

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 3518

INTERNET PRODAVAONICA RUKOTVORINA

Igor Bencetić

Zagreb, lipanj 2014.

Zagreb, 10. ožujka 2014.

ZAVRŠNI ZADATAK br. 3518

Pristupnik: **Igor Bencetić**
Studij: Računarstvo
Modul: Telekomunikacije i informatika

Zadatak: **Internet prodavaonica rukotvorina**

Opis zadatka:

Proučiti organizaciju internetskih dućana, uočiti zajedničke i specifične elemente. Oblikovati model baze podataka internetske prodavaonice rukotvorina. Napraviti web aplikaciju koja će administratorima omogućiti upravljanje prodavaonicom (definiranje proizvoda, cijena, popusta, itd.) a korisnicima standardne mogućnosti koje nudi internet prodavaonica (registracija, pretraživanjem, košarica, kupovina itd.).

Uz otisnuti primjerak rada priložiti optički disk s izgrađenom programskom potporom u izvornom i izvršnom obliku, te tekstom rada.

Zadatak uručen pristupniku: 14. ožujka 2014.

Rok za predaju rada: 13. lipnja 2014.

Mentor:




Doc.dr.sc. Igor Mekterović

Djelovođa:



Doc.dr.sc. Krešimir Pripužić

Predsjednik odbora za
završni rad modula:



Prof.dr.sc. Igor Sunday Pandžić

Sadržaj

Uvod	1
1. Korištene tehnologije	2
1.1. Apache Struts	2
1.2. Apache Tomcat	3
1.3. Bootstrap	4
1.4. Hibernate	4
1.5. MySQL	5
1.6. Sitemesh	6
1.7. Spring framewok	6
2. Internet prodavaonice	8
3. Model internet prodavaonice	10
4. Opis ostvarene funkcionalnosti	13
Zaključak	19
Literatura	20
Sažetak	21
Ključne riječi	22
Upute za instalaciju	23
JDK (<i>Java Development Kit</i>)	23
Apache Tomcat	24
MySQL	24

Uvod

Internetske prodavaonice (eng. *webshop*) su s razvojem interneta i poboljšanjem sigurnosti postale sve popularniji način distribucije robe. Najveća prednost internetskih prodavaonica je to što je krug korisnika mnogo veći nego kod fizičkih trgovina. Internetskoj prodavaonici moguće je pristupiti bilo gdje na svijetu, uz uvjet da postoji internetska veza. Ukoliko internetska prodavaonica dostavlja na zadanu adresu narudžba može biti izvršena bez da korisnik napusti svoj dom. Još jedna prednost za kupce je i velik izbor internetskih prodavaonica pa je moguće odabrati onu koja ima najpovoljnije uvjete.

Razvoj raznih programskih okvira (eng. *framework*) i dodatnih biblioteka (eng. *libraries*) za razvoj aplikacija uvelike je olakšana izrada internetskih aplikacija. Upravo zbog toga što je razvoj samih aplikacija postao jednostavniji pokretanje vlastite internetske prodavaonice nikada nije bilo lakše.

Internet prodavaonica rukotvorina, ostvarena u ovom radu, je primjer jedne jednostavne internetske aplikacije koja omogućava korisniku da upravlja svojom internetskom prodavaonicom.

1. Korištene tehnologije

Da bi se ubrzala i olakšala izrada internetskih aplikacija razvijen je velik broj radnih okvira i specijaliziranih biblioteka. Prvi zadatak programera je na osnovu zahtjeva projekta odabrati koje će tehnologije koristiti u razvoju aplikacije. U nastavku poglavlja će biti opisane korištene tehnologije.

1.1. Apache Struts

Apache Struts je radni okvir otvorenog koda za razvoj internetskih aplikacija u programskom jeziku Java. Podržava razvoj aplikacija baziranih na MVC modelu da bi se odvojili pojedini poslovi unutar aplikacije i olakšalo održavanje. Model se sastoji od tri komponente:

1. **Model** koji je zadužen za podatke i u pravilu sadržava bazu podataka i logiku za obradu i upravljanje podacima.
2. **Pogled** (eng. *view*) koji šalje zahtjeve u kojima traži podatke koji su mu potrebni da bi korisniku prikazao traženi izlaz.
3. **Upravljač** (eng. *controller*) koji posreduje u komunikaciji između pogleda i modela.

U ranijim radnim okvirima temeljenima na modelu MVC veći dio obrade podataka odvijao se na poslužiteljskoj strani, dok je klijent primao gotovu stranicu. Razvojem tehnologije internetske stranice postale su složenije i obrada podataka isključivo na poslužiteljskoj strani postala je nepraktična. Zbog toga je u novijim radnim okvirima temeljenima na modelu MVC omogućeno prebacivanje dijela odgovornosti za obradu podataka na klijentsku stranu. U vezanoj aplikaciji poslužiteljska strana se brine o komunikaciji s bazom podataka i šalje klijentu gotove podatke za obradu i prikazivanje.

```

<action name="DeleteAddress" class="actions.user.DeleteAddress">
  <result name="success" type="redirect">user/AddressManagement.action</result>
  <result name="input" type="redirect">user/AddressManagement.action</result>
</action>
<action name="AddAddress" class="actions.user.AddAddress">
  <result name="success" type="redirect">user/AddressManagement.action</result>
  <result name="input" type="redirect">user/AddressManagement.action</result>
</action>

```

Slika 1.1 Definiranje akcija u Struts radnom okviru

Kod rada sa Struts radnim okvirom klijent poziva akcije koje su definirane u datoteci s postavkama. Nakon poziva akcije upravljački dio poziva odgovarajući razred koji izvršava svoj kod nad modelom i od njega prima stranicu koju treba proslijediti pogledu. Podaci između modela i pogleda prenose se kao *JavaBeans* objekti iz kojih se podaci mogu čitati i prikazivati na željeni način. *JavaBean* je razred koji je oblikovan prema propisanoj specifikaciji tako da zadovoljava slijedeće zahtjeve:

- Implementira sučelje za ispis podataka u obliku niza bitova kako bi se omogućio njihov prijenos.
- Definirane javne metode za dohvat i postavljanje atributa razreda.
- Definirana javna metoda za stvaranje novog objekta bez argumenata.

1.2. Apache Tomcat

Apache Tomcat je internetski poslužitelj otvorenog koda koji ostvaruje Java Servlet i JSP (*JavaServer Pages*) specifikacije. Servlet je razred u programskom jeziku Java, a koristi se za proširivanje funkcionalnosti poslužitelja. Kod izrade aplikacije korišten je za obradu obrazaca, dohvat podataka iz baze podataka i upravljanje sjednicom (eng. *session*). Pošto HTTP (HyperText Transfer Protocol) nema trenutno stanje (poslužitelj ne zadržava podatke o prethodnim zahtjevima), pomoću *servleta* smo sjednici pridružili podatke koji su nam potrebni tokom korištenja aplikacije (npr. košarica s proizvodima). JSP je tehnologija koja programeru omogućava korištenje Java koda unutar stranice da bi se olakšala izrada dinamičnih internetskih stranica. U aplikaciji je korištena kao komponenta MVC modela koja predstavlja pogled.

1.3. Bootstrap

Bootstrap je skup alata izrađen kako bio olakšao i ubrzao izradu internetskih stranica. U sebi sadrži razne komponente koje pomažu programeru u izradi dizajna stranice. Optimiziran je za korištenje sa svim često korištenim preglednicima da programer ne bi morao prilagođavati stranicu ovisno o korištenom pregledniku. Posebno je prilagođen za pregledavanje na uređajima raznih veličina ekrana da bi korisnik mogao pristupiti svojoj funkcionalnosti stranice bez obzira na korišteni uređaj. Osim osnovnih alata postoji i cijeli niz nadogradnji koje se mogu preuzeti i koristiti u kodu. U izradi aplikacije korišten je za izradu obrazaca i izbornika. Korištene su i dodatne komponente za specifične potrebe (npr. povezani odabir datuma). Dodatne komponente se uglavnom oslanjaju na jQuery biblioteku.

1.4. Hibernate

Hibernate je biblioteka koja omogućava objektno-relacijsko preslikavanje, odnosno preslikavanje tablica relacijske baze podataka u razrede kao i preslikavanje između tipova podataka. Hibernate olakšava upravljanje vezama prema bazi podataka i sadrži metode za postavljanje upita i obradu rezultata. Osim tablica preslikavaju se i veze među tablicama te se tako unutar jednog upita omogućava dohvat svih podataka vezanih za određeni objekt. Na priloženoj slici možemo vidjeti Java razred *Category* u koji se preslikava tablica *category* baze podataka. Svaki stupac tablice povezan je s odgovarajućom varijablom razreda. Prikazana je i definicija veze između tablica *category* i *product*, prema kojoj se tablice spajaju pomoću glavnog ključa tablice *category*. Pošto svakoj kategoriji može pripadati više proizvoda definirana je veza *OneToMany*. Također, definirana je varijabla *products* u koju se zapisuju svi proizvodi povezani s kategorijom u pitanju.


```

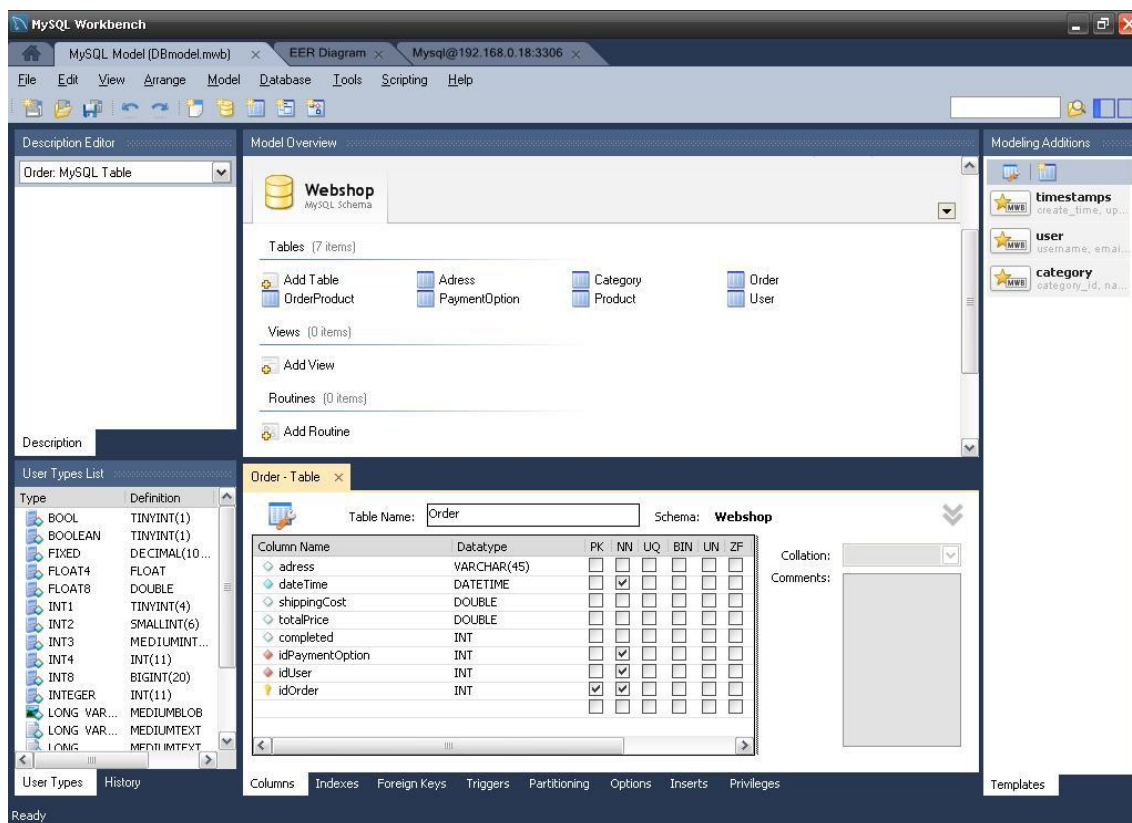
6
7 @Entity
8 @Table(name="category")
9 public class Category {
10
11     @Id @GeneratedValue
12     @Column(name="idCategory")
13     private int idCategory;
14
15     @Column(name="nameCategory")
16     private String nameCategory;
17
18     @Column(name="isDeleted")
19     private int isDeleted;
20
21
22     @OneToMany(mappedBy="category")
23     private Set<Product> products;
24
25     public Set<Product> getProducts(){
26         return products;
27     }

```

Slika 1.2 Preslikavanje tablice pomoću Hibernate biblioteke

1.5. MySQL

MySQL je sustav otvorenog koda za upravljanje relacijskom bazom podataka. Održava ga i razvija tvrtka Oracle. Za razvoj modela relacijske baze podataka korišten je alat MySQL Workbench.



Slika 1.3 MySQL Workbench

1.6. Sitemesh

Sitemesh je radni okvir za razvoj internetskih aplikacija u programskom jeziku Java. Koristi se za dekoraciju stranica, tako da presreće zahtjeve za stranice i ovisno o sadržaju generira konačnu stranicu. U aplikaciji je korišten za dodavanje dijelova stranice koji se nalaze u svim pogledima, u ovom slučaju to su navigacijska traka i podnožje stranice. Korištenjem Sitemesh radnog okvira izbjeglo se ponavljanje koda za generiranje uvijek prisutnih dijelova stranice. Također, poboljšana je čitljivost koda i omogućena izmjena često ponavljanih dijelova stranice na jednom mjestu.

1.7. Spring framewok

Spring je radni okvir otvorenog koda za razvoj aplikacija u programskom jeziku Java. Spring je modularan, što znači da je moguće koristiti samo dio njegove pune funkcionalnosti koji nam je potreban. U aplikaciji su, osim osnovnog

modula, korišteni i modul za integraciju sa Struts radnim okvirom, te modul za integraciju Hibernate sučelja.

Bitan koncept, ostvaren unutar osnovnog modula, je inverzija upravljanja (eng. *Inversion-of-Control*). Inverzija upravljanja je koncept kod kojega je tok upravljanja suprotan onome u imperativnom stilu programiranja. Kod inverzije upravljanja radni okvir poziva metode i procedure koje je napisao programer, za razliku od uobičajenog modela gdje programer poziva metode i procedure koje su definirane u radnom okviru. Ovaj se koncept koristi za ubacivanje zavisnih objekata (eng. *dependency injection*) kod pokretanja aplikacije. Zavisni objekti se ubacuju kod pokretanja pa programer ne mora voditi brigu o dostupnosti zavisnih objekata kod prevođenja. Primjer korištenja inverzije upravljanja u aplikaciji je razred koji služi za interakciju s bazom podataka (*Services*). Svi razredi koji rade s bazom podataka imaju definiranu nepostavljenu varijablu tipa *Services* i metodu za postavljanje iste. Kod stvaranja objekta nekog od razreda Spring će stvoriti novi objekt tipa *Services* i odgovarajućom varijablom unutar razreda pokazati na njega.

2. Internet prodavaonice

Koncept prodaje putem interneta osmislio je engleski poduzetnik Michael Aldrich još 1979. godine. Za razvoj internetskih prodavaonica kakvima ih danas poznajemo zaslužan je razvoj nekoliko tehnologija i otvaranje nekoliko, danas uspješnih, internetskih prodavaonica 90-ih godina prošlog stoljeća i to:

- Razvoj prvog *World Wide Web* poslužitelja i internetskog preglednika 1990. godine i njihova komercijalizacija 1991. godine.
- Razvoj internetskog bankarstva 1994. godine.
- Razvoj kriptografskog protokola SSL tvrtke Netscape.
- Otvaranje internetskih prodavaonica Amazon i eBay 1995. godine

Kako bi izvršio kupnju putem interneta kupac mora imati pristup internetu i važeće sredstvo plaćanja. Kupac proizvode može naći odlaskom izravno na stranice prodavatelja ili putem nekog od pretraživača trgovina koji uglavnom pruža i uslugu usporedbe cijena traženog proizvoda u različitim internetskim prodavaonicama. Kako bi se pojednostavila kupnja više proizvoda većina internetskih prodavaonica koristi virtualna kolica za kupovinu. Sljedeći korak u kupovini je potvrda narudžbe i unos podataka vezanih za plaćanje i adresu dostave. Najčešće korištena sredstva plaćanja su kreditne kartice i servisi za plaćanje (npr. PayPal), međutim postoji cijeli niz alternativa.

Internetske prodavaonice često nude i sustav registracije i prijave korisnika. Na taj način korisnik može samo jednom unijeti podatke potrebne za kupovinu. Također, na taj način je korisniku dostupan popis prošlih narudžbi.

Prodaja putem interneta ima sljedeće prednosti nad tradicionalnim fizičkim (eng. *brick and mortar*) trgovinama:

- Prodavaonice su otvorene 24 sata na dan. Osim što ne mora fizički putovati do trgovine kupac ima mogućnost obaviti narudžbu u bilo koje vrijeme što uklanja problem dolaska u prodavaonicu unutar radnog vremena.

- Iako kupac može procijeniti proizvod samo po priloženim slikama i opisu često postoji mogućnost pisanja recenzije naručenog proizvoda. Na taj način kupac može vidjeti iskustva drugih ljudi s nekim proizvodom i tako si olakšati odluku o kupovini.
- Pošto je putem internetske prodavaonice lakše pokriti veliko područje uslugom prodaje, kupci će u većini slučajeva imati širok izbor prodavaonica na raspolaganju.

Dok prodaja i kupovina putem interneta ima mnoge prednosti, ipak postoji i nekolicina nedostataka:

- Ukoliko je kupac nezadovoljan proizvodom uglavnom je odgovoran sam platiti cijenu dostave kod vraćanja proizvoda i zatim čekati zamjenski proizvod ili povrat sredstava.
- Kupac je pri pregledu proizvoda ograničen na priložene slike i opis, dok bi u fizičkoj trgovini mogao pregledati proizvod i konzultirati se s osobljem.
- Postoji rizik od internetskih prijevara. Kupcima je teže provjeriti kredibilitet internetske prodavaonice pa postoji rizik od krađe osobnih podataka.
- Krajnja cijena naručenih proizvoda nije uvijek očita. Postoji mogućnost dodatnih troškova na pošiljku u vidu carina i sl.

Ovaj rad je posvećen izradi internetske prodavaonice rukotvorina i zamišljen je kao privatna prodavaonica raznim ručno izrađenim predmetima.

3. Model internet prodavaonice

Korisnik

Za obavljanje kupovine korisnik se mora registrirati. Tijekom registracije unose se osnovni podaci: *ime*, *e-mail* i *lozinka*. Nakon registracije korisnik može uz svoj korisnički račun vezati više adresa između kojih odabire na koju želi primiti narudžbu.

Adresa

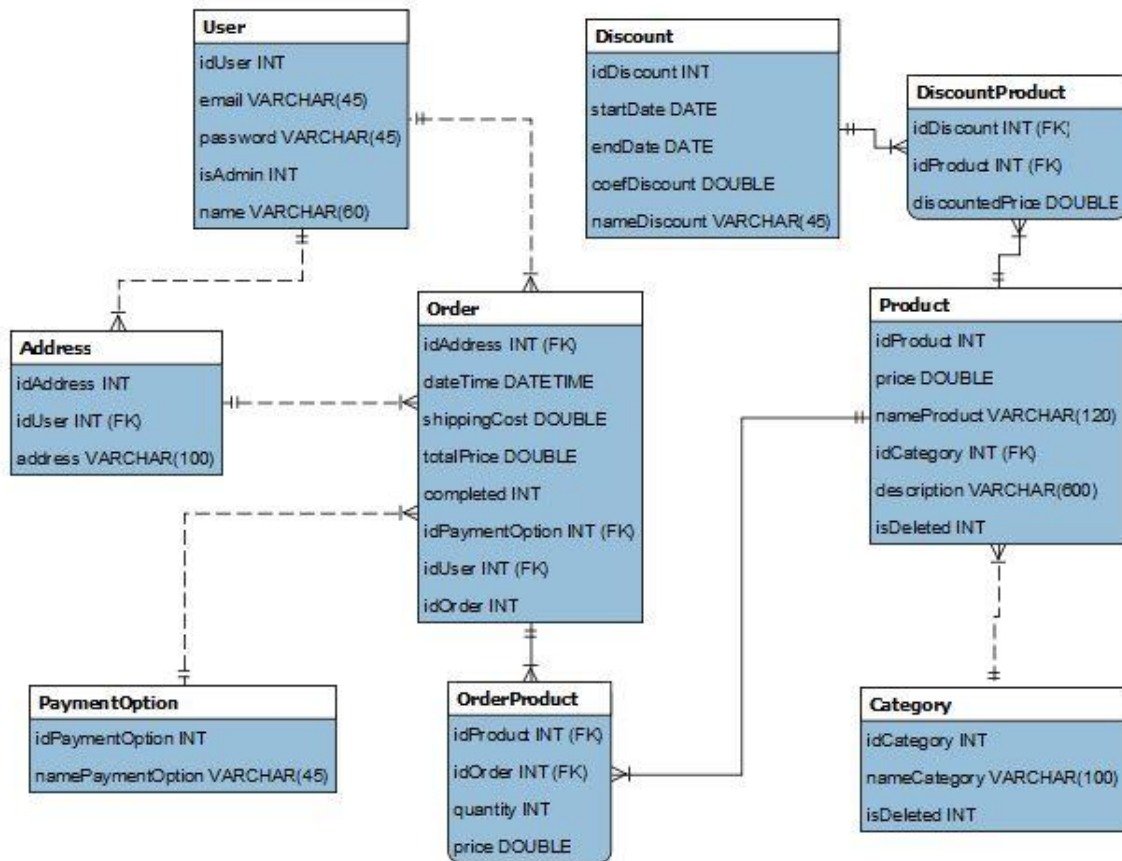
Svaka adresa je vezana za jednog korisnika i sadrži sve podatke koji su potrebni da bi pošiljka mogla stići na željenu destinaciju.

Proizvod

Proizvodi su temelj internetske prodavaonice. Svaki proizvod sadrži identifikacijski broj *idProduct* i identifikacijski broj kategorije proizvoda kojoj pripada *idCategory*. Polja koja opisuju proizvod su: ime (*name*), cijena (*price*) i opis (*description*). Proizvod može pripadati samo jednoj kategoriji. Posljednji stupac u tablici je *isDeleted*, koji služi kao pokazatelj da li je proizvod još u ponudi. Naime, ako bi umjesto postavljanja vrijednosti *isDeleted* proizvod izbrisali iz baze podataka on nam više ne bi bio dostupan kod pregleda prijašnjih narudžbi.

Kategorija

Kategorija je definirana poljima *name* i *idCategory*. Svaka kategorija može sadržavati više proizvoda. Kao i kod proizvoda postoji polje *isDeleted* kako bi sačuvali podatke za pregled, a uklonili ih iz pregleda na samoj stranici.



Slika 3.1 Relacijski model baze podataka

Narudžba

Tablica narudžba sadrži podatke o narudžbama i povezana je putem stranih ključeva s tablicama *User*, *PaymentOption* i *Address*. Pošto pretpostavljamo da će proći neko vrijeme između zaprimanja narudžbe i slanja proizvoda korisniku, tablica sadrži *dateOrdered* i *dateShipped*, kako bi korisniku kod pregledavanja narudžbi pružili informaciju o statusu narudžbe.

Popust

Tablica popust sadrži popuste koje administrator internetske prodavaonice može definirati za jedan ili više proizvoda. Osim primarnog ključa *idDiscount* i imena popusta *nameDiscount*, sadrži kolone *startDate* i *endDate* koje određuju

period unutar kojega su odabrani proizvodi na popustu. Kolona *coefDiscount* sadrži podatak o tome koliki postotak cijene se odbija za vrijeme trajanja popusta.

Opcije plaćanja

Tablica s opcijama plaćanja sadrži primarni ključ i ime opcije plaćanja. Pošto u sklopu rada ostvarivanje sigurnog plaćanja nije predviđeno, služi kako bi se pokazalo da postoji više mogućih načina plaćanja putem interneta.

Veza popust-proizvod

Dvosmjerna N:N veza između tablica popust i proizvod ostvarena je spojom tablicom *DiscountProduct* koja ima kompozitni ključ koji se sastoji od primarnih ključeva tablica *Product* i *Discount*. Sadrži i dodatnu kolonu koja predstavlja cijenu proizvoda nakon što se na nju primjeni vezani popust.

Veza narudžba-proizvod

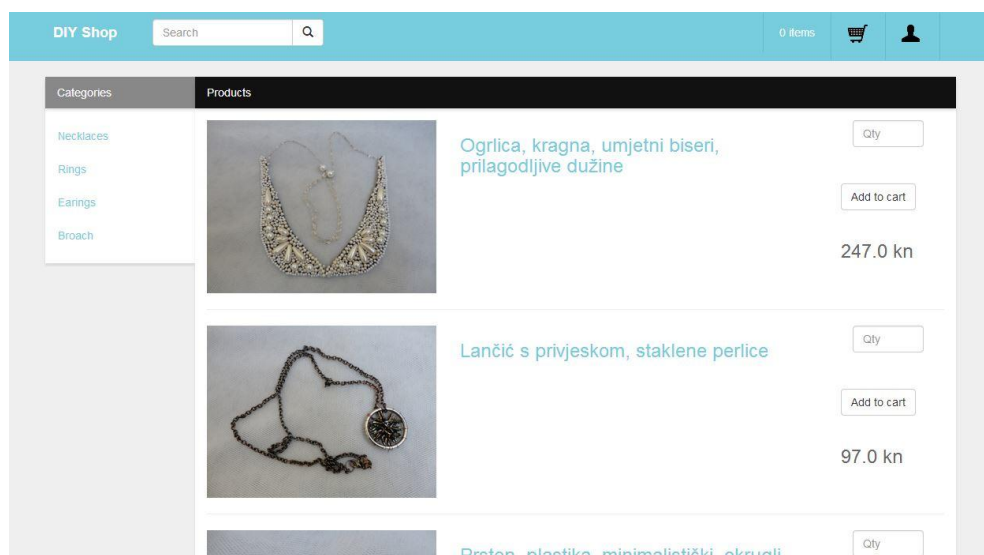
Dvosmjerna N:N veza između tablica narudžba i proizvod ostvarena je spojenom tablicom *OrderProduct* koja ima kompozitni ključ koji se sastoji od primarnih ključeva tablica *Order* i *Product*. Sadrži i dvije dodatne kolone. Prva dodatna kolona sadrži podatak o količini nekog proizvoda u narudžbi. Druga dodatna kolona služi kako bi se pohranila cijena proizvoda u trenutku kupovine. Pošto se cijena proizvoda može mijenjati definiranjem popusta, podatak o cijeni nam omogućava da ispišemo podatke o prošlim narudžbama koristeći cijene koje su bile aktualne u tom trenutku.

4. Opis ostvarene funkcionalnosti

Internetska prodavaonica ostvarena u sklopu zadatka sadrži sljedeće funkcionalnosti:

Pregledavanje artikala

Korisnik ima pristup pregledu dostupnih proizvoda, njihovih opisa i cijena. Svaki proizvod ima svoju stranicu na kojoj korisnik može vidjeti detaljniji opis. Proizvodi su razvrstani po kategorijama koje su ispisane u alatnoj traci i pomoću kojih korisnik može lakše pronaći željeni proizvod. Ostvarena je i jednostavna tražilica koja provjerava pojavljuje li se traženi pojam u imenu ili opisu proizvoda. Iz pregleda i detaljnog prikaza proizvoda korisniku ja omogućeno dodavanje proizvoljne količine u košaricu (eng. *shopping cart*).



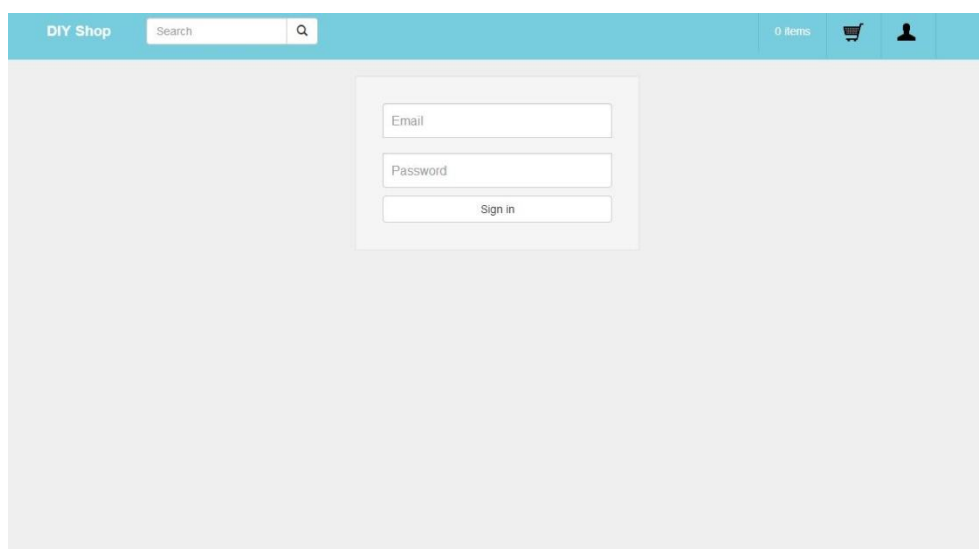
Slika 4.1 Prikaz proizvoda i kategorija

Košarica

Košarica se u internetskim prodavaonicama koristi kako bi korisnik mogao odabrati više različitih proizvoda, kao i odrediti količinu pojedinačnog odabranog proizvoda. Time se korisniku omogućuje da na završetku kupovine obavi jednu transakciju za sve odabrane proizvode dok bi u protivnom slučaju morao vršiti po jednu transakciju za svaki proizvod, a ukoliko bi želio veću količinu istog proizvoda ponavljao bi postupak kupnje onoliko puta koliko proizvoda želi. Kako bi se zadržali podatci o odabranim proizvodima dok korisnik pregledava sjednici je pridružen par: ključ, vrijednost pod imenom *košarica*.

Slanje narudžbe

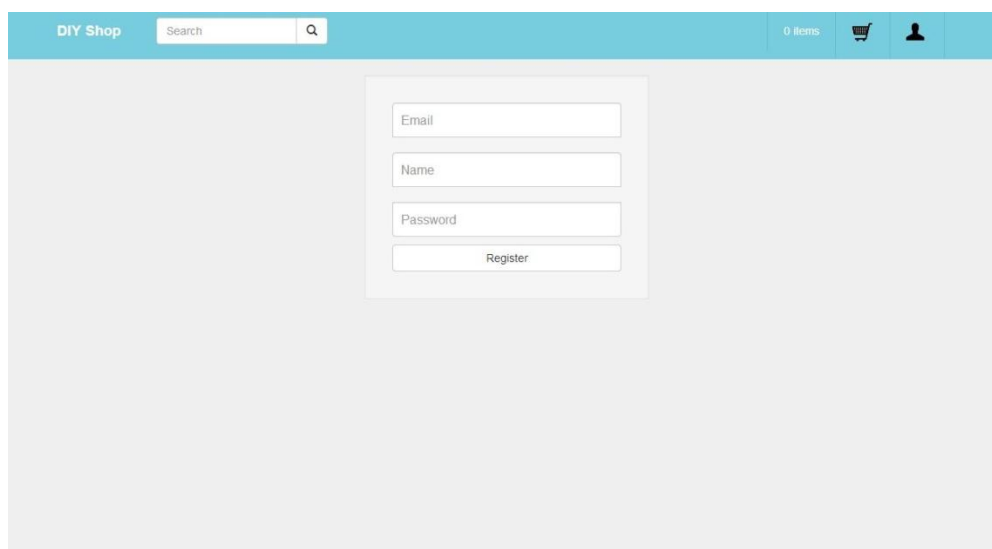
Korisniku je omogućeno da nakon odabira željenih proizvoda kroz poveznicu u pregledu košarice pošalje narudžbu na obradu. Prvi korak je provjera je li korisnik prijavljen u sustav. Ukoliko korisnik nije prijavljen biti će preusmjeren na stranicu za prijavu, a u protivnom ga aplikacija preusmjerava na pogled u kojemu odabire parametre narudžbe. Pod parametrima se podrazumijeva odabir jedne od prije definiranih adresa za dostavu narudžbe i odabir načina plaćanja naručenih proizvoda. U aplikaciji će se podrazumijevati da je transakcija uspješna i narudžba će biti poslana na obradu.

The image shows a web application interface for 'DIY Shop'. At the top, there is a teal header bar containing the shop name 'DIY Shop', a search bar with the placeholder text 'Search' and a magnifying glass icon, and on the right, a '0 Items' label, a shopping cart icon, and a user profile icon. The main content area has a light gray background. In the center, there is a white rectangular box with a thin gray border. Inside this box, there are two input fields: the top one is labeled 'Email' and the bottom one is labeled 'Password'. Below these two fields is a button labeled 'Sign in'.

Slika 4.2 Stranica za prijavu

Prijava i registracija

Kako bi obavio kupnju korisnik se mora prijaviti na sustav. Prijava je moguća samo ako postoji korisnički račun s kombinacijom e-mail adrese i lozinke. Kako bi stvorio novi korisnički račun korisnik odabire opciju registracije i unosi tražene podatke. Svaki zahtjev za pristup stranici se prije izvršavanja provjerava kako bi korisnik ne bi mogao pristupiti stranicama za koje nema odgovarajuće odobrenje. Tako, primjerice, korisnik može pregledavati proizvode i koristiti košaricu bez prethodne prijave u sustav, ali ne može poslati narudžbu ili pristupiti upravljanju adresama i pregledu narudžbi. Isto tako, samo korisnik koji je označen kao administrator ima pristup stranicama za upravljanjem internetskom prodavaonicom.

The image shows a web application interface for a shop named "DIY Shop". At the top, there is a teal header bar containing the shop name, a search bar with a magnifying glass icon, and navigation links for "0 Items", a shopping cart icon, and a user profile icon. The main content area is light gray and features a central white registration form. This form includes three input fields labeled "Email", "Name", and "Password", followed by a "Register" button.

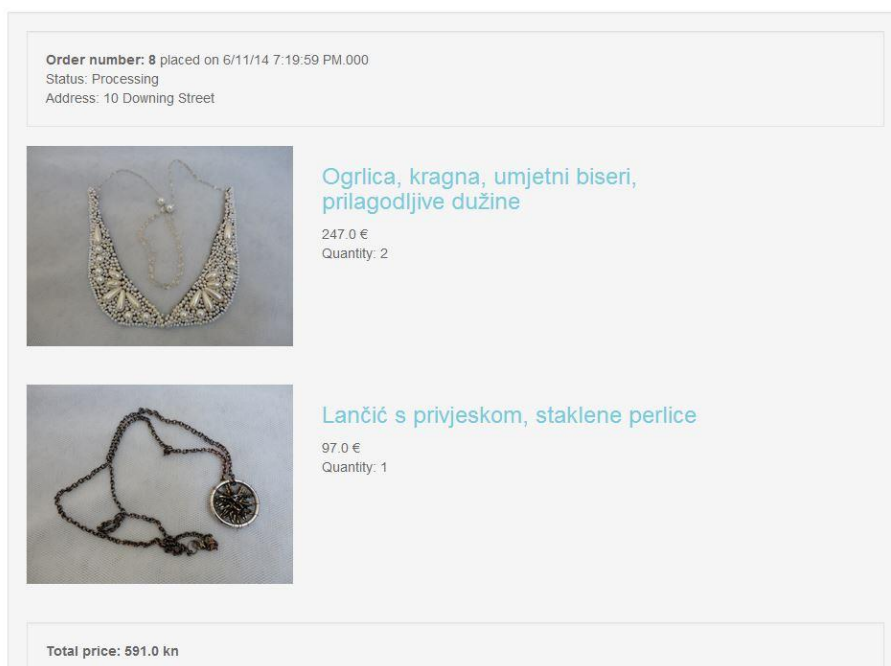
Slika 4.3 Stranica za registraciju

Upravljanje adresama

Korisnik koji je prijavljen u sustav ima mogućnost upravljanja adresama. U slučaju da želi dostavu naručene robe na različitu adresu od prije upisane, može dodati novu adresu koja će mu tada biti ponuđena za odabir kod kupnje. Postoji i mogućnost uklanjanja adrese ukoliko korisniku više nije potrebna.

Pregled narudžbi

Korisniku je unutar profila omogućeno pregledavanje povijesti naručivanja. Za svaku prošlu narudžbu ispisani su podaci o adresi, ukupnoj cijeni, kao i popis artikala, njihove količine i cijena. Ispisano je i stanje narudžbe koje govori korisniku jesu li naručeni proizvodi poslani na odabranu adresu ili je narudžba još u obradi.

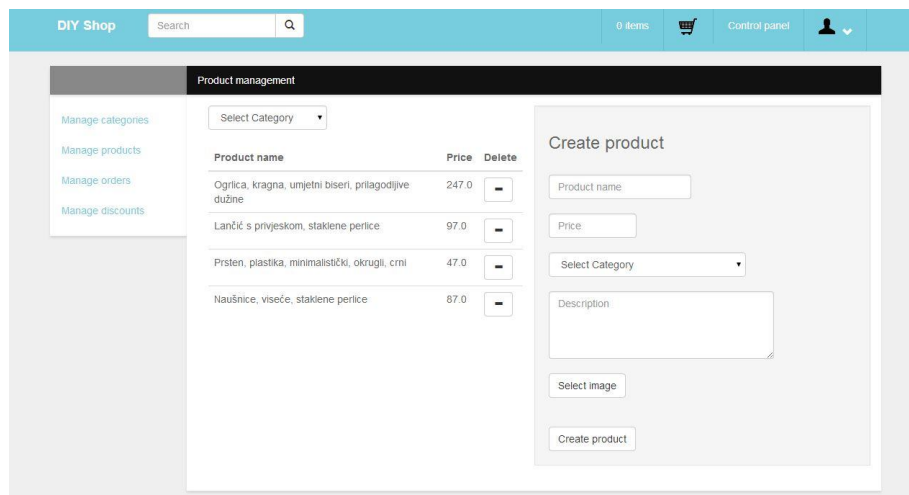


Slika 4.4 Pregled narudžbi

Upravljanje kategorijama i proizvodima

Da bi se omogućilo ažuriranje ponude internetske prodavaonice ostvareni su pogledi za upravljanje kategorijama i proizvodima. Kategorije razdvajaju skup proizvoda na manje cjeline kako bi se olakšalo pregledavanje i pronalaženje pojedinih proizvoda. Uklanjanjem kategorije iz ponude se izbacuju svi proizvodi

koji pripadaju toj kategoriji. Dodavanjem i uklanjanjem pojedinačnih proizvoda iz pojedine kategorije omogućeno je održavanje ponude aktualnom.



Slika 4.5 Upravljanje proizvodima

Definiranje popusta

Kako bi administrator internetske prodavaonice mogao ponuditi kupcima proizvode određene kategorije ili pojedine proizvode po sniženim cijenama ostvareno je definiranje popusta. Kod stvaranja novog popusta administrator odabire kategoriju ili pojedine proizvode koje želi pridružiti popustu. Nakon toga odabire datum početka i datum završetka popusta, te postotak za koji se smanjuje cijena odabranih proizvoda. U slučaju da je nad nekim proizvodom u određenom trenutku definirano više popusta oni se ne zbrajaju, nego se uzima onaj najveći.

Discount name	Start Date	End Date	Percent
a	6/2/14	6/4/14	0.03
bozic	6/5/14	6/6/14	0.1
proba	6/5/14	6/13/14	0.2

Slika 4.6 Upravljanje popustima

Upravljanje narudžbama


Administratoru prodavaonice je omogućen pregled neizvršenih narudžbi. Za svaku narudžbu ispisuju se podaci potrebni za dostavu tj. ime i adresa, pojedini proizvodi sadržani u narudžbi te njihove količine. U konačnici je ispisana ukupna cijena narudžbe. Administrator ima mogućnost označiti narudžbu izvršenom čime se ona više ne prikazuje na stranici za upravljanje narudžbama.

Order number: 8 placed on 6/11/14 7:19:59 PM.000

Status: Processing

User: admin


Address: 10 Downing Street



Ogrlica, kragna, umjetni biseri, prilagodljive dužine

247.0 €

Quantity: 2



Lančić s privjeskom, staklene perlice

97.0 €

Quantity: 1

Total price: 591.0 €

Slika 4.7 Upravljanje narudžbama

Zaključak

Zbog pogodnosti koje pruža kupcu, kao i zbog prednosti za prodavatelje internetske prodavaonice postale su sve učestalije. Ovisno o željenoj funkcionalnosti, postoji cijeli niz radnih okvira i biblioteka koji olakšavaju razvoj internetskih aplikacija. U sklopu rada ostvarena je aplikacija za čiji razvoj je korištena kombinacija više radnih okvira. Središnji radni okvir je Spring, koji podržava integraciju s radnim okvirom Struts i bibliotekom Hibernate i koristi se za ubacivanje zavisnosti. Radni okvir Struts korišten je za prikaz stranica i upravljanje tokom aplikacije. Hibernate se koristi za preslikavanje podataka iz baze podataka u objekte. Za izradu sučelja korišteni su radni okviri Bootstrap i Sitemesh. Nedostatak korištenja kombinacije radnih okvira svakako je potreba za dodatnom konfiguracijom.

Aplikacija ostvarena u radu oslikava jednostavnu internetsku prodavaonicu rukotvorina. Naglasak je stavljen na funkcionalnosti koje su ugrađene u većinu novijih prodavaonica. Putem ostvarenog grafičkog sučelja omogućeno je korištenje i upravljanje prodavaonicom.

Literatura

- [1] Oracle JavaBeans specifikacija --
<http://www.oracle.com/technetwork/java/javase/documentation/spec-136004.html> --Preuzeto 7. lipnja 2014.
- [2] Apache Struts framework -- <http://struts.apache.org> – Preuzeto 7. lipnja 2014.
- [3] Spring framework -- <http://projects.spring.io/spring-framework> -- Preuzeto 7. lipnja 2014.
- [4] Razvoj internetskih prodavaonica --
http://www.aldricharchive.com/internet_online_shopping.html -- Preuzeto 7. lipnja 2014.
- [5] Bootstrap -- <http://getbootstrap.com> -- Preuzeto 7. lipnja 2014.
- [6] Apache Tomcat server -- <http://tomcat.apache.org> -- Preuzeto 7. lipnja 2014.
- [7] Hibernate ORM -- <http://hibernate.org/orm> -- Preuzeto 7. lipnja 2014.
- [8] HTTP protokol -- <http://www.devpoets.com/http-is-a-stateless-protocol/> --
Preuzeto 7. lipnja 2014.
- [9] Sitemesh -- <https://today.java.net/pub/a/today/2004/03/11/sitemesh.html> --
Preuzeto 7. lipnja 2014.

Sažetak

Rad opisuje razvoj jednostavne internetske prodavaonice rukotvorina. Navedene su i opisane tehnologije koje su korištene pri izradi aplikacije. Ukratko je opisan razvoj internetskih prodavaonica. Opisan je model prema kojemu je aplikacija razvijana. Detaljnije su opisani entiteti modela i veze među njima. Sljedeće poglavlje sadrži opis ostvarene funkcionalnosti s priloženim slikama koje prikazuju važnije dijelove aplikacije. Slijede upute za instalaciju aplikacije u kojima je navedeno što je sve potrebno učiniti da bi se aplikacija pokrenula.

Summary

This thesis describes the development of a handicraft web shop. All the technologies used during development are listed and described. There is a short summary of web shop evolution in general. The following chapter contains a description of the development model and a closer look at the entities and their relation. The next chapter lists the features and contains photos that illustrate some of them. The appendix contains the installation manual.

Ključne riječi

- Apache Struts
- Spring
- Hibernate
- Internetska prodavaonica

Keywords

- Apache Struts
- Spring
- Hibernate
- Web shop

Upute za instalaciju

Uz izvorni kod, kod predanog zadatka biti će uključeni WAR (*Web application ARchive*) datoteka koja sadrži internetsku aplikaciju i skripta za kreiranje pripadne baze podataka. Da bi se uspješno pokrenula aplikacija potrebno je instalirati JDK (*Java Development Kit*), poslužitelj Apache Tomcat i pokrenut MySQL poslužitelj u koji je učitana skripta za kreiranje baze podataka. Slijede detaljnije upute za konfiguraciju pojedinih potrebnih komponenti.

JDK (*Java Development Kit*)

Za instalaciju JDK-a najprije je potrebno preuzeti zadnju verziju istog za vaš operacijski sustav izravno sa stranice proizvođača. Zadnja verzija dostupna je na sljedećoj poveznici:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>. Verzija JDK-a ne smije biti niža od verzije 7 ili u suprotnom aplikacija neće raditi ispravno. Nakon preuzimanja potrebno je pokrenuti izvršni program i obaviti instalaciju. Po završetku instalacije potrebno je još podesiti sadržaj varijabli okruženja PATH i JAVA_HOME. Na operacijskom sustavu Windows za verziju 1.7.0_15 JDK-a se to čini sljedećim naredbama iz komandne linije:

```
SET "JAVA_HOME=c:\Program Files (x86)\java\jdk1.7.0_15"
```

```
SET "PATH=%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin;%PATH%"
```

Apache Tomcat

Zadnja verzija poslužitelja Apache Tomcat dostupna je na stranicama proizvođača: <http://tomcat.apache.org/download-80.cgi>. Za preuzimanje potrebno je kliknuti na odgovarajuću poveznicu u sekciji “Core” koja se nalazi pod sekcijom “Binary Distributions”. Ovisno o vašem operacijskom sustavu moguće je skinuti različite verzije, a za detalje pogledati: <http://tomcat.apache.org/tomcat-8.0-doc/setup.html>. Pokretanje Apache Tomcat poslužitelja je moguće napraviti dvoklikom na *startup.bat* kod operacijskog sustava Windows (ili *startup.sh* kod operacijskog sustava Linux), koji se nalazi u direktoriju *bin* unutar direktorija koji sadrži datoteke poslužitelja. Gašenje se izvršava dvoklikom na *shutdown.bat* kod operacijskog sustava Windows (ili *shutdown.sh* kod operacijskog sustava Linux), koji se nalazi u istom direktoriju kao i datoteka za pokretanje. Aplikacija se u Tomcat dodaje tako da se direktorij *webapps* koji je u Tomcatovom direktoriju doda direktorij doda priložena WAR datoteka.

WAR datoteka sadrži sve podatke potrebne za distribuciju internetske aplikacije u komprimiranom formatu. Služi da bi se olakšala distribucija aplikacije pošto ju je potrebno samo kopirati u za to predodređeni direktorij, a poslužitelj sam raspakirava aplikaciju kod pokretanja.

MySQL

MySQL baza podataka dostupna je na: <http://dev.mysql.com/downloads/mysql>. Potrebno je izabrati verziju za odgovarajući operacijski sustav i skinuti arhivu. Nakon instalacije potrebno je pokrenuti priloženu skriptu kako bi se stvorila baza podataka koju koristi gore opisana internetska aplikacija.

Također je potrebno podesiti postavke za spajanje na bazu podataka. Postavke se nalaze u direktoriju WEB-INF i to u datoteci *applicationContext.xml*

```
<bean id="dataSource" class="com.mchange.v2.c3p0.ComboPooledDataSource" destroy-method="close">
  <property name="driverClass" value="com.mysql.jdbc.Driver" />
  <property name="jdbcUrl" value="jdbc:mysql://localhost:3306/webshop" />
  <property name="user" value="admin" />
  <property name="password" value="admin" />
</bean>
```

Slika 0.1 Konfiguracija spajanja na bazu podataka

U gore prikazanoj odsječku koda vidljivi su podatci koje je potrebno postaviti da bi se internetska aplikacija mogla uspješno spojiti na bazu podataka. Potrebni su nam ime i lozinka za pristup bazi podataka. Također je potrebno definirati koji pokretački program (eng. *driver*) koristimo za spajanje na bazu podataka, kao i adresu na kojoj možemo pristupiti bazi podataka.