

Oblikovanje programske podpore

Ak. god. 2018./2019.

Sjedi pa jedi

Dokumentacija, Rev. 2.

Grupa: *PrimjerGrupa*

Voditelj: *PrimjerVoditelj*

Datum predaje: 26. rujna 2019.

Nastavnik: *PrimjerNastavnik*

Ukoliko smatrate da se neki dio primjera dokumentacije može poboljšati ili ispraviti, molimo da u pdf dokumentu zalijepite komentar s ispravkom i pošaljete datoteku na opp@fer.hr s naslovom "Ispravak primjera dokumentacije".

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	7
3.1 Funkcionalni zahtjevi	7
3.1.1 Obrasci uporabe	9
3.1.2 Sekvencijski dijagrami	25
3.2 Ostali zahtjevi	31
4 Arhitektura i dizajn sustava	32
4.1 Baza podataka	34
4.2 Dijagram razreda	39
4.3 Dijagram stanja	44
4.4 Dijagram aktivnosti	45
4.5 Dijagram komponenti	47
5 Implementacija i korisničko sučelje	49
5.1 Korištene tehnologije i alati	49
5.2 Ispitivanje programskog rješenja	50
5.3 Dijagram razmještaja	52
5.4 Upute za puštanje u pogon	53
6 Zaključak i budući rad	58
Popis literature	60
Indeks slika i dijagrama	61
Dodatak: Prikaz aktivnosti grupe	62

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	*	14.10.2018.
0.2	Dodani opis projektnog zadatka i funkcionalni zahtjevi	*	16.10.2018.
0.2.1	Dodan prvi dio opisa obrazaca uporabe	*, *, *	19.10.2018.
0.3	Dodani svi opisi obrazaca uporabe	*, *, *, *	19.10.2018.
0.3.1	Popravljen prvi dio obrazaca uporabe	*	22.10.2018.
0.3.2	Popravljen drugi dio obrazaca uporabe	*	24.10.2018.
0.3.3	Ispravak obrazaca uporabe	*	25.10.2018.
0.3.4	Ispravak obrazaca uporabe	*	27.10.2018.
0.4	Dodani dijagrami obrazaca uporabe	*, *	27.10.2018.
0.4.1	Dodani novi obrasci uporabe	*	30.10.2018.
0.4.2	Izmjena dijagrama obrazaca uporabe	*	31.10.2018.
0.4.3	Dodan novi obrazac uporabe, izmjena dijagrama	*, *	01.11.2018.
0.5	Dodani sekvencijski dijagrami i ostali zahtjevi	*, *	03.11.2018.
0.5.1	Popravljeni sekvencijski dijagrami i dijagrami obrazaca uporabe	*	4.11.2018.
0.6	Arhitektura i dizajn sustava	*, *, *, *, *, *	15.11.2018.
0.6.1	Zaključak i budući rad, dodatak A, B i C	*, *	15.11.2018.
0.6.2	Dijagram objekata	*	15.11.2018.
0.6.3	Pojmovnik	*	15.11.2018.
0.7	Popravljena baza podataka	*, *	22.11.2018.
0.8	Dijagram razreda i dijagram objekata	*, *	22.11.2018.

Rev.	Opis promjene/dodatka	Autori	Datum
0.9	Opisi razreda i dodatak D	*	26.11.2018.
1.0	Korigiranje teksta i provjera dokumentacije	*, *	26.11.2018.
1.1	Implementacija	*	10.1.2019
1.2	Dodani dijagram stanja i dijagram aktivnosti	*, *	13.1.2019
1.3	Dodan dijagram razmještaja	*	14.1.2019
1.4	Dodan opis tehnologija i alata	*	15.1.2019
1.5	Dodan dio korisničkih uputa	*	16.1.2019
1.6	Dodan dijagram komponenti	*, *	17.1.2019
1.7	Konačne korisničke upute	*, *	17.1.2019
1.8	Dodana potrebna instalacija	*	17.1.2019
1.9	Pregledano i dodane slike s opisima	*	17.1.2019
2.0	Konačna verzija		

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije "Sjedi pa jedi" koja će korisniku omogućiti da nakon što sjedne za određeni stol u nekom restoranu, naruči i plati željeno jelo. Tako će jedini zadatak konobara biti da donesu određenu narudžbu za stol kako ne bi bili potkapacitirani i kako bi se primanje i obrada narudžbi svih gostiju odradila prihvatljivom brzinom.

Prilikom pokretanja sustava prikazuje se karta sa ucrtanim restoranima koji se nalaze u blizini.

Neregistriranom korisniku odabirom restorana otvaraju se opće informacije o restoranu: ime restorana, adresa, telefon, fax, OIB, IBAN, žiro račun i slika. Uz to, navedeni su svi proizvodi po kategorijama odabranog restorana: naziv jela, opis(sastojci, upozorenja na alergije i slično), slika i cijena. Neregistriranom korisniku je omogućeno prijavljivanje u sustav s postojećim računom (potrebno je upisati korisničko ime i lozinku) ili kreiranjem novog računa. Za kreiranje novog računa potrebni su sljedeći podaci:

- korisničko ime
- lozinka
- ime
- prezime
- broj mobitela
- email adresa
- broj kreditne kartice

Registracijom u sustav korisniku se dodjeljuju prava klijenta, a naknadno mu se mogu dodijeliti prava zaposlenika (kuhar, konobar), vlasnika ili administratora. Registrirani korisnik može pregledati, mijenjati osobne podatke i izbrisati svoj korisnički račun.

Klijent odabirom i dolaskom u restoran sjeda za stol koji je registriran tako da u aplikaciji ima svoj jedinstveni broj. Unošenjem tog broja u aplikaciju otvara se meni iz kojeg može naručiti željene proizvode. Kada završi s biranjem, završava kupnju i plaća račun. Potvrdom plaćanja narudžba postaje aktivna u sustavu i ona

se prosljeđuje zaposlenicima. Nakon obroka klijentu se aktivira opcija ocjenjivanja (1-5) i pisanja recenzije za proizvode koje je konzumirao i restoran u kojem je bio.

Kako klijenti ne bi ovisili isključivo o stolovima slobodnim u trenutku dolaska postoji mogućnost rezervacije jednog ili više stolova za određen datum u kalendaru. Klijent svojim rezervacijama može upravljati (mijenjati, brisati), a u slučaju da nema slobodnih stolova klijent će biti obaviješten kroz aplikaciju.

Uz klijenta postoje još 3 vrste korisnika, a to su:

- zaposlenik (kuhar/konobar)
- vlasnik
- admin

Zaposlenik restorana kroz aplikaciju ima pristup narudžbama onog restorana kojem je dodijeljen, a za sve ostale restorane on je samo klijent. Njegova zadaća je upravljanje narudžbama, tj. kada je narudžba gotova i dostavljena, mora ju označiti kao gotovu kako bi dalje klijent mogao recenzirati.

Vlasnik restorana ima širi spektar mogućnosti za rad u aplikaciji. Njemu je omogućeno ažuriranje podataka o stolovima i ponudi svog restorana te pregledavanje recenzija i rezervacija u svrhu vođenja restorana. Također on odgovara na recenzije i dodaje ili briše zaposlenike.

Administrator sustava ima najveće ovlasti. On ima pristup bazi s popisom registriranih korisnika i njihovim podacima te ih može brisati. Kao i vlasnik restorana, administrator ima mogućnosti ažuriranja ponude određenog restorana, ali i brisanju recenzija suprotnih pravilima korištenja aplikacije. Uz to, jedini je koji ima pristup statistici rada restorana.

Sustav treba podržavati rad više korisnika u stvarnom vremenu.

Napomena: Generalna primjedba je da opis projektnog zadatka ima samo nešto više od jedne stranice. Toliko obično ima tekst koji je zadan studentima. Taj tekst je potrebno proširiti s vlastitim opisom, lijepo strukturirati po točkama, dodati opis i slike sličnih rješenja, itd. Očekuje se min. 3, a poželjno 4-5 stranica opisa.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Vlasnik (naručitelj)
2. Klijenti restorana
3. Zaposlenici restorana
 - (a) Konobar
 - (b) Kuhar
4. Administrator
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) pregledati na karti pozicije restorana
 - (b) odabrati restoran i dobiti prikaz općih informacija (ime restorana, adresa, telefon, fax, OIB, IBAN, žiro račun, slika)
 - (c) odabrati restoran i uz opće informacije dobiti prikaz svih proizvoda (naziv jela, opis, slika, cijena) po kategorijama
 - (d) se registrirati u sustav, stvoriti novi korisnički račun za koji su mu potrebni korisničko ime, lozinka, ime, prezime, broj mobitela, e-mail adresa i broj kreditne kartice
2. Klijent (inicijator) može:
 - (a) pregledavati i mijenjati osobne podatke
 - (b) izbrisati svoj korisnički račun
 - (c) unošenjem broja stola za kojim sjedi naručiti i platiti željene proizvode iz menija
 - (d) rezervirati jedan ili više stolova
 - (e) svoje rezervacije urediti i brisati

- (f) pisati recenzije i dati ocjene
- (g) pregledati recenzije restorana

3. Zaposlenik (konobar/kuhar) (inicijator) može:

- (a) vidjeti popis aktivnih narudžbi za restoran u kojem je zaposlen
- (b) nakon što je narudžba pripremljena i dostavljena kupcu, označiti ju kao gotovu

4. Vlasnik (inicijator) može:

- (a) dodati, izbrisati i promijeniti broj stolova i njihov kapacitet u svom restoranu
- (b) dodati, izbrisati i promijeniti kategorije u meniju
- (c) dodati, izbrisati i promijeniti proizvode u kategorijama
- (d) pregledati narudžbe u stvarnom vremenu
- (e) pregledati rezervacije stolova u restoranu
- (f) odgovoriti na recenzije korisnika
- (g) po korisničkom imenu dodati ili maknuti klijenta kao svog zaposlenika
- (h) dodati ili obrisati restoran

5. Administrator (inicijator) može:

- (a) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (b) korisnike brisati i mijenjati im razinu pristupa aplikaciji (klijent, zaposlenik, vlasnik)
- (c) pristupiti statistici (npr. broj izvršenih narudžbi po restoranima po mjesecima, broj narudžbi po klijentu, najpopularnija jela, zarađeni novci i slično)
- (d) brisati recenzije koje su u suprotnosti s pravilima korištenja aplikacije
- (e) dodati ili obrisati restoran

6. Baza podataka (sudionik):

- (a) pohranjuje sve podatke o korisnicima i njihovim ovlastima
- (b) pohranjuje sve podatke o restoranima, njihovim kapacitetima i ponudi

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

UC1 - Pregled restorana

- **Glavni sudionik:** Korisnik, klijent
- **Cilj:** Pregledati restorane i ponude
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Karta je prikazana prilikom učitavanja aplikacije
 2. Korisnik na karti odabire restoran
 3. Prikazuju se informacije o restoranu i ponudi

UC2 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne korisničke podatke
 3. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC3 – Prijava u sustav

- **Glavni sudionik:** Klijent
- **Cilj:** Dobiti pristup korisničkom sučelju
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija

- **Opis osnovnog tijeka:**
 1. Unos korisničkog imena i lozinke
 2. Potvrda o ispravnosti unesenih podataka
 3. Pristup korisničkim funkcijama
- **Opis mogućih odstupanja:**
 - 2.a Neispravno korisničko ime/lozinka
 1. Sustav obavještava korisnika o neuspjehom upisu i vraća ga na stranicu za registraciju

UC4 - Pregled osobnih podataka

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Osobni podatci"
 2. Aplikacija prikazuje osobne podatke korisnika

UC5 – Promjena osobnih podataka

- **Glavni sudionik:** Klijent
- **Cilj:** Promijeniti osobne podatke
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabere opciju za promjenu podataka
 2. Korisnik mijenja svoje osobne podatke
 3. Korisnik sprema promjene
 4. Baza podataka se ažurira
- **Opis mogućih odstupanja:**
 - 2.a Korisnik promijeni svoje osobne podatke, ali ne odabere opciju "Spremi promjenu"
 1. Sustav obavještava korisnika da nije spremio podatke prije izlaska iz prozora

UC6 – Brisanje korisničkog računa

- **Glavni sudionik:** Klijent
- **Cilj:** Izbrisati svoj korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik pregledava osobne podatke
 2. Otvara se stranica s osobnim podacima korisnika
 3. Korisnik briše račun
 4. Korisnički račun se izbriše iz baze podataka
 5. Otvara se stranica za registraciju

UC7 – Rezervacija stola

- **Glavni sudionik:** Klijent
- **Cilj:** Rezervirati jedan ili više stolova
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Klijent odabire restoran u kojem želi rezervirati stol
 2. Klijent odabire opciju "Rezerviraj stol"
 3. Klijent odabire datum rezervacije
 4. Prikaže se stranica sa listom slobodnih stolova
 5. Klijent odabire jedan ili više stolova koje želi rezervirati
 6. Klijent odabire opciju "Potvrdi rezervaciju"
- **Opis mogućih odstupanja:**
 - 3.a U restoranu nema slobodnih stolova
 1. Sustav obavještava korisnika da nema slobodnih stolova i daje klijentu opciju mijenjanja datuma rezervacije

UC8 – Naručivanje proizvoda

- **Glavni sudionik:** Klijent
- **Cilj:** Naručiti proizvode
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen i postoji slobodan stol ili je korisnik rezervirao stol

- **Opis osnovnog tijeka:**

1. Klijent odabire restoran u kojem se nalazi
2. Klijent unosi broj stola za kojim sjedi
3. Prikaže se jelovnik restorana
4. Klijent označi proizvode koje želi naručiti
5. Klijent odabire opciju "Potvrdi narudžbu"
6. Prikaže se krajnja narudžba

- **Opis mogućih odstupanja:**

- Proizvod kojeg klijent želi naručiti nema više na jelovniku
 1. Sustav obavještava klijenta da proizvod nije dostupan

UC9 – Plaćanje narudžbe

- **Glavni sudionik:** Klijent
- **Cilj:** Platiti željene proizvode s jelovnika
- **Sudionici:** Baza podataka, zaposlenik
- **Preduvjet:** Klijent je prijavljen i napravio je narudžbu
- **Opis osnovnog tijeka:**
 1. Klijent se prikazuju podatci o narudžbi
 2. Klijent odabire opciju "Dovrši narudžbu"
 3. Narudžba se prikazuje u listi aktivnih narudžbi kod zaposlenika

UC10 – Uređivanje rezervacije

- **Glavni sudionik:** Klijent
- **Cilj:** Urediti rezervaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen i napravio je barem jednu rezervaciju
- **Opis osnovnog tijeka:**
 1. Klijent odabire opciju "Moje rezervacije"
 2. Prikaže se lista rezervacija korisnika
 3. Klijent odabire rezervaciju koju želi urediti
 4. Prikaže se lista slobodnih stolova
 5. Klijent promijeni svoju rezervaciju i odabire "Spremi"

UC11 – Brisanje rezervacije

- **Glavni sudionik:** Klijent
- **Cilj:** Obrisati rezervaciju
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen i napravio je barem jednu rezervaciju
- **Opis osnovnog tijeka:**
 1. Klijent odabire opciju "Moje rezervacije"
 2. Prikaže se lista rezervacija klijenta
 3. Klijent odabire opciju "Obriši rezervaciju"
 4. Rezervacija se briše iz baze podataka

UC12 - Ocjenjivanje restorana

- **Glavni sudionik:** Klijent
- **Cilj:** Napisati recenziju i ocijeniti restoran
- **Sudionici:** Baza podataka
- **Preduvjet:**
 - Klijent je prijavljen
 - Klijent je naručio proizvode i platio
 - Zaposlenik je označio narudžbu gotovom
- **Opis osnovnog tijeka:**
 1. Klijent napiše recenziju i ocijeni restoran
 2. Ocjena i osvrt se pohranjuju u bazu podataka i ažurira se prosječna ocjena restorana
- **Opis mogućih odstupanja:**
 - 1.a Klijent ne želi napisati osvrt
 1. Sustav zatvara prozor za ocjenjivanje

UC13 - Pregled recenzija

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati recenzije odabranoga restorana
- **Sudionici:** Baza podataka
- **Preduvjet:** Klijent je prijavljen
- **Opis osnovnog tijeka:**
 1. Klijent odabire opciju "Pregledaj recenzije"
 2. Prikažu se svi dotadašnji osvrti i ocjene od odabranoga restorana

UC14 - Pregled aktivne narudžbe

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Pogledati trenutno aktivne narudžbe
- **Sudionici:** Baza podataka
- **Preduvjet:** Narudžba je zaprimljena i plaćena
- **Opis osnovnog tijeka:**
 1. Zaposlenik odabere opciju "Aktivne narudžbe"
 2. Prikaže se lista trenutno aktivnih narudžbi

UC15 – Označavanje narudžbe gotovom

- **Glavni sudionik:** Zaposlenik
- **Cilj:** Označiti narudžbu kao gotovu
- **Sudionici:** Baza podataka, klijent
- **Preduvjet:** -
 - Zaposlenik je prijavljen
 - Narudžba je zaprimljena i plaćena
- **Opis osnovnog tijeka:**
 1. Zaposlenik odabire narudžbu koja je spremna
 2. Zaposlenik označava odabranu narudžbu gotovom
 3. Klijentu se omogućuje opcija recenziranja

UC16 – Dodavanje stola

- **Glavni sudionik:** Vlasnik
- **Cilj:** Dodati novi stol
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju pregleda kapaciteta restorana
 2. Vlasniku se otvara raspored stolova
 3. Vlasnik odabire opciju "Dodaj novi stol"
 4. Vlasnik unosi podatke o novom stolu
 5. Promjene se upisuju u bazu podataka

UC17 – Uređivanje stola

- **Glavni sudionik:** Vlasnik
- **Cilj:** Izmijeniti stol i njegov kapacitet
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju pregleda kapaciteta restorana
 2. Vlasniku se otvara raspored stolova
 3. Vlasnik odabire opciju izmjene broja stolova i kapaciteta
 4. Nakon uređivanja potvrđuje izmjenu
 5. Promjene se upisuju u bazu podataka

UC18 – Uklanjanje stola

- **Glavni sudionik:** Vlasnik
- **Cilj:** Ukloniti stol iz popisa
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju pregleda kapaciteta restorana
 2. Vlasniku se otvara raspored stolova
 3. Vlasnik odabire opciju brisanja stola
 4. Nakon uređivanja potvrđuje izmjenu
 5. Promjene se upisuju u bazu podataka

UC19 – Dodavanje kategorije

- **Glavni sudionik:** Vlasnik
- **Cilj:** Dodati kategoriju u jelovnik restorana
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju "Prikaz karte"
 2. Vlasniku se prikazuje karta samo njegovih restorana
 3. Vlasnik odabire restoran u čiji jelovnik želi dodati novu kategoriju jela
 4. Vlasnik dodaje novu kategoriju jela na jelovnik odabranoga restorana

UC20 – Brisanje kategorije

- **Glavni sudionik:** Vlasnik
- **Cilj:** Obrisati kategoriju iz jelovnika restorana
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju "Prikaz karte"
 2. Vlasniku se prikazuje karta samo njegovih restorana
 3. Vlasnik odabire restoran s čijeg jelovnika želi obrisati kategoriju jela
 4. Vlasnik briše kategoriju jela s jelovnika odabranoga restorana

UC21 – Uređivanje kategorije

- **Glavni sudionik:** Vlasnik
- **Cilj:** Urediti kategoriju u jelovniku restorana
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju "Prikaz karte"
 2. Vlasniku se prikazuje karta samo njegovih restorana
 3. Vlasnik odabire restoran na čijem jelovniku želi izvršiti promjenu kategorije jela
 4. Vlasnik mijenja kategoriju jela na jelovniku odabranoga restorana

UC22 – Dodavanje proizvoda u jelovnik

- **Glavni sudionik:** Vlasnik
- **Cilj:** Dodati proizvod u određenu kategoriju jela
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju "Prikaz karte"
 2. Vlasniku se prikazuje karta samo njegovih restorana
 3. Vlasnik odabire restoran na čijem jelovniku želi izvršiti promjenu kategorije jela
 4. Vlasnik odabire kategoriju u koju želi dodati proizvod
 5. Vlasnik dodaje proizvod u odabranu kategoriju

UC23 – Brisanje proizvoda s jelovnika

- **Glavni sudionik:** Vlasnik
- **Cilj:** Ukloniti proizvod iz kategorije jela određenog restorana
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju "Prikaz karte"
 2. Vlasniku se prikazuje karta samo njegovih restorana
 3. Vlasnik odabire restoran na čijem jelovniku želi izvršiti promjenu kategorije jela
 4. Vlasnik odabire kategoriju iz koje želi izbrisati proizvod
 5. Vlasnik briše proizvod iz odabrane kategorije

UC24 – Uređivanje proizvoda u jelovniku

- **Glavni sudionik:** Vlasnik
- **Cilj:** Urediti proizvod unutar kategorije jela određenog restorana
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik odabire opciju "Prikaz karte"
 2. Vlasniku se prikazuje karta svih restorana, a vlasniku samo njegovih
 3. Vlasnik odabire restoran na čijem jelovniku želi izvršiti promjenu kategorije jela
 4. Vlasnik odabire kategoriju unutar koje želi promijeniti proizvod
 5. Vlasnik mijenja proizvod unutar odabrane kategorije

UC25 – Pregled rezervacija restorana

- **Glavni sudionik:** Vlasnik
- **Cilj:** Pregledati rezervacije stolova u restoranu
- **Sudionici:** Baza podataka
- **Preduvjet:** Vlasnik je prijavljen
- **Opis osnovnog tijeka:**
 1. Vlasnik u aplikaciji odabere opciju "Rezervacije restorana"
 2. Prikaže se lista svih rezervacija stolova

UC26 – Odgovaranje na recenzije

- **Glavni sudionik:** Vlasnik
- **Cilj:** Odgovoriti na recenziju klijenta
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodijeljena su mu prava vlasnika te postoji barem jedna recenzija za restoran
- **Opis osnovnog tijeka:**
 1. Vlasnik u aplikaciji ode na karticu "Recenzije restorana" i otvori se prikaz svih recenzija
 2. Klikom na određenu recenziju otvara se opcija odgovaranja
 3. Nakon što odgovori na recenziju, odgovor postaje vidljiv ispod recenzije restorana

UC27 – Dodavanje zaposlenika

- **Glavni sudionik:** Vlasnik
- **Cilj:** Dodati zaposlenika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodijeljena su mu prava vlasnika
- **Opis osnovnog tijeka:**
 1. Vlasnik odabere opciju dodavanja zaposlenika
 2. Otvara se prozor za upis korisničkog imena zaposlenika
 3. Vlasnik upiše korisničko ime i potvrdi upis
 4. U bazu podataka se pohrani promjena
- **Opis mogućih odstupanja:**
 - 3.a Upisano je neispravno korisničko ime zaposlenika
 1. Sustav obavještava vlasnika o neuspjelom upisu

UC28 – Brisanje zaposlenika

- **Glavni sudionik:** Vlasnik
- **Cilj:** Obrisati zaposlenika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen i dodijeljena su mu prava vlasnika
- **Opis osnovnog tijeka:**
 1. Vlasnik odabere opciju brisanja zaposlenika
 2. Otvara se prozor za upis korisničkog imena zaposlenika

3. Vlasnik upiše korisničko ime i potvrdi upis
4. U bazu podataka se pohrani promjena
- **Opis mogućih odstupanja:**
 - 3.a Upisano je neispravno korisničko ime zaposlenika
 1. Sustav obavještava vlasnika o neuspjelom upisu

UC29 – Pregled korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati registrirane korisnike
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju pregledavanja korisnika
 2. Prikaže se lista svih ispravno registriranih korisnika s osobnim podacima

UC30 – Brisanje korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju uklanjanja korisnika
 2. Administrator pronalazi željenog korisnika
 3. Administrator uklanja željenog korisnika i njegove podatke iz baze podataka

UC31 – Promjena prava pristupa

- **Glavni sudionik:** Administrator
- **Cilj:** Promijeniti razinu pristupa korisnika (klijent, zaposlenik, vlasnik)
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator pronalazi željenog korisnika
 2. Administrator mijenja razinu pristupa željenom korisniku

UC32 – Pregled statistike

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati statistiku svih restorana
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju "Prikaz karte"
 2. Administratoru se prikazuje karta svih restorana
 3. Administrator odabire restoran
 4. Prikazuju se podatci i statistika odabranoga restorana

UC33 - Brisanje recenzije

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati recenziju
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabere recenzije za željeni restoran
 2. Klikom na recenziju prikaže mu se opcija "Izbriši"
 3. Recenzija se uklanja iz baze podataka

UC34 - Dodavanje restorana

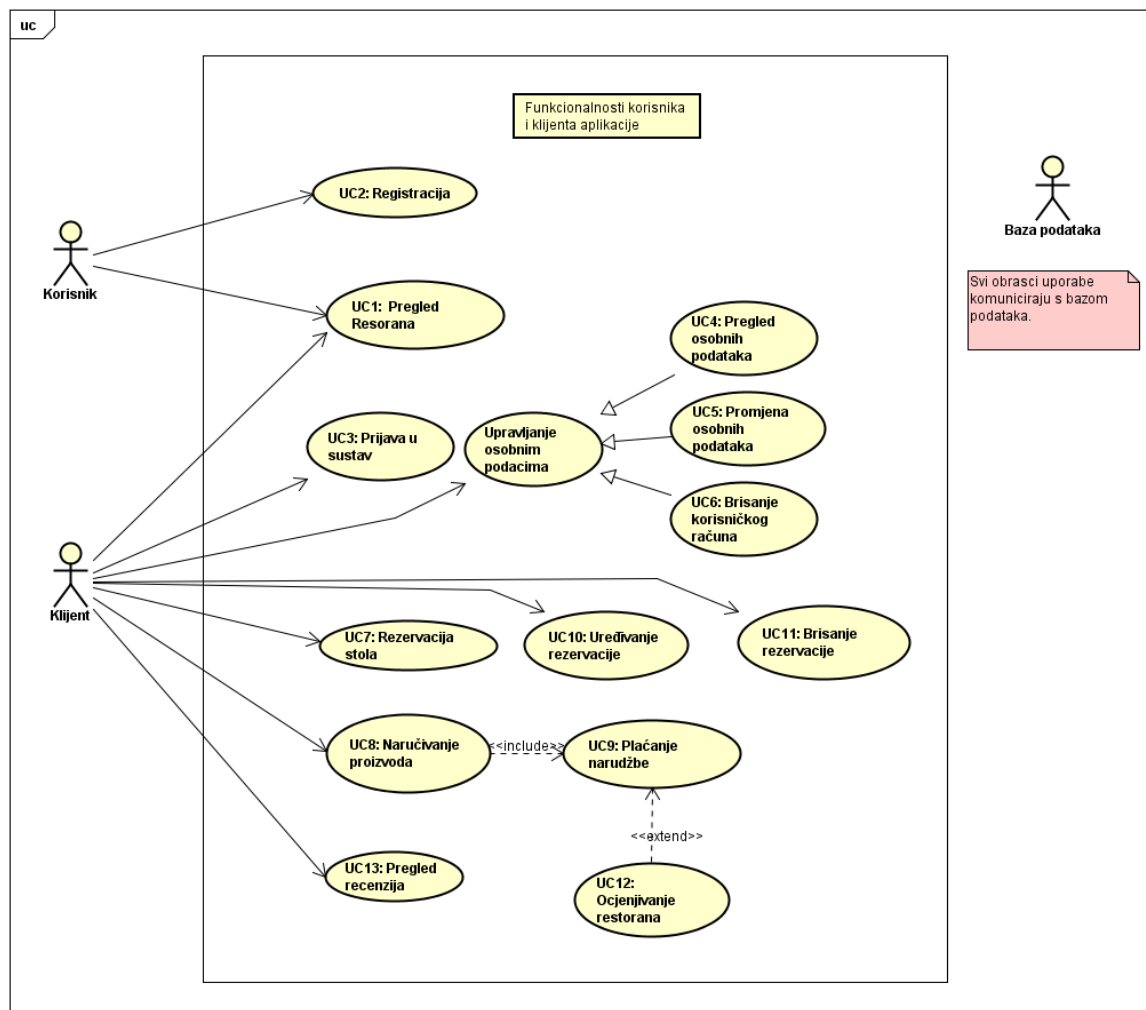
- **Glavni sudionik:** Administrator, vlasnik
- **Cilj:** Dodati restoran
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora ili vlasnika
- **Opis osnovnog tijeka:**
 1. Unosi se minimalni skup podataka o restoranu
 2. Administrator ili vlasnik odabiru opciju "Spremi"
 3. Restoran postaje vidljiv unutar aplikacije

UC35 - Brisanje restorana

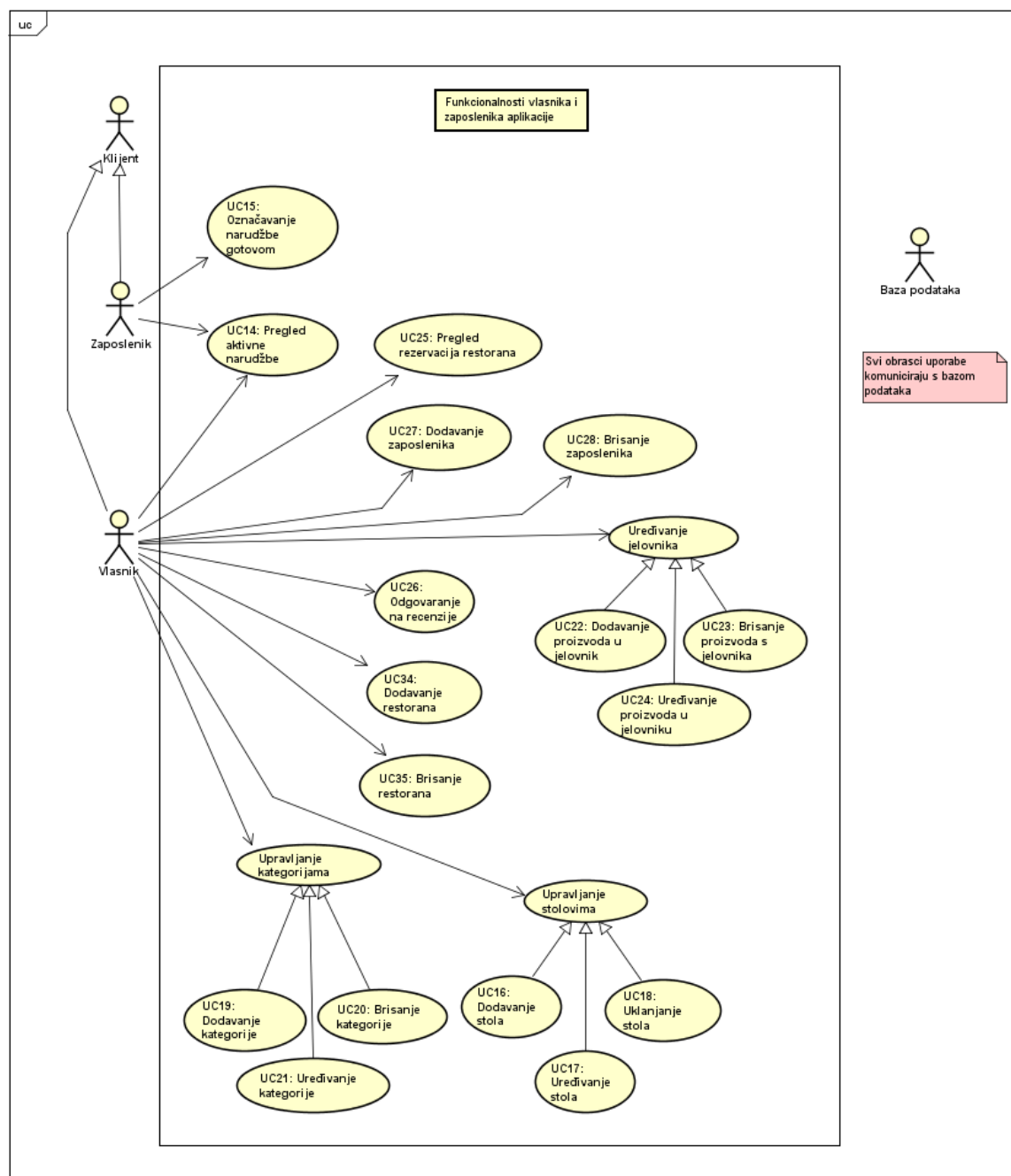
- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati restoran

- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je registriran i dodijeljena su mu prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator u aplikaciji odabiru opciju "Obriši restoran"
 2. Prikaže se lista restorana koji se mogu obrisati
 3. Administrator odabire "Obriši restoran"
 4. Restoran se uklanja iz baze podataka

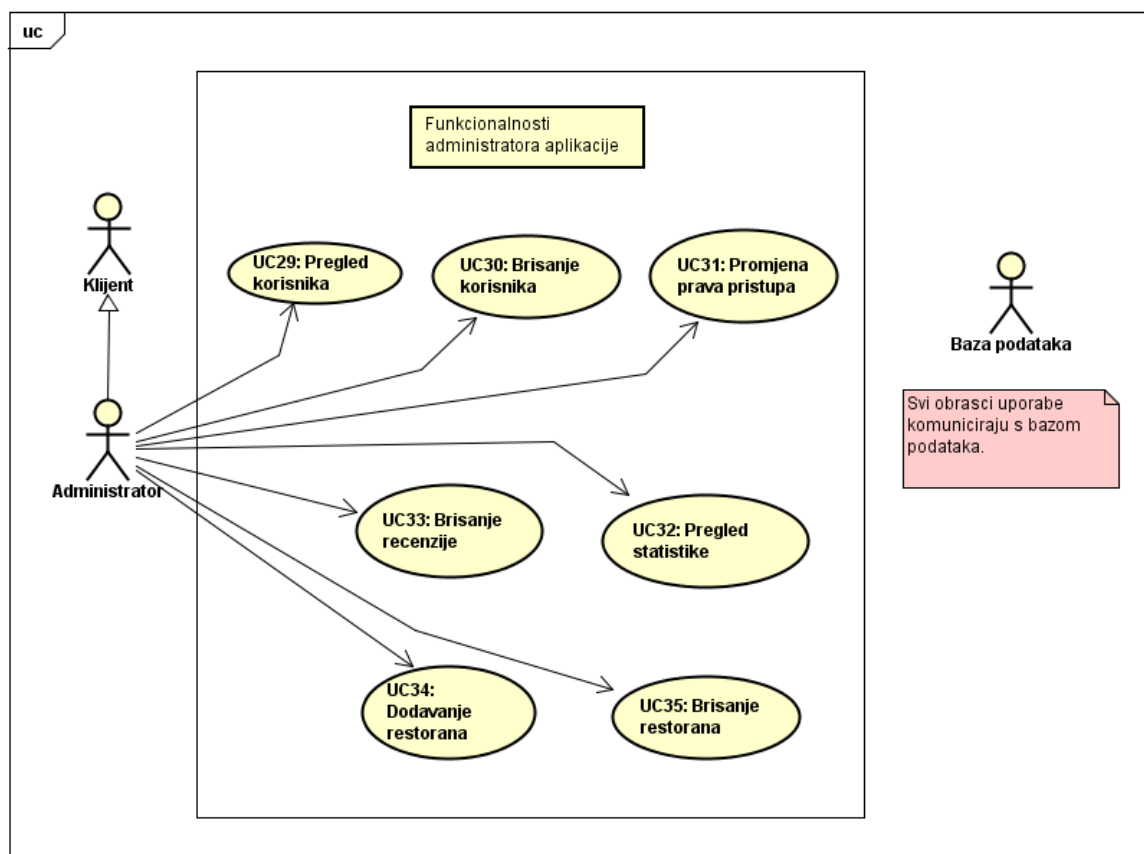
Dijagrami obrazaca uporabe



Slika 3.1: Dijagram obrasca uporabe, funkcionalnost korisnika i klijenta



Slika 3.2: Dijagram obrasca uporabe, funkcionalnost vlasnika i zaposlenika

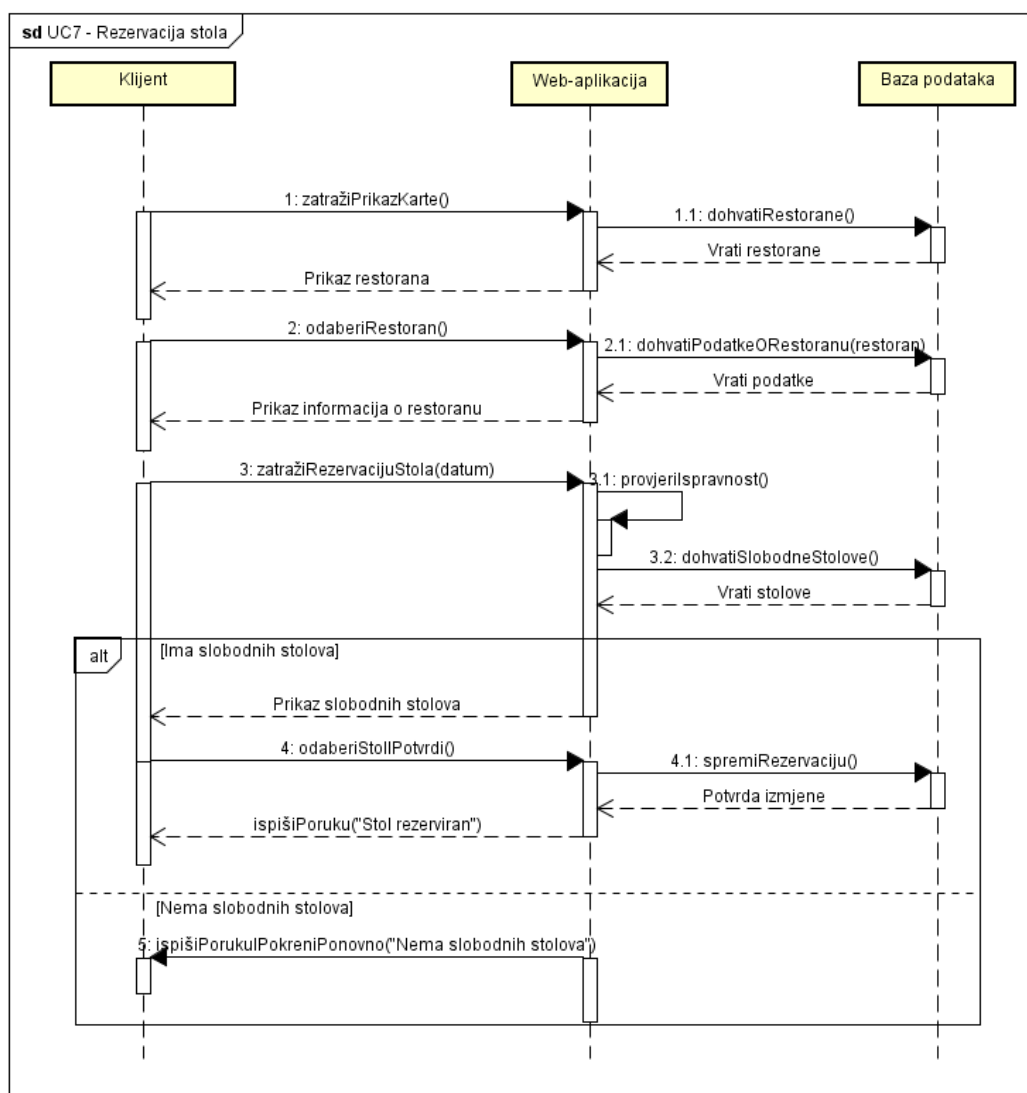


Slika 3.3: Dijagram obrasca uporabe, funkcionalnost administratora

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC7 – Rezervacija stola

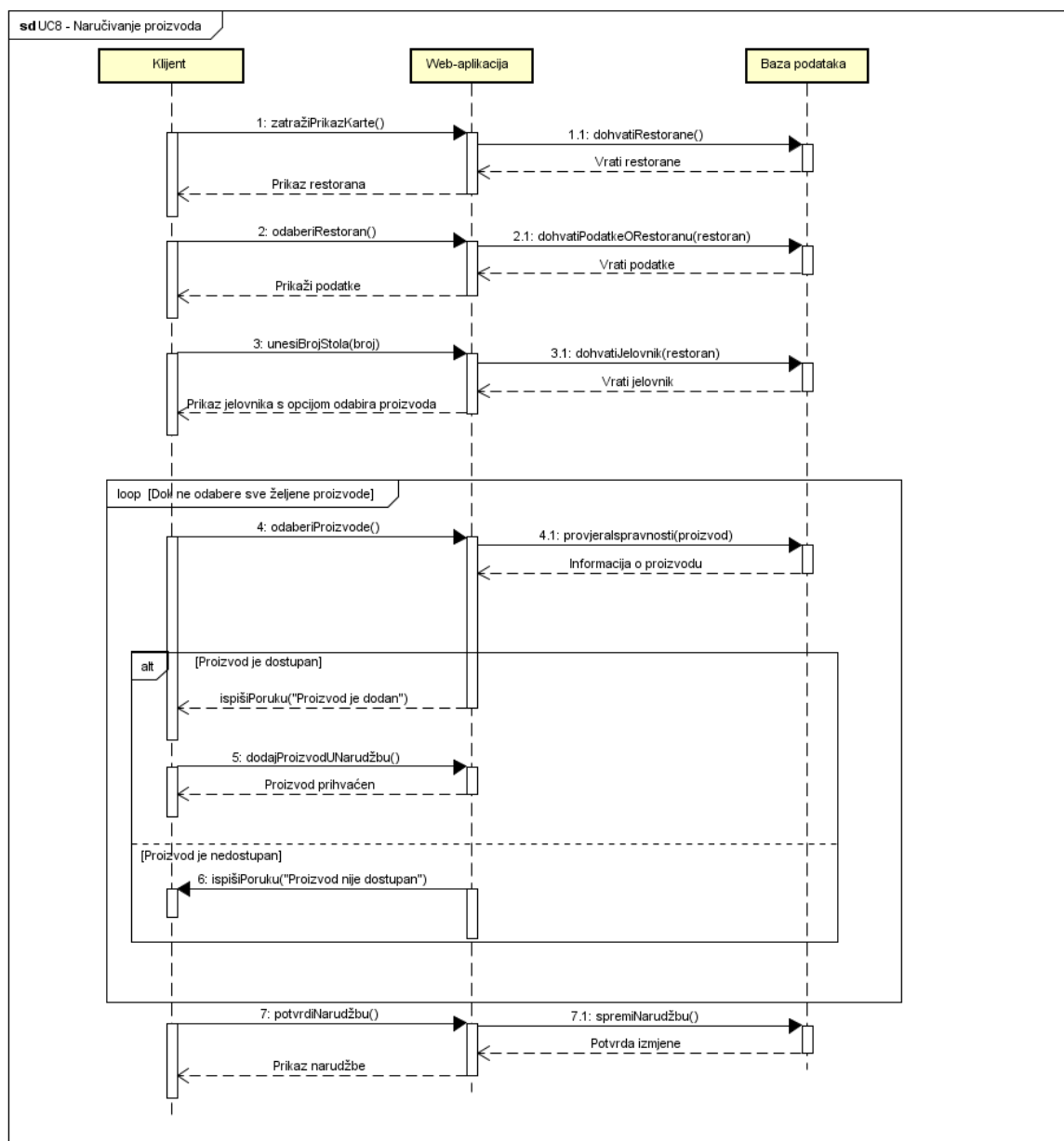
Klijent šalje zahtjev za prikaz karte s restoranima kako bi mogao odabrati restoran. Poslužitelj dohvaća trenutne restorane i prikazuje ih. Odabirom restorana, poslužitelj iz baze podataka dohvaća osnovne podatke o restoranu i prikazuje ih korisniku. Da bi započeo rezervaciju stola, klijent šalje zahtjev s datumom za koji želi rezervirati mjesto u odabranom restoranu. Poslužitelj provjerava ispravnost datuma te iz baze podataka dohvaća sve slobodne stolove za taj dan u restoranu. Ukoliko niti jedan stol nije slobodan, sustav obavještava klijenta o tome uz poruku. Ako postoji slobodan stol, proces rezervacije se nastavlja te klijent odabire jedan ili više stolova. Poslužitelj tu informaciju proslijeđuje bazi koja sprema promjenu.



Slika 3.4: Sekvencijski dijagram za UC7

Obrazac uporabe UC8 – Naručivanje proizvoda

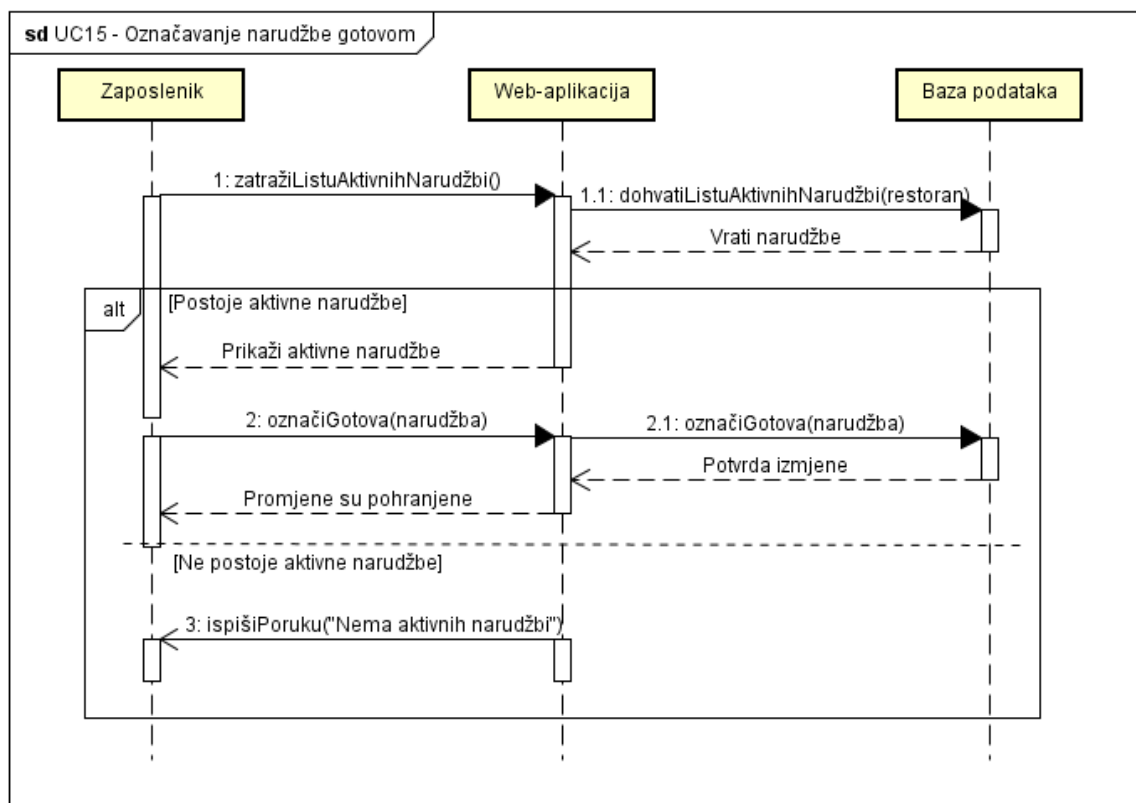
Klijent šalje zahtjev za prikaz karte s restoranima kako bi mogao odabrati restoran. Poslužitelj dohvaća trenutne restorane i prikazuje ih. Odabirom restorana, poslužitelj iz baze podataka dohvaća osnovne podatke o restoranu i prikazuje ih klijentu. Naručivanje započinje unosom broja stola za kojim sjedi, a zatim poslužitelj iz baze podataka dohvaća jelovnik odabranoga restorana. Dok ne potvrdi narudžbu, klijent ima pravo odabirati proizvode s jelovnika i dodavati ih u narudžbu. Prije nego se prihvate u narudžbu, poslužitelj s bazom podataka provjerava je li taj proizvod još dostupan. Ukoliko je, proizvod se dodaje, a ukoliko nije, ispiše se poruka i nastavlja se dalje sa biranjem.



Slika 3.5: Sekvencijski dijagram za UC8

Obrazac uporabe UC15 - Označavanje narudžbe gotovom

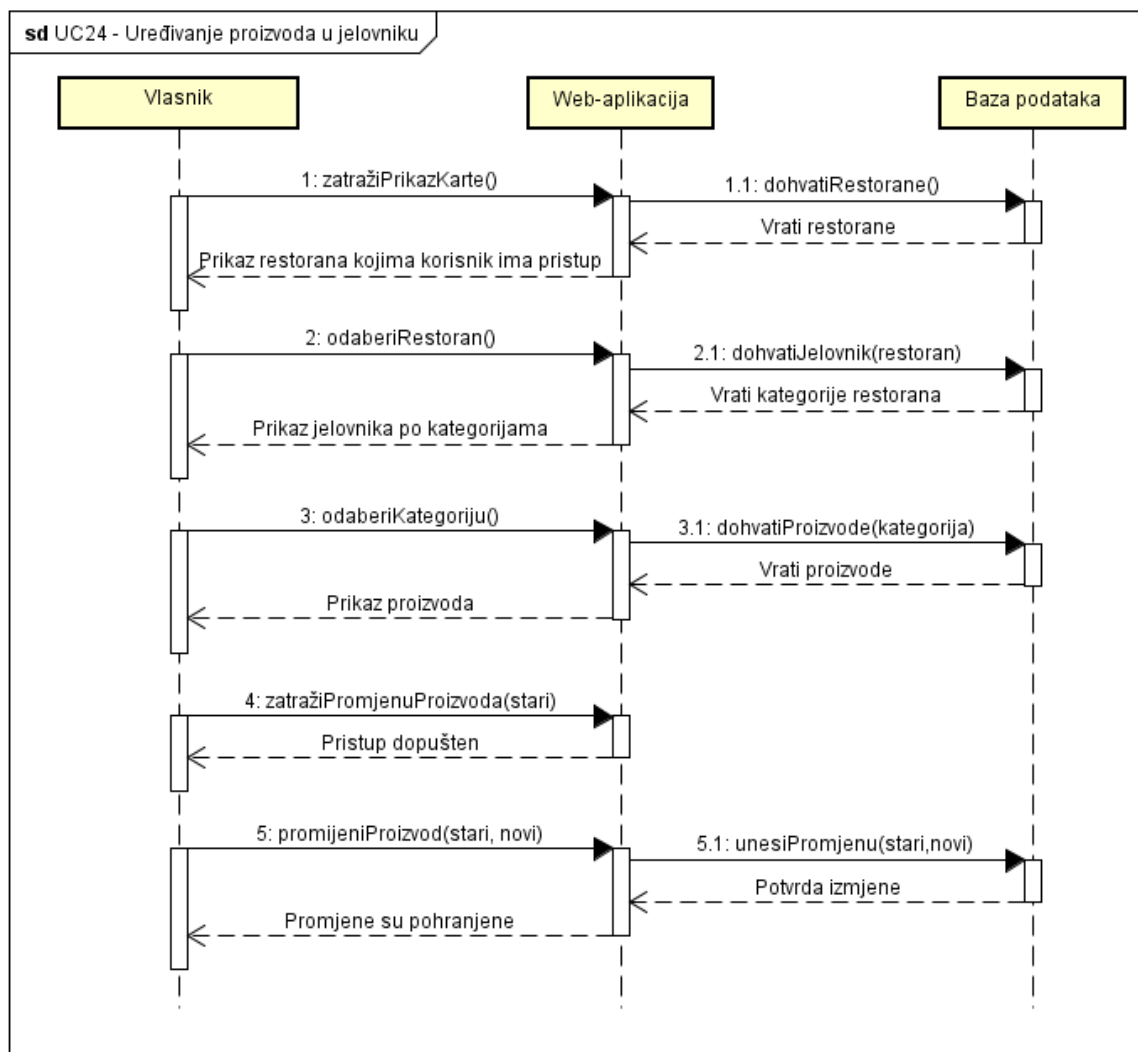
Zaposlenik šalje poslužitelju zahtjev za prikaz liste aktivnih narudžbi. Poslužitelj dohvaća listu aktivnih narudžbi iz baze podataka i ukoliko postoji barem jedna aktivna narudžba vraća je zaposleniku. Zaposlenik odabire narudžbu koju želi označiti kao gotovu. Poslužitelj prima taj zahtjev i prosljeđuje promjenu u bazu podataka. Ukoliko nema aktivnih narudžbi sustav obavještava korisnika o tome uz poruku.



Slika 3.6: Sekvencijski dijagram za UC15

Obrazac uporabe UC24 – Uređivanje proizvoda u jelovniku

Vlasnik odabire opciju "Prikaz karte". Poslužitelj dohvaća popis restorana iz baze podataka te ih prikazuje vlasniku na karti. Vlasnik odabire restoran. Poslužitelj dohvaća jelovnik restorana iz baze podataka te ga prikazuje vlasniku, a zatim isto napravi i za odabir kategorije. Vlasnik šalje zahtjev za promjenu proizvoda. Poslužitelj mu daje pristup i vlasnik unosi nove podatke o željenom proizvodu. Poslužitelj izmijeni proizvod u bazi podataka koja vraća potvrdu.



Slika 3.7: Sekvencijski dijagram za UC24

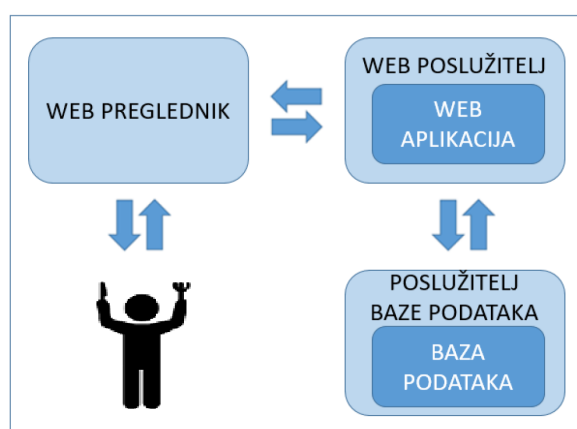
3.2 Ostali zahtjevi

- Sustav treba omogućiti rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu (dijakritičke znakove) pri unosu i prikazu tekstualnog sadržaja
- Izvršavanje dijela programa u kojem se pristupa bazi podataka ne smije trajati duže od nekoliko sekundi
- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- Neispravno korištenje korisničkog sučelja ne smije narušiti funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje, korisnici se moraju znati koristiti sučeljem bez opširnih uputa
- Nadogradnja sustava ne smije narušavati postojeće funkcionalnosti sustava
- Sustav kao valutu koristi HRK
- Veza s bazom podataka mora biti kvalitetno zaštićena, brza i otporna na vanjske greške
- Pristup sustavu mora biti omogućen iz javne mreže pomoću HTTPS.

4. Arhitektura i dizajn sustava

Arhitektura se može podijeliti na tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka



Slika 4.1: Arhitektura sustava

Web preglednik je program koji korisniku omogućuje pregled web-stranica i multimedijalnih sadržaja vezanih uz njih. Svaki internetski preglednik je prevoditelj. Dakle, stranica je pisana u kodu koji preglednik nakon toga interpretira kao nešto svakome razumljivo. Korisnik putem web preglednika šalje zahtjev web poslužitelju.

Web poslužitelj osnova je rada web aplikacije. Njegova primarna zadaća je komunikacija klijenta s aplikacijom. Komunikacija se odvija preko HTTP (engl. *Hyper Text Transfer Protocol*) protokola, što je protokol u prijenosu informacija na webu. Poslužitelj je onaj koji pokreće web aplikaciju te joj prosljeđuje zahtjev.

Korisnik koristi web aplikaciju za obrađivanje željenih zahtjeva. Web aplikacija obrađuje zahtjev te ovisno o zahtjevu, pristupa bazi podataka nakon čega preko poslužitelja vraća korisniku odgovor u obliku HTML dokumenta vidljivog u web pregledniku.

Programski jezik kojeg smo odabrali za izradu naše web aplikacije je C# zajedno s .NET radnim okvirom te programski jezik JavaScript. Odabrano razvojno

okruženje je Microsoft visual studio. Arhitektura sustava temeljiti će se na MVC (Model-View-Controller) konceptu. MVC koncept podržan je od strane .NET radnog okvira i kao takav ima gotove predloške koji nam olakšavaju razvoj web aplikacije.

Karakteristika MVC koncepta je nezavisan razvoj pojedinih dijelova aplikacije što za posljedicu ima jednostavnije ispitivanje kao i jednostavno razvijanje i dodavanje novih svojstava u sustav.

MVC koncept sastoji se od:

- **Model** - Središnja komponenta sustava. Predstavlja dinamičke strukture podataka, neovisne o korisničkom sučelju. Izravno upravlja podacima, logikom i pravilima aplikacije. Također prima ulazne podatke od Controllera
- **View** - Bilo kakav prikaz podataka, poput grafa. Mogući su različiti prikazi iste informacije poput grafičkog ili tabličnog prikaza podataka.
- **Controller** - Prima ulaze i prilagođava ih za prosljeđivanje Modelu ili Viewu. Upravlja korisničkim zahtjevima i temeljem njih izvodi daljnju interakciju s ostalim elementima sustava.

4.1 Baza podataka

Za potrebe našeg sustava koristit ćemo relacijsku bazu podataka koja svojom strukturom olakšava modeliranje stvarnog svijeta. Gradivna jedinka baze je relacija, odnosno tablica koja je definirana svojim imenom i skupom atributa. Zadaća baze podataka je brza i jednostavna pohrana, izmjena i dohvat podataka za daljnju obradu. Baza podataka ove aplikacije sastoji se od sljedećih entiteta:

- Korisnik
- Rezervacija
- Osvrt
- Narudžba
- Stol
- Restoran
- Kategorija
- Jelo
- Pripada-Restoranu

Opis tablica

Korisnik Ovaj entitet sadržava sve važne informacije o korisniku aplikacije. Sadrži attribute: Korisničko ime, lozinku, ime, prezime, broj mobitela korisnika, broj kartice za plaćanje, razinu ovlasti korisnika i njegov OIB. Ovaj entitet u vezi je *One-to-Many* s entitetom Narudžba preko atributa korisničko ime korisnika, u vezi *One-to-One* s Pripada-Restoranu preko korisničkog imena te u vezi *One-to-Many* s entitetom Rezervacija preko korisničkog imena i u vezi *One-to-Many* s entitetom Osvrt preko korisničkog imena.

Korisnik		
Korisničko ime	VARCHAR	jedinstveni identifikator korisnika
Lozinka	VARCHAR	hash lozinke
Ime	VARCHAR	ime korisnika
Prezime	VARCHAR	prezime korisnika
Broj mobitela	VARCHAR	telefonski broj korisnika
Email	VARCHAR	e-mail adresa korisnika
Broj kartice	INT	broj kartice korisnika

Korisnik		
Razina ovlasti	INT	razina ovlasti korisnika

Rezervacija Ovaj entitet sadržava sve važne informacije vezane za rezervaciju stola u restoranu. Sadrži attribute: Vrijeme rezervacije, šifru stola, šifru korisnika koji je rezervirao stol i broj rezervacije. Ovaj entitet u vezi je *Many-to-One* s entitetom Korisnik preko korisničkog imena i u vezi *One-to-Many* s entitetom Stol preko Šifre stola.

Rezervacija		
Vrijeme	DATETIME	vrijeme rezervacije
Šifra stola	INT	šifra stola, (stol.šifra)
Korisničko ime	VARCHAR	jedinstveni identifikator korisnika, (korisnik.korisničko ime)
Broj rezervacije	INT	jedinstveni brožčani identifikator

Osvrt Ovaj entitet sadržava sve važne informacije vezane za osvrt korisnika o pojedinom restoranu. Sadrži attribute: ID osvrta, ocjenu restorana, opis usluga, šifru korisnika i OIB restorana kojeg ocjenjuje. Ovaj entitet u vezi je *Many-to-One* s entitetom Korisnik preko korisničkog imena i u vezi *Many-to-One* s entitetom Restoran preko OIB-a restorana i odgovora na osvrt.

Osvrt		
ID	INT	jedinstveni brožčani identifikator
Ocjena	INT	ocjena za restoran
Opis	VARCHAR	osvrt na restoran i proizvode
Korisničko ime	VARCHAR	jedinstveni identifikator korisnika, (korisnik.korisničko ime)
OIB restorana	INT	OIB restorana, (restoran.OIB)
Odgovor	VARCHAR	odgovor na recenziju

Narudžba Ovaj entitet sadržava sve važne informacije vezane za narudžbu jela korisnika. Sadrži attribute: ID narudžbe, naziv jela kojeg naručuje, vrijeme narudžbe, korisničko ime korisnika koji naručuje i gotovo što prikazuje status narudžbe. Ovaj entitet u vezi je *Many-to-One* s entitetom Korisnik preko korisničkog

imena i u vezi *Many-to-One* s entitetom Jelo preko ID jela.

Narudžba		
ID	INT	jedinstveni brojčani identifikator
Vrijeme	TIME	vrijeme naručivanja
Korisničko ime	VARCHAR	jedinstveni identifikator korisnika, (korisnik.korisničko ime)
ID jela	INT	jedinstveni identifikator jela, (jelo.ID)
Gotova	BOOLEAN	oznaka je li narudžba završena

Stol Ovaj entitet sadržava sve važne informacije o stolovima u restoranu. Sadrži attribute: Šifra stola(id), OIB restorana kojem stol pripada i kapacitet osoba za tim stolom. Ovaj entitet u vezi je *Many-to-One* s entitetom Restoran preko OIB-a restorana i u vezi *Many-to-One* s entitetom Rezervacija preko šifre stola.

Stol		
Šifra	INT	jedinstveni brojčani identifikator
OIB restorana	INT	restorana kojem stol pripada (restoran.OIB)
Kapacitet	INT	kapacitet stola

Restoran Ovaj entitet sadržava sve važne informacije o restoranu. Sadrži attribute: ime restorana, adresa, telefon, fax, OIB restorana, žiro račun za uplatu i IBAN. Ovaj entitet u vezi *One-to-Many* s entitetima Stol, Osvrt preko OIB-a restorana, u vezi *One-to-Many* s Pripada Restoranu preko OIB-a restorana i u vezi *Many-to-Many* s Kategorija preko OIB-a restorana.

Restoran		
Ime	VARCHAR	ime restorana
Adresa	VARCHAR	adresa restorana
Telefon	VARCHAR	telefonski broj restorana
Fax	VARCHAR	fax restorana
OIB restorana	INT	OIB restorana
Iban	VARCHAR	Iban restorana
Žiro račun	VARCHAR	žiro račun restorana
Slika	LONGBLOB	slika restorana

Kategorija Ovaj entitet sadržava sve važne informacije o kategorijama jela koje korisnik želi izabrati. Sadrži attribute: ID kategorije, naziv kategorije i OIB restorana. Ovaj entitet u vezi je *Many-to-Many* s Restoran preko OIB-a restorana te *One-to-Many* s Jelo preko ID kategorije.

Kategorija		
ID	INT	jedinstveni brožčani identifikator
Naziv	VARCHAR	naziv kategorije
OIB restorana	INT	OIB restorana kojem kategorija pripada (restoran.OIB)

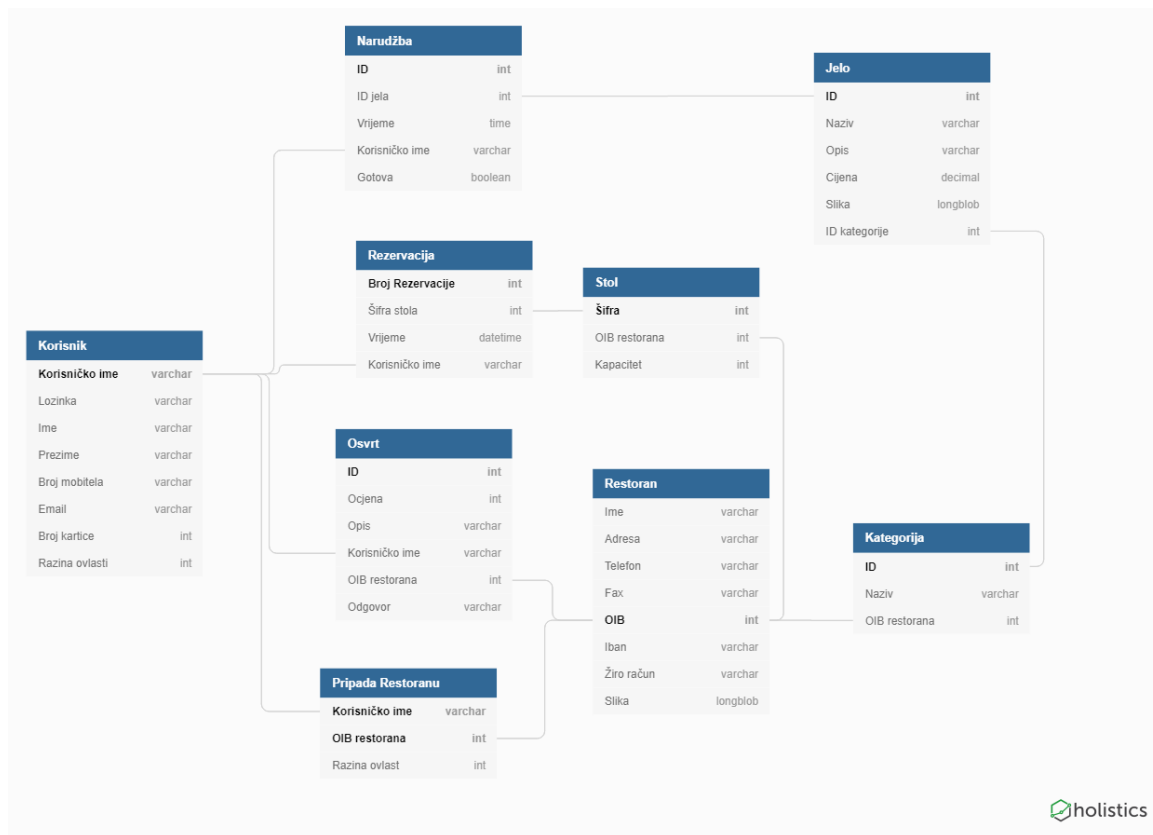
Jelo Ovaj entitet sadržava sve važne informacije o jelima u restoranima. Sadrži attribute: Naziv jela, ID jela, opis jela, slika jela, cijena jela i naziv kategorije. Ovaj entitet u vezi je *Many-to-One* s Kategorija preko naziva jela i u vezi *One-to-Many* s Narudžba preko ID jela.

Jelo		
ID	INT	jedinstveni brožčani identifikator
Naziv	VARCHAR	naziv jela
Opis	VARCHAR	sastojci, upozorenja na alergene
Cijena	DECIMAL	cijena jela
ID kategorije	INT	ID kategorije, (kategorija.ID)
Slika	LOB	slika jela

Pripada Restoranu Ovaj entitet sadržava sve informacije o odnosu korisnika i restorana odnosno kojem restoranu korisnik pripada i koja je njegova razina ovlasti. Sadrži attribute: Korisničko ime, OIB restorana i razina ovlasti korisnika. Ovaj entitet u vezi je *Many-to-One* s Restoranom preko naziva OIB-a restorana.

Pripada Restoranu		
Korisničko ime	VARCHAR	korisničko ime vlasnika ili zaposlenika restorana, (korisnik.korisničko ime)
OIB restorana	INT	OIB restorana, (restoran.OIB)
Razina ovlasti	INT	razina ovlasti korisnika

Dijagram baze podataka

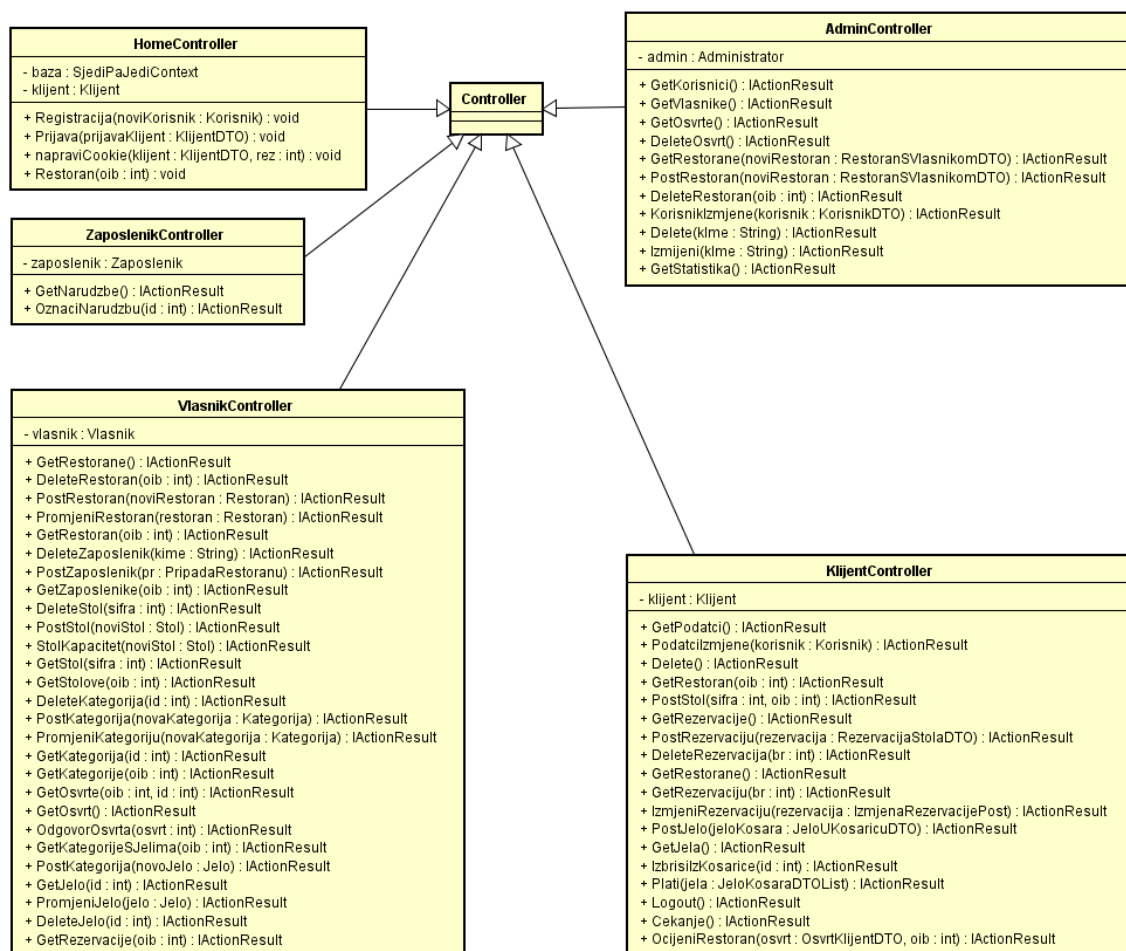


Slika 4.2: E-R dijagram baze podataka

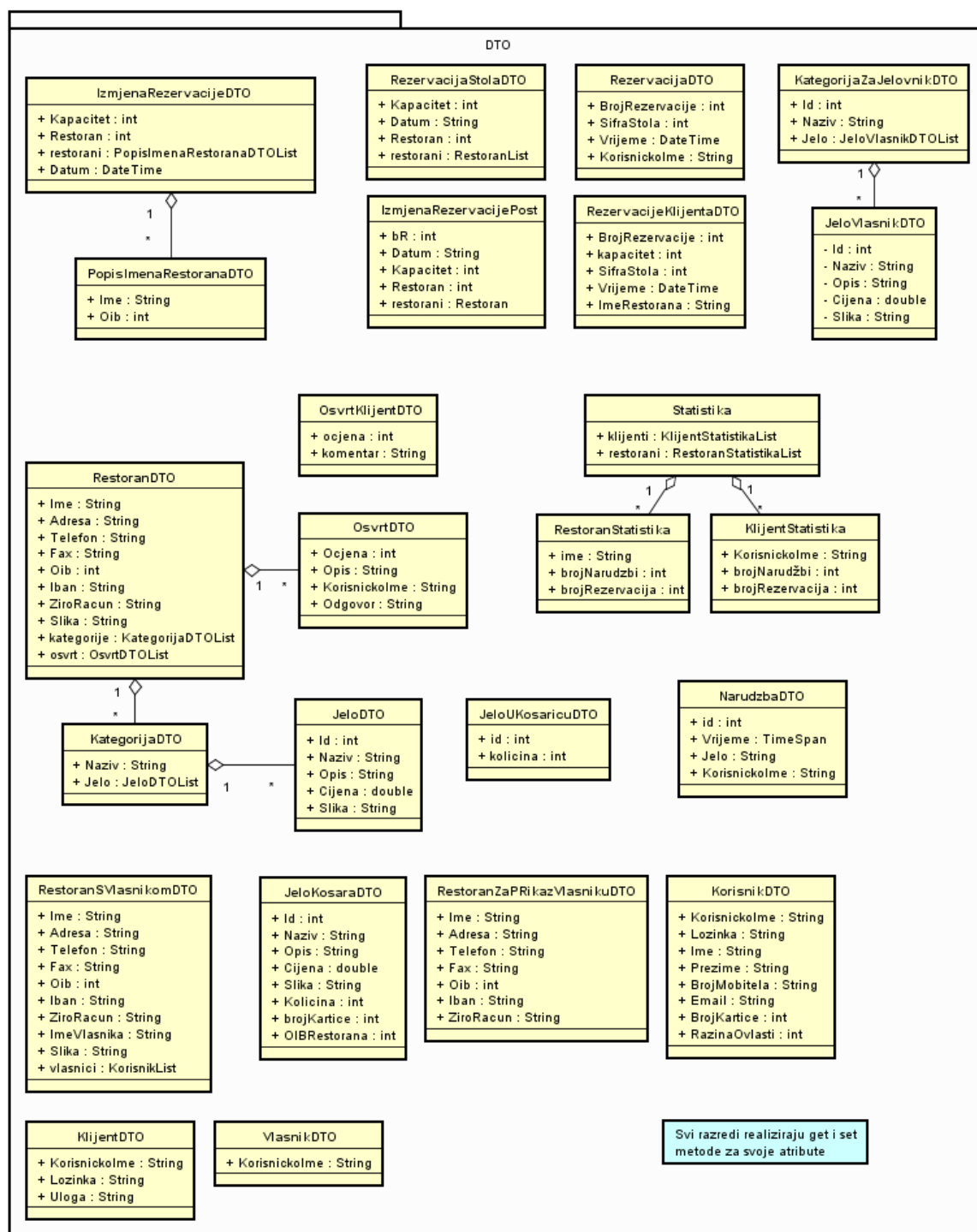
4.2 Dijagram razreda

Na slikama 4.3, 4.4 i 4.5 su prikazani razredi koji pripadaju *backend* dijelu MVC arhitekture. Razredi prikazani na slici 4.3 nasljeđuju Controller razred. Metode implementirane u tim razredima manipuliraju s DTO (*Data transfer object*), a oni su dohvaćeni pomoću metoda implementiranih u Model razredima. Metode implementirane u Controller razredima vraćaju JSON datoteke s html status kodom.

Zbog lakše organizacije, razredi su podijeljeni logički po pravu pristupa metodama određenih aktera, kako bi se smanjila prenapučenost unutar dijagrama, prikazane su samo ovisnosti između razreda koji pripadaju istom dijelu dijagrama. Iz naziva i tipova atributa u razredima može se zaključiti vrsta ovisnosti među različitim razredima.



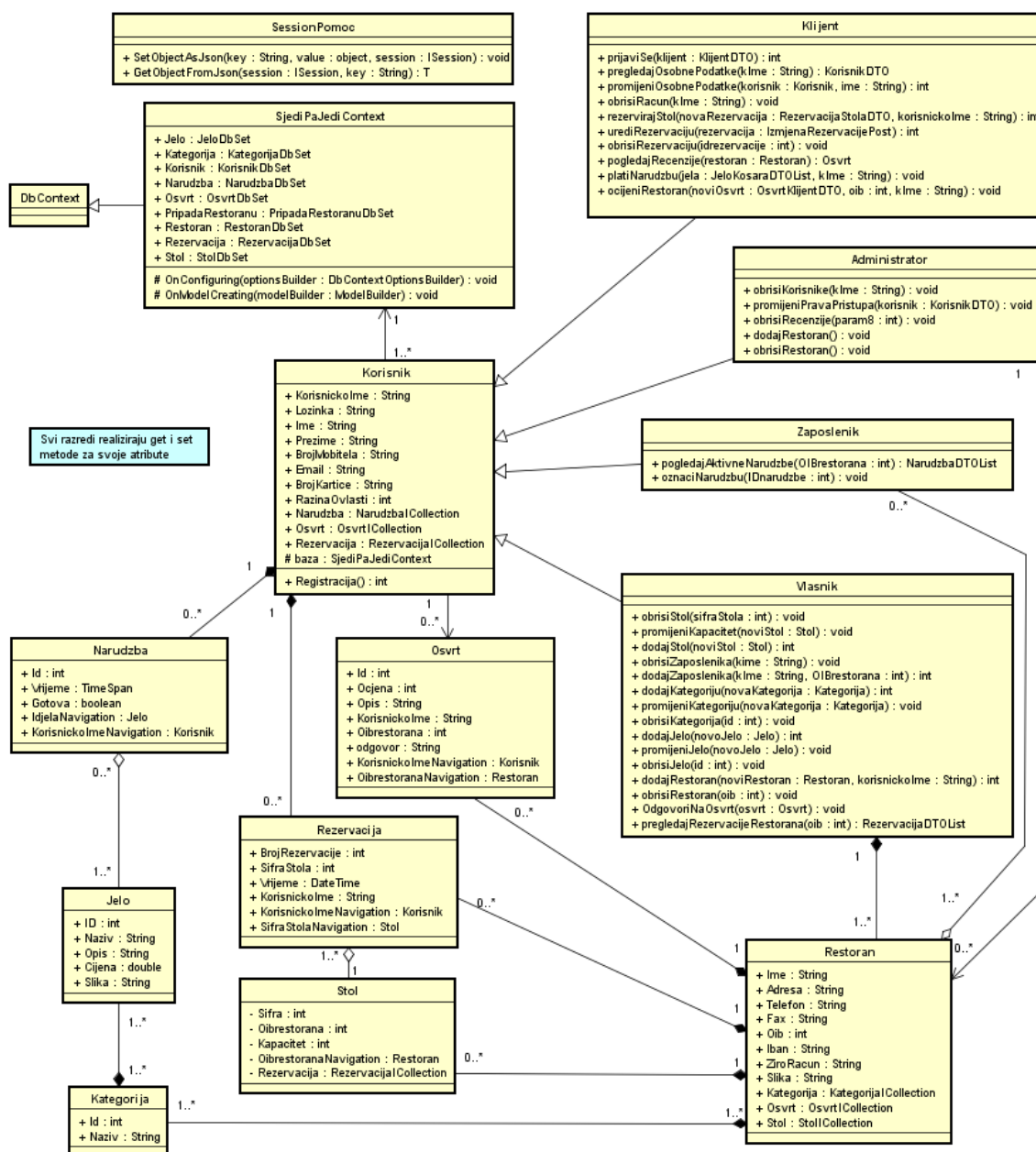
Slika 4.3: Dijagram razreda - dio Controllers



Slika 4.4: Dijagram razreda - dio Data transfer objects

Model razredi preslikavaju strukturu baze podataka u aplikaciji. Implementirane metode direktno komuniciraju s bazom podataka te vraćaju tražene podatke. Razred Korisnik predstavlja neregistriranog korisnika koji se može registrirati u

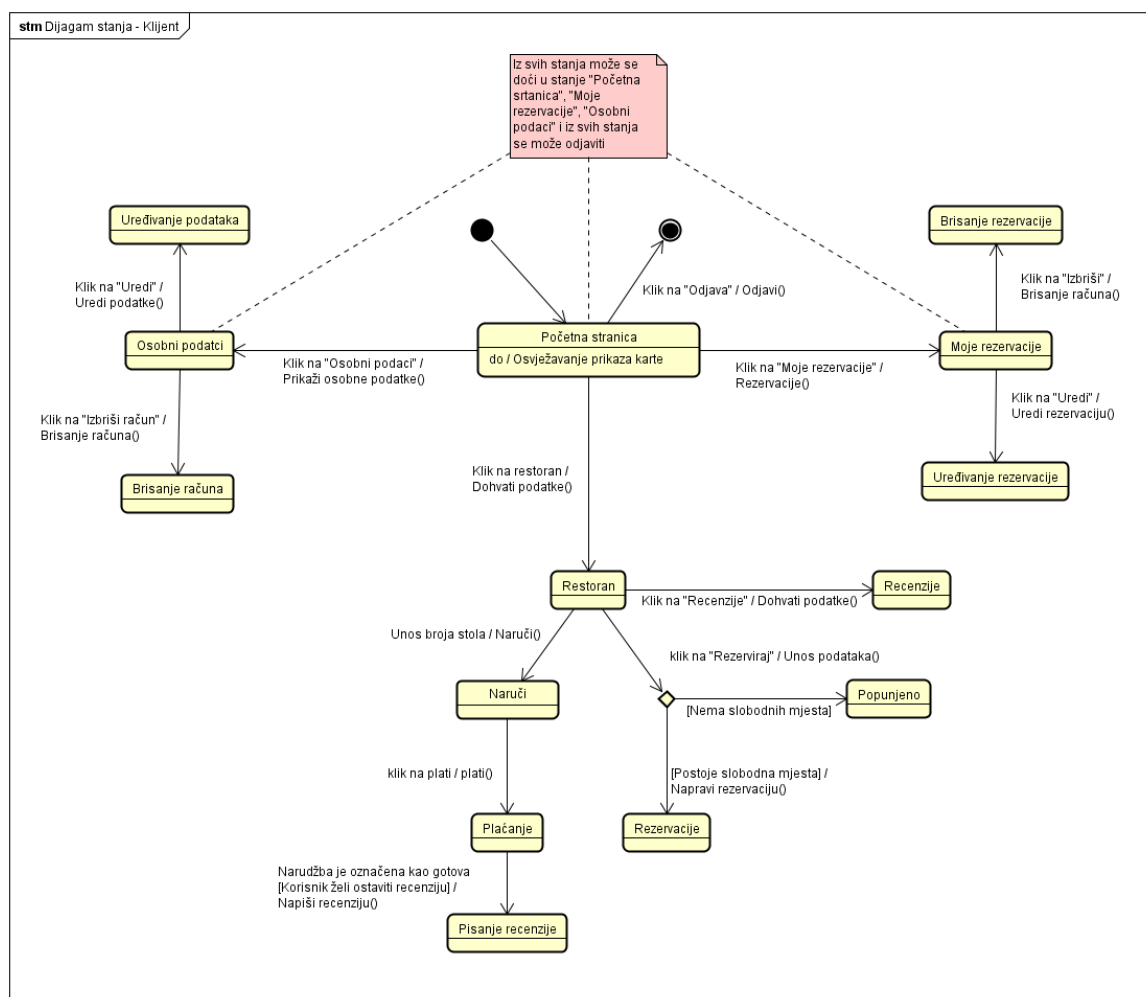
sustav unoseći osnovne informacije. Razred Klijent predstavlja klijenta koji je registriran u sustav i koji može koristiti njegove osnovne funkcionalnosti. Razred Administrator predstavlja administratora sustava koji ima najveće ovlasti. Razred Vlasnik predstavlja vlasnika jednog ili više restorana koji ima mogućnost upravljanja restoranima i zaposlenicima. Razred Zaposlenik predstavlja radnika u restoranu koji ima pristup trenutno aktivnim narudžbama. Razred Restoran predstavlja skup podataka koji su potrebni za registraciju restorana i koji se prikazuju korisniku i klijentu.



Slika 4.5: Dijagram razreda - dio Models

4.3 Dijagram stanja

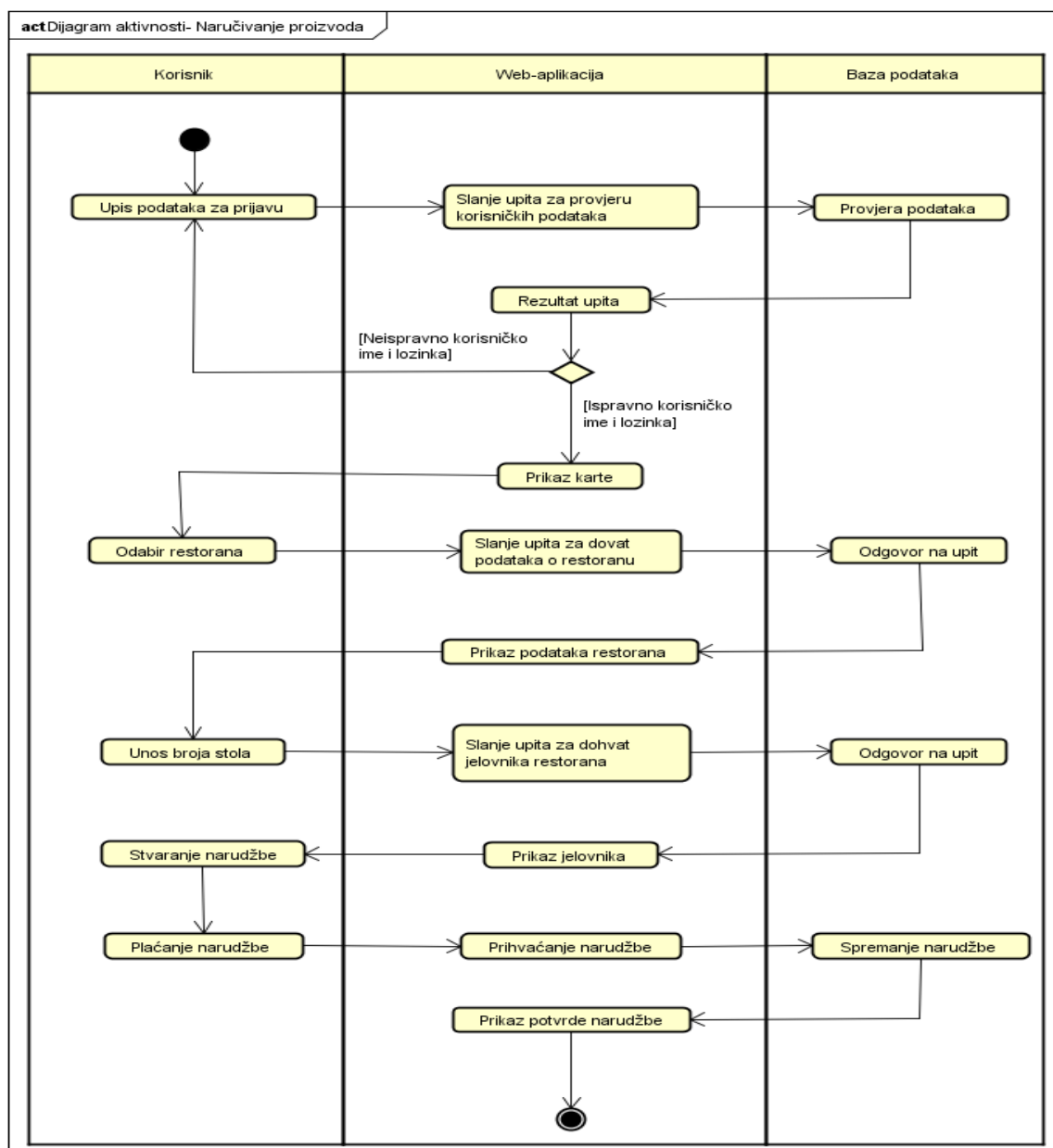
Dijagram stanja prikazuje stanja objekta te prijelaze iz jednog stanja u drugo temeljene na događajima. Na slici 4.6 prikazan je dijagram stanja za registriranog korisnika. Nakon prijave, klijentu se prikazuje početna stranica na kojoj može pregledati kartu i restorane. Za odabrani restoran ima opciju pregledavanja recenzija i ocjena restorana te opciju za stvaranje narudžbe (aktivira se unosom broja stola). Nakon plaćanja, kada je narudžba označena gotovom, korisniku se aktivira opcija pisanja osvrta na restoran. Također, klijent može u padajućem izborniku odabrati neku od ponuđenih mogućnosti. Klikom na "Osobni podatci" prikazuju mu se njegovi podatci, koje može urediti ili odabrati opciju brisanja račun. Odabirom "Moje rezervacije" otvara mu se popis rezervacije koje može urediti ili obrisati.



Slika 4.6: Dijagram stanja

4.4 Dijagram aktivnosti

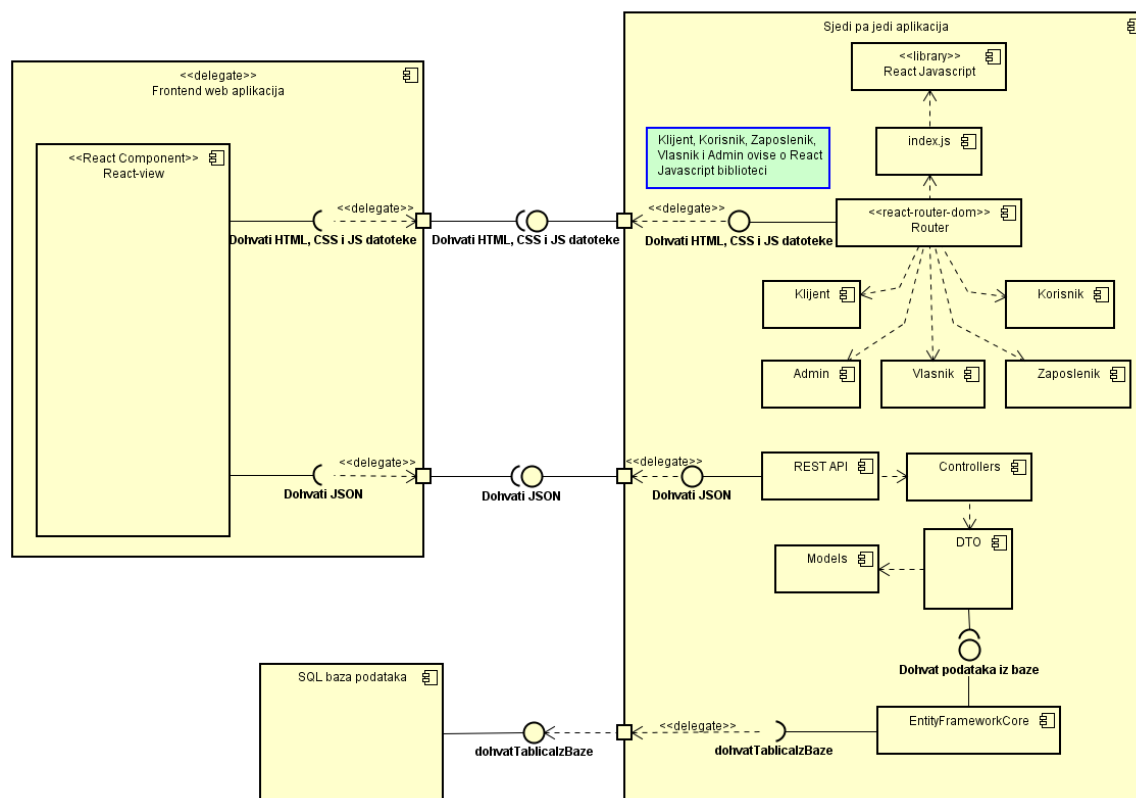
Dijagram aktivnosti primjenjuje se za opis modela toka upravljanja ili toka podataka. Ne upotrebljava se za modeliranje događajima poticanog ponašanja. U modeliranju toka upravljanja svaki novi korak poduzima se nakon završenog prethodnog, a naglasak je na jednostavnosti. Na dijagramu aktivnosti 4.7 prikazan je proces kreiranja narudžbe. Korisnik se prijavi u sustav, odabere restoran u kojem je trenutno te unosi broj stola za kojim sjedi. Zatim mu se prikazuje jelovnik restorana preko kojeg on odabire proizvode koje želi naručiti. Kada je zadovoljan svojom narudžbom, odabire plaćanje te narudžba postaje aktivna u sustavu.



Slika 4.7: Dijagram aktivnosti

4.5 Dijagram komponenti

Dijagram komponenti prikazan na slici 4.8 opisuje organizaciju i međuovisnost komponenti, interne strukture i odnose prema okolini. Sustavu se pristupa preko dva različita sučelja. Preko sučelja za dohvat HTML, CSS i JS datoteka poslužuju se datoteke koje pripadaju *frontend* dijelu aplikacije. Router je komponenta koja na upit s url određuje koja datoteka će se poslužiti na sučelje. *Frontend* dio se sastoji od niza JavaScript datoteka koje su raspoređene u logičke cjeline nazvane po tipovima aktera koji im pristupaju. Sve JavaScript datoteke ovise o React biblioteci iz koje dohvaćaju gotove komponente kao što su gumbi, forme i slično. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju *backend* dijelu aplikacije. EntityFrameworkCore je zadužen za dohvaćanje tablica iz baze podataka pomoću SQL upita. Podaci koji su pristigli iz baze se šalju dalje MVC arhitekturi u obliku DTO (*Data transfer object*). React-view komponenta preko dostupnih sučelja komunicira sa sjedi pa jedi aplikacijom te ovisno o korisnikovim akcijama osvježava prikaz i dohvaća nove podatke ili datoteke.



Slika 4.8: Dijagram komponenti

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

Komunikacija u timu realizirana je korištenjem aplikacije WhatsApp¹. Za izradu UML dijagrama korišten je alat Astah Professional², a kao sustav za upravljanje izvornim kodom Git³. Udaljeni repozitorij projekta je dostupan na web platformi GitLab⁴.

Kao razvojno okruženje korišten je Microsoft Visual Studio⁵ - integrirano je razvojno okruženje (IDE) tvrtke Microsoft. Prvenstveno se koristi za razvoj računalnih programa za operacijski sustav Windows, kao i za web-stranice, web-aplikacije, web-usluge i mobilne aplikacije. Visual Studio za razvoj softvera koristi Microsoftove platforme kao što su Windows API, Windows Forms, Windows Presentation Foundation, Windows Store i Microsoft Silverlight.

Aplikacija je napisana koristeći radni okvir .NET Framework⁶ i jezik C#⁷ za izradu *backenda* te React⁸ i jezik JavaScript⁹ za izradu *frontenda*. React, također poznat kao React.js ili ReactJS, je biblioteka u JavaScriptu za izgradnju korisničkih sučelja. Održavana je od strane Facebooka. React se najčešće koristi kao osnova u razvoju web ili mobilnih aplikacija. Složene aplikacije u Reactu obično zahtijevaju korištenje dodatnih biblioteka za interakciju s API-jem. Radni okvir .NET Framework nadograđuje mogućnosti samog operativnog sustava. Radi se o posebnoj infrastrukturi koja programerima nudi gotova rješenja i funkcionalnosti da bi ubrzala i pojednostavila razvoj aplikacija svih vrsta i oblika.

Baza podataka se nalazi na poslužitelju u oblaku Microsoft Azure¹⁰.

¹<https://www.whatsapp.com/>

²<http://astah.net/editions/professional>

³<https://git-scm.com/>

⁴<https://gitlab.com/>

⁵<https://visualstudio.microsoft.com/>

⁶<https://dotnet.microsoft.com/>

⁷<https://docs.microsoft.com/en-us/dotnet/csharp/>

⁸<https://reactjs.org/>

⁹<https://www.javascript.com/>

¹⁰<https://portal.azure.com/>

5.2 Ispitivanje programskog rješenja

Napomena: Od akademske godine 2019/2020 ispitivanje se provodi na strukturirani način pomoću unit i selenium testova. Kratke upute za pisanje ovog poglavlja potražite u predlošku dokumentacije, a konkretne ispitne slučajeve koje ćete obraditi u dokumentaciji definirajte s vašim nastavnikom.

Svi testovi izvršeni su ručno. Ispitivanje se radilo po obrascima uporabe kako bi se provjerila osnovna funkcionalnost sustava, ali i nasumičnim kretanjima po aplikaciji kako bi se pronašle neočekivane greške ("bugovi") ili nepredviđena ponašanja. Svaki dio sustava je ispitan, no zbog jednostavnosti u dokumentaciji će biti prikazan samo dio ispitivanja. Prikazivanje ispitivanja UC1, UC2, UC7 i UC8 (uključujući UC9).

Ispitni slučaj 1: Interakcija s kartom i pregledavanje dostupnih restorana

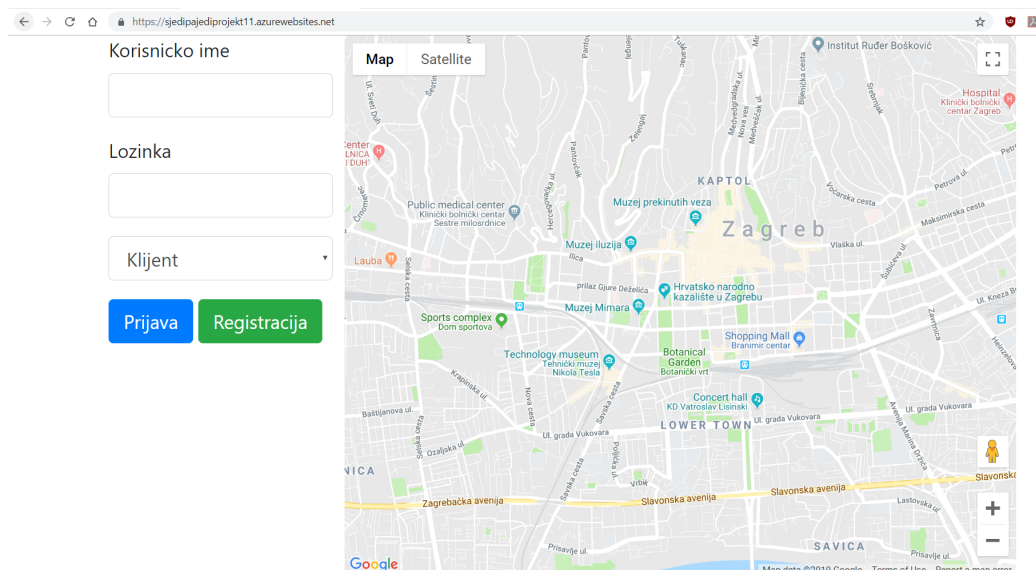
Ulaz:

1. Otvaranje početne stranice u web pregledniku.
2. Pritisak i pomicanje kursorom po karti.
3. Pomicanje kotačića na mišu dok je kursor iznad karte.

Očekivani rezultat:

- 1.a Prikazuje se karta.
- 1.b Oni restorani koji su uneseni u aplikaciju se prikazuju interaktivno na karti.
 - 2 Karta se pomiče po svakoj osi prateći pomicanje kursora.
 - 3 Karta se uvećava ili smanjuje ovisno o smjeru okretanja kotačića.

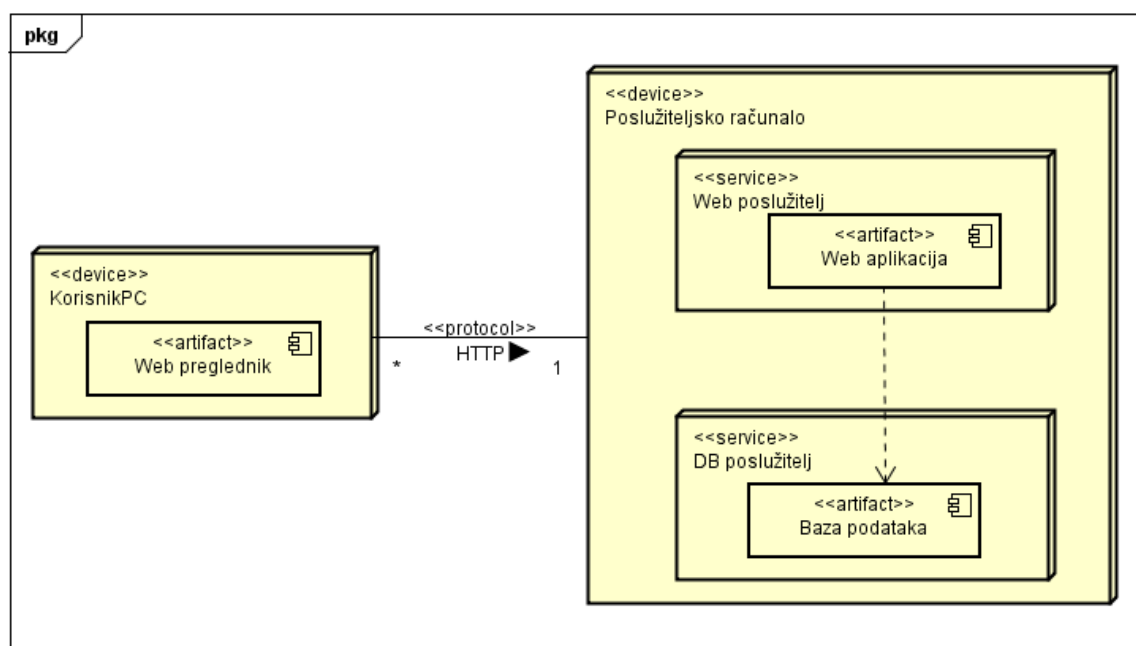
Rezultat: Očekivani rezultat [1.b] nije zadovoljen jer se prikazuje samo jedan restoran na točnoj lokaciji, iako je unutar aplikacije uneseno više restorana. Ostala očekivanja su zadovoljena. **Aplikacija nije prošla test.**



Slika 5.1: Pregledavanje restorana na karti

5.3 Dijagram razmještaja

Dijagrami razmještaja opisuju topologiju sklopovlja i programsku potporu koja se koristi u implementaciji sustava u njegovom radnom okruženju. Na poslužiteljskom računalu se nalaze web poslužitelj i poslužitelj baze podataka. Klijenti koriste web preglednik kako bi pristupili web aplikaciji. Sustav je baziran na arhitekturi "klijent – poslužitelj", a komunikacija između računala korisnika (klijent, zaposlenik, vlasnik, administrator) i poslužitelja odvija se preko HTTP veze.



Slika 5.2: Dijagram razmještaja

5.4 Upute za puštanje u pogon

Napomena: poglavlje je preuzeto iz dokumentacije drugog zadatka

Instalacija poslužitelja baze podataka

Potrebno je preuzeti Oracle MySQL bazu podataka (za operacijski sustav Windows). Poželjno je skinuti čitav paket Oracle Workbench. Nakon toga je potrebno provesti standardnu instalaciju s postavljanjem korisnika.

Konfiguracija poslužitelja baze podataka

Nakon instalacije, potrebno je pokrenuti MySQL Workbench i konfigurirati prvu konekciju na bazu. Potrebno je postaviti hostname (127.0.0.1), port (3306), korisničko ime i lozinku. Korisno je testirati konekciju pomoću tipke "Test Connection" u donjem desnom kutu formulara za postavljanje nove veze.

Server baze mora biti pokrenut. On će biti automatski pokrenut odmah nakon instalacije, a u slučaju da je ugašen moguće ga je pokrenuti iz MySQLWorkbench prema dijelu uputa za instalaciju pod nazivom "Pokretanje servera baze podataka".

U daljnjim uputama bit će opisana dva načina kako se može podignuti baza. Potrebno je provesti samo jedan od navedenih. Datoteke koje se spominju u tekstu se nalaze u repozitoriju projekta unutar direktorija Izvorni kod/ZOOService/database

Konfiguracija baze podataka

U trenutnoj konekciji na bazu odabiremo četvrtu ikonu u alatnoj traci (create schema) i nazovemo ju zoovrt te potvrdimo promjene. Odabiremo u alatnoj traci "File" - "Open Model.." i odabiremo preuzetu zooPopuliran.mwb datoteku. Zatim odabiremo "Database" - "Synchronize Model". Opcije konekcije bi trebale biti unaprijed postavljene, a ako nisu, postavljamo ih na vrijednosti korištenima pri likom podizanja servera baze podataka. Odabiremo "Next". U fazi Sync Options odabiremo "Include SQL Scripts Attached to Model", dva puta odabiremo "Next" te u fazi "Select schemata" odabiremo zoovrt model shemu. U donjem izborniku odabiremo kreiranu zoovrt shemu i odabiremo "Override Target" te potvrđujemo pritiskom na "Next". Zatim potvrđujemo do kraja formulara.

Punjenje baze podataka

Odabiremo u alatnoj traci pod "Server" - "Data Import". Otvara nam se novi tab za dodavanje podataka. Odabiremo "Import from Self-Contained File" za kojeg odaberemo preuzetu zoo.sql skriptu te bi zatim potvrđujemo izvođenje SQL skripte sa "Start Import".

Pokretanje poslužitelja baze podataka

Ako baza podataka nije pokrenuta možemo je pokrenuti iz MySQL Workbench. Odabiremo iz alatne trake "Server" - "Startup / Shutdown MySQL Server". Može se dogoditi da Workbench pretpostavi da je server podignut, no da ustvari nije i da piše "Running", a da nije uspostavljena veza na bazu podataka. U tom slučaju je potrebno zaustaviti i ponovno pokrenuti server.

Instalacija Tomcat poslužitelja

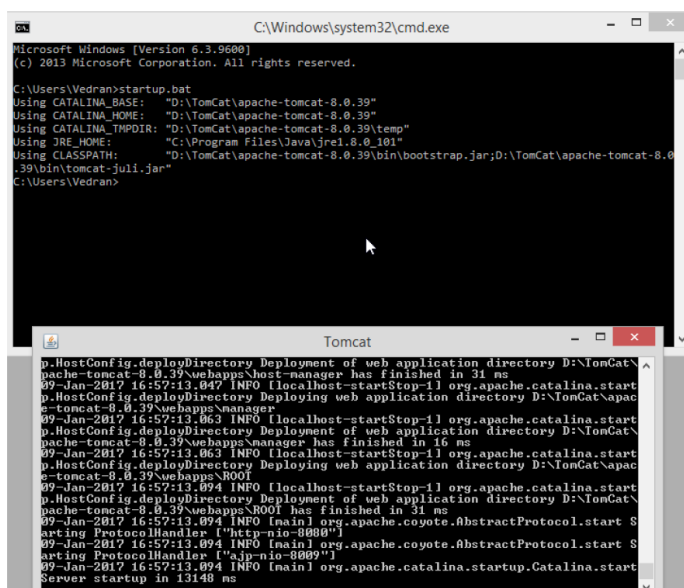
Potrebno je preuzeti Tomcat poslužitelj verzije 8.0.x koji je dostupan na <https://tomcat.apache.org/download-80.cgi>. U ovom projektu korištena je verzija 8.0.39.

Nakon preuzimanja Tomcat poslužitelja potrebno je postaviti varijable okruženja što je moguće napraviti na sljedećoj lokaciji: "Control Panel" - "System and Security" - "System" - "Advanced System Settings" - "Environmental variables". Klikom na "New" dodajemo novu sistemsku varijablu "CATALINA_HOME". Putanju treba postaviti na direktorij gdje je skinut Tomcat, npr. "C:\TomCat\apache-tomcat-8.0.39".

Nakon toga u varijablu Path potrebno dodati "%CATALINA_HOME%\bin". U uređivaču teksta potrebno je otvoriti datoteku users.xml u poddirektoriju "%CATALINA_HOME%\conf". Na kraju xml datoteke potrebno je dodati liniju prikazanu u nastavku.

```
<user username="user" password="blabla" roles="standard, managerscript, managergui"/>
```

"Username" i "password" su identifikatori za autorizaciju na Tomcat server koji će biti potrebni prilikom postavljanja aplikacije na server. Nakon završetka konfiguracije možemo pokrenuti Tomcat poslužitelj iz naredbenog retka pozivajući startup.bat skriptu iz "%CATALINA_HOME%\bin" direktorija kao što je prikazano na slici 5.3.



Slika 5.3: Podešavanje Tomcat servera unutar Eclipsea

Napomena: Potrebno je ostaviti otvorene prozore koje smo dobili u prošlom koraku, jer se gašenjem prozora gasi i web poslužitelj. Ujedno ispravan način gašenja Tomcat servera je pokretanjem shutdown.sh skripte koja se nalazi "%CATALINA_HOME%\bin" direktoriju jer se na taj način ispravno oslobode svi resursi operacijskog sustava.

Konfiguracija veze s bazom podataka

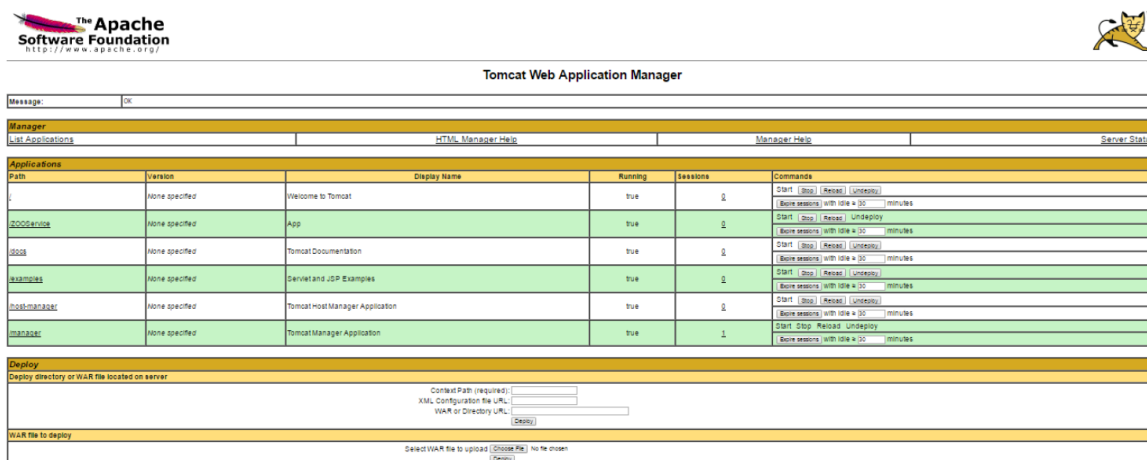
Da bi aplikacija uspješno koristila bazu podataka, potrebno je postaviti parametre konekcije s bazom podataka u datoteci persistence.xml u direktoriju META-INF. Na Tomcat poslužitelju direktorij će se nalaziti unutar ZOOService.war arhive pod WEB-INF\classes. Potrebno je promijeniti svojstva "username" i "password" na korisničko ime i lozinku koja se koristila prilikom stvaranja baze. Linije koje je potrebno izmijeniti dane su u nastavku.

```
<property name="hibernate.connection.username" value="root" />
<property name="hibernate.connection.password" value="root"/>
```

Pokretanje aplikacije izravno iz Tomcat sučelja

Nakon pokretanja Apache Tomcat poslužitelja, može mu se pristupiti na poveznici <http://localhost:8080/>. S desne strane odabiremo "Manager app" i zatim u polju "War file to deploy" odabiremo "Choose File" kao što je prikazano na slici. Potrebno je odabrati ZOOService.war datoteku koja je dostupna na gitlab repozitoriju

projekta unutar direktorija "Izvršni kod". Nakon toga treba kliknuti na "Deploy" i aplikacija je sada dostupna na adresi: `http://localhost:8080/ZOOService`.



Slika 5.4: Pokretanje aplikacije iz Tomcat sučelja

Moguće je i datoteku ZOOService.war postaviti u `\apache-tomcat-8.0.39\webapps` te resetirati Tomcat server. U tom slučaju će se koristiti autodeploy način i nije potrebno posebno postavljati put do te datoteke.

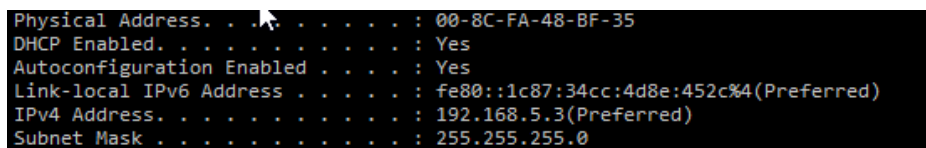
Definiranje adrese spajanja

Za ispravno pokretanje aplikacije na mobilnom uređaju ili emulatoru potrebno je podesiti varijablu baznog url-a sučelja na koje se spaja aplikacija. To je moguće napraviti na dva načina koji će biti objašnjeni u nastavku.

Prvi način je da se u izvornom kodu promjeni IP adresa poslužitelja. Potrebno je odabrati ApiClient.java razred u dao.rest paketu te zatim postaviti varijablu "HOST_IP" na sljedeći način:

```
private static String HOST_IP = "192.168.0.12";
```

U ovom primjeru URL je postavljen na adresu računala na kojem je bila pokrenuta poslužiteljska aplikacija 192.168.0.12. Taj dio je potrebno zamijeniti s adresom računala na kojem se trenutno pokreće aplikacija. Na operacijskom sustavu Windows možemo saznati adresu računala upisom ipconfig naredbe u komandni redak te možemo pročitati rezultat pod IPv4 poljem kao što je prikazano na slici 5.5.



```
Physical Address. . . . . : 00-8C-FA-48-BF-35
DHCP Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . : fe80::1c87:34cc:4d8e:452c%4(Preferred)
IPv4 Address. . . . . : 192.168.5.3(Preferred)
Subnet Mask . . . . . : 255.255.255.0
```

Slika 5.5: Određivanje IP adrese poslužitelja

U ovom slučaju bi promijenili varijablu `BASE_URL` na 192.168.5.3. Ukoliko se aplikacija pokreće pomoću emulatora Android uređaja, potrebno je koristiti ip adresu 10.0.22.

Drugi način je ručno podešavanje IP adrese poslužitelja unutar aplikacije. Oda- birom "Postavke" - "Podesi IP adresu poslužitelja" , možemo upisati IP adresu računala na koje se naša aplikacija spaja, pod pretpostavkom da je ta mrežna adresa dostupna unutar mreže na koju se spaja uređaj.

Pokretanje Android aplikacije

Potrebno je preuzeti `zoovrt.apk` datoteku s Gitlab repozitorija projekta pod di- rektorijem "Izvršni kod" na mobitel. Zatim ju je potrebno pokrenuti čime će se pokrenuti i instalacija Android aplikacije nakon čega klikom na ikonu aplikacije pokrećemo aplikaciju.

6. Zaključak i budući rad

Zadatak naše grupe bio je razvoj web aplikacije za online naručivanje jela uz mogućnost upravljanja rezervacijama, katalogom jela, zaposlenicima i recenzijama narudžbi. Nakon 17 tjedana rada u timu i razvoja, ostvarili smo zadani cilj. Sama provedba projekta bila je kroz dvije faze.

Prva faza projekta uključivala je okupljanje tima za razvoj aplikacije, dodjelu projektnog zadatka i intenzivan rad na dokumentiranju zahtjeva. Kvalitetna provedba prve faze uvelike je olakšala daljnji rad pri realizaciji osmišljenog sustava. Izrađeni obrasci i dijagrami (obrasci uporabe, sekvencijski dijagrami, model baze podataka, dijagram razreda) bili su od pomoći podtimovima zaduženima za razvoj *backenda* i *frontenda*. Izrada vizualnih prikaza idejnih rješenja problemskog zadatka uštedila je mnogo vremena u drugom ciklusu kada su članovi tima nailazili na nedoumice oko implementacije rješenja.

Druga faza projekta, iako nešto kraća od prve, bila je puno intenzivnija po pitanju samostalnog rada članova. Manjak iskustva članova u izradi sličnih implementacijskih rješenja primorao je članove na samostalno učenje odabranih alata i programskih jezika kako bi ispunili dogovorene ciljeve. Osim realizacije rješenja, u drugoj fazi je bilo potrebno dokumentirati ostale UML dijagrame i izraditi popratnu dokumentaciju kako bi budući korisnici mogli lakše koristiti ili vršiti preinake na sustavu. Dobro izrađen kostur projekta uštedio nam je mnogo vremena prilikom izrade aplikacije te smo izbjegli moguće pogreške u izradi koje bi bile vremenski skupe za ispravljanje u daljnjoj fazi projekta.

Komunikacija među članovima tima bila je putem Whatsappa čime smo postigli informiranost svih članova grupe o napretku projekta. Moguće proširenje postojeće inačice sustava je izrada mobilne aplikacije čime bi se cilj projektnog zadatka bio ostvaren u većoj mjeri no s web aplikacijom.

Sudjelovanje na ovakvom projektu bilo je vrijedno iskustvo svim članovima tima jer smo kroz intenzivnih nekoliko tjedana rada iskusili zajednički rad na istom projektu. Također, osjetili smo važnost dobre vremenske organiziranosti i koordiniranosti između članova tima. Zadovoljni smo postignutim bez obzira na golemi prostor za usavršavanje izrađene aplikacije što je posljedica neiskustva na takvim i

sličnim projektima.

Popis literature

1. Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langaniere, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

Indeks slika i dijagrama

3.1	Dijagram obrasca uporabe, funkcionalnost korisnika i klijenta . . .	22
3.2	Dijagram obrasca uporabe, funkcionalnost vlasnika i zaposlenika . .	23
3.3	Dijagram obrasca uporabe, funkcionalnost administratora	24
3.4	Sekvencijski dijagram za UC7	26
3.5	Sekvencijski dijagram za UC8	28
3.6	Sekvencijski dijagram za UC15	29
3.7	Sekvencijski dijagram za UC24	30
4.1	Arhitektura sustava	32
4.2	E-R dijagram baze podataka	38
4.3	Dijagram razreda - dio Controllers	40
4.4	Dijagram razreda - dio Data transfer objects	41
4.5	Dijagram razreda - dio Models	43
4.6	Dijagram stanja	44
4.7	Dijagram aktivnosti	46
4.8	Dijagram komponenti	48
5.1	Pregledavanje restorana na karti	51
5.2	Dijagram razmještaja	52
5.3	Podešavanje Tomcat servera unutar Eclipsea	55
5.4	Pokretanje aplikacije iz Tomcat sučelja	56
5.5	Određivanje IP adrese poslužitelja	57
6.1	prikaz aktivnosti na repozitoriju	67

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 10. listopada 2018.
- Prisustvovali: xx
- Teme sastanka:
 - sastanak s asistentom i demonstratorom
 - raščišćavanje osnovnih dilema funkcionalnosti
 - analiza zadatka

2. sastanak

- Datum: 16. listopada 2018.
- Prisustvovali:xx
- Teme sastanka:
 - konačan odabir alata i tehnologija
 - definiranje funkcionalnih zahtjeva
 - podjela opisa obrazaca uporabe

3. sastanak

- Datum: 23. listopada 2018.
- Prisustvovali:xx
- Teme sastanka:
 - sastanak s asistentom i demosom - prodiskutirati dosadašnji rad (fokus na opisima obrazaca uporabe)
 - podijeliti dalje poslove

4. sastanak

- Datum: 31. listopada 2018.
- Prisustvovali:xx
- Teme sastanka:
 - podjela zadataka (sekvencijski dijagrami + dodani obrasci + popravak dijagrama obrazaca uporabe)

5. sastanak

- Datum: 5. studenoga 2018.
- Prisustvovali:xx
- Teme sastanka:
 - sastanak s asistentom - komentiranje dijagrama obrazaca i sekvan-
cijskih dijagrama
 - daljnja podjela rada

6. sastanak

- Datum: 10. studenoga 2018.
- Prisustvovali:xx
- Teme sastanka:
 - osmišljavanje stranice i pojedinih pregleda

7. sastanak

- Datum: 16. studenoga 2018.
- Prisustvovali:xx
- Teme sastanka:
 - sastanak s asistentom i demosom - evaluacija dosadašnjeg rada

8. sastanak

- Datum: 20. studenoga 2018.
- Prisustvovali:xx
- Teme sastanka:
 - dogovor za popravljavanje stavki u dokumentaciji

9. sastanak

- Datum: 3. prosinca 2018.
- Prisustvovali:xx
- Teme sastanka:
 - komentiranje dokumentacije

10. sastanak

- Datum: 7. prosinca 2018.
- Prisustvovali:xx
- Teme sastanka:
 - plan daljnjeg napretka

11. sastanak

- Datum: 18. prosinca 2018.

- Prisustvovali:xx
- Teme sastanka:
 - podjela rada implementacije

12. sastanak

- Datum: 11. siječnja 2019.
- Prisustvovali:xx
- Teme sastanka:
 - sastanak s asistentom i demosom - demonstracija alfa inačice

13. sastanak

- Datum: 16. siječnja 2019.
- Prisustvovali:xx
- Teme sastanka:
 - završetak implementacije

14. sastanak

- Datum: 17. siječnja 2019.
- Prisustvovali:xx
- Teme sastanka:
 - dovršavanje pisanja dokumentacije

Tablica aktivnosti

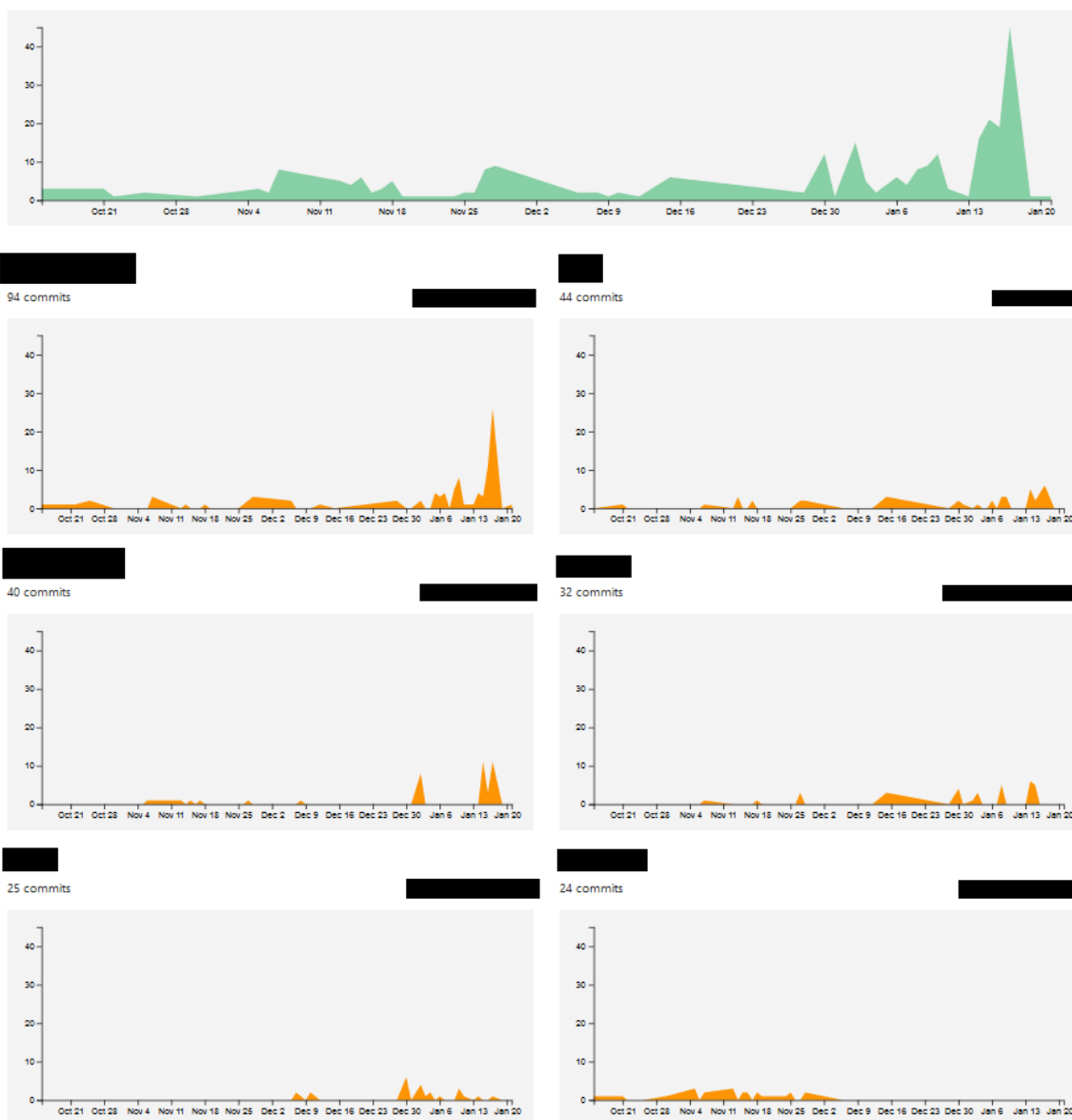
Kontinuirano osvježavanje

Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom						
Opis projektnog zadatka						
Opis funkcionalnih zahtjeva						
Dijagram obrazaca						
Opis pojedinih obrazaca						
Sekvencijski dijagrami						
Opis ostalih zahtjeva						
Svrha, opći prioriteti i skica sustava						
Baza podataka						
Dijagram razreda						
Dijagram stanja						
Dijagram aktivnosti						
Dijagram komponenti						
Korištene tehnologije i alati						
Ispitivanje programskog rješenja						
Upute za instalaciju						
Plan rada						
Dnevnik sastajanja						
Zaključak i budući rad						
Popis literature						
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>						

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
<i>npr. izrada početne stranice</i>						
<i>izrada baze podataka</i>						
<i>spajanje s bazom podataka</i>						
<i>back end</i>						

Dijagrami pregleda promjena



Slika 6.1: prikaz aktivnosti na repozitoriju