



UNIVERZITET U ZENICI

Politehnički fakultet

Softversko inženjerstvo

Razvoj informacijskih sistema



Berina Topalović

Informacioni sistem restorana (ISR)

Diplomski rad

Mentor:

V. prof. dr. Denis Čeke

Zenica, 2025.

BIBLIOGRAFSKA KARTICA RADA

NAUČNO PODRUČJE RADA: Tehničke nauke

NAUČNO POLJE RADA: Softversko inženjerstvo

NAUČNA GRANA RADA: Razvoj infomacijskih sistema

USTANOVA U KOJOJ JE RAD IZRAĐEN: Univerzitet u Zenici, Politehnički fakultet

MENTOR RADA: V. prof. dr. Denis Čeke

DATUM ODBRANE RADA:

ČLANOVI KOMISIJE ZA ODBRANU:

IZJAVA

Izjavljujem da sam diplomski rad pod naslovom „Informacioni sistem restorana (ISR)“ izradila samostalno pod mentorstvom profesora V. prof. dr. Denis Čeke. U radu sam koristila literaturu koja je navedena na kraju diplomskog rada.

Studentica:

Berina Topalović

SAŽETAK:

Informacioni sistem restorana (ISR) predstavlja savremeni informacioni sistem razvijen s ciljem unapređenja poslovanja restorana i poboljšanja korisničkog iskustva gostiju. Sistem je realizovan kao web aplikacija namijenjena administratorima i osoblju restorana, kao i krajnjim korisnicima – gostima restorana. Posebne funkcionalnosti sistema uključuju mogućnost online plaćanja, kao i skeniranje QR koda za brzu i jednostavnu online rezervaciju stolova, čime se znatno smanjuju gužve i olakšava organizacija rada.

Tehnologije korištene u razvoju sistema uključuju MSSQL kao bazu podataka, ASP.NET za razvoj backend-a aplikacije, te Angular za izradu frontend dijela aplikacije. Upravljanje projektom realizovano je korištenjem alata ProjectLibre, kojim je vođeno planiranje i praćenje aktivnosti tokom razvoja ovog informacionog sistema.

Ključne riječi: ISR, restoran, online plaćanje, QR rezervacija, MSSQL, ASP.NET, Angular, ProjectLibre.

SADRŽAJ

1. UVOD.....	6
2. PREGLED PLANA PROJEKTA.....	7
2. 1. Planiranje i istraživanje.....	7
2. 2. Dokumentacija sistema.....	7
2. 3. Dizajn sistema.....	8
2. 4. Implementacija sistema.....	8
2. 5. Održavanje sistema.....	8
3. PRIKUPLJANJE I ODREĐIVANJE KORISNIČKIH ZAHTJEVA.....	9
3. 1. Metode prikupljanja podataka.....	9
3. 2. Identifikacija glavnih problema.....	9
3. 3. Korisnički zahtjevi.....	10
4. SISTEMSKA ANALIZA PROBLEMA.....	11
4. 1. Use case dijagrami.....	11
4. 1. 1. Use case prijava, registracija, odjava.....	11
4. 1. 2. Use case meni.....	12
4. 1. 3. Use case narudžba.....	13
4. 1. 4. Use case račun.....	14
4. 1. 5. Use case rezervacija.....	14
4. 1. 6. Use case upravljanje zaposlenicima.....	15
4. 2. Dijagrami aktivnosti.....	16
4. 2. 1. Dijagram aktivnosti prijave, registracije i odjave.....	16
4. 2. 2. Dijagram aktivnosti upravljanja menijem.....	17
4. 2. 4. Dijagram aktivnosti kreiranja narudžbe.....	18
4. 2. 5. Dijagram aktivnosti izdavanja računa.....	18
4. 2. 6. Dijagram aktivnosti upravljanja zaposlenicima.....	19
4. 3. Dijagram klasa.....	20
4. 4. Dijagram komponenti.....	21
5. SISTEMSKI DIZAJN.....	23
5. 1. ER dijagram.....	23
5. 2. Dizajn baze podataka.....	24
5. 3. Dizajn interfejsa, formi, izvještaja i dijaloga putem mockup-a.....	25
5. 3. 1. Dizajn web aplikacije za administratora.....	26
5. 3. 2. Dizajn web aplikacije za korisnike.....	28
6. ARHITEKTURA SISTEMA.....	35
6. 1. Prezentacijski sloj.....	35
6. 2. Poslovni sloj.....	35
6. 3. Sloj baze podataka.....	36
7. TEHNOLOGIJE ZA BACKEND.....	37
7. 1. ASP .NET.....	37
7. 2. Swagger UI.....	37

7. 3. MSSQL server.....	38
7. 4. Visual Studio kao razvojno okruženje za backend.....	38
8. TEHNOLOGIJE ZA FRONTEND.....	39
8. 1. Angular.....	39
8. 2. HTML, CSS.....	39
8. 3. Visual Studio Code.....	39
9. UPRAVLJANJE IMPLEMENTACIJOM INFORMACIONOG SISTEMA RESTORANA..	
40	
9. 1. Pregled liste zadataka u Trello alatu.....	40
9. 2. Baza.....	41
9. 3. Web API.....	42
9. 4. Web aplikacija.....	43
10. PRAKTIČNA IMPLEMENTACIJA SISTEMA.....	46
10. 1. Baza podataka.....	46
10. 2. API.....	47
10. 2. 1. API sloj – prezentacijski i ulazni sloj aplikacije.....	48
10. 2. 2. BLL sloj (Business Logic Layer) – poslovna logika.....	48
10. 2. 3. DAL sloj (Data Access Layer) – pristup bazi podataka.....	48
10. 2. 4. DTO – data transfer object (prenos podataka između slojeva).....	49
10. 2. 5. Models – entiteti (reprezentacija tabela u bazi).....	49
10. 2. 6. Program.cs.....	49
10. 2. 7. appsettings.json.....	49
10. 3. Web aplikacija.....	50
10.3.1. Korijenska struktura projekta.....	50
10.3.2. src folder – Glavni direktorij aplikacije.....	51
10.3.3. app folder – Centralni dio aplikacije.....	51
10.3.4. Arhitekturna organizacija.....	52
11. ZAKLJUČAK.....	54
12. LITERATURA.....	55

1. UVOD

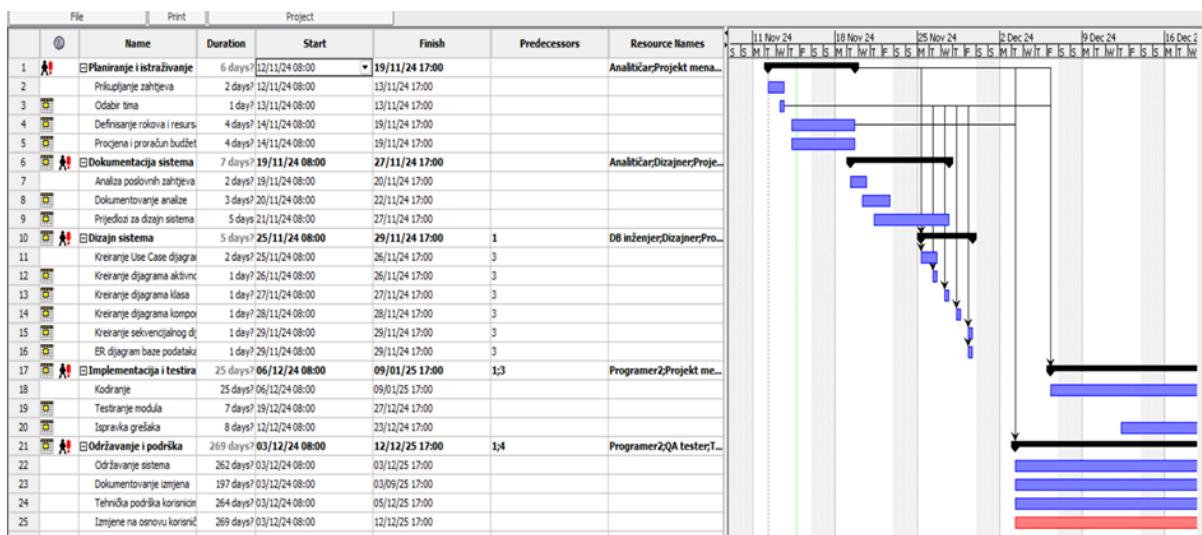
U vremenu u kojem trenutno živimo, možemo reći da je tehnologija mnoge aspekte naših života uzela pod svoje rukovodstvo, a to podrazumijeva pojam **digitalizacija**. U jedan od tih aspekata spada i ugostiteljstvo, gdje su brzina, tačnost i efikasnost ključni za adekvatno poslovanje. S obzirom na to, restorani, kao jedni od najdinamičnijih sektora, zahtijevaju kompleksno i dobro organizovano upravljanje procesima kako bi obezbijedili kvalitetnu uslugu gostima. Uzimajući sve to u obzir, rješenje koje se pokazalo kao najfunkcionalnije da odgovori svim ovim zahtjevima je optimizacija radnih procesa i implementacija informacionih sistema koji mogu automatizovati i unaprijediti poslovanje restorana. Informacioni sistemi predstavljaju revolucionarno rješenje koje omogućava upravljanje svim ključnim aspektima rada restorana – od prijave zaposlenih, preko upravljanja narudžbama, do kontrole skladišta– sve putem jedinstvenog digitalnog okruženja.

Informacioni sistem koji će biti implementiran u ovom diplomskom radu će uključivati osnovne funkcionalnosti kao što su: prijava i odjava korisnika, registracija novih zaposlenih, detaljno upravljanje jelovnikom, implementacija funkcija kreiranja, ažuriranja i brisanja jela u meniju, rezervacija određenih stolova, online plaćanje, kao i evidentiranje i praćenje narudžbi. Implementacijom ovog sistema, efikasnije će se moći upravljati procesima u restoranu koji uključuju naručivanje i posluživanje jela, a smanjit će se greške i poboljšati komunikacija.

Cilj ovog projekta je kreirati informacioni sistem koji će unaprijediti komunikaciju i saradnju među zaposlenicima, olakšati rukovodiocima praćenje rada i poslovanja, te povećati zadovoljstvo gostiju kroz bržu i tačniju uslugu. Na taj način, informacioni sistem će postati značajan faktor u poboljšanju efikasnosti. Digitalizacija i optimizacija poslovanja putem informacionog sistema doprinijet će rastu zadovoljstva gostiju, boljoj kontroli kvaliteta usluga, i smanjenju operativnih troškova.

2. PREGLED PLANA PROJEKTA

Najbolja realizacija određenog projekta se može implementirati onda kada je dobro definisan plan implementacije izrade. U ovom dijelu projekta će biti predstavljen plan za informacioni sistema restorana, razvijen u Project Libre alatu. Projekat se sastoji od nekoliko ključnih faza koje omogućavaju efikasno planiranje, razvoj i implementaciju sistema prema zahtjevima korisnika. Na slici 1. je prikazan pregled plana projekta.



Slika 1. Prikaz plana projekta

2. 1. Planiranje i istraživanje

Prva i temeljna faza u razvoju informacionog sistema jeste faza planiranja i istraživanja. U ovoj fazi se vrši prikupljanje zahtjeva od klijenata i osoblja restorana koji će svakodnevno koristiti sistem. Cilj ove faze je precizno razumijevanje potreba korisnika kako bi sistem odgovarao stvarnim poslovnim procesima.

U sklopu planiranja identificuju se osnovne funkcionalnosti koje sistem treba da posjeduje, kao što su prikupljanje relevantnih zahtjeva, odabir tima koji radi na projektu, definiranje rokova i resursa, te procjena i proračun budžeta.

2. 2. Dokumentacija sistema

Nakon planiranja prelazi se na fazu izrade dokumentacije, koja uključuje dubinsku analizu i dizajn sistema. Analizu najčešće vrši poslovni analitičar koji proučava sakupljene zahtjeve,

proučava tržište, postojeća rješenja i oblikuje početne prijedloge kako bi sistem bio što efikasniji i jednostavniji za korištenje.

2. 3. Dizajn sistema

Faza dizajna sistema uključuje izradu različitih dijagrama koji vizualizuju kako će sistem funkcionišati i izgledati. Ova faza obuhvata pet ključnih koraka:

- **Use case dijagram:** Prikazuje interakcije između korisnika i sistema;
- **Dijagram aktivnosti:** Opisuje tok radnji unutar sistema;
- **Dijagram klasa:** Modeliraju se objekti u sistemu i njihove međusobne veze;
- **Dijagram komponenti:** Prikazuje tehničke module sistema i njihovu međusobnu povezanost;
- **ER dijagram baze podataka:** Definiše sve tabele, njihove kolone i odnose između njih u okviru baze podataka.

Ovi dijagrami služe kao vodič za fazu implementacije i pomažu svim članovima tima da imaju zajedničku sliku o funkcionisanju sistema.

2. 4. Implementacija sistema

Faza implementacije i testiranja obuhvata konkretno razvijanje funkcionalnosti sistema prema prethodno izrađenom dizajnu, uz paralelno testiranje kako bi se otkrile i ispravile potencijalne greške u ranoj fazi razvoja. Dakle, to su:

- **Programiranje:** Programeri implementiraju sistem prema definisanom dizajnu;
- **Integracija i testiranje:** Ovaj proces uključuje QA inženjere, programere i IT stručnjake kako bi se garantovalo da je sistem potpuno funkcionalan prije njegove implementacije u praksi. Ova faza započinje tek nakon što se programiranje završi.

2. 5. Održavanje sistema

Posljednja faza traje najduže, odnosno 365 dana, i uključuje kontinuirano održavanje sistema. Plan održavanja kreira se odmah nakon implementacije, a za njega su zaduženi administratori sistema, programeri i menadžeri. Održavanje može uključivati periodična ažuriranja sistema i otklanjanje potencijalnih grešaka tokom korištenja.

3. PRIKUPLJANJE I ODREĐIVANJE KORISNIČKIH ZAHTJEVA

Faza prikupljanja i određivanja korisničkih zahtjeva predstavlja jedan od najvažnijih koraka u razvoju informacionog sistema za restorane. Kroz ovu fazu se analiziraju stvarne potrebe osoblja i menadžmenta restorana, identifikuju svakodnevni izazovi sa kojima se suočavaju u svom radu, te se na osnovu toga definišu funkcionalnosti koje će budući sistem morati da podrži.

3. 1. Metode prikupljanja podataka

Za kvalitetno definisanje korisničkih zahtjeva korištene su sljedeće metode:

- **Istraživanje poslovnih potreba restorana:** Provedeno je detaljno istraživanje trenutnih operativnih procesa u restoranu, uključujući upravljanje osobljem, proces naručivanja hrane i pića, upravljanje zalihamama, praćenje finansija i pripremu izvještaja. Fokus je bio na razumijevanju uloga različitih članova menadžerskog i operativnog tima, kao i identifikaciji zadatka koji se često obavljaju ručno ili neefikasno;
- **Analiza postojećih procedura i organizacione strukture:** Posmatrani su i analizirani tokovi informacija unutar restorana, od trenutka kada gost naruči hranu, preko pripreme u kuhinji, do naplate i analize prodaje. Identifikovani su segmenti poslovanja koji su podložni greškama, kašnjenjima ili preklapanju zadataka;
- **Komunikacija sa menadžerskim timom i osobljem:** Kroz direktne razgovore i neformalne intervjuje sa glavnim menadžerom i ostalim zaposlenima (konobarima, kuharima, pomoćnim osobljem), dobijene su informacije o specifičnim problemima koje bi informacioni sistem trebao da riješi.

3. 2. Identifikacija glavnih problema

Na osnovu sprovedenih analiza i istraživanja, identifikovani su sljedeći glavni problemi koji trenutno usporavaju ili otežavaju rad restorana:

- **Nepregledno upravljanje osobljem:** Menadžeri se često suočavaju sa izazovima u praćenju radnih smjena, zaduženja i performansi zaposlenih, što otežava raspodjelu zadatka i evaluaciju rada tima;

- **Ručno vođenje evidencija:** Mnogi procesi, poput rezervacije dolazaka gostiju, potrošnje zaliha ili dnevnog prometa, vode se ručno, što povećava mogućnost grešaka i troši vrijeme;
- **Nedostatak centralizovanog sistema pristupa:** Zaposleni nemaju jasnu podjelu pristupa podacima. Npr. kuhari često ne zna tačan redoslijed narudžbi, a konobari ne mogu u svakom trenutku vidjeti status obroka, što usporava cijelokupan servis.

3. 3. Korisnički zahtjevi

Na osnovu identifikovanih problema i potreba, definisani su sljedeći ključni korisnički zahtjevi za informacioni sistem restorana:

- **Praćenje strukture tima i pristupa sistemu:** Sistem mora omogućiti jasno definisanje uloga i pristupnih prava za sve članove osoblja, uključujući glavnog menadžera, šefove odjela, konobare, kuhare i pomoćno osoblje. Svaka uloga mora imati pristup samo onim podacima koji su im relevantni;
- **Centralizovano upravljanje narudžbama:** Konobari trebaju unositi narudžbe direktno u sistem, koji će ih automatski proslijediti kuharima. Status pripreme obroka treba biti vidljiv u realnom vremenu, čime se smanjuje potreba za dodatnom komunikacijom;
- **Praćenje zaliha i menija:** Menadžer kuhinje mora imati pregled nad zalihama namirnica, planiranjem menija i narudžbama koje dolaze iz sale, kako bi se izbjegli nedostaci i osigurala nesmetana priprema obroka;
- **Jednostavan i funkcionalan interfejs:** Sistem treba biti prilagođen svim korisnicima bez obzira na njihovu tehničku pismenost, sa preglednim menijima, jasnim ikonama i što manjim brojem koraka za obavljanje osnovnih funkcija.

Korisnički zahtjevi čine osnovu za naredne faze razvoja informacionog sistema. Njihova precizna definicija omogućava kreiranje dizajna i funkcionalnosti koji će unaprijediti organizaciju rada, smanjiti greške, ubrzati procese i doprinijeti boljem korisničkom iskustvu gostiju i osoblja.

4. SISTEMSKA ANALIZA PROBLEMA

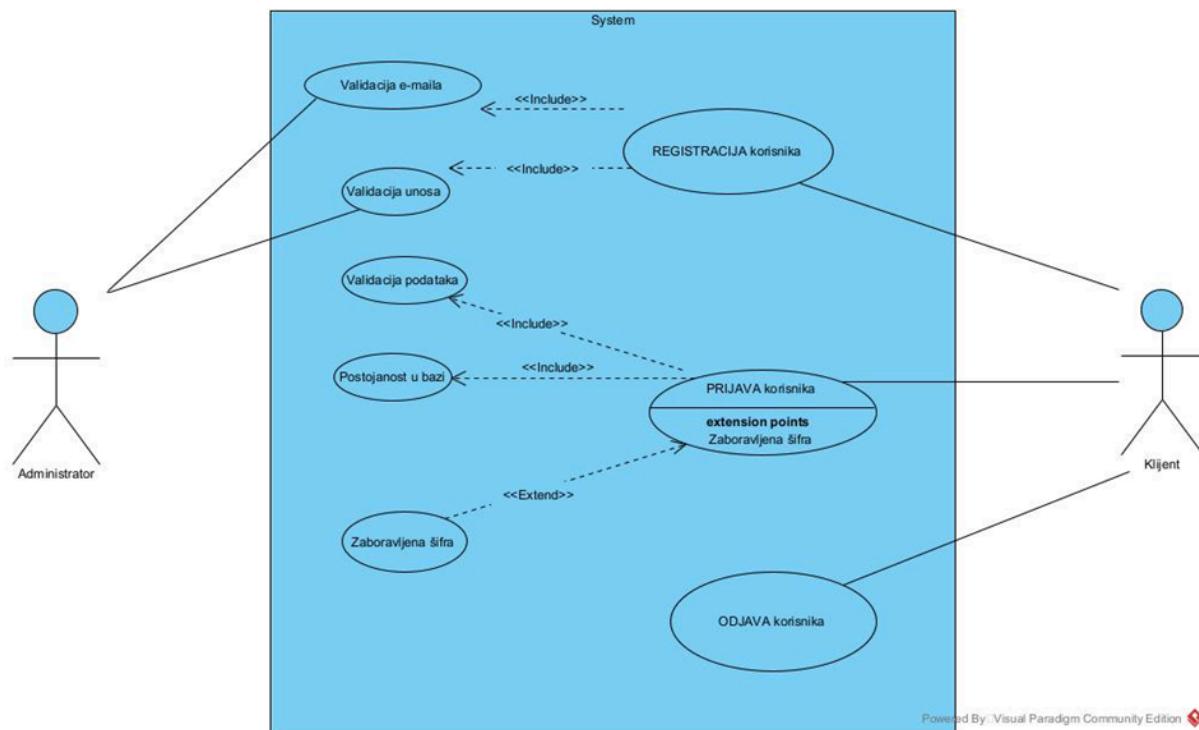
Sistemska analiza problema predstavlja treću fazu projekta u kojoj se prikazuju ključni dijagrami koji opisuju strukturu i funkcionalnosti informacionog sistema restorana. Sistemska analiza služi da razloži složenost poslovnih procesa i omogući bolji pregled u način na koji sistem podržava različite aktivnosti i korisničke interakcije. Na osnovu ove analize, detaljno će biti opisani svi aspekti funkcionisanja sistema, identifikovane glavne komponente i definisane relacije među njima.

4. 1. Use case dijagrami

Use Case dijagrami su bitni grafički prikazi koji ilustruju interakcije između korisnika (aktera) i sistema, prikazujući funkcionalnosti koje sistem omogućava korisnicima. To će biti od velikog značaja za detaljan prikaz funkcionalnosti informacionog sistema restorana.

4. 1. 1. Use case prijava, registracija, odjava

U prvom Use Case dijagramu se prikazuju osnovne funkcionalnosti sistema vezane za korisničke procese **prijave, registracije i odjave**. Cilj ovog dijagrama je vizualno prikazati interakcije između korisnika i sistema u kontekstu prijave na sistem, stvaranja novog korisničkog računa, te odjave sa sistema nakon završetka korištenja.



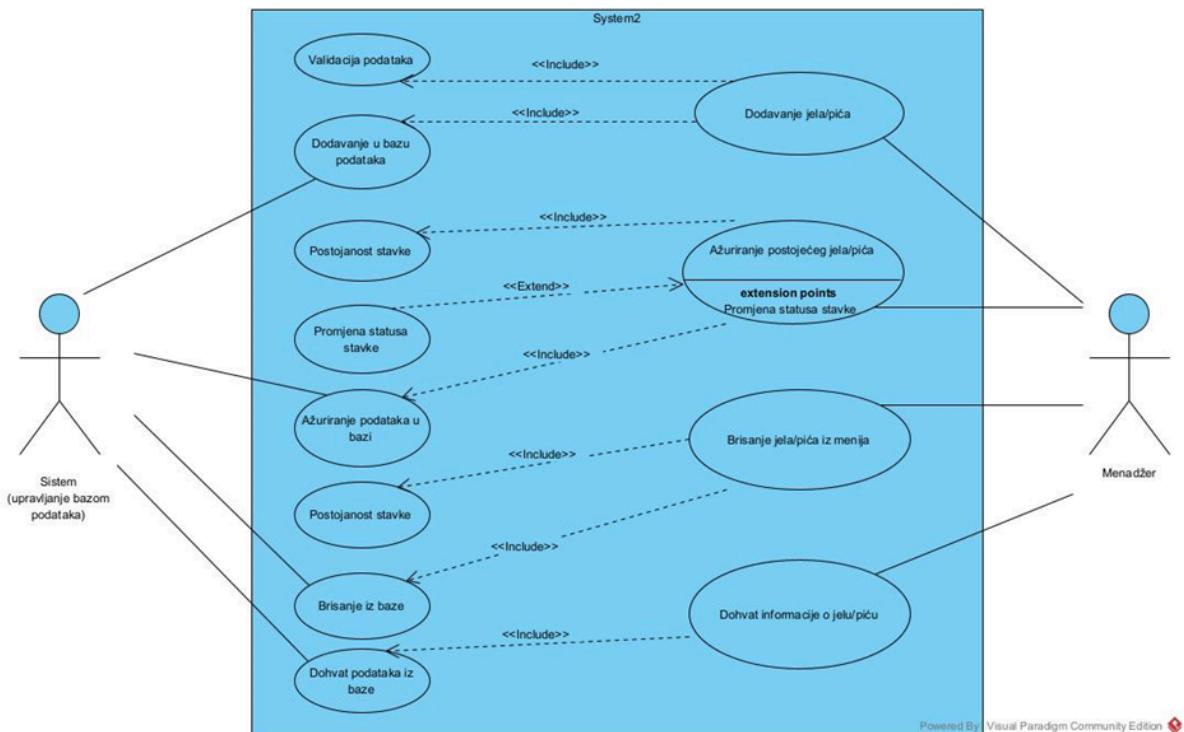
Slika 2. Use Case dijagram za prijavu, registraciju i odjavu korisnika

U ovom dijagramu identifikovana su dva aktera: **korisnik** (osnovni akter koji koristi sistem) i **administrator** (akter koji ima ulogu u administraciji, kao što je odobravanje novih registracija ili resetovanje šifri). Dijagram prikazuje tri glavne funkcionalnosti:

- **Registracija korisnika:** Proces u kojem korisnik unosi osnovne podatke za kreiranje novog korisničkog računa. Administrator je povezan s ovim procesom u slučaju da sistem zahtijeva njegovu intervenciju, kao što je verifikacija e-mail adrese;
- **Prijava korisnika:** Proces u kojem korisnik unosi svoje podatke (e-mail i šifru) kako bi pristupio svom računu;
- **Odjava korisnika:** Jednostavan proces gdje korisnik izlazi iz sistema.

4. 1. 2. Use case meni

U drugom Use Case dijagramu su prikazane funkcionalnosti sistema za kreiranje i upravljanje menijem restorana, koji omogućava menadžeru (administratoru) da dodaje, ažurira, briše i dohvati informacije o jelima i pićima. Sistem omogućava efikasno upravljanje ponudom restorana, uključujući mogućnost prilagodbe menija u skladu sa sezonskim promjenama i potrebama restorana.

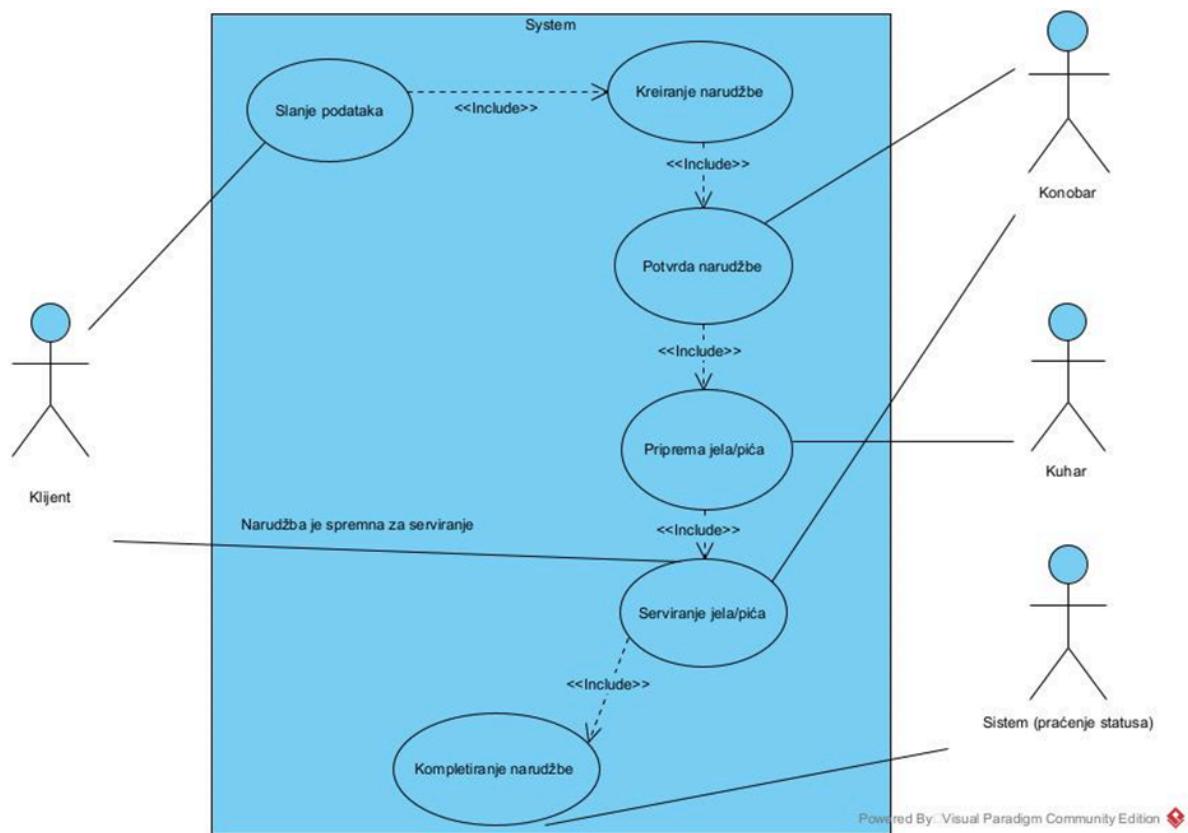


Slika 3. Use Case dijagram menija

Navedena su dva aktera ovog dijagrama. Glavni akter je menadžer, a sekundarni akter je sistem upravljanja bazom podataka, koji će sve promjene inicirane od menadžera (administratora) zabilježiti u svojoj bazi. Vezama include i extend kao i slučajevima upotrebe su detaljno opisani svi koraci izvršavanja ovog dijela projekta.

4. 1. 3. Use case narudžba

Dijagram za kreiranje narudžbe prikazuje interakciju između klijenta, konobara, kuhara i sistema tokom procesa naručivanja hrane u restoranu. Proces započinje kada klijent unese svoju narudžbu, a konobar je potvrđuje i šalje kuharu na pripremu. Nakon što jela budu pripremljena, konobar ih dostavlja klijentima, a sistem ažurira status narudžbe kao kompletну. Dijagram uključuje i opcionalnu funkcionalnost kao što je uputa za dostavu.



Slika 4. Use Case dijagram narudžbe

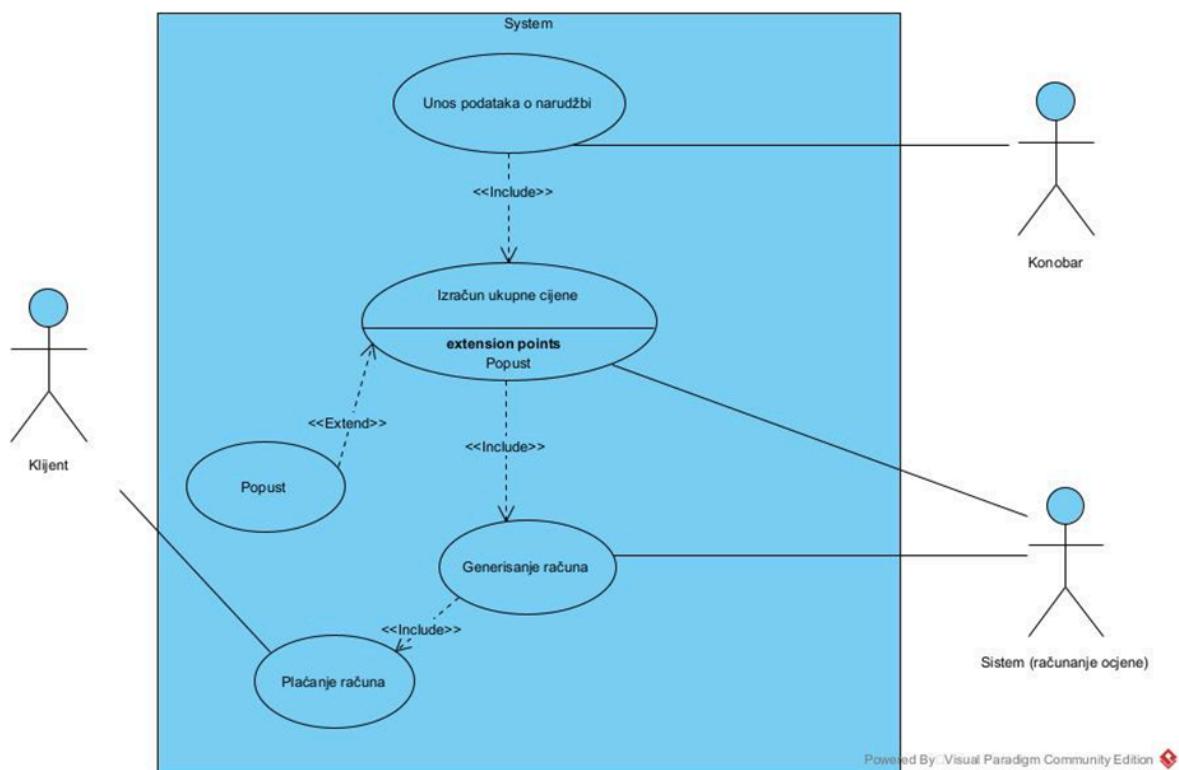
Kreiranjem i implementacijom ovog dijagrama omogućava se bolja organizacija sistema za efikasno upravljanje narudžbama, što olakšava rad konobara, kuhara i sistema.

4. 1. 4. Use case račun

Kroz ovaj dijagram, jasno se prikazuje kako sistem obračunava ukupnu cijenu naručene hrane i pića, kako generiše račun i kako se upravlja tim podacima.

Svaki korak procesa je jasno prikazan i povezan s odgovarajućim akterima. Dijagram započinje unosom podataka o narudžbi, gdje konobar unosi informacije o jelu/piću koje je klijent naručio. Ovi podaci prelaze u fazu izračuna ukupne cijene koju vrši sistem. Finaliziranjem ukupne cijene, dolazi se do narednog koraka, koji podrazumijeva generisanje računa.

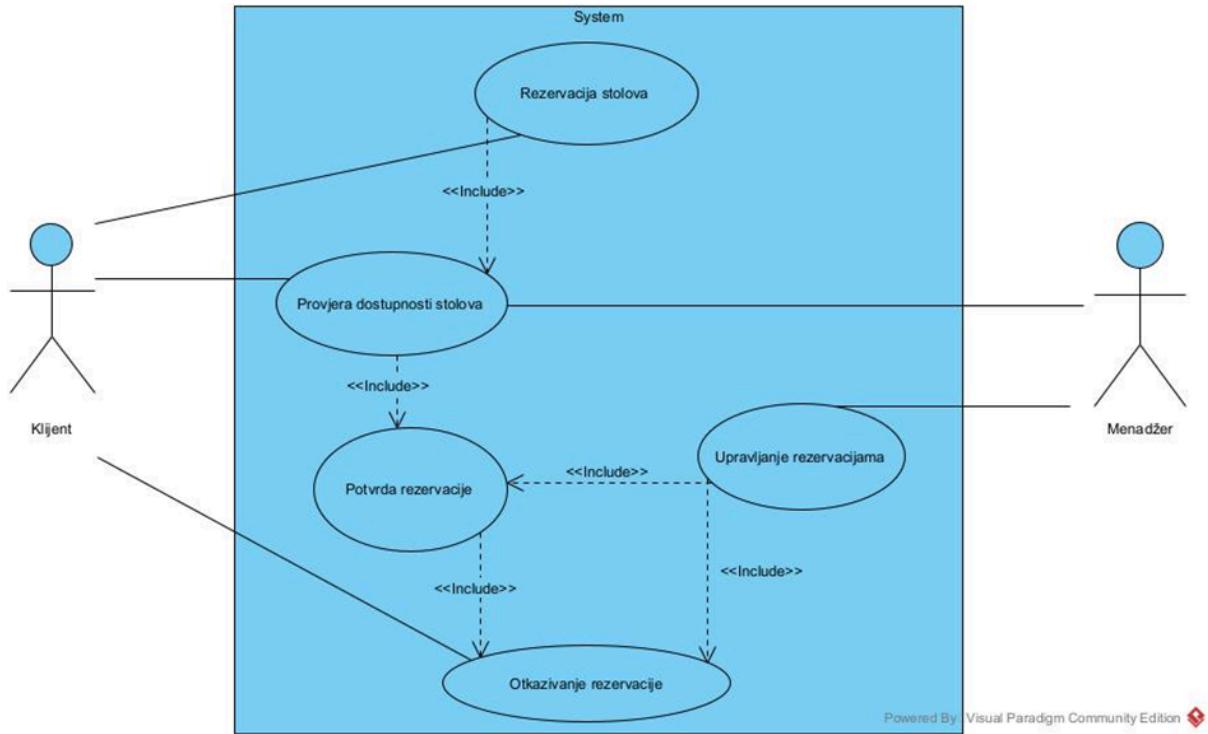
Pored osnovnog toka procesa, dijagram može uključivati opcionalne funkcionalnosti (kroz extend vezu), kao što je ovdje predstavljen popust, koji se aktivira samo u određenim situacijama.



Slika 5. Use Case dijagram računa

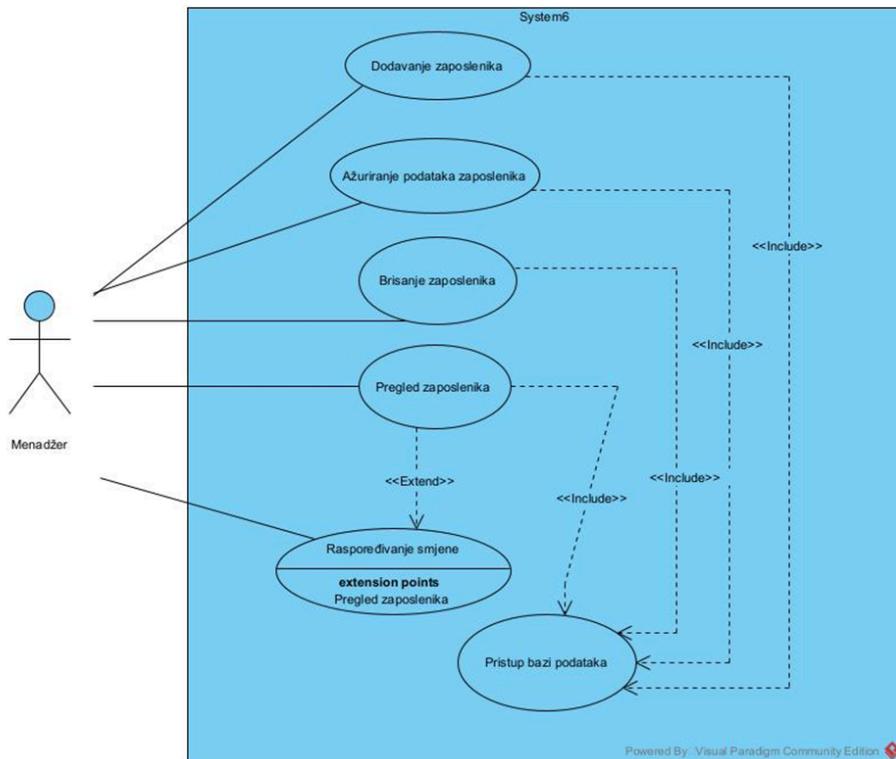
4. 1. 5. Use case rezervacija

Use Case dijagram za rezervaciju stolova prikazuje proces kojim klijenti mogu rezervisati sto u restoranu za specifičan datum i vrijeme.



Slika 6. Use Case Rezervacije

4. 1. 6. Use case upravljanje zaposlenicima



Slika 7. Use Case upravljanja zaposlenicima

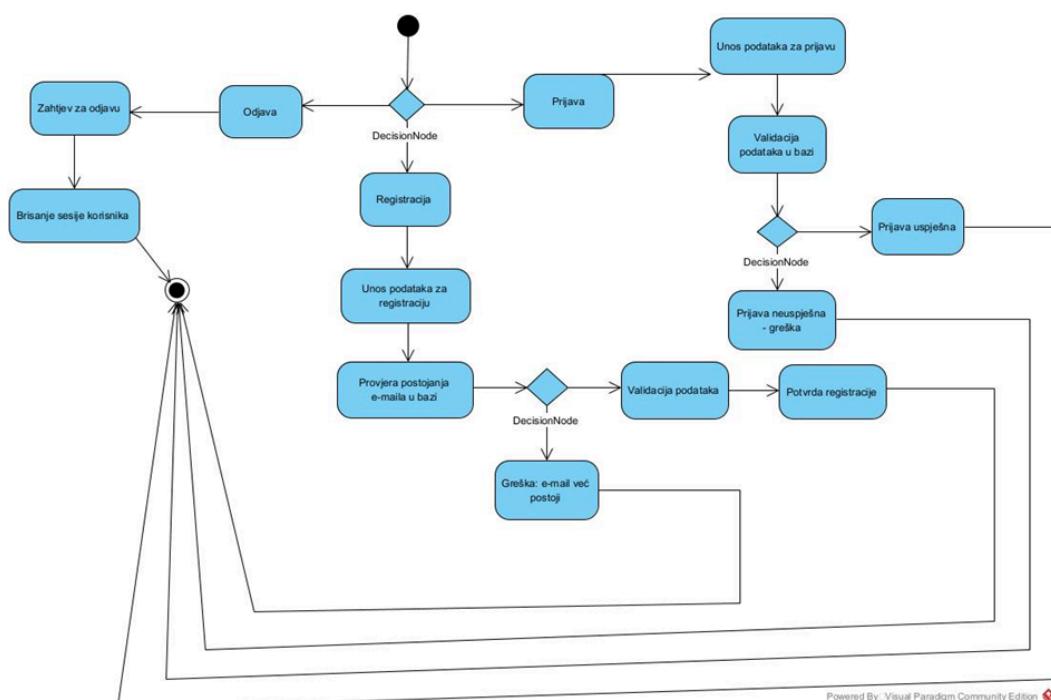
Slika 7. prikazuje na dijagramu upravljanje zaposlenicima koje obuhvata sve administrativne aktivnosti vezane za osoblje restorana. To uključuje dodavanje novih zaposlenih, ažuriranje postojećih podataka, brisanje određenih podataka i raspoređivanje smjena. Dijagram prikazuje kako menadžer upravlja podacima o zaposlenima, omogućava lakše praćenje osoblja i optimizaciju radnih procesa.

4. 2. Dijagrami aktivnosti

UML dijagrami aktivnosti prikazuju tok aktivnosti unutar sistema, predstavljajući korake u poslovnim procesima kroz povezane čvorove i prelaze. Ovi dijagrami omogućavaju vizualizaciju radnog procesa i toka informacija, što olakšava razumijevanje i praćenje složenih operacija. Pomoću ovih dijagrama detaljno će biti predstavljen i informacioni sistem restorana.

4. 2. 1. Dijagram aktivnosti prijave, registracije i odjave

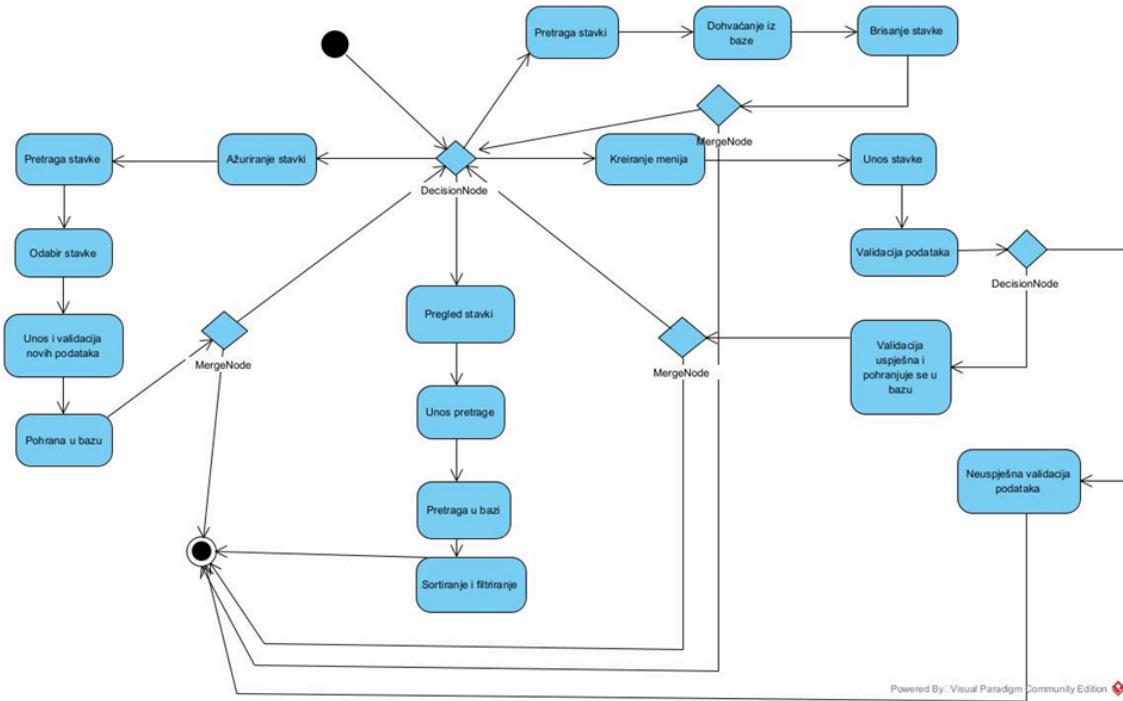
Dijagram aktivnosti autentifikacije korisnika prikazuje procesne tokove za prijavu, registraciju i odjavu korisnika. S obzirom na međusobnu povezanost ovih aktivnosti, dijagram omogućava lakše razumijevanje korisničkog toka i pojednostavljuje analizu rada sistema. Dijagram obuhvata sljedeće ključne tačke, koje će biti prikazane na slici broj 8.



Slika 8. Dijagram aktivnosti za autentifikaciju korisnika

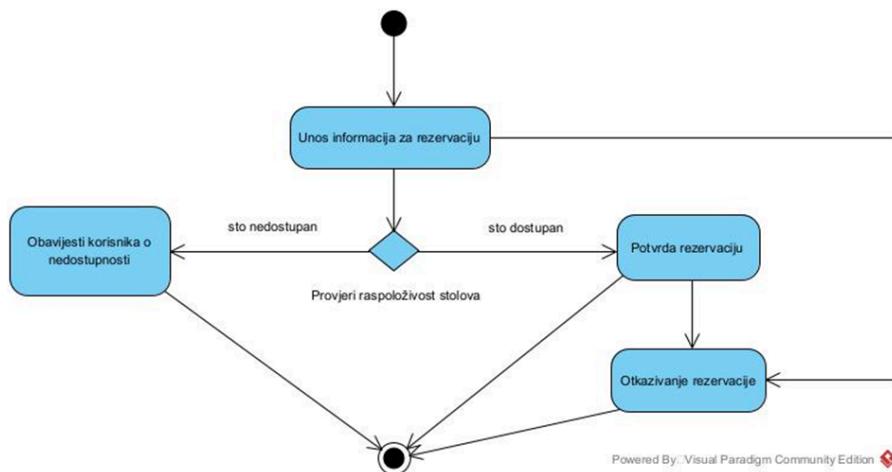
4. 2. 2. Dijagram aktivnosti upravljanja menijem

Upravljanje menijem uključuje osnovne operacije poput dodavanja novih stavki, ažuriranja postojećih stavki, brisanja stavki, i pregleda stavki menija. Svaka od ovih aktivnosti ima specifičan tok u dijagramu aktivnosti, čime se jasno prikazuju interakcije korisnika sa sistemom za upravljanje menijem.



Slika 9. Dijagram aktivnosti menija

4. 2. 3. Dijagram aktivnosti rezervacije stolova

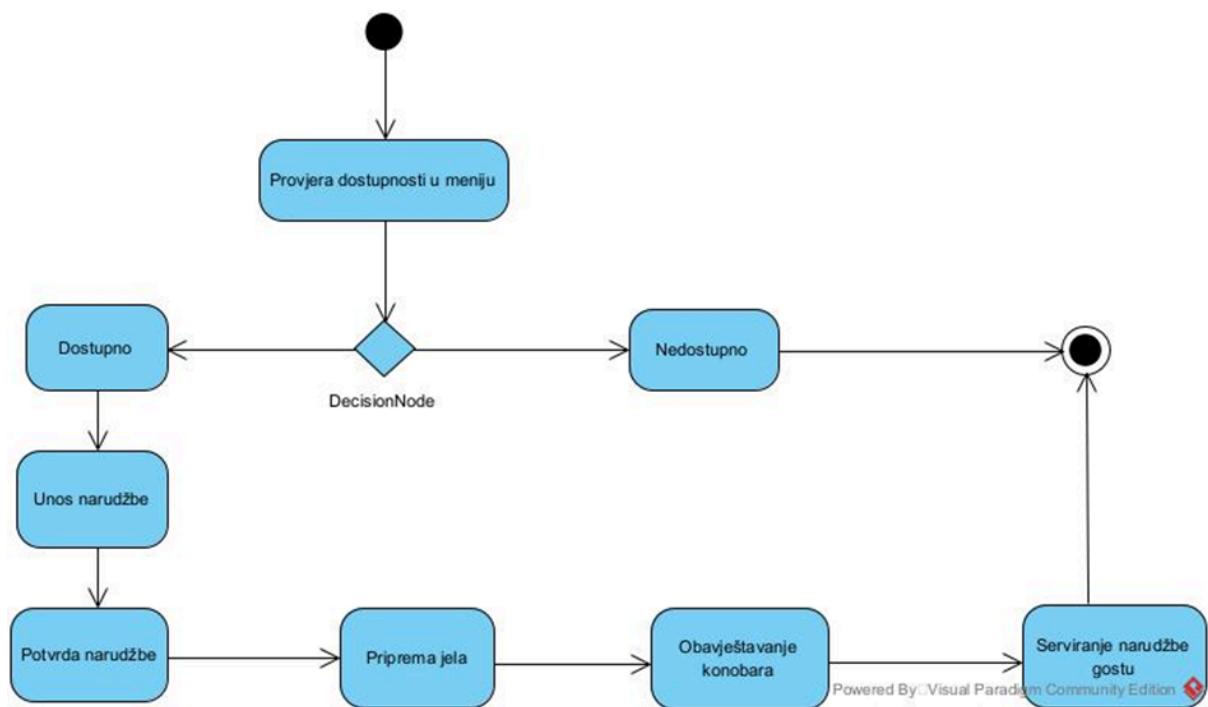


Slika 10. Dijagram aktivnosti rezervacije stolova

Ovaj dijagram prikazan na slici 10. pokazuje postupak od unosa potrebnih informacija (datum, vrijeme, broj osoba) do provjere dostupnosti, potvrde rezervacije i obavještavanja korisnika o statusu rezervacije. Dijagram takođe uključuje korake za otkazivanje rezervacije.

4. 2. 4. Dijagram aktivnosti kreiranja narudžbe

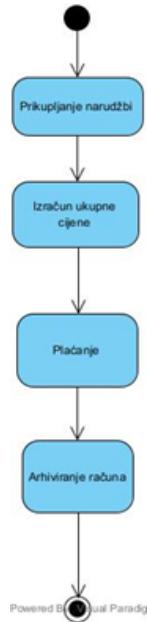
Dijagram aktivnosti za kreiranje narudžbe prikazuje sve korake kroz koje narudžba prolazi, od njenog unosa od strane konobara do serviranja gostu. Svaki korak, od provjere dostupnosti u meniju do obavještavanja konobara o spremnosti, je predstavljen kako bi se osigurao nesmetan rad i optimizacija vremena.



Slika 11. Dijagram aktivnosti kreiranja narudžbe

4. 2. 5. Dijagram aktivnosti izdavanja računa

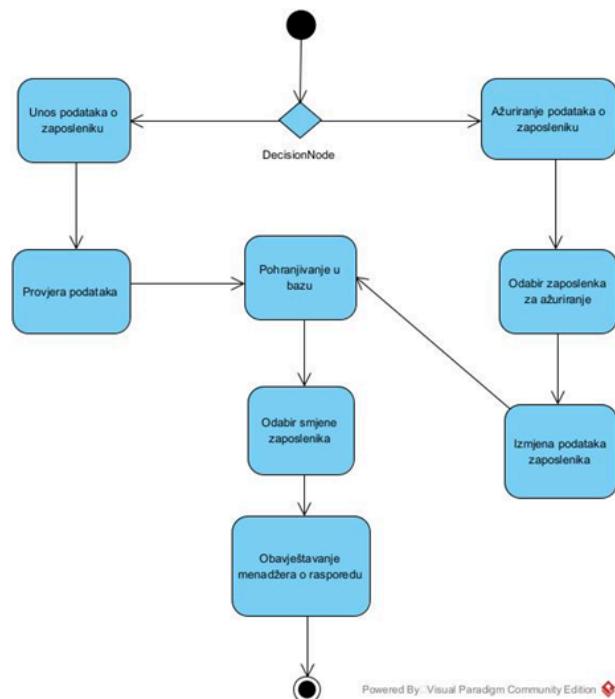
Proces izdavanja računa omogućava efikasno prikupljanje svih relevantnih informacija potrebnih za izdavanje računa, kao i transparentno praćenje transakcija za poslovne potrebe, na slici 12. će biti detaljno prikazan ovaj dijagram aktivnosti.



Slika 12. Dijagram aktivnosti računa

4. 2. 6. Dijagram aktivnosti upravljanja zaposlenicima

Dijagram aktivnosti za upravljanje zaposlenicima prikazuje ključne korake u procesu dodavanja novih zaposlenika, ažuriranja postojećih informacija te raspoređivanja smjena. Dijagram pomaže menadžmentu da prati i koordinira radne smjene, kao i da osigura ažurnu bazu podataka o zaposlenicima.

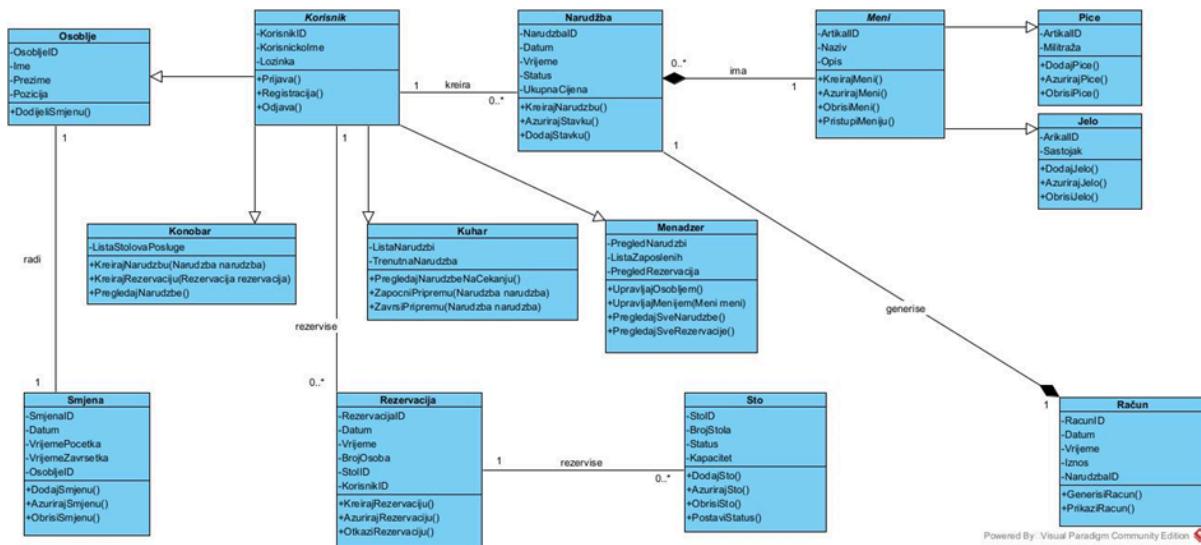


Slika 13. Dijagram aktivnosti upravljanja zaposlenika

4. 3. Dijagram klasa

Dijagram klasa je jedan od ključnih dijagrama u UML-u, koji prikazuje statičku strukturu sistema kroz klase, njihove atribute, metode i međusobne odnose. On pomaže u vizualizaciji i razumijevanju objektnog modela sistema, olakšavajući planiranje i implementaciju u narednim koracima izrade projekta.

Dijagram klasa u informacionom sistemu restorana omogućava jasan prikaz svih ključnih entiteta u sistemu, zajedno sa njihovim atributima i metodama. Ovaj dijagram koji je prikazan na slici 14. je od bitnog značaja za razvoj sistema jer omogućava efikasno upravljanje podacima i operacijama unutar restorana, kao i lakše održavanje i proširivanje sistema.



Slika 14. Prikaz dijagrama klasa informacionog sistema restorana

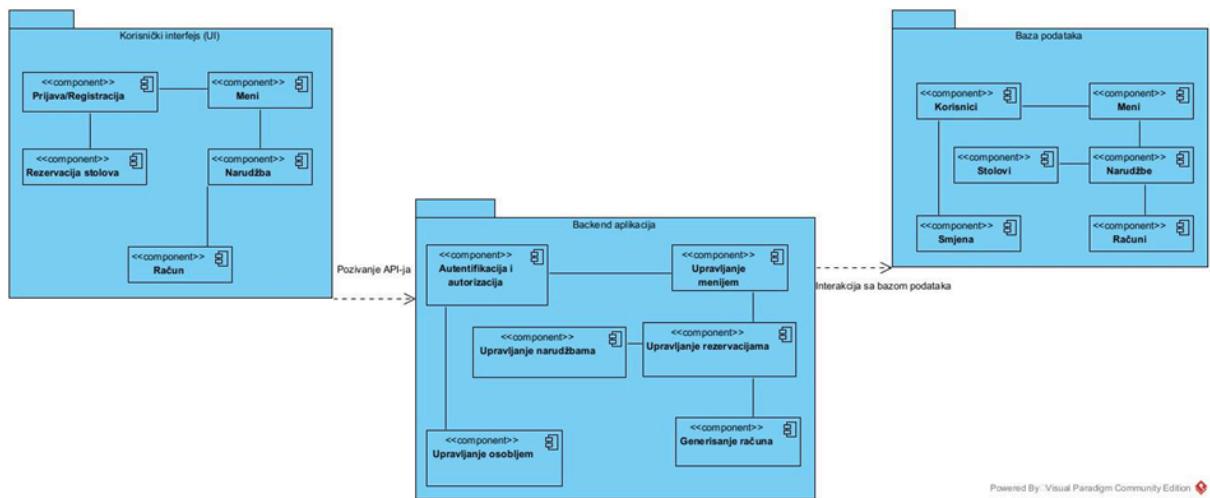
U okviru ovog dijagrama klasa osmišljen je princip strukture ovog informacionog sistema. Apstraktna klasa ‘Korisnik’ sadrži osnovne atribute i metode za rad s korisnicima sistema. Ova klasa je naslijedena ispred specifičnih uloga kao što su ‘Konobar’, ‘Kuhar’ i ‘Menadžer’, koje imaju dodatne specifične funkcionalnosti. Na taj način, korištenjem apstrakcije omogućeno je da se sve uloge korisnika tretiraju kroz zajednički okvir, ali sa određenom prilagodbom funkcionalnosti za svaku ulogu. Klasa ‘Narudžba’ povezana je s klasom Korisnik, jer svaki korisnik može kreirati narudžbu, a i svaka narudžba je povezana s odgovarajućim ‘Računom’ koji sadrži iznos i datum izdavanja računa. ‘Rezervacija’ omogućava korisnicima da rezervišu stolove, pri čemu je povezana s klasom ‘Sto’, koja predstavlja konkretne stolove u restoranu. Za sve ove veze postavljene su odgovarajuće kardinalnosti, kao što je npr. da jedan korisnik može imati više narudžbi, dok jedan sto može

biti rezervisan više puta. Klasa 'Meni' predstavlja jelovnik restorana i stavke menija (jelo i piće), sa atributima koji omogućavaju dinamičko dodavanje, uklanjanje i ažuriranje stavki. Ovom klasom restoran može upravljati svojim menijem i usklađivati ponudu s potrebama korisnika.

U navedenom dijagramu, korištena je generalizacija kao koncept apstrakcije, što omogućava dalje širenje sistema dodavanjem novih uloga ili funkcionalnosti bez potrebe za velikim promjenama u postojećem modelu. Dodatno, u dijagramu je korištena anotacija za označavanje veza između klasa, kao što su "kreira", "rezerviše", "generiše", itd., kako bi se jasno prikazale funkcije koje svaka klasa obavlja u sistemu.

4. 4. Dijagram komponenti

Dijagram komponenti prikazuju visok nivo arhitekture sistema i njegovih glavnih komponenata, kao i njihovih međusobnih odnosa. Ovi dijagrami omogućavaju razumijevanje kako se različite komponente unutar sistema međusobno povezuju i komuniciraju. Na slici 15. bit će prikazan dijagram komponenti informacionog sistema restorana.



Slika 15. Dijagram komponenti informacionog sistema restorana

Kreirane su tri glavne komponente: korisnički interfejs (UI), backend aplikacija i baza podataka. UI komponenta predstavlja sve što korisnik vidi i s čim komunicira, kao što su stranice za prijavu, registraciju, kreiranje narudžbi, menadžment stola i račun. Backend aplikacija služi za poslovnu logiku, kao što su autentifikacija korisnika, obrada narudžbi i povezanost sa bazom podataka. Baza podataka pohranjuje sve podatke sistema, uključujući informacije o korisnicima, meniju, narudžbama i rezervacijama. Nakon toga, dodane su

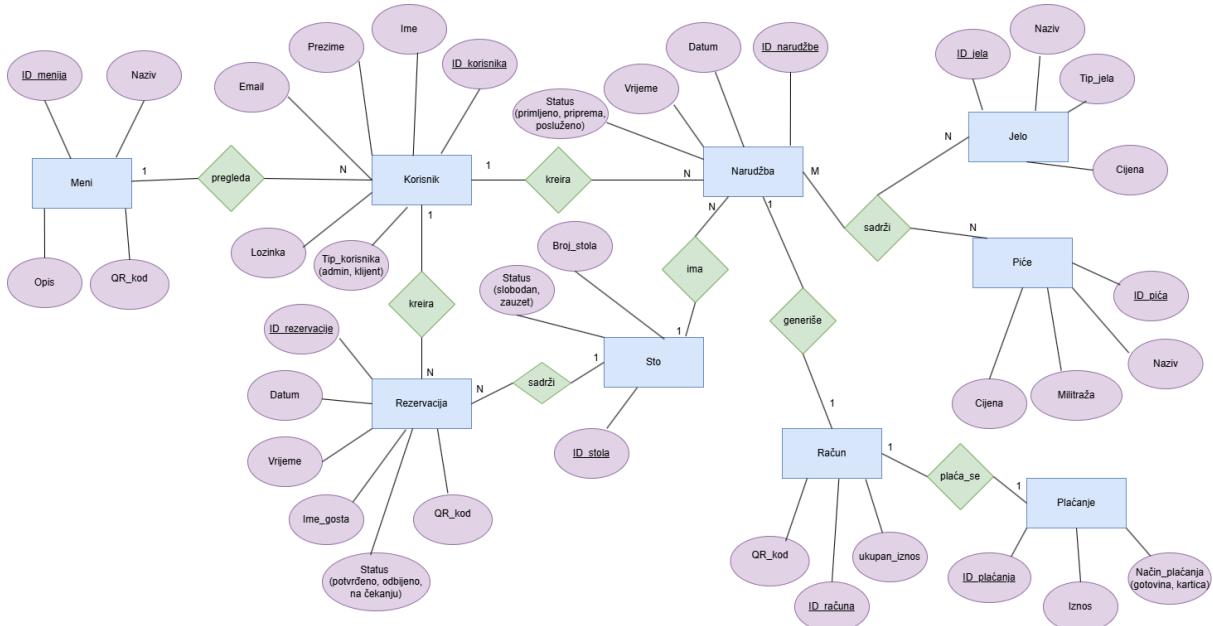
odgovarajuće podkomponente. Komponente smještene u paketima su povezane pomoću dependency veze, kako bi se jasno prikazale zavisnosti jedne od drugih.

5. SISTEMSKI DIZAJN

Značaj sistemskog dizajna za informacioni sistem restorana je veliki, jer omogućava efikasno i sigurno upravljanje svim aspektima poslovanja restorana. Pravilno dizajniran sistem omogućava restoranu da pruži brzo i kvalitetno korisničko iskustvo, poboljša interne procese, omogućava jednostavno praćenje finansija i omogućava zaštitu podataka.

5. 1. ER dijagram

Ovaj ER dijagram sistema upravljanja restoranima osmišljen je kako bi pružio potpuni i što jednostavniji pregled IT infrastrukture koja se koristi u restoranskom okruženju.

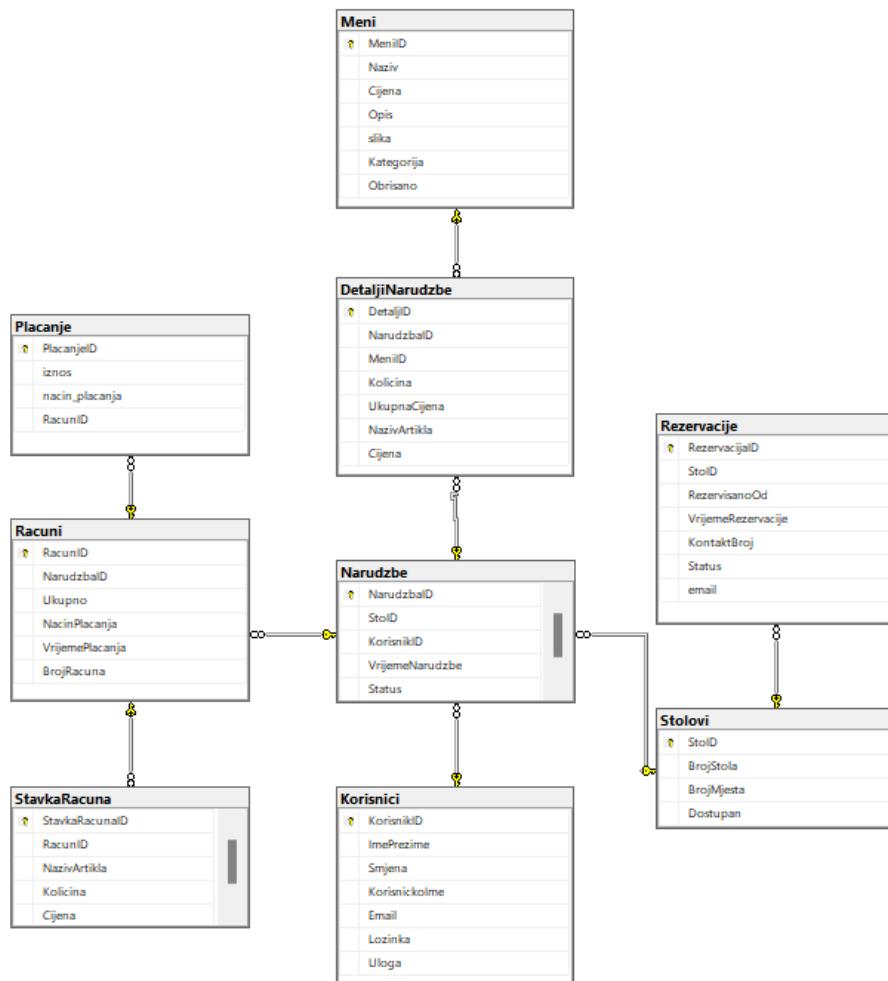


Slika 17. ER Dijagram restorana

Dijagram sadrži detaljan prikaz entiteta sa njihovim odgovarajućim atributima, kao i međusobnim vezama i numeracijom. Također pomaže da se prate podaci o klijentima, upravljanju u restoranu, kao i upravljanju finansijskim transakcijama. Može se koristiti za optimizaciju poslovanja restorana, optimizaciju resursa, smanjenje troškova i poboljšanje zadovoljstva kupaca. Nadalje, ovaj ERD daje detaljno razumijevanje komponenti sistema, omogućavajući administratorima da brzo diagnosticiraju, identificiraju i otklone tehničke probleme.

5. 2. Dizajn baze podataka

Dizajn baze podataka za ovaj informacioni sistem dizajniran je kako bi prikazao sve ključne entitete sistema i njihove međusobne veze. Ovaj dizajn može prikazati osnovne funkcionalnosti koje su važne za rad restorana, uključujući upravljanje korisnicima, menijem, stolovima, narudžbama, rezervacijama, i osobljem.



Slika 18. Prikaz dizajna baze podataka informacionog sistema restorana

Dizajn baze podataka prikazan na slici 18. je kreiran da omogući jednostavno praćenje i upravljanje različitim operacijama u restoranu, od prijave korisnika do rezervacije i naručivanja. Obuhvata nekoliko ključnih relacija, na način da je svaki entitet povezan s drugim u skladu sa pravilima. Korištene su veze: 1:1, 1:N i N:1 u zavisnosti od situacije.

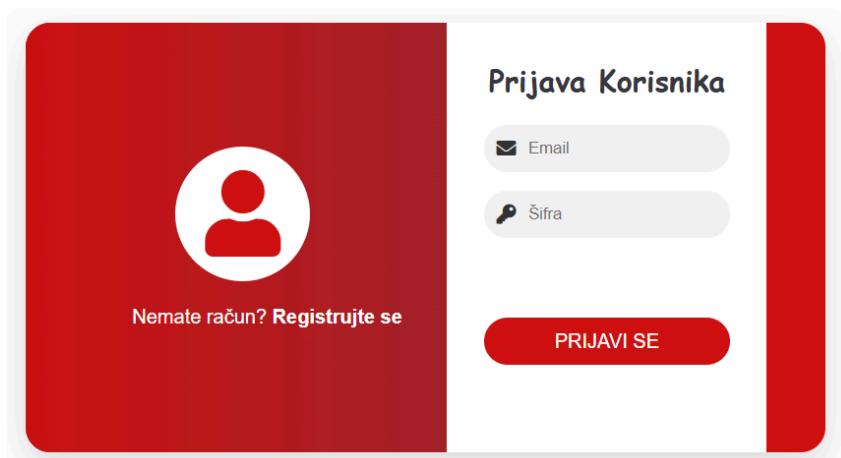
Entitet ‘Korisnik’ sadrži podatke za prijavu. Povezan je sa tabelom ‘Osoblje’ kroz relaciju 1:N što znači da jedan korisnik može biti povezan sa više članova osoblja. Osoblje sadrži podatke o zaposlenicima u restoranu. Osoblje je povezano sa tabelom ‘Korisnik’ (N:1) i ima vezu sa tabelom ‘Narudžba’ (1:N) jer jedan član osoblja može raditi na kreiranju više narudžbi. ‘Meni’ tabela sadrži ponudu restorana i ima vezu sa tabelama ‘Jelo’ i ‘Piće’ kroz relaciju 1:N. ‘Jelo’ i ‘Piće’ sadrže pojedinačne stavke menija. Povezane su sa tabelom ‘Meni’ kroz relaciju N:1, pošto više stavki može pripadati jednom meniju. Entitet ‘Sto’ je povezan sa tabelama ‘Narudžba’ i ‘Rezervacija’ kroz relacije 1:N, gdje jedan sto može imati više narudžbi i rezervacija u različitim periodima. Svaka narudžba je povezana sa jednim stolom (N:1) i može sadržati više stavki iz menija. ‘Rezervacija’ mogućava rezervaciju stolova za određeni datum i vrijeme. Povezana sa tabelom ‘Sto’ kroz relaciju N:1, omogućavajući više rezervacija za isti sto u različitim vremenskim intervalima.

5. 3. Dizajn interfejsa, formi, izvještaja i dijaloga putem mockup-a

U ovom poglavlju će biti izložene slike sa prikazom nekih dijelova interfejsa sistema, zajedno sa formama, izvještajima i dijalozima.



Slika 19. Logo restorana



Slika 20. Izgled forme za prijavu na sistem

The screenshot shows a registration form titled "Registracija Korisnika". The left side has a white background with a red vertical bar on the left and a red header bar at the top. The right side is a solid red color. The form fields are as follows:

- Ime i Prezime (Name and Surname) with a person icon
- Korisničko Ime (Username) with a user icon
- Email with an envelope icon
- Šifra (Password) with a lock icon

A red "REGISTRUJ SE" (Register) button is located at the bottom right of the red section.

Slika 21. Izgled forme prilikom registraciju na sistem

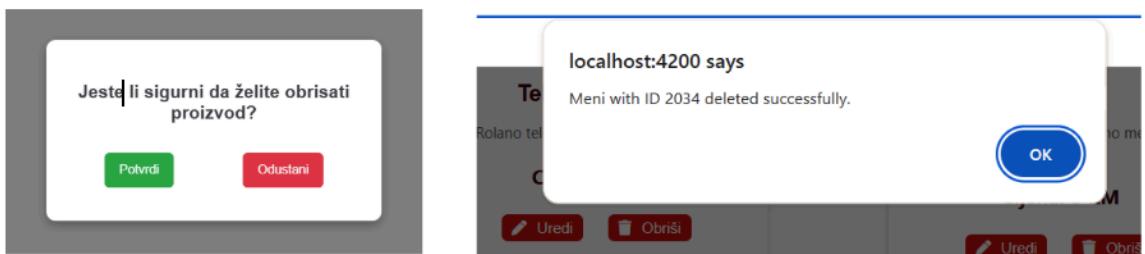
5. 3. 1. Dizajn web aplikacije za administratora

Ova web aplikacija predstavlja glavni alat za upravljanje radom restorana i njegovim korisnicima. Administrator sistema ima pristup svim ključnim funkcijama neophodnim za nesmetano funkcionisanje. Aplikacija je dizajnirana tako da omogući jednostavno i efikasno upravljanje velikim količinama podataka.



Slika 22. Dashboard menija kakav se prikazuje administratoru

Slika 23. Dizajn za uređivanje kartice menija



Slika 24. Dizajn prilikom brisanja određenog artikla sa menija

Slika 25. Dizajn forme za dodavanje određenog jela/pića na meni

Rezervacija ID	Sto	Rezervisao/la	Vrijeme Rezervacije	Kontakt Broj
2	Sto 2 - (6 mjestra)	Dina Zukic	12/21/24, 3:39 PM	062345678
3	Sto 2 - (6 mjestra)	Nedim Međić	12/20/24, 11:01 PM	063456789
5	Sto 2 - (6 mjestra)	Edina Edić	12/20/24, 11:02 PM	052447852
6	Sto 2 - (6 mjestra)	Berina Topalovic	12/21/24, 4:24 PM	123456789
7	Sto 2 - (6 mjestra)	Husna Hudić	12/21/24, 3:43 PM	784001455
8	Sto 1 - (4 mjestra)	Emir Hudić	12/21/24, 3:43 PM	784001744
9	Sto 2 - (6 mjestra)	Edo Edić	12/21/24, 9:35 AM	784354100
12	Sto 2 - (6 mjestra)	Esma Mehic	3/11/25, 5:22 PM	234567876
18	Sto 3 - (2 mjestra)	Senida Senić	6/24/25, 11:24 AM	2345432345

[-- Vrati se](#)

Slika 26. Prikaz svih rezervacija u restoranu sa odgovarajućim podacima

[-- Vrati se](#)

Narudžba 34				
Naziv jela	Količina	Cijena	Ukupno	
Cevapi	2	5.00	10.00	
<input type="button" value="Kreirana"/>	<input type="button" value="U pripremi"/>	<input type="button" value="Postavljeno"/>	<input checked="" type="button" value="Završeno"/>	<input type="button" value="Gotovina"/>
<input type="button" value="Kartica"/>	<input type="button" value="Generiši račun"/>			

Narudžba 35				
Naziv jela	Količina	Cijena	Ukupno	
Cappuccino	4	5.00	16.00	
<input type="button" value="Kreirana"/>	<input type="button" value="U pripremi"/>	<input type="button" value="Postavljeno"/>	<input checked="" type="button" value="Završeno"/>	<input type="button" value="Gotovina"/>
<input type="button" value="Kartica"/>	<input type="button" value="Generiši račun"/>			

Narudžba 36				
Naziv jela	Količina	Cijena	Ukupno	
Cappuccino	4	5.00	16.00	
<input type="button" value="Kreirana"/>	<input type="button" value="U pripremi"/>	<input type="button" value="Postavljeno"/>	<input checked="" type="button" value="Završeno"/>	<input type="button" value="Gotovina"/>
<input type="button" value="Kartica"/>	<input type="button" value="Generiši račun"/>			

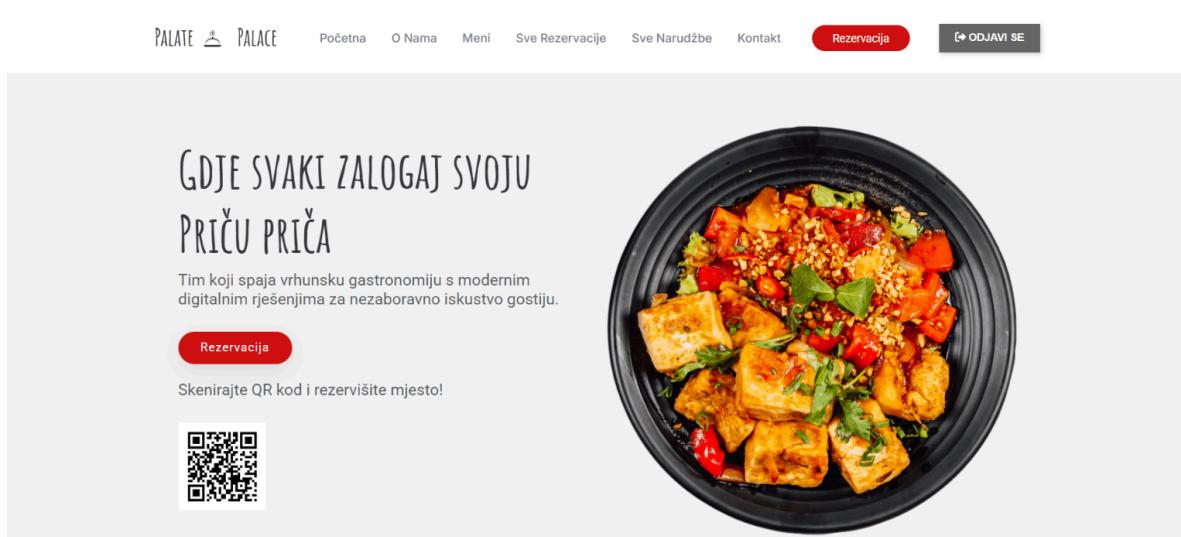
Narudžba 37				
Naziv jela	Količina	Cijena	Ukupno	
Pileća salata	2	12.00	24.00	
Cappuccino	1	5.00	4.00	
Topla cokolada	1	10.00	10.00	
<input type="button" value="Kreirana"/>	<input type="button" value="U pripremi"/>	<input type="button" value="Postavljeno"/>	<input checked="" type="button" value="Završeno"/>	<input type="button" value="Gotovina"/>
<input type="button" value="Kartica"/>	<input type="button" value="Generiši račun"/>			

Slika 27. Prikaz narudžbi u restoranu

5. 3. 2. Dizajn web aplikacije za korisnike

Web aplikacija za korisnike predstavlja njihov glavni alat za pretraživanje ponude restorana, rezervisanje, praćenje narudžbi, pregled računa, plaćanje i kontakt informacije. Aplikacija je kreirana tako da bude jednostavna za korištenje bez potrebe za dodatnim tehničkim znanjem. Korisnicima treba omogućiti brz pristup informacijama o jelovniku, dostupnim terminima za

rezervaciju, te mogućnost online narudžbi i plaćanja.



Slika 28. Prikaz početne stranice

A screenshot of the 'O nama' (About Us) section of the website. It features a large image of the restaurant's interior with wooden tables, chairs, and hanging lamps. Below the image is a box containing text about being a digital partner and a QR code. To the right is a section titled 'Saznajte više O nama' (Learn more about us) which includes a quote from Radmila, a testimonial from John Doe, and a section showing various breakfast items with points values: +50 for Radmila, +100 for Irena, +200 for Radmira, and +10 for Bojan.

Slika 29. Prikaz jednog dijela stranice 'O nama' za ISR

A screenshot of the second part of the 'O nama' section. It shows a testimonial from Radka Wigand, a contact form with fields for name, email, message, and a file attachment, and social media icons for Facebook, Instagram, and YouTube. At the bottom, there is footer information including address, phone number, email, opening hours, and a copyright notice.

Slika 30. Prikaz drugog dijela stranice 'O nama' za ISR

MENI

Pogledajte naš **Palate Palace meni**

 Kreiraj narudžbu

Jelo Piće

MENI

JELA



Pizza Margherita

Tijesto za pizzu, umak od paradajza, mozzarela, zelenina i moštovački ulje

Cijena: 15 KM

 Uredi  Obrisi



Špageti

Špageti u paradajz sosu

Cijena: 8 KM

 Uredi  Obrisi



Orada

Orada pripremljena na gradelima

Cijena: 45 KM

 Uredi  Obrisi



Pohovane lignje

Pohovane lignje u tjestu

Cijena: 25 KM

 Uredi  Obrisi

Slika 31. Prikaz jednog dijela stranice 'Meni' za ISR

PIĆA



Coca-Cola

Limenka od 0.33L
Cijena: 3.5 KM

 Uredi  Obrisi



Espresso kafa

Sveže pripremljena espresso kafa
Cijena: 3 KM

 Uredi  Obrisi



Limunada

Limon
Cijena: 4.5 KM

 Uredi  Obrisi



Topla cokolada

Vrlo ukusna i čokolanda
Cijena: 10 KM

 Uredi  Obrisi



Cijedena naranča

Pripremljeno na domaći način
Cijena: 7 KM

 Uredi  Obrisi



Cappuccino

sa mlijekom
Cijena: 5 KM

 Uredi  Obrisi



Jogurt

Jogurt, šumska vođa, müslji
Cijena: 4 KM

 Uredi  Obrisi



Sok od jagode

Prirodni sok od jagode
Cijena: 3.5 KM

 Uredi  Obrisi

Slika 32. Prikaz drugog dijela stranice 'Meni' za ISR

Pripremljeno na domaći način

Cijena: 7 KM

 Uredi  Obrisi

sa mlijekom

Cijena: 5 KM

 Uredi  Obrisi

Jogurt, šumska vođa, müslji

Cijena: 4 KM

 Uredi  Obrisi

Prirodni sok od jagode

Cijena: 3.5 KM

 Uredi  Obrisi

+ Dodaj Jelo/Pice

 Lokacija

New York, NY 535022
A108 Mike Street

 Kontakt inf.

Tel: +1 5589 55488 55
Email: palate-palace.com

 Radno Vrijeme

Pon-Sub: 11:00h - 23:00h
Ned: Neradno

Pratite Nas

© 2025 All rights reserved.
Designed by [Borislava Topalović](#)

Slika 33. Prikaz trećeg dijela stranice 'Meni' za ISR

Vrati se

Palate[△] Palace

Forma za Kreiranje Narudžbe

Ukupna cijena:

+ Dodaj X Poništi

✓ Potvrdi X Poništi

Slika 34. Prikaz forme za kreiranje narudžbe

Vrati se

Palate[△] Palace

Forma za Kreiranje Narudžbe

Naziv jelapića:

Količina: X

Naziv jelapića:

Količina: X

Ukupna cijena:

+ Dodaj X Poništi

✓ Potvrdi X Poništi

Slika 35. Prikaz forme za kreiranje više narudžbi

Palate[△] Palace

Početna O nama Meni Kontakt Rezervacija

REZERVACIJA

Želite uskoro i zdravo? **Tu smo za Vas!**

Ime i Prezime:
Međi/yyy:
Email:

Broj Telefona:
Broj Ljudi:

REZERVIRAJ

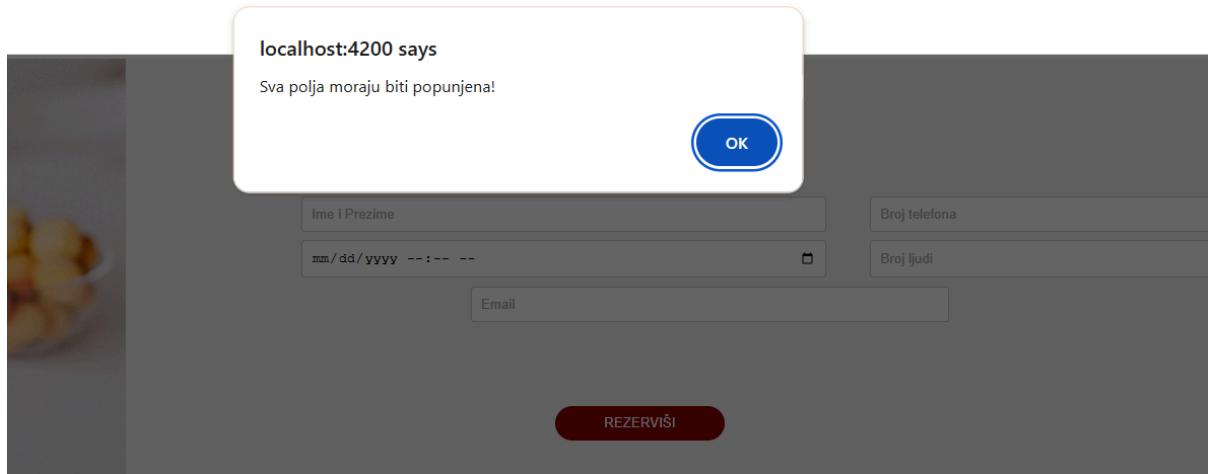
Lokacija
New York, NY 10022
A108 Milk Street

Kontakt Inf
Tel: +1 5589 55488 15
Email: palate-palace.com

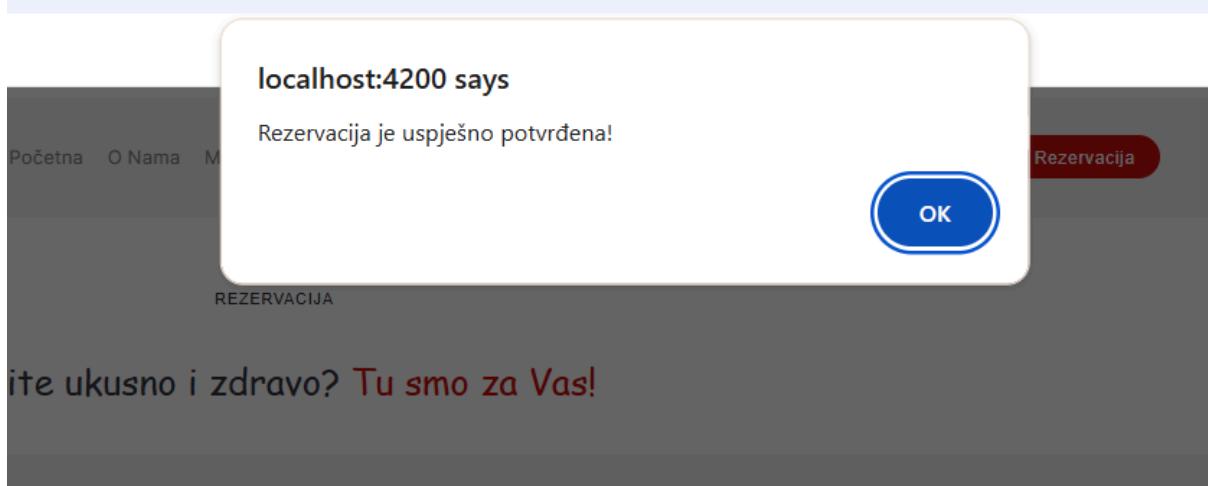
Račno Vrijeme
Pon-Sub: 11:00h - 23:00h
Net: Nasređe

Pritisci Na: Twitter Facebook Instagram YouTube

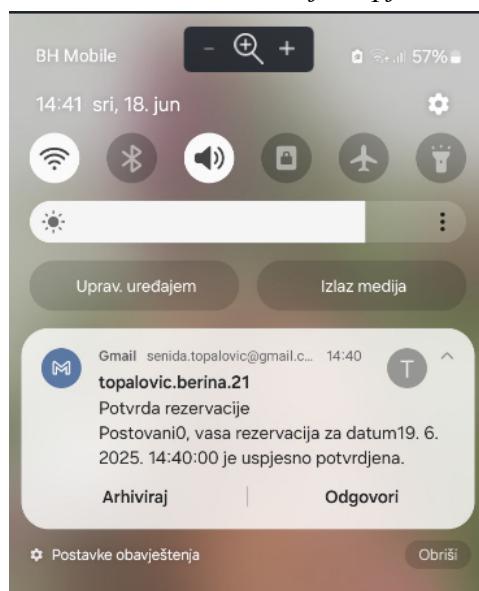
Slika 36. Prikaz stranice i forme za rezervaciju



Slika 37. Prikaz rezultata u slučaju neuspješne rezervacije



Slika 38. Prikaz rezultata u slučaju uspješne rezervacije



Slika 39. Prikaz notifikacije na mobilnom uređaju u obavijestima o uspješnosti rezervacije

Pizza Order
\$20.00

Pay with link

Or

Email: email@example.com

Card information:

1234 1234 1234 1234	
MM / YY	CVC

Cardholder name: Full name on card

Country or region: Bosnia & Herzegovina

Securely save my information for 1-click checkout
Pay faster on this site and everywhere Link is accepted.

Pay

Powered by stripe

Slika 40. Prikaz panela za plaćanje pomoći Stripe simulacije

Palate[®] Palace

New York, NY 535022
+1 5589 55488 55

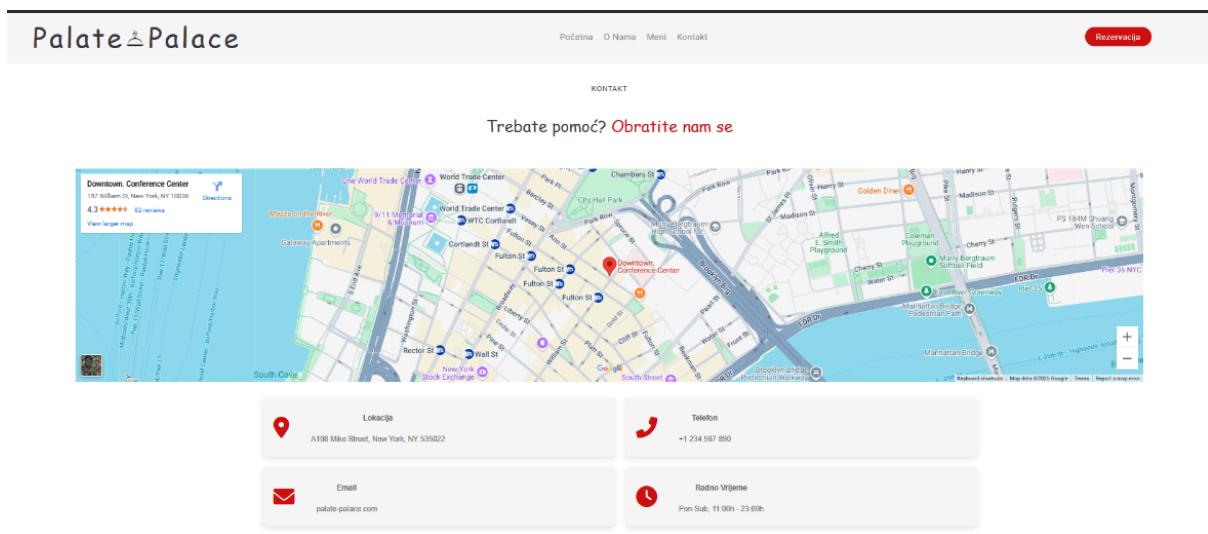
- ID Računa: R-638863883477329177
- ID Narudžbe: 34
- Datum i Vrijeme: 2025-06-24 18:59
- Status: Završeno

#	Naziv	Količina	Jedin. Cijena	Ukupna Cijena
1	Cevapi	2	5 KM	10 KM

Ukupno: 10 KM

PLATI

Slika 41. Prikaz generisanog računa određene narudžbe



Slika 42. Kontakt stranica ISR-a

6. ARHITEKTURA SISTEMA

U razvoju informacionog sistema za restoran primijenjena je **troslojna arhitektura**. Slojevi su međusobno povezani jedan sa drugim, ali i odvojeni kako bi omogućili fleksibilniji razvoj sistema. Ova struktura obuhvata tri glavna sloja:

- Prezentacijski sloj
- Poslovni sloj
- Sloj baze podataka

Ova vrsta arhitekture odabрана је јер омогућава развој система као web aplikacije уз једноставно одржавање и fleksibilnost. Помјене у једном слоју (нпр. prezentacijski sloj) не захтјевају помјене у другим слојевима (нпр. poslovnom sloju ili sloju baze podataka), што олакшава проширење и одржавање система.

Осим тога, раздвојеност слојева омогућава боље управљање приступом подацима, чиме се смањују потенцијалне грешке, а то знатно побољшава укупну сигурност система. API додатно побољшава сигурност тако што служи као зашtitni слој између бaze података и крајnjih корисника (prezentacijskog sloja), филтрирајући све заhtjeve приje nego što дођу до бaze података.

6. 1. Prezentacijski sloj

Prezentacijski sloj обухвата frontend dijelove aplikacije, код који генерише кориснички интерфејс и омогућава интеракцију с корисником. Према Springer извору, "frontend је интерфејс који повезује корисника с backend-ом и обухвата оно што корисник вidi и с чиме може комуникацији" [1]. Подаци за приказ долазе из poslovног слоја, али frontend nije usko vezan за njega. То значи да помјене у poslovном слоју неће nužno uticati na način na koji se podaci prikazuju u корисничком интерфејсу.

Ovaj слој укључује и ову израђену **web aplikaciju**, при чему се комуникаира с posловним слојем путем API-ја. Овaj слој представља интерфејс између корисника и система.

6. 2. Poslovni sloj

Poslovni слој или API слој, служи за obradu података и implementaciju poslovne logike. У ovom слоју se odvija dohvatanje podataka iz baze podataka, njihova obrada, te slanje

rezultata prema prezentacijskom sloju. Poslovni sloj ima direktnu komunikaciju sa slojem baze podataka i sa prezentacijskim slojem, ali ovi slojevi ostaju nezavisni.

API (Application Programming Interface) u ovom slučaju djeluje kao posrednik između baze podataka i frontend/mobilne aplikacije. API prima zahtjeve sa prezentacijskog sloja, obrađuje ih (u skladu sa poslovnom logikom) i odgovara podacima iz baze. Na ovaj način, isti API se može koristiti i za web i za mobilne aplikacije, što olakšava upravljanje aplikacijom i proširivanje funkcionalnosti.

6. 3. Sloj baze podataka

Sloj baze podataka čini baza u kojoj se nalaze svi podaci za rad sistema. Baza je strukturisana prema ER dijagramu, uz izvršenu normalizaciju radi optimizacije performansi i smanjenja redundancije podataka. Poslovni sloj pristupa bazi podataka kako bi dohvatao, obrađivao ili ažurirao podatke, koji se potom prikazuju korisniku kroz prezentacijski sloj.

7. TEHNOLOGIJE ZA BACKEND

Backend tehnologije čine osnovu informacionog sistema restorana, iako nisu vidljive krajnjem korisniku. One omogućavaju obradu zahtjeva koji dolaze s korisničkog interfejsa, upravljaju poslovnom logikom, komuniciraju s bazom podataka i osiguravaju sigurnu razmjenu podataka.

U ovom slučaju koristi se troslojna arhitektura, gdje je backend smješten u **poslovnom sloju**, između korisničkog interfejsa i baze podataka. Za realizaciju je odabrana web aplikacija, pri čemu backend prihvata zahtjeve korisnika, izvršava potrebne obrade i vraća odgovore putem API-ja. Korištenjem pouzdanih tehnologija, backend omogućava efikasan, siguran i stabilan rad sistema, čak i pri većem broju korisnika i kompleksnijim zadacima.

7. 1. ASP .NET

ASP.NET je server-side web framework kojeg je razvio **Microsoft** i omogućava izradu dinamičkih web aplikacija i servisa koristeći moderne paradigme za web-razvoj [2]. Izvodi se na .NET platformi, dajući pristup C# (ili drugim .NET jezicima), te omogućava korištenje moćnih biblioteka za autentifikaciju, autorizaciju i rad sa HTTP zahtjevima što je i iskorišteno u ovom diplomskom radu (.NET Core Identity Management) [3][4].

Dolaskom ASP.NET Core verzije, framework je postao **open-source**, **cross-platform** i **modularan**, omogućavajući razvoj aplikacija za Windows, Linux i macOS [5][6].

7. 2. Swagger UI

API je testiran koristeći Swagger, alat koji omogućava vizualizaciju i interakciju s resursima API-ja bez potrebe za implementacijom kompletne logike. Swagger UI se automatski generiše iz OpenAPI specifikacije, pružajući preglednu i vizualnu dokumentaciju koja olakšava razvoj backend sistema i korištenje API-ja na klijentskoj strani. Zahvaljujući Swaggeru, mogu se jednostavno pregledati i testirati funkcionalnosti API-ja, što doprinosi bržem razvoju i otklanjanju eventualnih grešaka. [7]

7. 3. MSSQL server

MSSQL (Microsoft SQL Server) je odličan sistem za upravljanje relacijskim bazama podataka. Ova tehnologija pruža sigurno i efikasno skladištenje podataka te njihovu manipulaciju, što je od velike važnosti za informacioni sistem restorana. Ključne karakteristike MSSQL-a uključuju visoke performanse, omogućavanje brzog izvršavanja upita i rukovanje velikim količinama podataka, kao i podršku za složene upite. MSSQL podržava napredne funkcionalnosti i nativnu integraciju s C# kroz biblioteke poput Entity Frameworka, što olakšava upravljanje podacima u aplikacijama razvijenim u ovom programskom jeziku. Njegova pouzdanost i skalabilnost omogućavaju dugoročnu stabilnost sistema [7].

7. 4. Visual Studio kao razvojno okruženje za backend

Visual Studio je jedno od najpopularnijih integrisanih razvojnih okruženja (IDE) za razvoj aplikacija. Razvijen od strane Microsofta, Visual Studio pruža sveobuhvatne alate i funkcionalnosti za rad s različitim tehnologijama, uključujući C#, ASP.NET i MSSQL, što ga čini idealnim izborom za razvoj backend dijela ovog informacionog sistema.

8. TEHNOLOGIJE ZA FRONTEND

Frontend tehnologije predstavljaju vidljivi dio aplikacije s kojim krajnji korisnici direktno komuniciraju. One su odgovorne za kreiranje korisničkog interfejsa (UI), omogućavanje interakcije, te osiguranje da korisnici imaju jednostavno i intuitivno iskustvo prilikom korištenja aplikacije. Cilj frontenda je povezivanje korisnika s backendom na način koji je vizuelno privlačan i funkcionalno efikasan.

8. 1. Angular

Za izradu frontend dijela sistema korišten je Angular, pogodan web framework koji omogućava razvoj brzih, responzivnih i skalabilnih aplikacija. Angular je razvijen i održava ga Google, te pruža bogat skup alata, komponenti i API-ja koji značajno ubrzavaju razvojni proces i olakšavaju održavanje aplikacija. Zahvaljujući modularnoj strukturi i podršci za dvosmjernu vezu podataka, Angular je pogodan za razvoj složenih web aplikacija koje zahtijevaju visoku interaktivnost i jednostavnu nadogradnju [9].

8. 2. HTML, CSS

Uz Angular, u izradi korisničkog interfejsa korišteni su i HTML i CSS. HTML (HyperText Markup Language) služi za strukturisanje sadržaja na web stranici, dok CSS (Cascading Style Sheets) omogućava stilizaciju tog sadržaja, dajući mu vizualni izgled. Ove tehnologije zajedno sa Angularom omogućavaju kreiranje preglednog i estetski ugodnog korisničkog interfejsa.

8. 3. Visual Studio Code

Visual Studio Code je popularno, lagano i prilagodljivo razvojno okruženje za frontend i backend razvoj. Pruža brojne funkcionalnosti koje olakšavaju rad s HTML-om, CSS-om i JavaScriptom. Korištenje Visual Studio Code-a olakšava rad na ovim elementima, zahvaljujući preglednom sučelju i naprednim alatima za produktivnost. Kombinacija HTML-a, CSS-a i VS Code-a omogućava razvoj atraktivnog, funkcionalnog i korisniku prilagođenog sistema koji efikasno zadovoljava potrebe restorana.

9. UPRAVLJANJE IMPLEMENTACIJOM INFORMACIONOG SISTEMA RESTORANA

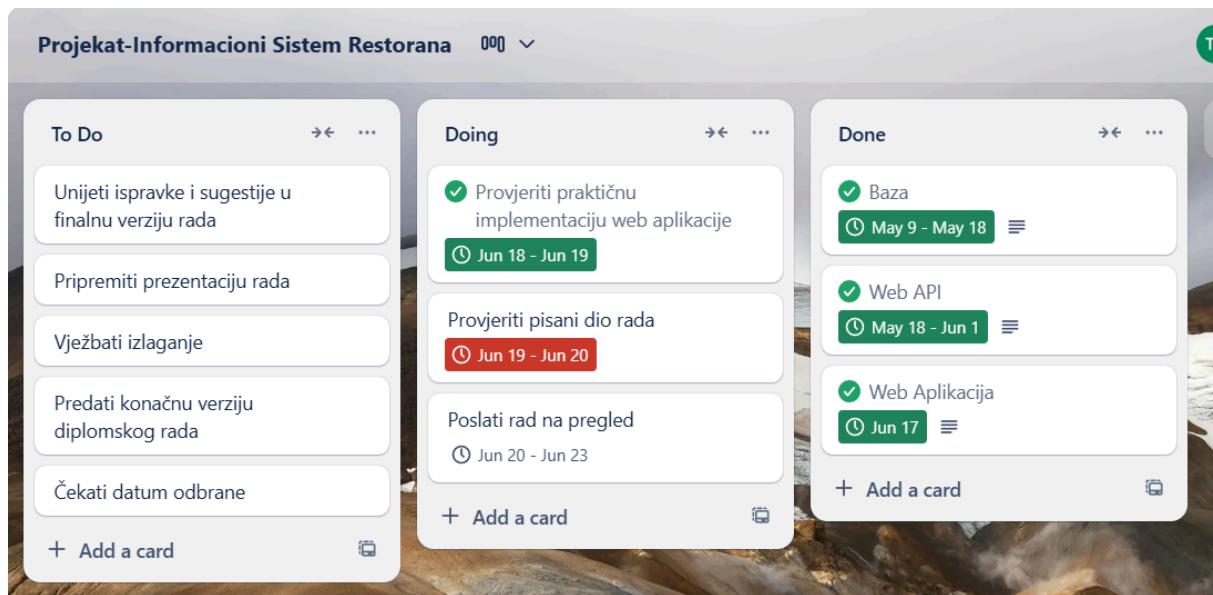
U okviru ovog projekta, upravljanje i praćenje svih faza razvoja obavljeno je korištenjem **Trello** alata. Trello je softverski alat za organizaciju zadataka i upravljanje projektima, koji koristi sistem ploča, listi i kartica za vizualno praćenje izrade projekta. Alat je posebno pogodan za samostalne i manje timske projekte zbog svoje jednostavnosti i fleksibilnosti, pa je zbog toga izabran i za praćenje ovog projekta. Trello omogućava korisnicima da jednostavno kreiraju zadatke, postavljaju rokove, prate napredak i organizuju radni tok na pregledan i intuitivan način. Zahvaljujući mogućnosti dodavanja napomena i komentara, Trello pomaže u efikasnom upravljanju svim aktivnostima vezanim za projekat. [9]

9. 1. Pregled liste zadataka u Trello alatu

Na slici 43. prikazana je lista zadataka unutar Trello alata, koja obuhvata glavne komponente informacionog sistema restorana. Svaki zadatak predstavljen je kao kartica, sa jasno definisanim naslovom, opisom, vremenskim rokovima i statusom, organizovanim unutar odgovarajućih lista (npr. To Do, Doing, Done). Svi zadaci vezani za izradu projekta su označeni kao ‘DONE’, što znači da su uspješno realizovani. Ova struktura omogućava jasan pregled svih ključnih faza projekta, kao i efikasno praćenje napretka razvoja sistema.

Kategorije zadataka:

- **Baza:** Ova grupa zadataka odnosi se na dizajn i implementaciju baze podataka sistema restorana. Zadaci uključuju analizu zahtjeva, definisanje entiteta i atributa, normalizaciju i implementaciju svih potrebnih veza i ograničenja;
- **Web API:** Zadaci u ovoj kategoriji obuhvataju razvoj API-ja koji omogućava komunikaciju između korisničkog interfejsa (web aplikacije) i serverske logike. To uključuje API rute za svaku od kategorija;
- **Web Aplikacija:** Zadaci u ovoj grupi odnose se na razvoj web aplikacije za restoran, koja sadrži stranice za različite korisničke uloge, uključujući administratore i klijente.



Slika 43. Pregled liste kategorija

9. 2. Baza

Na slici 44. prikazan je dio projekta koji pokriva sve glavne faze kreiranja baze podataka, od analize zahtjeva i identifikacije entiteta do definisanja atributa entiteta. Također, zadaci poput definisanja ključeva, ograničenja i odnosa između entiteta i njihovih ažuriranja od velike su važnosti za održavanje integriteta baze podataka, dok je normalizacija baze podataka izvršena kako bi se optimizovala baza, smanjili duplikati i poboljšala efikasnost u radu.

Baza

+ Add Labels Checklist Members Attachment

Dates
May 9 - May 18, 5:39 PM Complete

Description Edit

Zadaci:

1. Analiza zahtjeva i identifikacija entiteta
2. Definisanje atributa entiteta
3. Ključevi, ograničenja i odnosi
4. Normalizacija baze
5. Ažuriranja i dorade

Slika 44. Prikaz liste zadataka za kreiranje baze podataka

9.3. Web API

Na slici 45. prikazan je dio projekta koji obuhvata razvoj i implementaciju **API rutiranja** za ključne funkcionalnosti informacionog sistema restorana. Ovi API zadaci omogućavaju pouzdanu i sigurnu komunikaciju između korisničkog interfejsa (frontend) i serverskog dijela aplikacije (backend), osiguravajući efikasno rukovanje podacima unutar sistema. U okviru ovog segmenta projekta implementirani su različiti API-jevi, uključujući:

- **Dizajn i implementacija API rutiranja za funkcionalnosti sistema:** Zadatak predstavlja osnovu za API sistem, gdje je definisano glavno rutiranje koje omogućava siguran i efikasan pristup svim funkcionalnostima sistema;
- **API za prijavu, registraciju i odjavu korisnika:** Razvijen je API koji omogućava korisnicima prijavu na sistem, kreiranje korisničkih naloga i brisanje određene sesije korisnika sa sistema. Ova funkcionalnost obuhvata siguran prijenos podataka, te autentifikaciju korisnika;
- **API za upravljanje stolovima:** Odnosi se na obradu i upravljanje podacima o stolovima u restoranu u kontekstu rezervacija. Funkcionalnosti ovog API-ja omogućavaju kreiranje, izmjenu i brisanje zapisa o stolovima, kao i provjeru njihove dostupnosti u realnom vremenu, čime se olakšava proces planiranja i dodjele mjesta gostima;
- **API za upravljanje rezervacijama:** Obuhvata kreiranje API rutiranja koje omogućava korisnicima i osoblju da evidentiraju nove rezervacije, izvrše izmjene postojećih termina ili otkažu rezervaciju. Ova funkcionalnost je ključna za optimalno korištenje prostora restorana i unapređenje korisničkog iskustva kroz efikasno upravljanje dolascima gostiju;
- **API za upravljanje narudžbama:** Ova funkcionalnost obuhvata automatski prijenos narudžbi prema kuhinji, statusno praćenje („u pripremi“, „spremno“, „posluženo“) i administrativno upravljanje svim aspektima naručivanja;
- **API za upravljanje menjem:** Omogućava fleksibilno upravljanje ponudom jela i pića restorana. Funkcionalnosti uključuju kreiranje novih stavki, ažuriranje cijena i opisa, brisanje zastarjelih unosa, te pregled postojećih kategorija i artikala, čime se osigurava da korisnici i osoblje uvijek imaju pristup ažuriranim informacijama;
- **API za upravljanje računima:** Obuhvata kreiranje i pregled računa za svaku narudžbu ili grupu narudžbi, generisanje ukupnog iznosa za plaćanje, kao i praćenje

statusa svakog računa. Ova funkcionalnost je od suštinske važnosti za organizaciju finansijskih tokova unutar restorana;

- **API za upravljanje plaćanjem:** Omogućava realizaciju procesa plaćanja putem gotovine ili kartice. Ovaj API upravlja potvrdom uspješnog plaćanja, povezivanjem s računima i opcijama kao što su izdavanje fiskalnih računa, te evidentiranje svake transakcije u sistemu, čime se omogućava potpuna finansijska kontrola i transparentnost.

The screenshot shows a project management interface with the following details:

- Done ▾** button at the top left.
- Web API** title with a checkmark icon.
- Buttons:** + Add, Labels, Checklist, Members, Attachment.
- Dates:** May 18 - Jun 1, 5:39 PM, Complete.
- Description:** A list of tasks under the heading "Zadaci":
 1. Dizajn i implementacija API rutiranja za funkcionalnosti sistema
 2. API za prijavu, registraciju i odjavu korisnika
 3. API za upravljanje stolovima
 4. API za upravljanje rezervacijama
 5. API za upravljanje narudžbama
 6. API za upravljanje računima
 7. API za upravljanje plaćanjem
 8. API za upravljanje menijem
- Edit** button at the bottom right of the description area.

Slika 45. Prikaz liste zadataka za kreiranje web API-ja

9. 4. Web aplikacija

Na slici 46. prikazan je dio projekta koji se odnosi na sve neophodne zadatke prilikom izrade web aplikacije za informacioni sistem restorana. Web aplikacija je strukturirana tako da uključuje posebne stranice za administratora i različite vrste klijenata. **Administratorska stranica** predstavlja centralnu tačku za upravljanje čitavim sistemom, uključujući kontrolu nad korisnicima, narudžbama, rezervacijama i menijem. Stranice za klijente, omogućavaju pristup funkcionalnostima koje se odnose na evidentiranje narudžbi, online rezervacije stolova i hrane. Sve stranice su razvijene s posebnim fokusom na jednostavnost korištenja, preglednost, sigurnost, i prilagodljivost različitim ulogama korisnika unutar restoranskog sistema.

Done ▾

✓ Web Application

+ Add Labels Checklist Members Attachment

Due date

Jun 17, 12:00 AM Complete ▾

☰ Description

Edit

Zadaci:

1. Stranica za administratora (*predstavlja mjesto za upravljanje korisnicima, zaposlenima, menijem, narudžbama i rezervacijama-pregled, izmjene i brisanja*)
2. Stranica za klijente (*omogućava mogućnost slanja narudžbi, slanja rezervacija, mogućnost online plaćanja, pregled početne stranice, stranice o nama i kontakt stranice*)
3. Razvoj izgleda stranica (*implementacija intuitivnog korisničkog interfejsa*)
4. Razvoj funkcionalnosti stranica (*Upravljanje rezervacijama – kreiranje, pregled, izmjena i otkazivanje rezervacija stolova, Upravljanje narudžbama – dodavanje novih narudžbi, izmjena statusa narudžbe i pregled prethodnih narudžbi, Upravljanje korisnicima – registracija, prijava, dodjeljivanje uloga, implementacija metode plaćanja uz stripe simulaciju, implementacija generisanja računa*)

Slika 46. Prikaz liste zadataka za kreiranje web aplikacije

U okviru izrade web aplikacije za informacioni sistem restorana definisani su zadaci koji se odnose na razvoj ključnih stranica i funkcionalnosti:

- **Stranica za administratora:** Dizajnirana je kao centralna kontrolna tačka sistema, gdje administrator može upravljati korisnicima, zaposlenima, menijem, narudžbama i rezervacijama. Stranica sadrži pregledne tabele, forme za unos i izmjenu podataka, kao i funkcionalnosti za filtriranje i pretragu informacija;
- **Stranica za klijente:** Omogućava korisnicima da na jednostavan način izvrše rezervaciju, pošalju narudžbu, pregledaju meni i izvrše online plaćanje. Izgled stranice je prilagođen korisnicima, sa jasnim dugmadima i jednostavnom navigacijom;
- **Razvoj izgleda stranica:** Obuhvata kreiranje intuitivnog korisničkog interfejsa sa savremenim dizajnom, jasnim rasporedom elemenata i jednostavnim vizuelnim komponentama koje olakšavaju korištenje aplikacije;
- **Razvoj funkcionalnosti:** Uključuje povezivanje stranica sa bazom i backendom, implementaciju rezervacija, narudžbi, upravljanje korisnicima i plaćanja. Posebna

pažnja je posvećena sigurnosti, validaciji podataka i jasnoći ulogama korisnika (admin, konobar, kuhar, gost).

10. PRAKTIČNA IMPLEMENTACIJA SISTEMA

U nastavku će biti prikazana praktična implementacija informacionog sistema za restoran kroz strukturu foldera projekta. Organizacija projekta podijeljena je na nekoliko ključnih komponenti: baza podataka, serverski API, te web aplikacija. Svaka od ovih komponenti ima jasno definisani strukturi foldera, što omogućava jednostavno održavanje sistema, lakšu navigaciju kroz kod i jasno razdvajanje odgovornosti.

Struktura foldera osmišljena je na način koji olakšava dalji razvoj i proširenje sistema, kao i timski rad, iako je ovaj projekat realizovan samostalno. Poseban fokus stavljen je na modularnost – API servisi su odvojeni od korisničkog interfejsa, dok je pristup bazi centralizovan i organizovan kroz posebne rezervatorije i servisne klase.

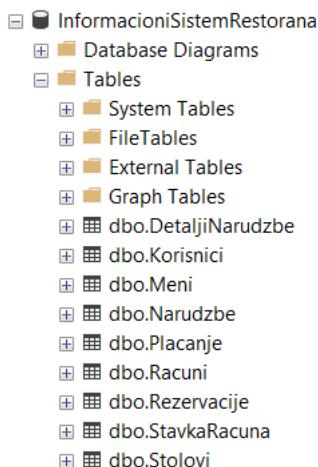
10. 1. Baza podataka

Na slici 47. prikazana je struktura baze podataka pod nazivom **InformacioniSistemRestorana**, koja je kreirana u Microsoft SQL Server-u. Baza sadrži više tabele koje zajedno omogućavaju funkcionalan rad informacionog sistema restorana. Svaka tabela ima svoju jasno definisani ulogu i odgovara određenom segmentu poslovanja restorana. U nastavku su opisane glavne tabele baze:

- **dbo.Korisnici:** Tabela u kojoj se nalaze podaci o svim korisnicima sistema, uključujući administratore, konobare i druge zaposlene;
- **dbo.Meni:** Ova tabela sadrži podatke o ponudi restorana – jela i pića;
- **dbo.Stolovi:** Tabela sadrži informacije o stolovima u restoranu, uključujući broj stola, broj sjedišta i status stola (slobodan/zauzet). Koristi se prilikom rezervacija i rasporeda sjedenja;
- **dbo.Rezervacije:** U ovoj tabeli se čuvaju informacije o rezervacijama stolova koje su kreirali korisnici ili administrator;
- **dbo.Narudzbe:** Tabela koja bilježi sve narudžbe gostiju;
- **dbo.DetaljiNarudzbe:** Ova tabela služi za evidentiranje stavki koje su dio svake narudžbe. Sadrži informacije o konkretnim jelima i pićima koje je gost naručio, kao i količinu i cijenu za svaki artikl;

- **dbo.StavkaRacuna:** Tabela u kojoj su evidentirane sve pojedinačne stavke na računima, s detaljima o narudžbi i povezanim menijskim artiklima. Omogućava precizan uvid u strukturu računa;
- **dbo.Racuni:** Ova tabela sadrži ukupne račune koji su generisani za goste. Svaki račun povezan je sa narudžbom i sadrži iznos, datum i status plaćanja;
- **dbo.Placanje:** Tabela koja bilježi podatke o plaćanju, uključujući način plaćanja (gotovina, kartica), vrijeme transakcije i povezanost s računom.

Ova struktura omogućava jasno i efikasno upravljanje svim aspektima poslovanja restorana, od rezervacija i narudžbi do upravljanja korisnicima i menijem i drugim funkcionalnostima. Povezanost između tabela omogućava jednostavnu pretragu, izvještavanje i proširivanje sistema u budućnosti.



Slika 47. Prikaz strukture baze podataka

10. 2. API

Na slici 48. prikazana je struktura projekta nazvanog **Restoran**, koji predstavlja informacioni sistem za upravljanje restoranom. Projekt je organizovan prema troslojnoj arhitekturi (API, BLL i DAL), čime je omogućeno jasno razdvajanje logike prezentacije, poslovne logike i pristupa podacima. Ova arhitektura doprinosi boljoj modularnosti, lakšem održavanju i skalabilnosti sistema.

10. 2. 1. API sloj – prezentacijski i ulazni sloj aplikacije

U folderu **API** nalaze se **kontroleri** koji služe kao ulazna tačka za HTTP zahtjeve i upravljaju komunikacijom između frontend dijela sistema i poslovne logike. Svaki kontroler je specijalizovan za određeni entitet i odgovoran je za obradu zahtjeva kao što su kreiranje, izmjena, brisanje i dohvatanje podataka.

U okviru API sloja koristi se **Swagger** za testiranje API ruta, kao i **AutoMapper** (MappingProfile.cs) za mapiranje između DTO objekata i entiteta baze. Konfiguracije kao što su konekcijski stringovi i postavke za Stripe (za online plaćanja) definirane su u datoteci **appsettings.json**, dok se u **Program.cs** nalazi početna tačka aplikacije, gdje se konfigurišu servisi, autentifikacija i middleware komponente. Osim osnovnih slojeva, struktura sadrži i dodatne komponente:

- DTO
- Models
- appsettings.json
- Program.cs

10. 2. 2. BLL sloj (Business Logic Layer) – poslovna logika

Folder **BLL** sadrži **interfejse i njihove implementacije**, koje definišu poslovnu logiku sistema. Sloj je organizovan u dvije ključne cjeline:

- **Interfaces:** Definisani su interfejsi;
- **Services:** Konkretne implementacije ovih servisa koje sadrže pravila, validacije i funkcionalnosti kao što su kreiranje rezervacija, slanje e-mailova, izračunavanje računa, upravljanje korisnicima i sl.

Ovaj sloj ne komunicira direktno s bazom podataka, nego koristi DAL sloj, što omogućava bolju kontrolu toka podataka i olakšava testiranje i proširivanje sistema.

10. 2. 3. DAL sloj (Data Access Layer) – pristup bazi podataka

U **DAL** sloju smješten je folder **Data**, koji sadrži **ApplicationDbContext.cs** – klasu koja definiše konekciju s bazom podataka i sadrži sve DbSet-ove za entitete. Struktura je dodatno podijeljena na:

- **Interfaces:** Interfejsi za pristup podacima;
- **Repositories:** Konkretne klase koje implementiraju CRUD operacije nad bazom.

10. 2. 4. DTO – data transfer object (prenos podataka između slojeva)

DTO sloj sadrži klase koje služe za prenos podataka između backend slojeva i frontend korisničkog interfejsa, bez direktnog izlaganja baza-modela (Models). Ove klase predstavljaju olakšanu i sigurnu verziju podataka, često sadrže samo ono što je potrebno za prikaz, unos ili validaciju.

DTO modeli se često automatski mapiraju na entitete korištenjem **AutoMapper** biblioteke, čime se omogućava čista i jasna razmjena podataka između slojeva aplikacije, uz povećanu sigurnost i efikasnost.

10. 2. 5. Models – entiteti (reprezentacija tabela u bazi)

U ovom folderu nalaze se klase koje predstavljaju **entitete sistema**, odnosno svaka klasa odgovara jednoj tabeli u bazi podataka. Ovi modeli direktno opisuju strukturu podataka unutar sistema restorana i koriste se u okviru ORM-a (Entity Framework) za kreiranje, čitanje, ažuriranje i brisanje zapisa u bazi.

Ove klase se koriste unutar **DbContext** klase u DAL sloju za kreiranje odgovarajućih **DbSet-ova**.

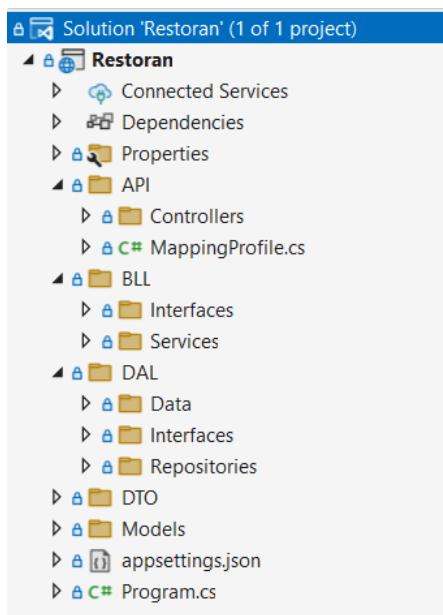
10. 2. 6. Program.cs

Glavni fajl koji definiše ulaznu tačku aplikacije. Ovdje se konfigurišu servisi, pokreće aplikacija i povezuju slojevi (npr. povezivanje API-ja s bazom i poslovnom logikom).

10. 2. 7. appsettings.json

Konfiguracijska datoteka u kojoj se nalaze parametri poput konekcije prema bazi podataka, postavki autentifikacije, API ključeva (npr. Stripe), i drugih važnih vrijednosti potrebnih za rad sistema.

Ovakva organizacija omogućava da sistem bude jasan, pregledan i spreman za eventualna buduća poboljšanja (bilo da se dodaju nove funkcionalnosti, nove rute API-ja ili čak integracija s drugim sistemima). Time se osigurava da je informacioni sistem restorana skalabilan, siguran i održiv.



Slika 48. Prikaz strukture projekta za API

10.3. Web aplikacija

Na slikama 49. i 50. prikazana je struktura frontend dijela informacionog sistema **noviRestoran**, koji je razvijen korištenjem Angular frameworka. Aplikacija je modularno organizovana, uz jasno razdvajanje komponenti, servisa i konfiguracijskih fajlova. Time je omogućena dobra preglednost, održavanje i proširivost aplikacije.

10.3.1. Korijenska struktura projekta

U korijenskom direktoriju projekta nalaze se različiti sistemske i konfiguracijske fajlovi koji omogućavaju pravilan rad Angular aplikacije:

- **angular.json**: Glavni konfiguracijski fajl za Angular CLI, definiše build opcije i strukturu projekta;
- **package.json** i **package-lock.json**: Sadrže informacije o korištenim paketima i njihovim verzijama;
- **server.ts**: Fajl koji omogućava server-side rendering aplikacije;

- **tsconfig.json, tsconfig.app.json, tsconfig.sprec.json:** Konfiguracijski fajlovi za TypeScript;
- **.editorconfig, .gitignore, README.md:** Pomoćni fajlovi za održavanje standarda u kodiranju i verzionisanju.

10.3.2. src folder – Glavni direktorij aplikacije

Unutar **src** foldera smješten je kompletan kod Angular aplikacije. Njegova struktura uključuje:

- **index.html:** Osnovna HTML stranica aplikacije, predstavlja ulaznu tačku aplikacije u browseru;
- **main.ts i main.server.ts:** Fajlovi za pokretanje aplikacije (klijentski i serverski dio);
- **styles.css:** Globalni fajl sa stilovima koji se primjenjuju na cijelu aplikaciju;
- **favicon.ico:** Ikonica aplikacije;
- **assets:** Folder koji sadrži statičke resurse kao što su slike.

10.3.3. app folder – Centralni dio aplikacije

Folder **app** čini osnovu cjelokupne Angular aplikacije. Struktura ovog foldera podijeljena je na funkcionalne cjeline:

- **features folder:** Sadrži sve vizuelne komponente aplikacije, koje su organizovane prema funkcionalnostima sistema:
 - **dodaj-umeni, uredi-meni, meni:** Komponente za kreiranje, izmjenu i prikaz jelovnika restorana;
 - **kreiraj-narudzbu, sve-narudzbe:** Upravljanje narudžbama korisnika;
 - **rezervacije, sve-rezervacije:** Komponente za rezervaciju i prikaz svih rezervacija stolova;
 - **prijava, registracija:** Komponente za autentifikaciju korisnika sistema;
 - **racun, placanje:** Funkcionalnosti za prikaz računa i online plaćanje;
 - **pocetna, kontakt, o-nama:** Informativne i promotivne stranice aplikacije.

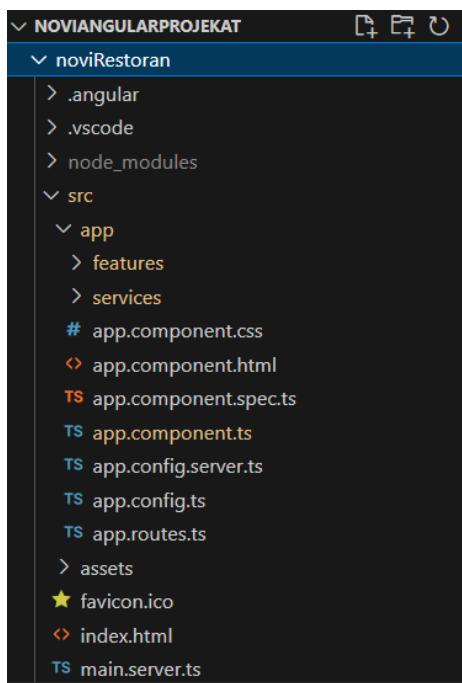
- **services folder:** Sadrži sve servise koji omogućavaju komunikaciju sa backend API-jem. Svaka funkcionalna cjelina ima svoj servis:
 - **auth.service.ts:** Upravljanje autentifikacijom korisnika (login, registracija);
 - **meni.ts, dodaj.ts, uredi.ts:** Za rad sa menijem (dodavanje, izmjena, prikaz);
 - **narudzba.ts, svenarudzbe.ts:** Servisi za kreiranje i pregled narudžbi;
 - **rezervacije.ts, sverezervacije.ts, stolovi.ts:** Upravljanje rezervacijama;
 - **placanje.ts, racun.ts:** Logika za online plaćanja i generisanje računa.
- **Ostali fajlovi u app folderu:**
 - **app.component.ts, app.component.html, app.component.css:** Glavne komponente aplikacije;
 - **app.routes.ts:** Definisanje ruta (navigacija kroz aplikaciju);
 - **app.config.ts, app.config.server.ts:** Konfiguracijski fajlovi specifični za klijentsku i serversku stranu.

10.3.4. Arhitekturna organizacija

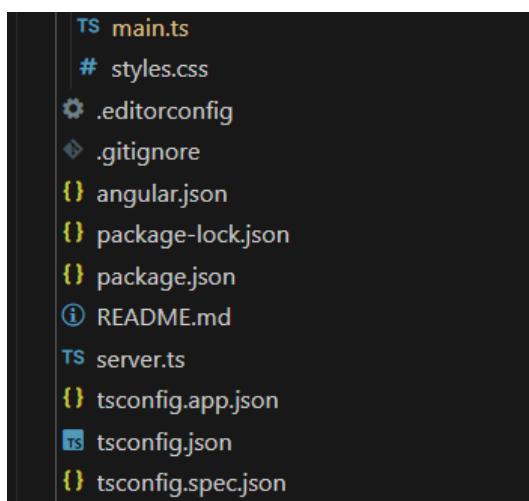
Frontend aplikacija slijedi modularni pristup, gdje su sve funkcionalnosti jasno podijeljene i organizovane unutar **features** i **services** foldera. Ovakva struktura omogućava:

- **Razdvajanje logike:** Poslovna logika nalazi se u servisima, dok su vizuelne komponente odvojene u **features**;
- **Jednostavno proširivanje sistema:** Dodavanje novih komponenti ne narušava postojeću strukturu;
- **Jednostavno testiranje i održavanje:** Komponente su izolovane i povezane kroz servise, što omogućava bolje testiranje aplikacije.

Ovakva frontend arhitektura je dizajnirana s ciljem da podrži funkcionalnosti informacionog sistema restorana na jasan način, omogućavajući korisnicima jednostavno naručivanje, rezervaciju i upravljanje podacima.



Slika 49. Prikaz strukture projekta web aplikacije ISR-a



Slika 50. Prikaz ostatka strukture projekta web aplikacije ISR-a

11.ZAKLJUČAK

ISR (Informacioni sistem restorana) predstavlja savremeno i efikasno rješenje za digitalizaciju poslovanja restorana, razvijen korištenjem modernih web tehnologija i u pogledu arhitekture podijeljen je na frontend i backend sloj. Kroz implementaciju web aplikacije za korisnike i administratore, sistem omogućava centralizovano upravljanje narudžbama, menijem, rezervacijama i korisničkim formama, čime se znatno unapređuje svakodnevni rad osoblja i kvalitet usluge prema gostima.

Administratori sistema imaju efikasan i jednostavan alat za upravljanje i manipulaciju podacima, čime se osigurava tačnost i poboljšava operativna efikasnost restorana. Omogućeno im je kreiranje, ažuriranje i brisanje određenih stavki, kao i upravljanje korisničkim formama, što doprinosi većoj organizovanosti i bržem donošenju odluka. Klijentima je značajno olakšan rad kroz automatizovane procese naručivanja, rezervacije i plaćanja, što omogućava nesmetan rad bez čekanja.

Ovaj informacioni sistem eliminiše potrebu za ručnim vođenjem narudžbi i rezervacija, čime se smanjuje mogućnost grešaka i povećava preciznost u radu. Digitalizacija poslovanja omogućava brži pristup informacijama, bolju organizaciju osoblja i transparentniju interakciju sa korisnicima. Takođe, pristup aplikaciji putem web interfejsa omogućava fleksibilnost u radu i pristupačnost sa različitih uređaja.

ISR opisan u ovom diplomskom radu je jasan primjer kako informacioni sistemi mogu unaprijediti poslovne procese u ugostiteljstvu, povećati efikasnost i podići nivo usluge. Implementacija ovog sistema je također omogućila bolje korisničko iskustvo, postavljajući visoke standarde za buduće slične sisteme u oblasti ugostiteljstva.

Cjelokupni praktični dio rada je moguće vidjeti na ovom linku:

<https://github.com/Berina1/InformacioniSistemRestoranaDR>

12.LITERATURA

- [1] Springer, *The Front End: Presentation, u Web Application Architecture and Development*, 1st ed., 2014. Dostupno na SpringerLink (PDF),
https://link.springer.com/chapter/10.1007/978-1-4302-6611-2_4 (pristupljeno 10.5.2025.)
- [2] Microsoft Learn, “What is ASP.NET?”,
<https://learn.microsoft.com/en-us/aspnet/overview/> (pristupljeno: 18.5.2025.)
- [3] Microsoft, “What is ASP.NET?”,
<https://dotnet.microsoft.com/en-us/apps/aspnet> (pristupljeno: 25.5.2025.)
- [4] OneClickITSolution, “Top 5 Benefits of ASP.NET Framework”,
<https://www.oneclickitsolution.com/top-benefits-of-asp-net-framework/>
(pristupljeno: 1.6.2025.)
- [5] GeeksforGeeks, “Why we should use ASP.NET?”,
<https://www.geeksforgeeks.org/why-we-should-use-asp-net/> (pristupljeno: 5.6.2025.)
- [6] Microsoft Learn, “Introduction to ASP.NET Core”,
<https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core>
(pristupljeno: 13.6.2025.)
- [7] Swagger - <https://swagger.io/tools/swagger-ui/> (pristupljeno: 15.6.2025.)
- [8] Microsoft Docs, “SQL Server Overview,”
<https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>
(pristupljeno: 20.6.2025.)
- [9] Valerio De Sanctis, *Full-Stack Web Development with ASP.NET Core 8 and Angular*, 6th ed., Packt Publishing, 2023.
Dostupno na GitHub-u: <https://github.com/PacktPublishing/ASP.NET-Core-8-and-Angular>
(pristupljeno: 20.6.2025.)
- [10] Trello, <https://trello.com> (pristupljeno: 21.6.2025.)

