

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

*Tim 21*

**Tea Jarčov, 39983/11-R**

**Luka Klancir, 39998/11-R**

**Josip Kolarić, 40000/11-R**

**Ivana Koren, 40004/11-R**

**Filip Rafajec, 40080/11-R**

**Appoteka.NET**

**PROJEKT IZ KOLEGIJA PROGRAMSKO INŽENJERSTVO**

**Varaždin, 2014.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Tea Jarčov, 39983/11-R, PS**

**Luka Klancir, 39998/11-R, IS**

**Josip Kolarić, 40000/11-R, PS**

**Ivana Koren, 40004/11-R, IS**

**Filip Rafajec, 40080/11-R, IS**

**Redoviti studenti**

**Preddiplomski studij**

## **Appoteka.NET**

### **PROJEKT IZ KOLEGIJA PROGRAMSKO INŽENJERSTVO**

**Mentor:**

Ivan Švogor, mag. inf.

**Varaždin, srpanj 2014.**

# Sadržaj

<b>1. UVOD .....</b>	<b>2</b>
<b>2. SPECIFIKACIJA ZAHTJEVA.....</b>	<b>3</b>
2.1 ZAHTJEVI VANJSKOG SUČELJA .....	3
2.1.1 Korisnička sučelja.....	3
2.1.2 Sklopovska sučelja .....	3
2.1.3 Programska sučelja.....	3
2.2 SISTEMSKA SVOJSTVA.....	4
2.2.1 Obavijest o minimalnom stanju zaliha lijeka.....	4
2.2.2 Prikaz stanja i lokacije skladištenja lijeka.....	5
2.3 IZVEDBENI ZAHTJEVI .....	6
2.4 OGRANIČENJE DIZAJNA .....	6
2.5 ATRIBUTI PROGRAMSKOG RJEŠENJA .....	7
2.6 OSTALI ZAHTJEVI .....	7
<b>3. PROJEKTNI PLAN .....</b>	<b>7</b>
3.1. PROJEKTNI TIM .....	9
3.2. TERMINSKI PLAN .....	9
3.3. PRORAČUN I BUDŽET .....	12
3.4. PONUDA NARUČITELJU .....	13
<b>4. OPIS DIZAJNA SUSTAVA .....</b>	<b>15</b>
4.1 ARHITEKTURA SUSTAVA .....	15
4.2. DIJAGRAM SLUČAJEVA KORIŠTENJA (USE CASE DIAGRAM) .....	16
4.3. DIJAGRAMI AKTIVNOSTI (ACTIVITY DIAGRAMS).....	17
4.3.1. Prijava.....	17
4.3.2. Naručivanje lijekova.....	17
4.3.3. Evidencija zaposlenika .....	18
4.3.4. Evidencija dobavljača .....	18
4.3.5. Evidencija lijekova.....	19
4.3.6. Evidencija ladica.....	19
4.3.7. Evidencija kategorija .....	20
4.3.8. Izrada i ispis računa.....	20
4.3.9. Provjera klijenta .....	21
4.3.10. Obavijest o minimalnom stanju zaliha lijekova.....	21
<b>5. DIJAGRAM KLASA (CLASS DIAGRAM) .....</b>	<b>22</b>

<b>6.</b>	<b>DIJAGRAM PODATAKA .....</b>	<b>23</b>
-----------	--------------------------------	-----------

# 1. Uvod

Poslovanje svake ljekarne na prvi pogled je vrlo slično poslovanju u maloprodaji s kakvim se susrećemo svakodnevno, ali nakon detaljnijeg proučavanja možemo zaključiti da u poslovanju ljekarni postoje brojna pravila i ograničenja kojima se onemogućavaju zloupotrebe u sustavu zdravstva kao što su primjerice podizanje lijekova s doktorskim receptom kojemu je istekao rok trajanja ili pak pokušaj kupnje lijekova čija je kupnja nedozvoljena bez dokorskog recepta. Ta pravila i ograničenja koje nalazimo u poslovanju ljekarne su uzrok složenijem poslovanju ljekarni. Prilikom stvaranja ideje našeg projekta imali smo prilike promotriti sustav poslovanja ljekarne te način na koji on funkcionira. Tijekom detaljne poslovne analize smo uvidjeli mnoge propuste i greške u poslovanju za koje smatramo da bi ovim projektom mogli korigirati i optimizirati, te programsko rješenje nastalo kao rezultat našeg projekta ponuditi određenoj ljekarni s ciljem poboljšanja poslovanja.

Naša vizija je da ovim projektom uvelike ubrzamo i poboljšamo cjelokupan sustav naručivanja, prodaje kao i evidencije zaliha unutar poslovanja. Tu viziju realizirati ćemo na način da sve bitne aspekte poslovanja jedne ljekarne objedinimo u jedno jedinstveno programsko rješenje, to jest aplikaciju za upravljanje poslovanjem. Poslovna aplikacija koju bi proizveli po završetku ovoga projekta bila bi pregledna i jednostavna za korištenje, te bi također bila prilagođena svim zahtjevima koje korisnik navede prije početka izrade same aplikacije.

Prvobitni cilj ovog projekta je ostvariti pozitivan rezultat iz oba gledišta, tako da krajnji korisnik aplikacije bude zadovoljan isporučenim proizvodom ali isto tako da se za uloženi trud ostvari određeni profit za naš projektni tim. Zadaci i strategije s kojima želimo ostvariti taj cilj su stvoriti aplikaciju upućenu ljekarnama sa svim zamišljenim funkcionalnostima i opcijama koje bi uvelike pomogle prilikom poslovanja svake ljekarne. Samim time, želimo ukloniti neučinkovito i nekvalitetno praćenje stanja i lokacija skladištenja lijekova uvođenjem tehnoloških trendova.

## **2. Specifikacija zahtjeva**

### **2.1 Zahtjevi vanjskog sučelja**

#### **2.1.1 Korisnička sučelja**

Aplikacija je zamišljena kao sustav sa jednim glavnim sučeljem i drugim (izlaznim) sučeljima do kojih se može doći, ovisno o izabranoj radnji, putem glavnog sučelja. Svi korisnici (zaposlenici) pripadati će istoj grupi korisnika, tj. svi su opunomoćeni u radu s aplikacijom te dodatna rješenja u svrhu selekcije neće biti potrebna.

Prilikom implementacije, prvi korisnik koji će biti registriran u aplikaciji biti će ujedno i u mogućnosti dodavati nove korisnike. Pretpostavka je da će prvi korisnik biti voditelj ljekarne te će on kao takav brinuti o drugim korisnicima aplikacije.

Sve mogućnosti i svojstva aplikacije biti će dostupna svim korisnicima iste i će oni moći samostalno upravljati cijelim sustavom (prodaja/narudžba).

Detaljnije o mogućnostima i svojstvima aplikacije nalazi se na točki 4. Opis dizajna sustava.

#### **2.1.2 Sklopovska sučelja**

Sklopovsko sučelje za ovaj sustav podijeljen je u dvije kategorije: klijentska računala i web servis i server. Klijentska računala nalaze se u poslovnicu u kojoj je aplikacija implementirana, tj. na računalima koje zaposlenici koriste.

Web servis i server sa bazom podataka nalazi se ili na kupljenom serveru od providera ili na našem računalu koje smo mi zamislili kao server. Ukoliko je server iznajmljen tada se smanjuje sama sklopovska podrška potrebna za cijeli sustav ali se povećavaju vremenski troškovi za sustav. Ukoliko je server računalo u vlasništvu ljekarne tada se u povećava održavanje cijele infrastrukture.

#### **2.1.3 Programska sučelja**

Programska sučelja podijeljena su u dva dijela: programska rješenja na strani servera i programska rješenja na klijentskoj strani.

Sučelja na strani servera zadužena su za obradu zahtjeva klijentskih računala (aplikacija) u koje ulazi ažuriranje baze podataka, komunikacija za HZZO bazom podataka, detaljno vođenje evidencije o svakom lijeku i arhiviranje poslovnih aktivnosti.

Sučelja na strani klijenta zadužena su za podršku zaposlenicima, tj. njihova namjena je komunikacija sa web servisom sustava i front-end prikaz svih podataka koje sustav sadržava.

## **2.2 Sistemska svojstva**

### **2.2.1 Obavijest o minimalnom stanju zaliha lijeka**

Automatizirano svojstvo koje s obzirom na postavljenu okidač u sustavu obavještava zaposlenika (korisnika) o zalihamu lijeka i služi kao podsjetnik i sigurnosna granica za naručivanje lijeka.

#### **2.2.1.1 Svrha svojstva**

Glavna svrha svojstva je poboljšanje i ubrzanje rada cijelog sustava ljekarne. Svaka ljekarna trguje sa puno lijekova i po svakom lijeku ne drži istu količinu na skladištu. Takav način rada zahtjeva detaljnu evidenciju trenutnog stanja količine lijeka u ljekarni što otežava i oduzima vrijeme zaposlenicima. Svojstvo minimalnog stanja zaliha omogućuje zaposlenicima postavljanje donje granice (količine lijeka), te kada do nje dođe, aplikacija će obavijestiti zaposlenika putem poruke da je lijek došao do donje granice. Daljnja svrha svojstva omogućuje zaposleniku da nakon pročitane obavijesti automatski označi lijek za naručivanje ili donese odluku o promjeni praćenja lijeka.

#### **2.2.1.2 Stimulacijski slijed**

Rezultat ovog svojstva trebao bi voditi zaposlenika do jednostavnijeg i bržeg naručivanja samog lijeka kao i pomoć pri odluci kod njegovog praćenja.

Odmah pri unosu lijeka u sustav ljekarne, zaposlenik će biti u mogućnosti postaviti na kojoj količini će željeti ponovo naručiti lijek (kada želi biti obaviješten o zalihi), što će stvoriti zatvoreni krug praćenja svih lijekova za koje će svojstvo biti definirano.

#### **2.2.1.3 Funkcionalni zahtjevi**

Svojstvo će biti uključeno i vidljivo pri svakom unosu lijeka te je direktno sa samim lijekom i povezano. Jedini dodatni zahtjev za ovo svojstvo je sama akcija zaposlenika, tj. njegovo definiranje svojstva na svakom lijeku pojedinačno.

##### ***2.2.1.3.1 Postojanje lijeka u sustavu***

Kako svaki lijek ima mogućnost praćenja zaliha, glavni zahtjev će biti postojanje lijeka na sustavu, tj. postojanje lijeka u određenoj količini na skladištu. Ukoliko lijek ne postoji ili ga nemamo na skladištu, postavljanje evidencije lijeka nema smisla.

### **2.2.1.3.2 Korisnička inicijativa**

Prilikom unošenja lijeka u sustav po prvi puta, nakon dostavljene narudžbe ili kod promjene odluke o vođenju lijeka, zaposlenik će biti u mogućnosti postaviti granicu (broj) na kojeg će aplikacija reagirati. Zaposlenik je taj koji mora donijeti odluku o granici i postaviti ju.

### **2.2.2 Prikaz stanja i lokacije skladištenja lijeka**

Svojstvo koje omogućuje zaposleniku precizan uvid gdje se lijek nalazi i u kojoj količini.

#### **2.2.2.1 Svrha svojstva**

U sustavu svake ljekarne nalazi se puno lijekova koji se ovisno o svojim količinama i veličini pakiranja u kojem dolaze razlikuju. Iako svaka ljekarna vodi neki svoj sustav evidencije i razvrstavanja, on se uvijek temelji na pamćenju samih zaposlenika. Svojstvo lokacije i stanja lijeka olakšati će zaposlenicima rad i ubrzati izvršavanje zahtjeva kupaca. Zaposlenik će moći u bilo kojem trenutku provjeriti dali lijek postoji i u kojoj količini, te će mu aplikacija također pokazati lokaciju traženog lijeka.

#### **2.2.2.2 Stimulacijski slijed**

Svaki lijek kada se unosi u sustav zahtjeva unos količine i lokaciju na koju će se spremati. Prilikom prodaje lijeka (prije nego se dosegne minimalna granica) zaposlenik ne zna u kojoj se količini lijek nalazi osim fizičkim pretraživanjem i prebrojavanjem lijeka. Nadalje taj problem na sebe nadovezuje još jedan problem, a to je lokacija ili lokacije lijeka. Ovim svojstvom zaposlenik će bez imalo truda saznati potrebne informacije.

#### **2.2.2.3 Funkcionalni zahtjevi**

Svojstvo količine i lokacije ažurirati će se prilikom intervencije zaposlenika. Podaci o količini će se ažurirati prilikom bilo kakve promjene, tj. narudžbe ili prodaje dok će se lokacija mijenjati ili proširivati prilikom narudžbe ili odluke o promjeni lokacije. Za realizaciju ove funkcionalnosti potrebno je definirati „spremište“ po već predviđenim sektorima i prilikom ažuriranja definirati tražene podatke.

##### **2.2.2.3.1 Ažuriranje količine**

Kako bi se mogla pratiti evidencija o lijeku potrebno je prilikom svake promjene ažurirati podatke o lijeku, tj. njegovu količinu. Ažuriranje, ili uvođenje novog lijeka obavlja se prilikom naručivanja lijeka ili prilikom prodaje istog.



#### **2.2.2.3.2 Klasifikacija „spremišta“**

Kako bi smo bili u mogućnosti pratiti lijek, ljekarna zahtjeva jednu razinu organizacijske strukture. U cijelom sustavu ljekarne, lijek je najmanja jedinica. Za ovo svojstvo predviđene su dvije karakteristike koje će jednoznačno definirati svaki lijek i time omogućiti cijelu funkcionalnost. Prva karakteristika je grupa u koju taj lijek pripada, tj. kategorija lijeka. Kategorija će nam omogućiti klasifikaciju lijeka po njegovim svojstvima i djelovanjima i samim time grupiranje lijekova iste kategorije u grupe. Druga karakteristika je „ladica“, tj. fizički prostor na kojem se lijek i/ili grupa lijeka nalaze. Kada je barem jedna karakteristika ispunjena rad sustava postaje znatno precizniji. Kada su obje karakteristike ispunjene zaposlenik zna točno na kojoj se lokaciji lijek nalazi.

### **2.3 Izvedbeni zahtjevi**

Zbog fizičke podjele aplikacije na dva dijela: serverski i klijentski dio, same performanse tražene od sustava se povećavaju.

Od klijentskog računala se očekuje podrška Windows OS-a sa barem 2gb radne memorije, i 120gb statične memorije. Kako se sva radnja odvija na web servisu, tj. podaci putuju preko Internet veze, najveći zahtjev ovog sustava je kvalitetna Internet veza. U slučaju proširenja sustava zahtjevi za vezom proporcionalno rastu.

Web servis i server bi trebali imati snagu proporcionalnu sustavu kojeg podržavaju. Ukoliko servis obrađuje podatke samo jedne podružnice na jednom do dva računala performanse neće trebati biti visoke dok se kod većih sustava proporcionalno traži više. Sam server i baza podataka, tj. prostor na kojem se pohranjuju podaci ovisan je direktno o veličini sustava i starosti samog sustava. Ukoliko se podaci o radu arhiviraju trajno tada će s vremenom sustav konstantno rasti. Shodno tome ukoliko se uvede pravilo zastare, sustavu će se optimizirati performanse.

### **2.4 Ograničenje dizajna**

Aplikacija ima svega par ograničenja. Prvo i najvažnije je ograničenje s obzirom na Internet vezu i samim time web servis i server. Aplikacija da bi mogla nesmetano obavljati sve funkcije mora biti povezana na Internet kako bi web servis na koji se nadovezuje baza mogao voditi evidenciju o radu sustava. Drugo ograničenje je zakonsko i to se odnosi na fiskalizaciju. Propisanim zakonom sustav mora imati fiskalizaciju i svoje poslovanje voditi shodno tim propisima.

## 2.5 Atributi programskog rješenja

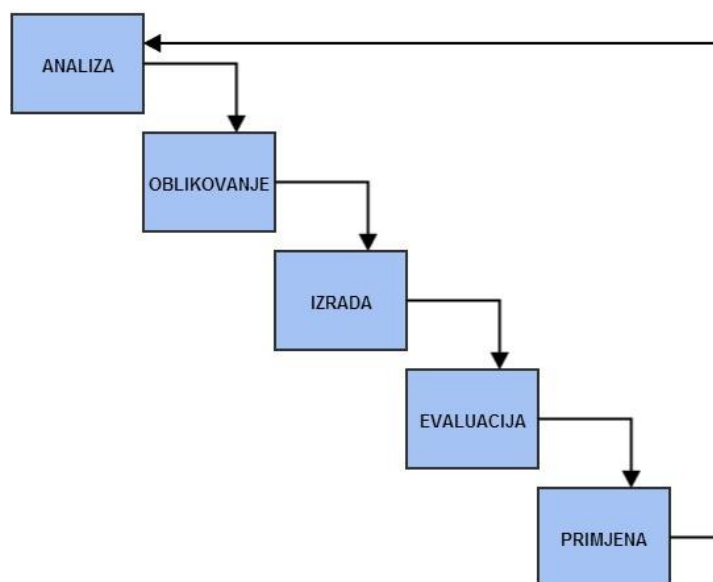
Programsko rješenje ponuditi će svim ljekarskim sustavima brži, precizniji i samim time kvalitetniji rad. Osnovne značajke vezane za sustav kao što su upozorenja kod kritičnih količina zaliha, precizno određivanje količine i lokacije lijeka i kvalitetno uređena baza u svrhu pretraživanja daju aplikaciji na težini i modernosti. Aplikaciju je moguće implementirati u bilo koji sustav što joj daje veliku adaptabilnost i zbog zakonske obveze za fiskalizacijom i Internet vezom, servis koji će obrađivati podatke biti će lako proširiv na više podružnica ukoliko sustav ima potrebu za tim.

## 2.6 Ostali zahtjevi

Aplikacija kao takva može raditi bez ijednog dodatnog svojstva, tj. bez njihovog specificiranja. Međutim bit ove aplikacije jest unaprjeđenje sustava što sa sobom nosi neke posljedice. Iz tih razloga glavni zahtjev ove aplikacije je uključenost zaposlenika sustava u aplikaciju i njeno održavanje.

## 3. Projektni plan

Za izradu projektnog plana izabrali smo vodopadni model, čija struktura je prikazana na sljedećoj slici.



Slika 3.1. Vodopadni model

Kod vodopadnog modela, faze razvoja se realiziraju strogo sekvencijalno, istovremeno za cijeli programski proizvod. Prednost ovog modela je dobra dokumentacija te istovremeni završetak svih podsustava. Dobra dokumentacija je temelj za jednostavno održavanje aplikacija. Isto tako, ovaj pristup daje garanciju da će u konačnom vremenu, doći do zadovoljavajućeg rješenja programskog proizvoda, a samim time i smanjenja rizika od neuspjeha.

Ovaj model smo izabrali iz razloga jer najviše opisuje naš rad na ovom projektu. Kako je i navedeno, prije početka rada bilo je potrebno napraviti kvalitetnu analizu. Analiza je obuhvaćala područje rada, određenu zahtjevnost projekta, specifikacije te prednosti i nedostatke pojedinog područja. Analizom smo stekli uvid u stvarno stanje određenih ljekarni te na temelju toga definirali naše ciljeve, odnosno zadatke koje planiramo provesti kako bi poboljšali njihov sustav poslovanja. Nakon analize slijedi oblikovanje, koje obuhvaća pripremu određenih dijelova projekta, izradu dijagrama i stvaranje osnovnog koncepta za rad. Sljedeća faza je sama izrada gdje se primjenjuje sve iz prethodne dvije faze. Kod faze evaluacije, potrebno je testirati određene dijelove projekta prije same primjene. U našem slučaju, aplikaciju je potrebno ispitati i testirati kako bi se utvrdile eventualne pogreške u radu. Nakon ispravne evaluacije, proizvod, odnosno aplikacija je spremna za primjenu.

Drugim riječima, aplikacija je spremna za korištenje od strane ljekarne. Ukoliko korisnici aplikacije nisu zadovoljni ili imaju dodatnih zahtjeva za promjenama, proces se ponavlja ispočetka počevši ponovno s prvom fazom – analizom.

### 3.1. Projektni tim

Članovi tima navedeni su na početku dokumentacije, podijeljeni u nekoliko uloga:

- Arhitekt – izrađuje SQL skriptu i UML dijagrame na temelju odluka analitičara
- Analitičar – donosi odluke o tome što će se modelirati, programirati i testirati te proučava korisničke zahtjeve i prilagođava ih realnoj implementaciji
- Dizajner – bavi se „vizualnom komunikacijom“; definira izgled i uređuje dokumentaciju, osmišlja i izrađuje dizajn aplikacije, vodi računa o tome da aplikacija bude pregledna, jednostavna i razumljiva korisnicima
- Programer – kreira pseudokod i piše programski kod, uređuje dio dokumentacije koji se odnosi na rješenje programa i testira aplikaciju, održava je i otklanja pogreške
- Voditelj projekta – izrađuje projektni plan te ga ažurira u skladu sa svim promjenama, definira potrebne resurse i raspoređuje ih po dostupnosti, dodjeljuje zadatke članovima tima i prati izvedbu projekta

Navedene uloge nisu specifično određene, već svi članovi tima sudjeluju u svim dijelovima projekta podjednako. Izrada projekta temelji se na dogovoru svih članova i pokušaja da svi utroše podjednako vremena na izradu.

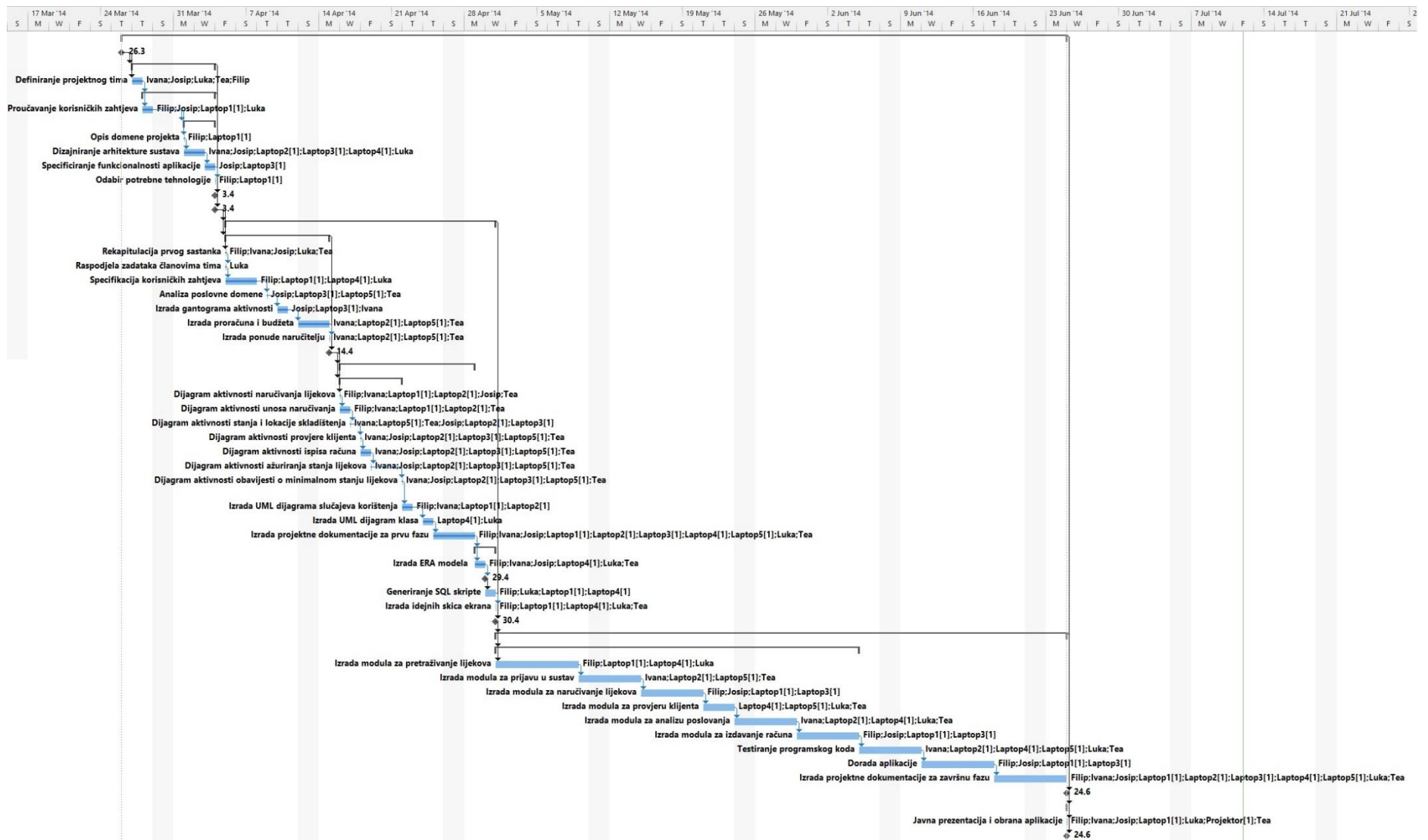
### 3.2. Terminski plan

Terminski plan našeg projekta izrađen je u Microsoftovom alatu MS Project koji nam je omogućio da bilježimo svaku aktivnost koju smo izvršili ili planiramo izvršiti u budućnosti te je na taj način uvelike olakšao planiranje vremena i raspodjelu zadataka u timu. Aktivnosti u terminskom planu su zapisane slijedno kako smo ih izvršavali i u stvarnosti, dok su one aktivnosti koje još uvijek nisu završene zapisane na način da prikazuju plan izvršavanja aktivnosti u budućnosti. Između pojedinih aktivnosti prikazana je i njihova međuovisnost, što znači da određena aktivnost ne može započeti sa izvršavanjem ako je uvjet njenog početka da prethodna aktivnost bude u potpunosti završena. Na svakoj aktivnosti određeno je njeno vrijeme trajanja, dok se u samom terminskom planu nalaze i brojne aktivnosti čije vrijeme trajanja nije određeno nego proračunato kao zbroj trajanja svih aktivnosti koje se izvršavaju kao pod aktivnosti unutar te aktivnosti. Datum početka prve aktivnosti zadan je kao datum kada je projekt

odobren. Datum završetka projekta nije zadan nego izveden, a dobiven je zbrojem trajanja svih aktivnosti u projektu. Aktivnostima su također dodani i resursi koji se koriste ili rade na tim aktivnostima, resurse smo podijelili na rad i materijal. Na slici 3.2. moguće je vidjeti sve aktivnosti koje se nalaze u projektu te podatke koji su vezani za te aktivnosti, dok se na slici 3.3. može vidjeti Ganttov dijagram projekta.

	Task Mode	Task Name	Duration	Start	Finish	Prec	Resource Names
1		▲ Ljekarna.NET	23 days	Wed 26.3.14	Mon 23.6.14		
2	✓	POČETAK PROJEKTA	0 days	Wed 26.3.14	Wed 26.3.14		
3	✓	▲ Planiranje projekta	2,63 days	Wed 26.3.14	Thu 3.4.14	2	
4	✓	Definiranje projektnog tima	4 hrs	Wed 26.3.14	Thu 27.3.14	2	Ivana;Josip;Luka;Tea;Filip
5	✓	▲ Prvi sastanak	2,13 days	Thu 27.3.14	Thu 3.4.14	4	
6	✓	Proučavanje korisničkih zahtjeva	5 hrs	Thu 27.3.14	Fri 28.3.14	4	Filip;Josip;Laptop1[1];Luka
7	✓	▲ Definiranje projekta	1,5 days	Mon 31.3.14	Thu 3.4.14	6	
8	✓	Opis domene projekta	2 hrs	Mon 31.3.14	Mon 31.3.14	6	Filip;Laptop1[1]
9	✓	Dizajniranje arhitekture sustava	5 hrs	Mon 31.3.14	Wed 2.4.14	8	Ivana;Josip;Laptop2[1];Laptop3[1];Laptop4[1];Luka
10	✓	Specificiranje funkcionalnosti aplikacije	4 hrs	Wed 2.4.14	Thu 3.4.14	9	Josip;Laptop3[1]
11	✓	Odabir potrebne tehnologije	1 hr	Thu 3.4.14	Thu 3.4.14	10	Filip;Laptop1[1]
12	✓	ZAVRŠETAK PRVOG SASTANKA	0 days	Thu 3.4.14	Thu 3.4.14	5	
13	✓	ZAVRŠETKA PLANIRANJA PROJEKTA	0 days	Thu 3.4.14	Thu 3.4.14	3	
14	✓	▲ Modeliranje	6,75 days	Fri 4.4.14	Tue 29.4.14	13	
15	✓	▲ Drugi sastanak	2,63 days	Fri 4.4.14	Mon 14.4.14	13	
16	✓	Rekapitulacija prvog sastanka	1 hr	Fri 4.4.14	Fri 4.4.14	13	Filip;Ivana;Josip;Luka;Tea
17	✓	Raspodjela zadataka članovima tima	1 hr	Fri 4.4.14	Fri 4.4.14	16	Luka
18	✓	Specifikacija korisničkih zahtjeva	4 hrs	Fri 4.4.14	Mon 7.4.14	17	Filip;Laptop1[1];Laptop4[1];Luka
19	✓	Analiza poslovne domene	3 hrs	Tue 8.4.14	Tue 8.4.14	18	Josip;Laptop3[1];Laptop5[1];Tea
20	✓	Izrada gantograma aktivnosti	6 hrs	Wed 9.4.14	Thu 10.4.14	19	Josip;Laptop3[1];Ivana
21	✓	Izrada proračuna i budžeta	4 hrs	Fri 11.4.14	Mon 14.4.14	20	Ivana;Laptop2[1];Laptop5[1];Tea
22	✓	Izrada ponude naručitelju	2 hrs	Mon 14.4.14	Mon 14.4.14	21	Ivana;Laptop2[1];Laptop5[1];Tea
23	✓	ZAVRŠETAK DRUGOG SASTANKA	0 days	Mon 14.4.14	Mon 14.4.14	15	
24	✓	▲ Modeliranje problemske domene UML dijagramima	3 days	Tue 15.4.14	Thu 24.4.14	23	
25	✓	▲ Izrada UML dijagrama aktivnosti	1,75 days	Tue 15.4.14	Mon 21.4.14	23	
26	✓	Dijagram aktivnosti naručivanja lijekova	2 hrs	Tue 15.4.14	Tue 15.4.14	23	Filip;Ivana;Laptop1[1];Laptop2[1];Josip;Tea
27	✓	Dijagram aktivnosti unosa naručivanja	2 hrs	Tue 15.4.14	Wed 16.4.14	26	Filip;Ivana;Laptop1[1];Laptop2[1];Tea
28	✓	Dijagram aktivnosti stanja i lokacije skladištenja	2 hrs	Wed 16.4.14	Wed 16.4.14	27	Ivana;Laptop5[1];Tea;Josip;Laptop2[1];Laptop3[1]
29	✓	Dijagram aktivnosti provjere klijenta	2 hrs	Thu 17.4.14	Thu 17.4.14	28	Ivana;Josip;Laptop2[1];Laptop3[1];Laptop5[1];Tea
30	✓	Dijagram aktivnosti ispisa računa	2 hrs	Thu 17.4.14	Fri 18.4.14	29	Ivana;Josip;Laptop2[1];Laptop3[1];Laptop5[1];Tea
31	✓	Dijagram aktivnosti ažuriranja stanja lijekova	2 hrs	Fri 18.4.14	Fri 18.4.14	30	Ivana;Josip;Laptop2[1];Laptop3[1];Laptop5[1];Tea
32	✓	Dijagram aktivnosti obavijesti o minimalnom stanju lijekova	2 hrs	Mon 21.4.14	Mon 21.4.14	31	Ivana;Josip;Laptop2[1];Laptop3[1];Laptop5[1];Tea
33	✓	Izrada UML dijagrama slučajeva korištenja	4 hrs	Mon 21.4.14	Tue 22.4.14	32	Filip;Ivana;Laptop1[1];Laptop2[1]
34	✓	Izrada UML dijagram klasa	4 hrs	Wed 23.4.14	Thu 24.4.14	33	Laptop4[1];Luka
35	✓	Izrada projektne dokumentacije za prvu fazu	2 hrs	Thu 24.4.14	Thu 24.4.14	34	Filip;Ivana;Josip;Laptop1[1];Laptop2[1];Laptop3[1];Laptop4[1];Laptop5[1];Luka;Tea
36	✓	▲ Modeliranje baze podataka	1,13 days	Fri 25.4.14	Tue 29.4.14	35	
37	✓	Izrada ERA modela	2 hrs	Fri 25.4.14	Fri 25.4.14	35	Filip;Ivana;Josip;Laptop4[1];Luka;Tea
38	✓	PRVA PROVJERA	0 days	Tue 29.4.14	Tue 29.4.14	37	
39	✓	Generiranje SQL skripte	2 hrs	Tue 29.4.14	Tue 29.4.14	38	Filip;Luka;Laptop1[1];Laptop4[1]
40	✓	Izrada idejnih skica ekrana	1 hr	Tue 29.4.14	Tue 29.4.14	39	Filip;Laptop1[1];Laptop4[1];Luka;Tea
41	✓	ZAVRŠETAK MODELIRANJA	0 days	Tue 29.4.14	Tue 29.4.14	14	
42	✓	▲ Realizacija programskog rješenja	13,5 days	Wed 30.4.14	Fri 20.6.14	41	
43	✓	▲ Izrada aplikacije u .NET	8,75 days	Wed 30.4.14	Tue 3.6.14	41	
44	✓	Izrada modula za pretraživanje lijekova	14 hrs	Wed 30.4.14	Wed 7.5.14	41	Filip;Laptop1[1];Laptop4[1];Luka
45	✓	Izrada modula za prijavu u sustav	12 hrs	Wed 7.5.14	Tue 13.5.14	44	Ivana;Laptop2[1];Laptop5[1];Tea
46	✓	Izrada modula za naručivanje lijekova	12 hrs	Tue 13.5.14	Mon 19.5.14	45	Filip;Josip;Laptop1[1];Laptop3[1]
47	✓	Izrada modula za provjeru klijenta	10 hrs	Mon 19.5.14	Thu 22.5.14	46	Laptop4[1];Laptop5[1];Luka;Tea
48	✓	Izrada modula za analizu poslovanja	10 hrs	Fri 23.5.14	Wed 28.5.14	47	Ivana;Laptop2[1];Laptop4[1];Luka;Tea
49	✓	Izrada modula za izdavanje računa	10 hrs	Thu 29.5.14	Tue 3.6.14	48	Filip;Josip;Laptop1[1];Laptop3[1]
50	✓	Testiranje programskog koda	10 hrs	Wed 4.6.14	Mon 9.6.14	49	Ivana;Laptop2[1];Laptop4[1];Laptop5[1];Luka;Tea
51	✓	Dorada aplikacije	12 hrs	Tue 10.6.14	Fri 13.6.14	50	Filip;Josip;Laptop1[1];Laptop3[1]
52	✓	Izrada projektne dokumentacije za završnu fazu	10 hrs	Mon 16.6.14	Fri 20.6.14	51	Filip;Ivana;Josip;Laptop1[1];Laptop2[1];Laptop3[1];Laptop4[1];Laptop5[1];Luka;Tea
53	✓	REALIZACIJA ZAVRŠENA	0 days	Fri 20.6.14	Fri 20.6.14	42	
54	✓	▲ Isporuka programskog rješenja	0,13 days	Mon 23.6.14	Mon 23.6.14	53	
55		Javna prezentacija i obrana aplikacije	1 hr	Mon 23.6.14	Mon 23.6.14	53	Filip;Ivana;Josip;Laptop1[1];Luka;Projektor[1];Tea
56		PROJEKT ZAVRŠEN	0 days	Mon 23.6.14	Mon 23.6.14	1	

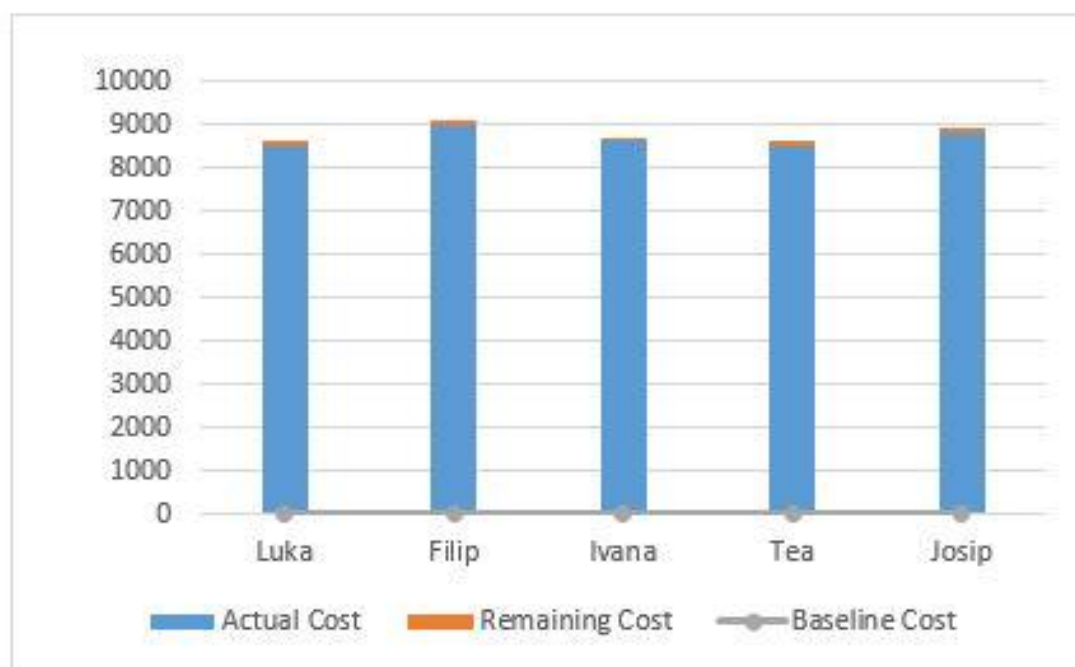
**Slika 3.2. Terminski plan aktivnosti**



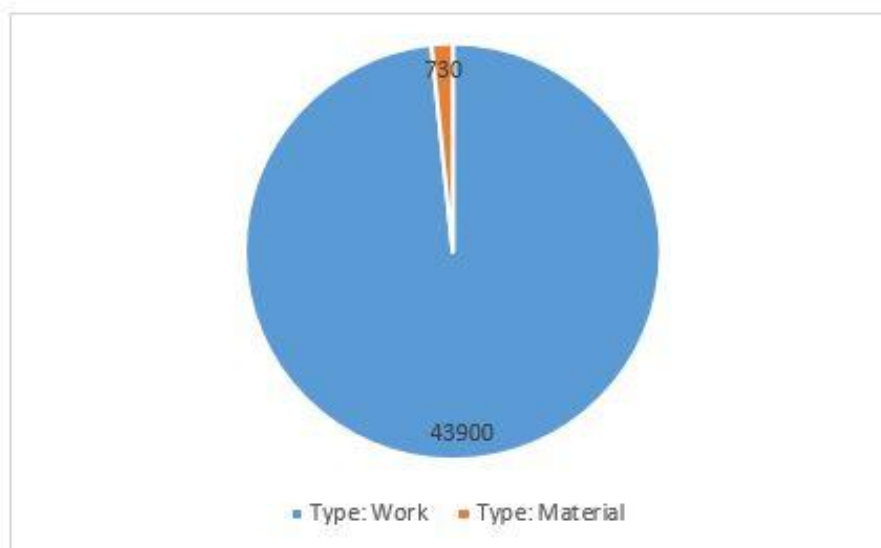
Slika 3.3. Ganttov dijagram

### 3.3. Proračun i budžet

Proračun i budžet projekta dobiveni su također uz pomoć Microsoftovog alata MS Project koji pruža mogućnost kreiranja izvještaja koristeći podatke iz unesenih aktivnosti. Proračun i budžet našeg projekta dobiven je na način da smo odredili koliki su troškovi pojedinih resursa s obzirom na vrijeme njihova korištenja. Tako smo primjerice za plaću ljudskih resursa odredili iznos od 100 kuna po satu, dok je trošak korištenja materijalnih resursa 10 kuna po satu za svaki od laptopa koji se koriste u projektu te za projektor koji je potreban za prezentaciju projekta. Cijena troška ljudskih resursa po satu je određena sukladno sa cijenama satnice ljudskih resursa u poduzećima koja se bave izradom programskih rješenja. Na slikama 3.4., 3.5. i 3.6. prikazani su troškovi resursa u našem projektu.



Slika 3.4. Status troškova za ljudske resurse



**Slika 3.5. Podjela troškova s obzirom na tip resursa**

Name	Actual Work	Actual Cost	Standard Rate	Cost	Work
Luka	85 hrs	8.500,00 kn	100,00 kn/hr	8.600,00 kn	86 hrs
Filip	90 hrs	9.000,00 kn	100,00 kn/hr	9.100,00 kn	91 hrs
Ivana	86 hrs	8.600,00 kn	100,00 kn/hr	8.700,00 kn	87 hrs
Tea	85 hrs	8.500,00 kn	100,00 kn/hr	8.600,00 kn	86 hrs
Josip	88 hrs	8.800,00 kn	100,00 kn/hr	8.900,00 kn	89 hrs

**Slika 3.6. Detalji troškova za ljudske resurse**

### 3.4. Ponuda naručitelju

Nakon što smo definirali terminski plan i proračun projekta, izradili smo ponudu za naručitelja, koji je u našem slučaju ljekarna. U ponudi je navedena okvirna cijena aplikacije, module i usluge koje obuhvaća aplikacija.



*Tim 21*

*Ludbreška 3*

*42 000 Varaždin*

*Sveučilište u Zagrebu*

*Fakultet organizacije i informatike*

*Pavlinska 2*

*42 000 Varaždin*

U Varaždinu, 26. ožujka 2014.

## **PONUDA**

Ponudili bismo Vam aplikaciju koja je trenutno u izradi i za koju smatramo da bi Vam bila veoma korisna te omogućila brže i kvalitetnije poslovanje. Neke od glavnih specifikacija koje bi aplikacija omogućavala:

- Mogućnost narudžbe određenih proizvoda
- Provjeru stanja i lokacije skladištenja pojedinog proizvoda
- Obavijest o minimalnom stanju zaliha
- Prilagodbu cijene klijentima s dopunskim osiguranjem

Nakon izvedbe plana projekta, predviđena cijena aplikacije je 65.000 kn, a u cijenu su uključeni troškovi analize poslovnog slučaja, izrada svih potrebnih dijagrama i baze podataka te sama realizacija i izrada aplikacije. Predviđeno trajanje projekta je tri mjeseca.

Za sva pitanja možete nam se obratiti putem e-maila ili možemo ugovoriti sastanak na kojem bi se rješavale nedoumice vezane uz projekt.

Spremni smo na dogovore i radujemo se budućoj suradnji.

S poštovanjem,

*Tea Jarčov (tjarcov@foi.hr)*

*Luka Klancir (lklancir@foi.hr)*

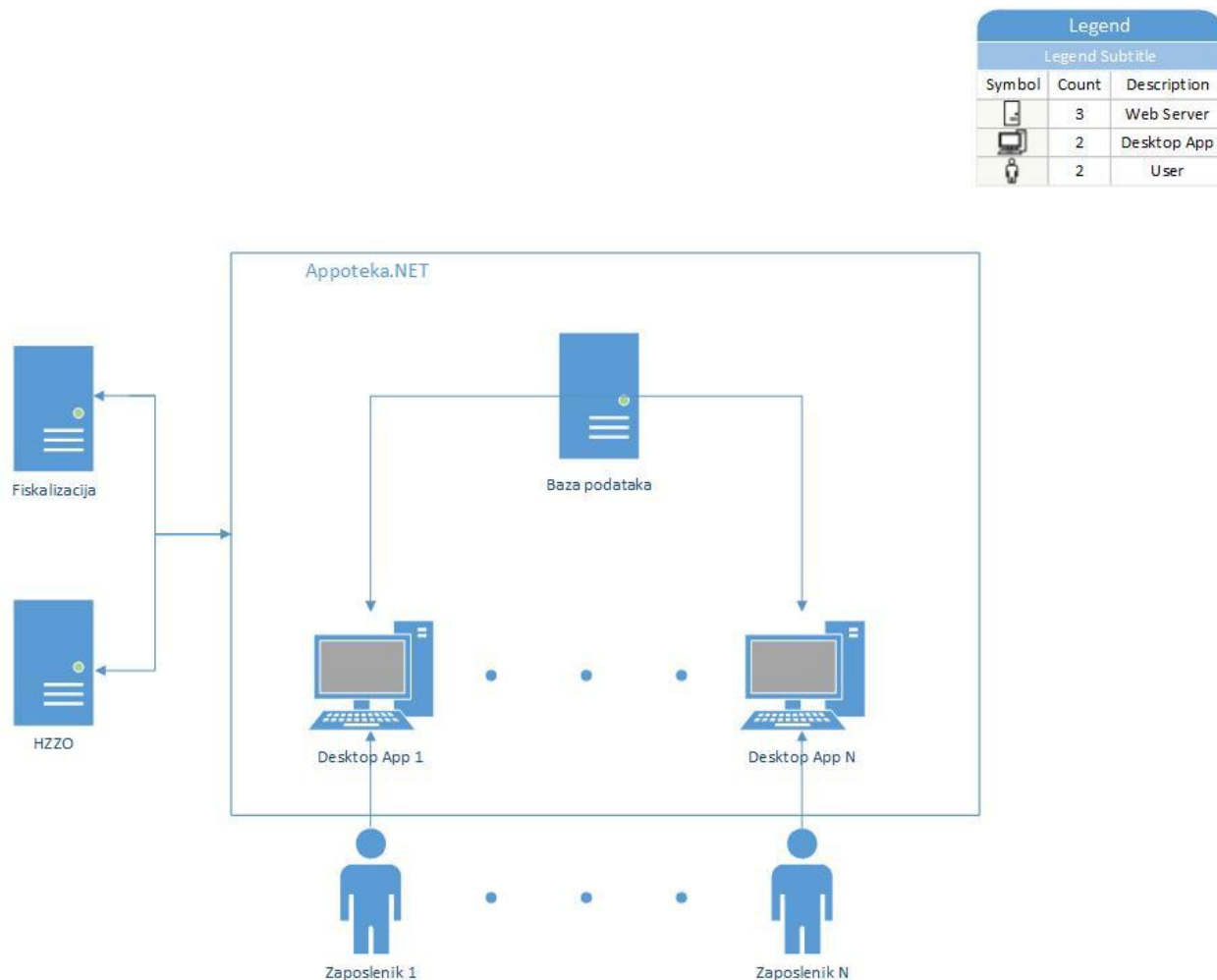
*Josip Kolarić (jkolaric@foi.hr)*

*Ivana Koren (ikoren@foi.hr)*

*Filip Rafajec (frafajec@foi.hr)*

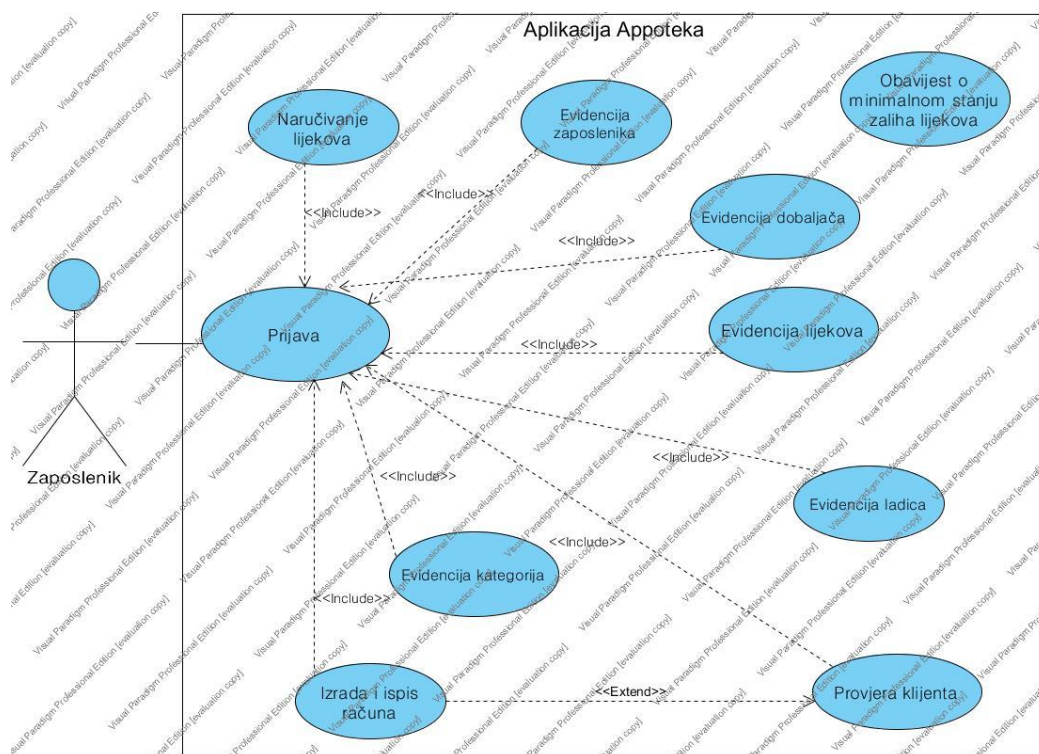
## 4. Opis dizajna sustava

### 4.1 Arhitektura sustava



Sljedeća arhitektura sustava prikazuje našu aplikaciju Appoteka.NET koja je instalirana na desktop računalu te povezana sa web bazom podataka. Odlučili smo se za bazu podataka na web serveru zbog veće mogućnosti skalabilnosti i integracije samog sustava unutar grupacija određenih ljekarni. Također sama aplikacija komunicira sa bazom podataka HZZO zbog preuzimanja prepisanih recepata naših klijenata. Zbog potrebe fiskalnog računa aplikacija po potrebi komunicira sa poreznom upravom.

## 4.2. Dijagram slučajeve korištenja (Use Case Diagram)



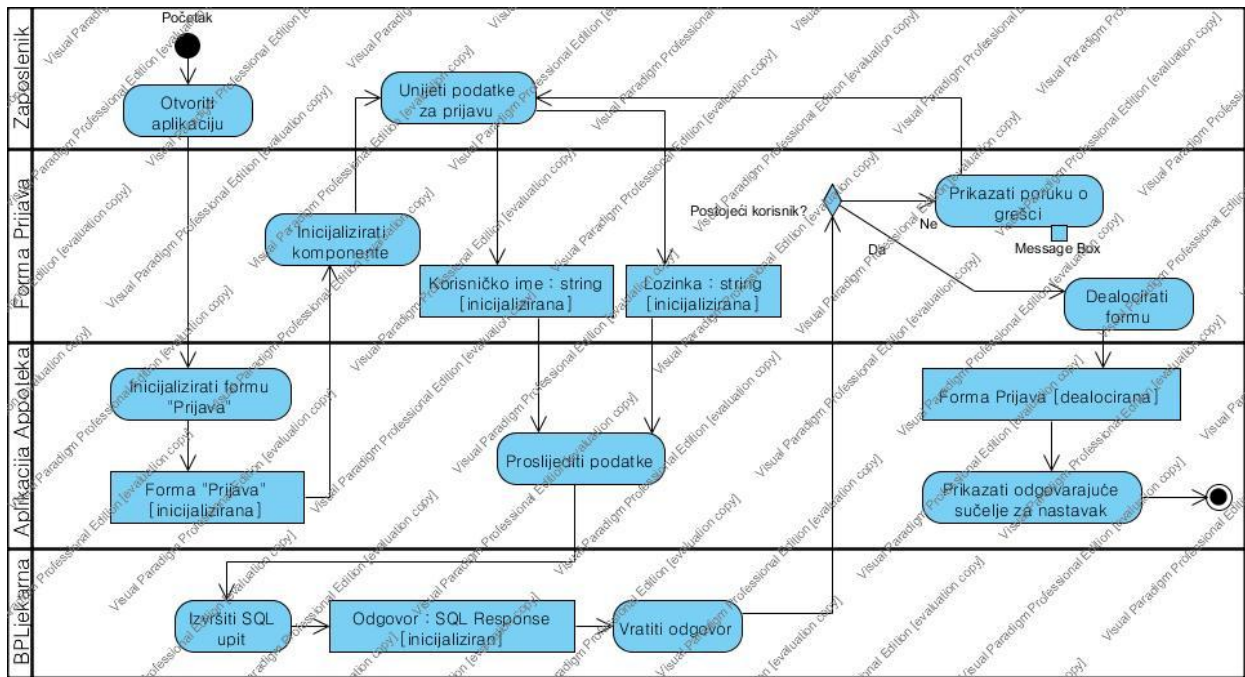
Slika 4.2. Dijagram slučajeve korištenja

Primarni i jedini učesnik u našem sustavu je zaposlenik, jer samo on koristi i ima pristup aplikaciji.

Kod pristupanja aplikaciji *Apoteka*, zaposlenik mora izvršiti *Prijavu* u sustav. Kod prijave on unosi svoje korisničko ime i lozinku kako bi mogao pristupiti ostalim slučajevima aplikacije.

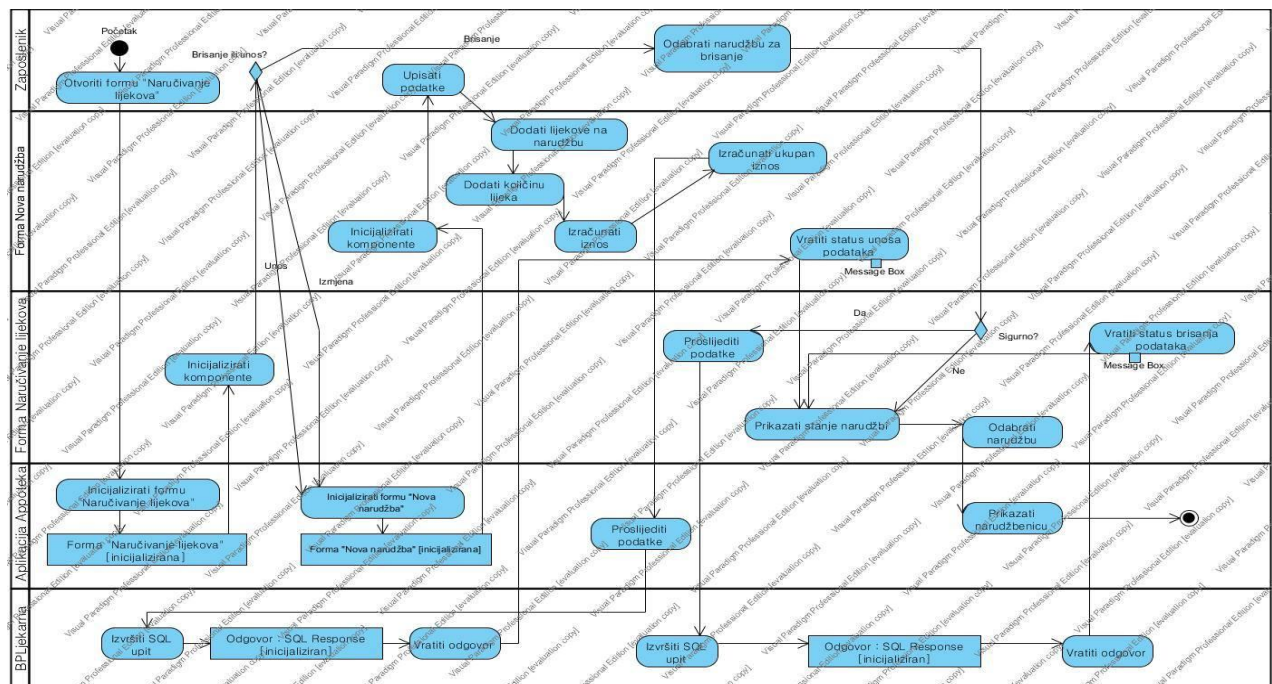
## 4.3. Dijagrami aktivnosti (Activity Diagrams)

### 4.3.1. Prijava



Slika 4.3.1 Dijagram aktivnosti *Prijava*

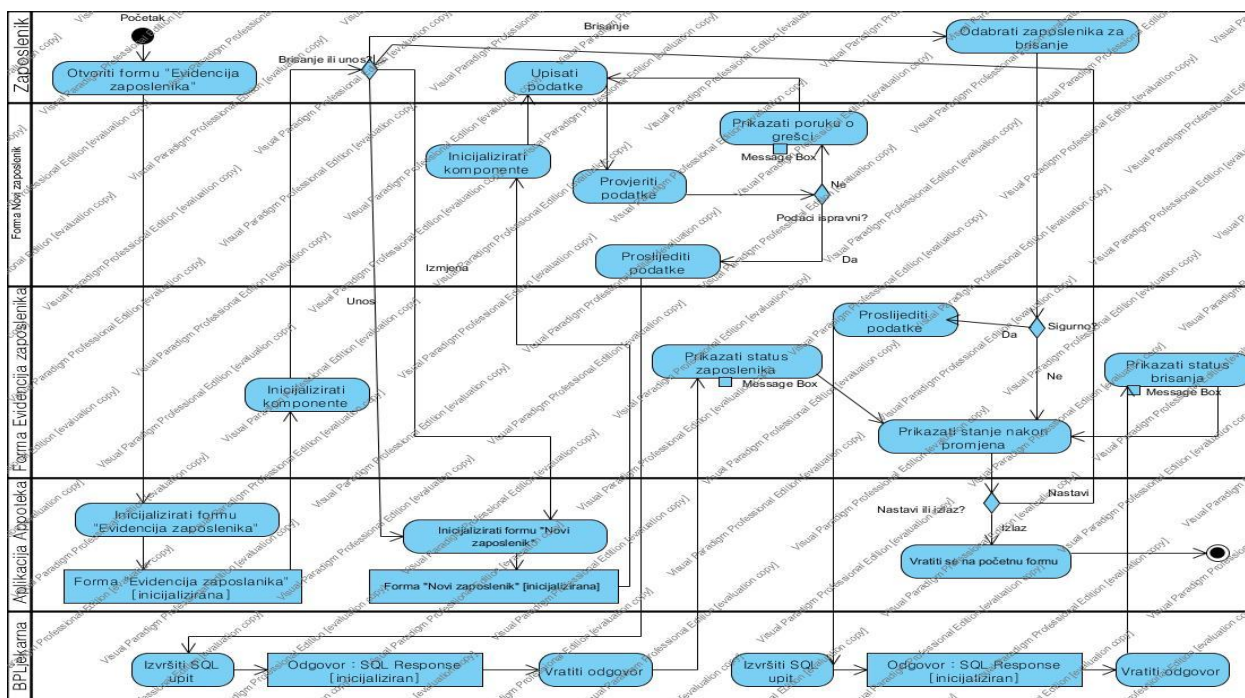
### 4.3.2. Naručivanje lijekova



Slika 4.3.2 Dijagram aktivnosti *Naručivanje lijekova*

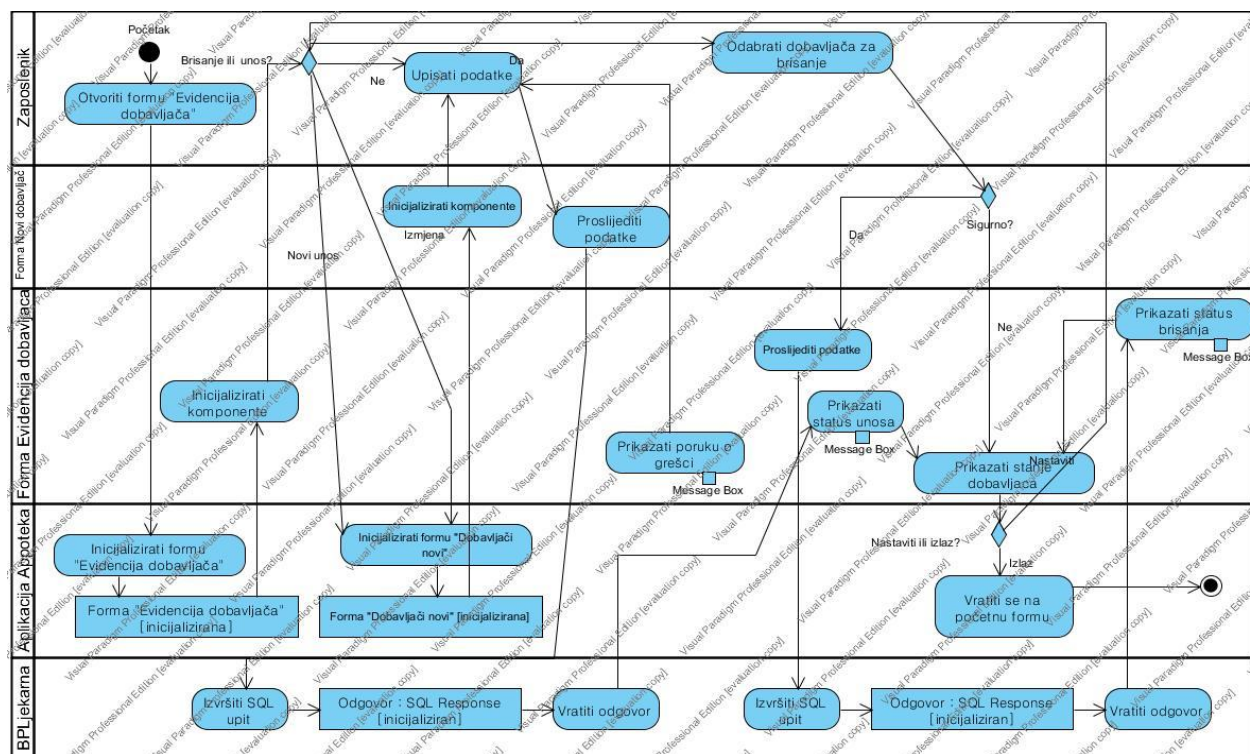


### 4.3.3. Evidencija zaposlenika



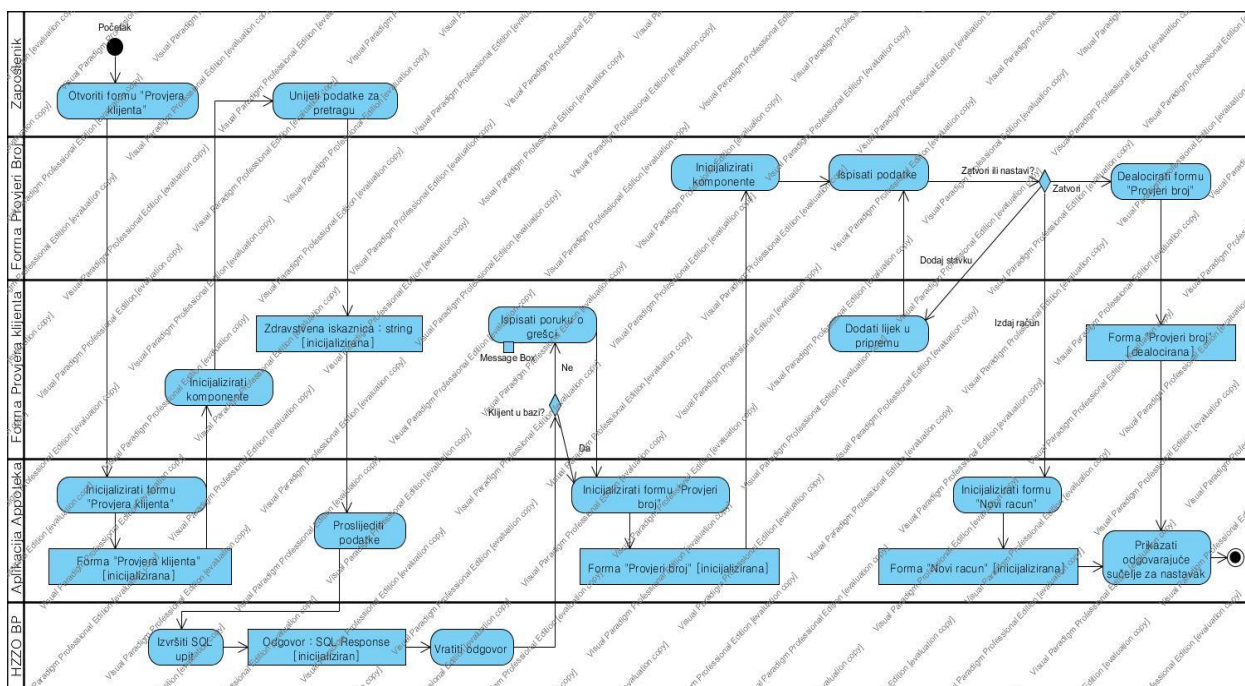
Slika 4.3.3 Dijagram aktivnosti *Evidencija zaposlenika*

### 4.3.4. Evidencija dobavljača



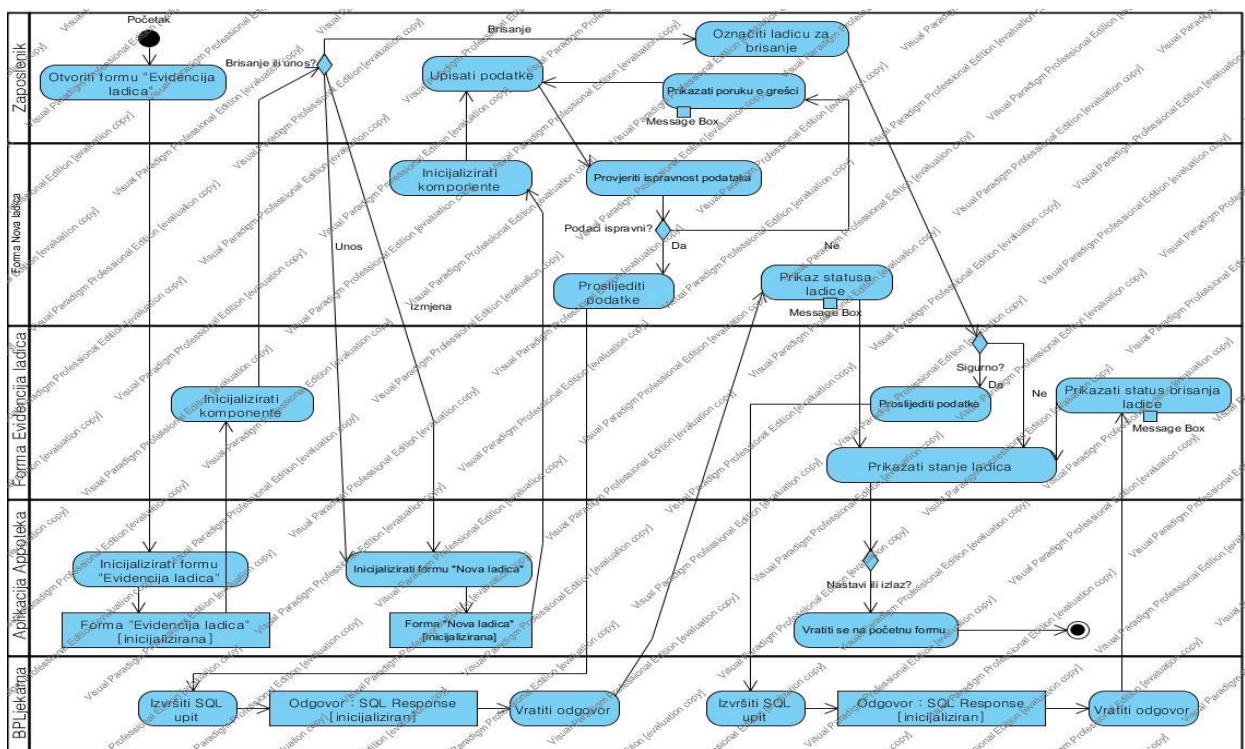
Slika 4.3.4 Dijagram aktivnosti *Evidencija dobavljača*

### 4.3.5. Evidencija lijekova



Slika 4.3.4 Dijagram aktivnosti *Provjera klijenta*

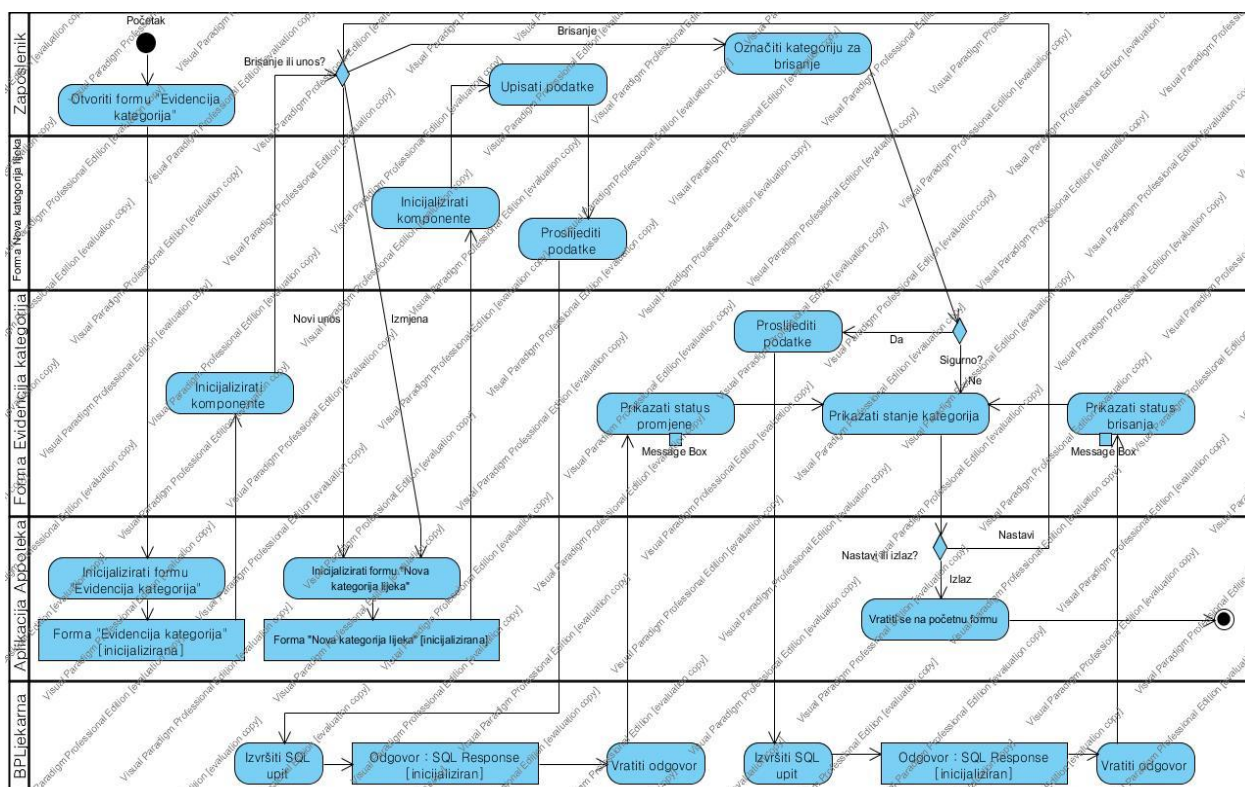
### 4.3.6. Evidencija ladic



Slika 4.3.6 Dijagram aktivnosti *Evidencija ladic*

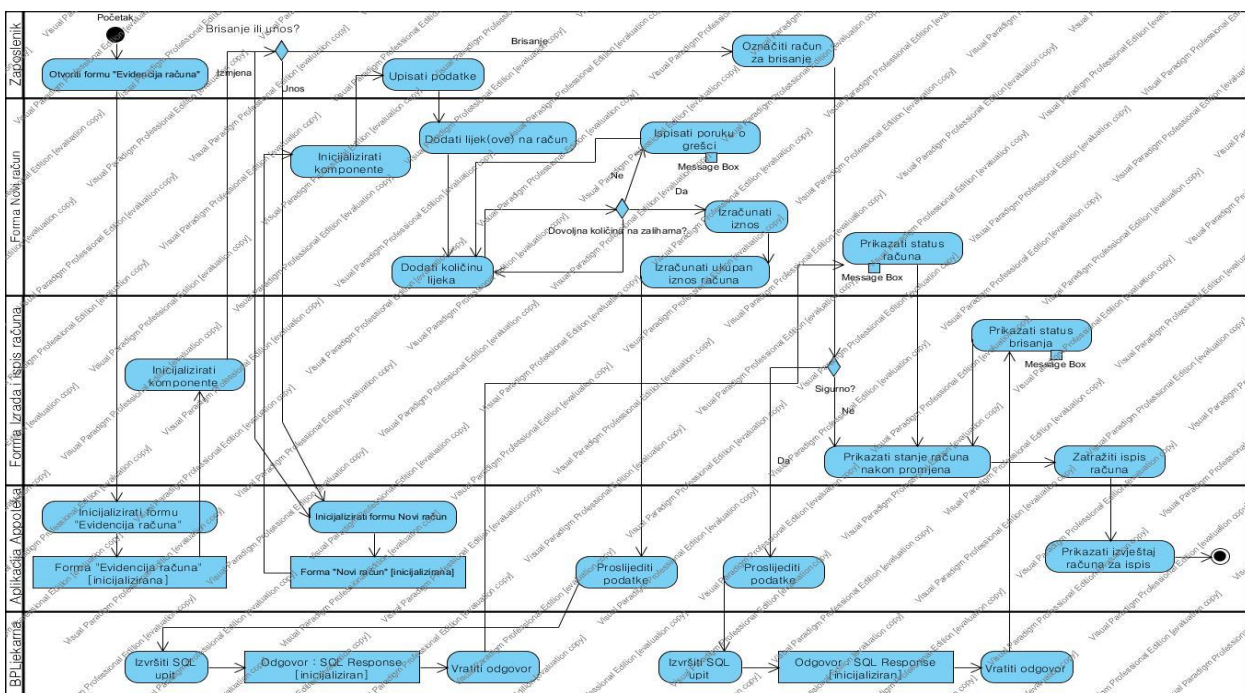


### 4.3.7. Evidencija kategorija



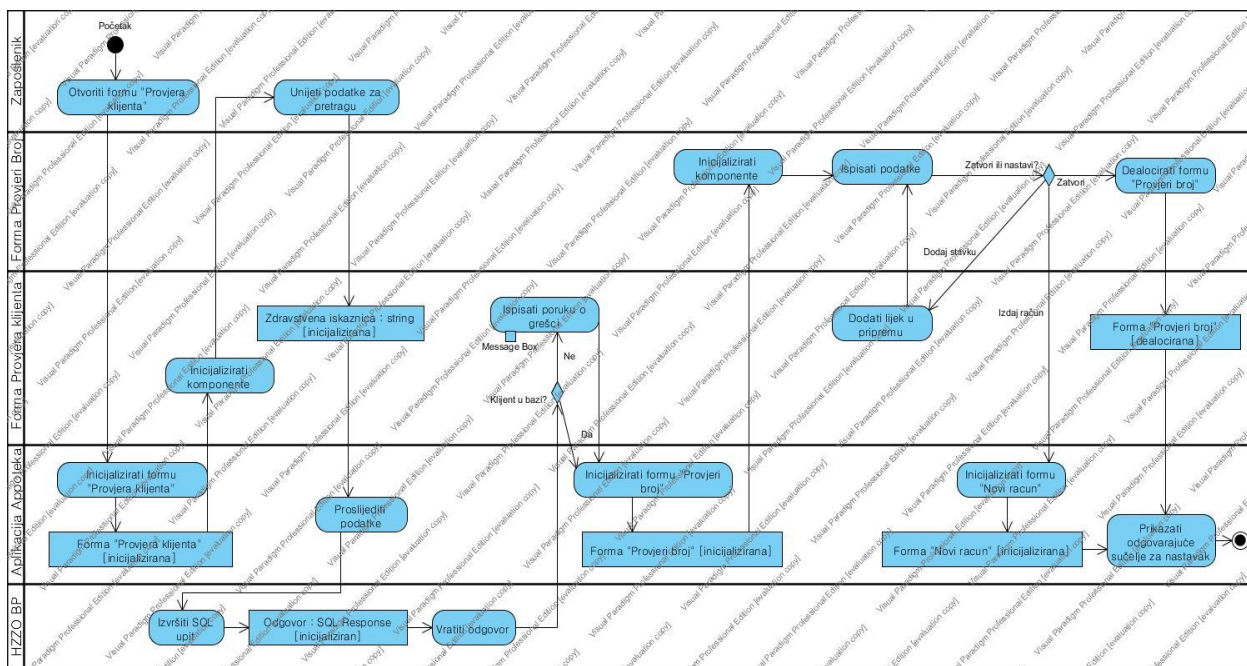
Slika 4.3.7 Dijagram aktivnosti *Evidencija kategorija*

### 4.3.8. Izrada i ispis računa



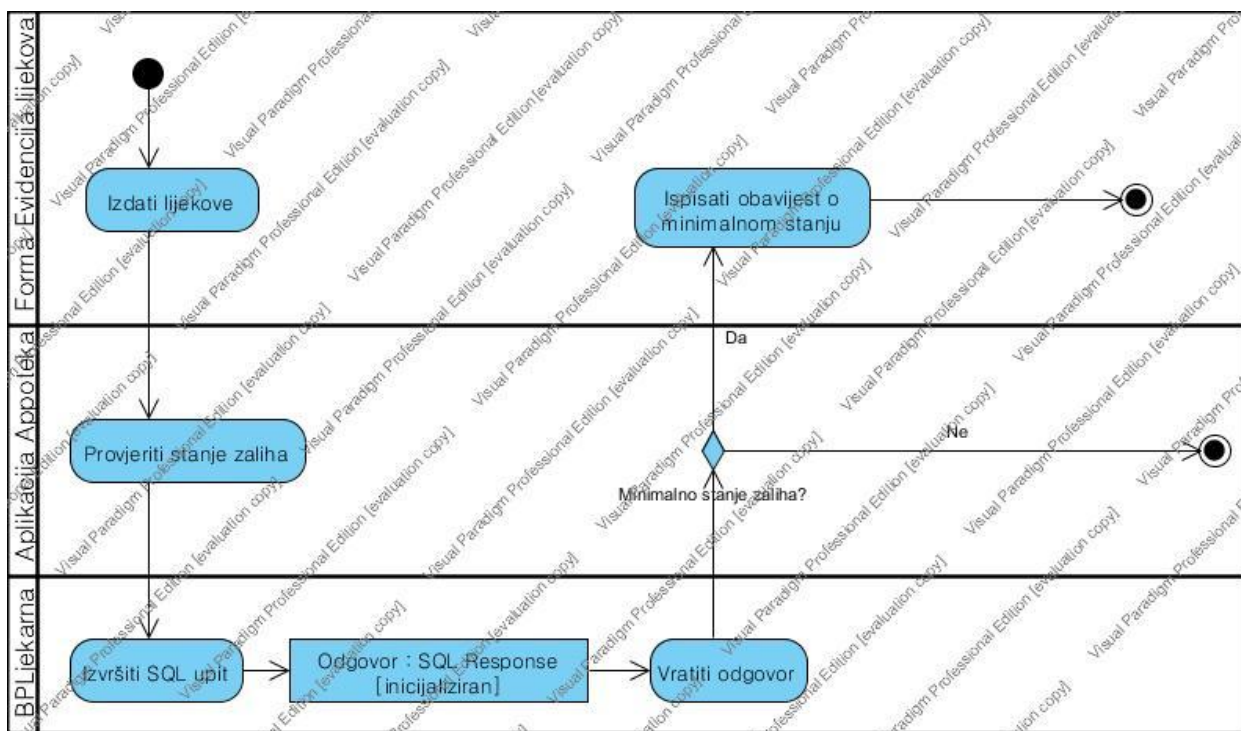
Slika 4.3.8 Dijagram aktivnosti *Izrada i ispis računa*

#### 4.3.9. Provjera klijenta



Slika 4.3.9 Dijagram aktivnosti *Provjera klijenta*

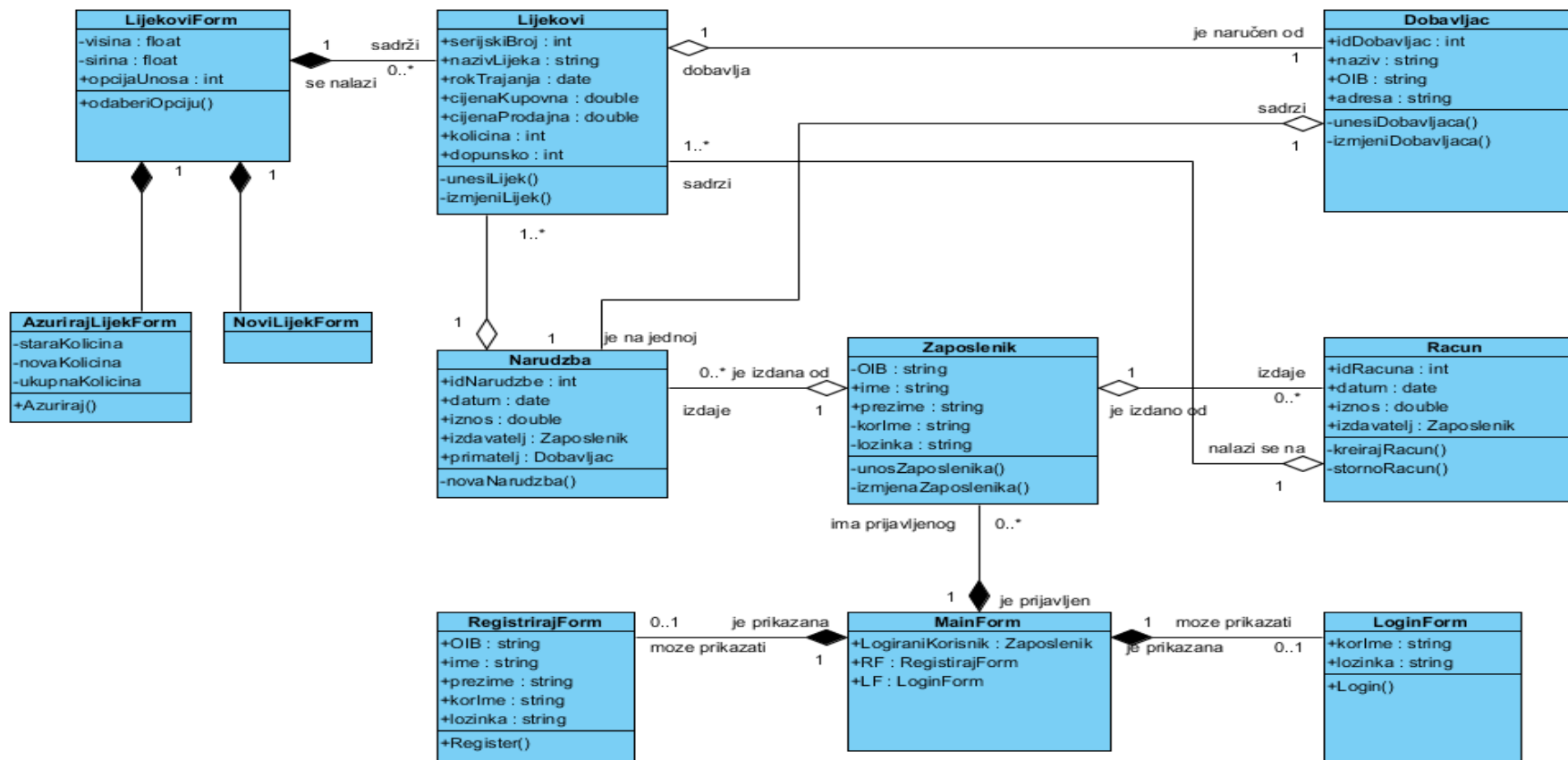
#### 4.3.10. Obavijest o minimalnom stanju zaliha lijekova



Slika 4.3.10 Dijagram aktivnosti *Obavijest o minimalnom stanju zaliha lijekova*



## 5. Dijagram klasa (Class Diagram)



Slika 5.1. Dijagram klasa aplikacije Appoteka.NET

Dijagram klasa prikazuje nam strukturu aplikacije te klase koje ćemo koristiti kod izrade iste. Klasa **Zaposlenik** sadrži informacije o zaposlenicima koji su korisnici aplikacije. Zaposlenik mora prvotno biti registriran da bi se mogao ulogirati u sustav i koristiti funkcionalnosti. To radi preko forme **RegistrirajForm**, a nakon registracije se svaki puta potrebno prijaviti putem **LoginForm** u sustav. Nadalje zaposlenik se koristi u klasama **Narudzba** i **Racun** gdje se kod izdavanja bilo kojeg od tih dvoje poslovnih dokumenata mora naznačiti koja osoba odnosno zaposlenik ju je izdao. Klasa **Lijekovi** sadrži sve informacije o lijekovima, a povezana je sa klasom **Dobavljac** koja klasificira određene dobavljače za određene lijekove. **LijekoviForm** je forma za unos/provjeru/ažuriranje lijekova.

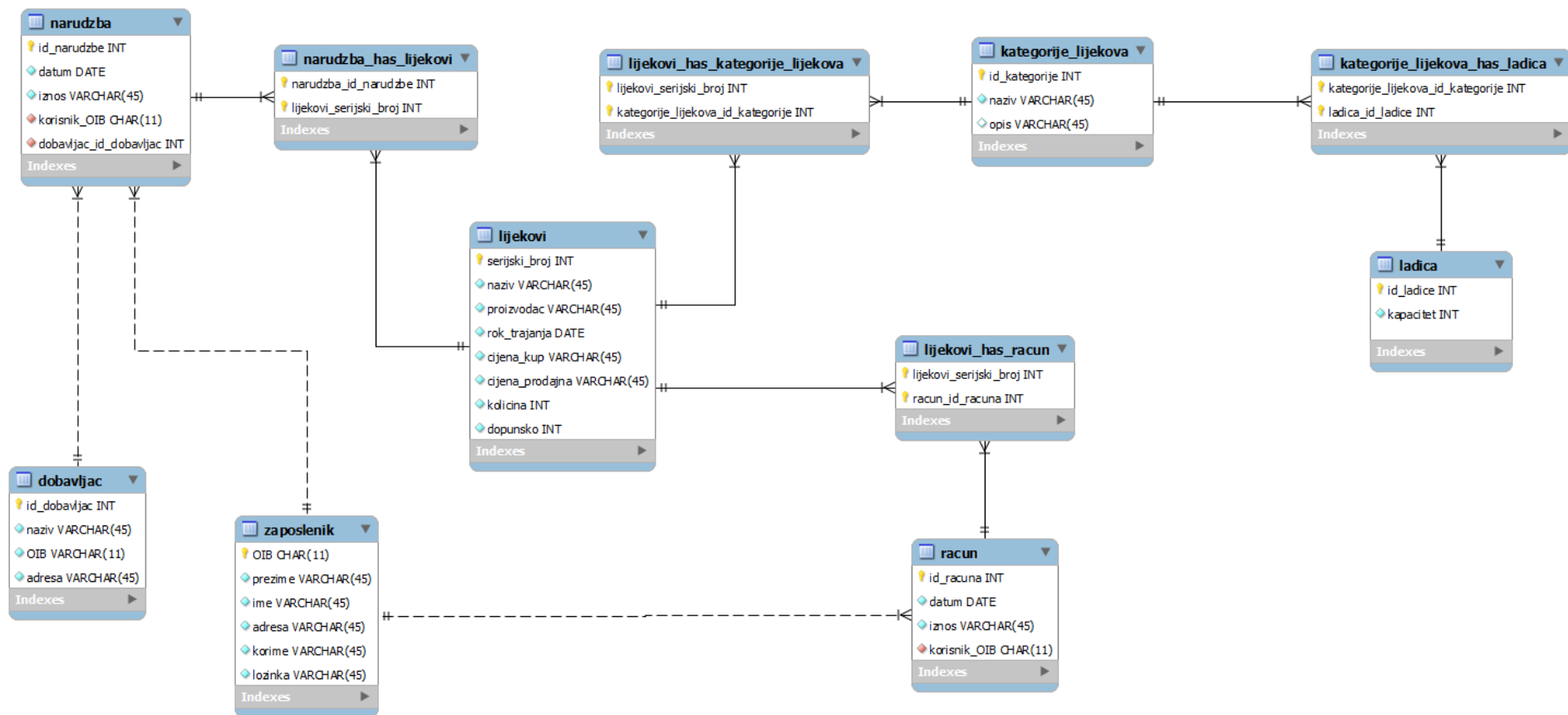
*\*Dijagram klasa nije finalni te je podložan ažuriranju tokom razvoja aplikacije*

## 6. Dijagram podataka

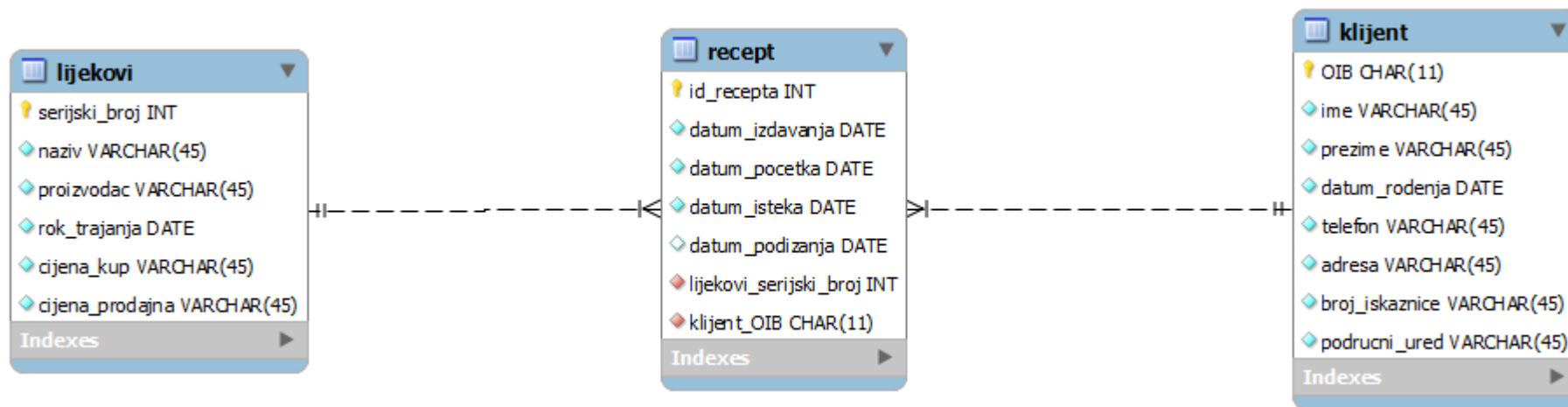
Sljedeći dijagram podataka je ERA model podataka izrađen u alatu MySQL Workbench. ERA model podataka sastoji se od sljedećih entiteta:

- Lijekovi – svaki lijek spada pod određenu kategoriju koja spada pod određenu kategoriju, a ista kategorija može biti raspoređena kroz više spremišta (ladica) ovisno o količinama koje se naručuju
- Zaposlenici – korisnici aplikacije
- Narudzbe – narudžbe lijekova
- Racun – izdavanje računa
- Dobavljac – dobavljači u našem sustavu nabave

Također smo zbog potrebe funkcionalnosti simulirali smo bazu HZZO koja nam je potrebna za povlačenje klijenata i njihovih prepisanih recepata za lijekove.



Slika 6.1 ERA model podataka za bazu Appoteka.NET



Slika 6.2 ERA model podataka sa simulaciju baze HZZO