

**SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO – MATEMATIČKI FAKULTET  
MATEMATIČKI ODSJEK**

Tamara Bucić

**BAZE PODATAKA ZA ELEKTRONIČKE KNJIŽNICE**

Diplomski rad

Voditelj rada:  
prof.dr.sc. Nenad Antić

Zagreb, veljače 2012.

Ovaj diplomski rad obranjen je dana \_\_\_\_\_ pred ispitnim povjerenstvom u sastavu:

1. \_\_\_\_\_, predsjednik
2. \_\_\_\_\_, član
3. \_\_\_\_\_, član

Povjerenstvo je rad ocijenilo ocjenom \_\_\_\_\_.

Potpisi članova povjerenstva:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_

## **Zahvala**

*Zahvaljujem svima koji su svojim prijedlozima, savjetima i podrškom pridonjeli izradi ovog rada. Posebno se zahvaljujem svom mentoru prof.dr.sc. Nenadu Antoniću na strpljenju, pomoći, vodstvu i izuzetnoj suradnji tijekom izrade rada. Također se zahvaljujem dr.sc. Vedranu Šegi na bezrezervnoj pomoći i pruženom znanju kad god bih zapela na bilo kojem području programskog dijela rada. Moram također spomenuti i svoga brata, koji mi je također priskočio u pomoć pri izradi programskog dijela rada, pogotovo pri instalaciji aplikacije na web server.*

*Ovim putem želim također zahvaliti i svim svojim kolegama i prijateljima koje sam upoznala na fakultetu što su svojim prisustvom uljepšali moje studentsko razdoblje.*

*A posebna zahvala mojim roditeljima koji su mi omogućili život i studiranje u drugom gradu i što su me poticali u odabiru studija koji volim i koji me zanima.*

# Sadržaj

<b>Sadržaj</b> .....	<b>iii</b>
<b>Uvod</b> .....	<b>1</b>
<b>Poglavlje 1. Elektroničke knjižnice</b> .....	<b>2</b>
1.1. Uvod .....	2
1.2. Tradicionalne i elektroničke knjižnice .....	4
1.3. Način digitalizacije knjiga .....	7
1.4. Klasifikacija knjiga.....	9
1.5. Prednosti i nedostaci elektroničkih knjižnica .....	11
1.6. Budućnost.....	12
<b>Poglavlje 2. Baze podataka</b> .....	<b>14</b>
2.1. Što su baze podataka.....	14
2.2. Modeliranje entiteta i veza .....	15
2.3 Relacijski model.....	18
2.4. Pretvaranje ER sheme u relacijsku .....	19
2.5. Normalizacija relacijske sheme .....	20
2.6. SQL i SQLite.....	22
<b>Poglavlje 3. Web programiranje</b> .....	<b>27</b>
3.1. World Wide Web.....	27
3.2. HTML .....	28
3.3. CSS .....	30
3.4. PHP .....	32
<b>Poglavlje 4. Praktični dio diplomskog rada</b> .....	<b>35</b>
4.1. Struktura modela.....	35
4.2. Presjek poslovne logike .....	36
4.2.1. Sloj podataka .....	37
4.2.2. Sloj poslovne logike .....	40
4.2.3. Sloj korisničkog sučelja .....	51
4.3. Presjek sigurnosne logike .....	69
4.3.1. Sloj podataka .....	70
4.3.2. Sloj poslovne logike .....	70

4.3.3. Sloj korisničkog sučelja .....	72
4.4. Razine korisnika .....	73
4.5. Preduvjeti za instalaciju.....	76
<b>Zaključak.....</b>	<b>77</b>
<b>Sažetak .....</b>	<b>78</b>
<b>Summary .....</b>	<b>80</b>
<b>Literatura.....</b>	<b>78</b>
<b>Životopis.....</b>	<b>81</b>

## Uvod

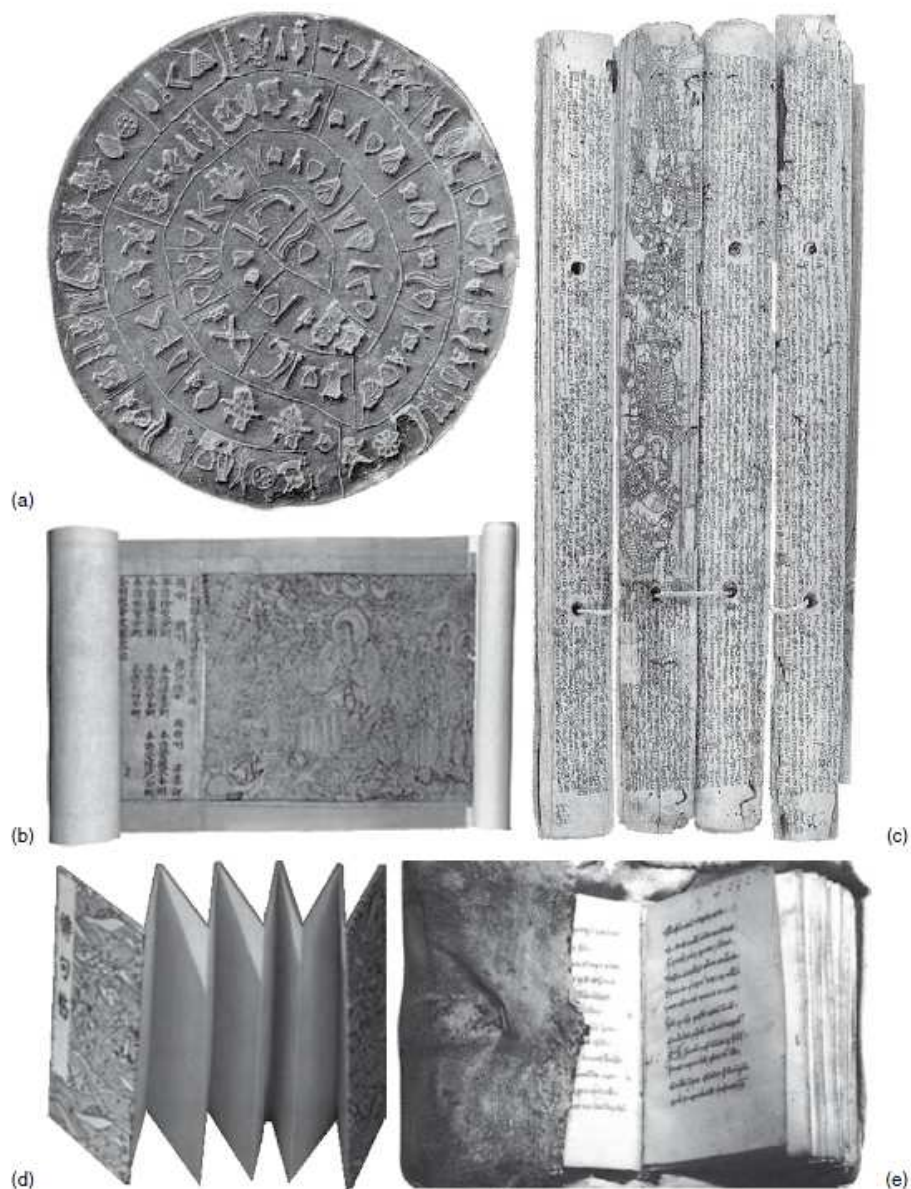
*„Knjige su najtiši i najstalniji prijatelji; one su najprihvatljiviji i najmudriji savjetnici i najstrpljiviji učitelji.“*

*Charles Eliot*

Povijest knjige stara je više od 7000 godina. Priča o knjizi počinje u bogatoj ravnici u južnom dijelu Mezopotamije, gdje su Sumerani, jedan od najtajanstvenijih naroda u povijesti čovječanstva, stvorili izvanredno visoku civilizaciju. Na području Sumera pronađeni su najstariji zapisi. Najčešće se pisalo tvrdim pisaljka po glinenim pločicama. Kako je ostajao otisak u obliku klina, njihovo je pismo nazvano klinasto pismo. Od Sumerana se sačuvalo najstarije književno djelo - Ep o Gilgamešu.

U tih više od 7 tisućljeća knjiga je mijenjala svoj oblik, ovisno o materijalu od kojeg je izrađivana i o načinu pisanja i umnožavanja tog materijala, a pri pisanju pribor se mijenjao kroz povijest. Najstarije kineske knjige bile su sastavljene od bambusovih daščica, a kasnije su ih zamijenili svici od svile. Stari Egipćani pisali su na drvenim i kamenim pločama, te na svicima papirusa. Stari Grci i Rimljani služili su se pravokutnim drvenim ili voštanim pločama za kratke bilješke, trgovačke obračune ili za pisanje školskih zadaća. Po dvije pločice mogle su biti svezane zajedno, pa bi oblikovale „bilježnicu“. Knjige od papirusa nisu imale korice nego tobole. Najstariji kožni uvez imalo je koptsko sveto pismo iz 6. stoljeća. Taj je uvez bio ukrašen utisnutim ornamentima koji su kasnije postali jako popularni. Tako je kodeks dobio zbijen oblik, spretan za korištenje i jednostavan za pregledavanje. U njemu je bilo spojeno puno stranica, tako da su knjige mogle biti puno deblje i opširnije.

Prve tiskane knjige, tzv. inkunabule u svemu su oponašale kodekse. No od početka 16. stoljeća knjige su sve više poprimale današnji izgled. Iako su Kinezi već u 6. stoljeću rabili rezbarene drvene ploče za umnožavanje tekstova, a pokretna metalna slova korištena su za tiskanje u Koreji sredinom 13. stoljeća, u Europi su se do sredine 15. stoljeća knjige umnožavale prepisivanjem rukom, što ih je činilo skupima i rijetkima. Izum tiskarskog stroja, kojeg je oko 1430. izumio Nijemac Johannes Gutenberg, omogućio je eksplozivno širenje znanja u renesansnoj Europi. Stari tiskari bi najprije u okvir složili olovna slova pa ih onda premazali bojom. Potom bi na njih stavili papir i pritisnuli ga drvenom prešom tako da se boja upije u papir. Do 19. stoljeća svi su se poslovi oko izrade knjiga obavljali ručno, no danas to obavljaju strojevi.



Slika 1. Razni načini za pisanje (a) glinena pločica; (b) svitak; (c) listovi palme; (d) harmonika; (e) kodeks

U 20. stoljeću knjiga je poprimila i druge oblike – knjige na mikrofilmu, na magnetskoj vrpci (tzv. govorne knjige), itd.. Najvažniju promjenu donijela je elektronička revolucija i brz razvoj tehnologije. Tekst iz knjiga preselio se na računalne ekrane, te je tako stvoren novi oblik knjige – elektronička knjiga. Začetnikom značajnije pojave knjiga u elektroničkom obliku (e-knjiga) možemo smatrati studenta sveučilišta u Illinoisu, Michaela S. Harta, koji je 1971. godine započeo „Projekt Gutenberg“ time što je u elektronički oblik prebacio američku Deklaraciju nezavisnosti. Projekt Gutenberg je od tada volontaristički projekt za digitaliziranje i arhiviranje kulturnih djela, koji potiče stvaranje i distribuciju e-knjiga, te se smatra najstarijom digitalnom knjižnicom. Većina stavki u zbirci su puni tekstovi knjiga iz javnog vlasništva. Prema podacima Projekta Gutenberg, u lipnju 2011. godine u njihovoj zbirci nalazilo se više od 36.000 e-knjiga. Ciljevi projekta su da se knjige

## UVOD

pretvaranjem u elektronički oblik sačuvaju (jer papir trune), te da budu dostupne preko Interneta svima kojima su knjige zanimljive i koji su željni znanja.

*„Knjiga koja želi trajati je ona koja se može čitati na različite načine. U svakom slučaju treba omogućiti različito, promjenjivo čitanje. Svaka generacija čita velika djela na drugačiji način.“*

*Horhe Luis Borhes*

Moj diplomski rad sastoji se iz dva dijela – praktičnog i teorijskog. Praktični dio sastoji se od izrade elektroničke knjižnice, koja u ovom slučaju ne služi samo za pohranu podataka o elektroničkim knjigama, nego i kao baza podataka za pohranu informacija o knjigama u papirnatom obliku. Teorijski dio je ova radnja koja se upravo nalazi pred Vama u kojoj sam obradila teme vezane uz izradu praktičnog dijela rada, a to su – elektroničke knjižnice, baze podataka, web programiranje te opis praktičnog dijela rada.



## Poglavlje 1.

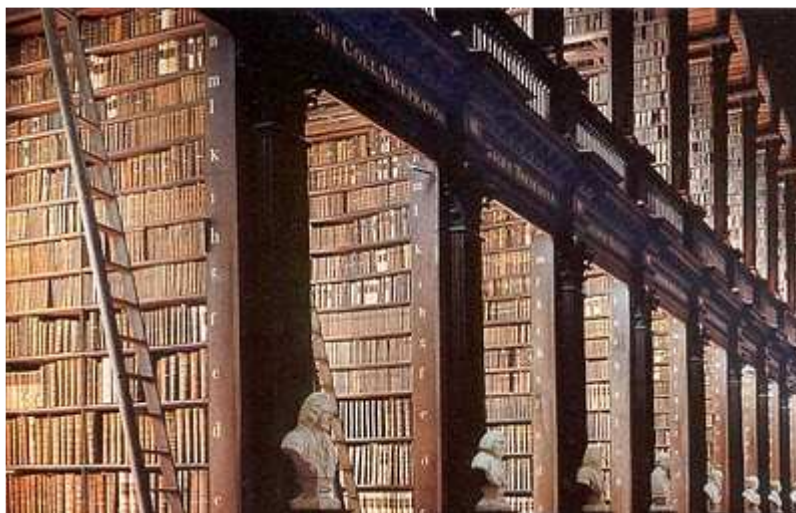
### Elektroničke knjižnice

Što su to knjižnice u digitalnom dobu? Koju svrhu služe u današnje vrijeme u kojem nova, digitalna građa zauzima centralno mjesto? Što su elektroničke knjižnice i koja je razlika između elektroničkih i klasičnih knjižnica? Koje su im prednosti i mane? Kakva je budućnost knjižnica uopće? Sva ova i još neka dodatna razmatranja obrađena su u ovom poglavlju.

#### 1.1. Uvod

Informacijsko doba uslijedilo je nakon industrijskog doba, i obilježava ga brzina kretanja informacija koja je veća od brzine fizičkog kretanja. Njegova glavna odrednica je rasprostranjenost informacijske tehnologije koja povećava brzinu i učinkovitost prijenosa informacija. U današnje doba ubrzanog razvoja informacijske i komunikacijske tehnologije (ICT tehnologije), sve više se ukazuje beskrajna potreba za pohranjivanje, organizaciju, te lakši i efikasniji pristup informacijama.

Tradicionalne knjižnice i upotreba knjiga u papirnatom obliku postaju zastarjeli. Tradicionalne knjižnice postaju prevelike i nepregledne, zatrpane starim i prašnjavim knjigama, koje je teško organizirati i pronaći. Dok knjiga prođe put od autora do čitatelja, prođe mnogo vremena unutar kojega se tehnologija još mnogo puta promijeni.



Slika 2. Nepreglednost tradicionalnih knjižnica

Četiri su glavna razloga za uvođenje automatizacije u knjižnice:

1. povećani opseg posla (uzrokovano povećanom količinom informacija)
2. potreba za većom djelotvornošću (ušteda radnog vremena i novca)
3. nove službe (razne vrste popisa, statistika, ...)
4. kooperacija i centralizacija (mogućnost diobe podataka koja vodi do zajedničkih sustava)

Mogućnost pohrane podataka na različitim medijima sve većeg kapaciteta i sve manjih cijena dala je zamah mnogim eksperimentalnim skladištima teksta, slika i numeričkih podataka, dohvatljivih s bilo kojeg mjesta na svijetu koje ima neophodnu infrastrukturu. Neizmjerne mogućnosti koje nude nove tehnologije uzrokovale su bespoštednu utrku između različitih poduzeća uključenih u informacijsku industriju. Iskustva su pokazala da će spremišta podataka zahtijevati sve manje i manje prostora, da će pristup informacijama biti sve brži i učinkovitiji i da bi predstavljanje informacija u tiskanom obliku moglo sasvim nestati. Tradicionalne knjižnice postoje preko 25 stoljeća, dok elektroničke knjižnice postoje jedva 15 godina. Međutim, u današnjem informacijskom društvu elektroničke knjižnice će se sigurno pokazati jednim od najvažnijih i najutjecajnijih institucija novog doba. Ipak, sumnjam da će elektroničke knjižnice u potpunosti zamijeniti tradicionalne, barem ne u skoroj budućnosti.

Pa što je onda elektronička knjižnica? Definicija elektroničke knjižnice bila bi ukratko:

- Uređeni skup digitalnih objekata, uključujući tekst, video i audio zajedno s metodama za pristup i dohvaćanje podataka, te za odabir, organizaciju i održavanje podataka.

Elektroničke knjižnice sastoje se od izvora dostupnih u digitalnom obliku kojima se može pristupiti lokalno (pohranjeni na hard disk) ili preko mreže (javne ili privatne). One sadrže određene pokaznice za odabir sadržaja kako bi se održala dosljednost skupa podataka. Kako se bilježi sve veći porast izvora informacija na Webu, korisnici se suočavaju s problemom traženja autoritativnih i autentičnih informacija. E-knjižnice mogu poslužiti i kao „vodič“ do izvora dostupnog na webu. Također vode korisnika do e-knjiga, e-časopisa, web siteova i multimedijских sadržaja. U njima korisnik može pronaći opise e-knjiga, recenzije čitatelja, popise najčitanijih i najpopularnijih knjiga, a može i kupiti knjigu.

## 1.2. Tradicionalne i elektroničke knjižnice

Je li elektronička knjižnica institucija ili dio tehnologije? Izraz elektronička knjižnica, kao i sama riječ knjižnica ima različito značenje različitim ljudima. Mnogi ljudi imaju sliku knjižnice kao fizičkog objekta od cigle i žbuke, mirnog mjesta na kojem se čuvaju knjige. Profesionalnim knjižničarima to su institucije za pohranu, prikupljanje i organizaciju materijala, kao i za pristup istome. Materijal u knjižnici nisu nužno knjige, postoje i zbirke umjetnina, filmova, audio zapisa, botaničkih uzoraka i kulturnih objekata. Istraživačima su knjižnice mreže koje omogućuju pristup svjetskom znanju gdje god se ono nalazilo. Današnji studenti znanosti i tehnologije, nažalost, sve više misle o knjižnici kao o *World Wide Web-u*, tj. oni pogrešno shvaćaju Web kao univerzalnu knjižnicu.

Elektronička knjižnica nije samo „digitalizirana knjižnica“. Elektronička knjižnica je okvirno novi način bavljenja znanjem – pohranjivanje, prikupljanje, organizacija, širenje znanja i pristup znanju, a ne uništenje postojećih institucija. U prošlom podpoglavlju elektronička knjižnica je definirana kao:

- Uređeni skup digitalnih objekata zajedno s metodama za pristup i dohvaćanje podataka, te za odabir, organizaciju i održavanje podataka.

Ova široka interpretacija „digitalnih objekata“ (ne samo teksta) se očituje u tome da se osim audio i video zapisa mogu uključiti i simulacije, dinamičke vizualizacije i virtualna stvarnost. Drugi dio definicije (pristup i dohvaćanje podataka) odnosi se na običnog korisnika, dok se treći dio definicije (odabir, organizacija i održavanje podataka) odnosi na knjižničara i to je centralni dio ideje knjižnice. Knjižničar je tu da odluči što uključiti u zbirku podataka, što ne, te da na prikladan način organizira i održava informacije. To razlikuje elektroničke knjižnice od neuređenog skupa informacija koji zovemo *World Wide Web*.



Slika 3. Knjižničarka u tradicionalnoj knjižnici

Elektroničke knjižnice inače jasnu razliku između korisnika i knjižničara ponekad čine ne baš tako jasnom. Ali u svakom slučaju, vrlo je bitno imati na umu razliku između te dvije strane. Programi elektroničkih knjižnica pružaju podršku korisniku za pregledavanje i pretraživanje zbirke. Isto tako, pružaju podršku knjižničarima u njihovim nastojanjima da pruže odgovarajuće organizacijske strukture i da ih učinkovito održavaju.

Elektroničke knjižnice su knjižnice bez zidova, ali i one također trebaju granice. Sam pojam zbirke implicira granicu – činjenicu da su određene stvari unutar zbirke znači da ostale moraju ostati izvan nje. I zbirke zahtijevaju neku vrstu konceptualnog integriteta koji im daje jedinstvenost i identitet. Svaka zbirka mora imati jasnu svrhu, koja definira ciljeve koji se namjeravaju postići, kao i skup pravila koja služe kao smjernice koje će voditi odluke knjižničara o tome što će biti uključeno u zbirku i – jednako važno – što će biti isključeno iz nje. To nije nimalo lagan posao.

Kod elektroničkih zbirki vidimo samo neprovidnu pojavu – zaslon (najčešće Web stranica) s nimalo naznaka o tome što se nalazi iza njega i u kojoj količini. Nalazi li se iza tog zaslona dobro probrano blago ili hrpa bezvrijednih prolaznih stvari? Nalazi li se tamo samo nekolicina dokumenata ili nekoliko milijuna dokumenata? Kod fizičkih knjižnica je drukčije, one zauzimaju fizički prostor i izlažu opipljivu fizičku organizaciju. Na ulazu u takvu knjižnicu čovjek osjeti prisutnost i nepromjenjivost koji odražavaju trud uložen u izgradnju i održavanje jedne takve zbirke. Nitko ne može to zamijeniti sa hrpom beskorisnog smeća. Dok u virtualnom svijetu razlika nije toliko očigledna.

U gornjem tekstu već je određena razlika između elektroničke knjižnice i *World Wide Web-a* – Web ima manjak neophodnih obilježja poput selekcije i organizacije. Također je bitno razlikovati elektroničku knjižnicu od Web stranice, čak i one stranice koja nudi uređenu zbirku dobro organiziranog materijala. Web stranice koje pružaju obilje digitalnih objekata zajedno s prikladnim metodama za pristup podacima i njihovom dohvaćanju ne moraju nužno biti „knjižnice“. Knjižnice su skladišta podataka koja omogućavaju jednostavno dodavanje novih podataka. Mnogo dobro organiziranih web stranica stvorene su putem ručno izrađenih hipertekstualnih povezanih struktura. Ali jednako kao što dodavanje novih znanja u fizičke knjižnice ne uključuje i njihovo pretakanje u knjige i obradu, tako ni novi materijal ne može postati dio elektroničke knjižnice bez ručne obrade struktura za pristup i dohvaćanje podataka.

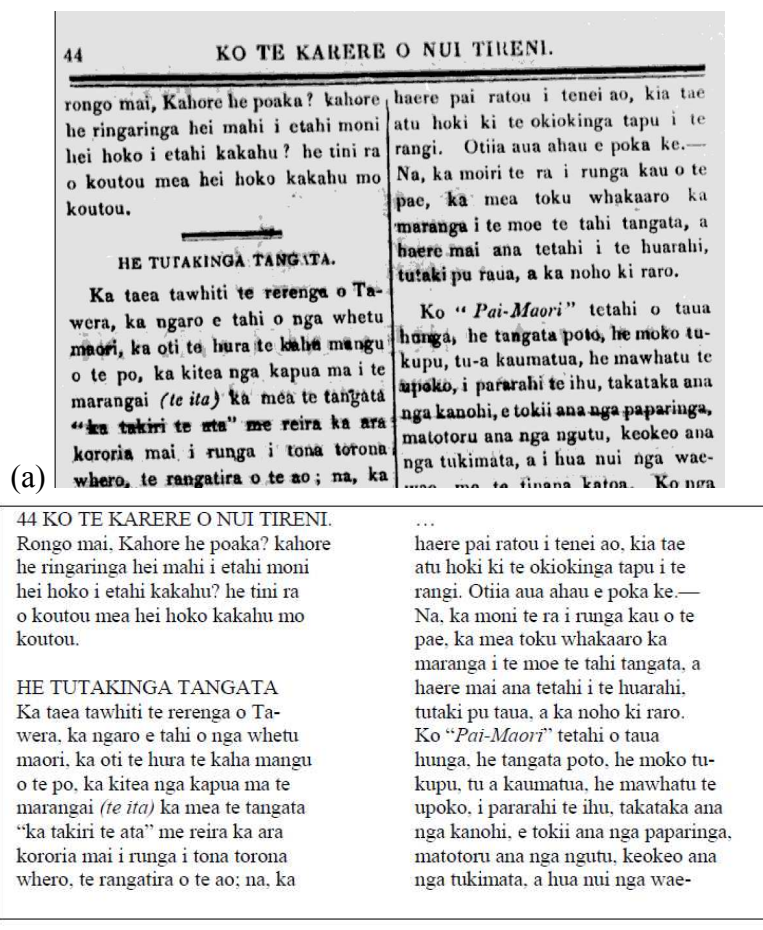
Ono što čini novi materijal u opipljivom formatu dijelom zbirke fizičke knjižnice je dijelom i njegova pozicija na policama knjižnice, ali bitnija je informacija o samom materijalu koja se nalazi u katalogu knjižnice. U elektroničkim knjižnicama postoji slična „katalogizacija“, tj. *metapodaci* (podaci o podacima) i ona predstavlja važan dio u elektroničkim knjižnicama.

Tablica 1. Prednosti i nedostaci elektroničkih i tiskanih knjiga

	<b>Prednosti</b>	<b>Nedostaci</b>
<b>E-knjiga</b>	<ul style="list-style-type: none"> <li>- Tekst može biti pretraživan, osim kada je u slikovnom obliku</li> <li>- Zauzima malo mjesta (otprilike 500 prosječnih e-knjiga se može spremi na jedan cd)</li> <li>- E-knjige se mogu čitati na slabom svjetlu, pa i u totalnom mraku</li> <li>- Brzo kopiranje</li> <li>- Niska cijena distribuiranja</li> <li>- Ekonomski i ekološki su prihvatljive jer smanjuju potrošnju papira i tinte</li> <li>- Ne istroše se tijekom vremena</li> </ul>	<ul style="list-style-type: none"> <li>- Može biti inkompatibilna s novim hardwareom i softwareom</li> <li>- Zahtijeva pažnju u rukovanju i spremanju dokumenata, da bi se izbjeglo gubljenje podataka</li> <li>- Program može onemogućiti ispis</li> <li>- Baterija se može isprazniti</li> <li>- Mnogi korisnici preferiraju papir umjesto kompjuterskog ekrana</li> <li>- Prosječna cijena je mnogim korisnicima preskupa u odnosu na tiskane knjige</li> </ul>
<b>Tiskana knjiga</b>	<ul style="list-style-type: none"> <li>- Može se čitati na različitim mjestima</li> <li>- Robusna i izdržljiva</li> <li>- Može se čitati i kada je oštećena</li> <li>- Ne zahtjeva izvor struje kao računalo</li> <li>- Ima veću kolekcionarsku vrijednost (npr. prva izdanja, potpisane knjige)</li> </ul>	<ul style="list-style-type: none"> <li>- Ne može biti lako kopirana</li> <li>- Ne može se čitati u mraku</li> <li>- Zauzimaju mnogo mjesta, teške su za prenošenje (pogotovo na putovanjima)</li> <li>- Mogu se prestati tiskati – što uvelike otežava dobavljanje</li> <li>- Ekološki su upitne jer se za njih koriste velike količine papira</li> </ul>

### 1.3. Način digitalizacije knjiga

Digitalizacija je prijenos građe u digitalni oblik, postupak snimanja, pohranjivanja i obrade sadržaja korištenjem digitalne kamere, skenera i računala. Moguće je digitalizirati sve vrste gradiva, od teksta, preko audio i video zapisa, sve do trodimenzionalnih objekata. Stavljanje građe na Web omogućava veću i bržu dostupnost korisnicima diljem svijeta. Mogu se izraditi visokokvalitetne kopije jer se pri umnožavanju ne gubi na kvaliteti. Digitalna građa ne oštećuje se korištenjem, tj. kvaliteta s vremenom ne nestaje i ne umanjuje se uporabom. Tekstualna građa može se digitalizirati na tri načina; prepisivanjem, skeniranjem i fotografiranjem digitalnim fotoaparatom. U slučaju skeniranja i fotografiranja potrebno je provesti i optičko prepoznavanje slova (eng. OCR - *Optical character recognition*).



Slika 4. (a) isječak iz Māori novina; (b) OCR tekst

Prepisivanje je najdugotrajniji i najskuplji od nabrojanih postupaka i najčešće se koristi kod rukopisa, tiskanog teksta s preslabim kontrastom ili tiskanog teksta kojem su pridodane rukom pisane bilješke.

Skeniranje teksta može se izvršiti s originala ili s mikrofilma. Skenirati se može u boji ili u crno-bijeloj tehnici. Ako se nakon skeniranja provodi OCR, tekst će se skenirati kao crno-bijeli dokument kako bi se postigao optimalni kontrast.

Fotografiranje teksta vrši se digitalnim fotoaparatom. Najčešće se koristi kod građe za koju postoji opasnost od oštećenja.

Jednom kada je knjiga digitalno skenirana, slike su dostupne za brzu distribuciju, reprodukciju i čitanje na ekranu. Takve slike se spremaju u nekoliko vrsta formata, najčešći su: DjVu, Portable Document Format (PDF), ili Tagged Image File Format (TIFF). Koristeći OCR za pretvaranje slika određenih stranica knjige u običan tekst, znatno se smanjuje kapacitet potreban za pohranu knjige i omogućuje da se tekst ponovno oblikuje, pretražuje, ili koristi kao predložak aplikacijama za obradu teksta.

Trenutno postoje 3 glavna načina na koje se velike organizacije oslanjaju: eksternaliziranje, skeniranje u „kući“ korištenjem komercijalnih skenera knjiga i skeniranje „u kući“ koristeći robotizirana skenerska rješenja. U slučaju eksternaliziranja, knjige se često dostavljaju na skeniranje jeftinim destinacijama kao što su Indija i Kina. Isto tako, zbog praktičnosti, sigurnosti i napretka tehnologije, mnoge organizacije pak odabiru same vršiti skeniranje, ili pomoću tradicionalnih skenera kod kojih proces skeniranja ipak zna poprilično trajati, ili pomoću drugih rješenja, npr. digitalni fotoaparati koji su bitno brži. Takvim rješenjima se koriste Internet Arhiva te Google. Tradicionalne metode skeniranja uključuju rezanje samog uveza knjige te ubacivanje stranica u skener koji ima mogućnost njihovog automatskog učitavanja. U posljednjih nekoliko godina, softverski upravljani strojevi i roboti razvijeni su za skeniranje knjiga bez potrebe za rezanjem uveza, kako bi se očuvao i sadržaj dokumenta i digitalna foto arhiva njegovog sadašnjeg stanja. Ovaj trend je omogućen prvenstveno zbog poboljšanja tehnologija snimanja i skeniranja koje omogućuju snimanje visoko kvalitetnih digitalnih arhiva slika s malo ili bez ikakvih oštećenja rijetkih ili krhkih knjiga u razmjerno kratkom vremenu. Određeni visoko kvalitetni sustavi skeniranja upotrebljavaju vakuum i statički naboj za okretanje stranica dok se snimanje odvija automatski, obično pomoću kamere visoke razlučivosti koja se nalazi iznad podesive „kolijevke“ u obliku slova V.

Zbog pitanja autorskih prava, većina skeniranih knjiga su one koje ih nemaju, odnosno koje su u javnom vlasništvu. Međutim, Google Book Search je poznat po skeniranju knjiga koje su još uvijek zaštićene autorskim pravima osim ako im nakladnik to striktno ne zabrani.

## 1.4. Klasifikacija knjiga

### ISBN

Međunarodni standardni knjižni broj ili ISBN (International Standard Book Number = međunarodni standardni knjižni broj) jedinstven je identifikator knjiga i drugih omeđenih publikacija bez obzira na medij na kojem se objavljuju. Njegova primjena počinje 1966. godine, kada se počeo upotrebljavati u Velikoj Britaniji kao SBN (Standard Book Number = standardni knjižni broj), a obuhvaćao je samo izdavače s engleskog govornog područja. Sedamdesetih godina prošlog stoljeća SBN postaje međunarodno prihvaćen i preimenovan je u ISBN. 1972. godine osnovan je Međunarodni ured za ISBN sa sjedištem u Berlinu koji koordinira rad nacionalnih ureda za ISBN i prati primjenu sustava. SBN je imao 9 znamenaka. Deseteroznamenasti format ISBN-a razvila je Međunarodna organizacija za standardizaciju (ISO) i objavila 1970. kao međunarodni standard ISO 2108. Poseban ISBN je dodijeljen svakom izdanju i varijaciji knjige (osim reprinta). Deseteroznamenasti ISBN se dodjeljivao izdanjima prije 2007., a nakon 2007. se dodjeljuje trinaesteroznamenasti.

ISBN se sastoji od 4 ili 5 dijelova:

1. za trinaesteroznamenasti ISBN prvi dio je troznamenasti prefiks: 978 ili 979;
2. oznaka nacionalne, zemljopisne ili jezične grupe – može imati 1 do 5 znamenaka;
3. oznaka izdavača;
4. naslov knjige;
5. kontrolni broj – oblik provjere redundancije koda, koristi se za detekciju neispravnosti koda. Sastoji se od jedne znamenke izvedene od ostalih znamenaka u kodu.

Broj znamenaka u trećoj i četvrtoj skupini ovisi o nakladničkom planu, odnosno broju naslova koje nakladnik namjerava objaviti.



Slika 5. Dijelovi deseteroznamenastog ISBN koda, pripadajući EAN-13 i barkod



Kontrolni broj se računa prema slijedećoj formuli:

$$(10x_1 + 9x_2 + 8x_3 + 7x_4 + 6x_5 + 5x_6 + 4x_7 + 3x_8 + 2x_9 + x_{10}) \bmod 11 \equiv 0$$

Da bi ovaj uvjet bio zadovoljen, vrijednost broja  $x_{10}$  može biti jednaka 10; u tom slučaju se stavlja 'X'.

Dvije najčešće pogreške u radu s ISBN-om (npr. pri unosu) su zamjena susjednih znamenaka ili pogrešan unos neke od znamenki. Kontrolni broj osigurava da će ove pogreške biti otkrivene.

### ISBN-13

Od 1. siječnja 2007. oznaka međunarodnog standardnog knjižnog broja promijenila se iz deseteroznamenaste u trinaesteroznamenastu tako da je deseteroznamenasta oznaka dobila troznamenasti prefiks 978 i novi kontrolni broj. Na taj način međunarodni standardni knjižni broj u potpunosti je usklađen s crtičnim kodom EAN-13 (European Article Numbering - europsko označavanje proizvoda brojevima). Izdavači koji su se uključili u sustav ISBN prije 1. siječnja 2007. ne smiju koristiti deseteroznamenaste ISBN oznake koje su im dodijeljene prije tog datuma, već ih moraju preračunati u trinaesteroznamenaste ISBN oznake.

Kontrolni broj za ISBN-13 izračunava se prema slijedećoj formuli:

$$x_{13} = (10 - (x_1 + 3x_2 + x_3 + 3x_4 + \dots + x_{11} + 3x_{12})) \bmod 10 \bmod 10$$

Ovaj način provjere ne može uhvatiti sve pogreške zamjene znamenki.

### AMS (MSC)

MSC (Mathematics Subject Classification, hrv. Klasifikacija matematičkih tema) je alfanumerička metoda klasifikacije. Koristi se za mnogo matematičkih časopisa koji traže autore istraživačkih radova da navedu kodove tema iz *klasifikacije matematičkih tema* u svojim radovima. Trenutna verzija je MSC2010.

MSC je hijerarhijska shema sa strukturom od tri nivoa. Klasifikacija može biti dvo-, tro- ili peteroznamenasta, ovisno o tome koliko nivoa klasifikacijske sheme se koristi.

Prvi nivo je prikazan dvoznamenkastim brojem, drugi slovom, a treći nekim drugim dvoznamenkastim brojem. Primjer:

- **53** je klasifikacija za diferencijalnu geometriju
- **53A** je klasifikacija za klasičnu diferencijalnu geometriju
- **53A45** je klasifikacija za vektorsku analizu

## UDK

UDK (Univerzalna decimalna klasifikacija) je način klasifikacije knjiga koji su razvili Belgijski knjižničari Paul Otlet i Henri La Fontaine krajem 19. stoljeća. Ovo je najistaknutija svjetska višejezična shema klasifikacije za sva područja znanja, sofisticirani alat za indeksiranje i pristup podacima. UDK je bio modificiran i proširen tokom godina da bi izašao na kraj sa svim disciplinama ljudskog znanja i dan danas se konstantno pregledava da bi uzeo u obzir nova otkrića. Dokumenti klasificirani s UDK mogu biti u bilo kojem obliku. Najčešće će biti neki oblik literature (tj. pisani dokumenti), ali mogu biti i ostali mediji poput filma, video i audio zapisa, ilustracija...

UDK klasifikacija koristi arapske brojeve i bazirana je na dekadskom sistemu. Radi olakšanog čitanja nakon svake treće znamenke postoji prekid. Tako nakon 61 – „medicinske znanosti“ dolazi podvrsta 611 do 619; pod 611 „Anatomija“ dolazi njegoova podvrsta 611.1 do 611.9; nakon 611.1 dolaze njegove podvrste prije nego se pojavi 611.2 i tako dalje. Prednost ovog sistema je da je beskonačno ekstenzibilan i pojava novih vrsta neće ometati postojeću alokaciju brojeva.

## 1.5. Prednosti i nedostaci elektroničkih knjižnica

### Prednosti:

1. Nema fizičke granice; korisnik digitalne knjižnice ne mora fizički ići u knjižnicu, ljudi diljem svijeta imaju pristup istim informacijama dok god je Internet dostupan,
2. Nema radnog vremena; pristup u bilo koje vrijeme dana ili noći,
3. Višestruki pristup; istim izvorima može pristupati više korisnika istodobno,
4. Strukturirani pristup; pristup bogatijem i strukturiranom sadržaju, kretanje poveznica i hipertekstovima,
5. Pronalazak informacija; pronalazak pomoću pretraživanja i upisivanja jedne riječi ili skupa riječi,
6. Umnožavanje i očuvanje; može se napraviti identična kopija bez razlike u kvaliteti,
7. Prostor; dok klasičnim knjižnicama treba skladište, digitalne knjižnice mogu pospremiti puno veći broj informacija jer treba jako malo fizičkog prostora,
8. Umrežavanje; digitalna knjižnica može prosljediti link koji vodi drugoj digitalnoj knjižnici, dakle koristi se i dijeljenje arhiva,
9. Cijena; održavanje digitalne knjižnice je jeftinije od održavanja klasične knjižnice, samo je proces digitalizacije malo skuplji.

**Nedostaci:**

1. Ograničenost; autorsko pravo je kompleksna stavka koja strogo ograničava izbor materijala za digitalizaciju pa mnoge kulturne ustanove izbjegavaju kompleksnosti zakona o autorskom pravu digitalizirajući samo materijale koji nisu zaštićeni autorskim pravom i pripadaju u tzv. javnom vlasništvu. Ta praksa znači da mnoge zbirke knjižnične građe nisu opsežno digitalizirane. Bitan korak u procesu odabira materijala za digitalizaciju je upoznavanje sa svime što je pokriveno autorskim pravom.
2. Visoki početni troškovi; troškovi hardvera su prilično visoki,
3. Propusnost; elektroničke knjižnice trebaju vrlo visoku propusnost za transfer multimedijalnih sredstava, ali propusnost se povećava svakim danom.
4. Efikasnost; radi povećanog obujma digitalne informacije, pronalaženje pravog materijala za određeni zadatak postaje sve teže
5. Ozračje; elektroničke knjižnice ne mogu proizvesti ozračje tradicionalnih knjižnica. Mnogim ljudima je lakše čitati tiskani materijal nego na kompjuterskom zaslonu.
6. Održavanje; zbog ubrzanog tehnološkog razvoja, elektroničke knjižnice mogu jako brzo postati zastarjele i njihovi podaci mogu postati nedostupni.

**1.6. Budućnost**

Elektroničke knjižnice su mnogo fleksibilnije nego tradicionalne. Prenosive su, dostupne u bilo koje vrijeme i na bilo kojem mjestu – kod kuće, u avionu, na plaži ili na ulici. Ekstenzivne su, daju pristup osobnoj zbirci knjiga, gradskoj javnoj knjižnici i sveučilišnoj knjižnici. Štoviše, bit će integrirane sa nacionalnim i međunarodnim izvorom informacija. Pružat će pristup svjetskom znanju i definitivno neće biti statične.



Slika 6. Čitaonica narodne knjižnice New Yorka



Slika 7. Čitaonica elektroničke knjižnice

Unutar računala, vaša knjižnica će biti složena onako kako vi to želite i personalizirana za osobne stvari kojima se inače bavite. Izvana može izgledati kao na slici 6. (jednostavno donesite laptop i priključite ga) ili kao na slici 7. (sa već postavljenim kompjuterima). U svakom slučaju, postojat će sučelje – moguće nekakav zaslon, premda u će to u budućnosti možda jednostavno biti unutar vašeg zida ili vaših naočala; možda će imati tipkovnicu, premda i to može biti zamijenjeno zaslonom osjetljivim na dodir ili mikrofonom; moguće da će imati miša, iako i to može biti zamijenjeno zaslonom osjetljivim na dodir. Ali to sučelje će se smatrati knjižnicom više nego računalom. Računala će postati nevidljiva, baš kao što su i bezbrojni elektromotori u vašim kućanstvima nestali u sušila za kosu, ventilatore i razno razne uređaje.

Osim što će biti personalizirane, bit će i dinamičke. Ne samo u smislu da će uvijek posjedovati svježije informacije, nego će i raditi usporedo s vama – pratit će vaše aktivnosti i reagirati samostalno stavljajući ono što biste mogli zatrebati u kontekstu onoga što trenutno radite tamo gdje to možete vidjeti i pročitati. Kada završite s danom, nastavit će raditi za vas, locirati bitne informacije, klasificirati i kategorizirati ih, postavljati implikacije i istraživati u korist vama, tako da idućeg dana možete početi tamo gdje ste stali.

Današnje mobilne elektroničke knjižnice trebaju mrežni pristup serverima elektroničke knjižnice. Umjesto toga, zamislite mogućnost nošenja elektroničke knjižnice bilo gdje sa sobom. Tokom pretraživanja i pregledavanja, možete imati potpuni pristup milionima podataka – tekst, glazba ili video – gdje god da se nalazite, bez potrebe za pristup internetu i čekanjem da se sadržaj s weba transferira u vaš uređaj. Sve putuje s vama, bez restrikcija u avionima, bez razmišljanja o povezivosti u drugim zemljama.

## Poglavlje 2.

### Baze podataka

Od samog početka razvoja računala i informatike, obrada podataka bila je jedan od najzanimljivijih i najznačajnijih segmenata tog razvoja. Potrebe za što sveobuhvatnijom obradom podataka, jednostavnom manipulacijom, te što bržim i sigurnijim pristupom podacima, dovele su do razvoja nekoliko modela podataka i čitavog niza programskih paketa za upravljanje podacima. Ovo poglavlje pruža uvid u osnove danas najdominantnijeg, relacijskog modela podataka, način modeliranja entiteta i veza, normalizaciju baze, te ukratko opisuje najpopularniji jezik za rad s bazama – SQL i jezik koji je korišten za bazu podataka iz praktičnog dijela ovog rada - SQLite

#### 2.1. Što su baze podataka

Baza podataka je organizirana zbirka podataka. Termin je izvorno nastao unutar računalne industrije, a njegovo se značenje proširilo popularnom upotrebom toliko da Europska direktiva za baze podataka (koja za baze podataka donosi prava za intelektualno vlasništvo) uključuje i neelektronske baze podataka unutar svoje definicije.

Baza podataka je skup međusobno povezanih podataka, pohranjenih u vanjskoj memoriji računala. Podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima. Ubacivanje, promjena, brisanje i čitanje podataka obavlja se putem posebnog softvera, takozvanog sustava za upravljanje bazom podataka (DBMS-a). Korisnici i aplikacije pritom ne moraju poznavati detalje fizičkog prikaza podataka, već se referenciraju na neku idealiziranu logičku strukturu baze.

Središnji koncept baze podataka je jednak onome zbirke zapisâ ili dijelova znanja. Za danu bazu podataka tipično postoji strukturni opis vrste činjenica sadržanih u toj bazi podataka: taj opis naziva se shema. Shema opisuje predmete koji su prikazani u bazi podataka i odnose među njima. Postoje brojni načini organiziranja sheme, to jest modeliranja strukture baze podataka: oni se zovu modeli baza podataka (ili modeli podataka). Model u najraširenijoj upotrebi danas je relacijski model, koji se temelji na matematičkom pojmu relacije, a podaci i veze među podacima prikazane su pomoću „pravokutnih“ tablica. Ovaj model prikazuje odnose upotrebom vrijednosti koje su zajedničke za više od jedne tablice. Ostali modeli, poput hijerarhijskog i mrežnog modela, koriste prikaze i odnose koji su eksplicitniji.

Naziv *baza podataka* se strogo govoreći odnosi na zbirku zapisa, a na softver bi se trebalo odnositi kao na *sustav za upravljanje bazom podataka* ili DBMS. Sustav za upravljanje bazom podataka oblikuje fizički prikaz baze u skladu s traženom logičkom strukturom. On u

ime klijenta obavlja sve operacije s podacima i u stanju je podržati razne baze od kojih svaka može imati svoju logičku strukturu, ali mora biti u skladu s istim modelom. Također, brine se za sigurnost podataka, te automatizira administrativne poslove s bazom.

Ipak, kada je kontekst nedvojbjen, mnogi programeri i administratori za baze podataka ipak koriste termin *baza podataka* da pokriju oba značenja.

Ciljevi koji se nastoje postići korištenjem baza podataka:

- Fizička neovisnost podataka
- Logička neovisnost podataka
- Fleksibilnost pristupa podacima
- Istovremeni pristup do podataka
- Čuvanje integriteta
- Mogućnost oporavka nakon kvara
- Zaštita od neovlaštenog korištenja
- Zadovoljavajuća brzina pristupa
- Mogućnost podešavanja i kontrole

## 2.2. Modeliranje entiteta i veza

U stvarnim situacijama teško je direktno pogoditi relacijsku shemu. Zato postoji pomoćna faza koja se zove modeliranje entiteta i veza (Entity-Relationship Modelling, tzv. ER-shema). U ovoj fazi oblikuje se konceptualna shema koja predstavlja apstrakciju realnog svijeta. Ta ER shema se dalje automatski pretvara u relacijsku. U ovoj fazi svijet se promatra kroz tri kategorije:

**entiteti:** skup objekata koji su nam od interesa;

**veze:** odnosi između entiteta koji su nam od interesa;

**atributi:** svojstva entiteta i veza koja su nam od interesa.

**Entitet** je nešto o čemu želimo spremati podatke, može postojati ili ne postojati. Sve ono što se može jednoznačno identificirati. To može biti stvar, biće, događaj ili pojava (npr. knjiga, autor, student, auto ...).

**Atribut** je obilježje, odnosno svojstvo, entiteta (npr. naziv knjige, broj stranica, ime i prezime autora, ...). Ako neki atribut zahtijeva svoje atribute, onda ga se treba smatrati novim entitetom (npr. izdavač). Isto vrijedi i za atribut koji može istovremeno imati više vrijednosti (npr. izdanje). Ime entiteta, zajedno s pripadnim atributima, određuje **tip** entiteta. Može postojati mnogo **primjeraka** entiteta zadanog tipa (npr. AUTOR je tip čiji su primjerci August Šenoa, Dobriša Cesarić, ...).

**Kandidat za ključ** je atribut, ili skup atributa, čije vrijednosti mogu identificirati entitet. Ne mogu postojati dva različita primjerka entiteta istog tipa s istim vrijednostima kandidata za ključ (Na primjer za tip entiteta AUTO, kandidat za ključ je atribut REGISTARSKI\_BROJ). Ukoliko jedan tip entiteta ima više kandidata za ključ, tada biramo jednog od njih i proglašavamo ga **primarnim ključem**.

**Veze** se uspostavljaju između dva ili više tipova entiteta (npr veza NAPISAO između AUTOR i NASLOV). Radi se o binarnoj ili k-narnoj relaciji između primjeraka entiteta određenih tipova. U binarnoj vezi razlikujemo slijedeće funkcionalnosti:

- Jedan-naprama-jedan (1:1) – jedan primjerak prvog tipa entiteta može biti u vezi s najviše jednim primjerkom drugog tipa entiteta i obrnuto. Npr. veza JE\_PREDSTOJNIK između tipova entiteta NASTAVNIK i ZAVOD
- Jedan-naprama-mnogo (1:N) – jedan primjerak prvog tipa entiteta može biti u vezi s 0, 1 ili više primjeraka drugog tipa entiteta, no jedan primjerak drugog tipa može biti u vezi s najviše jednim primjerkom prvog tipa. Na primjer veza PREDAJE između tipova entiteta NASTAVNIK i KOLEGIJ
- Mnogo-naprama-mnogo (M:N) – jedan primjerak prvog tipa entiteta može biti u vezi s 0, 1 ili više primjeraka drugog tipa entiteta i obrnuto. Na primjer veza NAPISAO između tipova entiteta AUTOR i KNJIGA

Veza može imati i svoje atribute koji se ne mogu pripisati niti jednom od tipova entiteta (na primjer veza NAPISAO može imati atribut REDNI\_BROJ\_AUTORA).

Ako svaki primjerak entiteta nekog tipa mora sudjelovati u zadanoj vezi, tada kažemo da tip entiteta ima **obavezno članstvo** u toj vezi. Inače tip entiteta ima neobavezno članstvo.

Do sada su obrađene samo binarne veze, tj. one koje povezuju dva različita tipa entiteta, i takve veze su najčešće i najpoželjnije zbog jednostavnosti. Ali postoje situacije kada binarne veze nisu dovoljne, tada su nam potrebne složenije veze. Slijedi opis tri vrste složenih veza:

- Involuirana veza - povezuje jedan tip entiteta s tim istim tipom. Npr. tip entiteta OSOBA i veza U\_BRAKU\_S. Stanje ove veze opisuje se kao skup uređenih parova primjeraka entiteta istog tipa koji su povezani. Funkcionalnost takve veze opet može biti 1:1, 1:M, odnosno M:M.

- Pod-tip i nad-tip entiteta - Tip entiteta  $E_1$  je pod-tip tipa entiteta  $E_2$  ako je svaki primjerak od  $E_1$  također i primjerak od  $E_2$ . Pritom  $E_1$  nasljeđuje sve atribute od  $E_2$ , no  $E_1$  može imati i dodatne atribute. Npr. tip entiteta OSOBA ima atribute IME, PREZIME, DATUM\_ROĐENJA,... i tip entiteta AUTOR koji sadrži dodatne atribute poput MR\_AUTHOR\_ID i slično. Između ta dva tipa entiteta stavimo vezu JE. Tada je OSOBA nad-tip entiteta AUTOR, odnosno AUTOR je pod-tip entiteta OSOBA
- Ternarna veza - uspostavlja se između tri tipa entiteta. Stanje ternarne veze opisuje se kao skup uređenih trojki primjeraka entiteta koji su trenutno povezani. Svojstva ternarne veze teže je opisati nego kod binarne veze, budući da postoji više vrsta funkcionalnosti, te više kombinacija za obaveznost članstva. Za istu vezu mogu se promatrati tri kardinalnosti, tako da se na jedan od tri načina odaberu dva od tri tipa entiteta, fiksiraju se primjerci entiteta odabranih dvaju tipova, te se gleda broj primjeraka trećeg tipa koji su u vezi s dva fiksirana primjerka.

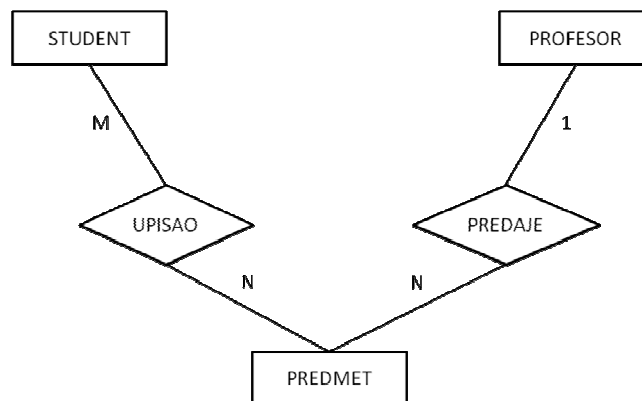
### Prikaz ER sheme pomoću dijagrama

ER shema obično se prikazuje pomoću dijagrama. Simboli u ER-dijagramima su:

entitet – pravokutnik

veza – romb

Veze su povezane bridovima s odgovarajućim tipovima entiteta. U dijagramu na slici 8. prikazani su imena tipova entiteta i veza, te funkcionalnost veze. Također se može prikazati i obaveznost članstva.



Slika 8. Primjer ER dijagrama



## 2.3 Relacijski model

U relacijskom modelu baza podataka se sastoji od skupa tablica ili relacija. Pojedini stupac relacije sadrži vrijednost određenog atributa za entitet ili vezu, pa se stupac povezuje s atributom i obratno. **Domena atributa** je definirani skup dozvoljenih vrijednosti za atribut. Jedan redak relacije obično predstavlja jedan primjerak entiteta ili bilježi vezu između dva ili više primjerka i naziva se **n-torka**. U relaciji ne smiju postojati dvije jednake n-torke. Broj atributa je **stupanj** relacije, a broj n-torki **kardinalnost** relacije. Npr. relacija AUTORI s atributima ID, IME i PREZIME ima sljedeću tablicu:

Tablica 2. Relacija AUTORI

ID	IME	PREZIME
1	August	Šenoa
2	Dobriša	Cesarić
3	Ivan	Kozarac

Relacija ne propisuje redosljed svojih n-torki ili atributa. Permutacijom stupaca ili redaka dobiva se drugi zapis, ali ista relacija.

Uvedena terminologija potječe iz matematike. Naime, neka je  $R$  relacija stupnja  $n$  i neka su domene njenih atributa redom  $D_1, D_2, \dots, D_n$ . Tada se  $R$  može interpretirati kao podskup Kartezijevog produkta  $D_1 \times D_2 \times \dots \times D_n$ . Dakle, pojam relacije odgovara matematičkom pojmu  $n$ -arne relacije.

Terminologija u DBMS, matematici i programskim jezicima:

Tablica 3. Terminologija u DBMS, matematici i programskim jezicima

DBMS	Relacijski račun	Programski jezici
tablica	relacija	datoteka
redak	n-torka	zapis
stupac	atribut	polje

**Primarni ključ** je atribut ili skup atributa koji jedinstveno identificiraju svaki element entiteta (redak utablici). Primarni ključ mora zadovoljavati tri osnovna uvjeta:

1. **Jedinstvenost** - U tablici ne mogu postojati dva redka s istom vrijednošću primarnog ključa
2. **Minimalnost** - Ako je primarni ključ složen tj. sastoji se od više atributa, tada se niti jedna njegova komponenta nemože ukloniti a da se ne naruši pravilo jedinstvenosti.
3. **Pravilo integriteta primarnog ključa** - Niti jedna komponenta primarnog ključa ne smije imati null vrijednost tj. atribut koji je sastavni dio primarnog ključa mora imati definiran kardinalitet (1,1). Pravilo integriteta posljedica je pravila minimalnosti. Kada bi bilo dopušteno da neki atribut kao dio primarnog ključa poprimi null vrijednost, to bi značilo da se elementi entiteta mogu identificirati i bez poznavanja vrijednosti tog atributa. Iz toga proizlazi da takav atribut uopće ne treba biti dio primarnog ključa, tj. narušilo bi se pravilo minimalnost

## 2.4. Pretvaranje ER sheme u relacijsku

### 1. Pretvorba entiteta i atributa

Svaki tip entiteta prikazuje se jednom relacijom. Atributi entiteta postaju atributi relacije. Jedan primjerak entiteta prikazan je jednom n-torkom. Primarni ključ entiteta postaje primarni ključ relacije. Na primjer, tip entiteta **AUTOR** postaje relacija:

**AUTOR** ( id, ime, prezime, dodaci, . . . )

Također, sudjelovanje entiteta u vezama može zahtijevati da se u relaciju dodaju još neki atributi koji nisu postojali u odgovarajućem tipu entiteta.

### 2. Pretvorba binarnih veza

Pravilo 1. Ako tip  $E_2$  ima obavezno članstvo u N:1 vezi s entitetom  $E_1$ , tada relacija za  $E_2$  treba uključiti primarne attribute (primarni ključ) od  $E_1$ .

Pravilo 2. Ako tip  $E_2$  ima neobavezno članstvo u N:1 vezi s entitetom  $E_1$ , tada vezu možemo prikazati:

- a. na prethodni način uvođenjem ključa, ili
- b. uvođenjem nove relacije čiji su atributi primarni atributi (primarni ključevi) od entiteta  $E_1$  i  $E_2$ .

Pravilo 3. Veze tipa N:M uvijek se prikazuju posebnom relacijom koja uključuje primarne attribute oba entiteta te još možda dodatne koje ima sama veza.

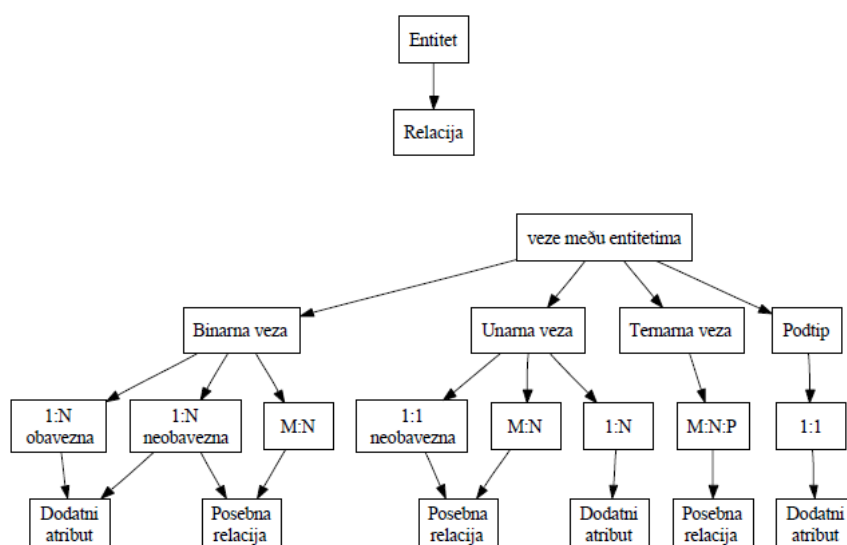
Pravilo 4. Involuiranu vezu tipa 1:1 prikazujemo posebnom relacijom.

Pravilo 5. Involuiranu vezu tipa N:M prikazujemo posebnom relacijom.

Pravilo 6. Involuiranu vezu tipa 1:N možemo prikazati dodatnim atributom.

Pravilo 7. Podtip se prikazuje posebnom relacijom koja sadrži primarne attribute nadređenog tipa, te attribute specifične za taj podtip.

Pravilo 8. Ternarna veza se prikazuje posebnom relacijom koja sadrži primarne attribute svih triju entiteta, te moguće dodatne attribute specifične za vezu.



Slika 9. Skica pretvaranja ER sheme u relacijsku

## 2.5. Normalizacija relacijske sheme

Dobro oblikovana relacijska baza podataka je skup podataka koji:

- nije redundantan
- posjeduje semantički integritet (ne pokazuje anomalije pri unosu, brisanju ili promjeni podataka)

**Redundancija** - govori da se istovrsni podatak memorira na više mjesta (to se smatra nedostatkom jer se obrada redundantnih podataka mora istovremeno obaviti na više mjesta). Posljedično se javljaju anomalije pri unosu, brisanju i promjeni podataka, te se zbog moguće

nekonzistentnosti istog podatka, koji se memorira više puta, mogu pojaviti i neispravni odgovori na zadani upit.

Relacijska shema, dobivena iz ER-sheme na osnovu uputa iz prethodnog podpoglavlja, može sadržavati nedorečenosti koje treba otkloniti prije implementacije. Proces daljnjeg dotjerivanja sheme zove se **normalizacija**. Normalizacija je postupak prevođenja jedne ili skupa relacija iz niže u višu normalnu formu, a služi se operacijama prirodnog spajanja i projekcije. Teorija normalizacije zasnovana je na pojmu **normalnih formi**, od čega praktičnu važnost ima njih pet. Ovdje su nabrojane polazeći od nižih ili blažih prema višim ili strožim normalnim formama:

- Prva normalna forma (1NF)
- Druga normalna forma (2NF)
- Treća normalna forma (3NF)
- Boyce-Coddova normalna forma (BCNF)
- Četvrta normalna forma (4NF)

Slijedi kratak opis svake od normalnih formi:

### **Prva normalna forma**

Relacija se nalazi u prvoj normalnoj formi (1NF) ako je svaki njezin atribut atomiziran, što znači da sadrži samo jednu vrijednost, nikako više vrijednosti. Svi neključni atributi su funkcijski zavisni o primarnom ključu. Ovo pravilo je jednostavno i proizlazi iz definicije primarnog ključa. Budući je primarni ključ jedinstven, tj. ne mogu se pojaviti dvije iste vrijednosti primarnog ključa u jednoj tablici, onda se podrazumijeva da su svi ostali atributi funkcijski ovisni o ključu. Baza podataka je u 1. normalnoj formi, ako su sve tablice u 1. normalnoj formi.

### **Druga normalna forma**

Ova normalna forma rješava problem djelomične funkcijske zavisnosti. Neka su  $X$  i  $Y$  skupovi atributa relacije, relacija zadovoljava potpunu funkcijsku zavisnost  $X \rightarrow Y$  ako je  $Y$  funkcijski zavisan od  $X$ , ali ne i od bilo kojeg podskupa od  $X$ .

Relacija se nalazi u 2. normalnoj formi (2NF) ako se nalazi u 1. normalnoj formi (1NF), i ako su svi neključni atributi potpuno funkcijski zavisni o bilo kojem ključu relacije. Pretvorba relacije iz 1NF u 2NF obavlja se prema pravilima pouzdane dekompozicije. Svi atributi koji djelomično zavise o ključu, zajedno s dijelom ključa po kojem zavise, izdvajaju se u novu relaciju.

### Treća normalna forma

rješava problem kada imamo nepotrebnu (redundantnu) funkcijsku zavisnost. Relacija se nalazi u trećoj normalnoj formi (3NF) ako se nalazi u 2NF i ako niti jedan neključni atribut nije tranzitivno zavisian o bilo kojem ključu relacije. Svaki neključni atribut mora zavisiti o cijelom ključu i ni o čemu drugom. Pretvorba relacije iz 2NF u 3NF obavlja se prema pravilima pouzdane dekompozicije. Izdvajaju se svi neključni atributi između koji postoji funkcijska zavisnost u posebnu relaciju.

U relaciji R koja sadrži attribute X, Y i A, vrijedi da je A **tranzitivno funkcijski zavisian** o atributu X, ako je  $X \rightarrow Y$ ,  $Y \not\rightarrow X$ ,  $Y \rightarrow A$ ,  $A \not\rightarrow Y$ . Simbolički se tranzitivna zavisnost prikazuje:  $X \rightarrow Y \rightarrow A$ .

### Boyce-Coddova normalna forma

**Determinanta** je atribut (ili kombinacija atributa) o kojem je neki drugi atribut potpuno funkcionalno ovisan.

Relacija je u Boyce-Codd-ovoj normalnoj formi (BCNF) ako je svaka njezina determinanta ujedno i kandidat za ključ. Očito je relacija koja je u BCNF također i u 2NF i 3NF. No postoje relacije koje su u 3NF, no nisu u BCNF.

### Četvrta normalna forma i višeznačna zavisnost

Ako su X i Y disjunktni skupovi atributne relacije R, a njihov komplement je  $Z=R-XY$ , relacija R zadovoljava **višeznačnu zavisnost**  $X \twoheadrightarrow Y$  ako uz dvije promatrane n-torke  $xy_1z_1$  i sadrži n-torke  $xy_1z_2$  i  $xy_2z_1$ . Višeznačna zavisnost postoji u relacijama koje opisuju dvije nezavisne veze tipa M:M.

Relacija u kojoj je zadan skup funkcijskih i višeznačnih zavisnosti je u četvrtoj formi (4NF) ako je svaka višeznačna zavisnost  $X \twoheadrightarrow Y$  trivijalna ili je X ključ relacije.

## 2.6. SQL i SQLite

### SQL

Structured Query Language (SQL) je najpopularniji računalni jezik za izradu, pretraživanje, uređivanje i brisanje podataka iz relacijskih baza podataka. Razvoj SQL-a kao jezika za manipulaciju podacima tekao je usporedno s razvojem relacijskog modela podataka, čije je osnove 1971. godine postavio E.F.Codd. Kada je Codd postavio svoj koncept relacijskog modela podataka, ustvrdio je da „... usvajanje relacijskog modela podataka dopušta razvoj

univerzalnog jezika podataka baziranog na primjeni relacijske algebre”. Iako je odmah uočio zahtjeve koje bi takav jezik trebao ispunjavati, kao i njegove prednosti pri manipulaciji podacima, tada nije pokušao razviti takav jezik u praksi.

SQL je uglavnom zasnovan na relacijskom računu, s time da je matematička notacija zamijenjena ključnim riječima nalik na govorni engleski jezik. No lagano se realiziraju i sve operacije iz relacijske algebre. Osim postavljanja upita, jezik također omogućuje: definiranje relacija, ažuriranje relacija (upis, promjena, brisanje n-torki), sortiranje i formatiranje ispisa, neke aritmetičke operacije s podacima, definiranje „pogleda“ (virtualnih relacija izvedenih iz postojećih), utjecaj na fizičku građu baze (na primjer stvaranje tzv. indeksa), te kontrolu sigurnosti.

SQL se dijeli na četiri „podjezika“:

- **Data Definition Language (DDL)** sastoji se od naredbi za definiranje objekata baze podataka: npr. CREATE, DROP i ALTER
- **Data Manipulation Language (DML)** sastoji se od naredbi za manipuliranje podacima u bazi: npr. INSERT, DELETE, UPDATE
- **Data Control Language (DCL)** sastoji se od naredbi za kontrolu podataka i njihovu sigurnost: npr. GRANT i REVOKE
- **Data Query Language (DQL)** sastoji se od naredbi za pretraživanje i dohvaćanje podataka u bazi : npr. SELECT

#### **Najčešće korištene naredbe u SQL-u:**

SELECT naredba ne mijenja niti dodaje i ne briše podatke iz baze podataka. Ona postavlja upit na relaciji te odabire retke ili kolone neke tablice ili više tablica. SELECT naredba se može sastojati od više klauzula, a moraju biti navedene ovim redom:

- SELECT koje kolone iz pojedine tablice ili tablica
- FROM iz koje tablice ili tablica
- WHERE uvjet
- ORDER BY koristimo za izmjenu sortirnog pojma ili redoslijeda

Samo su SELECT i FROM klauzule obvezne, radi kolona i tablica nad kojima treba izvršiti upit. Još su neke SELECT klauzule koje mogu biti dio upita: GROUP BY i HAVING. Može se koristiti i klauzula LIMIT s kojom se ograničava broj pronađenih n-torki. LIMIT se može kombinirati s OFFSET koja regulira koji će skup n-torki biti dobiven DISTINCT koristimo želimo izbjeći višestruko ponavljanje nekog od atributa u upitu.

INSERT naredba umeće novi redak u tablicu:

- INSERT (naziv stupca 1, naziv stupca 2, ..., naziv stupca n)
- INTO naziv tablice
- VALUES (vrijednost 1, vrijednost 2, ..., vrijednost n)

DELETE naredba briše redak ili tablicu

- DELETE
- FROM iz koje tablice ili tablica
- WHERE uvjet

UPDATE naredba mijenja sadržaj pojedinog polja (ili više njih) u tablici

- UPDATE ime tablice
- SET ime stupca = nova vrijednost
- WHERE uvjet

Dvije ili više tablica spajaju se pomoću naredbe JOIN. Ta naredba kombinira tablice pomoću vrijednosti atributa koje su im zajedničke. Kao rezultat kombiniranja tablica dobijemo novu, privremenu tablicu koja se zove *joined table*. Osnovni tipovi JOIN naredbe su INNER JOIN i OUTER JOIN (OUTER JOIN se djeli na LEFT OUTER JOIN, krace LEFT JOIN, RIGHT OUTER JOIN, krace RIGHT JOIN i FULL OUTER JOIN).

INNER JOIN je dio SELECT izraza koji dohvaća zapise iz više tablica i daje jedan skup zapisa kao riješenje upita. Ispisuje sve retke obje tablice koje smo uspjeli povezati istom vrijednosti atributa. Ostale retke ne ispisuje.

LEFT JOIN kao rezultat vraća sve retke iz lijeve tablice i odgovarajuće retke iz desne tablice. Ako neki redak iz lijeve tablice nema odgovarajući redak u desnoj, tada mu se pridružuje NULL vrijednost.

RIGHT JOIN kao rezultatni skup vraća sve retke iz desne tablice i odgovarajuće retke iz lijeve. Ako za neki redak ne postoji odgovarajući, pridružuje mu se NULL redak.

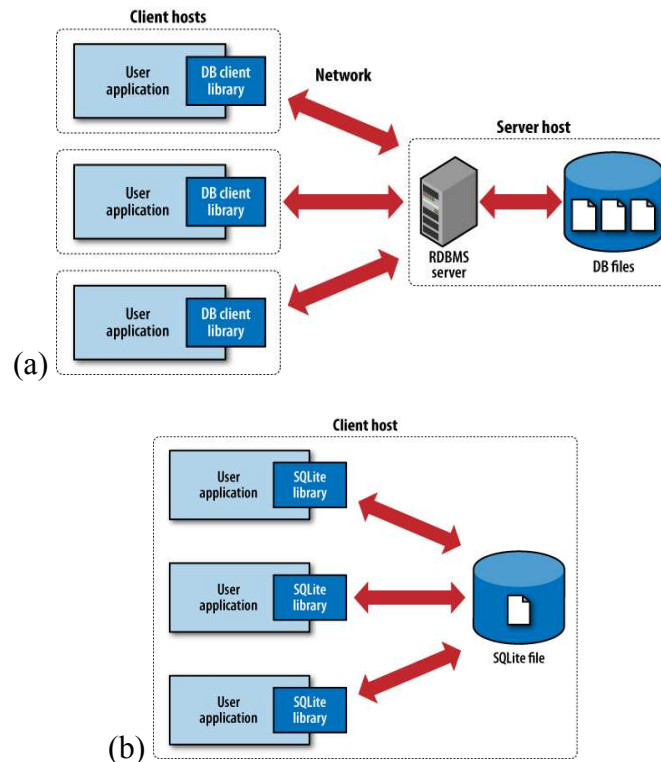
FULL OUTER JOIN vraća sve retke iz lijeve i desne tablice tako da spoji one koji odgovaraju prema kriterijima spajanja, one koji ne odgovaraju dopunjava sa NULL vrijednostima.

## SQLite

SQLite je relacijska baza podataka temeljena na maloj C programskoj biblioteci. Glavne značajke SQLite baze podataka su:

- Nema potrebe za odvojenim poslužiteljem ili sistemom da bi radio. SQLite pristupa direktno svojim podacima
- Nulta konfiguracija; nema potrebu za prilagođavanjem i administracijom.
- Cijela baza podataka smještena u jednoj datoteci na disku.
- Podaci iz baze se mogu slobodno dijeliti između više računala.
- Podržava nekoliko terabajta velike baze i nekoliko gigabajta velike nizove podataka.
- Brža izvedba standardnih operacija od ostalih popularnih baza podataka.
- Jednostavno korisničko sučelje
- Transakcije funkcioniraju po ACID (*atomic, consistent, isolated, and durable*) principu. Podaci ostaju postojani pri padu sustava ili nestanku električne energije.
- Podržava većinu SQL92 standarda.

Sve u svemu, SQLite pruža vrlo funkcionalnu relacijsku bazu podataka koja koristi minimalne resurse i stvara minimalnu gnjavažu za programere i korisnike. „Lite“ u nazivu SQLite se ne odnosi na njegove mogućnosti. Točnije, SQLite je lagan što se tiče složenosti, administracije i upotrebe.



Slika 10. (a) klasična RDBMS klijent/poslužitelj arhitektura, (b) SQLite arhitektura bez poslužitelja



### Primjena SQLite baze podataka

- **Web stranice** – SQLite baza podataka je odlična za korištenje na web stranicama za malu ili srednju količinu prometa. Generalno govoreći idealna je za stranice koje imaju oko 100 000 klikova na dan. Dokazano je da može raditi sa 10 puta većim prometom (1 000 000 klikova na dan).
- **Ugrađeni uređaji i aplikacije** – SQLite baza podataka zahtijeva malo ili nimalo administracije što ju čini idealnim izborom za uređaje ili usluge koje trebaju raditi automatski i bez ljudske podrške. Odlično je usklađena za primjenu u mobilnim telefonima, dlanovnicima, upravljačkim kutijama i ostalima uređajima i napravama.
- **Ad hoc disk datoteke** – Odlična je zamjena za ad hoc disk datoteke koje koriste fopen(), fread() i fwrite() funkcije za kreiranje datoteka i manipulaciju.

Sav programski kod SQLite baze zasnovan je na public domain principu. To znači da se programski kod u cijelosti, kao i njegovi dijelovi, slobodno mogu kopirati, modificirati, objavljivati, koristiti, prodavati i distribuirati u originalnom SQLite formatu, otvorenom kodu ili u kompajliranom obliku za bilo koju svrhu komercijalnu ili ne komercijalnu.

### Nedostaci

Iako je SQLite-ova implementacija SQL-a prilično dobra, postoje neke stvari koje trenutno nisu implementirane:

- ALTER TABLE nije u potpunosti podržan; samo RENAME TABLE i ADD COLUMN naredbe su podržane. Ostale naredbe poput DROP COLUMN, ALTER COLUMN, i ADD CONSTRAINT nisu implementirane
- Ugnježdene transakcije; SQLite dozvoljava samo pojedinačne transakcije. Ugnježdene transakcije dozvoljavaju kontrolu nad većim i kompleksnijim operacijama.
- RIGHT i FULL OUTER JOIN; LEFT OUTER JOIN je implementiran, ali RIGHT OUTER JOIN i FULL OUTER JOIN nisu. RIGHT OUTER JOIN može biti implementiran jednostavnim obrtanjem redoslijeda tablica i modificiranjem „join“ uvijeta. FULL OUTER JOIN može biti implementiran kao kombinacija ostali relacijskih operacija koje SQLite podržava

## Poglavlje 3.

# Web programiranje

Web programiranje je stvaranje računalnog programa čija je funkcionalnost primarno usmjerena na web. Ono uključuje analitičnost zahtjeva, programski dizajn, matematičko – logičku pragmatičnost i poznavanje tehnološke platforme. Web programiranje nije čvrsto vezano uz sklopovlje računala ili operativni sustav, već je prvenstveno okrenuto efikasnosti izvedbe korisničkog zadovoljstva upotrebe aplikacije. Ovo poglavlje pruža uvid u osnove Web-a i web programiranja – što je to WWW, što je HTML, kako se oblikuju stranice (CSS), te kako se rade dinamičke web stranice i pristupa bazi podataka putem web aplikacije (PHP).

### 3.1. World Wide Web

World Wide Web – skraćeno WWW, W3, ili samo Web, naziv je koji dolazi iz engleskog jezika a može se prevesti kao 'svjetska mreža'. To je veliki repozitorij dokumenata i jedna od najkorištenijih usluga Interneta koja omogućava dohvaćanje hipertekstualnih dokumenata. Web dokumenti se spremaju i zapisuju u jeziku koji se zove *HyperText Markup Language* (HTML) (detaljnije u slijedećem poglavlju). Dokumenti mogu sadržavati tekst, slike i multimedijalne sadržaje a međusobno su povezani tzv. hiperlinkovima. Za dohvaćanje i prikaz sadržaja koriste se računalni programi koji se nazivaju web-preglednici. Pregledavanje weba odvija se u potpunosti u skladu sa paradigmom klijent-poslužitelj. Naime, web preglednik je klijent koji traži od odgovarajućeg web poslužitelja da mu pošalje web dokument. Taj web poslužitelj je program koji radi na onom računalu na kojem je pohranjen web dokument. Za komunikaciju web klijenta i web poslužitelja koristi se TCP veza i posebni aplikacijski protokol koji se zove *HyperText Transfer Protocol* (HTTP). Ukoliko web stranica sadrži multimedijske objekte i tekst, za svaki objekt se otvara nova TCP/IP veza.

HTTP protokol je relativno jednostavan. Sastoji se od svega nekoliko naredbi.

- GET: zahtijeva određeni podatak od poslužitelja.
- HEAD: traži informaciju o statusu dokumenta
- PUT: šalje podatke poslužitelju, koje poslužitelj koristi da bi zamjenio određeni dokument.
- POST: šalje podatke poslužitelju, koje poslužitelj dodaje danom dokumentu.

Osim HTTP postoje i drugi protokoli za komunikaciju web klijenata i poslužitelja, na primjer HTTPS koji se koristi za šifriranu komunikaciju. Također, osim HTML postoje i drugi jezici

za zapisivanje dokumenata - takozvani markup jezici - na primjer XML, VXML. Sve su to razlozi zašto današnji web preglednici imaju prilično složenu građu.

Projekt kojim je počeo razvoj Weba predložili su 1990. u CERN-u engleski inženjer i znanstvenik Tim Berners-Lee i belgijski znanstvenik Robert Cailliau.

Web se često pogrešno koristi kao sinonim za Internet. Internet je globalni komunikacijski sustav čiji hardver i softver omogućuje povezivanje putem računala, dok je WWW jedna od usluga koja se koristi putem interneta.

U posljednjih 15-tak godina WWW se intenzivno razvijao te su se u njega uklopile brojne dodatne tehnologije. Zahvaljujući takvom razvoju, današnji web dokumenti ne moraju isključivo biti tekstovi u HTML-u ili nekom sličnom markup jeziku, već također mogu poprimiti složenije oblike. Točnije, današnji web dokumenti dijele se na sljedeća tri tipa:

- **Statički** - klasični web dokumenti zapisani na primjer u HTML-u, koji se nalaze u jednoj datoteci i prilikom svakog pozivanja izgledaju isto.
- **Dinamički** - Web poslužitelj stvara dinamički web dokument prilikom svakog poziva. Dokument se realizira izvršavanjem nekog dodatnog programa na strani poslužitelja čiji ispis poslužitelj šalje klijentu. Koriste se tehnologije poput Common Gateway Interface (CGI) - ASP, JSP, PHP, ....
- **Aktivni** - sastoji se od programa kojeg web poslužitelj pri svakom pozivu šalje pregledniku za lokalno izvršavanje (na strani preglednika). Koriste se tehnologije poput Java, JavaScript, ...

## 3.2. HTML

HTML je kratica za HyperText Markup Language i čini strukturu svake web stranice. Zapis dokumenta u HTML-u samo ugrubo specificira smjernice za prikazivanje dokumenta, te dopušta pregledniku određivanje detalja. HTML predstavlja dogovor o zajedničkom jeziku između autora stranica i pretraživača - komunikacija i razmjena informacija je moguća isključivo ako obje strane govore isti jezik. Svaki web dokument identificira se svojom *Universal Resource Location* (URL) adresom, koja se zapisuje u formatu:

```
protokol://ime_računala/ime_dokumenta
```

HTML je standardizirani jezik, za što se brine W3C konzorcij.

## Osnovna struktura HTML dokumenta

HTML dokument ima ekstenziju .html

Pri pisanju HTML dokumenta, moraju se poštivati određena pravila:

- 1) Sve HTML naredbe moraju biti unutar oznaka <...>

Sve što se nalazi unutar tih oznaka tretira se kao naredba HTML-u i neće biti vidljivo u web pregledniku. Sve što je izvan tih oznaka vidljivo je u web pregledniku. Npr.

```
<b>Podebljani tekst</b>
```

Većina naredbi ima oznaku za početak i kraj. Oznaka za kraj je uvijek znak /.

- 2) Svaki HTML dokument mora imati određenu strukturu:

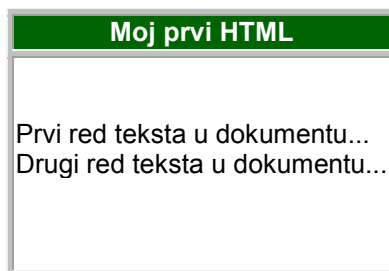
- redak s informacijom o HTML verziji
- zaglavlje <HEAD> </HEAD> unutar kojeg se upisuju opće definicije, poput kodne stranice, podatka o autorstvu, opisa dokumenta i dr.
- tijelo <BODY> </BODY> unutar kojeg se upisuje sve ono što treba biti vidljivo u web pregledniku
- cjelina koju čine deklaracija zaglavlja i tijelo treba biti obuhvaćena elementom HTML koji ima oznaku početka <HTML> i oznaku kraja </HTML>

Primjer jednog HTML dokumenta s objašnjenjem:

(HTML naredbe označene su crvenom bojom)

Izvorni kod HTML-a (vidljiv u tekst editoru)	Objašnjenje
<pre>&lt;!DOCTYPE HTML PUBLIC "-// //W3C//DTDHTML4.01//EN" "http://www.w3.org/TR/html4/strict.dtd"&gt; &lt;HTML&gt; &lt;HEAD&gt; &lt;TITLE&gt;Moj prvi HTML&lt;/TITLE&gt; &lt;/HEAD&gt; &lt;BODY&gt; Prvi red teksta u dokumentu...&lt;BR&gt; Drugi red teksta u dokumentu... &lt;/BODY&gt; &lt;/HTML&gt;</pre>	<p>Redak s informacijom o HTML verziji</p> <p>početak HTML-a</p> <p>početak zaglavlja</p> <p>naziv dokumenta</p> <p>kraj zaglavlja</p> <p>početak tijela</p> <p>tekst koji će biti vidljiv u browseru</p> <p>&lt;BR&gt; prekida red</p> <p>kraj tijela</p> <p>kraj HTML-a</p>

Rezultat koji se vidi u pregledniku:



### 3.3. CSS

CSS je kratica za *Cascading Style Sheets*. Radi se stilskom jeziku, koji rabimo za opis prezentacije dokumenta napisanog pomoću HTML jezika.

Kako se web razvijao, prvotno su u HTML ubacivani elementi za definiciju prezentacije (<font> oznaka itd.), ali je dovoljno brzo uočena potreba za stilskim jezikom koji će HTML osloboditi potrebe prikazivanja sadržaja (što je prvenstvena namjena HTML-a) i njegovog oblikovanja (čemu nam danas služi CSS). Drugim riječima, stil definira kako prikazati HTML elemente. Konkretno, pomoću CSS stilskog jezika se definira izgled i raspored web stranice. Dakle, pomoću CSS-a može se definirati, između ostalog, izgled, veličina i boje slova (font), izgled, definiciju i sve ostale elemente sadržaja, koji je napisan u HTML ili nekom drugom programskom jeziku. Na taj način se poboljšava pristup sadržaju, dobija se veća fleksibilnost i praktički apsolutna kontrola u specifikacijama prezentacije sadržaja. Prednost CSS-a je mogućnost „pozivanja“ različitih stilova za pojedine elemente web stranice. Radi se o takozvanim pravilima, preko kojih CSS određuje prioritete i kreira „shemu“ za stilove i pravila.

CSS je prvi puta uveden kao jedan od standarda 1996. godine, iako se prvi „tragovi“ pojavljuju još 1970. godine. Naravno, bilo je, kao i uvijek kada je web razvoj u pitanju, određenih problema, poglavito s prikazom u različitim preglednicima, ali se standardizacijom to uglavnom dovelo na zadovoljavajuću razinu.

Trenutačno postoje tri „razine“ CSS-a; CSS 1, CSS 2 i CSS 3. Treba naglasiti da je CSS 2 najviše upotrebljavan, CSS 1 polako odlazi u povijest, dok je CSS 3 trenutačno u fazi svojevrsnog testiranja. CSS 2 je još 1998. godine definiran kao jedan od standarda i objavljen je kao preporuka.

U današnje vrijeme razvoja web aplikacija i Internet stranica nemoguće je zamisliti profesionalnu web stranicu koja u sebi nema CSS zato što je preporuka da se definicije elemenata sadržaja određuju upravo kroz CSS, u suprotnom stranica neće proći provjeru valjanosti, niti HTML, a niti, naravno, CSS.

CSS se može pisati unutar same HTML stranice na dva načina.:

- kao stilove u zaglavlju dokumenta

```
<style type="text/css">
  h1 { color: blue }
</STYLE>
```

- unutar samih HTML tagova, npr. `<p style="color: red;">Crveni tekst</p>` što daje: **Crveni tekst**

Ili se može definirati unutar posebnog dokumenta i rabiti pomoću poziva

- `<link rel="stylesheet" href="layout.css" type="text/css">` ili
- `<style type="text/css">`  
`@import url(http://www.neki.url/neki_stil);`  
`</style>`

### Anatomija CSS elementa

CSS element se sastoji od (slika 11 Slika 11. Anatomija CSS elementa):

#### 1) deklaracijskog bloka

kod koji dolazi iza imena elementa uključuje sve mogućnosti oblikovanja koje se primjenjuju na ime elementa. Blok počinje s otvorenom vitičastom zagradom ({} i završava sa zatvorenom vitičastom zagradom ({}))

#### 2) deklaracije

između otvorene i zatvorene vitičaste zagrade deklaracijskog bloka dolazi jedna ili više deklaracija ili instrukcija za oblikovanje. Svaka deklaracija ima dva dijela: atribut i vrijednost, te završava znakom točka-zarez (;)

#### 3) selektora

daje uputstva web pregledniku koji element na web stranici treba oblikovati (npr, naslov, odlomak, link, sliku, tablicu, formu i sl...)

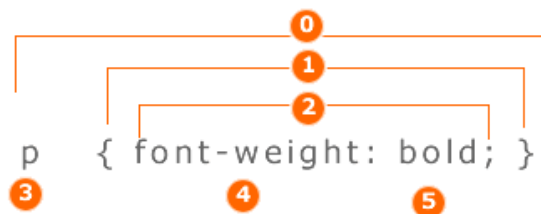
#### 4) atributa

CSS nudi mnogo različitih mogućnosti oblikovanja koje se zovu atributi. Atributi definiraju određeni stilski efekt.

#### 5) vrijednosti atributa

definira kako će se oblikovati određeni atribut; različiti atributi zahtijevaju različite tipove vrijednosti, npr.: mjerne duljine > 12px, 2em ili 1in; boja > *black* ili #000000; URL > slike/slika-1.jpg) ili specifična ključna riječ > *top, center, left*)

0. Pravilo (*Rule set, Rule*)
1. Deklaracijski blok (*Declaration block*)
2. Deklaracija (*Declaration*)
3. Selektor (*Selector*)
4. Atribut (*Property*)
5. Vrijednost atributa (*Value*)



Slika 11. Anatomija CSS elementa

### 3.4. PHP

PHP (Hypertext Preprocessor) je „open-source server-side“ skriptni programski jezik za dinamičko generiranje HTML koda. Drugim riječima, PHP je skriptni programski jezik pomoću kojeg možete kreirati HTML stranicu na serveru prije nego što je ona poslana klijentu popunjena dinamičkim sadržajem. Ovim načinom generiranja sadržaja klijent ne može vidjeti kod (skriptu) koji je generirao prikazani sadržaj, već ima pristup čistom HTML kodu. „Open-source“ u gornjoj definiciji znači da svatko tko želi može skinuti izvorne PHP kodove pisane u C-u i, ukoliko ih razumije, može ih mijenjati po svojoj volji te dodavati nove funkcije PHP-u. PHP je jedna od najnaprednijih i najkorištenijih server-side skriptnih tehnologija danas u upotrebi. On je svojom sintaksom sličan mnogim drugim sličnim jezicima, čak i ima istoznačne (iste po sintaksi i funkcionalnosti) funkcije kao i neki drugi jezici kao što su C ili Perl. To znači da se jedna radnja može izvesti korištenjem više različitih funkcija. Ono što PHP stavlja još više ispred ostalih web skriptnih tehnologija je njegova podrška za baratanje širokom paletom baza podataka. Podržava sve popularnije baze podataka kao MySQL, SQLite, PostgreSQL, dBase, Oracle, ODBC...

Sav PHP kod mora se nalaziti unutar oznake <body> u HTMLu (osim ako nije drugačije navedeno) i nalazi se unutar oznaka: <?php code ?> ili <? code ?>. Te oznake su znak interpreteru na serveru da se taj dio koda mora izvršiti.

Za razliku od C-a, u PHP-u nije potrebno deklarirati tip varijable. PHP automatski određuje tip varijable pri njenoj inicijalizaciji. Oznaka \$ ispred imena varijable označava da se radi o varijabli. Npr:

```
$mojTekst = "tekst"
```

ili

```
$broj = 150
```

Tekstualne varijable se pišu pod navodnicima

Podaci se u PHP-u dohvaćaju preko posebnih globalnih varijabli (npr. GET, POST, REQUEST, FILE).

PHP se može koristiti na 3 načina:

- Skriptiranje sa strane poslužitelja – ovo je njegova prvenstvena namjena (izrada dinamičnih web stranice).
- Naredbeno skriptiranje – znači da može izvoditi skripte iz naredbene linije (slično Unix shell-u).
- Skripte se mogu koristiti za poslove administracije sustava. GUI aplikacije sa strane klijenta – koristeći se PHP-GTK-om (gtk.php.net) na PHP-u se možmo napisati potpuno razvijene GUI (graphical user interface) aplikacije.

Može se koristiti na većini operacijskih sustava, od Unix-a do Windows-a i Mac OS X-a.

### PHP sesijske varijable

Sesije (sessions) su vrlo bitna stvar pri izradi Internet aplikacija i služe kao mehanizam za pamćenje vezanih uz jedan posjet korisnika našoj aplikaciji, odnosno Web site-u. Sesijske varijable koriste se kada je potrebno prenijeti podatke s jedne stranice na drugu. Normalna HTML stranica to ne može napraviti, jer se svi podaci s jedne stranice izgube s učitavanjem nove stranice. To čini problem kod stranica poput online trgovine gdje se informacije o proizvodima koje je korisnik označio moraju prenositi s jedne stranice na drugu.

PHP sesije rješavaju ovaj problem tako što spremaju informacije na server za kasniju upotrebu (npr. korisničko ime, lozinka, proizvodi iz košarice, ...). Sesijske varijable su privremene i postoje sve dok je korisnik na site-u ili ako negdje u kodu nije postavljeno da budu izbrisane. Kada korisnik prestane otvarati stranice na site-u PHP nakon zadanog vremena briše sve informacije koje su spremljene tokom sesije na čvrsti disk servera radi uštede prostora.

Sesije rade tako da kreiraju jedinstveni identifikacijski broj (UID) za svakog posjetitelja i prema varijable bazirano na tom identifikacijskom broju.

Prije početka spremanja informacija u sesijske varijable, potrebno je najprije započeti sesiju. To se radi na početku koda prije nego što je bilo koji HTML ili text unesen.

```
<?php
    session_start(); // pokretanje PHP sesije
?>
```

Ovaj mali dio koda registira korisnikovu sesiju sa serverom i omogućuje spremanje korisničkih informacija i dodjelu UID-a za sesiju tog korisnika. Kada se podaci žele spremiti



u sesiju, koristi se `$_SESSION` asocijativno polje. Iz tog polja se također mogu i dohvatiti podaci.

```
<?php
    session_start();
    $_SESSION['views'] = 1; // spremanje podataka
    echo „Pageviews = „. $_SESSION['views']; //dohvaćanje podataka
?>
```

Postojanje sesijske varijable može se provjeriti pomoću funkcije `isset()`, ta funkcija provjerava da li je varijabla postavljena, odnosno da li joj je dodjeljena određena vrijednost.

```
<?php
    session_start();
    if(isset($_SESSION['views']))
        $_SESSION['views'] = $_SESSION['views']+ 1;
    else
        $_SESSION['views'] = 1;

    echo „views = „. $_SESSION['views'];
?>
```

Pri prvom pokretanju gornje skripte na novootvorenom Web pregledniku, if uvjet neće proći jer još ništa nije spremljeno u tu varijablu.

Iako su sesijske varijable privremene i automatski se brišu nakon nekog vremena, mogu se također i ručno obrisati pomoću funkcije `unset()` koja služi za brisanje jedne varijable ili pomoću funkcije `session_destroy()` koja briše cijelu sesiju.

## Poglavlje 4.

### Praktični dio diplomskog rada

Nakon što je u prethodnim poglavljima objašnjeno što su elektroničke knjižnice, te obrađene teme iz osnova baza podataka i web programiranja, slijedi opis modela - struktura modela, izgled baze, opis tablica, dijelovi koda, te preduvjeti za instalaciju.

#### 4.1. Struktura modela

Model je podijeljen na logične presjeka i slojeve. Presjeci grupiraju funkcionalnosti s obzirom na korisnika. Svaki presjek je dalje podijeljen na slojeve koji pružaju funkcionalnosti na različitim nivoima apstrakcije. Općenito, funkcionalnosti viših razina oslanjaju se na funkcionalnosti nižih, a razlike u apstrakcijama između slojeva mogu biti objašnjeni na primjeru podatkovnog sloja naspram sloja poslovne logike. U pravilu, podaci su pohranjeni u tablice, ali tablice ne diktiraju niti objašnjavaju kako bi podaci trebali biti interpretirani, kombinirani ili kako se njima upravlja. Podatke interpretiraju i njima upravljaju procedure u slojevima poslovne logike.

Unutar ovog modela postoje tri presjeka:

- Presjek poslovne logike – pruža funkcionalnosti za poslovnog korisnika. To je najkompleksniji presjek i sadrži najviše funkcionalnosti modela
- Presjek sigurnosne logike – pruža funkcionalnosti za dokaz autentičnosti i autorizaciju korisnika pri pokušaju pristupa funkcionalnostima modela
- Presjek administracije – ovaj presjek je referenciran samo zbog cjelovitosti. U modelu nije implementirana niti jedna dodatna administracijska funkcionalnost. (Misli se na administraciju modela, ne na administraciju baze)

Svaki presjek podijeljen je na tri sloja:

- Sloj podataka – sačinjavaju tablice koje sadrže podatke. Također, ovaj sloj sadrži relacijska ograničenja između podataka.
- Sloj poslovne logike – sadrži poslovnu logiku svakog sloja. U pravilu je implementiran pomoću procedura i funkcija u PHP-u i SQLite-u.
- Sloj korisničkog sučelja – ovaj sloj sadrži funkcionalnosti potrebne da bi korisniku pružile mogućnost upravljanja funkcionalnostima pojedinog presjeka. Sloj je implementiran pomoću HTML-a i CSS-a.

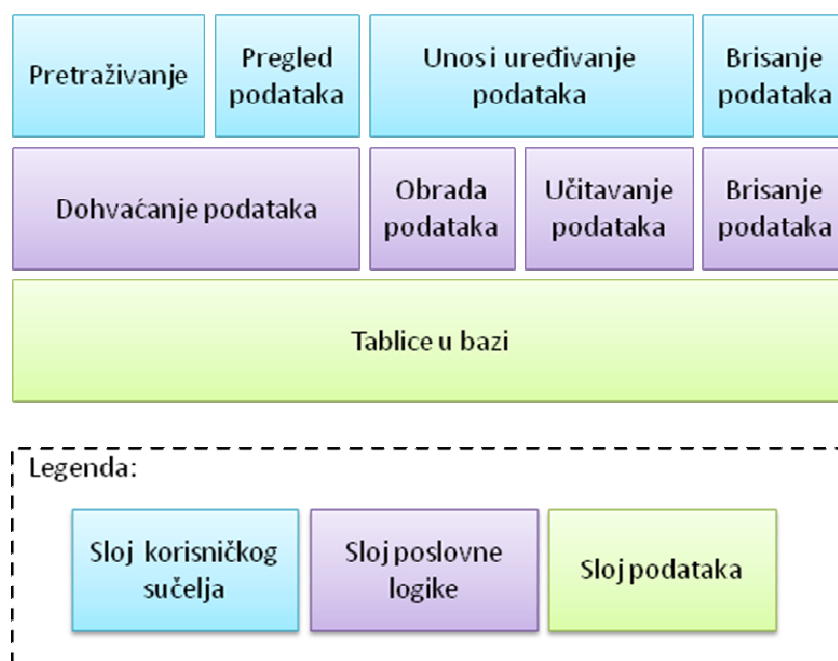
U modelu postoje tri razine korisnika:

- Običan korisnik – omogućeno jedino pretraživanje podataka
- Knjižničar – uz pretraživanje, omogućen unos novih podataka
- Administrator baze – uz pretraživanje i unos novih podataka, omogućeno je uređivanje i brisanje istih.

## 4.2. Presjek poslovne logike

Presjek poslovne logike je presjek koji sadrži većinu funkcionalnosti modela i također je presjek koji je u direktnoj interakciji s poslovnim korisnicima. Pod pojmom poslovni korisnik se misli na ljudsku osobu čiji je cilj, ovisno o razini funkcionalnosti koje su joj dodijeljene, dohvaćanje, učitavanje i brisanje podataka iz baze.

Struktura presjeka poslovne logike prikazana je na slici 12.

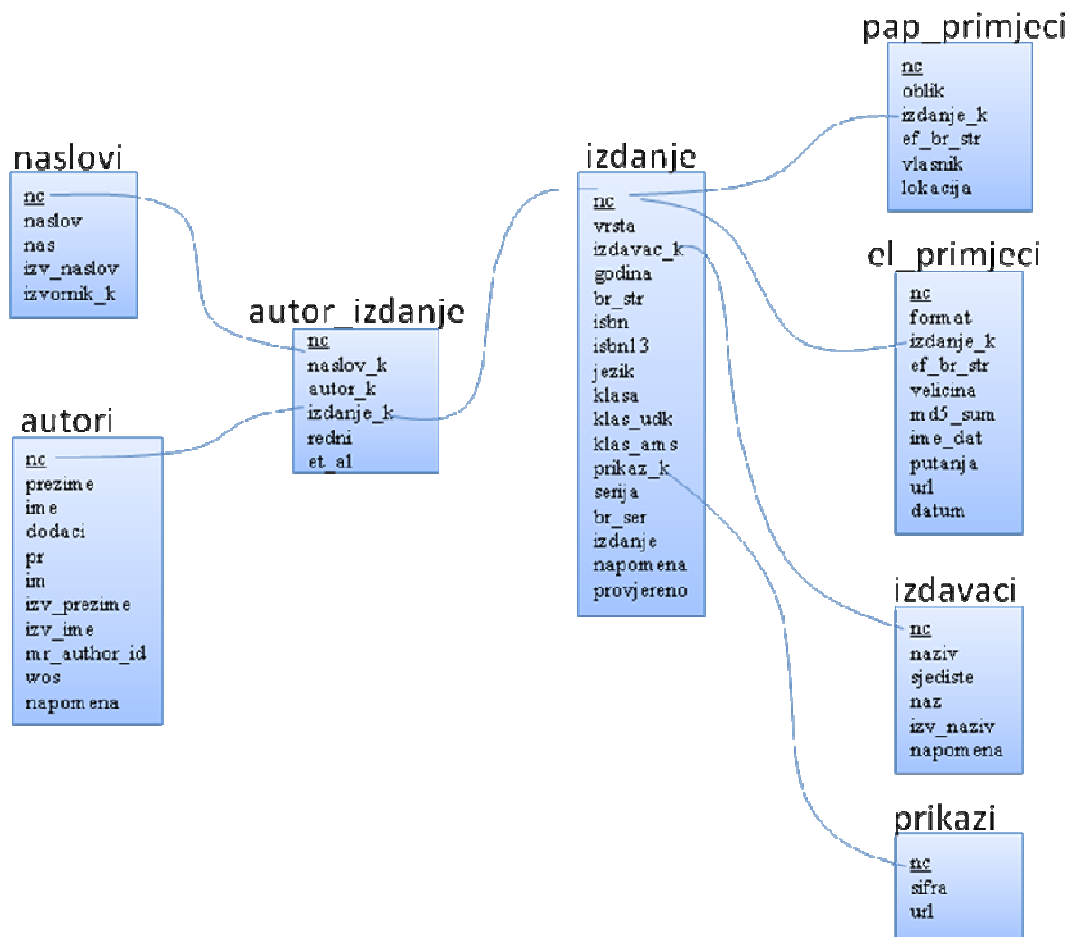


Slika 12. Struktura presjeka poslovne logike

Slojevi su ugrubo objašnjeni u prethodnom poglavlju i u ovom presjeku slijede istu logiku. S druge strane, slojevi su specifični za svaki presjek pa detaljan opis svakog od slojeva ovog presjeka slijedi u tekstu ispod.

### 4.2.1. Sloj podataka

Kao što je gore navedeno ovaj sloj sačinjavaju tablice koje sadrže podatke. Tablice i povezanost među njima prikazane su na slici 13, dok je struktura pojedine tablice opisana ispod slike.



Slika 13. Tablice u bazi „knjiznica.db“

#### Tablica „naslovi“

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
naslov	string	Naslov knjige [utf8]
nas	string	[ascii, auto generiran iz naslov]
izv_naslov	string	Naslov na izvornom pismu [utf8]
izvornik_k	string	Naslov na izvornom jeziku (samo za prijevode)

**Tablica „autori“**

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
prezime	string	Prezime autora [utf8]
ime	string	Ime autora [utf8]
dodaci	string	Dodaci imenu (poput jr., pl., III, ...)
pr	string	[ascii, auto generiran iz prezime]
im	string	[ascii, auto generiran iz ime]
izv_prezime	string	Izvorno prezime [utf8]
izv_ime	string	Izvorno ime [utf8]
mr_author_id	string	Jedinstveni identitet matematičkih autora u bazi podataka Mathematical Reviews®
wos	string	WOS klasifikacija
napomena	string	Napomena

**Tablica „izdanje“**

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
vrsta	string	Autorski, zbornik, ...
izdavac_k	int	Veza na tablicu „izdavaci“
Godina	int	Godina izdanja
br_str	string	Broj stranica [npr. xix+231]
isbn	string	ISBN klasifikacija [0..9,X]
isbn13	string	ISBN13 klasifikacija [auto generiran iz isbn]
jezik	string	[zarezima odijeljen niz ISO oznaka]
klasa	string	Gruba klasifikacija
klas_udk	string	UDK klasifikacija [odijeljeno s „:“]
klas_ams	string	AMS klasifikacija [odijeljeno s „:“]
prikaz_k	int	Veza na tablicu „prikazi“
serija	string	Ime serije (poput Lecture notes in

		mathematics)
br_ser	string	Redni broj u seriji
izdanje	string	Broj izdanja
napomena	string	Napomena
provjereno	string	Stupanj ispravnosti unosa (uređuje samo administrator)

**Tablica „autor\_izdanje“**

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
naslov_k	int	Veza na tablicu „naslovi“
autor_k	int	Veza na tablicu „autori“
izdanje_k	int	Veza na tablicu „izdanje“
redni	int	Redni broj autora
et_al	int	Et al

**Tablica „izdavaci“**

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
naziv	string	Naziv izdavača [utf8]
sjediste	string	Sjedište izdavača [utf8]
naz	string	[asci, auto generiran iz naziv]
izv_naziv	string	Naziv na izvornom pismu
napomena	string	Napomena

**Tablica „pap\_primjerci“**

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
oblik	string	{K,k,F,f}
izdanje_k	int	Veza na tablicu „izdanje“

ef_br_str	int	Efektivni broj stranica
vlasnik	string	Ime osobe koja posjeduje knjigu
lokacija	string	Opis mjesta na kojem se nalazi (ormar, ...)

**Tablica „el\_primjerci“**

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
format	string	.djvu, .pdf
izdanje_k	int	Veza na tablicu „izdanje“
ef_br_str	int	Efektivni broj stranica
velicina	int	Veličina [u kb]
md5_sum	string	32-znamenkasti HEX [auto generiran]
ime_dat	string	Ime datoteke
putanja	string	Putanja
url	string	URL
datum	string	Datum

**Tablica „prikazi“**

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
sifra	string	Šifra review-a
url	string	Lokacija review-a

#### 4.2.2. Sloj poslovne logike

Sloj poslovne logike sadrži većinu funkcionalnosti modela, implementiran je pomoću procedura i funkcija u PHP-u i SQLite-u i omogućuje unos podataka u bazu, te dohvaćanje i brisanje istih. U ovom podpoglavlju opisana je samo procedura koja se odvija u pozadini. U slijedećem podpoglavlju prikazano je kako to vidi korisnik na ekranu.

## 1. Pretraživanje podataka

Bazu je moguće pretraživati prema:

- naslovu knjige,
- autoru (imenu i/ili prezimenu autora ili MR Author ID),
- ISBN, ISBN13, AMS ili UDK klasifikaciji.

Baza se pretražuje pomoću funkcije `search($selection, $what)` napisane u PHP-u

`$selection` – tekstualna varijabla koja se odnosi na korisnikovu selekciju u za pretraživanje baze (naslov, autor, klasifikacija). Potrebna je radi određivanja uvjeta za pretraživanje podataka

`$what` – tekstualna varijabla koja se odnosi na niz znakova koji se treba naći u bazi.

Na primjer, ako se traži autor Dobriša Cesarić, tada je varijabla `$selection = „autor“`, a varijabla `$what = "Dobriša Cesarić"`.

Dohvaćanje traženih podataka u bazi izvršava se pomoću slijedećeg SQLite upita:

```
$sql = "Select naslovi.nc, naslov, izvornik_k, ime, prezime, et_al, izdanje, vrsta,
jezik, godina, serija, br_ser, isbn, isbn13, klas_ams, klas_udk, naziv, sjediste
from (((naslovi inner join autor_izdanje on naslovi.nc = autor_izdanje.naslov_k)
inner join autori on autori.nc = autor_izdanje.autor_k)
inner join izdanje on izdanje.nc = autor_izdanje.izdanje_k)
inner join izdavaci on izdavaci.nc = izdanje.izdavac_k ";
```

Crvenom bojom napisan je SQLite upit te spremljen u PHP varijablu `$sql`.

Ovaj upit nije cjelovit, jer sadrži samo selekciju podataka koji su potrebni da bi se ispisali za pretraživanje i tablice koje sadrže potrebne podatke. Ne sadrži uvjet za pretraživanje podataka. Radi preglednosti koda, uvjet za pretraživanje se kasnije „lijepi“ na gore navedeni upit te se upit nakon toga izvršava.

Tekstualna varijabla `$condition` koja sadrži uvjet pretraživanja lijepi se na gore navedeni upit pomoću PHP operatora „.“.

```
$sql .= $condition;
```

Varijabla `$condition` generira se pomoću ugrađene switch-case funkcije u PHP-u.

```
switch ($selection){
    case "naslov";
        $condition = "where ((upper(naslov) like upper('%".$what."%')) and
        (provjereno < 10)) order by naslovi, redni;";
    break;
    case "autor";
```



```

$autor = explode(',', $what);
array_walk($autor, 'trim_value');

if(isset($autor[1]))
    $tmp_condition = "where ((upper(prezime) like
        upper('%".$autor[0]."%')) and (upper(ime) like
        upper('%".$autor[1]."%'))) order by naslovi, redni";
else $tmp_condition = "where ((upper(prezime) like
        upper('%".$autor[0]."%')) or (upper(ime) like
        upper('%".$autor[0]."%')) or mr_author_id like
        ('%".$autor[0]."%')) order by naslovi, redni";

$tmp_sql = "select naslovi.nc from ((naslovi inner join autor_izdanje
on naslovi.nc = autor_izdanje.naslov_k) inner join autori on autori.nc
= autor_izdanje.autor_k) ".$tmp_condition ;

$condition = "where ((naslovi.nc in (" . $tmp_sql. ")) and (provjereno <
10));";
unset($autor);
break;
case "klasifikacija";
$condition = "where (((upper(isbn) like upper('%".$what."%')) or
(upper(isbn13) like upper('%".$what."%')) or (upper(klas_udk) like
upper('%".$what."%')) or (upper(ams) like upper('%".$what."%'))) and
(provjereno < 10)) order by naslovi, redni";
break;
}

```

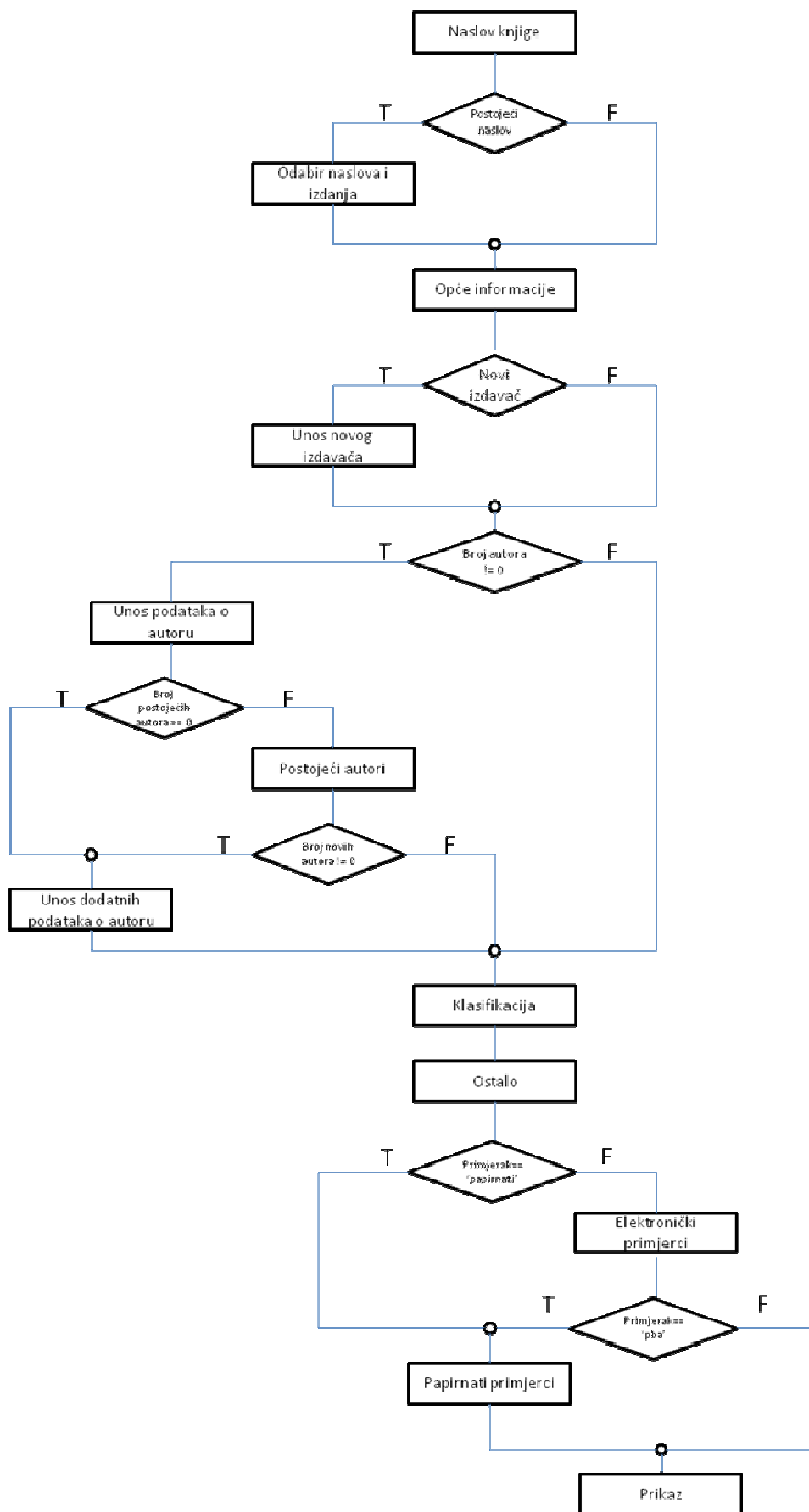
## 2. Unos podataka

U ovom modelu moguć je unos sljedećih podataka:

- novi naslov,
- novi autor,
- novi izdavač.

Unos podataka zamišljen je u obliku „čarobnjaka“ (eng. wizard) koji vodi korisnika od jedne do druge forme za unos podataka.

Procedure za unos novog autora i novog izdavača su relativno jednostavne, jer se sastoje od samo jedne forme (detaljnije u poglavlju 4.2.3), dok se procedura za unos novog naslova sastoji od više formi i uvjeta. Iz tog razloga slijedi opis samo procedure unosa naslova. Pseudo kod procedure za unos naslova vidljiv je na slici 14 na sljedećoj stranici, a opis slijedi u tekstu iza slike.



Slika 14. Pseudo kod procedure za unos naslova

Prvi korak je unos naslova knjige. Nakon toga slijedi provjera postoji li već u bazi uneseni naslov i ako postoji ispisuje sve pojave unesenog naslova skupa sa izdanjima koja za taj naslov postoje. Korisnik može odabrati unos podataka o novom izdanju knjige s određenim naslovom ili odabrati unos nove knjige.

Zadim slijedi unos općih informacija o naslovu (datoteka UnosKnjigeOpcenito.php):

- Izdanje (postavljena vrijednost je '1')
- Izdavač (padajući izbornik, moguće je izabrati izdavača koji već postoji u bazi ili odabrati unos novog.)
- Broj autora (ukoliko to polje ostane prazno ili je unesena vrijednost '0', tada se autor pohranjuje kao Anonimus)
- Izvorni naslov – na izvornom pismu, npr. ćirilica (ovo i iduće polje se pojavljuju samo ako se radi o potpuno novoj knjizi)
- Izvorni naslov – na izvornom jeziku, npr. ruski

Ako korisnik u izborniku „Izdavač“ izabere vrijednost „--Novi izdavač--“, slijedeći korak je unos podataka o novom izdavaču. Ako korisnik izabere jednog od postojećih izdavača u bazi, korak za unos podataka o izdavaču se preskače i provjerava se polje „Broj autora“. Ako je u to polje unesena neka vrijednost različita od nula otvara se forma za unos imenâ i prezimenâ autorâ knjige. Ta forma je automatski generirana, tj. polja „Ime autora“ i „Prezime autora“ generiraju se onoliko puta kolika je postavljena vrijednost polja „Broj autora“. Slijedi primjer koda.

Na početku datoteke UnosKnjigeIzdavaci.php stoji uvjet:

```
if($_SESSION['izdavac'] != 'novi') header( "Location: UnosKnjigeAutori.php" );
```

Što znači ako je odabran neki izdavač koji postoji u bazi, prebacuje na stranicu UnosKnjigeAutori.php, u suprotnom ostaje na stranici UnosKnjigeIzdavaci.php.

Isto tako, na početku datoteke UnosKnjigeAutori.php stoji:

```
if($_SESSION['brojAutora'] == 0) header( "Location: UnosKnjigeKlasifikacija.php" );
```

Dakle, ako je vrijednost polja „Broj autora“ postavljena na 0, autor se pohranjuje kao Anonimus i prelazi se na stranicu UnosKnjigeKlasifikacija.php

Ako je vrijednost polja „Broj autora“ veća od nula, generira se forma za unos imenâ i prezimenâ autorâ knjige, odnosno generira se  $n$  polja „Ime autora“ i „Prezime autora“, gdje je  $n$  vrijednost polja „Broj autora“

```

$counter = 1;
$n = $_SESSION['brojAutora'];
while ( $counter <= $n) {
    echo "<tr>";
    echo "<td \"/>";
    echo "<td id = \"left\"><label for=\"imeAutor\" . $counter.\">Ime autora:</label></td>";
    echo "<td id = \"right\"> <input name=\"imeAutor\" . $counter.\"\"
        type=\"text\" maxlength=\"100\" class=\"text\" \" ";

        Ispis('imeAutor' . $counter);

        echo " required/></td>";
    echo "</tr>";
    echo "<tr>";
    echo "<td id = \"left\"><label for=\"prezimeAutor\" . $counter.\">Prezime autora:</label></td>";
    echo "<td id = \"right\"> <input name=\"prezimeAutor\" . $counter.\"\"
        type=\"text\" maxlength=\"100\" class=\"text\" \" ";

        Ispis('prezimeAutor' . $counter);

        echo "/></td>";
    echo "</tr>";
    $counter++;
}

```

\$counter – numerička varijabla, brojač

\$n – vrijednost polja „Broj autora“

Pomoću while petlje, dok je  $\$counter \leq \$n$  generira se tablica koja sadrži formu za unos imenâ i prezimenâ autora. Na kraju while petlje, varijabla \$counter se povećava za 1, kad vrijednost varijable \$counter prijeđe vrijednost varijable \$n, izlazi iz petlje.

Nakon unosa imenâ i prezimenâ autorâ, provjerava se postoje li već ti autori u bazi i ponudi se korisniku da odabere ako se radi o autoru iz baze ili unos novog autora s tim imenom i prezimenom. Slijedi dio koda uz objašnjenje.

```

for ($i = 1; $i <= $n; $i++){
    $novi[$i] = $i; // sprema redne brojeve autora
    $ime = "imeAutor" . $i;
    $prezime = "prezimeAutor" . $i;
    $sql = "select * from autori where (upper(ime) = upper('$baza->escapeString($_SESSION[$ime]). ''')
        and upper(prezime) = upper('$baza->escapeString($_SESSION[$prezime]). '''))";
    //pronadi u bazi autore sa unesenim imenom i prezimenom i spremi ih u varijablu result
    $result = $baza->query($sql);

    unset($ponavljanje);
    $ponavljanje = array(); //ako u bazi već postoje više autora s istim imenom i prezimenom, ali različitim ostalim podacima
    while($res = $result->fetchArray(SQLITE3_ASSOC)){
        unset($row);
        $row = array();
        if(!isset($res['nc'])) continue;
        $row['nc'] = $res['nc'];
        $row['prezime'] = $res['prezime'];
        $row['ime'] = $res['ime'];
        $row['dodaci'] = $res['dodaci'];
        $row['izv_prezime'] = $res['izv_prezime'];
        $row['izv_ime'] = $res['izv_ime'];
        $row['mr_author_id'] = $res['mr_author_id'];
        $row['wos'] = $res['wos'];
        $row['napomena'] = $res['napomena'];

        array_push($ponavljanje, $row); //ubaci u polje ponavljanje polje row
        unset($novi[$i]); // iz polja novi makni redni broj i, tako da na kraju ostanu spremljeni samo redni brojevi novih autora
    }
    if(count($ponavljanje) != 0){
        $postojeci[$i] = $ponavljanje;
        // polje koje sadrži polje ponavljanje na mjestu i koje je redni broj autora
        array_push($_SESSION['postojeciAutori'], $i); // spremi ga u session polje za kasnije
    }
}
$_SESSION['noviAutori'] = $novi;
if (count($_SESSION['postojeciAutori']) == 0) header( "Location: UnosKnjigeNoviAutori.php" );
//ako niti jedan od unesenih autora ne postoji u bazi, idi na unos podataka o novim autorima

```

\$novi – polje koje na početku pohranjuje redne brojeve svih autora. Ako neki autor postoji u bazi, njegov redni broj se kasnije briše.

\$sql – tekstualna varijabla u koju se pohranjuje SQLite upit za dohvaćanje svih autora iz baze s imenom i prezimenom autora s rednim brojem \$i

\$row – pomoćno polje u koje se spremaju informacije iz baze o postojećem autoru.

\$ponavljanje – polje u koje se spremaju informacije iz baze o postojećem autoru. S obzirom da u bazi može postojati više autora s istim imenom i prezimenom, ovo je dvodimenzionalno polje

\$postojeci – polje koje sadrži sve informacije o svim autorima koje je korisnik unio, a postoje u bazi. Na mjestu \$i sadrži podatke iz baze o svim autorima s istim imenom i prezimenom autora unesenim pod rednim brojem \$i

Radi lakšeg shvaćanja koda slijedi objašnjenje koda pomoću slika i opisa. Razmotrimo slijedeći primjer - pretpostavimo da smo postavili da postoje tri autora, dakle \$n = 3. Pretpostavimo da se jedan zove „Novi Autor“, drugi „Postojeći Autor“, a treći „Još jedan Autor“. Dakle \$imeAutor1 = 'Novi', \$prezimeAutor1 = 'Autor', \$imeAutor2 = 'Postojeći', \$prezimeAutor2 = 'Autor', te \$imeAutor3 = 'Još jedan', \$prezimeAutor2 = 'Autor' Pretpostavimo da prvi autor (tj. „Novi Autor“) ne postoji u bazi, da drugi („Postojeći Autor“) i treći („Još jedan Autor“) postoje u bazi, te neka postoje dva zapisa o drugom autoru („Postojeći Autor“):

Prvi zapis		Drugi zapis	
nc	1	nc	2
prezime	Autor	prezime	Autor
ime	Postojeći	ime	Postojeći
dodaci	I.	dodaci	II.
pr		pr	
im		im	
izv_prezime		izv_prezime	
izv_ime		izv_ime	
mr_author_id	123	mr_author_id	12345
wos		wos	
napomena	Prvi zapis za Postojeći Autor	napomena	Drugi zapis za Postojeći Autor

Znači, u for petlji \$i ide do 3.

\$i = 1

\$novi[1] = 1

\$novi 

1
---

1

S obzirom da podaci o „Novi Autor“ ne postoje u bazi, while petlja se preskače, polje \$ponavljanje ostaje prazno, preskače se i if uvjet i ide se na drugi korak

\$i = 2 \$novi[2] = 2

\$novi 

1	2
---	---

1                      2

Autor „Postojeći Autor“ postoji u bazi, while petlja se izvršava:

```
$row['nc'] = 1, $row['prezime'] = 'Autor', $row['ime'] =
'Postojeći', $row['dodaci'] = 'I.', $row['izv_prezime'] =
'', $row['izv_ime'] = '', $row['mr_author_id'] = '123',
$row['wos'] = '', $row['napomena'] = 'Prvi zapis za
Postojeći Autor'
```

\$row	1	'Autor'	'Postojeći'	'I.'	'123'	Prvi zapis za Postojeći Autor
	nc	prezime	ime	dodaci	mr_author_id	napomena

\*Radi jednostavnijeg prikaza, izostavljena su prazna polja.

```
array_push($ponavljanje, $row)
```

\$ponavljanje:

1	1	'Autor'	'Postojeći'	'I.'	'123'	Prvi zapis za Postojeći Autor
	nc	prezime	ime	dodaci	mr_author_id	napomena

```
unset($novi[2])
```

\$novi	1	<del>2</del>
	1	2

S obzirom da postoje 2 zapisa u bazi o autoru „Postojeći Autor“, while petlja će se odvrtni još jedan put, spremi u polje \$row drugi zapis o autoru, te polje \$row spremi u polje \$ponavljanje. Sada polje \$ponavljanje izgleda ovako:

\$ponavljanje:

1	1	'Autor'	'Postojeći'	'I.'	'123'	Prvi zapis za Postojeći Autor
	nc	prezime	ime	dodaci	mr_author_id	napomena
2	2	'Autor'	'Postojeći'	'II.'	'12345'	Drugi zapis za Postojeći Autor
	nc	prezime	ime	dodaci	mr_author_id	napomena

\$ponavljanje ima 2 zapisa, dakle uvjet `if(count($ponavljanje) != 0)` prolazi.

```
$postojeci[2] = $ponavljanje;
```

Špostojeci:

1							
2	1	1	'Autor'	'Postojeći'	'I.'	'123'	Prvi zapis za Postojeći Autor
		nc	prezime	ime	dodaci	mr_author_id	napomena
2	2	2	'Autor'	'Postojeći'	'II.'	'12345'	Drugi zapis za Postojeći Autor
		nc	prezime	ime	dodaci	mr_author_id	napomena

Ši = 3 Analogno kao u drugom koraku, samo što polje Šponavljanje ima jedan zapis Šponavljanje:

1	3	'Autor'	'Još jedan'			Zapis za Još jedan Autor
	nc	prezime	ime	dodaci	mr_author_id	napomena

Na kraju for petlje imamo:

Šnovi

1
---

1

Špostojeci:

1							
2	1	1	'Autor'	'Postojeći'	'I.'	'123'	Prvi zapis za Postojeći Autor
		nc	prezime	ime	dodaci	mr_author_id	napomena
2	2	2	'Autor'	'Postojeći'	'II.'	'12345'	Drugi zapis za Postojeći Autor
		nc	prezime	ime	dodaci	mr_author_id	napomena
3	1	3	'Autor'	'Još jedan'			Zapis za Još jedan Autor
		nc	prezime	ime	dodaci	mr_author_id	napomena



Ta dva polja pohranjuju se u session polja, tako da se mogu koristiti i kasnije.

Ako je polje `$postojeci` neprazno, korisniku se nudi da odabere o kojem se autoru iz baze radi ili da odabere unos novog autora s istim imenom. Ako je korisnik odabrao unos novog autora ili ako je unio ime i prezime autora koji ne postoji u bazi, ili ako je polje `$postojeci` prazno, slijedeći korak je `UnosKnjigeNoviAutori.php` na kojoj se unose podaci o novom autoru. U suprotnom, ako ne postoje novi autori, tj. ako je korisnik odabrao samo one autore koji postoje u bazi, taj se korak preskače i ide se na stranicu `UnosKnjigeKlasifikacija.php` na kojoj se unose podaci o klasifikaciji knjige:

```
if (count($_SESSION['noviAutori']) == 0) header( "Location: UnosKnjigeKlasifikacija.php" );
```

Poslije klasifikacije slijedi unos ostalih podataka o knjizi - broj stranica, jezik, vrsta primjerka itd.

Polje „Vrsta primjerka“ napravljeno je kao polje za odabir gdje korisnik može izabrati jednu od tri mogućnosti:

- Elektronički primjerak
- Papirnati primjerak
- Oba primjerka

Ovisno o tome koji primjerak je odabran, otvara se forma za unos informacija o primjerku. Ako je označen odabir „Oba primjerka“, najprije se otvara forma za unos informacija o elektroničkom primjerku, zatim o papirnatom.

Zadnji korak je prikaz svih unesenih informacija kako bi korisnik mogao pregledati unesene podatke, te ih eventualno ispraviti ili kao takve unjeti u bazu.

### 3. Pregled podataka

Moguće je odabrati jedan od tri pregleda:

- pregled izdanja,
- pregled autora (svi autori ili autori koji nisu vezani za izdanje)
- pregled izdavača (svi izdavači ili izdavači koji nisu vezani za izdanje)
- pregled članova (neriješeni zahtjevi, knjižničari i administratori)

Na stranicama koje omogućuju pregled podataka iz baze moguće je odabrati brisanje ili uređivanje podataka.

Primjer SQLite upita za odabir autora koji nisu vezani za izdanje:

```
mysql = "SELECT autori.nc as prim, prezime, ime, dodaci, pr, im, izv_prezime, izv_ime, mr_author_id, wos, napomena
FROM autori left join autor_izdanje on autori.nc = autor_izdanje.autor_k
where autori.nc not in(select autor_k from autor_izdanje) and autori.nc <> '0'";
```

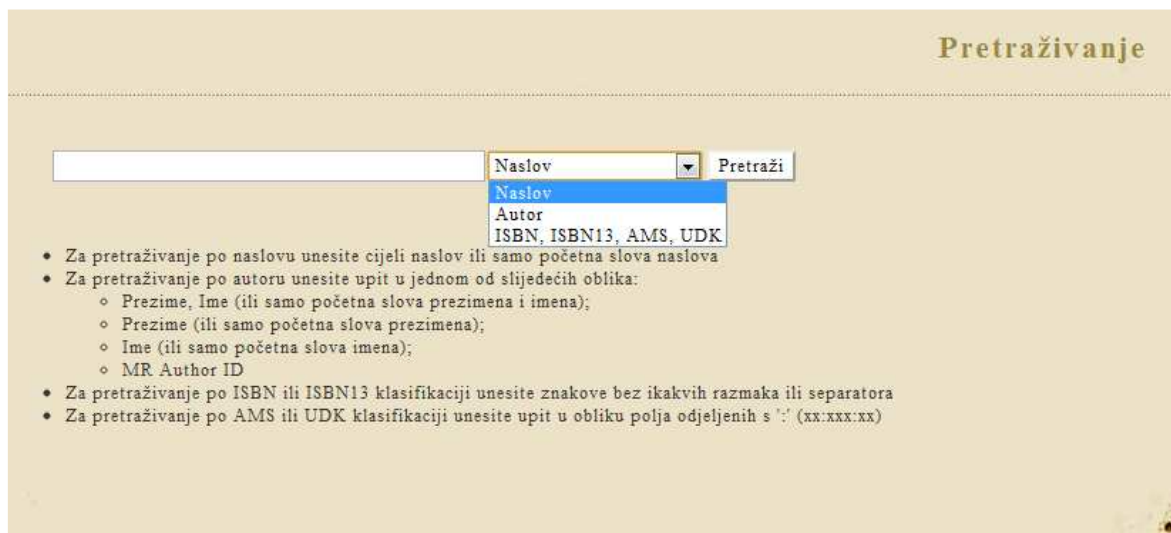
Relacije „autori“ i „autor\_izdanje“ povezane su na temelju atributa autori.nc (primarni ključ u relaciji „autori“) i autor\_izdanje.autor\_k (veza na autor u tablici „autor\_izdanje“). Operacija left join između relacija uključuje sve zapise iz relacije „autori“ čak i ako oni ne postoje u tablici „autor\_izdanje“. Zato na kraju stoji uvjet where autori.nc not in (select autor\_k from autor\_izdanje), kako bi se uzeli u obzir samo oni autori koji postoje u relaciji „autori“ a ne postoje u relaciji „autor\_izdanje“. Drugi dio uvjeta (autori.nc <> '0') služi da bi se izbjegao autor „Anonimus“, jer je njegov primarni ključ 0.

### 4.2.3. Sloj korisničkog sučelja

Ovo poglavlje po strukturi prati prethodno i prikazuje ono što korisnik vidi prilikom otvaranja stranica, dok je u prethodnom poglavlju opisana logika u pozadini cijele procedure.

Ovaj sloj implementiran je pomoću HTML-a i CSS-a.

#### 1. Pretraživanje podataka



Slika 15. Pretraživanje baze

Kao što je već gore spomenuto, bazu je moguće pretraživati prema:

- naslovu knjige,
- autoru (imenu i/ili prezimenu autora ili MR Author ID),
- ISBN, ISBN13, AMS ili UDK klasifikaciji.

## 2. Unos podataka

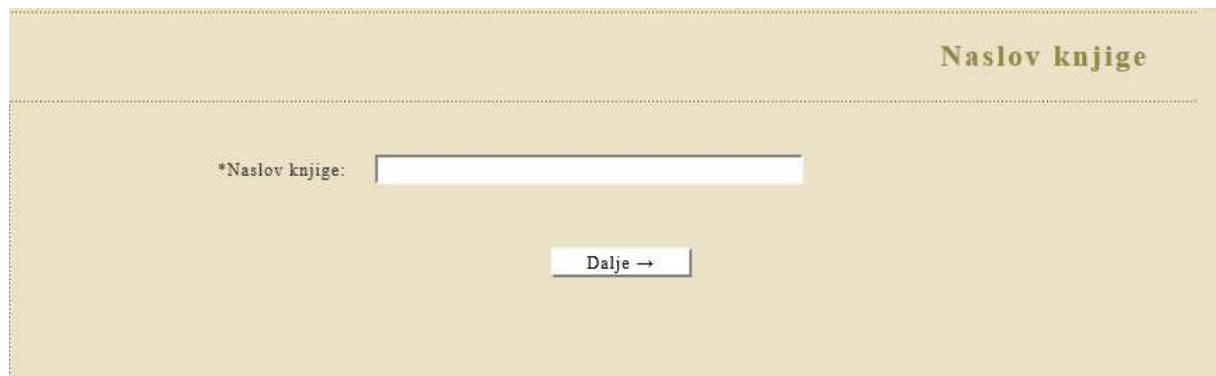
Moguć je unos sljedećih podataka:

- novi naslov,
- novi autor,
- novi izdavač i
- novi član/izmjena korisničkih podataka

### Unos knjige

Za skicu ove procedure pogledati poglavlje 4.2.2, sliku 14

- i) Prvi korak je unos naslova knjige. Ukoliko naslov već postoji u bazi, sljedeći korak je odabir knjige s tim naslovom kako bi se mogli unijeti podaci o novom izdanju te knjige, ili odabir nove knjige. Ako naslov ne postoji u bazi, taj se korak preskače i ide se na sljedeći.



Naslov knjige

\*Naslov knjige:

Dalje →

Slika 16. Forma za unos naslova knjige

ii) Odabir knjige

Naslov knjige

Slijedeća izdanja s naslovom knjiga su pronađena u bazi:

Izvorni naslov - izvorni jezik:  
 Izvorni naslov - izvorno pismo: The Book

**Knjiga / imel prezime1**

Izdavač: Zagreb: Školska knjiga, 2011  
Izdanje:  
Vrsta: izvješće  
Serija: 2, 2  
Jezik: engleski

Klasifikacija:

- ISBN: 81-7525-766-0
- ISBN13:
- AMS:
- UDK:

**Knjiga / ime2 prezime2**

Izdavač: središte: izdavač, 1986  
Izdanje: second edition  
Vrsta: journal  
Jezik: engleski

Klasifikacija:

- ISBN: 81-7525-766-0
- ISBN13:
- AMS:
- UDK:

Nova knjiga

Dalje →

Slika 17. Forma odabir knjige

iii) Zatim slijedi unos općih podataka

- izdanje (postavljena vrijednost je '1')
- izdavač (padajući izbornik, moguće je izabrati već unesene izdavače ili odabrati unos novog.)

U slučaju odabira unosa novog izdavača slijedeći korak je unos podataka o izdavaču, u suprotnom se taj korak preskače.

- Broj autora (postavljena vrijednost je '0')

Ukoliko to polje ostane prazno ili je unesena vrijednost '0', tada se autor pohranjuje kao Anonimus.

- Izvorni naslov – na izvornom pismu, npr. ćirilica (ovo i iduće polje se pojavljuje samo ako se radi o novoj knjizi, ili naslovu koji ne postoji u bazi)
- Izvorni naslov – na izvornom jeziku, npr. ruski

Opće informacije

Izdanje:   
(Ako je polje prazno, izdanje će biti spremljeno kao prvo)

Izdavač:

Broj autora:  (Ako je označen broj 0 ili je polje prazno, autor će biti spremljen kao Anonimus)

Izvorni naslov:  
(Izvorno pismo)

Izvorni naslov:  
(Na izvornom jeziku)

Dalje →

Slika 18. Forma za unos općih informacija

Ako je u ovom koraku u polju „Izdavač“ označeno „--Novi izdavač--“, sljedeći korak je unos podataka o novom izdavaču. U suprotnom, ako je izabran jedan od postojećih izdavača, taj se korak preskače i ide se na sljedeći korak.

iv) Unos podataka o izdavaču:

- Naziv izdavača (obavezno polje)
- Sjedište izdavača (obavezno polje)
- Izvorni naziv (na izvornom jeziku)
- Napomena

Podaci o izdavaču

Naziv:

Sjedište:

Izvorni naziv:

Napomena:

Dalje →

Slika 19. Forma za unos podataka o izdavaču

- v) Ukoliko je u prvom koraku polje „Broj autora“ ostavljeno prazno, ili je unesena vrijednost '0', autor se sprema kao „Anonimus“ i ovaj korak se preskače. U suprotnom

se otvara forma za unos podataka o autorima. Polja se automatski generiraju ovisno o broju autora koje je korisnik unio. Na slici 20. je prikazana forma za dva autora.

U ovu formu potrebno je unjeti ime i prezime pojedinog autora (obavezna polja), te u slučaju da nisu svi autori navedeni (et al) označiti okvir za izbor „Et al“.

The screenshot shows a form titled 'Autori' with a light beige background. On the left, there are two vertical labels '1|' and '2|' corresponding to two rows of input fields. Each row has a label 'Ime autora:' followed by a text input field, and 'Prezime autora:' followed by another text input field. Below these fields is a checkbox labeled 'Et al:'. At the bottom center, there is a button labeled 'Dalje ->'.

Slika 20. Forma za unos podataka o autoru

Ukoliko jedan od autora već postoji u bazi, otvara se forma za izbor autora iz baze. Također, korisnik može izabrati i unos novog autora. S druge strane, ako niti jedan od autora ne postoji u bazi, taj se korak preskače i otvara se forma za unos podataka o novim autorima. U slučaju da neki autori postoje, a neki ne postoje u bazi ide se na oba koraka. Za potrebe ovog primjera unesen je jedan postojeći i jedan novi autor.

vi) U ovom koraku otvara se forma za izbor postojećeg autora iz baze. Slika 21.

The screenshot shows a form titled 'Autori' with a light beige background. Below the title, there is a text instruction: 'Slijedeći autori su pronađeni u bazi. Označite ako se radi o tom autoru ili odaberite unos novih podataka.' Below this is a table with a header row and three data rows. The header row is labeled 'postojeći autor' in red. The columns are 'Izvorno ime', 'Izvorno prezime', 'MR Author ID', 'WOS', and 'Napomena'. The first data row has radio button 'a', 'a', 'b', '123', '123'. The second data row has radio button 'b', 'b', 'c', '1234', '1234'. The third data row has radio button 'Novi autor'. At the bottom center, there is a button labeled 'Dalje ->'.

postojeći autor					
	Izvorno ime	Izvorno prezime	MR Author ID	WOS	Napomena
<input checked="" type="radio"/>	a	b	123	123	
<input type="radio"/>	b	c	1234	1234	
<input type="radio"/>	Novi autor				

Slika 21. Forma za izbor postojećeg autora

Za potrebe ovog primjera, u bazi postoje dva autora s imenom i prezimenom „postojeći autor“, ali imaju drugačije ostale podatke. Korisnik može izabrati ili jednog

od ponuđenih autora ili izabrati unos novog autora s tim imenom i prezimenom. U ovom primjeru izabran je prvi ponuđeni autor.

**vii)** Slijedi forma za unos podataka o autorima koji ne postoje u bazi.

Od korisnika se traži da unese slijedeće podatke:

- Ime i prezime (postavljena vrijednost je ona već prije unesena)
- Dodaci (npr. jr., pl., III., ...)
- Izvorno ime i prezime
- WOS
- MR Author ID
- Napomena

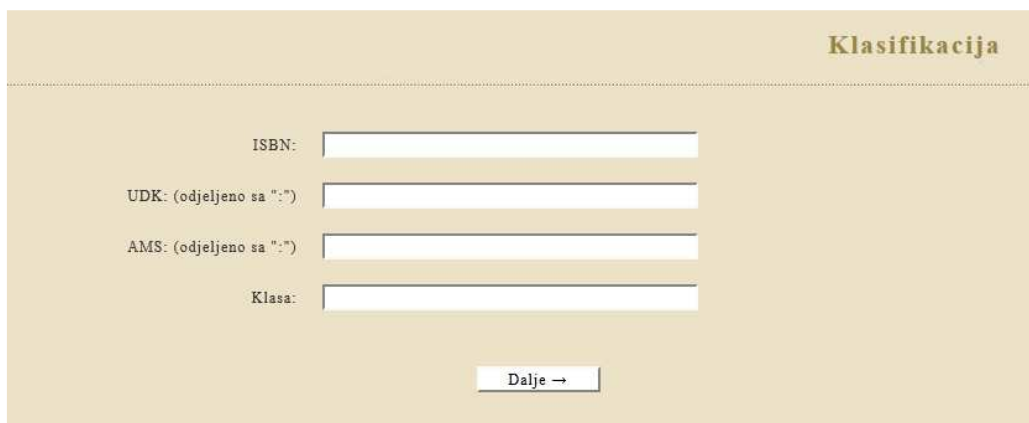


The image shows a web form titled 'Avtori' (Authors) with a sub-section 'Podaci o autoru 2' (Data about author 2). The form contains several input fields: 'Ime autora:' with the value 'novi', 'Prezime autora:' with the value 'autor', 'Dodaci: (poput jr., pl., ...)' which is empty, 'Izvorno ime:' which is empty, 'Izvorno prezime:' which is empty, 'WOS:' which is empty, 'MR Author ID:' which is empty, and a larger text area for 'Napomena:' (Note) which is also empty. At the bottom of the form is a button labeled 'Dalje →' (Next).

Slika 22. Forma za unos podataka o novim autorima

**viii)** Slijedeći korak je unos podataka o klasifikaciji knjige

- ISBN
- UDK
- AMS
- Klasa



Klasifikacija

ISBN:

UDK: (odjeljeno sa ":")

AMS: (odjeljeno sa ":")

Klasa:

Dalje →

Slika 23. Forma za unos podataka o klasifikaciji knjige

**ix)** Slijedi unos ostalih informacija:

- Broj stranica
- Jezik
- Vrsta
- Godina izdanja
- Prikaz
- Serija
- Redni broj u seriji
- Vrsta primjerka (papirnati, elektronički ili oba)
- Opis / napomena



Ostale informacije

Broj stranica:

Jezik:

Vrsta:

Godina izdanja:

Prikaz:

Seriya:

Redni broj u seriji:

Vrsta primjerka:  Elektronički primjerak  
 Papirnati primjerak  
 Oba primjerka

Opis/napomena:

Dalje →

Slika 24. Forma za unos ostalih informacija o knjizi

Ukoliko korisnik odabere „Elektronički primjerak“, otvara se forma za unos podataka o elektroničkom primjerku, ukoliko odabere „Papirnati primjerak“, otvara se forma za unos podataka o papirnatom primjerku, a ako odabere „Oba primjerka“, otvaraju se obje forme. U ovom primjeru označit ćemo odabir „Oba primjerka“.

- x) U formi za unos elektroničkog primjerka od korisnika se traži unos slijedećih podataka:
- Format (.djvu, .pdf, ...)
  - Efektivni broj stranica
  - Veličina datoteke (u kb)
  - Ime datoteke
  - Putanja
  - URL
  - Datum

Elektronički primjerak

Format (.djvu, .pdf, ...):

Efektivni broj stranica:

Veličina (u kb):

Ime datoteke:

Putanja:

URL:

Datum:

Dalje →

Slika 25. Forma za unos podataka o elektroničkom primjerku

xi) U formi za unos papirnatom primjerka od korisnika se traži unos sljedećih podataka:

- Oblik
- Efektivni broj stranica
- Vlasnik
- Lokacija

Papirnat primjerak

Oblik (K, k, F, f):

Efektivni broj stranica:

Vlasnik:

Lokacija (opis ormara, ...):

Dalje →

Slika 26. Forma za unos podataka o papirnatom primjerku

Nakon unosa svih podataka, korisnik može vidjeti sve unesene podatke, te potom odabrati treba li ih preurediti ili ih želi spremi u bazu (slika 27.)

Ukoliko odabere uređivanje podataka, cijeli gore navedeni postupak se ponavlja, samo što su vrijednosti već postavljene na one definirane od strane korisnika.

Pregled unesenih informacija

---

**Općenite informacije**

Naslov knjige:  
 Izdanje:  
 Broj autora: 2  
 Izvorni naslov:  
 (Izvorno pismo)  
 Izvorni naslov:  
 (Na izvornom jeziku)

---

**Izdavač**

Naziv: \*  
 Sjedište: \*  
 Izvorni naziv:  
 Napomena:

---

**Autori**

Autor 1: postojeći autor  
 Autor 2: novi autor  
 Et al.: Ne

---

**Klasifikacija**

ISBN:  
 UDK:  
 AMS:  
 Klasa:

---

**Ostale informacije**

Broj stranica:  
 Jezik:  
 Vrsta:  
 Godina izdanja:  
 Prikaz:  
 Scrija:  
 Redni broj u seriji:  
 Primjerak: oba  
 Napomena:

---

**Elektronički primjerak**

Format:  
 Efektivni broj stranica:  
 Veličina:  
 Ime datoteke:  
 Putanja:  
 URL:  
 Datum:

**Papirnat primjerak**

Oblik:  
 Efektivni broj stranica:  
 Vlasnik:  
 Lokacija:

Slika 27. Pregled unesenih informacija

### Unos novog autora

S obzirom da ne postoji puno podataka o novom autoru, za unos podataka postoji samo jedna forma (nije napravljeno u obliku čarobnjaka).

Od korisnika se traži unos slijedećih podataka

- Ime autora (obavezno polje)
- Prezime autora (obavezno polje)
- Dodaci
- Izvorno ime
- Izvorno prezime
- WOS
- MR Author ID
- Napomena



The image shows a web form titled "Podaci o autoru" (Author Data) on a light beige background. The form contains several input fields for author information:

- Ime autora: [text input field]
- Prezime autora: [text input field]
- Dodaci: (poput jr., pl., ...) [text input field]
- Izvorno ime: [text input field]
- Izvorno prezime: [text input field]
- WOS: [text input field]
- MR Author ID: [text input field]
- Napomena: [large text area]

At the bottom center of the form is a button labeled "Unesi u bazu" (Enter into database).

Slika 28. Forma za unos podataka o autoru

### Unos novog izdavača

Unos podataka o novom izdavaču također nije zamišljen u obliku čarobnjaka, nego postoji samo jedna forma za unos sljedećih podataka o izdavaču:

- Naziv izdavača (obavezno polje)
- Sjedište (obavezno polje)
- Izvorni naziv
- Napomena



The image shows a web form titled "Podaci o izdavaču" (Publisher Data). The form is set against a light beige background. It contains four input fields: "Naziv:" (text), "Sjedište:" (text), "Izvorni naziv:" (text), and "Napomena:" (text area). A "Unesi u bazu" (Save to database) button is located at the bottom center of the form.

Slika 29. Forma za unos podataka o izdavaču

### Unos novog člana/izmjena korisničkih podataka

Pri unosu novih članova, tj. slanja zahtjeva za registraciju novih članova potrebno je unijeti sljedeće podatke:

- Korisničko ime (najmanje 6 znakova, bez razmaka – obavezno polje)
- Lozinka (najmanje 6 znakova, bez razmaka – obavezno polje)
- Ponovni unos lozinke (obavezno polje)
- E-mail (obavezno polje)
- Ime
- Prezime

Forma za registraciju korisnika prikazana je na slici 30., dok je forma za izmjenu korisničkih podataka prikazana je na slici ispod

**Registracija korisnika**

\*Korisničko ime:  
Najmanje 6 znakova, bez razmaka

\*Lozinka:  
Najmanje 6 znakova, bez razmaka

\*Ponovite lozinku:

\*E-mail:

Ime:

Prezime:

OK

Slika 30. Forma za zahtjev za registraciju korisnika

**Uređivanje korisničkih podataka**

\*Stara lozinka: ●●●●●●

\*Nova lozinka:  
Najmanje 6 znakova, bez razmaka

\*Ponovite novu lozinku:

\*E-mail: a.b@a.com

Ime:

Prezime:

OK

Slika 31. Forma za uređivanje korisničkih podataka

### 3. Pregled podataka

Moguće je odabrati jedan od tri pregleda:

- pregled izdanja,
- pregled autora (svi autori ili autori koji nisu vezani za niti jedno izdanje)
- pregled izdavača (svi izdavači ili izdavači koji nisu vezani za niti jedno izdanje)
- pregled članova (neriješeni zahtjevi, knjižničari i administratori)

Korisničko sučelje za pregled podataka nije grafički uređeno, radi veće iskoristivosti prostora za same podatke, umjesto za dizajn.

#### Pregled izdanja

Kao što je vidljivo na slici 32., odabirom na pregled izdanja prikazuju se podaci o svim izdanjima unesenim u bazu. Na istoj slici dolje također su vidljivi gumbi „Uredi“ i „Obriši“ koji omogućuju administratoru uređivanje postojećih podataka ili brisanja istih.

Klikom na gumb „Uredi“ otvara se forma za uređivanje podataka, ovisno o tome na kojem mjestu je korisnik kliknuo gumb.

Klikom na gumb „Obriši“ brišu se svi podaci o izdanju, ali ne i podaci o autoru i izdavaču.

< Prev | 1 | 2 | Next >

### Kako napraviti digitalnu knjižnicu

Naslov	ASCII naslov	Izvorni naslov: (Izvorno pismo)	Izvorni naslov: (Na izvornom jeziku)
Kako napraviti digitalnu knjižnicu		How to build a digital library	How to build a digital library

[Uredi](#)

**Izdavač**

USA, Morgan Kaufmann Publishers  
 ASCII naziv:  
 Izvorni naziv: Morgan Kaufmann Publishers  
 Napomena:

[Uredi](#)

**Autori**

1  
 Ime i prezime: Ian H.Witten  
 ASCII ime i prezime:  
 Izvorno ime i prezime: Ian H.Witten  
 MR Author ID:  
 WOS:  
 Napomena: Neznam ništa o njemu

2  
 Ime i prezime: David Bainbridge  
 ASCII ime i prezime:  
 Izvorno ime i prezime: David Bainbridge  
 MR Author ID:  
 WOS:  
 Napomena:

3  
 Ime i prezime: David M.Nichols  
 ASCII ime i prezime:  
 Izvorno ime i prezime: David M.Nichols  
 MR Author ID:  
 WOS:  
 Napomena:

[Uredi](#)

**Izdanje**

Izdanje: 2  
 Godina izdanja: 2003  
 Serija:  
 Redni broj u seriji:  
 Vrsta:  
 Broj stranica: 629  
 Jezik: engleski  
 ISBN: 0-12-374857-7  
 ISBN13:  
 UDK:  
 AMS:  
 Klasa:  
 Napomena:

[Uredi](#)

**Primjerak**

Elektronički primjerak

Format:  
 Efektivni broj stranica:  
 Veličina (kb):  
 MD 5: [Uredi](#)  
 Ime datoteke:  
 Putanja:  
 URL:  
 Datum:

[Dodaj papirni primjerak](#)  
[Dodaj papirni i ukloni elektronički primjerak](#)

PROVJERENO:

[Spremi](#) [Obriši](#)

Slika 32. Pregled izdanja



## Pregled autora

Moguće je izabrati pregled svih autora i pregled onih autora kojima nije dodjeljen niti jedan naslov.

Na slici 33. prikazan je pregled svih autora, dok je na slici 34. prikazan pregled autora koji nemaju dodjeljeno izdanje.

< Prevl 1 | 2 | 3 | Next >

**Pregled autora**

Ime i prezime	Dodaci	ASCII ime i prezime	Izvorno ime i prezime	MR Author ID	WOS	Napomena	
1 2	4		6		3q		Uredi
q 2							Uredi
das asd			fa fas				Uredi
postojeći autor			a b	123	123		Uredi

Slika 33. Pregled svih autora

< Prevl 1 | Next >

**Sljedeći autori nisu vezani za niti jedno izdanje**

Obriši sve

Ime i prezime	Dodaci	ASCII ime i prezime	Izvorno ime i prezime	MR Author ID	WOS	Napomena	
1 2	4		6		3q		Obriši
q 2							Obriši
postojeći autor			b c	1234	1234		Obriši
bla bla			a d				Obriši

Slika 34. Pregled autora koji nemaju izdanje

Kao što je vidljivo iz prethodnih slika, kod pregleda svih autora, moguće je samo urediti autore, da bi se izbjegla situacija brisanja autora koji je vezan za određeno izdanje. Kod pregleda autora bez izdanja moguće obrisati autore jednog po jednog ili sve odjednom.

### Pregled izdavača

Kao i u prethodnom slučaju, moguće je izabrati pregled svih izdavača i pregled onih kojima nije dodjeljeno niti jedano izdanje.

Na slici 35., odnosno 36., prikazani su redom: pregled svih izdavača i pregled izdavača koji nemaju dodjeljeno niti jedno izdanje.

< Prev | 1 | Next >

**Pregled izdavača**

Naziv	Sjedište	ASCII naziv	Izvorni naziv	Napomena	
Cambridge University Press	Cambridge				Uredi

Naziv	Sjedište	ASCII naziv	Izvorni naziv	Napomena	
Cambridge University Press	Cambridge, UK				Uredi

Naziv	Sjedište	ASCII naziv	Izvorni naziv	Napomena	
izdavač	sjedište		sa		Uredi

Naziv	Sjedište	ASCII naziv	Izvorni naziv	Napomena	
Morgan Kaufmann Publishers	USA		Morgan Kaufmann Publishers		Uredi

Slika 35. Pregled svih izdavača

< Prev | 1 | Next >

**Slijedeći izdavači nisu vezani za niti jedno izdanje**

Naziv	Sjedište	ASCII naziv	Izvorni naziv	Napomena	
Novi izdavač	da				Obriši

Slika 36. Pregled izdavača bez izdanja

Također, iz istog razloga kao i kod autora, kod pregleda svih izdavača, moguće je samo uređivanje izdavača, dok je kod onih koji nemaju izdanje moguće brisanje jednog po jednog ili svih izdavača.

## Pregled članova

U ovom slučaju moguće je odabrati pregled neriješenih zahtjeva, pregled knjižničara ili administratora.

Neriješeni zahtjevi su tek pristigli korisnički zahtjevi za registraciju i polje „Odobren“ ima vrijednost „NE“.

< Prevl 1 | Next >

**Pregled neriješenih zahtjeva**

Datum zahtjeva	Korisničko ime	Ime	Prezime	E-mail	Status	Odobren (DA/NE)		
2012-01-19	123456789	1		tamara.bucic@gmail.com	knjiznicar	NE	Spremi	Obrisi

Datum zahtjeva	Korisničko ime	Ime	Prezime	E-mail	Status	Odobren (DA/NE)		
2012-01-19	tamara			tamara.bucic@gmail.com	knjiznicar	NE	Spremi	Obrisi

Slika 37. Pregled neriješenih zahtjeva

Administrator može korisniku promijeniti status („knjiznicar“ / „admin“) ili odobren („NE“ / „DA“)

U pregledu knjižničara i administratora nalaze se članovi koji imaju status „knjiznicar“, odnosno „admin“ (slika 38. i 39.)

< Prevl 1 | Next >

**Pregled knjižničara**

Ako želite obrisati nekog člana, najprije u polju "Odobren" postavite vrijednost "NE", te ga zatim obrišite na stranici s neriješenim zahtjevima.

Korisničko ime	Ime	Prezime	E-mail	Status	Odobren (DA/NE)	
knjižničar				knjiznicar	DA	Spremi

Korisničko ime	Ime	Prezime	E-mail	Status	Odobren (DA/NE)	
123456	t	b	tamara.bucic@gmail.com	knjiznicar	DA	Spremi

Slika 38. Pregled knjižničara

< Prevl 1 | Next >

**Pregled administratora**

Ako želite obrisati nekog člana, najprije u polju "Odobren" postavite vrijednost "NE", te ga zatim obrišite na stranici s neriješenim zahtjevima.

Korisničko ime	Ime	Prezime	E-mail	Status	Odobren (DA/NE)	
admin			a.b@a.com	admin	DA	Spremi

Slika 39. Pregled administratora

### 4.3. Presjek sigurnosne logike

Presjek sigurnosne logike značajno je manje kompleksan nego presjek poslovne logike. Ovaj presjek ima dvije primarne funkcionalnosti:

1. Provjera autentičnosti
2. Autorizacija

Naravno, i ovaj sloj podijeljen je u tri sloja:

- Sloj podataka
- Sloj poslovne logike
- Sloj korisničkog sučelja

Struktura presjeka sigurnosne logike prikazana je na slici 40.



Slika 40. Struktura presjeka sigurnosne logike

Moduli u presjeku sigurnosti su slijedeći:

- Akreditacija – skup tablica koje pohranjuju korisničke akreditacije i nivoe autorizacije potrebne za određene funkcionalnosti (u presjeku poslovne logike).
- Provjera autentičnosti – sadrži procedure potrebne za provjeru autentičnosti korisnika nad tablicama za akreditaciju.
- Autorizacija – sadrži procedure koje pružaju mjerodavan odgovor može li korisnik pristupiti određenoj funkcionalnosti ili ne.
- Sigurnosno korisničko sučelje – To uključuje formu za prijavu koja je u interakciji sa korisnikom u svrhu sigurnosti.

### 4.3.1. Sloj podataka

Ovaj sloj sastoji se od baze „clanovi.sqlite“ koja sadrži samo jednu tablicu pod nazivom „clanovi“ sa sljedećom strukturom:

Ime stupca	Tip podataka	Opis
nc	int	Primarni ključ
usn	string	Korisničko ime
pwd	string	Sigurnosna lozinka
status	string	Nivo autorizacije dodjeljen korisniku („admin“ ili „knjiznicar“)
email	string	e-mail adresa korisnika
ime	string	Ime korisnika
prezime	string	Prezime korisnika
odobren	string	Polje koje poprima vrijednost „DA“ ili „NE“ ovisno o tome je li administrator odobrio korisnički račun
code	string	Jedinstveni kod koji se dodjeljuje novom korisniku i šalje u linku za potvrdu zahtjeva za registraciju na e-mail korisnika. Nakon što korisnik potvrdi korisnički zahtjev, polje se postavlja na NULL
date_request	date	Datum slanja zahtjeva za registraciju
date_confirm	date	Datum potvrde korisničkog zahtjeva

### 4.3.2. Sloj poslovne logike

Sadrži module za provjeru autentičnosti i autorizaciju. I u ovom presjeku sloj poslovne logike implementiran je pomoću PHP-a i SQLite-a.

#### 1. Provjera autentičnosti

Provjera autentičnosti odvija se na stranici „Login.php“ koja na početku sadrži sljedeći kod:

```

if(isset($_POST["usn"]) && isset($_POST["pwd"])){
    $baza = new SQLite3("clanovi.sqlite");
    $sql = "select * from clanovi where
            (usn = '". $baza->escapeString(trim($_POST["usn"]))."'
            and pwd = '". $baza->escapeString(trim($_POST["pwd"]))."'";

    $result = $baza->query($sql) or die ('error');
    while($res = $result->fetchArray(SQLITE3_ASSOC)){
        if(!isset($res['nc'])) continue;
        $status = $res['status'];
        $odobren = $res['odobren'];
        $nc = $res['nc'];
    }

    if(!isset($status) || ($odobren == 'NE')) $msg = "Neispravno korisni&#269;ko ime ili lozinka";
    else{
        $_SESSION['status'] = $status;
        $_SESSION['statusNC'] = $nc;
        header('Location: index.php' );
    }
}

require("main.php");

```

Odnosno, ako je uneseno korisničko ime i sigurnosna lozinka, pronađi podatke u bazi o korisniku s tim korisničkim imenom i lozinkom. Potom spremi informaciju o nivou autorizacije korisnika u tekstualnu varijablu `$status`, informaciju da li je korisnički račun odobren u varijablu `$odobren`, te informaciju o primarnom ključu („nc“) u varijablu `$nc`. Ako je varijabla `$status` prazna, tj. ako ne postoji korisnik s unesenim korisničkim imenom i lozinkom, i ako je vrijednost varijable `$odobren` = „NE“, tj. ako korisnički račun još nije odobren, spremi u varijablu `$msg` poruku „Neispravno korisničko ime ili lozinka“. Ta poruka se ispiše iznad forme za logiranje. U suprotnom spremi varijablu `$status` u sesijsku varijablu `$_SESSION['status']`, varijablu `$nc` u sesijsku varijablu `$_SESSION['statusNC']`, i idi na stranicu „index.php“

## 2. Autorizacija

Kao što je gore već spomenuto, ovaj modul sadrži procedure koje pružaju mjerodavan odgovor može li korisnik pristupiti određenoj funkcionalnosti ili ne. Odnosno procedure se razlikuju u ovisnosti o tome koja je funkcionalnost namjenjena za kojeg korisnika. Npr. običan korisnik ne može pristupiti unosu i pregledu podataka, knjižničar ne može pristupiti pregledu podataka, dok administrator baze može pristupiti svemu (više o razinama korisnika i dodjeljenim funkcionalnostima u poglavlju 4.4).

Tako na početku stranicâ koje omogućuju unos podataka u bazu (odnosno kojima mogu pristupiti administrator baze i knjižničar) stoji slijedeći dio koda:

```

if(!isset($_SESSION['status']))
    header('Location: Login.php');

```

Odnosno, ako ne postoji sesijska varijabla `$_SESSION['status']` vrati na stranicu „Login.php“. Provjera postojanja varijable `$_SESSION['status']` vrši se iz razloga jer ona može poprimiti samo vrijednosti 'admin' ili 'knjiznicar'.

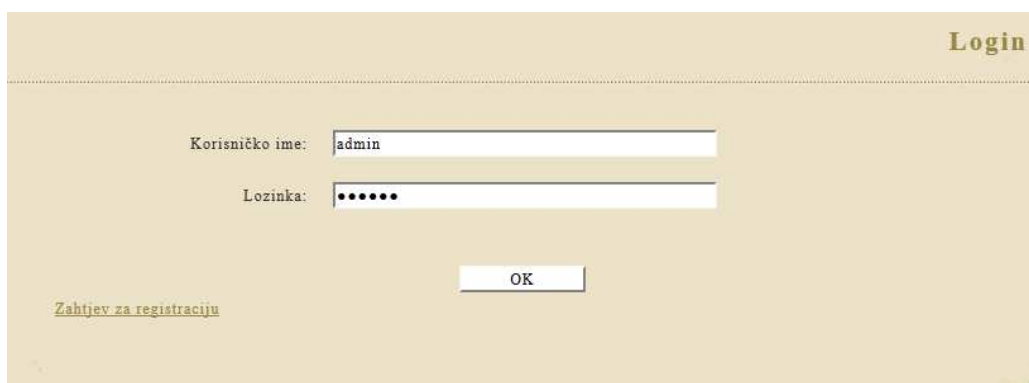
Na početku stranicâ koje omogućuju pregled podataka u bazi (odnosno kojima može pristupiti samo administrator baze) stoji slijedeći dio koda:

```
if(!isset($_SESSION['status']) || ($_SESSION['status'] != "admin"))  
    header('Location: Login.php');
```

Odnosno, ako ne postoji sesijska varijabla `$_SESSION['status']` ili postoji, a vrijednost joj je različita od „admin“, vrati na stranicu „Login.php“.

### 4.3.3. Sloj korisničkog sučelja

Prikaz korisničkog sučelja za prijavu korisnika prikazan je na slici 41.



Slika 41. Forma za prijavu korisnika

Ukoliko je uneseno ispravno korisničko ime i lozinka, prebacuje se na stranicu „index.php“ za pretraživanje baze.

Ukoliko korisničko ime i/ili lozinka nisu ispravne, ili korisnički račun nije odobren, pojavljuje se poruka (slika 42.)



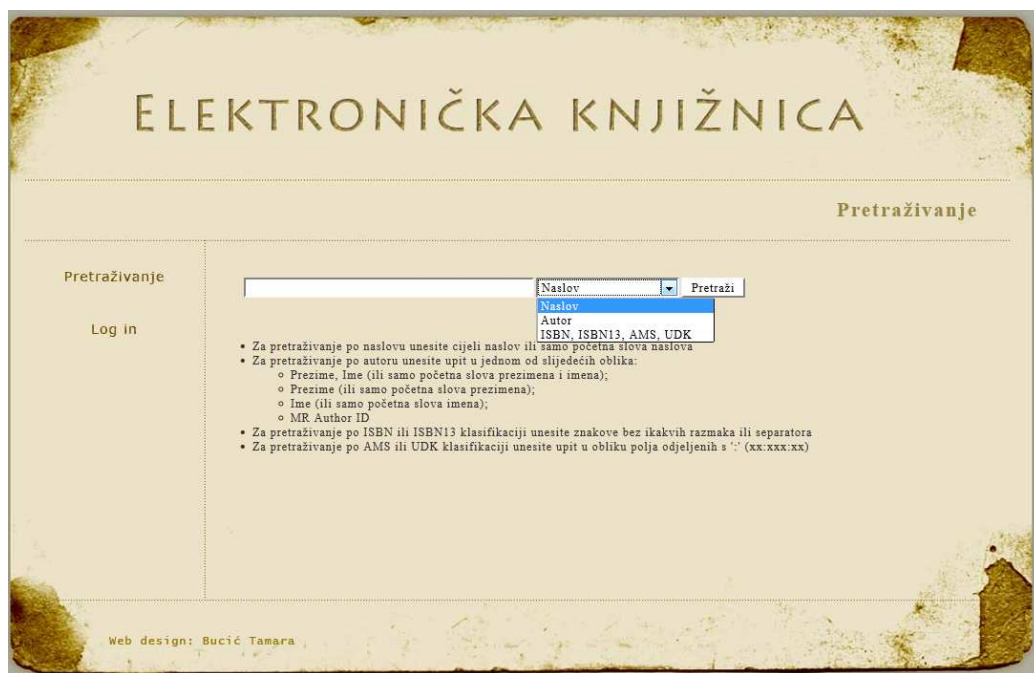
Slika 42. Poruka u slučaju neispravnosti korisničkog imena ili sigurnosne lozinke

## 4.4. Razine korisnika

Kako je spomenuto, postoje tri razine korisnika: običan korisnik, knjižničar i administrator baze. Svakom od njih dodjeljene su različite razine funkcionalnosti.

### 1. Običan korisnik

Običnom korisniku omogućeno je jedino pretraživanje podataka. Prikaz koji on vidi prikazan je na slici 43.



Slika 43. Prikaz za običnog korisnika

### 2. knjižničar

Uz funkcionalnosti običnog korisnika, ovom korisniku je omogućen unos podataka u bazu.

Na slici 44. prikazan je prikaz korisničkog sučelja za knjižničara. Na izborniku s lijeve strane vide se dodatne funkcionalnosti za knjižničara.



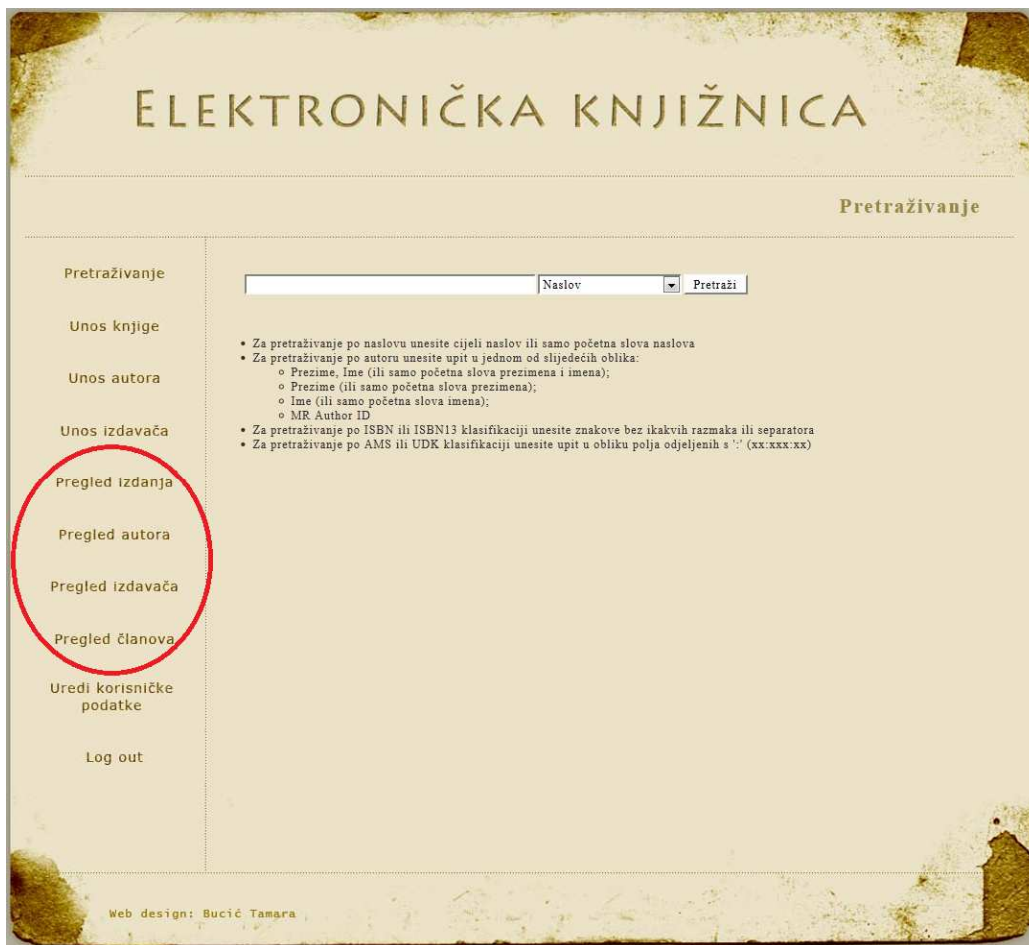


Slika 44. Prikaz za knjižničara

### 3. Administrator baze

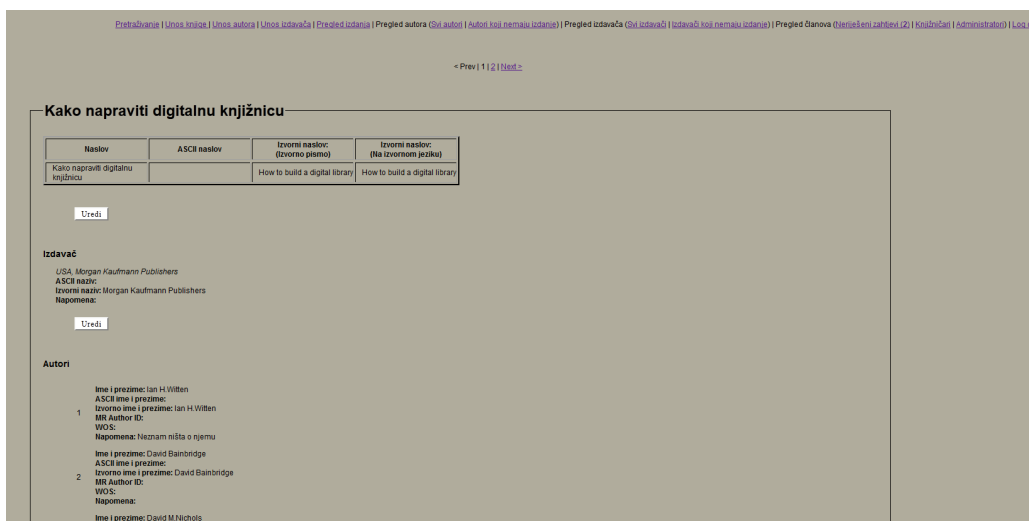
Osim funkcionalnosti omogućenih knjižničaru i običnom korisniku, administrator može pregledati sve podatke unesene u bazu, može ih preurediti i obrisati. Na slici 45. na izborniku s lijeve strane su vidljive dodatne mogućnosti dodijeljene administratoru baze, dakle:

- pregled izdanja,
- pregled autora,
- pregled izdavača i
- pregled članova



Slika 45. Prikaz za administratora

Korisničko sučelje za mogućnosti dodjeljene administratoru baze nije grafički uređeno, radi veće iskoristivosti prostora za same podatke, umjesto za dizajn – slika 46.



Slika 46. Pregled izdanja

## 4.5. Preuvjeti za instalaciju

Za instalaciju ove aplikacije, potrebna je slijedeća arhitektura:

- Apache2 http server verzije 2.2.0+
- Php5 procesor verzije 5.2.0+
- sqlite3 baza podataka verzije 3.5.1+

Datoteke koje služe za bazu podataka (tj. knjiznica.db i clanovi.sqlite) moraju imati rw (read i write) postavke za korisničko ime pod kojim je pokrenut web server, odnosno:

```
chown www-data.www-data clanovi.sqlite
```

```
chown www-data.www-data knjiznica.db
```

- umjesto `www-data` staviti korisničko ime i Linux grupu pod kojom je pokrenut web server na Linux-u (npr. `apache2`, `no-data`)

Potrebno je postaviti i read-write prava za korisničko ime i grupu pod kojom je pokrenut web server na Linux-u, odnosno:

```
chmod 664 clanovi.sqlite
```

```
chmod 664 knjiznica.db
```

- umjesto 664 može ići i 660

## Zaključak

Elektroničke knjižnice možda neće zamijeniti postojanje knjiga i fizičkog dokumenta, ali bez sumnje da bi se zadovoljila sadašnja potražnja, te da bi se prilagodilo modernom i ubrzanom načinu života, sve više dokumenata morat će se digitalizirati. Inicijalni trošak digitalizacije možda je visok, ali jednom kad se digitalizacija počne provoditi, trošak digitalizacije određene zbirke knjiga će biti manja nego trošak iste takve zbirke u tradicionalnoj knjižnici. Dan za danom trošak digitalizacije se smanjuje, internetsko izdavaštvo se povećava, kao i potrebe korisnika za bržom i lakšom dostupnošću knjiga.

Web aplikacija koja je nastala kao programski dio ovog diplomskog rada trenutno ima samo svrhu upravljanja podacima o korisnikovoj zbirci knjiga, bilo elektroničkih, bilo fizičkih. Današnja tehnologija i programski jezici spomenuti u radnji omogućili su realizaciju iste. Trenutno ne posjedujem dovoljno znanja, ljudi niti tehnologije za izradu prave elektroničke knjižnice. Međutim, ova aplikacija bi mogla jednog dana postati polazišna točka u daljnjem razvoju u projekta elektroničke knjižnice putem koje će korisnici moći pristupiti knjigama spremljenim u bazi, a ne samo podacima o toj knjizi, kako je trenutno napravljena.

# Literatura

- Ian H. Witten, David Bainbridge, David M. Nichols: How to build a digital library, Elsevier, 2010.
- Anne Peterson Bishop, Barbara P. Buitendijk and Nancy A. Van House: Digital Library Use: Social Practice in Design and Evaluation, MIT Press, 2003.
- Stipčević, A. Povijest knjige. 2. prošireno i dopunjeno izd. Zagreb : Matica hrvatska, 2006.
- M. Owens: The definitive guide to SQLite, Apress, 2006.
- Jay A. Kreibich: Using SQLite, O'Reilly Media, 2010.
- Skripta - R. Manger: „Baze podataka“ - drugo izdanje - veljača 2011
- <http://hr.wikipedia.org/>
- <http://www.udcc.org/>
- <http://www.ams.org/>
- [www.web-upotrebljivost.com](http://www.web-upotrebljivost.com), Dario Šuveljak, Zagreb
- <http://www.tizag.com/phpT/>

## Sažetak

Ovaj dio diplomskog rada sastoji se od četiri poglavlja koja bi se logički mogla podijeliti na dva dijela. Prvi dio (1., 2. i 3. poglavlje) opisuje pojmove koji su potrebni za razumijevanje drugog dijela (4. poglavlje) koji je zapravo tehnička dokumentacija web aplikacije.

Da bi se razumio programski dio diplomskog rada korisnik mora razumjeti samu ideju elektroničke knjižnice, razliku između elektroničke i tradicionalne knjižnice, kao i način rada i strukturu elektroničke knjižnice. Također, da bi se mogla koristiti web aplikacija koja je nastala kao programski dio diplomskog rada, potrebno je znati nešto o načinu klasifikacije knjiga i autora, te o postojećim formatima elektroničkih knjiga. Time se bavi prvo poglavlje – „Elektroničke knjižnice“.

Drugo poglavlje – „Baze podataka“ služi da bi olakšalo ideju o načinu spremanja podataka koje koristi web aplikacija. To poglavlje ukratko opisuje glavna područja vezana uz baze podataka - čemu služe baze podataka, što je relacijski model, što su entiteti i kakve veze između njih postoje, normalizacija baze, što su SQL i SQLite, te kakvi upiti postoje i čemu služe.

Za realizaciju ove web aplikacije, potreban pojam koji se zove „Web programiranje“ - upravo time se bavi treće poglavlje. U ovom poglavlju napisano je ukratko o WWW-u, HTML-u, CSS-u i PHP-u. Što je pojedini pojam, koja je struktura programskih jezika koji se koriste za web programiranje i primjeri koda.

Četvrto poglavlje – „Praktični dio diplomskog rada“ je zapravo tehnička dokumentacija, ali ujedno i uputstvo za upotrebu ove web aplikacije. Opisuje strukturu modela, odnosno logičku podjelu modela na presjeke i slojeve, te detaljno opisuje svaki od njih. Na kraju poglavlja napisana je arhitektura potrebna za instalaciju ove aplikacije na web server.

# Summary

This thesis consists of four chapters, which would logically be divided into two parts. The first part (Chapter 1, 2 and 3) describes the concepts needed to understand the second part (Chapter 4), which is actually the technical documentation of the web application.

In order to understand the programming part of graduate work the user must understand the very idea of electronic libraries, the difference between electronic and traditional libraries, as well as the operation and structure of the electronic library. Also, in order to use a web application that was created as part of the graduate program of work; you need to know something about the classification of books and authors, and the existing format of electronic books. The first chapter - "Electronic libraries" deals with those terms.

The second chapter - the "Databases" is used to explain how to store data in databases used by web application. This section briefly describes the main issues on the databases – what are they used for, what is the relational model, what are the entities, and what relationships between them there are, the normalization of database, what are SQL and SQLite, what queries are and what they do.

For the realization of this web application, a term named "Web programming" is needed – with this term precisely deals the third chapter. This chapter is written briefly on the World Wide Web, HTML, CSS and PHP. What a single term is, what does the structure of the programming languages used for web programming looks like and code examples.

Chapter Four - "The practical part of of graduate work" is actually the technical documentation, but also the instructions for use of this web application. This chapter describes the structure of the model, i.e. a logical segregation of the model cross sections and layers, and describes in detail each of them. At the end of the chapter there is an architecture required for the installation of this application on the web server.

# Životopis

Tamara Bucić rođena je 16.11.1986. u Slavonskom Brodu, gdje je pohađala Osnovnu i Srednju školu. Od malena je pokazivala interese za programiranje i tehniku, te u jesen 2001. upisuje Srednju tehničku školu u Slavonskom Brodu, usmjerenje *Tehničar za telekomunikacije*. Maturirala je u ljeto 2005. s projektom pod nazivom „Projekt za izazove 21. stoljeća – povezivanje PLC laboratorija u RH i EU“ s kojim sudjeluje na raznim izložbama, sajmovima inovacija i natjecanjima, te osvaja zlatno odličje na međunarodnom sajmu inovacija INOVA u Zagrebu i nagradu Matice hrvatske za najbolji maturalni rad u području automatizacije.

U jesen 2005. upisuje preddiplomski sveučilišni studij *Matematika*, PMF – Matematički odsjek u Zagrebu koji završava 2008. s akademskim nazivom Sveučilišna prvostupnica matematike. U jesen iste godine upisuje diplomski sveučilišni studij *Financijska i poslovna matematika*, PMF – Matematički odsjek, Zagreb. U drugom semestru prve godine diplomskog studija (zima, 2009.) prijavljuje se za ERASMUS stipendiju i odlazi studirati u Beč na jedan semestar, gdje akademske predmete sluša na engleskom jeziku i paralelno pohađa intenzivni tečaj njemačkog jezika. Nakon povratka iz Beča, ponovo upisuje prvu godinu diplomskog studija, te počinje raditi na automatizaciji financijskih modela na raznim projektima u poduzećima. Diplomski studij završava u zimu 2012. s akademskim nazivom Magistar poslovne i financijske matematike.

U slobodno vrijeme bavi se sportom (gimnastika, ples, aerobik, teretana i biciklizam) i glazbom (od 2010. redovita polaznica glazbene škole, usmjerenje solo pjevanje).