

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Slobodan Jovanović

**Razvoj napredne spletne trgovine z upoštevanjem
zgodovine nakupov**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Igor Rožanc

Ljubljana, 2016

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.¹

¹ V dogovoru z mentorjem lahko kandidat diplomsko delo s pripadajočo izvorno kodo izda pod katero izmed alternativnih licenc, ki ponuja določen del pravic vsem: npr. *Creative Commons* [6] in *GNU GPL* [10]. Zgornje besedilo je opis licence, ki ga po potrebi lahko tudi prilagodite. Če se kandidat odloči, da diplomskega dela ne bo izdal pod omenjenimi licencami, je potrebno zgornje besedilo spremeniti v naslednje: »Rezultati diplomskega dela so intelektualna lastnina avtorja. Za objavljanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.« **V obeh primerih pa iz končnega besedila odstranite to opombo.**

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomski nalogi prikažite sistematičen razvoj učinkovite spletne trgovine od ideje do izvedbe, ki omogoča izboljšano uporabniško izkušnjo. Pri tem izhajajte iz običajnih zahtev za spletno trgovino, na podlagi katerih načrtujte razvoj z uporabo primernih tehnologij. Jedro naloge je prikaz nadgradnje rešitve z intuitivnim uporabniškim vmesnikom, uporabniku prilagojenim prikazom izdelkov na podlagi njegovih predhodnih nakupov ter učinkovitim sistemom obveščanja. Nalogo zaključite z analizo uporabe.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Slobodan Jovanović sem avtor diplomskega dela z naslovom:

Razvoj napredne spletne trgovine z upoštevanjem zgodovine nakupov

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Igorja Rožanca,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 1. februarja 2016

Podpis avtorja:

Zahvaljujem se svoji družini, puncu, prijateljem ter sodelavcem, ki so ves čas stali ob meni ter me spodbujali v času študija. Prav tako se zahvaljujem mentorju, viš. pred. dr. Igorju Rožancu za pomoč, nasvete ter izjemno prijaznost.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Tehnologije in orodja	3
2.1	Uporabljene tehnologije.....	3
2.1.1	ASP.Net	3
2.1.2	Ogrodje Entity Framework.....	4
2.1.3	Poizvedovalni jezik LINQ.....	5
2.1.4	SUPB Microsoft SQL Server	6
2.1.5	Jezika HTML in CSS.....	6
2.1.6	Knjižnica Ajax Control ToolKit.....	7
2.2	Uporabljenjena orodja	7
2.2.1	Okolje Visual Studio Express.....	7
2.2.2	Okolje SQL Server Management Studio Express	7
Poglavje 3	Razvoj spletne aplikacije.....	8
3.1	Zahteve.....	8
3.2	Opis rešitve	8
3.3	Načrtovanje	9
3.3.1	Uporabniški tipi	9
3.3.2	Načrt podatkovne baze	12
3.3.3	Grafični vmesnik	15
3.4	Izvedba.....	16
3.4.1	Osnovne funkcionalnosti	17
3.4.2	Komunikacija s podatkovnim strežnikom.....	20

3.4.3	Napredna spletna trgovina	21
Poglavje 4	Analiza.....	31
4.1	Primerjava z drugimi spletnimi trgovinami.....	31
4.1.1	Prikaz priljubljenih izdelkov	31
4.1.2	Primerjava izgleda.....	31
4.2	Mnenje uporabnikov.....	32
Poglavje 5	Sklepne ugotovitve	33
Literatura.....		35

Seznam uporabljenih kratic

kratica	angleško	slovensko
HTML	Hyper Text Markup Language	označevalni jezik za nadbesedila
CSS	Cascading Style Sheets	kaskadne slogovne podloge
LINQ	Language INtegrated Query	intergriran poizvedovalni jezik
SQL	Structured Query Language	strukturiran povpraševalni jezik
SMTP	Simple Mail Transfer Protocol	protokol za prenos pošte
DBMS	DataBase Management System	sistem za upravljanje s podatkovno bazo
EDMX	Entity Data Model tool	orodje za modeliranje podatkov
URL	Uniform Resource Locator	enolični krajevnik vira

Povzetek

Cilj diplomskega dela je izdelati spletno trgovino z dodatnimi funkcionalnostmi, ki poleg običajnih funkcionalnosti (prijave uporabnikov, pregleda izdelkov, izvedbe nakupa) uporabnikom spletne trgovine prikaže izdelke na bolj učinkovit način ter povečuje dobiček spletni trgovini. V ta namen smo razvili mehanizem, ki omogoča kupcu prilagojen prikaz izdelkov.

Najprej so na kratko omenjene tehnologije, ki so bile uporabljene pri razvoju. Kot osnova pri razvoju spletne strani je bil uporabljen programski jezik C#. Prav tako pa so bila uporabljena ogrodja in knjižnice (ASP.Net, Entity Framework, LINQ) ter ostale spletne tehnologije (HTML, CSS, AJAX). Za shranjevanje in manipulacijo s podatki so bile zgrajene tabele v MS SQL podatkovni bazi.

V ostalih poglavjih se dotaknemo zahtev, ideje ter same logike rešitve. Pokažemo načrt in kako določene funkcionalnosti delujejo v primeru različnih tipov uporabnikov. Osredotočimo se na prikaz sistema za učinkovit prikaz izdelkov, enostavnega uporabniškega vmesnika ter sistema za obveščanje uporabnikov. Razvoj zaključimo z analizo, kjer naredimo primerjavo s podobnimi rešitvami, ter povprašamo uporabnike za mnenje. Za konec omenimo tudi kje so se pojavile težave in zaključimo z opažanji kaj bi lahko izboljšali v prihodnosti.

Ključne besede: spletna trgovina, spletne tehnologije, statistika, LINQ, Asp.Net

Abstract

The goal of thesis is to develop a typical web shop application with some additional functionality. This functionality enables web shop customers to browse products in a more efficient way and thus makes shop more profitable. For this purpose, we developed a specific mechanism that handles product presentation in customer adapted way.

First we describe technologies used for development. Programming language C# is presented shortly as well as some other frameworks (ASP.net, Entity framework,), libraries (LINQ) and other web technologies (HTML, CSS, AJAX). For storing and manipulating data a database with tables in MS SQL database is created.

Furthermore we take a look at requirements, idea and logic of solution. We present solution design and present how specific functionality behaves in case of different user types. We present a solution analysis where a comparison with other similar solutions and user tests are shown. Finally we discuss problems during the development and possibilities about the future improvements.

Keywords: web shop, web technologies, statistics, LINQ, Asp.Net

Poglavje 1 Uvod

Z rastočim številom uporabnikov svetovnega spleta v zadnjih letih se povečuje obseg nakupovanja preko spletnih trgovin. Po rezultatih projekta, ki ga je opravil Center za metodologijo in informatiko znotraj Fakultete za družbene vede, naj bi se delež oseb, ki so opravile spletni nakup v Sloveniji, povečal iz 15% v letu 2006 na 39% v letu 2012 [1]. Podatki Statističnega urada Republike Slovenije kažejo podoben trend glede števila oseb, ki so opravile spletne nakupe. Tako naj bi v letu 2014 nakupovalo preko spleta že 26% oseb, v letu 2009 pa 14% [2]. Podobna slika je tudi v svetovnem merilu, kjer beležijo vsako letno rast števila oseb, ki opravi spletni nakup [3].

Posledica rasti spletnega nakupovanja je tudi število spletnih trgovin. Danes večina večjih trgovin ponuja svoje storitve tudi na spletu, kjer obveščajo svoje kupce o popustih ter ponudbi izdelkov. Predvsem v športni, tekstilni in tehnični panogi je veliko spletnega nakupovanja, medtem ko v živilski panogi to ni tako pogost pojav. Zaradi tega je v živilski panogi še veliko prostora za razvoj spletne trgovine, saj se večina še vedno odloča za obisk klasičnih trgovin. Omeniti je potrebno, da kupci v živilskih trgovinah velikokrat kupujejo izdelke v nekem ustaljenem vzorcu, kar pomeni, da se lahko izdelki ponavljajo v njihovih nakupih. Opazno je tudi v drugih spletnih aplikacijah, ki se ukvarjajo s ponujanjem izdelkov ali storitev (npr. Booking¹, Kuponko²), da imajo sisteme, ki na nek način spremljajo uporabnike. Kuponko omogoča svojim uporabnikom v nastavitvah profila izbiro kategorij, ki jih največ zanimajo. Pri Bookingu pa opazamo malce drugačen pristop. Njihov sistem beleži zgodovino naročanj in poizvedovanj ter na podlagi teh podatkov obvešča uporabnike o posebnih ponudbah.

Na podlagi teh spoznanj smo se odločili za izdelavo spletne trgovine, ki jo bomo nadgradili na tak način, da jo naredimo učinkovitejšo ter enostavnejšo za uporabo tako za uporabnike kot za skrbnike.

¹<http://www.booking.com>

²<http://www.kuponko.si>

Zato je bil glavni cilj implementacija sistema, ki omogoča bolj učinkovit ter prilagodljiv prikaz izdelkov glede na posameznega uporabnika, enostaven uporabniški vmesnik ter obveščanje uporabnikov. Poleg tega smo želeli dodati analizo podatkov, ki bi skrbnikom ponudilo statistične podatke o prodaji ter omogočalo lažje odločanje o nadaljnji prodajni strategiji.

Na začetku diplomske naloge bomo predstavili tehnologije in knjižnice, saj so nam nekatere zelo olajšale delo. Kasneje bomo predstavili zgradbo podatkovne baze ter nekaj besed namenili tudi o grafičnem vmesniku. Največji del diplome je namenjen predstavitvi rešitve ter njeni analizi.

Poglavje 2 Tehnologije in orodja

Razvoj spletne rešitve je potekal v programskem jeziku C# s pomočjo ogrodja ASP.Net [4]. Ogradje je narejeno predvsem za potrebe razvoja spletnih rešitev. V povezavi z ASP.Net ogrodjem smo uporabili še knjižnici Entity Framework [6,7] ter LINQ [8]. Knjižnici nam poenostavita povezovanje s podatkovno bazo ter manipulacijo s podatki.

Podatkovno bazo smo zgradili v Microsoft SQL Server SUPB [8]. Uporabniški vmesnik je bil narejen s pomočjo HTML in CSS označevalnim jezikom. Vključili smo tudi ASP.Net Ajax Control Toolkit knjižnico, ki ponudi dodatne gradnike in funkcionalnosti v uporabniškem vmesniku.

2.1 Uporabljene tehnologije

2.1.1 ASP.Net

ASP.Net je Microsoftovo razvojno ogrodje, ki omogoča razvoj dinamičnih spletnih strani kar pomeni da se vsebina spletne strani lahko spreminja glede na uporabnika in njegove zahteve [4]. V večini primerov se dinamične spletne strani povezujejo na podatkovne baze iz katerih črpajo podatke ter jih prikažejo uporabniku. Trenutna zadnja verzija je ASP.Net 5 ki je bila izdala v letu 2015. Vključuje nove lastnosti kot so Linux, OSX podpora ter podpora za knjižnice NODE [21], AngularJS [20]...

Ogradje omogoča tri glavne programske modele in sicer spletne obrazce (ang. Web Forms), spletne strani (ang. Web Pages) ter model Model-View-Controller. Vsak model ima svoje specifike in odločitev razvijalca je, katera je bolj ustrezna.

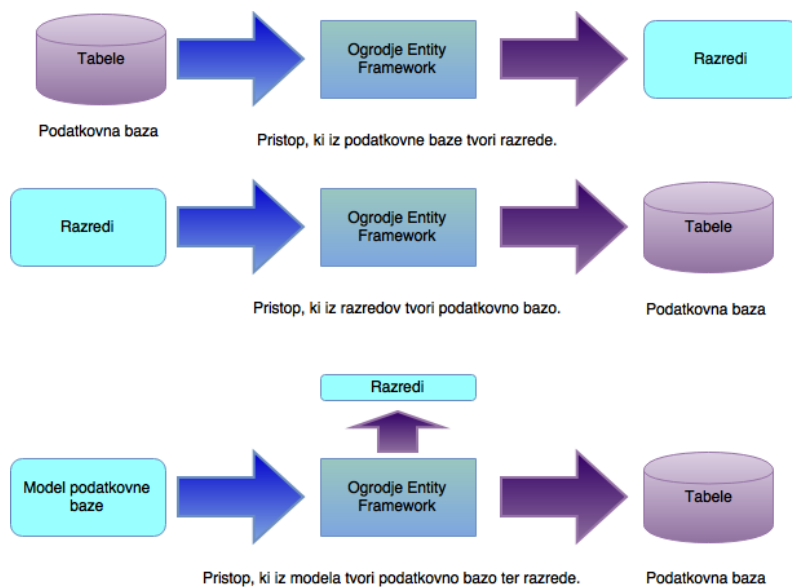
Spletni obrazec je model, ki je visoko nivojsko orientiran in je najstarejši. Za prikaz vsebine uporablja gradnike kot su gumbi ter tabele. Trenutno največ novih projektov uporablja Model-View-Controller model, vendar je še veliko obstoječih projektov v tem modelu.

2.1.2 Ogradje Entity Framework

Ogradje Entity Framework omogoča objektno-relacijsko preslikavanje podatkovne baze. Poenostavi dostop do baze, saj razvijalcu ni potrebno vedno znova pisati povezovalne logiko. Vsebuje orodje za vizualno kreiranje podatkovnih model ter omogoča delo z različnimi vrstami podatkovnih baz (Microsoft SQL server [9], Oracle [22], DB2 [23]). Dobra lastnost je tudi ta, da se ob spremembi strukture v podatkovni bazi samodejno posodobijo modeli.

Ogradje v verziji 6 ponuja tri različne pristope k uporabi (slika 1):

- **Code First**, pri katerem razvijalec prvo spiše razrede ter na podlagi le teh ogradje generira podatkovno bazo.
- **Model First** pristop določa, da razvijalec generira podatkovne modele ter relacije v orodju EDMX.
- **Database First** na podlagi obstoječe podatkovne baze generira podatkovne modele.



Slika 1: Vizualni prikaz pristopov

2.1.3 Poizvedovalni jezik LINQ

LINQ (ang. Language Integrated Query) je integriran poizvedovalni jezik, ki je ena izmed knjižnic Microsoft .Net ogrodja, namenjen prvotno za C# programski jezik [8]. LINQ je bil prvič vključen v ogrodje .NET leta 2007 v verziji .NET 3.5.

LINQ razširja programski jezik, ki omogoča pisanje poizvedovalnih ukazov podobnim SQL ukazom nad podatkovnimi strukturami. Predstavlja vmesni člen med objektno orientiranim programskim jezikom ter poizvedovalnim SQL jezikom. Najbolj smiselna uporaba je nad relacijskimi bazami, saj LINQ knjižnica pretvori poizvedbe v SQL ukaze.

V osnovi ponuja dva načina pisanja ukazov in sicer v poizvedovalni sintaksi, ki je podobna pri pisanju SQL ukazov (slika 2), ter lambda sintaksi (slika 3) [25].

Primer poizvedovalne sintakse:

```
var results = from c in SomeCollection
              where c.SomeProperty < 10
              select new {c.SomeProperty, c.OtherProperty};

foreach (var result in results)
{
    Console.WriteLine(result);
}
```

Slika 2: Poizvedovalna sintaksa

Primer lambda izraza:

```
var results =
    SomeCollection
        .Where(c => c.SomeProperty < 10)
        .Select(c => new {c.SomeProperty, c.OtherProperty});

foreach (var result in results)
{
    Console.WriteLine(result.ToString());
}
```

Slika 3: Lambda izraz

LINQ podpira več operatorjev kot so *Where*, *Select*, *GroupBy*, *Join*, *OrderBy*, *ToList*, *Sum*, *Take*, *Union* ipd.

Preprostost uporabe ter hitra implementacija povpraševalnih ukazov v programski kodi omogočata hiter razvoj aplikacij. Prav tako pa LINQ vsebuje standardne funkcije za manipulacijo nad objekti, ki so shranjeni v pomnilniku.

2.1.4 SUPB Microsoft SQL Server

Microsoft SQL Server je sistem za upravljanje s podatkovnimi bazi razvit s strani Microsofta [9]. Glavne funkcionalnosti sistema so shranjevanje in poizvedovanje nad podatki na podlagi zahtevkov različnih programskih aplikacij. Ponuja tudi druge storitve kot so analiza podatkov, izdelava poročil, integracijske storitve itd. Deluje na osnovi T-SQL programski jezika, ki je razširitev osnovnega programskega jezika SQL [24].

Orodje, ki omogoča delo s podatkovno bazo, se imenuje Microsoft SQL Server Management Studio [10]. Orodje vsebuje skriptni urejevalnik ter grafično orodje za delo s podatki.

2.1.5 Jezika HTML in CSS

HTML je označevalni jezik, s katerim gradimo spletne strani [11]. S pomočjo značk postavljamo različne elemente, ki omogočajo prikaz in vnos podatkov. Brskalniki iz HTML datotek preberejo značke ter jih interpretirajo in izrišejo na zaslon.

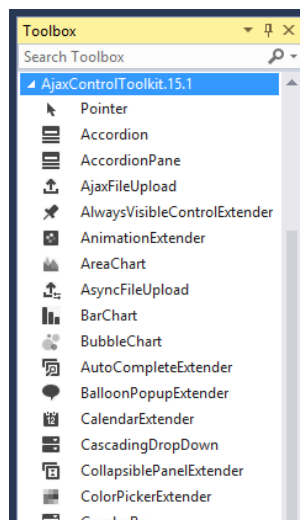
Zadnja verzija se imenuje HTML5 ter prinaša nove elemente in lastnosti ter omogoča boljše delo z grafiko in mediji.

CSS je jezik, za predloge, ki skrbijo za vizualno predstavitev spletnih strani [13]. Z njimi določamo pravila z prikazovanje HTML elementov na strani. S pravili lahko določimo kakšne barve, velikosti in obrobe naj imajo HTML gradniki ter postavitev teh na spletni strani. Prav tako lahko skrbimo za dogodke, ki jih uporabnik strani sproži nad elementi, npr. postavitev miškekega kazalca na povezavo.

CSS3 je najnovejša verzija CSS, ki vsebuje dodatne funkcionalnosti kot so novi selektorji, zaobljeni koti, podpora več brskalnikom, nove oblikovne lastnosti...

2.1.6 Knjižnica Ajax Control Toolkit

Ajax Control Toolkit je knjižnica, ki ponuja več kot 40 gradnikov (slika 4) in funkcionalnosti [15]. Ponuja prikaz podatkov v obliki grafov, spustne sezname, menije, gradnik za kontrolo moči gesla in mnoge druge. Knjižnica izboljša samo odzivnost in interakcijo med uporabnikom in spletno stranjo.



Slika 4: Gradniki knjižnice Ajax ControlTool Kit

2.2 Uporabljen orodja

2.2.1 Okolje Visual Studio Express

Microsoft Visual Studio je Microsoftovo razvojno okolje, ki je namenjeno razvoju programov, spletnih strani, spletnih aplikacij in storitev [16]. Podpira programske jezike C, C++, C#, Visual Basic, Python, Ruby. Vsebuje različna orodja, ki omogočajo izdelavo grafičnih vmesnikov, razredov, podatkovnih shem. Za lažje odkrivanje napak uporabljamo vgrajeni razhroščevalnik. Obstajajo več različic programa kot so Community, Professional, Enterprise, Express. Trenutna zadnja verzija ki je izšla se imenuje Visual Studio 2015.

2.2.2 Okolje SQL Server Management Studio Express

SQL Management Studio je program namenjen za urejanje in upravljanje s podatkovnimi bazami na strežnikih [17]. Vsebuje skriptni urejevalnik prav tako pa grafično orodje za delo z objekti in dodatki. Omogoča kreiranje tabel, pogledov, procedur, funkcij, opravi ter uporabniških računov, s katerimi omejujemo dostope uporabnikom do podatkov. Objektni raziskovalec ponuja pregled in urejanje vseh objektov, ki se nahajajo v podatkovni bazi.

Poglavje 3 Razvoj spletne aplikacije

3.1 Zahteve

Implementirati želimo mehanizem, ki bo znal glede na uporabnika, prikazati primeren nabor podatkov oz. izdelkov. Spletna trgovina mora imeti več različnih tipov uporabnikov, ki morajo imeti določene pravice in omejitve. Uporabnikov mora omogočati pregled in naročanje izdelkov, skrbnikom v zalednem sistemu pa pregled naročil in urejanje podatkov. Ker je spletna aplikacija namenja predvsem prodaji živilskih izdelkov lahko pričakujemo, da bo veliko uporabniki kupovalo izdelke v nekem ustaljenem vzorcu. To pomeni, da mora mehanizem iz zgodovine nakupov ugotoviti uporabnikove najpogosteje kupljene izdelke. Poleg tega želimo spletno aplikacijo narediti čim bolj dobičkonosno, zato mora mehanizem pri prikazu izdelkov upoštevati tudi ta pogoj.

Pri implementaciji moramo vzeti v obzir tudi odzivnost in hitrost spletne aplikacije. V primeru, ko bi veliko uporabnikov hkrati dostopalo do spletne aplikacije ali pa bi bil velik nabor podatkov za računanje, bi lahko mehanizem upočasnil delovanje aplikacije. Zato je potrebno mehanizem implementirati tako, da čim manj upočasnjuje delovanje ostalih funkcionalnosti.

Spletna aplikacija mora omogočati skrbnikom prikaz statistični podatkov o zgodovini nakupov. Podatki morajo biti predstavljeni na čim bolj enostaven in vizualen način. V spletno aplikacijo želimo vključiti tudi obveščanje uporabnikov po elektronski pošti ob različnih dogodkih. Prav tako bomo obveščali skrbnike v primeru nepravilnega delovanja sistema.

3.2 Opis rešitve

Na začetku bomo ustvarili osnovno spletno trgovino in podatkovno bazo. Dodali bomo spletne obrazce za prijavo, registracijo, pregled izdelkov, pregled naročil, urejanje izdelkov in oseb. Kasneje bomo spletno trgovino nadgradili z dodatnimi funkcionalnostmi, kot so učinkovit prikaz izdelkov, prikaz statistike ter obveščanje uporabnikov, ki jo bodo naredili bolj učinkovito.

Začetna rešitev za prikaz najpogosteje kupljenih izdelkov je bila, da sistem prebere podatke iz baze ter ugotovi, katere izdelke je uporabnik največkrat kupil. Ta proces bi se izvajal vsakič, ko bi uporabnik želel pregledovati izdelke. Tu se je pojavilo vprašanje hitrosti mehanizma, saj bi vnovično ugotavljanje lahko trajajo predolgo časa. Zato smo se odločili za implementacijo

dodatne tabele v podatkovni bazi, kjer bi za vsakega uporabnika posebej hranili podatke o najpogosteje kupljenih izdelkih. S tem imamo podatke vnaprej pripravljene in mehanizem manj obremenjuje sistem.

Prikazovanje statistike uporabnikom bomo implementirali s pomočjo knjižnice Ajax Control Toolkit, saj vsebuje gradnike, ki prikazujejo podatke v obliki grafov. S tem bomo naredili hiter in vizualen pregled statističnih podatkov. Za pošiljanje elektronske pošte bomo razvili metodo, za komunikacijo s SMTP strežnikom, ki posreduje poslano elektronsko pošto k naslovniku.

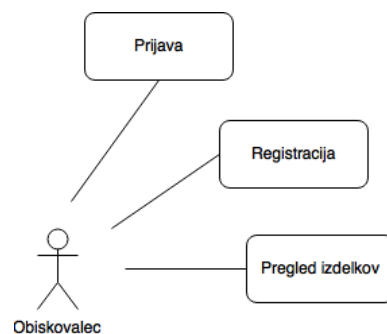
3.3 Načrtovanje

Na začetku razvoja smo se odločili za tri glavne tipe uporabnikov in sicer **obiskovalec**, **registrirani uporabnik** in **administrator**. Med razvojem smo ugotovili, da potrebujemo še dodatni tip uporabnika, ki smo ga poimenovali **moderator**. Vsaka skupina uporabnikov ima določene pravice, ki dovoljujejo ali omejujejo uporabo določenih funkcionalnosti. V nadaljevanju bomo z diagrami predstavili uporabniške tipe. V tem koraku razvoja smo načrtovali tudi tabele in relacije med njimi v podatkovni bazi ter funkcionalnosti, ki jih bomo implementirali.

Načrtovali smo tudi grafični vmesnik oz. postavitev na spletni aplikaciji. Elemente smo logično razdeli glede na potrebe določenih izpisov in funkcionalnosti.

3.3.1 Uporabniški tipi

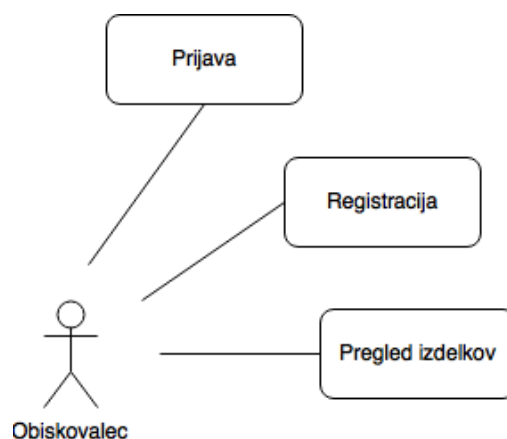
Uporabniške tipe lahko razdelimo na dva dela in sicer na prednji in zaledni del. Prednji del je namenjen obiskovalcem ter kupcem, ki pregledujejo izdelke in izvajajo naročila, zadnji del pa služi skrbnikom spletne trgovine, ki urejajo izdelke, pregledujejo naročila ter obveščajo kupce.



Slika 5: Diagram uporabniškega tip obiskovalec

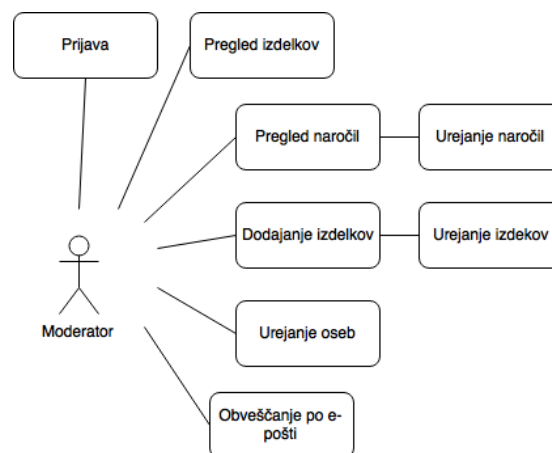
Obiskovalec je tip uporabnika, ki ima najmanj pravic, saj lahko samo pregleduje izdelke, ne more pa jih naročiti (slika 5). Lahko obišče druge podstrani z dodatnimi informacijami in kontakti ter podstrani za registracijo ali prijavo.

Pri registraciji uporabnik navede svoje ime in priimek, geslo, elektronski naslov, naslov za dostavo ter označi ali dovoljuje obveščanje po elektronskem naslovu. Po uspešni registraciji obiskovalec prejme na svoj elektronski naslov potrditev o uspešni registraciji.



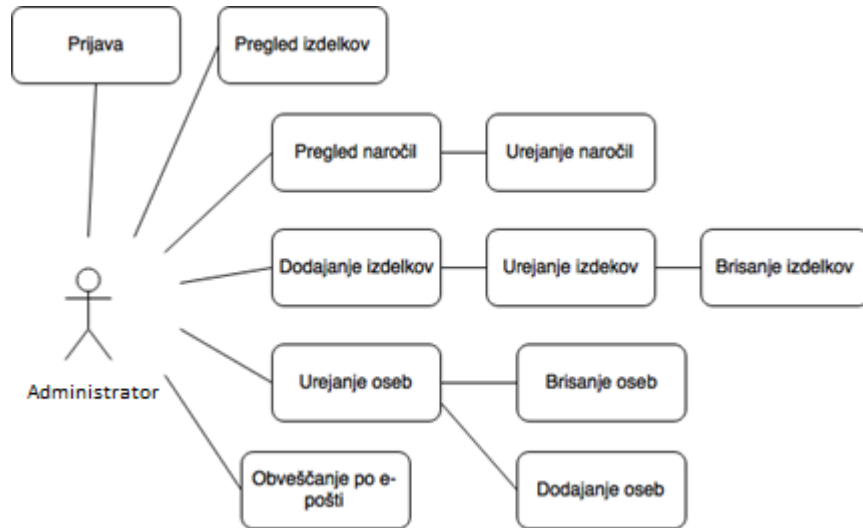
Slika 6: Diagram uporabniškega tipa uporabnik

Po registraciji in prijavi v sistem obiskovalec spletne strani dobi status **uporabnika** (slika 6). Poleg vseh pravic, ki jih ima obiskovalec, dobi uporabnik še dodatno pravico naročanja izdelkov in pregled zgodovine naročil. Lahko ureja tudi osebne podatke kot so naslov za dostavo, telefonska številka ter geslo.



Slika 7: Uporabniški tip moderator

Moderator je uporabniški tip, ki obsega zaledni del sistema in je namenjen urejanju podatkov ter dodajanju novih (slika 7). Moderator lahko ureja izdelke in dodaja nove izdelke, ne more pa jih brisati. Prav tako lahko ureja uporabnike spletne trgovine, ne sme pa jih brisati. Lahko pregleduje naročila, statistične grafe ter obvešča uporabnike po elektronski pošti. Sistem običajno vsebuje več moderatorjev.



Slika 8: Uporabniških tip Administrator

Administrator je uporabniški tip, ki ima največ pravic in prav tako predstavlja zaledni del sistema (slika 8). Vsebuje vse pravice, ki jih ima moderator, poleg teh pa še nekaj dodatnih kot so brisanje izdelkov, brisanje oseb ter dodajanje oseb z moderatorskimi pravicami.

3.3.2 Načrt podatkovne baze

Pomemben del razvoja aplikacije je načrtovanje podatkovne baze. Na začetku razvoja smo naredili nekaj osnovnih tabel in povezav med njimi. Med razvojem smo dodajali nove tabele ter dodatne attribute za potrebe posameznih funkcionalnosti.

Glavne tabele so *User*, *Product* ter *Order*. Prav tako smo dodali nekaj pomožnih kot so *UserType*, *ProductCategory*, *UserStatistics* ter *Employee*.

- Tabela *User* predstavlja vse registrirane stranke spletne trgovine. V tabeli se ob registraciji shranijo osnovni podatki kot so:
 - `userID,`
 - `name,`
 - `lastName,`
 - `password,`
 - `email,`
 - `address,`
 - `inform,`
 - `usertypeID.`

- Tabela *Employee* vsebuje podatke o vseh zaposlenih. Vsebuje podatek o tipu zaposlenega, ki določa, kakšne pravice bo imel. Ostali podatki so:
 - `employeeID,`
 - `name,`
 - `lastname,`
 - `email,`
 - `address,`

- telephone,
 - usertypeID.
-
- Tabela *UserType* je pomožna tabela, ki hrani podatek o uporabniških tipih ki nastopajo v aplikaciji. Povezana je na osnovno tabelo *User* ter *Employee*, kjer se pojavi kot tuj ključ.

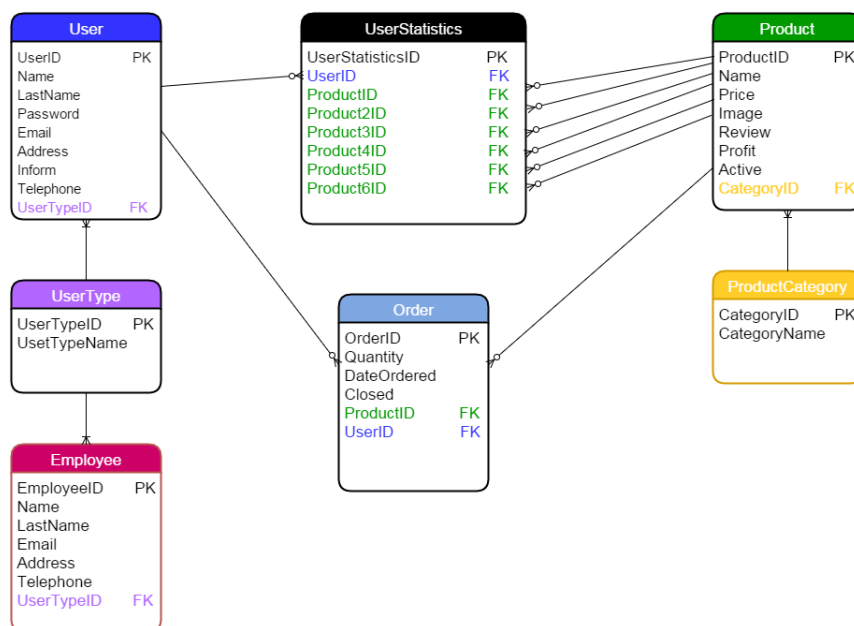
 - Tabela *Product* shranjuje podatke o vseh artiklih ki so v ponudbi. Vsebuje naslednje podatke:
 - productID,
 - name,
 - price,
 - image,
 - review,
 - profit,
 - active,
 - categoryID.

 - Tabela *ProductCategory* je pomožna tabela, ki se povezuje na osnovno tabelo *Product*, kjer nastopa kot tuj ključ. Hrani podatke o različnih kategorijah artiklov, ki nastopajo v aplikaciji.

 - Tabela *Order* vsebuje podatke o naročilih, ki so jih izvedli uporabniki. V njej med drugimi podatki najdemo tuja ključa, ki kažeta na tabeli *User* ter *Product*. Ostali podatki so:

- orderID,
- quantity,
- dateOrdered,
- closed,
- productID,
- userID.

Tabela *UserStatistics* je pomožna tabela, v katero zapisujemo podatke o uporabnikovih najpogostejših nakup. Ob registraciji se generira zapis, v katerega zapišemo podatek o uporabniku. V kasnejših procesih se zapišejo še podatki o artiklih, ki jih uporabnik največkrat naročil.



Slika 9: Relacijski diagram podatkovne baze

3.3.3 Grafični vmesnik

V tem razdelku se bomo dotaknili načrtovanja grafičnega vmesnika oz. postavitve spletne aplikacije (slika 10). Vmesnik smo načrtovali tako, da omogoča učinkovit in enostaven prikaz izdelkov in ostalih funkcionalnosti. Grafični vmesnik je razdeljen na tri dele in sicer na zgornji, glavni in spodnji del. V vsakem delu se nahaja eden ali več dodatnih delov, ki ločujejo posamezne grafične elemente.

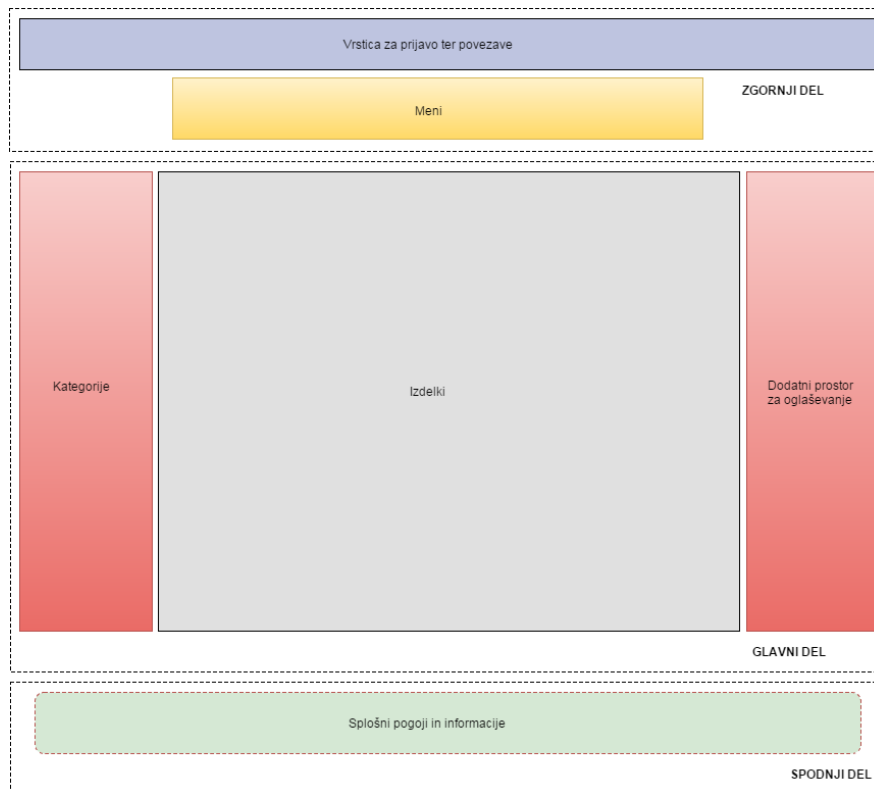
Zgornji del je vsebuje dva dela:

- V prvem delu se nahaja vrstica, kjer so povezave do podstrani za prijavo ter izpis imena uporabnika, ki je prijavljen. Prav tako se prijavljenim uporabnikom prikažejo povezave do podstrani, kjer lahko pregledujejo zgodovino nakupov, ter urejajo osebne podatke, skrbnikom spletnih strani pa se pokaže povezava do nadzorne plošče.
- Drugi del vsebuje vrstico kjer se nahaja meni. V meniju so povezave na domačo stran, izdelke ter do kontaktnih podatkov.

Glavni del je razdeljen na tri dele:

- V levem delu se nahaja meni v katerem so povezave do posameznih kategorij izdelkov. Ob kliku na povezavo sistem preko parametra, ki se nahaja v URL naslovu, dobi `CategoryID`, na podlagi katerega prikazuje izdelke iz izbrane kategorije.
- Srednji del prikazuje izdelke. Je dinamičen, tukaj se vsebina spreminja glede na uporabnikove zahteve.
- V desnem delu je prostor za oglaševanje.

V spodnjem delu se nahaja vrstica, kjer so navedeni podatki o splošnih pogojih ter druge informacije.



Slika 10: Postavitev spletne aplikacije

3.4 Izvedba

Po načrtovanju podatkovne baze, grafičnega vmesnika ter definicije uporabniških tipov smo se lotili kodiranja spletne aplikacije. Najprej smo tvorili osnovno stran (ang. master page), kjer smo s pomočjo CSS označevalnega jezika določili obliko spletne aplikacije. Na podlagi te osnovne strani smo izpeljali več drugih spletnih obrazcev, na katere smo postavljali gradnike ali pa izpisovali podatke. Osnovna stran deluje kot predloga. S tem omogoča, da lahko hitro dodajamo nove spletne obrazce.

Za tem smo naredili nekaj osnovnih spletnih obrazcev kot so recimo obrazec za prijavo, registracijo, kontaktne podatke, prikaz izdelkov, pregled naročil, pregled registriranih uporabnikov ipd.

S temi postopki smo naredili temelje, nad katerimi smo gradili druge funkcionalnosti, ki bodo bolj podrobno opisane v nadaljevanju.

3.4.1 Osnovne funkcionalnosti

Registracija in prijava uporabnika

Ena izmed osnovnih funkcionalnosti ki jih uporabnik lahko izvaja je to, da se registrira v sistem (slika 11). Ob registraciji uporabnik izpolni osnovne podatke ter pošlje zahtevek za registracijo. Sistem preveri ali uporabniško ime že obstaja in v primeru, da obstaja, prikaže obvestilo, da je uporabnik s tem imenom že registriran, v nasprotnem primer pa izvede registracijo. Uporabnik dobi potrditveno pošto na svoj elektronski naslov in s tem se proces registracije zaključi.



The image shows a registration form interface. At the top, there are three navigation buttons: 'HOME' (yellow), 'SHOP' (blue), and 'CONTACTS' (red). Below these is a registration form with the following fields and controls:

- UserName:
- Name:
- Lastname:
- Password:
- Confirm Password:
- E-mail:
- Deliver Address:
- Telephone number:
- Allow informing to mail:
- Register
- Cancel

Slika 11: Registracijski obrazec

Naslednji korak je prijava. Za prijavo v sistem mora uporabnik poznati uporabniško ime ter geslo, ki ju je določil ob registraciji. Ob uspešni prijavi uporabnika preusmerimo na začetno spletno stran in prikažemo njegovo ime v zgornjem desnem kotu.

Urejanje osnovnih podatkov

Uporabnik lahko naknadno ureja svoje osnovne podatke, vendar ne vseh. Ureja lahko podatke kot so naslov, telefonska številka, geslo. Prav tako pa lahko omogoči ali onemogoči obveščanje po elektronski pošti.

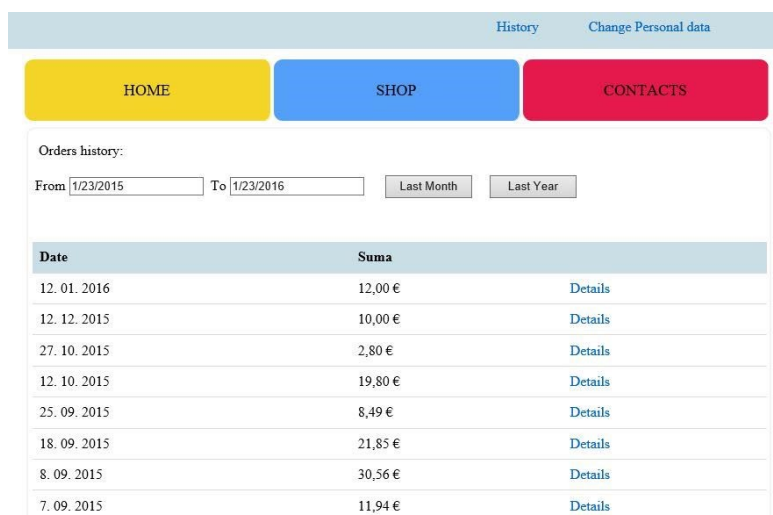
Naročanje

Uporabnik izvrši naročanje tako, da pri vsakem izdelku, ki ga želi kupiti, navede količino. Sistem mu nato izpiše vse izdelke, ki jih je označil, ter zahteva še zadnjo potrditev. V primeru, da uporabnik potrdi, se na nakup zabeleži v tabeli naročil v nasprotnem primeru pa se izvajanje procesa prekine.

Zgodovina naročil

Do zgodovine naročil lahko vsak uporabnik dostopa preko povezave, ki se nahaja v zgornjem delu aplikacije (slika 12). Za prikaz naročil mora uporabnik izbrati začetni in končni datum, na podlagi katerih se izvedbe poizvedba na podatkovno bazo. Poleg izbire datuma smo dodali še dva gumba za hiter izpis naročil. Ob kliku na prvega se pokažejo vsa naročila, ki so bila naročena v zadnjem mesecu, ob kliku na drugega pa vsa naročila naročena v zadnjem letu.

V spisku naročil so vidni podatki o datumu naročila ter skupnem znesku. Vsako naročilo ima tudi povezavo da podrobnosti, kjer ob kliku uporabnika prikažemo vse izdelke, ki se nahajajo v okviru tega naročila (slika 13). Ob izdelkih pokažemo še dodatne informacije o ceni ter naročeni količini.



Date	Suma	
12. 01. 2016	12,00 €	Details
12. 12. 2015	10,00 €	Details
27. 10. 2015	2,80 €	Details
12. 10. 2015	19,80 €	Details
25. 09. 2015	8,49 €	Details
18. 09. 2015	21,85 €	Details
8. 09. 2015	30,56 €	Details
7. 09. 2015	11,94 €	Details

Slika 12: Pregled naročil



Date	Product	Quantity	Price
12. 01. 2016	Črni hlebec	1	2,80 €
12. 01. 2016	Zelje	2	1,45 €
12. 01. 2016	Goveje meso	1	6,30 €
			All: 12,00 €

[Back](#)

Slika 13: Podrobnosti naročila

Skrbniki spletne aplikacije imajo skoraj identični prikaz zgodovine naročil le z nekaj spremembami. Prav tako jim je omogočena izbira glede na začetni in končni čas. Skrbnikom smo dodali tri gumbе, ki nastavljajo filtre glede na čas ter stanje naročila . Prvi gumb pokaže naročila v zadnjem tednu, drugi gumb pa naročila v zadnjih 30 dneh. Poleg teh dveh je še možnost, da pokažemo naročila, ki niso zaključena. Skrbniki prav tako lahko pogledajo podrobnosti posameznega naročila, ter ga imajo možnost zaključiti.

Nadzorna plošča

Nadzorna plošča predstavlja del, ki je namenjen administratorjem ter moderatorjem spletne aplikacije (slika 14). Pri delovanju upošteva pravice uporabniških tipov in recimo moderatorjem preprečuje brisanje podatkov. Glavni del nadzorne plošče je pregled in urejanje izdelkov ter bližnjice do drugi funkcionalnosti.

Funkcionalnosti, ki jih nadzorna plošča omogoča:

- (a) pregled vseh uporabnikov, izdelkov in naročil,
- (b) dodajanje izdelkov,
- (c) deaktiviranje izdelkov,
- (d) urejanje podatkov o uporabnikih, izdelkih ter zaključevanje naročil,
- (e) obveščanje uporabnikov po e-pošti,
- (f) pregled statistik.



	ProductID	CategoryID	Name	Price	image	Review	Profit	Active
Edit	1007	5	Beli hlebec	2,99	kruh1.jpg	1kg	1,30	<input checked="" type="checkbox"/>
Edit	1008	5	Črni hlebec	2,80	kruh2.jpg	900g	1,40	<input checked="" type="checkbox"/>
Edit	1009	5	Polbeli hlebec	2,50	kruh3.jpg	1kg	1,00	<input checked="" type="checkbox"/>
Edit	1010	5	Štruca s semeni!	2,70	kruh4.jpg	800g	1,10	<input checked="" type="checkbox"/>
Edit	1012	5	Bela štruca	1,99	kruh5.png	1kg	0,70	<input checked="" type="checkbox"/>
Edit	1013	1	Pišč. bedra	5,00	meso1.jpg	1kg	1,50	<input checked="" type="checkbox"/>
Edit	1014	1	Pišč. file	5,50	meso2.jpg	1kg	2,10	<input checked="" type="checkbox"/>
Edit	1015	1	Goveje meso	6,30	meso3.jpg	1kg	2,80	<input checked="" type="checkbox"/>
Edit	1016	1	Svinjsko meso	7,00	meso4.jpg	1kg	2,35	<input checked="" type="checkbox"/>

Slika 14: Nadzorna plošča

Dodajanje izdelkov

Na nadzorni plošči se nahaja obrazec, kjer lahko skrbniki dodajajo nove izdelke (slika 15). Pri dodajanju morajo izpolniti podatke kot so ime, cena, dobiček ter iz spustnega seznama izbrati kategorijo novega izdelka. Poleg tega morajo določiti sliko za prikaz, ki jo lahko izberejo v spustnem seznamu ali pa dodajo novo.

Slika 15: Obrazec za dodajanje izdelkov

3.4.2 Komunikacija s podatkovnim strežnikom

Pri razvoju sta nam zelo pomagala ogrodje Entity Framework ter knjižnica LINQ, saj sta izjemno pohitrila razvoj. Aplikacija veliko komunicira s podatkovnim strežnikom, zato smo pridom izkoristili njune funkcionalnosti.

Ob implementaciji novih funkcionalnosti smo z ogrodjem Entity Framework hitro vzpostavili povezavo med tabelo v podatkovni bazi ter razredom v programskem okolju. Knjižnica LINQ pa nam je omogočila enostavno in hitro pisanje poizvedovalnih ukazov.

Na sliki 16 vidimo primer nastavitve preslikave med SQL podatkovnimi tabelami in razredi v programskem okolju. Slika 17 prikazuje dva LINQ poizvedovalna ukaza, kjer so med drugimi uporabljene funkcije grupiranja, razvrščanja in selekcije ter pretvorbe podatkov v seznam.

```
public DbSet<User> User { get; set; }
public DbSet<Order> Order { get; set; }
```

Slika 16: Primer preslikav


```

var list = db.Order.Where(u => u.UserID == UserIdparam).OrderByDescending(u => u.date).Take(20).ToList();

var groupedList = list.GroupBy(g => g.ProductID).Select(g => new { Symbol = g.Key, Count = g.Count() })
    .OrderByDescending(g => g.Count).ThenBy(g => g.Symbol).Take(6).ToList();

```

Slika 17: LINQ poizvedovalna ukaza

3.4.3 Napredna spletna trgovina

Poleg osnovnih smo implementirali še dodatne funkcionalnosti, ki spletno trgovino naredijo bolj učinkovito in uporabno. Dodatne funkcionalnosti lahko razdelimo v dva sklopa in sicer na tiste, ki so namenjene uporabnikom in tiste, ki so namenjene skrbnikom. V tem poglavju bomo pojasnili delovanje posameznih funkcionalnosti ter kako smo do njih prišli.

Sistem za učinkovitejši prikaz izdelkov

Sistem za učinkovitejši prikaz izdelkov je del zalednega sistema aplikacije. Sistem je definiran z nekaj postopki in deluje samostojno, kar pomeni skrbnikom ni potrebno skrbeti za delovanje procesa. Glavni namen sistema je prikaz uporabnikovih njegove najpogosteje kupljenih izdelkov. Ko se uporabnik prijavi in postavi na pregled izdelkov se mu najprej prikažejo tisti, ki jih je največkrat kupil. Za namen je bil razvita metoda, ki iz tabele `UserStatistics` prebere uporabnikove priljubljene izdelke.

Poleg teh sistem prikaže še druge izdelke, ki prinašajo večji dobiček spletni trgovini in predstavljajo po nekaj izdelkov iz vsake kategorije. Za ta namen smo med razvojem dopolnili tabelo o izdelkih s tem, ko smo dodali podatek, koliko dobička prinaša določen izdelek. Na podlagi tega podatka sistem s pomočjo LINQ poizvedbe izbere po nekaj najbolj dobičkonosnih izdelkov iz vsake kategorije (slika 18). Metoda v nabor podatkov ne vključi izdelkov, ki se nahajajo v naboru najpogosteje kupljenih izdelkov, da ne bi prihajalo do prikaza podvojenih izdelkov.

```

public static List<Artikel> additionalArtikels(List<Int32> productIds, ShopContext db)
{
    try
    {
        var notContainsProduct = db.Artikel.Where(a => !productIds.Contains(a.ProductID)).ToList();
        var categories = db.Category.Select(c => new { c.CategoryID }).ToList();
        List<Artikel> missingData = new List<Artikel>();
        foreach (var item in categories)
        {
            var temp = notContainsProduct.Where(c => c.CategoryID == item.CategoryID).OrderByDescending(c => c.Profit).Take(2).ToList();
            missingData = missingData.Concat<Artikel>(temp).ToList();
        }
        missingData.Shuffle<Artikel>();
        return missingData;
    }
    catch (Exception ex)
    {
        SendMail("Napaka v funkciji additionalArtikels/" + ex.ToString(), "Sistemska napaka", "████████████████████");
        throw ex;
    }
}

```

Slika 18: Metoda za nabor izdelkov

Ob registraciji uporabnika se tvori zapis v tabeli, kjer beležimo statistiko. Dogodek, ki proži osveževanje statistike, se zgodi po tem, ko uporabnik opravi nakup. Prav tako smo omejili nabor podatkov, ki je vključen v računanje. V računanje vključimo samo tiste izdelke, ki so bili izbrani v zadnjih 20 nakupih. Ta omejitev predstavlja dodatno pohitritev sistema.

Logiko sistema lahko povzamemo v naslednjih korakih:

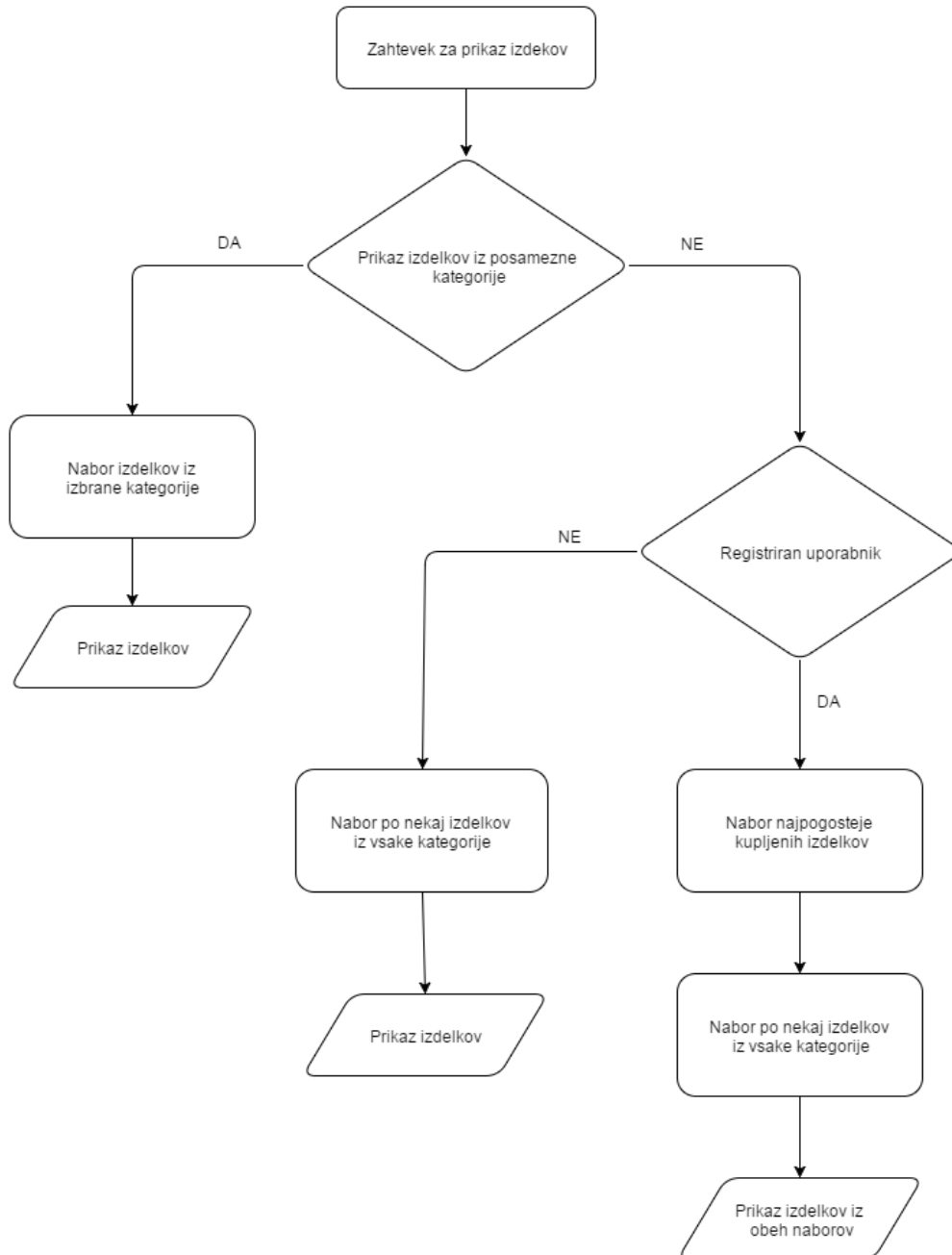
- (a) Registracija uporabnika – ob registraciji uporabnika tvorimo zapis v tabeli `User` (v primeru da uporabniško ime že obstaja, prekinemo nadaljnje izvajanje ter obvestimo, da uporabniško ime že obstaja).
- (b) Kreiranje zapis v tabeli `UserStatistics` – če se je uporabnik uspešno registriral v tem koraku kreiramo zapis v tabeli `UserStatistics`. V tem trenutku zapis vsebuje samo podatek na katerega uporabnika je vezan. Podatki o izdelkih so prazni.
- (c) Nakup – Uporabnik izvrši prvi nakup. Vnesejo se podatki o nakupu v tabelo o naročilih.
- (d) Osveževanje statistike – Če se korak (c) uspešno izvrši je to dogodek, ki pomeni, da je potrebno osveževanje statistike. V tem koraku sistem iz zadnjih 20 nakupov (v tem primeru je samo en nakup) ugotovi, kateri izdelki so se največkrat pojavili (grupacija po izdelkih) ter osveži podatke o izdelkih v zapisu tabele `UserStatistics`.
- (e) Ob naslednjem pregledovanju izdelkov se uporabniku prikazujejo osveženi podatki oz. njegovi najpogosteje kupljeni izdelki.

Prikaz izdelkov za obiskovalce se razlikuje od prikaza pri registriranem uporabniku. Uporabniku se najprej pokažejo izdelki, ki jih je najpogosteje kupil, ter nekaj dodatnih, ki prinašajo največji dobiček. Obiskovalcu se prikaže le nekaj izdelkov iz vsake kategorije. Teh nekaj izdelkov predstavlja najbolj dobičkonosne izdelke iz kategorije v katero spada. Pri pregledu izdelkov lahko uporabniki in obiskovalci izberejo posamezne kategorije in pregledujejo samo eno vrsto izdelkov.

Sistem je narejen tako, da ugotovi za kakšen tip uporabnika gre, ter ali uporabnik želi splošni pregled ali pa pregled po kategorijah. Tip uporabnika sistem ugotovi iz seje, ki se kreira v trenutku prijave uporabnika v sistem. Na podlagi teh ugotovitev se odloči, kakšen nabor izdelkov bo prikazal. Kljub temu, da se prikazan nabor izdelkov večinoma ponavlja (dokler ne

pridejo novi izdelki ali se spremeni statistika), se vrstni red prikazovanja spreminja. Zato poskrbi funkcija, ki nabore izdelkov vsakič naključno premeša.

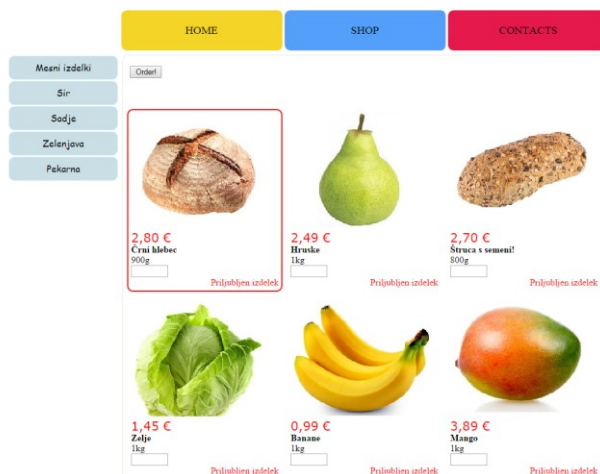
Slika 19 prikazuje diagram poteka, ki predstavlja odločitvene pogoje in postopke pri izbiri.



Slika 19: Diagram poteka

Prikaz izdelkov

Prikaz priljubljenih izdelkov se vizualno razlikuje od prikaza ostalih izdelkov, saj imajo dodatno oznako »Priljubljen izdelek«. Prav tako se obroba okoli njih obarva v rdečo barvo, ko se postavimo nad njimi z miškinim kazalcem (slika 20). Pri ostalih izdelkih se obroba obarva v drugi barvi ter ne vsebujejo nobenega napisa (slika 21).



Slika 20: Priljubljeni izdelki



Slika 21: Ostali izdelki

Za tovrstne vizualne učinke smo morali določiti nekaj oblikovnih lastnosti v CSS datoteki kot so odmiki, barve, pozicija teksta, stil obrobe itd. (slika 22). Določili smo tudi lastnosti za stanje, ko se uporabnik postavi na element z miškinim kurzorjem. Priljubljenim izdelkom spremenimo markirni element z drugim CSS selektorjem, tako da pridobi drugačne lastnosti od ostalih izdelkov.

```
#ContentPlaceholder1_PnProduct div
{
  border: 2px solid white;
  border-radius: 10px;
  padding: 5px;
  text-align: left;
  float: none;
  display: inline-block;
  margin-left: 7px;
  margin-bottom: 10px;
}

#ContentPlaceholder1_PnProduct div:hover
{
  border: 2px solid #bc9cb8;
  -webkit-transition: background-color 0.5s;
  -moz-transition: background-color 0.5s;
  -o-transition: background-color 0.5s;
  transition: 0.5s;
}
```

Slika 22: CSS Lastnosti

Pri priljubljenih izdelkih so CSS lastnosti malce drugačni pri stanju, ko se uporabnik postavi na element z miškinim kazalnikom (slika 23).

```
.panelUserArtikel:hover{
  border: 2px solid red !important;
  -webkit-transition: background-color 0.5s;
  -moz-transition: background-color 0.5s;
  -o-transition: background-color 0.5s;
  transition: 0.5s;
}
```

Slika 23: CSS lastnosti priljubljenih izdelkov

Obveščanje po elektronski pošti

Pomembna in uporabna funkcionalnost spletne aplikacije, je tudi obveščanje uporabnikov in skrbnikov po elektronski pošti. Večina sporočil se pošilja samodejno ob določenih dogodkih kot so uspešna registracija, uspešen nakup itd.

Obveščanje uporabnikov o novi akcijah ali ponudbah, pa je povsem v domeni skrbnikov, kar pomeni, da lahko določajo vsebino ter naslov sporočila. Prav tako lahko skrbniki izbirajo, katerim uporabnikom se bo sporočilo poslalo.

Imajo možnosti pošiljanja:

- vsem uporabnikom,
- uporabnikom, ki niso opravili nakupa zadnjih nekaj mesecev (število mesecev je nastavljivo),
- ter nekaj najboljšim kupcem (število kupcev je nastavljivo).

Uporabniki ob registraciji lahko označijo opcijo, če želijo, da jim spletna aplikacija pošilja sporočila. V primeru, da ne želijo obveščanj, se jim ne pošilja sporočil razen samodejnih.

Tipi sporočil, ki jih aplikacija pošilja:

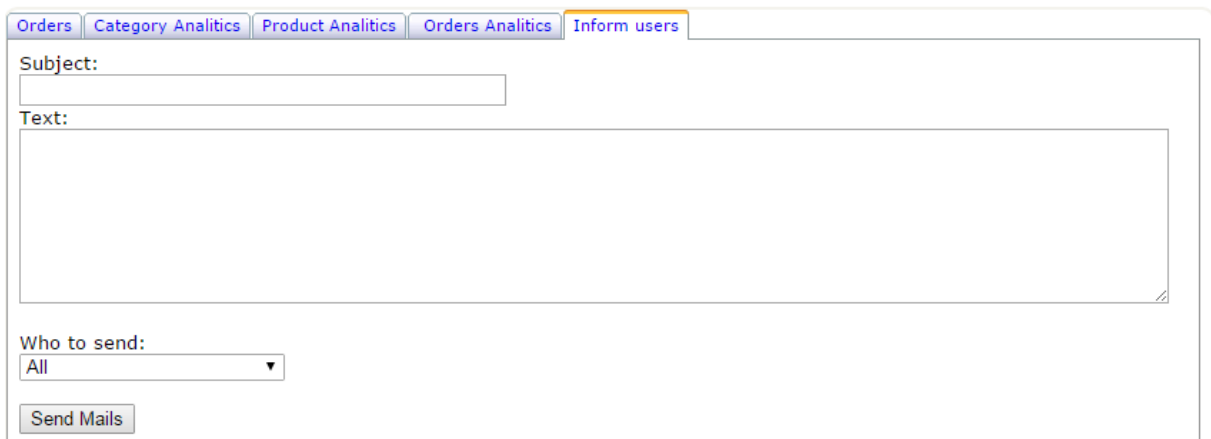
- (a) Obveščanje uporabnikov
 - Registracija uporabnika
 - Opravljen nakup
 - Ostala sporočila (akcije, ponudbe)
- (b) Obveščanje skrbnikov
 - Novo naročilo
 - Napaka v sistemu

Za pošiljanje sporočil smo spisali splošno metodo, ki za parametre dobi podatke o naslovniku sporočila, vsebini ter naslovu. To metodo sistem uporablja ob pošiljanju vseh vrst sporočil. Za potrebe razvoja in testiranja smo uporabljali posebej narejen Googlov uporabniški račun, saj registriranim uporabnikom Google omogoča uporabo SMTP strežnikov za pošiljanje odhodnih sporočil. Na sliki 24 so vidne nastavitve (vrata, naslov do strežnika, protokol), ki jih je potrebno nastaviti, da se pošiljanje uspešno izvede. Na sliki 25 je prikazan obrazec za pošiljanje elektronske pošte.

```
public static void SendMail(String text, String subject, String to)
{
    try
    {
        MailMessage mail = new MailMessage();
        SmtpClient client = new SmtpClient();
        client.Port = 587;
        client.Host = "smtp.gmail.com";
        client.EnableSsl = true;
        client.Timeout = 10000;
        client.DeliveryMethod = SmtpDeliveryMethod.Network;
        client.UseDefaultCredentials = false;
        client.Credentials = new System.Net.NetworkCredential("██████████", "██████████");

        MailMessage mm = new MailMessage("██████████", to, subject, text);
        mm.BodyEncoding = UTF8Encoding.UTF8;
        mm.DeliveryNotificationOptions = DeliveryNotificationOptions.OnFailure;
        client.Send(mm);
    }
    catch(Exception ex){
        notifError(ex.ToString(), "SendMail");
    }
}
```

Slika 24: Metoda za pošiljanje elektronske pošte



Orders Category Analytics Product Analytics Orders Analytics Inform users

Subject:

Text:

Who to send:
All ▼

Send Mails

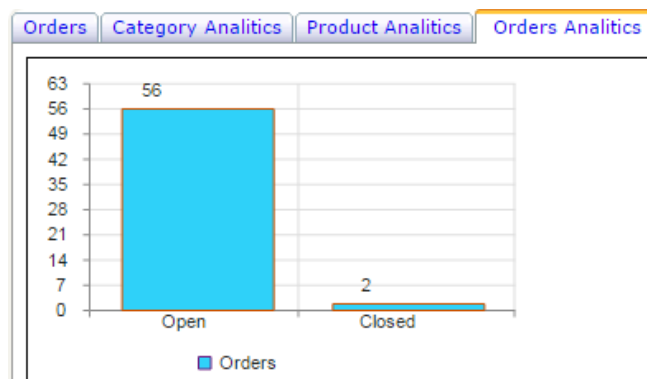
Slika 25: Obrazec za pošiljanje elektronske pošte

Statistika

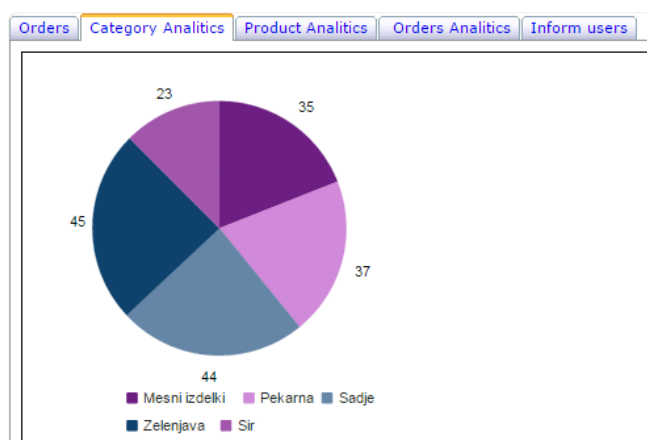
Skrbnikom spletne aplikacije je omogočeno spremljanje statističnih podatkov na nadzorni plošči. S pomočjo statistike skrbniki na enem mestu hitro lahko ugotovijo stanje naročil ter na podlagi podatkov o izdelkih se lahko odločijo o nadaljnjih morebitnih oglaševanjih za določene izdelke.

Za implementacijo statistike smo si pomagali s knjižnico AjaxControlTool Kit, ki vsebuje gradnike za prikaz grafov. Gradniki omogočajo enostavno implementacijo saj za svoje delovanje potrebujejo samo vhodne podatke. Za vhodne podatke skrbijo metode, ki berejo iz podatkovne baze ter podatke obdelajo tako, da so v obliki kot jih zahtevajo gradniki.

Na sliki 26 vidimo število zaključenih, ter še vedno aktivnih naročil, medtem ko slika 27 vsebuje graf, ki prikazuje število prodanih izdelkov glede na posamezno kategorijo.



Slika 26: Graf naročil

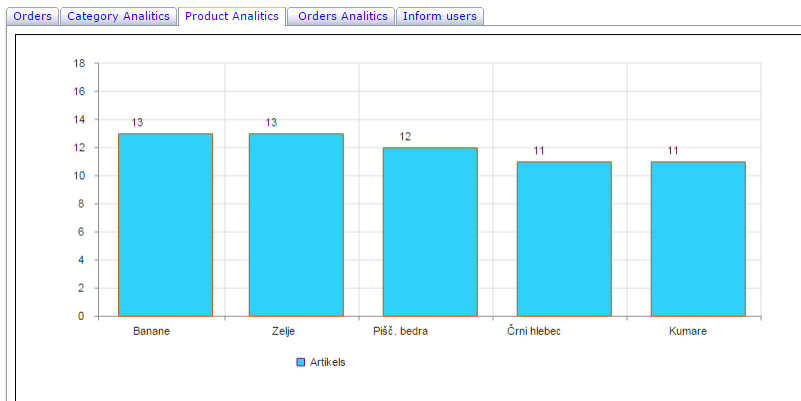


Slika 27: Graf kategorij



Slika 28: Prikaz prometa

Na sliki 28 vidimo graf, na katerem je prikazan promet v zadnjih dveh dneh.



Slika 29: Graf izdelkov

Slika 29 prikazuje graf, na katerem je prikazanih pet najbolj prodajanih izdelkov.

```
#region FillBarChart
public void FillBarChart(Dictionary<String, decimal> list)
{
    string category="";

    decimal[] values = new decimal[list.Count];
    int i = 0;
    foreach(var dt in list){
        category = category + "," + dt.Key;
        values[i]= dt.Value;
        i++;
    }
    BarChart1.CategoriesAxis = category.Remove(0, 1);
    BarChart1.Series.Add(new AjaxControlToolkit.BarChartSeries { Data = values, BarColor="#2fd1f9", Name = "Artikels" });
}
#endregion
```

Slika 30: Metoda za polnjene grafa

Na sliki 30 je prikazana metoda, ki polni graf o najbolj prodajanih izdelkih.

Poglavje 4 Analiza

Med pregledovanjem spleta smo ugotovili, da obstaja nekaj podobnih spletnih trgovin, ki ponujajo predvsem živilske izdelke, ter jih primerjali z našo spletno trgovino. Poleg tega smo za mnenje vprašali uporabnike, ki so testirali aplikacijo.

4.1 Primerjava z drugimi spletnimi trgovinami

Za primerjavo z našo trgovino smo uporabili spletne trgovine večjih trgovinskih podjetji Leclerc¹, Mercator² in Spar³. Spletne trgovine smo primerjali med seboj po funkcionalnostih ter izgledu.

4.1.1 Prikaz priljubljenih izdelkov

Leclercova spletna trgovina vsebuje podobno funkcionalnost kot smo jo implementirali v naši spletni trgovini. Trgovina omogoča prikaz priljubljenih izdelkov uporabnika, vendar pa je potrebno to funkcionalnost poiskati v meniju. Pri naši spletni trgovini so priljubljeni izdelki vidni takoj, ko se uporabnik postavi na pregled izdelkov. Pri Mercatorjevi spletni trgovini te možnosti nismo našli medtem, ko Sparova spletna trgovina ne omogoča spletnih nakupov temveč samo ponuja katalog izdelkov.

Pri spletni trgovini Leclerca smo opazili funkcionalnost, ki se nam zdi zelo uporabna in bi jo bilo smotno implementirati tudi v naši aplikaciji. Ta omogoča, da izberemo nakup iz preteklosti ter ga opravimo še enkrat.

4.1.2 Primerjava izgleda

Prikaz kategorij in izdelkov je pri Leclercovi in Sparovi trgovini narejen na podoben način kot pri naši trgovini, kajti skupine izdelkov so logično združene pod posamezne kategorije. Rezultat tega je, da lahko hitro poiščemo iskane izdelke.

¹<http://fd1-www.leclercdrive.si/default.aspx>

²<http://trgovina.mercator.si/>

³https://www.spar.si/sl_SI.html

Pri Sparovi trgovini je organizacija izdelkov malce drugačna, saj so izdelki združeni v promocijske kategorije, zato je iskanje posameznega izdelka bolj zahtevno za uporabnika.

Prednost spletnih trgovin Leclerca in Mercatora je v nakupovalnih košaricah, saj omogočajo naknadno spreminjanje količin posameznim izdelkom. To pri naši spletni trgovini trenutno še ni omogočeno. Prav tako spletni aplikaciji vsebujeta več možnosti za filtriranje izdelkov. Prednost naše aplikacije v primerjavi z drugimi je v tem, da uporabnik hitro lahko opazi izdelke, ki jih je najpogosteje kupil, saj ti izdelki odstopajo po izgledu od ostalih.

4.2 Mnenje uporabnikov

Testiranje aplikacije je izvedlo deset uporabnikov. Polovica je pregledovala samo del, ki je namenjen kupcu, polovica pa tudi dele, ki jih vidijo skrbniki. Za potrebe testiranja smo projekt objavili na strežnik ponudnika spletnega gostovanja Somee¹. Pri ponudniku spletnega gostovanja smo izbrali opcijo, ki je zastoj in je zato bolj namenjena testiranju. Kljub temu je bilo delovanje aplikacije zanesljivo in dovolj hitro.

Splošno mnenje uporabnikov je bilo, da je aplikacija dovolj pregledna in enostavna za uporabo. Zanimiv jim je bil sistem za prikazovanje priljubljenih izdelkov, saj ni potrebe po dodatnih nastavitvah ali iskanjih po aplikaciji. Predlagali so izboljšavo nakupovalne košarice, dodatna razvrščanja izdelkov po drugih kriterijih ter iskalnik, saj bi bila s tem aplikacija še bolj uporabna.

Uporabniki, ki so imeli vpogled v zaledni sistem, so pohvalil nadzorno ploščo predvsem zaradi tega, ker združuje vse funkcionalnosti na enem mestu. Prav tako jim je bil všeč prikaz podatkov v obliki grafov, saj so pregledi in vsebujejo koristne informacije. Izpostavili so tudi vgrajene filtre na pregledu naročil.

¹<https://somee.com/>

Poglavje 5 Sklepne ugotovitve

Izdelovanje diplomske naloge nam je predstavljalo izziv, saj se do sedaj še nismo srečali s tako širokim problemom. Prav tako smo se z nekaterimi tehnologijami, ki smo jih vključili v razvoj, prvič srečali, zato je bilo samo izdelovanje toliko bolj zanimivo in poučno. Sproti smo iskali rešitve za določene probleme, hkrati pa so se nam odpirale nove ideje in možnosti. Trenutna rešitev, bi ob smiselni nadgradnji, zadoščala za dejansko spletno trgovino.

Nekajkrat smo prišli do primera, ko nam je manjša sprememba oziroma dodelava povzročila nedelovanje drugih delov aplikacije. Zato je pomembno je omeniti, da smo med razvojem prišli do ugotovitve, da je zelo pomembna dobra analiza problema, dobro načrtovanje rešitve ter predvidevanja o nadaljnjih izboljšavah.

Med razvojem so se nam pojavile nove ideje, ki jih nismo vseh mogli realizirati. Predvsem je veliko možnosti na izboljšanju oblike in dodatnih funkcionalnosti. Uporabnikom bi lahko omogočili filtriranje izdelkov in razvrščanje izdelkov glede na ceno in ime. Izboljšali bi lahko tudi delovanje in prikaz nakupovalne košarice. Poleg obveščanja po elektronski pošti bi lahko razmislili o implementaciji obveščanja preko sms sporočil.

Za skrbnikom spletne aplikacije bi lahko omogočili dodajanje popustov in akcij na določene izdelke. Omogočili bi jim vpogled v stanje zalog izdelkov ter jih obveščali, ko določenega izdelka zmanjka na zalogi.

Glede na popularnost mobilnih telefonov in tablic bi bil smotrni razmislek tudi o implementaciji android aplikacije za mobilne naprave.

Literatura

- [1] (2016) Raba interneta v Sloveniji [Online]. Dosegljivo:
<http://www.ris.org/index.php?fl=2&lact=1&bid=9399&parent=26p1=276&p2=285&p3=1354&p4=1351&p4=1356&id=1356> [Dostopano 24.01.2016]
- [2] (2016) Statistični urad republike Slovenije [Online]. Dosegljivo:
<http://www.stat.si/StatWeb/glavnanavigacija/podatki/prikazistaronovico?IdNovice=6680> [Dostopano 24.01.2016]
- [3] (2016), Svetovni podatki o rabi interneta. Dosegljivo:
<http://www.statista.com/statistics/251666/number-of-digital-buyers-worldwide/>
[Dostopano 24.01.2016]
- [4] (2016) ASP.Net [Online]. Dosegljivo:
<http://www.asp.net/> [Dostopano 24.01.2016]
- [5] Matthew MacDonald, "ASP.Net 4 in C# 2010", 2010
- [6] (2016) Entity framework [Online]. Dosegljivo:
<https://msdn.microsoft.com/en-us/data/ef.aspx> [Dostopano 24.01.2016]
- [7] (2016) Entity framework [Online]. Dosegljivo:
<http://www.entityframeworktutorial.net/> [Dostopano 24.01.2016]
- [8] (2016) LINQ [Online]. Dosegljivo:
<https://msdn.microsoft.com/en-us/library/bb397926.aspx> [Dostopano 24.01.2016]
- [9] (2016) SUPB Microsoft SQL Server [Online]. Dosegljivo:
<https://www.microsoft.com/en-us/server-cloud/products/sql-server/> [Dostopano 24.01.2016]
- [10] (2016) SQL Server Management studio [Online]. Dosegljivo:
<https://msdn.microsoft.com/en-us/library/mt238290.aspx> [Dostopano 24.01.2016]

- [11] (2016) HTML [Online]. Dosegljivo:
http://www.w3schools.com/html/html_intro.asp [Dostopano 24.01.2016]
- [12] Sanders Kleifeld, "HTML5 for publisher", 2011
- [13] (2016) CSS [Online]. Dosegljivo:
<https://www.w3.org/Style/CSS/> [Dostopano 24.01.2016]
- [14] Peter Shaw, "CSS3 Succinctly", 2015
- [15] (2016) Ajax Control Toolkit [Online]. Dosegljivo:
<https://ajaxcontroltoolkit.codeplex.com/> [Dostopano 24.01.2016]
- [16] (2016) Visual Studio [Online]. Dosegljivo:
<https://www.visualstudio.com/> [Dostopano 24.01.2016]
- [17] (2016) SQL Management Studio. Dosegljivo:
[https://technet.microsoft.com/en-us/library/ms174173\(v=sql.105\).aspx](https://technet.microsoft.com/en-us/library/ms174173(v=sql.105).aspx) [Dostopano 24.01.2016]
- [18] Andrew Troelsen. "Pro C# with .NET 3.0" , 2008
- [19] (2016) C# Programming language [Online]. Dosegljivo:
<https://msdn.microsoft.com/en-us/library/z1zx9t92.aspx> [Dostopano 31.01.2016]
- [20] (2016) AngularJS. Dosegljivo:
<https://angularjs.org/> [Dostopano 31.01.2016]
- [21] (2016) NodeJS. Dosegljivo:
<https://nodejs.org/en/> [Dostopano 31.01.2016]
- [22] (2016) Oracle. Dosegljivo:
<http://www.oracle.com/index.html> [Dostopano 31.01.2016]
- [23] (2016) DB2. Dosegljivo:
<http://www-01.ibm.com/software/data/db2/> [Dostopano 31.01.2016]
- [24] (2016) Transact-SQL. Dosegljivo:
[https://technet.microsoft.com/en-us/library/ms189826\(v=sql.90\).aspx](https://technet.microsoft.com/en-us/library/ms189826(v=sql.90).aspx) [Dostopano 31.01.2016]

[25] (2016) Lambda. Dosegljivo:

<https://msdn.microsoft.com/en-us/library/bb397687.aspx> [Dostopano 31.01.2016]

