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 hranljivih 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

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 promenljivih 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.

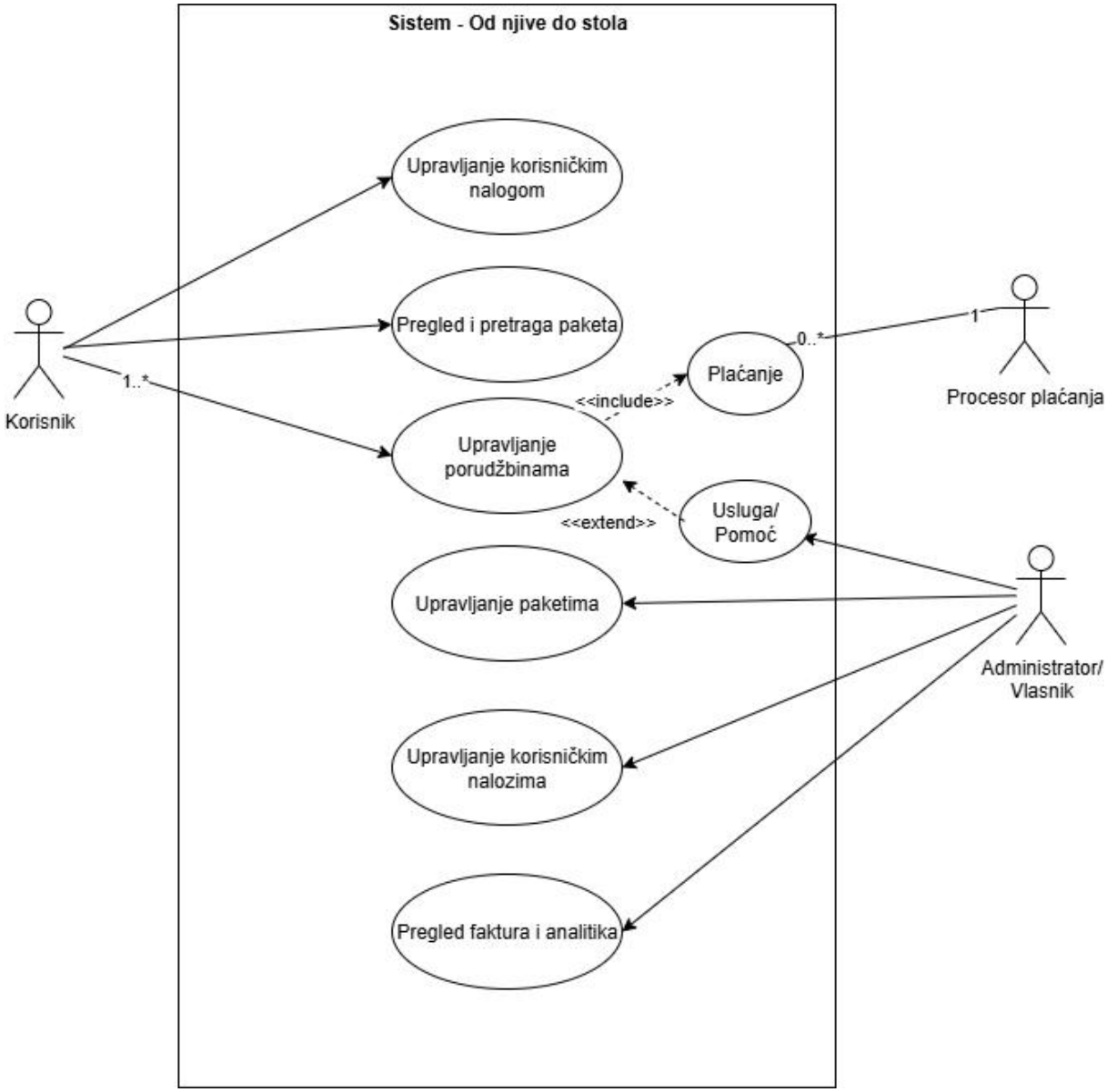
UML dijagrami

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:

1.1) Korisnik je akter koji komunicira sa sistemom kroz tri glavna slučaja. Registracija/prijava, pregled paketa, naručivanje paketa.

1.2) Administrator/Vlasnik je akter koji upravlja sadržajem sistema, a to se odnosi na upravljanje paketima, korisničkim nalozima, paketima korisnika i fakturama.

2) Slučajevi upotrebe:

2.1) Upravljanje korisničkim nalogom

Akteri: Korisnik

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

2.2) Pregled i pretraga paketa

Akteri: Korisnik

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

2.3) 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.

2.4) Upravljanje paketima

Akteri: Administrator/Vlasnik

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

2.5) 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

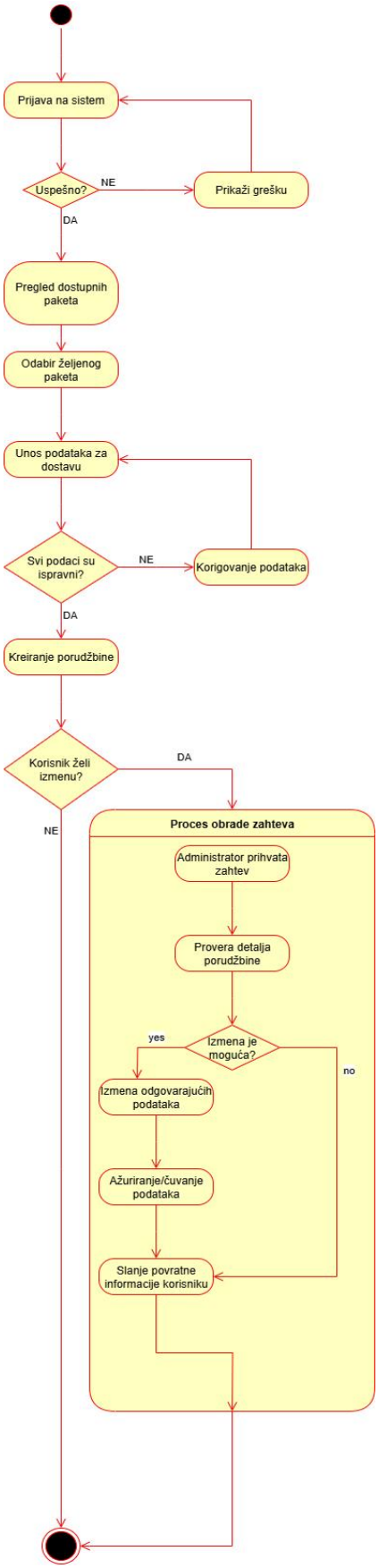
2.6) Pregled faktura i analitika

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

1.1) Korisnik se uspešno prijavio

1.2) 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

3.1) Korisnik je uneo ispravne podatke

3.2) 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

4) Tačka završetka

Dijagram sekvenci

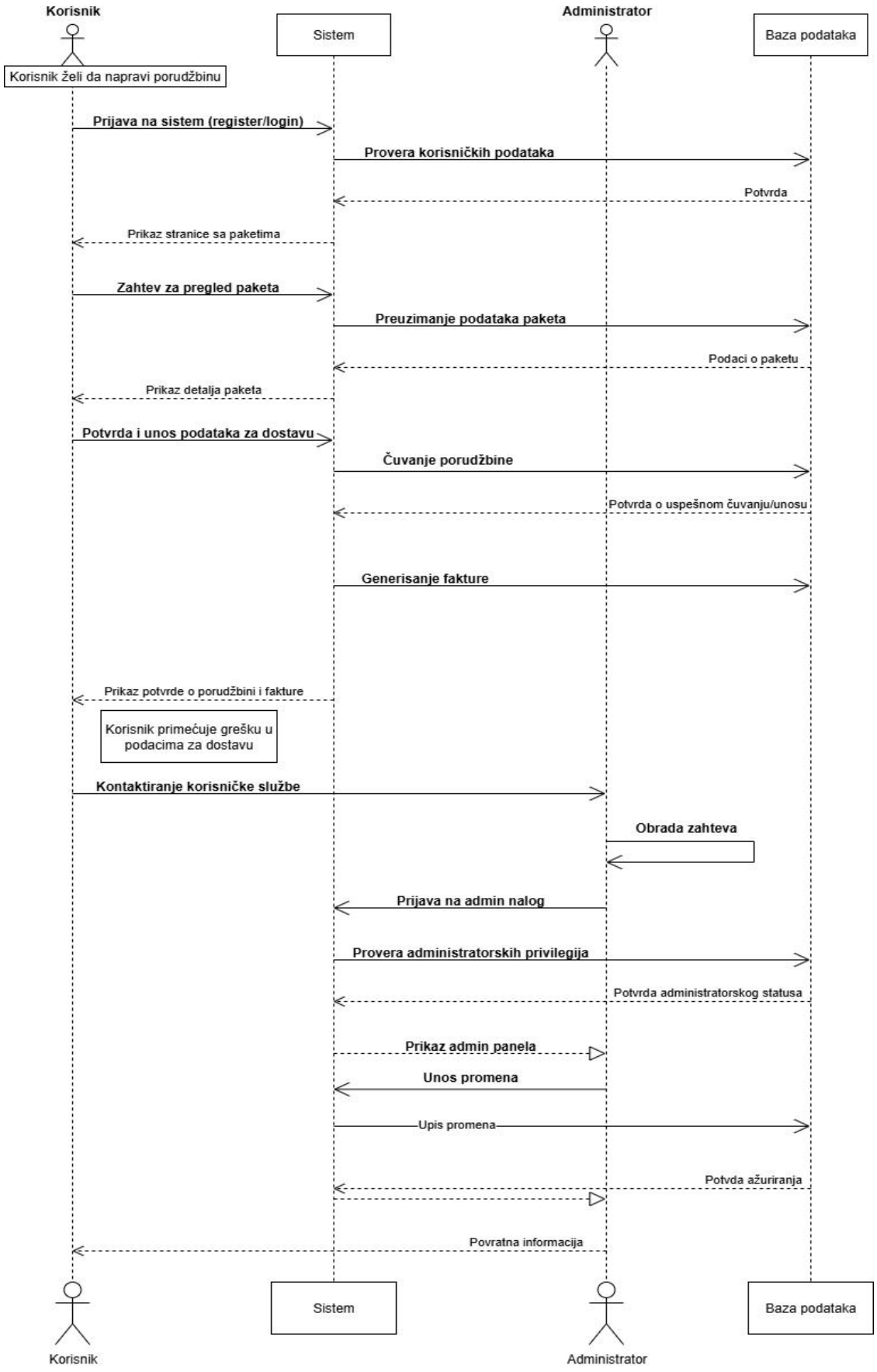
Dijagram sekvenci detaljno prikazuje komunikaciju/interakciju između objekata/komponenti sistema kroz vreme. Glavna svrha dijagrama sekvenci je dokumentovanje dinamičkog ponašanja sistema, redosleda i sadržaja interakcija potrebnih za realizaciju nekog scenarija (naprimer dekompozicija nekog slučaja 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)

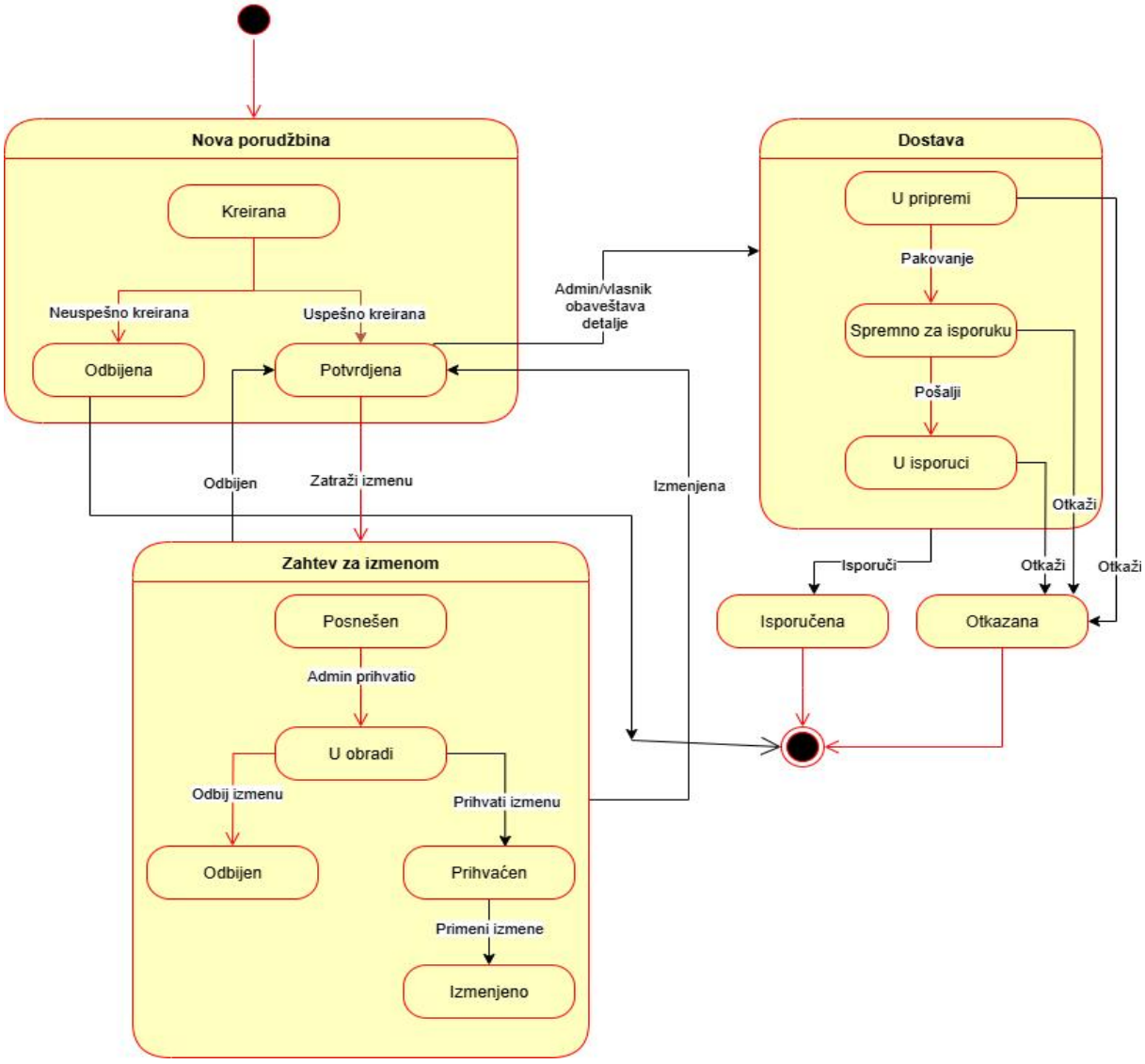
25) Baza podataka -> Veb aplikacija -> Administrator : Potvrda ažuriranja podataka

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

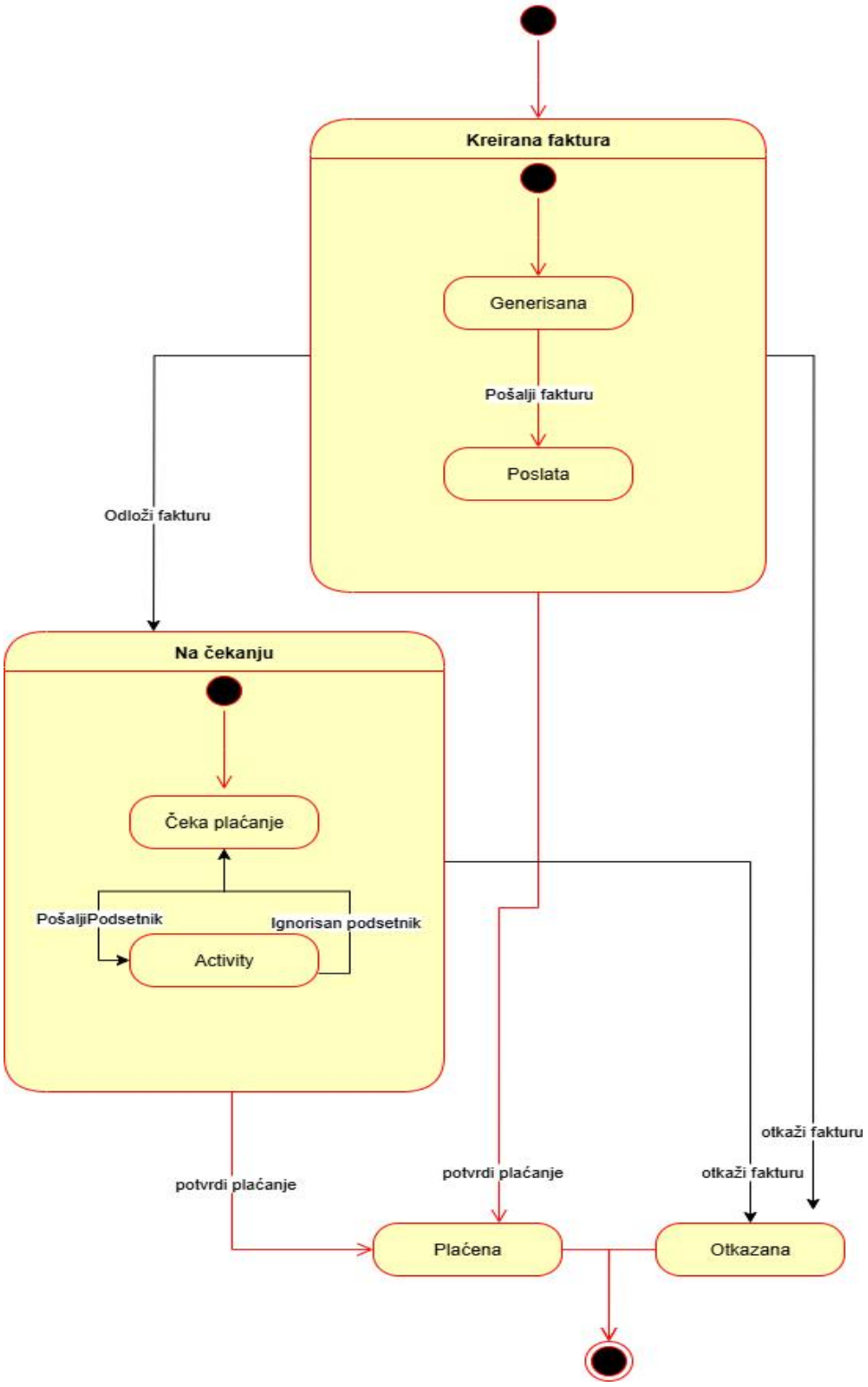
2) Pošalji - poslovni partneri koji se bave dostavom su u pokretu do adresa korisnika

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

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. [6]

Pri kreiranju tabela preko migracija svaka tabela podrazumevano ima atribute “created_at” i “updated_at” koji služe za više stvari. Utiče na integritet podataka, pomaže pri analitici podataka i može da bude korisno i u oblasti bezbednosti. Tačno vreme kreiranja porudžbine je izuzetno bitan podatak i tačno vreme ažuriranja podatka mogu pomoći pri rešavanju sporova, proveru reklamacija ili analize greški pri poslovnom procesu.

Olakšavaju jednostavno pretraživanje, segmentaciju korisnika i pretplata po vremenu kada su nastale ili izmene koja je uglavnom momenat plaćanja u slučaju odloženog plaćanja. Ovi atributi su ključni za upravljanje pretplatama.

Analitika u vidu najprometnijeg perioda u godini je još jedna od prednosti koja može biti omogućena uz pomoć ovih atributa. Datumi dostave mogu da se izračunaju na osnovu ovih podataka. U slučaju anomalija ili incidenta pronalaženje bugova i generalna forenzika je olakšana pri pronalazku korisnika ili odgovornog procesa koji je doveo do stanja.

Ovi atributi imaju i bitnu ulogu u arhivama baze podataka.

Upotreba tipa nvarchar u sql bazi je praktično i bezbedno rešenje i za tekstualne atrbute a i za atribute koji mogu imati specijalne karaktere ili međunarodne znakove. Nvarchar obezbeđuje bilo koji unos od nestanka karaktera koji može da se desi pri korisničkom unosu. Nvarchar (255) obično pokriva sve tipične tekstualne upise poput imena, adrese elektronske pošte, adresa dostave itd.[7] Naravno za polja koja nikada ne mogu imati veći broj karaktera od nekog maksimuma se ne postavlja dužina od 255 što će biti napomenuto ispod.

Svaka tabela u sistemu ima primarni ključ “id”. U praksi se ovakav pristup pokazao kao najpraktičnije rešenje, a i u sistemu koji je baziran na Laravelu mnogo je lakše izkoristiti pun potencijal ovog radnog okvira uz upotrebu ovakvog pristupa dodeli atributa. Naravno svi primarni ključevi se automatski povećavaju za svaku novu stavku u sistemu i unikatni su za svaku stavku.

Korisnik (tabela users)

Korisnici sistema predstavljeni su tabelom users sa njenim atributima. Jedan korisnik može da napravi više paketa i može imati više paketa u isto vreme. Možda korisnik ima više adresa na kojima želi dostavu i želi različite pakete na tim adresama i to ovako postavljen sistem omogućava.

Atributi:

name nvarchar(255) NOT NULL - U sistemu “Od njive do stola” razdvajanje imena i prezimena nije neophnodno jer je najbitnija adresa elektronske pošte korisnika koja je obavezna za korišćenje usluga ovog servisa. Name u ovom sistemu više služi kao tekst za prikaz teksta na predviđenim mestima za poboljšano iskustvo korisnika.

email nvarchar(255) NOT NULL - email je centralno i najbitnije polje u ovoj tabeli jer služi i za logovanje na sistem. Dužina je 255 zato što lokalni deo adrese elektronske pošte (pre “@” karaktera) je ograničen na 64 karaktera, domen je ograničen na 63 karaktera i sa DNS domenom je ukupan broj karaktera 255 znakova. Zbog ovih ograničenja u SMTP implementacijama praktičan maksimum koji većina sistema podržava je 254 karaktera. [8] Naravno uzimajući u obzir navedene podatke ovaj atribut je obavezan u svakom zapisu.

email_verified_at datetime NULLABLE - Pri generisanju Laravel 10 projekta preko Vemto [9] alata ovo polje je podrazumevano. Implementacija servera za automatsko slanje poruka na elektronsku poštu korisnika nije urađena ali je kreirana osnova za imlementaciju.

password nvarchar(255) NOT NULL - lozinka korisnika se hešuje pri upisu i nije čitljiva. Standardna maksimalna dužina lozinki je 255 a nvarchar kao tip omogućava širok spektar prihvatljivih karaktera.

remember_token nvarchar(100) NULLABLE - pri logovanju korisnika na sistem on može da odabere da veb pregledač koji koristi registruje i zapamti za buduće posete bez potrebe za unosom adrese elektronske pošte i lozinke

Tip paketa (tabela tip_paketas)

Koncept usluge je da korisnik ne mora da ručno dodaje svaki artikal pojedinačno kao nanekim online piljarama već korisnik jako brzo može samo jednim klikom da odabere celu korpu/gajbicu u kojoj dolazi više povrća i voća koje je u tom periodu godine sveže i najzastupljenije, a ako je alergičan na neki proizvod koji je možda napoment u paketu može da to naglasi. Korisnik može da odabere više tipova paketa što ćemo videti u relaciji u tabeli paket_korisnikas.

Atributi:

cena_godisnje_pretplate decimal(10,2) NULLABLE

cena_mesečne_pretplate decimal(10,2) NULLABLE - Cena godišnje/mesečne pretplate služe za računanje cene pri kreiranju fakture. Decimalni broj sa deset cifara pre decimalne tačke i dve cifre posle decimalne tačke za preciznost je standardan tip koji se koristi pri čuvaju iznosa valuta. “decimal” čuva vrednosti tačno onako kako je specificirano za razliku od FLOAT, real ili nekih drugih tipova koje možemo videti ali Microsoft više ne podržava i može da se desi greška pri zaokruživanju pošto su to samo približni tipovi podataka. Skoro sve cene sa kojima se susrećemo su zaokružene na dve decimale osim na nalozima u bankama ili sličnim servisima gde je veća preciznost obavezna. Decimal (10, 2) pokriva i mnogo više od cena pretplata bilo da su mesečne ili godišnje ali je ostavljen veći prostor u slučaju da dinar dodatno opadne u vrednosti. Ovi atributi su NULLABLE zato što je moguće uneti paket koji je dostupan samo određeni deo godine i nema opciju godišnje pretplate pa se na ovaj način obrađuje ovakav slučaj.

opis nvarchar(MAX) NULLABLE - Atribut opis predstavlja unos koji vrši vlasnik sistema. Opis može biti mnogo duži od 255 karaktera i da bi se osigurao integritet podataka dozvoljen

je maksimalni mogući unos. Međutim ovo ne pravi problem sa memorijom jer će nvarchar rezervisati samo onoliko memorije koliko je potrebno za određeni string. Ovo polje je takođe NULLABLE zbog mogućnosti da je ime paketa dovoljan opis ili vlasnik mora naknadno da unese opis zbog nekog razloga ali je i takav slučaj obrađen.

naziv nvarchar(64) NOT NULL - Atribut naziv paketa je ograničen na 64 karaktera jer ne bi trebao da bude duži od toga zbog prikaza na ekranu korisnika. Naziv je obavezno polje i ne može se uneti tip paketa bez naziva.

Paket korisnika (tabela paket_korisnikas)

Tabela paket korisnika može da ima jednog korisnika i jedan tip paketa ali može postojati više paketa korisnika sa istim tipom i korisnik može da ima više odabranih paketa. Paket korisnika služi da se kreira faktura i čuva istorija podataka. Korisnik unosi vezuje adresu za paket korisnika a ne za tabelu korisnika zbog mogućnosti da mu je potreban drugi paket na drugoj adresi. (Primer scenarija, neki član porodice nije u mogućnosti da nabavlja namirnice zbog povrede, bolesti ili slično, a nema iskustva sa korišćenjem tehnologije, a porodica nije blizu da pomogne i onda korisnik može da uplati pretplatu na dostavu hrane na drugu adresu sa drugim paketom)

godisnja_pretplata bit DEFAULT 0 - Ovaj atribut služi kao buleanovo logičko polje, ako je čekirano korisnik je odabrao godišnju pretplatu i u obračunu se računa odgovarajuća cena, u suprotnom ako nije čekirano podrazumevano se uzima u obračun cena mesečne pretplate. Ovo polje ima podrazumevanu vrednost 0.

tip_paketa_id bigint NOT NULL - Ovo je strani kluč iz tabele tip_paketas. Povezuje dva entiteta i obavezno je polje jer ne može postojati paket korisnika koji nema tip odabranog paketa. Tip polja je bigint pošto je bigint standard u laravelu za sva polja koja imaju veze sa primarnim ključevima

user_id bigint NOT NULL - Ovaj atribut je strani kluč iz tabele users. Povezuje entitet paket korisnika sa entitetom korisnik. Ovaj atribut ima obavezan unos pošto ne može da postoji paket KORISNIKA bez korisnika koji je kreirao paket.

adresa nvarchar(255) NOT NULL - Atribut adresa se odnosti na adresu dostave koja može biti dugačka i ostavljeno je 255 karaktera za unos. Adresa je obavezno polje jer se usluga ne može izvršiti bez tačne adrese.

uputstvo_za_dostavu nvarchar(255) NULLABLE - Atribut uputstvo za dostavu je u mnogim situacijama jako bitno polje. Mnoge adrese nemaju očigledan fizički pristup i dostavljač mora da ima opis lokala i/ili okoline kako bi brzo i efikasno izvršio dužnost. Dužina je 255 zbog procene da ovo polje neće “preliti” ovu vrednost. Imajući u vidu da su mnogi brojevi kuća/zgrada jasno vidljivi i stanovi dobro obeleženi u većini slučajeva nema potrebe da se dodatno opisuje adresa pa je polje NULLABLE.

broj_telefona nvarchar(18) NULLABLE - Atribut broj telefona služi kako bi se korisnik kontaktirao u slučaju da postoji neka vest vezana da pretplatu koju korisnik treba hitno da čuje ili se koristi u toku dostave radi provere mogućnosti korisnika da preuzme paket ili prosto najavi da je dostava izvšena i da je narudžbina ispred vrata adrese. Dužina ovog polja je ograničena na 18 karaktera pošto broj telefona ne može biti duži od 18. Polje jeste nullable jer korisnik možda nema broj telefona koji bi podelio (ali adresa elektronske pošte je obavezna informacija o kontaktu)

poštanski_broj nvarchar(20) NOT NULL - Poštanski broj je bitan zbog precizne adrese. Ograničen je na 20 karaktera zato što poštanski broj ne može biti duži od dvadeset.

Faktura (tabela fakturas)

Red u tabeli faktura se kreira u isto vreme kada korisnik potvrdom na formi unese podatke i kreira red u tabeli paket_korisnikas. Tabela faktura se kreira uz pomoć stranog ključa koji je vezan za tabelu paket korisnika. Polje cena se računa na osnovu polja godišnja pretplata u tabeli paket_korisnikas i na osnovu podatak o ceni u tabeli tip_paketas. Tabela faktura bi trebalo da je povezana sa procesorom plaćanja uz povratnu informaciju od procesora plaćanja ali u ovom sistemu nije implementiran procesor plaćanja.

Atributi:

paket_korisnika_id bigint NOT NULL - Strani ključ koji nam govori od kog paketa korisnika je faktura, a pošto je id korisnika obavezan u tabeli paket korisnika onda je tabela fakturas uvek vezana za korisnika. Ovo polje je obavezno

Cena decimal (10, 2) NOT NULL - Ovo polje se izračunava uz pomoć tabele paket korisnika gde se nalazi odabran parametar godišnje ili mesečne pretplate i tip paketa gde se nalazi sama cena.

tekst nvarchar (255) NULLABLE - Polje tekst sadrži tekst koji se nalazi na fakturi i bolje je opisuje

placeno bit NULLABLE - Polje plaćeno ima tip buleanove logike i ako je vresnot 1 to znači da je korisnik uspešno izvršio uplatu bilo da je preko procesora plaćanja ili nekom drugom metodom. Zbog toga polje može imati NULL vrednost u slučaju da korisnik nije uspeo uspešno da izvši uplatu pri kreiranju narudžbine ali zato ima mogućnost da naknadno uplati.

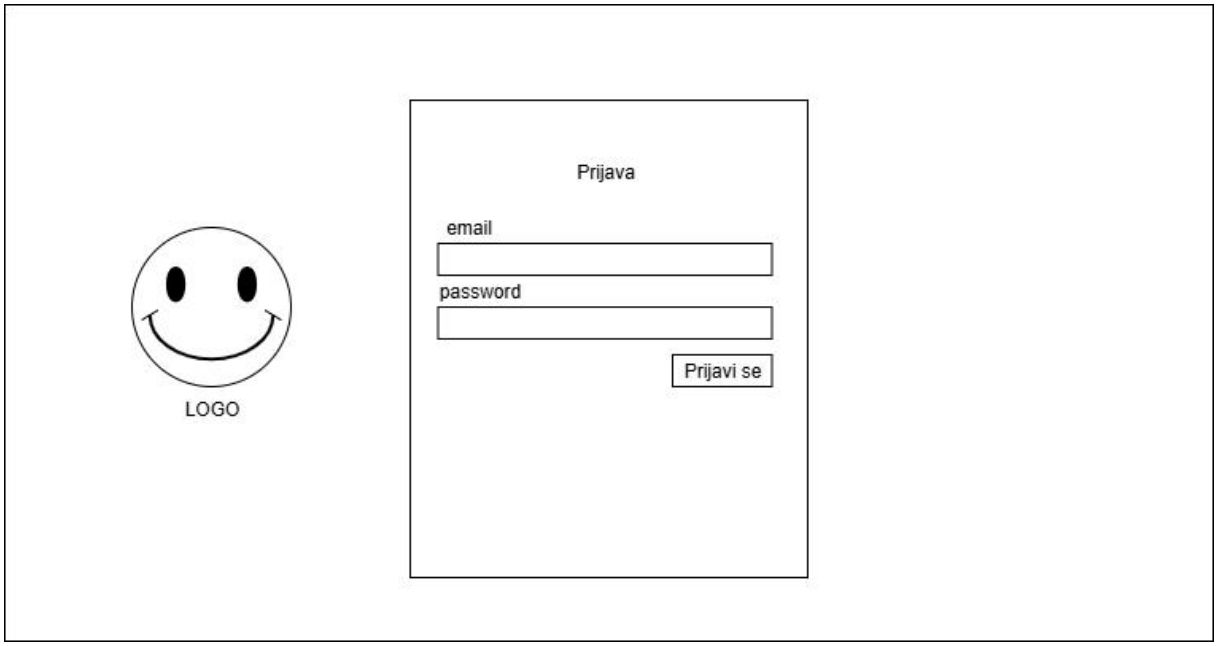
Brisanje redova u sistemu je jako retko i nema puno podataka ali u slučaju brisanja podataka reda u tabeli paket korisnika briše se i odgovarajuća faktura.

Projektovanje korisničkog interfejsa

Filozofija pristupa dizajnu korisničkog interfejsa je funkcionalni minimalizam. Ako nešto ne mora da bude prisutno onda ne postoji. Cilj usluge je da korisnici uštede vreme na odlazak i prebiranje namirnica, isto tako i proces naručivanja treba da bude što brži i jednostavniji.

Niskoverna maketa skeleta dizajna

Sledeći crteži skiza su niskoverna ideja koja služi naručiocu softvera kako bi stvorio generalnu ideju konačnog izgleda proizvoda. Dimenzije nisu precizne i ne mora da obradi svaki element i sadržaj koji će biti prisutan.

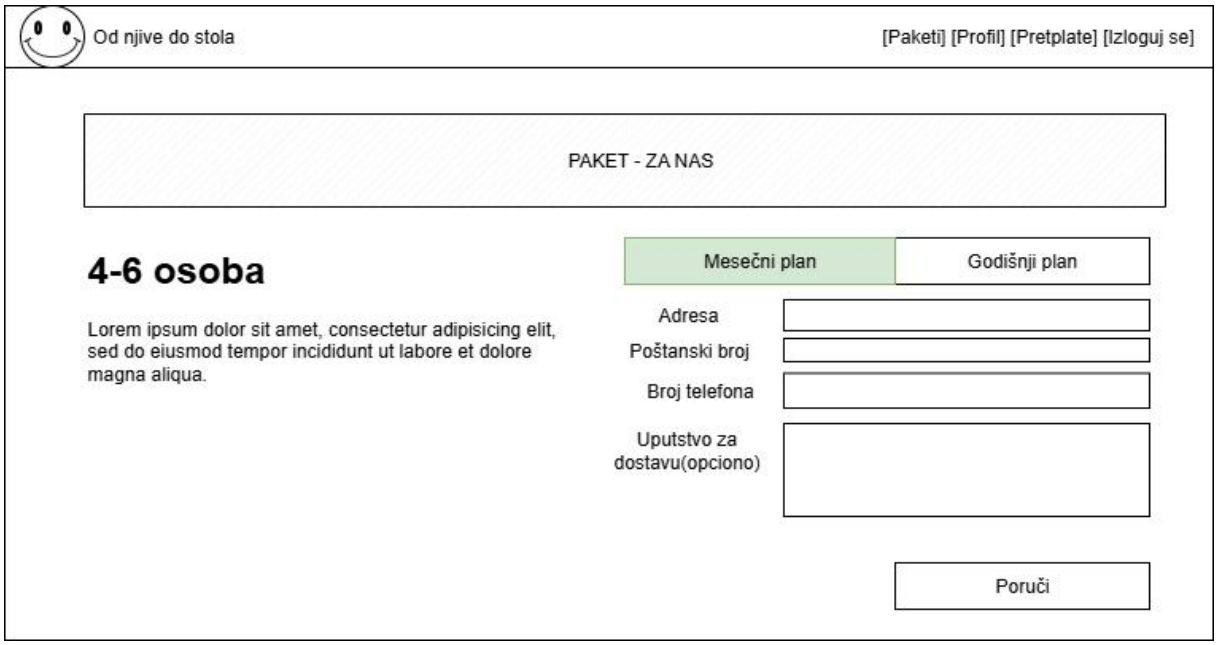


Slika 7. Prikaz skice dizajna stranice za logovanje na sistem

U toku dizajniranja niskoverne makete dizajna login stranice veb aplikacije “Od njive do stola” nije dodato dugme za registrovanje korisnika. Ivice elemenata su oštre što takođe nije bilo implementirano u krajnjoj verziji dizajna. Stranica ima minimalan broj elemenata što znači da korisnik ima jasnu ideju šta treba da uradi.

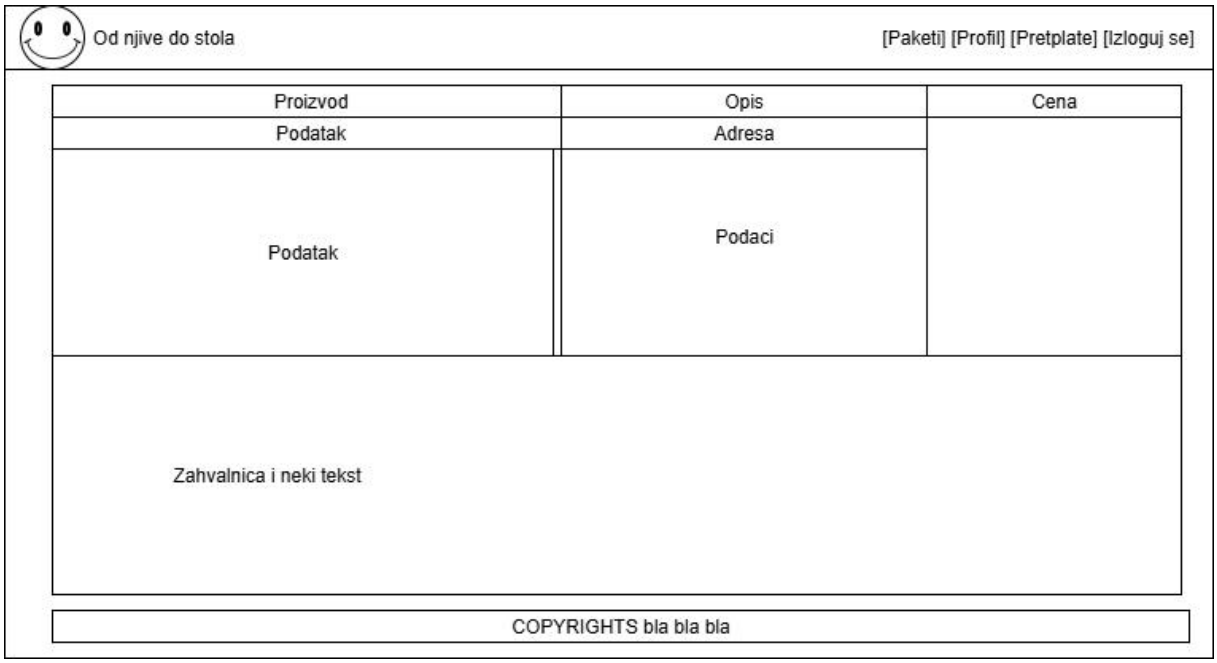
Slika 8. Prikaz početne/glavne stranice

Ne postoji posebna stranica sa listom proizvoda, ne postoji razlog da korisnik mora da traži gde je lista proizvoda. Korisnik je ušao na sajt da bi pregledao ponudu i video cilj servisa. Navigacionog menija i liste proizvoda prikazana je samo kratka poruka dobrodošlice i opis usluge. Od dugmadi koja vrše redirekciju postoji dugme Paketi koja je zamišljeno da vodi na trenutnu stranicu, dugme liste pretplata koje je korisnik obavio, dugme za log off i svaki element u listi ima svoju odgovarajući url koji sprovodi korisnika do stranice sa detaljima paketa i formom za naručivanje i plaćanje.



Slika 9. Prikaz dizajna stanice za prikaz detalja paketa

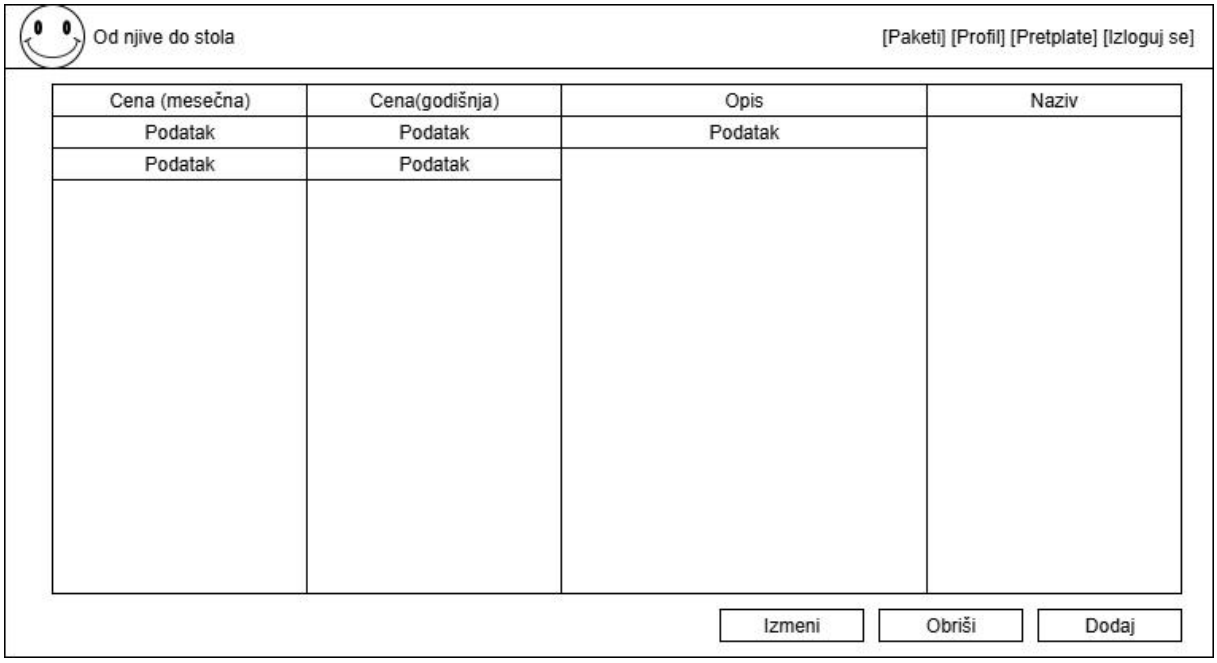
Na stranici se nalaze svi bitni detalji o paketu i odmah je dostupna forma. U skici nisu prikazani elementi vezani za obradu isplate. Cilj je implementacija procesora plaćanja na istoj stranici, međutim to nije moguće sa svim procesorima plaćanja.



Slika 10. Prikaz dizajna veb prikaza potvrđene pretplate i detalja fakture

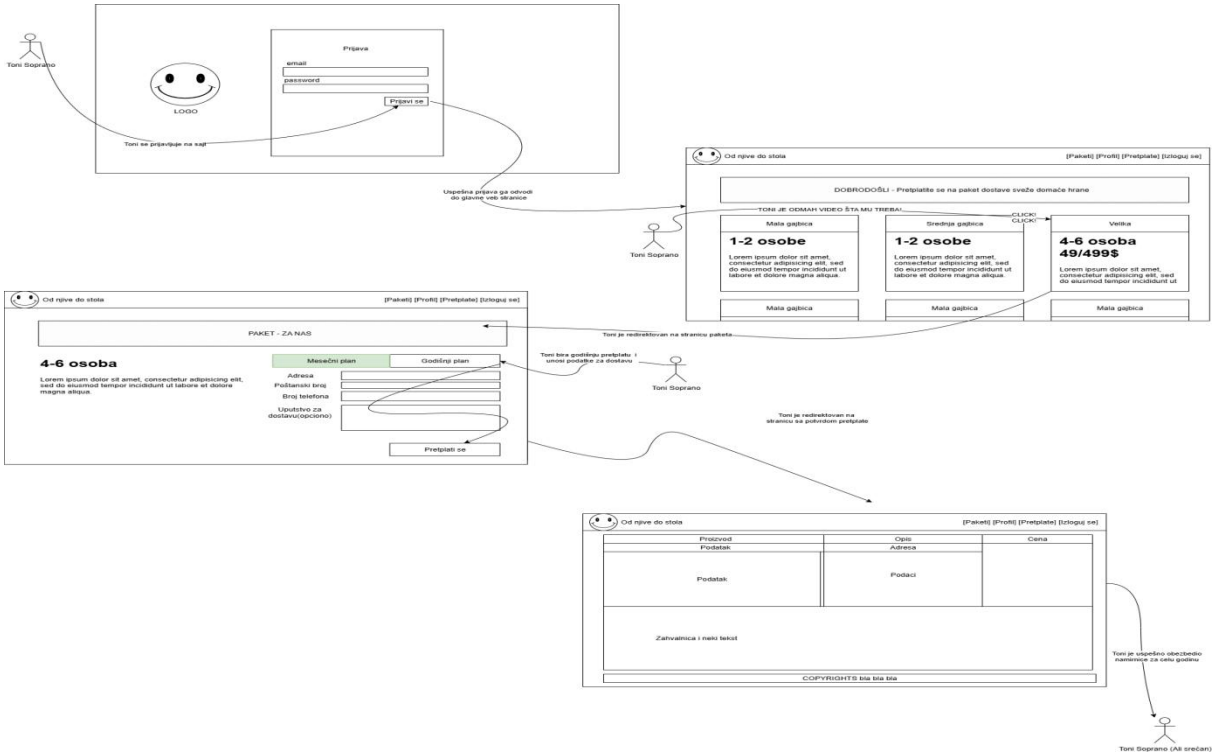
Nakon što je korisnik uneo sve podatke u formi i potvrdio pretplatu veb aplikacija redirektuje korisnika na prikaz na slici 10. Ovo služi korisniku kao potvrda da je uspešno kreirao

porudžbine i može da pregleda opet sve podatke kao podsetnik ili proveru ispravnosti podataka.



Slika 11. Prikaz dizajna veb prikaza stranice administratora za upravljanje podataka

Slika 11 prikazuje šablon izgleda administratorskog panela za podatke u bazi podataka koje može da izmeni preko veb aplikacije. Finalna verzija ne izgleda isto ali ovaj niskoverni prikaz je dobra početna tačka za razgovor sa naručiocem softvera i objašnjenje funkcionalnosti.

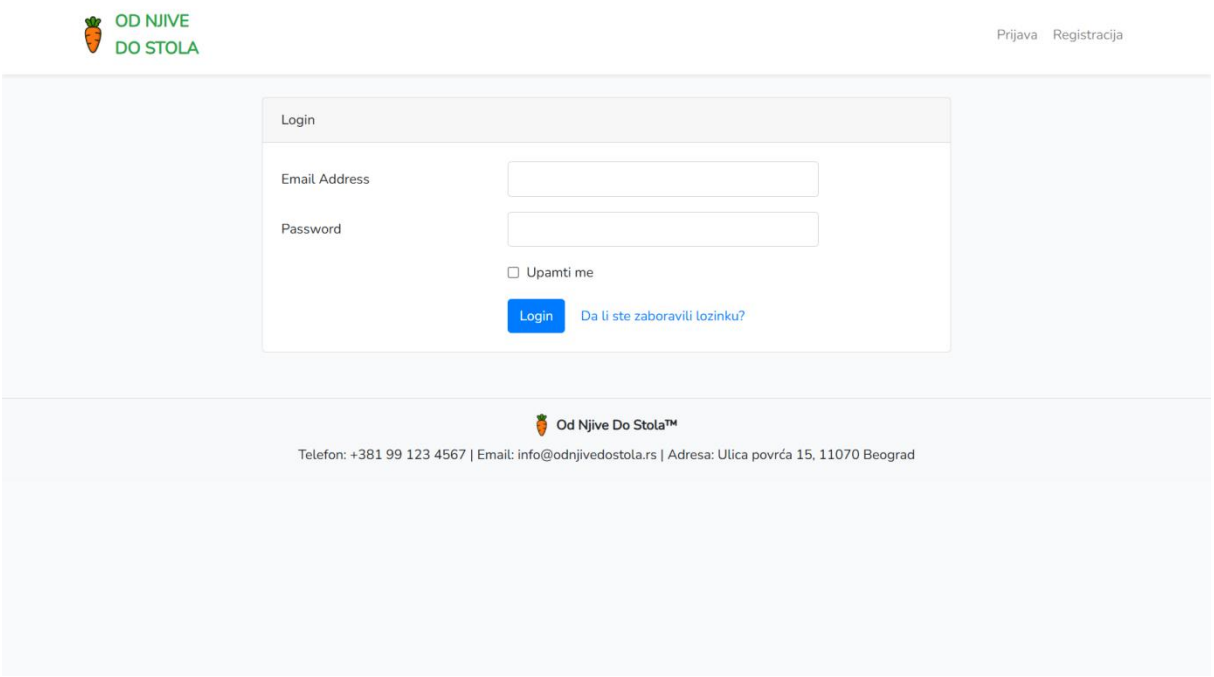


Slika 12. Prikaz šeme putanje kroz ekrane

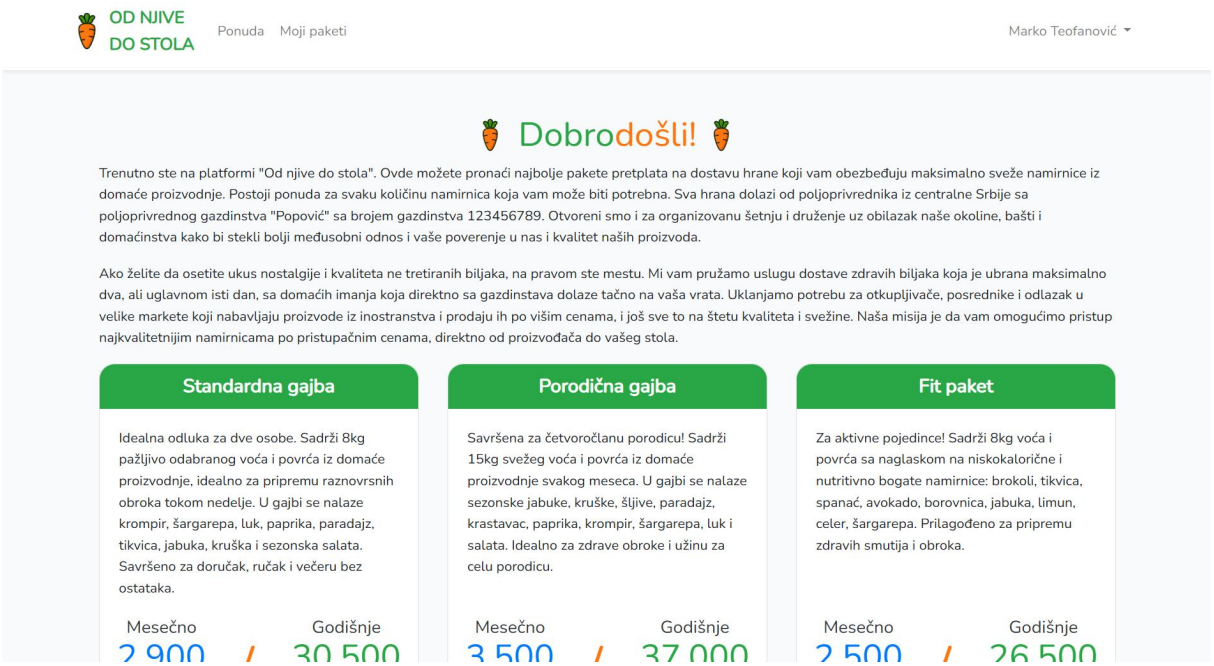
Visokoverni prototip dizajna aplikacije

Bootstrap [10] dizajn je odaran zbog mnogih prednosti. Bootstrap lako omogućava sajt koji se prilagođuje svim ekranima, obezbeđuje konzistentnost dizajna između elemenata i daje moderan potpis. Veliki benefit je "GRID" sistem prilagođavanja dimenzija i rasporeda elemenata. Zbog benefita bootstrap radnog okvira je i dizajn prilagođen da prati takav izgled.

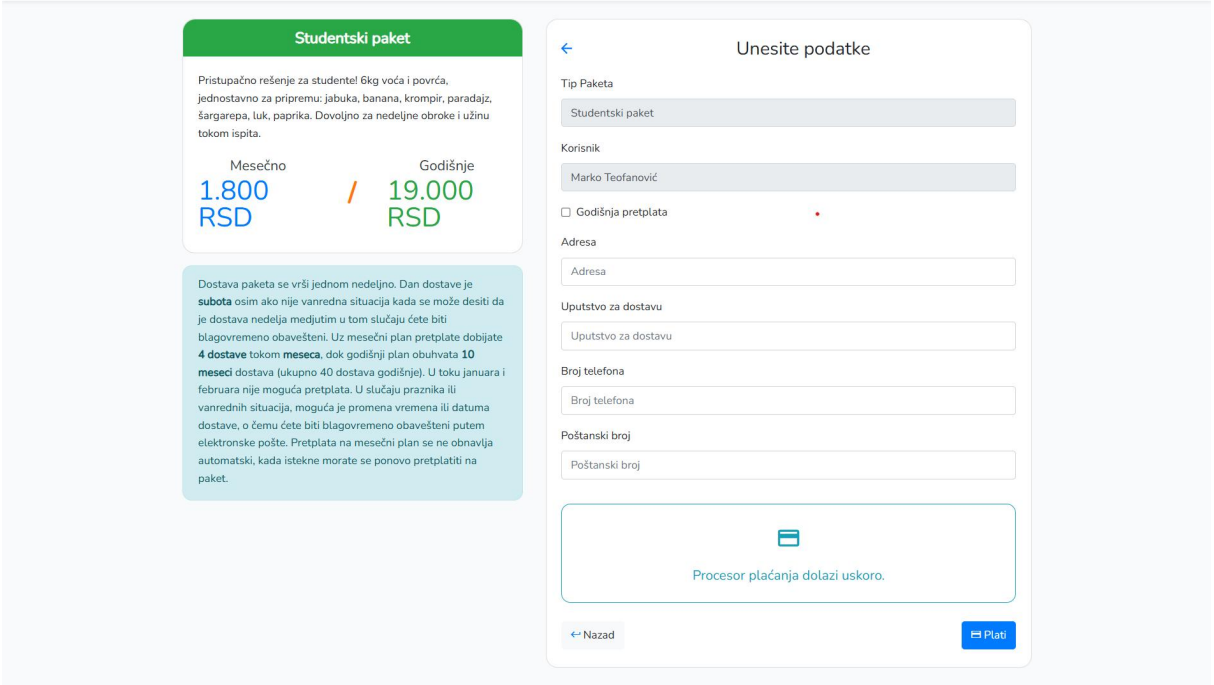
Dizajn sajta nema pregršt boja, paleta je pažljivo odabrana da odgovara temi usluge. Logo i generalno dizajn bootstrap dizajn nemaju oštre ivice pošto je za sajtove gde je tematika hrana preporučljivo da tako izgledaju elementi.



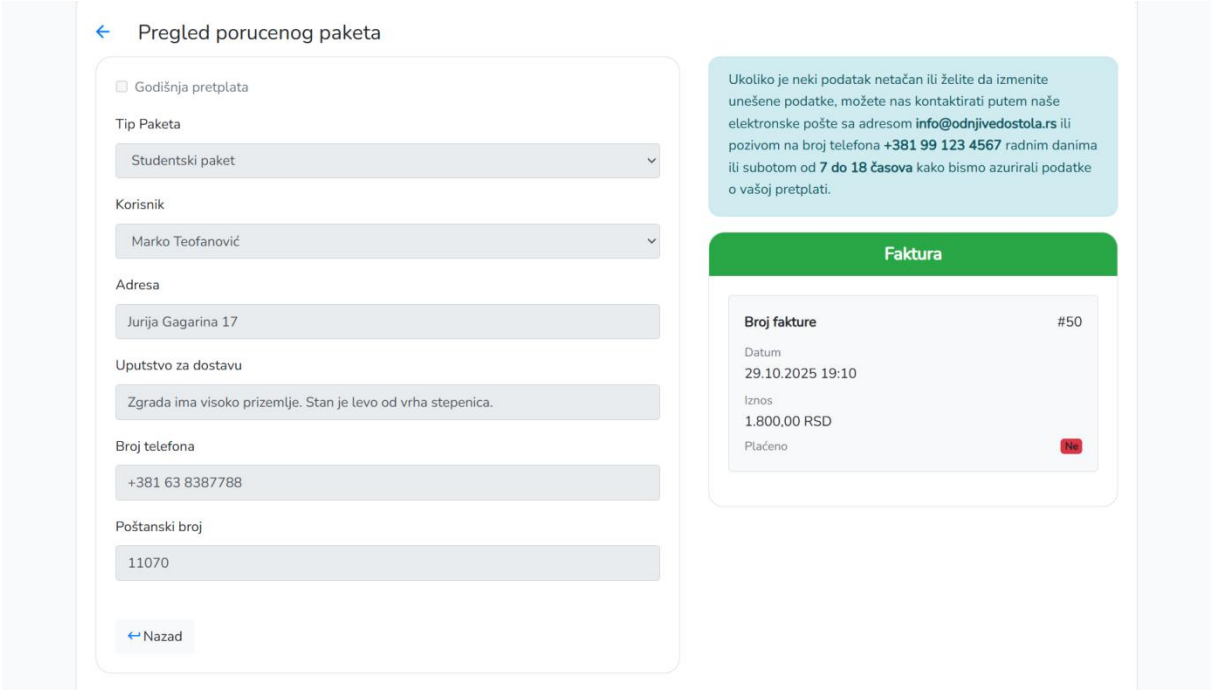
Slika 13. Prikaz login stranice



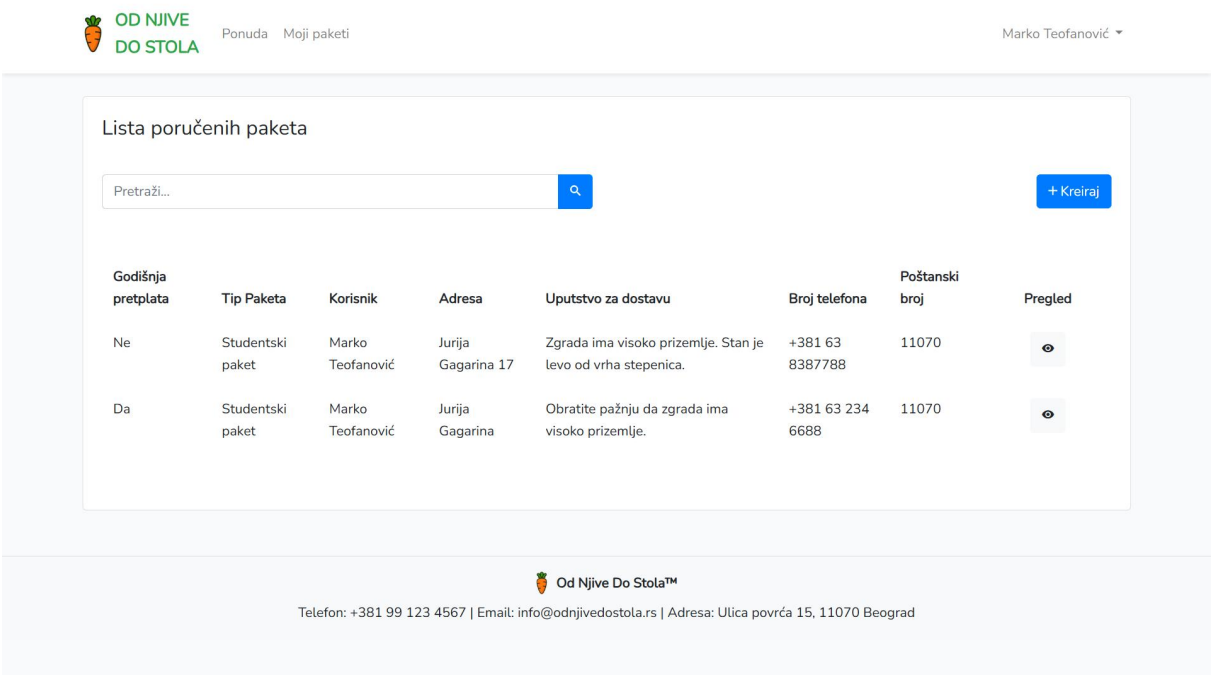
Slika 14. Prikaz dizajna početne stranice



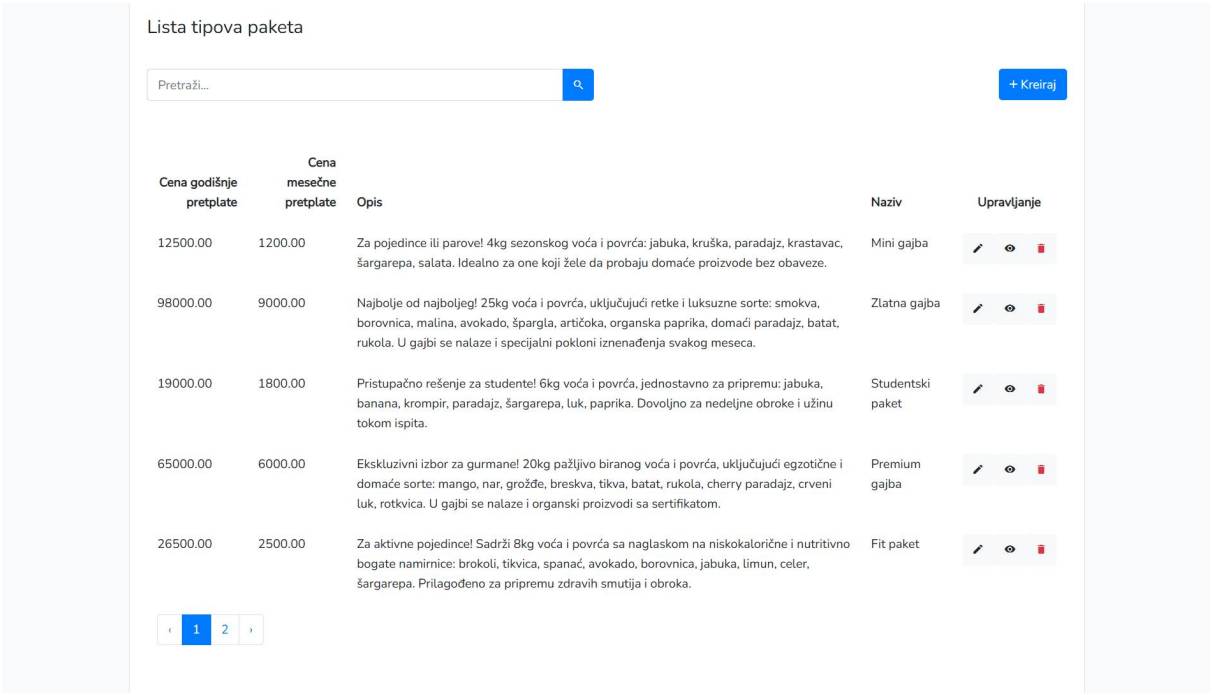
Slika 15. Prikaz dizajna stranice detalja paketa



Slika 16. Prikaz dizajna detalja pretplate

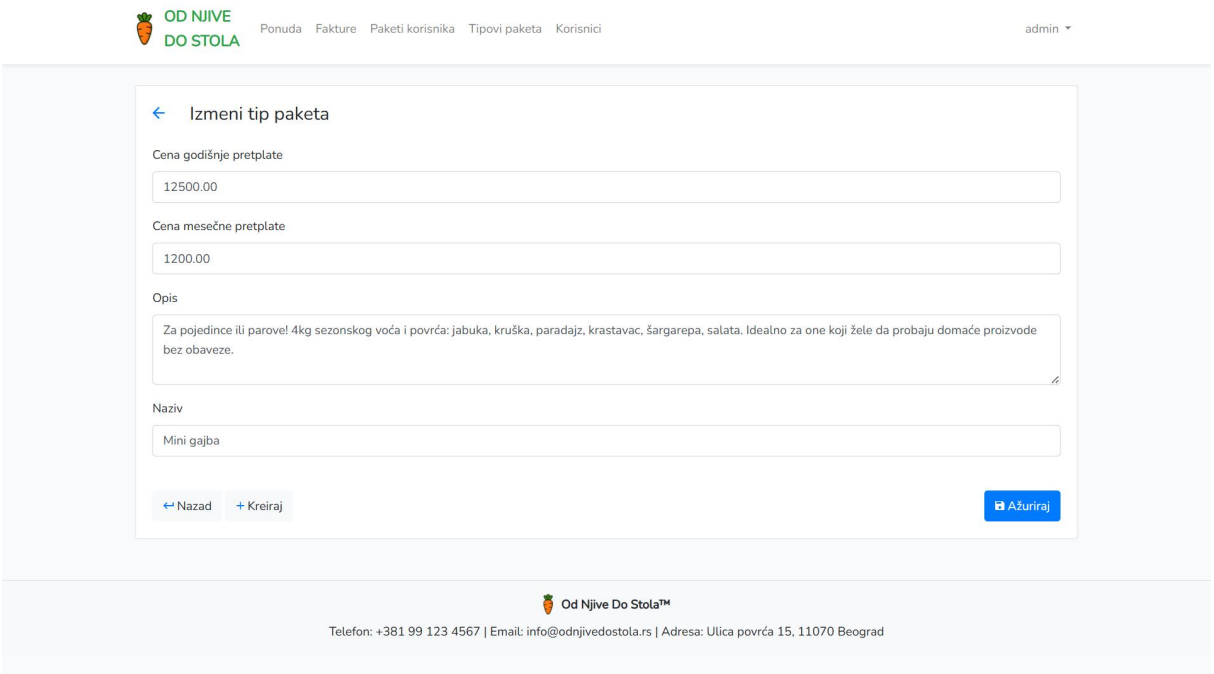


Slika 17. Prikaz dizajna stranice istorije pretplata



Slika 18. Prikaz stranice admin panela

Ostali admin paneli za fakture, pakere korisnike i korisnike imaju isti dizajn kao primer u slici 18.



Slika 19. Prikaz admin stranice za izmenu podataka

PHP

Projekat “Od njive do stola” je napisan u php programskom jeziku. Php je skriptni jezik opšte namene ali koji se najčešće koristi u domenu veb razvoja. Php je brz što ga čini pogodnim za serversku logiku dinamičnih aplikacija koje su postale veoma zastupljene. Za pokretanje aplikacije osnovni zahtev je php instalacija verzije 8.3.25. [13]

Laravel

Laravel je radni okvir php programskog jezika zasnovan na MVC tipu arhitekture. U sastav laravela spada i ORM (objektno-relacioni maper) sa nazivom “Eloquent” uz pomoć kog je omogućeno jednostavno rukovanje bazom podataka mapiranjem tabela SQL baze podataka na modele u kodu. U projektu “Od njive do stola” su implementirane i migracije koje pruža laravel zajedno sa seederima koji imaju realistične test podatke što olakšava razvoj i testiranje aplikacije. Lokalno pokretanje projekta se vrši komandom “**php artisan serve**” i uz pomoć artisan komandi može da se ubrza mnogo akcija jer moguće je generisati različite delove aplikacije (više u dokumentaciji) . [12][11]

Composer

Composer je “menadžer” alat za upravljanje zavisnostima u php-u koji je neophodan za rad sa laravel projektom. Composer automatski ažurira ili popuni sve neophodne podatke o datotekama tj. bibliotekama u datoteci composer.json nakon čega ih automatski instalira ili ažurira. Na ovaj način se lako integrišu eksterni laravel paketi kao što su test alati ili biblioteke koje pomažu pri implementaciji autentifikovanja ili druge php biblioteke bez ručnog ubacivanja koda.[15]

SQL server

Projekat “Od njive do stola” je realizovan uz pomoć Microsoft SQL Server-a za dugotrajno skladištenje podataka. Laravel zvanično podržava verzije od 2017. Konfiguracija za SQL server koristi sqlsrv(PDO) drajver za povezivanje sa bazom podataka. Da bi sistem funkcionisao potrebno je imati zvanične Microsoftove php ekstenzije sqlsrv i pdo_sqlsrv sa odgovarajućim postfiksom za sistem na kojem se pokreće i ODBC drajverom.

Node.js i npm

Node.js je javascript radni okvir, a npm je standardni menadžer paketa za node.js. Node.js i npm su neophodni uslovi za korišćenje laravela jer oni obrađuju front-end zadatke.

Instaliranje i kompajliranje bootstrapa koji je korišćen u projektu “Od njive do stola” je primer upotrebe. [14]

Bootstrap

Front-end radni okvir bootstrap služi za uključivanje već spremnih css klasa koje mogu dalje da se pozivaju u HTML-u. Obezbeđuje sajt koji je prilagodljiv na različitim ekranima. Bootstrap klase mogu i ručno da se promene u slučaju da je dizajn drugačije zamišljen ali je svakako koristan jer nije potrebno kucati sve klase od nule i smanjuje se potreba za boilerplate poslom. Bootsrap postavlja podrazumevani moderni izgled za svaki tipičan element koji se koristi u savremenim veb stranicama. Takođe ima grid sistem rasporeda elemenata. [16]

Git

Verzionisanje koda je omogućeno uz pomoć git alata. Git je distribuirani sistem kontrole verzija (DVCS) koji lokalno čuva kompletnu istoriju repozitorijuma. Korišćenje gita omogućava rad i bez stalne veze sa centralnim serverom. Uz pomoć gita je olakšano praćenje promena kroz git grane i commit-ove i lakša je koordinacija razvoja jer lako vratiti se na ranije verzije koda po potrebi. [17]

Vemto

Uslov za generisanje laravel projekta su sve prethodno navedene tehnologije. Vemto je open-source aplikacija koja služi za generisanje laravel. Vemto omogućava kreiranje modela, migracija, kontrolera i drugih komponenti kao i API-jeva.

MVC pristup

Laravel koristi Model-View-Controller arhitekturu kako bi se jasno razdvojio sloj poslovne logike, sloj podataka i prezentacioni sloj.

Kontroleri obrađuju HTTP zahteve i koordiniraju tok aplikacije. Modeli koji su kreirani uz pomoć eloquent-a upravljaju strukturom podataka i poslovnom logikom. Blade pogledi(šabloni izgleda stranice) čine prezentacioni sloj i prikazuju odobrene podatke korisniku. Dakle kontroler prihvata zahtev, poziva odgovarajući model za rad sa podacima i onda se rezultati prosleđuju blade pogledu i konačno se generiše HTML koji se šalje na veb pregledač.

Blade šabloni su smešteni u resources/views direktorijumu i imaju jednostavnu sintaksu za ugrađivanje podataka. Na taj način je moguće potpuno izbeći PHP kod međutim idalje je moguće uneti php blok ako za to ima potrebe. Ali ovakva postavka olakšava održavanje.

Inicijalni skelet modela i migracija je uz pomoć vemto alata kreiran pri kreaciji samog laravel projekta. Modeli se nalaze u direktorijumu App/Models, a kontroleri u App/Http/Controllers.

Migracije i seederi

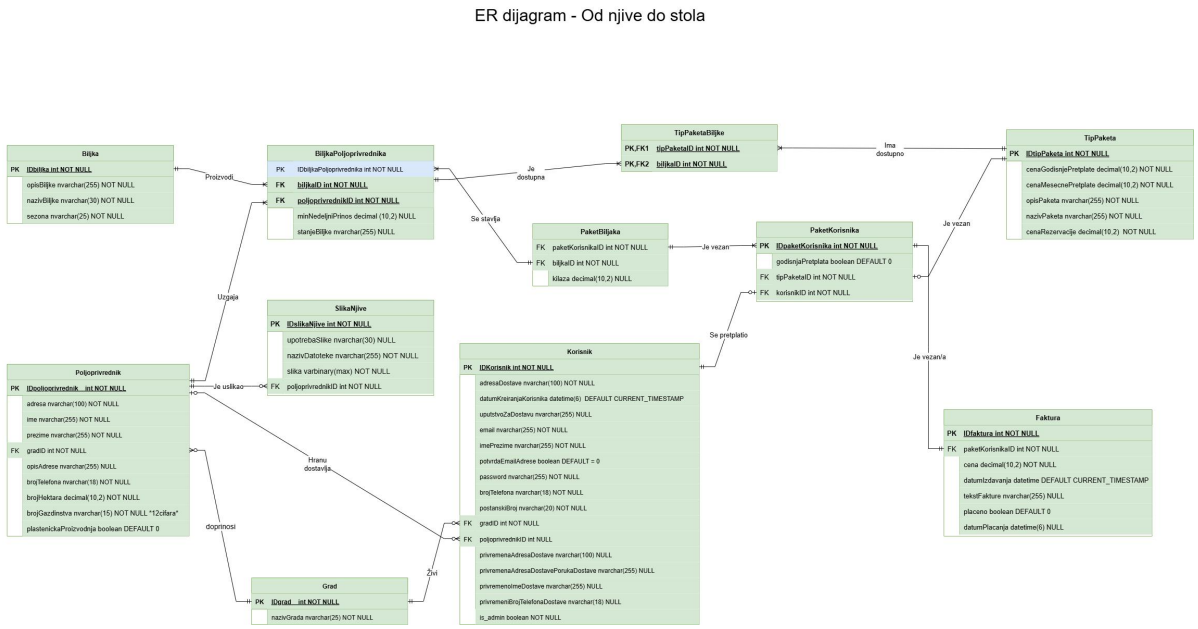
Migracije u laravelu služe kao sistem upravljanja verzijama šeme baze podataka. One omogućavaju programeru da se definiše struktura (model)baze, a to se odnosi na tabele, kolone, indekse i relacije. Migracije se sastavljaju sa php kodom što znači da programer ne mora da kuca SQL kod. Pri pokretanju migracija laravel kreira ili ažurira tabele u bazi podataka na osnovu sadržaja migracionih datoteka. Prednost je olakšana prenosivost i automatizacija, svi koji rade na projektu mogu lako primeniti najnoviju šemu unosom prostih migracionih komanti u terminalu. Migracije nisu vezane za jednu vrstu baze podataka jer laravel može lako da se prilagodi i MySQL-u, PostfreSQL-u, SQLite-u i SQL serveru koji je korišćen u ovom projektu, ovo znači da iz jednog koda sa jednom sintaksom se obezbeđuje višestruka kompatibilnost.

Migracije su iskorišćene za kreiranje svih tabela, kolona i relacija koje su napomenute u ER dijagramu u SQL server bazi podataka. Komanda ***php artisan migrate*** pokreće migracione datoteke i kompletna baza podatka se ažurira ili kreira po definisanim specifikacijama.

Seederi funkcionišu slično kao migracije i omogućuju popunjavanje baze podataka test podacima ili podrazumevanim podacima koji su predviđeni da uvek budu unešeni u bazu. Nalaze se u direktorijumu database/seeders. U okviru seedera se piše kod koji ubacuje podatke uz pomoć eloquent modela. Moguće je definisati veći broj korisnika ili bilo kog drugog entiteta i popuniti tabele podacima. U projektu Od njive do stola su kreirani seeder-i sa realističnim podacima koji se mogu koristiti pri pokretanju aplikacije kao simulacija upotrebe servisa. Alternativa su fabrike podataka koje omogućavaju lažne zapise koji odgovaraju tipu podatka atributa međutim u nekim slučajevima kao što je testiranje ovog proizvoda to nije prihvatljivo rešenje jer su podaci besmisleni i nečitki.

Zaključak

Izbor tehnologija i pristupa omogućio je organizovan i uspešan razvoj rešenja realnog problema u modernom svetu. Problem koji je rešen je vrlo specifičan međutim dalja vizija bi bila proširenje sistema da ima mogućnost obsluženja bilo kog regiona i naprednije funkcionalnosti. Trenutni projekat je dobra osnova za testiranje ovakvog koncepta poslovanja. U slučaju rapidnog uspeha sledeći cilj i početak bi bio reorganizacija modela i veza, a primer naprednije baze bi izgledao ovako:



Slika 20. Budući dijagram entiteta i veza

Uveo bi se u direktno u informacioni sistem poljoprivrednik. Poslovanje ne bi bilo isključivo u Beogradu već bi postojao i entitet gradovi i bilo bi moguće poslovanje u bilo kom regionu dokle god ima poljoprivrednika. Korisnik bi mogao da unese i alternativnu adresu dostave i mogao bi da promeni aktivnu adresu. Tabela slike bi omogućila poljoprivrednicima da se kvalitetnije izreklamiraju. Korisnik bi mogao da bira pojedinačno biljke i pored opcije već predodređenih paketa u slučaju da želi specifičnu korpu.

Literatura

[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*

[6] *Taylor Otwell, https://laravel.com/docs/10.x/eloquent, © 2025 Laravel*

[7] *Microsoft, https://learn.microsoft.com/en-us/sql/t-sql/data-types/nchar-and-nvarchar-transact-sql?view=sql-server-ver17, 2025*

[8] *Network Working Group - J.Klensin, Simple Mail Transfer Protocol - SMTP, https://datatracker.ietf.org/doc/html/rfc5321, October 2008*

[9] *Open source VemtoOrg, https://vemto.app, 2025*

[10] *Bootstrap team, https://getbootstrap.com/docs/5.3/getting-started/introduction/, 2024*

[11] *Laravel LLC, Eloquent ORM https://laravel.com/docs/10.x/eloquent, February 14, 2023.*

[12] *Laravel LLC, Database https://laravel.com/docs/10.x/migrations, February 14, 2023.*

[13] *The PHP Group, PHP https://www.php.net/, 2025*

[14] *OpenJS Foundation, NodeJS https://nodejs.org/en, 2025*

[15] *Nils Adermann, Jordi Boggiano i mnogi drugi benefaktori zajednice, Composer https://getcomposer.org/, 2025*

[16] *Bootstrap Team, Bootstrap https://getbootstrap.com/, 2025*

[17] *Scott Chacon i Ben Straub,https://git-scm.com/book/en/v2, 2014*

[18] *Računarski fakultet, predavanja, 2022-2025*