

Baze podataka (napredni kurs)

-dokumentacija mini-projekta-

Tema: **Online apoteka (E-pharmacy)**

Projekat radili:

1. Ervin Ciguljin 61/21
2. Nevena Jestrović 69/21
3. Neira Bošković 68/21
4. Nikola Ostojić 114/21
5. Seid Burdzović 07/20

Podgorica 21.5.2024. godine

Sadržaj

APSTRAKT	3
ABSTRACT	3
1. Uvod	4
1.2 Opis projekta	4
1.2.1 Izbor tehnologija	5
2. Analiza zahtjeva	6
2.1 Funkcionalni zahtjevi	6
2.2 Nefunkcionalni zahtjevi	8
2.3 Use Case, Class dijagram, SSD dijagram	9
3. Dizajn sistema	13
3.1 Mockup ekrani	13
3.2 Sequence dijagram, Design Class, ERD dijagram	16
4. Zaduženja članova grupe	19
5. Reference	20

Tabela slika

Slika br. 1 – Use case dijagram.....	9
Slika br. 2 – Class dijagram	10
Slika br. 3 – SSD dijagram 1	11
Slika br. 4 – SSD dijagram 2	12
Slika br. 5 – Mockup ekran (početna stranica)	14
Slika br. 6 – Mockup ekran (korpa proizvoda)	15
Slika br. 7 – Sequence dijagram	16
Slika br. 8 – Design Class dijagram.....	17
Slika br. 9 – ERD dijagram.....	18

APSTRAKT

Ovaj projekat se bavi razvojem *online* apoteke, koja omogućava korisnicima da pregledaju sve dostupne lijekove u apoteci, kao i da naruče potrebne lijekove putem interneta. Korisnici mogu jednostavno unijeti recept, odnosno napraviti porudžbinu za lijekove koje su im potrebni. Farmaceuti imaju mogućnost da pregledaju ove narudžbine, pripreme lijekove i obavijeste pacijente kada su njihovi lijekovi spremni za preuzimanje ili dostavu. Platforma obuhvata širok spektar funkcionalnosti koje omogućavaju jednostavno i brzo pretraživanje dostupnih lijekova, detaljne informacije o svakom proizvodu i mogućnost direktnog naručivanja i praćenja statusa porudžbine.

Ključne riječi: online apoteka, naručivanje lijekova, digitalna farmacija, recepti, farmaceut, dostupnost lijekova, farmaceutska usluga, HTML, CSS, JavaScript, Flask.

ABSTRACT

This project deals with the development of an online pharmacy, which allows users to view all available drugs in the pharmacy, as well as to order the necessary drugs online. Users can simply enter a prescription, or place an order for the medicine they need. Pharmacists have the ability to review these orders, prepare medications, and notify patients when their medications are ready for pickup or delivery. The platform includes a wide range of functionalities that allow a simple and quick search of available medicines, detailed information about each product and the possibility of direct ordering and monitoring of the order status.

Keywords: online pharmacy, ordering medicines, digital pharmacy, prescriptions, pharmacist, availability of medicines, pharmaceutical service, HTML, CSS, JavaScript, Flask.

1. Uvod

U savremenom svijetu, digitalizacija je postala ključan faktor u mnogim industrijama, uključujući i zdravstvenu industriju. Brži razvoj tehnologije donio je promjene u načinu na koji se zdravstvene usluge pružaju, omogućavajući brže i pristupačnije pružanje usluga. [1] Ovaj projekat predstavlja inovativan pristup koji olakšava pristup lijekovima i farmaceutskim uslugama. Cilj ovog projekta je razviti platformu koja će omogućiti korisnicima da pregledaju dostupne lijekove u apoteci, odaberu i naruče potrebne proizvode dok će apotekar (farmaceut) pripremiti lijekove za korisnika (pacijenta).

U apotekama je potrebno da pacijenti fizički odlaze kako bi kupili lijekove, dobili savjet od farmaceuta. Ovaj model nosi sa sobom određene izazove. Radno vrijeme apoteke je ograničeno, što može predstavljati problem za pacijente koji zbog posla ili drugih obaveza ne mogu da posjete apoteku tokom radnog vremena.

Pandemija COVID-19 dodatno je naglasila potrebu za alternativnim rješenjima koja omogućavaju pristup apotekama digitalnim putem. Ovaj period je pokazao koliko su digitalne platforme i sistemi važni za kontinuitet zdravstvene njege. Zdrastveni radnici i farmaceuti su često preopterećeni poslom, pa im *online* sistemi pružaju neophodnu podršku i olakšavaju njihov posao. [2]

1.2 Opis projekta

Cilj ovog projekta je razvoj platforme za *online* apoteku koja će korisnicima pružiti mogućnost da pregledaju dostupne lijekove, naruče potrebne proizvode. Svaki od dostupnih proizvoda treba da sadrži neke osnovne informacije (ime proizvoda, opis proizvoda, dostupan broj proizvoda). Glavna ideja projekta je da korisnici (pacijenti) mogu na jednostavan način pregledati sve dostupne proizvode (lijekove, vitamine i ostalo) i odabrati željene proizvode iz apoteke, odnosno napisati porudžbinu (recept) za te proizvode. Farmaceut može imati uvid u ove porudžbine sortirane po korisnicima, proizvodima i količinama i spremati ih za pacijenta.

Kroz rad će biti obuhvaćen niz radnji potrebnih za kreiranje ovakvog sistema (*web* aplikacije) koja treba da zadovolji ove uslove i funkcionalnosti. Osim toga, u radu će biti pomenute tehnologije koje su korišćene prilikom izrade ove *web* aplikacije kao i neki od glavnih slučajeva korišćenja u sistemu. Za specifikaciju, vizuelizaciju i razvoj ovog sistema biće korišteni **UML dijagrami**. Ovi dijagrami predstavljaju grafičke prikaze dizajna sistema, omogućavajući programerima, arhitektama i ostalim korisnicima u projektu da komuniciraju, planiraju i implementiraju odgovarajuća softverska rešenja. [3]

1.2.1 Izbor tehnologija

Arhitektura ovog projekta podijeljena je na *frontend* i *backend* dio. Za ovaj projekat korišćena je kombinacija **HTML**-a, **CSS**-a, **JavaScript**-a i **Flask**-a. Svaka od ovih tehnologija je ključna za realizaciju ovog projekta. [4]

HTML je osnovni jezik za izradu web stranica i služi za strukturisanje sadržaja na platformi.

CSS se koristi za stilizovanje HTML elemenata na stranici. CSS omogućava kreiranje dizajna po volji korisnika i kreiranje responzivnih stranica za brojne uređaje.

JavaScript je programski jezik koji se koristi za implementaciju dinamičkog ponašanja na *web* stranicama. Pomoću ovog programskog jezika razvija se logika brojnih funkcionalnosti, validiraju forme na stranicama. [4]

Flask je *web framework* namijenjen za Python programski jezik, koji se koristi za izradu *web* aplikacija. Ključne prednosti ovog *framework*-a su jednostavnost i fleksibilnost. Flask je dizajniran da bude lak i jednostavan za korišćenje, ali i dovoljno fleksibilan da podrži složenije aplikacije. [5]

2. Analiza zahtjeva

Analiza zahtjeva je najbitnija faza u razvoju bilo kojeg softverskog rješenja uključujući i razvoj *web* aplikacija. U ovoj fazi identifikuju se potrebe i očekivanja korisnika u vezi sa sistemom koji se razvija. Cilj je da se jasno definišu **funkcionalni zahtjevi**, odnosno šta sistem treba da radi i **nefunkcionalni zahtjevi**, odnosno kako sistem treba da radi. [6]

2.1 Funkcionalni zahtjevi

Funkcionalni zahtjevi služe za opis specifičnih aktivnosti i operacija koje sistem treba da radi, odnosno opis onoga šta sistem treba da omogući korisnicima. Funkcionalni zahtjevi uključuju sve funkcionalnosti koje jedan sistem ima i koji se pružaju korisnicima. Korisniku su ove funkcionalnosti vidljive i ostvaruje interakciju sa njima. [6]

Konkretno, ova *web* aplikacija ima funkcionalne zahtjeve za korisnike (pacijente) i zaposlene radnike u apoteci (farmaceuti ili neko drugi zadužen za operacije na sistemu). [6]

Funkcionalni zahtjevi za korisnike su:

- **Registracija korisnika** – Korisnik ima mogućnost da kreira svoj nalog na online apoteci kako bi mogao da upravlja svojom korpom proizvoda. Ovaj funkcionalni zahtjev je bitan za korisnike kako bi mogli da odaberu željene proizvode, upravljaju količinom proizvoda i ostalo. Apotekar ili admin ovog sistema treba da ima uvid u registrovane korisnike i njihove porudžbine.
- **Prijava korisnika** – Nakon registracije, korisnik ima opciju da se prijavi na svoj nalog nakon što mu on bude odobren. Kada se korisnik prijavi, ima uvid u svoju korpu proizvoda kojom može upravljati.
- **Korpa proizvoda** – Nakon što se korisnik prijavi, ima uvid u korpu proizvoda koja je na početku prazna. U toj korpi, korisnik može dodavati nove proizvode, brisati postojeće, mijenjati količinu proizvoda u skladu sa željama. U korpi, pored naziva proizvoda treba da

stoji broj dostupnih proizvoda u apoteci. Korisnik ne može odabrati ni manje ni više proizvoda od dostupne količine.

Funkcionalni zahtjevi za farmaceut, admin sistema:

- **Informacije o korisnicima** – Admin sistema ima pristup informacijama o registrovanim korisnicima i proizvodima koje su poručili.
- **Potvrda o narudžbi** – Admin sistema ima mogućnost da potvrdi korisnikovu odabranu korpu proizvoda i da spremi željene proizvode za korisnika (pacijenta).
- **Dodavanje proizvoda** – Admin (farmaceut) ima opciju da doda nove proizvode na platformi. Informacije koje neki proizvod ima su: naziv proizvoda, opis proizvoda, sifra proizvoda i dostupna količina proizvoda.
- **Ažuriranje stanja proizvoda** – Admin može ažurirati trenutno stanje proizvoda. Admin (farmaceut) ima mogućnost da ukloni proizvod sa platforme ili doda više proizvoda.

Funkcionalni zahtjevi koje sistem obuhvata:

- **Kreiranje korisnika** – Nakon što korisnik kreira svoj nalog čeka se odobrenje od sistema. Nakon što sistem odobri nalog korisnik može da nastavi dalju radnju (bira željene proizvode, ažurira ih i ostalo).
- **Korpa odabranih proizvoda** – Korisnik bira proizvode i može da upravlja njima. Sistem omogućava korisniku da mijenja količinu proizvoda, dodaje nove proizvode ili briše postojeće. Svaka izmjena koju korisnik napravi se ažurira u bazi podataka.

2.2 Nefunkcionalni zahtjevi

Nefunkcionalni zahtjevi predstavljaju opis onoga kako sistem treba radi. Korisnik nema vidljivu interakciju sa ovim zahtjevima kao što je to slučaj sa funkcionalnim zahtjevima. Nefunkcionalni zahtjevi su bitni za jedan sistem jer oni pokrivaju aspekte kao što su: performanse, sigurnost, pouzdanost, [6]

Primjer nefunkcionalnog zahtjeva je brzina kojom se učitava web stranica ili sigurnost transfera novca putem neke platforme. Ovi zahtjevi obuhvataju čitav niz procesa, kako bi se zadovoljili neki etički i pravni aspekti, ali i kako bi se korisniku osigurala sigurnost i pouzdanost na sistemu koji ima odgovarajuće performanse. [6]

Nefunkcionalni zahtjevi za **E-pharmacy** *web* aplikaciju:

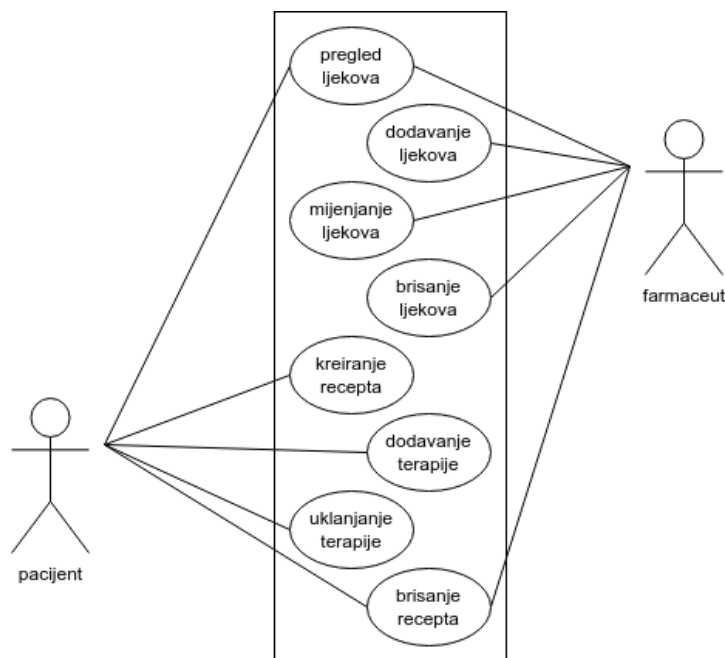
- **Performanse** – Sistem treba da odgovori na korisničke zahtjeve prenosa podataka u što kraćem roku. Važnost ovog nefunkcionalnog zahtjeva je bitna za korisnike. Korisnici očekuju da aplikacija bude brza u izvršavanju zadataka. Prilikom izrade platforme, potrebno je kompresovati slike, video zapise, optimizovati kod kako bi aplikacija bila brža. Aplikacija treba da podrži veliki broj korisnika u istom momentu.
- **Sigurnost** – Ovaj nefunkcionalni zahtjev je izuzetno važan za korisnike i za njihova prava. Ovaj zahtjev je vezan za sva etička pitanja i način prikupljanja podataka od korisnika, naročito kada je riječ o zdravstvenim informacijama kod korisnika. Podaci korisnika moraju biti šifrovani tokom procesa prenosa i skladištenja podataka u bazi, kako bi njihova zaštita bila na većem nivou. Sistem treba da bude zaštićen od uobičajenih sigurnosnih prijetnji. Korisnici treba da se na platformi osjećaju bezbjedno i sigurno.
- **Pouzdanost** – Sistem treba da bude pouzdan kada je u pitanju prikupljanje podataka od korisnika. U ovom slučaju, korisnik unosi informacije o farmaceutskim proizvodima i korisnik očekuje od aplikacije da bude pouzdana i da će njegovi podaci stići do farmaceuta kako bi primili njegovu porudžbinu i spremili je za preuzimanje.
- **Prilagodljivost** – Sistem (aplikacija) treba da bude prilagodljiva za korisnike i njihove potrebe. Platforma treba da obuhvati sve funkcionalnosti koje korisnici očekuju. Takođe,

bitno je da aplikacija ima jednostavan dizajn interfejsa, kako bi korisnicima bilo olakšano korišćenje i snalaženje na ovom sistemu. Sistem treba da bude responzivan, odnosno prilagodljiv na svim uređajima uključujući mobilne i tablet uređaje svih veličina. Takođe, platforma treba da bude prilagodljiva i na svim verzijama dostupnih pretraživača uključujući Chrome, Operu, Mozillu i ostale pretraživače. [6]

2.3 Use Case, Class dijagram, SSD dijagram

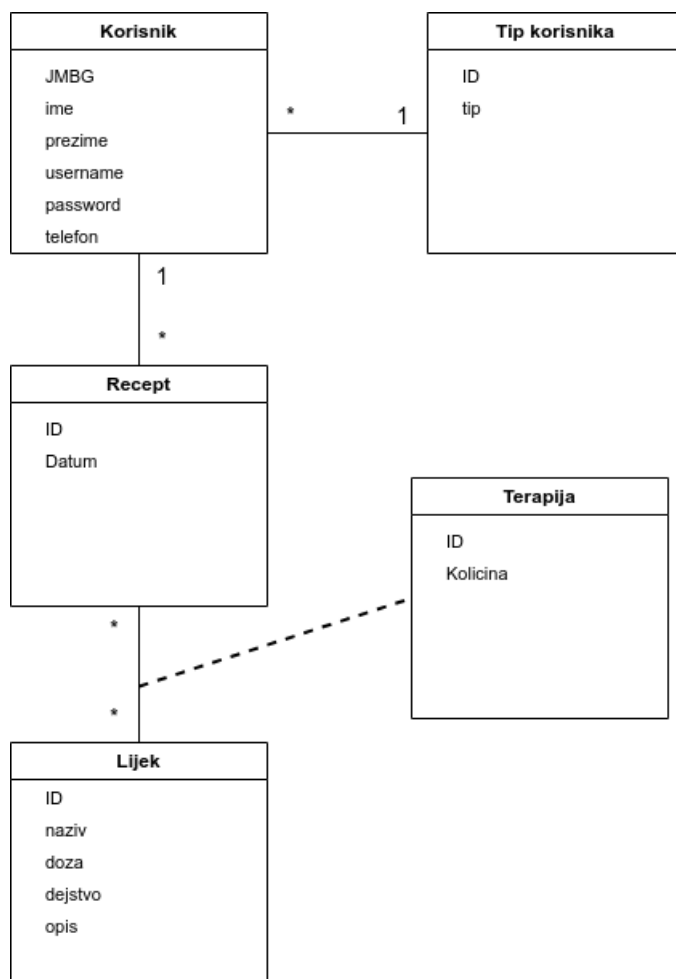
Use Case dijagram služi za grafički prikaz funkcionalnosti sistema iz perspective korisnika. Ovaj dijagram prikazuje različite slučajeve korišćenja (*use cases*) i učesnike (*actors*) koji su u interakciji sa sistemom. [7]

Konkretno, u ovom projektu postoje dva učesnika (pacijent i farmaceut) koji su u interakciji sa sistemom. Funkcionalnosti kojima pacijent može upravljati su: pregled lijekova, kreiranje recepta, dodavanje terapije, uklanjanje terapije i brisanje recepta. Funkcionalnosti kojima farmaceut može upravljati su: pregled lijekova, dodavanje lijekova, mijenjanje lijekova, brisanje lijekova, dodavanje terapije, uklanjanje terapije i brisanje recepta.



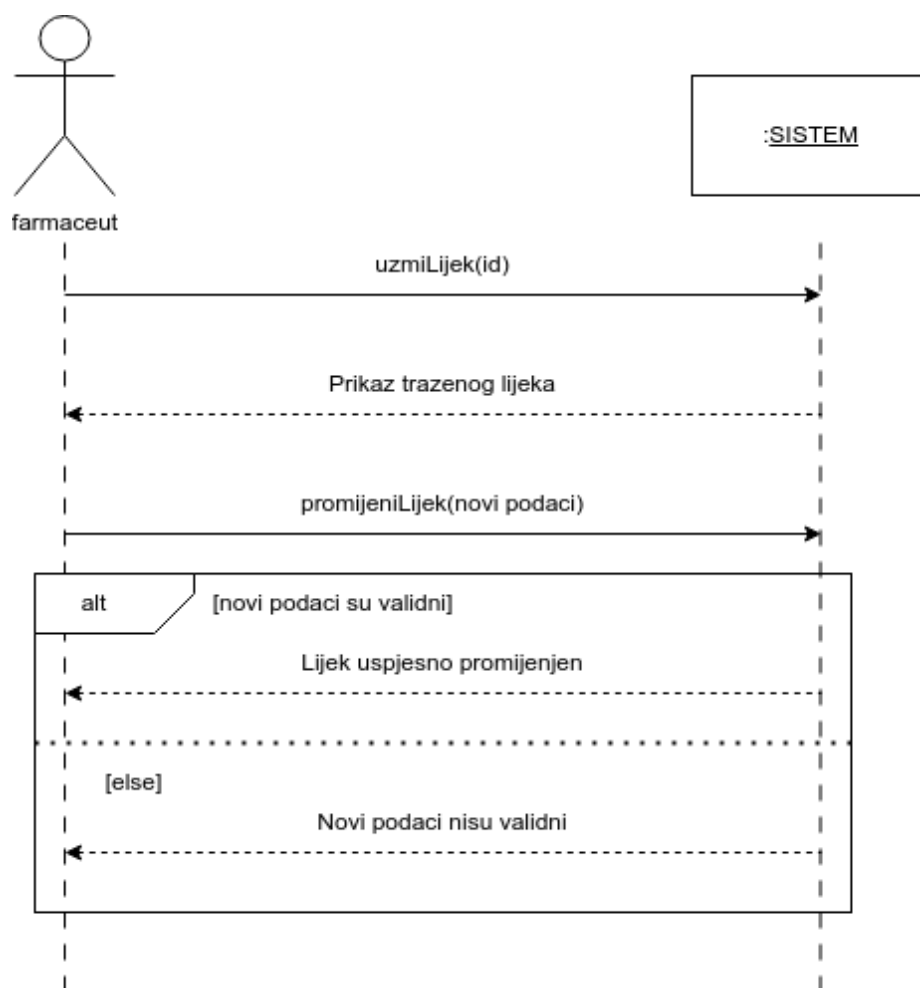
Slika br. 1 – Use case dijagram

Class dijagram je dijagram koji prikazuje strukturu sistema kroz klase, njihove attribute, metode i odnose među njima. Suštinski, ovaj dijagram prikazuje kako su različite klase međusobno povezane i kako komuniciraju. [8]

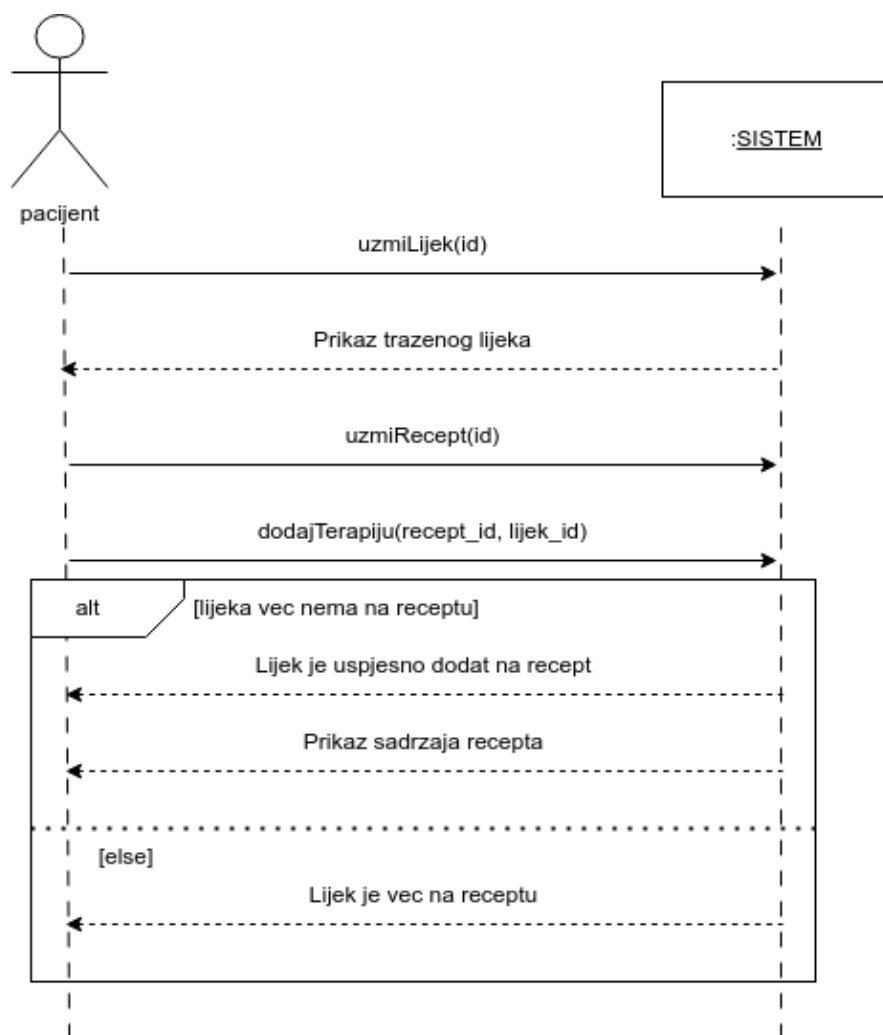


Slika br. 2 – Class dijagram

SSD dijagram je dijagram ponašanja. Ovaj dijagram prikazuje interakciju objekata u sistemu i pravi se za jedan *use case* slučaj. Prvi SSD dijagram (*Slika br. 3*) je vezan za *use case* slučaj “promijeni lijek”, a drugi za *use case* slučaj “dodavanje terapije” (*Slika br. 4*). [9]



Slika br. 3 – SSD dijagram 1



Slika br. 4 – SSD dijagram 2

3. Dizajn sistema

Dizajn sistema treba da obuhvati jednostavan korisnički interfejs sa privlačnim notatama boja kako bi korisnicima bio jednostavan za korišćenje. Kreiranje svakog sistema se vrši na osnovu nekog prototipa dizajna kako treba da izgleda. Prilikom kreiranja ove web aplikacije, pomoću Nicepage alata, kreiran je dizajn web aplikacije, odnosno njenih stranica na kojima se nalaze neke od glavnih funkcionalnosti platforme.

U ovom poglavlju, osim izgleda aplikacije biće opisana i arhitektura sistema. Dizajn sistema često je opisan UML dijagramima koji objašnjavaju koji su glavni slučajevi korišćenja i šta oni treba da rade, odnosno kako treba da rade funkcionalnosti na platformi. Osim toga, prikazani će biti i *mockup* ekrani samog sistema, odnosno *web* aplikacije.

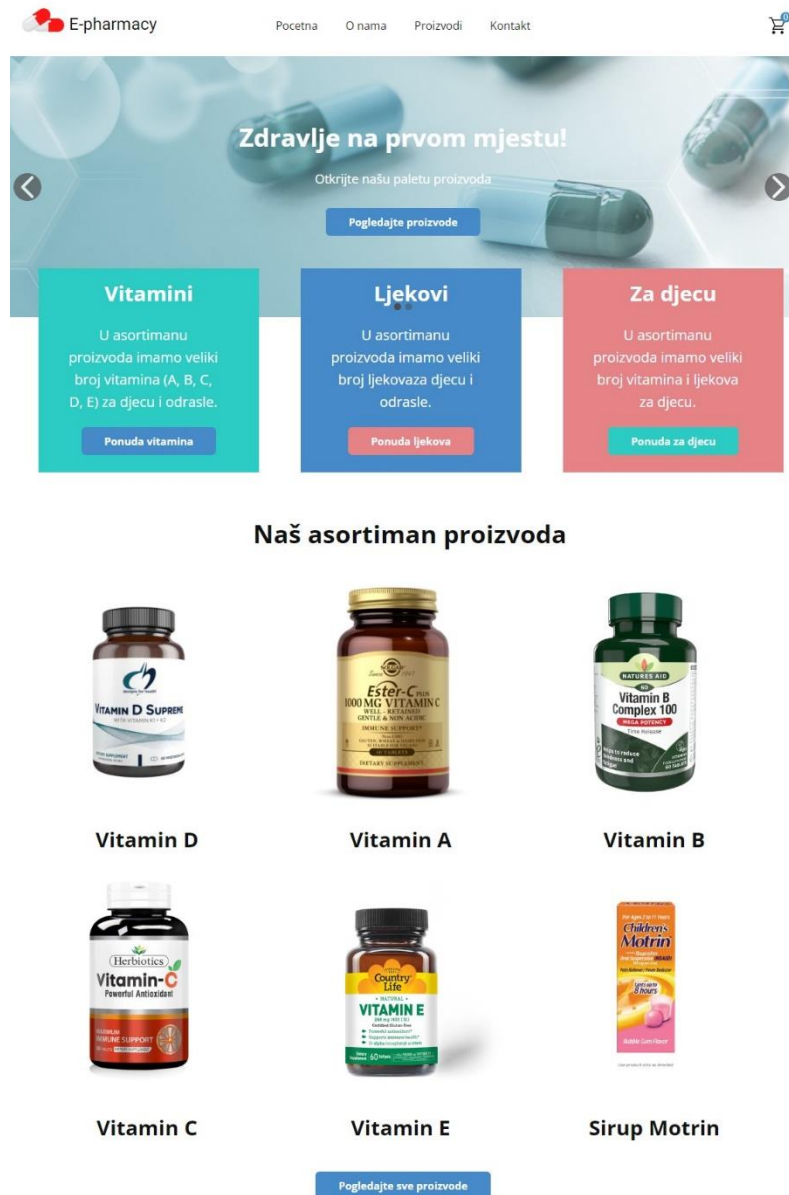
3.1 Mockup ekrani

Mockup ekrani su vizuelni prikaz korisničkog interfejsa dizajna aplikacije ili *web* stranice. Ovi ekrani pružaju pregled izgleda i funkcionalnosti tog sistema, uključujući raspored elemenata, boje, ikonice i druge komponente. [10]

Za kreiranje *mockup* ekrana za ovaj porijekat korišćen je **Nicepage** alat koji korisnicima pruža jednostavno kreiranje *web* stranica. Ovaj alat je namijenjen za kreiranje *web* stranica ali finalni rezultat se može *export*-ovati i kao slika u .jpg formatu, što je izuzetno korisno za kreiranje mockup ekrana. [11]

Ovi ekrani služe *developer*-ima da uz pomoć njih (po uzoru na njih) kreiraju *web* stranicu sa dizajnom sličnim ili istim.

Na prvom primjeru (*Slika br. 5*) je prikazan ekran koji objašnjava ako treba da izgleda početna stranica ove *web* aplikacije. Ova stranica je infomativnog tipa i sadrži osnovne informacije o proizvodima.



Slika br. 5 – Mockup ekran (početna stranica)

Na drugom primjeru (Slika br. 6) se nalazi ekran koji grafički prikazuje kako treba da izgleda čekirani proizvod od strane korisnika iz asortimana dostupnih proizvoda. Klikom na dugme „Sjedeće“ korisnik prelazi na stranicu na kojoj se nalazi korpa odabranih proizvoda. Korisnik može upravljati korpom odabranih proizvoda, brisati proizvode i mijenjati njihovu količinu.



Korpa odabranih proizvoda

Naziv proizvoda	Opis proizvoda	Slika proizvoda	Kolicina odabranih proizvoda	Upravljaj proizvodima
Vitamin C		Vitamin C je jedan od najboljih proizvoda. Mogu ga koristiti i djeca i odrasli.	3	+ - Ukloni
Vitamin E		Vitamin E nije namijenjen za djecu ispod 9 godina. Preporučuje se mladima preko 9 godina.	1	+ - Ukloni
Sirup Motrin		Sirup Motrin je namijenjen isključivo za djecu uzrasta do 9 godina. Idealan za prehladu.	1	+ - Ukloni

[Sledeće](#)

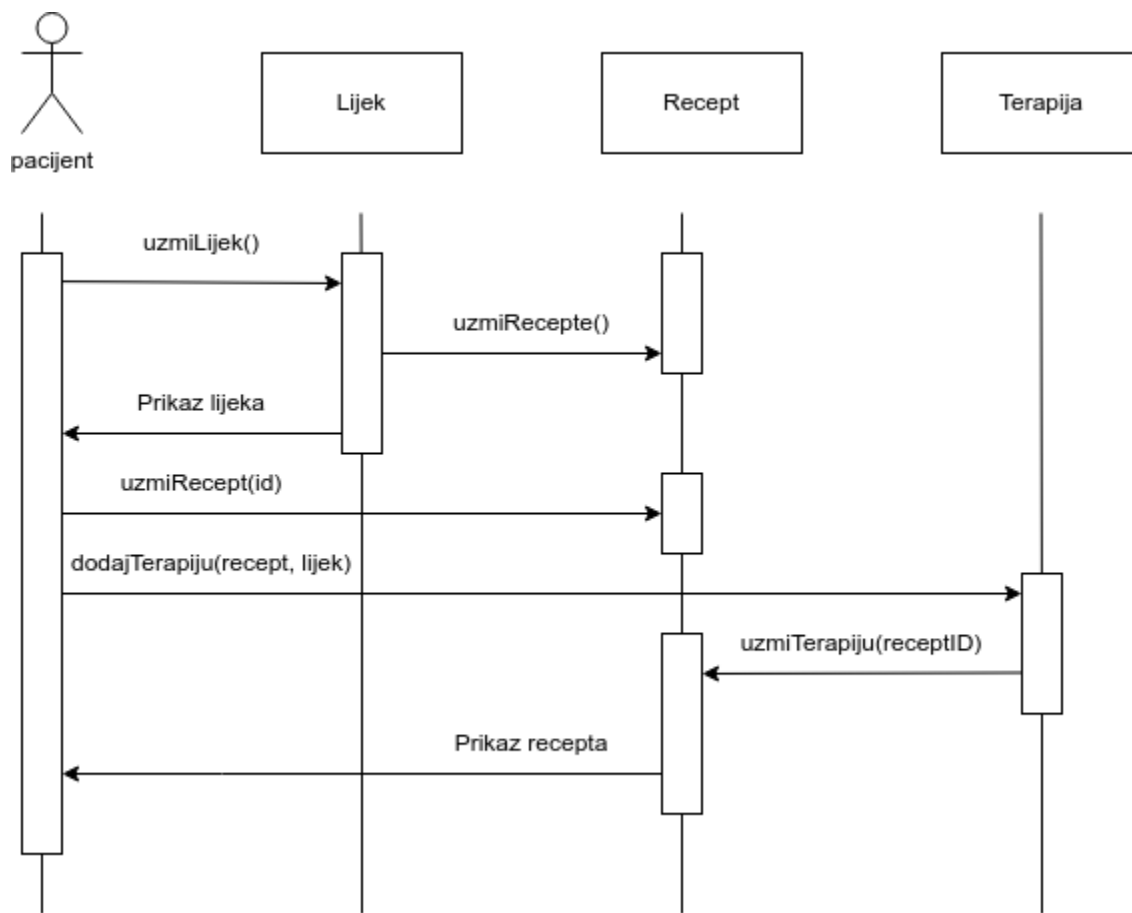
Naš asortiman proizvoda

**Vitamin C**☒ Izabrano**Vitamin E**☒ Izabrano**Sirup Motrin**☒ Izabrano[Sledeće](#)

Slika br. 6 – Mockup ekran (korpa proizvoda)

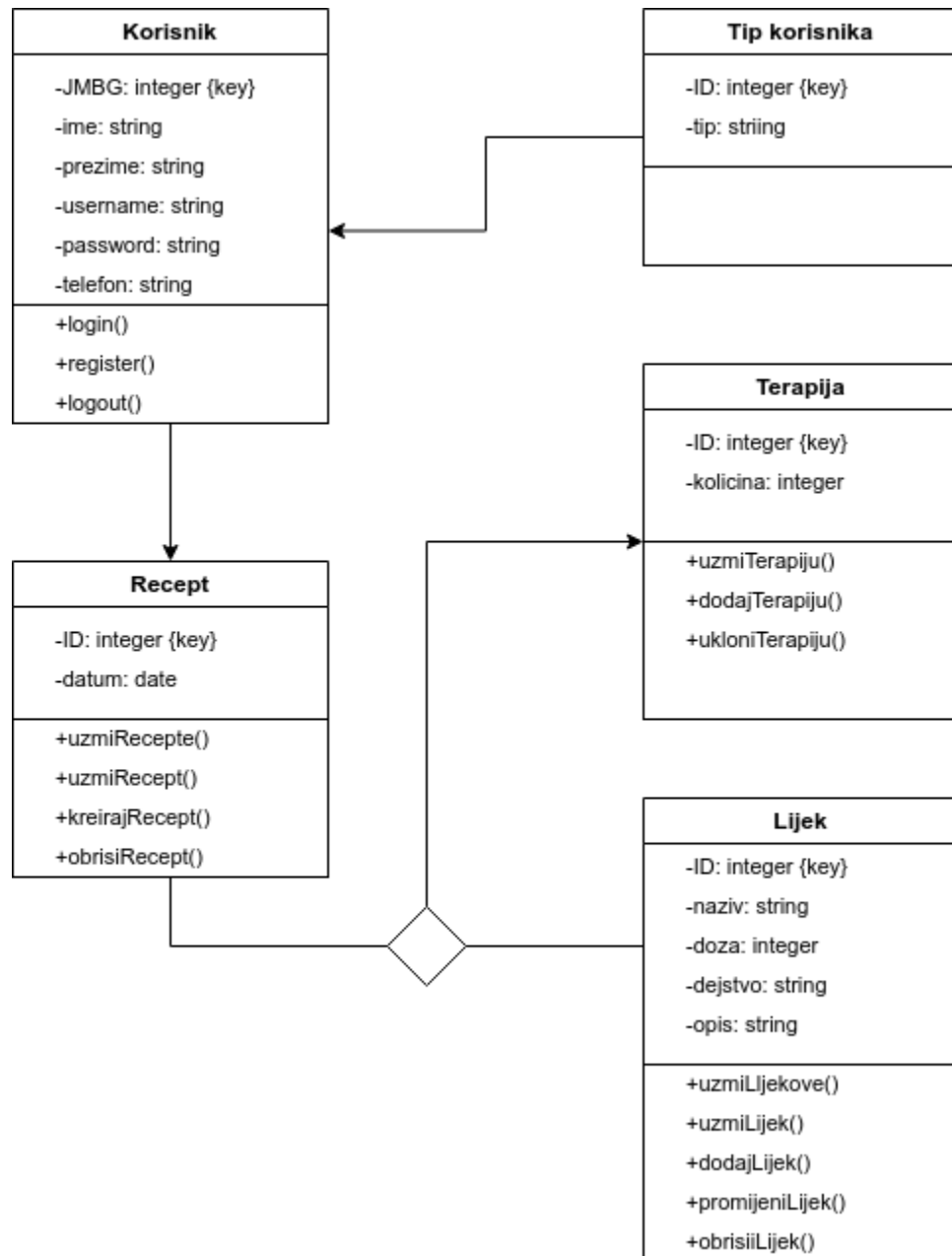
3.2 Sequence diagram, Design Class, ERD diagram

Sequence diagram je dijagram sličan SSD dijagramu, ali u ovom slučaju interakcija nije između korisnika i sistema nego između klasa i korisnika. Skiciran je jedan sequence diagram za *use case* „dodaj terapiju“ (Slika br. 7). [12]



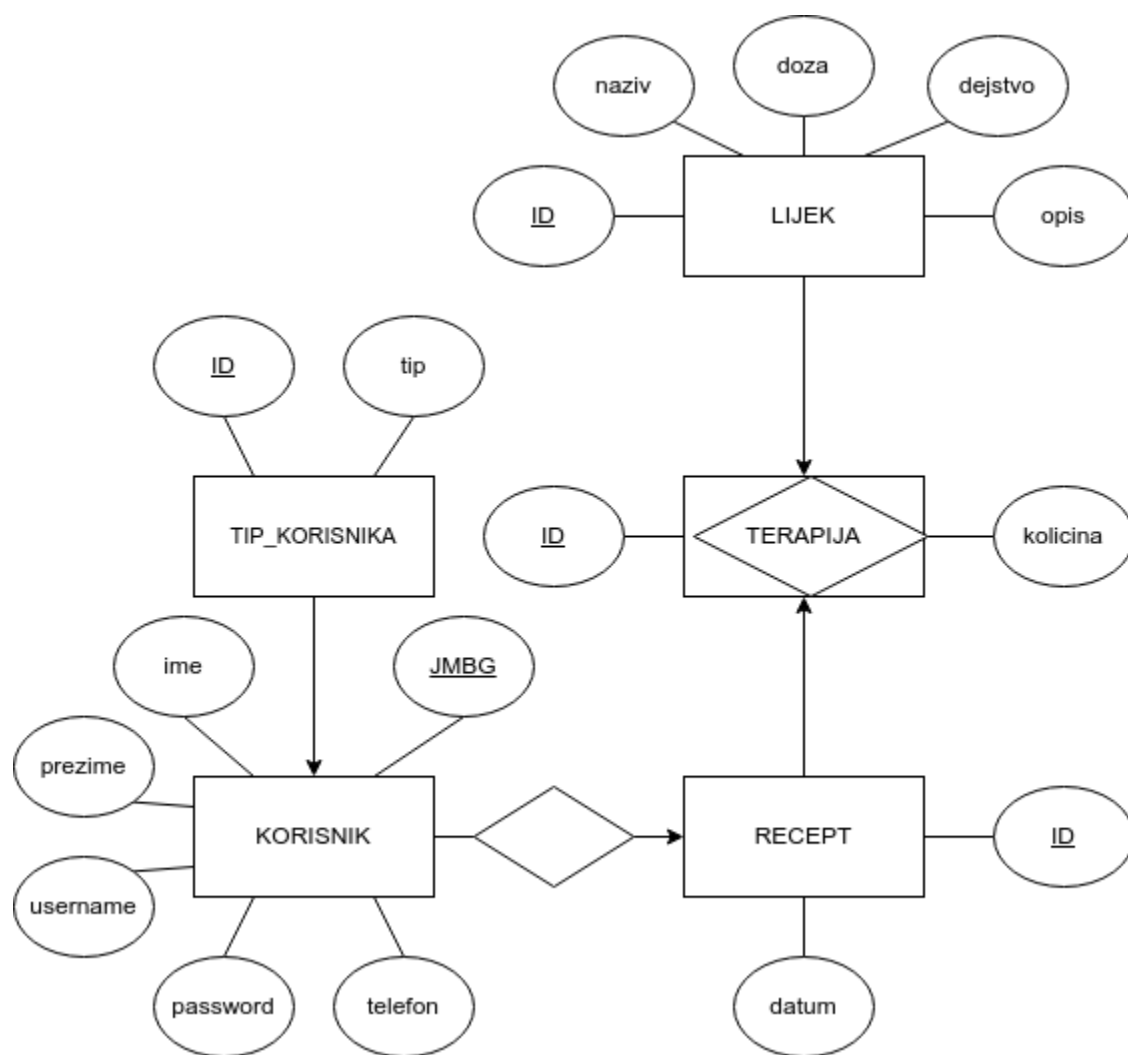
Slika br. 7 – Sequence dijagram

Design class dijagram je nadograđeni class dijagram. Design class dijagram je dijagram koji obuhvata nazive metoda i tipove podataka. Dijagram prikazan na konkretnom slučaju na slici ispod (*Slika br. 8*). [12]



Slika br. 8 – Design Class dijagram

ERD dijagram je dijagram za modelovanje baze podataka koji se koristi za prikazivanje entiteta, njihovih atributa i odnosa među njima. Entiteti su prikazani kao pravougaonici i predstavljaju glavne objekte ili koncepte u bazi podataka (*Slika br. 9*). [12]



Slika br. 9 – ERD dijagram

4. Zaduženja članova grupe

Svaki od članova grupe učestvuje podjednako u izradi ovog projektnog zadatka. Za *frontend* dio zaduženi su Ervin Ciguljin i Nikola Ostojić dok su ostali članovi grupe (Nevena Jestrović, Neira Bošković i Seid Burdzović) zaduženi za *backend* dio ove *web* aplikacije.

Iako su članovi na neki način podijeljeni na *frontend* i *backend* dio, svi članovi grupe podjednako istražuju i međusobno pomažu jedni drugima. Za uspješnu izradu ovog projektnog zadatka timski rad igra ključnu ulogu.

Za kreiranje UML dijagrama zaduženi su bili svi članovi grupe.

- Za *use case* slučajeve korišćenja i class dijagram zadužena je bila Nevena Jestrović.
- Za Design class dijagram zadužen je bio Ervin Ciguljin.
- Za ERD dijagram zadužen je bio Seid Burdzović.
- Za Sequence dijagram zadužen je bio Nikola Ostojić.
- Za SSD dijagrame zadužena je bila Neira Bošković.

5. Reference

1. Päivi Parviainen, Jukka Kääriäinen, Maarit Tihinen and Susanna Teppola, “Tackling the digitalization challenge: how to benefit from digitalization in practice”, March 2017
2. Fernando Almeida, Jose Duarte Santos and Jose Augusto Monteiro, “The Challenges and Opportunities in the Digitalization of Companies in a Post-COVID-19 World”, May 2024
3. Gianna Reggio, Maurizio Leotta, Filippo Ricca and Diego Clerissi, “What are the used UML diagrams?”, University at Genova, Italy
4. Jennifer Niederst Robbins, “Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web graphics”, September 2016
5. Miguel Grinberg, “Flask web development, developing web applications with python”, May 2014
6. Angela Hathaway and Thomas Hathaway, “Functional and Non-Functional Requirements”, 2016
7. Kurt Bitnner and Ian Spence, “Use case modeling”, August 2004
8. Scott W. Ambler, “The Elements of UML 2.0 style”, New York, 2005
9. Sinan si Alhir, “Learning UML”, July 2003
10. Sufyan bin Uzayr, “Mastering UI Mockups and Frameworks”, 2022
11. Nicepage, Available on link: <https://nicepage.com/>
12. Sinan si Alhir, “Learning UML”, July 2003