

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 5677

**Realizacija portala za potporu
aktivnostima ljekarne**

Mario Kovačević

Zagreb, lipanj 2018.

*Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu \izvornik.*

SADRŽAJ

1. Uvod	1
2. Opis problema	2
3. Modeli	4
3.1. Model baze podataka	4
3.2. Model aplikacije	8
4. Korištene tehnologije	21
5. Opis rješenja	23
6. Zaključak	30
7. Literatura	31
8. Naslov, sažetak i ključne riječi	32
9. Title, abstract and keywords	33

1. Uvod

Još od početka vremena, poznato je da su ljudi otkrivali ljekovite sastojke u prirodi kao i načine za njihovu primjenu. Tada još nisu postojali nikakvi objekti koji bi omogućili njihovo skladištenje niti širokoliku uporabu. No početkom srednjega vijeka dolazi do izraženog povećanja rasta i migracije stanovništva kao i napredovanja trgovine te se javlja potreba za stvaranjem zdravstvenih objekata u kojima bi lijekovi bili dostupni svima i u svako vrijeme. Prvom ljekarnom smatra se ona iz Bagdada koja datira u 8. stoljeće, dok se prve ljekarne u Europi pojavljuju u 12. stoljeću.

Napredak farmacije i tehnologije omogućio je stvaranje boljih i djelotvornijih lijekova, kao i metoda za njihovu distribuciju. Dolaskom modernih vremena u kojima je često izražen manjak vremena, brzina i učinkovitost postaju presudni. Stvaranjem portala za potporu aktivnostima ljekarne, odnosno Sanitas ljekarne, omogućila bi se brza, sigurna i učinkovita kupnja farmaceutskih proizvoda za vlastito zdravlje. Osim javno dostupnih općih informacija o proizvodima i proizvođačima, portal bi omogućio i slanje upita ljekarnicima koji bi u najkraćem roku usmjerili korisnika na proizvod koji odgovara njegovim simptomima i potrebama.

2. Opis problema

Dolaskom interneta pojavila se mogućnost koja korisniku nudi pregledavanje internetskog sadržaja u udobnosti vlastitog doma umjesto, često iscrpljujućeg, vlastitog istraživanja. Kako se internet razvijao, tako je napredovala i internetska trgovina te postala gotovo svakodnevna stvar. Time se otvorila prilika i ljekarnama da svoje proizvode postavе na internet i tako omogućе zainteresiranim kupcima uvid u dostupne lijekove i farmaceutske preparate.

Ostvarivanjem portala za potporu aktivnostima ljekarne, odnosno Sanitas ljekarne, omogućila bi se kupovina proizvoda iz ljekarne internetom kao i njihova dostava na kućna vrata. Time korisnik u samo par trenutaka naručuje željene proizvode i ne mora brinuti o njihovom transportu.

Jedna od najbitnijih stavki internetske trgovine je sigurnost. Portal osigurava privatnost korisničkih podataka kao i da te podatke neće prenositi trećim stranama.

Traženje pomoći ili savjeta od ljekarnika često je prikazano kao najveća prednost neinternetske kupovine farmaceutskih proizvoda, odnosno najveći problem internetske.

Ljekarnik je stručna osoba koja radi u ljekarni, poznaje lijekove i ovlašten je prodavati ih. Za razliku od većine portala koje prodaju farmaceutske proizvode preko interneta, ljekarna Sanitas omogućuje slanje upita ljekarnicima koji pomažu korisniku u nedoumici čime se postiže osjećaj sigurnosti i bezbrižnosti jer korisnik zna da mu je preporuku dao ovlašteni ljekarnik.

Još jedna bitna razlika u odnosu na većinu sličnih portala je ta što ljekarna Sanitas omogućuje postavljanje komentara za svaki proizvod koji zatim postaju vidljivi svim korisnicima. Time potencijalni kupac dobiva bolji uvid u određeni proizvod upravo od njegovih korisnika.

Druga bitna stavka je detaljan opis samog proizvoda. Svaki proizvod ima istaknutog proizvođača, datum proizvodnje i isteka trajanja, vrstu pakiranja, veličinu, moguće nuspojave i bolesti za koje je primjenjiv. Isticanje nuspojava, kao i bolesti na koje je proizvod primjenjiv, omogućuje korisniku da precizno odabere proizvod koji mu odgovara.

Određeni proizvodi imaju i zamjenske proizvode čime se korisniku daje mogućnost biranja između više sličnih lijekova sa sličnim djelovanjem, ovisno o vlastitim preferencijama.

Također, omogućeno je pretraživanje proizvoda po nazivu, njihovo sortiranje po cijeni i datumu isteka valjanosti, kao i filtriranje ovisno o bolestima na koje je primjenjiv. Ostvarivanjem ugodnog dizajna i olakšanog korištenja portala, doprinosi se osjećaju sigurnosti i opuštenosti korisnika.

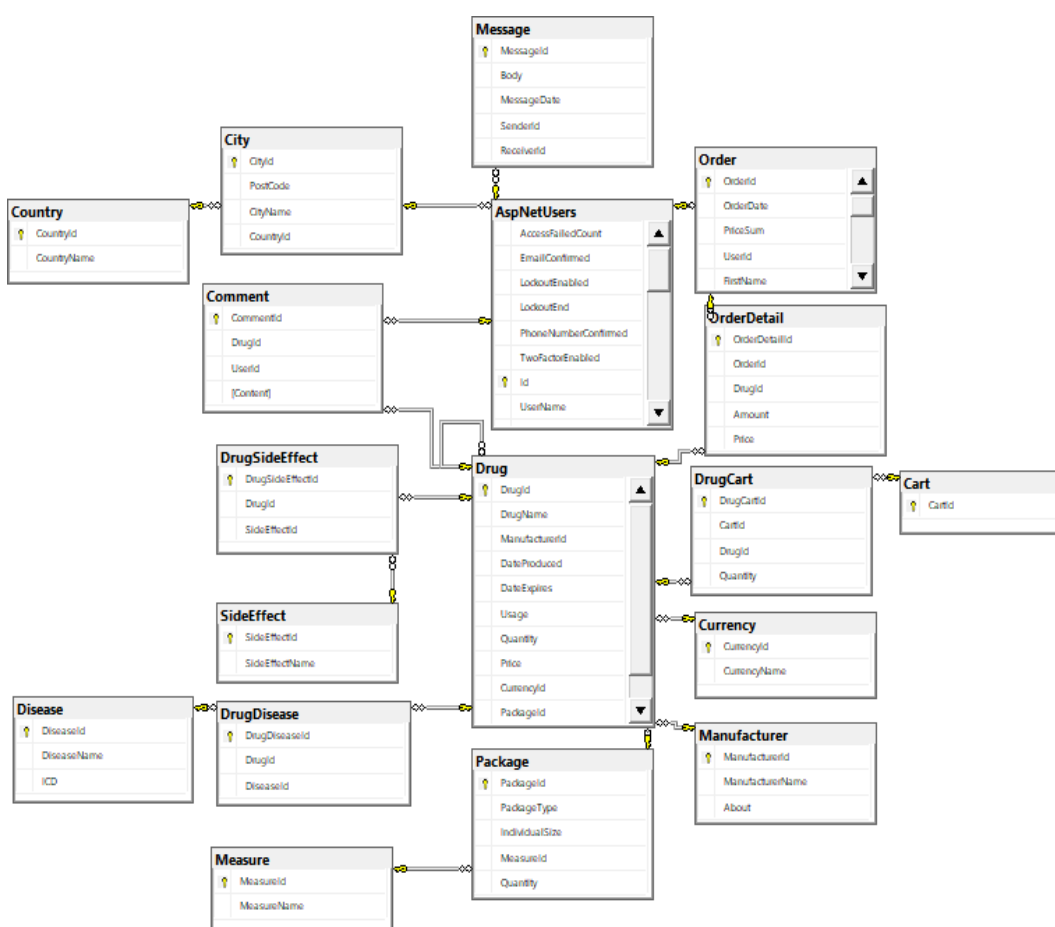
Navedene razlike prikazuju velike prednosti ljekarne Sanitas u odnosu na većinu portala s istom djelatnosti. Povećana je sigurnost, kvaliteta i zadovoljstvo korisnika kao i samo korištenje portala koje je prilagođeno za sve korisnike, neovisno o informatičkom znanju.

3. Modeli

3.1. Model baze podataka

Baza podataka ključan je dio za pohranjivanje podataka ljekarne Sanitas kao i za općenitu dinamičnost podataka cijele aplikacije. U bazi se kreiraju relacije koje olakšavaju dostupnost podataka korisnicima te omogućavaju ubrzano pretraživanje.

Baza se sastoji od modela od kojih svaki predstavlja određeni entitet. Svaki entitet opisan je s jednim ili više atributa te sadrži vlastiti identifikacijski broj koji predstavlja njegov primarni ključ koji služi za jedinstveno određivanje entiteta u bazi. Za stvaranje veze između dva ili više entiteta koriste se strani ključevi.



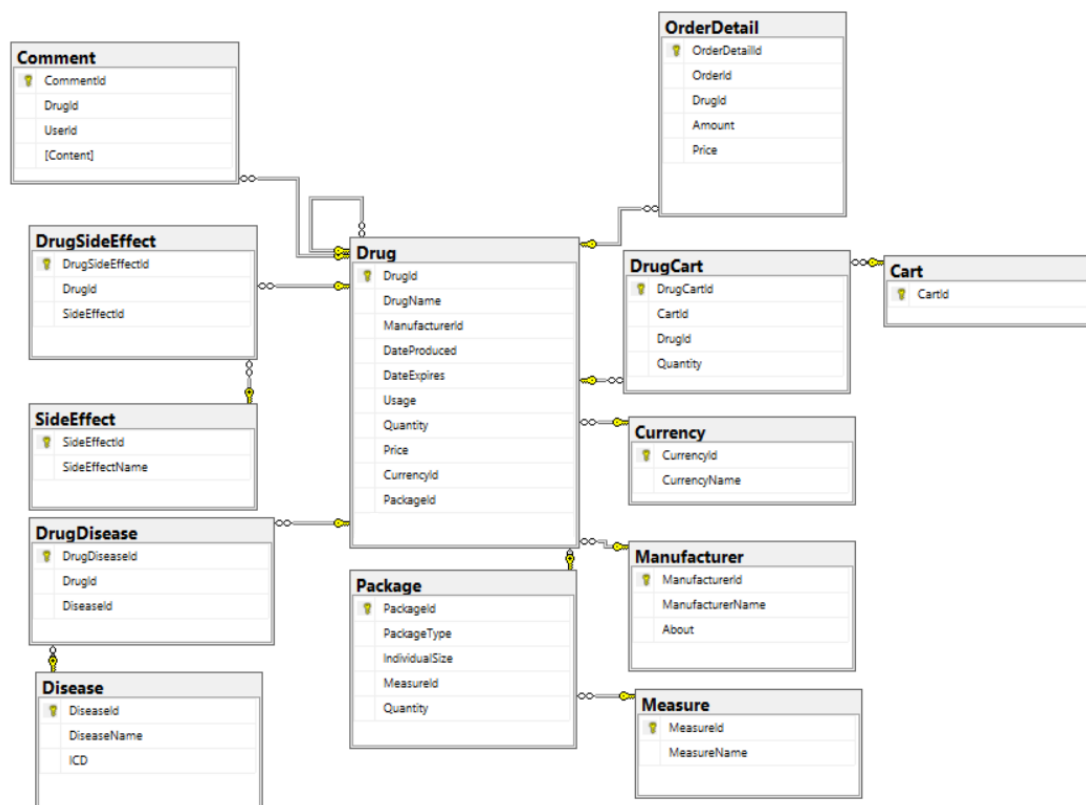
Slika 1. Svi modeli baze podataka

Središnji i najvažniji entitet baze je proizvod (*Drug*) koji sadrži attribute koji ga definiraju, a to su: naziv, datum proizvodnje i isteka valjanosti, način uporabe, količina na skladištu i cijena.

Kao strane ključeve, entitet sadrži identifikacijske brojeve entiteta: proizvođača, valute te proizvoda što omogućuje stvaranje rekurzivne veze.

Također, sadrži kolekcije nuspojava, bolesti, komentara, zamjenskih lijekova i košarica za kupnju.

Lijek ima 1-n vezu s entitetima: komentara i narudžbe, n-1 vezu s: pakiranjem i proizvođačem, a n-n vezu s entitetima: nuspojave, bolesti i košarice.



Slika 2. Povezanost modela proizvoda s ostalim modelima

Entitet proizvođača (*Manufacturer*) sadrži ime te kratki opis dotičnog proizvođača, a bolest (*Disease*) sadrži naziv i međunarodni klasifikacijski broj bolesti (*ICD*). Mjerna jedinica (*Measure*), valuta (*Currency*) i nuspojava (*SideEffect*) sadrže samo naziv. Komentar (*Comment*) se sastoji od sadržaja te je uz narudžbu izravna veza između korisnika i lijeka.

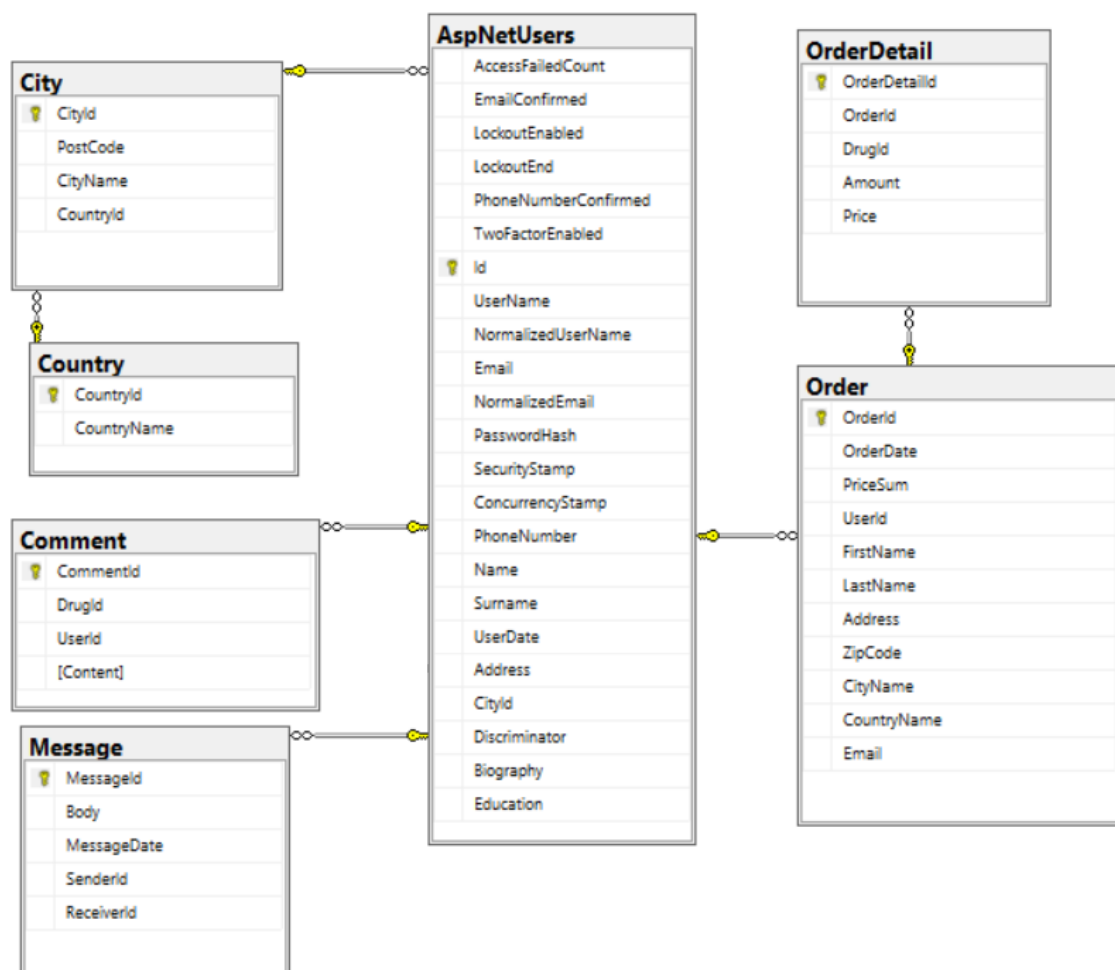
Pakiranje (*Package*) se sastoji od imena tipa pakiranja, količine u pakiranju, veličine pojedinačne stavke u pakiranju kao i pripadajuće mjerne jedinice.

Veza između proizvoda i nuspojave, odnosno proizvoda i bolesti je proširena s tablicama *DrugDisease* i *DrugSideEffect* koje omogućuju stvaranje potrebne n-n veze.

Drugi najvažniji entitet je korisnik (*AspNetUsers*). Osim već ugrađenih atributa, ovaj entitet sadrži još i ime, prezime, datum registriranja i adresu.

Od stranih ključeva, sadrži identifikacijski broj grada. Također, sadrži kolekciju poruka koje je korisnik primio i poslao te kolekciju komentara i narudžbi.

Entitet korisnika ima n-1 vezu s entitetom grada, 1-n s narudžbom, porukom i komentarom.



Slika 3. Povezanost modela korisnika s ostalim modelima

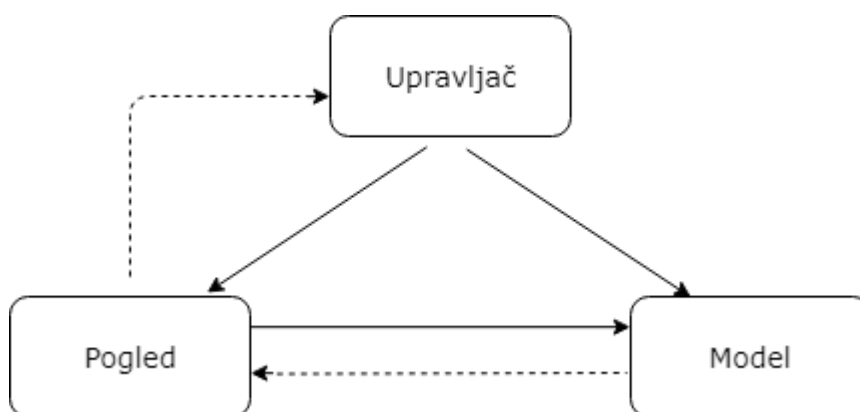
Entitet poruke (*Message*) sadrži tijelo poruke, datum i vrijeme slanja te pošiljatelja i primatelja. Grad (*City*) sadrži naziv, poštanski broj i državu u kojoj se nalazi.

Narudžba (*Order*) sadrži ukupnu cijenu, datum narudžbe te podatke potrebne za dostavu proizvoda.

3.2. Model aplikacije

Arhitektura mrežne aplikacije sastoji se od tri dijela: modela, pogleda i upravljača te je ta arhitektura poznata kao model-pogled-upravljač ili MVC arhitektura. Dvije glavne ideje nalaze se u osnovi ove arhitekture, a to su ponovno korištenje postojećeg koda i jasna raspodjela zadaća među različitim dijelovima sustava.

Prva ideja omogućuje da se jednom napisani kod uz minimalne ili čak nikakve izmjene može koristiti u više različitih projekata. Druga ideja je podjela sustava na više međusobno nezavisnih cjelina od kojih svaka ima svoj zadatak i koje se pojedinačno mogu mijenjati bez straha da će se te promjene odraziti na ostatak sustava. (Freeman, 2013)



Slika 4. Ovisnost dijelova u MVC arhitekturi

Model sadrži podatke u obliku pogodnom za konkretnu primjenu. Osim toga, sadrži i logiku aplikacije, odnosno definira što se sve može raditi s danim podacima. Mnoge aplikacije (kao i Sanitas ljekarna) koriste mehanizme za trajno čuvanje podataka (bazu podataka) te onda model predstavlja pojedinačni entitet u bazi. (Freeman, 2013)

```

using System;
using System.Collections.Generic;
using System.Text;

namespace DrugData.Models
{
    5 references
    public class SideEffect
    {
        10 references | 0 exceptions
        public int SideEffectId { get; set; }

        0 references | 0 exceptions
        public virtual ICollection<DrugSideEffect> DrugSideEffect { get; set; }
        11 references | 0 exceptions
        public string SideEffectName { get; set; }
    }
}

```

Slika 5. Kod MVC modela

Slika 5 prikazuje primjer izgleda koda modela, u ovom slučaju radi se o modelu nuspojave (*SideEffect*). Svaki model sadrži svoj identifikacijski broj u obliku *nazivModelaId* te atribute koji ga definiraju kao i strane ključeve.

Upravljač (*Controller*) koordinira modele i poglede, odnosno odlučuje kako model treba promijeniti te koji pogled koristiti i to uglavnom na osnovi korisnikova unosa. Kada dođe do nekog događaja, primjerice, klika na neko dugme, upravljač obavještava model o tome i prikazuje potrebni pogled. (Freeman, 2013)

```

public SideEffectController(ApplicationDbContext context,
    IActionDescriptorCollectionProvider actionDescriptorCollectionProvider)
{
    _dbContext = context;
    _actionDescriptorCollectionProvider = actionDescriptorCollectionProvider;
}

3 references | 0 requests | 0 exceptions
public async Task<IActionResult> Index(string sortOrder, string currentFilter, string searchString, int? page)...
```

```

[HttpGet]
0 references | 0 requests | 0 exceptions
public IActionResult Add()
{
    return View(new SideEffectViewModel());
}

[HttpPost]
0 references | 0 requests | 0 exceptions
public IActionResult Create(SideEffectViewModel model)
{
    if (ModelState.IsValid)
    {
        var sideEffect = new SideEffect { SideEffectName = model.Name};
        var x = _dbContext.SideEffect.FirstOrDefault(g => g.SideEffectName == sideEffect.SideEffectName);
        _dbContext.SideEffect.Add(sideEffect);

        TempData["Success"] = true;
        _dbContext.SaveChanges();
        TempData[Constants.Message] = $"Nuspojjava je dodana";
        TempData[Constants.ErrorOccurred] = false;
    }

    return RedirectToAction(nameof(Index));
}

```

Slika 6. Kod MVC upravljača za dodavanje

```

[HttpPost]
0 references | 0 requests | 0 exceptions
public IActionResult Delete(int id){

    var sideEffect = _dbContext.SideEffect.FirstOrDefault(p => p.SideEffectId == id);
    _dbContext.SideEffect.Remove(sideEffect);
    _dbContext.SaveChanges();

    return RedirectToAction(nameof(Index));
}

[HttpGet]
0 references | 0 requests | 0 exceptions
public IActionResult Edit(int id){

    var ses = _dbContext.SideEffect.FirstOrDefault(p => p.SideEffectId == id);
    var model = new EditSideEffectViewModel
    {
        SideEffect = ses
    };
    return View(model);
}

[HttpPost]
0 references | 0 requests | 0 exceptions
public IActionResult Update(int id, EditSideEffectViewModel model){

    if (ModelState.IsValid){

        var ses = _dbContext.SideEffect.FirstOrDefault(m => m.SideEffectId == id);
        ses.SideEffectName = model.SideEffect.SideEffectName;

        _dbContext.SaveChanges();
    }

    return RedirectToAction(nameof(Index));
}

```

Slika 7. Kod MVC upravljača za brisanje i uređivanje

Slika 6 prikazuje izgled koda upravljača za entitet nuspojave (*SideEffect*) s pripadajućim metodama za dodavanje. Pritisak na gumb za dodavanje nove nuspojave, poziva u upravljaču metodu *Add* koja u pogled za dodavanje nuspojave šalje odgovarajući model za pogled (*ViewModel*) koji služi za lakše prenošenje podataka između modela i pogleda, a zatim se u pogledu definiraju potrebni atributi. Odabir spremanja unesenih vrijednosti poziva metodu *Create* koja, u slučaju da je sve valjano, definiranu nuspojavu dodaje u bazu.

Na Slici 7 je definiran kod za brisanje i uređivanje entiteta nuspojave. Pritisak na gumb za brisanje poziva metodu *Delete* te joj šalje identifikacijski broj nuspojave koju je potrebnu obrisati. Isto tako, gumb za uređivanje poziva metodu *Edit* kojoj također šalje identifikacijski broj nuspojave za uređivanje te se otvara pogled koji omogućuje mijenjanje atributa. Spremanjem promjena, poziva se metoda *Update* koja promijenjenu nuspojavu sprema u bazu podataka.

Operacije dodavanja, uređivanja i brisanja entiteta se smatraju najosnovnijim operacijama upravljača. U ljekarni Sanitas upravljač gotovo svakog entiteta ima implementirane te operacije te je administrator u većini slučajeva jedina vrsta korisnika koja ih može izvoditi.

Pogled (*View*) prikazuje podatke iz modela u formatu pogodnom za interakciju te može omogućiti korisniku mijenjanje podataka. Pogled nikako ne treba biti odgovoran za skladištenje podataka, osim kada se koristi spremanje u priručnu memoriju kao mehanizam poboljšanja performansi. Osim toga, može biti odgovoran za prikaz jednog ili više objekata iz modela.

Unutar jedne aplikacije može postojati više pogleda koji su prilagođeni različitim situacijama te prikazuju podatke iz istog modela. (Freeman, 2013)

```
@model DrugData.Models.ViewModels.SideEffectViewModel

<form asp-action="Create" method="POST">
  <div asp-validation-summary="All" class="text-danger"></div>
  <h2>
    Nova nuspojava
  </h2>
  <div class="form-group">
    <label asp-for="Name">Naziv </label>
    <input asp-for="Name" class="form-control" />
  </div>

  <div class="text-center">
    <button type="submit" class="btn btn-primary">Spremi</button>
    <a asp-action="Index">
      <button type="button" class="btn btn-secondary">Odustani</button>
    </a>
  </div>
</form>
```

Slika 8. Kod MVC pogleda

Nova nuspojava

Naziv

Spremi

Odustani

Slika 9. Izgled pogleda unutar internet preglednika

Slika 8 prikazuje kod jednog od pogleda napisanog u HTML-u, dok Slika 9 prikazuje izgled istog tog pogleda u internetskom pregledniku. Radi se o pogledu koji predstavlja dodavanje nove nuspojave, odnosno definiranje njezinih atributa (imena nuspojave). Također, sadrži i gumb za spremanje, odnosno odustajanje. Pritiskom na gumb za spremanje, poziva se metoda *Create* u upravljaču koja definiranu nuspojavu sprema u bazu.

Glavne prednosti MVC arhitekture već su spomenute – ponovno korištenje koda i podjela sustava na međusobno nezavisne cjeline. Projekti su sistematičniji, njihovi pojedinačni dijelovi mogu se lako mijenjati i poboljšavati, lakše ih je testirati. Paralelni razvoj aplikacije može se lako organizirati i samim time povećava se produktivnost.

Naravno MVC ima i svoje mane. Previše je kompleksan za implementaciju kod razvoja manjih aplikacija i njegovo korištenje u tim slučajevima dovodi do pogoršanja dizajna i performansi. Ponekad se može dogoditi da uslijed čestih promjena modela pogled postane preplavljen zahtjevima za izmjenom. (Freeman, 2013)

3.3. Obrasci uporabe

Obrasci uporabe predstavljaju pogled na sustav koji naglašava njegovo vanjsko ponašanje prema korisniku te opisuju sve moguće interakcije sustava. Prikazuju se pomoću dijagrama obrazaca uporabe koji uz njih, opisuju i korisnike te njihove međusobne odnose. (Sommerville, 2007)

Ljekarna Sanitas ima pet vrsta korisnika: vlasnika portala, administratora, ljekarnika te neregistriranog i registriranog korisnika.

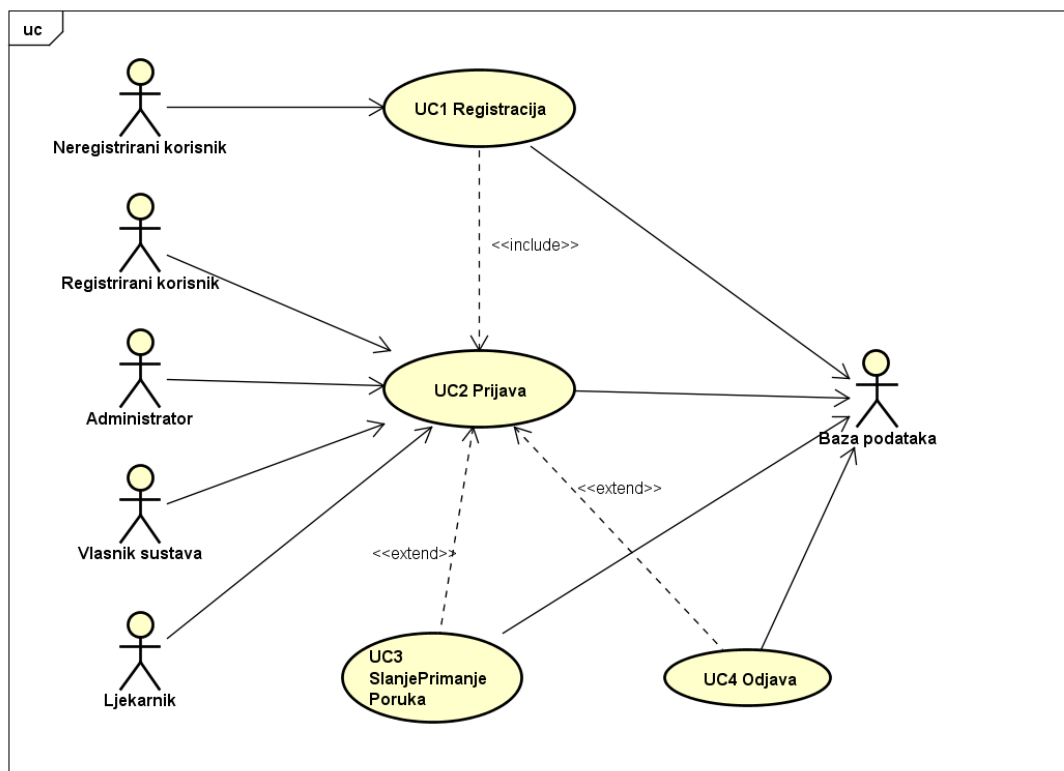
Vlasnik portala najviše je tijelo u hijerarhiji aplikacije. Ima najveću ovlast, a jedina zadaća mu je dodavanje, uređivanje i brisanje administratora. Vlasnika definiraju ime, prezime, korisničko ime (koje je identično njegovoj adresi elektroničke pošte) te lozinka. Te podatke može mijenjati i upisivati isključivo on. Samo je jedan vlasnik sustava. Poruke može slati isključivo administratorima.

Administrator je najvažnije tijelo koje omogućuje rad ljekarne Sanitas. Njegova primarna zadaća je upravljanje proizvodima što uključuje njihovo dodavanje, uređivanje i brisanje kao i ostalih entiteta koji su s njime povezani. Osim upravljanja proizvodima, može dodavati, brisati i uređivati ljekarnike koji pomažu korisnicima na portalu. Također, administrator je zadužen za osiguravanje reda i sigurnosti na portalu. Omogućeno mu je brisanje komentara koje smatra neprimjerenim, kao i slanje poruka svih korisnicima portala, uključujući i vlasnika. U slučaju nedoličnog ponašanja, admin može ukloniti korisnika iz baze što onemogućuje njegov daljnji rad na stranici.

Registrirani i neregistrirani korisnik imaju mogućnost pregleda svih proizvoda, proizvođača i ljekarnika, kao i njihovo filtriranje i sortiranje. To su jedine mogućnosti neregistriranog korisnika. Registrirani može stavljati proizvode u košaricu te iste i naručiti. U slučaju nedoumice, može poslati upite ljekarniku po želji. Također, može ostavljati komentare za svaki proizvod, koje može naknadno brisati i uređivati. Osim toga, može i uređivati vlastite podatke te pregledati prošle kupovine.

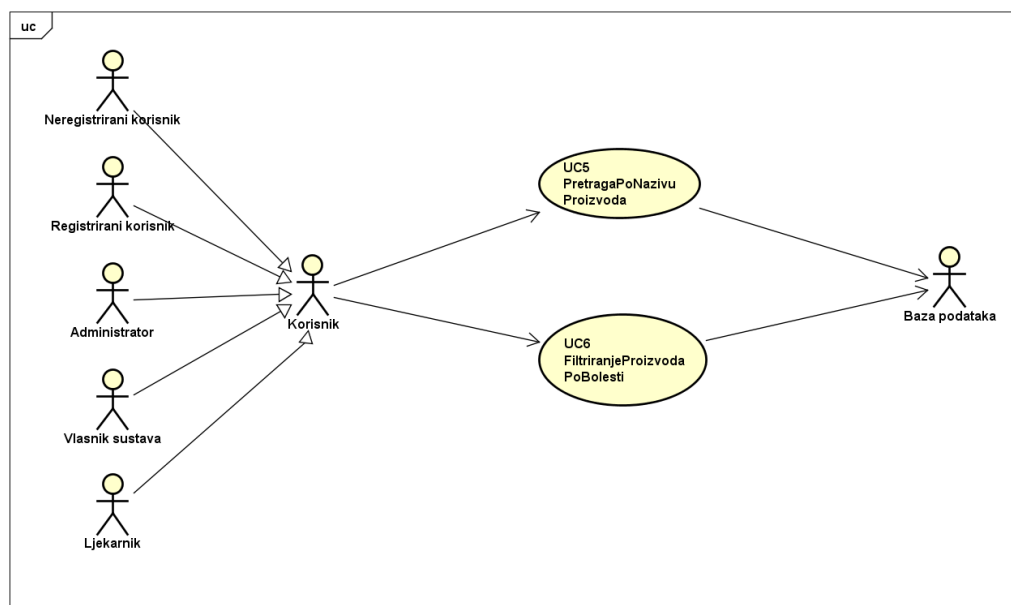
Ljekarnik ima jednaku mogućnost pregleda portala kao i registrirani korisnik, osim što može slati poruke administratorima, registriranim korisnicima, kao i drugim ljekarnicima.

Slijedeći dijagrami će prikazati i opisati svaki obrazac uporabe u Ljekarni Sanitas koji uključuju korisnika i bazu podataka kao sudionike.



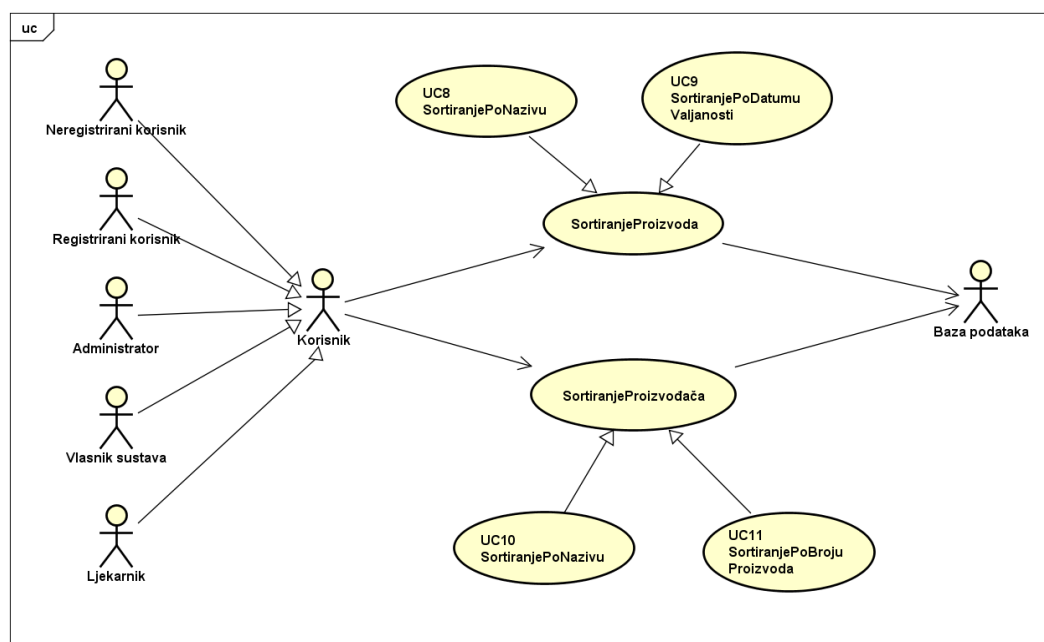
Slika 10. Obrasci uporabe registracije, prijave, odjave i slanja te primanja poruka

Slika 10 prikazuje obrasce uporabe za registraciju, prijavu, odjavu te slanje i primanje poruka. Registrirati se može samo neregistrirani korisnik. Nakon uspješne registracije, neregistrirani korisnik postaje registriran te je automatski prijavljen na portal i nakon toga, kao i svi ostali korisnici, može slati i primiti poruke, kao i odjaviti se s portala.



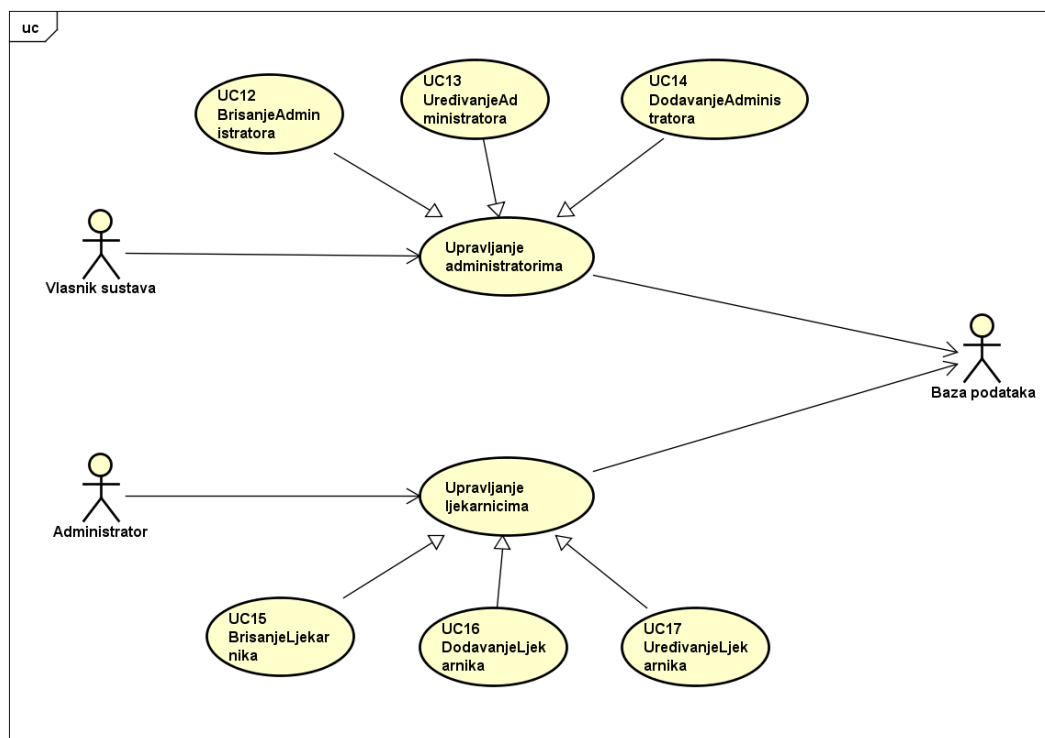
Slika 11. Obrasci uporabe filtriranja i pretrage proizvoda

Slikom 11 primjećuju se obrasci uporabe pretrage te filtriranja proizvoda. Filtriranje uključuje mogućnost filtracije po bolestima na koje je lijek primjenjiv. Pretraživanju i filtriranju mogu pristupiti svi korisnici.



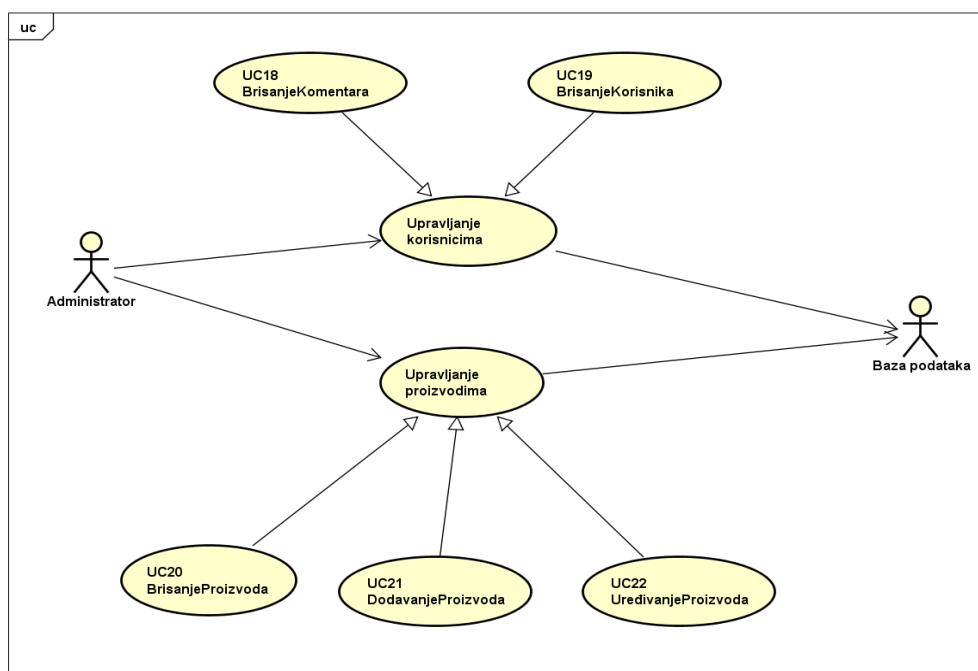
Slika 12. Obrazac uporabe sortiranja

Obrasci uporabe sortiranja vidljivi su na Slici 12. Moguće je sortirati proizvode i proizvođače. Proizvode po nazivu i datumu isteka valjanosti, a proizvođače po broju proizvoda i nazivu. Sve vrste sortiranja moguće je obaviti u uzlaznom, odnosno silaznom redoslijedu.



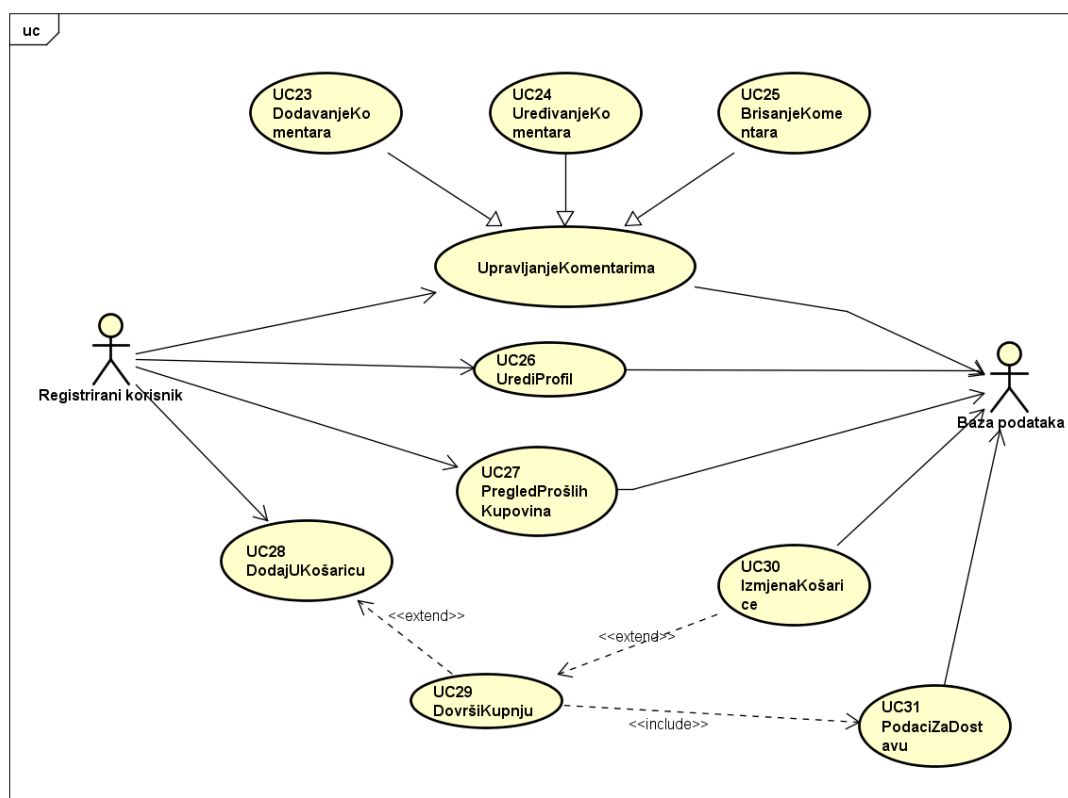
Slika 13. Obrasci uporabe upravljanja administratorima i ljekarnicama

Slika 13 prikazuje obrasce uporabe upravljanja administratorima i ljekarnicama. Vlasnik sustava jedini je koji može upravljati s administratorima, odnosno može ih dodavati, uređivati i brisati. Administrator upravlja ljekarnicama te nad njima ima iste mogućnosti kao i vlasnik nad administratorima, odnosno dodavanje, brisanje i uređivanje.



Slika 14. Obrasci uporabe upravljanja korisnicima i proizvodima

Na Slici 14 primjećuju se obrasci uporabe upravljanja korisnicima i proizvodima. To je ujedno i posljednji dijagram obrazaca uporabe koji uključuje administratora kao glavnog sudionika. Proizvode može dodavati, uređivati i brisati, dok korisnike može jedino obrisati kao i njihove komentare. Brisanje korisnika uključuje i brisanje svih njegovih komentara.



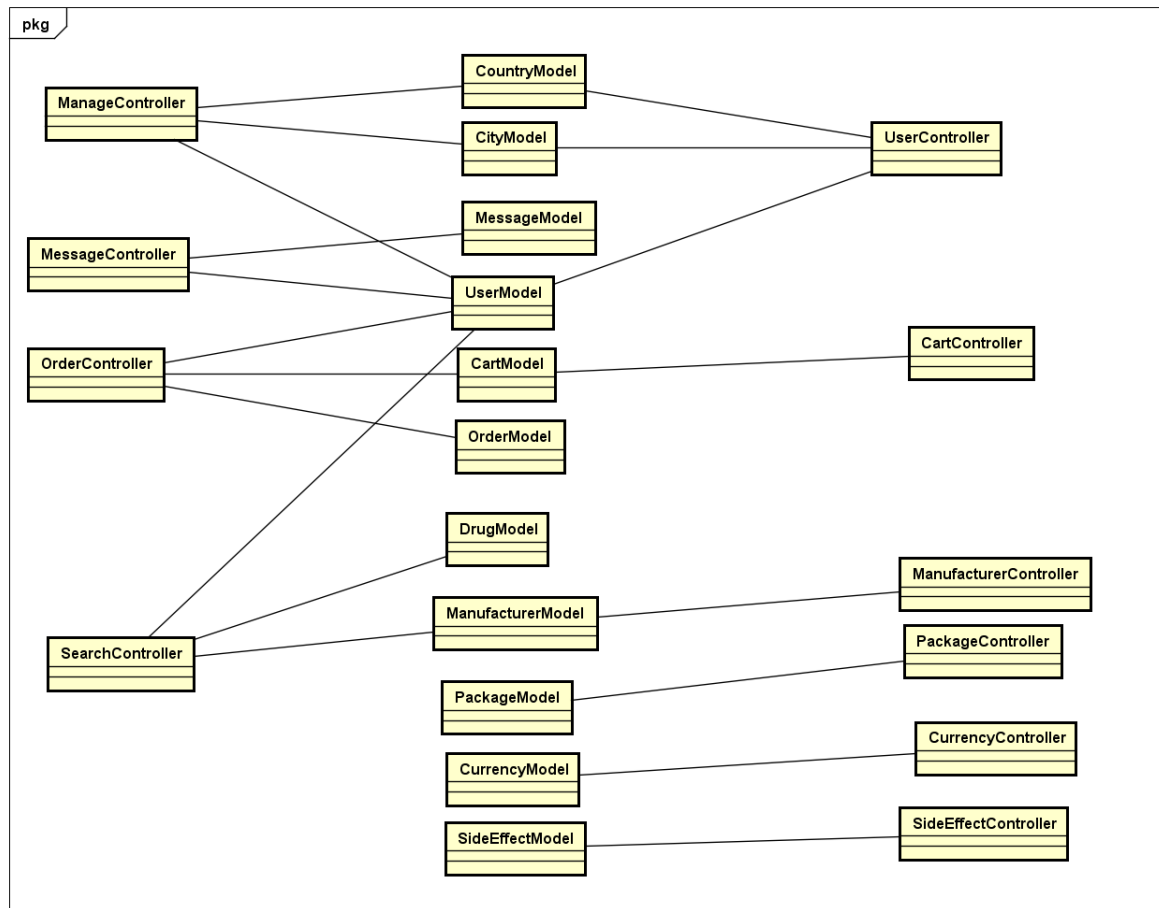
Slika 15. Ostali obrasci uporabe registriranog korisnika

Posljednji dijagram obrazaca uporabe opisuje preostale moguće operacije registriranog korisnika, a to su: upravljanje komentarima (dodavanje, uređivanje i brisanje), uređivanje vlastitog profila, pregled prošlih kupovina, dodavanja proizvoda u košaricu što daje mogućnost za dovršenje kupnje. Tada je moguće izmijeniti košaricu ili unijeti podatke za dostavu što dovršava postupak kupnje.

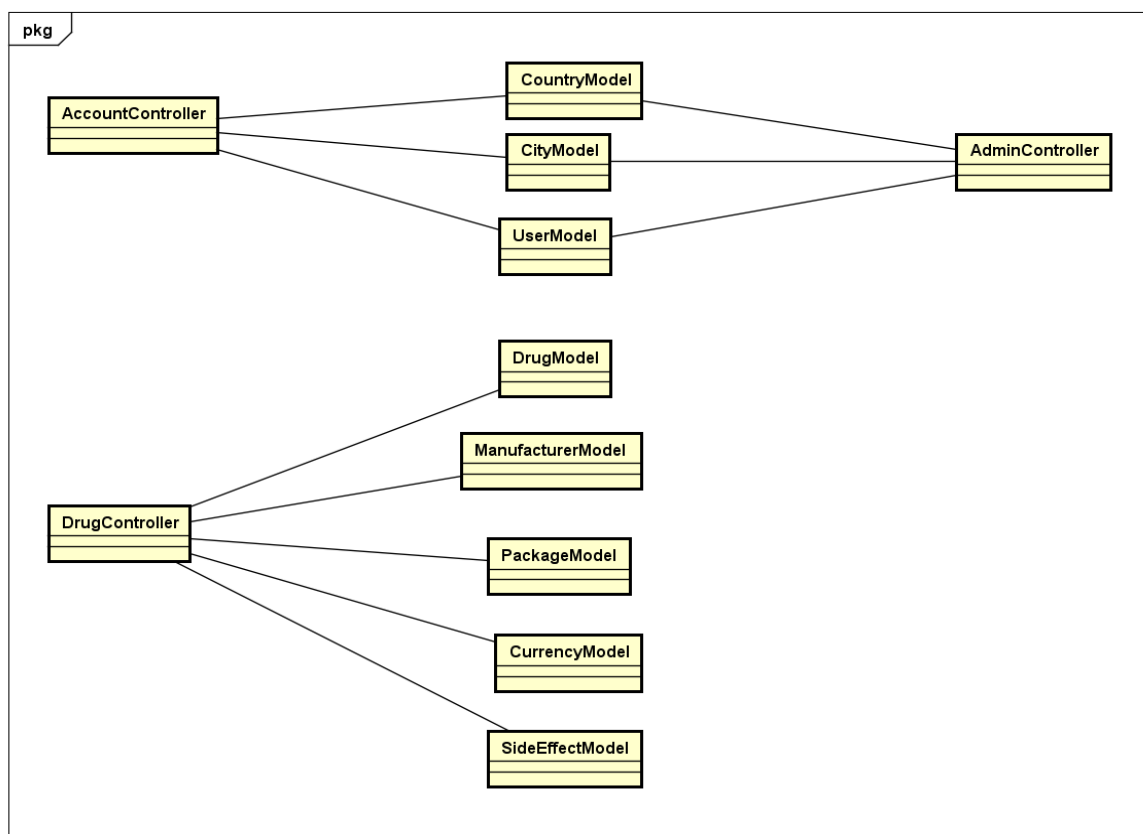
3.4. Dijagrami razreda

Dijagrami razreda prikazuju strukturu sustava preko razreda, njihovih atributa i odnosa unutar sustava. (Sommerville, 2007)

U ovome slučaju dijagrami razreda predstavljaju modele i upravljače te njihovu međusobnu povezanost.



Slika 16. Dijagram razreda (prvi dio)



Slika 17. Dijagram razreda (drugi dio)

Iz dijagrama sa slika 16 i 17 očito je da je upravljač za proizvod (*DrugController*) povezan s najviše modela što je i logično jer proizvod predstavlja glavni entitet u bazi. Upravljači za rad s korisnicima su također povezani s velikim brojem modela.

4. Korištene tehnologije

Pri izradi ljekarne Sanitas korišten je objektno orijentirani jezik C#. Kao radni okvir odabran je .Net Core, odnosno konkretno Asp.Net MVC Core koji omogućuje izradu modernih konzolnih i internetskih aplikacija te podržava rad u već navedenom C#-u.

Baza podataka rađena je u SQL-u. Korištenjem Entity Framework Corea omogućeno je komuniciranje s bazom preko C# klasa, odnosno MVC modela. Entity Framework Core koristi objektno-relacijsko mapiranje što omogućuje sinkronizaciju baze podataka putem sustava migracija. Također, obrađuje podatke iz baze lokalno putem komponente zvane DataSet koja predstavlja stvarne podatke pohranjene lokalno u memoriju. Entity Framework Core također izvodi automatske SQL upite. (Smith, 2018)

Za potrebe autentifikacije korisnika korištena je Identity biblioteka koja omogućuje sigurniji i brži rad jer pruža već gotove komponente koje se lako integriraju u postojeće klase. Upravo ta biblioteka omogućuje funkcionalnost prijave i registracije, kao i ostalih operacija koje uključuju rad korisnika.

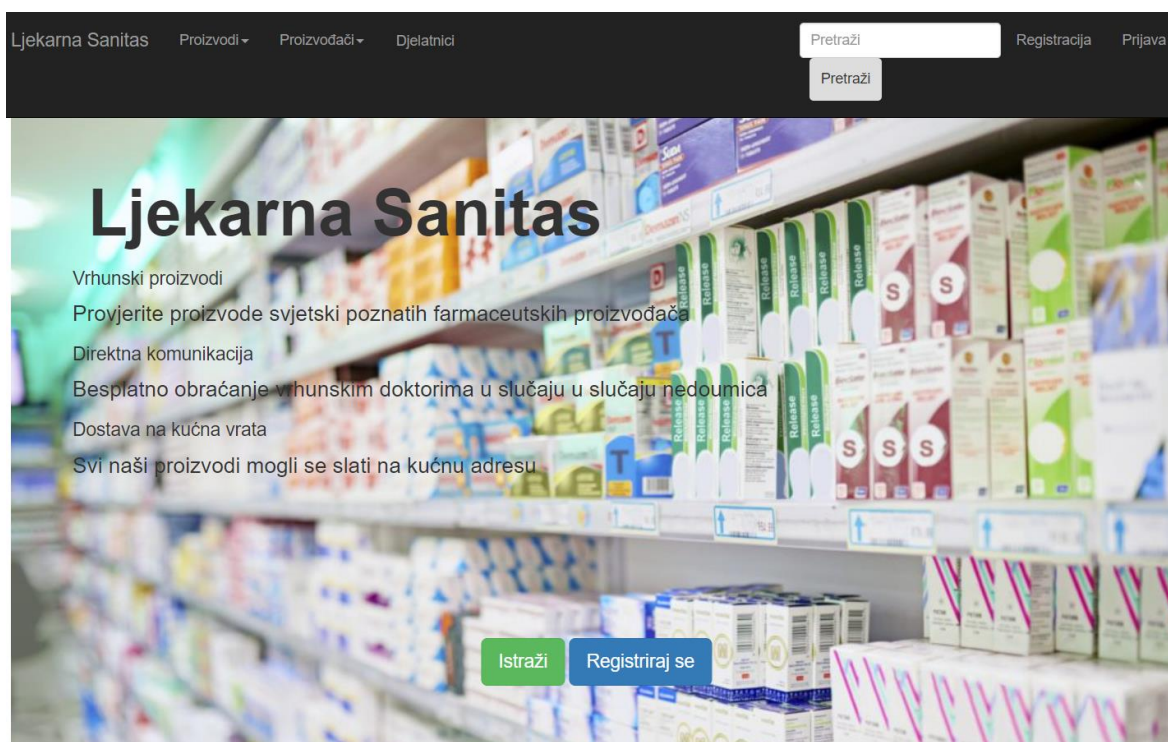
Sučelje portala izgrađeno je pomoću HTML5, CSS3 i JavaScript jezika.

HTML (*HyperText Markup Language*) predstavlja prezentacijski jezik za izradu web stranica. On nije programski jezik te služi samo za opis hipertekstualnih dokumenata, odnosno dokumenta koji sadrži poveznice. Poveznice unutar HTML dokumenata povezuju dokumente u uređenu hijerarhijsku strukturu i time određuju način na koji posjetitelj doživljava sadržaj stranice. Osnovni građevni element svake stranice su znakovi (*tags*) koji opisuju kako će se nešto prikazati u internetskom pregledniku. (HTML, 30. travanj 2018., HTML, <https://hr.wikipedia.org/wiki/HTML> , 3. lipanj 2018.)

CSS (*Cascading Style Sheets*) je stilski jezik koji se rabi za uređivanje izgleda i rasporeda dokumenta napisanog u HTML-u. (CSS, 27. svibanj 2017., CSS, <https://hr.wikipedia.org/wiki/CSS> , 3. lipanj 2018.)

JavaScript je skriptni programski jezik koji se izvršava u internetskom pregledniku na strani korisnika. JavaScript s AJAX tehnikom (*Asynchronous JavaScript and XML*) omogućuje internetskim stranicama komunikaciju s poslužiteljskim programom, odnosno čini internetsku aplikaciju interaktivnijom i jednostavnijom za korištenje. (JavaScript, 28. listopada 2015., *JavaScript*, <https://hr.wikipedia.org/wiki/JavaScript> , 3. lipanj 2018.)

Osnovu dizajna čini Bootstrap 4 radni okvir koji ima na raspolaganju veliki broj različitih izbora za dizajn komponenti te njihovo olakšano implementiranje.



Slika 18. Izgled stranice ostvaren pomoću HTML-a, CSS-a i Bootstrapa

Osim JavaScripta, korištena je i biblioteka jQuery koja olakšava upravljanje HTML-om.

5. Opis rješenja

Opisani sustav sadrži nekoliko karakterističnih dijelova, od kojih su najvažniji oni koji se tiču korisnika, a to su dijelovi za logiranje ili registraciju, dio s prikazima proizvoda, proizvođača i ljekarnika te dijelovi u kojima se ostvaruje kupnja i šalje upit ljekarniku.

Što se tiče admina, najvažniji dijelovi su oni u kojima se omogućuju dodavanje, uređivanje i brisanje stavke, odnosno konkretnog lijeka, proizvođača, nuspojave, bolesti, pakiranja, mjerne jedinice i valute.

Za ljekarnika je najbitniji dio s porukama koji mu omogućuje odgovaranje na postavljene upite, dok je vlasniku portala najbitniji dio s upravljanjem administratora.

U predstojećim karakterističnim ekranima, prikazan je izgled i je opis dotičnog karakterističnog dijela.

Ljekarna Sanitas Proizvodi ▾ Proizvođači ▾ Djelatnici Pretraži.. Registracija Prijava

Registracija

Ime

Prezime

Email

Lozinka

Potvrdite lozinku

Adresa

Grad
10000 Zagreb ▾

Hrvatska

[Registriraj se](#)

Slika 19. Karakteristični ekran registracije

Karakteristični ekran registracije predstavlja prvu potrebnu akciju korisnika za pristupanje svim posebnostima portala. Korisnik je dužan upisati svoje ime, prezime, adresu elektroničke pošte, lozinku, adresu i mjesto s pripadajućim poštanskim brojem. Sva polja moraju biti ispunjena, a u slučaju unosa elektroničke pošte koja već pripada nekom korisniku, prikazat će se greška te će se uputiti korisnika da promijeni upisanu elektroničku poštu. U slučaju uspješne registracije, korisnik i njegovi podatci spremaju se u bazu podataka te je spreman za rad.

Ljekarna Sanitas Proizvodi ▾ Proizvođači ▾ Djelatnici Pretraži.. Registracija Prijava

Prijava

Koristite lokalni račun za prijavu.

Email

josip@gmail.com

Lozinka

.....

☐ Zapamti me?

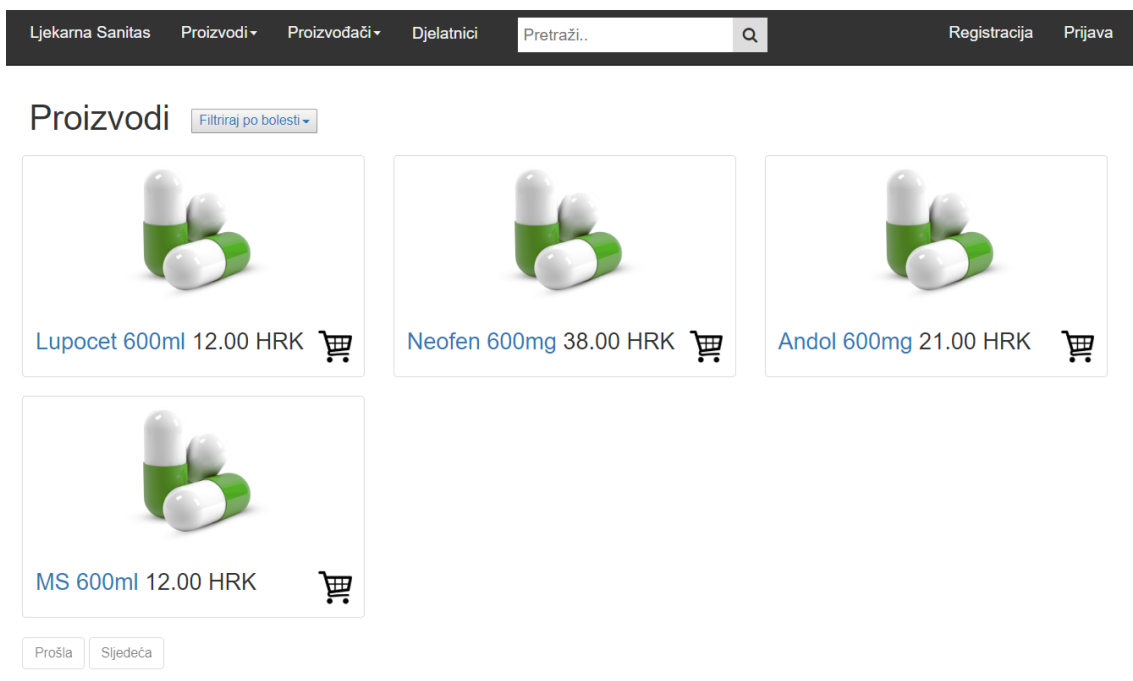
Prijava

[Registriraj se kao novi korisnik](#)

© 2018 - Sanitas

Slika 20. Karakteristični ekran prijave

Ako je korisnik registriran, mora se prvo prijaviti da pristupi svim sadržajima portala. Kao što je vidljivo sa slike 20, prilikom prijave dužan je unijeti adresu elektroničke pošte koja je i njegovo korisničko ime te pripadajuću lozinku. Ako lozinka ne pripada navedenoj adresi elektroničke pošte ili ta elektronička adresa uopće ne postoji, prikazuje se pripadajuća pogreška. Ako je sve ispravno, korisnik se uspješno prijavljuje te je preusmjeren na dio s prikazom svih proizvoda.



Slika 21. Karakteristični ekran prikaza proizvoda

Slika 21 prikazuje ekran s prikazom svih proizvoda. Za svaki proizvod dostupna je slika, cijena i njena valuta, slika košarice te ime proizvoda. Pritiskom na sliku košarice, odabrani proizvod sprema se u košaricu, dok pritiskom na ime proizvoda otvara se stranica s detaljnim informacijama o proizvodu. Na vrhu ekrana nalazi se filter proizvoda, a na dnu ekrana dva gumba koji omogućuju prelazak s prošle na sljedeću stranicu ili obrnuto.



Lupocet 600 ml

Cijena: **12.00 HRK**
Datum proizvodnje: **01-01-2018**
Datum isteka valjanosti: **01-01-2018**
Proizvođač: **Sandoz**
Pakiranje: **Otopina**
Način uporabe: **Oralno**

Primjenjivo za bolesti:

•**Gluakom**

Moguće nuspojave:

•**Glavobolja**

•**Temperatura**

Trenutno nema zamjenskih proizvoda

Komentari

[Komentiraj](#)



izvrsno, prekrasno, fantasticno

Marko Zacc

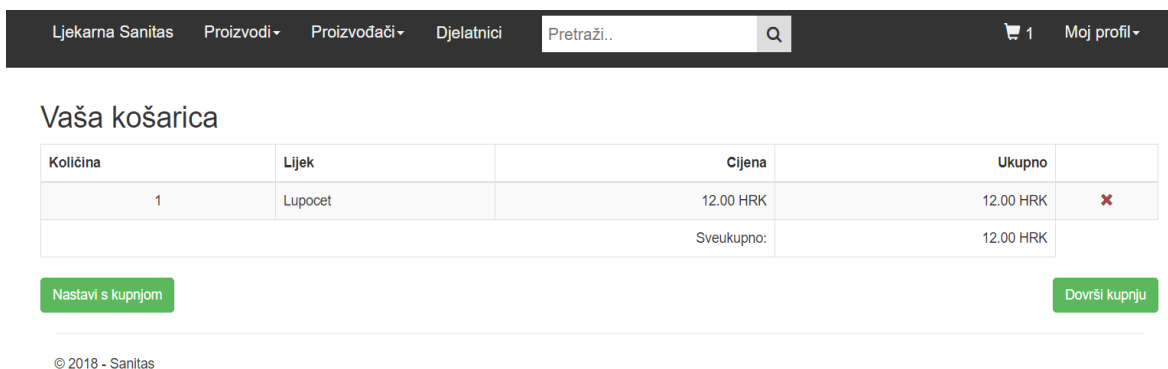
Član od: 29-05-2018

Slika 22. Karakteristični ekran detaljnog prikaza proizvoda

U ekranu detaljnog prikaza proizvoda, osim njegovog imena i cijene, dostupni su i datumi proizvodnje i isteka valjanosti, proizvođač, pakiranje, veličina, količina unutar pakiranja, način uporabe, moguće nuspojave te bolesti za koje je lijek primjenjiv.

Također, ako proizvod ima zamjenske proizvode, oni su navedeni.

Na dnu ekrana nalaze se komentari te svaki prijavljeni korisnik može ostaviti komentar koji kasnije samo on može urediti i obrisati (administrator može obrisati svaki komentar).



Slika 23. Karakteristični ekran košarice

Pritisak na gumb za dodavanje u košaricu prikazuje korisniku ekran košarice te se pojavljuje i ikona košarice na vrhu stranice. U košarici se nalaze proizvodi uz koje je navedeno njihovo ime, pojedinačna i ukupna cijena te količine. Ako korisnik odabere više količine proizvoda nego što postoji na skladištu, javlja mu se greška koja ga obavještava koliko najviše proizvoda smije kupiti.

Dostupni su i gumbi koji omogućuju smanjenje količine proizvoda u košarici te nastavak ili dovršavanje kupnje.

Unesite podatke za slanje

Ime

Prezime

Adresa

Poštanski broj

Grad

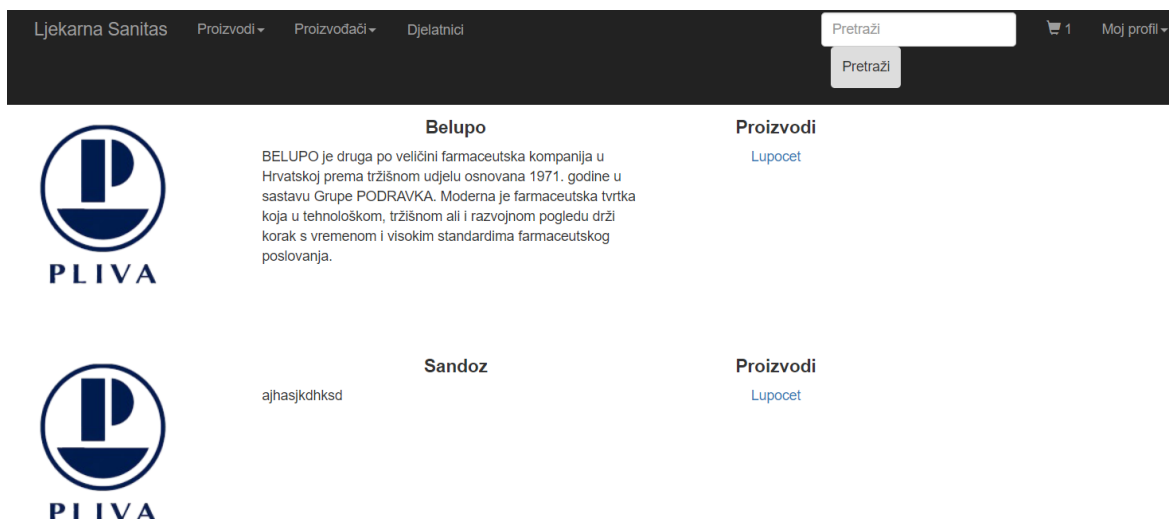
Država

Email

Pošalji

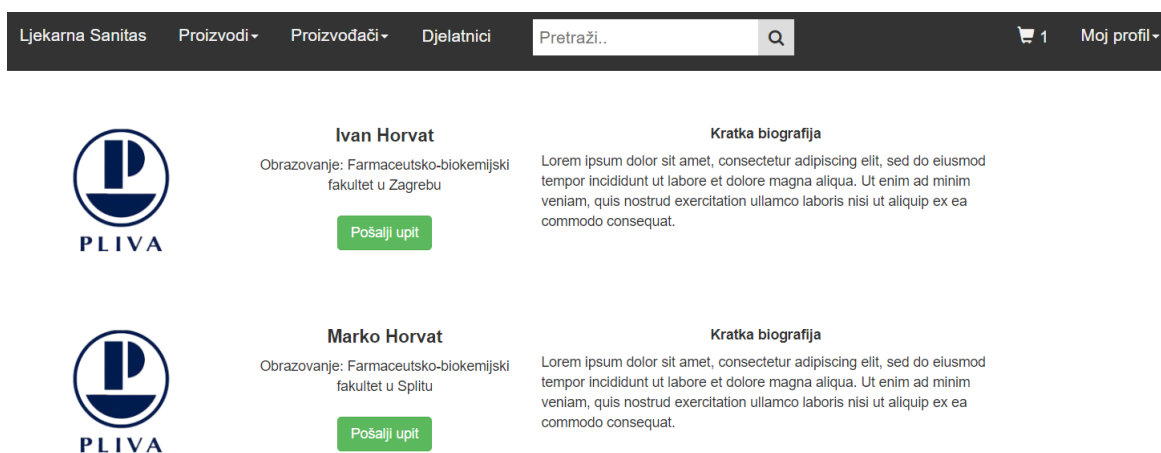
Slika 24. Karakteristični ekran dovršavanja kupnje

Nastavak kupnje vraća korisnika na pregled svih proizvoda, dok dovršavanje kupnje vodi u proces popunjavanja potrebnih podataka za slanje koji je prikazan na slici 24. Ako su svi podaci ispunjeni, košarica se prazni te odabrani proizvodi odlaze na dostavljanje, a korisnik se preusmjerava na dio s prikazom svih proizvoda.



Slika 25. Karakteristični ekran prikaza proizvođača

Prikaz, osim naziva i slike proizvođača, prikazuje i opis te proizvode tog proizvođača koji su dostupni za kupnju. Klikom na naziv proizvoda, otvara se ekran s detaljnim opisom tog proizvoda.



Slika 26. Karakteristični ekran prikaza ljekarnika

Prikaz svih djelatnika, odnosno ljekarnika na portalu, osim imena i slike, sadrži i informacije o njegovom obrazovanju te kratku biografiju. Također, postoji i gumb koji omogućuje slanje upita određenom ljekarniku.

Pritisak na taj gumb vodi neprijavljenog korisnika na dio s prijavom, a prijavljenog na ekran sa slanjem upita koji je prikazan na idućoj slici. Poslana poruka sprema se u spremnik korisnika koji ju je poslao, kao i ljekarnika kojemu je namijenjena.

Nova poruka


Primatelj

Ivan Horvat ▾

Unesite poruku

Pošalji


Slika 27. Karakteristični ekran slanja upita

Ljekarna Sanitas Proizvodi ▾ Proizvođači ▾ Djelatnici 

Moj profil ▾ Admin ▾

Ljekarnici

Pretraži: | [Cijeli popis](#)

Datum registriranja	Ime i prezime	
29-05-2018	Ivan Horvat	<input type="button" value="Uredi"/> <input type="button" value="Obriši"/>
<div> <div>  <div> <p>Ivan Horvat</p> <p>Mail: horvat@gmail.com</p> <p>Obrazovanje: Farmaceutsko-biokemijski fakultet u Zagrebu</p> <p>Djelatnik od: 29-05-2018</p> <p>Adresa:</p> <p>Pantovcak 21 10000 Zagreb Hrvatska</p> </div> </div> <div> <p>Kratka biografija</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.</p> </div> </div>		
29-05-2018	Marko Horvat	<input type="button" value="Uredi"/> <input type="button" value="Obriši"/>

Slika 28. Karakteristični ekran prikaza ljekarnika preko administratorskog sučelja

Primjer administratorskog sučelja dan je na slici 28 te njemu može pristupiti jedino administrator. Za sve stavke dostupno je pripadajuće administratorsko sučelje koje mu omogućuje lakše upravljanje. Administrator može pretražiti svaku stavku po imenu preko dostupne tražilice te dodati novu. Osim već opisanih informacija o stavci koje su dostupne svima, postoje i informacije koje jedino administrator može vidjeti, u ovome slučaju kod ljekarnika to su: adresa, datum kada je postao djelatnik na portalu te adresa elektroničke pošte. Za svaku stavku u administratorskom sučelju dostupno je brisanje i uređivanje.

6. Zaključak

Osnivanje farmaceutske znanosti i ljekarne kao produkta te znanosti predstavlja jedno od najvećih dostignuća čovjeka u nastojanju za poboljšanjem vlastitog života i životnog vijeka. Pojava raspostranjenosti Interneta pridonjela je novome načinu trgovine te su se i ljekarne pridružile tome trendu i tako omogućile prodaju svojih proizvoda preko interneta.

Temeljni zadatak ljekarne Sanitas bio je omogućavanje internetske kupovine farmaceutskih proizvoda, no sustav je dodatno nadograđen s dijelovima koji omogućuju korisniku slanje upita ljekarnicima, kao i ostavljanje komentara na sve proizvode. Te dvije stavke omogućile su da se ljekarna Sanitas uvelike razlikuje od drugih portala sa sličnim aktivnostima.

Na portalu svakodnevno radi više administratora kao i dežurnih ljekarnika te u slučaju manjka administratora ili ljekarnika, vlasnik odnosno administrator, može dodati novoga.

Korištenjem MVC arhitekture koja pruža trodijelnu strukturu, sustav je dobio na stabilnosti, kvaliteti i lakšoj organizaciji. Asp.Net MVC Core omogućio je korištenje postojećih komponenti za lakše stvaranje modernih internetskih aplikacija što je uvelike smanjilo nepotreban napor prilikom njene izrade.

Primjena HTML-a, CSS-a, JavaScripta i Bootstrapa omogućila je da portal ima ugodan dizajn te dinamično i interaktivno okruženje.

Kod ljekarne Sanitas, kao i kod svih aplikacijskih sustava, ima mjesta za dodatnu izgradnju. Jedan od tih prijedloga bilo bi postavljanje tjednih akcija i sniženja na određene proizvode. Davanje popusta korisnicima koji često kupuju te mogućnost kartičnog plaćanja su također prijedlozi koji bi se mogli razmatrati u budućnosti. Isto tako, portal bi se mogao prilagoditi za osobe izvan Republike Hrvatske.

7. Literatura

- [1] Adam Freeman, Pro ASP.NET MVC 5, Apress, 2013.
- [2] Ian Sommerville, Software engineering, 8th ed., AddisonWesley, 2007.
- [3] John Smith, Entity Framework Core in Action, 1st ed., Manning, 2018.
- [4] Wikipedia, HTML, 30.5.2018.
<https://hr.wikipedia.org/wiki/HTML>
- [5] Wikipedia, CSS, 30.5.2018.
<https://hr.wikipedia.org/wiki/CSS>
- [6] Wikipedia, JavaScript, 30.5.2018.
<https://hr.wikipedia.org/wiki/JavaScript>

8. Naslov, sažetak i ključne riječi

Realizacija portala za potporu aktivnostima ljekarne

Sažetak

U ovome radu prikazano je implementiranje portala za potporu aktivnostima ljekarne, odnosno sustav ljekarne Sanitas. U uvodnom dijelu opisan je problem i dana je motivacija za stvaranjem navedenog portala. Zatim su objašnjeni razlozi za njegovo stvaranje te je napravljena analiza implementiranoga portala s već postojećim sustavima jednakih aktivnosti. U trećem poglavlju, opisan je izgled i funkcionalnost baze podataka, arhitektura portala te obrasci uporabe. Nakon toga je dan opis svih korištenih tehnologija. U petom poglavlju je prikazan opis rješenja koji se sastoji od karakterističnih dijelova i ekrana. Na kraju je dan zaključak te prijedlozi za daljnju izgradnju sustava.

Ključne riječi

Ljekarna, internetska trgovina, proizvod, korisnik, administrator, baza podataka, upravljač, model, pogled, C#

9. Title, abstract and keywords

Web portal to support pharmacy activities

Abstract

The implementation of web portal for pharmacy activities or Sanitas pharmacy, is described in this paper. In introduction, the problem is discussed and motivation for creation of this web portal is given. Then, the reasons for it's creation are explained along with the analysis in which Sanitas pharmacy is compared to other web portals with similar activities.

In the third chapter, the look and functionality of database is presented along with portal's architecture and use cases. After that, there's a description of used technologies.

In the fifth chapter, the description of the solution is presented which contains of characteristic parts and screens. In the end, there are conclusion and suggestions for further system development.

Keywords

Pharmacy, web shop, product, user, administrator, database, controller, model, view, C#