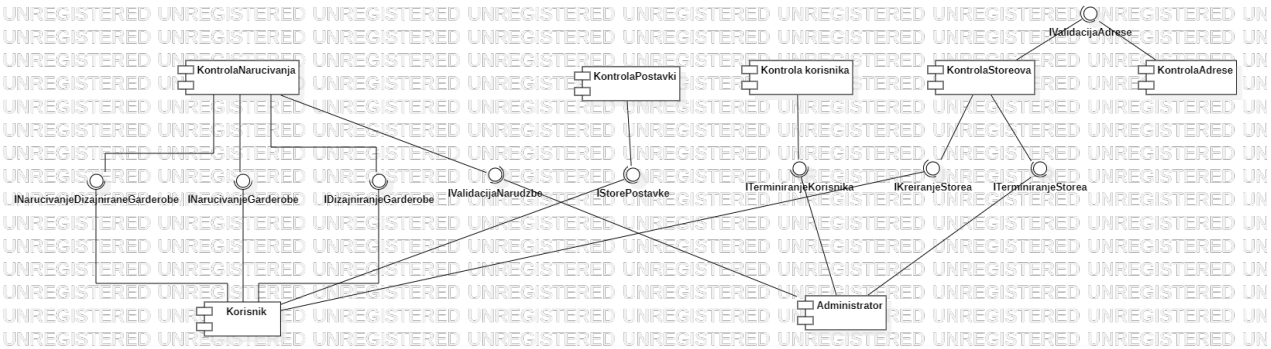
Naručivanje dizajnirane garderobe:



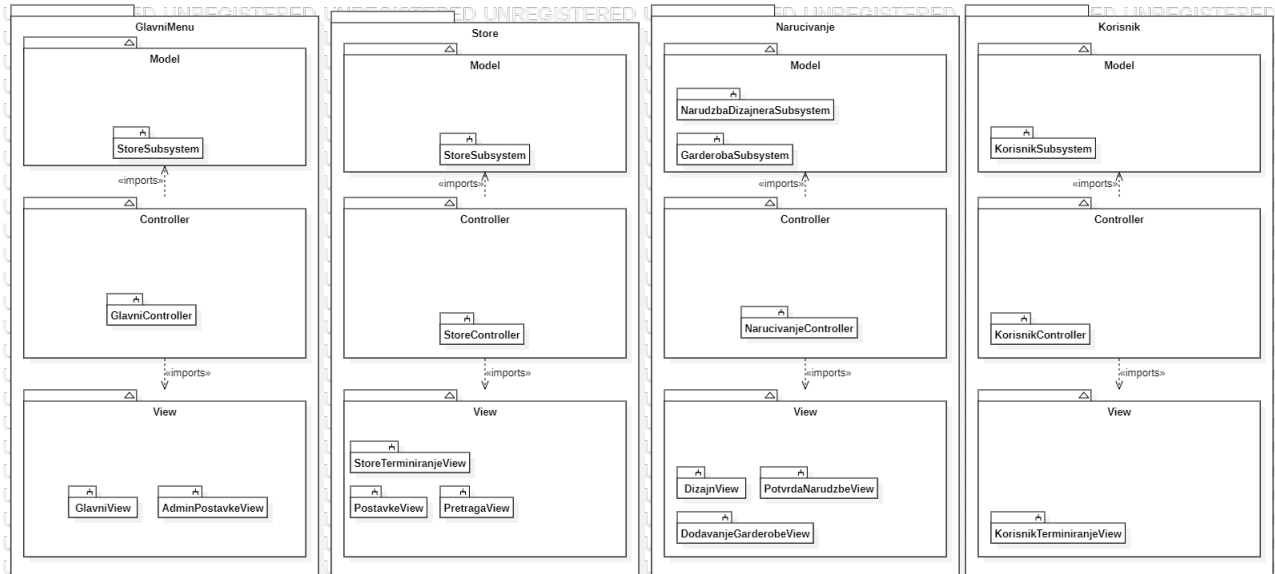
Naručivanje garderobe:



Dijagram komponenti



Dijagram paketa



Dijagram raspoređivanja

