

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Mitja Petan

**Pospeševanje prodaje preko spletne trgovine**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE  
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana 2013



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*





Št. naloge: 00551 / 2013  
Datum: 15.9.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **MITJA PETAN**

Naslov: **POSPEŠEVANJE PRODAJE PREKO SPLETNE TRGOVINE  
SALES PROMOTION THROUGH THE ONLINE STORE**

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija prve stopnje

Tematika naloge:

Kandidat naj preuči sodobne spletne tehnologije za razvoj spletnih trgovin. Najprej izberite primerno programsko okolje, zgradite strukturo podatkovne baze in implementirajte delujoč model spletne trgovine. Model naj poskuša povečati obseg prodaje s pomočjo popustov, ki so na voljo šele, ko kupec sam uspe privabiti še druge kupce. Pri tem uporabite funkcionalnosti obstoječih družabnih omrežij. Na ta način si kupec lahko pridobi količinski popust, čeprav kupi samo en izdelek.

Mentor:

doc. dr. Luka Šajn



Dekan:

prof. dr. Nikolaj Zimic



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Mitja Petan, z vpisno številko **63010262**, sem avtor diplomskega dela z naslovom:

*Pospeševanje prodaje preko spletne trgovine*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Luka Šajna,
- 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 v zbirki "Dela FRI".

V Ljubljani, dne 15. novembra 2013

Podpis avtorja:



*Zahvalil bi se rad svojemu mentorju, doc. dr. Luki Šajnu, za vodenje in pomoč pri izdelavi diplomske naloge. Zelo sem hvaležen tudi moji družini in vsem ostalim, ki ste me podpirali in spodbujali v času študija.*



# Kazalo

Slike	iii
Seznam kratic	v
Povzetek	vii
Abstract	ix
<b>1 Uvod</b>	<b>1</b>
<b>2 Metode</b>	<b>3</b>
<b>3 Tehnologije in orodja</b>	<b>5</b>
3.1 ASP.NET . . . . .	6
3.2 ADO.NET . . . . .	7
3.3 javascript . . . . .	9
3.4 SQL Server . . . . .	9
<b>4 Rezultati</b>	<b>11</b>
4.1 Uporaba spletne trgovine . . . . .	13
4.2 Kaj potrebujemo za delovanje spletne trgovine . . . . .	15
4.3 Podatkovna baza . . . . .	18
4.4 Velikost spletne trgovine . . . . .	20
4.5 Kje deluje . . . . .	21

<b>5</b>	<b>Diskusija</b>	<b>23</b>
5.1	Varnost nakupov prek spletne trgovine . . . . .	23
5.2	Hitrost spletnih trgovin . . . . .	24
5.3	Uporaba metode popust na naročila pri obstoječih trgovcih . .	25
<b>6</b>	<b>Zaključek</b>	<b>27</b>
	<b>Literatura</b>	<b>29</b>
	<b>Dodatek</b>	<b>31</b>

# Slike

4.1	Izgled spletne trgovine. . . . .	12
4.2	Obrazec za vpis osebnih podatkov. . . . .	14
4.3	Diagram zaporedja oddaje naročila. . . . .	16
4.4	Shema podatkovne baze. . . . .	19
6.1	Struktura datotek. . . . .	32



# Seznam kratic

<b>DNS</b>	Domain Name System
<b>IIS</b>	Internet Information Server
<b>IP</b>	Internet Protocol
<b>ASP</b>	Active Server Pages
<b>.NET</b>	The .NET Framework is a software framework developed by Microsoft that runs primarily on Microsoft Windows
<b>MVC</b>	Model view controller
<b>C#</b>	Programming language
<b>ADO</b>	ActiveX Data Objects
<b>HTML</b>	Hypertext Markup Language
<b>CSS</b>	Cascading Style Sheets
<b>JavaScript</b>	Programming language
<b>AJAX</b>	Asynchronous JavaScript and XML
<b>XML</b>	Extensible Markup Language



# Povzetek

V času, ki ga živimo ta trenutek v katerem je gospodarska kriza neizprosna, mora vsak trgovec najbolje oglaševati svoje izdelke in s tem pritegnil čimvečje število kupcev. Zato se mnogi trgovci odločijo, da na svoje izdelke dodajo posebne popuste, darila ali kakšne druge ugodnosti, da bi tako pritegnili morebitnega kupca v nakup svojega izdelka. Na spletu je komunikacija med trgovcem in kupcem še posebej omejena, zato mora biti trgovec še posebej inovativen, da prepriča morebitnega kupca v nakup. V tem diplomskem delu sem razvil spletno trgovino, ki uporabniku ponudi poseben popust, ko število naročil preseže določeno število. Stem spodbudimo kupcu zanimanje za nakup, ker bo izdelek dobil po ugodnejši ceni. Trgovec pa dobi zadostno število naročil, da se mu po stroškovni plati izplača, ponuditi izdelek po nižji ceni.

## **Ključne besede:**

spletna trgovina, popust na količino naročil



# Abstract

Nowadays when the economic crisis is relentless, each trader have to do their best in advertising products. To attract potential costumers, traders choose to add special discounts, gifts or other benefits on products. Since the online communication between traders and costumers is one way communication only, traders have to be innovative in order to stimulate potencial customer to purchase.

We developed an online store that offers special discount when customer exceeds a certain number of orders. In this way dealer gets a sufficinet number of orders and can offer product with a lower price.

## **Keywords:**

internet store, discount to order quantity



# Poglavje 1

## Uvod

Spletne trgovine so realnost. Vse večje število ljudi uporablja spletne trgovine za svoje nakupe, saj lahko danes preko spletne trgovine, kupimo že skoraj vse kar je na voljo tudi v fizičnih trgovinah. Tu lahko kupimo živila v Mercatorjevi spletni trgovini<sup>1</sup>, zdravila brez recepta v lekarni Nove Poljane<sup>2</sup>, oblačila v spletni trgovini Stilago<sup>3</sup>, ležišča, odeje, vzglavnike v spletni trgovini Dormeo<sup>4</sup> orodja in pripomočke za gradnjo v Merkurjevi spletni trgovini<sup>5</sup>, priljubljeni izdelki iz tv oglasov v spletni trgovini Topshop<sup>6</sup>, zabavno elektroniko v trgovini Conrad<sup>7</sup>. Ne moremo pa mimo ne da bi naštel dveh največjih spletnih centrov Mimovrste<sup>8</sup> in Enaa<sup>9</sup>. Pravzaprav so že tako razširjene, da obstaja že več seznamov spletnih trgovin<sup>10</sup>. Poteka pa tudi vsakoletni izbor, kjer med vsemi prijavljenimi spletnimi trgovinami, izberejo najboljšo v tekočem letu<sup>11</sup>.

---

<sup>1</sup>Dostopno na spletnem naslovu: <http://trgovina.mercator.si/market/>

<sup>2</sup>Dostopno na spletnem naslovu: <http://www.lekarnar.com/>

<sup>3</sup>Dostopno na spletnem naslovu: <http://www.stilago.si/>

<sup>4</sup>Dostopno na spletnem naslovu: <http://www.dormeo.net/>

<sup>5</sup>Dostopno na spletnem naslovu: <http://www.merkur.si/>

<sup>6</sup>Dostopno na spletnem naslovu: <http://www.topshop.si/>

<sup>7</sup>Dostopno na spletnem naslovu: <http://www.conrad.si/>

<sup>8</sup>Dostopno na spletnem naslovu: <http://www.mimovrste.com/>

<sup>9</sup>Dostopno na spletnem naslovu: <http://www.ena.com/>

<sup>10</sup>Dostopno na spletnem naslovu: <http://www.trgovine.net/>

<sup>11</sup>Dostopno na spletnem naslovu: <http://www.spletnitrgovecleta.si/>

Seveda si vsi trgovci želijo čim večje prodaje svojih izdelkov. To pomeni, da si posledično želijo tudi večje število obiskov svoje spletne trgovine. Za doseg tega se trgovci poslužujejo različnih metod in akcij, da bi povečali obiskanost svojih spletnih strani oglašujejo na vseh razpoložljivih medijih od radija in televizije do časopisov, revij in plakatov. Za spletne trgovine je morda še najbolj pomembno spletno oglaševanje. Ta način povečevanja prometa je najbolj dostopen, saj s plačilom za oglas, ki je na drugi spletni strani, lahko na trgovčevo stran pripelje veliko obiskovalcev. Drugi težji način je optimizacija spletne trgovina za spletne iskalnike, kot je na primer Najdi<sup>12</sup> ali Google<sup>13</sup>. Trgovec si lahko povečuje ogled strani tudi tako, da na elektronske naslove, ki jih je zbral z zbiranjem prijav na elektronske novice pošilja elektronska sporočila, ki oglašujejo spletno trgovino in hkrati ponujajo kupcu kvalitetno vsebino, da se vedno znova vrača na njo. Da bi trgovec prepričal kupca v končen nakup, se le-ta poslužuje trženjskih metod prodaje. V spletnih trgovinah ponujajo razne popuste, darila, brezplačno dostavo izdelkov in podobno.

V tej diplomski nalogi se bomo osredotočili na trženjsko metodo prepričevanja nakupa, ki kupcu omogoča popust na izbrani izdelek ob zadostnem številu naročil. To pomeni, da bo kupec dobil popust in naročene izdelke šele, ko bo trgovec zbral zadostno število naročil. To število trgovec nastavi za vsak izdelek posebej. Tako bo na koncu kupec zadovoljen, ker bo dobil izdelek po znižani ceni, zadovoljen bo tudi trgovec, saj bo dosegel željeno ekonomijo obsega.

Glavna naloga oziroma cilj tega diplomskega dela je razviti spletno trgovino, ki bo omogočala nakupe po trženjski metodi popust na količino naročil.

---

<sup>12</sup>Dostopno na spletnem naslovu: <http://www.najdi.si/>

<sup>13</sup>Dostopno na spletnem naslovu: <http://www.google.si/>

# Poglavje 2

## Metode

Za začetek si najprej na kratko pogledimo obstoječe metode trženja v spletnih trgovinah. Najbolj pogosta trženjska metoda prepričevanja kupca je popust na izdelek. Tako bomo velikokrat zasledili v spletnih trgovinah ob predstavitvi izdelka, da je izdelek znižan za določen popust, na primer za 10%. Nakup preko spletne trgovine prinaša tudi dodaten strošek, ki ga načeloma kupci v fizičnih trgovinah ne zaznajo. Ta dodaten strošek je poštnina, ki se prišteje ceni zato, da lahko trgovec dostavi naročeni izdelek na kupcu na dom. Zaradi tega stroška v spletnih trgovinah se trgovci poslužujejo trženjske metode prepričevanja kupca in ponujajo kupcu brezplačno dostavo izdelkov na dom, lahko tudi po določenih drugih pogojih, kot je na primer nakup nad določenim zneskom, na določen dan itd. Darilo, ki je pogojeno z nakupom določenega izdelka, prav tako lahko premami kupca v nakup. Občutek, da bo kupec dobil nekaj več za svoj denar premami marsikaterega kupca.

Darilo lahko trgovec ponudi včasih tudi na vsak nakup ob različnih pogojih, kot je na primer nakup nad določenim zneskom ali nakup v določenem časovnem obdobju na primer med božičnimi prazniki.

Metoda 3 za 2 in podobne kombinacije in variacije, kjer trgovec kupcu

ponudi 3 izdelke za ceno dveh in mu najcenejšega podari. Ta metoda je zelo prepričljiva metoda in jo trgovec lahko ponudi kupcem v spletnih trgovinah.

Trgovec lahko uvede tudi kupone, ki so običajno nekajmestni in so sestavljeni iz črk in števil in jih oglašuje v medijih in z njimi ponuja različne ugodnosti. To so na primer popusti v obliki vrednosti, ali popust v odstotkovni obliki na primer 10% popusta na celoten nakup, darilo za nakup, brezplačna dostava, itd. je tudi lahko prepričljiva trženjska metoda prepričevanja kupca v nakup in lahko prepriča marsikaterega kupca.

Spletna trgovina 1nadan<sup>1</sup> omogoča nakupe kuponov, ki jih objavljajo dnevno s popusti od 50 do 90%. Da njihova ponudba uspe, se mora za nakup odločiti določeno število kupcev. Ko ponudba uspe in se zbere določeno število kupcev, se nakupi aktivirajo in razpošljejo kuponi, ki jih kupci izkoristijo pri ponudnikih. Če naročil ni dovolj se nakupi ne aktivirajo in kupcem se naročene kupone ne zaračuna. Podobno kot na spletni strani 1nadan deluje tudi spletna stran kolektiva<sup>2</sup>, kjer prav tako kupci kupujejo kupone in v primeru, da je ponudba uspela in je zadostno število naročil, kupci dobijo kupon po promocijski ceni. V primeru, da ponudba ne uspe kupcem denar vrnejo na transakcijski račun.

---

<sup>1</sup>Dostopno na spletnem naslovu: <http://1nadan.si/pogosta-vprasanja>

<sup>2</sup>Dostopno na spletnem naslovu: <http://www.kolektiva.si/ljubljana/faq/>

## Poglavje 3

# Tehnologije in orodja

Spletna trgovina, ki smo ji dali glavni pomen, popustu na število naročil, je bila razvita v razvojnem okolju Microsoft Visual Studio Ulitimate 2012<sup>1</sup> na osebem računalniku s konfiguracijo razvidno iz tabele 3.1. V razvojnem okolju Micorsoft Visual Studio Ultimate 2012 smo ustvarili projekt ASP.NET<sup>2</sup> MVC4 Web Application<sup>3</sup> v okolju .NET Framework 4.0<sup>4</sup>, ki kot osnovni programski jezik uporablja programski jezik C#. Pri razvoju vizaulnega dela aplikacije, ki se prikaže uporabniku v spletnem brskalniku smo v okolju ASP.NET uporabili osnovne spletne tehnologije HTML, CSS in javascript s knjižnico jQuery<sup>5</sup>. Sistem za upravljanje podatkovnih baz Microsoft SQL Server 2005 Express<sup>6</sup>, smo uporabili za upravljanje naših podatkov. Za administracijo in upravljanje podatkovnih baz smo, za urejanje in spreminjanje podatkov, ter za ustvarjanje podatkovnih shem in relacij, in pisanje stored procedur uporabili orodje Microsoft SQL Server Management Studio 2008. Za dostop do spletne trgovine smo med razvojem le-te, uporabljali spletni brskalnik Mozilla Firefox<sup>7</sup>.

---

<sup>1</sup>Dostopno na spletnem naslovu: <http://www.visualstudio.com/>

<sup>2</sup>Dostopno na spletnem naslovu: <http://www.asp.net/>

<sup>3</sup>Dostopno na spletnem naslovu: <http://www.asp.net/mvc/mvc4>

<sup>4</sup>Dostopno na spletnem naslovu: <http://www.microsoft.com/en-us/download/details.aspx?id=17851>

<sup>5</sup>Dostopno na spletnem naslovu: <http://jquery.com/>

<sup>6</sup>Dostopno na spletnem naslovu: <http://www.microsoft.com/en-us/download/details.aspx?id=21844>

<sup>7</sup>Dostopno na spletnem naslovu: <http://www.mozilla.si/>

Komponenta	Konfiguracija
Procesor	Intel(R) Core(TM)2 Quad CPU Q6600 @ 2.40GHz
Pomnilnik	4,00 GB RAM DDR2 (400 MHz)
Trdi disk	Western Digital (1.5 TB 7200 rpm)
Operacijski sistem	Windows 7 Ultimated SP1 64-bit

Tabela 3.1:

### 3.1 ASP.NET

ASP.NET je ogrodje, ki je namenjeno za gradnjo spletnih aplikacij in se zanj uporablja objektno orientiran pristop k izvajanju spletnih strani. Znotraj ASP.NET ogrodja smo uporabili MVC (Model View Controller) pristop, ki nam je v osnovi omogočil ločevanje: podatkovnih struktur, kontrolne logike ter prikaza ali izrisa HTML strukture. Za prenos podatkov iz kontrolne logike v izris HTML-ja smo uporabili podatkovno strukturo ViewBag, ki jo uporabljata obe strani.

Poglejmo si vzorčni primer, kako prenesti podatke iz kontrolne logike k pogledu.

```
public class HomeController : Controller
{
    public ActionResult Index()
    {
        ViewBag.NumberOfOrdersCreated = 2;
        ViewBag.NumberOfOrdersPercent = 10;
        return View();
    }
}
```

Koda 3.1: Programska koda na strani kontrolne logike.

```
<p>PRODANIH: <span class="focuscontent">
    @ViewBag.NumberOfOrdersCreated
</span>
</p>

<div class="offer-progress-bar-container">
    <div class="offer-progress-bar"
        style="width:@ViewBag.NumberOfOrdersPercent;">
    </div>
</div>

<p>POTREBUJEMO: <span class="focuscontent">
    @ViewBag.NumberOfOrdersRequired
</span>
</p>
```

Koda 3.2: Programska koda na strani izrisa HTML-ja.

## 3.2 ADO.NET

ADO.NET je tehnologija za pridobivanje podatkov iz podatkovnih baz. Zato smo tudi tehnologijo ADO.NET učinkovito izkoristili za preslikavo podatkov iz podatkovne baze v lokalne podatkovne modele v ogrodju ASP.NET. Kot primer si oglejmo, kako pridobiti podatke o izdelku s pomočjo ADO.NET tehnologije iz podatkovne baze in jih preslikati v podatkovni model izdelek.

```
public ProductModel GetProduct() {
    ProductModel product = new ProductModel();
    var connectionString = GetConnectionString();

    string sql = "EXEC_[]dbo].[Products_GetProduct]";

    using (SqlConnection connection =
new SqlConnection(connectionString)){
        SqlCommand command = new SqlCommand(sql, connection);

        connection.Open();
        SqlDataReader reader = command.ExecuteReader();

        if (reader.HasRows){
            while (reader.Read()){
                product.Product_ID = (int)reader["Product_ID"];
                product.VatRate = (double)reader["VatRate"];
                product.Name = (string)reader["Name"];
                product.Description = (string)reader["Description"];
                product.ShortDescription =
                    (string)reader["ShortDescription"];
                product.Price = (double)reader["Price"];
                product.Image = (string)reader["Image"];
                product.SortOrder = (int)reader["SortOrder"];
            }
        }
        reader.Close();
        command.Dispose();
        connection.Close();
    }
    return product;
}
```

Koda 3.3: Programska koda za preslikavo podatkov iz podatkovne baze v lokalni podatkovni model.

## 3.3 javascript

Vsi sodobni brskalniki podpirajo izvajanje javascript programske kode, ki se izvaja na strani uporabnika. Običajno se ta tehnologija uporablja za razne animacije, validacije, spremembe v HTML strukturi, izvajanje AJAX klicev. Tudi mi smo to tehnologijo izkoristili. V spodnjem vzorčnem primeru, si lahko ogledamo, kako smo s pomočjo javascript kode in z uporabo JQuery knjižnice, ki je napisana v javascript programskem jeziku napisali akcijo klika na sidrno povezavo, ki z animiranim pomikom do sidra, pripelje stran, na zaslonu, do željene pozicije.

```
var $document = $(document);
$document.ready(function () {
    $(".JS-GoToAnchor").click(function () {
        var target = $("#" + $(this).data("anchor"));
        $("html, body").animate({
            scrollTop: target.offset().top
        }, 1000);
    });
});
```

Koda 3.4: Programska koda premakne vidno polje strani na sidrišče.

## 3.4 SQL Server

Običajno vse dinamične spletne strani uporabljajo v ozadju sistem za upravljanje podatkovnih baz. Tako smo se mi odločili, da uporabimo sistem za upravljanje s podatkovnimi bazami, Microsoft SQL Server 2005 Express. S pomočjo Microsoft SQL Server Management Studia smo shranjevali stored procedure na omenjeni podatkovni strežnik. Želeli smo imeti shranjena vsa SQL povpraševanja na enem mestu, kar nam sprida omogoča podatkovni strežnik, saj omogoča shranjevanje stored procedur v katere vpišemo SQL povpraševanja.



# Poglavje 4

## Rezultati

Da bi razvili sistem za popust za količino naročil je bilo potrebno izdelati spletno trgovino, ki bo podpirala popust za količino naročil. Tako smo izdelal spletno trgovino, ki se sestoji iz glave, telesa in noge. V glavi tako lahko najdemo logotip trgovine, ki je na levi strani, telefonsko številko trgovine za hitro pomoč kupcem na sredini in manjši meni, kjer so na voljo odstrani o podjetju in kontaktni podatki. Na vrhu telesa spletne trgovine najdemo ime izdelka, kratek opis izdelka, ki mu sledi slika produkta pod njo pa je sklop informacij o cenah in številu naročil. V sklopu informacij o cenah na levi strani pokažemo ceno izdelka, popust v odstotkih ter prihranek v evrih. Na sredini prikažemo aktualno ceno izdelka ter pod njo velikost davka na dodano vrednost v evrih. Ostane nam še desna stran, kjer imamo število prodanih izdelkov, in kot pogoj število potrebnih nakupov, da ponudba uspe, ter grafični prikaz. Pod tem se nahaja facebook socialni vtičnik[10]. Pod sklopom informacijo o cenah in številu naročil se nahaja opis izdelka, ki zaključuje sklop informacijo o izdelku, ki se prodaja. Pod opisom se nahaja obrazec za nakup, kjer kupec vnese svoje osebne podatke spol, ime, priimek, naslov, pošta, telefon, e-naslov ter polje za sporočilo, kamor lahko kupec vnese svoje posebne želje, če jih ima. Pod obrazcem se nahaja še noga kjer so podatki o avtorju spletne trgovine. Izgled spletne trgovine si lahko ogledate na sliki 4.1.

## Obisk Celjskega gradu

Izkoristite to izjemno priložnost in si zagotovite vstop v srednjeveški slovenski biser po izjemni ceni. Pogledite si njegove arheološke ostanke in izjemen pogled na mesto Celje.



VREDNOS: 195,50 €

POPUST: **61,38%**

PRIHRANEK: **75,50 €**

VAŠA CENA: **120,00 €**

OD TEGA DDV: 21,64 €

PRODANIH: **6**

POTREBUJEMO: **20**

ODDAJ NAROČILO

Prilporeči Deli z ostalimi 8

## Opis Izdelka

### Zgodovina

Najstarejša omemba Celjskega gradu izhaja iz leta 1322 »purch Cylie«, kasneje pa so ga omenjali pod imeni »vest Cili« (1341), »castrum Cilie« (1451), »gloss Obercili« (1468) ipd. Značilno je, da se je ime »Obercili« - Gornje Celje pojavilo šele po izumrtju roda grofov Celjskih. Prvotno se je imenoval grad Celje.

Prvotno grajsko poslopje (romanski kastelni grad) so v prvi polovici 13. stoletja postavili grofje Vovbrški s Koroške na zahodni skalni kopi grajskega grebena. Imelo je značaj nepravilnega obodnega gradu. V talni ploskvi je imel obliko petkotnika, oziroma štirikotnika z zalomljeno južno stranico, ki se je prilagodila terenu. Prvi zapisi o gradu segajo med letoma 1125 in 1137; verjetno ga je zgradil mejni grof Gunter. V zahodnem delu je stala večnadstropna stavba, palačij, od katere so se v delih ohranili obodni zidovi. V vzhodnem delu je bilo zaprt dvorišče z velikim vodnim zbiralnikom. Na vzhodni, najbolj izpostavljeni strani, je bil zidni štiti plašč nekako 3 m višji od preostalega obodnega zidovja. Na vrhu je bil zavarovan s hurdo, obrambnim hodnikom. V takšni obliki je predstavljal tipični ministerialski grad tedanjega časa.

### Grad v lasti Žovneških in Celjskih

Prvotni grad je verjetno zgorel in propadel v bojih med Žovneškimi gospodi in Aufensteinovci. Vhod v grad so Žovneški svobodniki uredili s severne strani. Gradu so še pred letom 1300 dodali novi venec obzidja in ga na severni strani okrepili z obrambnim stolpičem, ki je varoval dohod v notranjost grajskega jedra. Obzidje je segalo od naravne zajede na vzhodu do prepadnih sten na severovzhodu. Vhod so prestavili na južno stran, kjer je še danes.

Po izumrtju njihove rodbine leta 1334 je grad prešel v roke Žovneških gospodov, od leta 1341 grofov Celjskih, in od leta 1436 Celjskih knezov. Grofje Celjski so začeli utrdbo preurejati v udobnejše stanovanjsko bivališče in uradno prebivališče. Okoli leta 1400 so dodali štirinadstropni obrambni stolp in ga kasneje imenovali Friderikov stolp (bergfrid). Na vzhodni strani dvorišča je bil velik trinadstropni stanovanjski stolp, ki je najbolj ohranjeni del gradu za Friderikovim stolpom. Na zahodnem delu je bila prava stanovanjska stavba (palas). Tu so bili prostori za ženske (kamenate). Ta del se je končal ob ozkem predgradu in je slabo ohranjen. Na južni strani je bil na palas prizidan Andrejev stolp. V nadstropju je imel kapelico svetega Andreja. S srednjeveškim orožjem gradu ni bilo moč zavzeti, le sestradali so ga lahko, čeprav je iz gradu vodil skriven prehod k žitnici. Celjski grofje so v tem času že prenehali bivati na gradu in so imeli na njem gradnika (kastelana), poveljnika gradu z oboroženim spremstvom.

Celjski grad, spredaj Savinja v Pečovniku proti severovzhodu in ob njenem levem bregu železniška proga proti Ljubljani.

Celjski grad, 23 m visoki Friderikov stolp (bergfrid), zgrajen na umetni vzpetini, pogled proti vzhodu

Ob potresu leta 1348 se je porušil del romanskega palačija s pečino vred. Porušeni del so na novo pozidali in ga prestavili proti notranjosti dvorišča. V 15. stoletju so razširili predgradje na vzhodni strani grebena vse do skalne kope. Tu se je obzidje povežalo z močnim peterokotnim obrambnim stolpom. V drugi polovici 16. stoletja je grad doživel ponovno prezidavo. Obrambno zidovje se je zvišalo v jedru kot tudi v predgradu. Uredili so tudi notranje dvorišče. Posodobljen del obzidja so opremili z renesančnimi strelnimi linami.

Slika 4.1: Izgled spletne trgovine.

## 4.1 Uporaba spletne trgovine

Uporaba spletne trgovine je močno povezana s potekom nakupnega procesa zato si tega pogledjmo od blizu.

Ob vstopu kupca v spletno trgovino, se temu prikaže vstopna stran, kjer so vidni podatki o izdelku, ki se prodaja. Podatki o izdelku so ime, kratek opis izdelka, slika, vrednost, višina popusta v odstotkih, prihranek, znižana cena oziroma kupčeva cena, količina davka na dodano vrednost v evrih, trenutno število prodanih izdelkov, število potrebnih izdelkov, priporočilni vmesnik socialnega omrežja facebook in opis izdelka.

Naslednji korak nakupnega procesa se zgodi, ko se kupec odloči, da želi izdelek kupiti, za nakup je primoran vnesti svoje osebne podatke. Ti so: spol, ime, priimek, naslov, pošta, telefon ter e-naslov. V kolikor ima kupec morda kakšne posebne želje lahko te zapiše v polje sporočilo. Ob končanem vnosu podatkov pritisne na gumb oddaj naročilo. Izgled spletnega obrazca naše spletne trgovine si lahko ogledate na sliki 4.2.

V naslednjem koraku se ti podatki preverijo in če ustrezajo zadanim pogojem, kot so: vsa izpolnjenja polja ter ali vpisani e-naslov ustreza obliki, ki je določen za e-naslov. V kolikor so podatki napačni se kupcu nepravilnosti označijo, da jih lahko popravi. Ob izpolnjenih vseh pogojih se ti podatki zabeležijo v bazo podatkov.

Po uspešnem zapisu podatkov v bazo podatkov sestavimo potrditveno sporočilo, kjer kupcu sporočimo, da smo uspešno prejeli njegovo naročilo. V njem so podatki o kupcu ter podatki o naročenem izdelku. Tako sporočilo potem pošljem na e-naslov kupca, ki ga je vnesel v obrazec.

**Vaši podatki**

Spol:

Ženska  Moški

Ime:

Priimek:

Naslov:

Pošta:

Telefon:

E-naslov:

Vaše sporočilo:

**ODDAJ NAROČILO**

© 2013 - Mitja Petan

Slika 4.2: Obrazec za vpis osebnih podatkov.

Po pošiljanju potrditvenega sporočila kupca preusmerimo na zahvalno stran, kjer kupcu tako, kot pri potrditvenem sporočilu prikažemo njegove podatke, ki jih je vnesel in podatke o naročenem izdelku. S preusmeritvijo na zahvalno stran zaključimo nakupni proces. Naročilo je tako uspešno oddano. Potek nakupnega procesa si lahko ogledate tudi v diagramu zaporedja, ki je viden na sliki 4.3.

Danes so socialna omrežja zelo priljubljena med uporabniki svetovnega spleta, kupec lahko, v želji po pridobitvi popusta, uporabi socialni facebook vtičnik, ki je dostopen na spletni strani, z njim lahko deli ponudbo na socialnem omrežju facebook, med svoje prijatelje in tako oglašuje izdelek. Tako oglaševanje ima poseben socialen pristop, saj lahko besede prijatelja, bolj prepričajo neodločenega kupca, kot pa besede trgovca.

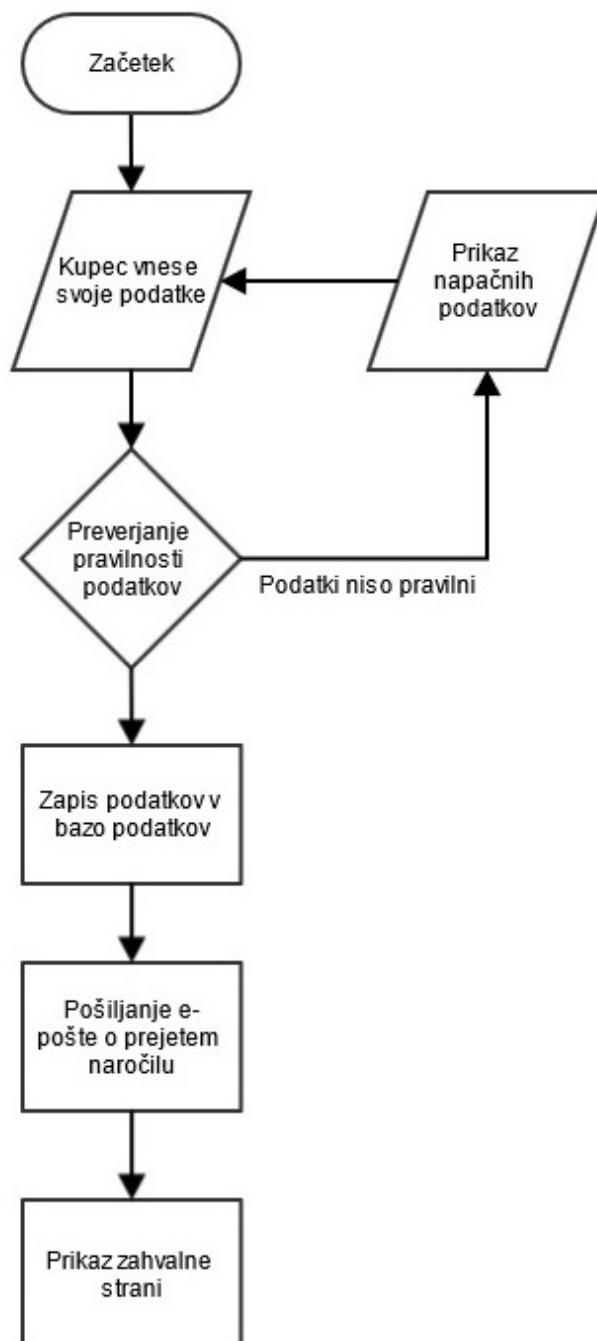
## 4.2 Kaj potrebujemo za delovanje spletne trgovine

Za delovanje spletne trgovine najprej potrebujemo spletni naslov oziroma domeno, ki jo kupimo in registriramo pri enem od registrarjev<sup>1</sup>. Z domeno predstavljamo spletno trgovino, jo oglašujemo in je običajno ime spletne trgovine. Poznamo tudi veliko vrsto vrhnjih domen. To je prvi podatek uporabniku. Tako v Sloveniji podjetja uporabljajo vrhnjo domeno .si in s tem sporočajo uporabnikom, da gre za slovensko spletno stran. Ko registriramo domeno, pridobimo spletni naslov, na primer <http://www.popustnakolicino.si/>, ki bo uporabnikom prikazoval našo spletno trgovino. Nadalje za delovanje potrebujemo DNS strežnik, običajno lahko DNS strežnik zakupimo pri registrarju. Na voljo so nam tudi DNS strežniki, ki nam za svoje storitve na zaračunajo niti centa. Eden takih je FreeDNS<sup>2</sup>. Na DNS strežniku nastavimo, pot od

---

<sup>1</sup>Dostopno na spletnem naslovu: <http://www.register.si/seznam-registrarjev.html/>

<sup>2</sup>Dostopno na spletnem naslovu: <http://freedns.afraid.org/>



Slika 4.3: Diagram zaporedja oddaje naročila.

spletnega naslova do našega spletnega strežnika, kjer hranimo kodo spletne trgovine. Za delovanje je poleg kode, ki predstavlja spletno trgovino, najpomembnejši spletni strežnik. Za delovanje potrebujemo računalnik, ki je povezan z internetnim omrežjem, na katerem je nameščen operacijski sistem. V našem primeru je to Microsoft Server 2003 Web Edition, na katerem je nameščen spletni strežnik IIS. V večini primerov se lahko izognemo lastnemu strežniku, z zakupom gostovanja na spletnem strežniku pri registratorjih, ki dodeli nekaj prostora na trdem disku, kjer shranimo našo kodo. Naš spletni strežnik mora prepoznati spletni naslov, katerega nam posreduje DNS strežnik, zato moramo zagotoviti, nastavitve spletnega strežnika na način, da ko dobi strežnik zahtevo po spletnem naslovu, izvede programsko kodo, ki je namenjena zahtevanemu spletnemu naslovu. To zagotovimo tako, da na samem IIS-u nastavimo spletni naslov, ter pot do datotek, kjer se nahaja koda spletne trgovine. Spletna trgovina je napisana v okolju Microsoft Visual Studio 2012, kot projekt ASP.NET MVC4 Web Application v okolju .NET Framework 4.0 in kot osnovni jezik na strežniški strani uporablja C#, zato moramo na spletnem strežniku imeti nameščen .NET Framework 4.0, ki zna izvesti kodo in njen rezultat posredovati do brskalnika.

Spletna trgovina pošilja elektronska sporočila kupcem, za delovanje pošiljanja elektronskih sporočil potrebujemo SMTP strežnik. V našem primeru izkoriščamo SMTP strežnik, ki je vključen v IIS 6.0.

Podatke o izdelkih in naročilih shranjujemo v bazo podatkov. Da bi spletna trgovina delovala, potrebujemo podatkovni strežnik. Uporabili smo Microsoft SQL Server 2005 Express, ki je nameščen na spletnem strežniku.

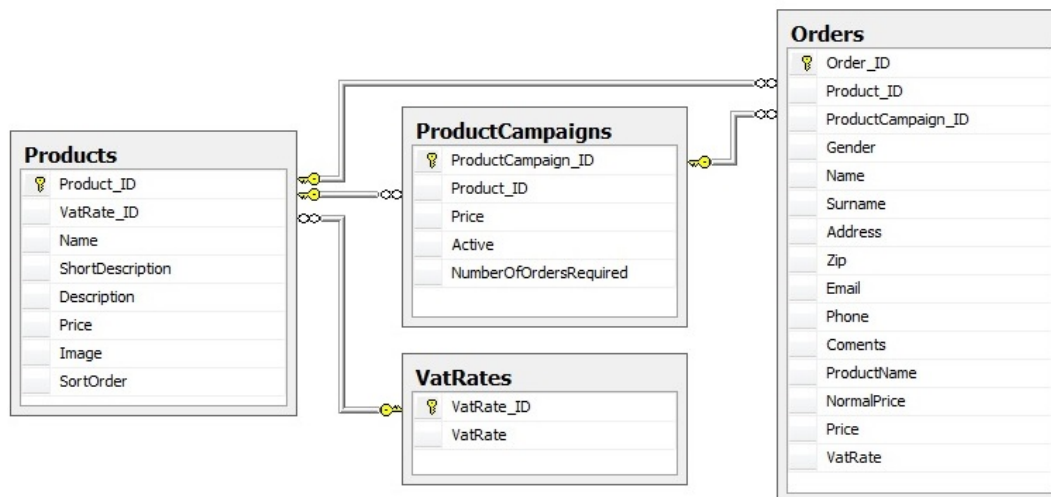
## 4.3 Podatkovna baza

V podatkovno bazo shranjujemo podatke. Za namen spletne trgovine v podatkovno bazo shranjujemo podatke o izdelkih in naročilih. Za predstavitev oglaševanega izdelka v spletni trgovini podatke o izdelku shranimo v podatkovno tabelo Izdelki. Podatkovna tabela izdelkov ima polje Product.ID, v katerem je shranjena unikatna vrednost produkta temu polju pravimo tudi primarni ključ. Tako imamo v tabeli izdelkov več izdelkov. Prikazali pa bomo točno določenega, saj se lahko sklicujemo na to unikatno vrednost in tako pridobimo samo podatke željenega izdelka. Poglejmo si primer SQL povpraševanja, kjer pridobimo podatke o izdelku glede na unikatno vrednost izdelka v polju Product\_ID, katerega vrednost v našem primeru je enaka 1, SQL povpraševanje vrne tudi podatek o višini davčne stopnje iz podatkovne table davčnih stopenj.

```
SELECT
    P.Product_ID ,
    VR.VatRate ,
    P.Name ,
    P.ShortDescription ,
    P.[Description],
    P.Price ,
    P.[Image],
    P.SortOrder
FROM dbo.Products AS P
    INNER JOIN dbo.VatRates AS VR
        ON P.VatRate_ID = VR.VatRate_ID
WHERE P.Product_ID = 1;
```

Koda 4.1: SQL povpraševanje po podatkih o izdelku.

Za prikaz izdelka v naši spletni trgovino potrebujemo podatke ime izdelka, kratek opis, opis, cena, slika in vrstni red po katerem določimo, kateri



Slika 4.4: Shema podatkovne baze.

izdelek je na vrsti za prikaz na naši spletni trgovini. V tabeli izdelkov imamo tudi tuji ključ s katerim lahko povežemo tabelo davčnih stopenj in tako pridobimo davčno stopnjo izdelka. V Sloveniji imamo dve davčni stopnji, in sicer 9.5% za hrano, knjige in nekatere storitve ter 22% davčno stopnjo za preostale stvari. Tako imamo v našem primeru v tabeli davčnih stopenj samo dve vrednosti. S časoma bomo imeli vse več izdelkov v naši podatkovni bazi. Vsak izdelek bo imel tako nastavljeno eno od vrednosti. V primeru, da se davčne stopnje spremenijo, spremenimo vrednosti v tabeli davčnih stopenj in tako na hiter in enostaven način spremenimo oziroma popravimo višino davčne stopnje vsem izdelkom. Znižano ceno pridobimo iz podatkovne tabele prodajnih akcij, ki je na shemi baze podatkov, slika 4.4, poimenovana `ProductCampaigns`. V tej tabeli se poleg že omenjene znižane oziroma akcijske cene nahaja še pogoj za znižano ceno to je število potrebnih naročil, da ponudba uspe. Na koncu je tu še polje `Aktivno`, kjer povemo ali je prodajna akcija aktivna ali ne. Tako smo s pomočjo podatkovnih tabel `Products`, `VatRates` in `ProductCampaigns` v podatkovni bazi podatkov pripravili vse

potrebno za shranjevanje podatkov o izdelkih. Na shemi baze podatkov se nahaja še ena tabela, to je tabela naročil. V tabeli naročil, kot prvo shranjujemo podatke o kupcih. Kupčevi podatki, se shranjujejo v podatkovno tabelo naročil tako, da kupec vpiše svoje osebne podatke v spletni obrazec, ki se nahaja v spletni trgovini. Da lahko odda naročilo mora v spletni obrazec vpisati svoje osebne podatke: spol, ime, priimek, naslov, poštna številka in ime pošte, e-naslov, telefonska številka in morebitne želje. Ti osebni podatki se nato shranijo v podatkovno tabelo naročil v isto imenska polja. V shemi baze podatkov, bomo tako našli polja: Gender, Name, Surname, Address, Zip, Email, Phone, Coments. V tabeli naročil bomo poleg podatkov o kupcih našli še podatke o izdelku, ki ga je kupec naročil. Podatki o izdelku, ki se shranjujejo v podatkovno tabelo naročil so: ime, cena, znižana cena, davčna stopnja, unikatna številka produkta ter unikatna številka prodajne akcije po kateri je kupljen izdelek. Gotovo se sprašujete zakaj shranjujemo ime izdelka, ceno, znižano ceno in davčno stopnjo v podatkovno tabelo naročil. Vse te podatke bi lahko pridobili iz tabel izdelkov, prodajnih akcij in davčnih stopenj. Čeprav lahko pridobimo te podatke iz omenjenih tabel pa ne moremo zagotoviti, da bodo te vrednosti vedno enake. Na primer: kupec je včeraj kupil izdelek po ceni 100€, nato pa se danes trgovec odloči, da ta izdelek ne bo več prodajal po tej ceni in mu tako spremni ceno iz 100€ na 110€, ko bi nato pregledovali naročila bi vsi izdelki označeni kot, da so bili kupljeni po ceno 110€, čeprav temu ni tako. V tem poglavju smo si v celoti pogledali zgradbo tabel in njenih relacij v podatkovni bazi, ki si jo lahko ogledate na sliki 4.4.

## 4.4 Velikost spletne trgovine

Naša spletna trgovina je v primerjavi z ostalimi tipičnimi trgovinami, zelo omejena mala spletna trgovina, saj omogoča prodajo samo enega izdelka v istem času. V spletni trgovini sicer lahko menjujemo izdelke v njej, vendar je kupcu omogočen nakup samo enega izdelka. Če bi želeli v spletni trgovini

prodajati več izdelkov hkrati bi morali nekoliko spremeniti našo spletno trgovino. Gotovo bi začeli z implementacijo košarice v spletni trgovini, tako da bi kupec lahko dodajal izdelke v njo. Za implementacijo košarice bi morali spremeniti strukturo podatkovne baze, tako da bi dodali novo podatkovno tabelo. V tabeli košarice bi se nahajali podatki: Session\_ID oziroma id seje, ki jo generira spletni strežnik za vsakega obiskovalca spletne trgovine posebej in je unikatna za vsakega obiskovalca. Poleg id-ja seje bi potrebovali vsaj še unikatno številko izdelka ter njegovo količino. V spletni trgovini bi morali implementirati tudi dodajenje v košarico. V naši mali trgovini kupec s pritiskom na gumb odda naročilo, ki se zapiše v tabelo naročil. To bi v primeru, če želimo podpreti nakup večje količine izdelkov hkrati, morali nekoliko spremeniti. Gumb naroči, bi spremenili tako, da bi ga zamenjali z dodaj v košarico, ki bi dodal izdelek v košarico in kupca pripeljal na spletno stran za pregled košarice. Tu bi se nahajal gumb za naročilo izdelkov, ki bi sprožil ukaz za zapis naročila v podatkovno bazo. Pri prodaji večjega števila izdelkov se hitro pokaže potreba po kategorizaciji teh izdelkov. Meni kategorij in kategorična stran z izdelki, ki spadajo na takšno stran je naslednja logična razširitev trgovine. Za tako razširitev bi potrebovali v bazi podatkov novo tabelo v katero bi shranjevali kategorije ter povezovalno tabelo izdelkov in kategorij s katero bi določil kateri izdelki spadajo v katero kategorijo. S temi modifikacijami bi že močno razširili našo spletno trgovino. Seveda, pa bi lahko našo spletno trgovino razširili še z mnogo drugimi stvarmi.

## 4.5 Kje deluje

Spletna aplikacija oziroma spletna trgovina je dostopna vsem elektronskim napravam, ki imajo dostop do spleta in imajo nameščen namenski spletni brskalnik za dostop do spletnih strani. Za prikaz spletne trgovine mora znati interpretirati HTML kodo, CSS stile to je znati interpretirati obliko HTML elementov ter javascript programski jezik, ki se izvaja v samem brskalniku pravimo tudi na strani uporabnika. Tako spletna trgovina, danes

Brskalniki	Verzije	Delovanje
Microsoft Internet Explorer	6 ali višje	✓
Mozilla Firefox	1 ali višje	✓
Google Chrome	1 ali višje	✓
Opera	6 ali višje	✓
Apple Safari	1 ali višje	✓

Tabela 4.1:

deluje v vseh popularnih spletnih brskalnikih, kot so Mozilla Firefox<sup>3</sup>, Google Chrome<sup>4</sup>, Microsoft Internet Explorer<sup>5</sup>, Opera<sup>6</sup>, Safari<sup>7</sup> podjetja Apple ter drugi. Spletna trgovina se bo prikazovala tudi na mobilnikih in pametnih telefonih ter tablicah, ki danes zelo hitro pridobivajo na popularnosti. Na pametne telefone in tablice lahko namestimo mobilne verzije brskalnikov, ki prav tako podpirajo standarde za prikaz spletnih strani to je HTML, CSS in javascript, kar uporabnikom omogoča prikaz spletnih strani na njihovih napravah. Čeprav naša spletna trgovina ni prilagojena tem naprava, bodo uporabniki lahko pregledovali vsebino naše spletne trgovine in opravili nakup.

<sup>3</sup>Dostopno na spletnem naslovu: <http://www.mozilla.org/sl/firefox/new/>

<sup>4</sup>Dostopno na spletnem naslovu: <http://www.google.com/intl/sl/chrome/>

<sup>5</sup>Dostopno na spletnem naslovu: <http://windows.microsoft.com/sl-si/internet-explorer>

<sup>6</sup>Dostopno na spletnem naslovu: <http://www.opera.com/>

<sup>7</sup>Dostopno na spletnem naslovu: <http://www.apple.com/safari/>

# Poglavje 5

## Diskusija

### 5.1 Varnost nakupov prek spletne trgovine

Pogosto se dogaja, da nepridipravi uporabnike spleta zavedejo in jim od-tujijo, ali jih zavedejo na način, da le-ti ostanejo brez denarja. Velikokrat za take in drugačne prevare izvemo iz medijev. To v nas vzbuja strah, ker si nihče ne želi ostati brez svojih prihrankov, zato je varnost kupcu zelo pomembna. Trgovci se tega dobro zavedajo in si želijo ohraniti oziroma pri-dobiti dobro ime v družbi kateri ponujajo svoje izdelke in storitve. Da bi to dosegli trgovci pogosto varnost oglašujejo z napisi, kot je: "100% varen nakup!" in podobnimi, kar lahko daje kupcu eno dodatno prepričanje v var-nost nakupov prek spleta. Da pa ne bi bilo vse tako črnogledo si poglejmo, kako iz varnostnega vidika izgleda nakup prek spleta. Ko morebitini kupec pride v spletno trgovino najprej opazi ime trgovca. Ime trgovca je kupcu prva informacija, kjer lahko ugotovi ali je trgovec resničen ali je samo nekdo, ki ga želi oguljufati. Običajno ljudje ne kupujejo pri trgovcih, ki jih ne po-znajo. Pogosto za spletno trgovino slišijo iz medijev, prijateljev znancev, več ko imajo informacij o trgovcu in opravljenimi nakupi svojih prijateljev bol temu trgovcu zaupajo. Ko kupec dovolj zaupa trgovcu in se odloči za nakup preko spletne trgovine, kupec lahko odda zgolj naročilo za nakup za metodo plačila pa izbere plačilo po povzetju, tako se lahko izogne plačevanju

prek spleta, saj bo kupec plačal izdelke dostavljalcu ob prevzemu in se s tem izognil tveganju za izgubo denarja in pomislekom, ki jih morda ima. Tudi plačila prek spleta bi lahko rekli, da so varna. Saj trgovec običajno za plačilo prek spleta kupca preusmeri na strani banke, na kateri opravi plačilo, ki pa je obvezno na varni https povezavi, kjer ima banka nameščen zaupanja vreden certifikat in se tako legitimno predstavi kupcu kdo pravzaprav je. Pri varnosti bi še rad omenil, da številke kreditne kartice ne vpisujemo v vsako polje, ki ga vidimo in je ne shranjujmo v bazah spletnih trgovcev čeprav to morda nekateri ponujajo.

## 5.2 Hitrost spletnih trgovin

Na hitro si še pogledajmo zakaj je pomembna hitrost nalaganja spletne trgovine. Spletna trgovina, ki se uporabniku nalaga v nedogled ne bo koristila nikomur. Morebitni kupec bi se naveličal čakati, da se stran naloži šele po nekaj sekundah in hitro obupal in odnehal od nakupa. Optimalno je da se uporabniku stran odpre v nekaj milisekundah, vendar je včasih to težko zagotoviti saj lahko število sočasnih povezav na spletno trgovino privde do tega, da se stran začne počasi prikazovati. Razlogov za počasno nalaganje je lahko tudi v sami programski kodi spletne trgovine, ali morda kaj tretjega. Pri vsem tem je pomembno, da raziščemo zakaj je odzivnost spletne strani počasna in ustrezno ukrepamo. Ukrepamo lahko tako da, popravimo programsko kodo, nastavimo predpomnilnik in tako ne obremenjujemo podatkovne baze po nepotrebnem, poskusimo optimizirati programsko kodo, poskusimo optimizirati SQL povraševanje če ugotovimo, da tu nastajajo težave. In kot skrajno možnost če ne gre drugače zamenjamo strojno opremo z boljšo.

---

## 5.3 Uporaba metode popust na naročila pri obstojećih trgovcih

Prodaje izdelkov po trženjski metodi popust na količino naročil v ostalih slovenskih spletnih trgovinah ne zasledimo. Še najbližje tej metodi je spletna prodajalna kuponov Kolektiva, ki zbira naročila, da lahko ponudijo kupcem kupone po ugodnih cenah za izdelke in storitve, ki jih potem lahko koristijo pri ponudnikih. Opažam, da trgovci, prepričujejo kupce v največji meri z popustom na izdelek.



## Poglavje 6

### Zaključek

Cilj tega diplomskega dela je bil razviti spletno trgovino, ki bo omogočala kupcu uveljaviti popust na količino. Na začetku dela, smo si ogledali najbolj pogoste metode trženja v spletnih trgovinah, nato smo vam predstavili tržensko metodo popust na količino. Kot osrednji del diplomskega dela pa smo vam predstavili uporabo in delovanje spletne trgovine. Osnovni cilj izdelati spletno trgovino, ki bo omogočala nakup na količino smo dosegli in tako pokazali, da je tak pristop mogoče razviti v spletni trgovini. Seveda prototip spletne trgovini ni popoln. Pri sami spletni trgovini je še veliko prostora za optimizacijo in razširitev. Tako bi lahko razvili avtomatiko kaj se zgodi z naročili, ko dosežemo ustrezno število naročil. Ustrezno bi izbrali vsa naročila in kupce obvestili o uspešni ponudbi. Naslednja možnost izboljšanja bi bila obveščanje kupcev po določenem času, da ponudba še ni uspela in pozvali kupce, da posredujejo ponudbo svojim prijateljem, ki jih to morda zanima. Osnovni cilj implementirati spletno trgovino, ki omogoča popust na količino naročil smo dosegli.



# Literatura

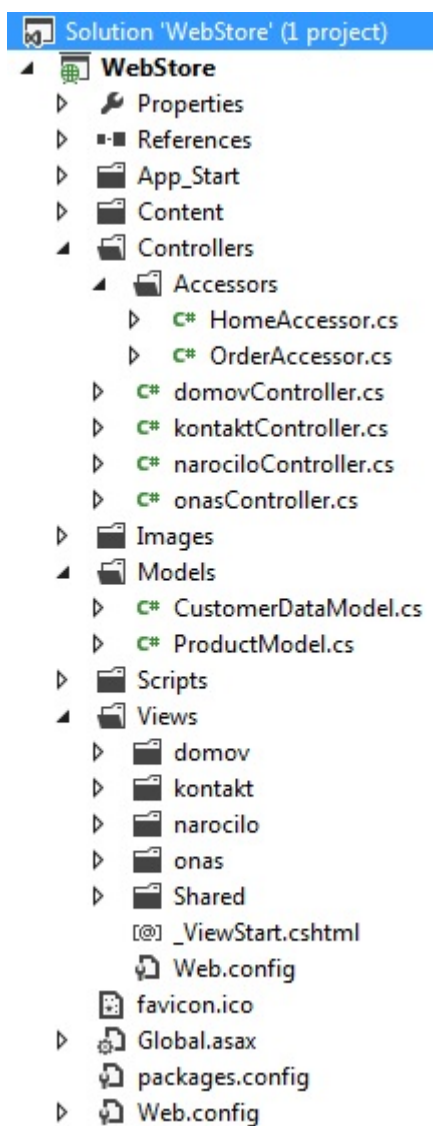
- [1] Adam Freeman. "Pro ASP.NET MVC 4", 2012.
- [2] Jon Galloway, Phil Haack, Brad Wilson, K. Scott Allen. "Professional ASP.NET MVC 4", 2012.
- [3] Bill Evjen, Scott Hanselman, Devin Rader. "Professional ASP.NET 4 in C# and VB", 2010.
- [4] Dino Esposito. "Programming Microsoft ASP.NET 3.5", 2008.
- [5] Christian Nagel, Bill Evjen, Jay Glynn, Karli Watson, Morgan Skinner. "Professional C# 2005 with .NET 3.0", 2007.
- [6] G. Andrew Duthie. "Microsoft ASP.NET programming with Microsoft visual C# .NET step by step", 2003.
- [7] Itzik Ben-Gan, Lubor Kollar, Dejan Sarka, Steve Kass. "Inside Microsoft SQL Server 2008: T-SQL Querying", 2008.
- [8] Itzik Ben-Gan, Lubor Kollar, Dejan Sarka. "Inside Microsoft SQL Server 2005: T-SQL Querying", 2005.
- [9] Itzik Ben-Gan, Dejan Sarka, Roger Wolter. "Inside Microsoft SQL Server 2005: T-SQL Programming", 2005.
- [10] Facebook, inc. "Documentation", Facebook Developers (online), (posodobljeno 20.8.2013, citirano 15.11.2013). Dostopna na naslovu: <https://developers.facebook.com/docs/plugins/share-button/>



# Dodatek

V dodatku podajamo nekaj vzorcev kode, vključno z izvorno kodo spletne trgovine popust na število naročil. Zaradi preglednosti smo določene elemente spletne trgovine nekoliko prilagodili.

Na začetku si na sliki 6.1 najprej pogledjmo datotečno strukturo našega projekta. Ker je smo za izdelavo aplikacije izbrali MVC projekt je strukturirana v tri glavne sklope Views, kjer se nahaja izris HTML kode, Models kjer so nahajajo podatkovni model izdelkov ter model osebnih podatkov kupca ter Controllers, kjer se nahaja programska logika naše aplikacije. Poleg naštetega je tu še sklop Content, kjer se nahaja CSS datoteka, ki pove kako se naj oblikovno prikaže naša HTML koda. Sklop Images, kjer se nahajajo slike ter na koncu še Scripts kamor sodijo javascript datoteke.



Slika 6.1: Struktura datotek v MVC4 projektu.

```
<div class="body-holder">
  <div class="body-productName">
    <h1>@ViewBag.ProductName</h1>
  </div>
  <div class="body-productShortDescription">
    <h2>@ViewBag.ProductShortDescription</h2>
  </div>
  <div class="body-offer">
    <div class="body-image">
      
    </div>
    <div class="body-prices">
      <table class="body-offer-prices">
        <tr>
          <td class="body-offer-prices-first">
            <p>VREDNOS: <span class="strikethrough">@ViewBag.
              ProductPrice</span></p>
            <p>POPUST: <span class="focuscontent">@ViewBag.
              SavingPercent</span></p>
            <p>PRIHRANEK: <span class="focuscontent">@ViewBag.
              Saving</span></p>
          </td>
          <td class="body-offer-prices-second">
            <div class="main-offer-focus">
              <p class="yourprice-text">VASA CENA:
                <span class="yourprice-value">@ViewBag.
                  ProductCampaignPrice</span>
              </p>
              <p class="taxtext">OD TEGA DDV:
                <span class="">@ViewBag.Tax</span>
              </p>
              <div class="buybutton-holder JS-GoToAnchor" data-anchor
                ="anchororderform">
                <p class="buybutton-text">ODDAJ NAROCILO</p>
              </div>
            </div>
          </td>
          <td class="body-offer-prices-last">
```

```

<p>PRODANIH: <span class="focuscontent">@ViewBag.
    NumberOfOrdersCreated</span></p>
<div class="offer-progress-bar-container">
  <div class="offer-progress-bar" style="width:@ViewBag.
    NumberOfOrdersPercent;">
  </div>
</div>
<p>POTREBUJEMO:
  <span class="focuscontent">@ViewBag.
    NumberOfOrdersRequired</span>
</p>
<div class="fb-like" data-href="@Request.Url" data-width
  ="450"
  data-layout="button_count" data-action="recommend"
  data-show-faces="true" data-send="true">
</div>
</td>
</tr>
</table>
</div>
</div>
<div class="body-description">
  <h2>Opis Izdelka</h2>
  <p>@Html.Raw(ViewBag.ProductDescription)</p>
</div>
<div class="body-form">
  @Html.Partial("OrderForm")
</div>
</div>

```

Koda 6.1: HTML struktura vstopne strani.

```

private void SetHomeData()
{
    HomeAccessor Accessor = new HomeAccessor();
    ProductModel Product = new ProductModel();
    CultureInfo Culture = new CultureInfo("sl-SI");
    Culture.NumberFormat.CurrencySymbol = "€";
}

```

```
Product = Accessor.GetProduct();
Product = Accessor.GetOrderCampaign(Product);

ViewBag.ProductName = Product.Name;
ViewBag.ProductShortDescription = Product.
    ShortDescription;
ViewBag.ProductDescription = Product.Description;
ViewBag.ProductImage = Product.Image;

ViewBag.ProductPrice = Product.Price.ToString("C2",
    Culture);
ViewBag.ProductCampaignPrice = Product.
    ProductCampaignPrice.ToString("C2", Culture);

ViewBag.Saving = (Product.Price - Product.
    ProductCampaignPrice).ToString("C2", Culture);
ViewBag.SavingPercent = (Product.ProductCampaignPrice
    / Product.Price).ToString("P", Culture);

ViewBag.Tax = (Product.ProductCampaignPrice - (
    Product.ProductCampaignPrice / (1 + (Product.
    VatRate / 100))))).ToString("C2", Culture);

ViewBag.NumberOfOrdersRequired = Product.
    NumberOfOrdersRequired;
ViewBag.NumberOfOrdersCreated = Product.
    NumberOfOrdersCreated;

decimal numberOfOrdersCreated = decimal.Divide(
    Product.NumberOfOrdersCreated, Product.
    NumberOfOrdersRequired);

if (numberOfOrdersCreated > 1)
{
    numberOfOrdersCreated = 1;
}

ViewBag.NumberOfOrdersPercent = numberOfOrdersCreated
```

```
        .ToString("P0", Culture);  
    }  
}
```

Koda 6.2: Zbirko ViewBag napolnimo v metodi SetHomeData(), ki se nahaja v mapi Controllers.

```
public ProductModel GetProduct()  
{  
    ProductModel product = new ProductModel();  
    var connectionString = GetConnectionString();  
  
    string sql = "EXEC_□[dbo].[Products_GetProduct]";  
  
    using (SqlConnection connection = new SqlConnection(  
        connectionString))  
    {  
        SqlCommand command = new SqlCommand(sql,  
            connection);  
  
        connection.Open();  
        SqlDataReader reader = command.ExecuteReader(  
            );  
  
        if (reader.HasRows)  
        {  
            while (reader.Read())  
            {  
                product.Product_ID = (int)  
                    reader["Product_ID"];  
                product.VatRate = (double)  
                    reader["VatRate"];  
                product.Name = (string)reader  
                    ["Name"];  
                product.Description = (string  
                    )reader["Description"];  
                product.ShortDescription = (  
                    string)reader["  
                        ShortDescription"];  
            }  
        }  
    }  
}
```

```
        product.Price = (double)
            reader["Price"];
        product.Image = (string)
            reader["Image"];
        product.SortOrder = (int)
            reader["SortOrder"];
    }
}
reader.Close();

command.Dispose();
connection.Close();
}

return product;
}
```

Koda 6.3: Model Product napolnimo iz podatkovne baze s pomočjo metode GetProduct(), ki se nahaja v HomeAccessor.cs datoteki.

```
SELECT TOP (1)
    P.Product_ID ,
    VR.VatRate ,
    P.Name ,
    P.ShortDescription ,
    P.[Description],
    P.Price ,
    P.[Image],
    P.SortOrder
FROM dbo.Products AS P
    INNER JOIN dbo.VatRates AS VR ON P.VatRate_ID = VR.
        VatRate_ID
ORDER BY
    P.SortOrder DESC ,
    P.Product_ID DESC;
```

Koda 6.4: SQL poizvedba za pridobivanje podatkov o produktu.

```
public ProductModel GetOrderCampaign(ProductModel product)
```

```
{  
    var connectionString = GetConnectionString();  
  
    string sql = "EXEC [dbo].[  
        ProductCampaigns_GetCampaignByProductID]   
        @Product_ID";  
  
    using (SqlConnection connection = new SqlConnection(  
        connectionString))  
    {  
        SqlCommand command = new SqlCommand(sql,   
            connection);  
  
        command.Parameters.Add("@Product_ID",   
            SqlDbType.Int);  
        command.Parameters["@Product_ID"].Value =   
            product.Product_ID;  
  
        connection.Open();  
        SqlDataReader reader = command.ExecuteReader   
            ();  
  
        if (reader.HasRows)  
        {  
            while (reader.Read())  
            {  
                product.ProductCampaignPrice   
                    = (double)reader["Price"];  
                product.  
                    NumberOfOrdersRequired = (  
                        int)reader["  
                            NumberOfOrdersRequired"];  
                product.NumberOfOrdersCreated   
                    = (int)reader["  
                            NumberOfOrdersCreated"];  
            }  
        }  
        reader.Close();  
    }  
}
```

```
        command.Dispose();
        connection.Close();
    }
    return product;
}
```

Koda 6.5: Pridobivanje podatkov o številu naročil in o številu potrebnih naročil.

```
SELECT TOP (1)
    PC.ProductCampaign_ID,
    PC.Price,
    PC.NumberOfOrdersRequired,
    COUNT(O.Order_ID) AS NumberOfOrdersCreated
FROM dbo.ProductCampaigns AS PC
    LEFT JOIN dbo.Orders AS O ON O.
        ProductCampaign_ID = PC.ProductCampaign_ID
WHERE PC.Product_ID = @Product_ID
    AND PC.Active = 1
GROUP BY
    PC.ProductCampaign_ID,
    PC.Price,
    PC.NumberOfOrdersRequired;
```

Koda 6.6: SQL poizvedba za pridobitev akcijske cene.

```
<p id="anchororderform"></p>
<h2>Vaši podatki</h2>
@using (Html.BeginForm())
{
    <div class="body-form-gender">
        <div class="body-form-label">
            <label>Spol:</label>
        </div>
        <div class="body-form-input">
            @Html.RadioButtonFor(model => model.Gender, 1,
                new { @checked = "checked", @id = "f" })
            @Html.LabelFor(model => model.Gender, "Ženska",
                new { @for = "f" })
        </div>
    </div>
}
```

```
        @Html.RadioButtonFor(model => model.Gender, 2,
            new { @id = "m" })
        @Html.LabelFor(model => model.Gender, "Moški",
            new { @for = "m" })
    </div>
<div class="body-form-error">
    @Html.ValidationMessageFor(model => model.Gender)
</div>
</div>
<div class="body-form-name">
    <div class="body-form-label">
        @Html.LabelFor(model => model.Name, "Ime:")
    </div>
    <div class="body-form-input">
        @Html.TextBoxFor(model => model.Name)
    </div>
    <div class="body-form-error">
        @Html.ValidationMessageFor(model => model.Name)
    </div>
</div>
<div class="body-form-surname">
    <div class="body-form-label">
        @Html.LabelFor(model => model.Surname, "Priimek:")
    </div>
    <div class="body-form-input">
        @Html.TextBoxFor(model => model.Surname)
    </div>
    <div class="body-form-error">
        @Html.ValidationMessageFor(model => model.Surname)
    </div>
</div>
<div class="body-form-naslov">
    <div class="body-form-label">
        @Html.LabelFor(model => model.Address, "Naslov:")
    </div>
```

```
<div class="body-form-input">
    @Html.TextBoxFor(model => model.Address)
</div>
<div class="body-form-error">
    @Html.ValidationMessageFor(model => model.Address
    )
</div>
</div>
<div class="body-form-zip">
    <div class="body-form-label">
        @Html.LabelFor(model => model.Zip, "Pošta:")
    </div>
    <div class="body-form-input">
        @Html.TextBoxFor(model => model.Zip)
    </div>
    <div class="body-form-error">
        @Html.ValidationMessageFor(model => model.Zip)
    </div>
</div>
<div class="body-form-phone">
    <div class="body-form-label">
        @Html.LabelFor(model => model.Phone, "Telefon:")
    </div>
    <div class="body-form-input">
        @Html.TextBoxFor(model => model.Phone)
    </div>
    <div class="body-form-error">
        @Html.ValidationMessageFor(model => model.Phone)
    </div>
</div>
<div class="body-form-email">
    <div class="body-form-label">
        @Html.LabelFor(model => model.Email, "E-naslov:")
    </div>
    <div class="body-form-input">
        @Html.TextBoxFor(model => model.Email)
    </div>
    <div class="body-form-error">
```

```

        @Html.ValidationMessageFor(model => model.Email)
    </div>
</div>
<div class="body-form-comment">
    <div class="body-form-label">
        @Html.LabelFor(model => model.Coments, "Vaše
            sporočilo:")
    </div>
    <div class="body-form-input">
        @Html.TextAreaFor(model => model.Coments)
    </div>
    <div class="body-form-error">
        @Html.ValidationMessageFor(model => model.Coments
            )
    </div>
</div>
<div class="body-form-submit">
    <button type="submit">ODDAJ NAROČILO</button>
</div>
}

```

Koda 6.7: Pod podatki o izdelku izrišemo obrazec, kamor kupec vnese osebne podatke, da opravi naročilo.

```

[HttpPost]
public ActionResult Index(CustomerDataModel customer)
{
    HomeAccessor Accessor = new HomeAccessor();

    OrderAccessor OAccessor = new OrderAccessor();
    ProductModel product = new ProductModel();

    if (ModelState.IsValid)
    {
        int orderId = Accessor.InsertOrder(customer);

        product = OAccessor.GetProductByOrderID(
            orderId);
    }
}

```

```
        SendOrderConfirmationEmail(customer, product,
                                   orderId);

        return RedirectToAction("Index", "narocilo",
                                   new { id = orderId });
    }

    SetHomeData();
    return View();
}
```

Koda 6.8: Implementacija zapisa podatkov o naročilu se nahaja v Intex metodi v domovController.cs datoteki.

```
public int InsertOrder(CustomerDataModel customer)
{
    int order_id = 0;

    var connectionString = GetConnectionString();

    string sql = "EXEC [dbo].[Orders_InsertOrder] @Gender
                , @Name , @Surname , @Address , @Zip , @Email , @Phone ,
                @Coments ";

    using (SqlConnection connection = new SqlConnection(
        connectionString))
    {
        SqlCommand command = new SqlCommand(sql,
            connection);

        command.Parameters.Add("@Gender", SqlDbType.
            Int);
        command.Parameters["@Gender"].Value =
            customer.Gender;

        command.Parameters.Add("@Name", SqlDbType.
            NVarChar);
```

```
command.Parameters["@Name"].Value = customer.
    Name;

command.Parameters.Add("@Surname", SqlDbType.
    NVarChar);
command.Parameters["@Surname"].Value =
    customer.Surname;

command.Parameters.Add("@Address", SqlDbType.
    NVarChar);
command.Parameters["@Address"].Value =
    customer.Address;

command.Parameters.Add("@Zip", SqlDbType.
    NVarChar);
command.Parameters["@Zip"].Value = customer.
    Zip;

command.Parameters.Add("@Email", SqlDbType.
    NVarChar);
command.Parameters["@Email"].Value = customer
    .Email;

command.Parameters.Add("@Phone", SqlDbType.
    NVarChar);
command.Parameters["@Phone"].Value = customer
    .Phone;

command.Parameters.Add("@Coments", SqlDbType.
    NVarChar);
if (customer.Coments != null)
{
    command.Parameters["@Coments"].Value
        = customer.Coments;
}
else
{
    command.Parameters["@Coments"].Value
```

```
        = DBNull.Value;
    }

    try
    {
        connection.Open();
        order_id = (int)command.ExecuteScalar
            ();
    }
    catch (Exception exception)
    {
        throw exception;
    }

    return order_id;
}
}
```

Koda 6.9: Pošiljanje podatkov v podatkovno bazo se zgodi v metodi InsertOrder.

```
ALTER PROCEDURE [dbo].[Orders_InsertOrder]
    @Gender INT,
    @Name NVARCHAR(255),
    @Surname NVARCHAR(255),
    @Address NVARCHAR(255),
    @Zip NVARCHAR(255),
    @Email NVARCHAR(255),
    @Phone NVARCHAR(255),
    @Coments NVARCHAR(4000) = NULL
AS
BEGIN
    SET NOCOUNT ON;
    SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;

    DECLARE @Product_ID INT;
    DECLARE @ProductCampaign_ID INT;
    DECLARE @ProductName NVARCHAR(500);
```

```
DECLARE @NormalPrice FLOAT;
DECLARE @Price FLOAT;
DECLARE @VatRate FLOAT;

SELECT TOP(1)
    @Product_ID = P.Product_ID,
    @ProductCampaign_ID = PC.ProductCampaign_ID,
    @ProductName = P.Name,
    @NormalPrice = P.Price,
    @Price = PC.Price,
    @VatRate = VR.VatRate
FROM dbo.Products AS P
    INNER JOIN dbo.VatRates AS VR ON VR.
        VatRate_ID = P.VatRate_ID
    LEFT JOIN dbo.ProductCampaigns AS PC ON PC.
        Product_ID = P.Product_ID AND PC.Active =
        1
ORDER BY
    P.SortOrder DESC,
    P.Product_ID DESC;

INSERT INTO dbo.Orders(
    Product_ID,
    ProductCampaign_ID,
    Gender,
    Name,
    Surname,
    [Address],
    Zip,
    Email,
    Phone,
    Coments,
    ProductName,
    NormalPrice,
    Price,
    VatRate
)
VALUES (
```

```
        @Product_ID ,
        @ProductCampaign_ID ,
        @Gender ,
        @Name ,
        @Surname ,
        @Address ,
        @Zip ,
        @Email ,
        @Phone ,
        @Coments ,
        @ProductName ,
        @NormalPrice ,
        @Price ,
        @VatRate
    );
```

```
        SELECT CAST(SCOPE_IDENTITY() AS INT);
END
```

Koda 6.10: Za zapis v podatkov v podatkovno bazo poskrbi naslednja SQL poizvedba, ki je shranjena kot stored procedura.

```
private void SendOrderConfirmationEmail(CustomerDataModel
    customer, ProductModel product, int orderId)
{
    MailMessage message = new MailMessage();
    message.From = new MailAddress("info@mitjapetan.si");

    message.To.Add(new MailAddress(customer.Email));
    message.Bcc.Add(new MailAddress("mitja.petan@gmail.com"));

    message.Subject = "Hvala za vaše naročilo";
    message.Body = CreateOrderConfirmationMessage(
        customer, product, orderId);
    message.IsBodyHtml = true;

    SmtClient client = new SmtClient();
```

```
        client.Send(message);  
  
        client.Dispose();  
        message.Dispose();  
    }
```

Koda 6.11: Pošiljanje elektronskega sporočila.