

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 | 8 |
| 3.1. Dijagram sekvenci | 8 |
| 3.2. Proširen model objekti-veze | 10 |
| 3.3. Relacioni model..... | 10 |
| 4. Zaključak | 11 |

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 ćemo proći kroz 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 poslovnicu, 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.

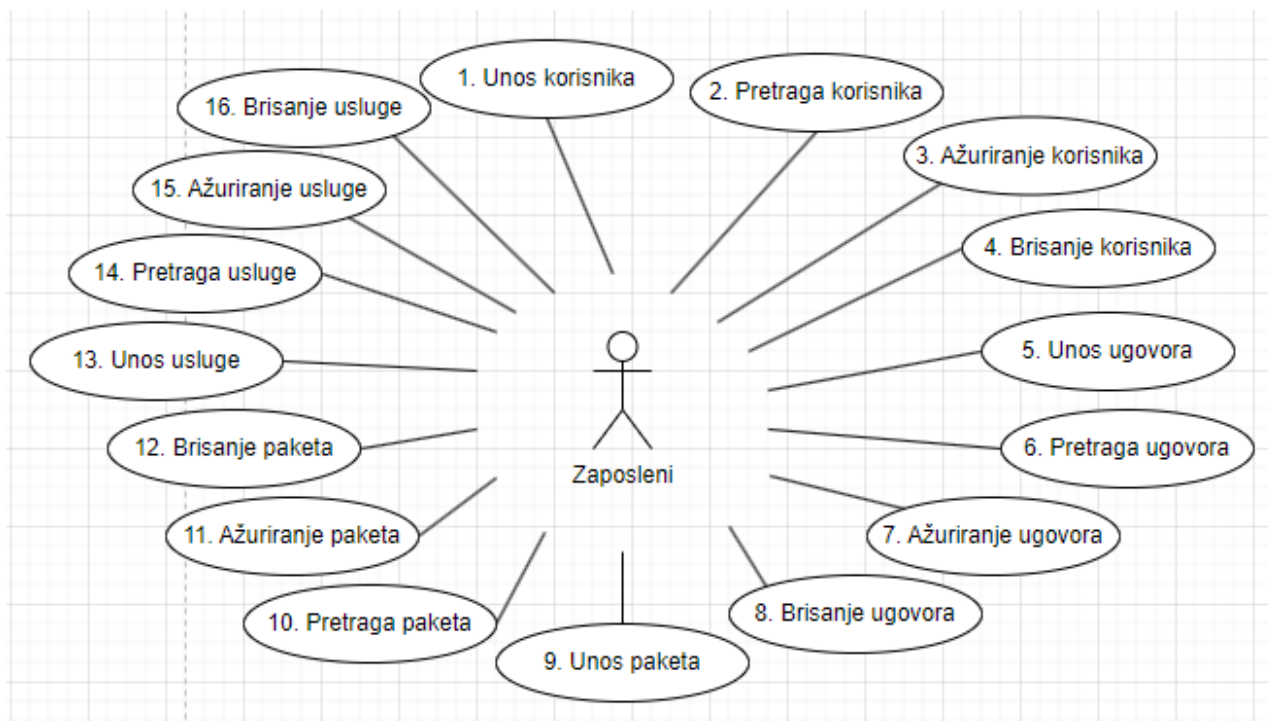
Aplikacija treba da omogući unos, pregled, ažuriranje i brisanje korisnika, ugovora, paketa i usluga.

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 (složen slučaj)
16. Brisanje usluge



Slika 1: Slučajevi korišćenja

SK1: Slučaj korišćenja - Unos korisnika

Naziv SK

Unos korisnika

Aktori SK

Zaposleni

Učesnici SK

Zaposleni, sistem

Preduslov

Sistem je uključen i prikazana je forma za unos korisnika.

Osnovni scenario SK

1. Zaposleni unosi podatke o novom korisniku.
2. Zaposleni proverava da li je uneo ispravne podatke.
3. Zaposleni poziva sistem da zapamti novog korisnika.
4. Sistem pamti novog korisnika.
5. Sistem prikazuje poruku "Uspešno unet korisnik".

Alternativna scenarija

5.1. Ukoliko sistem ne upiše podatke, sistem prikazuje poruku "Korisnik nije unet".

SK2 Slučaj korišćenja - Ažuriranje usluge

Naziv SK

Ažuriranje usluge

Aktori SK

Zaposleni

Učesnici SK

Zaposleni, sistem

Preduslov

Sistem je uključen i prikazana je forma za pretragu usluga.

Osnovni scenario SK

1. Zaposleni unosi podatke za pretragu usluga.
2. Zaposleni poziva sistem da pronađe usluge.
3. Sistem traži usluge na osnovu unetih podataka.
4. Sistem prikazuje zaposlenom listu usluga sa porukom "Pronađene su usluge".
5. Zaposleni bira uslugu iz liste.
6. Zaposleni poziva sistem da učitava podatke o usluzi.
7. Sistem učitava podatke o usluzi.
8. Sistem prikazuje podatke o usluzi.
9. Zaposleni menja podatke o usluzi.
10. Zaposleni poziva sistem da zapamti izmenjene podatke o usluzi.
11. Sistem pamti podatke o usluzi.
12. Sistem prikazuje poruku "Uspešno ažurirana usluga".

Alternativna scenarija

- 4.1. Ukoliko sistem ne pronađe usluge, sistem prikazuje poruku "Za zadate parametre usluge ne postoje".
- 12.1. Ukoliko sistem ne upiše podatke, sistem prikazuje poruku "Usluga nije ažurirana".

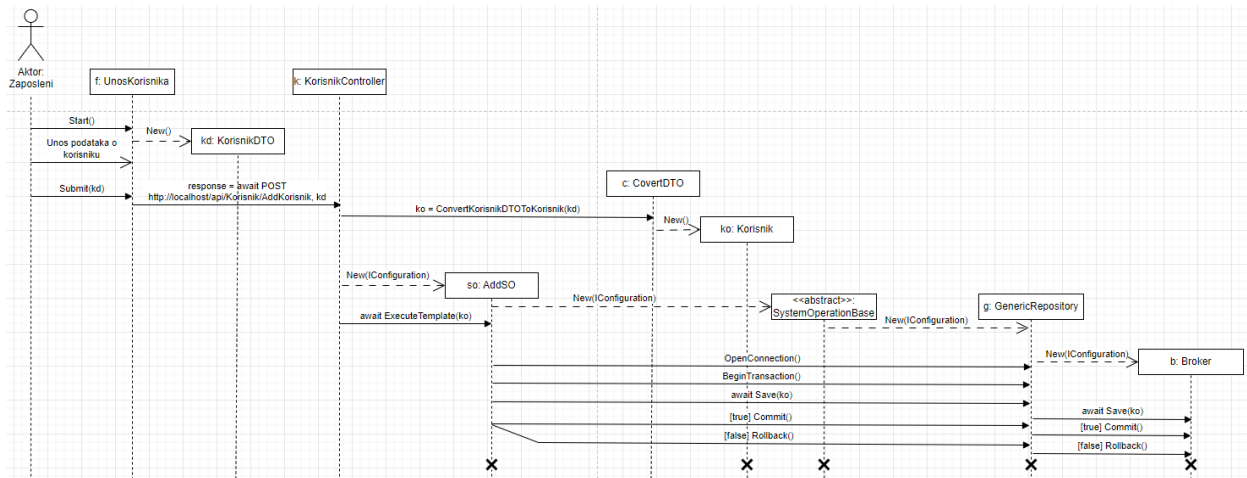
3. Analiza i projektovanje

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



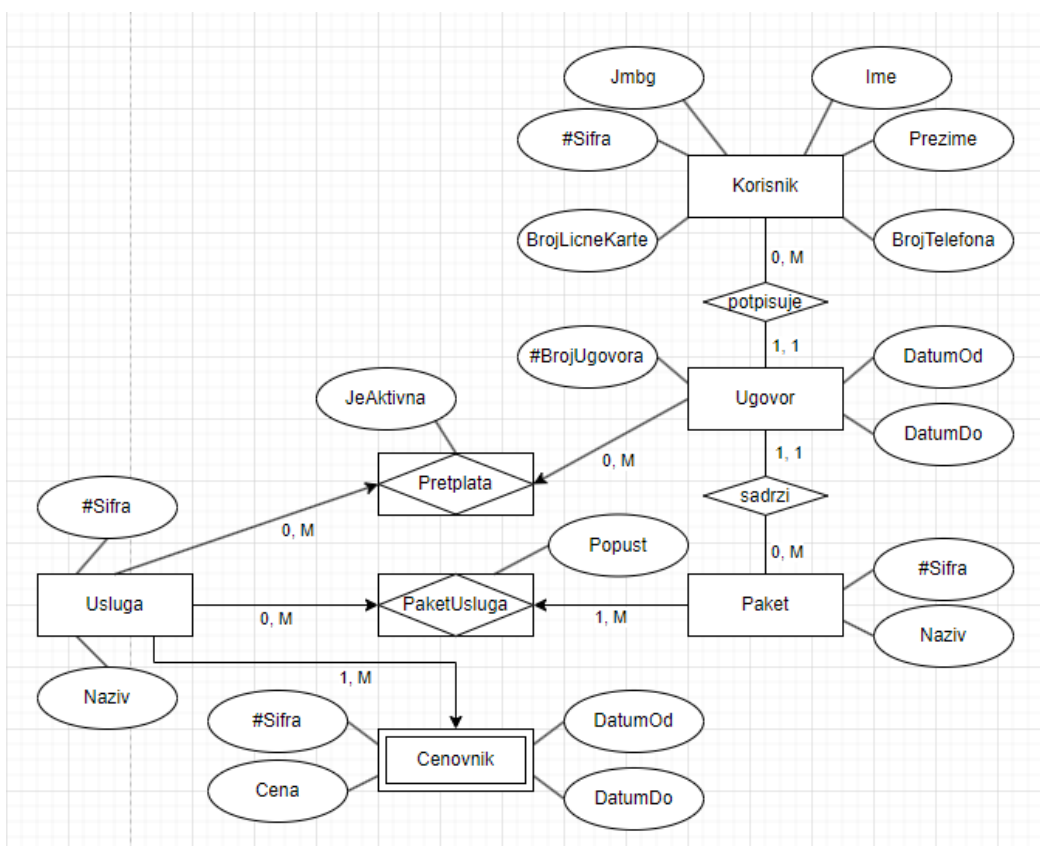
Slika 2: Dijagram sekvence slučaja korišćenja - Unos korisnika

[illegible]

9

3.2. Proširen model objekti-veze

Na slici u nastavku možemo 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 4: 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 nisam naišao na veće poteškoće, 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 frejmwork 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, moja topla preporuka svakome je da obavezno istraže mogućnosti novog Microsoft-ovog alata i prođu kroz onsove koncepte koji su potrebni za razvoj aplikacije.

Na samom kraju, pomenuo bih da, pored razvoja standardne web aplikacije, Blazor pruža mogućnost da se razvije web aplikacija 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 ažuriranju podataka, ukoliko je korisnik vršio izmene u trenutku korišćenja aplikacije bez konekcije na internet.