

UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA

Seminarski rad iz predmeta

Izabrana poglavlja iz informacionih sistema

Tema: Aplikacija za evidenciju pružanja telekomunikacionih
usluga

Student:

Sergej Aleksandar Gojković 3069/2023

Beograd 2024.

1. Uvod.....	3
2. Prikupljanje korisničkih zahteva	4
2.1. Verbalni opis	4
2.2. Slučajevi korišćenja	5
3. Analiza i projektovanje	10
3.1. Dijagram sekvenci	10
3.2. Proširen model objekti-veze	14
3.3. Relacioni model.....	14
4. Zaključak	15

1. Uvod

Cilj ovog seminarskog rada je upoznavanje sa relativno novom tehnologijom Blazor, koja je zvanično postala dostupna 2018. godine. Blazor je proizvod kompanije Microsoft koji pruža brojne mogućnosti, među kojima su razvoj klijentske, ali i serverske web aplikacije, kao i njihovo automatsko povezivanje.

Stranice i delovi stranica Blazor-a, kao i kod Angular-a i React-a, funkcionišu na nivou komponenti. Svaka komponenta se može iskoristiti u okviru neke od stranica, ali može biti i sama stranica, ukoliko se navede njena putanja.

Jedna od prednosti ove tehnologije u odnosu na slične proizvode, kao što su Angular i React, leži u tome što celokupan projekat može da se razvija u .Net tehnologiji, bez primene JavaScript jezika, ali uz ostavljanje mogućnosti da se koristi, ukoliko je to potrebno. Ovakav način razvoja, programerima omogućava da se oslone na jedan programski jezik koji im je poznat i fokus stave isključivo na njegovo unapređivanje.

Ovaj moćni alat, pored pomenutog, pruža jednostavno povezivanje klijentske i serverske aplikacije sa drugim bibliotekama, definisanjem servisa koji će se koristiti na samom serveru ili klijentu. Dobro poznat koncept tagova, omogućava lakši i brži razvoj, jednostavno održavanje, kao i bolju čitljivost i strukturu koda. Mogućnost interakcije korisnika sa aplikacijom i rad u realnom vremenu, još jedna je u nizu prednosti koju ovaj frejmwork nudi.

Treba obratiti pažnju da je kod na klijentskoj aplikaciji dostupan svakome ko pristupi stranici, dok kod servera to nije slučaj. Ovakav princip je zastupljen i kod pomenutih alternativnih tehnologija, pa je poznata preporuka da se sama logika smesti na server.

U nastavku je opisano prikupljanje korisničkih zahteva gde se nalazi definisan verbalni opis, kao i neki od slučajeva korišćenja prepoznatih u ovom projektu. U trećem poglavlju se nalaze analiza i projektovanje, odnosno fizička i logička struktura softvera definisana preko dijagrama sekvenci, proširenog modela objekti-veze i relacionog modela. Na samom kraju se nalazi zaključak u kojem se nalaze utisci stečeni u radu na ovom projektu.

2. Prikupljanje korisničkih zahteva

U ovom poglavlju je opisana struktura i ponašanje sistema kroz verbalni opis i slučajeve korišćenja, sa fokusom na funkcionalne zahteve.

2.1. Verbalni opis

Kompanija korisnicima pruža telekomunikacione usluge koje uključuju kablovsku televiziju, internet i fiksnu telefoniju.

Kada korisnik dođe u poslovnici, sa namerom da se pretplati na neku od usluga, zaposleni prikuplja podatke o korisniku i unosi ih u aplikaciju. Nakon što se korisnik izjasni o paketu i uslugama koje želi da koristi, zaposleni u aplikaciju unosi novi ugovor sa podacima o izabranom paketu, uslugama i korisniku, posle čega korisnik potpisuje ugovor. Prilikom potpisivanja ugovora, korisnik mora odabrati jedan paket, a pored toga može izabrati i neke od dodatnih usluga.

Ukoliko se ukaže potreba, zaposleni može dodati novi paket ili uslugu. Svaki paket se sastoji od jedne ili više usluga, i svaka usluga u okviru paketa ima odgovarajući popust na cenu usluge. Svaka usluga ima odgovarajuću cenu, pri čemu će period važenja cene biti neograničen kada je u pitanju nova cena usluge. Kada dođe do promene cene usluge, usluga se ažurira tako što se dodaje nova cena koja će imati neograničen period važenja, dok će stara cena dobiti svoj kranji datum.

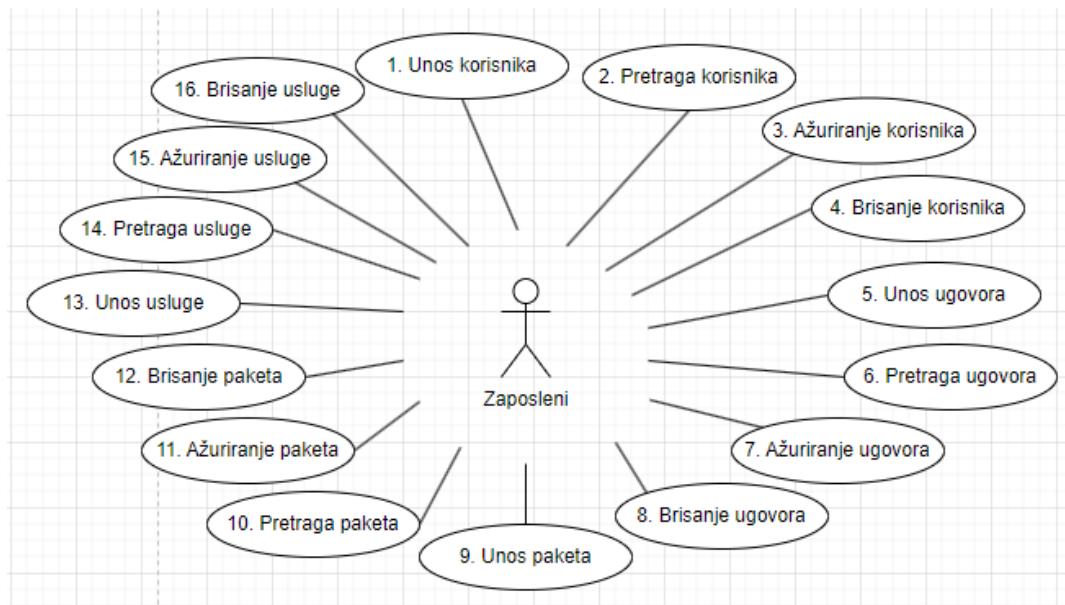
Aplikacija treba da omogući unos, pregled, ažuriranje i brisanje korisnika, ugovora, paketa i usluga.

2.2. Slučajevi korišćenja

U nastavku se nalaze funkcionalni zahtevi predstavljeni preko modela slučaja korišćenja, koji se sastoji od skupa slučaja korišćenja, aktora i njihovih međusovnih veza.

Postoje sledeći slučajevi korišćenja:

1. Unos korisnika (prost slučaj)
2. Pretraga korisnika
3. Ažuriranje korisnika
4. Brisanje korisnika
5. Unos ugovora
6. Pretraga ugovora
7. Ažuriranje ugovora
8. Brisanje ugovora
9. Unos paketa
10. Pretraga paketa
11. Ažuriranje paketa
12. Brisanje paketa
13. Unos usluge
14. Pretraga usluge
15. Ažuriranje usluge
16. Brisanje usluge



Slika 1: Slučajevi korišćenja

SK1: Slučaj korišćenja - Unos korisnika

Naziv SK

Unos korisnika

Aktori SK

Zaposleni

Učesnici SK

Zaposleni, sistem

Preduslov

/

Osnovni scenario SK

1. Zaposleni pokreće formu za unos korisnika.
2. Sistem istancira objekte.
3. Zaposleni unosi podatke o korisniku u polja ime, prezime, broj telefona, JMBG i broj lične karte.
4. Zaposleni poziva sistem da zapamti novog korisnika pritiskom na dugme sačuvaj.
5. Sistem prihvata poziv i pamti novog korisnika.
6. Sistem prikazuje poruku “Uspešno unet korisnik”.

Alternativna scenarija

6.1. Ukoliko sistem ne upiše podatke, sistem prikazuje poruku “Korisnik nije unet”.

SK2: Slučaj korišćenja - Ažuriranje korisnika**Naziv SK**

Ažuriranje korisnika

Aktori SK

Zaposleni

Učesnici SK

Zaposleni, sistem

Preduslov

/

Osnovni scenario SK

1. Zaposleni pokreće formu za pretragu korisnika.
2. Sistem istancira objekte.
3. Zaposleni unosi podatke za pretragu korisnika.
4. Zaposleni poziva sistem da pronađe korisnike pritiskom na dugme pretraži.

5. Sistem prihvata poziv i traži korisnike u bazi.
6. Sistem prikazuje zaposlenom listu korisnika sa porukom “Pronađeni su korisnici”.
7. Zaposleni poziva sistem da učitava podatke o korisniku klikom na dugme izmeni.
8. Sistem prihvata poziv, prikazuje formu za ažuriranje korisnika i podatke o korisniku.
9. Zaposleni menja podatke o korisniku.
10. Zaposleni poziva sistem da sačuva izmene pritiskom na dugme sačuvaj.
11. Sistem prihvata poziv i pamti izmene korisnika.
12. Sistem prikazuje poruku “Uspešno ažuriran korisnik”.

Alternativna scenarija

- 6.1. Ukoliko sistem ne pronađe korisnike, sistem prikazuje poruku “Za zadate parametre korisnici ne postoje”.
- 8.1. Ukoliko sistem ne učitava podatke, sistem prikazuje poruku o grešci.
- 11.1. Ukoliko sistem ne zapamti podatke, sistem prikazuje poruku “Korisnik nije ažuriran”.

SK3: Slučaj korišćenja - Unos usluge

Naziv SK

Unos usluge

Aktori SK

Zaposleni

Učesnici SK

Zaposleni, sistem

Preduslov

/

Osnovni scenario SK

1. Zaposleni pokreće formu za unos usluge.
2. Sistem istancira objekte.
3. Zaposleni unosi naziv usluge.
4. Zaposleni unosi datum od, datum do i cenu za cenovnik.
5. Zaposleni poziva sistem da doda cenovnik u listu pritiskom na dugme dodaj.
6. Sistem prihvata poziv, dodaje cenovnik u listu i prikazuje ga u tabeli.
7. Ukoliko želi, zaposleni može da pozove sistem da obriše određeni cenovnik iz tabele, pritiskom na dugme obriši.
8. Sistem prihvata poziv i briše cenovnik iz liste.
9. Ukoliko želi da ažurira određeni cenovnik, zaposleni pritiska dugme ažuriraj.

10. Sistem prihvata poziv, učitava cenovnik i prikazuje podatke zaposlenom.
11. Zaposleni menja podatke o cenovniku.
12. Zaposleni poziva sistem da ažurira podatke cenovnika pritiskom na dugme dodaj.
13. Sistem prihvata poziv, menja podatke o cenovniku i prikazuje promene u tabeli.
14. Zaposleni ponavlja korake od 4 do 13 dokle god ima potrebu za nekom od opcija.
15. Zaposleni poziva sistem da sačuva uslugu i cenovnike pritiskom na dugme sačuvaj.
16. Sistem prihvata poziv i pamti podatke u bazi pod transakcijom.
17. Sistem prikazuje poruku “Uspešno uneta usluga”.

Alternativna scenarija

- 6.1. Ukoliko sistem ne doda cenovnik u listu, sistem prikazuje poruku o grešci.
- 13.1. Ukoliko sistem ne promeni podatke, sistem prikazuje poruku o grešci.
- 17.1. Ukoliko sistem ne upiše podatke, sistem prikazuje poruku “Usluga nije uneta”.

SK4 Slučaj korišćenja - Ažuriranje usluge

Naziv SK

Ažuriranje usluge

Aktori SK

Zaposleni

Učesnici SK

Zaposleni, sistem

Preduslov

/

Osnovni scenario SK

1. Zaposleni pokreće formu za pretragu usluga
2. Sistem istancira objekte.
3. Zaposleni unosi podatke za pretragu usluga.
4. Zaposleni poziva sistem da pronađe usluge pritiskom na dugme pretraži.
5. Sistem prihvata poziv i traži usluge na osnovu unetih podataka.
6. Sistem prikazuje zaposlenom listu usluga sa porukom “Pronađene su usluge”.
7. Zaposleni poziva sistem da učitá podatke o usluzi klikom na dugme izmeni.
8. Sistem prihvata poziv, prikazuje formu za ažuriranje usluge i podatke o usluzi i cenovnicima.
9. Ukoliko ima potrebe, zaposleni menja naziv usluge.
10. Ukoliko ima potrebe, zaposleni poziva sistem da obriše cenovnik pritiskom na dugme obriši.

11. Sistem prihvata poziv, dodeljuje cenovniku operaciju delete i to prikazuje u tabeli.
12. Ukoliko ima potrebe, zaposleni priprema podatke za unos novog cenovnika unosom podataka datum od, datum do i cena.
13. Zaposleni poziva sistem da doda cenovnik u listu pritiskom na dugme dodaj.
14. Sistem privata poziv, dodaje cenovnik sa operacijom add u listu i prikazuje ga u tabeli.
15. Ukoliko želi da ažurira određeni cenovnik, zaposleni pritiska dugme ažuriraj.
16. Sistem prihvata poziv, učitava cenovnik i prikazuje podatke zaposlenom.
17. Zaposleni menja podatke o cenovniku.
18. Zaposleni poziva sistem da ažurira podatke cenovnika pritiskom na dugme dodaj.
19. Sistem prihvata poziv, menja podatke o cenovniku, dodeljuje operaciju update i prikazuje promene u tabeli.
20. Zaposleni ponavlja korake od 9 do 19 dokle god ima potrebu za nekom od opcija.
21. Zaposleni poziva sistem da sačuva uslugu i cenovnike pritiskom na dugme sačuvaj.
22. Sistem prihvata poziv i pamti podatke u bazi pod transakcijom.
23. Sistem prikazuje poruku “Uspešno ažurirana usluga”.

Alternativna scenarija

- 5.1. Ukoliko sistem ne pronađe usluge, sistem prikazuje poruku “Za zadate parametre usluge ne postoje”.
- 8.1. Ukoliko sistem ne prikaže podatke o usluzi, sistem prikazuje poruku o grešci.
- 14.1. Ukoliko sistem ne doda cenovnik u listu, sistem prikazuje poruku o grešci.
- 19.1. Ukoliko sistem ne izmeni podatke, sistem izbacuje poruku o grešci.
- 23.1. Ukoliko sistem ne upiše podatke, sistem prikazuje poruku “Usluga nije ažurirana”.

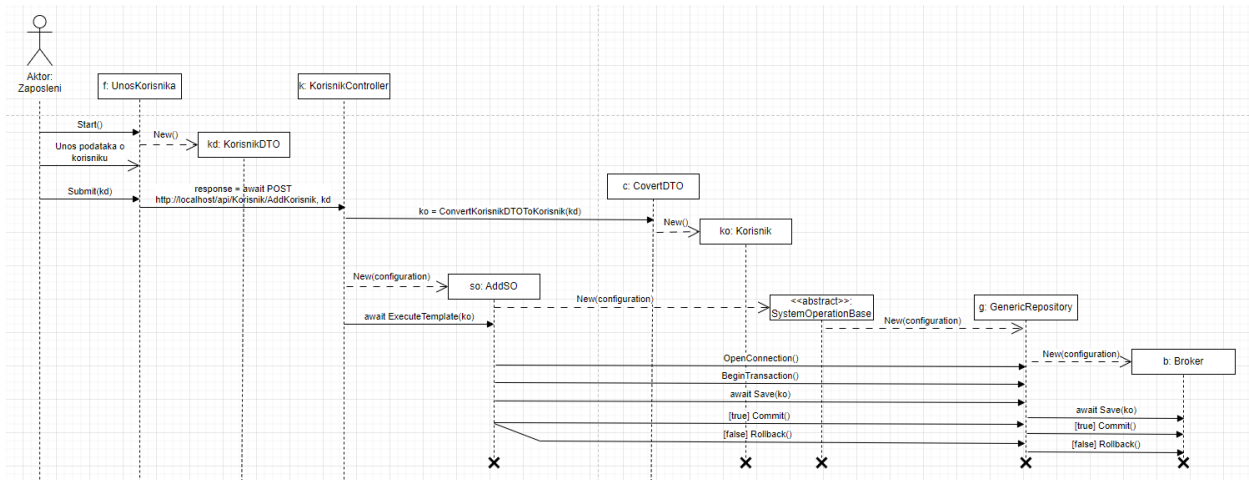
3. Analiza i projektovanje

U ovom poglavlju su opisane logička i fizička struktura i ponašanje softvera, pomoću dijagrama sekvenci, proširenog modela objekti-veze i relacionog modela.

3.1. Dijagram sekvenci

Dijagrami sekvenci, navedeni u nastavku, odnose se na jedan odabran prost i jedan složen slučaj korišćenja.

DS1: Dijagram sekvence slučaja korišćenja - Unos korisnika



Slika 2: Dijagram sekvence slučaja korišćenja - Unos korisnika

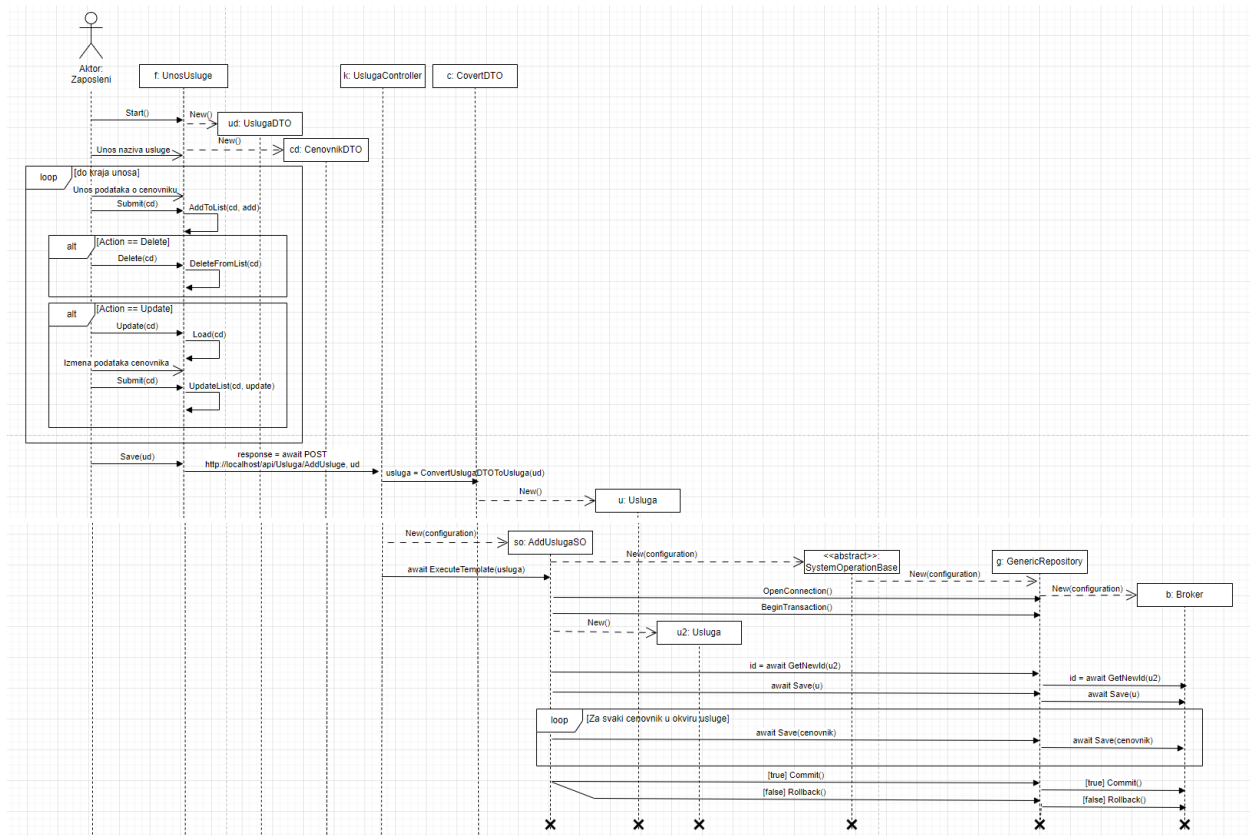
```

sequenceDiagram
    actor A as Aktor: Zaposleni
    participant f as f: PretragacKorisnika
    participant k as k: KorisnikController
    participant c as c: CovertDTO
    participant ko as ko: Korisnik
    participant so as so: GetSO
    participant sys as <<abstract>>: SystemOperationBase
    participant g as g: GenericRepository
    participant b as b: Broker
    participant f2 as f2: UnosKorisnika
    participant kd2 as kd2: KorisnikDTO
    participant ko2 as ko2: Korisnik
    participant so2 as so2: GetSO
    participant sys2 as <<abstract>>: SystemOperationBase
    participant g2 as g2: GenericRepository
    participant b2 as b2: Broker
    participant ko3 as ko3: Korisnik
    participant so3 as so3: UpdateSO
    participant sys3 as <<abstract>>: SystemOperationBase
    participant g3 as g3: GenericRepository
    participant b3 as b3: Broker

    A->>f: Start()
    f->>f: Unos podataka sa pretragu korisnika
    f->>f: Submit(kd)
    f->>f: New(i)
    f->>f: korisnici = await GET http://localhost/api/Korisnik/GetKorisnici.url
    f->>k: New(sifra, ime, prezime, brojTelefona, brojLK)
    k->>c: New(sifra, ime, prezime, brojTelefona, brojLK)
    c->>ko: New(sifra, ime, prezime, brojTelefona, brojLK)
    ko->>so: New(configuration)
    so->>sys: await ExecuteTemplate(ko)
    sys->>g: New(configuration)
    g->>b: New(configuration)
    b->>b: OpenConnection()
    b->>b: BeginTransaction()
    b->>b: result = await Get(ko)
    b->>b: [true] Commit()
    b->>b: [false] Rollback()
    b->>g: result = await Get(ko)
    g->>sys: [true] Commit()
    g->>sys: [false] Rollback()
    sys->>so: result = await Get(ko)
    so->>f: korisnici = ConvertKorisnicToKorisnicDTO(so.Result)
    f->>f: Update(kd)
    f->>f: Redirect(kd.Sifra)
    f->>f2: New()
    f2->>f2: kd2 = (await GET http://localhost/api /Korisnik/GetKorisnici.sifra(Korisnika).FirstOrDefault())
    f2->>f2: New(sifra, ime, prezime, brojTelefona, brojLK)
    f2->>ko2: New(sifra, ime, prezime, brojTelefona, brojLK)
    ko2->>so2: New(configuration)
    so2->>sys2: await ExecuteTemplate(ko2)
    sys2->>g2: New(configuration)
    g2->>b2: New(configuration)
    b2->>b2: OpenConnection()
    b2->>b2: BeginTransaction()
    b2->>b2: result = await Get(ko2)
    b2->>b2: [true] Commit()
    b2->>b2: [false] Rollback()
    b2->>g2: result = await Get(ko2)
    g2->>sys2: [true] Commit()
    g2->>sys2: [false] Rollback()
    sys2->>so2: result = await Get(ko2)
    so2->>f2: korisnici = ConvertKorisnicToKorisnicDTO(so.Result)
    f->>f: Izmena podataka korisnika
    f->>f: Submit(kd2)
    f->>f: response = (await POST http://localhost/api /Korisnik/UpdateKorisnik.kd2)
    f->>f: korisnik = ConvertKorisnikDTOToKorisnik(kd2)
    f->>ko3: New()
    ko3->>so3: New(configuration)
    so3->>sys3: await ExecuteTemplate(korisnik)
    sys3->>g3: New(configuration)
    g3->>b3: New(configuration)
    b3->>b3: OpenConnection()
    b3->>b3: BeginTransaction()
    b3->>b3: await Update(korisnik)
    b3->>b3: [true] Commit()
    b3->>b3: [false] Rollback()
    b3->>g3: await Update(korisnik)
    g3->>sys3: [true] Commit()
    g3->>sys3: [false] Rollback()
    sys3->>so3: await Update(korisnik)
    so3->>f: response = await POST http://localhost/api /Korisnik/UpdateKorisnik.kd2
    f->>f: response = await POST http://localhost/api /Korisnik/UpdateKorisnik.kd2
    
```

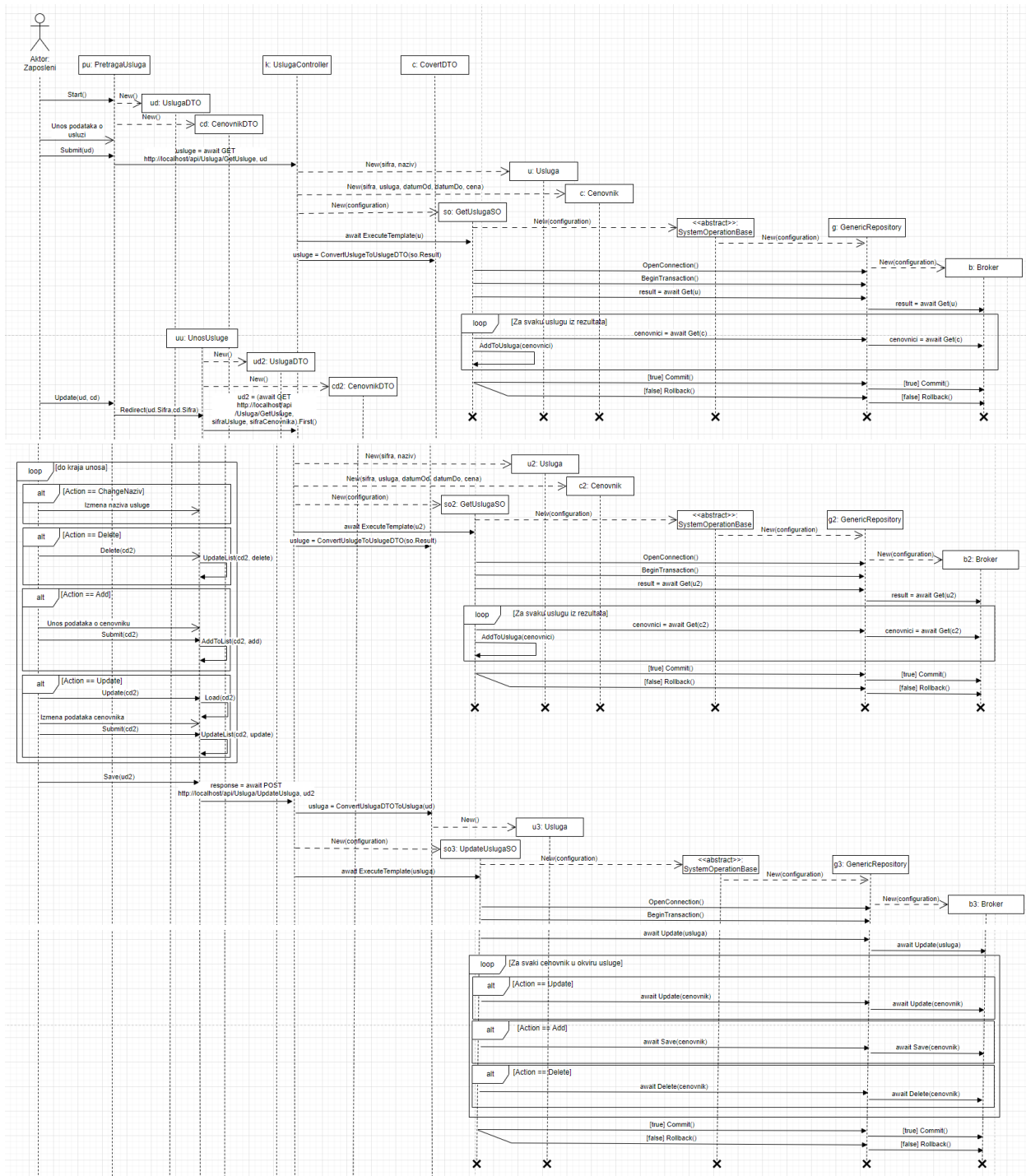
11

DS3: Dijagram sekvence slučaja korišćenja - Unos usluge



Slika 4: Dijagram sekvence slučaja korišćenja - Unos usluge

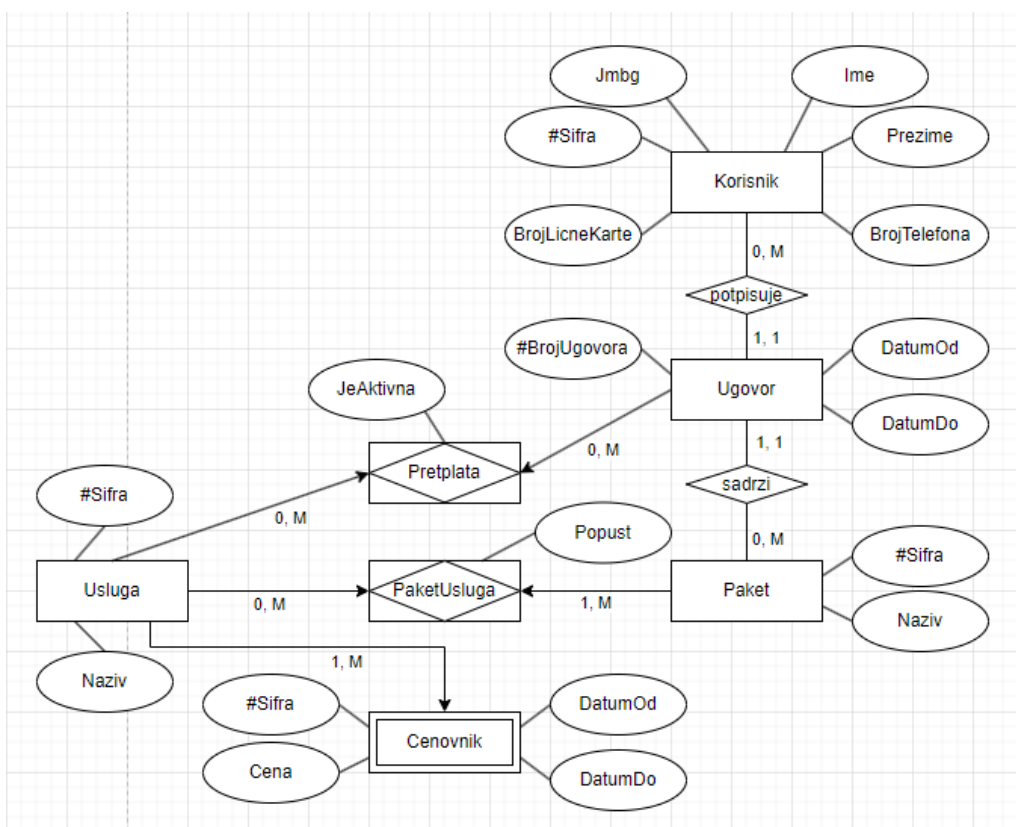
DS4: Dijagram sekvence slučaja korišćenja - Ažuriranje usluge



Slika 5: Dijagram sekvence slučaja korišćenja - Ažuriranje usluge

3.2. Proširen model objekti-veze

Na slici u nastavku se može videti da jedan korisnik može imati više ugovora, dok se jedan ugovor odnosi na jednog i samo jednog korisnika. Prilikom potpisivanja ugovora, korisnik mora izabrati tačno jedan paket, koji može postojati u okviru više ugovora. Svaki od paketa, koji se nalaze u ponudi na ugovoru, mora sadržati barem jednu uslugu, dok svaka usluga može postojati u više različitih paketa. Svaka usluga mora imati barem jednu cenu, i ona može postojati u okviru više ugovora, kao što i ugovori mogu imati više određenih usluga.



Slika 6: Prošireni model objekti-veze

3.3. Relacioni model

Korisnik(Sifra, Jmbg, Ime, Prezime, BrojTelefona, BrojLicneKarte)

Ugovor(BrojUgovora, DatumOd, DatumDo, *SifraKorisnika*, *SifraPaketa*)

Paket(Sifra, Naziv)

Usluga(Sifra, Naziv)

Cenovnik(Sifra, *SifraUsluge*, DatumOd, DatumDo, Cena)

PaketUsluga(*SifraUsluge*, *SifraPaketa*, Popust)

Pretplata(*SifraUsluge*, *BrojUgovora*, JeAktivna)

4. Zaključak

Svaki projekat pruža mogućnost sticanja novog znanja i potencijalno otvara nove puteve za dodatni razvoj. Prilikom razvoja aplikacije u Blazor frejmvorku nije bilo većih poteškoća, što ukazuje na njegovu intuitivnost i jednostavnost korišćenja, kao i kompletnost i kvalitet dokumentacije koja se može pronaći na zvaničnom sajtu Microsoft kompanije. Pored dokumentacije, postoje i brojni video zapisi koji ukazuju na različite mogućnosti, ali na koje se ne treba oslanjati u potpunosti jer je ova tehnologija još uvek u razvoju i prethodne tehnike brzo zastarevaju.

Kao što je već pomenuto, ovaj frejmvork usmerava korisnike da svoj fokus premeste na usavršavanje jednog programskog jezika, umesto istraživanja brojnih tehnologija koje zahteva dosta vremena. Na kraju razvoja ovog seminarskog rada, preporuka je da se obavezno istraže mogućnosti novog Microsoft-ovog alata i prođe kroz onsove koncepte koji su potrebni za razvoj aplikacije.

Pored razvoja standardne web aplikacije, Blazor pruža mogućnost razvoja web aplikacije koja može da se preuzme na lokalnu stanicu i na taj način je omogućena simulacija desktop aplikacije. U tom slučaju, ne radi se o pravoj desktop aplikaciji, već o web aplikaciji koja ima svoju ikonicu na radnoj površini računara, preko koje je moguće otvoriti aplikaciju koja će izgledati kao da je u pitanju desktop i koja pruža mogućnost rada čak i u offline modu. Kao i u radu sa drugim aplikacijama u offline modu, treba obratiti pažnju na ažuriranje aplikacije kada dođe do promene verzije, kao i ažuriranje podataka, ukoliko je korisnik vršio izmene u trenutku korišćenja aplikacije bez konekcije na internet.