



UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I RAČUNARSTVO

Primena baze podataka za sajt E-book store

- Diplomski rad -

Mentor:

Prof. dr Mladen Veinović

Student:

Ivaniš Nebojša

Br. indeksa: 97/2006

Beograd, 2010.



FAKULTET ZA INFORMATIKU I RAČUNARSTVO

UNIVERZITET SINGIDUNUM

FAKULTET ZA INFORMATIKU I RAČUNARSTVO

Beograd, Zorana Đinđića 44

Kandidat: Nebojša Ivaniš

Broj indeksa: 97/2006

Smer: Projektovanje i programiranje

Tema: Primena baze podataka za sajt E-book store

Zadatak: Realizovati aplikaciju za elektronsku trgovinu prodaje knjiga. Aplikacija treba da podrži dinamički rad sajta i punu funkcionalnost za elektronsku prodaju knjiga. Odabrati tehnologiju za realizaciju sajta i odgovarajući sistem za upravljanje bazama podataka. Posebno objasniti modelovanje i tehniku povezivanja Web aplikacije i baze podataka.

Datum odobrenja rada:
Beograd, _____._____._____.

MENTOR

Prof. dr Mladen Veinović

DEKAN

Prof. dr Mladen Veinović

Primena baze podataka za sajt E-book store

Sažetak: U ovom diplomskom radu obrađen je proces modelovanja i projektovanja baze podataka knjige.mdb, koja se implementira na sajt E-book store. Uspostavljanjem konekcije, preko ODBC driver-a, između web aplikacije koja je pisana u programskom jeziku C# i HTML i CSS uz pomoć kojih je urađen dizajn stranica, kompletiran je izgled sajta E-book store, a uz obavezno prisustvo baze podataka knjige.mdb sajt stiče punu funkcionalnost. Kako bi projektovanje baze podataka moglo da se obavi na jednostavan, precizan ali i korektan način korišćeni su UML dijagrami. Time su se stekli uslovi za jednostavan rad i pravljenje troslojne arhitekture, a modelovanje je, proces realizacije svelo na mehaničko pisanje koda. Sajt E-book store postaje deo elektronskog biznisa i stiče konkurentnost na Internet tržištu.

Ključne reči: baza podataka, modelovanje, projektovanje, ODBC driver, C#, HTML, CSS

Database Application for E-book Store Website

Abstract: This graduate paper deals with the process of modeling and database design for books.mdb, which is implemented in the e-book store website. Establishing a connection through ODBC drivers, between a web application written in C# programming language and the unavoidable HTML and CSS languages used to design the pages, e-book store's website layout has been completed. Adding the required books.mdb database, website will achieve full functionality. In order to design the database in a simple, precise and accurate manner, UML diagrams have been used. Thus the conditions for simple operation and making three-fold layer architecture have been met, while modeling has reduced the implementation process to a mechanical writing code. E-book store website will become an integral part of the electronic business and will acquire competitiveness in the Internet market.

Keywords: database, modeling, designing, ODBC driver, C#, HTML, CSS

SADRŽAJ

1	Uvod	- 2 -
2	Primena baze podataka u elektronskoj prodaji knjiga	- 4 -
2.1	Arhitektura baze podataka.....	- 6 -
2.2	Komunikacija baze podataka i sajta.....	- 8 -
3	Realizacija baze podataka na sajtu E-book store.....	- 9 -
3.1	Proces modelovanja baze podataka	- 10 -
3.2	Proces konekcije baze podataka i sajta E-book store	- 22 -
4	Zaključak.....	- 37 -
	Literatura	- 38 -

1 Uvod

U diplomskom radu osnovna tematika je upotreba baze podataka radi lakše, dinamičnije, funkcionalnije primene i upotrebe sajta za prodaju knjiga u elektronskoj formi. Baze podataka nam služe radi lakše organizacije, manipulacije i čuvanja podataka. Pojedinci ili kompanije svakodnevno koriste baze podataka, za šta možda i sami nisu svesni, ali pristupanjem nekom programu ili sajtu koji daje određene informacije na upit korisnika, aktivira se baza podataka i daje odgovor putem određenih metoda. Kao što je poznato, Internet kao pojava nosi veliki broj informacija za široku primenu u raznim segmentima poslovanja. Sve informacije moraju biti smeštene, najčešće fizički ali kako se infrastruktura globalne mreže svakodnevno širi i doživljava ekspanziju, podaci se, sve češće čuvaju i u elektronskom obliku uz pomoć baza podataka. U vezi sa tim, ljudska populacija mora da se „opismeni“ i prihvati Internet kao globalnu mrežu i kao svakodnevnicu, naravno uz poštovanje određenih zakona i administrativnih procedura.

Globalna mreža u današnje vreme polako prevazilazi samu sebe tako što postaje zasićena mnogobrojnim mogućnostima zarade i trgovine, što je i paradoks, jer je i sama ideja postojanja Interneta bila da se olakšaju transakcije novčanih sredstava i pojednostave i poboljšaju komercijalne potrebe. Naravno da globalna mreža ima još mnoštvo drugih mogućnosti i donosi pozitivan ton novim poslovima i omogućava dobitke kako velikim kompanijama tako i korisnicima kao jedinkama. Sa druge strane ima i mnoštvo nepravilnosti i zloupotreba i može napraviti pravne i ekonomske probleme firmama i zadati glavobolju običnom korisniku.

Kako se Internet širi i popunjava novim korisnicima, tako se javljaju novi programi za rad i edukaciju, ali i mogućnosti predstavljanja samih firmi i pojedinaca na istom. Samo povećanje uslovljava automatski i proširivanje dosadašnje infrastrukture što je povezano sa bazama podataka, jer se i same baze podataka moraju proširivati i menjati, tj. postajati fleksibilnije i dinamičnije kako bi mogle da odgovore svim upitima koje postavljaju korisnici. Kako bi se prodajna mreža i distribucija razvila čitavom planetom, tako postojanje i izrada sajtova dobijaju na težini i funkcionalnosti ali bez primene baze podataka sajtovi bi izgledali na putujuće slike koje bi kružile Internetom. Primena baze podataka je obavezna ako želimo da ostvarimo konkurentnost u odnosu na druge, na Internetu, tačnije globalnom mega marketu. Prednost ostvarujemo u dinamici misleći na brzinu odgovora na upit korisnika, a fleksibilnost, da se sa jednog mesta prilagodimo korisniku i njegovim prohtevima kao poslodavac i obezbedimo sigurnost kako bi korisnik imao osećaj da su mu zaštićeni integritet i identitet, što je i najbitnija stavka u svetu poslovanja u elektronskom biznisu.

Diplomski rad se bavi izradom sajta za prodaju knjiga u elektronskom formatu i prateće baze podataka na kojoj su smeštene knjige. Projekat E-book store u sebi sadrži opciju ne samo dostavljanja knjiga, jer je ta opcija postala ustaljena na mnogim sajtovima konkurencijske prodaje, već i da kupac može preuzeti sadržaj knjige koju kupi u elektronskoj formi. Kao što je na početku uvoda navedeno, Internet i samo postojanje informacionih tehnologija omogućavaju korisnicima da sve što im je potrebno za rad ne nose fizički sa sobom, nego da jednostavno, za sve svoje potrebe se obrate određenom sajtu. Tome u prilog idu i nove tehnologije TFT i Plazma monitora, koji funkcionalnost i operativnost omogućavaju dodirnom na sam ekran i time ostvaruju veliku lepezu opcija.

Zamislimo, da klijent sa druge strane zemljine lopte želi da kupi određenu knjigu i nađe je na sajtu. Postavlja se pitanje koliko dugo treba da čeka na isporuku. Odgovor na to pitanje je, otprilike nedelju dana. Ali ako je omogućeno da pri kupovini knjige korisnik ima mogućnost da skine sadržaj u elektronskoj formi, automatski firma postaje ekspeditivnija od svojih konkurenata.

Ne treba zaboraviti da je korišćenje knjiga kako stručnih tako i književnih u velikoj ekspanziji, kako kod mladih tako i kod starijih osoba. Time u prilog ide ideja o pravljenju sajta koji će omogućiti korisnicima da se jednostavno registruju, pristupe sajtu sa svojim korisničkim imenom i lozinkom i izborom knjige stavivši je u korpu, klikom na dugme, plate. U sledećem trenutku imaju opciju da izvrše preuzimanje knjige na vlastiti računar ili neki uređaj i tada knjiga postaje njihovo vlasništvo, uz obavezno poštovanje autorskih prava i zakonskih regulativa.

Rad je podeljen na dve celine. U prvom delu razmatra se problem modelovanja baze podataka, izrada odgovarajuće web aplikacije i njihovo međusobno povezivanje. Takođe, biće navedeni i objašnjeni osnovni pojmovi koji su korišćeni u radu.

Drugi deo rada odnosi se na sopstvene rezultate. Iznose se činjenice, podaci i rezultati do kojih se došlo istraživanjem i rešavanjem problema.

2 Primena baze podataka u elektronskoj prodaji knjiga

Nastankom marketinga i komercijalne prodaje na Internetu, došli smo do pojave elektronskog poslovanja. Sama pojava elektronskog poslovanja veoma brzo je zaživela u zemljama, koje su imale uređen pravni sistem. Od tada se velika pažnja daje bezbednosti podataka koji moraju biti smešteni u prvom trenutku na nekoj određenoj bazi podataka, a potom se distribuiraju. Vrativši se u prošlost, svaka odštampana knjiga da bi doživela sudbinu da je neko kupi, morala je biti izložena u vitrini ili izlogu nekog dućana. Prodaja knjige bi trajala danima, a ponekad i nedeljama, da bi se saznalo da je ta knjiga uopšte i odštampana i puštena u prodaju. Marketing je bio dosta težak sve do pojave radija ili putem televizije. I onda bi korisnik došao i kupio knjigu, ali šta ako je ta knjiga odštampana i puštena u prodaju u samo jednoj zemlji. Korisniku koji bi trebao da pređe okean i još par zemalja kako bi je kupio. Putovao bi nedeljama a nekada i više, ali pojavom prevoznih sredstava ili aviona put bi ubrzao. To opet ne bi bilo dovoljno brzo, jer i u ovom slučaju bi morao ili da sačeka isporuku ili da izađe iz svog doma i krene na putešestvija. Postoji mogućnost da sačeka da se knjiga pojavi i u njegovom komšiluku, ali reč „sačeka“ objašnjava, da knjigu opet ne može da dobije istog momenta.

Nastankom sajta i baze podataka koja opslužuje sajt za elektronsku prodaju knjiga, prebacivanjem knjige iz fizičkog oblika u elektronski i smeštanjem knjige u bazu podataka, dobija se na ekspektivnosti. Na taj način baze podataka su postale toliko moćne da mogu u sebi da nose informacije o autoru, imenu knjige, ceni, jer knjiga u elektronskoj formi može da se kopira i proda i na hiljade puta, a da pri tom, kada korisnik izabere i plati knjigu može je preuzeti istog trenutka, pročitati je kod kuće ili parku ili poslu ili bilo gde na planeti zemlji samo pod uslovom da ima konekciju ka Internetu. Problem konekcije je prevaziđen u velikom broju slučajeva, sa uvođenjem bežičnog ili satelitskog interneta uz korišćenje određenih uređaja koji to i podržavaju.

Samim tim, sajt za elektronsku prodaju knjiga postaje konkurentan i uz pomoć baze podataka na kojoj se čuvaju svi podaci vezani za prodaju knjiga, spisak svih članova sajta, podaci o karticama kojima se može izvršiti plaćanje, podaci o naručenim ili preuzetim knjigama i podaci o novčanim transakcijama koje su ostvarene na samom sajtu. Isto tako, baza podataka se koristi i za pretragu knjiga kako bi korisniku omogućila da lakše i brže dođe do knjige koju želi da ima u svojoj digitalnoj biblioteci. Uloga baze podataka je velika, što se može videti, da se sa svim ovim podacima može manipulirati, obrađivati, podnositi izveštaj ako to menadžment kompanije traži.

Podaci se mogu lako ažurirati ili brisati po potrebi, a da pri tom se po zahtevu uklone iz svih tabela ili da se ostave ukoliko je to potrebno, a da se cela funkcionalnost sajta i dinamika korišćenja baze podataka ne menja.

Mogući problemi o kojima treba da se vodi računa pre modelovanja i projektovanja baze podataka su:

- 2.1 Arhitektura baze podataka - organizovanost relacionih tabela i podataka u njima, u odnosu na potrebe sajta i zastupljenost podataka koji su predstavljeni
- 2.2 Komunikacija baze podataka i sajta - korišćenje drajvera, SQL upita i logičkog sloja pisanog programskim jezikom C#

Za uspešno postizanje strukture baze podataka i sajta za elektronsku prodaju knjiga korišćeni su:

- ASP.NET
- Visual Studio 2008
- Access database 2007
- C#
- SQL
- HTML
- CSS

2.1 Arhitektura baze podataka

Arhitektura baze podataka je najvažniji deo izrade web aplikacije ili bilo koje aplikacije koja se koristiti za izradu baze podataka. Internet tehnologije služe za olakšavanje deljenja podataka i informacija. Svi podaci su smešteni u bazu podataka. Dobro modelovana baza podataka će omogućiti korisniku pojednostavljenje problema i posmatranje samo sa stanovišta bitnih za ciljeve analize. Na sajtu za prodaju knjiga, glavni cilj je prezentovati knjigu i navesti korisnika da kupi knjigu kako bi se ostvario profit za firmu.

Knjiga kao objekat posmatranja, može imati više osobina ili atributa od kojih u datom trenutku analize može biti dovoljan samo manji broj, kao na primer, autor knjige, ime knjige, opis knjige, slika knjige, cena knjige. Čovek kao jedinka, obdaren sposobnostima apstraktnog načina mišljenja i shvatanja, stvara jedan svet apstraktnog modela realnog sveta. Da bi pristupili modelovanju baze podataka moramo raditi paralelno sa analizom potreba sajta i korisnika koji će pristupati samom sajtu. Model čini osnovu za osmišljavanje, definisanje i implementaciju baze podataka. Kako se informacije prikupljaju, objekti se indentifikuju, dodeljuju im se imena korišćenjem termina bliskim korisnicima. Dijagram arhitekture se može pregledati od strane dizajnera i krajnjeg korisnika da bi se osigurala njegova kompletnost i tačnost. Ako kojim slučajem model nije dobro napravljen on se može modifikovati jer je lakše to uraditi preko dijagrama nego već implementiranu bazu podataka prepravljati. To može zadati velike poteškoće. Zbog toga proces arhitekture i modelovanja baze podataka traje sedamdeset posto vremena, a nekad i više, od celokupnog precesa projektovanja i implementiranja baze podataka. Ciklus pregledanja i modifikovanja treba da traje sve dok se ne dobije potvrda da je model korektan.

Objekti iz realnog vremena se opisuju u računarskoj primeni kao podaci. Zato se kaže da su podaci apstrakcija realnosti. Modelovanje, kao postupak kojim se realni svet prevodi u određeni broj podataka, predstavlja veoma kompleksan posao i sastoji se od nekoliko koraka, kao što su izbor, imenovanje i klasifikacija. Ovako navedeni koraci zovu se konceptualni modeli. Ispitivanjima se došlo do podataka da svaki model podataka ima tri neodvojive komponente, a to su struktura podataka, operacija nad podacima i ograničenja.

Model koji se koristi u izradi baze podataka vezane za sajt elektronske prodaje knjiga je objektno orjentisani model. Bazira se na konceptu objekata koji predstavljaju skup podataka i operacija koje se na njima mogu izvršiti. U objektno orjentisanom modelu podataka svaki objekat se može predstaviti klasom. Klasa obuhvata attribute i ponašanje entiteta. Direktna veza između objekata u web aplikaciji i objekata u bazi podataka daje za rezultat bolje performanse baze podataka i samog sajta. U objektno orjentisanom modelu, informacija je sačuvana kao stalni objekat, a ne kao red u tabeli. Ovo sistem čini efikasnijim kako je gore predhodno navedeno, u smislu prostora potrebnog za čuvanje i smeštanje podataka. To osigurava da korisnik ili administrator uređuju podatke, manipulišu istim, kao i brzinu u dobijanju odgovora na upite, ali samo na onakav način na koji je programer ili administrator baze podataka ili administrator sajta odredio.

Postoje i relacije između objekata koje su korišćenje u izradi baze podataka vezane za web aplikaciju elektronske prodaje knjiga, a one su 1:1, 1:N i N:M. Veze postoje jer objekti nisu međusobno izolovani nego se nalaze u interakciji. U modelu podataka ne postoje samo atributi objekata već postoje i njihove veze. Prvo se selektuju objekti, imenuju se, a zatim se analiziraju tipovi odnosa koji se uspostavljaju između objekata. Odnosi između objekata posmatranja, prikazuju se najčešće primenom logike skupova i preslikavanja elemenata.

Primena podataka ili informacija iz baze podataka na sajt za elektronsku prodaju knjiga, je bitan šraf u celokupnoj izradi same web aplikacije. Veoma je bitno da se podaci, koji se upitima povlače iz baze podataka, budu smešteni na pravo mesto na starnici sajta, kako bi vizuelnim delom omogućili korisniku veću funkcionalnost i operisanje samim sajtom. Najvažnije je da korisnik pravovremeno sazna određene informacije zbog kojih je i postavio upit. Ovaj deo posla je usko stručno vezan za pravila dizajniranja web satranice, jer time moramo voditi računa o nekolicini segmenata, kao što su izgled stranice ali ne samo statički, nego i dinamički jer kada korisnik izvrši pritisak na određeno dugme, šalje se upit u bazu podataka, baza podataka izvršava upit i putem izveštaja obaveštava korisnika. Bitno je, da taj izveštaj bude vidljiv i dizajnirano korektan kako bi korisnik prihvatio sajt kao privlačan, zadržao se dugo ili vratio i po drugi put na njega, ali i ostvario novčanu transakciju, što je i sama ideja elektronskog poslovanja.

2.2 Komunikacija baze podataka i sajta

Komunikacija baze podataka i sajta za elektronsku prodaju knjiga je od velike važnosti jer se sam sajt zasniva na dinamišnosti i efikasnosti, što bez upotrebe baze podataka ne bi bilo moguće. Komunikacija se ostvaruje preko ODBC drajvera koji služe da povežu hardverski server na kome se nalazi sajt i bazu podataka. Izvršavanje povezanosti u softverskom domenu se ostvaruje preko SQL upita koji su upisani u delove koda C# jezika.

Za određene objekte imamo metode koji služe da se izvrši konekcija sa bazom podataka, ili da se konekcija prekine. Uspostavljanjem konekcije se otvara mogućnost za nove zahteve, a samim tim i u logičkom sloju C# programskog jezika, za nove metode vezane za određene upite i izveštaje. Tako da, ako korisnik želi da vidi određeni podatak iz baze podataka, kao na primer informaciju o nekoj knjizi, pritiskom na sliku knjige aktivira prvo konekciju, a zatim izvršava upit u bazi podataka. Baza podataka prosleđuje izveštaj koji posebnim metodom i delom koda se izvršava na samoj web stranici. Tekst se ispisuje na web stranici. Dizajn i font teksta je obrađen u višem sloju web aplikacije uz upotrebu HTML i CSS jezika.

3 Realizacija baze podataka na sajtu E-book store

Za realizaciju baze podataka koristi se Microsoft tehnologija, a za modelovanje i projektovanje baze podataka Microsoft Access iz paketa Office 2007. Prvi korak za realizaciju i povezivanje sajta i baze podataka je izrada Web aplikacije u Microsoft Visual studio 2008 i konekcija driver ODBC koji je već instaliran pri pokretanju servera ASP.NET. Kompletan kod je rađen u C# programskom jeziku i on simulira logički sloj Web aplikacije. U tom kodu su metode za konekciju sa bazom podataka, upitima, izveštajima i prekidima konekcije.

Modelovanje baze podataka je zahtevalo objektno orijentisani model. Objektno orijentisani model je jedan od najnovijih modela baza podataka. Istraživači su za njega postali zainteresovani krajem 70-tih godina i početkom 80-tih godina prošlog veka i tada je zaživeo uz podršku programskih jezika. Bazira se na skupu objekata i operacija koje se na njemu mogu izvršiti. Baš to je omogućilo da se modeluje baza podataka i da u njoj budu smešteni svi objekti koji su potrebni za funkcionalnost samog sajta za elektronsku prodaju knjiga. To zahteva da se pre samog modelovanja baze podataka donese odluka, za jednu vrstu ispitivanja, kako bi se došlo do informacija potrebnih kako sajt uopšte treba da izgleda i bude vidljiv na Internetu. Ispitivanja i istraživanja pre samog početka i pisanja programskog koda su najbitnija, jer je jednostavnije ispraviti greške koje se naprave u procesu modelovanja i projektovanja, nego kada se kod već napiše. To zna da zada velike glavobolje i timu projektanata, a kamoli pojedincu.

U toku ispitivanja je vršeno upoređivanje ideje vezane za prodaju knjiga elektronskim putem i dostavljanje istih. Došlo se do saznanja, da se ideja elektronske prodaje knjiga visoko kotira na Internet tržištu. Taj prvi korak je bitan, jer morate imati saznanja da li je ideja uspešna i materijalno isplativa.

3.1 Proces modelovanja baze podataka

Proces modelovanja baze podataka je veoma složen. Proces iziskuje minimum sedamdeset posto ukupnog vremena, koje je potrebno koristiti u realizaciji i implementaciji web aplikacije. Veoma je bitno da se imaju sve informacije vezane za zahtev klijenta jer dodatno modelovanje baze podataka može predstavljati veliki problem ako je potrebno uklanjati određene relacione tabele.

U procesu izrade baze podataka je korišćen alat Access iz Office paketa verzije 2007. Microsoft je napravio jednostavan koncept funkcionalnosti za povezivanje baze podataka i sajta koji je pisan u C# kodu, koristeći alat Visual Studio 2008.

Prvi problem sa kojim se susrećemo, je da se moraju imati sve informacije o članovima koji će biti registrovani na sajtu. Problem je razrešen tako što je projektovan objektno orjentisani model baze podataka sa tabelom – members (članovi). Prikaz slike 3.1.1 tabele članovi, vizuelno govori o objektu članovi i njegovim metodama, koji su uneti u kolone tabele sa podešenim parametrima.

members	
Field Name	Data Type
member_id	AutoNumber
member_login	Text
member_password	Text
member_level	Number
first_name	Text
last_name	Text
email	Text
phone	Text
address	Text

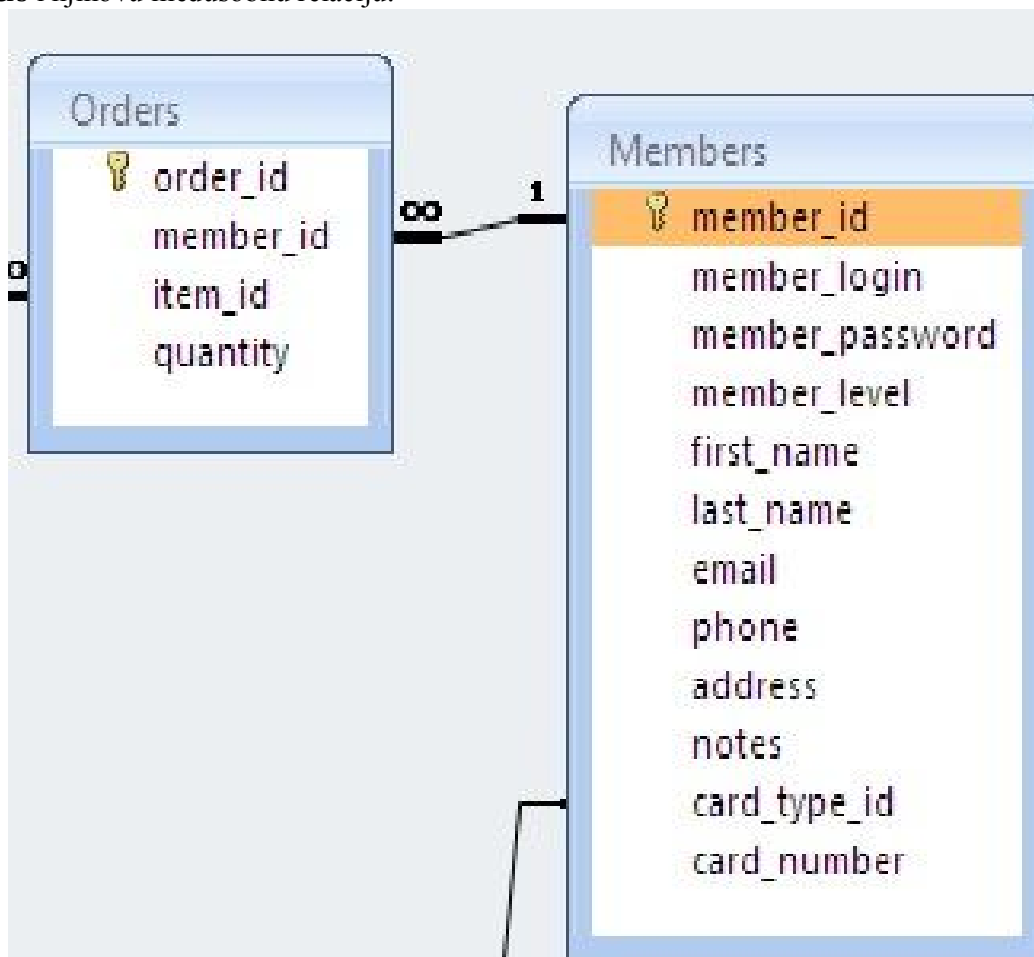
General	Lookup
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

Slika 3.1.1. Tabela members (članovi)

Kao što se vidi iz priloženog, slika3.1.1 tabela members, kao objekat ima nekoliko atributa koji ga opisuju. Atribut objekta members je member_id, koji je podešen kao AutoNumber. To je opis samog atributa, jer će svaki novi uneti korisnik dobiti identifikacioni broj prvi sledeći u nizu i time će se izbeći mogućnost duplikata indetifikujućeg broja korisnika, što je veoma bitno, tako da svaki korisnik će biti smešten u bazu podataka pod određenim i jedinstvenim brojem. Pošto su pomenute reči „jedinstvenim brojem“, automatski taj atribut će poslužiti i dobiti još jednu funkcionalnost, a to je da postane primarni ključ. Ostali atributi će biti upotrebljeni, kako bi svakog korisnika kao objekta ili kao virtuelni podatak opisali sa što više informacija, kako bi potrebne informacije mušterija ostavi prilikom registracije. Time bi imali što bolji uvid o svakom pojedinom korisniku kao potecialnoj mušteriji.

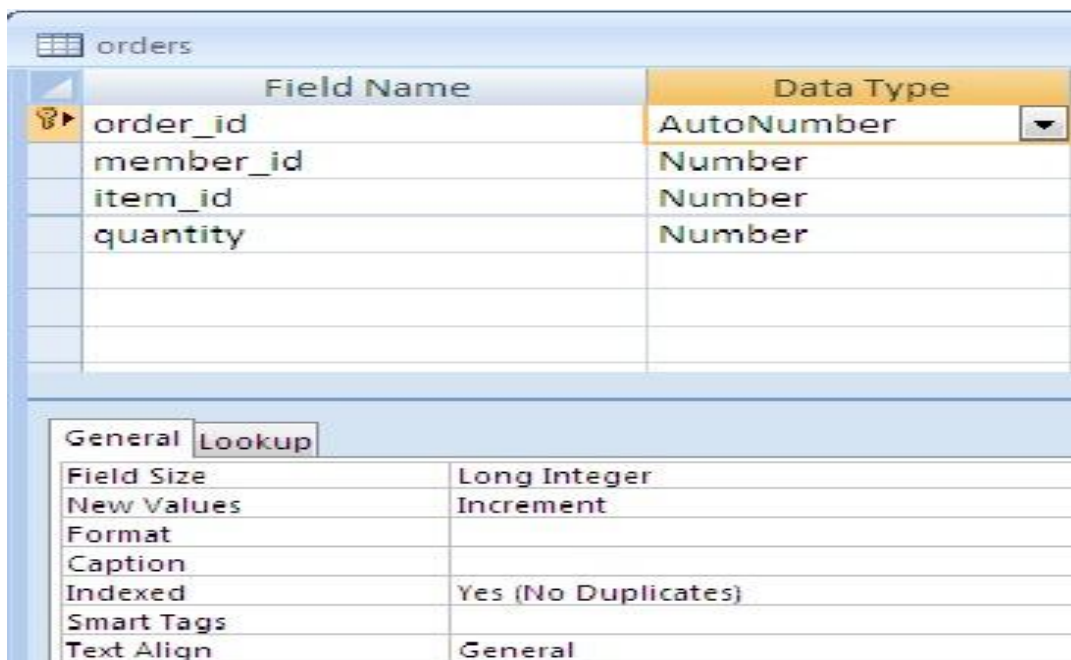
Polje index parametra atributa member_id je podešeno opcijom Yes (No Duplicate), što opisuje, da taj atribut koji opisuje svaki objekat ponaosob, neće moći da se duplira pri unosu i biće jedinstven. To je jedna od predradnji, da bi u relaciji sa drugim relacionim objektno orjentisanim tabelama omogućili da korisnik uvek ostane jedinstven.

Sledeći zahtev je da naši korisnici koji bi se registrovali na sajt elektronske prodaje knjiga, mogu da izvrše i neke porudžbine. Slika3.1.2 koja sledi opisuje tebele members i orders i njihovu međusobnu relaciju.



Slika 3.1.2 Relacioni model veze između objekata Members i Orders
(članovi i porudžbine)

Relacija između tabela members i orders pokazuje da se radi o 1:N relaciji, gde je tabela members glavna tabela, tj. glavni objekat. Time se postiže postavljanje spoljnog ključa u tabelu orders, a to je atribut member_id. To će omogućiti objektu members da poruči više knjiga po potrebi ako to i želi. Kod objekta orders je jasno da ga opisuje nekoliko atributa, od kojih je najvažniji jedinstveni atribut order_id, kojem je podešen parametar Yes (No Duplicat), što ga i čini jedinstvenim i tako simulira primarni ključ.

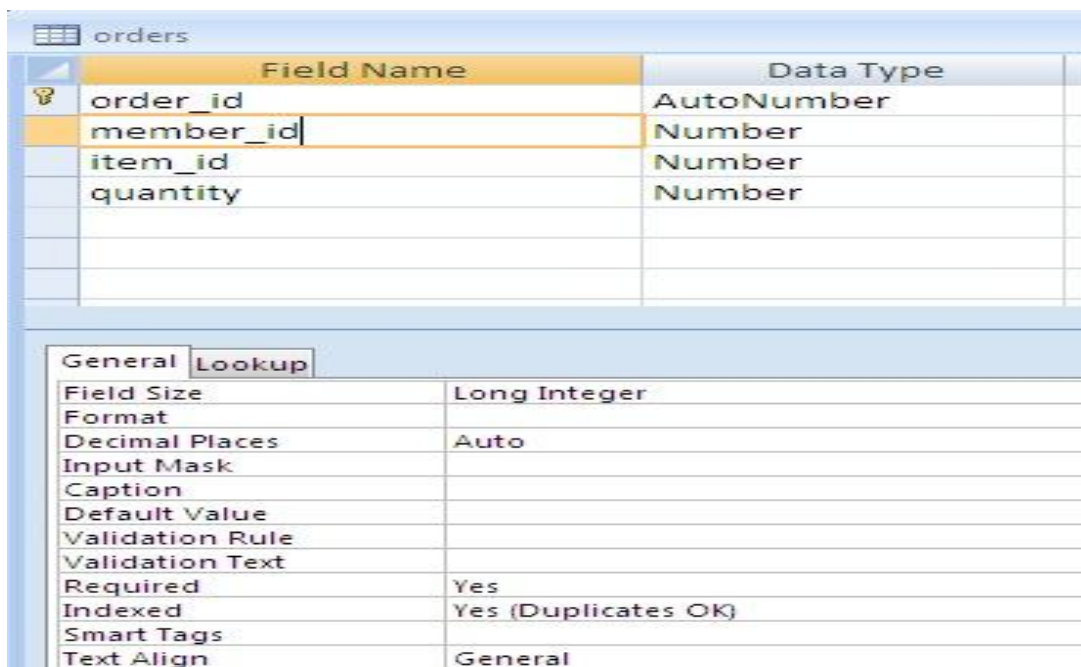


Field Name	Data Type
order_id	AutoNumber
member_id	Number
item_id	Number
quantity	Number

General	
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

Slika3.1.3 Tabela orders (porudžbine)

Ono što je bitno, je podešavanje parametara jednog atributa objekta orders a to je member_id.



Field Name	Data Type
order_id	AutoNumber
member_id	Number
item_id	Number
quantity	Number

General	
Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (Duplicates OK)
Smart Tags	
Text Align	General

Slika3.1.4 Tabela orders i atribut member_id(porudžbine i atribut korisnik id)

Gore napomenuti spoljni ključ `member_id` je atribut koji opisuje objekat `orders` i koji čini vezu sa objektom `members`. Da bi relacija bila korektna, opis atributa `member_id` iz tabele `members` je `AutoNumber`. Kako bi spoljni ključ `member_id` u tabeli `orders` bio funkcionalan, potrebno je da njegov opis bude `Number`, jer će time podržavati upis svakog korisnika kao objekta u tabelu `orders`. Jedinstveni broj iz tabele `members` će biti upisan, zato što opis atributa `member_id` u tabeli `orders` je postavljen da može da vrši upis brojeva. Što se tiče parametara atributa `member_id` u tabeli `orders` bitno je polje `index`, koje daje informaciju da postoji mogućnost duplikata kako bi porudžbina korisnika bila moguća i ako se posle jedne konekcije vrati, ponovi proces, za novu porudžbinu.

Zbog potreba za ubacivanjem objekta `items`, što u prevodu znači stavka ili artikal, napravljena je tabela `items`. Pošto pravimo web aplikaciju koja se bavi prodajom knjiga, moraju se smestiti sve knjige koje se mogu ponuditi korisnicima sajta, tačnije kupcima. Ovde se javio problem sa atributima koji opisuju objekat, jer morao se naći način koji bi bio adekvatan da se knjiga kao artikal opiše sa svim elementima koje korisnik zahteva. Ono što je olakšalo ispitivanje potrebnog opisa objekta je to da se u prvi mah stave opisi vezani za najbitnije informacije, kao naprimer, autor knjige, ime knjige, cena, slika knjige, opis i kako je rangirana knjiga na sajtu od strane korisnika. Time bi se stvorila slika u marketinškom pogledu, da li je knjiga rangirana kao top naslov ili ako nije prihvaćena od strane čitalaca, kako bi bila uklonjena iz reklamne ponude sa prve stranice sajta. Time bi se postigao bolji marketinški funkcionalan dinamičan sajt, prihvatljiv od strane korisnika.

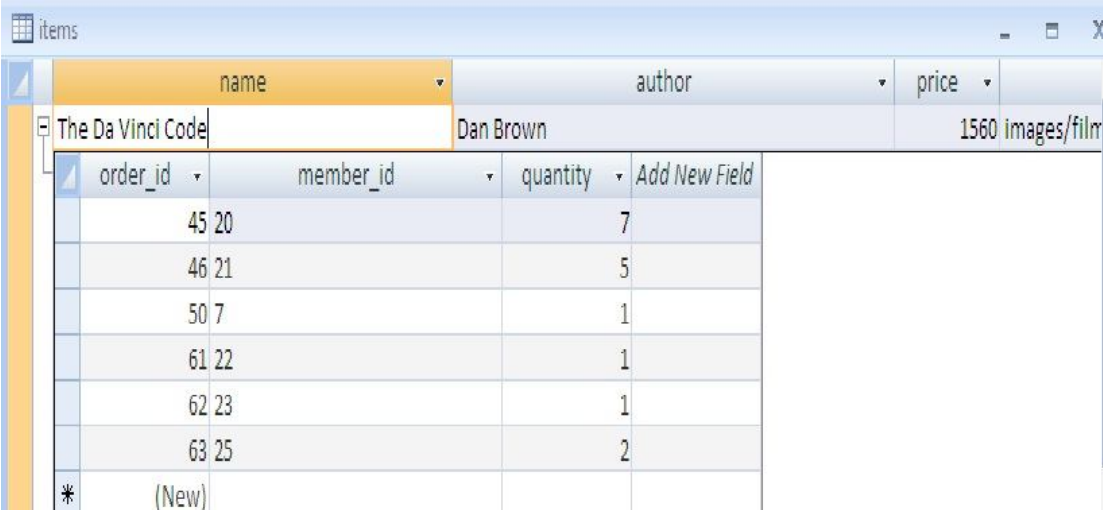
Field Name	Data Type
item_id	AutoNumber
category_id	Number
name	Text
author	Text
price	Number
image_url	Text
notes	Memo
is_recommended	Number
rating	Number

General	Lookup
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

Slika3.1.5 Tabela `items` (stavke ili artikli)

Kao što se vidi iz priloženog, slika 3.1.5, objekat `items` je opisan sa svim potrebnim atributima počevši od `item_id` koji ima funkciju primarnog ključa, a parametrom podešen na `Yes (No Duplicate)` i čini ga jedinstvenim. Ovo je bitno jer ne bi bilo dobro da se pojavi ista knjiga na dva različita mesta u objektu `items` i time napravi problem kako administraciji sajta tako i računovodstvu. To može izazvati ozbiljne finansijske probleme.

Atributi redom opisuju kategoriju određene knjige ili žanra, što je opisano u sledećoj tabeli categories. Ideja je da sajt ima pretragu čime bi se olakšao upit iz logičkog sloja programskog jezika C# i tako određenom žanru ili kategoriji dodelio identifikacioni broj. Atribut name i author imaju funkcionalnost opisa knjiga. Objekat items u ovom slučaju daje informaciju ko je knjigu napisao i koje ime nosi knjiga. Isto važi i za atribut price u kojem administrator ostavlja cenu knjige koju su mu prosledili iz menadžmenta firme. Menadžmentu će biti omogućeno, da jednostavnim putem izvršavaju korekcije cena kao bi rasteretili administratora od dodatnog posla, pristupom na sajt uz korišćenje određenog korisničkog imena i lozinke. Notes ima ulogu da kao kontejner tekstualnog fajla omogući unos teksta i opisan je kao atribut sa memo opcijom. Uloga mu je da sadržaj knjige u kratkim crtama prenese kao informaciju korisnicima na sajt i time privuče pažnju ili omogući im lakši odabir knjige. Atribut image_url je opisan kao tekst polje, koje služi da prihvati tekstualnu informaciju u bazu podataka. To u prvom trenutku i radi, ali u samom logičkom delu koda C# programskog jezika, postoji mogućnost da informaciju koja se upitom povlači iz baze podataka i koja je tekstualnog tipa, prosledi do izvornih fajlova – slika ili images-a, koji se nalaze u određenom podfolderu. Podfolder je deo foldera web aplikacije. Time će se na sajtu predstaviti slike, naslovne strane ili korice neke knjige koje su smeštene u podfolderu Images.

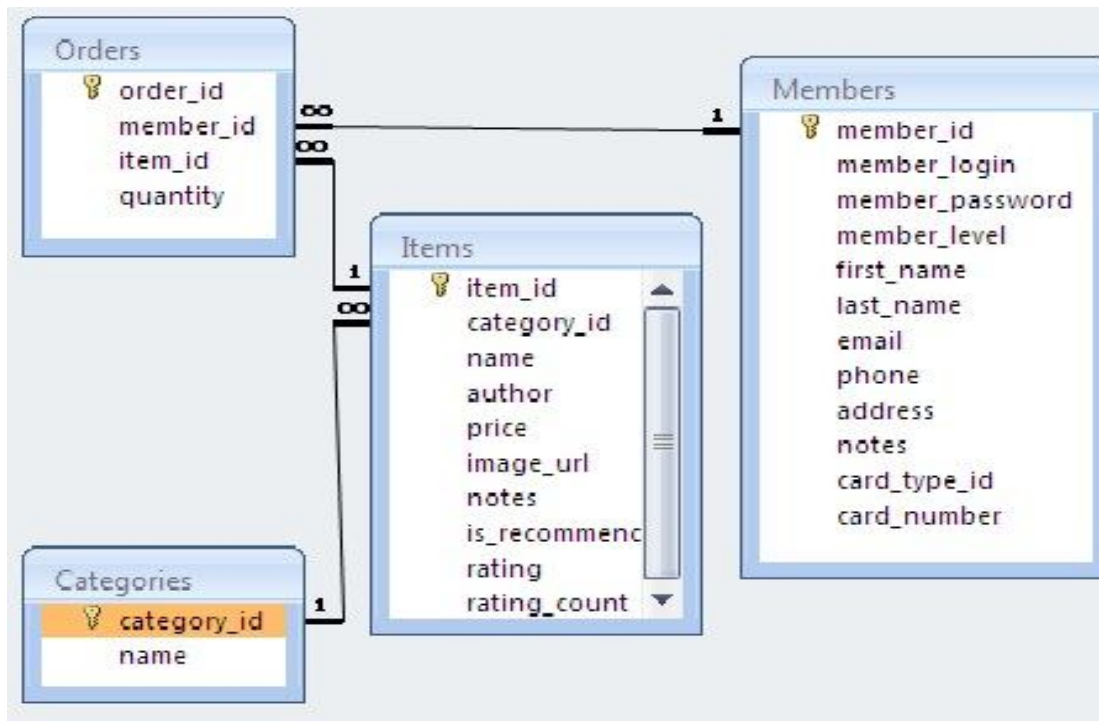


name	author	price	order_id	member_id	quantity
The Da Vinci Code	Dan Brown	1560			
			45	20	7
			46	21	5
			50	7	1
			61	22	1
			62	23	1
			63	25	2

Slika3.1.6 Tabela items (stavke ili artikli)

Slika 3.1.6 predstavlja pogled na objekat items i na attribute koji ga opisuju kao i jedan unos imena, autora, cene i putanje do odgovarajućeg izvornog fajla slike. Vidi se da je uspostavljena relacija i funkcionalnost korektna. Sa slike 3.1.6 se vide identifikujući brojevi porudžbina, identifikujući brojevi korisnika i količina knjiga koja je poručena.

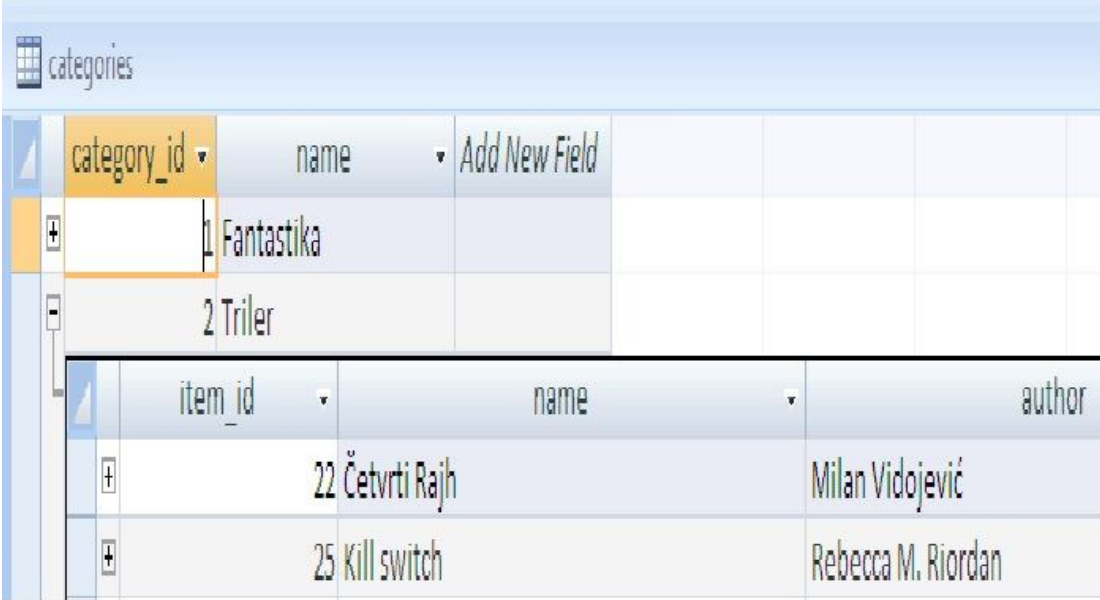
Da bi korisnik imao lakšu mogućnost pretrage modelovan je objekat categories koji je opisan sa atributima category_id i name.



Slika 3.1.7 Relacioni model objekta categories (kategorije ili žanrovi) i relacija sa objektima Orders, Items, Members

Ostvarena je mogućnost da se određene knjige vežu za neku od kategorija i time se olakša pretraga knjiga, kako korisnik ne bi morao da kuca tačan naslov knjige u polju za pretragu, već mu je omogućeno da izabere odgovarajuću kategoriju ili žanr čime bi olakšao brži pristup do željene knjige.

Relacijom je omogućeno da tabela categories bude glavna u odnosu na tabelu items i time se dobija mogućnost da objekat categories bude vezan za više različitih knjiga što je i cilj.



categories	
category_id	name
1	Fantastika
2	Triler

item_id	name	author
22	Četvrti Rajh	Milan Vidojević
25	Kill switch	Rebecca M. Riordan

Slika 3.1.8 Tabela categories (kategorije ili žanrovi)

Za opis atributa `category_id` uz prisustvo relacije i korektne funkcionalnosti, vezale su se informacije od objekta `items`, kao što je prikazano na slici 3.1.8. Klikom na dugme „plus“ koje prikazuje „stablo“, vidimo da je broj 2 koji je u opisu atributa `category_id` označen kao jedinstven za određenu kategoriju. Opis atributa `name` je podešen kao triler i time pridodat jedinstvenom broju 2 kao kategoriji. Pošto je relacija korektna i vezuje objekat `items` u kome su smeštene informacije o knjigama, dolazi se do veze, kao i što sama reč „relacija“ znači, da postoji veza između objekata `categories` i `items`. Na taj način, pridodati su atributima `category_id` i `name` svi atributi objekta `items` i informacije koje nose sa sobom, a to su ime knjige, autor, cena, kratak opis knjige, putanja do slike koja će biti predstavljena na stranici sajta kao slika, koja je smeštena u podfolderu `Images`.

Sada funkcionalnost baze podataka postaje veća, jer korisnik pored registrovanja na bazu podataka, može da ima uvid u izbor knjiga i da vrši porudžbinu knjiga koje bi želeo da kupi i ima usvojoj digitalnoj biblioteci.

Pošto je zahtev da sajt može da izvršava i novčane transakcije i omogući korisnicima elektronsko plaćanje određenim karticama, od kuće iz njihovih fotelja, baza podataka mora da se modeluje tako da sa još par objekata koji bi u sebi nosili određene informacije, provere i memorisanja novčanih transakcija, omogući menadžmentu firme jasniji uvid u novčane transakcije na jednostavan način uz korišćenje određenih upita, kao i uvid u izveštaje iz baze podataka.

Pravljenjem objekta `processor` koji ima za funkciju da smešta u sebe informacije o svim procesima na sajtu, misleći na novčane transakcije, dobija se mogućnost da u bazu podataka smestimo sve zapise vezane za novčane transakcije.

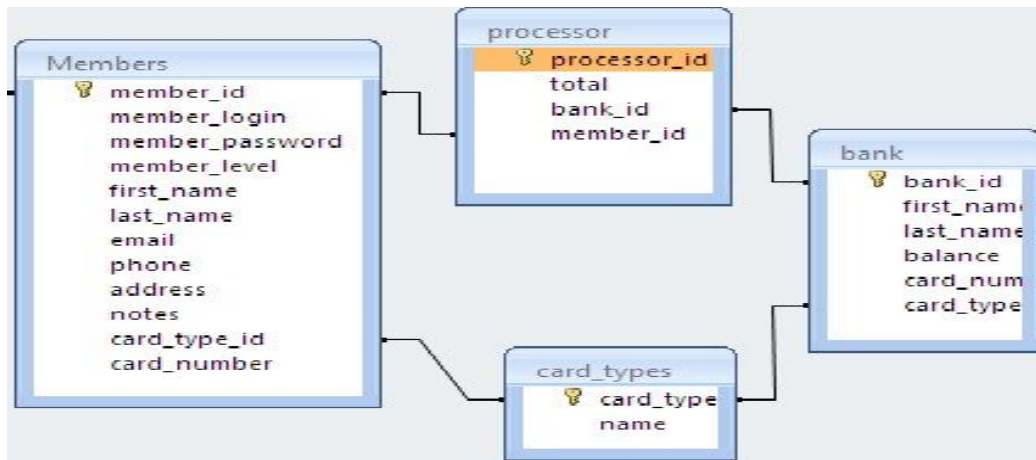
processor	
Field Name	Data Type
processor_id	AutoNumber
total	Currency
bank_id	Number
member_id	Number
General Lookup	
Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

Slika 3.1.9 Tabela processor

Opisni atribut `processor_id` objekta `processor`, predstavljen na slici 3.1.9, ima ulogu jedinstvenog primarnog ključa i opisan je parametrom kao `AutoNumber`, što ga čini jedinstvenim. Ova tabela, tačnije objekat `processor`, je vezana za banku i kreditni biro. Banka vrši proveru stanja kao i otvaranje novih računa za korisnike. Kreditni biro ima ulogu supervizora pri svakoj transakciji jer proverava validnost kreditnih i debitnih kartica. Zbog veze sa bankom i kreditnim biroom u tabeli `processor` se pojavljuju još dva atributa koji opisuju objekat, a to su, `bank_id` i `member_id`.

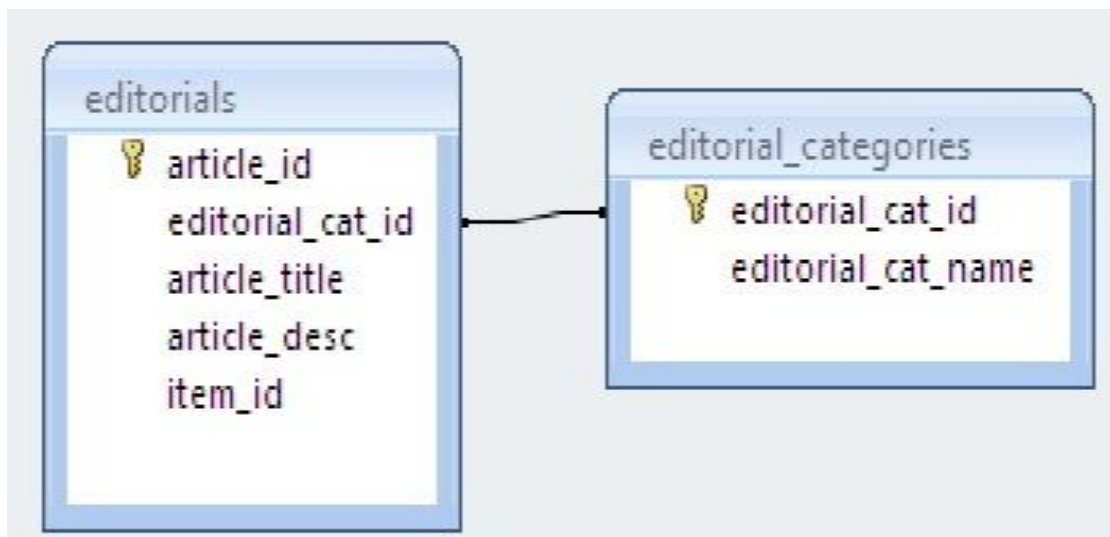
Atribut `bank_id` je spoljni ključ primarnog ključa tabele `bank` i time omogućava da se izvršene transakcije memorišu u objekat `processor`.

Atribut `member_id` omogućava da svaku novčanu transakciju veže za određenog korisnika, tačnije člana baze podataka `knjige.mdb`, što je i predstavljeno na slici 3.1.10.



Slika 3.1.10 Relacioni model baze sa bankom i kreditnim biroom

Cela baza podataka je modelovana tako što su joj dodate još dve tabele koje nisu sa njom fizički vezane. Njihova funkcionalnost je osmišljena tako da u sebi nose informacije o novitetima i određenim knjigama koje će se naći na popustu ili akciji. Time će privući korisnika na povoljniju ali dobru kupovinu.



Slika 3.1.11 Tabele editorials i editorials_categories

Objekat `editorial_categories`, koji je predstavljen na slici 3.1.11, ima funkciju da napravi kategorije akcijske prodajne ponude na sajtu, tako da se korisnicima da do znanja da postoje prodajne akcije i snižene cene. Ovaj deo je bitan za marketing prodaju samog menadžmenta sajta. Atribut `editorial_cat_id` je primarni ključ, koji omogućava da se za određenu kategoriju ponude veže odgovarajući tekst. Tekst može biti, na primer, misleći na aktuelne teme, na nove naslove, nedeljnu ponudu i slično. Objekat `editorials` ima funkcionalnost da prihvati unos marketing službe sajta, za sve informacije koje ga opisuju, naravno uz korišćenje atributa. Primarni ključ je `article_id`, kojem je uloga da svaki artikal bude jedinstven, da ne bi došlo do dupliranja istog na dva ili više mesta u bazi podataka, što bi moglo da izazove probleme, misleći ako korisnik pokrene upit za neki naslov knjige koja je na akcijskoj prodaji, izveštaj iz baze podataka mogao bi da bude dupliran i da napravi konfuzno stanje kako kod korisnika tako i kod marketig službe firme.

Atribut `editorial_cat_id` ima funkciju spoljnog ključa i pravi relaciju između objekata. Atributi `artical_title` i `artical_desc` imaju funkciju da prime tekstualne informacije, koje će biti prezentovane na sajt u kolumni „Šta se čita“ i „Novi naslovi“. Ove informacije će korisnik moći da vidi. Informacije se mogu vrlo lako menjati i njima se lako može manipulirati jer se administriranje obavlja ili iz baze podataka ili sa sajta odgovarajućim korisničkim nalogom.

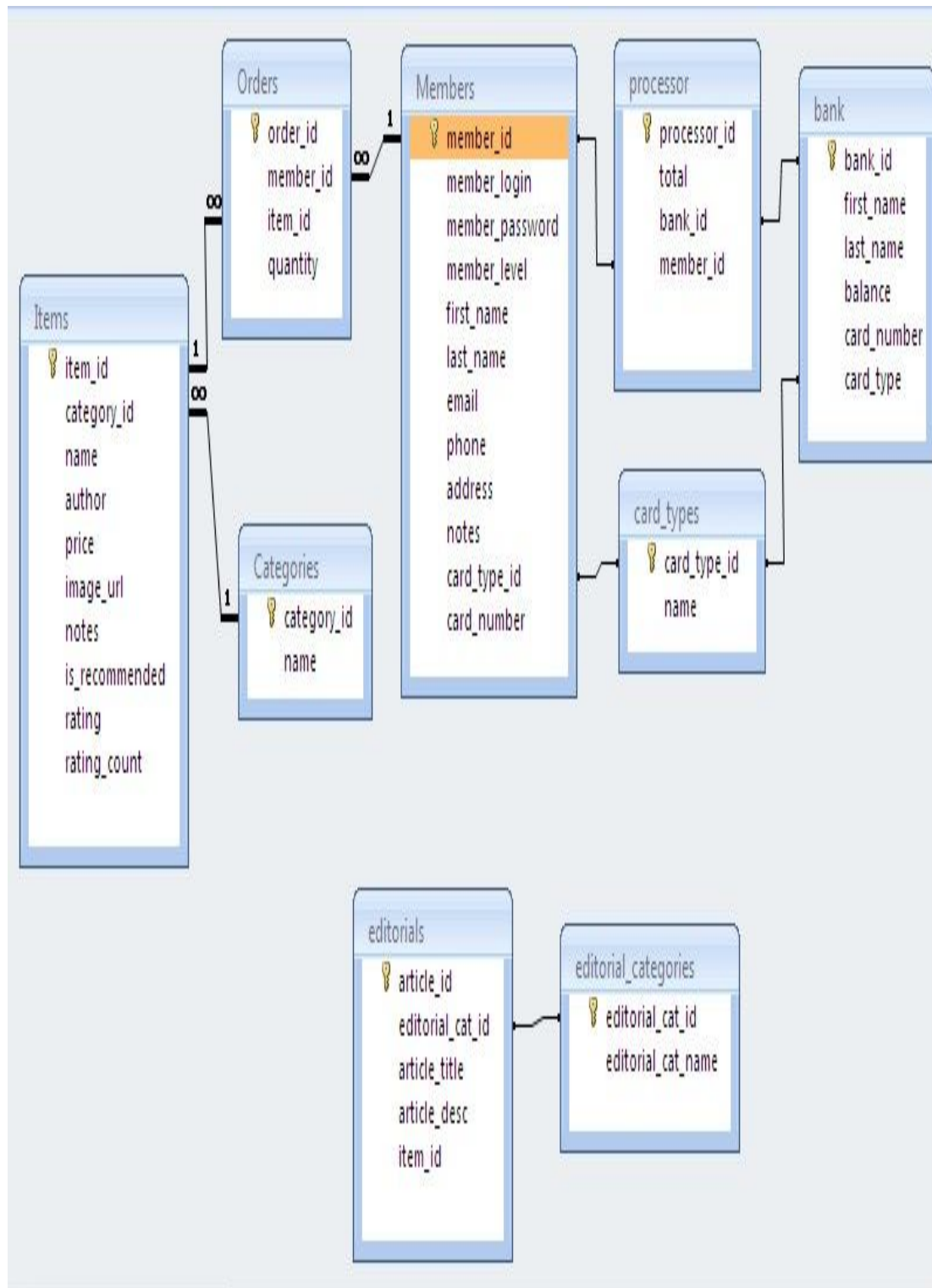
Jednostavan unos podataka i funkcionalnost u tabele `editorial_categories` i `editorials` omogućavaju da se marketing služba lako nosi sa svim problemima vezanim za povećanje prodaje knjiga i omogući korisnicima povoljnije cene u odnosu na druge konkurentne sajtove.

editorial_cat_id	editorial_cat_name	article_id	article_title	article_desc	item_id
1	Šta se čita				
			Paulo Koeljo - Zahir	Zahir je očaravajuća priča o putovanju, opsednutosti i	
		*	(New)		
2	Novo				
3	Nedeljno				
4	Opšta				

Slika 3.1.12 Tabela `editorial_categories`

Veza između objekata `editorials` i `editorial_categories` je funkcionalna i korektna, prikazana je na slici 3.1.12. Ona pokazuje, na koji način je vezan atribut `editorial_cat_id` sa atributima objekta `editorials`, koji u sebi nose opisne informacije.

Modelovanje baze podataka kao što se može videti iz priloženog je kompleksan proces. Kada se uspostave relacije između objekata i kada se izvrše unosi u polja njihovih atributa, postaje veoma teško, a nekada skoro i nemoguće, izvršiti promene kompletne infrastrukture. U nekim slučajevima, moguće je ostvariti ubacivanje novih opisnih atributa kako bi se povećao broj informacija potrebnih za određeni objekat.



Slika 3.1.13 Relacioni model objekto orjentisane baze podataka knjige.mdb

Za implementaciju baze podataka knjige.mdb i realizaciju korišćen je alat Access database iz Microsoft Office 2007 paketa. Slike koje su predstavljene, su urađene uz pomoć printscreen-a iz Access database 2007 paketa, korišćenjem fajla knjige.mdb.

Implementacija baze podataka na sajt je proces koji zahteva hardversku vezu između baze podataka i servera na kome će biti postavljena web aplikacija koja se pokreće uz pomoć servera ASP.NET.

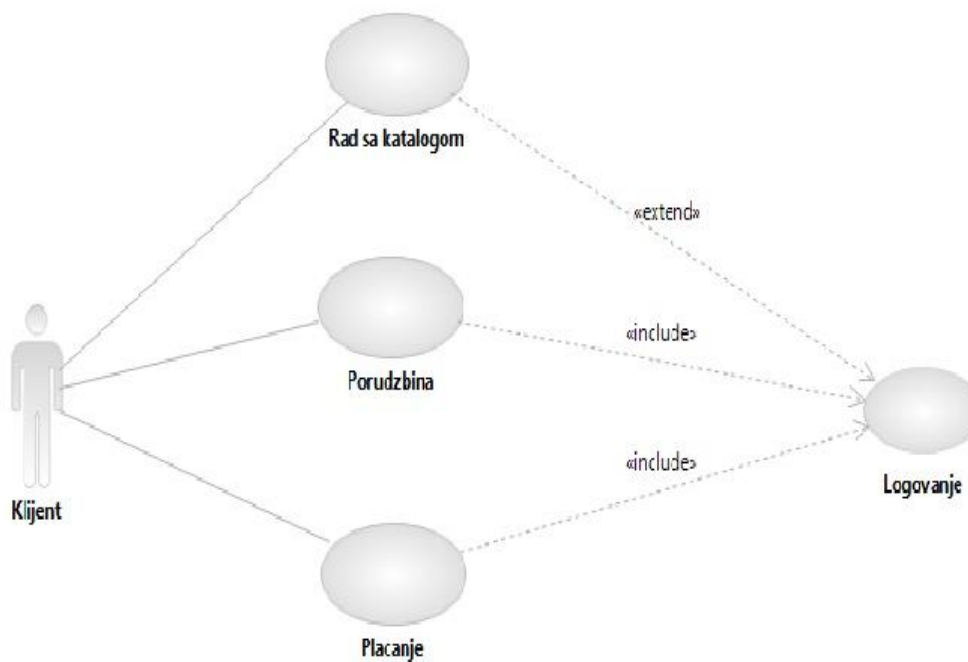
Web aplikacija se implementira i postavlja u rad preko ODBC driver-a. ODBC driver se može koristiti za Microsoft proizvode i instaliran je uz sistem Windows7, kao podrška. Postoje još i OLE DB driver-i, koji su implementirani u Microsoft pakete, ali ono nisu korišćeni za konekciju u ovom diplomskom radu i imaju istu ulogu kao i ODBC driver-i, ali se malo razlikuju u funkcionalnosti.

3.2 Proces konekcije baze podataka i sajta E-book store

Kako bi sajt za elektronsku prodaju knjiga zaživeo i bio implementiran u globalnu mrežu, mora se pristupiti celokupnom modelovanju procesa pravljenja sajta i pisanja programskog koda. Ovaj proces se sastoji od nekolicine stavki, ali najvažnija stavka kojom se bavi diplomski rad je konekcija između baze podataka i web servera na kome će se pokretati web aplikacija.

Za povezivanje potrebni su:

- Driver ODBC
- Metodi ili funkcije u C# programskom jeziku
- SQL – jezik upita
- Baza podataka knjige.mdb
- ASP.NET



Slika 3.2.1 Use case dijagram

Kada klijent otvori početnu stranu, tada se uspostavlja konekcija web aplikacije i servera ali pošto je sama aplikacija podržana od strane baze podataka, a većina podataka nalazi se u njoj, tada se uspostavlja i konekcija između baze podataka i servera preko ODBC driver-a.

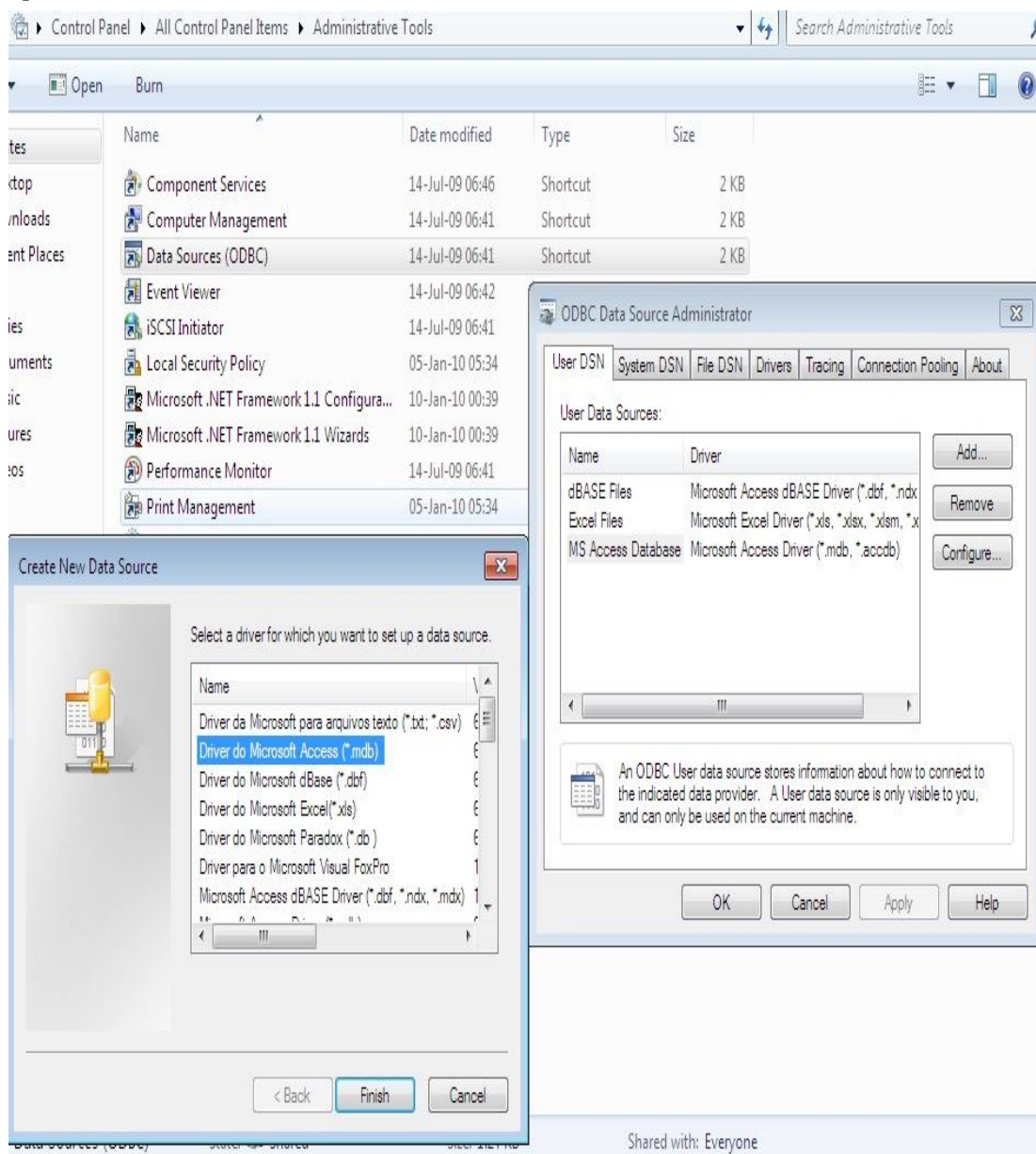
ODBC driver je standardni interfejs za pristup SQL bazama podataka koji je definisala kompanija Microsoft. Definisan je u Windows7 operativnom sistemu korišćenjem ODBC data Source Administrator alata. Pristup ovom alatu se sprovodi kroz Control Panel prozor u Windows operativnom sistemu.

Administrator dodaje ODBC driver-e konfigurišući drajvere da rade sa određenom datotekom specifične baze podataka. U diplomskom radu povezuju se baza podataka knjige.mdb sa ODBC driver-om.

Takvim povezivanjem se omogućavaju izvršenja SQL naredbi, koje preko upita, jer je SQL jezik upita, omogućavaju korisniku da bude dinamički uslužen od strane web aplikacije E-book store koja je povezana sa bazom podataka knjige.mdb.

SQL (Structured Query Language) je standardni relacioni upitni jezik (ANSI – American National Standards Institute - standard). Služi za kreiranje, organizaciju i manipulaciju podacima u relacionim bazama podataka. Danas je SQL ugrađen u sve vodeće SUBP SQL i uspešno primenjen u sistemima za upravljanje bazom podataka kao što su MS Access. U terminologiji SQL-a umesto pojma relacije koristi se pojam tabele.

Proces povezivanja baze podataka knjige.mdb i ODBC driver-a je jednostavan na operativnom sistemu Microsoft Windows7.

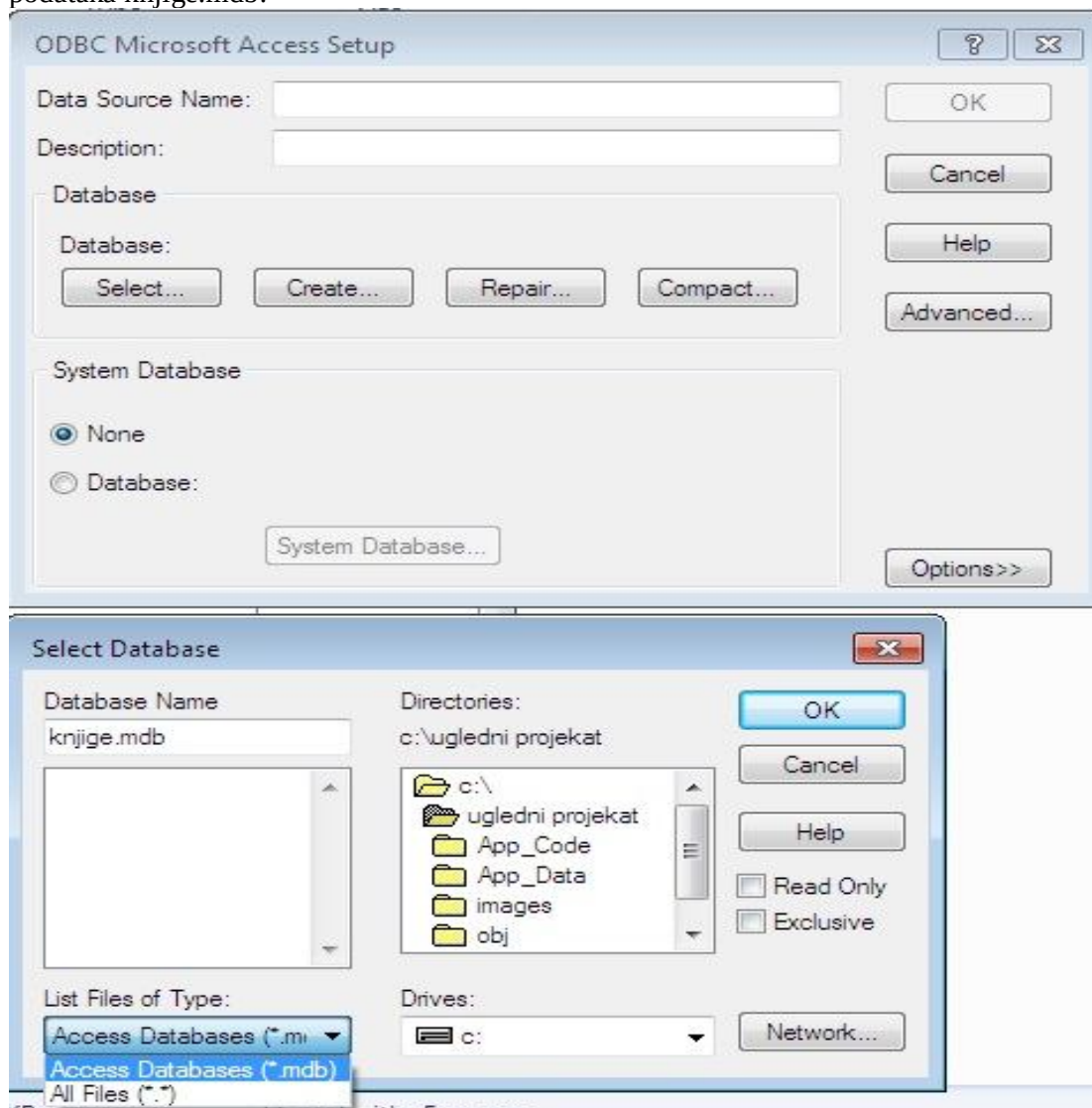


Slika 3.2.2 Pokretanje ODBC driver-a

Kada se otvori Control Panel bira se opcija Administrative tools. Ovde biramo opciju Data Sources (ODBC) kao što se može videti i sa slike 3.2.2.

Otvoriće se novi prozor u kome se bira opcija MS Access Database i kao što se sa slike 3.2.2 može uočiti da su u ovoj opciji podržane ekstenzije .mdb. Takva ekstenzija je i potrebna za izvršenje projekta jer je baza podataka knjige.mdb u istoj ekstenziji. Time je osigurano da će ODBC driver korekto izvršiti povezivanje baze podataka knjige.mdb i izvršenje SQL naredbi.

Pritiskom na dugme Finish otvara se prozor u kome treba navesti putanju do baze podataka knjige.mdb.



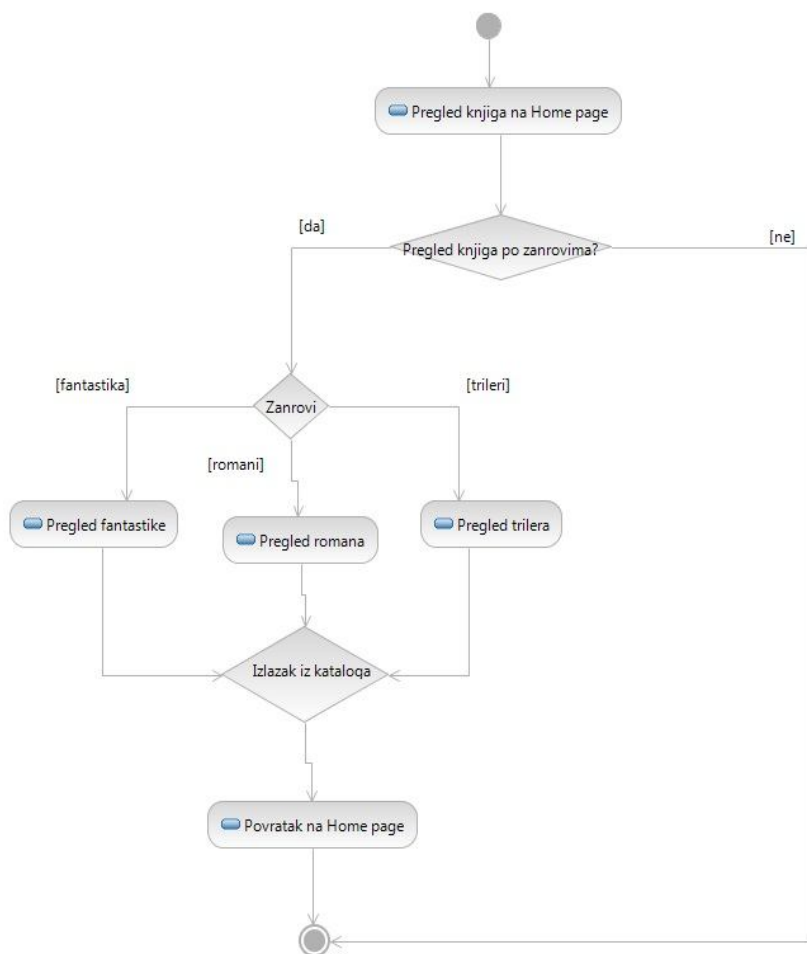
Slika 3.2.3 Projektovanje ODBC driver-a

Kao što se može videti sa slike 3.2.3, izborom opcije Select otvara se novi prozor Select Database. Prvo je potrebno izabrati na kom delu ili particiji se nalazi projekat. Pošto se projekat nalazi na c particiji hard disc-a, jednostavnim odabirom particije i samog foldera u kome se nalazi baza podataka knjige.mdb i odabirom ekstenzije .mdb dolazimo do baze podataka knjige.mdb koju koristimo za izradu sajta elektronse prodaje knjiga.

Kada je driver ODBC podešen i korektno setovan, trebalo je napraviti projekat sajta za prodaju knjiga. Pošto se diplomski rad ne bavi projektovanjem sajta kroz pisanje programskog koda u jeziku C#, ali web aplikacija je morala biti napravljena, kako bi gotov projekat u fizičkom obliku mogao biti predstavljen, a da je funkcionalno korektno urađena aplikacija, predstavljen je deo procesa UML modelovanja koji je predhodio izradi sajta.

Slika 3.2.1, s početka poglavlja predstavlja Use case dijagram, koji govori o tome koje funkcionalnosti korisnik može da koristi na sajtu a da pri tom mora da bude ulogovan ili prijavljen, tj. prvo registrovan a potom dobija mogućnost da se uloguje ili prijavi na određene stranice.

Kako bi se videle određene aktivnosti koje određene stranice trebaju da pruže korisniku – predstavljena su tri dijagrama aktivnosti.



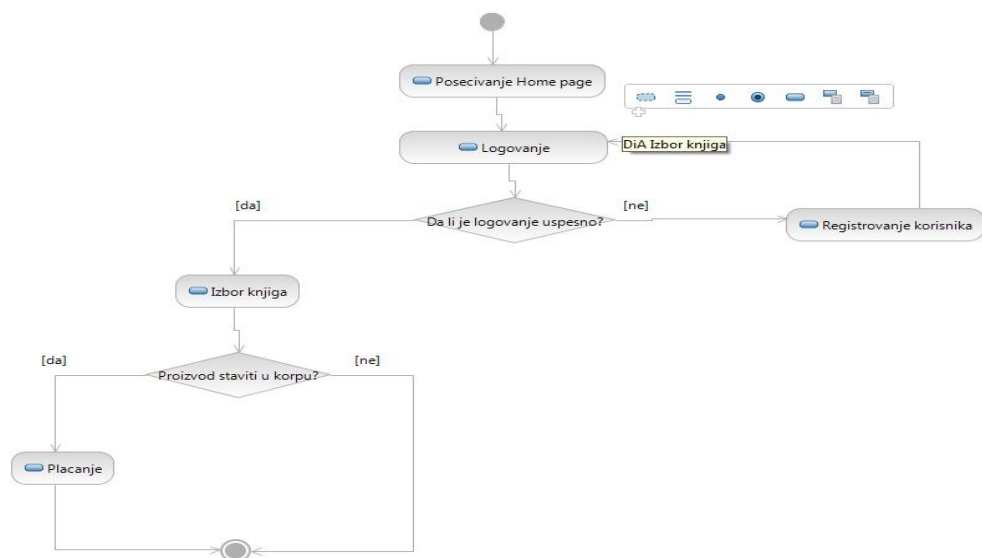
Slika 3.2.4 Dijagram aktivnosti pregleda knjiga na Home Page

Ukratko, korisnik može imati izbor kataložskog pregleda knjiga. Ako bi odabrao ne, ostao bi na Home Page, početnoj stranici. Ali ako bi korisnik odabrao izbor knjiga, imao bi opciju da izborom žanrova suzi selekciju i jednostavnije nađe određenu knjigu. Ovaj korak je bitan jer ima veze sa projektovanjem baze podataka knjige.mdb koja je objašnjena u poglavlju 3.2. Pristupajući radi, objekat categories iz baze podataka knjige.mdb, upućuje na to da se vidi da će se na sajtu pojaviti baš te kategorije knjiga iz baze podataka.

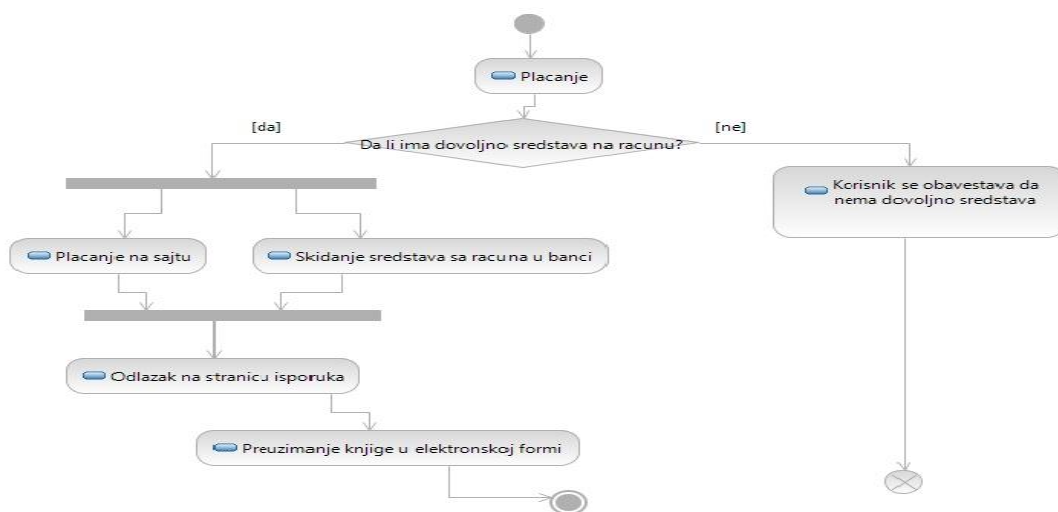
Time se postiže jedna funkcionalnost i dokaz da je dobro modelovana objektno orijentisana baza podataka veoma bitan šraf u celokupnom procesu i radu web aplikacije.

U nastavku aktivnosti vidi se da korisnik kada završi pregled kataloga automatski se vraća na Home Page, početnu stranicu. Ovu aktivnost mogu da obave svi korisnici bilo da su registrovani ili ne.

Sledeća dva dijagrama aktivnosti, opisuju situaciju i mogućnost akcije korisnika kao aktivnosti, ako je registrovan. Opcijom prijavljivanja, omogućeno je da korisnik napravi šoping korpu i da, kada se odluči, plati knjige koje je odabrao.



Slika 3.2.5 Dijagram aktivnosti logovanje i korpa



Slika 3.2.6 Dijagram aktivnosti plaćanje

Što se tiče dijagrama aktivnosti sa slika 3.2.5 i 3.2.6 očigledna je veza baze podataka knjige.mdb i sajta za prodaju knjiga, jer aktivnost logovanja je povezana sa objektima baze podataka members i preko SQL jezika se vrši upit u bazu podataka da bi se dobile informacije da li je korisnik registrovan ili ne.

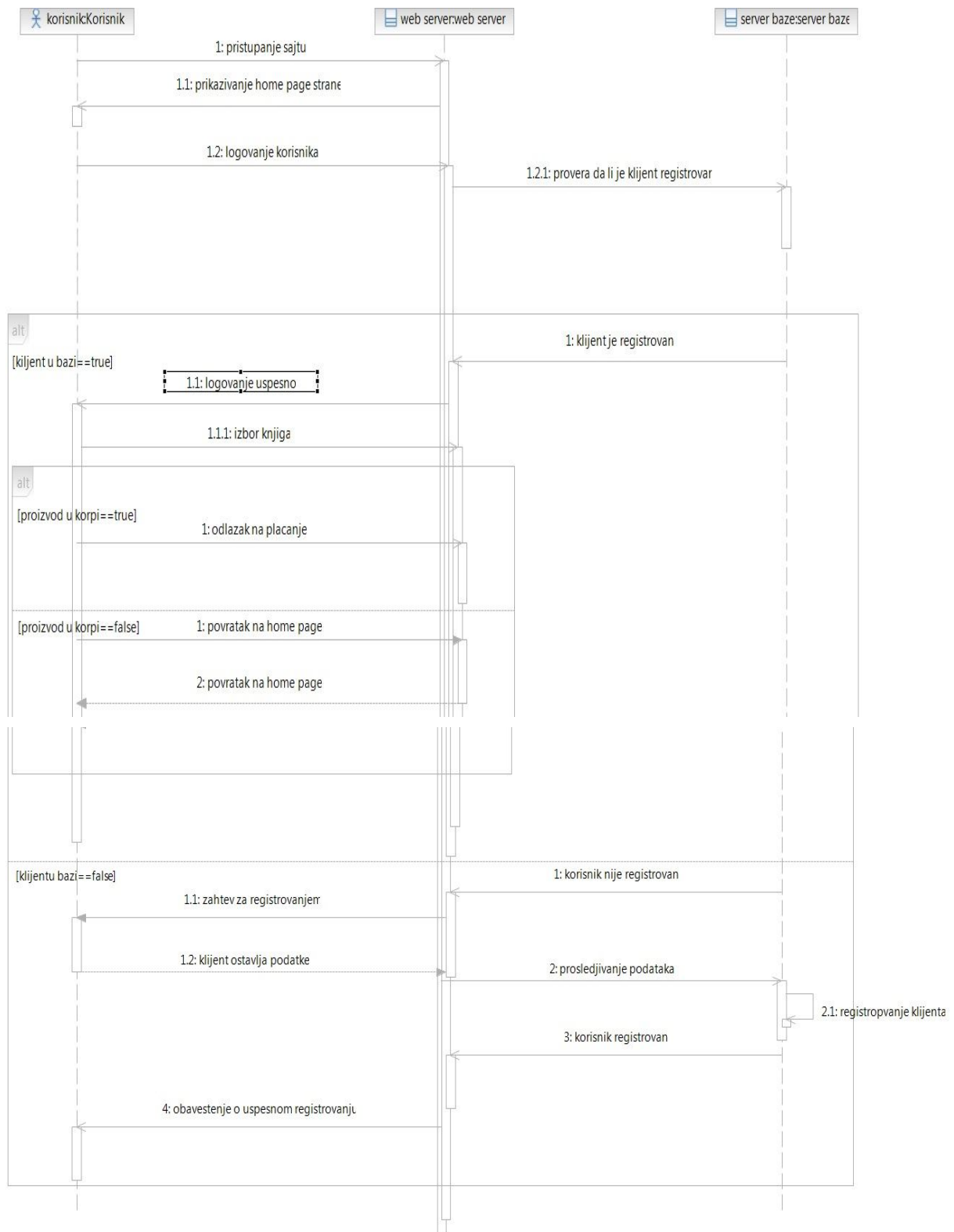
Jer ako korisnik nije u objektu members u objektu orjentisanoj bazi podatak knjige.mdb, mora da izvrši registraciju popunivši sva polja na stranici Registracija.aspx, a to su ustvari atributi koji opisuju objekat members i koji se prosleđuju bazi podataka. Svako polje se popunjava i to po principu kako je atribut opisan, tačnije rečeno, da li je polje tekstualnog, brojevnog ili datumskog tipa.

Dijagram aktivnosti koji opisuje aktivnost korisnika koji je napravio korpu i sada želi da plati, kako bi došao do knjige koju je odabrao u katalogu. Veza sa bazom podataka knjige.mdb je u tome što se sada koriste relacije, tačnije veze između objekata members i card_type koji imaju funkciju da provere da li korisnik koji je ulogovan, plaća kreditnom ili debitnom karticom i da li je pritom takva kartica i vrsta novčane transakcije, uopšte podržana od firme, odnosno da li je firma uspostavila ugovor sa platnim prometom te vrste kartica.

Druga veza se odnosi na objekte u bazi podataka, a to su, members i bank. Banka vrši proveru da li korisnik ima uopšte otvoren račun i da li ima dovoljno novčanih sredstava da izvrši novčanu transakciju.

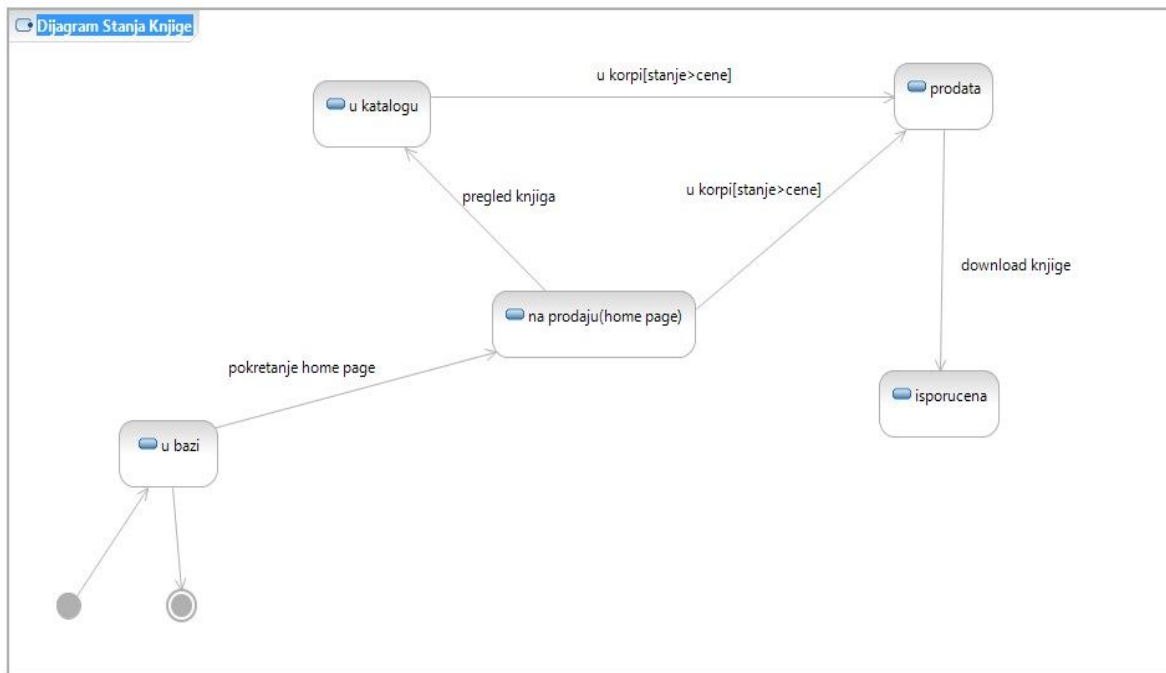
Ako je korisnik uspešno obavio novčanu transakciju i banka mu je skinula sredstva sa računa, aktivira se objekat processor u bazi podataka knjige.mdb koji u sebi beleži novčanu transakciju i korisnika koji je u njoj učestvovao. Time se vidi da je prisustvo baze podataka knjige.mdb koja je objektno orjentisano modelovana, bitno i obezbeđuje funkcionalnost i dinamičnost. Ostaje da se modeluje kompletan sajt za prodaju knjiga, kako bi se upiti korisnika i vizuelni prikaz odgovora, podstaknut dizajnom, izvršavao u realnom vremenu.

U procesu UML modelovanja sajta E-book store, korišćeni su dijagram sekvenci, dijagram komunikacije, dijagram stanja knjige, dijagram klasa, dijagram komponenti i dijagram uvođenja ili hardverske implementacije.



Slika 3.2.7 Dijagram sekvenci

Dijagram sekvenci, prikazan na slici 3.2.7, sastoji se od uspravnih linija (Lifelines) koje predstavljaju objekte, dok linije između, predstavljaju poruke - radnje koje se odvijaju između objekata. Radnje su poredane hronološki. Dijagram sekvenci prikazuje objekte koji učestvuju u realizaciji određene radnje i akcije koje se dešavaju. Obrađen je dijagram sekvenci koji se odnosi na izbor knjiga. Pošto se sajt E-book store bavi prodajom knjiga, uzet je kao glavni objekat knjiga ili kako je u bazi podataka knjige.mdb objekat nazvan kao items.



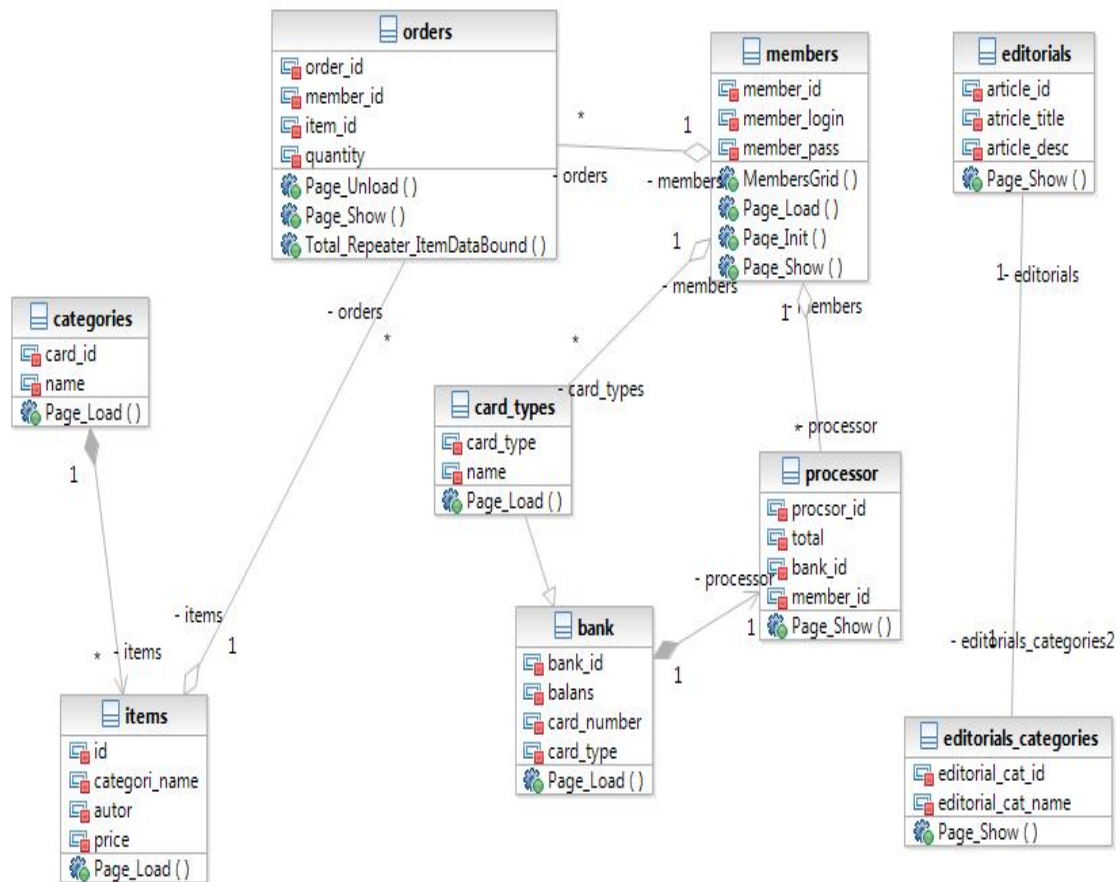
Slika 3.2.8 Dijagram stanja knjige

Kod dijagrama stanja koji je predstavljen na slici 3.2.8, jasno se vidi kroz koja sve stanja prolazi objekat items – knjiga. Linije koje ih povezuju predstavljaju metode.

Na prvi pogled deluje apstraktno, kada se pomisli u kojim sve stanjima može da se nađe knjiga, ali ako se posmatra, kako je ranije navedeno, kao objekat, jasno je da može menjati stanja u zavisnosti od potrebe same aplikacije.

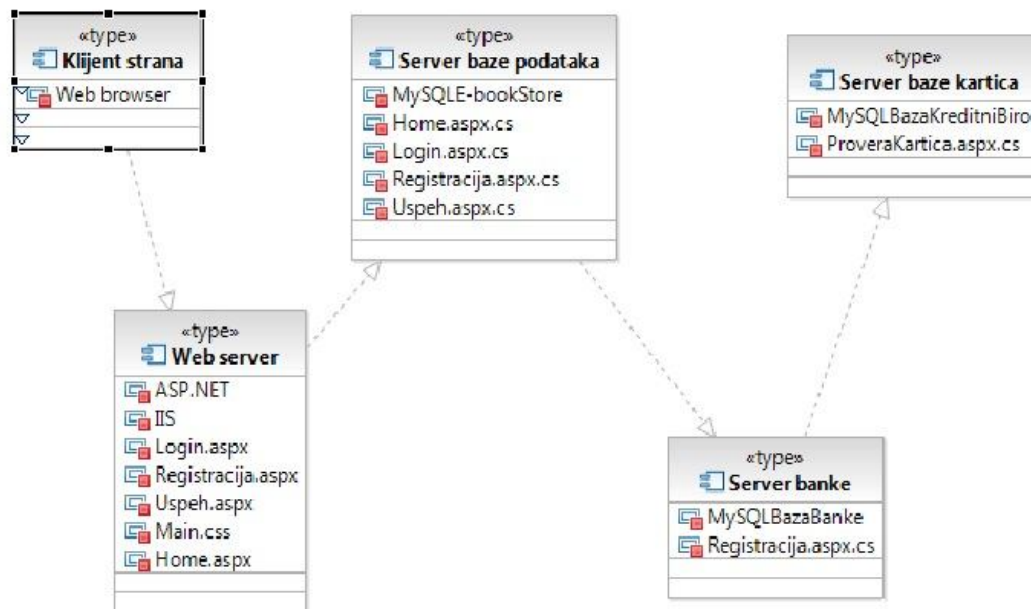
Daljim modelovanjem pomoću UML došlo se do projektovanja dijagrama klasa. Projektovanje sistema podrazumeva grafičko crtanje klasa, prikaz stanja objekata, njihove međusobne veze. Završni deo projektovanja odnosi se na grafičko predstavljanje, implementacije – hardvera, potrebnog za uspešno realizovanje samog sajta E-book store kao i njegovo uspešno funkcionisanje. Projektovanje sistema postoji u cilju olakšavanja posla programerima. Ideja je da programeri mehanički kucaju unapred „pripremljen“ kod.

Kod dizajna klasa vrlo je bitno da se prekontrolišu svi objekti, atributi i metode koje treba da sadrži sam projekat. Iz tih razloga je dijagram klasa veoma koristan, jer se mnogo lakše na dijagramu klasa mogu ispraviti greške nego kasnije u samom kodu. Dijagram klasa je predstavljen na slici 3.2.9.



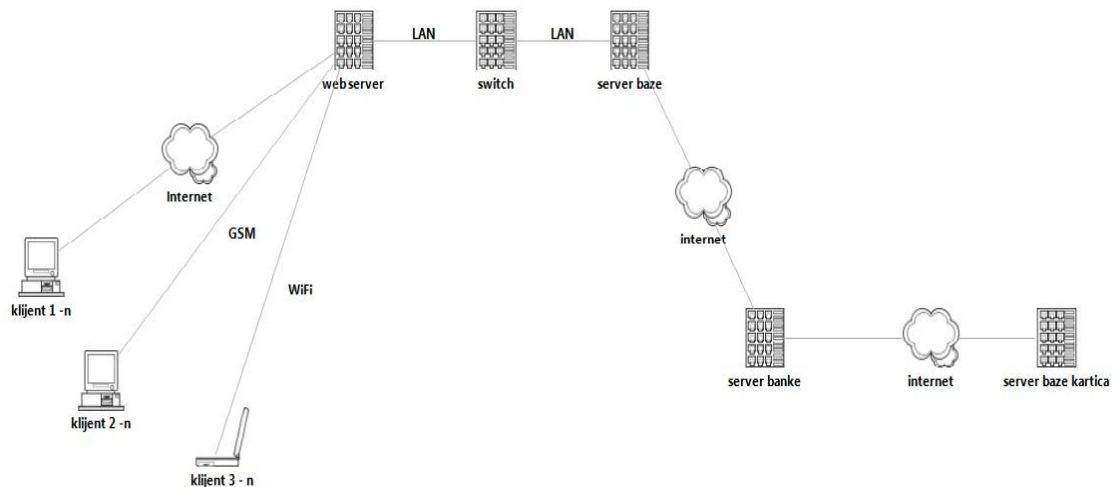
Slika 3.2.9 Dijagram klasa

Pošto se ceo projekat sastoji od tri sloja, takav sistem modelovanja se naziva troslojna arhitektura. Sastoji se od sloja baze podataka, aplikacionog sloja i logičkog sloja.



Slika 3.2.10 Dijagram komponenti

Dijagrami komponenti se prave da bi pokazali komponente koje čine fizičku realizaciju sistema. Dijagram komponenti vezan za sajt E-book store je predstavljen na slici 3.2.10.



Slika 3.2.11 Dijagram uvođenja

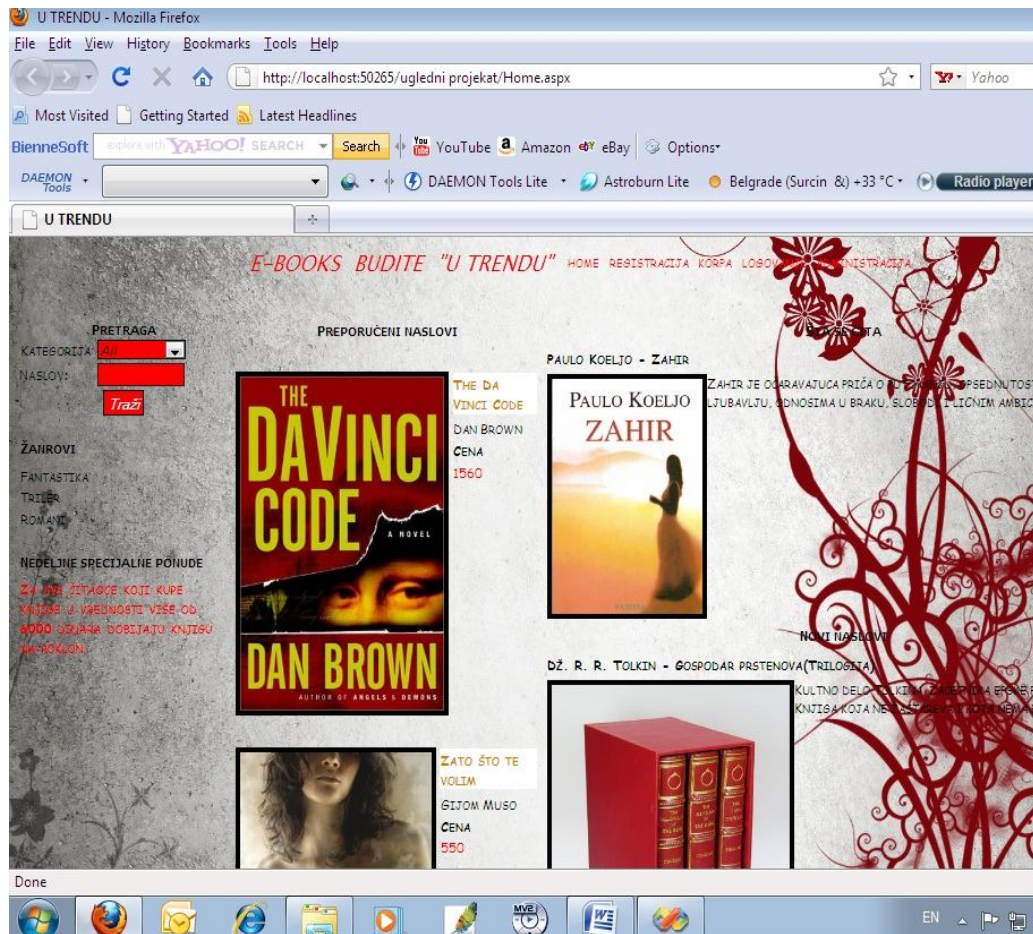
Dijagram uvođenja, predstavljen na slici 3.2.11, najslikovitije objašnjava koji je to hardver potreban za implementaciju i koji su sve to akteri koji se mogu javiti kao deo sistema. Takođe na slici 3.2.11 mogu se videti koje sve vrste veza u smislu mreža podržava celokupan sistem. Navedeni dijagram predstavlja osnovnu strukturu sistema.

Sistem se sastoji od servera baze na kome je smeštena baza podataka, koja je LAN mrežom povezana preko switch-a na web server. Na hardverskoj podlozi web servera, smeštena je aplikacija ASP.NET koja propušta pakete preko porta 50265. Pristup korisnici mogu vršiti preko adrese URL <http://localhost:50265/ugledniprojekat> . To je ujedno i ROOT URL samog projekta.

Pošto se početna strana automatski pokreće startovanjem ASP.NET servera, a stranica nosi ime „U TRENDU“ tj. Home.aspx, kako je memorisana u alatu Visual Studio 2008, korisnici mogu pristupiti početnoj strani putem internet konekcije, GSM ili WiFi bežične mreže. Serveri su hardverski opremljeni sa po 64Mb RAM memorije i sa procesorima sa četiri jezgra kako bi svaki zahtev bio razrešen u realnom vremenu. Propusna moć kabla preko kojeg korisnici vrše konekciju igra bitnu ulogu ali sa strane web servera na koje je smeštena web aplikacija. Preporuka je da se koristi optički kabal na web serveru kome korisnici pristupaju, jer sam optički kabal ima veliku propusnu moć i time omogućava transfer velikog broja paketa podataka u veoma kratkom vremenskom intervalu. Konekcija sa strane korisnika nije toliko bitna za rad same web aplikacije, već je bitna da korisnik omogući tim putem sebi brži pristup, ali pošto sajt E-book store nije grafički zahtevan, a poznato je da grafički zahtevni sajtovi traže da konekcije budu sa velikom propusnom moći, tako da otvaranje stranica sajta E-book store i funkcionalnost istog ne zahtevaju određenu niti specifičnu vrstu konekcije od strane korisnika. Ovo je veoma bitan zahtev koji treba ispoštovati, jer se time omogućava da svaki korisnik može pristupiti sajtu nezavisno kakvu konekciju poseduje u zavisnosti od brzine i tipa. Server na kome se nalazi implementirana baza podataka knjige.mdb je povezan putem Internet konekcije sa serverom banke i serverom baze kartica.

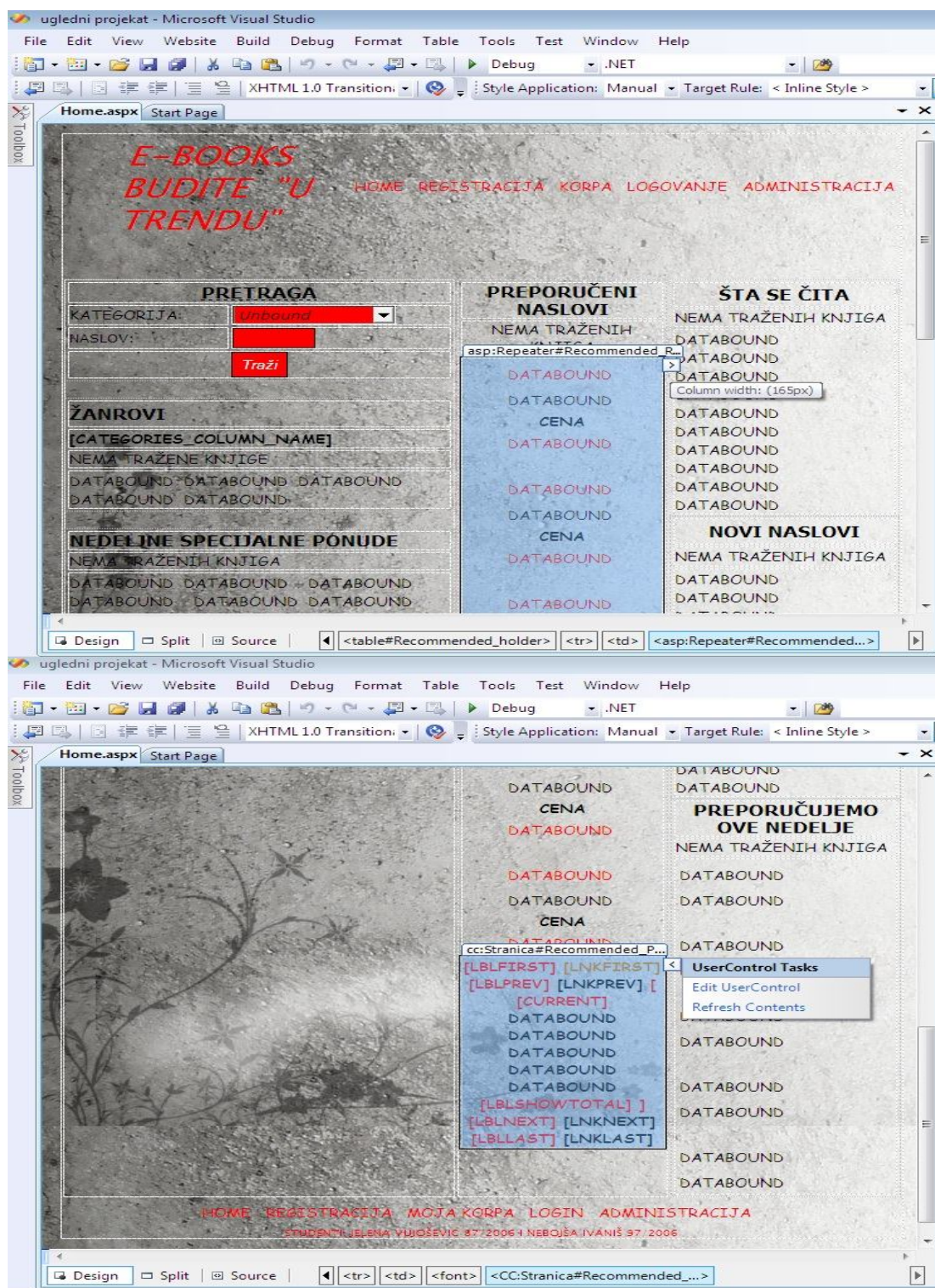
Objašnjenjem same projekcije sajta za prodaju knjiga tokom poglavlja 3.2 i objašnjenjem modelovanja i projektovanja baze podataka knjige.mdb stiče se slika o funkcionalnosti sajta i njegovoj kompleksnosti. Obradom koda pisanog u programskom jeziku C# se nije bavilo u diplomskom radu, jer to nije zadata tema.

Potrebno je objasniti konekcije i metode vezane za uspostavljanje veze između baze podataka knjige.mdb i logičkog sloja web aplikacije E-book store.



Slika 3.2.12 Home Page

Slika 3.2.12 prikazuje početnu stranu. Ovo je bitno da bi se videlo, kako bi se ušlo u srž problema vezanog za konekciju web aplikacije E-book store i baze podataka knjige.mdb.



Slika 3.2.13 Alat Visual studio 2008 – projektovanje početne strane

Slika 3.2.13 pokazuje da će se na početnoj strani Home.aspx prikazivati informacije iz baze podataka knjige.mdb. Očigledno je, u odnosu na početnu stranu Home page predstavljenu na slici 3.2.12, da su mesta na kojima piše DATABOUND, mesta na početnoj stranici Home.aspx gde će se informacije iz baze podataka knjige.mdb pojavljivati.

Dizajnerski deo je urađen u HTML-u i CSS-u kako bi se korisniku pružile prave ali i vizuelno privlačne informacije.

U nastavku je predstavljen deo koda koji daje informacije o fizičkoj lokaciji baze podataka knjige.mdb.

```
<add key="sBook_StoreDBConnectionString"
value="Provider=Microsoft.Jet.ODBC.4.0;UserID=Admin;Data
Source=C:\ugledniprojekat\App_Data\knjige.mdb;Persist Security
Info=False"/>
```

Slika 3.2.14 Predstavljanje lokacije baze podataka knjige.mdb

Ovaj deo koda je putanja do baze podataka knjige.mdb. Kod koji je prikazan na slici 3.2.14 je bitan zbog mogućnosti da se fizički može promeniti mesto baze podataka ako je to potrebno. Tako da se otvara mogućnost, da celokupnu web aplikaciju E-book store sa bazom podataka knjige.mdb, premestimo na drugi računar. I pri toj aktivnosti neće se izgubiti funkcionalnost sajta E-book store. Sledeća slika 3.2.15 zajedno sa slikom 3.2.14 predstavlja uspostavljanje konekcije ODBC driver-a sa bazom podataka knjige.mdb.

```
public void DBOpen()
{
    // uzmi apsolutnu putanju baze iz fajla Config.web
    string sConnectionString =
System.Configuration.ConfigurationSettings.AppSettings["sBook_StoreDBConn
ectionString"];
    // otvori bazu koriscenjem OdbcConnection
    Connection = new OdbcConnection(sConnectionString);
    Connection.Open();
}
```

Slika 3.2.15 Metod konektovanja sa bazom podataka knjige.mdb

Konektovanje sa bazom podataka knjige.mdb kako je predstavljeno na slici 3.2.15, govori o metodu koji uspostavlja konekciju sa bazom podataka i čita gde se nalazi baza podataka, što je i predstavljeno u komentaru na slici. Potom uzima ODBC driver, pravi novu instancu klase OdbcConnection i tada konstruktoru prosleđuje parametre za uspostavljanje konekcije value, koji je predstavljen na slici 3.2.14 i sa funkcijom Connection.Open() otvara konekciju prema bazi podataka knjige.mdb. Posle otvaranja konekcije stiče se mogućnost da se uspostavi korektna veza i svim funkcijama iz programskog jezika C# omogućiti normalan rad sa bazom podataka knjige.mdb u pogledu upita i povlačenja izveštaja.

Obradom u logičkom sloju web aplikacije dolazi se do delova koda vezanih za relacione veze baze podataka knjige.mdb i programskog jezika C#.

```
void Search_Show()  
{  
  
    Utility.buildListBox(Search_category_id.Items, "select  
category_id,name from categories order by 2", "category_id", "name",  
"All", "");  
  
    //Pretraga  
    string s;  
  
    s = Utility.GetParam("category_id");  
  
    try  
    {  
        Search_category_id.SelectedIndex =  
Search_category_id.Items.IndexOf(Search_category_id.Items.FindByValue(s))  
;  
    }  
    catch { }  
  
    s = Utility.GetParam("name");  
    Search_name.Text = s;  
  
}
```

Slika 3.2.16 C# kod pretrage po žanrovima

Kod koji je predstavljen na slici 3.2.16 opisuje metod Search_Show(). Metod služi da povuče informaciju iz baze podataka knjige.mdb vezane za izbor žanrova određene knjige. Upotrebom SQL-a omogućeno je da se izvrši upit u bazu podataka knjige.mdb na određeni objekat categories i na njegove attribute koji ga opisuju. Tekst boks omogućava pretragu po kategoriji žanra knjige, ali pruža i mogućnost pretrage i po imenu knjige. Prvenstveno je potrebno u promenljivu string – s, smestiti naziv kategorije. Pošto su svi žanrovi smešteni u bazu podataka u atribut category_id, zbog toga je prosleđena brojčana vrednost iz atributa category_id kao parametar. GetParam() služi kao pomoć da bi se dobila informacija da li postoji parametar u QueryString-u, koji mu se prosledjuje, kao na primer, category_id. Ako bi bio prosleđen parametar koji ne postoji vratio bi prazan string.

Povezanost svih elemenata informacionih tehnologija korišćenih za izradu web aplikacije E-book store, za elektronsku prodaju knjiga, kao što se moglo videti iz celokupnog rada, pospešuje jednostavniju administraciju u održavanju sajta i mogućnost da korisnici imaju lagodan odnos u interakciji sa samim sajtom.

4 Zaključak

Sajt je realizovan tako da prati nove tendencije i očekivanja na Internet i e-komerc tržištu. Banch marking omogućava uvid u uspešnost poslovanja. Radi dinamičnosti i funkcionalnosti, modelovana je i dizajnirana baza podataka knjige.mdb tako da prati novi razvoj tehnologija.

Za realizaciju baze podataka knjige.mdb primenjene su savremene tehnologije Microsoft Visual Studio 2008 i Access database 2007.

Kako hardver i potrebni alati čine samo temelj dobre baze podataka, zadatak koji se nameće, jeste sposobnost pravljenja dobre arhitekture.

Za modelovanje korišćen je UML (Unicode Markup Language) jezik. Modelovanje baze podataka i web aplikacije u UML-u pomoglo je da se sagledaju procesi koji su neophodni za dobro funkcionisanje sajta E-book store.

S' obzirom da je cela web aplikacija objektno orjentisana, radi uspešnog modelovanja baze podataka bilo je potrebno sagledati koji su to učesnici, aktivnosti, objekti, atributi, klase i metode prisutne u projektu koji obrađuje diplomski rad.

Dijagrami aktivnosti, sekvenci i klasa omogućili su dobar preduslov za dizajniranje baze podataka. UML projektovanje sprečava pojavu grešaka u samoj logici arhitekture baze podataka. Kada je baza podataka dobro projektovana, modelovanje je svedeno na automatizam.

Zahvaljujući integrisanosti Visual Studio i Access database-a, kada je baza podataka uspešno modelovana i realizovana, upotrebom ODBC driver-a vezana je za web aplikaciju.

Sajt E-book store bavi se elektronskom prodajom knjiga. Jedan od najvećih problema koji se može javiti, jeste isporuka u realnom vremenu. Korišćenjem dobre hardverske platforme obezbediće se stabilan rad baze podataka kao i njen kapacitet. Sa druge strane, visoka propusna moć konekcije prema sajtu od strane korisnika je omogućila brži odziv na upite koje korisnik postavlja. Sve to je obrađeno i implementirano kako se ne bi narušila funkcionalnost i dinamičnost sajta E-book store, a kako bi se omogućilo korisniku da nesmetano i u realnom vremenu bude uslužen, a isporuka se obavi na željenom nivou.

Razvoj baza podataka i objektno orjentisanog modelovanja mora da prati razvoj na Internet tržištu kako bi mogao da odgovori na zahteve korisnika.

Postignuto je prilagođavanje baze podataka na taj način, da ona obezbedi funkcionalnost, laku upotrebu administratoru i zaposlenima, a sa druge strane efikasnost korisnicima.

Literatura

- [1] Veinović, M., **Baze podataka**, drugo izdanje, Univerzitet Singidunum, Beograd, 2008.
- [2] Lippman, S., **C# izvornik**, drugo izdanje, CET Computer Equipment and Trade, Beograd, 2003.