|  |  |  |
| --- | --- | --- |
|  | UNIVERZITET „UNION“  **RAČUNARSKI FAKULTET**  Knez Mihailova 6/VI  11000 BEOGRAD | Broj: |
|  |
| Datum: |
| 31.8.2025 |

|  |
| --- |
| **OSNOVNE STRUKOVNE STUDIJE**  **Informacione tehnologije**  **DIPLOMSKI RAD**  **Servis za pretplatu na dostavu hrane**  **– Od njive do stola**  *Kandidat*: **Marko Teofanović**  *Broj indeksa: IT 46/2022*  *Mentor:* **Dr Dragoljub M. Pilipović**    **Beograd, 2025.** |

# Apstrakt

“Od njive do stola” je servis koji povezuje domaće proizvođače hrane sa potrošačima u Beogradu. Direktno, transparentno i bez posrednika. Cilj projekta je da se unapredi domaća poljoprivreda kroz digitalizaciju prodaje, praćenje proizvodnje i promociju zdrave lokalne hrane. Promoviše se i zdravija ishrana pošto je većina hrane u velikim gradovima problematična. Pojačivači ukusa, neusklađeni odnos sastojaka, prskano povrće i sorte koje nemaju visok nivo hranjljivih sastojaka su postali standard u gradovima. Digitalizacija logistike i prodaje može direktno da spoji proivođače sa potrošačima. Korisnici mogu odabrati paket pretplate, a proizvođači dobijaju zagarantovanu prodaju za razliku od odlazka na pijacu.

Servis “Od njive do stola” je projektovan uz pomoć upotrebe dijagrama po UML standardu i osnovnih skica, a aplikacija je implementirana koristeći “PHP” programski jezik uz “Laravel 10” razvojni okvir.

Servis je prilagođen da podrži poslovanje isključivo u Beogradu, ali dalja vizija i cillj su kreiranje platforme koja se može prilagoditi i primeniti u svakoj državi kao jedan univerzalni šablon.

# Sadržaj

[Apstrakt 2](#_Toc22303)

[Sadržaj 3](#_Toc14734)

[Uvod 4](#_Toc23858)

[Definisanje problema koji se rešava 5](#_Toc7518)

[Rešenje problema 6](#_Toc30655)

[Korišćene tehnologije i tehnike 8](#_Toc864)

[UML dijagrami 8](#_Toc3552)

[Dijagram slučajeva upotrebe 9](#_Toc11554)

[Dijagram aktivnosti 11](#_Toc32330)

[Dijagram sekvenci 14](#_Toc31868)

[Dijagram stanja 18](#_Toc29080)

[Literatura 24](#_Toc13592)

# Uvod

U ovom radu obrađuje se proces planiranja, dizajniranja i kreiranja veb aplikacije „Od njive do stola“. Aplikacija je razvijena u PHP programskom jeziku uz pomoć “Laravel 10” okvira(framework-a). Projekat objedinjuje konceptualne i tehničke aspekte izrade softverskog rešenja. Od definisanja funkcionalnih zahteva na osnovu problema tj. zahteva naručioca softvera i primitivnog nacrta skice dizajna, do modelovanja dijagrama po UML standardu i kreiranja aplikacije. Aplikacija je dizajnirana da obrađuje potrebe poslovanja isključivo u Beogradu jer se tu nalazi ciljna grupa za ovakav tip usluge i jedino ovakav model može da opravda ekonomsku isplativost za predviđene uslove. Međutim dalji korak i vizija je kreiranje platforme koja može da se prilagodi bilo kom regionu.

# Definisanje problema koji se rešava

Cilj servisa je da se olakša nabavka svežeg voća i povrća u gradu. U modernom dobu čovek ima sve manje i manje slobodnog vremena. Ciljna grupa su ljudi koji žive u gradovima i nemaju lak pristup kvalitetnim i svežim namirnicama. Mnogi građani nemaju brz pristup pijaci. Odlazak i dolazak iz pijace mnogima oduzima dragoceno vreme. Boravak na pijaci mnogima može biti stresan, nepristupačan, iritantan ili nepraktičan. Vikendom, kada uglavnom poljoprivrednici dodju i donesu svoje proizvode, su jako česte gužve, a mnogi ljudi jedino tada mogu da obave nabavku. Na pijacama postoji veliki procenat ljudi koji se bave preprodajom robe što direktno ide na štetu manjih proizvođača. Cene robe nisu standardizovane i to se zloupotrebljava kako bi se otkupna cena smanjila na najmanju moguću kako bi marža otkupljivača i velikih kompanija bila što veća.

Druga opcija za građane je nabavka proizvoda u lancima velikih marketa. Činjenica je da supermarketi imaju prskane proizvode i/ili proizvode koji dolaze iz inostranstva. Čak i kada se kupuju proizvodi na pijaci, kupac ne može da ima garanciju da je taj proizvod svež i da nije prskan. Proizvodi koji se uvoze iz inostranstva moraju da budu tretirani kako bi stigli u prihvatljivom stanju za prodaju. Što znači da hrana u supermarketima nije sveža i jeste prskana raznim preparatima. Takvi proizvodi mogu da prouzrukuju mnoge zdravstvene probleme jer: nemaju kvalitetna ili dovoljno vlakna, mogu imati neusklađen procenat šećera, ili je manja koncentracija hranljivih sastojaka zbog povišenog udeljka vode u masi, izazivanje alergija kod dece (ali i odraslih) zbog korišćenja pesticida insekticida i raznih otrova koji se mogu nelegalno koristiti u inostranstvu.

Broj poljoprivrednika u Srbiji se smanjuje i poljoprivreda kao zanimanje se sve manje i manje vrednuje. Mnogi poljoprivrednici i pored generalnih neizvesnosti i promenjljivih vezanih za to zanimanje imaju i problem jer zavise od otkupljivača i broja ljudi na pijacama. Iz godine u godinu sve manji broj ljudi posećuje pijace već narod nabavalja namirnice u supermarketima. Poljoprivrednik nema zagarantovanu prodaju i količina robe koju donosi na pijacu je isključivo spekulativna i nikada ne može biti tačna. Zbog toga se dešava da poljoprivrednici često moraju da odbacuju proizvode koji propadnu. Otkupljivači uglavnom nude otkupne cene koje nisu ekonomski isplative za poljoprivrednike. Cena za koju proizvođač prodaje robu i cena koju potrošači plaćaju uglavnom nije opravdana.

Neizvesnost vremenske prognoze i uvećan broj štetočina ili bolesti biljaka nije uvek lako predvideti i sprečiti. Međutim gore navedeni problemi mogu da se reše.

Voće i povrće nije isto kao što je bilo pre samo 30 godina. Kada je stanje tržišta trenutno takvo da su čak i biljke problematične i manjeg kvaliteta, stvara se statistika koja nam implicira da je glavni uzrok većine bolesti i smrti u modernom svetu loša i nekvalitetna ishrana.

# Rešenje problema

Servis “Od njive do stola” može da reši neke od gore navedenih problema i da pruži alternativan i praktičan način nabavke namirnica. “Od njive do stola” omogućava korisnicima da im voće i povrće, bude dostavljeno direktno na njihovu adresu. Hrana koja se dostavlja je maksimalno sveža jer se berba vrši dan pred dostavu. Korisnici imaju garanciju da će biti odabrani samo najbolje biljke bez ikakvih skrivenih rupa ili oštećenja. Korisnicima je zagarantovan kvalitet i transparentna je proizvodnja jer su proizvođači otvoreni za posetu imanja i proveru načina uzgoja. Korisnici samo trebaju da odvoje par minuta i da sačuvaju puno vremena koje bi inače potrošili u odlazku na pijacu i u prebiranju proizvoda, i ne moraju da troše keš već mogu i da plate jednom online i da su obezbeđeni za sledeći period koji su odabrali.

Poljoprivrednici imaju benefit zato što imaju unapred plan količine koju moraju da pripreme što minimizuje gubitke i imaju zagarantovanu prodaju. Ne moraju ceo vikend da izgube na tezgi gde im prodaja zavisi od više faktora. Poljoprivrednici koji bi sarađivali sa servisom “Od njive do stola” ne moraju da razlišljaju o poziciji tezge, broju prolaznika i izlaganju hladnom ili pretoplom vremenu, već samo trebaju da odvoje tačnu količinu i predaju transportnoj službi.

U ovoj verziji “Od njive do stola” dostavu hrane ne vrši direktno pružalac usluge “Od njive do stola” ili naručioc proizvoda. Dostava se angažuje izvan ovog sistema u vidu dugotrajne saradnje sa firmama koje se bave transportom i/ili taksi službama koje pružaju usluge ponavljajućih prevoza. “Od njive do stola” je specifično prilagođen za manje poslovanje, a ne za obradu masovnog broja narudžbenica i korisnika. Unapređenje servisa bi zahtevalo mnogo više vremena i resursa i usluga bi bila skuplja što nije trenutni cilj.

Implementirano rešenje omogućava registraciju korisnika, izbor i pretplatu na paket kao i administraciju svih podataka od strane administratora. Nakon registracije ili prijave, korisniku se prikazuju dostupni paketi sa opisima i cenama. Korisnik može da izabere paket i od prve subote nakon pretplate kreće dostava. Postoje četiri dostave po mesecu, a godišnja pretplata obuhvata 40 dostava jer zimi usluge nisu dostupne. Prilikom naručivanja korisnik popunjava podatke o dostavi. Sve tranzakcije se čuvaju uz pomoć tabela paket\_korisnikas i fakturas (imaju s na kraju jer je razvojnom okviru “Laravel 10” to konvencija imenovanja tabela zbog korišćenja svih mogućnosti koje pruža, više pod Korišćene tehnologije i tehnike). Ako korisnik želi da promeni neki podatak o sebi ili paketu to ne može da uradi direktno već mora da kontaktira administratora. Na ovakav način funkcioniše jer ne bi trebalo da korisnik može tek tako da promeni podatke pošto je onda moguće da se desi da destruktivni korisnik pravi problem davaocu usluge. Korisniku su dostupne kontakt informacije kao što je adresa elektronske pošte i broj telefona koji je dostupan u određenom radnom vremenu.

“Od njive do stola” pomaže i korisnicima i unapređuje poljoprivredu u Srbiji koja nažalost opada, a deo razloga su gore navedeni problemi gde su glavni problem veliki lanci marketa i način na koji posluju tj. ne sarađuju sa domaćim proizvođačima. Izmena ključnih podataka mora da bude u nadležnosti administratora. Administrator ima poseban nalog koji može da pristupi svim rutama na sajtu i može da izmeni preko veb aplikacije podatke u svakoj tabeli iz baze. Admin može da pregleda, keira, ažurira i obriše sve dok običan korisnik može samo da pregleda svoje porudžbine. Sve administratorske rute običnom korisniku samo daju opciju da se vrati na početnu stranicu. Predviđeno je da sistem podrži plaćanje i ostavljen je prostor u grafičkom prikazu na mestu gde bi trebao da bude procesor plaćanja ali da bi se izbeglo promovisanje bilo koje kompanije implementacija sistema procesora trenutno nije implementirana.

# Korišćene tehnologije i tehnike

Na osnovu zahteva naručioca softvera su kreirani UML dijagrami i skice kako bi se što bolje razumeo ideja i koncept.

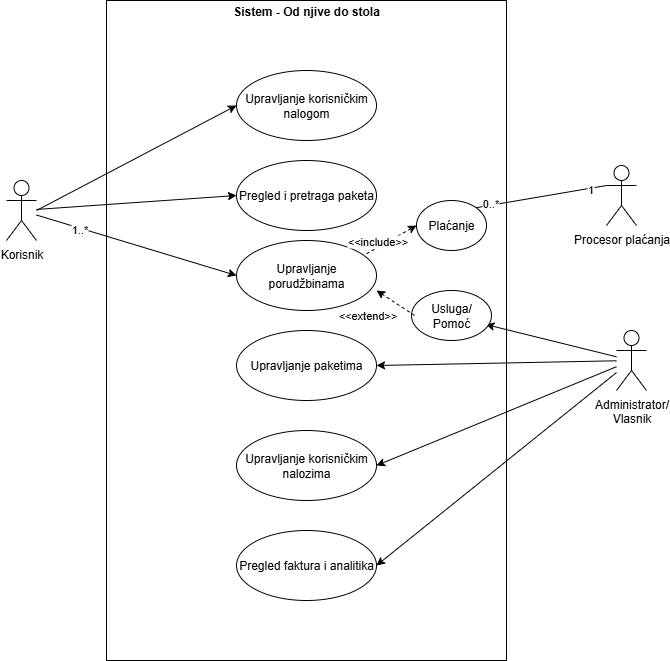
## UML dijagrami

Ujedinjeni jezik za modelovanje je standardizovani grafički jezik za specifikaciju, vizualizaciju i dokumentovanje artefakata[1] softverskih sistema. UML predstavlja skup dijagrama i notacija koji omogućavaju apstraktno modelovanje strukturnog aspekta informacionog sistema i način ponašanja informacionog sistema nezavisno od tehnologije ili načina na koji se sistem implementira. Omogućuje jasan, precizan i formalan način da se prikažu zahtevi, arhitektura, tok podataka i interakcija objekata u sistemu što sve olakšava potvrdu dizajna granica i mogućnosti sistema, smanjuje rizik grešaka i omogućava prenosivost znanja među učesnicima koji imaju dodir sa projektom bilo da su trenutni ili neko ko će se u budućnosti naknadno upoznati sa sistemom. [2][3][4]

Zbog široke primene UML standarda i podrške za širok broj tipova dijagrama kao što su dijagram slučaja upotrebe, klasnog dijagrama, dijagrama entiteta i relacija, dijagram aktivnosti, dijagram stanja i slični, ovaj standard je odabran za predstavljanje servisa “Od njive do stola”. U projektu su prisutni svaki od gore navedenih dijagrama. Predstavljeni su svi detalji kao što je tok i obrada podataka, model baze, ponašanje objekata tokom životnog ciklusa pretplate. Takodje korišćenjem UML standarda olakšava buduće proširenje projekta što je još jedan razlog za odabir UML-a kao glavni jezik za tehničku dokumentaciju.

### Dijagram slučajeva upotrebe

Dijagram slučajeva upotrebe nam služi kao opšti prikaz identifikacije i predstavljanja funkcionalnih zahteva sistema iz perspektive njegovih aktera (korisnik, administrator, eksterni sistemi). Glavni cilj ovog dijagrama je da što prostije uputi čitaoca šta sistem radi i koje usluge postoje bez ulaza u tehničke detalje kako se te usluge ostvaruju. Dijagram slučajeva upotrebe se kreira u prvoj fazi projekta radi brze komunikacije sa članovima tima i svih zainteresovanih strana radi provere da li su opšti zahtevi ispravno shvaćeni i izkomunicirani. Takođe se koristi kao osnova za dalju dekompoziciju tokom projektovanja sistema i crtanja drugih dijagrama. Za svaki bitan slučaj se izrađuje dijagram aktivnosti koji dalje opisuje grananja unutar dijagrama slučaja upotrebe. Sekvencni dijagram za redosled interakcija aktera sa objektima sistema. Dijagram entiteta i relacija je lakše napraviti zbog brzog prepoznavanja entiteta i veza



Slika 1. Prikaz dijagrama slučajeva upotrebe

1. Akteri:
   1. Korisnik je akter koji komunicira sa sistemom kroz tri glavna slučaja. Registracija/prijava, pregled paketa, naručivanje paketa.
   2. Administrator/Vlasnik je akter koji upravlja sadržajem sistema, a to se odnosi na upravljanje paketima, korisničkim nalozima, paketima korisnika i fakturama.
2. Slučajevi upotrebe:
   1. Upravljanje korisničkim nalogom

Akteri: Korisnik

Opis: Korisnik se može registrovati na platformu ili prijaviti sa već postojećim nalogom

* 1. Pregled i pretraga paketa

Akteri: Korisnik

Opis: Korisnik može da pregleda dostupne pakete pretplate i da pročita detalje paketa

* 1. Upravljanje porudžbinom

Akteri: Korisnik, Administrator/Vlasnik, Procesor plaćanja

Opis: Korisnik unosi podatke i kreira porudžbinu. Administrator može naknadno da promeni podatke na zahtev korisnika.

Alternativni tok: U slučaju da korisnik želi da promeni podatak mora da kontaktira korisničku podršku pozivom na broj telefona ili elektronskom poštom. Ovakav pristup osigurava usaglašenost podataka sa obe strane.

Drugi alternativni tok je ako procesor plaćanja odbije uplatu, gde korisnik mora ponovo da unesi podatke.

* 1. Upravljanje paketima

Akteri: Administrator/Vlasnik

Opis: Administrator/Vlasnik može da unese novi paket, ažurira postojeće pakete ili da obriše paket

* 1. Upravljanje korisničkim nalozima

Akteri: Administrator/Vlasnik

Opis: Administrator može da vidi sve registrovane naloge i ispravi greške u slučaju da je korisnik uneo pogrešan podatak ili želi da napravi promenu

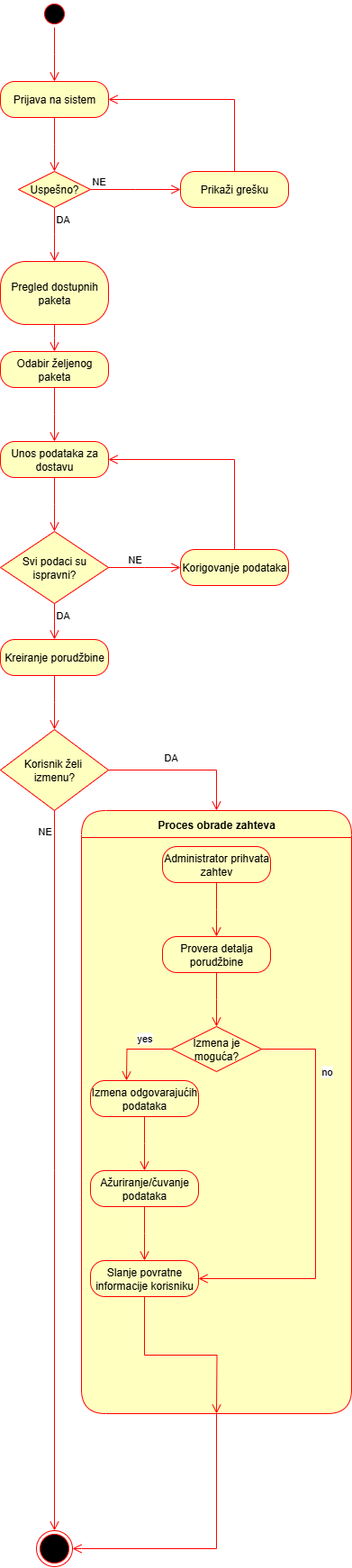
* 1. Pregled faktura i analitika

Akteri: Administrator/Vlasnik

Opis: Administrator/Vlasnik može da vidi podatke o svim narudžbenicama i pregleda detalje kao što su, da li je uplata uspešno izvršena, ili da vidi analitiku prodaje

### Dijagram aktivnosti

Dijagram aktivnosti prikazuje tok izvršavanja poslovnog procesa ili funkcionalnog scenarija kroz niz povezanih aktivnosti, odluka i paralelnih tokova. Cilj dijagrama je da jasno prikaže korake i tok radnje, tačke grananja i odgovornosti pri obavljanju određenog postupka u sistemu. Uz pomoć dijagrama aktivnosti možemo videti tačan opis poslovnog procesa koji se odvija u sistemu pre implementacije. Takođe je lakše uvideti konkurentne ili asinhrone korake i daje uvid u to šta treba da se paralelizuje i na taj način optimizuje radi boljeg iskustva. Aktivnosti se lako mogu porediti sa komponentama u Laravel okviru i može da pomogne lakšoj vizualizaciji koda. Pored svega navedenog ova vrsta dijagrama može da pomogne i pri kreiranju test scenarija sa alternativnim tokovima i mogućim greškama pri radu. Dijagram aktivnosti je dekompozicija iz dijagrama slučajeva upotrebe. Aktivnosti koje se dešavaju iniciraju entiteti koji se nalaze u klasnom dijagramu.



Slika 2. Prikaz dijagrama aktivnosti

Priloženi dijagram aktivnosti prikazuje korisničke i administratorske aktivnosti u procesu upravljanja narudžbinom uključujući i alternativne tokove.

Glavni tok:

1. Korisnik se prijavljuje na sistem
   1. Korisnik se uspešno prijavio
   2. Korisniku se prikazuje da je došlo do greške tokom prijave i vraća se na korak 1
2. Korisnik vidi prikaz dostupnih paketa pretplate
3. Korisnik unosi podatke o dostavi
   1. Korisnik je uneo ispravne podatke
   2. Korisnik ponovo unosi podatke
4. Sistem kreira pretplatu

Alternativni tok:

1. Korisnik je pogrešio ili mora da promeni neki podatak koji je već unešen u sistem i kontaktira korisničku podršku
2. Administrator je primio zahtev
3. Administrator pregleda unešene podatke i zahtev

3.1) Administrator je odobrio ažuriranje podataka i obrađuje zahtev

3.2) Podaci su uspešno sačuvani u sistem

3.3) Administrator šalje povratnu informaciju korisniku o izmeni podataka

3.4) Administrator odbija zahtev

1. Tačka završetka

### Dijagram sekvenci

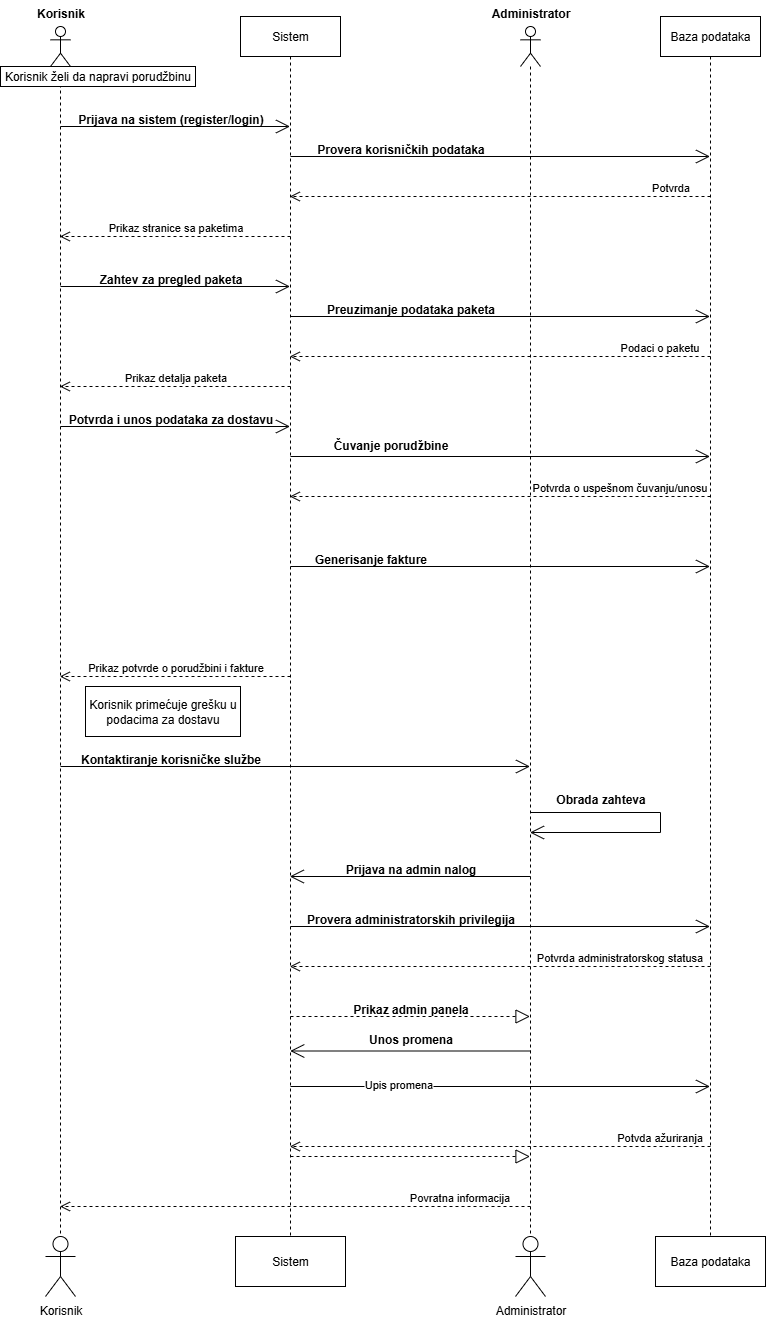
Dijagram sekvenci detaljno prikazuje komunikaciju/interakciju između objekata/komponenti sistema kroz vreme. Glavna svrha dijagrama sekvenci je dokumentovanje dinamičkog ponašanja sistema, redosleda i sadržaja interakcija potrebnih za realizaciju nekog scenarija (naprimer dekompozicija nekog slučaja upotrebe iz dijagrama slučaja upotrebe). Dijagram sekvenci je dijagram sa najviše detalja i najpreciznije opisuje funkcionisanje sistema. Prikazuje tačno u kom trenutku šta se poziva što je ključno za razumevanje protokola toka informacija, zavisnosti i tačaka gde se mogu uvesti asinhrone funkcije ili tranzakcije.

Sekvencni dijagram utiče i na razvoj grafičkog interfejsa jer može da ukaže precizno redosled prikaza elemenata ili stranica na ekranu pri razvoju softverskog rešenja za informacioni sistem. Kada je dokumentovana vremenska relacija programer može bolje da isplanira zadatke i upotrebu “redova čekanja” koja pruža Laravel PHP radni okvir.

Takođe pri kreiranju tranzakcija se uz pomoć dijagrama sekvenci može odrediti početak i kraj tranzakcije, naprimer kreiranje pretplate i fakture u jednoj ACID tranzakciji (atomičnost, konzistentnost, izolacija, trajnost). Dijagram sekvenci može i da pomogne pri povezivanju vizuelnog dizajna sa imenima metoda i objektima u kodu.

Dijagram sekvenci jasno i detaljno opisuje alternativne tokove i moguće greške poput neuspešne validacije ili greške vezane za bazu.

U kombinaciji sa dijagramom slučaja upotrebe i dijagramom aktivnosti obrađeni su svi detalji procesa šta sistem radi korak po korak sa tačnim redosledom.



Slika 3. Prikaz dijagrama sekvenci

Učesnici (lifelines) koji učestvuju u scenariju:

1. Korisnik
2. Sistem/Veb aplikacija
3. Administrator
4. Baza podataka

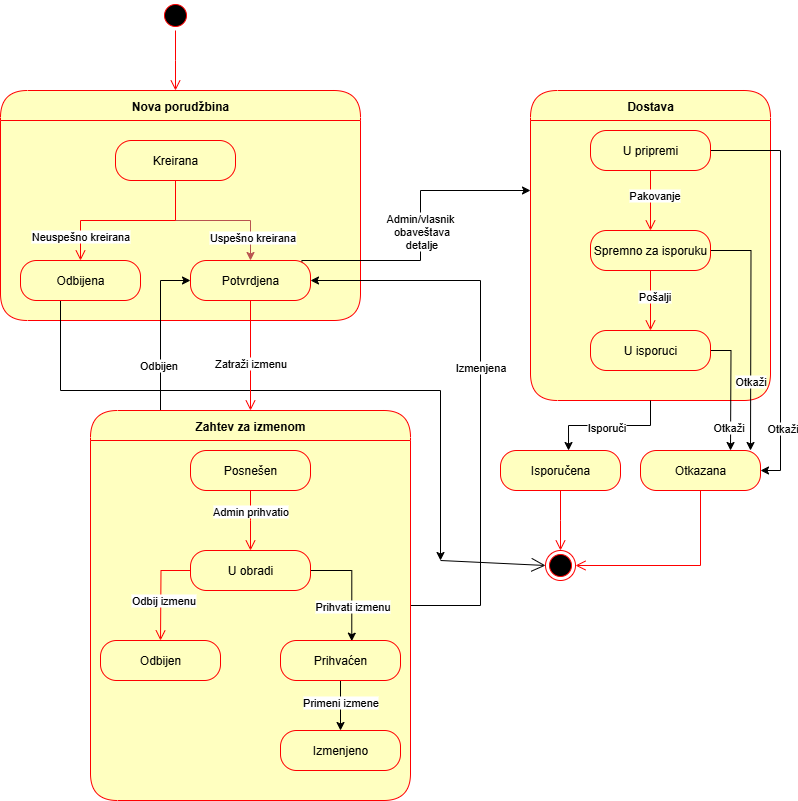
Poruke/procesi:

1. Aktivacioni blok : Korisnik želi da napravi narudžbenicu
2. Korisnik -> Veb aplikacija : Korisnik se registruje ili loguje na sistem
3. Veb aplikacija-> Baza podataka : Provera korisničkih podataka
4. Baza podataka -> Veb aplikacija : Povratna informacija statusa
5. Veb aplikacija -> Korisnik : Prikaz početne stranice
6. Korisnik -> Veb aplikacija : Zahtev za pregled detalja paketa
7. Veb aplikacija -> Baza podataka : Upit za preuzimanje podataka o paketu
8. Baza podataka -> Veb aplikacija : Podaci o traženom paketu
9. Veb aplikacija -> Korisnik : Prikaz stranice popunjene sa podacima o paketu koji je korisnik odabrao
10. Korisnik -> Veb aplikacija : Korisnik unosi podatke o dostavi i plaćanju u formu i potvrđuje (HTTP POST )
11. Veb aplikacija -> Baza podataka : Upit za unos stavki paketa korisnika
12. Veb aplikacija -> Baza podataka : Upit za unos stavki fakture
13. Baza podataka -> Veb aplikacija : Potvrda o uspešnom unosu
14. Veb aplikacija -> Korisnik : Prikaz detalja narudžbenice i fakture
15. Aktivacioni blok : Korisnik je primetio da je uneo neispravan podatak ili se predomislio oko zahteva
16. Korisnik -> Administrator : Korisnik uspostavlja kontakt sa korisničkom službom
17. Administrator : Obrada i analiza zahteva
18. Administrator -> Veb aplikacija : Administrator se prijavljuje na admin nalog
19. Administrator -> Veb aplikacija : Administrator otvara stranicu sa panelom za promenu podataka
20. Veb aplikacija : Provera administratorskih privilegija
21. Veb aplikacija -> Baza podataka : Upit za dostavljanje odgovarajućih podataka
22. Veb aplikacija -> Administrator : Generisanje i prikaz stranice
23. Administrator -> Veb aplikacija : Unos promena
24. Veb aplikacija -> Baza podataka : Upit za upis promena u bazi (HTTP POST )
25. Baza podataka -> Veb aplikacija -> Administrator : Potvrda ažuriranja podataka
26. Administrator -> Korisnik : Administrator obaveštava korisnika o uspešnoj promeni podataka

### Dijagram stanja

Dijagram stanja prikazuje način na koji se pojedinačni entitet menja kroz niz jasno definisanih stanja usled događaja, uslova ili u toku akcija. Glavni cilj dijagrama stanja je dokumentovanje životnog ciklusa entiteta, koja stanja entitet može da ima, koji događaji uzrokuju prelaz između stanja i koje se akcije izvršavaju prilikom ulaska ili izlaska iz stanja. Za razliku od sekvencnog dijagrama koji obrađuje samo poruke i interakcije između objekata ili komponenti, dijagram stanja fokusira se isključivo na ponašanje jednog entiteta.

Pomoću dijagrama stanja se zapisuju dozvoljena stanja i tranzicije što može biti korisno pri uklanjanju nejasnoća pri dizajnu i implementaciji poslovne logike u aplikaciji. Složeniji uslovi za ulaz u stanje mogu biti pregledniji i vidljiviji. Takođe je dijagram sekvenci sa definisanim tranzicijama koristan pri kreiranju jediničnih testova i testa integracije što čini sistem pouzdanijim. Dijagram stanja nam takođe služi za prepoznavanje i planiranje automatizovanih radnji jer ukazuje na to kakve promene trebaju da se dogode i pokrenu događaj ili za postavljanje zakazanih zadataka (cron jobs).



Slika 4. Prikaz dijagrama stanja

Na slici 4. je prikazan niz stanja kroz koja prolazi polazi pretplata korisnika u sistemu “Od njive do stola”.

Nova porudžbina/pretplata:

Ulazna akcija:

1. Unos korisnika na formi

Lista stanja:

1. Biva kreirana
2. Odbijena/neaktivna
3. Potvđena/aktivna

Lista tranzicija:

1. Neuspešno kreirana - nalazi se između kreirana i odbijena/neaktivna
2. Uspešno kreirana - nalazi se između kreirana i potvrđena/prihvaćena
3. Kraj - ako je pretplata neaktivna ili odbijena
4. Zatraži izmenu - korisnik inicira proces promene podataka iako je potvrđena pretplata
5. Admin/Vlasnik obaveštava dostavu - Aministrator ili vlasnik sprovodi informacije o dostavi poslovnim partnerima koji vrše dostavu

Zahtev za izmenu:

Ulazna akcija:

1. Stavka broj četiri pod nova narudžbina/pretplata

Lista stanja:

1. Podnešen - Zahtev je uspešno pristigao
2. U obradi - Zahtev je pristigao do Administratora/vlasnika i u toku analize je
3. Odbijen - Zahtev biva ugašen ukoliko je nemoguće ispuniti isti, narudžbenica se potpuno otkazuje ili se nastavlja u prvobitnom stanju
4. Prihvaćen - Zahtev biva prihvaćen, upisuje se novo stanje i natavlja proces
5. Ispunjen - Zahtev je uspešno ispunjen

Lista tranzicija:

1. Administrator/Vlasnik je prihvatio - Administrator ili vlasnik je prihvatio podnešen zahtev i čita elekronsku poštu ili služa zahtev preko poziva
2. Odbij izmenu - Administrator ili vlasnik je odbio izmenu zbog tehničkih problema ili nevalidnih zahteva
3. Prihvati zahtev za izmenu - Administrator ili vlasnik je prihvatio izmenu i menja detalje narudžbenice
4. Primeni izmene narudžbenice - Administrator ili vlasnik unosi i primenjuje izmene nad narudžbenicom

Izlazna akcija:

Povratna sprega na stanje 3 iz kreiranja nove narudžbenice/pretplate

Dostava:

Ulazna akcija:

1. Informacije administratora/vlasnika - administrator ili vlasnik obaveštava poslovne partnere koji se bave dostavom

Lista stanja:

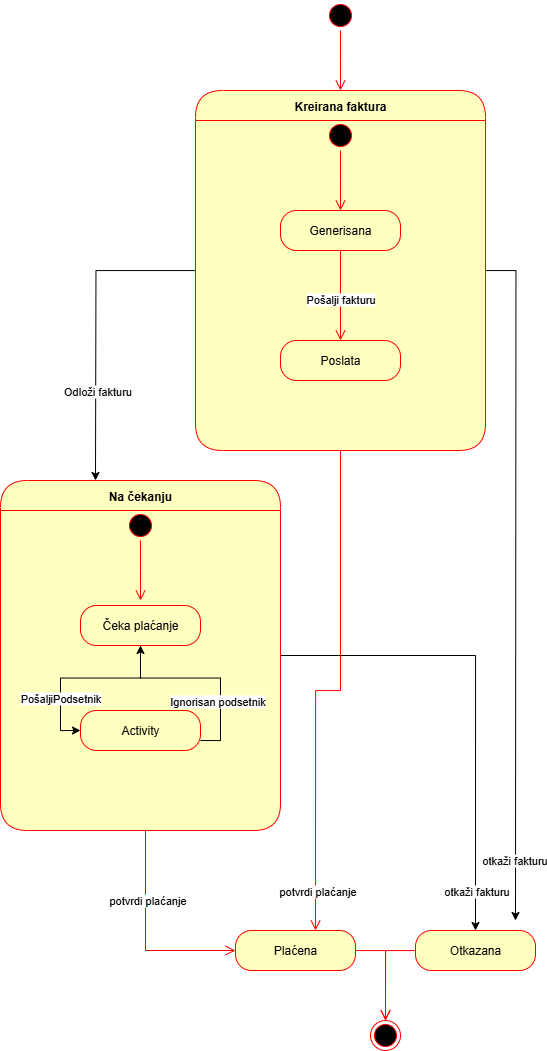
1. U pripremi - poljoprivrednici spremaju i odvajaju porudžbine
2. Spremno za isporuku - stanje koje je uglavnom aktivno od ponoći do jutra kada gajba čeka dostavljače
3. U isporuci - porudžbina je na putu do Beograda ili u Beogradu na redu da bude isporučena
4. Isporučena - paket je uspešno dostavljen i prihvaćen od strane klijenta
5. Otkazana - paket je neuspešno dostavljen usled neplaniranih okolnosti

Lista tranzicija:

1. Pakovanje - odnosi se i na odvajanje zasebnih gajbi i na utovar u prevozno vozilo
2. Pošalji - poslovni partneri koji se bave dostavom su u pokretu do adresa korisnika
3. Isporuči - dostavljači su u toku predaje robe
4. Otkaži - neplanirani događaji mogu da dovedu do potpunog otkaza dostave i moguć je otkaz dostave u svakom od prva tri stanja

Izlazne akcije:

Krajnji status dostave nas dovodi do kraja dijagrama



Slika 5. Prikaz dijagrama stanja

Na slici 5. je prikazan dijagram stanja fakture koja se kreira pri odabiru paketa.

Kreirana faktura:

Ulazna akcija:

1. Korisnik je potvrdio podatke unešene u formi

Lista stanja:

1. Generisana - faktura je generisana uporedo sa upisom podataka o paketu korisnika u bazu podataka
2. Poslata korisniku - faktura je poslata korisniku kako bi odmah imao povratnu informaciju i znao da se uspešno pretplatio na uslugu

Lista tranzicija:

1. Pošalji fakturu - sistem generiše prikaz fakture sa svim bitnim detaljima korisniku

Izlazne akcije:

1. Uplata je uspešno izvšena i potvrđena
2. Odloži fakturu - ako je došlo do problema sa procesuiranjem uplate ili je korisnik pokrenuo proceduru povratka novca faktura se odlaže na čekanje
3. Otkaži fakturu - faktura je otkazana od strane korisnika ili od strane administratora/vlasnika zbog nevalidnih podataka

Na čekanju:

Ulazna akcija:

Odložena faktura

Lista stanja:

1. Čeka plaćanje - korisnik idalje ima opciju da se pretplati na paket do dana dostave u slučaju da procesor plaćanja nije uspešno realizovao uplatu
2. Obavešten korisnik - Vlasnik ili administrator ili sistem šalje poruku elektronskom poštom ili poziva korisnika radi podsećanja ili napomene trenutnog stanja fakture
3. Plaćena - faktura je upotpunjena
4. Otkazana - faktura dobija oznaku neizvršene uplate

Lista tranzicija:

1. Pošalji podsetnik
2. Korisnik ignoriše podsetnik

Izlazne akcjie:

1. Potvrdi plaćanje - korisnik je rešio problem i uspešno izvršio uplatu
2. Otkaži fakturu

## Dijagram entiteta i veza

Dijagram entiteta i veza predstavlja model podataka koji vizuelno predstavlja entitete ( tabele ), njihove atribute i veze ( relacije ) između entiteta. U projektu “Od njive do stola” dijagram entiteta i veza je centralni artefakt za dizajn baze podataka. Uz pomoć dijagrama entiteta i veza se kreira šablon upisa podataka na koji nadograđujemo sve ključne mogućnosti sistema. Definiše strukturu podataka koja omogućava funkcionalnosti registracije korisnika, pretplate korisnika na tip paketa, fakturisanje i istoriku paketa korisnika. Dijagram entiteta i veza modeluje podatke i način čuvanja tj. Strukturu podataka, veze i kardinalnosti 1:1, 1:N, N:M uz pomoćne tabele, primarne i strane ključeve koji obezbeđuju referencijalni integritet, ograničenja podataka ili obavezu unosa podataka, čuvanje podataka bez nekontrolisane redudanse i obezbeđuje zaštitu od anomalija. [5]

Dijagram entiteta i veza jasno prikazuje uključene entitete koje aplikacija obrađuje.



Slika 6. Prikaz dijagrama entiteta i veza

### Entiteti

Napomena: tabele se završavaju sa karakterom “s”. Implementacija ovakvog naziva tabela je obavezna zbog omogućavanja Laravel radnog okvira da prepozna tabele i da bi se iskoristio pun potencijal Laravel okvira. U engleskom jeziku se slovo “s” dodaje na kraj reči kako bi se označila množina reči.

#### Korisnik (tabela users)

Korisnici sistema su predstavljeni tabelom users sa njenim atributima.

# Literatura

|  |  |
| --- | --- |
| [1] | *Autor, Naziv knjige, godina izdanja....* |
| [1] | *Dr Marko N. Mladenović, Računarski fakultet, skup prezentacija iz predmeta “Uvod u softversko inženjerstvo”, 2024* |
| [2] | *The Object Management Group®, What is UML?,*  *<https://www.omg.org/uml/what-is-uml.htm,> 2025* |
| [3] | *R. Booch, J. Rumbaugh, I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999.* |
| [4] | *Object Management Group, UML® 2.0 Superstructure Specification, OMG, 2005.* |
| [5] | *Peter Pin-Shan Chen, The Entity-Relationship Model-Toward a Unified View of Data, 1976* |