SVEUČILIŠTE U ZAGREBU

**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Seminar iz kolegija

Objektno oblikovanje

Kozmetički salon

Tina Knežević

Magdalena Lisica

Marta Posavec

Zagreb, siječanj 2019.

Sadržaj

[**1.** **Uvod** 1](#_Toc536093056)

[***1.1.*** ***Design story*** 1](#_Toc536093057)

[**2.** **Funkcionalni zahtjevi** 3](#_Toc536093058)

[***2.1.*** ***Funkcionalnosti sustava*** 3](#_Toc536093059)

[***2.2.*** ***Razrada funkcionalnih zahtjeva kroz obrasce uporabe*** 3](#_Toc536093060)

[*2.2.1.* *Prikaz narudžbi klijenata za trenutni datum* 3](#_Toc536093061)

[*2.2.2.* *Promjena datuma za prikaz narudžbi klijenata* 4](#_Toc536093062)

[*2.2.3.* *Dodavanje nove narudžbe (Desktop aplikacija)* 4](#_Toc536093063)

[*2.2.4.* *Dodavanje nove narudžbe (Web aplikacija)* 5](#_Toc536093064)

[*2.2.5.* *Prikaz svih artikala na skladištu* 6](#_Toc536093065)

[*2.2.6.* *Promjena količina artikala na skladištu* 7](#_Toc536093066)

[*2.2.7.* *Dodavanje novog artikla na skladište* 7](#_Toc536093067)

[*2.2.8.* *Prikaz svih usluga kozmetičkog salona* 7](#_Toc536093068)

[*2.2.9.* *Dodavanje i uklanjanje usluga za salon* 8](#_Toc536093069)

[*2.2.10.* *Unos nove nabave artikala* 8](#_Toc536093070)

[*2.2.11.* *Pregled prošlih nabava artikala* 9](#_Toc536093071)

[*2.2.12.* *Detaljan prikaz nabave artikala* 9](#_Toc536093072)

[*2.2.13.* *Prikaz svih zaposlenika salona* 9](#_Toc536093073)

[*2.2.14.* *Dodavanje novog zaposlenika* 9](#_Toc536093074)

[*2.2.15.* *Uklanjanje zaposlenika* 10](#_Toc536093075)

[*2.2.16.* *Uređivanje podataka o zaposleniku* 11](#_Toc536093076)

[*2.2.17.* *Pregled detalja o zaposleniku* 11](#_Toc536093077)

[*2.2.18.* *Pregled zarade zaposlenika po mjesecima* 12](#_Toc536093078)

[**3.** **Opis objektnog modela** 13](#_Toc536093079)

[***3.1.*** ***ER dijagram*** 13](#_Toc536093080)

[***3.2.*** ***Dijagram razreda*** 14](#_Toc536093081)

[***3.3.*** ***Opis modela*** 15](#_Toc536093082)

[**4.** **Opis implementacije perzistencije** 20](#_Toc536093083)

[**5.** **Opis izgrađene desktop aplikacije** 24](#_Toc536093084)

[**6.** **Opis izgrađene web aplikacije** 28](#_Toc536093085)

[***6.1.*** ***Modeli i modeli za prikaz*** 28](#_Toc536093086)

[***6.2.*** ***Kontroleri*** 29](#_Toc536093087)

[***6.3.*** ***Prikazi*** 31](#_Toc536093088)

[***6.4.*** ***Ekrani web aplikacije*** 32](#_Toc536093089)

# **Uvod**

Cilj seminara jest opisati način rada aplikacije koja je zamišljena da se koristi u svrhu vođenja poslovanja kozmetičkih salona. Aplikacija je izvedena primjenom principa i tehnika objektnog oblikovanja.

## ***Design story***

Kako bi si ubrzali i olakšali poslovanje, djelatnici kozmetičkog salona odlučili su staromodno zapisivanje u bilježnice i rokovnike zamijeniti računalnim vođenjem poslovanja, kroz desktop aplikaciju. Do sad su u rokovnike bilježili narudžbe klijenata te ručno zapisujući vodili evidenciju o artiklima koje imaju na skladištu, te njihovim dobavljačima i nabavama. Dakle, aplikacija koju će koristiti treba podržavati naručivanje klijenata te vođenje evidencije o artiklima, što uključuje pregled stanja artikala na skladištu te narudžbe artikala od dobavljača. Dodatno, aplikacija će omogućiti pregled svih zaposlenika salona, zajedno s uslugama koje ti zaposlenici pružaju, pregled svih usluga koje salon pruža te pregled svih prošlih i budućih narudžbi klijenata i prošlih nabavi artikala. Dakle, aplikacija salonu treba nuditi:

* popis zaposlenika
* popis usluga
* popis narudžbi klijenata
* popis artikala
* popis nabavi artikala
* dodavanje i brisanje zaposlenika
* dodavanje i brisanje usluga
* unos nove narudžbe klijenta
* promjenu stanja artikala na skladištu
* nabavu artikala

Svaki zaposlenik ima osnovne podatke koje ima svaka osoba (ime, prezime, OIB, adresa), smjenu u kojoj radi (jutarnja ili popodnevna) te popis usluga koje zna obavljati. Svaka usluga ima naziv, cijenu, okvirno trajanje, kategoriju usluge, opcionalni opis usluge te popis zaposlenika koji obavljaju tu uslugu. Svaka narudžba klijenta sadrži ime i kontakt klijenta, uslugu za koju se klijent naručio, datum i vrijeme narudžbe te zaposlenika koji će primiti tog klijenta. Svaki artikl ima naziv, cijenu, kategoriju, proizvođača, dobavljača te količinu tog artikla na skladištu. Dobavljači imaju naziv, OIB, kontakt i adresu. Svaka nabava artikala ima dobavljača od kojeg se nabavljaju artikli, datum, ukupnu cijenu te popis svih naručenih artikala, zajedno s količinama.

Kako bi se mogućnost korištenja ove aplikacije proširila na veći broj salona, dodatno je realizirano i web sučelje, kroz koje se različiti saloni mogu prijaviti u aplikaciju te ju koristiti s istom funkcionalnošću kao kod opisane desktop aplikacije.

# **Funkcionalni zahtjevi**

## ***Funkcionalnosti sustava***

1. Prikaz narudžbi klijenata za trenutni datum
2. Promjena datuma za prikaz narudžbi klijenata
3. Dodavanje nove narudžbe
4. Prikaz svih artikala na skladištu
5. Promjena količine artikala na skladištu
6. Dodavanje novog artikla na skladište
7. Prikaz svih usluga kozmetičkog salona
8. Dodavanje i uklanjanje usluga za salon
9. Unos nove nabave artikala
10. Pregled prošlih nabava artikala
11. Detaljan prikaz nabave artikala
12. Prikaz svih zaposlenika
13. Dodavanje novog zaposlenika
14. Uklanjanje zaposlenika
15. Uređivanje podataka o zaposleniku
16. Pregled zarade zaposlenika po mjesecima

## ***Razrada funkcionalnih zahtjeva kroz obrasce uporabe***

### *Prikaz narudžbi klijenata za trenutni datum*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje početni zaslon odabranog kozmetičkog salona.

**Postcondition:** Sustav prikazuje sve narudžbe za trenutni dan.

**Glavni uspješni scenarij:**

1. Zaposleniku je prikazan početni zaslon.
2. Zaposlenik odabire gumb „Narudžbe“
3. Prikaz svih narudžbi za danas.

### *Promjena datuma za prikaz narudžbi klijenata*

**Primary actor:** Zaposlenik

**Precondition:** Klikom na gumb „Narudžbe“ otvorio se prikaz svih narudžbi. (2.1.1)

**Postcondition:** Zaslon prikazuje popis narudžbi za odabrani datum.

**Glavni uspješni scenarij:**

1. Glavni scenarij 2.1.1. (Odabir prikaza svih narudžbi)
2. Odabir željenog datuma
3. Odabrati opciju „Promjena datuma“.
4. Prikaz svih narudžbi za odabrani datum.

### *Dodavanje nove narudžbe (Desktop aplikacija)*

**Primary actor:** Zaposlenik

**Precondition:** Klikom na gumb „Narudžbe“ otvorio se prikaz svih narudžbi. (2.1.1)

**Postcondition:** Dodana nova narudžba u sustav.

**Glavni uspješni scenarij:**

1. Glavni scenarij 2.1.1. (Odabir prikaza svih narudžbi)
2. Klik na gumb „Nova narudžba“
3. Odabir željene usluge
4. Odabir datuma i vremena nove narudžbe
5. Klik na gumb „Pronađi zaposlenike“
6. Odabir jednog od zaposlenika koji pruža odabranu uslugu za odabrano vrijeme
7. Klik na gumb „Provjeri dostupnost“
8. Pojavljuje se prozor „Odabrani zaposlenik je SLOBODAN u odabranom terminu“
9. Unos imena i prezimena klijenta
10. Unos kontakta klijenta
11. Klik na gumb „Dodaj narudžbu“
12. Pojavljuje se poruka "Nova narudžba dodana.“

**Alternativni scenarij:**

**A)**

1. Glavni scenarij 2.1.1. (Odabir prikaza svih narudžbi)
2. Klik na gumb „Nova narudžba“
3. Odabir željene usluge
4. Odabir datum i vrijeme nove narudžbe
5. Klik na gumb „Pronađi zaposlenike“
6. Odabir jednog od zaposlenika koji pruža odabranu uslugu za odabrano vrijeme
7. Klik na gumb „Provjeri dostupnost“
8. Pojavljuje se prozor „Odabrani zaposlenik je ZAUZET u odabranom terminu“
9. Unos imena i prezimena klijenta
10. Unos kontakta klijenta
11. Klik na gumb „Dodaj narudžbu“
12. Pojavljuje se prozor "Unijeti novu narudžbu iako je zaposlenik zauzet?"
13. Odabir „Yes/No“
14. [YES]->Pojavljuje se poruka "Nova narudžba dodana.“
15. [NO]-> Pojavljuje se poruka "Poništeno.“

**B)**

1. Glavni scenarij 2.1.1. (Odabir prikaza svih narudžbi)
2. Klik na gumb „Nova narudžba“
3. Odabir željene usluge
4. Odabir datum i vrijeme nove narudžbe
5. Klik na gumb „Pronađi zaposlenike“
6. Odabir jednog od zaposlenika koji pruža odabranu uslugu za odabrano vrijeme
7. Klik na gumb „Provjeri dostupnost“
8. Pojavljuje se prozor „Odabrani zaposlenik je ZAUZET u odabranom terminu“
9. 🡪korak 6.

### *Dodavanje nove narudžbe (Web aplikacija)*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut.

**Postcondition:** Dodana nova narudžba u sustav.

**Glavni uspješni scenarij:**

1. Klikom na gumb „Nova narudžba“ otvara se forma za unos nove narudžbe.
2. Zaposlenik unosi ime i kontakt klijenta i odabire uslugu i datum za narudžbu te klikće na gumb „Dalje“.
3. Otvara se nova stranica, gdje je potrebno odabrati termin, zajedno sa zaposlenikom koji će odraditi tu narudžbu.
4. Nakon odabira željenog termina i klijenta, potrebno je kliknuti na gumb „Gotovo“ i time je narudžba unesena u sustav.

**Alternativni scenarij:**

**A)**

1. Klikom na gumb „Nova narudžba“ otvara se forma za unos nove narudžbe.
2. Zaposlenik nije unio ime ili kontakt klijenta i kliknuo je na gumb „Dalje“.
3. Sustav porukom javlja zaposleniku da treba unijeti podatke.
4. Kad zaposlenik unese sve podatke, scenarij se nastavlja od točke 3) po uspješnom scenariju.

**B)**

1. Klikom na gumb „Nova narudžba“ otvara se forma za unos nove narudžbe.
2. Zaposlenik unosi ime i kontakt klijenta i odabire uslugu i datum za narudžbu te klikće na gumb „Dalje“.
3. Za odabrani datum ne postoji raspoloživi termin. Ponovno se otvara forma za odabir datuma, s porukom da za odabrani datum nema raspoloživog termina te da je potrebno odabrati drugi datum.
4. Korisnik odabire drugi datum i scenarij se nastavlja od točke 3) po uspješnom scenariju.

### *Prikaz svih artikala na skladištu*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje početni zaslon odabranog kozmetičkog salona.

**Postcondition:** Sustav prikazuje sve artikle kozmetičkog salona.

**Glavni uspješni scenarij:**

1. Zaposleniku je prikazan početni zaslon.
2. Zaposlenik odabire gumb „Artikli“
3. Prikaz svih artikala

### *Promjena količina artikala na skladištu*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazan je popis svih artikala

**Postcondition:** Prikaz količine pojedinih artikala je promijenjen

**Glavni uspješni scenarij:**

1. Glavni scenarij 2.2.5. (Prikaz svih artikala).
2. Promjena polja „Količina“ za pojedini artikl na skladištu.
3. Klik na gumb „Promjeni količine“.
4. Količine su uspješno pohranjene.

### *Dodavanje novog artikla na skladište*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje prikaz svih artikala

**Postcondition:** Popis artikala sadrži novi artikl

**Glavni uspješni scenarij:**

1. Glavni scenarij 2.2.5. (Odabir prikaza artikala)
2. Klik na gumb „Dodaj novi artikl“
3. Odabir novog artikla iz popisa
4. Promjena količine novog artikla
5. Klik na gumb „Spremi“
6. Artikl je uspješno dodan.

### *Prikaz svih usluga kozmetičkog salona*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje početni zaslon odabranog kozmetičkog salona.

**Postcondition:** Sustav prikazuje sve usluge kozmetičkog salona

**Glavni uspješni scenarij:**

1. Zaposleniku je prikazan početni zaslon.
2. Zaposlenik odabire gumb „Usluge“
3. Prikazan je popis svih usluga kozmetičkog salona

### *Dodavanje i uklanjanje usluga za salon*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje sve usluge kozmetičkog salona.

**Postcondition:** Sustav prikazuje novi popis usluga kozmetičkog salona.

**Glavni uspješni scenarij:**

1. Glavni scenarij 2.2.8. (Prikaz svih usluga kozmetičkog salona).
2. Zaposlenik odabire gumb „Dodaj/Ukloni uslugu“.
3. Za dodavanje novih usluga označiti *check-box* ispred usluge.
4. Za uklanjanje usluga odznačiti *check-box* ispred označene usluge.
5. Klik na gumb „Spremi“.
6. Usluge su uspješno dodane/uklonjene.

### *Unos nove nabave artikala*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje sve artikle po kategoriji i dobavljačima kozmetičkog salona.

**Postcondition:** Sustav kreira novu nabavu artikala za dobavljače.

**Glavni uspješni scenarij:**

1. Zaposleniku je prikazan početni zaslon.
2. Zaposlenik odabire gumb „Nabava artikala“.
3. Za artikle koje želi nabaviti mijenja polje „Količina“ u željenu količinu tog artikla.
4. Klik na gumb „Nabavi“.
5. Stvoreno je onoliko novih nabava koliko različitih dobavljača imaju artikli odabrani za nabavu.

### *Pregled prošlih nabava artikala*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje sve artikle po kategoriji i dobavljačima kozmetičkog salona.

**Postcondition:** Sustav prikazuje sve nabave, dobavljače nabave i ukupnu cijenu nabave.

**Glavni uspješni scenarij:**

1. Zaposleniku je prikazan početni zaslon
2. Zaposlenik odabire gumb „Nabava“
3. Prikazuju se sve prethodno kreirane nabave

### *Detaljan prikaz nabave artikala*

**Primary actor:** Zaposlenik

**Precondition:** Sustav je pokrenut i prikazuje sve artikle po kategoriji i dobavljačima kozmetičkog salona.

**Postcondition:** Sustav prikazuje sve artikle nabave, uz cijene artikala.

**Glavni uspješni scenarij:**

1. Glavni scenarij 2.2.11. (Pregled nabava)
2. Zaposlenik odabire gumb „Pregledaj“ za nabavu koju želi pregledati
3. Prikazuje se detaljan prikaz nabave

### *Prikaz svih zaposlenika salona*

**Primary actor:** Korisnik

**Precondition:** Sustav je pokrenut.

**Postcondition:** Sustav prikazuje sve zaposlenika salona

**Glavni uspješni scenarij:**

1. Korisnik odabire opciju „Zaposlenici“.
2. Korisniku se prikazuje stranica s popisom svih zaposlenika salona.

### *Dodavanje novog zaposlenika*

**Primary actor:** Korisnik

**Precondition:** Sustav je pokrenut i prikazan je popis svih zaposlenika.

**Postcondition:** Novi zaposlenik uspješno je dodan u sustav.

**Glavni uspješni scenarij:**

1. Korisnik odabire opciju „Dodaj novog zaposlenika“.
2. Otvara se stranica s formom za unos imena, prezimena, OIB-a i adrese, odabir grada i smjene te popisom svih usluga, gdje treba odabrati usluge koje taj zaposlenik zna obavljati.
3. Korisnik unosi sve potrebne podatke te odabire opciju „Gotovo“.
4. Novi zaposlenik je uspješno dodan kao zaposlenik trenutnog salona u sustav.

**Alternativni scenarij:**

1. Korisnik odabire opciju „Dodaj novog zaposlenika“.
2. Otvara se stranica s formom za unos imena, prezimena, OIB-a i adrese, odabir grada i smjene te popisom svih usluga, gdje treba odabrati usluge koje taj zaposlenik zna obavljati.
3. Korisnik nije unio sve potrebne podatke te odabire opciju „Gotovo“.
4. Sustav korisniku javlja poruku da treba unijeti sve potrebne podatke.
5. Kada korisnik unese sve potrebne podatke, scenarij se nastavlja od točke 3) kao uspješni scenarij.

### *Uklanjanje zaposlenika*

**Primary actor:** Korisnik

**Precondition:** Sustav je pokrenut i prikazan je popis svih zaposlenika.

**Postcondition:** Zaposlenik je uspješno uklonjen iz sustava.

**Glavni uspješni scenarij:**

1. Korisnik odabire opciju „Obriši“ pokraj zaposlenika kojeg želi ukloniti.
2. Sustav „postavlja pitanje“ korisniku je li siguran da želi ukloniti tog zaposlenika.
3. Korisnik odabire opciju „Obriši“ te je time zaposlenik uspješno uklonjen iz sustava.

**Alternativni scenarij:**

1. Korisnik odabire opciju „Obriši“ pokraj zaposlenika kojeg želi ukloniti.
2. Sustav „postavlja pitanje“ korisniku je li siguran da želi ukloniti tog zaposlenika.
3. Korisnik odabire opciju „Natrag na popis“ i zaposlenik nije uklonjen iz sustava.

### *Uređivanje podataka o zaposleniku*

**Primary actor:** Korisnik

**Precondition:** Sustav je pokrenut i prikazan je popis svih zaposlenika.

**Postcondition:** Podaci o zaposleniku su uspješno uređeni.

**Glavni uspješni scenarij:**

1. Korisnik odabire opciju „Uredi“ pokraj željenog zaposlenika.
2. Otvara se stranica s formom za ažuriranje imena, prezimena, OIB-a i adrese, odabir grada i smjene te popisom svih usluga, gdje treba odabrati usluge koje taj zaposlenik zna obavljati.
3. Korisnik ažurira podatke koje želi te odabire opciju „Gotovo“.
4. Podaci o zaposleniku su uspješno uređeni.

**Alternativni scenarij:**

1. Korisnik odabire opciju „Uredi“ pokraj željenog zaposlenika.
2. Otvara se stranica s formom za ažuriranje imena, prezimena, OIB-a i adrese, odabir grada i smjene te popisom svih usluga, gdje treba odabrati usluge koje taj zaposlenik zna obavljati.
3. Korisnik je ostavio neko od polja prazno te odabire opciju „Gotovo“.
4. Sustav korisniku javlja poruku da treba unijeti sve potrebne podatke.
5. Kada korisnik unese sve potrebne podatke, scenarij se nastavlja od točke 3) kao uspješni scenarij.

### *Pregled detalja o zaposleniku*

**Primary actor:** Korisnik

**Precondition:** Sustav je pokrenut i prikazan je popis svih zaposlenika.

**Postcondition:** Prikazani su detalji o željenom zaposleniku.

**Glavni uspješni scenarij:**

1. Korisnik odabire opciju „Detalji“ pokraj željenog zaposlenika.
2. Otvara se stranica s prikazom detaljnih podataka o odabranom zaposleniku.

### *Pregled zarade zaposlenika po mjesecima*

**Primary actor:** Korisnik

**Precondition:** Sustav je pokrenut i prikazani su detalji o zaposleniku (2.2.17.).

**Postcondition:** Prikazan je pregled zarade zaposlenika po mjesecima za trenutnu godinu.

**Glavni uspješni scenarij:**

1. Korisnik odabire opciju „Pregled zarade“ na prikazu detalja o željenom zaposleniku.
2. Otvara se stranica s popisom svih mjeseci te ukupnom zaradom, izračunatom prema uslugama koje je taj zaposlenik odradio u mjesecu u trenutnoj godini.

# **Opis objektnog modela**

Kao što je opisano u uvodu, osnovni modeli potrebni za realizaciju aplikacije su sljedeći:

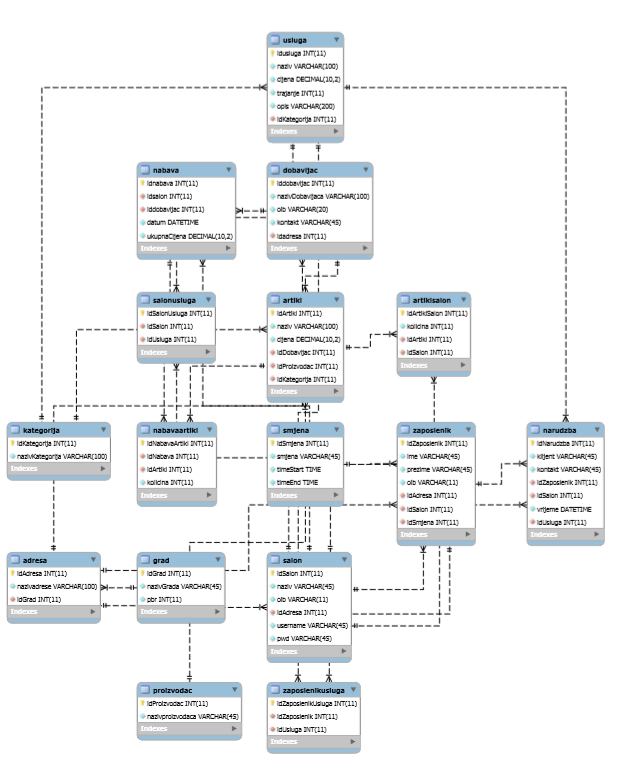
* Adresa
* Artikl
* Dobavljač
* Grad
* Kategorija
* Nabava
* Narudžba
* Proizvođač
* Salon
* Smjena
* Usluga
* Zaposlenik

Osim navedenih osnovnih modela, potrebni su i modeli koji predstavljaju N-N veze među osnovnim modelima. Konkretno, to su veze između modela:

* Artikl i Salon
* Artikl i Nabava
* Zaposlenik i Usluga

## ***ER dijagram***

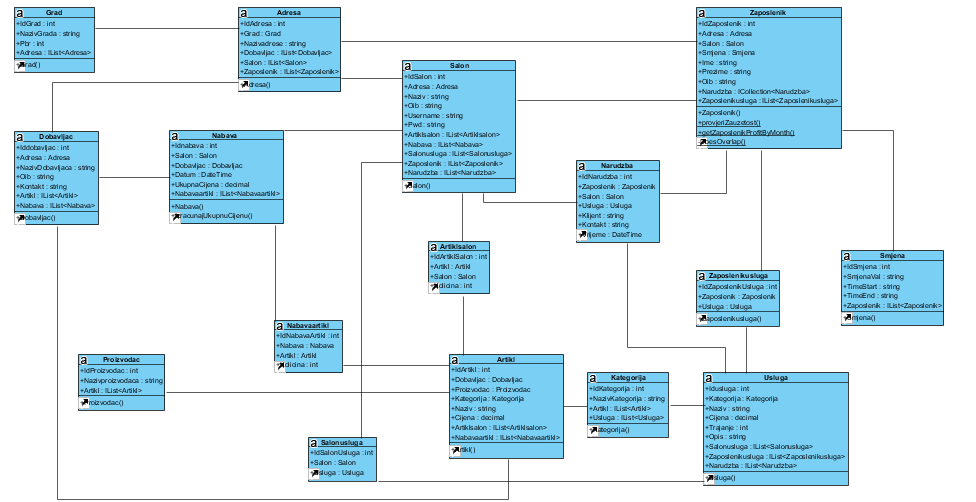
Upravo navedeni modeli i veze predstavljaju sve entitete korištene baze podataka. Ti entiteti te njihove međusobne veze prikazani su ER dijagramom na slici 3.1.



**Slika 3.1.** ER dijagram

## ***Dijagram razreda***

Na temelju navedenih entiteta u bazi podataka prikazanih ER dijagramom na slici 3.1., mapiranjem su stvoreni razredi koji predstavljaju model domene sustava. Slika 3.2. prikazuje dijagram razreda koji predstavlja korišteni model domene.



**Slika 3.2.** Dijagram razreda

## ***Opis modela***

Razred ***Salon***sadrži podatke o kozmetičkom salonu koji koristi aplikaciju. Svaki salon ima atribute *IdSalon*, *Adresa*, *Naziv*, *Oib*, te atribute *Username* i *Pwd* koje koristi za prijavu putem web aplikacije. Razred Salon sadrži i listu svih svojih nabavi artikala, listu svih zaposlenika salona te listu svih narudžbi klijenata u tom salonu. Također, sadrži i listu objekata tipa *Artiklsalon* te listu objekata tipa *Salonusluga*, koje predstavljaju N-N veze između razreda *Salon* i razreda *Artikl*, odnosno *Usluga*. Ovi su razredi opisani kasnije.

Razred ***Adresa***povezan je s razredom ***Grad***, te se koristi za realizaciju podatka o adresama salona i zaposlenika. Atributi rezreda *Adresa* su *IdAdresa*, *Grad* (realiziran razredom *Grad*) i *Naziv*, a atributi razreda *Grad* su *IdGrad*, *Naziv* i *Pbr* (poštanski broj) te lista svih adresa koje se nalaze u tom gradu. Osim ovih atributa, razred Adresa sadrži listu svih dobavljaču, listu svih salona te listu svih zaposlenika koji imaju tu adresu.

Svaki kozmetički salon može pružati različite usluge. Svaka usluga pripada različitoj kategoriji, ima određenu cijenu i trajanje u minutama. Razred ***Usluga*** realizira uslugu i sadrži atribute *Idusluga*, *Kategorija* (realizirana razredom *Kategorija*), *Naziv*, *Cijena*, *Trajanje* i *Opis*. Osim toga, razred Usluga sadrži i listu narudžbi klijenata koje su sadržavale tu uslugu, listu objekata tipa *Salonusluga* te listu objekata tipa *Zaposlenikusluga*, koje predstavljaju N-N veze s razredima *Salon* i *Zaposlenik*. Ovi su razredi opisani kasnije.

Povezanost salona s uslugom realizirana je razredom ***Salonusluga***. Objekti tog razreda govore koji salon pruža koje usluge, a atributi su im *IdSalonusluga*, *Salon* (realiziran razredom *Salon*) te *Usluga* (realizirana razredom *Usluga*).

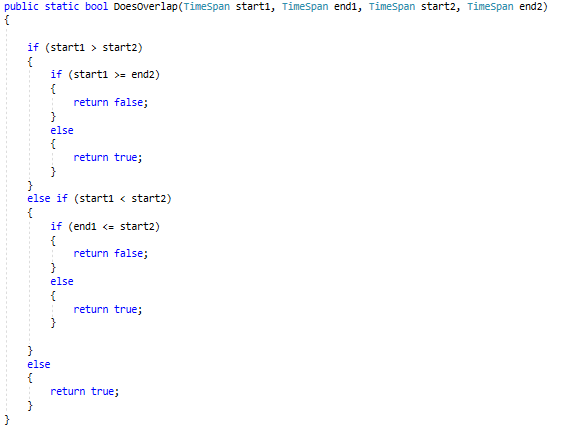
Svaki kozmetički salon ima svoje zaposlenike, koji imaju sposobnost obavljanja pojedinih usluga. Razred ***Zaposlenik***sadrži atribute *IdZaposlenika*, *Ime*, *Prezime*, *Oib*, *Adresa* (realizirana razredom *Adresa*), *Salon* (realiziran razredom *Salon*) te *Smjena* (realizirana razredom *Smjena*). Atribut *Salon* određuje salon u kojem zaposlenik radi, a atribut *Smjena* određuje vrijeme u kojem zaposlenik radi u salonu. Dodatno, svaki zaposlenik sadrži i listu narudžbi klijenata za koje je on zadužen te listu objekata tipa *Zaposlenikusluga*, koji predstavlja N-N vezu između razreda *Zaposlenik* i *Usluga* te koji je opisan kasnije. Razred *Zaposlenik* sadrži i metode:

* ProvjeriZauzetost (IList<Narudzba> svenarudzbeZaposlenika, TimeSpan vrijemeNarudzbe, int trajanjeUsluge): Boolean
* DoesOverlap (TimeSpan start1, TimeSpan end1, TimeSpan start2, TimeSpan end2): bool
* GetZaposlenikProfitByMonth (ICollection<Narudzba> nar, int mjesec, int godina): decimal

Metoda ProvjeriZauzetost(), prikazana na slici 3.3., služi za provjeru je li zaposlenik slobodan ili zauzet za preuzimanje neke narudžbe klijenta. Metoda kao ulazne argumente prima listu svih svojih već naručenih narudžbi klijenata, odabrano vrijeme za novu narudžbu te trajanje odabrane usluge, a vraća Boolean vrijednost true ako je zaposlenik slobodan za preuzeti tu narudžbu, a u suprotnom false. Ova metoda koristi metodu DoesOverlap(), prikazanu na slici 3.4., koja kao ulazne argumente prima dva vremenska intervala te vraća bool vrijednost true ako se oni preklapaju, a u suprotnom vraća false.

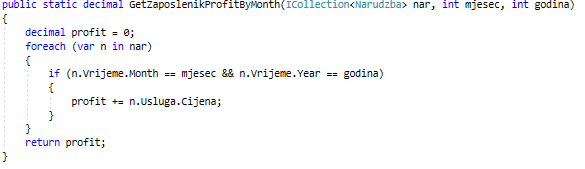


**Slika 3.3.** Metoda ProvjeriZauzetost()



**Slika 3.4.** Metoda DoesOverlap()

Metoda GetZaposlenikProfitByMonth(), prikazana na slici 3.5., računa zaradu zaposlenika za zadani mjesec u zadanoj godini, koji se, uz listu svih narudžbi tog zaposlenika, predaju kao ulazni argumenti metodi. Metoda vraća decimalni broj koji predstavlja ukupnu zaradu tog zaposlenika u zadanom mjesecu zadane godine.



**Slika 3.5.** Metoda GetZaposlenikProfitByMonth()

Smjene zaposlenika su realizirane razredom ***Smjena***i sadrže atribute *IdSmjena*, *SmjenaVal* (jutarnja ili popodnevna), *TimeStart* kao početak radnog vremena, i *TimeEnd* kao kraj radnog vremena zaposlenika. Svaki objekt ovog razreda sadrži i listu zaposlenika koji rade u toj smjeni.

Usluge koje pruža zaposlenik realizirane su razredom ***Zaposlenikusluga***, koji sadrži atribute *IdZaposlenikusluga*, *Zaposlenik* (realiziran razredom *Zaposlenik*) te *Usluga* (realizirana razredom *Usluga*).

Kreiranje narudžbi realizirano je razredom ***Narudzba***. Narudžba sadrži atribute *IdNarudzbe*, *Zaposlenik* (realiziran razredom *Zaposlenik*) koji je zadužen za odrađivanje narudžbe, *Usluga* (realizirana razredom *Usluga*) koja se traži u narudžbi, *Salon* (realiziran razredom *Salon*) u kojem se traži narudžba, atribute *Klijent* i *Kontakt* koji nude podatke o klijentu te *Vrijeme* u kojem će se izvršiti narudžba.

Artikli su opisani razredom ***Artikl***. Atributi artikala su *IdArtikl*, *Dobavljac* (realiziran razredom *Dobavljac*), *Proizvodac* (realiziran razredom *Proizvodac*), *Kategorija* (realizirana razredom *Kategorija*), *Naziv* i *Cijena*. Dodatno, razred Artikl sadrži listu objekata tipa *Artiklsalon* te listu objekata tipa *Nabavaartikl*, koje predstavljaju N-N veze s razredima *Salon* i *Nabava*.

Dobavljači artikala realizirani su razredom ***Dobavljac****,* koji sadrži atribute *Iddobavljac*, *Adresa* (realizirana razredom *Adresa*), *Naziv*, *Oib* i *Kontakt*. Također, razred *Dobavljac* sadrži listu svih artikala čiji je on dobavljač te listu svih nabavi u kojima je sudjelovao. Svaki artikl je povezan sa određenim dobavljačem tog artikla. Razred ***Proizvodac***koristi atribute *IdProizvodac* i *NazivProizvodac*, a također sadrži i listu svih artikala čiji je on proizvođač.

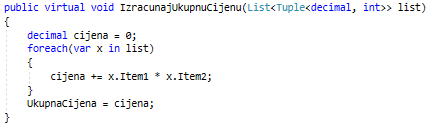
Artikli i usluge pripadaju i različitim kategorijama, koje su opisane razredom ***Kategorija***. Svaka kategorija određena je atributima *NazivKategorija* i *IdKategorija*. Ovaj razred također sadrži listu svih artikala i listu svih usluga koje se nalaze u toj kategoriji.

Artikli i Salon povezani su razredom ***Artiklsalon****,* koji određuje količinu pojedinog artikla u određenom kozmetičkom salonu. Ovaj razred sadrži atribute *IdArtiklsalon*, *Artikl* (realiziran razredom *Artikl*), *Salon* (realiziran razredom *Salon*) te *Količina*.

Kozmetički salon ima mogućnost kreiranja nabave artikala za određene dobavljače. Razred ***Nabava***sadrži atribut *IdNabava*, atribut *Salon* (realiziran razredom *Salon*) koji definira za koji salon je definirana nabava, *Dobavljac* (realiziran razredom *Dobavljac*) od kojeg se traži nabava, *Datum* kada je nabava realizirana, te atribut *UkupnaCijena* nabave za tog dobavljača. Dodatno, ovaj razred sadrži i listu objekata tipa *Nabavaartikl*, koji predstavlja N-N vezu između razreda *Nabava* i *Artikl,* kao što je kasnije i opisano. Razred *Nabava* sadrži i metodu

IzracunajUkupnuCijenu (List<Tuple<decimal, int>> list): void.

Ova metoda služi za izračun ukupne cijene prilikom stvaranja nove nabave. Metoda kao ulazni argument prima listu parova cijene, tipa decimal, i količine, tipa int, za svaki naručeni artikl od istog dobavljača. Metoda ništa ne vraća, već postavlja atribut *UkupnaCijena* razreda *Nabava* na izračunatu ukupnu cijenu, kao što je prikazano na slici 3.6.



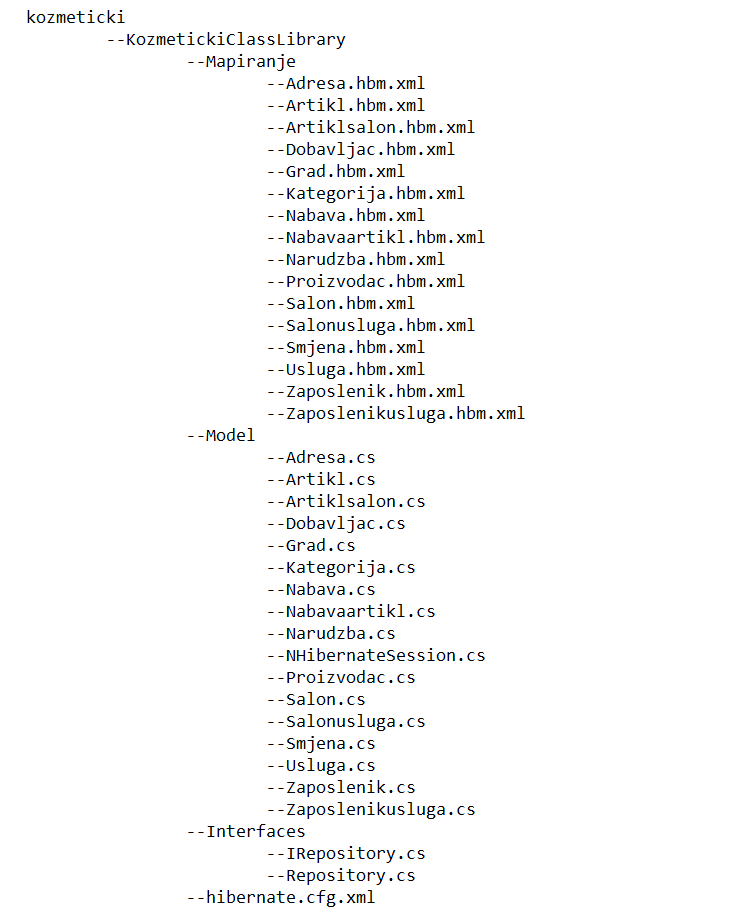
**Slika 3.6.** Metoda IzracunajUkupnuCijenu()

Konačno, razred ***Nabavaartikl***povezuje artikl koji je naručen u nabavi, zajedno s količinom artikla koja se naručuje, pa koristi atribute *Nabava* (realizirana razredom *Nabava*), *Artikl* (realiziran razredom *Artikl*) i *Količina*.

# **Opis implementacije perzistencije**

Kompletna aplikacija realizirana je kroz četiri zasebna projekta, a to su projekti *KozmetickiClassLibrary,* koji sadrži model domene i vezu na bazu podataka, *KozmetickiSalonDesktop,* pomoću kojeg je realizirana desktop aplikacija, *KozmetickiSalonWeb,* kojim je realizirana web aplikacija, te *UnitTest,* koji sadrži sve potrebne Unit testove vezane za aplikaciju.

Projekt *KozmetickiClassLibrary,* sadrži modele, njihove datoteke za mapiranje i sučelje *IRepository,* kao što je prikazano na slici 4.1. Osim navedenog, u ovom je projektu sadržan i direktorij *ViewModels*, koji sadrži modele za prikaz, a koji će biti opisani u poglavlju 6.



**Slika 4.1.** Organizacija direktorija u projektu *KozmetickiClassLibrary*

Unutar direktorija *Model* smješteni su svi korišteni modeli. Svaki model sadrži vlastite atribute i metode koje ga određuju. Njegova pripadajuća datoteka za mapiranje nalazi se unutar direktorija *Mapiranje*. U njoj je prikazan odnos između pripadajuće tablice („zaposlenik“) u bazi podataka i modela (*Zaposlenik*). *Id* je postavljen na automatsko povećavanje pomoću oznake <generator class=“increment“>. Svi vlastiti atributi definiraju se na sljedeći način:

<property name="NazivAtributaUModelu">

<column name="NazivRetkaUTablici" sql-type="TipPodatka"

not-null="true/false" />

</property>

Veza *many-to-one* definira se u datoteci za mapiranje unutar oznake <many-to-one name=““> gdje *name* označava naziv atributa koji je tipa pripadajućeg objekta u modelu za kojeg je definirana veza na strani ključ. Na primjer, u bazi podataka tablica „zaposlenik“ u sebi sadrži strani ključ *idAdresa* koji se referencira na tablicu „adresa“. U modelu je ta veza prikazana na način da model *Zaposlenik* sadrži objekt tipa *Adresa*. Nadalje, veza *one-to-many* definira se unutar oznake <bag name=““>. U modelu se ta veza očituje u stvaranju kolekcije objekata tog tipa. Na primjer, u modelu *Zaposlenik* veza *one-to-many* s modelom *Narudzba* prikazuje se stvaranjem kolekcije objekata tipa *Narudzba*: ICollection<Narudzba> Narudzba. U modelu *Narudzba* ta je veza prikazana kao *many-to-one.* Na analogni način kao i *Adresa* u modelu *Zaposlenik*, u modelu *Narudzba* stvara se objekt tipa *Zaposlenik*. Slika 4.2. prikazuje kako je implementiran model *Zaposlenik.cs*, a slika 4.3. prikazuje datoteku za mapiranje tog modela, *Zaposlenik.hbm.xml*.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

**Slika 4.2.** Model *Zaposlenik.cs*

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

**Slika 4.3.** Datoteka za mapiranje *Zaposlenik.hbm.xml*

*Unit of Work* realiziran je unutar klase *Repository,* koja implementira sučelje *IRepository*. Unutar sučelja *IRepository* definirane su funkcije za dohvaćanje, spremanje, brisanje i ažuriranje podataka. Unutar konstruktora *ZaposleniksController-*a stvara se objekt *Repository* tipa *IRepository*:

private IRepository zaposlenikRepository;

public ZaposleniksController() {

this.zaposlenikRepository = new Repository();

}

Prilikom pozivanja konstruktora za stvaranje objekta *Repository* otvara se nova sesija za rad s podacima iz baze podataka. Za spremanje promjena u sesiji nad njom se poziva funkcija Save() ili za brisanje Delete(). Kako bi se promjene koje su odrađene nad podacima pohranile u bazu podataka, poziva se funkcija Commit() nad objektom tipa *ITransaction*. Slika 4.4. prikazuje kako je implementiran razred *Repository.cs*.

Slika na kojoj se prikazuje snimka zaslona

Opis je automatski generiran

**Slika 4.4.** Razred *Repository.cs*

# **Opis izgrađene desktop aplikacije**

Desktop aplikacija izgrađena je kao Visual Studio C# Windows Forms aplikacija. Ona koristi referencu na Class Library projekt, koji sadržava model domene. Preko dobivene reference aplikacija pristupa potrebnim modelima. Forme pristupaju podacima iz baze podataka stvarajući objekt *Repository* tipa *IRepository* koji otvara novu sesiju, kao što je opisano u poglavlju 4.

Aplikacija se sastoji od glavne forme *PocetnaForm.cs* i dodatnih formi koje su povezane sa glavnom. Dodatnim formama pristupa se preko različitih gumba u glavnoj formi.

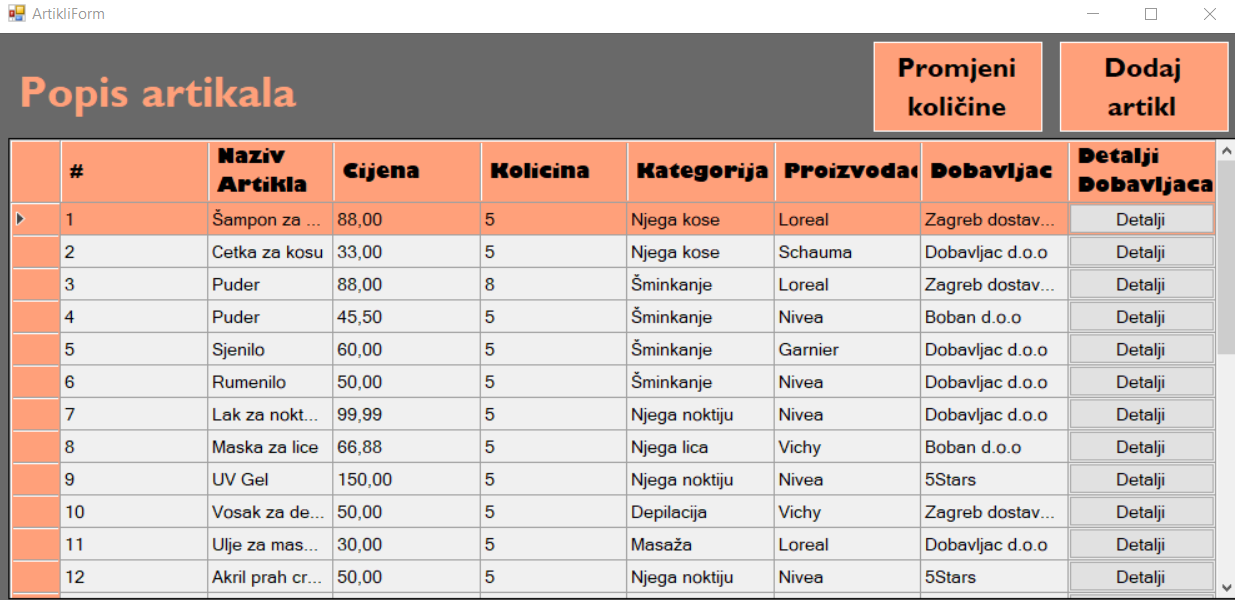
Kreiranje svake forme obavlja se na jednak način: unutar prozora „Solution Explorer“ gdje je prikazan sadržaj trenutno otvorenog projekta potrebno je desnim klikom kliknuti na ime projekta → „Add“ → „Windows form“. U donjem dijelu novootvorenog prozora potrebno je upisat željeno ime forme i pritisnuti „OK“. Povezivanje svih formi obavlja se unutar koda glavne forme. Prvo je potrebno deklarirati nove forme te zatim definirati kada će se pojedina forma definirati i otvoriti. Pošto je za pristup ostalim formama korišten izbornik, dvostrukim klikom na odgovarajuću stavku unutar njega otvara se generirani kod unutar kojeg se definira koja će se forma otvoriti. Slike u nastavku prikazuju forme koje realiziraju neke od glavnih obrazaca uporabe.

Konkretno, slika 5.1. prikazuje početni ekran desktop aplikacije realiziran formom *PocetnaForm.cs*. Početni ekran sadrži naziv salona te popis svih narudžbi klijenata za trenutni datum. Također, na lijevoj strani ekrana nalazi se navigacijska traka, preko koje je moguće pristupiti bilo kojoj od opcija koje nudi aplikacija.



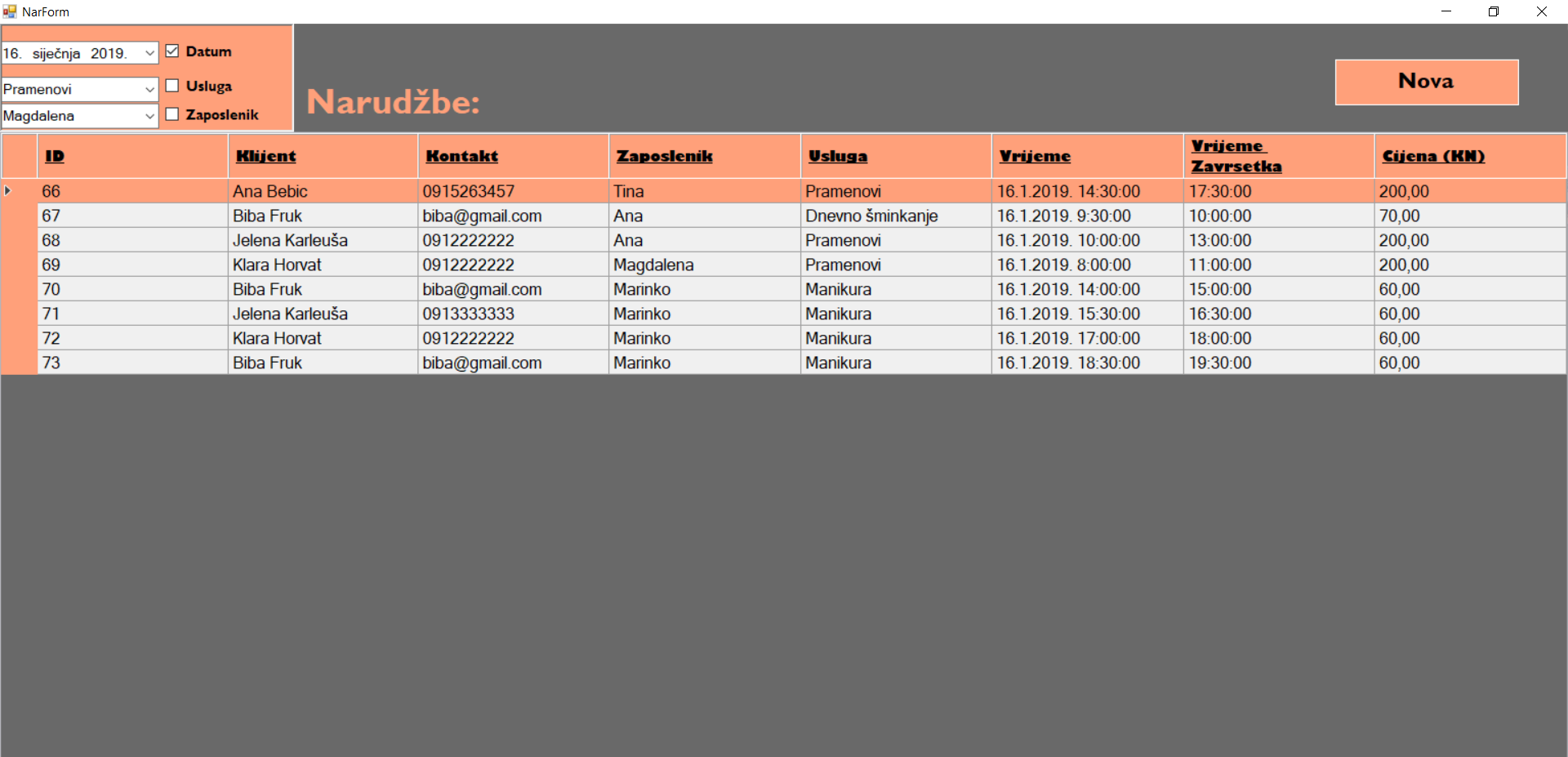
**Slika 5.1.** Početni ekran desktop aplikacije

Nadalje, slika 5.2. prikazuje popis svih artikala na skladištu kozmetičkog salona, realiziran formom *ArtikliForm.cs*. Za svaki artikl na popisu vidljiv je naziv, cijena, količina, kategorija, proizvođač te dobavljač, a moguće je i pregledati detalje o dobavljaču. Također, artiklima je moguće promijeniti količine na skladištu, klikom na gumb „Promijeni količine“ te je moguće dodati novi artikl na skladište, klikom na gumb „Dodaj artikl“.



**Slika 5.2.** Popis svih artikala na skladištu

Slika 5.3. prikazuje popis svih narudžbi klijenata za određeni datum, realiziran formom *NarForm.cs*. Za svaku narudžbu na popisu vidljivo je ime i kontakt klijenta, zaposlenik, usluga, vrijeme početka i okvirno vrijeme završetka te cijena. Narudžbe se, osim po datumu, mogu filtrirati i po usluzi ili po zaposleniku, što je moguće učiniti u gornjem lijevom kutu ekrana. Također, moguće je unijeti novu narudžbu klijenta, klikom na gumb „Nova“ u gornjem desnom kutu ekrana.



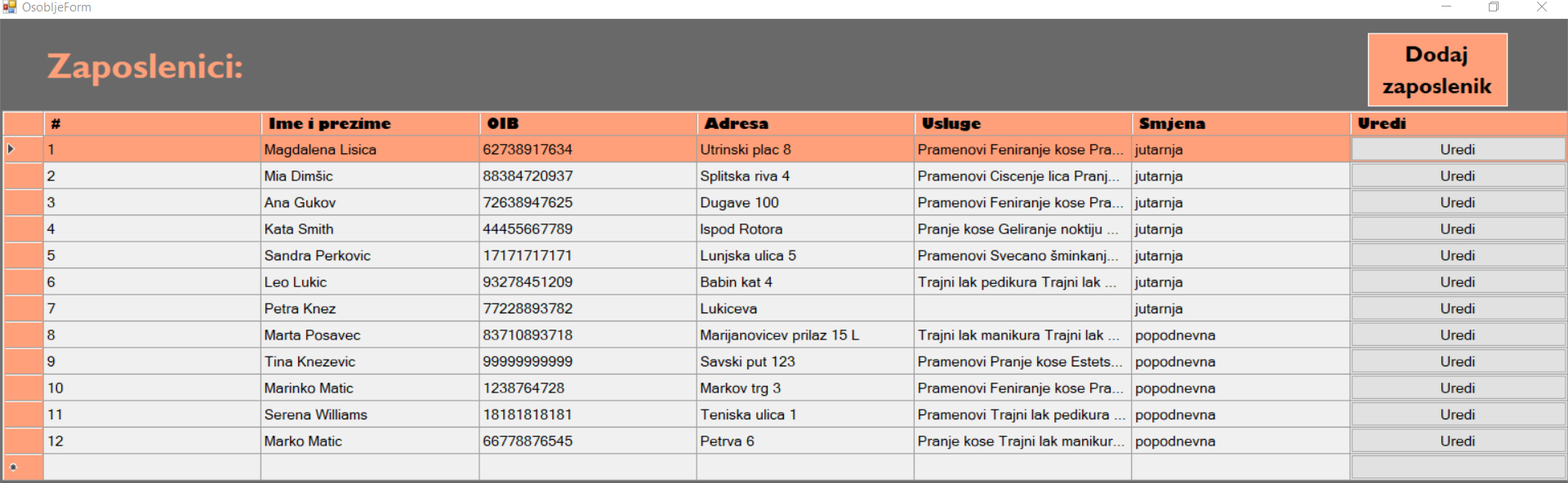
**Slika 5.3.** Popis svih narudžbi klijenata za određeni datum

Slika 5.4. prikazuje formu za nabavu artikala, realiziranu formom *NabaveForm.cs*. Prilikom stvaranja nove nabave artikala, potrebno je na popisu svih artikala pored svakog artikla koji se želi naručiti unijeti željenu količinu tog artikla. Nakon toga, klikom na gumb „Nabavi“ u gornjem desnom kutu ekrana, stvara se onoliko novih nabavi artikala koliko je različitih dobavljača među odabranim artiklima.



**Slika 5.4.** Forma za nabavu artikala

Slika 5.5. prikazuje popis svih zaposlenika kozmetičkog salona, realiziran formom *OsobljeForm.cs*. Za svakog zaposlenika na popisu vidljivo je ime i prezime, OIB, adresa, popis usluga koje on nudi i smjena u kojoj radi. Zaposlenika je moguće urediti klikom na gumb „Uredi“ pored željenog zaposlenika, čime se otvara forma za uređivanje podataka o zaposleniku. Također, moguće je dodati novog zaposlenika, klikom na gumb „Dodaj zaposlenika“ u gornjem desnom kutu ekrana.

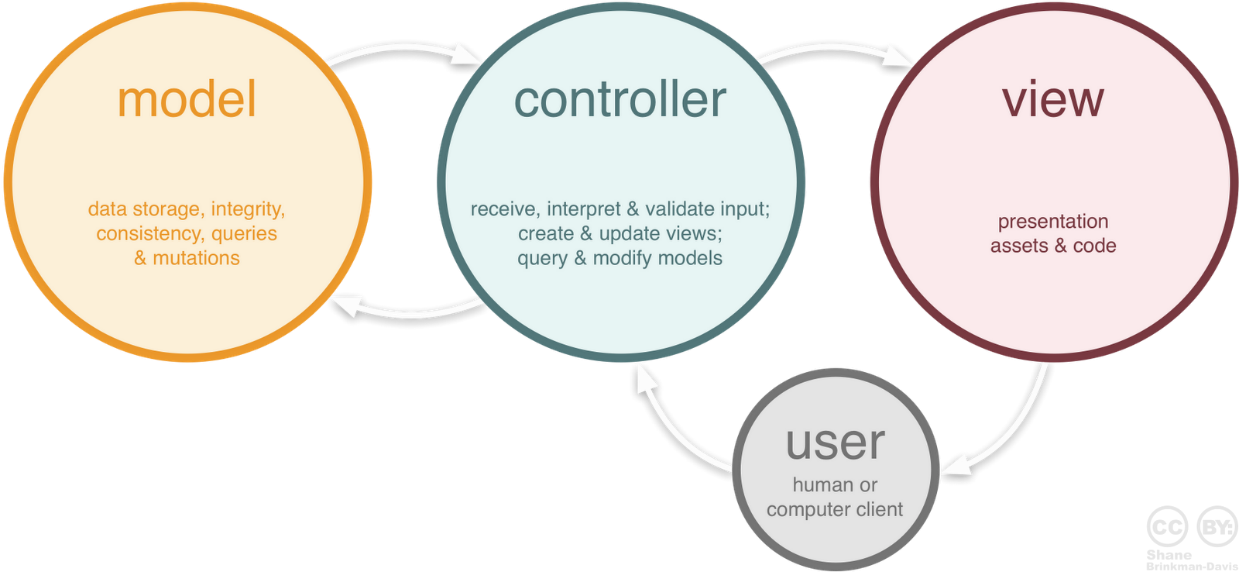


**Slika 5.5.** Popis svih zaposlenika kozmetičkog salona

# **Opis izgrađene web aplikacije**

Web aplikacija izgrađena je kao Visual Studio C# Model-View-Controller Web aplikacija. Ona koristi referencu na Class Library projekt, koji sadržava model domene te preko dobivene reference pristupa potrebnim modelima, pa i bazi podataka. Iako je aplikacija tipa Model-View-Controller, sami projekt zapravo sadrži samo „View-Controller“ dio, odnosno kontrolere i prikaze, dok je „Model“ dio sadržan u opisanom Class Library projektu.

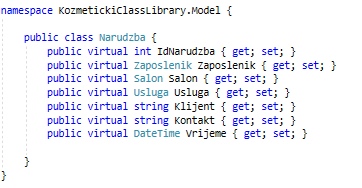
Način na koji funkcionira MVC web aplikacija, odnosno uloge svake komponente te komunikacija među komponentama, prikazan je na slici 6.1.



**Slika 6.1.** Arhitektura MVC web aplikacije

## ***Modeli i modeli za prikaz***

Web aplikacija bazirana na MVC sustavu funkcionira tako da su kroz modele pohranjeni svi podaci, zajedno s integritetskim i konzistencijskim ograničenjima. Dodatno, u svrhu jednostavnijeg baratanja i prikaza podataka, izrađeni su i modeli za prikaz (engl. *View Models*), koji sadrže atribute koji su prilagođeni za prikaz podataka te samo potrebne atribute. Primjer modela i odgovarajućeg modela za prikaz za slučaj narudžbe prikazani su isječcima koda na slikama 6.2., odnosno 6.3. Iz ovih je isječaka koda vidljivo kako je u modelu za prikaz realizirana validacija prilikom unosa nove narudžbe te je definiran način prikaza DateTime atributa. Ovo je realizirano u modelu za prikaz obzirom da se on koristi prilikom unosa nove narudžbe. Također, u modelu za prikaz umjesto cijelih objekata povezanih modela (Zaposlenik, Salon i Usluga), atributi su njihovi ključevi, što uvelike pojednostavljuje rad.



**Slika 6.2.** Model: razred Narudzba



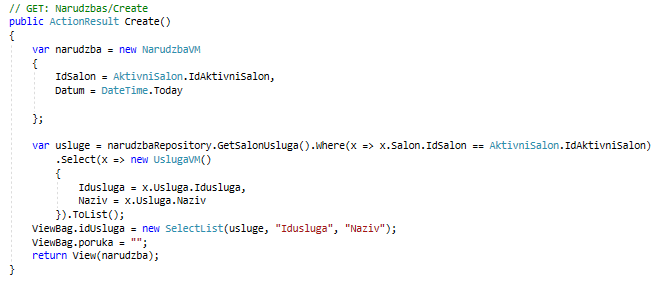
**Slika 6.3.** Model za prikaz: razred NarudzbaVM

Svi modeli u organizaciji naših direktorija nalaze se u direktoriju *KozmetickiClassLibrary/Model*, a svi modeli za prikaz nalaze se u direktoriju *KozmetickiClassLibrary/ViewModels*, koji je dio Class Library projekta.

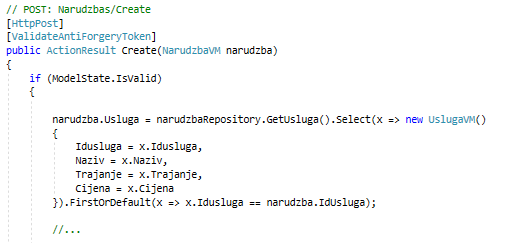
## ***Kontroleri***

Kontroleri (engl. *Controller*), u pravom smislu tog naziva, kontroliraju i modele i prikaze, odnosno omogućuju interakciju korisnika sa sustavom. Dakle, kontroler prima podatke, bilo od modela, bilo od prikaza te nad tim podacima odradi akcije za koje je zadužen. Konkretno, to su uglavnom akcije stvaranja novih podataka te ažuriranja ili brisanja postojećih u slučaju kada je metoda tipa POST u kontroleru, odnosno kada prima podatke od prikaza. S druge strane, metode tipa GET u kontroleru pripremaju podatke u obliku modela za prikaz, pa te podatke prosljeđuje u prikaz. Kontroleri pristupaju podacima iz baze podataka stvarajući objekt *Repository* tipa *IRepository* koji otvara novu sesiju, kao što je opisano u poglavlju 4.

Slike 6.4. i 6.5. prikazuju isječke koda, odnosno metode iz kontrolera *NarudzbasController.cs*. Konkretno, slika 6.4. prikazuje GET metodu za stvaranje nove narudžbe u kontroleru zaduženom za narudžbe, a slika 6.5. POST metodu za stvaranje nove narudžbe u kontroleru zaduženom za narudžbe.



**Slika 6.4.** GET metoda Create()



**Slika 6.5** POST metoda Create()

Kao što je vidljivo na sli 6.4., metoda tipa GET ne prima ulazne parametre, a vraća podataka tipa ActionResult, konkretno, prikaz modela za prikaz tipa NarudzbaVM. Pomoću atributa ViewBag, moguće je prenijeti dodatne podatke osim modela u prikaz.

S druge strane, kao što se vidi iz slike 6.5., metoda tipa POST kao ulazni argument prima objekt tipa NarudzbaVM, o kojem je, kroz prikaz, korisnik unio podatke i ažurira taj objekt, a što se dalje događa u toj metodi za svrhu ovog primjera nije bitno.

Kad se govori o organizaciji direktorija ove aplikacije, svi se kontroleri nalaze u direktoriju *KozmetickiSalonWeb/Controllers*, koji je dio projekta koji predstavlja web aplikaciju, a nazivlje kontrolera je, kao i kod kontrolera za narudžbe, ime modela pa riječ „Controller“.

## ***Prikazi***

Prikazi (engl. *View*) služe, kako im i samo ime govori, za prikaz podataka, koje su u obliku nekog modela za prikaz primili od GET metode iz kontrolera. Također, oni služe i kao forme za unos podataka od strane korisnika, koji se onda prosljeđuju POST metodi u kontroleru za daljnju obradu i pohranu. Isječak koda iz prikaza *Create.cshtml* iz direktorija */Views/Narudzbas* prikazan je na slici 6.6.



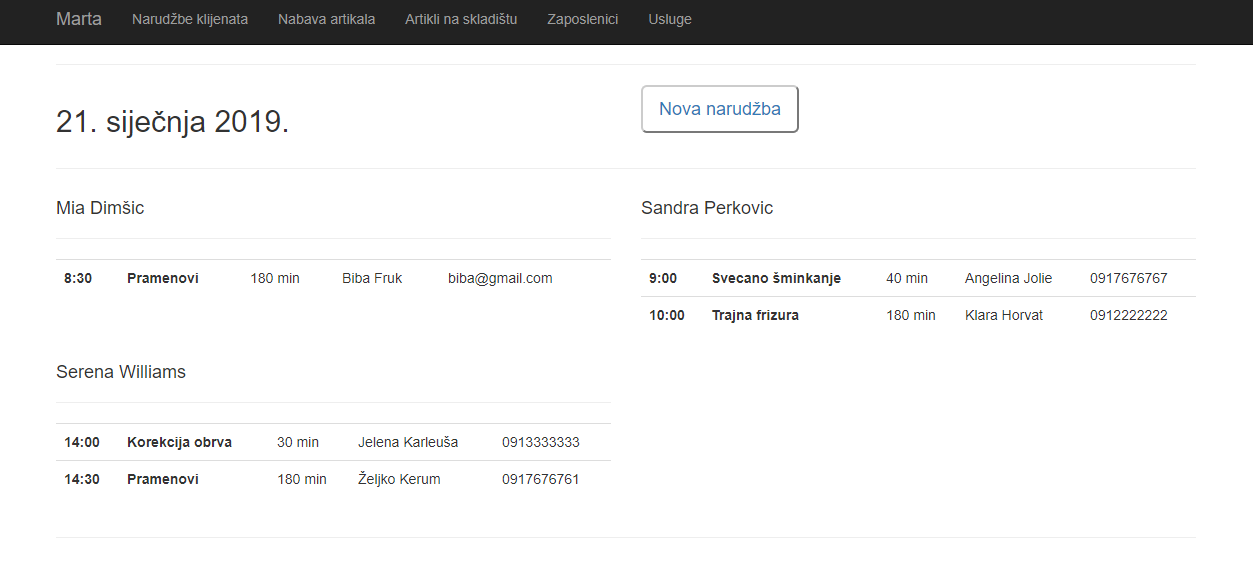
**Slika 6.6.** Prikaz *Create.cshtml*

Kao što je vidljivo u prvom retku koda na slici 6.6., korišteni model za prikaz u ovom je slučaju NarudzbaVM, kao što je i spomenuto u poglavlju 6.2. Također, u dijelu koda na slici za odabir usluge moguće je uočiti korištenje podataka prenesenih pomoću atributa ViewBag, spomenutom u poglavlju 6.2. Prikaz na slici služi za unos podataka od strane korisnika, što znači da treba sadržavati formu, koja započinje kodom @using (Html.BeginForm()). Korištenje validacije, definirane u modelu kao što je opisano u poglavlju 6.1., definira se kodom @Html.ValidationSummary (true). U nastavku koda sa slike, između ostalog postoji i gumb potvrde unosa (engl. *Submit*), kojim se potvrđuje unos podataka te prosljeđuje model u POST metodu kontrolera.

Što se tiče organizacije u direktorije, svi se prikazi nalaze u direktoriju *KozmetickiSalonWeb/Views*, gdje su također raspoređeni u direktorije, koji odgovaraju kontrolerima. Dakle, kao što je navedeno za primjer narudžbe, kontroler *NarudzbasController.cs* se nalazi u direktoriju *KozmetickiSalonWeb/Controllers*, a svi prikazi vezani za narudžbu, odnosno njen kontroler, nalaze se u direktoriju *KozmetickiSalonWeb/Views/Narudzbas*.

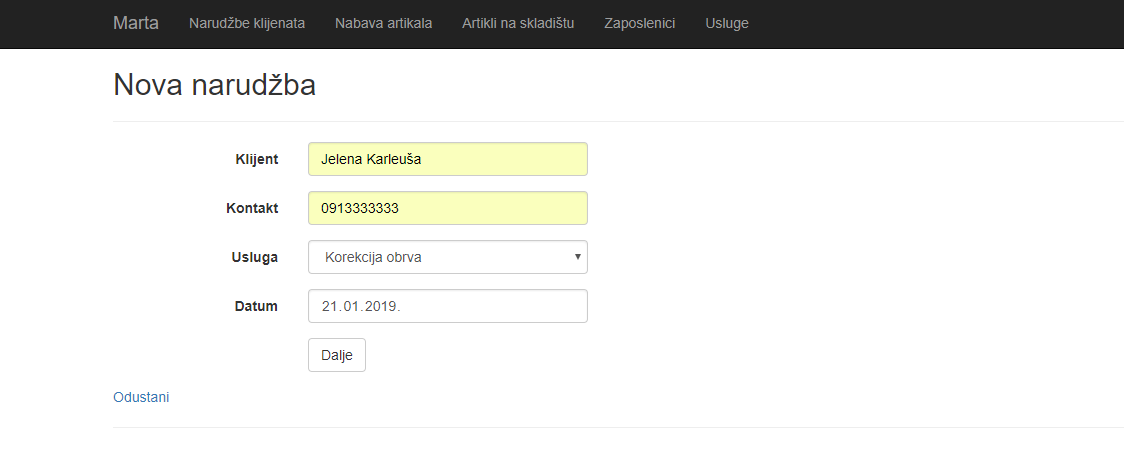
## ***Ekrani web aplikacije***

U ovom će poglavlju biti prikazano nekoliko glavnih ekrana web aplikacije. Prvi od njih je, naravno, početni ekran, na kojem je, osim trenutnog datuma, moguće vidjeti popis svih narudžbi za trenutni datum, raspoređenih po zaposlenicima. Početni ekran web aplikacije prikazan je na slici 6.7.

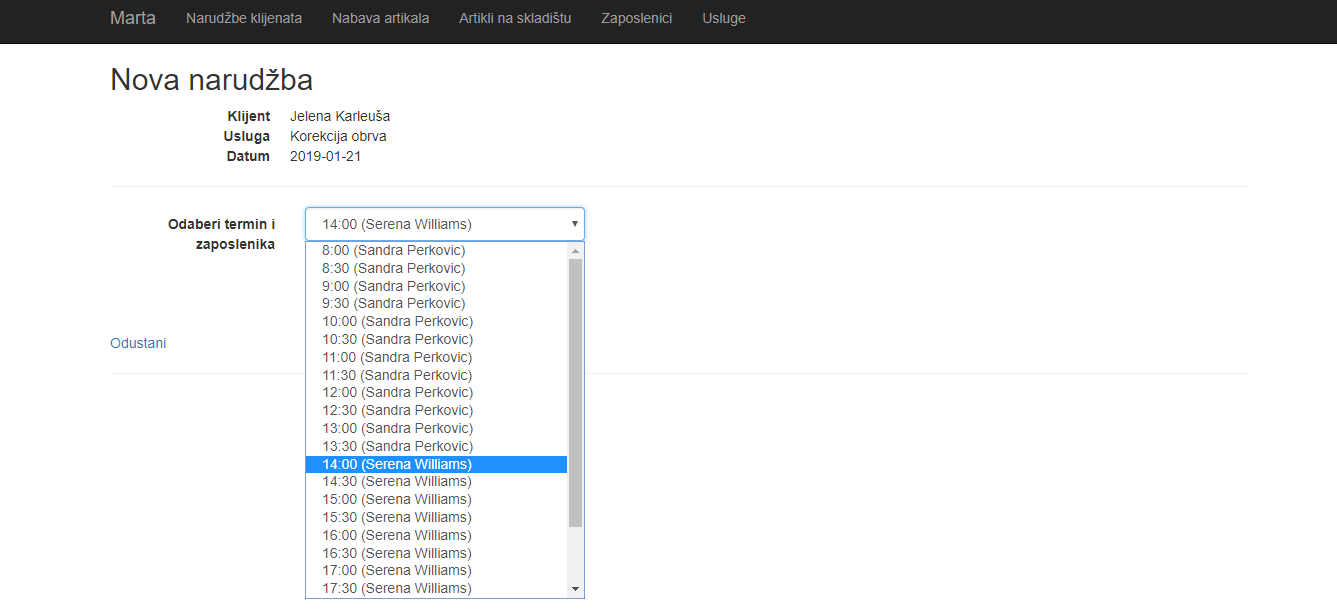


**Slika 6.7.** Početni ekran web aplikacije

Nadalje, jedna od najbitnijih funkcionalnosti sustava jest unos nove narudžbe klijenta. To se provodi kroz dvije forme, tako da se kroz prvu formu unosi ime i kontakt klijenta te se odabire željena usluga i datum. Ova je forma prikazana na slici 6.8. U drugoj se formi, prikazanoj na slici 6.9., nakon što sustav generira sve mogućnosti, odabire termin za narudžbu, zajedno sa zaposlenikom koji će odraditi tu narudžbu.

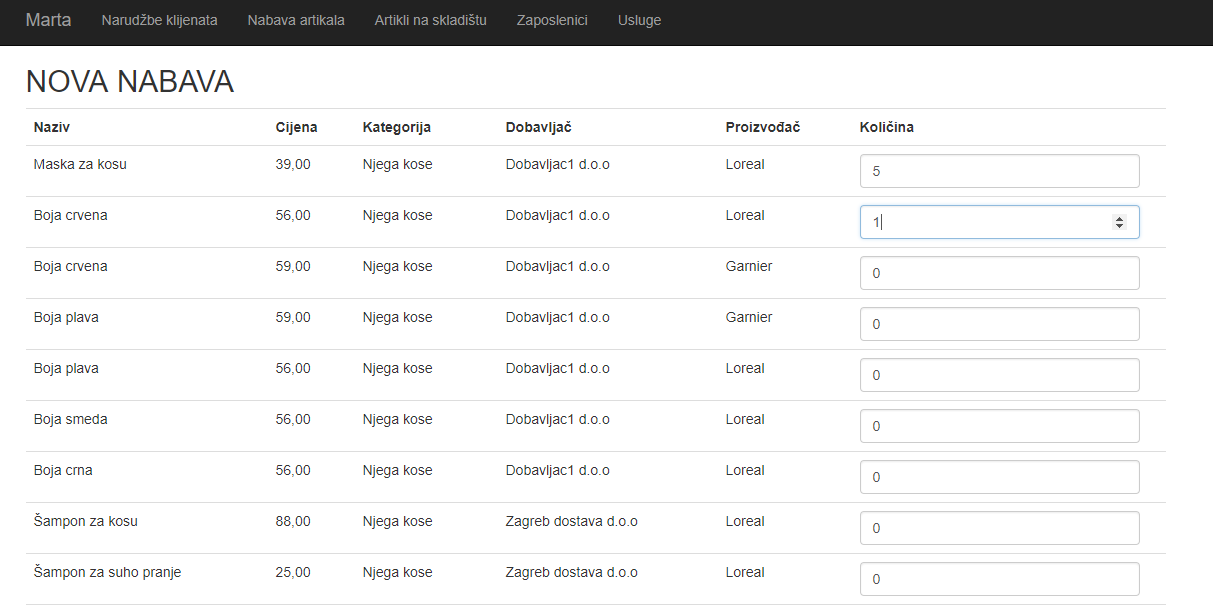


**Slika 6.8.** Prva forma prilikom unosa nove narudžbe



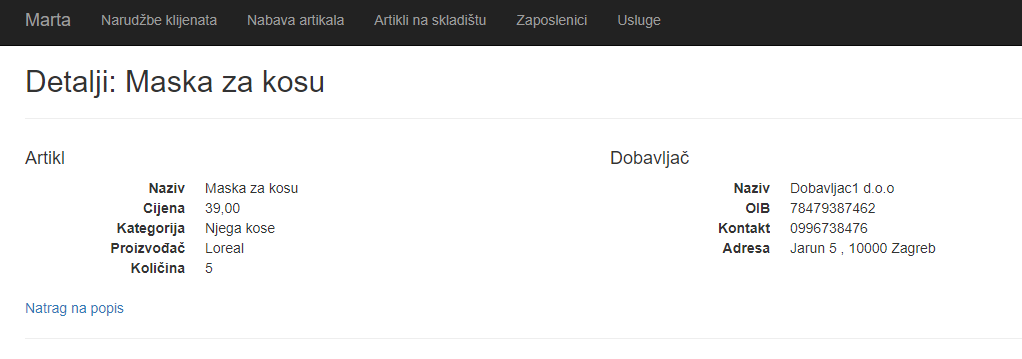
**Slika 6.9.** Druga forma prilikom unosa nove narudžbe

Još jedna od bitnijih funkcionalnosti sustava jest nabava artikala. Ekran za nabavu artikala sadrži popis svih mogućih artikala za nabaviti, gdje je pokraj svakog željenog artikla potrebno unijeti željenu količinu za nabavu. Ova je forma prikazana na slici 6.10.



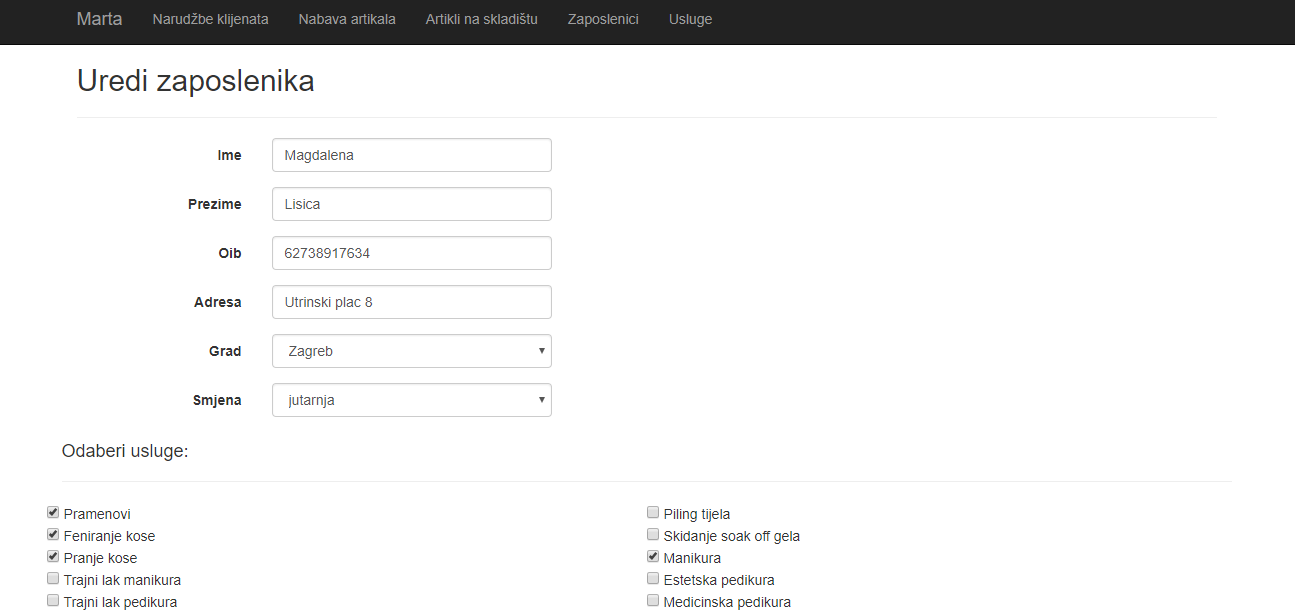
**Slika 6.10.** Forma za nabavu artikala

Za svaki artikl koji salon ima na skladištu, moguće je pregledati detalje. Detalji o artiklu uključuju sve podatke o samom artiklu, što su naziv, cijena, kategorija, proizvođač i količina na skladištu, te sve podatke o dobavljaču tog artikla, što su naziv, OIB, kontakt i adresa. Ovaj je ekran prikazan na slici 6.11.



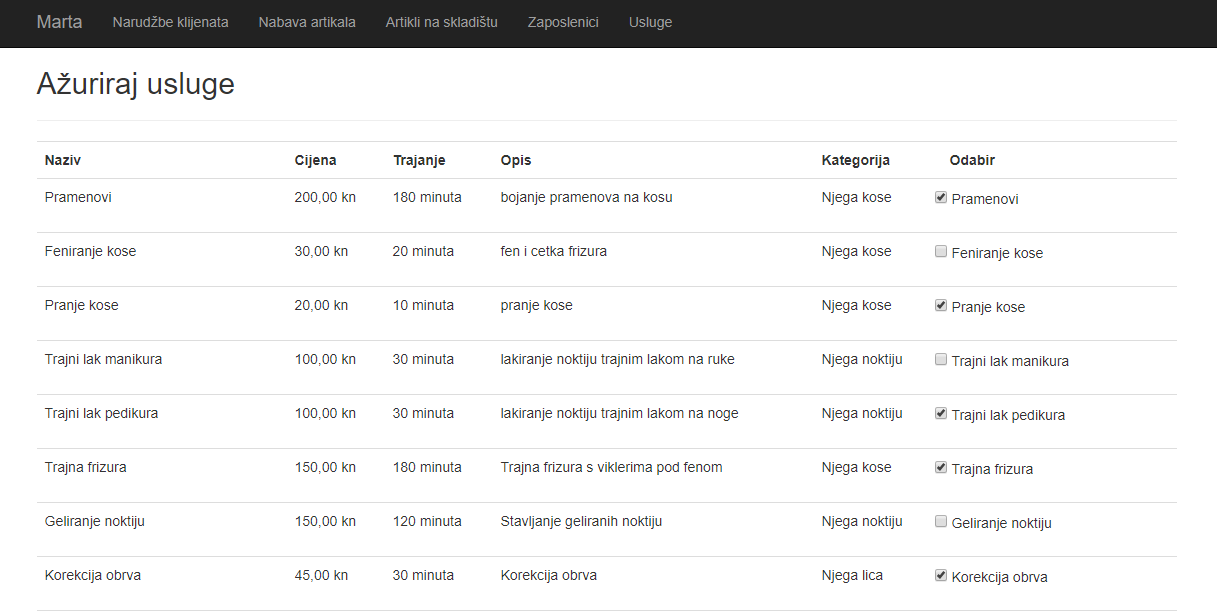
**Slika 6.11.** Ekran s detaljima o artiklu

Nadalje, na slici 6.12. prikazana je forma za uređivanje podataka o zaposleniku salona, a njoj je jednaka i forma za dodavanje novog zaposlenika. Prilikom ažuriranja podataka o zaposleniku, moguće je uređivati ime, prezime, OIB, adresu te promijeniti odabir grada i smjene. Također, na dnu forme nalazi se popis svih mogućih usluga, gdje svaka od njih pored sebe ima *check-box*, pomoću kojeg se može odabrati zna li zaposlenik odrađivati tu uslugu. Ovdje se prilikom uređivanja zaposlenika usluge mogu označiti ili odznačiti.



**Slika 6.12.** Forma za uređivanje podataka o zaposleniku

Konačno, zadnji ekran koji je potrebno demonstrirati jest ekran prilikom ažuriranja usluga koje nudi kozmetički salon. Ovaj ekran sadrži popis svih mogućih usluga, zajedno s detaljima o svakoj usluzi te se pokraj svake usluge nalazi check-box, pomoću kojeg je moguće dodati ili maknuti uslugu s popisa usluga koje taj salon nudi. Ova je forma prikazana slikom 6.13.



**Slika 6.13.** Forma za ažuriranje usluga koje salon nudi