ePharm

Electronic Pharmacy

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 03.05.2021. | 1.0 | Inicijalna verzija | Stefan |
| 09.05.2021. | 1.0 | Revizija | Dušan |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 9

6.2.1 Prijavljivanje 9

6.2.2 Pregled spiska lekova 10

6.2.3 Sortiranje lekova po šifri 10

6.2.4 Sortiranje lekova po imenu 10

6.2.5 Pregled određenog leka 10

6.2.6 Pretraga lekova po šifri 10

6.2.7 Pretraga lekova po imenu 10

6.2.8 Izdavanje lekova 10

6.2.9 Dodavanje novog leka 10

6.2.10 Ažuriranje podataka o leku 10

6.2.11 Brisanje postojećeg leka 10

6.2.12 Pregled porudžbina 10

6.2.13 Sastavljanje nove porudžbine 11

6.2.14 Korigovanje sastavljenje porudžbine 11

6.2.15 Brisanje postojeće porudžbine 11

6.2.16 Prihvatanje/odbijanje porudžbine 11

6.2.17 Pregled spiska zaposlenih 11

6.2.18 Kreiranje novog korisničkog naloga 11

6.2.19 Brisanje postojećeg korisničkog naloga zaposlenog 11

6.2.20 Ažuriranje podataka postojećeg zaposlenog 11

6.2.21 Pregled spiska svih zaposlenih 11

6.2.22 Pregled spiska svih porudžbina 11

6.2.23 Pregled prihoda i rashoda 11

6.2.24 Grafički prikaz podataka o izabranoj apoteci 11

7. Pogled na logičku arhitekturu sistema 11

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 12

7.1.1 Korisnički interfejs 12

7.1.2 Aplikaciona logika 12

7.1.3 Pristup podacima 13

7.1.4 HTML 13

7.1.5 JavaScript 13

7.1.6 C# (ASP.NET) 13

7.1.6 MySQL 13

8. Pogled na procese 13

8.1 Procesi 13

8.1.1 Web čitač 13

8.1.2 Web server 14

8.1.3 Server (ASP.NET) 14

8.1.4 MySQL Server 14

9. Pogled na raspoređivanje sistema 14

9.1 Klijent 14

9.2 Web server 14

9.3 DBMS server 14

10. Pogled na implementaciju sistema 15

10.1 Model domena 15

10.2 Šema baze podataka 16

10.3 Komponente sistema 16

10.3.1 Komponente korisničkog interfejsa 16

10.3.2 Komponente aplikacione logike 17

10.3.3 Komponente za pristup podacima 17

11. Performanse 18

12. Kvalitet 18

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je specifikacija zahteva u pogledu detaljnog opisa slučajeva korišćenja ePharm sistema.

# Opseg dokumenta

Dokument se odnosi na ePharm aplikaciju koja će biti razvijena od strane DSoft-a. ePharm predstavlja skraćenicu od Electronic Pharmacy (srp. Elektronsko Apotekarstvo). Namena sistema je efikasno i uspešno poslovanje jednog lanca apoteka.

# Reference

Spisak korišćene literature:

1. ePharm – Predlog projekta, DSoft-ePharm-01, V1.0, 2021, DSoft.
2. ePharm – Planirani raspored aktivnosti na projektu, V1.0, 2021, DSoft.
3. ePharm – Plan realizacije projekta, V1.0, 2021, DSoft.
4. ePharm – Vizija sistema, V1.0, 2021, DSoft.
5. ePharm – Specifikacija zahteva, V1.0, 2021, DSoft.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. ePharm aplikacija će biti implementirana kao Web aplikacija zasnovana na JavaScript skripting jeziku, C# jeziku i MySQL bazi podataka [4].
2. Klijentski deo ePharm aplikacije će biti optimizovan za sledeće Web čitače: Microsoft Edge, Opera, Firefox (Mozilla) kao i Google Chrome. [4].
3. Svi zahtevi u pogledu performansi dati u [5] moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

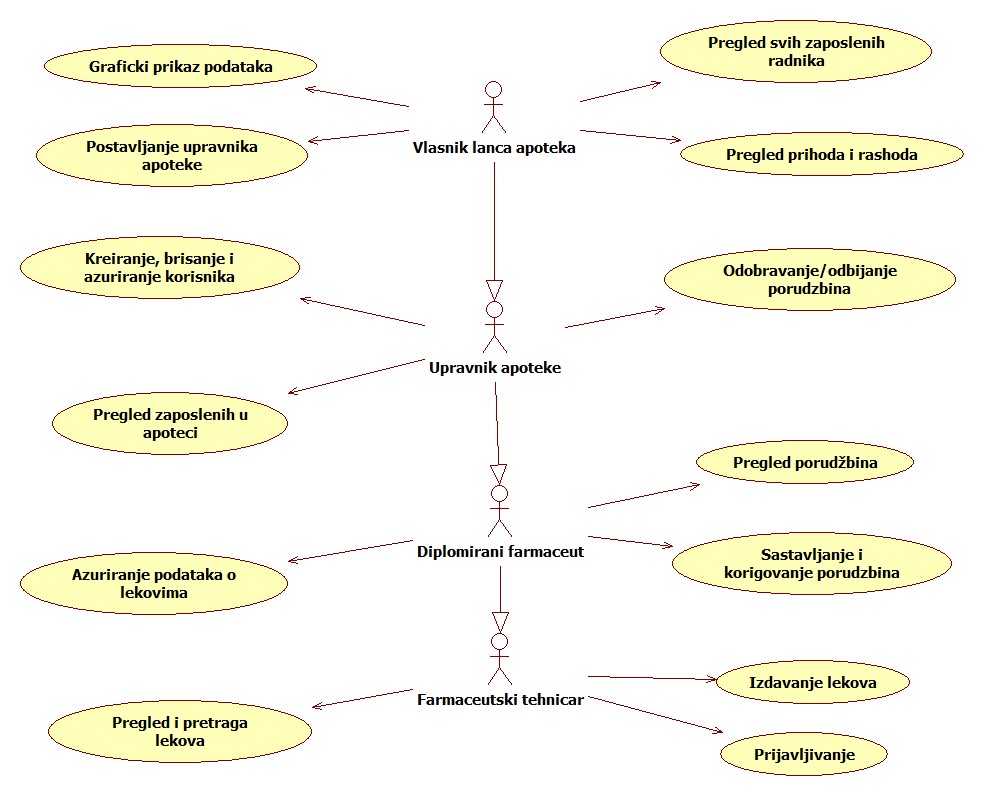
Slučajevi korišćenja ePharm aplikacije su:

* Prijavljivanje
* Pregled spiska lekova
* Sortiranje lekova po šifri
* Sortiranje lekova po imenu
* Pregled odredjenog leka
  + Pregled lekova po šifri
  + Pregled lekova po imenu
* Izdavanje lekova
  + Dodavanje novog leka
  + Ažuriranje podataka o leku
  + Brisanje postojećeg leka
* Pregled porudžbina
  + Sastavljanje nove porudžbine
  + Korigovanje postojeće porudžbine
  + Brisanje postojeće porudžbine
  + Prihvatanje/odbijanje porudžbine
* Pregled spiska zaposlenih
  + Kreiranje novog korisničkog naloga
  + Brisanje postojećeg korisničkog naloga zaposlenog
  + Ažuriranje podataka postojećeg zaposlenog
* Pregled spiska svih zaposlenih
* Pregled spiska svih porudžbina
* Pregled prihoda i rashoda
* Grafički prikaz podataka o izabranoj apoteci

Ove slučajeve korišćenja mogu da iniciraju farmaceutski tehničar, diplomirani farmaceut, upravnik apoteke i vlasnik lanca apoteka.

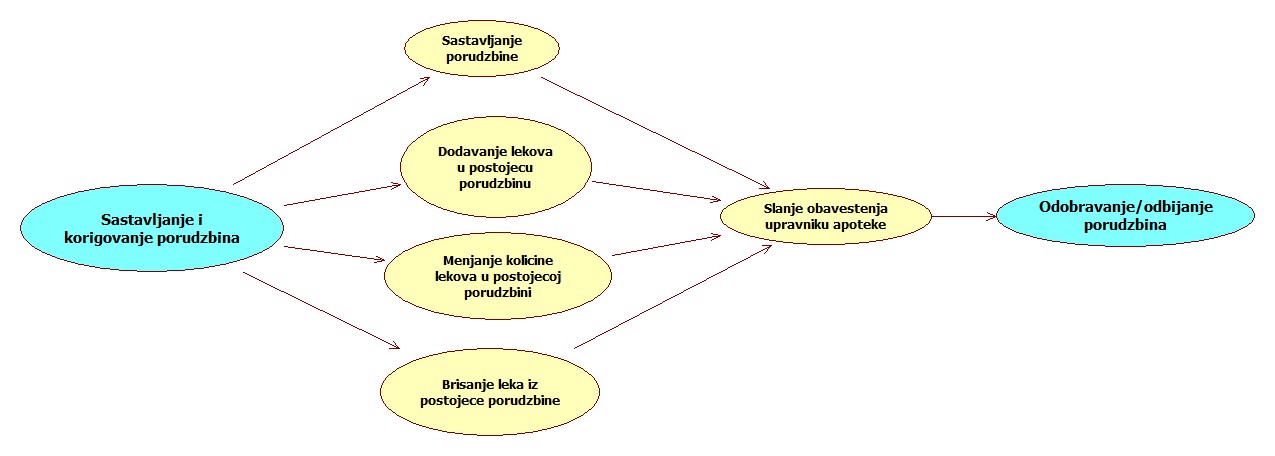
## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja ePharm aplikacije prikazan je na sledećoj slici:

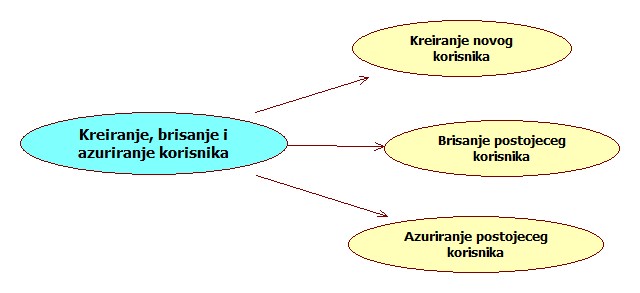


Slučajevi korišćenja *sastavljanje i korigovanje porudžbina*, *kreiranje, brisanje i ažuriranje korisnika, ažuriranje podataka o lekovima* *i pregled i pretraga lekova* obuhvataju složenije radnje koje se mogu razložiti dalje razložiti na pojedinačne slučajeve korišćenja.

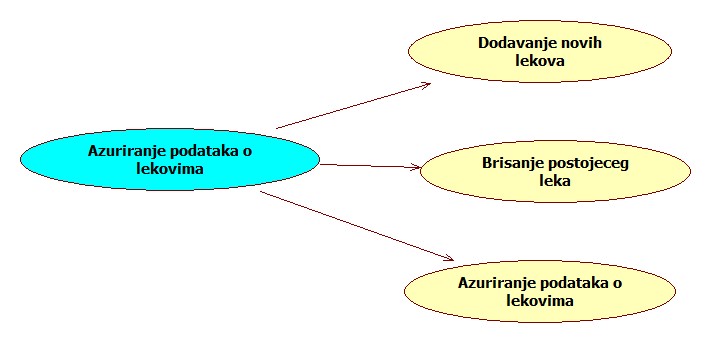
Detaljni UML dijagram za slučaj korišćenja *sastavljanje i korigovanje porudžbina* je prikazan na sledećoj slici:



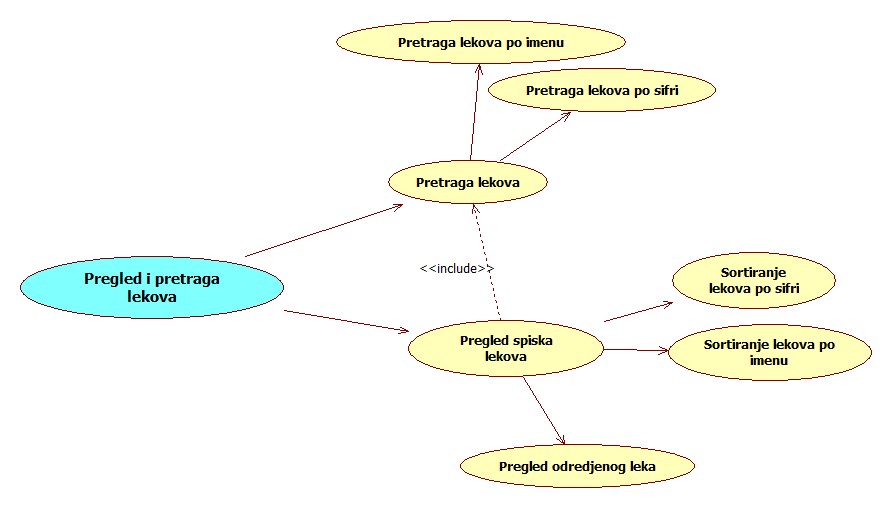
Detaljni UML dijagram za slučaj korišćenja *kreiranje, brisanje i ažuriranje korisnika* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *ažuriranje podataka o lekovima* je prikazan na sledećoj slici:



Detaljni UML dijagram za slučaj korišćenja *pregled i pretraga lekova* je prikazan na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Prijavljivanje

Kratak opis: Prijavljivanje zaposlenih u cilju pristupa funkcionalnostima aplikacije.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Pregled spiska lekova

Kratak opis: Prikaz stranice sa lekovima.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Sortiranje lekova po šifri

Kratak opis: Prikaz stranice sa lekovima sortiranim po šifri.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Sortiranje lekova po imenu

Kratak opis: Prikaz stranice sa lekovima sortiranim po imenu.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Pregled određenog leka

Kratak opis: Prikaz stranice sa informacijama o određenom leku.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Pretraga lekova po šifri

Kratak opis: Prikaz stranice sa lekovima koji sadrže uneti deo šifre ili lek sa potpunom šifrom.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Pretraga lekova po imenu

Kratak opis: Prikaz stranice sa lekovima čije ime sadrži uneti tekst.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Izdavanje lekova

Kratak opis: Izdavanje leka građaninu.

Akteri koji iniciraju slučaj korišćenja: Farmaceutski tehničar, Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Dodavanje novog leka

Kratak opis: Dodavanje novog leka u bazu lekova od strane nadležnog farmaceuta.

Akteri koji iniciraju slučaj korišćenja: Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Ažuriranje podataka o leku

Kratak opis: Izmena podataka leka.

Akteri koji iniciraju slučaj korišćenja: Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Brisanje postojećeg leka

Kratak opis: Brisanje postojećeg leka iz baze lekova.

Akteri koji iniciraju slučaj korišćenja: Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Pregled porudžbina

Kratak opis: Prikaz stranice sa porudžbinama.

Akteri koji iniciraju slučaj korišćenja: Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Sastavljanje nove porudžbine

Kratak opis: Sastavljanje nove porudžbine lekova od strane nadležnog farmaceuta.

Akteri koji iniciraju slučaj korišćenja: Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Korigovanje sastavljene porudžbine

Kratak opis: Izmena količine izabranih lekova i promena izbora lekova porudžbine koja nije odobrena.

Akteri koji iniciraju slučaj korišćenja: Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Brisanje postojeće porudžbine

Kratak opis: Biranje postojeće porudžbine iz baze podataka.

Akteri koji iniciraju slučaj korišćenja: Farmaceut, Upravnik apoteke, Vlasnik lanca apoteka.

### Prihvatanje/odbijanje porudžbine

Kratak opis: Prihvatanje/odbijanje sastavljenih porudžbine od strane upravnika apoteke.

Akteri koji iniciraju slučaj korišćenja: Upravnik apoteke, Vlasnik lanca apoteka.

### Pregled spiska zaposlenih

Kratak opis: Prikaz stranice sa listom korisničkih naloga zaposlenih.

Akteri koji iniciraju slučaj korišćenja: Upravnik apoteke, Vlasnik lanca apoteka.

### Kreiranje novog korisničkog naloga

Kratak opis: Kreiranje korisničkog naloga za novog zaposlenog.

Akteri koji iniciraju slučaj korišćenja: Upravnik apoteke, Vlasnik lanca apoteka.

### Brisanje postojećeg korisničkog naloga zaposlenog

Kratak opis: Brisanje korisničkog naloga i podataka za postojećeg zaposlenog.

Akteri koji iniciraju slučaj korišćenja: Upravnik apoteke, Vlasnik lanca apoteka.

### Ažuriranje podataka postojećeg zaposlenog

Kratak opis: Ažuriranje osnovnih podataka postojećeg naloga zaposlenog.

Akteri koji iniciraju slučaj korišćenja: Upravnik apoteke, Vlasnik lanca apoteka.

### Pregled spiska svih zaposlenih

Kratak opis: Prikaz stranice sa listom korisničkih naloga svih zaposlenih u svim apotekama.

Akteri koji iniciraju slučaj korišćenja: Vlasnik lanca apoteka.

### Pregled spiska svih porudžbina

Kratak opis: Prikaz stranice sa spiskom porudžbina u svim apotekama.

Akteri koji iniciraju slučaj korišćenja: Vlasnik lanca apoteka.

### Pregled prihoda i rashoda

Kratak opis: Prikaz stranice sa prikazom prihoda i rashoda.

Akteri koji iniciraju slučaj korišćenja: Vlasnik lanca apoteka.

### Grafički prikaz podataka o izabranoj apoteci

Kratak opis: Grafički prikaz podataka o izabranoj apoteci.

Akteri koji iniciraju slučaj korišćenja: Vlasnik lanca apoteka.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

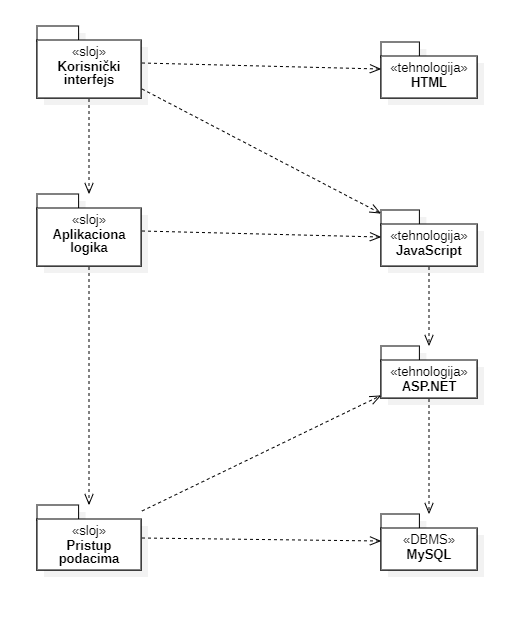
Logički pogled na ePharm sistem obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Web stranice, skripte(Javascript) i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja serverski deo sistema koji je zadužen za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži funkcije koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržane sve HTML, multimedijalni sadržaji i JS skripte koje generišu HTML stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i JS.

### Aplikaciona logika

Sloj aplikacione logike predstavlja serverski deo sistema. Sadrži C# kod koji realizuje funkcionalnost karakterističnu za domen primene sistema i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i C# serverskog dela.

### Pristup podacima

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži C# kod zadužen za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MySQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od C#-a i MySQL baza podataka.

### HTML

tehnologija

Tehnologija HTML-a definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### JavaScript

tehnologija

Tehnologija JS obezbeđuje mehanizam za pisanje i izvršavanje skripti. Ove skripte mogu da generišu HTML kod koji realizuje korisnički interfejs i služe za komunikaciju sa serverskim delom aplikacije.

### C# (ASP.NET)

tehnologija

Tehnologija ASP.NET obezbeđuje mehanizam za pisanje i izvršavanje serverskog dela koda. Ovaj kod odgovara na zahteve klijenata i pristupa bazi podataka u cilju pribavljanja, unosa i ažuriranja podataka.

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju ePharm sistema.

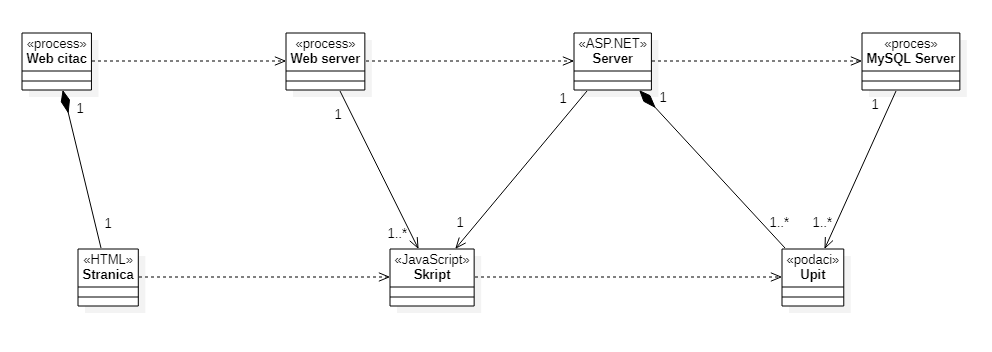
# Pogled na procese

U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje ePharm sistema kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju ePharm sistema. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na klijent-server (JS – C#) arhitekturi i MySQL bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz HTML stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu HTML stranicu.

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuću HTML stranicu na zahtev.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača i predstavlja interfejs između čitača i serverskog dela aplikacije.

### Server (ASP.NET)

Server proces obavlja posao obrade zadatog zadatka i generiše odgovarajući sadržaj koji Web server šalje Web čitaču. Ovaj proces može da zahteva usluge MySQL servera-a. Komunikacija između serverskog procesa i MySQL servera se obavlja preko prosleđivanja upita i vraćanja rezultat.

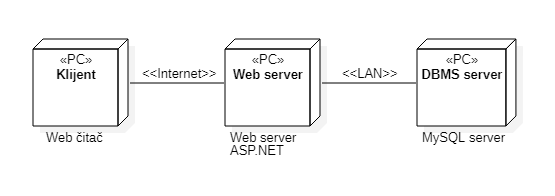
### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja ePharm sistema.



## Klijent

Pristup aplikaciji se obavlja preko klijentskih računara na kojima se izvršava Web čitač. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i procesi servera koji vrše obradu zahteva. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN) ali može da se izvršava i na istoj mašini na kojoj se izvršava i serverski deo aplikacije.

## DBMS server

DBMS server je računar na kome se izvršava MySQL Server proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

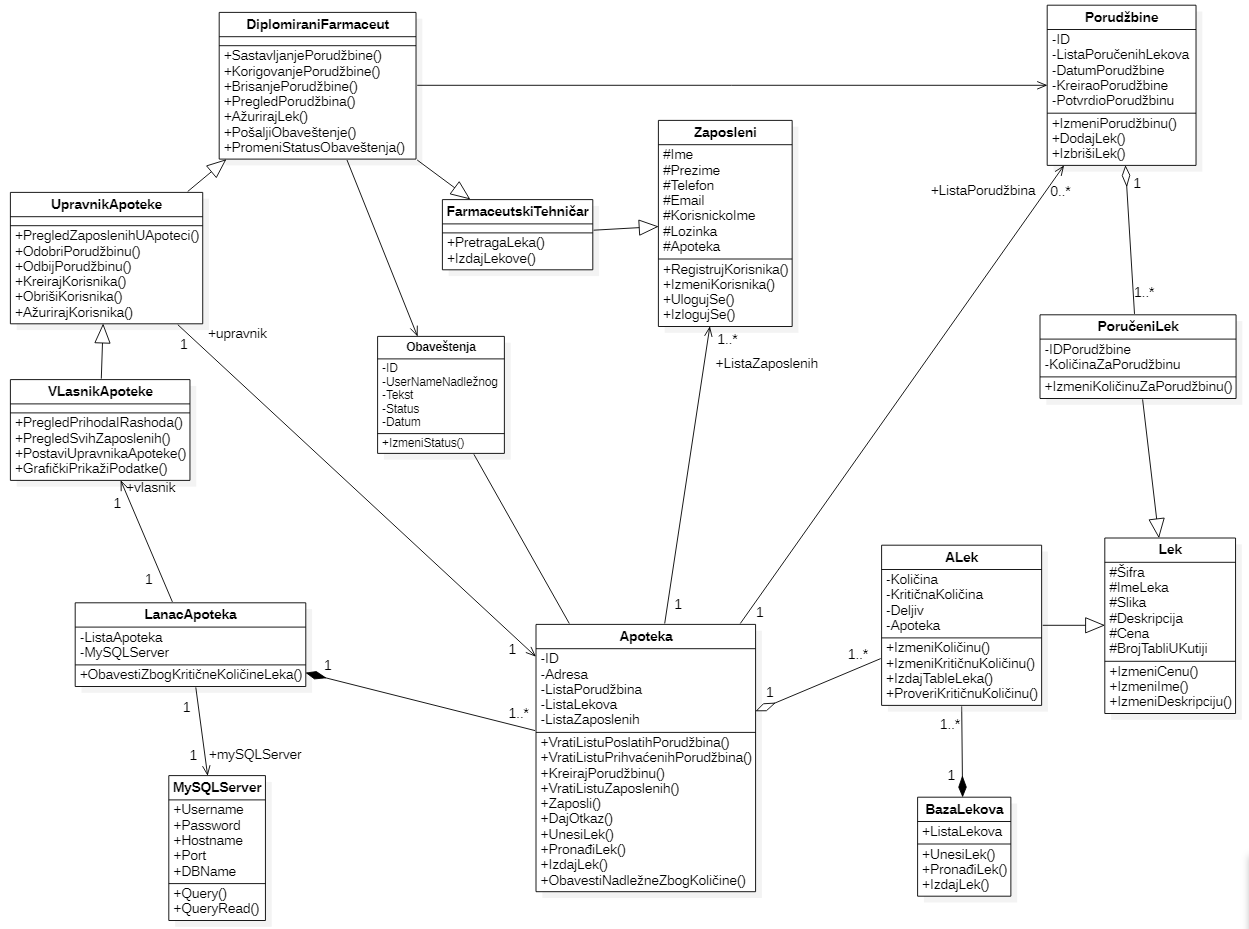
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju ePharm aplikacije ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

Model domena za koji se ePharm sistem projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

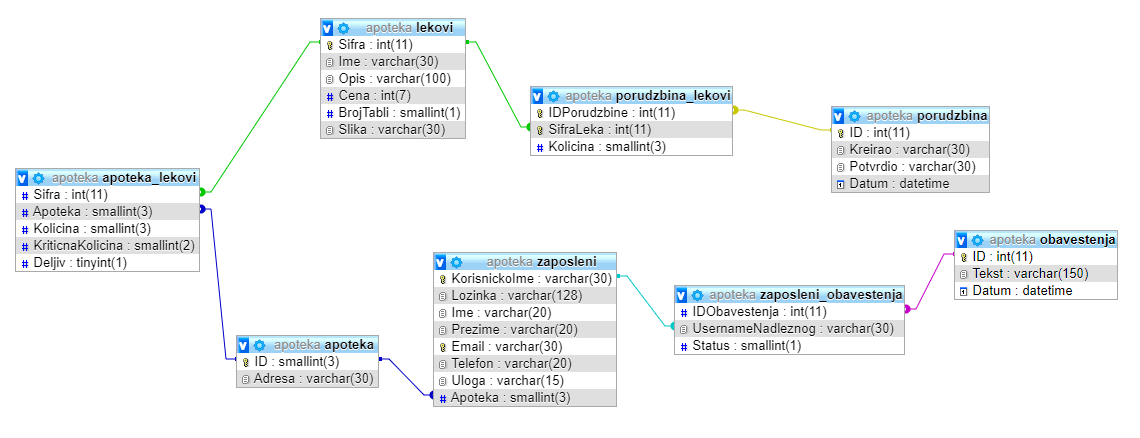
Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu.

Prikazana šema je eksportovana pomoću alata *phpMyAdmin*.



## Komponente sistema

Komponente sistema ePharm aplikacije su JS skripte i C# serverski deo čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa.

### Komponente korisničkog interfejsa

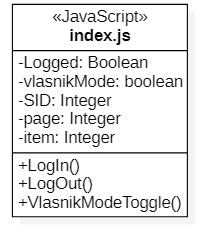
Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



Komponenta **index.js** implementira stranicu aplikacije čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

Parametri koji utiču na prikaz stranice ilustrovani su sledećim dijagramom klasa:

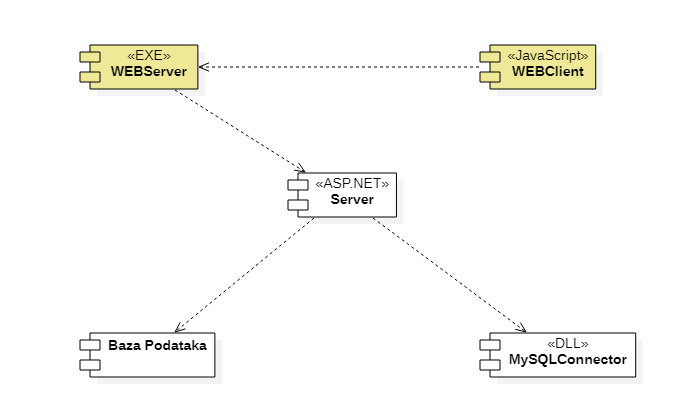


Značenje atributa je sledeće:

* Logged – da li je zaposleni ulogovan;
* vlasnikMode - da li je u podešavanjima čekirano da se prikažu opcije za vlasnika;
* SID - identifikator sesije kada je korisnik ulogovan;
* page - interni identifikator stranice (o apoteci, zaposlenima, porudžbinama)
* item - identifikator stavke koja se deteljno prikazuje na stranici (određeni zaposleni ili lek)

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo prekokomponente **Lanac apoteka**. Lanac apoteka se stara o zahtevima . Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



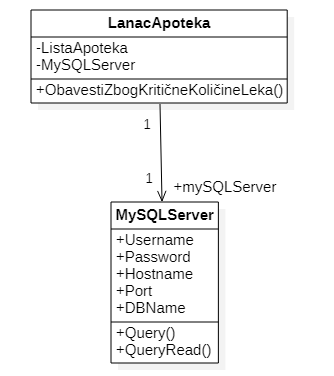
Opis komponenti:

* WEBClient – Šalje zahteve do servera putem WEBServera, obrađuje i prikazuje pristigle podatke.
* WEBServer – Omogućava komunikaciju klijentskog i serverskog dela aplikacije.
* Server – Odgovara i obrađuje zahteve klijenta. Upisuje, ažurira i briše podatke iz baze podataka.
* Baza podataka – Skladišti podatke aplikacije.
* MySQLConnector – Omogućava komunikaciju serverskog dela aplikacije sa bazom podataka.

### Komponente za pristup podacima

Pristup bazi podataka je u potpunosti zatvoren u funkcije koje su definisane u okviru klase *MySQLServer*. Data klasa se instancira u glavnoj klasi – LanacApoteka i putem nje pristižu upiti.

Na sledećem UML dijagramu klasa pobrojane su funkcije za pristup podacima iz baze i prikazana je veza sa glavnom klasom:



Navedene funkcije obavljaju sledeće zadatke:

* Query() – Izvršava INSERT, UPDATE I DELETE upite. U slučaju INSERT upita vraća poslednji uneti ID (LastInsertedId).
* QueryRead() – Izvršava SELECT upite i vraća rezultat.

U okviru ovih funkcija se izvršava i povezivanje na bazu podataka pomoću datih atributa klase.

# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi [5]:

1. Sistem će da podrži do 100 simultanih pristupa korisnika aplikaciji.
2. Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme [5].

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti [5]:

1. ePharm aplikacija će biti dostupna 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupna ne sme da pređe 10%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 120 sati.