**VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA INFORMACIONE TEHNOLOGIJE**



DIPLOMSKI RAD

**Razvoj web aplikacije za upravljanje poslovanjem restorana primenom Laravel okvira sa integracijom geolokacije korisnika**

Mentor: dr Aleksandar Simović, prof. strukovnih studija Datum predaje:

Student: Marko Đokić

**Beograd Decembar, 2022.**

SADRŽAJ:

[1. UVOD 4](#_Toc123499685)

[2. METODOLOGIJA RAZVOJA SOFTWERA 5](#_Toc123499686)

[2.1. MODEL VODOPADA 5](#_Toc123499687)

[2.2. V-MODEL 5](#_Toc123499688)

[2.3. INKREMENTALNI I ITERATIVNI RAZVOJ 6](#_Toc123499689)

[2.4. SPIRALNI MODEL 7](#_Toc123499690)

[2.5. AGILNE METODE 7](#_Toc123499691)

[2.6. SCRUM RAZVOJ 8](#_Toc123499692)

[2.7. LARMANOVA METODA 9](#_Toc123499693)

[3. PRIMENJENE TEHNOLOGIJE ZA RAZVOJ WEB APLIKACIJE 10](#_Toc123499694)

[3.1. PHP 10](#_Toc123499696)

[3.2. LARAVEL 10](#_Toc123499697)

[3.3. MARIADB SERVER 10](#_Toc123499698)

[3.4. JAVASCRIPT 11](#_Toc123499699)

[3.5. BOOTSTRAP 11](#_Toc123499700)

[3.6. LIVEWIRE 11](#_Toc123499701)

[3.7. MODEL-VIEW-CONTROLLER KONCEPT 11](#_Toc123499702)

[3.8. REPOSITORY DESIGN PATTERN 12](#_Toc123499703)

[4. SPECIFIKACIJA KORISNIČKIH ZAHTEVA 13](#_Toc123499704)

[4.1. VERBALNI OPIS MODELA 13](#_Toc123499706)

[4.2. SLUČAJEVI KORIŠĆENJA 14](#_Toc123499707)

[4.3. OPIS SLUČAJEVA KORIŠĆENJA 15](#_Toc123499709)

[5. FAZA ANALIZE 22](#_Toc123499710)

[5.1. SISTEMSKI DIJAGRAMI SEKVENCI ZA SLUČAJEVE KORIŠĆENJA 22](#_Toc123499711)

[5.2. DEFINISANJE UGOVORA O SISTEMSKIM OPERACIJAMA 25](#_Toc123499712)

[5.3. KONCEPTUALNI (DOMENSKI) MODEL 28](#_Toc123499713)

[5.4. RELACIONI MODEL 30](#_Toc123499714)

[6. FAZA PROJEKTOVANJA 31](#_Toc123499715)

[6.1. DIJAGRAMI SEKVENCI I KOLABORACIONI DIJAGRAMI ZA SISTEMSKE OPERACIJE 31](#_Toc123499716)

[6.2. PROJEKTOVANJE BAZE PODATAKA 34](#_Toc123499717)

[6.3. PROJEKTOVANJE KORISNIČKOG INTERFEJSA 39](#_Toc123499718)

[7. ZAKLJUČAK 43](#_Toc123499719)

[8. LITERATURA 44](#_Toc123499720)

[9. PRILOZI 46](#_Toc123499721)

[9.1. SPISAK SLIKA 52](#_Toc123499722)

[9.2 Spisak upotrebljenih akronima 54](#_Toc123499723)

**Rezime**

U završnom radu ,,Razvoj web aplikacije za upravljanje poslovanjem restorana primenom Laravel okvira sa integracijom geolokacije korisnika”, projektovana je I implementirana Web aplikacija za organizaciju I upravljanje ugostiteljskog objekta, kao I pružanje usluga naručivanja I dostave hrane krajnjim korisnicima. Namena Web aplikacije je da olakša i unapredi poslovanje restorana automatizovanjem pojedinih procesa, kao što su evidentiranje novokreiranih porudžbina, obračunavanje zarade i mnoge druge statistike značajne za uspešan i napredan rad. Takođe namena Web aplikacije je i da olakša krajnjem korisniku naručivanje i pregled samih artikala u svega par jednostavnih koraka. Web aplikacija poseduje nekoliko korisničkih rola, kojima su jasno definisane i raspodeljene funkcionalnosti sistema. Svaka korisnička rola je podjednako bitna i međusobno zavisna od drugih, tako da se može lako utvrditi na čijoj strani je došlo do eventualnih problema i grešaka u poslovanju.

Rad je implementiran u PHP programskom jeziku, upotrebom Laravel okvira. Za skladištenje i manipulaciju podacima zadužena je MYSQL relaciona baza podataka. Za sam dizajn odgovoran je CSS i bootstrap okvir, dok su obaveštenja i validacije podataka implementirane putem JavaScript-a.  
  
**Ključne reči:**  
  
Web aplikacija, porudžbina, korisnik, Laravel, PHP, MySQL.

# UVOD

U završnom radu ,,Razvoj web aplikacije za upravljanje poslovanjem restorana primenom Laravel okvira sa integracijom geolokacije korisnika”, obrađen je kompletan sistem rada jednog ugostiteljskog objekta, prolazeći pojedinačno kroz svaku rolu koja je podjednako bitna za ceo sistem. Funkcionalnosti su mnogobrojne i svaka ništa manje bitna od neke druge, s toga je ovaj rad sažet što je više moguće. Sistem poseduje šest korisničkih rola, a to su: fizičko lice, pravno lice, konobar, dostavljač, šef restorana i menadžer.

Značaj završnog rada je da olakša poslovanje zaposlenima ugostiteljskog objekta, kao i da unapredi i poboljša poslovanje u značajnoj meri. Sama mogućnosti pregleda artikala iz asortimana olakšava korisniku da na brži i jednostavniji način odluči šta bi želeo da naruči na svoju kućnu adresu ili u samom restoranu. Pored toga su sve ostale funkcionalnosti koje korisniku pružaju mogućnost da na brz i lak način dođe do željenog obroka.

Projektovanje samog rada obavljeno je upotrebom Larmanove metode razvoja softvera. Poglavlja (četvrto, peto i šesto) u nastavku predstavljaju faze Larmanove metode.

U drugom poglavlju opisane su neke od značajnijih metodologija razvoja softvera, među kojima je i Larmanova metoda.

Trećim poglavljem predstavljene su tehnologije korišćene prilikom izrade završnog rada.

Četvrto poglavlje je opis specifikacija korisničkih zahteva kroz detaljan opis slučajeva korišćenja, sa strane svih korisničkih rola u sistemu.

U petom poglavlju opisana je faza analize upotrebom dijagrama sekvenci, gde su prikazane relacije među korisnicima i sistema. Definisani su ugovori o sistemskim operacijama, prikaz konceptualnog modela i relacionog modela koji će predstavljati osnovu za projektovanje baze podataka.

Kroz kolaboracione i dijagrame sekvenci, u šestom poglavlju opisana je faza projektovanja za sistemske operacije. U daljem nastavku prikazana je baza podataka korišćena u projektovanju. Na kraju poglavlja dat je prikaz korisničkog interfejsa aplikacije kako bi se korisnici na lakši način upoznali sa postojećim funkcionalnostima.

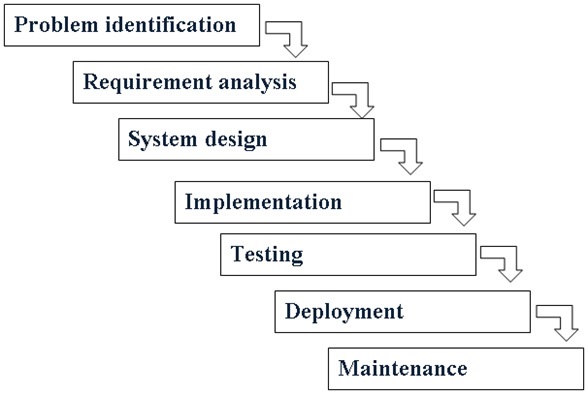
Zaključak samog rada kao i razmatranje mogućih unapređenja obuhvaćeno je u poslednjem poglavlju.

# METODOLOGIJA RAZVOJA SOFTWERA

Metodologija razvoja softvera se odnosi na strukturirane procese uključene u rad na projektu. To je spoj filozofije dizajna i pragmatičnog realizma koji se postiže od ranih dana računarstva. Cilj je da se obezbedi sistematski pristup razvoju softvera. Tokom godina uvedene su različite metodologije razvoja softvera kako bi se iskoristile dostupne tehnologije i resursi. Metodologija razvoja softvera obezbeđuje platformu za programere da efikasnije rade zajedno kao tim. Ona formalizuje komunikaciju i određuje kako se informacije dele unutar tima [1].

## MODEL VODOPADA

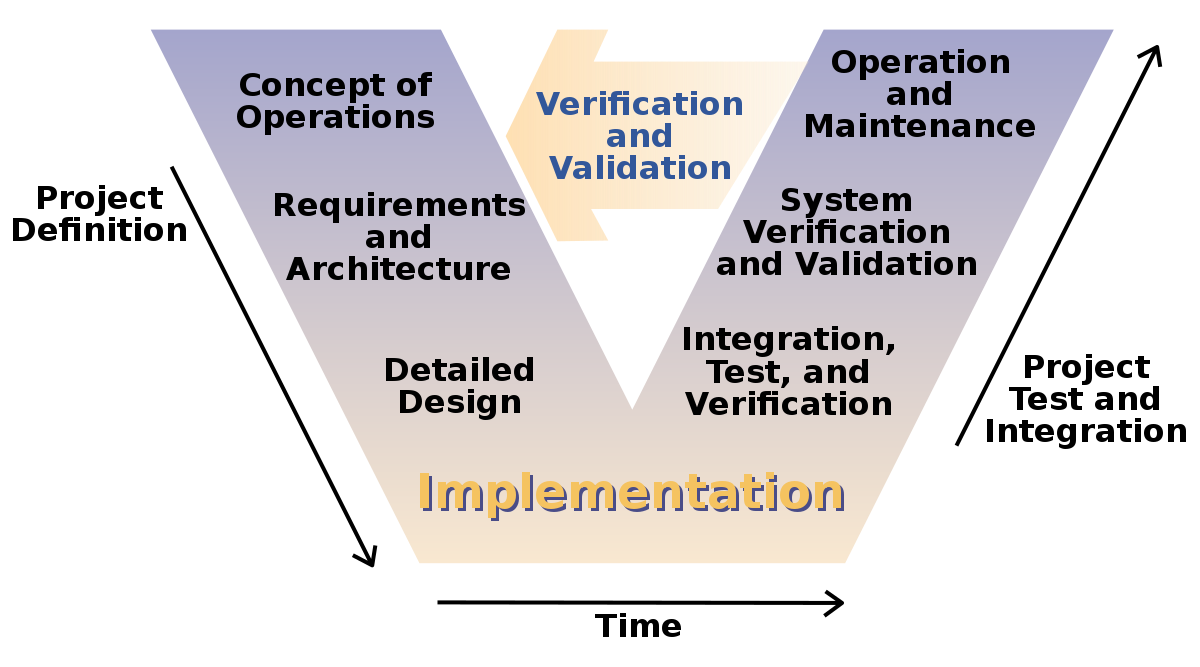
Model vodopada je linearni, sekvencijalni pristup životnom ciklusu razvoja softvera koji je popularan u softverskom inženjerstvu i razvoju proizvoda. Model vodopada naglašava lično napredovanje koraka. Slično u kom pravcu voda teče preko ivice litice, određene krajnje tačke i ciljevi su postavljeni za svaku fazu razvoja i ne mogu se ponovo razmatrati nakon završetka. Termin je prvi put uveden u radu koji je objavio dr. Winston W. Royce i nastavlja da se koristi u primenama industrijskog dizajna [2].  
Na slici 1. prikazan je model vodopada.



Slika 1. Model vodopada [3]

## V-MODEL

V Model je model validacije i verifikacije. Baš kao i model vodopada, životni ciklus u V modelu je sekvencijalni put izvršenja procesa. Svaka faza mora biti završena pre nego što počne sledeća faza. Dotični model je jedan od mnogih modela razvoja softvera. Testiranje proizvoda je paralelno sa odgovarajućom fazom razvoja u V modelu [4].  
Na slici 2. prikazan je V model.

Slika 2. V-Model [5]

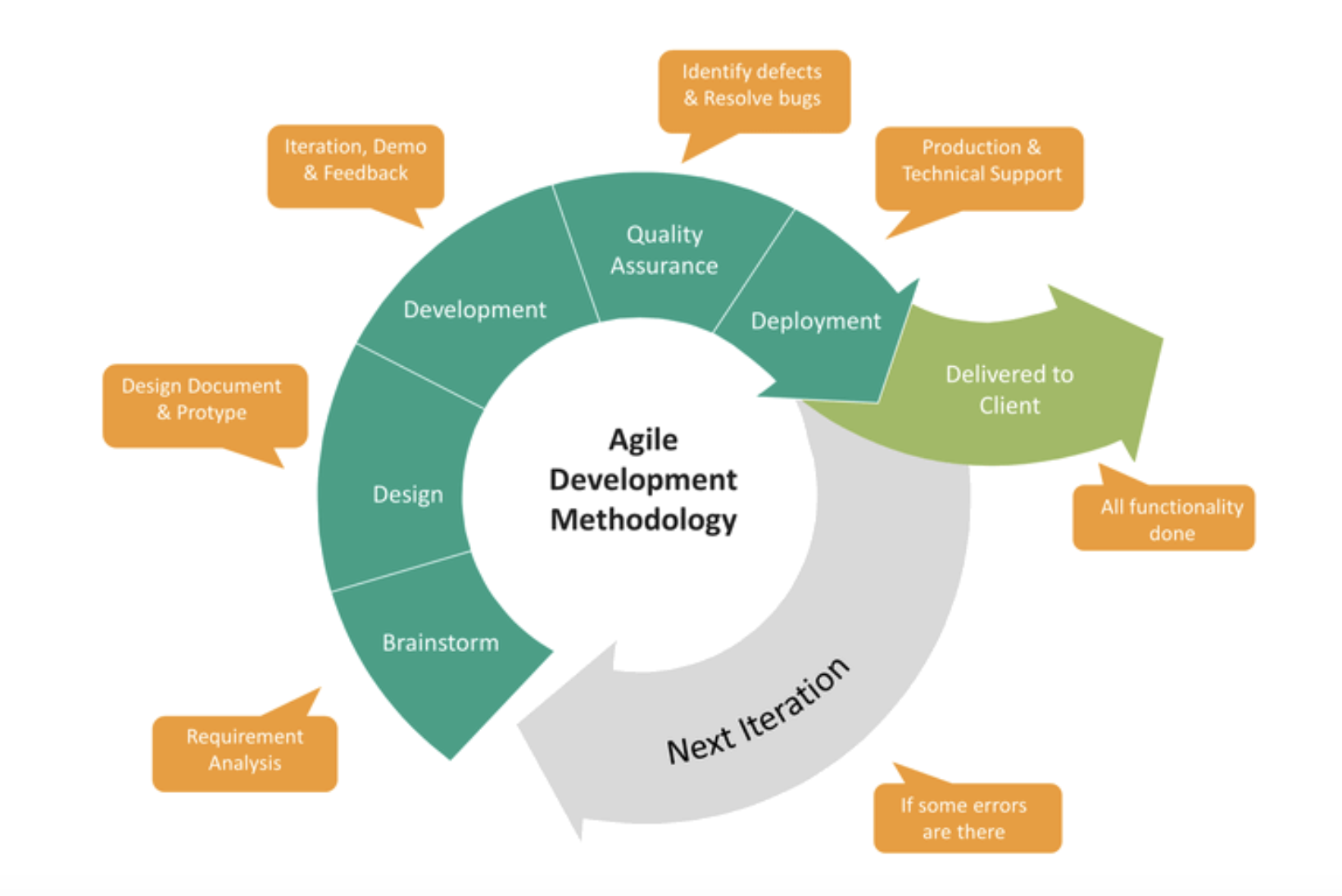
## INKREMENTALNI I ITERATIVNI RAZVOJ

Inkrementalni i iterativni razvoj je proces koji kombinuje iterativni metod projektovanja sa inkrementalnim modelom izgradnje. Koriste ga programeri za pomoć u upravljanju projektima.

Možemo ga podeliti na dva dela:

- Inkrementalni: Razbija proces razvoja softvera na male delove kojima se može upravljati, poznate kao inkrementi. Svaki inkrement se nadograđuje na prethodnu verziju tako da se poboljšanja vrše korak po korak.

- Iterativni: Iterativni model znači da se aktivnost razvoja softvera sistemski ponavljaju u ciklusima poznatim kao iteracije. Nova verzija softvera se proizvodi nakon svake iteracije sve dok se ne postigne optimalni proizvod [6].  
Na slici 3 prikazan je iterativni i inkrementalni razvoj softvera.

  
Slika 3. Iterativni i inkrementalni razvoj [7]

## SPIRALNI MODEL

Spiralni model je sličan inkrementalnom razvoju softvera, sa većim naglaskom na analizu rizika. Spiralni model ima četiri faze:  
  
- Planiranje

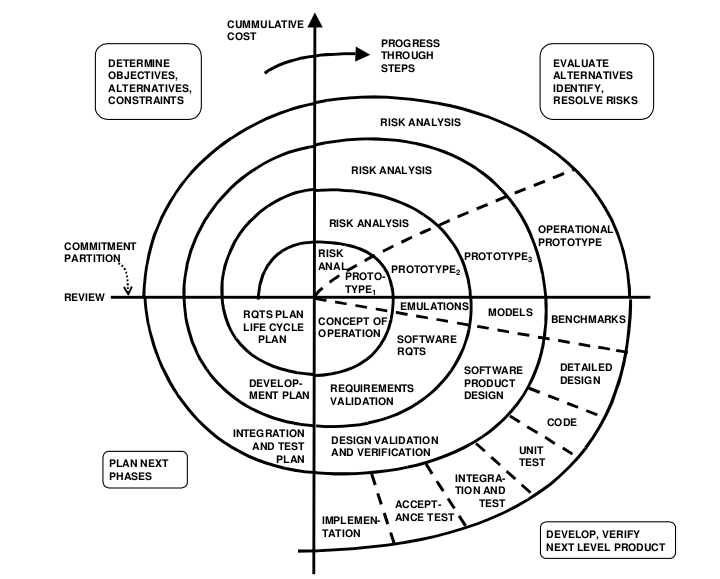
- Projektovanje

- Konstrukcija

- Evaluacija

Softverski projekat više puta prolazi kroz ove faze u iteracijama, koje u ovom modelu nazivamo spiralama [8].

Na slici 4 je grafički prikazan spiralni model razvoja softvera.

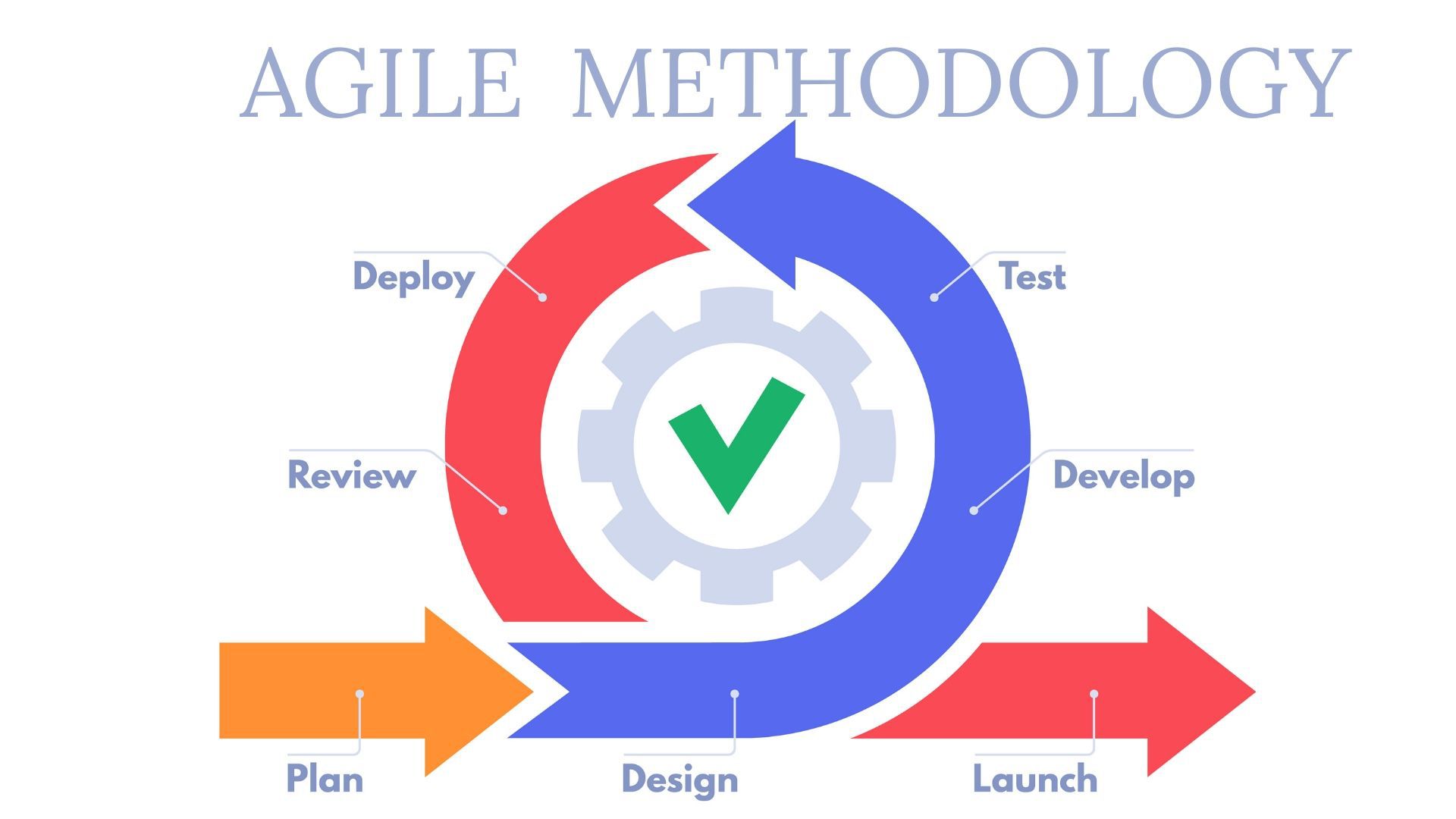


Slika 4. Spiralni model [9]

## AGILNE METODE

Agilna metodologija je praksa koja promoviše kontinuirano ponavljanje razvoja i testiranja tokom životnog ciklusa razvoja softvera. U agilnom modelu u testiranju softvera, aktivnost razvoja i testiranja su istovremene, za razliku od modela vodopada [10].

Na slici 5 grafički je ilustrovana agilna metoda softverskog razvoja.

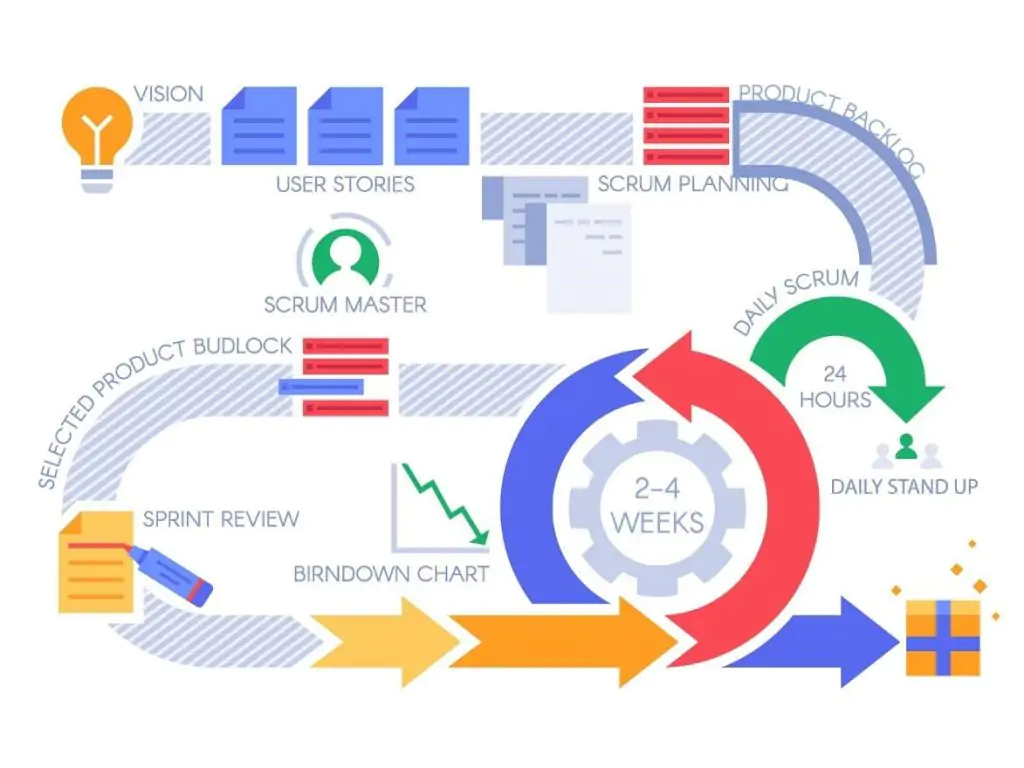


Slika 5. Agilna metoda razvoja softvera [11]

## SCRUM RAZVOJ

Skram je agilna razvojna metodologija koja se koristi u razvoju softvera zasnovana na iterativnim i inkrementalnim procesima. Skram je prilagodljiv, brz, fleksibilan i efikasan agilni okvir koji je dizajniran da pruži vrednost tokom celog razvoja projekta. Primarni cilj skrama je da zadovolji potrebe korisnika kroz okruženje transparentnosti u komunikaciji, kolektivne odgovornosti i kontinualnog napretka. Razvoj počinje od opšte ideje šta treba da se izgradi, razrade liste karakteristika poređanih po prioritetu koje vlasnik proizvoda želi da dobije [12].

Skram razvoj je slikovito prikazan na slici 6.



Slika 6. Skram razvoj [13]

## LARMANOVA METODA

Larmanova metoda je metodologija softverskog inženjerstva za razvoj softvera koja omogućava razvoj softvera koji je modularan, skalabilan, upotrebljiv, pouzdan i lako se održava. Larmanova pojednostavljena metoda obuhvata sve faze životnog ciklusa softvera [14]:

- Prikupljanje korisničkih zahteva

Kupac, odnosno naručilac softverskog proizvoda pre svega treba pravilno da definiše svoje zahteve, a analitičar zahteva da ih prikupi na isti način na koji namerava da ih prenese.

Cesto nije moguće da analitičar zahteva ispravno postavi korisničke zahteve na početku, zbog čega je dobra praksa uključiti sve zainteresovane strane koje bi mogle uticati na krajnji proizvod.

Prikupljanje korisničkih zahteva moguće je na brojne načine kao što su: korisničke priče, slučajevi korišćenja, dokument na govornom jeziku kupca, dijagrami toka i mnogi drugi. Korisničke priče su jedna od popularnijih metoda za prikupljanje zahteva, jer su u osnovi skup korisničkih unosa na njihovom govornom jeziku.

- Analiza zahteva

U ovoj fazi različite zainteresovane strane održavaju sesije ideja. Analiziranjem prikupljenih zahteva traže izvodljivost da ih primene. Međusobnim razgovorom se svaka nejasnoća rešava

Ovaj korak je jako bitan iz razloga što kupac može tražiti zahteve koje je nemoguće primeniti . [15]

- Projektovanje  
Projektovanje softvera je proces transformisanja korisničkih zahteva u neki odgovarajući oblik, koji pomaže softverskim inženjerima u programiranju i implementaciji softvera. Tokom faze projektovanja softvera kreira se projektni dokument, na osnovu zahteva naručioca onako kako je dokumentovano u SRS (eng. software requirements specification) dokumentu. Cilj ove faze je da se SRS dokument transformiše u projektni dokument. [16]

- Implementacija

Faza implementacije razvoja softvera je proces razvoja softverskog proizvoda, koji treba da se isporuči naručiocu. Ponekad ovo uključuje odvojene aktivnosti projektovanja i programiranja. Međutim ako se koristi agilan pristup razvoja, projektovanje i implementacija se prepliću bez formalnih projektnih dokumenata koji se proizvode tokom procesa. [17]

- Testiranje

Testiranje softvera je metod provere da li softverski proizvod odgovara očekivanim zahtevima i provera da li softverski proizvod poseduje defekte. Faza testiranja uključuje izvršavanje softverskih komponenata upotrebom ručnih ili automatizovanih alata za procenu. Svrha testiranja softvera je da se identifikuju greške, praznine ili nedostajući zahtevi u odnosu na stvarne zahteve. [18]

# PRIMENJENE TEHNOLOGIJE ZA RAZVOJ WEB APLIKACIJE

# U ovom poglavlju obradićemo šablone i tehnologije korišćene prilikom izrade web aplikacije

* 1. PHP PHP (Eng. Hypertext Preprocessor) je skriptni jezik otvorenog kôda na strani servera koji mnogi programeri koriste za Web razvoj. To je takođe jezik opšte namene koji može da se koristi za izradu mnogo projekata, uključujući grafičke korisničke interfejse (GUI) [19].  
       
     Skraćenica PHP je prvobitno značila Personal Homepage. Ali to je sada akronim za HyperText Preprocessor (Rekurzivno je u smislu da je prva reč sama po sebi skraćenica, tako da puno značenje ne prati skraćenicu).  
       
     Prva verzija PHP-a je pokrenuta pre 26 godina. Sada je na verziji 8, objavljenoj u novembru 2020. godine, ali verzija 7 i dalje ostaje najkorišćenija.  
       
     PHP radi na Zend engine-u, koji je najpopularnija implementacija. Postoje i neke druge implementacije kao što su parrot, HPMV (Hip Hop Virtual Machine), i Hip Hop kreiran od strane Facebook-a.

PHP se uglavnom koristi za pravljenje Web servera. Pokreće se u pretraživaču i takođe je sposoban da radi u komandnoj liniji. Dakle, ako ne želite da prikažete izlaz kôda u pretraživaču, možete ga prikazati u komandnoj liniji, odnosno terminalu.

* 1. LARAVELLaravel je višeplatformski PHP okvir za kreiranje Web aplikacija. To je platforma zasnovana na serveru koji upravlja podacima, koristeći obrazac dizajna Model-View-Controller (MVC), dele pozadinsku arhitekturu aplikacije na logičke delove [20].  
       
     PHP okvir je skup biblioteka kôda sa unapred programiranim funkcijama kao što su alati za autentifikaciju, alati za rutiranje, ili HTML šabloni koji omogućavaju korisnicima da brže prave aplikacije. Laravelov najbolji PHP okvir ubrzava proces Web razvoja i poboljšava kvalitet finalnog proizvoda.

Laravel Web aplikacije su veoma skalabilne i imaju dobro struktuirane baze kôda. Laravelov modularni sistem pakovanja i snažno upravljanje zavisnošću omogućavaju programerima da brzo prošire funkcionalnost svojih aplikacija.

* 1. MARIADB SERVER MariaDB je sistem za upravljanje relacionim bazama podataka otvorenog kôda (DBMS) koji je kompatibilna zamena za široko korišćenu MySQL (Eng. My Structured Query Language) tehnologiju baze podataka. Napravljen je od strane programera koji su igrali ključne uloge u izgradnji originalne baze podataka. Osmislili su MariaDB 2009. godine kao odgovor na kupovinu MySQL-a od strane Oracle-a [21].  
       
     MariaDB je baziran na SQL-u i podržava obradu podataka u ACID (Eng. atomicity, consistency, isolation, durability) stilu sa zagarantovanom autonomnošću, doslednošću, izolacijom i izdržljivošću za transakcije. Između ostalih karakteristika, baza podataka takođe podržava JSON (Eng. JavaScript Object Notation) API-je (Eng. application programming interface), paralelnu replikaciju podataka i više mehanizama za skladištenje, uključujući i InnoDB, MyRocks, Spider, Aria, TokuDB, Cassandra and MariaDB ColumnStore.

Veliki deo rada na razvoju baze podataka otvorenog kôda fokusiran je na postizanju pariteta karakteristika između MariaDB i MySQL. MariaDB corp., pokretačka snaga MariaDB-a, kaže da je za sve praktične svrhe baza podataka binarno kompatibilna sa MySql-om. Kao rezultat toga većina korisnika može da prelazi između dve tehnologije jednostavnim definisanjem MySQL-a i instaliranjem MariaDB-a na njegovo mesto.

# **JAVASCRIPT**

JavaScript je skriptni ili programerski jezik koji vam omogućava da implementirate složene funkcije na Web serverima. Svaki put kada Web stranica radi nešto vise od prikazivanja statičkih informacija koje možemo da pogledamo, prikazujući pravovremeno ažuriran sadržaj, interaktivne mape, 2D/3D (Eng. two-dimensional/three-dimensional) grafičke animacije itd. U većini slučajeva je uključen javascript. To je treći sloj torte slojeva standardnih Web tehnologija [22].

# **BOOTSTRAP**

Bootstrap je besplatan frontend razvojni okvir otvorenog kôda za kreiranje Web stranica i Web aplikacija. Dizajniran da omogući responzivan razvoj Web sajtova namenjenih mobilnim uređajima. Bootstrap obezbeđuje kolekciju sintakse za dizajn šablona [23].  
  
Kao okvir bootstrap uključuje osnove za responzivan Web razvoj, tako da programeri samo treba da ubace kôd u unapred definisani sistem mreže. Bootstrap okvir je izgrađen na jeziku za označavanje hiperteksta HTML (Eng. HyperText Markup Language), kaskadnim stilovima CSS (Eng. Cascading Style Sheets) i javascript-u. Web programeri koji koriste bootstrap mogu da prave Web stranice mnogo brže bez trošenja vremena na brige o osnovnim komandama i funkcijama.

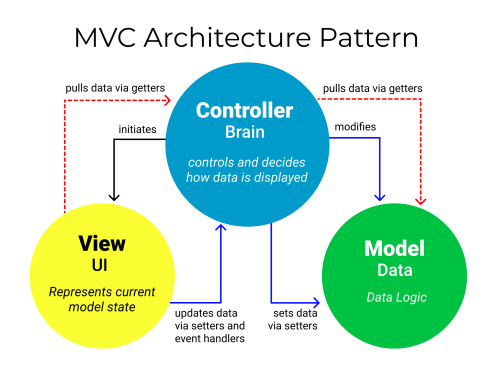
# **LIVEWIRE**

Livewire je kompletan okvir za Laravel koji čini izgradnju dinamičkih interfejsa jednostavnim, bez napuštanja udobnosti Laravel-a [24].   
  
Livewire prikazuje početni izlaz komponente sa stranicom. Na ovaj način je prilagođeno SEO-u (Eng. search engine optimization).  
Kada dođe do interakcije, livewire šalje AJAX (Eng. Asynchronous JavaScript And XML) zahtev serveru sa ažuriranim podacima.  
  
Server ponovo prikazuje komponentu i odgovara sa novim HTML-om.  
Livewire zatim inteligentno mutira DOM (Eng. Document Object Model) u skladu sa stvarima koje su se promenile.

# **MODEL-VIEW-CONTROLLER KONCEPT**

Model predstavlja pozadinu koja sadrži svu logiku podataka.   
View odnosno pogled je frontend odnosno grafički korisnički interfejs GUI (Eng. Graphical User Interface).  
Kontroler je mozak aplikacije koji kontroliše kako se podaci prikazuju [25].  
  
Koncept MVC-ja (Eng. Model-View-Controller) prvi je uveo Trygve Reenskaug, koji ga je predložio kao način za razvoj GUI aplikacija za desktop računare.  
  
Danas se MVC obrazac koristi za moderne Web aplikacije jer omogućava da aplikacija bude skalabilna, održiva i lako proširiva.

Na slici 7. slikovito je prikazan MVC koncept.

  
Slika 7. Model-View-Controller [26]

## REPOSITORY DESIGN PATTERN

Repozitorijumi su klase ili komponente koje obuhvataju logiku potrebnu za pristup izvorima podataka. Oni centralizuju zajedničku funkcionalnost pristupa podacima, obezbeđuju bolju mogućnost održavanja razdvajajući infrastrukturu ili tehnologiju koja se koristi za pristup bazama podataka od sloja modela domena. Omogućava nam da se fokusiramo na logiku postojanosti podataka, a ne na mrežu za pristup podacima [27].  
  
U osnovi, repozitorijum vam omogućava da popunite podatke u memoriji koji dolaze iz baze podataka u obliku entiteta domena. Jednom kada se entiteti nađu u memoriji, mogu se promeniti i zatim nastaviti nazad u bazu podataka kroz transakcije.

# SPECIFIKACIJA KORISNIČKIH ZAHTEVA

# U ovom poglavlju prikazan je verbalni opis modela na osnovu koga se dalje utvrđuju slučajevi korišćenja svih aktera u sistemu, kako bi bile opisale sve njihove funkcionalnosti u skladu sa zahtevima korisnika. U nastavku prikazan je detaljni opis svakog slučaja korišćenja pojedinačno.

# **VERBALNI OPIS MODELA**

Potrebno je projektovati I implementirati web aplikaciju ugostiteljske usluge i dostavu hrane restorana. Sistem će imati šest korisnika, a to su: fizičko/pravno lice, sef restorana, konobar restorana menadžer i dostavljač.

Fizičko i pravno lice imaju identične funkcionalnosti, ali sa jednom bitnom razlikom, pravna lica mogu naručiti i do 150 artikala u jednoj porudžbini, dok je za fizičko lice taj broj ograničen na 50. Pored te razlike, pravna lica među svoje podatke mogu navesti i ime zastupnika, odnosno firme u kojoj rade. Osim tih razlika, ostale funkcionalnosti su identične, kao što su pregled artikala, dodavanje artikala u korpu, brisanje dodatog artikla, promena količine artikala u korpi, kreiranje porudžbine, otkazivanje i lociranje iste. Pored toga moguća je i izmena ličnih podataka, kao i korisničke šifre.

Sa druge strane, korisničke role zadužene za poslovanje restorana imaju znatno drugačije funkcionalnosti.

Konobar ima ograničene funkcionalnosti na prihvatanje novokreirane porudžbine, čime se dobija evidencija da je sama porudžbina u procesu pripreme. Takođe pored prihvatanja konobar takođe može i otkazati pripremu, ukoliko je to iz bilo kog razloga potrebno, čime se stanje porudžbine ponovo vraća u početno, odnosno stanje čekanja. Pored toga, konobar ima mogućnost i ručnog kreiranja porudžbine, van trenutne liste artikala, što može u budućnosti i sugerisati na eventualno proširivanje asortimana.

Dostavljač onog trenutka kada konobar prihvati porudžbinu dobija evidenciju da će uskoro biti spremna ista, nakon čega dostavljač može prihvatiti istu. Kad se porudžbina prihvati sa dostavljačeve strane, korisnik dobija evidenciju da je njegova porudžbina u procesu isporuke, nakon čega se dobija mogućnost lociranja iste. Proces lociranja je usko povezan sa komunikacijom između ove dve role, odnosno dostavljača i korisnika. Korisnik nakon što dostavljač prihvati porudžbinu, dobija mogućnost za slanjem zahteva za trenutnu lokaciju dostavljača. Dostavljač nakon stizanja zahteva u dva klika šalje svoju trenutnu lokaciju. Ovaj proces je sveden na minimum, međutim i dalje je moguće automatizovati ga, što će biti urađeno kroz neke novije nadogradnje.

Menadžer restorana ima uvid u kompletnu statistiku, kao što su recimo prihodi, najčešće poručivani artikli, broj otkazanih porudžbina i lokacije korisnika, odnosno lokacije za isporuku. Ovom korisničkom rolom, pažljivom analizom svih podataka se može poboljšati poslovanje restorana u znatnoj meri.

Šef, kao najviša rola u ovom sistemu je taj koji ima mogućnost dodavanja i smanjivanja broja zaposlenih, odnosno zapošljavanja ili davanja/prihvatanja otkaza. Šef restorana ima uvid u sve pojedinačne funkcionalnosti ostalih korisničkih rola i na osnovu uvida i same procene može doneti odluku da li je nekome potrebno dati otkaz. Takođe kao što je navedeno, šef ima mogućnost i proširivanja broja radnika, odnosno prilikom zapošljavanja, evidentiranje novozaposlenog u sistemu, kreiranjem njegovog korisničkog naloga. Šef je taj koji jedini ima mogućnost da registruje svoje zaposlene, dok oni mogu samo da se prijave na svoj korisnički nalog.

U nekim daljim nadogradnjama moguća je dodatna automatizacija procesa tako da sistem sam na osnovu obračuna daje šefu statistiku koji radnici su podobni a koji nepodobni za naše poslovanje. U daljem nastavku rada obrađena je pojedinačno svaka korisnička rola i njene funkcionalnosti.   
Sistem poseduje još jednu korisničku rolu zaduženu za proširivanje ili smanjivanje našeg asortimana artikala, ali zbog ograničenog opsega završnog rada nju nećemo pismeno obrađivati.

# **SLUČAJEVI KORIŠĆENJA**

Definišu interakciju korisnika sa sistemom.

Često se povlače imenice da bi identifikovali koncepte u domenu problema.

Slučajevi korišćenja prikazani su Use Case dijagramom, na slici 8.

# Na osnovu verbalnog modela uočeni su sledeći slučajevi korišćenja za fizičko lice:

1.Kreiranje korisničkog naloga

2.Prijavljivanje na nalog

3.Brisanje korisničkog naloga

4.Menjanje šifre korisničkog naloga

5.Dodavanje artikla u korpu (Maksimalna količina 50 artikala)

6.Brisanje artikla iz korpe

7.Menjanje količine artikla u korpi

8.Kreiranje porudžbine

9.Otkazivanje porudžbine

10.Lociranje porudžbine

Na osnovu verbalnog modela uočeni su sledeći slučajevi korišćenja za pravno lice:

1.Kreiranje korisničkog naloga

2.Prijavljivanje na nalog

3.Brisanje korisničkog naloga

4.Menjanje šifre korisničkog naloga

5.Dodavanje artikla u korpu (Maksimalna količina 150 artikala)

6.Brisanje artikla iz korpe

7.Menjanje količine artikla u korpi

8.Kreiranje porudžbine

9.Otkazivanje porudžbine

10.Lociranje porudžbine

Na osnovu verbalnog modela uočeni su sledeći slučajevi korišćenja za sefa restorana:

1.Prijavljivanje na korisnički nalog

2.Menjanje šifre korisničkog naloga

3. Brisanje zaposlenih

4. Dodavanje zaposlenih

Na osnovu verbalnog modela uočeni su sledeći slučajevi korišćenja za konobara:

1.Prijavljivanje na korisnički nalog

2.Prihvatanje porudžbine

3.Otkazivanje porudžbine

4.Rucno unošenje porudžbine

Na osnovu verbalnog modela uočeni su sledeći slučajevi korišćenja za menadžer:

1.Prijavljivanje na korisnički nalog

2.Menjanje šifre korisničkog naloga

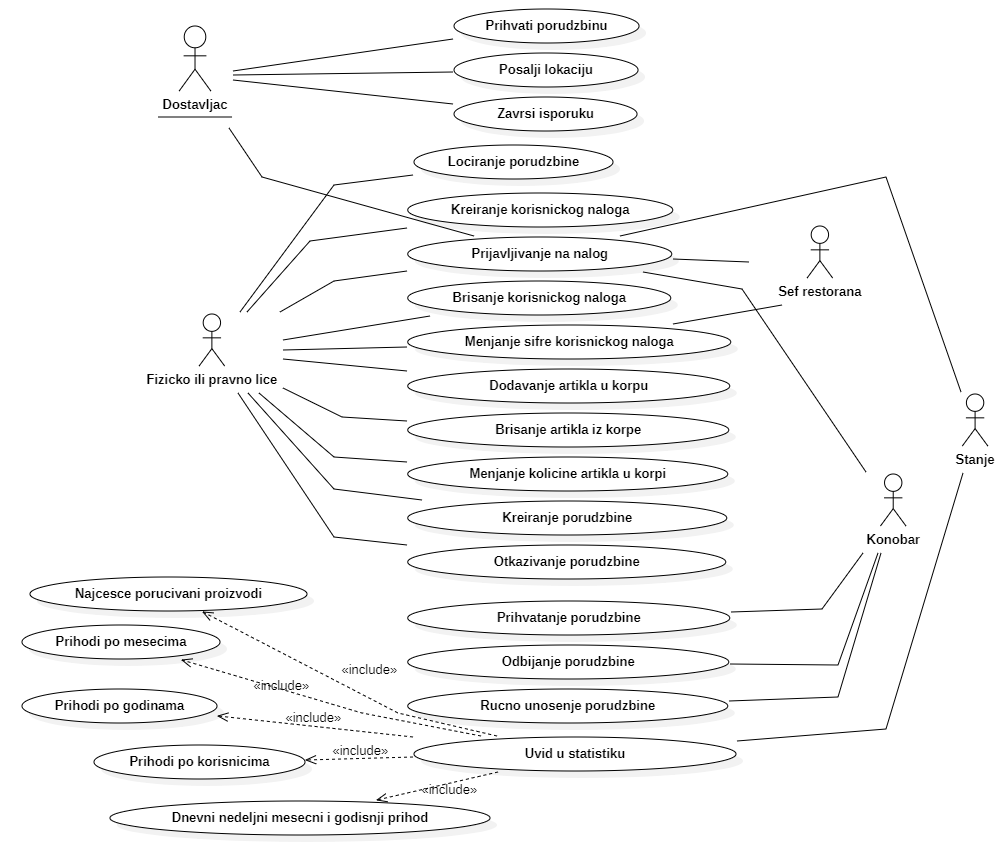
3.Uvid u statistiku

Na osnovu verbalnog modela uočeni su sledeći slučajevi korišćenja za dostavljača:

1.Prihvati porudžbinu

2.Posalji lokaciju

3.Zavrsi isporuku

  
Slika 8. Use Case diagram

## OPIS SLUČAJEVA KORIŠĆENJA

**SK1: Kreiranje korisničkog naloga**

Naziv: Kreiranje korisničkog naloga

Aktori: Fizičko lice, pravno lice

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma za logovanje ili kreiranje novog naloga.

Osnovni scenario:

1.Korisnik bira opciju za kreiranje naloga (APUSO)

2.Sistem pokreće proces za kreiranje naloga (SO)

3.Sistem prikazuje formu za kreiranje naloga (IA)

4.Korisnik unosi tražene podatke kako bi kreirao nalog (APUSO)

5.Korisnik poziva sistem kako bi izvršio kreiranje naloga sa unetim podacima (APSO)

6.Sistem proverava postojanje korisničkog imena (SO)

7.Sistem kreira korisnički nalog sa zadatim podacima (SO)

8.Sistem prikazuje poruku o uspešnosti kreiranja naloga (IA)

Alternativni scenariji:

5.1. Sistem pronalazi korisnički nalog sa već unetim korisničkim imenom, prikazuje poruku korisniku I prekida izvršavanje scenarija (IA)

7.1. Sistem ne može da kreira nalog sa zadatim podacima I prikazuje poruku korisniku (IA)

**SK2: Prijavljivanje na nalog**

Naziv: Prijavljivanje na nalog

Aktori: Fizičko lice, pravno lice, sef restorana, konobar, menadžer

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma za logovanje ili kreiranje novog naloga.

Osnovni scenario:

1.Korisnik bira opciju za prijavljivanje na nalog (APUSO)

2.Korisnik unosi tražene podatke(APUSO)

3.Korisnik poziva sistem da proveri postoji li nalog sa unetim podacima (APSO)

4.Sistem prijavljuje korisnika na njegov korisnički nalog (SO)

5.Sistem pokazuje poruku o uspešnosti logovanja (IA)

Alternativni scenariji:

4.1 Sistem ne pronalazi nalog sa unetim podacima I obustavlja proces prijavljivanja(IA)

**SK3: Brisanje korisničkog naloga**

Naziv: Brisanje naloga

Aktori: Fizičko lice, pravno lice, sef restorana

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa informacijama korisničkog naloga.

Osnovni scenario:

1.Korisnik bira opciju za brisanje naloga (APUSO)

2.Korisnik poziva sistem da izvrši operaciju brisanja (APSO)

3.Sistem izvršava operaciju brisanja (SO)

4.Sistem obaveštava korisnika da je brisanje uspešno izvršeno (IA)

Alternativni scenariji:

3.1 Došlo je do prekida internet veze (IA)

**SK4: Menjanje šifre korisničkog naloga**

Naziv: Menjanje šifre naloga

Aktori: Fizičko lice, pravno lice, sef restorana

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa informacijama korisničkog naloga.

Osnovni scenario:

1.Korisnik bira opciju za menjanje šifre naloga (APUSO)

2.Korisnik unosi postojeću šifru I novu šifru (APUSO)

3.Korisnik poziva sistem da izvrši promenu šifre (APSO)

4.Sistem proverava identičnost unete I već postojeće šifre (SO)

5.Sistem izvršava operaciju promene šifre (SO)

6.Sistem obaveštava korisnika da je promena šifre uspešno izvršena (IA)

Alternativni scenariji:

4.1 Šifre se ne poklapaju (IA)

**SK5: Dodavanje artikla u korpu**

Naziv: Dodavanje artikla u korpu

Aktori: Fizičko lice, pravno lice

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa postojećim artiklima

Osnovni scenario:

1.Korisnik bira opciju za prikaz artikala (APUSO)

2.Korisnik bira željeni artikal I količinu (APUSO)

3.Korisnik poziva sistem da smesti artikal u korpu (APSO)

4.Sistem proverava da li je korisnik prijavljen (SO)

5.Sistem izvršava operaciju smeštanja artikla u korpu (SO)

6.Sistem obaveštava korisnika da je artikal dodat u korpu (IA)

Alternativni scenariji:

4.1 Korisnik nije prijavljen (IA)

**SK6: Brisanje artikla iz korpe**

Naziv: Brisanje artikla iz korpe

Aktori: Fizičko lice, pravno lice

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa postojećim artiklima u korpi

Osnovni scenario:

1.Korisnik bira opciju za prikaz korpe (APUSO)

2.Korisnik bira željeni artikal za brisanje (APUSO)

3.Korisnik poziva sistem da izbriše artikal iz korpe (APSO)

4.Sistem izvršava operaciju brisanja artikal iz korpe (SO)

5.Sistem obaveštava korisnika da je artikal izbrisan iz korpe (IA)

Alternativni scenariji:

4.1 Korisnik nije prijavljen (IA)

**SK7: Menjanje količine artikla u korpi**

Naziv: Menjanje količina artikala u korpi

Aktori: Fizičko lice, pravno lice

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa postojećim artiklima u korpi

Osnovni scenario:

1.Korisnik bira opciju za promenu količine artikala (APUSO)

2.Korisnik unosi novu količinu (APUSO)

3.Korisnik poziva sistem da izvrši promenu količine (APSO)

4.Sistem izvršava operaciju promene količine (SO)

5.Sistem obaveštava korisnika da je količina promenjena (IA)

Alternativni scenariji:

4.1 Uneta količina je nula (IA)

**SK8: Kreiranje porudžbine**

Naziv: Kreiranje porudžbine

Aktori: Fizičko lice, pravno lice

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa korpom

Osnovni scenario:

1.Korisnik bira opciju za potvrđivanje porudžbine (APUSO)

2.Korisnik unosi lokaciju za isporuku (APUSO)

3.Korisnik poziva sistem da kreira porudžbinu (APSO)

4.Sistem izvršava operaciju kreiranja porudžbine (SO)

5.Sistem obaveštava korisnika da je porudžbina kreirana (IA)

Alternativni scenariji:

4.1 Korpa je prazna (IA)

**SK9: Otkazivanje porudžbine**

Naziv: Otkazivanje porudžbine

Aktori: Fizičko lice, pravno lice

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa kreiranom porudžbinom

Osnovni scenario:

1.Korisnik bira opciju za otkazivanje porudžbine (APUSO)

2.Korisnik poziva sistem da otkaze porudžbinu (APSO)

3.Sistem izvršava operaciju otkazivanja porudžbine (SO)

4.Sistem obaveštava korisnika da je porudžbina otkazana (IA)

Alternativni scenariji:

3.1 Ne postoji ni jedna kreirana porudžbina (IA)

**SK10: Prihvatanje porudžbine**

Naziv: Prihvatanje porudžbine

Aktori: Konobar

Učesnici: Konobar I sistem

Preduslovi: Sistem je uključen I konobaru je prikazana forma sa kreiranim porudžbinama

Osnovni scenario:

1.Konobar bira opciju za prihvatanje porudžbine ( APUSO)

2.Konobar poziva sistem da prihvati porudžbinu (APSO)

3.Sistem izvršava operaciju prihvatanja porudžbine (SO)

4.Sistem obaveštava konobara da je porudžbina prihvaćena (IA)

Alternativni scenariji:

3.1 Porudžbina je već kreirana (IA)

**SK11: Odbijanje porudžbine**

Naziv: Odbijanje porudžbine

Aktori: Konobar

Učesnici: Konobar I sistem

Preduslovi: Sistem je uključen I konobaru je prikazana forma sa kreiranim porudžbinama

Osnovni scenario:

1.Konobar bira opciju za odbijanje porudžbine (APUSO)

2.Konobar poziva sistem da odbije porudžbinu (APSO)

3.Sistem izvršava operaciju odbijanja porudžbina (SO)

4.Sistem obaveštava konobara da je porudžbina prihvaćena (IA)

Alternativni scenariji:

3.1 Porudžbina je već odbijena (IA)

**SK12: Ručno unošenje porudžbine**

Naziv: Odbijanje porudžbine

Aktori: Konobar

Učesnici: Konobar I sistem

Preduslovi: Sistem je uključen I konobaru je prikazana forma sa kreiranim porudžbinama

Osnovni scenario:

1.Konobar bira unosi neophodne podatke za porudžbinu (APUSO)

2.Konobar poziva sistem da kreira porudžbinu (APSO)

3.Sistem izvršava operaciju kreiranja porudžbine (SO)

4.Sistem obaveštava konobara da je porudžbina kreirana (IA)

Alternativni scenariji:

3.1 Nisu uneti validni podaci (IA)

**SK13: Uvid u statistiku**

Naziv: Odbijanje porudžbine

Aktori: Menadžer

Učesnici: Menadžer I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa neophodnim podacima za vođenje statistike

Osnovni scenario:

1.Korisnik bira opciju za prikaz statistike (APUSO)

2.Korisnik poziva sistem da prikaze podatke (APSO)

3.Sistem prikazuje tražene podatke (SO)

4.Sistem obaveštava korisnika da su podaci uspešno obračunati (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK13.1: Najčešće poručivani proizvodi**

Naziv: Najčešće poručivani proizvodi

Aktori: Menadžer

Učesnici: Menadžer I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa neophodnim podacima za vođenje statistike

Osnovni scenario:

1.Korisnik bira opciju za prikaz liste proizvoda koji su najčešće poručivani (APUSO)

2.Korisnik poziva sistem da prikaze podatke (APSO)

3.Sistem prikazuje tražene podatke (SO)

4.Sistem obaveštava korisnika da su podaci uspešno obračunati (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK13.2: Prihodi po mesecima**

Naziv: Prihodi po mesecima

Aktori: Menadžer

Učesnici: Menadžer I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa neophodnim podacima za vođenje statistike Osnovni scenario:

1.Korisnik bira opciju za prikaz ostvarenih prihoda po mesecima (APUSO)

2.Korisnik poziva sistem da prikaze podatke (APSO)

3.Sistem prikazuje tražene podatke (SO)

4.Sistem obaveštava korisnika da su podaci uspešno obračunati (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK13.3: Prihodi po godinama**

Naziv: Prihodi po godinama

Aktori: Menadžer

Učesnici: Menadžer I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa neophodnim podacima za vođenje statistike

Osnovni scenario:

1.Korisnik bira opciju za prikaz ostvarenih prihoda po godinama (APUSO)

2.Korisnik poziva sistem da prikaze podatke (APSO)

3.Sistem prikazuje tražene podatke (SO)

4.Sistem obaveštava korisnika da su podaci uspešno obračunati (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK13.4: Prihodi po korisnicima**

Naziv: Prihodi po korisnicima

Aktori: Menadžer

Učesnici: Menadžer I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa neophodnim podacima za vođenje statistike

Osnovni scenario:

1.Korisnik bira opciju za prikaz ostvarenih prihoda po korisnicima (APUSO)

2.Korisnik poziva sistem da prikaze podatke (APSO)

3.Sistem prikazuje tražene podatke (SO)

4.Sistem obaveštava korisnika da su podaci uspešno obračunati (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK13.5: Dnevni nedeljni mesečni i godišnji prihod**

Naziv: Dnevni nedeljni mesečni i godišnji prihod

Aktori: Menadžer

Učesnici: Menadžer I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa neophodnim podacima za vođenje statistike Osnovni scenario:

1.Korisnik bira opciju za prikaz ostvarenih prihoda na dnevnom, nedeljnom, mesečnom I godišnjem nivou (APUSO)

2.Korisnik poziva sistem da prikaze podatke (APSO)

3.Sistem prikazuje tražene podatke (SO)

4.Sistem obaveštava korisnika da su podaci uspešno obračunati (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK14: Prihvati porudžbinu**

Naziv: Prihvati porudžbinu

Aktori: Dostavljač

Učesnici: Dostavljač I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa spremnim porudžbinama za isporuku

Osnovni scenario:

1.Korisnik poziva sistem da prihvati porudžbinu (APSO)

2.Sistem dodeljuje porudžbinu dostavljaču (SO)

3.Sistem obaveštava korisnika da je porudžbina prihvaćena (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK15: Pošalji lokaciju**

Naziv: Pošalji lokaciju

Aktori: Dostavljač

Učesnici: Dostavljač I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa spremnim porudžbinama za isporuku

Osnovni scenario:

1.Korisnik poziva sistem da pošalje trenutnu lokaciju (APSO)

2.Sistem smesta trenutnu lokaciju u bazu podataka (SO)

3.Sistem obaveštava korisnika da je lokacija pribavljena (IA)

Alternativni scenariji:

3.1 Nema podataka u bazi (IA)

**SK16: Završi isporuku**

Naziv: Završi isporuku Aktori: Dostavljač

Učesnici: Dostavljač I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa spremnim porudžbinama za isporuku Osnovni scenario:

1.Korisnik poziva sistem da oznaci porudžbinu kao isporučenu (APSO)

2.Sistem menja stanje porudžbine (SO)

3.Sistem obaveštava korisnika da je stanje promenjeno (IA)

Alternativni scenariji:

2.1 Nema podataka u bazi (IA)

**SK17: Lociranje porudžbine**

Naziv: Završi isporuku

Aktori: Fizičko I pravno lice

Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I je prikazana forma sa kreiranim porudžbinama

Osnovni scenario:

1.Korisnik poziva sistem da pošalje zahtev za lociranje porudžbine (APSO)

2.Sistem menja stanje zahteva za lociranje (SO)

3.Sistem obaveštava korisnika da je zahtev poslat (IA)

Alternativni scenariji:

2.1 Nema podataka u bazi (IA)

# FAZA ANALIZE

## SISTEMSKI DIJAGRAMI SEKVENCI ZA SLUČAJEVE KORIŠĆENJA

Dijagrami prikazani u fazi analize spadaju u UML dijagrame sekvenci. Kao što se iz samog naziva može zaključiti, dijagrami prikazuju sekvence za pojedine slučajeve korišćenja. Zbog ograničenog obima završnog rada prikazani su svega par karakterističnih dijagrama sekvenci.

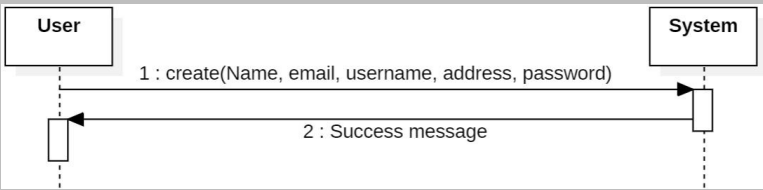
**DSSK1: Kreiranje korisničkog naloga**

Naziv: Kreiranje korisničkog naloga   
Aktori: Fizičko lice, pravno lice   
Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma za logovanje ili kreiranje novog naloga.   
Osnovni scenario:

1.Korisnik poziva sistem kako bi izvršio kreiranje naloga sa unetim podacima (APSO)

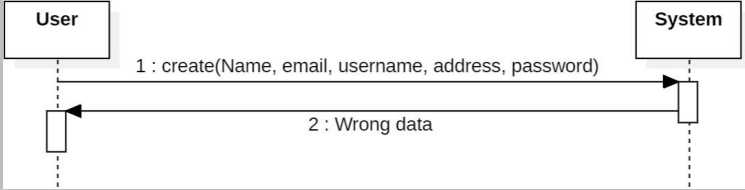
2.Sistem prikazuje poruku o uspešnosti kreiranja naloga (IA)



Slika 9. Kreiranje korisničkog naloga

Alternativni scenariji:

* 1. Sistem pronalazi korisnički nalog sa već unetim korisničkim imenom, prikazuje poruku korisniku I prekida izvršavanje scenarija (IA)



Slika 10. Kreiranje korisničkog naloga alternativni scenario

**DSSK2: Prijavljivanje na nalog**

Naziv: Prijavljivanje na nalog

Aktori: Fizičko lice, pravno lice, šef restorana, konobar, menadžer, dostavljač

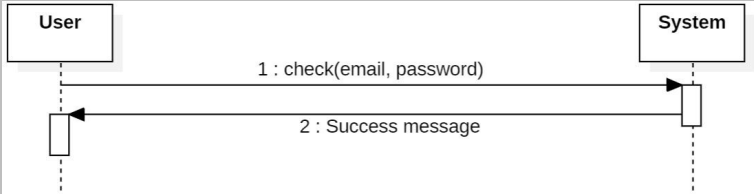
Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma za logovanje ili kreiranje novog naloga.

Osnovni scenario:

1. Korisnik poziva sistem da proveri postoji li nalog sa unetim podacima (APSO)

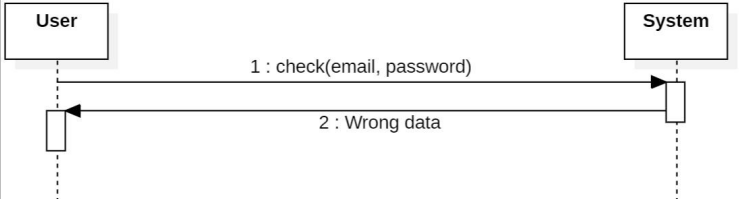
2. Sistem pokazuje poruku o uspešnosti logovanja (IA)



Slika 11. Prijavljivanje na nalog

Alternativni scenariji:

3.1Sistem ne pronalazi nalog sa unetim podacima I obustavlja proces prijavljivanja(IA)



Slika 12. Prijavljivanje na nalog alternativni scenario

**DSSK3: Brisanje korisničkog naloga**

Naziv: Brisanje naloga

Aktori: Fizičko lice, pravno lice, sef restorana

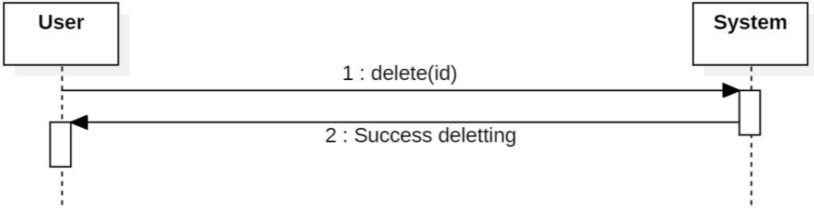
Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa informacijama korisničkog naloga.

Osnovni scenario:

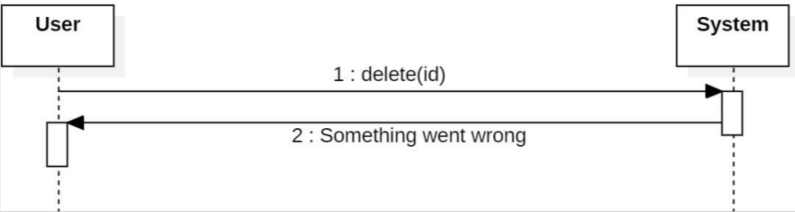
1.Korisnik poziva sistem da izvrši operaciju brisanja (APSO)

2.Sistem obaveštava korisnika da je brisanje uspešno izvršeno (IA)

  
Slika 13. Brisanje korisničkog naloga

Alternativni scenariji:

1.1 Došlo je do prekida internet veze (IA)



Slika 14. Brisanje korisničkog naloga alternativni scenario

**DSSK4: Menjanje šifre korisničkog naloga**

Naziv: Menjanje šifre naloga

Aktori: Fizičko lice, pravno lice, sef restorana,

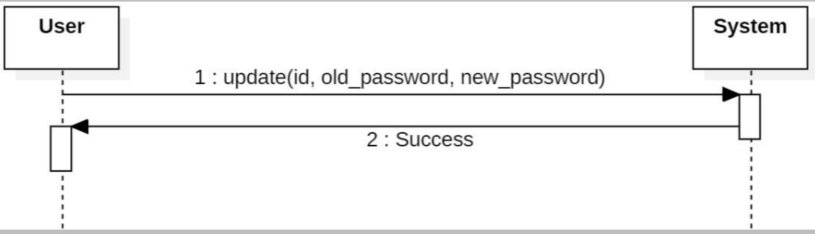
Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa informacijama korisničkog naloga.

Osnovni scenario:

1.Korisnik poziva sistem da izvrši promenu šifre (APSO)

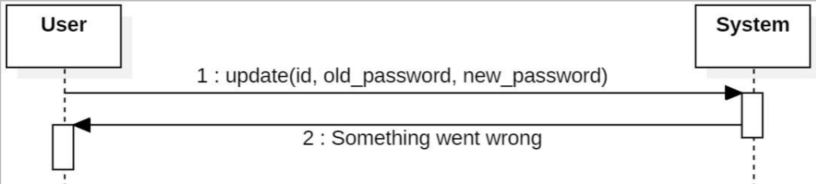
2.Sistem obaveštava korisnika da je promena šifre uspešno izvršena (IA)



Slika 15. Menjanje šifre korisničkog naloga

Alternativni scenariji:

1.1 Šifre se ne poklapaju (IA)



Slika 16. Menjanje šifre korisničkog naloga alternativni scenario

**DSSK5: Dodavanje artikla u korpu**

Naziv: Dodavanje artikla u korpu

Aktori: Fizičko lice, pravno lice

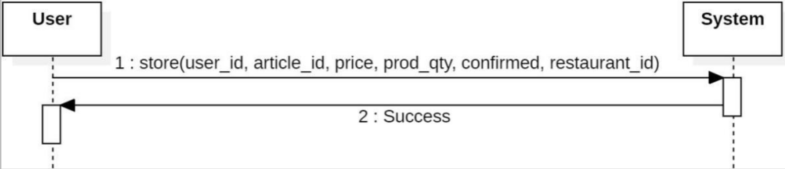
Učesnici: Korisnik I sistem

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa postojećim artiklima

Osnovni scenario:

1.Korisnik poziva sistem da smesti artikal u korpu (APSO)

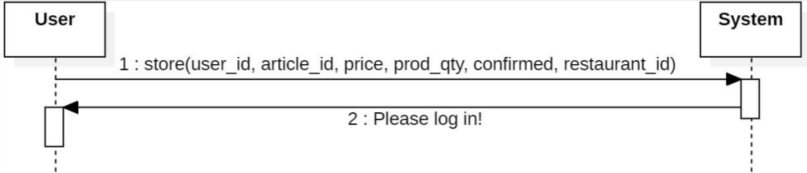
2.Sistem obaveštava korisnika da je artikal dodat u korpu (IA)



Slika 17. Dodavanje artikla u korpu

Alternativni scenariji:

1.1 Korisnik nije prijavljen (IA)



Slika 18. Dodavanje artikla u korpu alternativni scenario

Slika 52. Lociraj porudžbinu alternativni scenario

## DEFINISANJE UGOVORA O SISTEMSKIM OPERACIJAMA

Za svaku od uočenih sistemskih operacija prave se ugovori (Eng. contracts). Ugovori opisuju ponašanje sistemske operacije, tako što opisuje šta operacija radi, ali ne i kako. Jedan ugovor vezan je za jednu sistemsku operaciju. Uočene su sledeće sistemske operacije

1. createAccount(name, username, email, address, password)
2. Check(email, password)
3. showProducts(name, price)
4. addToCart(article\_id, user\_id, article\_name, prod\_qty, price)
5. deleteFromCart(id)
6. updateQty(id, prod\_qty)
7. CreateOrder(user\_id, location\_id, order\_date, payment\_method, shipping\_address, name, company\_name, phone, comment, price, article\_name, prod\_qty, accepted, cenceled, )
8. CancelOrder(id, cancceled)
9. AcceptOrder(id, accepted)
10. RejectOrder(id, accepted)
11. ShowData(id, sum(), avg() )
12. popularArticles(cart\_count)
13. amounts(order\_date, price)
14. months(order\_date, price)
15. years(order\_date, price)
16. users(user\_id, price)
17. AcceptOrder(id, boss\_id)
18. sendLocation(id, location)
19. finishDelivery(id, delivered)
20. locate(order\_locations.id)

**Ugovor G1: createAccount**

Operacija: createAccount(name, username, email, address, password): bool

Veza sa SK: SK1

Preduslovi: Nalog ne postoji u bazi

Postuslovi: Kreiran je novi nalog

**Ugovor G2: check**

Operacija: check(username, password): bool

Veza sa SK: SK2

Preduslovi: Nalog postoji u bazi

Postuslovi: Korisnik je pristupio aplikaciji

**Ugovor G3: showProducts**

Operacija: showProducts(name, price): bool

Veza sa SK: SK3

Preduslovi: Baza nije prazna

Postuslovi: Korisnik je poslao zahtev za prikaz podataka

**Ugovor G4: addToCart**

Operacija: addToCart(article\_id, user\_id, article\_name, prod\_qty, price): bool

Veza sa SK: SK4

Preduslovi: Korisnik je poslao zahtev za unos artikla u korpu

Postuslovi: Artikal je dodat u korpu

**Ugovor G5: deleteFromCart**

Operacija: deleteFromCart(id): bool

Veza sa SK: SK5

Preduslovi: Korisnik je poslao zahtev za brisanje artikla iz korpe

Postuslovi: Artikal je izbrisan iz korpe

**Ugovor G6: updateQty**

Operacija: updateQty(id, prod\_qty): bool

Veza sa SK: SK6

Preduslovi: Korisnik je poslao zahtev za promenu količine artikla iz korpe

Postuslovi: Količina artikla u korpi je promenjena

**Ugovor G7: createOrder**

Operacija: createOrder(user\_id, location\_id, order\_date, payment\_method, shipping\_address, name, company\_name, phone, comment, price, article\_name, prod\_qty, accepted, cenceled, ): bool

Veza sa SK: SK7

Preduslovi: Korisnik je poslao zahtev za kreiranje porudžbine

Postuslovi: Porudžbina je kreirana

**Ugovor G8: cancelOrder**

Operacija: cancelOrder(id, cancceled): bool

Veza sa SK: SK8

Preduslovi: Korisnik je poslao zahtev za otkazivanje porudžbine

Postuslovi: Porudžbina je otkazana

**Ugovor G9: acceptOrder**

Operacija: acceptOrder(id, accepted): bool

Veza sa SK: SK9

Preduslovi: Konobar je poslao zahtev za prihvatanje porudžbine

Postuslovi: Porudžbina je prihvaćena

**Ugovor G10: rejectOrder**

Operacija: rejectOrder(id, accepted): bool

Veza sa SK: SK10

Preduslovi: Konobar je poslao zahtev za otkazivanje porudžbine

Postuslovi: Porudžbina je otkazana

**Ugovor G11: showData**

Operacija: showData(id, sum(), avg() ): bool

Veza sa SK: SK11

Preduslovi: Korisnik je poslao zahtev za obračunavanje I prikaz podataka

Postuslovi: Podaci su obračunati I prikazani

**Ugovor G12: popularArticles**

Operacija: popularArticles(id, cart\_id, name ): bool

Veza sa SK: SK13.1

Preduslovi: Korisnik je poslao zahtev za obračunavanje I prikaz podataka

Postuslovi: Podaci su obračunati I prikazani

**Ugovor G13: months**

Operacija: months(id, order\_date, price): bool

Veza sa SK: SK13.2

Preduslovi: Korisnik je poslao zahtev za obračunavanje I prikaz podataka

Postuslovi: Podaci su obračunati I prikazani

**Ugovor G14: years**

Operacija: years(id, order\_date, price): bool

Veza sa SK: SK13.3

Preduslovi: Korisnik je poslao zahtev za obračunavanje I prikaz podataka

Postuslovi: Podaci su obračunati I prikazani

**Ugovor G15: users**

Operacija: users(id, user\_id, price): bool

Veza sa SK: SK13.4

Preduslovi: Korisnik je poslao zahtev za obračunavanje I prikaz podataka

Postuslovi: Podaci su obračunati I prikazani

**Ugovor G16: amounts**

Operacija: amounts(id, order\_date, price): bool

Veza sa SK: SK13.5

Preduslovi: Korisnik je poslao zahtev za obračunavanje I prikaz podataka

Postuslovi: Podaci su obračunati I prikazani

**Ugovor G17: AcceptOrder**

Operacija: AcceptOrder(id, boss\_id): bool

Veza sa SK: SK14

Preduslovi: Porudžbina nije u procesu isporuke

Postuslovi: Porudžbina je prihvaćena

**Ugovor G18: sendLocation**

Operacija: sendLocation(id, location):bool

Veza sa SK: SK15

Preduslovi: Porudžbina je u procesu isporuke

Postuslovi: Porudžbina je prihvaćena

**Ugovor G19: finishDelivery**

Operacija: finishDelivery(id, delivered):bool

Veza sa SK: SK16

Preduslovi: Porudžbina nije isporučena

Postuslovi: Porudžbina je isporučena

**Ugovor G20: locate**

Operacija: locate(order\_locations.id):bool

Veza sa SK: SK17

Preduslovi: Porudžbina je kreirana

Postuslovi: Zahtev za lociranje je poslat

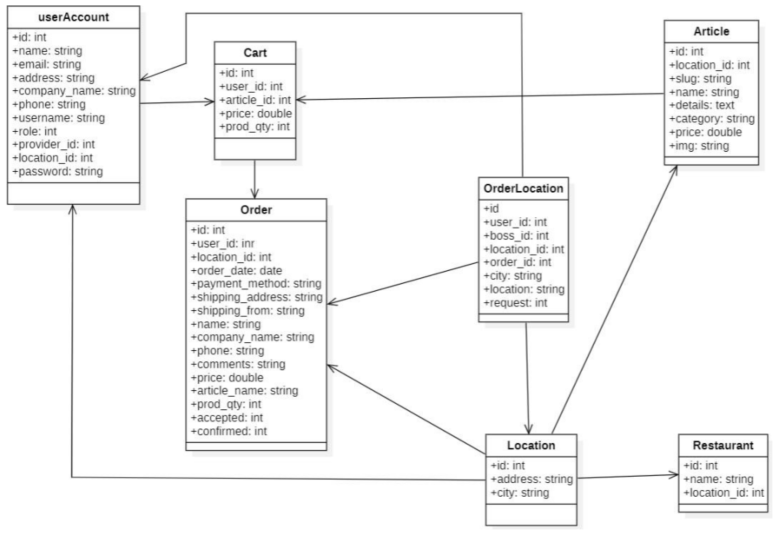
# **KONCEPTUALNI (DOMENSKI) MODEL**

Koriste podvučene imenice iz slučajeva korišćenja da bi kreirali koncepte u konceptualnom modelu.

Neke od imenica, ako identifikuju jednostavne tipove podataka, koriste se za kreiranje atributa ovih koncepata. Kreiraju se asocijacije između pojmova.

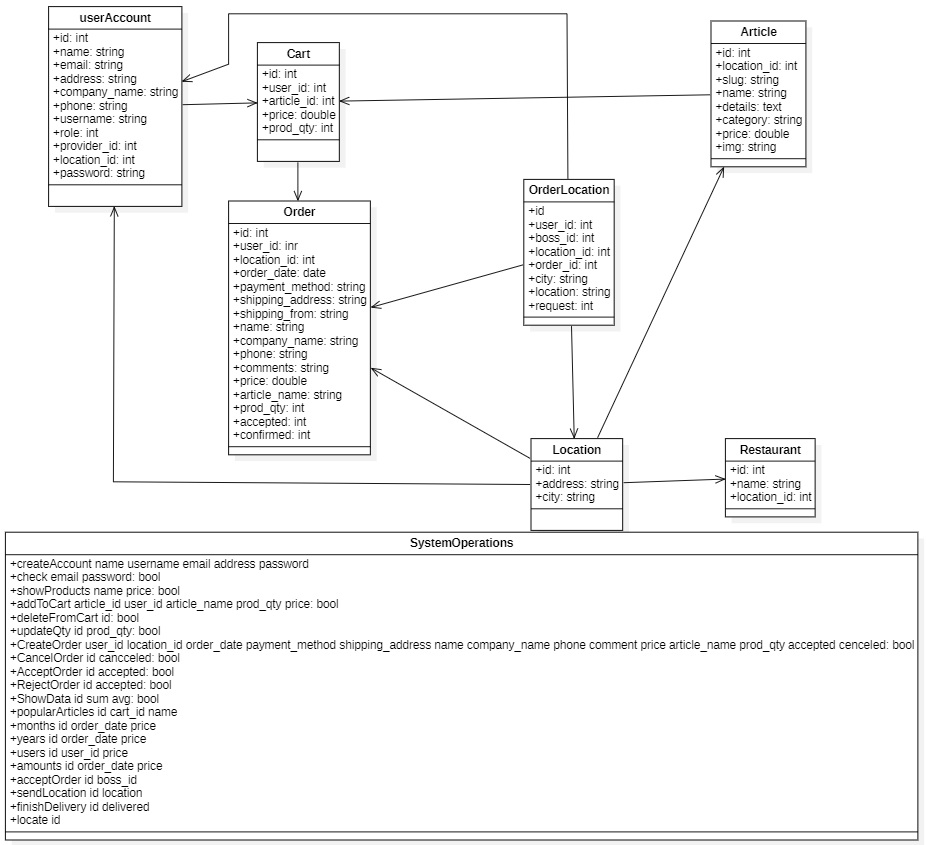
Konceptualni model nastaje na osnovu podataka iz funkcionalnog zahteva i slučajeva korišćenja.

Konceptualni model kreira se klasnim dijagramom. Dodaju se metode i atributi koji su otkriveni u dijagramima kolaboracije klasama u konceptualnom modelu.



Slika 53. Konceptualni model

Kao rezultat analize scenarija SK i pravljenja konceptualnog modela dobija se logička struktura i ponašanje softverskog sistema.



Slika 54. Struktura sistema

## RELACIONI MODEL

Na osnovu konceptualnog modela može se napraviti relacioni model, koji će predstavljati osnovu za projektovanje relacione baze podataka.

**UserAccount**(ID, name, email, address, company\_name, phone, username, role, provider\_id, location\_id, password)

**Cart**(ID, User\_id, Article\_id, price, prod\_qty)

**Article**(ID, Location\_id, slug, name, details, category, price, img)

**Order**(ID, user\_id, location\_id, order\_date, payment\_method, shipping\_address, shipping\_from, name, company\_name, phone, comments, price, article\_name, prod\_qty, accepted, confirmed)

**Location**(ID, address, city)

**Restaurant**(ID, name, Location\_id)  
**OrderLocation**(ID, user\_id, boss\_id, location\_id, order\_id, city, location, request)

# FAZA PROJEKTOVANJA

## DIJAGRAMI SEKVENCI I KOLABORACIONI DIJAGRAMI ZA SISTEMSKE OPERACIJE

Sekvencijalni dijagrami se kreiraju za svaki scenario korišćenja. Svaki događaj sekvence na dijagramu odgovara interakciji korisnika sa sistemom navedenim u proširenom slučaju korišćenja.

Kreira se kolaboracioni dijagram za svaki sistemski događaj u dijagramima sekvence.

Kolaboracionim dijagramima dodeljuje se odgovornost klasama u konceptualnom modelu da bi se ispunili naknadni uslovi u ugovorima.

Koriste se asocijacije iz konceptualnog modela u kombinaciji sa obrascima da bi se dodelile odgovornosti.

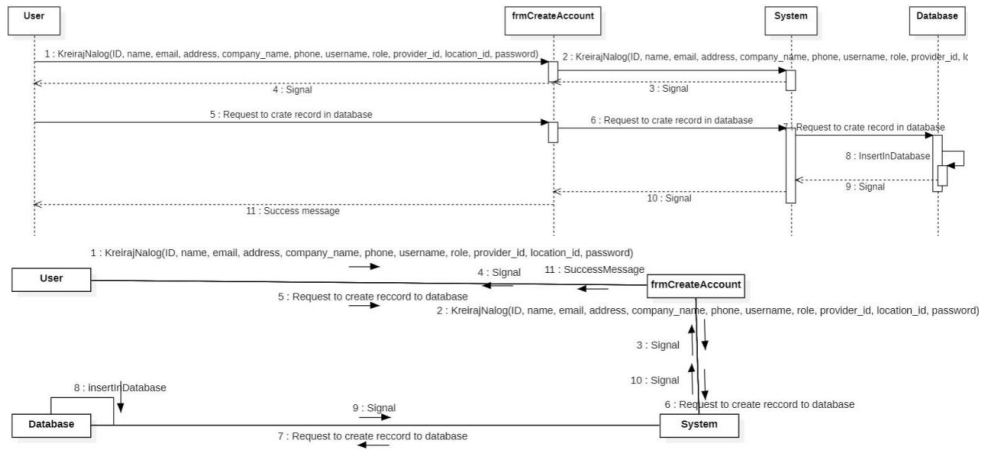
Zbog ograničenog obima završnog rada prikazani su svega par ugovora sistemskih operacija, koji se odnose na pet karakterističnih operacija.

**Ugovor G1: createAccount**

Operacija: createAccount(name, username, email, address, password): bool

Preduslovi: Nalog ne postoji u bazi

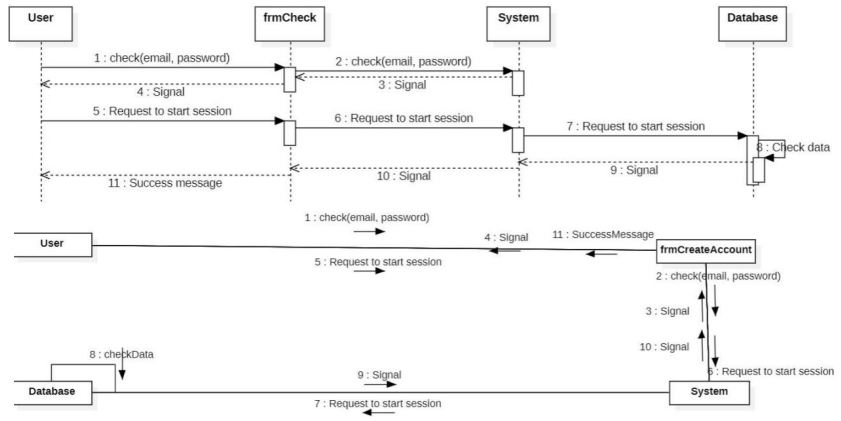
Postuslovi: Kreiran je novi nalog



Slika 55. Sekvencijalni I kolaboracioni dijagram createAccount

**Ugovor G2: check**

Operacija: check(username, password): bool   
Preduslovi: Nalog postoji u bazi   
Postuslovi: Korisnik je pristupio aplikaciji



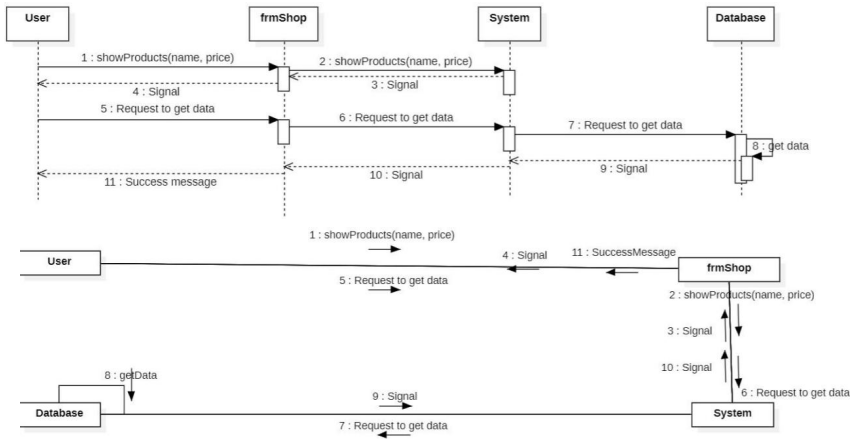
Slika 56. Sekvencijalni I kolaboracioni dijagram check

**Ugovor G3: showProducts**

Operacija: showProducts(name, price): bool

Preduslovi: Baza nije prazna

Postuslovi: Korisnik je poslao zahtev za prikaz podataka



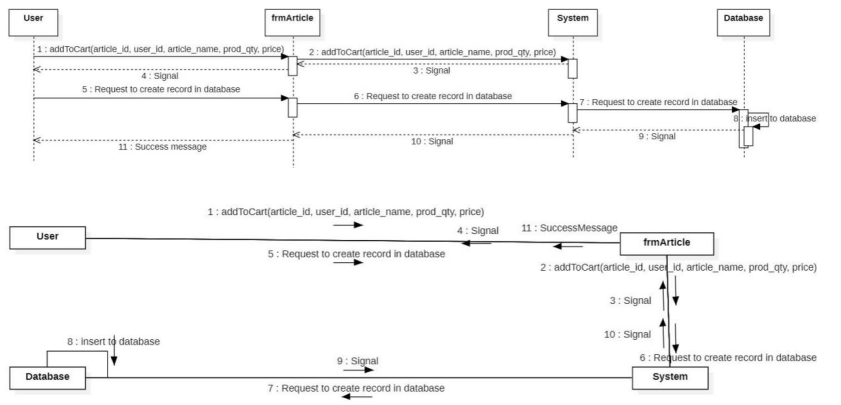
Slika 57. Sekvencijalni I kolaboracioni dijagram showProducts

**Ugovor G4: addToCart**

Operacija: addToCart(article\_id, user\_id, article\_name, prod\_qty, price): bool

Preduslovi: Korisnik je poslao zahtev za unos artikla u korpu

Postuslovi: Artikal je dodat u korpu



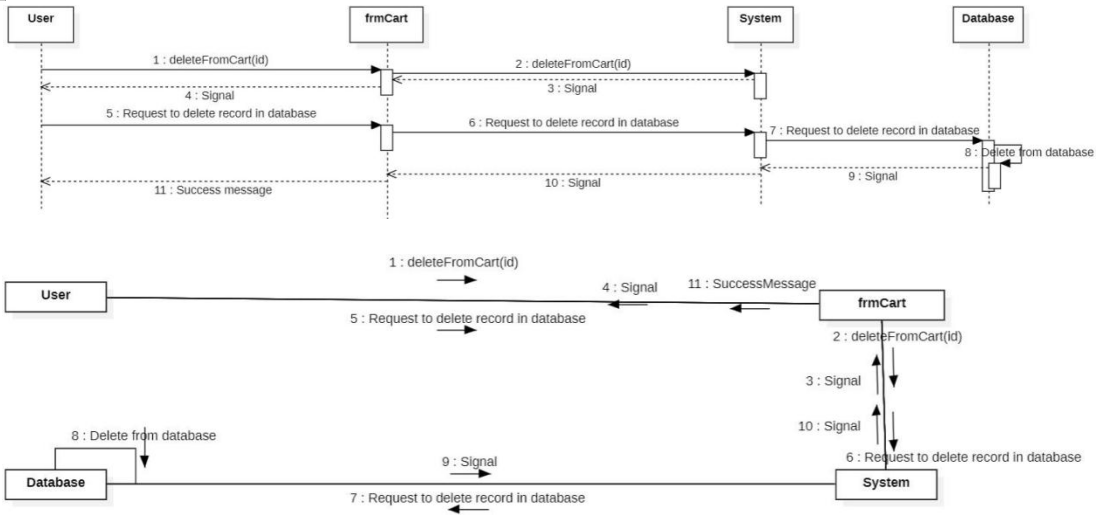
Slika 58. Sekvencijalni I kolaboracioni dijagram addToCart

**Ugovor G5: deleteFromCart**

Operacija: deleteFromCart(id): bool

Preduslovi: Korisnik je poslao zahtev za brisanje artikla iz korpe

Postuslovi: Artikal je izbrisan iz korpe



Slika 59. Sekvencijalni I kolaboracioni dijagram deleteFromCart

**Ugovor G6: updateQty**

Operacija: updateQty(id, prod\_qty): bool

Preduslovi: Korisnik je poslao zahtev za promenu količine artikla iz korpe

Postuslovi: Količina artikla u korpi je promenjena

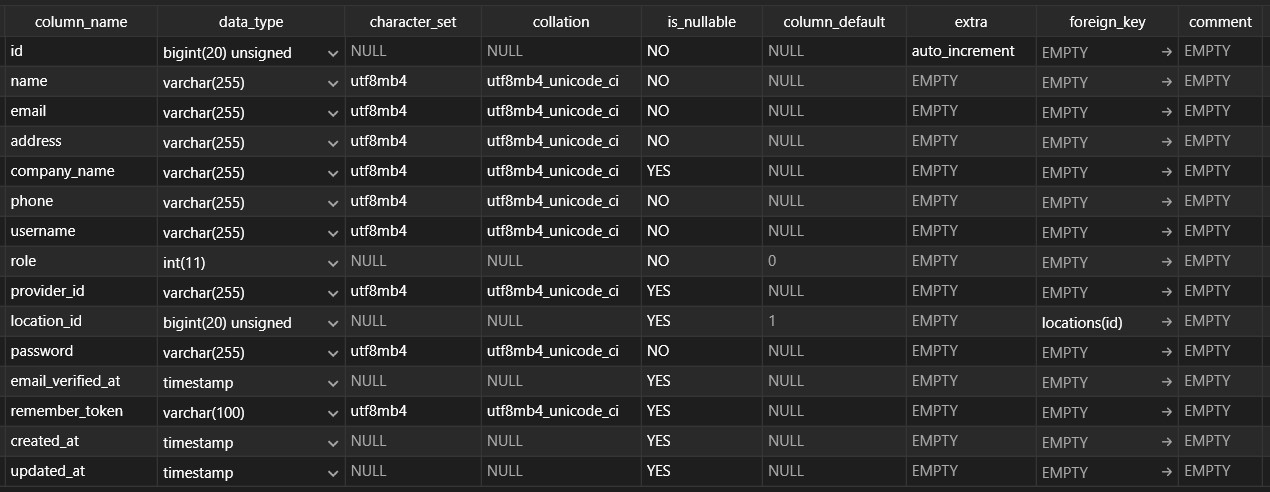


Slika 60. Sekvencijalni I kolaboracioni dijagram updateQty

# **PROJEKTOVANJE BAZE PODATAKA**

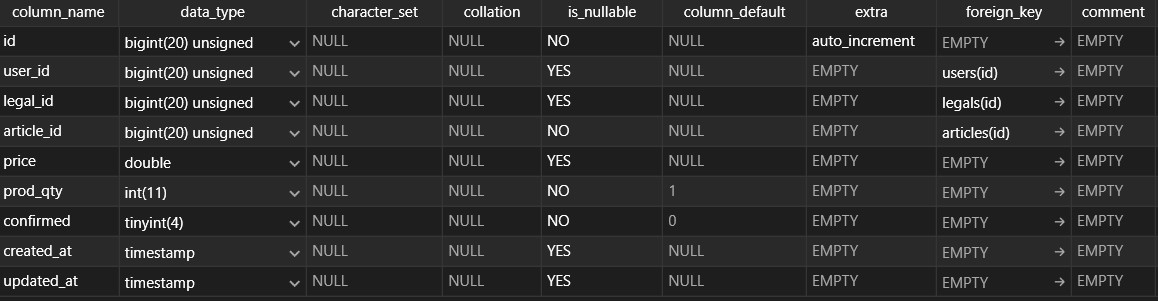
Skladište podataka se projektuje na osnovu softverskih klasa strukture. Kao rezultat se dobijaju tabele relacionog sistema za upravljanje bazom podataka.

Tabela: users (korisnici)

  
  
Slika 75. Users tabela

U datoj tabeli beleže se podaci svih registrovanih korisnika našeg sistema. Korisnici svih rola nalaze se u ovoj tabeli, međutim sistem raspoznaje korisnike na osnovu kolone “role”, što u velikoj meri smanjuje kompleksnost baze podataka. Svaki registrovani korisnik neophodno je da poseduje mejl adresu koja je jedinstven i unikatan podatak, adresne podatke radi isporuke, i jako bitan podatak “location\_id”. Navedeni podatak je podrazumevano za sve korisnike 1, sto je nasleđeni podatak iz “location” tabele. U daljim proširivanjima sistema biće moguće dodati više restorana, od kojih će pojedini imati lokale na više različitih lokacija, što nam omogućava da znamo tačno u kom lokalu je određeni korisnik zaposlen.

Tabela: cart (korpa)

  
  
Slika 76. Cart tabela

U datoj tabeli beleže se artikli dodati u korpu. Ovo omogućava korisniku da i nakon ponovnog prijavljivanja na nalog ima sačuvane artikle koje je nekom prilikom dodao u korpu. Data tabela nasleđuje ključeve iz tabela “Users”, “Articles” koji omogućavaju da se zna tačno koji artikal je koji korisnik dodao u korpu. Pored toga su naravno bitne informacije cena i količina artikala u korpi.

Tabela article (artikli)

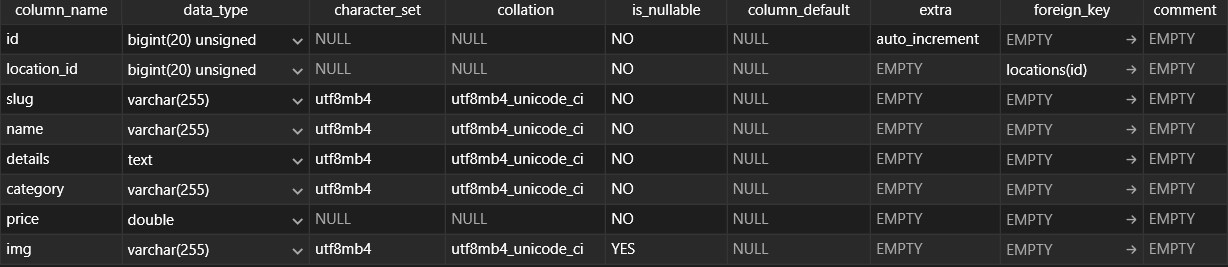
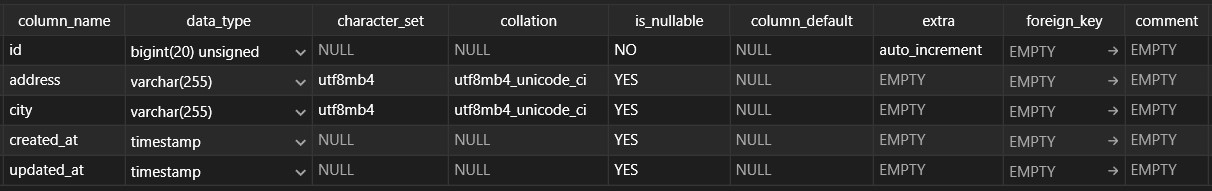
  
  
Slika 77. Article tabela

Tabela koja pruža informacije o svim artiklima, kao što su naziv, detalji kategorija i cena samog artikla. Kolona slug je bitna radi prikaza pojedinačnih artikala u Web aplikaciji. Tabela artikal nasleđuje spoljni ključ iz tabele Location, što je jako bitno u budućem proširivanju sistema, u kome će biti uključeni mnogi drugi restorani, koji će imati različite asortimane artikala.

Tabela Locations (Lokacije)

  
  
Slika 78. Locations tabela

Kao što je već pomenuto, u budućem proširivanju poslovanja, u sistem biće uključeni brojni restorani, od kojih će svaki imati svoju adresu i grad. To omogućava lako razvrstavanje korisnika po dotičnim restoranima.

Tabela Restaurants (Restorani)

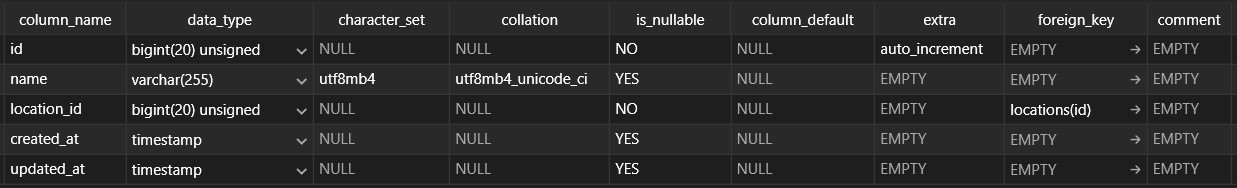
  
  
Slika 79. Restaurants tabela

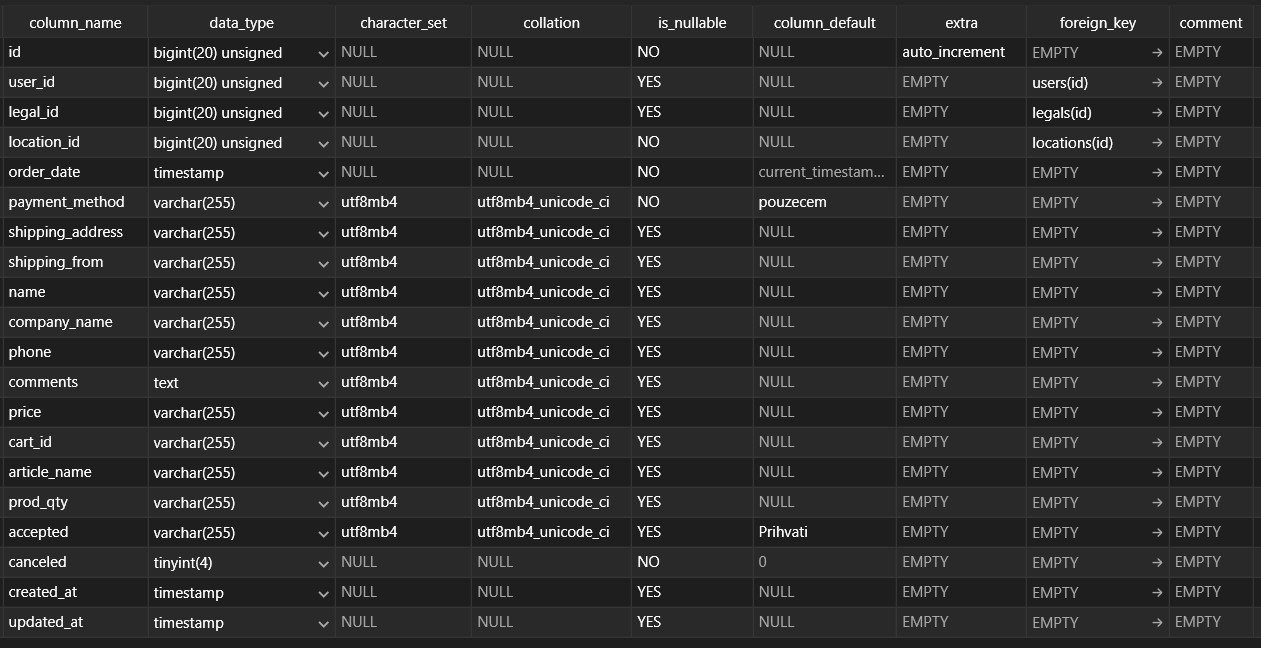
Tabela restorani pruža informacije o nazivu restorana, međutim kako je pomenuto da pojedini restorani mogu imati lokale na više različitih lokacija, tako će uvek postojati samo jedan naziv restorana, koji će informacije o adresi i gradu povlačiti iz tabele lokacije, pomoću nasleđenog ključa location\_id.

Tabela OrderLocation (Lokacija porudžbine)

  
  
Slika 80. OrderLocation tabela

Tabela lokacije porudžbina je usko vezana sa krajnjim korisnikom i dostavljačem (boss\_id kolona bi zapravo trebala biti deliverer\_id). Naime, redovi u tabeli se kreiraju onog trenutka kada konobar prihvati porudžbinu i tog trenutka se popunjavaju nasleđeni ključevi user\_id i order\_id. U trenutku kada određeni dostavljač prihvati porudžbinu popunjava se i kolona boss\_id. Kada su ove kolone popunjene može se otpočeti proces isporuke i komunikacije između korisnika i dostavljača. Prilikom traženja lokacije odnosno slanja zahteva za istom, popunjava se kolona request, čime dostavljač dobija informaciju da je potrebno da pošalje svoju lokaciju. Prilikom slanja lokacije, popunjava se kolova location sa trenutnim koordinatama a kolona request odnosno zahtev se briše. Takođe jako bitne kolone su delivered i canceled, koje pružaju informacije da li je porudžbina otkazana od strane korisnika i da li je ista dostavljena korisniku.

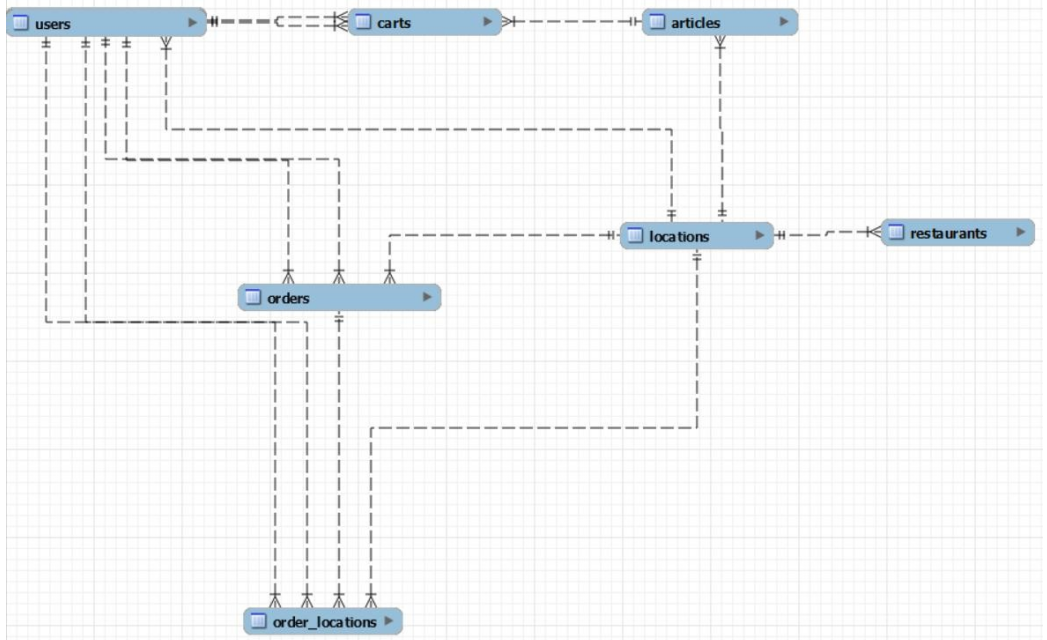
Tabela Orders (Porudžbine)



Slika 81. Orders Tabela

Tabela orders beleži sve kreirane porudžbine i daje informacije o tome koji korisnik je kreirao istu, preko nasleđenog ključa user\_id, kao i location\_id, odnosno tačno određeni lokal u kom je porudžbina kreirana, o čemu je već bilo reči ranije. Kolone kao što su accepted i canceled su bitne za konobarsku rolu, na osnovu kojih konoba ima evidenciju da li je porudžbina prihvaćena radi samog procesa pripreme iste, ili je korisnik otkazao porudžbinu što konobaru daje informaciju da porudžbina ne treba da se priprema. Kolone shipping\_from i shipping\_address daju informaciju gde porudžbina treba da se isporuči i odakle se isporučuje. Kolona shipping\_address se u prvi mah čini suvišnom, jer je mogla da se povuče iz tabele User, međutim kako korisnik ima mogućnost unosa neke druge adrese prilikom naručivanja, ova kolona je neophodna. Isto važi i za Name kolonu. Kolone article\_name i prod\_qty su zapravo nizovi artikala i količina tih artikala iz korisničke korpe, kako bi uklonili potrebu za kreiranjem više redova u bazi prilikom kreiranja porudžbine. Kolona price obračunava se na osnovu ukupne cene svih artikala u korpi. Kolona comment je opciona, u nju korisnik može upisati bilo kakvu vrstu napomene prilikom poručivanja.

Database diagram

  
  
Slika 82. Dijagram baze podataka

# **PROJEKTOVANJE KORISNIČKOG INTERFEJSA**

Za svaki slučaj korišćenja koji se izvodi nad ekranskom formom treba kreirati odgovarajuću ekransku formu i dati opis akcije.

Ova faza projektovanja bitna je kako za korisnike koji su radili ili će raditi na projektu, tako i za krajnje korisnike koji bi želeli da se na slikovit i prosto opisan način upoznaju sa radom i funkcionalnostima naše web aplikacije.

U nastavku prikazani su opisi, propraćeni slikama, pojedinih slučajeva korišćenja. Neki od njih će biti preskočeni ili u određenoj meri kraće opisani zbog ograničenog obima završnog rada.

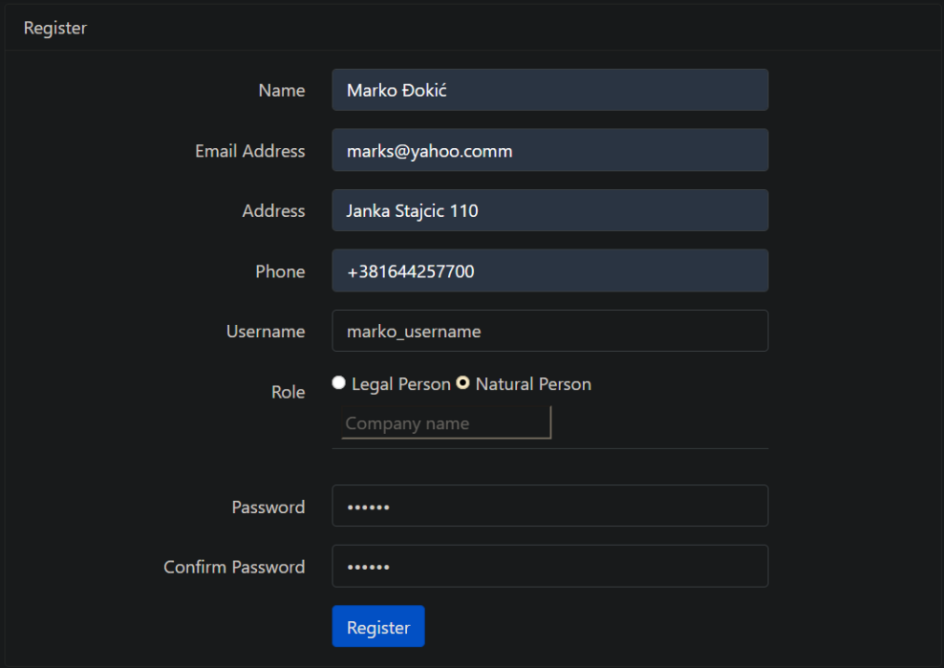
Kao što je navedeno već par puta, zbog ograničenog obima završnog rada, kroz slučajeve korišćenja biće obrađeni svega nekoliko istih.

Neke od ostalih funkcionalnosti Web aplikacije biće obrađene u poglavlju 9. Prilozi.  
  
**SK1: Kreiranje korisničkog naloga**

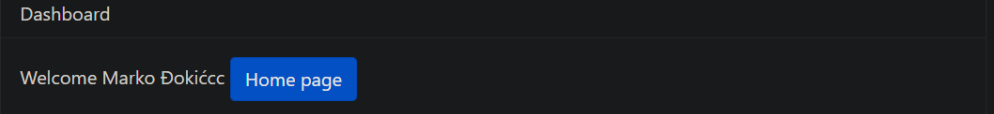
Preduslovi: Sistem je uključen I korisniku je prikazana forma za logovanje ili kreiranje novog naloga.

Osnovni scenario:

Korisniku je pri prvom ulasku na sajt prikazana forma za registraciju ili logovanje, gde je potrebno odabrati opciju za registraciju I uneti sve neophodne podatke. Nakon unosa svih podataka ukoliko su svi validni I ne postoji već takav registrovan korisnik, kreiraće se korisnički nalog.

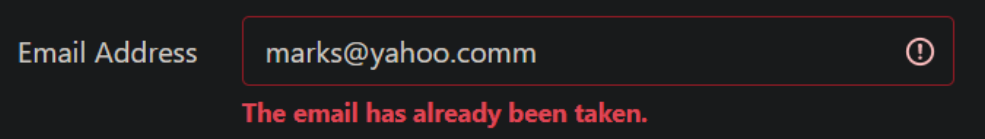


Slika 83. Registraciona forma

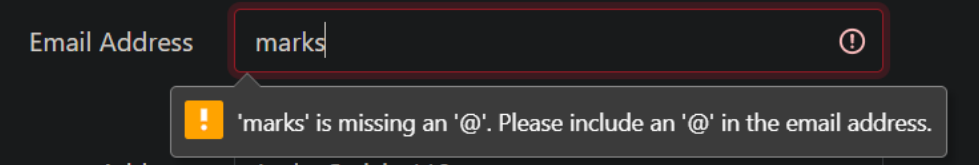


Slika 84. Pozdravna poruka

Međutim, ukoliko korisnik već postoji u bazi podataka ili neki od unetih podataka nije validan, pojaviće se odgovarajuća poruka o grešci.



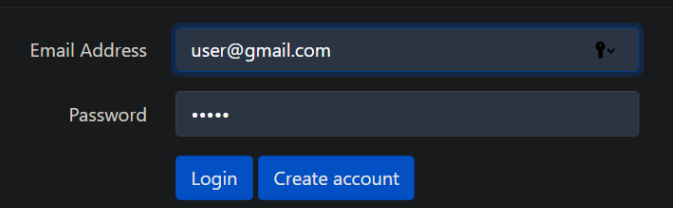
Slika 85. Postojeća mejl adresa

  
Slika 86. Nevalidni podaci

**SK2: Prijavljivanje na korisnički nalog**

Preduslovi: Sistem je uključen I korisniku je prikazana forma za logovanje ili kreiranje novog naloga.

Korisnik pri prvoj poseti web stranici bira opciju za prijavljivanje na nalog. Ukoliko nalog postoji i uneti su ispravni podaci korisniku će biti ispisana pozdravna poruka.



Slika 87. Forma za prijavljivanje



Slika 88. Pozdravna poruka

Ukoliko uneta šifra nije ispravna ili korisnički nalog ne postoji biće prikazana odgovarajuća poruka.

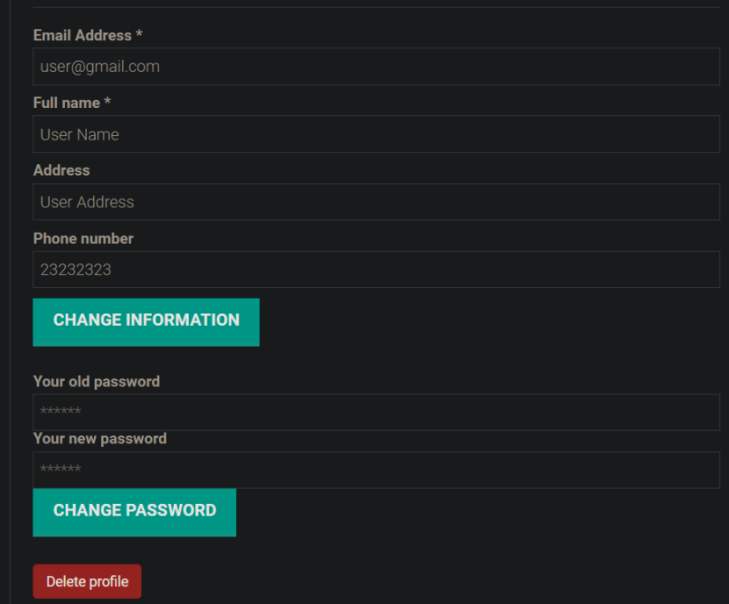


Slika 89. Poruka o grešci

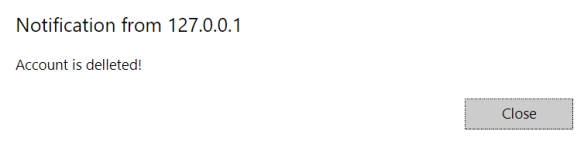
**SK3: Brisanje korisničkog naloga**

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa informacijama korisničkog naloga.

Korisnik bira opciju za brisanje korisničkog naloga.



Slika 90. Forma za promenu podataka i brisanje naloga

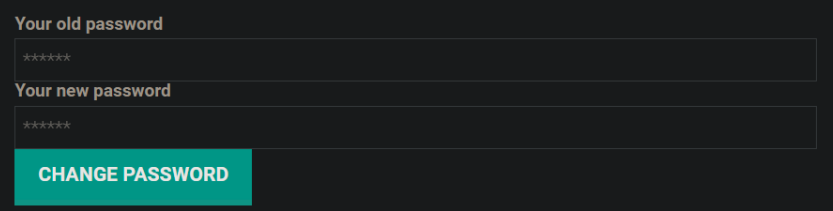


Slika 91. Poruka o izvršenoj operaciji brisanja

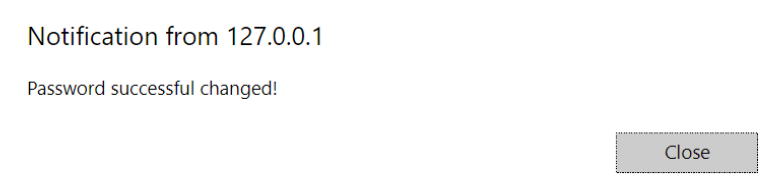
**SK4: Menjanje šifre korisničkog naloga**

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa informacijama korisničkog naloga.

Korisnik unosi već postojeću šifru i novu šifru nakon čega sistem proverava da li se uneta šifra poklapa sa šifrom iz baze podataka. Ukoliko se šifra poklapa korisniku će biti ispisana poruka o uspešnom ažuriranju podataka.

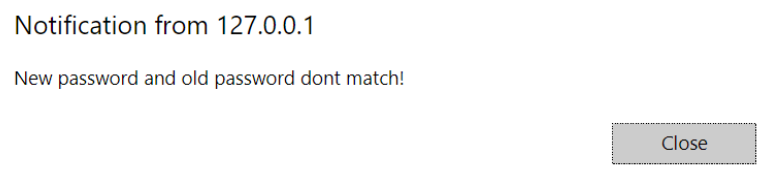


Slika 92. Forma za promenu podataka.



Slika 93. Poruka o uspešnom ažuriranju podataka

Ukoliko se stara šifra ne poklapa sa šifrom iz baze biće ispisana odgovarajuća poruka.

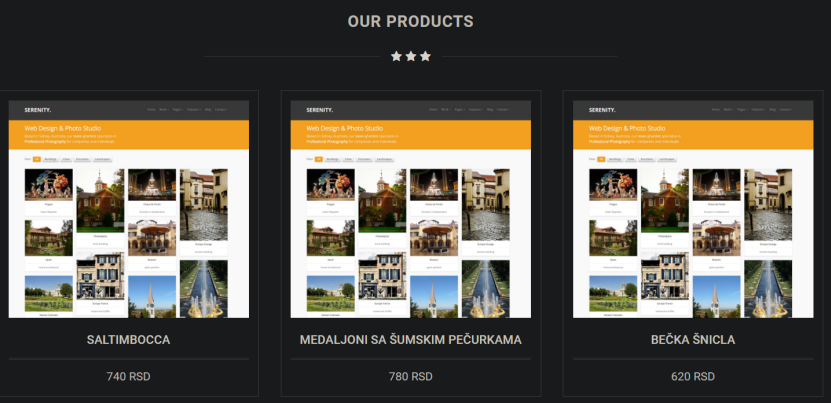


Slika 94. Poruka o pogrešno unetoj šifri

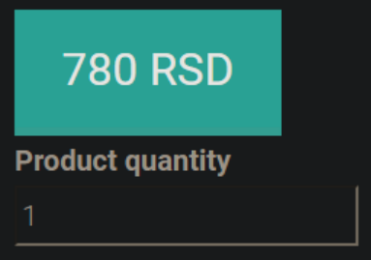
**SK5: Dodavanje artikla u korpu**

Preduslovi: Sistem je uključen I korisniku je prikazana forma sa postojećim artiklima

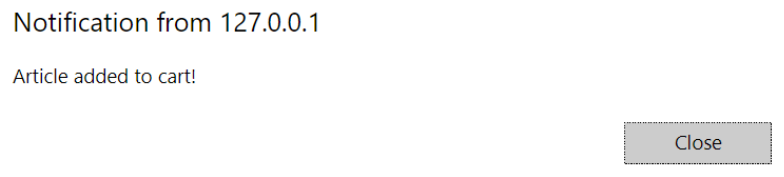
Korisniku je prikazana lista svih artikala iz baze podataka, među kojima korisnik bira željeni artikal, nakon čega korisnik unosi željenu količinu i dodaje artikal u korpu. Ukoliko je korisnik prijavljen na nalog biće ispisana poruka o uspešnom dodavanju artikla u korpu.



Slika 95. Lista artikala

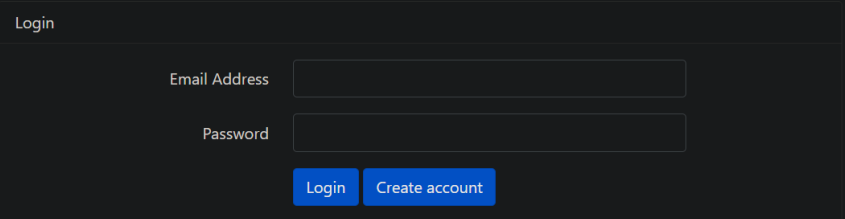


Slika 96. Forma za dodavanje artikla u korpu



Slika 97. Poruka o uspešnom dodavanju artikla u korpu

Ukoliko korisnik nije prijavljen, biće redirektovan na formu za prijavljivanje.



Slika 98. Forma za prijavljivanje na nalog

# ZAKLJUČAK

U završnom radu projektovana je i implementirana web aplikacija primenom PHP Laravel okvira zasnovanom na MVC šablonu.

Cilj završnog rada je izrada aplikacije za podršku i unapređenje u radu ugostiteljskog objekta koji se bavi dostavom hrane. Aplikacija pruža korisniku jednostavan interfejs sa prikazom celokupnog asortimana artikala u ponudi, kao i sve detalje pojedinih artikala koji bi korisnika mogli da zanimaju. Pored toga korisnik je u mogućnosti da kroz svega par minuta kreira svoju porudžbinu, nakon čega se sav dalji proces, opet na vrlo jednostavan način, odvija među zaposlenima u restoranu. Ono sto ovu aplikaciju izdvaja od pojedinih je na izgled jednostavna ali vrlo bitna funkcija, praćenja same porudžbine.

Aplikacija ima potencijal za mnogobrojna unapređenja.

Implementacija push notifikacija. Kako je u aplikaciji prisutna komunikacija među različitim aktorima u sistemu, vrlo je verovatna buduća nadogradnja aplikacije implementacijom push notifikacija, radi poboljšanja te komunikacije.

Izrada Android i IOS aplikacija. Web aplikacija će se uglavnom koristiti na mobilnim uređajima. Od krajnjih korisnika, preko konobara i dostavljača koji isporučuju naručene artikle. Na osobu toga, gotovo je neminovna izrada mobilnih aplikacija u budućnosti.

Automatizacija i poboljšanje pojedinih procesa. Ovo je domen unapređenja svakog informacionog sistema koji gotovo nikada ne zastareva. Dokle god ima prostora za unapređenjem, sistem se čini "živim". Pa tako i prikazani sistem ima prostora za mnogobrojna unapređenje u vidu poboljša performansi i funkcionalnosti.

Pored toga jedna od vrlo verovatnih unapređenja je implementacija online platnog prometa, kako bi prilikom poručivanja hrane korisnik mogao i da plati samu porudžbinu online.

# LITERATURA

[1] 8 Best Software Development Methodologies, dostupno na: <https://www.uptech.team/blog/software-development-methodologies>, (datum pristupa 19.12.2022.)

[2] Waterfall model <https://www.techtarget.com/searchsoftwarequality/definition/waterfall-model>, (datum pristupa 19.12.2022.)

[3] SDLC Waterfall Model in Management Information System <https://www.includehelp.com/MIS/sdlc-waterfall-model-in-management-information-system.aspx> , (datum pristupa 19.12.2022.)

[4] SDLC - V-Model <https://www.tutorialspoint.com/sdlc/sdlc_v_model.htm> , (datum pristupa 19.12.2022.)

[5] V-Model <https://en.wikipedia.org/wiki/V-Model>, (datum pristupa 19.12.2022.)

[6] What is Iterative and Incremental Development <https://www.wrike.com/project-management-guide/faq/what-is-iterative-incremental-development/>, (datum pristupa 19.12.2022.)

[7] Why We’re Moving to Agile Software Development <https://orionadvisortech.com/blog/why-were-moving-to-agile-software-development/?print=print>, (datum pristupa 19.12.2022.)

[8] What is 'Spiral Model' <https://economictimes.indiatimes.com/definition/spiral-model>, (datum pristupa 19.12.2022.)

[9] Spiral Model: Software Development For Critical Projects <https://blog.airbrake.io/blog/sdlc/spiral-model>, (datum pristupa 19.12.2022.)

[10] Agile Methodology: What is Agile Model in Software Testing? <https://www.guru99.com/agile-scrum-extreme-testing.html>, (datum pristupa 19.12.2022.)

[11] What is Agile Methodology? <https://www.devteam.space/blog/what-is-an-agile-methodology/> , (datum pristupa 19.12.2022.)

[12] What Is Scrum Methodology? &

Scrum Project Management <https://www.digite.com/agile/scrum-methodology/>, (datum pristupa 19.12.2022.)

[13] What is Scrum? <https://www.digite.com/agile/scrum-methodology>, (datum pristupa 19.12.2022.)

[14] Larman's UML Process <https://www.objectsbydesign.com/books/larman_process.html> (datum pristupa 19.12.2022.)

[15] What is requirement analysis <https://sr.myservername.com/what-is-requirement-analysis> (datum pristupa 19.12.2022.)

[16] Introduction of Software Design process | Set 2 <https://www.geeksforgeeks.org/introduction-of-software-design-process-set-2/> (datum pristupa 19.12.2022.)

[17] Software design and implementation <https://www.collegenote.net/curriculum/software-engineering-csit/52/302/> (datum pristupa 19.12.2022.)

[18] What is Software Testing? Definition <https://www.guru99.com/software-testing-introduction-importance.html>

(datum pristupa 19.12.2022.)

[19] What is PHP? The PHP Programming Language Meaning Explained <https://www.freecodecamp.org/news/what-is-php-the-php-programming-language-meaning-explained/> (datum pristupa 19.12.2022.)

[20] A brief guide through Laravel <https://mdevelopers.com/blog/a-brief-guide-through-laravel> (datum pristupa 19.12.2022.)

[21] MariaDB <https://www.techtarget.com/searchdatamanagement/definition/MariaDB> (datum pristupa 19.12.2022.)

[22] What is JavaScript? <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript> (datum pristupa 19.12.2022.)

[23] Bootstrap <https://www.techtarget.com/whatis/definition/bootstrap> (datum pristupa 19.12.2022.)

[24] Laravel Livewire <https://laravel-livewire.com/> (datum pristupa 19.12.2022.)

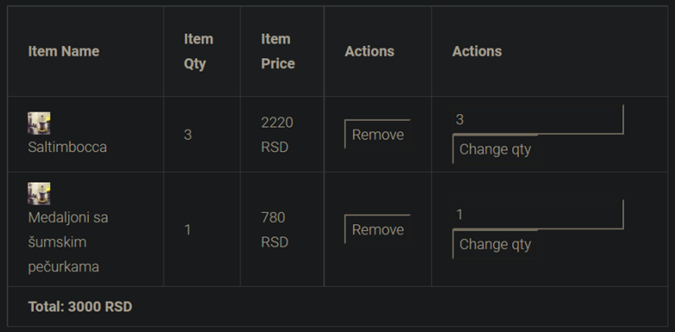
[25] model-view-controller (MVC) <https://www.techtarget.com/whatis/definition/model-view-controller-MVC> (datum pristupa 19.12.2022.)

[26] The Model View Controller Pattern – MVC Architecture and Frameworks Explained <https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/> (datum pristupa 19.12.2022.)

[27] Introduction to Repository Design Pattern <https://cubettech.com/resources/blog/introduction-to-repository-design-pattern/> (datum pristupa 19.12.2022.)

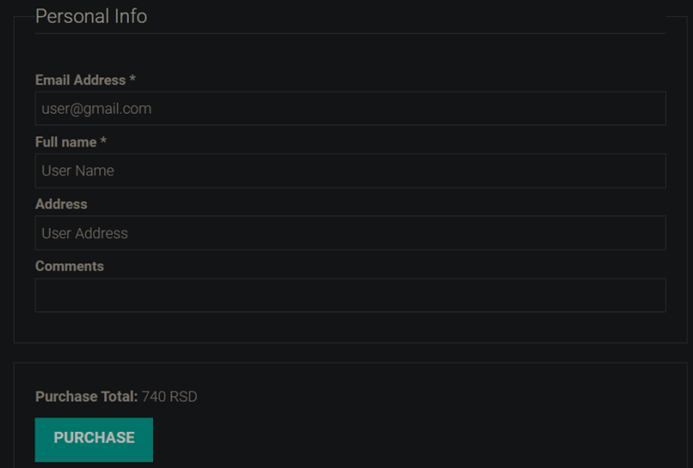
# PRILOZI

**Prilog A1**

****

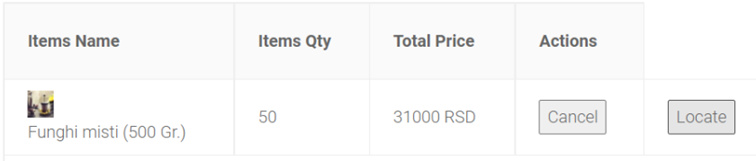
Opis: Prikazana je početna strana korpe sa prethodno dodatim proizvodima. Korisnik ovim ima uvid u sve artikle koje je dodao u korpu, bilo namerno ili slučajno, što se može ispraviti klikom na dugme Remove (obrisati). Ukoliko je pak uneta pogrešna količina ili iz nekog razloga istu želimo da promenimo, to je moguće učiniti prostim unosom nove količine i klikom na dugme Change qty (Promeniti količinu). Takođe prikazane su i osnovne informacije o pojedinačnim cenama artikala u korpi, količinama i ukupnoj ceni porudžbine, ukoliko se ona kreira.

**Prilog A2**

****

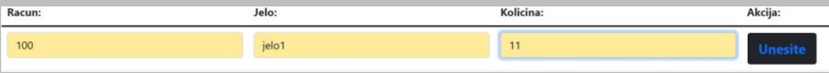
Opis: prikazan je deo strane ispod artikala koji se nalaze u korpi. Na ovom delu korisnik ima uvid u svoje informacije koje je uneo prilikom registracije kao i informaciju o ukupnoj vrednosti porudžbine, ukoliko ista bude kreirana. Korisnik ima mogućnost da unese druge informacije umesto postojećih, ukoliko je to iz bilo kog razloga potrebno. Kreiranje porudžbine, onda kada je korisnik siguran da su sve informacije onakve kakve treba da budu, se vrši prostim klikom na dugme Purchase (Kupovina).

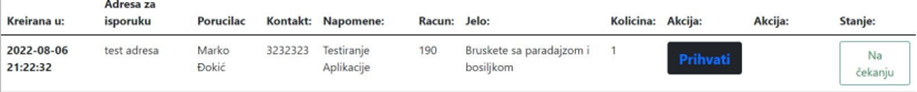
**Prilog A3**

****

Opis: Ukoliko se korisnik iz bilo kog razloga predomisli i više ne želi svoju porudžbinu, ima mogućnost da je otkaže u bilo kom trenutku, prostim klikom na dugme Cancel (Otkaži). Pored toga, korisnik ima uvid u informacije kao što su artikli koji čine tu porudžbinu, ukupna količina artikala i ukupan iznos za plaćanje iste. Takođe korisnik može i locirati svoju porudžbinu od onog trenutna kada dostavljač započne isporuku iste, klikom na dugme Locate (lociraj).

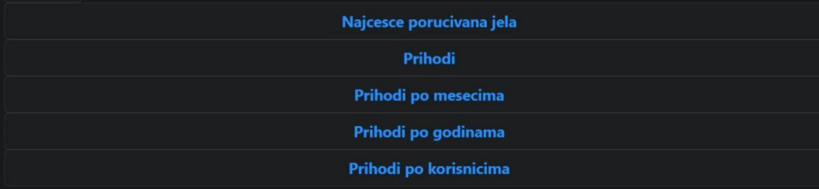
**Prilog A4**





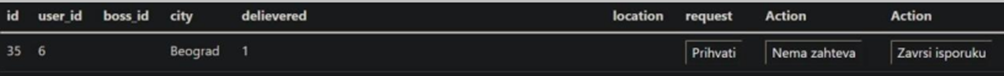
Opis: Konobari koji su zaduženi za evidentiranje novokreiranih porudžbina i pripremu istih, prostim panelom mogu obavljati svoja zaduženja. Onog trenutka kada se porudžbina prihvati dostavljači će dobiti evidenciju da je u toku priprema porudžbine koju bi trebalo uskoro isporučiti na traženu adresu. Konobari imaju uvid u sve neophodne informacije vezane za samu porudžbinu, kao što su datum i vreme kreiranja iste, adresa za isporuku, Ime, broj telefona i napomene poručioca. Pored toga tu su i naravno neophodne informacije o iznosu računa, naziv jela koje je potrebno pripremiti i količina jela za pripremu. Pored toga tu je i jako bitna informacija o tome da li je korisnik otkazao porudžbinu ili ne, putem zelenog dugmeta ,,Na čekanju“ ili crvenog dugmeta ,,Otkazano“, ukoliko je korisnik otkazao istu. Konobar pored svega ima i opciju ručnog unošenja porudžbine, koja može biti nezavisna od trenutnog menija, čime možemo proširivati isti i ispunjavati posebne želje naručioca.

**Prilog A5**



Opis: Uvid u statistiku je funkcionalnost vezana za menadžera. U svega par klikova i na vrlo jednostavan način menadžer može dobiti neophodne informacije za uspešno vođenje poslovanja, kao što su prihodi po mesecima, godinama, korisnicima. Takođe imamo uvid i u dnevne, nedeljne, ovomesečne i ovogodišnje prihode. Pored toga imamo uvid i u to koji korisnici su koliko puta otkazivali svoje porudžbine, što može biti vrlo bitna informacija o tome da li je potrebno nekog korisnika banovati iz sistema, što će biti obrađeno u budućim nadogradnjama sistema.

**Prilog A6**



Opis: Izgled panela namenjenom dostavljačima je prilično jednostavan i sličan onom koji je namenjen konobarima. Naime, dostavljač ima sve neophodne informacije za isporuku porudžbina, kao što su adresa i grad za isporuku iste, ima naručioca i datum kreiranja iste. Pored toga, jako bitne funkcionalnosti namenjene samo dostavljačkoj roli za prihvatanje porudžbine, završetak isporuke iste i slanje trenutne lokacije. Takođe, kao i konobari, dostavljači će imati evidenciju o tome da li je korisnik otkazao porudžbinu ili je ista i dalje aktuelna za isporuku.

## SPISAK SLIKA

Slika 1. Model vodopada

Slika 2. V-Model

Slika 3. Iterativni i inkrementalni razvoj

Slika 4. Spiralni model

Slika 5. Agilna metoda razvoja softvera

Slika 6. Skram razvoj

Slika 7. Model-View-Controller

Slika 8. Use Case diagram

Slika 9. Kreiranje korisničkog naloga

Slika 10. Kreiranje korisničkog naloga alternativni scenario

Slika 11. Prijavljivanje na nalog

Slika 12. Prijavljivanje na nalog alternativni scenario

Slika 13. Brisanje korisničkog naloga

Slika 14. Brisanje korisničkog naloga alternativni scenario

Slika 15. Menjanje šifre korisničkog naloga

Slika 16. Menjanje šifre korisničkog naloga alternativni scenario

Slika 17. Dodavanje artikla u korpu

Slika 18. Dodavanje artikla u korpu alternativni scenario

Slika 19. Brisanje artikla iz korpe

Slika 20. Brisanje artikla iz korpe alternativni scenario

Slika 21. Menjanje količine artikla u korpi

Slika 22. Menjanje količine artikla u korpi alternativni scenario

Slika 23. Kreiranje porudžbine

Slika 24. Kreiranje porudžbine alternativni scenario

Slika 25. Otkazivanje porudžbine

Slika 26. Otkazivanje porudžbine alternativni scenario

Slika 27. Prihvatanje porudžbine

Slika 28. Prihvatanje porudžbine alternativni scenario

Slika 29. Odbijanje porudžbine

Slika 30. Odbijanje porudžbine alternativni scenario

Slika 31. Ručno unošenje porudžbine

Slika 32. Ručno unošenje porudžbine alternativni scenario

Slika 33. Prikaz statistike

Slika 34. Prikaz statistike alternativni scenario

Slika 35. Najčešće poručivani proizvodi

Slika 36. Najčešće poručivani proizvodi alternativni scenario

Slika 37. Prihodi po mesecima

Slika 38. Prihodi po mesecima alternativni scenario

Slika 39. Prihodi po godinama

Slika 40. Prihodi po godinama alternativni scenario

Slika 41. Prihodi po korisnicima

Slika 42. Prihodi po korisnicima alternativni scenario

Slika 43. Dnevni nedeljni mesečni i godišnji prihod

Slika 44. Dnevni nedeljni mesečni i godišnji prihod alternativni scenario

Slika 45. Prihvati porudžbinu

Slika 46. Prihvati porudžbinu alternativni scenario

Slika 47. Prihvati porudžbinu

Slika 48. Prihvati porudžbinu alternativni scenario

Slika 49. Završi isporuku

Slika 50. Završi isporuku alternativni scenario

Slika 51. Lociraj porudžbinu

Slika 52. Lociraj porudžbinu alternativni scenario

Slika 53. Konceptualni model

Slika 54. Struktura sistema

Slika 55. Sekvencijalni I kolaboracioni dijagram createAccount

Slika 56. Sekvencijalni I kolaboracioni dijagram check

Slika 57. Sekvencijalni I kolaboracioni dijagram showProducts

Slika 58. Sekvencijalni I kolaboracioni dijagram addToCart

Slika 59. Sekvencijalni I kolaboracioni dijagram deleteFromCart

Slika 60. Sekvencijalni I kolaboracioni dijagram updateQty

Slika 61. Sekvencijalni I kolaboracioni dijagram createOrder

Slika 62. Sekvencijalni I kolaboracioni dijagram cancelOrder

Slika 63. Sekvencijalni I kolaboracioni dijagram acceptOrder

Slika 64. Sekvencijalni I kolaboracioni dijagram rejectOrder

Slika 65. Sekvencijalni I kolaboracioni dijagram showData

Slika 66. Sekvencijalni I kolaboracioni dijagram popularArticles

Slika 67. Sekvencijalni I kolaboracioni dijagram months

Slika 68. Sekvencijalni I kolaboracioni dijagram years

Slika 69. Sekvencijalni I kolaboracioni dijagram users

Slika 70. Sekvencijalni I kolaboracioni dijagram amounts

Slika 71. Sekvencijalni I kolaboracioni dijagram AcceptOrder

Slika 72. Sekvencijalni I kolaboracioni dijagram sendLocation

Slika 73. Sekvencijalni I kolaboracioni dijagram finishDelivery

Slika 74. Sekvencijalni I kolaboracioni dijagram locate

Slika 75. Users tabela

Slika 76. Cart tabela

Slika 77. Article tabela

Slika 78. Locations tabela

Slika 79. Restaurants tabela

Slika 80. OrderLocation tabela

Slika 81. Orders Tabela

Slika 82. Dijagram baze podataka

Slika 83. Registraciona forma

Slika 84. Pozdravna poruka

Slika 85. Postojeća mejl adresa

Slika 86. Nevalidni podaci

Slika 87. Forma za prijavljivanje

Slika 88. Pozdravna poruka

Slika 89. Poruka o grešci

Slika 90. Forma za promenu podataka i brisanje naloga

Slika 91. Poruka o izvršenoj operaciji brisanja

Slika 92. Forma za promenu podataka.

Slika 93. Poruka o uspešnom ažuriranju podataka

Slika 94. Poruka o pogrešno unetoj šifri

Slika 95. Lista artikala

Slika 96. Forma za dodavanje artikla u korpu

Slika 97. Poruka o uspešnom dodavanju artikla u korpu

Slika 98. Forma za prijavljivanje na nalog

Slika 99. Lista artikala u korpi

Slika 100. Poruka o uspešnom brisanju.

Slika 101. Forma sa listom artikala u korpi

Slika 102. Poruka o uspešno promenjenoj količini

Slika 103. Poruka o grešci

Slika 104. Forma sa ličnim podacima i ukupnom cenom artikala u korpi

Slika 105. Poruka o uspešno kreiranoj porudžbini

Slika 106. Prikazana forma ukoliko je korpa prazna

Slika 107. Forma sa kreiranom porudžbinom

Slika 108. Poruka o uspešno otkazanoj porudžbini

Slika 109. Forma prikazana korisniku ukoliko nema kreiranih porudžbina

Slika 110. Forma za prikaz I prihvatanje porudžbine

Slika 111. Obaveštenje o prihvatanju porudžbine

Slika 112. Alternativni scenario prilikom prihvatanja porudžbine

Slika 113. Forma za ručni unos porudžbine

Slika 114. Obaveštenje o uspešno kreiranoj porudžbini

Slika 115. Obaveštenje o nevalidnim podacima

Slika 116. Početna strana menadžera

Slika 117. Alternativni scenario prilikom prikaza podataka.

Slika 118. Forma za prikaz I prihvatanje porudžbine

Slika 119. Obaveštenje o prihvatanju porudžbine.

Slika 120. Alternativni scenario prilikom prikaza porudžbina za isporuku

Slika 121. Tabelarni prikaz porudžbina spremnih za isporuku

Slika 122. Forma za slanje trenutne lokacije

Slika 123. Obaveštenje o poslatoj lokaciji

Slika 124. Alternativni scenario prilikom prikaza porudžbina za isporuku

Slika 125. Tabelarni prikaz porudžbine

Slika 126. Alternativni scenario prilikom završetka isporuke

Slika 127. Tabelarni prikaz kreirane porudžbine

Slika 128. Obaveštenje o uspešno poslatom zahtevu za lociranje

Slika 129. Alternativni scenario prilikom prikaza porudžbine

## 9.2 Spisak upotrebljenih akronima

SRS (eng. software requirements specification)

MySQL (eng. My Structured Query Language)

ACID (eng. atomicity, consistency, isolation, durability)

JSON (eng. JavaScript Object Notation)

API-je (eng. application programming interface)

2D/3D (eng. two-dimensional/three-dimensional)

HTML (eng. HyperText Markup Language)

CSS (eng. Cascading Style Sheets)

SEO-u (eng. search engine optimization)

AJAX (eng. Asynchronous JavaScript And XML)

DOM (eng. Document Object Model)

GUI (eng. Graphical User Interface)

MVC (eng. Model-View-Controller)

Ugovori (eng. contracts)