

Prilmljeno/Submitted: 30.04.2023.  
Prihvaćeno/Accepted: 15.06.2023.

Stručni rad  
Professional paper  
*JEL Classification: 030, 039*

## **PRIJEDLOG RJEŠENJA RAZVOJA INFORMACIONOG SISTEMA BRZE POŠTE – ANALIZA I DIZAJN**

### **SUGGESTED FOR THE DEVELOPMENT OF THE EXPRESS MAIL INFORMATION SYSTEM - ANALYSIS AND DESIGN**

Vernes Vinčević\*  
Ibrahim Obhodaš\*\*

#### **SAŽETAK**

Svjedoci smo da sve više kompanija i trgovina pored fizičke trgovine unapređuje svoje poslovanje kroz online trgovinu i nudi usluge dostave putem brzih pošti. Živimo u vremenu gdje je nezamislivo živjeti bez dostavljanja proizvoda na kućnu adresu. Počevši od mlađih generacija pa do penzionera, svi imamo potrebe za dostavom nekog željenog produkta na kućnu adresu. Brza pošta djeluje na zahtjev gotovo svih potreba korisnika od namirnica, tehnologije, odjeće, bijele tehnike, itd. Veoma važan segment poslovnog sistema kompanije čija je djelatnost usluga brze pošte je informacioni sistem. Kako bi se ispravno projektovao predmetni sistem potrebno mu je pristupiti metodološki i sistemski. U radu će biti prikazano idejno rješenje razvoja informacionog sistema posebno faza analize i sistemskog dizajna, dok će ostale faze biti kratko opisane. U radu će biti korištena metodologija životnog ciklusa razvoja sistema ( System Development Life Cycle, SDLC) koja je najčešće korištena metodologija za razvoj sistema u mnogim organizacijama.

**Ključne riječi:** Informacioni sistem, sistemska analiza, sistemski dizajn, use case dijagrami, arhitektura sistema, životni ciklus razvoja sistema.

---

\*Doktorant na Fakultetu informacionih tehnologija, Sveučilište/Univerzitet "VITEZ", Vitez, e-mail: [vernes.vincevic@unvi.edu.ba](mailto:vernes.vincevic@unvi.edu.ba)

\*\*Prof. dr. sc. Fakultet informacionih tehnologija, Sveučilište/Univerzitet "VITEZ", Vitez, e-mail: [ibrahim.obhodjas@unvi.edu.ba](mailto:ibrahim.obhodjas@unvi.edu.ba)

## ABSTRACT

We are witnessing that more and more companies and shops, in addition to physical trade, are improving their business through online trade and offering delivery services via express mail. We live in a time where it is unimaginable to live without home delivery of products. Starting from younger generations to pensioners, we all have a need for the delivery of a desired product to our home address. Express mail operates on the request of almost all user needs from food, technology, clothing, white goods, etc.. A very important segment of the business system of a company whose activity is express mail services is the information system. In order to correctly design the subject system, it is necessary to approach it methodologically and systematically. The paper will present the conceptual solution for the development of the information system, especially the phases of analysis and system design, while the other phases will be briefly described. The paper will use the system development life cycle methodology (System Development Life Cycle, SDLC), which is most often methodology for system development in many organizations.

**Keywords:** Information system, system analysis, system design, use case diagrams, system architecture, system development life cycle.

## UVOD

Poslovanje jedne brze pošte je skoro u potpunosti zasnovano na informacionom sistemu, koji pruža podršku korisnicima usluga. Od same ideje kako bi trebao da izgleda jedan funkcionalni informacioni sistem, u radu će biti pažljivo prikazani svi neophodni elementi za razvoj samog sistema, dok će akcenat biti na analizi i dizajnu. Rezultat rada je izrada projektne dokumentacije koja će u daljnim fazama biti podloga za kodiranje prema programerskom timu koji će na osnovu iste razvijati informacioni sistem. "Informacioni sistem je sistem koji prikuplja, pohranjuje, čuva, obrađuje i isporučuje informacije važne za organizaciju i društvo, tako da budu dostupne i upotrebljive za svakog ko se želi njima koristiti, uključujući poslovodstvo, klijente, zaposlene i ostale" (Lindsay, 2000, str. 30).

### 1. RAZVOJ INFORMACIONOG SISTEMA BRZE POŠTE

#### Pregled plana razvoja informacionog sistema

Kako bi se adekvatno projektovao odgovarajući informacioni sistem neophodno je izraditi kvalitetan plan razvoja informacionog sistema u odgovarajućem softveru za upravljanje projektima, koji sadrži korake odnosno faze razvoja informacionog sistema. Unutar plana uneseni su i neophodni resursi za razvoj ovog sistema, kako izgleda na slici 1.

Slika 1: Plan razvoja u softverskom alatu Project Libre

	📅	Name	Duration	Start	Finish	Predecessors	Resource Names
1	📅	📁 PLANIRANJE	2 days?	29/09/22 08:00	30/09/22 17:00		
2		Detaljno opisivanje sistema	2 days?	29/09/22 08:00	30/09/22 17:00		Project manager
3		Izrada potreba za informacijski sistem	2 days	29/09/22 08:00	30/09/22 17:00		Project manager
4		Formiranje projektnih timova	1 day?	29/09/22 08:00	29/09/22 17:00		Project manager
5	📅	📁 ANALIZA	36 days?	03/10/22 08:00	21/11/22 17:00		
6		Definisanje zahtjeva	5 days	03/10/22 08:00	07/10/22 17:00	1	Project manager
7		Definisanje specifikacija i izrada istih	4 days	10/10/22 08:00	13/10/22 17:00	6	Project manager
8		Uraditi detaljnu analizu tržišta	2 days	03/10/22 08:00	04/10/22 17:00	1	Project manager
9	📅	Uraditi analizu marketinga	1 day?	05/11/22 08:00	07/11/22 17:00	1	Project manager
10	📅	Uraditi SWOT analizu	1 day?	18/11/22 08:00	18/11/22 17:00	7	Project manager
11	📅	Izrada mogućih rizika	2 days	18/11/22 08:00	21/11/22 17:00	7	Project manager
12	📅	📁 DIZAJN	16 days?	22/11/22 08:00	13/12/22 17:00		
13		Opis svakog modula sistema	1 day?	22/11/22 08:00	22/11/22 17:00	5	Full stack developer
14		Opis svih funkcionalnosti	1 day?	22/11/22 08:00	22/11/22 17:00	5	Full stack developer
15		Dizajniranje UML dijagrama	4 days	22/11/22 08:00	25/11/22 17:00	5	Front end developer;UI/UX ...
16		Dizajniranje mockup/prototipweb stranice po UI/UX p	3 days	28/11/22 08:00	30/11/22 17:00	15	UI/UX designer
17		Dizajniranje mockup mobilne aplikacije	5 days	01/12/22 08:00	07/12/22 17:00	16	UI/UX designer
18	📅	Identifikacija svih tabela baze podataka	4 days	08/12/22 08:00	13/12/22 17:00	17	UI/UX designer
19		📁 IMPLEMENTACIJA	66 days?	14/12/22 08:00	15/03/23 17:00		
20		Izrada frontend aplikacije	10 days	14/12/22 08:00	27/12/22 17:00	18	Front end developer[50%];...
21		Kreiranje baze podataka	7 days	14/12/22 08:00	22/12/22 17:00	18	Back end developer
22		Implementacija funkcionalnosti zahtjeva za pošiljkom	7 days	14/12/22 08:00	22/12/22 17:00	18	Full stack developer
23		Implementacija backend-a	15 days	23/12/22 08:00	12/01/23 17:00	16;22	Full stack developer[50%];...
24		Izrada mobilnih aplikacija	7.5 days	13/01/23 08:00	24/01/23 13:00	23	Android developer[50%];iO...
25	📅	Dokumentiranje	10 days	03/02/23 09:00	17/02/23 09:00		Data Analyst[50%];Back en...
26	📅	Pojedinačna testiranja	3 days?	20/02/23 08:00	22/02/23 17:00		QA engineer
27	📅	Povezivanje frontend i backend dijela aplikacije	1 day?	21/02/23 08:00	21/02/23 17:00		Back end developer;Full sta...
28	📅	Uskladjivanje web i mobilne aplikacije	10 days	02/03/23 08:00	15/03/23 17:00		Back end developer;Full sta...
29	📅	Unos testnih podataka u bazu	2 days?	06/03/23 08:00	07/03/23 17:00		
30	📅	📁 TESTIRANJE	21 days?	07/03/23 08:00	04/04/23 17:00		
34		📁 DEPLOYMENT	10 days	05/04/23 08:00	18/04/23 17:00		
37		📁 Održavanje	10 days?	19/04/23 08:00	02/05/23 17:00		

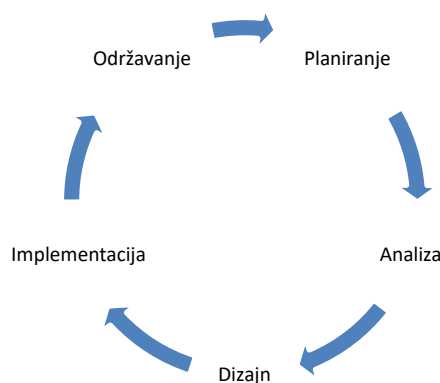
Izvor: Obrada autora u programu Project Libre

## 2. METODOLOGIJA

U radu je korištena metodologija životnog ciklusa razvoja sistema ( System Development Life Cycle, SDLC) koja je uobičajena metodologija za razvoj sistema u mnogim organizacijama. „Prema Elliott (2004) razvoj životnog ciklusa sistema (SDLC) može se smatrati da je najstarija formalizovana metodologija okvira za izgradnju informacionih sistema. Osnovna ideja SDLC je bila "da nastavi razvoj informacionih sistema na organizovan i metodičan način, zahtjevajući svaku fazu životnog ciklusa - od nastanka ideje do isporuke finalnog sistema - da bude obavljan strogo i redom” (Geoffrey, 2004, str. 51).

Životni ciklus razvoja softvera mogu i prilagođavaju timovi za razvoj softvera na osnovu filozofije, metodologije i okvira koji usvajaju prilikom izgradnje datog softverskog sistema ili između organizacija. Danas u svijetu imamo razne metodologije, okvire, koncepte i standarde razvoja softverskih sistema.

Slika 2: Metodološki okvir SDLC (Životni ciklus razvoja sistema) podjeljen u osnovne faze



Izvor: <https://eternalsunshineofthemind.wordpress.com/2013/02/13/system-development-life-cycle-sdlc/>  
(pristupano, 25.1.2023)

## Korisnički zahtjevi sistema

Korisnički zahtjevi definišu šta konkretno sistem treba raditi i kako će pomoći krajnjim korisnicima prilikom rada. Korisnički zahtjevi obrađeni tokom ovog projekta prikupljeni su raznim metodama za prikupljanje zahtjeva kao što su analiziranje tržišta, istraživanje interneta te postojećih softverskih rješenja, upoznavanje strategija specifičnih za datu oblast, poređenje sa sličnim ranije razvijenim sistemima, promatranje i sl. Ukoliko bi se razvoj ovakvog sistema radio za stvarnog klijenta koristile bi se sljedeće metode:

**Prikupljanje informacija** – prvi korak prilikom utvrđivanja korisničkih zahtjeva je prikupljanje informacija. U ovoj fazi se radi istraživanje tržišta kako bi se dobio bolji uvid u zahtjeve.

**Utvrđivanje korisničkih potreba** – u ovoj fazi se provodi intervjuisanje svih učesnika koji su uključeni u projekt. Prilikom intervjuisanja često se za korisnike provodi scenariji kojima se opisuje hipotetska situacija za koju korisnik iskazuje želju kako želi da se programska podrška ponaša.

**Planiranje i vrednovanje** – kada se prikupi dovoljno informacija moguće je početi s procesom utvrđivanja korisničkih potreba.

**Specifikacija zahtjeva** – kada se jednom svi korisnički zahtjevi definišu i utvrde potrebno ih je dokumentovati na odgovarajući način kako bi se mogli održavati. Dodatno, zahtjeve je potrebno kategorizirati po domeni kojoj pripadaju (npr. zahtjevi sistema, funkcionalnosti) te po važnosti ili prioritetu, što će biti prikazano u narednim fazama.

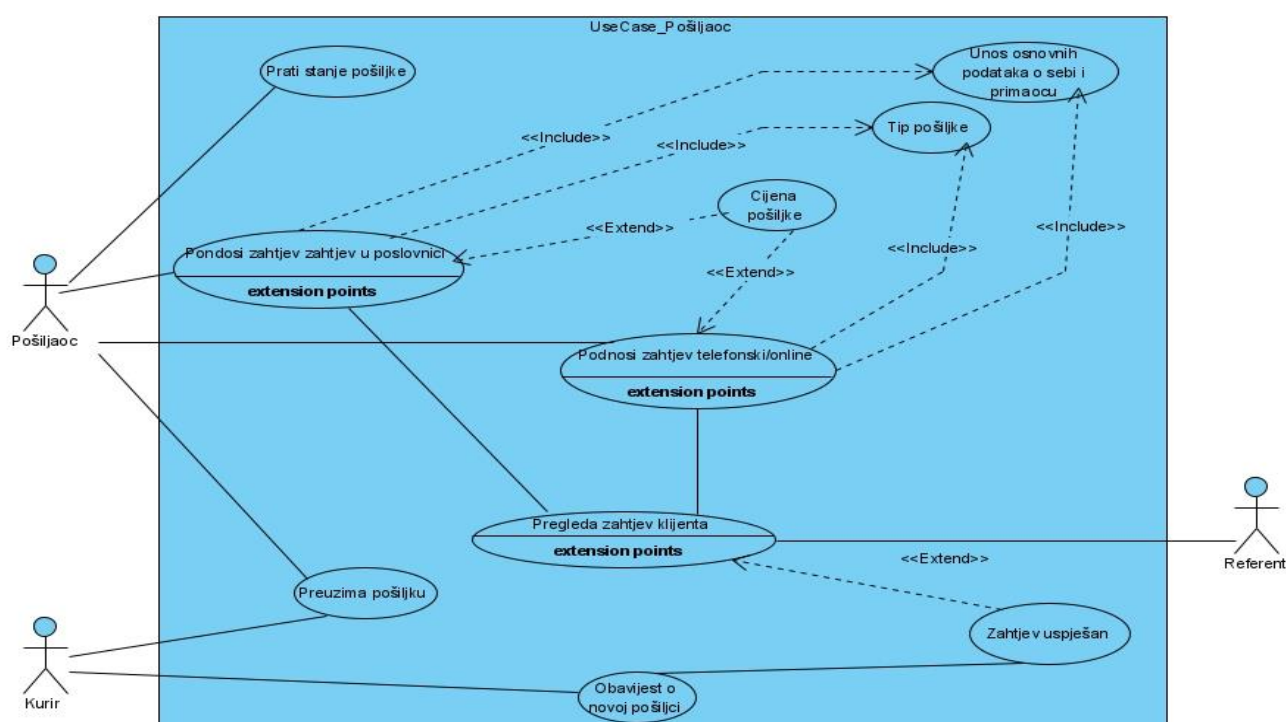
### 3. SISTEMSKA ANALIZA PROBLEMA

Sistemska analiza obuhvata definisanje UML dijagrama sistema koja sadrži USE Case dijagram, dijagrame aktivnosti, dijagram klasa koji su prikazani u nastavku. Modeliranje poslovnih procesa za analizu predmetnog informacionog sistema u radu će bit prikazani raznim UML dijagramima aktivnosti, sekvence, slučajevima upotrebe i dr.

#### Dijagrami slučajeva upotrebe

„Dijagrami slučajeva upotrebe odnosno Use case dijagrami predstavljaju najbolji način za prikazivanje interakcije između korisnika i sistema“ (Hoffer, George, Valacich, 2017, str. 152).

Slika 3: USE Case slanje pošiljke

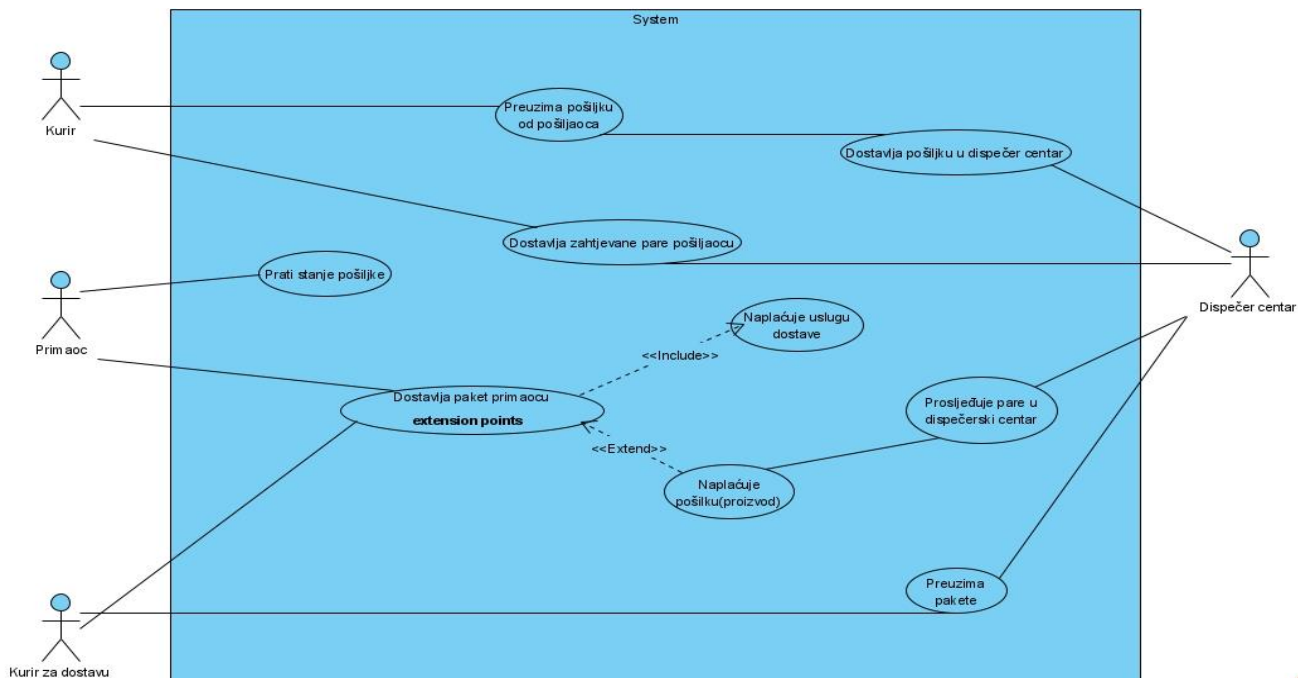


Izvor: Obrada autora u softverskom alatu Visual Paradigm ver. 16.3

Na prvom USE case dijagramu imamo tri aktera, te je opisano slanje pošiljke. Pošiljaoc ima mogućnost podnošenja zahtjeva za pošiljkom u poslovnicu kao i telefonski/online. Prilikom podnošenja zahtjeva obavezno je dostaviti podatke o sebi i primaocu kao i o kakvoj pošiljci se radi odnosno tip pošiljke. Ako je korisnik podnio zahtjev u poslovnicu, pošiljka je preuzeta na licu mjesta, što se odražava na drugi USE case odnosno, kurir nema potrebu za preuzimanjem pošiljke.

Referent pregleda zahtjev za pošiljku te obavještava kurira o novoj pošiljci, te kurir kurir preuzima pošiljku od pošiljaoca i dostavlja je u dispečer centar radi daljnog transporta u dispečer centar u regiji u kojoj se nalazi primaoc pošiljke. Pošiljaoc ima mogućnost praćenja trenutne lokacije pošiljke korištenjem tracking numbera.

Slika 4: USE Case dostavljanje pošiljke

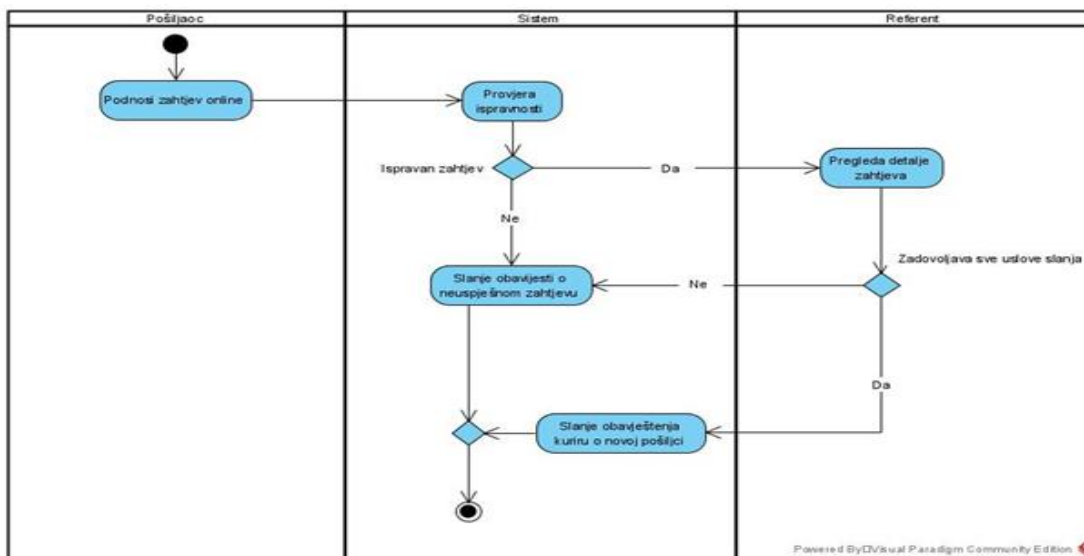


Izvor: Obrada autora u softverskom alatu Visual Paradigm ver. 16.3

**Dijagrami aktivnosti**

Dijagram aktivnosti odnosno Activity dijagram je vizualni dijagram baziran na procesnom pristupu, tj. opisuje procese u datom sistemu te nam ovi dijagrami govore kako sistem radi.

Slika 5: Zahtjev za pošiljku online



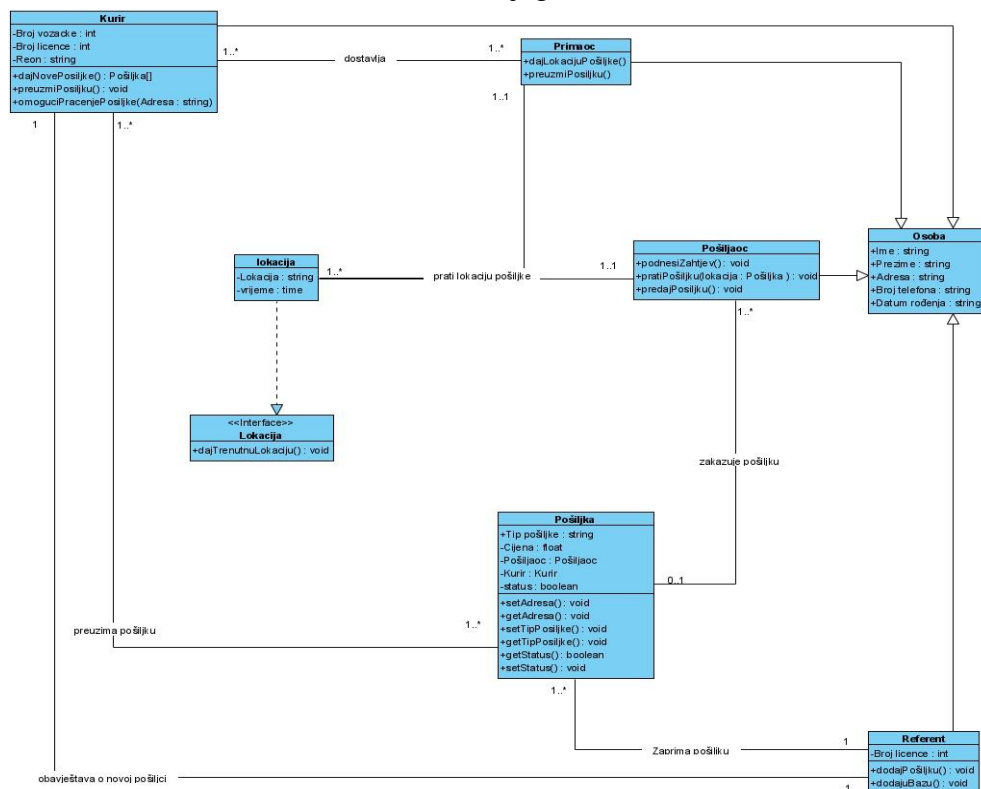
Izvor: Obrada autora u softverskom alatu Visual Paradigm ver. 16.3

Na prvom dijagramu aktivnosti imamo tri aktera: pošiljaoc, referent i sistem. Korisnik podnosi zahtjev na stranici, sistem automatski provjerava da li je ispravan zahtjev odnosno da li su unešeni svi potrebni podaci, ako je zahtjev neispravan, sistem šalje obavijest korisniku o neispravnosti gdje se završava aktivnost. Ukoliko je zahtjev unešen ispravno, referent pregleda ispravnost adrese i podataka i ukoliko je unešeno ispravno šalje obavijest kuriru o novoj pošiljci te se aktivnost završava, ukoliko nije ispravno šalje obavijest korisniku o neispravnom zahtjevu kako bi popunio ponovo te se aktivnost završava.

#### 4. SISTEMSKI DIZAJN INFORMACIONOG SISTEMA

U fazi sistemskog dizajna se prevodi preporučeno rješenje u logičke a zatim u fizičke specifikacije sistema. Prikazani će biti potrebni dijagrami koji opisuju ovu fazu prije same implementacije. „Dijagram klasa opisuje tipove objekata u sistemu i različite vrste statičkih veza koje postoje među njima, ovi dijagrami, također, prikazuju svojstva i operacije klasa“.

Slika 6: Dijagram klasa



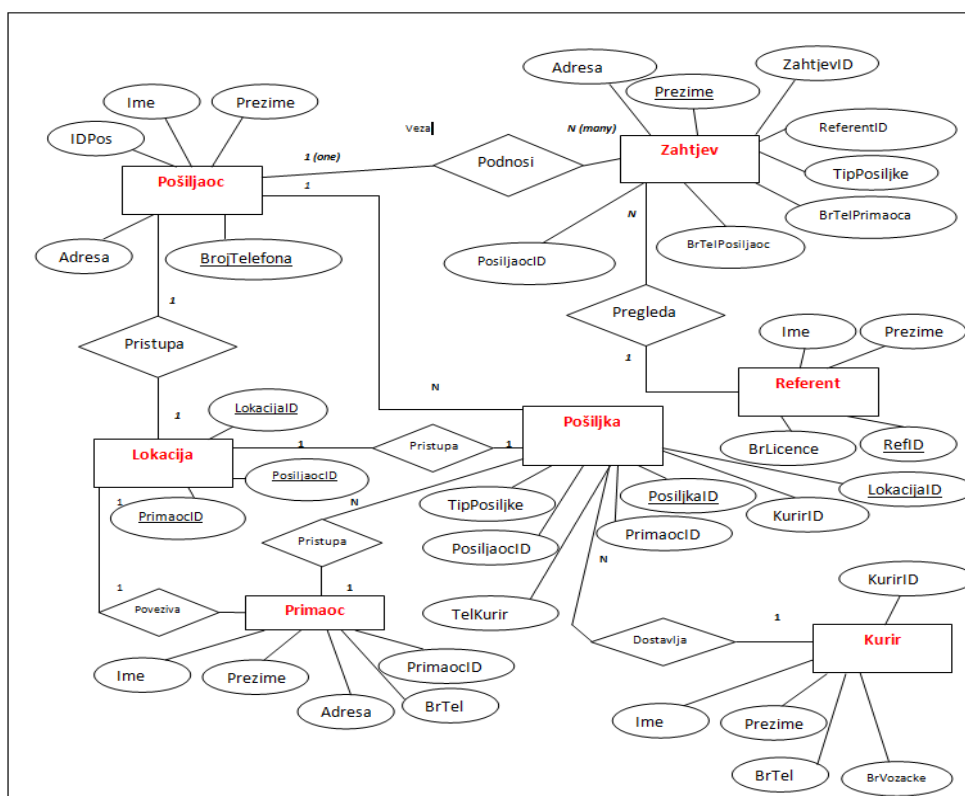
Izvor: Obrada autora u softverskom alatu Visual Paradigm ver. 16.3

Dijagram klasa prikazuje povezivanje klasa u sistemu, i obuhvata 7 klasa a to su Pošiljaoc, Referent, Primaoc, Kurir, Osoba, Lokacija i Pošiljka. Svaka od klasa sadrži potrebne atribute i metode. Klasa osoba sadrži sve potrebne atribute za osobe tj. aktere, te klase Pošiljaoc, Primaoc, Referent i Kurir generalizuju odnosno nasljeđuju atribute klase osoba.

## Dijagram baze podataka

Na dijagramu su prikazani entiteti i relacija u bazi podataka. Postoji 7 entiteta koji posjeduju svoje atribute. Entitet Pošiljaoc u relaciji je sa entitetom Zahtjev i ima vezu one to many odnosno jedan pošiljaoc može podnijeti više zahtjeva, dok više zahtjeva može biti podneseno od jednog pošiljaoca. Također i kod veze Referenta i Zahtjeva, jedan referent može pregledati više zahtjeva, dok više zahtjeva može biti pregledano od jednog referenta.

Slika 7: Dijagram baze podataka



Izvor: Obrada autora u softverskom alatu Draw.io

Entitet pošiljka pored primarnog ključa sadrži atribut tipPosiljke te foreign ključeve tabela Kurir, Pošiljaoc, Primaoc i Lokaciju koji omogućavaju povezivanje i pristupanje podacima između tih tabela. Tabela Kurir povezana je samo sa pošiljkom preko veze one to many odnosno jedan kurir može dostaviti više pošiljki, dok više pošiljki je moguće dostaviti od strane samo jednog kurira. Entitet Lokacija je u vezama one to one sa Primaocem i Pošiljaocem iz razloga što jedan korisnik pristupa jednoj lokaciji, i jednoj lokaciji pristupa samo jedan korisnik odnosno Pošiljaoc i Primaoc.



## Dizajn korisničkog interfejsa web aplikacije informacionog sistema

Slika 8: Mogući (mockup -nacrt) izgleda web stranice



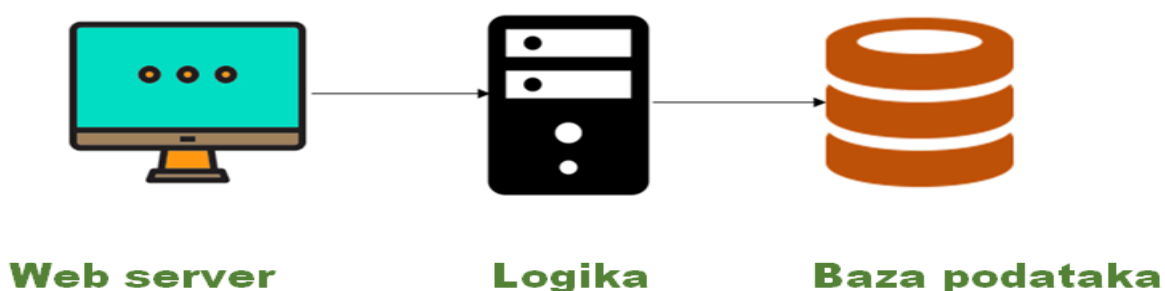
Izvor: Obrada autora u softveru Adobe XD

Mockup stranice je osmišljen kako bi klijentima olakšao vizuelizaciju određenih grafičkih projekata u pogledu tekstura, oblika i dimenzija.

## 5. MOGUĆA IZVEDBA ARHITEKTURE I IMPLEMENTACIJA SISTEMA

Nakon analize i potreba sistema planirano je korištenje troslojne arhitekture sistema tj. arhitektura sa odvojeno povezanim korisničkim interfejsom, logikom i bazom podataka. Troslojna arhitektura je klijent-preglednik arhitektura u kojoj se funkcionalna procesna logika, pristupa podacima, pohranjivanje podataka i korisnički interfejs razvijaju i održavaju kao neovisni moduli na zasebnim platformama. Troslojna arhitektura je obrazac dizajna softvera i dobro uspostavljena softverska arhitektura. Na strani baze podataka planirana je upotreba relacionih baza podataka, radi potrebe sortiranja podataka u tabele i relacija između njih.

Slika 9: Ilustracija troslojne arhitekture korištene u radu



Izvor: Obrada autora u softveru draw.io

Još jedan od razloga upotrebe troslojne arhitekture jeste mogućnost pohranjivanja podataka na više lokacija, kao i korištenje cross platforme odnosno omogućavanje web i mobilne aplikacije za Android i iOS.

### Korištene tehnologije za implementaciju frontend i backend dijelova sistema

Korisnički interfejs je jako bitan dio prilikom razvoja softverskog rješenja. Korisnički interfejs je jedino što krajnji korisnici sistema vide. Korisnički interfejs za informacijski sistem brze pošte sadrži navbar meni sa funkcionalnostima i olakšava kretanje kroz sistem.

Za implementaciju frontenda korišten je Angular 13.1.1 framework koji je TypeScript bazirani i također open source framework a kreiran od strane Google-a i koristi se za kreiranje web aplikacija. Angular nam pomaže strukturirati JavaScript kod, lahko ga je testirati. Stranica je podijeljena u komponente.

*Back-end* je dio programskog koda koji se izvodi na serveru, koji prima zahtjeve od klijenata i sadrži logiku za slanje odgovarajućih podataka natrag klijentu. Pozadina također uključuje bazu podataka koja će trajno pohranjivati sve podatke za aplikaciju. Logika koja se izvršava na server strani zahtjeva upotrebu nekih od logičkih/server side programskih jezika. Za implementaciju logičke strane ovog informacijskog sistema planirana je upotreba Node.js

„Postoje dva tipa komunikacije između servera i baze. Dva osnovna tipa koja se koriste prilikom implementacije softvera, pa tako i informacionog sistema, su upis u bazu podataka i čitanje podataka iz baze“.(Sommerville, *Software engineering*, 2007, str. 15).

Za pohranjivanje podataka u bazu koristiti će se relaciona baza odnosno MySQL baza. Razlog upotrebe MySQL baze podataka jeste mogućnost pohranjivanja podataka u tabele i relacija između tabela, visoke performanse baze, te omogućavaju fleksibilnost i skalabilnost.

## ZAKLJUČAK

Percipirano je da ovaj tip kompanije, koji se bavi brзом dostavom zahtjeva odgovarajući informacioni sistem. Kako bi se adekvatno i ispravno projektovao ovaj informacioni sistem, te kako bi se omogućila kvalitetna implementacija ovog sistema, napravio sam plan projekta koji sadrži korake za izradu ovog informacionog sistema podijeljen u više faza, te su uneseni resursi neophodni za razvoj ovog sistema. Također jer prikazan dizajn formi odnosno kako bi trebao grafički izgledati unos podataka na stranici, te je dizajniran izgled izvještaja tj. gdje će se ispisivati ti podaci. Nakon sistemskog dizajna, dizajniran je sami Mockup stranice odnosno dizajn korisničkog interfejsa, te je obuhvaćeno i prikazano na koji način rade dijalozi ovog informacionog sistema.

Sam rad na izradi ovog rješenja se pokazao kao značajan za naš tim u smislu sticanja iskustva u projektovanju informacioni sistema po nekoj metodologiji i nekog novog pogleda na izazove razvoja informacionih sistema, koji po svojoj prirodi nisu jednostavan zadatak. Rezultat rada je projektna dokumentacija potrebna za daljni razvoj sistema namjenjena da olakša programerskom timu u finalnoj realizaciji sistema.

## LITERATURA

1. Geoffrey Elliott (2004) *Global Business Information Technology: an integrated systems approach*.
2. J. A. Hoffer, J. F. George, J. S. Valacich: *Modern Systems Analysis and Design*, Prentice Hall College Div, 2017
3. J. L. Whitten, L. D. Bentley, K. C. Dittman: *Systems Analysis & Design Methods*, McGraw-Hill Higher Education, 2004
4. L. Maciaszek: *Requirements Analysis and System Design: Developing Information Systems with UML*, Addison Wesley Higher Education, 2007
5. Lindsay, John (2000). [Information Systems – Fundamentals and Issues](#). Kingston University, School of Information Systems
6. Rainer, R. Kelly and Cegielski, Casey G. (2009). ["Introduction to Information Systems: Enabling and Transforming Business, 3rd Edition"](#)
7. Sommerville, *Software Engineering*, 8th Edition, Pearson Education Limited, Harlow, 2007.

8. The Institute of Electrical and Electronics Engineers, IEEE Standard Glossary of Software Engineering Terminology: IEEE Std. 610.12-1990, The Institute of Electrical and Electronics Engineers, New York, 1990.
9. <https://angular.io/guide/what-is-angular> (pristupano, 15.1.2023)
10. [https://data.fesb.unist.hr/public/news/Srdjana\\_kvalifikacijski\\_rad\\_4%205-2996ede56e.pdf](https://data.fesb.unist.hr/public/news/Srdjana_kvalifikacijski_rad_4%205-2996ede56e.pdf) (pristupano, 13.1.2022)
11. [https://en.wikipedia.org/wiki/Information\\_system](https://en.wikipedia.org/wiki/Information_system) (pristupano, 13.1.2022)
12. <https://www.cis.hr/files/dokumenti/CIS-DOC-2011-06-015.pdf> (pristupano, 14.1.2022)
13. <https://www.codecademy.com/article/back-end-architecture> (pristupano, 18.1.2023)
14. <https://www.visual-paradigm.com/download/community.jsp> (pristupano, 21.1.2023)