Dizajn i implemetacija elektronske bilblioteke

Bota, Mirjana

Master's thesis / Diplomski rad

2014

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:706617

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-07-16**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

ELEKTROTEHNIČKI FAKULTET

Sveučilišni studij

DIZAJN I IMPLEMENTACIJA ELEKTRONSKE BIBLIOTEKE

Diplomski rad

Mirjana Bota

Osijek, 2014. godina.

Sadržaj

1. UVOD 1

2. PREGLED UML-A 3

2.1. DIZAJN SUSTAVA I MODELIRANJE 3

2.2. DIJAGRAMI MODELA 5

2.3. ZAHTJEVI NA SUSTAV 9

3. PREGLED ELEKTRONSKE BIBLIOTEKE 10

4. MODELIRANJE ELEKTRONSKE BIBLIOTEKE 12

4.1. DIJAGRAM SLUČAJA KORIŠTENJA 12

4.2. DIJAGRAM AKTIVNOSTI 22

4.3. DIJAGRAM RASPOREĐIVANJA 23

4.4. KLASNI DIJAGRAM 24

4.5. SEKVENCIJALNI DIJAGRAM 27

5. IZRADA APLIKACIJE 34

5.1. PROGRAMSKI JEZICI ZA IZRADU INTERNET APLIKACIJE 34

5.2. PROGRAMSKI JEZIK ZA IZRADU I UPRAVLJANJE BAZOM PODATAKA 38

5.3. PROGRAMI ZA RAD S BAZOM PODATAKA I INTERNET PRISTUPOM 40

6. OPIS APLIKACIJE 43

6.1. BAZA PODATAKA 44

6.2. PRIJAVA KORISNIKA 45

6.3. PRETRAGA KNJIGE 48

6.4. KONTAKT FORMA 50

6.5. ADMINISTRACIJSKO SUČELJE 51

7. TESTIRANJE INTERNET APLIKACIJE 53

8. ZAKLJUČAK 57

1. UVOD

Internet je postao nezamjenjivo mjesto za edukaciju korisnika gdje je dostupan sve veći broj informacija. Brzi razvoj tehnologija i novo informacijsko doba promijenilo je današnje tradicionalne knjižnice koje su postale prevelike i loše organizirane te ograničene količinom medija i mogućnostima posudbe. Potreba za skladištenjem rastućeg broja medija i smanjenje vremena pretraživanja postaju ciljevi elektronske biblioteke koja nastaje digitalizacijom tradicionalne knjižnice.

Iako digitalna i tradicionalna knjižnica imaju istu svrhu: pružiti korisniku medij za čitanje, slušanje ili gledanje, razlikuju se u nabavi materijala što postaje prednost digitalne knjižnice. Svatko, tko ima Internet, može pristupiti sustavu elektronske biblioteke i preuzeti svoj medij. Jedan od većih problema digitalne biblioteke je brzo pretraživanje velike količine digitalnog medija te što točniji pronalazak rješenja koje je potrebno korisniku. Za rješavanje ovog problema koriste se dvije metode. Prva metoda reducira opseg pretrage pretraživanjem baze knjiga preko određenih kategorija. Druga metoda koristi meta podatke koje označavaju ključne riječi unutar određenog medija. Ta metoda još uvijek nije dovoljno pouzdana i točna pri pronalasku stoga se bolje ograničiti na sigurna rješenja.

Najznačajniji element o kojem treba voditi računa pri kreiranju, implementaciji i upravljanju digitalne biblioteke je njezino povezivanje s drugim knjižnicama u virtualnom prostoru. Iako knjižnice mogu otvoriti međusobni pristup datotekama, važno je dogovoriti ponude s vlasnicima digitalnih medija. Veliki problem digitalne knjižnice je sigurnost podataka te održavanje autorskih prava zbog neautoriziranog skidanja i umnožavanja datoteka koje su pod zaštitom. Stoga je drugi problem rješavanje sigurnosnog i validnog dijeljenja medija.

Zadatak ovog rada počinje obradom zahtjeva korisnika koji se služi elektronskom bibliotekom. Potom se izrađuje glavni cilj ovog rada, dizajn e-biblioteke u kojoj korisnik pretražuje i skida željeni materijal te implementacija sustava u obliku internet aplikacije prema prethodno izrađenom dizajnu. Internet aplikacija mora zadovoljiti potrebe korisnika i administratora elektronske knjižnice. Spomenuta implementacija mora pratiti prethodni dizajn sustava koji opisuje od čega su elementi sustava sastavljeni, kako su ti elementi međusobno povezani te kako sustav funkcionira.

Drugo poglavlje prikazuje UML dijagrame kao podlogu za rješavanje sustava elektronske biblioteke. Treće poglavlje opisuje elektronsku biblioteku te njezin princip rada.

Četvrto poglavlje prikazuje UML dijagrame elektronske biblioteke, a peto poglavlje prikazuje tehnologije korištene za rad aplikacije. Šesto poglavlje prikazuje realiziranu internet aplikaciju te mogućnosti korisnika i administratora u radu sa stranicom. Sedmo poglavlje prikazuje testiranje internet stranice bazirano na mogućim akcijama korisnika i načinu pregleda internet stranice. Osmo poglavlje opisuje zaključak ovog rada.

2. PREGLED UML-A

Svaki projekt, koji se radi prema zahtjevu klijenta, započinje obradom ciljeva koje sustav mora postići u radu. Razvojni tim započinje ispitivanjem tijeka procesa te razvija najvažnije slučajeve korištenja na kojima se bazira projekt. Kvalitetna bazna arhitektura za izgradnju dobre programske podrške postaje temelj kreiranja projekta unutar zadanog proračuna i vremena. U ovom radu korištene su grafičke tehnike uniformnog jezika modeliranja ili UML-a (engl. *Unified Modeling Language*) za kreiranje baznog vizualnog modela objektno-orijentiranog sustava. Objektno-orijentirani pristup je važan u razvoju programske podrške zbog raširene domene problema i rada s kompleksnim sustavima.

2.1. DIZAJN SUSTAVA I MODELIRANJE

Dizajn sustava na svakoj razini je vrlo težak i zahtjevan neovisno radi li se o jednostavnoj desktop aplikaciji ili višejezgrenom sustavu od više stotina softverskih i hardverskih komponenti. Modeliranje sustava pomaže boljoj vizualizaciji i kontroli arhitekture sustava te ujedno određuje strukturu ili ponašanje što su osnove za izgradnju sustava prema [2]. Za efektivno modeliranje sustava koristi se UML jezik koji pomaže obradi složenih sustava. Razlog zašto je UML postao standardnim jezikom modeliranja je programsko-jezična nezavisnost koja uz malo preinaka može popuniti bilo koji posao. UML se koristi za dizajniranje i komunikaciju softvera ili poslovnog procesa, dohvaćanje detalja za analizu sustava te dokumentiranje postojećeg sustava, procesa ili organizacije prema [3].

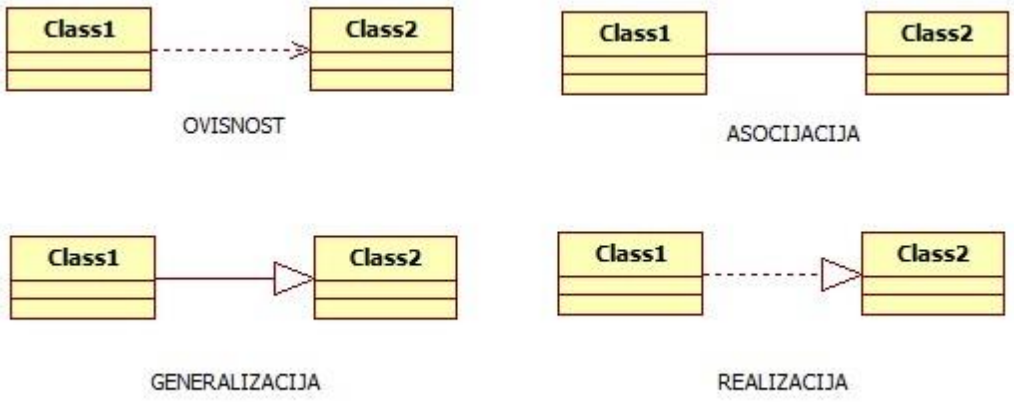
Jedna od pogreški u programiranju je kreiranje projekta bez plana ili modela. UML je zato dobar alat jer su grafički elementi jednostavni za razumijevanje uz tekst koji ih dodatno opisuje. To je omogućeno tzv. inženjerstvom unaprijed *(*engl. *forward engineering)*. Inženjerstvo unaprijed je tradicionalni postupak kod kojih se s viših nivoa apstrakcije prelazi na niže slojeve projektiranja i izvođenja. Tijekom izrade prati se i dokumentacija arhitekture sustava (projektni plan, testovi i sl.) što je važno za kontrolu, mjerenje i komunikaciju tijekom razvoja sustava i nakon njezina pokretanja. UML je namijenjen sustavima koji se oslanjaju na programsku podršku, a najčešće se koristi na području zdravstva, financija, zaštita i prodajnih sustava, no pojavljuje se i drugdje.

Konceptualan model UML jezika se sastoji od tri glavna elementa: osnovnih građevnih blokova, pravila njihova korištenja i osnovnih mehanizama primjene UML-a prema [2]. UML jezik stoga sadrži rječnik i pravila kombiniranja riječi. Pravila i rječnik govore kako kreirati i čitati napravljene modele, ali ne govore koje modele koristiti te kada ih kreirati.

To je uloga razvojnog procesa koji pomaže korisniku tijekom odlučivanja koje komponente i aktivnosti koristiti te kako ih kreirati. Rječnik UML-a sadrži tri vrste građevinskog bloka: stvari kao apstrakcije, relacije za povezivanje i dijagrame za grupiranje istih elemenata. Apstrakcije mogu biti statički dijelovi modela (klasa, sučelje, suradnja, slučaj korištenja ili objekt aktivne klase) odnosno konceptualni ili fizički elementi (komponenta ili čvor). Dinamički dijelovi modela predstavljaju ponašanje kroz prostor i vrijeme te se predstavljaju interakcijom ili automatom stanja. Dodatni oblici opisivanja apstrakcije mogu biti paketi koji povezuju elemente tijekom razvojnog procesa ili komentari (*engl. note)* koji detaljnije opisuju pripadnu apstrakciju.

Relacije se unutar UML modela mogu prikazati na četiri načina, a grafički se relacije označavaju isprekidanom ili punom linijom s mogućom oznakom smjera prema slici 2.1. Vrste relacija su:

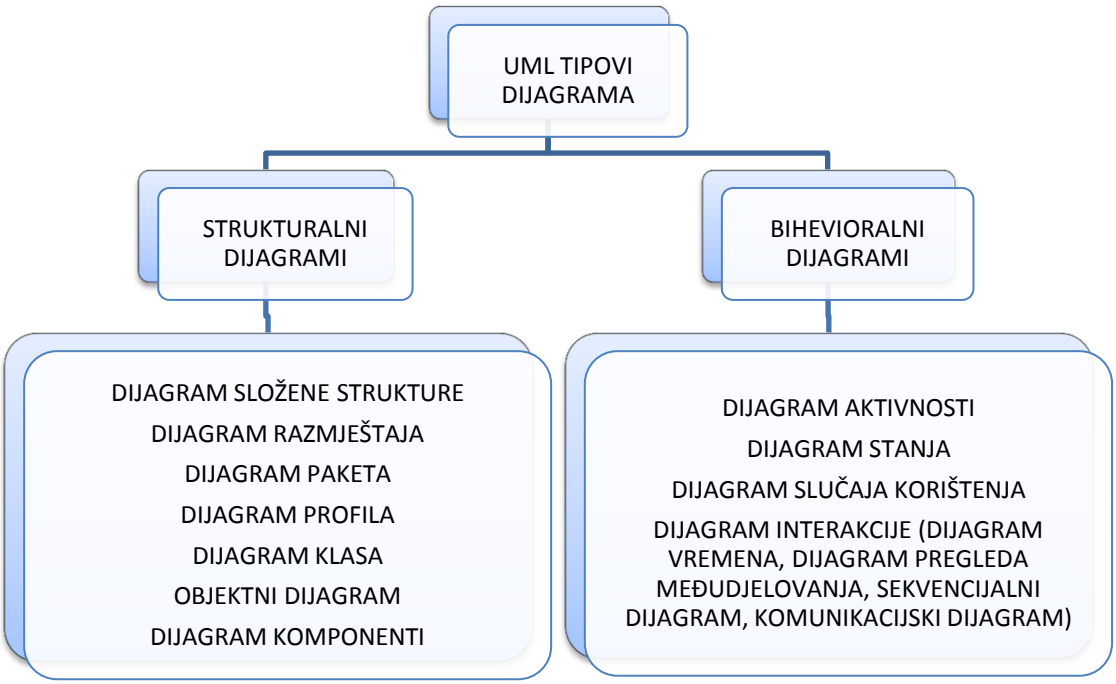
- ✓ ovisnost (engl. *dependency)* – semantička relacija gdje promjena jednog objekta utječe na promjenu ovisnog objekta.
- ✓ asocijacija (engl. *association)* – strukturalna relacija kao skup veza između objekata. Kombinacija predstavlja specijalnu vezu između cjeline i njezinih dijelova.
- ✓ generalizacija (engl. *generalization)* – relacija u kojoj se objekti specijaliziranih elemenata (djeca) mijenjaju prema objektima generaliziranih elemenata (roditelja). Djeca tako dijele strukturu i ponašanje roditelja.
- ✓ realizacija (engl. *realization)* – semantička veza između više klasifikatora gdje jedan specificira ugovor koji drugi klasifikator izvršava. Realizacija se ostvaruje između sučelja i klasa ili komponenata koje ih realiziraju te između slučajeva korištenja i suradnji koje ih realiziraju.



Sl. 2.1. Grafički prikaz relacija unutar UML modela.

2.2. DIJAGRAMI MODELA

Dijagram je grafička prezentacija skupa elemenata prikazana kao grafovi koji povezuju objekte i relacije u modelu u bilo kojoj kombinaciji. Dijagrami predstavljaju poboljšani prikaz elemenata koji čine sustav. U praksi se samo mali broj kombinacija pojavljuje, a te kombinacije su vezane s pogledima na najkorisnije dijelove sustava.

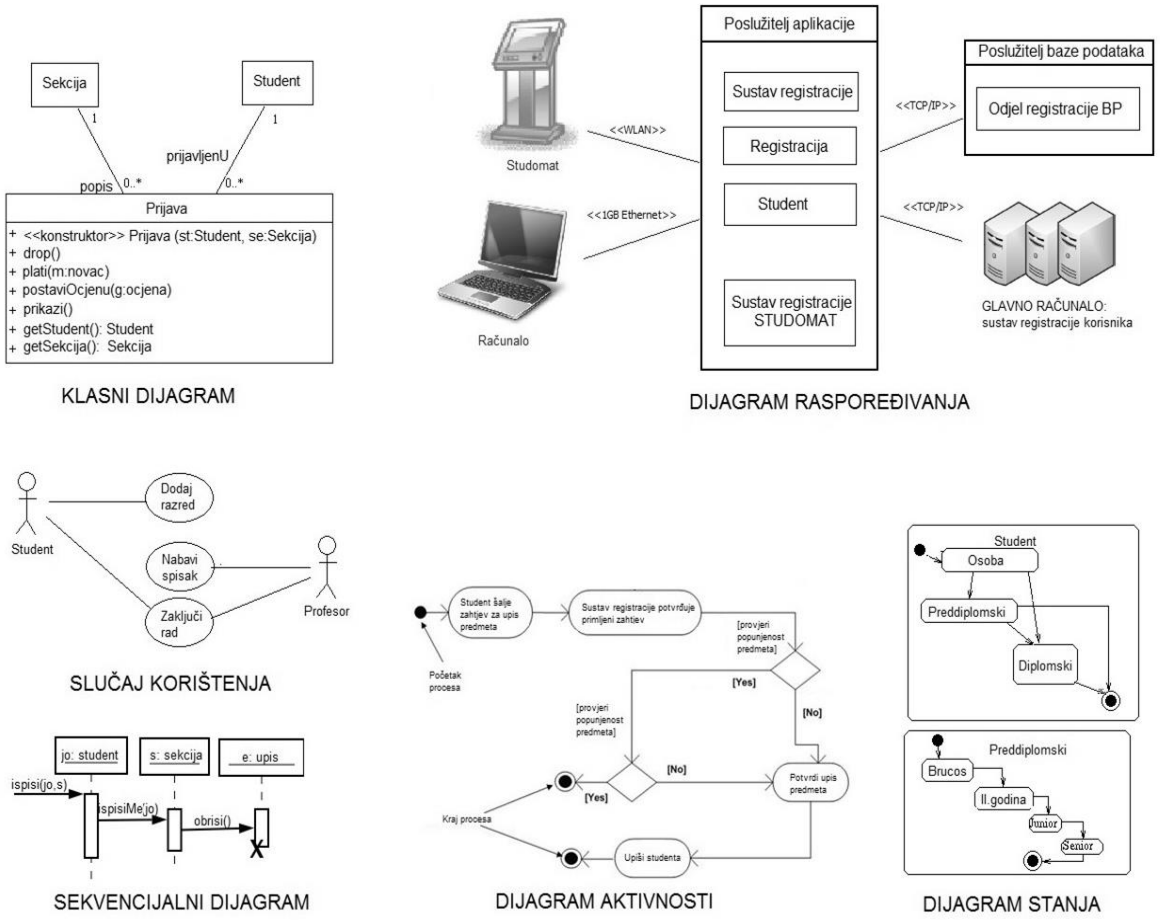


Sl. 2.2. Prikaz vrste UML dijagrama.

Prema slici 2.2. UML dijagrami prikazuju dva različita pogleda: strukturalni i bihevioralni pogled za prikaz sistema. Statički ili strukturalni pogled: naglašava statičku strukturu korištenjem objekata, atributa, operacija i odnosa. Strukturalni pogled uključuje klasni dijagram i ostale složene strukturalne dijagrame.

Dijagram klasa (engl. *class diagram*) opisuje statičnu strukturu sustava odnosno pokazuje kako se entiteti međusobno odnose. Koristi se za prikaz logičkih klasa poput banaka ili za prikaz implementacijske klase. Klasa predstavlja apstrakciju entiteta sa zajedničkim karakteristikama, a opisana je imenom, atributima i operacijama ili metodama. Aktivna klasa inicira tok aktivnosti, dok pasivna pohranjuje podatke te obavlja radnje za aktivnu klasu. Na slici 2.3. prikazani su neki od UML dijagrama koji prikazuju stanja i procese sustava.

U klasnom dijagramu klasa prijava je povezana s druge dvije klase: prijava mora sadržavati jednog studenta kao što mora imati i jedan predmet koji se prijavljuje. Student može napraviti neograničeni broj prijava. Svaki predmet (sekcija) može imati neograničen broj prijava, dok jedna prijava uključuje samo jedan predmet. Klasa prijava nema atribute, ali ima slijedeće operacije: prijava studenta, plati semestar, postavi ocjenu, prikazi listu, dohvati studenta i dohvati predmet.



Sl. 2.3. UML dijagrami prijave studenta.

Dijagram komponenti pruža fizički pogled na vremenski nepromjenjiva svojstva sustava. Njegova namjena je vizualizacija sustava, kreiranje izvršnih datoteka korištenjem inženjeringa unaprijed i unazad te opis organizacije i odnosa između komponenti. Komponenta je fizička i stvarna implementacija logičkih elemenata s pet stereotipa (izvršna datoteka, biblioteka, tablica baze podataka, dokument s kodom ili podacima te dokument sadržaja). Komponente međusobno komuniciraju putem sučelja koje predstavlja kolekciju operacija za specificiranje usluge razreda ili komponente povezane konektorima.

Dijagram raspoređivanja (engl. *deployment diagram*) pokazuje kako će sustav biti fizički implementiran u hardverskom okruženju. Pokazuje fizičko pokretanje različitih komponenti sustava i njihovu međusobnu komunikaciju. Sastoji se od čvorova, komponenti i njihove međusobne veze. Komponente izvršavaju sustav, a čvorovi su računalni resursi koji pokreću i izvršavaju komponente. Veza između njih se označava komunikacijskim protokolom ili tehnologijom koju koriste. Klasični blokovi se zamjenjuju vizualnim slikama radi lakšeg razumijevanja i boljeg izgleda. Prema slici 2.3. fizički prikaz sustava upisa se sastoji od osobnog računala ili uređaja za prijavu, poslužitelja na kojem se nalazi aplikacija, poslužitelja koji sadrži bazu podataka i glavnog računala. Između svih stanica označene su vrste komunikacijske veze.

Dijagram paketa (engl. package diagram) prikazuje povezanost dijelova sustava u logičke cjeline. Koristan je za prikaz hijerarhije razreda i njihovo grupiranje u funkcionalne cjeline te objašnjava funkciju modeliranog sustava. Paket predstavlja različite UML objekte. Svaki paket sadržava razrede različite vidljivosti dostupne samo hijerarhijski višem paketu. Između paketa postoji dva tipa ovisnosti: uvoz (elementi ciljanog paketa se unose u izvorni paket) i spajanje paketa (sadržaji paketa se kombiniraju). Elementi dijagrama su: paket (grupa sličnih elemenata i dijagrama), klasa (struktura i ponašanje objekta), sučelje (specifikacija ponašanja), objekt (instanca klase) i tablica (stereotip klase).

Dinamički ili bihevioralni pogled naglašava dinamičko ponašanje sustava pokazivanjem međudjelovanja između objekta te se mijenja u interna stanja objekta. Taj pogled uključuje dijagram slučaja korištenja, dijagram aktivnosti, dijagram stanja, sekvencijalni dijagram i komunikacijski dijagram.

Dijagram slučaja korištenja (engl. *use-case diagram*) su usluge ili funkcije koje sustav pruža korisniku. Vizualizira funkcionalne zahtjeve uključujući odnose aktera prema procesu i odnose između različitih slučajeva korištenja. Sastoji se od aktera, slučaja korištenja te njihovog međusobnog odnosa. Akteri mogu ostvariti više uloga u sustavu, a komuniciraju sa sustavom preko poruka (koje se ne prikazuju u ovom dijagramu). Iako se koriste samo ključni slučajevi važni za funkcioniranje sustava, dijagram se može proširiti posebnim slučajevima. Prema slici 2.3. na dijagramu slučaja korištenja postoje dva aktera, aktivni akter student koji inicira upis razreda te pasivni akter profesor koji obavlja radnje: nabavlja spisak, provjerava stanje te upisuje ako ima slobodnih mjesta. Student mora potvrditi upis stoga je zadnji slučaj korištenja iniciran s obje strane.

UML dijagram aktivnosti prikazuje proceduralni tijek kontrole između dva ili više objekata klase. Koristi se za kreiranje tijeka stanja procesa i unutarnjih operacija ili budućih ishoda istog zadatka. Prikazuje slijedne operacije na jednostavan način. Elementi dijagrama stanja su: aktivnosti, veze, uvjeti i ograničenja. Na slici 2.3. dijagram aktivnosti opisuje stanja u kojem se sustav može naći tijekom upisa studenta. Nakon što student podnese zahtjev za upisom, sustav provjerava slobodna mjesta. Ako ima slobodnih mjesta, student mora potvrditi upis, inače radnja završava. Provjera se može napraviti više puta.

Dijagram stanja (engl. *statechart diagram*) opisuje različita stanja u kojima se pojedina klasa može naći. Započinje ulaznim stanjem (puni krug) te završava izlaznim stanjem (tzv. meta). Između se prikazuju različita stanja objekta (stanja koja zadovoljavaju određene kriterije, izvode određene akcije ili čekaju na određeni događaj). Prijelaz između stanja sadržava događaj koji ga je izazvao, ograničenje ili akciju koja se događa ako je ograničenje zadovoljeno. Prijelazi su kratkog trajanja i ne mogu se prekinuti. Na slici 2.3. dijagram stanja opisuje izbor studenta: upisati preddiplomski ili diplomski studij (ovisno o uvjetima) te upisati diplomski studij nakon preddiplomskog. Student nakon svakog studija može završiti studiranje. Prvi dio se proširuje opisom tijeka preddiplomskog studija gdje student počinje kao brucoš, a završava nakon četvrte godine.

Sekvencijalni dijagram prikazuje izmjenu poruka između više objekata u vremenu. Opisuje detaljan tijek za specifični slučaj korištenja ili čak dio specifičnog slučaja korištenja. Dijagram sadrži dvije dimenzije: vertikalnu dimenziju koja pokazuje poruke u vremenskom redoslijedu događanja i horizontalnu dimenziju koja pokazuje instance objekta prema kojima je poruka poslana. Na slici 2.3. sekvencijalni dijagram opisuje vremenski tijek ispisa gdje objekt student podnosi zahtjev za ispisom (traže se podaci o studentu i predmetu) potom objekt predmet traži ispis studenta (traže se podaci samo o studentu) te objekt upis koji briše studenta s predmeta.

UML ima pravila za oblikovanje modela koji opisuju komponente i izvođenje dinamičkog modela sustava. Specifikacijom se opisuje sadržaj i izvedba građevnih blokova nakon čega se crtaju dijagrami kao rezultat spomenute specifikacije. Osim opisa atributa i operacija pojedinog građevnog bloka mogu mu se opisati i dodatne uloge poput pravo pogleda na atribut (javno, privatno i zaštićeno).

2.3. ZAHTJEVI NA SUSTAV

Veliki broj zahtjeva klijent ne navede prije razvojnog procesa ili se određeni problemi tek otkriju u stvarnom radu. Stoga je velika prednost UML-a njegova otvorenost prema promjenama u postojećem sustavu. Postojećim klasama se mogu nadodati dodatne klase s novim operacijama ili se mogu postaviti nova ograničenja klase odnosno objekta. Tijekom svih promjena potrebno je paziti da se ne mijenja tijek slanja informacija potrebnih za realizaciju. Poslovni ili korisnički zahtjevi (engl. *user requirements*) definiraju što sustav treba raditi u stvarnom vremenu. Što su ti zahtjevi bolje definirani, lakše je mijenjati proces tijekom razvoja. Kroz zahtjeve se prepoznaju ključne karakteristike koje opisuju aktivnosti i ponašanja sustava u određenim situacijama te karakteristike softvera i ograničenja na svojstva sustava.

Svaki korisnik elektronske biblioteke ima glavni zahtjev za pregledom ili kupnjom traženog medija. Zato je potrebno ostvariti upravljanje korisničkim računom, pregledom medija koji se skida te pružanjem odgovora na željenu uslugu. Administrator biblioteke traži zahtjev za ažuriranjem liste medija i liste korisnika usluge. Ujedno može obavljati sve druge poslove kao i sam korisnik. Aplikacija knjižnice mora omogućiti pretraživanje knjiga, provjeravati stanje liste medija te davati sadržaje medija spremljene u bazi. Zahtjevi su u ovom slučaju na strani korisnika: pretraga baze, pregled sadržaja i skidanje medija na vlastitu jedinicu za pohranu. Specificiranje slučaja korištenja je postupak detaljne identifikacije slučaja korištenja i prikaz u preglednoj formi prikazano u tablici 2.1.

Tab. 2.1. Specifikacija zahtjeva u elektronskoj biblioteci

<table>
<tr><th colspan="3">AKTERI</th></tr>
<tr><th>KORISNIK</th><th>ADMINISTRATOR</th><th>APLIKACIJA S BAZOM</th></tr>
<tr><td colspan="2">PRETRAGA MEDIJA</td><td>ISPIS REZULTATA</td></tr>
<tr><td colspan="2">PRIJAVA NA SUSTAV (PRIJAVA, REGISTRACIJA)</td><td>PROVJERA PODATAKA</td></tr>
<tr><td colspan="2">PREGLED REZULTATA KNJIGA</td><td>ISPIS SADRŽAJA KNJIGE</td></tr>
<tr><td colspan="2">ODABIR I PREGLED SADRŽAJA MEDIJA</td><td>ISPIS RAČUNA</td></tr>
<tr><td colspan="2">SKIDANJE KNJIGE</td><td>NAPLATA RAČUNA</td></tr>
<tr><td colspan="2">OCJENJIVANJE KNJIGE</td><td>PRIJEDLOG KNJIGA</td></tr>
<tr><td colspan="2">OZNAČAVANJE KNJIGE</td><td rowspan="2">SLANJE AUTOMATSKIH E-MAILOVA</td></tr>
<tr><td>SKIDANJE MEDIJA</td><td>NARUČIVANJE MEDIJA</td></tr>
<tr><td>PLAĆANJE MEDIJA</td><td>UREĐIVANJE BAZE KNJIGA I KORISNIKA</td><td rowspan="2"></td></tr>
<tr><td>KONTAKT – SLANJE PORUKA I UPITA</td><td>ODGOVOR KORISNIKU NA E-MAIL RAČUN</td></tr>
</table>

3. PREGLED ELEKTRONSKE BIBLIOTEKE

Internet se ponajviše koristi za prikupljanje podataka, slanje elektronske pošte i čitanje dnevnih novina. Sve je veći porast korištenja interneta u svrhu edukacije gdje digitalna biblioteka ima efikasan utjecaj. Elektronska biblioteka, digitalna ili hibridna knjižnica pruža korisnicima novi korak s vremenom i tehnologijom. E-biblioteka omogućava jednostavan i brži pristup medijima koje korisnik želi pogledati. Zamišljena je kao otvoreni sustav stvaranja, objavljivanja, pretraživanja i korištenja svih vrsta digitalizirane i multimedijske građe. Cilj biblioteke je pretraživanje velike količine medija iz udobnosti vlastitog doma u samo nekoliko sekundi te pohrana odabranog medija na vlastitu jedinicu zapisa. E-biblioteka može sadržavati pokaznice za odabrani medij ukoliko ne sadrži traženi pojam. Korisnika se može odvesti i do e-knjiga, e-časopisa, raznih multimedijskih sadržaja te internet stranica.

Tradicionalne knjižnice postupno prelaze na digitalni oblik zbog praćenja tehnologije i zaštite dokumenata kojima prijeti propadanje. Postupkom digitalizacije tiskana građa se prenosi u digitalni oblik postupkom snimanja, pohranjivanja i obrade sadržaja. Osim tiskane građe digitaliziraju se audio i video zapisi sa starih medija, a samim postupkom postoje minimalni gubici kvalitete i količine. Svaka digitalna knjižnica je nadograđivana na postojeći materijal u digitalnoj formi što je smanjilo potrebe za ljudskim resursom, vremenom i financijskim sredstvima. Međudjelovanje i održivost ključ su za viziju digitalnih knjižnica koje međusobno komuniciraju.

Digitalna i tradicionalna knjižnica imaju istu svrhu: pružiti korisniku medij (tekst, audio, video i grafika) no razlikuju se u nabavi materijala. Prednosti digitalne knjižnice su količina dostupnog medija i veći broj korisnika, prostor skladištenja, povezanost većeg broja knjižnica i cijena održavanja. Nedostaci digitalne knjižnice su previsoka cijena medija, dugotrajni proces digitalizacije starih knjiga te zahtjev za kompatibilnošću s najnovijim softverima i hardverima.

Elektronske biblioteke su fleksibilnije od tradicionalnih. Dostupne su na svakome mjestu, proširive su te kreiraju bazu svjetskog znanja povezivanjem više elektronskih knjižnica u jednu. Kvaliteta digitalnih knjižnica se mjeri prema razini usluge koja se pruža korisnicima u pogledu nalaženja odgovarajućih informacija, a ne prema broju nabavljenih medija. Svaki uređaj za pregled e-knjige može se personalizirati i proširiti, a preglednik prateći korisnikove interese nudi korisniku traženi odgovor bolje i brže na svakoj slijedećoj pretrazi. Pretraživanje velike količine medija i dalje ostaje veliki problem e-biblioteke što je zasad riješeno smanjenjem pretrage prema određenim kategorijama i uvođenjem meta podataka.

Meta podaci su od ključnog značaja za digitalni svijet. Opisuju sadržaj digitalne građe prema ključnim riječima što može uključiti temu građe, jezik u kojem je pisan itd. Meta podaci omogućuju lakše kreiranje, implementiranje, upravljanje i pretraživanje digitalne biblioteke.

Primarne aktivnosti digitalne knjižnice su odabir medija, organiziranje i pružanje prava pristupa te zadržavanje informacija unutar digitalnog svijeta. Veliki problem razvoja primarnih aktivnosti je tehnička infrastruktura koja je nužna za ostvarivanje zadataka. Fizička komponenta digitalne knjižnice uključuje hardver, softver, osoblje i prikladnu kolekciju medija. Softverski gledano, svaka se digitalna knjižnica realizira preko internet stranice što uglavnom uključuje rad s HTML-om, bazama podataka, različitim programima za grafički dizajn i OCR softverom (engl. *optical character recognition*) koji prema [1] olakšava digitalizaciju tiskanih medija.

Svaki digitalni medij sadrži tip opreme koja se koristi u procesu elektronske komunikacije odnosno čitanja, slušanja ili gledanja elektronskim putem. Elektronska knjiga (e-knjiga) je rađena za razmjenu te se može čitati na računalu ili čitaču e-knjiga poput Amazon Kindle, eReader Prestigio ili Barnes & Noble Nook Simple Touch pa čak i na običnim „pametnim“ telefonima.

Dobra programska podrška mora zadovoljiti rastuće potrebe korisnika. Za dobru programsku podršku dugotrajne kvalitete, potrebna je solidna bazna arhitektura. Razvojni proces je važan za kreiranje projekta unutar zadanog proračuna i vremena. Izgradnja digitalne knjižnice prvo traži obradu zahtjeva te određivanje glavnih slučajeva korištenja koji se mogu mijenjati tijekom izvedbe projekta. U ovom radu koristi se uniformni modularni jezik ili UML.

4. MODELIRANJE ELEKTRONSKE BIBLIOTEKE

U ranoj fazi razvijanja sustava koristi se strukturna analiza koja uključuje analizu i modeliranje sustava. Modeliranje je proces u kojem se utvrđuju najvažniji elementi sustava te veza između njih što se vizualno prikazuju primjenom tehnika grafičkog prikaza. Modeliraju se procesi, arhitektura te logička i fizička struktura sustava. Korištenjem različitih metoda i tehnika za modeliranje sustava povećava se kvaliteta softvera.

Prikazivanjem u obliku UML dijagrama olakšava se prepoznavanje najvažnijih slučajeva korištenja. Za opis sustava elektronske biblioteke te prikaz funkcija rada pojedinih dijelova koriste se slijedeći dijagrami koji opisuju aktere u sustavu te pojedine scenarije:

- dijagram slučaja korištenja
- dijagram aktivnosti
- dijagram stanja
- dijagram raspoređivanja
- klasni dijagram
- sekvencijalni dijagram.

4.1. DIJAGRAM SLUČAJA KORIŠTENJA

Slučajevima korištenja potrebno je modelirati korisničke zahtjeve i scenarije ispitivanja sustava. Ako se koristi metoda oblikovanja programske potpore zasnovana na slučajevima korištenja, potrebno je započeti s njima te iz njih izvesti strukturne i bihevioralne modele sustava. U drugom slučaju potrebno je osigurati konzistentnost prethodno spomenutih modela. Svaki slučaj korištenja mora imati veći dio korištenog sustava te biti razumljiv stručnjacima dane domene i programerima kako bi se iz njih razvijali ostali dijagrami.

Tijekom definiranja zahtjeva naručitelji projekta i projektni tim raspravljaju o slučajevima korištenja te mogućim scenarijima u kojima se opisuje tijek događaja, izvanredne situacije te paralelne aktivnosti. Slučajevi korištenja (engl. use cases) predstavljaju metodu preuzetu iz UML standarda. Temelji se na ideji scenarija, a opisuje moguće interakcije sustava. Slučaj korištenja opisuje funkcionalnost sustava kroz slučajeve razumljive korisnicima ili akterima. Tri temeljna elementa u modelima obrazaca uporabe su: slučajevi korištenja, akteri te odnosi ili relacije među akterima kako je prikazano u tablici 4.1. Detalji unutar dijagrama slučajeva korištenja mogu se predstaviti tekstom ili dodatnim dijagramima interakcije između elemenata sustava.

Tab. 4.1. Elementi u modelu obrasca korištenja.

ELEMENT	SLUČAJ UPORABE	AKTERI	GRANICA SUSTAVA
OPIS	Akcija koju sustav obavlja u interakciji s akterima sustava.	Skup uloga koje imaju korisnici u interakciji s slučajevima korištenja.	Granica fizikalnog sustava i različitih aktera koji su u interakciji s fizikalnim sustavom.

Akteri su jedini entiteti koji komuniciraju sa sustavom. To može biti drugi sustav ili vanjski uređaj. Akteri su uglavnom inicijatori koji teže određenom cilju. Mogu biti pasivni ili aktivni ovisno jer sami vrše radnju ili su sudionici za izvršenje određene radnje (tablica 4.2. i 4.3.).

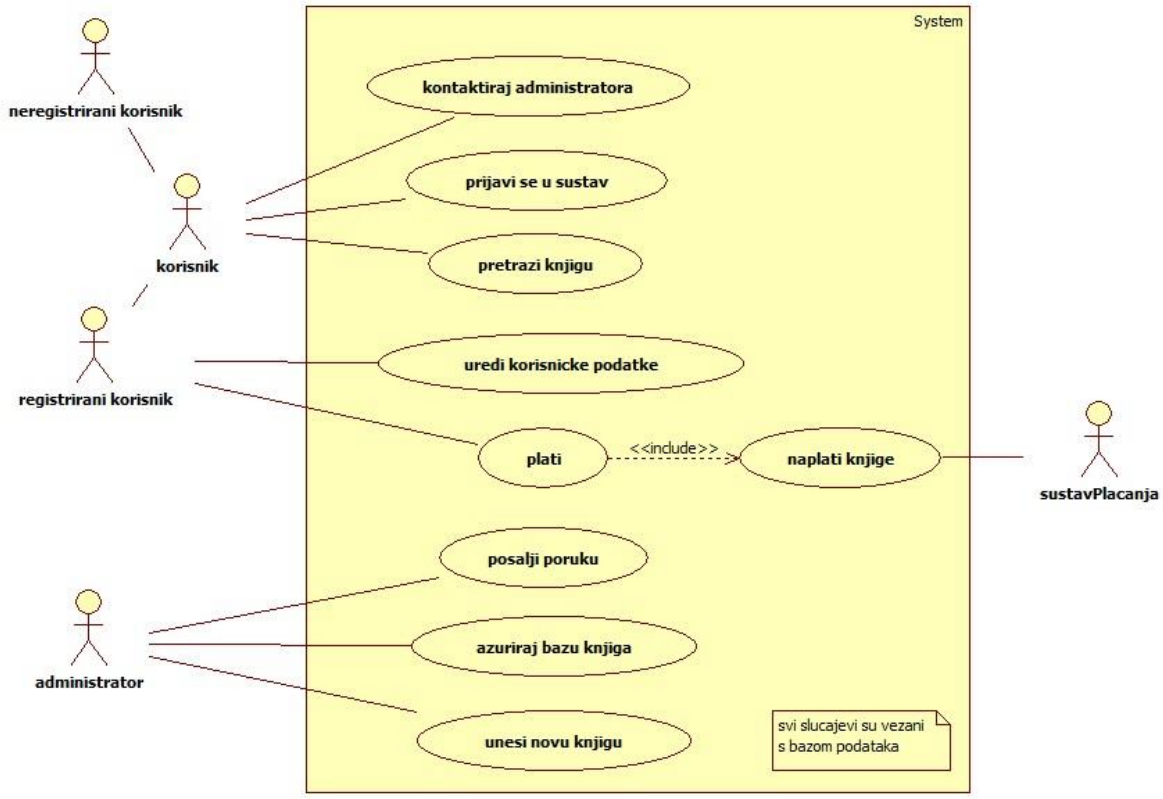
Tab. 4.2. Glavni akteri i njihove funkcije.

Korisnik	**Korisnik** je član elektronske biblioteke koji pregledava medije na internetu i skida ih na vlastito računalo.
Administrator	**Administrator** je knjižničar koji ažurira podatke o medijima i korisnicima sustava. Može pretraživati i pregledati knjige.

Tab. 4.3. Sporedni akteri i njihove funkcije.

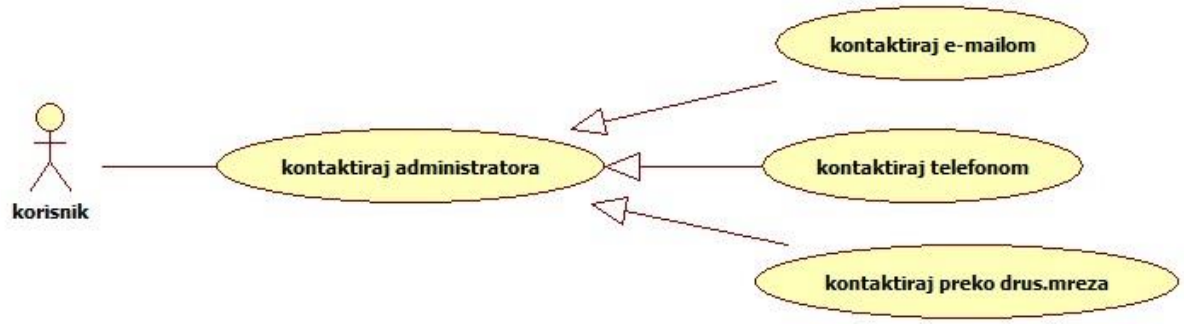
Baza podataka	**Baza podataka** knjižnice prikazuje rezultate pretrage i ispituje ispravnost korisničkih podataka. Uključuje bazu korisnika i knjiga.
Sustav plaćanja	**Sustav plaćanja** provjera iznos računa, stanje korisnikova računa te vrši naplatu. Na kraju šalje sustavu potvrdu plaćanja.

Glavni akteri u e-biblioteci su registrirani i neregistrirani korisnik koji su generalizirani u korisnika općeg pojma te administrator. Sporedni akteri su sustav plaćanja i baza podataka elektronske biblioteke. Na slici 4.1. korisniku su dane najvažnije funkcije: kontaktiraj administratora, prijavi se u sustav i pretraži knjigu dok su registriranom korisniku dane dodatne opcije: uredi korisničke podatke i plati knjige. Administrator ima slijedeće funkcije: pošalji poruku, ažuriraj bazu knjiga i unesi novu knjigu. Svi slučajevi korištenja osim kontaktiranja administratora su povezani s bazom podataka koja se ovdje ne spominje da ne djeluje kao glavni akter sustava. Slučaj korištenja *plati* inicijaliziran od korisnika uključuje slučaj *naplati knjige* kojeg provodi sustav plaćanja. Svi slučajevi korištenja nalaze se unutar granica sustava, a akteri se nalaze izvan njega.



Sl. 4.1. Dijagram slučaja korištenja elektronske biblioteke.

Korisnik može kontaktirati administratora kao što se vidi na slici 4.2. Kontaktiranje može ići putem e-maila, telefona ili društvenih mreža. Administrator biblioteke rješava upite odgovaranjem na e-mailove poslane preko kontakt-forme ili direktno na službeni e-mail račun, a opisan je slučajem korištenja *pošalji poruku* na slici 4.1.



Sl. 4.2. Prošireni slučaj korištenja „kontaktiraj administratora“.

Sl. 4.3. Prošireni slučaj korištenja „prijavi se u sustav“.

Na slici 4.3. generalizirani korisnik sudjeluje u prijavi. Ako korisnik nije registriran, poziva se akcija registriraj se, inače se poziva slučaj prijavi se s podacima. Svaka registracija uključuje potvrdu registracijskog koda koji se šalje korisniku na e-mail račun, a prijava zahtijeva autentifikaciju korisnika koju obavlja baza korisnika.

Ime: Registracija korisnika

Inicijator: Korisnik

Cilj: registrirati se na stranicu e-biblioteke

1. Aplikacija nudi formular za upis podataka
2. Korisnik ispunjava svoje podatke
3. Baza korisnika provjerava podatke i šalje registracijski kod
4. Korisnik potvrđuje registracijski kod
5. Baza korisnika registrira korisnika.

Dodaci:

3. Baza pronalazi korisničko ime ili e-mail kao postojeći u bazi korisnika

a. Aplikacija traži ponovni upis

b. Vrati se na korak 2.

3. Lozinka ne odgovara zadanom standardu

a. Aplikacija traži ponovni upis

b. Vrati se na korak 2.

4. Korisnik nije potvrdio registracijski kod ili ga nije dobio na e-mail račun

a. Korisnik traži ponovno slanje

b. Vrati se na korak 3.

4. Registracijski kod ne odgovara onom u bazi korisnika.

a. Vrati se na korak 4.

Ime: Prijava u sustav

Inicijator: Korisnik

Cilj: Prijaviti se na stranicu e-biblioteke

1. Aplikacija nudi korisniku prijavu na stranicu
2. Korisnik upisuje svoje korisničke podatke
3. Baza korisnika provjerava upisane podatke
4. Baza korisnika potvrđuje prijavu.

Dodaci:

2. Korisnik nije prethodno prijavljen

a. Aplikacija traži registraciju korisnika

b. Korisnik se registrira

c. Ponovi korak 2

4. Korisnički podaci nisu ispravni

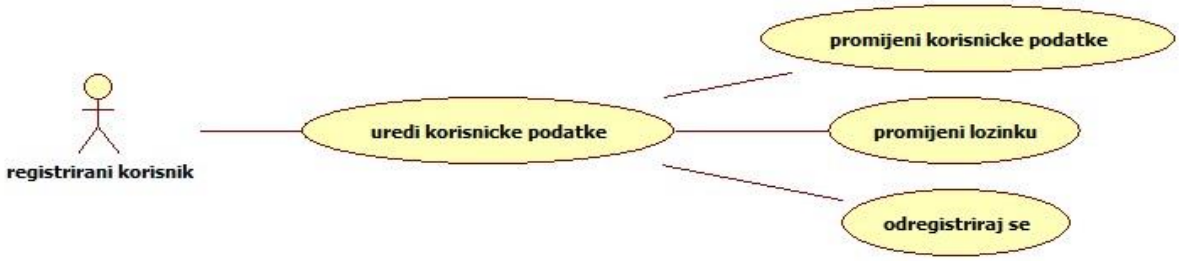
a. Aplikacija omogućava korisniku ponovni upis podataka

b. Vrati se na korak 2

4. Korisnički podaci nisu ispravni triput zaredom

a. Aplikacija nudi korisniku slanje korisničkih podataka na prijavljenu e-mail adresu

b. Vrati se na korak 2.



Sl. 4.4. Proširenu slučaj korištenja „uredi korisničke podatke“.

Na slici 4.4. prikazan je prošireni slučaj korištenja upravljanja korisničkim računom. Inicijator je registrirani korisnik, a slučaj korištenja uključuje promjenu korisničkih podataka (ime, prezime, kontakt, ...), promjenu lozinke i trajnu odjavu korisnika odnosno gašenje korisničkog računa.

Ime: Promjena korisničkih podataka

Inicijator: Korisnik

Cilj: Promjena postojećih podataka korisnika

1. Korisnik se prijavljuje
2. Korisnik odabire promjenu korisničkih podataka
3. Korisnik unosi tražene podatke
4. Baza korisnika provjerava podatke i ažurira ih.

Dodaci:

4. Korisnik nije promijenio postojeće podatke

b. Vrati se na korak 3.

4. Korisnik je unio krive podatke ili podatke koji nisu prema zadanoj formi.

    a. Vrati se na korak 3.

Ime: Promjena lozinke korisnika

Inicijator: Korisnik

Cilj: Promjena lozinke korisnika

1. Korisnik se prijavljuje
2. Korisnik odabire promjenu lozinke
3. Korisnik unosi staru lozinku i dvaput novu lozinku.
4. Baza korisnika provjerava i ažurira podatke.

Dodaci:

4. Sustav provjerava podatke, unesena je kriva trenutna lozinka

    a. Sustav nudi korisniku ponovni upis podataka

    b. Vrati se na korak 3

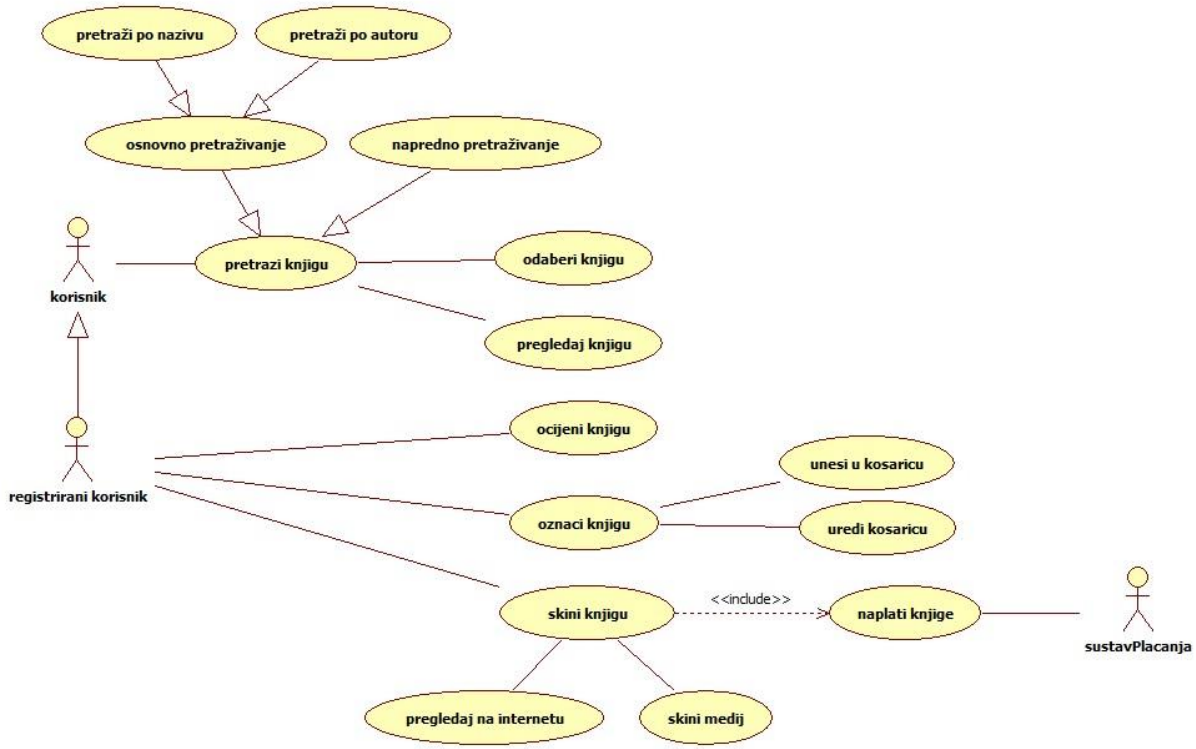
4. Sustav provjerava podatke, nove lozinke se ne podudaraju

    a. Sustav nudi korisniku ponovni upis podataka

    b. Vrati se na korak 3

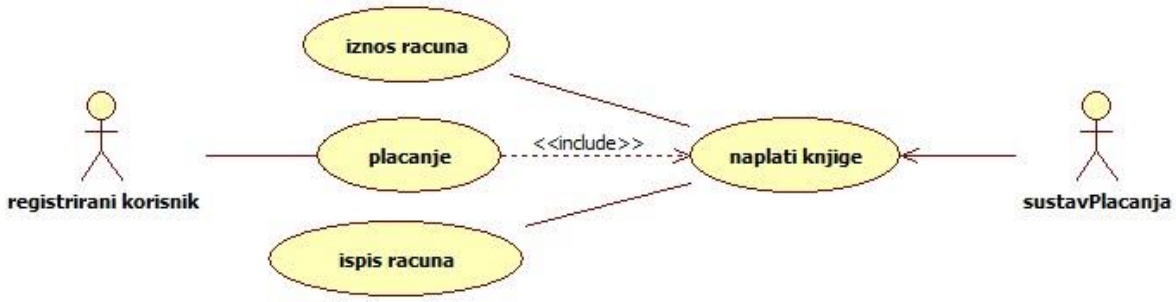
Na slici 4.5. obuhvaćen je proširenи slučaj korištenja pretrage knjige koji uključuje pet dodatnih slučajeva: odaberi knjigu, pregledaj knjigu, ocijeni knjigu, označi i skini knjigu. Pretraga se dijeli na osnovno i napredno pretraživanje. Kod osnovnog pretraživanja ključni pojmovi se mogu pretražiti prema nazivu ili autoru knjige.

Ostali slučajevi korištenja zahtijevaju prethodnu prijavu korisnika. Skidanje knjige uključuje provjeru plaćanja koju izvršava baza e-biblioteke, dok se naplata vrši preko sustava plaćanja. Registrirani korisnik može pročitati knjigu na internetu ili ju spremiti na vlastiti medij zapisa. Ocjenjivanje knjige daje prosječnu ocjenu svih registriranih korisnika, dok se označavanjem knjige stavljaju u korisnikovu košaricu za kasniji pregled što uključuje unos i uređivanje košarice.



Sl. 4.5. Proširenи slučaj korištenja „pretrazi knjigu“.

Ako se knjiga naplaćuje, aplikacija šalje račun te sustav plaćanja obavlja transakciju opisanu na slici 4.6. Korisnik potvrđuje plaćanje što uključuje slučaj *naplati knjige* od strane sustava plaćanja koji ima dodatne opcije: *provjeri iznos računa* i *ispiši račun* kao potvrdu obavljene transakcije. Kada aplikacija dobije potvrdu, korisniku je dozvoljeno skidanje knjige.



Sl. 4.6. Proširenи slučaj korištenja transakcija sustava plaćanja.

Ime: Pretraživanje knjige

Inicijator: Korisnik

Cilj: Skinuti medij iz baze podataka

1. Korisnik upisuje traženi pojam u tražilicu ili odabire iz ponuđene liste medija
2. Baza knjiga pretražuje pojam ili korisnikov odabir s liste
3. Aplikacija prikazuje rezultat pretrage
4. Korisnik odabire željeni medij
5. Aplikacija otvara sadržaj medija
6. Registrirani korisnik ocjenjuje medij
7. Registrirani korisnik označuje medij
8. Registrirani korisnik skida medij

Dodaci:

3. Baza knjiga nije našla traženi pojam

    a. Aplikacija nudi korisniku novu pretragu

    b. Vrati se na korak 1

6. (ili 7. ili 8.) Aplikacija provjerava je li korisnik prijavljen

    a. Ako korisnik nije prijavljen, aplikacija traži prijavu

    b. Korisnik se prijavljuje

    b. Nastavi s korakom 6 (ili 7 ili 8)

6. Registrirani korisnik je već ocijenio knjigu

    a. Aplikacija prikazuje samo prosječnu ocjenu

7. Registrirani korisnik je već označio medij

    a. Baza knjiga ne uvodi istu knjigu dvaput u košaricu

8. Baza knjiga provjerava plaćanje, knjiga je besplatna

    a. Korisnik može odmah skinuti knjigu

    b. Korisniku se otvara knjiga u pregledniku

8. Sustav provjerava plaćanje, knjiga se naplaćuje

a. Aplikacija šalje račun sustavu plaćanja

b. Sustav plaćanja naplaćuje korisnikov račun te šalje aplikaciji potvrdu plaćanja

c. Kada je potvrda dobivena, aplikacija šalje registriranom korisniku mogućnost skidanja

d. Knjiga se otvara u pregledniku

Komentar: Ako na internet stranici ne postoji mogućnost plaćanja, aplikacija može odbiti skidanje knjige i prikazati samo sadržaj navedene knjige. Plaćanje se provodi spajanjem s određenim sustavima naplate koji osim banke može biti bilo koja tvrtka za internet plaćanje poput Paypala.

Sl. 4.7. Proširenи slučaj korištenja „ažuriraj bazu knjiga“.

Na slici 4.7. administrator inicira ažuriranje baze knjiga koja uključuje promjenu podataka postojeće knjige i brisanje knjige iz baze knjiga. Za ta dva slučaja zahtijeva se prethodna pretraga knjige. Tek kada aplikacija prikaže sadržaj tražene knjige, administrator ju može mijenjati.

Ime: Ažuriranje baze knjiga

Inicijator: Administrator

Cilj: Ažurirati popis knjiga

1. Administrator pretražuje medij
2. Baza knjiga vraća rezultate pretrage
3. Administrator pregledava željeni medij
4. Administrator odabire promjenu sadržaja medija
5. Baza knjiga ažurira podatke

Dodaci:

2. Sustav prikazuje da nema rezultata za traženi pojam

a. Administrator pregledava listu nabavke

b. Administrator naručuje novi medij od dobavljača

c. Administrator unosi podatke o novom mediju

d. Nastavi s korakom 5

4. Administrator pregledava podatke o knjizi

a. Administrator unosi nove podatke o knjizi ako je nabavljena knjiga s drugim izdanjem.

b. Nastavi s korakom 5

4. Administrator pregledava status posudbe ili skinutu količinu medija

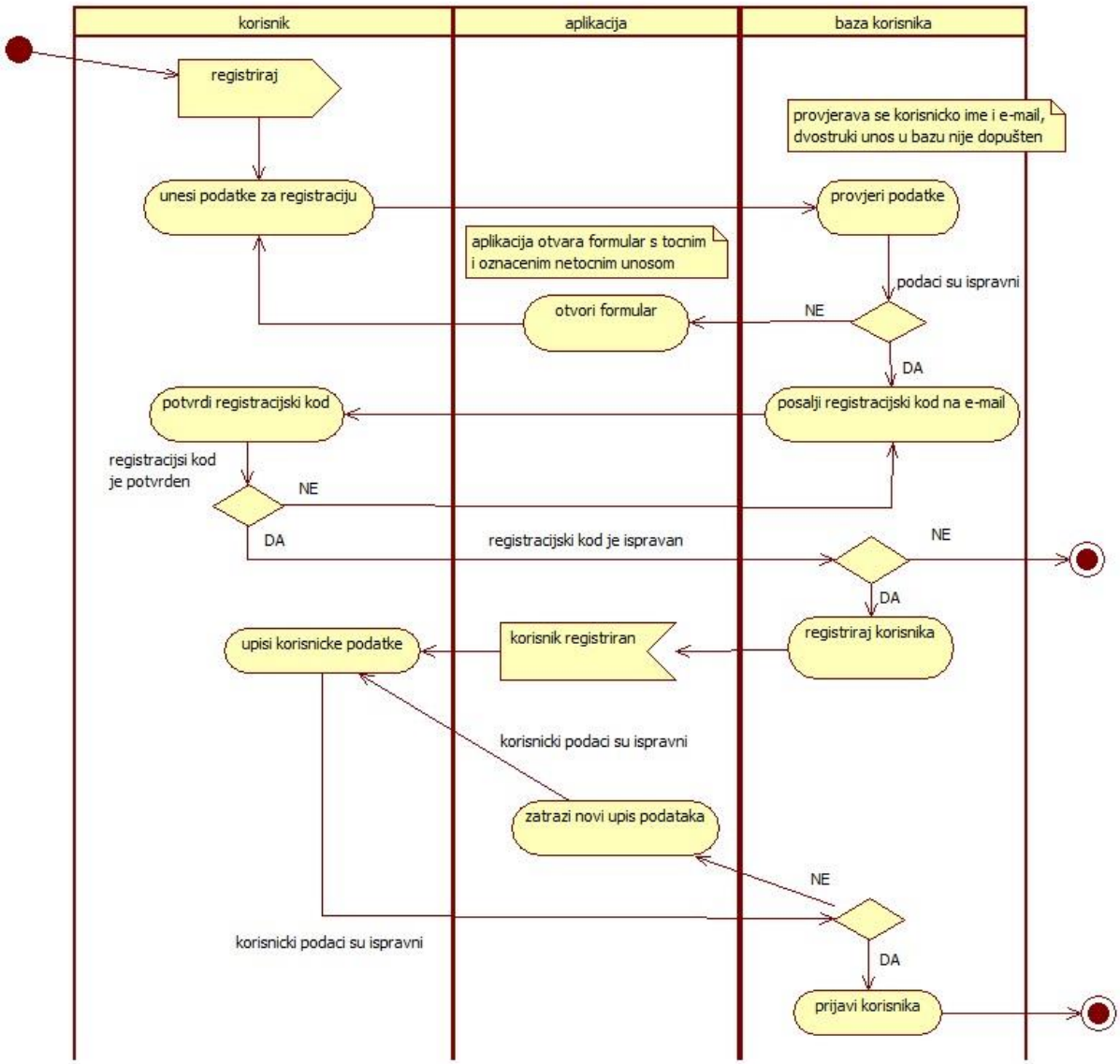
a. Ako medij nije bio pregledan ili skinut duže vrijeme, administrator sprema knjigu u arhivu starih medija ili ju briše.

b. Nastavi s korakom 5.

4.2. DIJAGRAM AKTIVNOSTI

Dijagram aktivnosti se sastoji od tri aktera prikazana u plivajućim trakama (engl. *swimlanes*): korisnik, aplikacija e-biblioteke i baza korisnika (slika 4.8.). Početni događaj prikazan je punim krugom s lijeve strane prve plivajuće trake. Prvi događaj započinje kada korisnik pošalje signal za registraciju. Strelica označava kretanje aktivnosti. Aktivnost aktera označena je elipsom. Korisnik u slijedećoj aktivnosti upisuje podatke za registraciju. Strelica vodi do baze korisnika gdje se provjeravaju podaci (ispravan e-mail, podudaranost s postojećim korisničkim imenom i sl.). Podaci su smješteni u bazi podataka gdje se odlučuje o ispravnosti. Odluka je predstavljena rombom. Ako su podaci neispravni, strelica vraća aktivnost na aplikaciju koja otvara formular za ponovni upis podataka, u suprotnom baza automatski šalje korisniku registracijski kod na prijavljeni e-mail račun. Unutar svoje trake korisnik mora potvrditi kod dobiven na e-mail te se isti kod uspoređuje u bazi korisnika. Sustav potom registrira korisnika te aplikacija navodi korisnika na prijavu. Korisnik se mora prijaviti na sustav s registriranim podacima. Ako su podaci neispravni, baza preko aplikacije traži ponovni upis podataka, inače baza prijavljuje korisnika i time aktivnost završava.

Kraj događaja označava se „metom“ koja izlazi na desnoj strani plivajućih traka. Žuti kvadratići označavaju komentare koji nemaju aktivnu funkciju, ali pomažu čitatelju razumjeti događaj.



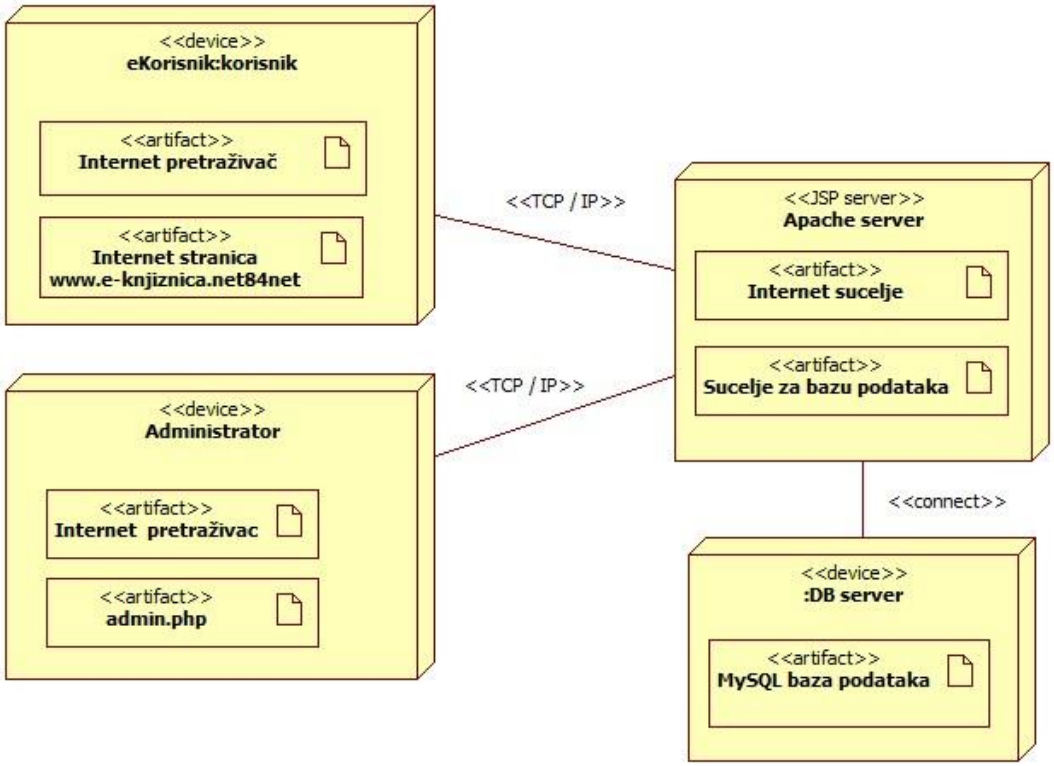
Sl. 4.8. Dijagram aktivnosti registracije i prijave korisnika.

4.3. DIJAGRAM RASPOREĐIVANJA

Dijagram raspoređivanja prikazuje hardversku i softversku arhitekturu sustava. Čvor korisnik sadržava internet stranicu E-biblioteke i internet pretraživač koji je potreban korisniku za pregled internet stranice skinute na korisnikovo računalo. Internet aplikaciju uređuje administrator koji pregledava stranicu preko internet pretraživača te koristi internet aplikaciju admin.php za njezino uređivanje. Oba čvora prikazuju računalo korisnika odnosno administratora.

U čvoru internet poslužitelja, koji djeluje kao razvojna okolina za stranicu, nalazi se internet sučelje i sučelje za bazu podataka. Čvor servera je označen kao JSP server (engl. *Java* Server *Pages*) koji je u ovom primjeru web poslužitelj otvorenog koda Apache. Stranica se preko poslužitelja povezuje s bazom podataka rađenoj u MySQL okruženju (programu otvorenog koda). Između dva korisnika i internet poslužitelja postoji komunikacijska veza koja je prikazan oznakom TCP/IP.

Ovo je jednostavna konfiguracija koja opisuje funkciju elektronske biblioteke na mreži u stvarnom vremenu (slika 4.9.).

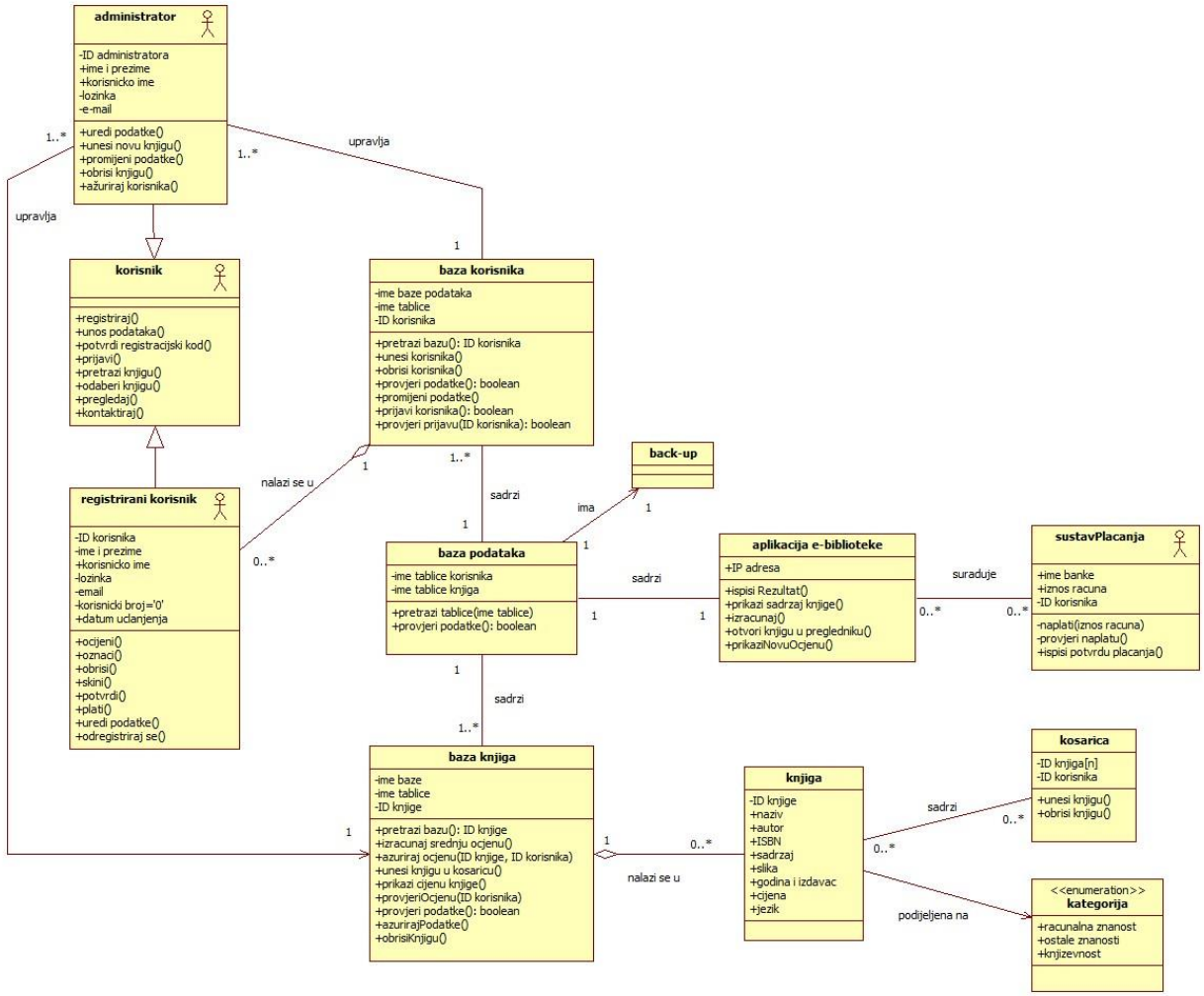


Sl. 4.9. Dijagram raspoređivanja elektronske biblioteke.

4.4. KLASNI DIJAGRAM

Klasni dijagram se koristi za vizualizaciju i povezivanje elemenata statičkog dijela sustava, opis funkcionalnosti izvođenja te dokumentiranje i izgradnju izvršnog koda softverske aplikacije. Klasni dijagram prikazuje klase sustava odnosno instance objekta koji sudjeluju u procesu. Dohvaća klase modela te sadržava atribute (varijable), operacije (funkcije) i odnose prema drugim klasama koji su važni za trenutni dijagram odnosno sustav.

U ovom slučaju glavni akter je član knjižnice ili korisnik elektronske biblioteke. Za svakog sudionika slučaja korištenja potrebno je odabrati oznaku uloge i dati tekstualni opis kako je prikazano u tablici 4.2. i 4.3. Administrator ima iste operacije kao korisnik uz dodatno ažuriranje baze podataka. Baza podataka sadržava imena tablica (s podacima o korisniku ili knjizi) u kojima se vrši pretraga. Aplikacija prikazuje rezultate pretrage i sadržaj knjige, a podacima pristupa preko ID-a.



Sl. 4.10. Klasni dijagram elektronske biblioteke.

U klasnom dijagramu (slika 4.10.) nalaze se četiri glavne klase: generalizirani korisnik, baza podataka, aplikacija e-biblioteke i sustav plaćanja. Klasa *korisnik* ima dvije podklase: *registrirani korisnik i administrator* (koji može imati funkcije običnog korisnika). Klasa *aplikacija e-biblioteke* povezana je s još dvije klase: *bazom podataka* (baza korisnika i baza knjiga) te *sustavom plaćanja* koji obavlja naplatu knjige.

Svaka klasa označena je jedinstvenim imenom klase, svojim atributima koji opisuju što klasa sadržava te operacijama koje označavaju radnju koje klasa može obaviti unutar sustava. Klasa *aplikacija e-biblioteke* sadrži atribut IP (*engl. Internet protocol*). IP adresa označava lokaciju na internetu preko koje korisnik pristupa aplikaciji. Operacije dodijeljene ovoj klasi su *ispisiRezultat()* pretrage baze, *prikaziSadrzajKnjige*, *izracunaj* cijenu knjiga koje korisnik želi skinuti, *otvoriKnjiguUPregledniku* i *prikaziNovuOcjenu* nakon ocjenjivanja.

Klasa *baza podataka* povezana je s dvije klase (*baza korisnika i baza knjiga*). Baza korisnika sadržava ime baze i imena tablica koji su potrebni za vezu s općom bazom podataka te ID korisnika koji je jedinstveni ključ pretrage. Operacija ove klase uključuju akcije vezane za prijavu korisnika, promjenu njegovih korisničkih podataka te pretraživanje baze gdje klasa vraća određenog korisnika prema njegovom ID-u ili korisničkom imenu. Klasa *baza knjiga* sadržava jednake atribute i slične operacije kao prethodno spomenuta klasa, ali se ovi podaci odnose na knjige unutar baze. Ova klasa je cjelina sastavljena od klase *knjiga* koja sadržava atribute: ID knjige, naziv, autor, ISBN, godina i izdavač preko kojih se vrši pretraga baze te dodatne atribute poput sadržaja, slike, cijene i jezika koji dodatno opisuju određenu knjigu. Klasa *knjiga* je podijeljena (grupirana) na navedene tri kategorije: računalna znanost, ostale znanosti i književnost (slika 4.10.) što skraćuje pretraživanje baze knjiga. Svaka knjiga može biti sadržana u korisničkoj košarici koja unosi i briše knjige prema ID-u korisnika i ID-u knjige. Svaka *baza podataka* treba imati svoju sigurnosnu kopiju (engl. *back-up*) u slučaju oštećenja ili gubitka podataka.

Nakon pretrage korisnik može skinuti knjigu samo ako se potvrdi da je knjiga besplatna ili plaćena. Provjeru cijene knjige aplikacija radi preko atributa *cijena* u klasi *knjiga*, dok provjeru plaćenog računa izvršava klasa *sustav plaćanja* koja se veže na klasu *aplikacija e-biblioteke*. Klasa *sustav plaćanja* sadržava ime banke, iznos računa i ID korisnika, dok od operacija sadrži provjeru naplate, naplatu računa prema korisniku te ispis potvrde plaćenog računa. Nakon pozitivne potvrde, sustav plaćanja šalje aplikaciji poruku da je račun plaćen.

Aplikacija e-biblioteke povezana je s klasom *korisnik* koja je dio klase *baze korisnika*. Klasa registrirani korisnik sadržava slijedeće atribute: ID korisnika, ime i prezime, korisničko ime i lozinku te e-mail. Ovi podaci razlikuju korisnike te pomažu tijekom prijave korisnika na sustav. Samo registrirani korisnik može obaviti operacije vezane za uređivanje korisničkih podataka te skidanje knjige.

Klasa korisnik je generalizirana klasa koja se sastoji od dvije podklase: *registrirani korisnik* (koji ima dodatna dva atributa: datum učlanjenja i korisnički broj koji označava razinu pristupa podacima) i *administrator*. Klasa administrator se može ponašati kao klasa korisnik, ali ima dodatne mogućnosti. Administrator može ažurirati i obrisati knjige unutar baze podataka te riješiti upite i probleme korisnika. Atribut koji razlikuje administratora od registriranog korisnika je korisnički broj koji se u ovom slučaju izjednačuje s "1". Aplikaciju mogu koristiti i neregistrirani korisnici koji se ne moraju prijaviti, ali bez prijave ne mogu skidati knjige.

Aplikacija e-biblioteke ne mora surađivati sa sustavom plaćanja jer knjige mogu biti besplatne, ali mora sadržavati barem jednu bazu korisnika i jednu bazu knjiga. Baza knjiga sastoji se od neograničenog broja knjiga koje se dijele na tri numeričke kategorije. Baza korisnika e-biblioteke ima neograničen broj korisnika koji koriste jednu aplikaciju e-biblioteke. Korisnici se nalaze unutar baze korisnika kojom upravlja jedan ili više administratora jednako kao i bazom knjiga. Baza korisnika i baza knjiga su sadržane unutar baze podataka povezane s aplikacijom e-biblioteke iako se ovo može proširiti ako se postojeći sustav poveže s drugim bibliotekama. Onda se baza podataka proširuje na više aplikacija e-biblioteke gdje se svaka knjiga iz drugog sustava povezuje preko spomenutih hiperlinkova.

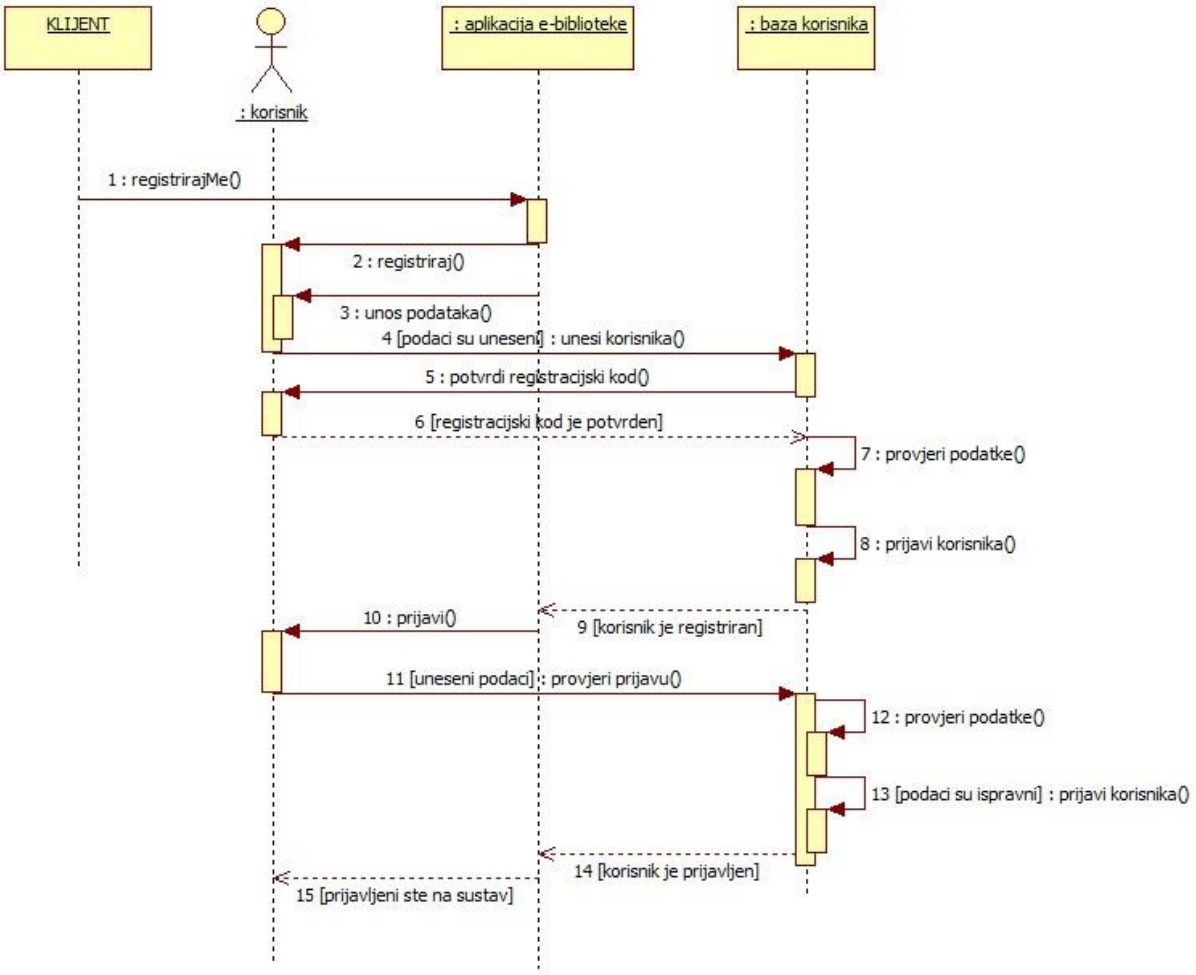
4.5. SEKVENCIJALNI DIJAGRAM

Sekvencijalni dijagram prikazuje interakciju između objekata prema vremenskom redoslijedu u kojemu se te interakcije odvijaju. U dijagramu na slici 4.11. upisana su tri objekta: korisnik, aplikacija e-biblioteke i baza korisnika. Svaki objekt predstavljen je vertikalnom isprekidanom linijom koja se naziva „linija života“ te je definiran u određenom vremenu. Između dva objekta šalje se poruka koja sadržava redni broj i naziv poruke. Označava se strelicom usmjerenoj prema primatelju.

Svaka poruka može imati povratni efekt ili ne mora vraćati povratnu informaciju što se označava isprekidanom strelicom otvorenog trokuta (oznaka *return()*). Puna linija s otvorenom strelicom je poslana poruka (*engl. send*) ovisno o ishodu prethodne akcije drugog objekta.

Vanjski korisnik (nazvan klijent) poziva određenu akciju koja je u ovom slučaju registracija korisnika. Aplikacija šalje zahtjev korisniku *registriraj()* te traži unos podataka. Kada korisnik unese podatke, baza korisnika ih unosi te šalje registracijski kod korisniku na potvrdu. Ako je registracijski kod potvrđen, baza korisnika *provjeri podatke* te *prijavi korisnika.* Potom vraća poruku aplikaciji da je korisnik registriran te aplikacija šalje korisniku poziv *prijavi se.*

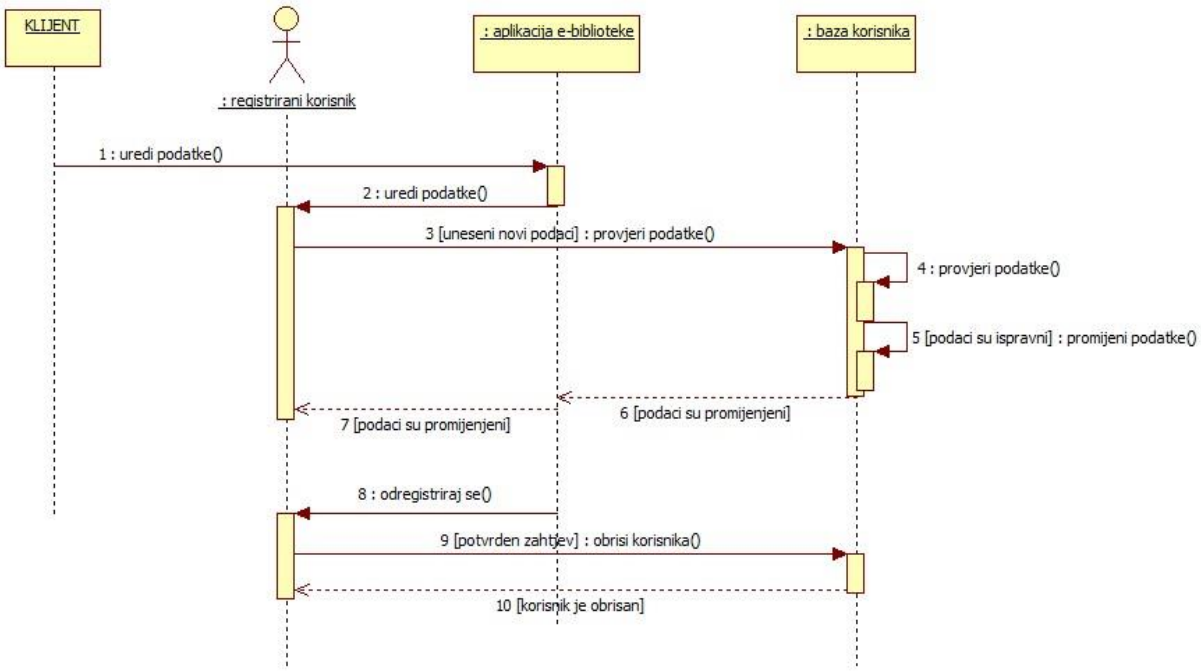
Nakon unosa podataka baza *provjeri podatke* te *prijavi korisnika*. Aplikacija vraća korisniku potvrdnu poruku da je prijavljen.



Sl. 4.11. Sekvencijalni dijagram prijave korisnika.

Samo registrirani korisnik može mijenjati svoje podatke ili korisničku košaricu. Prema slici 4.12. klijent pokreće zahtjev za promjenom podataka. Aplikacija šalje poziv korisniku *uredi podatke()* nakon čega baza korisnika *provjeri podatke().* Podaci moraju biti ispravni i različiti od starih. Ako su podaci ispravni, baza korisnika *promijeni podatke()* te aplikacija obavještava registriranog korisnika.

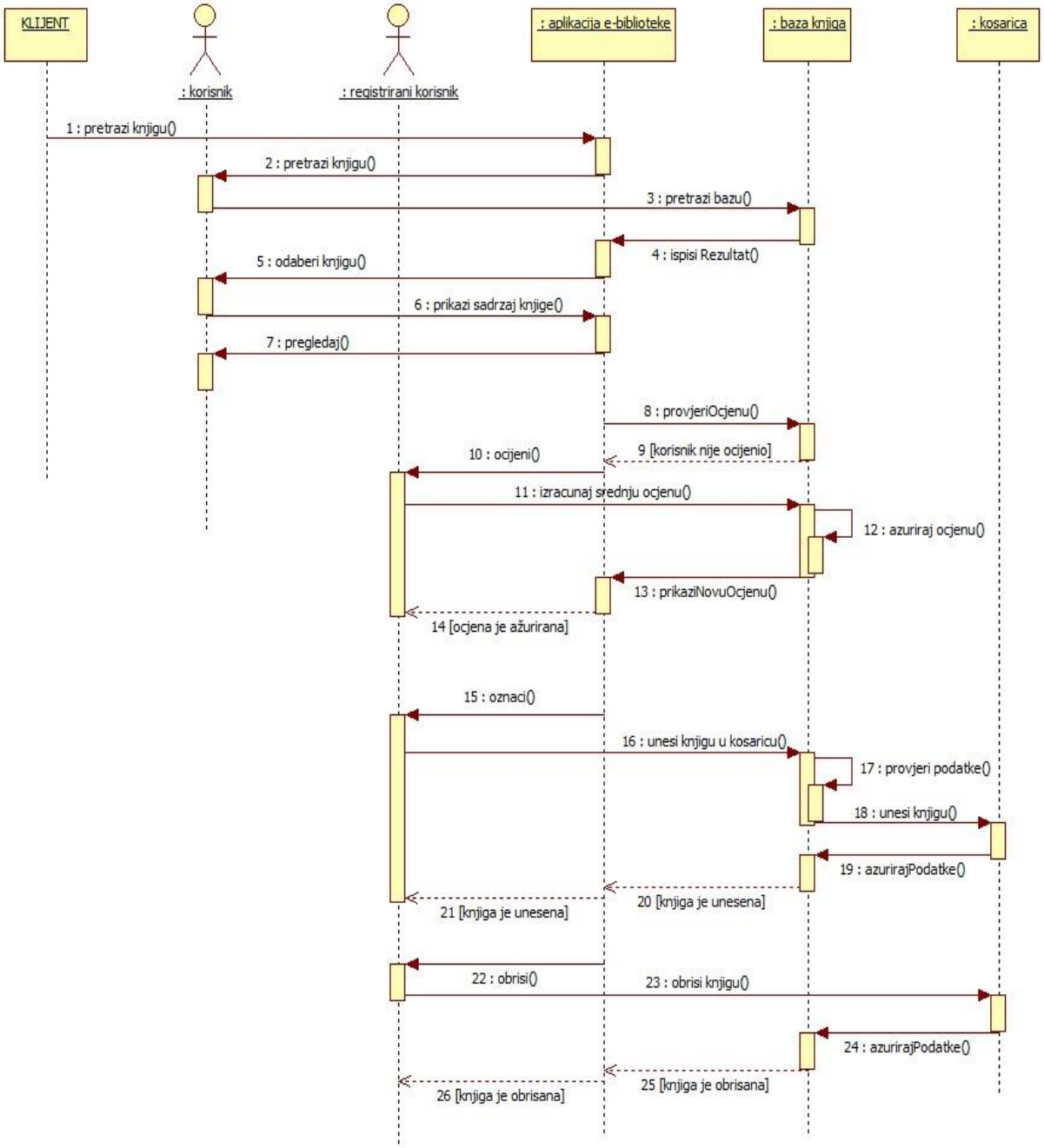
Korisnik se može odjaviti trajno s liste korisnika. Korisnik šalje poziv aplikaciji e-biblioteke koja traži potvrdu korisnika *odregistriraj se()*. Nakon što korisnik potvrdi, baza korisnika prima poziv *obriši korisnika()* s liste.



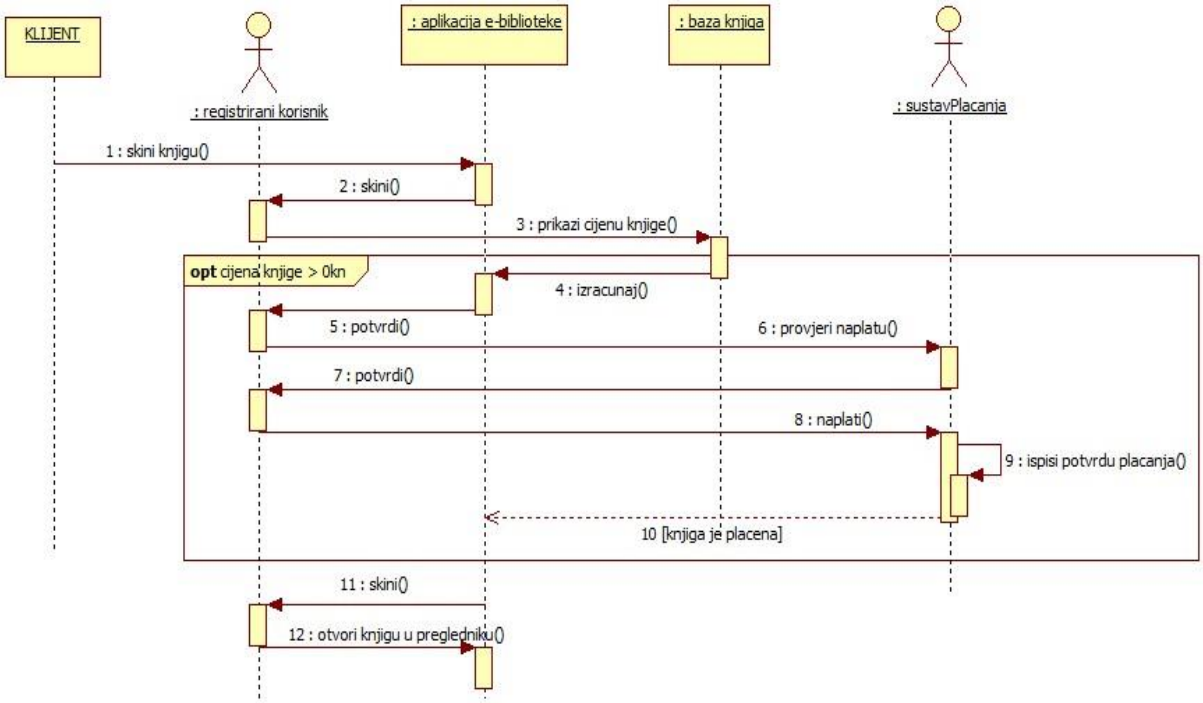
Sl. 4.12. Sekvencijalni dijagram promjene podataka korisnika.

Unutar sekvencijalnog dijagrama na slici 4.13. nalazi se pet objekata: korisnik, registrirani korisnik, aplikacija e-biblioteke, baza knjiga i košarica. Klijent poziva novu funkciju pretrage knjige. Aplikacija otvara formu korisniku da *pretraži knjigu()*. Nakon što korisnik unese pojam pretrage, baza knjiga *pretraži bazu()*. Dobiveni rezultati se šalju aplikaciji koja ispisuje rezultate. Korisnik odabire knjigu te pregledava detalje nakon što aplikacija prikaže sadržaj knjige.

Drugi dio podijeljen je u tri dijela: ocjenjivanje knjige, označavanje i brisanje knjige iz košarice te je vezan samo za registriranog korisnika. Kod ocjenjivanja aplikacija *provjeri ocjenu* u bazi knjiga. Ako korisnik nije ocijenio knjigu, aplikacija šalje korisniku poziv *ocijeni knjigu.* Ako je korisnik obavio poziv, baza knjiga izračuna srednju ocjenu te ažurira podatke na temelju ID knjige i unesene ocjene korisnika. Kod označavanja knjige postoje dva različita zahtjeva: korisnik javlja aplikaciji da želi označiti knjigu ili izbrisati knjigu iz košarice. Nakon obje poruke sustav šalje signal bazi knjiga *ažuriraj podatke()*. Ako se knjiga ne nalazi u košarici, košarica će unijeti knjigu te poslati poziv bazi da se ažurira. Baza knjiga šalje poruku aplikaciji da je poziv odrađen. Tada aplikacija vraća poruku korisniku da je knjiga unesena.

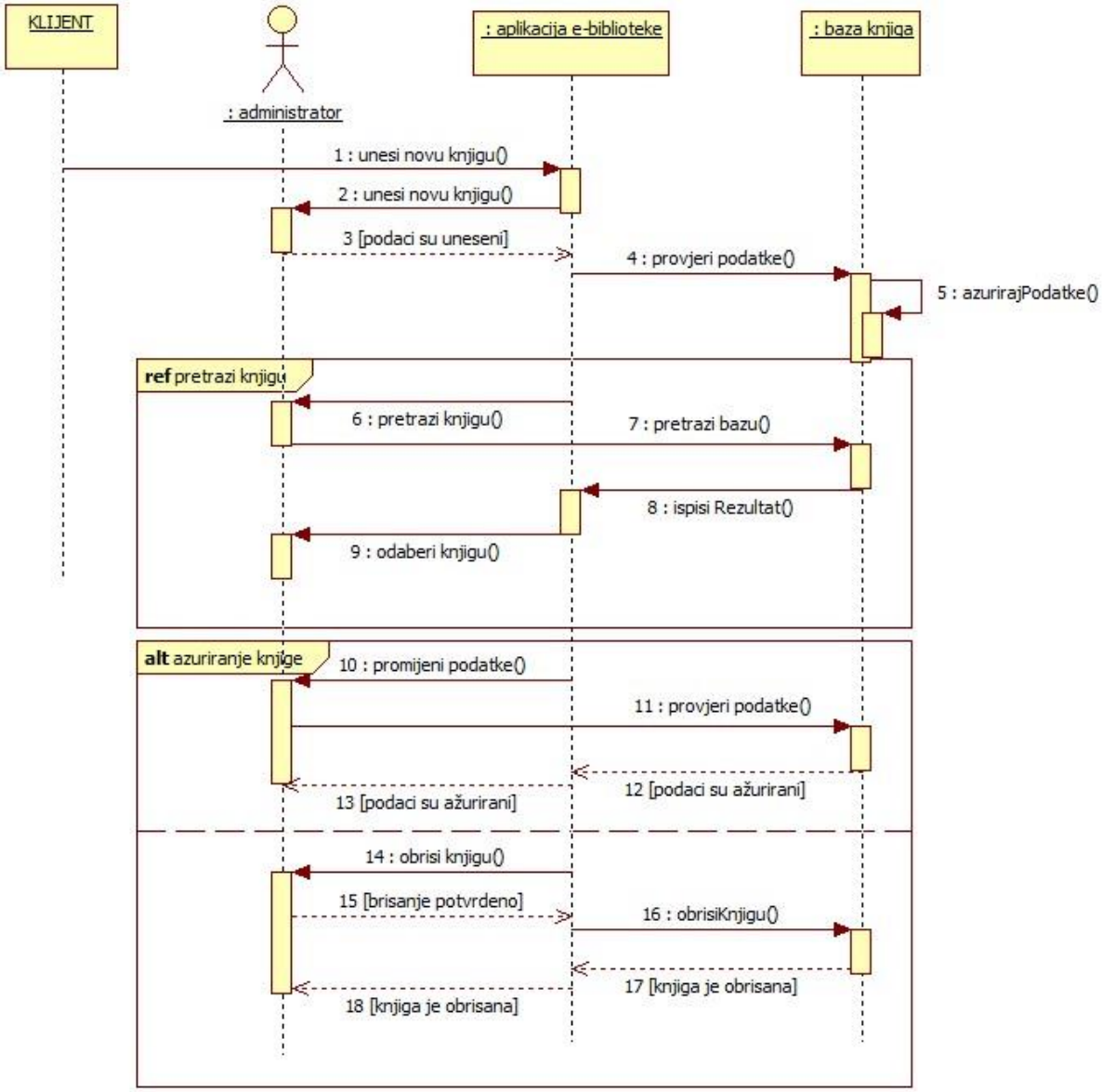


Sl. 4.13. Sekvencijalni dijagram pretrage knjige.



Sl. 4.14. Sekvencijalni dijagram skidanja knjige.

Klijent prvo poziva funkciju skidanja knjige. Registrirani korisnik se može odlučiti za skidanje knjige nakon čega baza knjiga *prikazuje cijenu knjige.* Prema slici 4.14. aplikacija izračuna ukupni račun te šalje poziv korisniku da *potvrdi plaćanje.* Sustav plaćanja provjerava naplatu te zahtijeva od registriranog korisnika da *potvrdi plaćanje.* Ako je korisnik potvrdio, sustav plaćanja naplaćuje račun preko korisnikovog bankovnog ili internet računa te ispisuje potvrdu plaćanja. Sustav plaćanja šalje poruku aplikaciji da je račun plaćen koja potom šalje poziv korisniku *skini knjigu().* Kada registrirani korisnik klikne na skidanje, aplikacija *otvara knjigu u pregledniku.* Ovo je opcionalno, gdje je uvjet za odrađivanje da je cijena knjige veća od nula kuna odnosno da nije besplatna. Ako je knjiga besplatna, zadnja poruka se automatski prosljeđuje korisniku, tražena knjiga se otvara u pregledniku te ju samo registrirani korisnik može pohraniti na vlastiti medij zapisa.



Sl. 4.15. Sekvencijalni dijagram ažuriranja baze knjiga.

Na slici 4.15. prikazan je sekvencijalni dijagram ažuriranja baze knjiga koji sadrži tri objekta: administratora koji jedini može mijenjati bazu knjiga, aplikaciju e-biblioteke te bazu knjiga. Kao i kod promjene korisničkih podataka, klijent poziva funkciju promjene baze podataka, potom aplikacija provjerava prijavu korisnika ili administratora u ovom slučaju. Taj dio se ne spominje već se sekvencijalni dijagram na slici 4.11. od poziva broj 10 može uključiti u ostale dijagrame kao polazna točka.

Administrator odabire jedno od tri načina rada: unos nove ili ažuriranje (promjena ili brisanje) postojeće knjige u bazi knjiga. Aplikacija poziva administratora da *unese novu knjigu.*

Kada su podaci uneseni i provjereni (ID i ISBN knjige se mora razlikovati od postojećih knjiga), baza knjiga *ažurira podatke*. Druge dvije opcije zahtijevaju pretragu knjige pa se prije njih uključuje referentni okvir *pretraži knjigu* sa slike 4.13. Nakon odabira knjige, aplikacija šalje poziv administratoru da *promijeni podatke*. Kada su podaci uneseni, baza knjiga *provjerava podatke* (ID i ISBN knjige moraju ostati isti) te ih *ažurira.* Nakon ažuriranja podataka, baza knjiga šalje poruku aplikaciji koja vraća poziv administratoru da je promjena izvršena.

Pri brisanju knjige administrator mora potvrditi zahtjev *obrisi knjigu* nakon čega se knjiga briše iz baze knjiga koja vraća poruku aplikaciji da je knjiga obrisana.

5. IZRADA APLIKACIJE

5.1. PROGRAMSKI JEZICI ZA IZRADU INTERNET APLIKACIJE

HTML (*engl. HyperText Markup Language*) je glavni prezentacijski jezik koji pretvara tekstualni dokument u internet stranice i aplikacije. Osnovna namjera HTML kao prezentacijskog jezika je pružanje semantičkog opisa sadržaja stranice i uspostavljanje strukture odnosno hijerarhije elemenata. Opisuje stranicu, ali ne izvršava nikakve zadaće [4].

Svaki HTML dokument izgrađen je od HTML elemenata koji sadrže oznaku (*engl. tag*) i atribute za opis svojstva elementa. Oznake su kodovi za označavanje tipa oblikovanja teksta. HTML element može sadržavati tekst, dodatne oznake, komentare i druge tekstualne sadržaje.

< početak_imeOznake atributi> HTML element </kraj_imeOznake>

Tab. 5.1. Struktura HTML dokumenta

HTML KOD	Opis HTML koda
<!DOCTYPE html> *<html>*	Oznaka formirane internet stranice Početak internet stranice.
<head> *</head>*	Predstavlja zaglavlje HTML dokumenta. Uključuje naslov stranice, uređeni stil stranice, te skripte napisane u Javascript jeziku i linkove na vanjske PHP i CSS dokumente.
<body> *</body>*	HTML dokumenta – zaglavlje, podnožje i tijelo stranice, vidljivi sadržaj.
</html>	Kraj internet stranice.

HTML elementi (tablica 5.1.) su dijelovi internet stranice koji kreiraju strukturu za tekst poput zaglavlja, paragrafa, listi, linkova, navoda i drugih dijelova. Može ugraditi kodove pisane u drugim jezicima poput JavaScripta ili PHP-a koji utječu na ponašanje internet stranice jer „čisti“ HTML daje samo „dizajnersku“ formu. Svaki HTML dokument se može kreirati u bilo kojem uređivaču teksta (npr. Notepad ++) ili funkcionalnom alatu (npr. Adobe Dreamweaver).

Uređivanje elemenata na stranici definira strukturu, a izgled stranice radi se u drugom jeziku. CSS (engl. *Cascading Style Sheets*) je stilski obrazac za oblikovanje HTML dokumenta. Postoje tri stilska obrasca koja ujedno predstavljaju prioritete tijekom prikaza stranice napisane u drugim jezicima poput JavaScripte ili PHP-a:

1. vezani (engl. *external*) - stilska obilježja su pohranjena u zasebnoj datoteci *main.css*. Pogodan je ako se stil primjenjuje na više stranica jer je potrebno mijenjati samo jedan dokument. Uključuje se unutar HTML dokumenta naredbom:

   <link rel="stylesheet" type="text/css" href="main.css">

2. ugrađeni (engl. *internal*) - koristi se kada pojedina stranica ima jedinstven stil, a definira se unutar <*style*> oznaka na početku HTML dokumenta kao što je primjer unutar *index.php* stranice.

3. linijski (engl. *inline*) - svakoj pojedinoj HTML oznaci se dodaju vlastita stilska obilježja, ali se time ne mogu iskoristiti sve mogućnosti CSS stila.

CSS pravilo pisanja se dijeli na dva glavna dijela: selektor i deklaracije (slika 5.1.).

Sl. 5.1. Dva glavna dijela CSS elementa.

Pisanjem „čistog“ HTML dokumenta ne dobiva se dovoljna dinamika internet stranice. Stoga je potrebno uključiti JavaScript jezik. JavaScript je skriptni programski jezik koji s AJAX tehnikom (*engl. Asynchronous JavaScript and XML*) omogućuje komunikaciju stranice s poslužiteljskim programom. To je jezik na strani korisnika (engl. *client-side scripting language*) koji radi unutar internet preglednika. Preko njega se definira izvršavanje skripte nakon određenog događaja, promjena sadržaja HTML dokumenta, potvrđivanje podataka, detektiranje preglednika, kreiranje kolačića i još mnogo drugih funkcija prema [9].

Skripta se može unijeti unutar tijela stranice <*body*> ili unutar zaglavlja <head> što je bolje jer se prema [8] funkcija može odmah upotrijebiti nakon otvaranje stranice pisanjem meta oznaka JavaScripta. Osim direktnog pisanja u HTML dokumentu, JavaScript se može uključiti putem vanjskih dokumenata s nastavkom *.js* ili s bilo kojeg mjesta na Internetu:

<script type="text/javascript" src="ime_datoteke ili hiperlink"> funkcija_za_obradu </script>.

JavaScript izjave se izvršavaju tijekom učitavanja stranice. Češće se određeni Javascript kod mora izvršiti kada se dogodi određeni događaj poput klika mišom. U ovom radu korištene su JavaScript funkcije za uključivanje originalnih gumba s komunikacijskih i društvenih mreža poput Facebooka, Skype i Twittera. Koristi se za prikaz rezultata pretraživanja ako je broj rezultata veći od 100, za promjenu slike tijekom prelaska miša ili za pokretanje dvije različite funkcije koje se trebaju obaviti istovremeno nakon klika određenog gumba. Upravo zato je JavaScript postao popularan jezik među internet programerima jer čini internet stranicu zanimljivijom i jednostavnom za korištenje. Postoji veliki broj besplatnih skripti koje poboljšavaju učitavanje stranice, kreiraju različite efekte te poboljšavaju navigaciju internet stranice.

Za uspostavljanje dinamike internet stranice koristi se i programski jezik koji pruža podršku u radu s bazama podataka. PHP (*engl. PHP Hypertext Predprocessor*) je skriptni jezik otvorenog koda koji se najviše koristi za izradu dinamičkih internet stranica. PHP kod se izvršava na poslužitelju, a korisniku se prikazuje samo krajnji rezultat za razliku od HTML dokumenta gdje korisnik može vidjeti pozadinski kod. PHP se pokreće na različitim platformama (Windows, Linux…), kompatibilan je s većinom današnjih poslužitelja te podržava široki raspon baze podataka, besplatan je i jednostavan za učenje. Sve to ga je učinilo široko rasprostranjenim.

Postoji tri osnovna područja gdje se koristi PHP skripta:

- Skripta na strani poslužitelja – najčešće područje za PHP koje koristi PHP parser (CGI (*engl. common gateway interface*) ili modul poslužitelja), internet poslužitelj i pretraživač,
- Komandna linija skriptiranja – zahtijeva samo PHP parser poput skripta pokretanih raspoređivačem zadataka (engl. *Task sheduler)* u Windowsima,
- Desktop aplikacije – nije najbolji odabir, ali se koristi pomoću PHP-GTK.

PHP nema ograničenja, a obavlja većinu dinamičkih radnji na internet stranici. PHP radi s dokumentima na poslužitelju, prikuplja podatke, šalje i prima kolačiće (*engl. cookies*), mijenja i šifrira podatke u bazi. Jedna od najvažnijih značajki PHP-a je podrška za bazu podataka koja postoji na lokalnom ili udaljenom poslužitelju. Lokalni poslužitelj je korisnikovo vlastito računalo. Udaljeni poslužitelj predstavlja bilo koje računalo kojem korisnik pristupa preko internet veze. Nakon provjere korisnika može se pristupiti bilo kojem dokumentu dostupnom na istom poslužitelju.

Povezivanje HTML forme s PHP skriptom je druga važna značajka. U ovom radu svaki HTML dokument dohvaća podatke iz baze podataka pomoću PHP naredbi, a veza s bazom podataka uspostavljena je u datoteci *config.php*. PHP kod se uključuje unutar internet stranice oznakom *<?php* te završava s *?>* između čega se piše PHP izjava. Kada se napravi potvrda preko gumba u HTML formi, unesene informacije postanu automatski dostupne PHP skripti preko, primjerice, POST i GET metode. Time se uspostavlja zahtjev-odziv između klijenta i poslužitelja. Prije nego pretraživač pošalje podatke, šifrira ih shemom URL šifriranja. Ime i vrijednost su spojeni znakom jednakosti, a različiti parovi su odvojeni „*&*“ znakom. Razmak je označen znakom „+“, a posebni znakovi su zamijenjeni ekvivalentnim heksadecimalnim vrijednostima.

GET metoda šalje šifrirane podatke ovisno o zahtjevu stranice. Stranica i šifrirani podaci su odvojeni „?“ znakom:

http://www.e-knjiznica.net84.net/knjiga.php?id=7.

GET metoda stvara duži niz koji se pojavljuje u lokaciji pretraživača ili takozvanoj kutiji (engl. *box*). Ograničena je na 1024 znaka, ne preporučuje slanje osjetljivih informacija i ne može slati slike ili dokumente. Pristup podacima se dobiva varijablom okoline upita (engl. *QUERY_STRING)*. PHP omogućava *$_GET* asocijativnu matricu kao pristup informacijama korištenjem GET metode kao što je naredba *$_GET['id']* iz prethodnog primjera.

POST metoda prenosi podatke kroz HTTP zaglavlje. Informacija je kodirana kako je objašnjeno u GET metodi. POST metoda nema ograničenja na veličinu podataka za slanje, a koristi se za slanje ASCII i binarnih podataka. Sigurnost ipak ovisi o HTTP protokolu, stoga se podaci mogu zaštiti korištenjem HTTPS-a (*engl. HTTP secure*). Za pristup podacima koristi se asocijativna matrica *$_POST['id']*.

Ako se želi napraviti dinamička i funkcionalna stranica koja omogućava korisniku jednostavnije kretanje, potrebno je uključiti PHP sesiju. PHP sesija (engl. *session*) je vrijeme koje korisnik provede na stranici tijekom kojeg se čuvaju svi uneseni podaci. Funkcije sesije pružaju jedinstveni identifikator korisniku tzv. UID (engl. *unique ID*). UID se sprema u kolačiće ili se šalje kroz URL (*engl. Uniform Resource Locator*). Potrebne informacije, povezane na zadani UID, dostupne su kroz sve stranice u jednoj aplikaciji.

Kada korisnik posjeti stranicu na kojoj je omogućena sesija, ili je novi identifikator alociran ili je korisnik povezan s već uspostavljenim identifikatorom iz prošle sesije. Svaka varijabla, povezana sa sesijom, dostupna je kroz $_SESSION superglobalne varijable u kojima se sprema ime i vrijednost varijable prema [6].

Postoje tri osnovne naredbe kojima se pristupa varijablama radi njihove obrade: *session_start(), unset($_SESSION['ime_varijable']) te session_destroy()*.

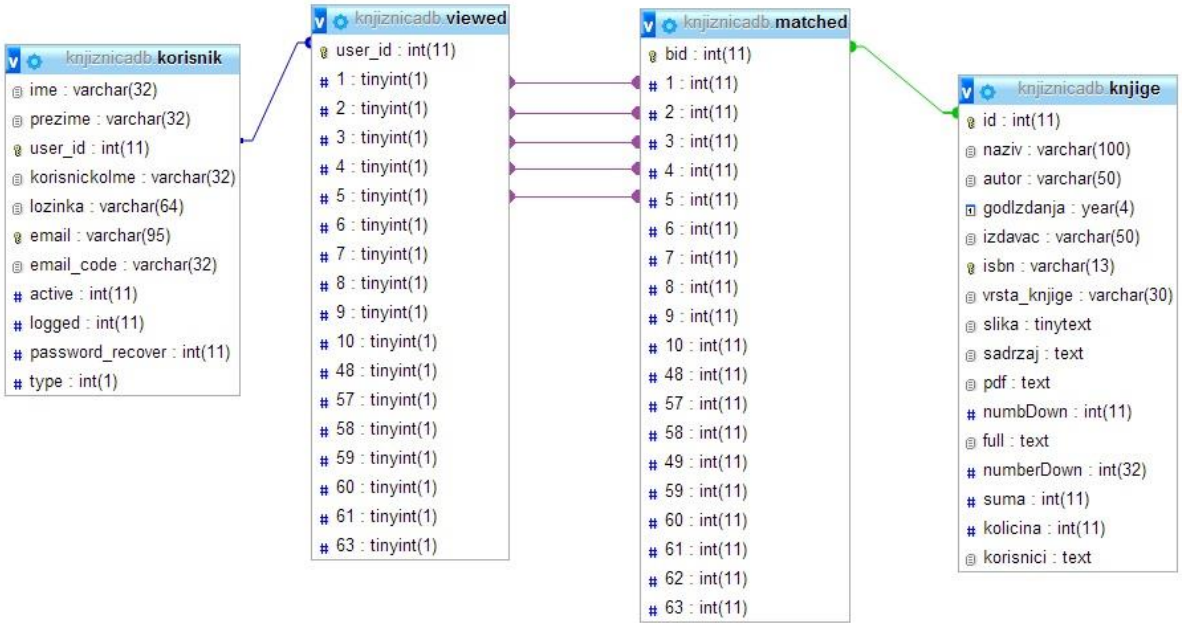
Pristupanje dokumentima, izvršavanje naredbi i stvaranje mrežnih veza na poslužitelju čine ovaj način rada nesigurnim. Mnoge dodatne konfiguracije kontroliraju ovo ponašanje. Potpuna sigurnost nije moguća, ali je potrebno napraviti balans između rizika i korisnosti prema [5]. Jedan od najčešćih napada sigurnosti internet stranice je unos i promjena baze podataka preko malicioznih skripti. Ako se ograniči promjena baze podataka preko internet stranice, onemogućavaju se korisne promjene administratora u bazi podataka. Stoga se zaštita može napraviti ograničenom dozvolom PHP korisnika te provjerom svih unesenih varijabli.

5.2. PROGRAMSKI JEZIK ZA IZRADU I UPRAVLJANJE BAZOM PODATAKA

Sesija je privremeno rješenja za dohvaćanje podataka o korisniku, jer kada korisnik izađe iz internet stranice, podaci se automatski brišu. Prema [7] za trajno čuvanje bolje je koristiti spremanje informacija u MySQL bazu podataka. Baza podataka je strukturirana kolekcija podataka pohranjena na računalnom sustavu i organizirana radi bržeg pretraživanja i dohvaćanja podataka prema [8]. SQL (engl. *Structured Query Language*) je jezik baziran na engleskom jeziku te se koristi u drugim bazama poput Oracle ili Microsoft SQL Servera. Omogućava jednostavne zahtjeve na bazu podataka poput odabira, unošenja, ažuriranja i brisanja.

MySQL je najpoznatiji sustav upravljanja baze podataka preko internet poslužitelja. Besplatno korištenje programa je jedan od razloga velike rasprostranjenosti. Iznimno je snažan te dovoljno brz u izvedbi, vrlo je skalabilan i razvija se s internetom stranicom zbog čega je u uskom krugu s Oracle (jednoj od najtraženijih i najrazvijenijih baza podataka). MySQL baza podataka se sastoji od jedne ili više tablica koje sadržavaju evidencije (engl. *records*) i redove (engl. *rows*). Unutar redaka se nalaze kolumne polja s podacima. Svaka kolumna je ujedno ime polja unutar retka.

Za prikaz veza između tablica koristi se dijagram odnosa entiteta *(engl. entity-relation diagram)* koji prikazuje entitete, njihove atribute i međusobne veze u bazi podataka. Entitet predstavlja pojedinu tablicu, atributi su njihova obilježja dok su veze međusobni odnosi koji povezuju jedan ili više entitet. Tako tablica „*korisnik*“ (slika 5.2.) sadrži slijedeće atribute: ime, prezime, korisničko ime, korisnički ID, email, verifikacijski kod, stanje registracije (0 ili 1), stanje prijave (0 ili 1), kod za unos nove lozinke te tip korisnika (korisnik ili administrator). Tablica „*viewed*“ povezana je s tablicom „*korisnik*“ preko atributa korisničkog ID-a, dok je s tablicom *„matched“* povezana preko atributa istog ID-a knjige.



Sl. 5.2. Dijagram relacije entiteta baze podataka a7065913_kdb.

Kada korisnik pregleda određenu knjigu, stanje knjige se unutar tablice *„viewed“* mijenja s 0 na 1 unutar stupca korisnikovog ID-a. Tablica *„viewed“* prati stanje tablice *„matched“* te se u svakom retku pojedine knjige označava broj korisnika koji su pregledali ostale knjige. Ove dvije tablice kreirane su kao matrice $n \, x \, n$ gdje je n ukupan broj korisnika odnosno ukupan broj knjiga. Kako je riječ o manjem broju knjiga, ovaj način je funkcionalan. Tablica *„knjige“* sadržava podatke o knjizi te je preko ID-a knjige povezana s tablicom *„matched“*. Svaki put kada se kreira nova knjiga, ona se automatski unosi u tablicu *„viewed“* te potom u tablicu *„matched“* što dovodi do zaključka da je tablica *„knjige“* roditeljska tablica tablici *„viewed“* koja ima tablicu dijete „*matched“*.

Podaci iz baze podataka dohvaćaju se MySQL naredbama te se potom obrađuju unutar HTML ili PHP dokumenta. Obrađeni podaci vraćaju se u bazu MySQL naredbama. Ispod se nalaze neke od osnovnih MySQL naredbi korištenih u radu:

ALTER TABLE ime_tablice ADD (ime_kolumne modifikatori, ...); – dodaje kolumnu u postojećoj tablici,
CREATE TABLE ime_tablice (ime_kolumne modifikatori, ...); - kreira tablicu u bazi,
DROP TABLE ime_tablice; - briše tablicu iz baze podataka,
INSERT [INTO] ime_tablice (ime_kolumne, ...) VALUES (vrijednost, ...); - dodaje novi redak u tablici u vrijednostima,

SELECT ime_kolumne, ... FROM ime_tablice WHERE uvjet_pretrage; - traži određeno polje ili redak u tablici.

MySQL baza je funkcionalna, stabilna te ima podršku brojnih programskih jezika poput spomenutog PHP-a, Jave, Perl i drugih. MySQL baza podataka je relacijskog tipa koji se pokazao najboljim oblikom za pohranu i pretragu velikog broja podataka što omogućava brzi pristup informacijama.

5.3. PROGRAMI ZA RAD S BAZOM PODATAKA I INTERNET PRISTUPOM

Većina današnjih internet stranica ima potrebu sačuvati određene podatke bilo kakvog oblika poput korisničkih podataka ili sadržaja internet stranice. Iako se podaci o korisnikovim akcijama čuvaju na korisničkom računalu (u obliku kolačića), podaci koji se prikazuju na internet stranici moraju imati mjesto zapisa. Za to se koristi baza podataka koja sprema potrebne podatke te ih prikazuje prema zahtjevu korisnika ili kodu programera.

Ukoliko se internet stanica kreira lokalno, koristi se internet poslužitelj poput Apache. To je besplatan internet poslužitelj otvorenog koda s programibilnim sučeljem i bazom podataka zasnovanim na DBMS-u koje daju ograničen pristup podacima samo omogućenim korisnicima.

Sustav za upravljanje bazom podataka ili DBMS (engl *database management system*) je niz programa za ulaz, organizaciju i odabir podataka iz baze. Osnovna svrha DBMS-a je softverski sustav koji omogućava definiranje, kreiranje, upite, ažuriranja i administraciju baze podataka, a najpoznatiji su: MySQL, Microsoft SQL i Oracle. Iako portabilnost nije oznaka ovih sustava, korištenjem standarda poput SQL-a, ODBC-a ili JDBC-a omogućava aplikaciji rad s više baza podataka. Različitog su tipa od malih sustava na osobnim računalima do velikih sustava koji se pokreću na središnjem računalu (engl. *mainframe*). Osim knjižnog sustava primjer su još sustav za rezervaciju aviokarata i inventarni sustav. Najveći problem kod velikih baza podataka je upravljanje podacima što uključuje pretraživanje, unos novih i brisanje starih podataka. Zahtjevi za podacima iz baze podataka rade se preko upita (engl. *query*) pisanih u jeziku upita (engl. *query language*). Različiti DBMS-ovi imaju različite jezike upita, ali svi koriste semi-standardizirani jezik SQL.

Poslužitelj baze podataka je pozadinski sustav (engl. *back-end system*) koji koristi arhitekturu klijent-poslužitelj gdje se obrada podataka dijeli na korisnikovu radnu stanicu (engl. *front-end*) i pozadinski sustav (engl. *back-end).*

Standardni pozadinski sustavi uključuju sustave upravljanja bazom podataka (DBMS), sustav slanja poruka (npr. Lotus Notes ili Microsoft Exchange), izlaze prema postojećim sustavima (IBM host) i sustave upravljanja mrežama.

Korisnik je u interakciji s aplikacijama prednjeg sustava kako bi poslao zahtjev na pozadinski sustav koji procesira zahtjev. Zadaća mu je analiza podataka, pohrana, upravljanje podacima, arhiviranje i drugi nekorisnički zadaci. Ponekad se uvodi srednji sustav koji obrađuje procese od klijenta ili poslužitelja, ali najvažnije je da održava poslovnu logiku (pravila, procedure ili operacijske nizove) koju dijele sve aplikacije.

Ako se internet stranica postavlja lokalno bez pristupa Interneta, onda se koriste programi koji emuliraju rad poslužitelja na osobnom računalu. Najčešće korišteni programi su: XAMPP, WAMP, LAMP ili MAMP ovisno o operacijskom sustavu koji se koristi.

XAMPP je skup paketa (engl. s*oftware stack / solution stack*) programa otvorenog koda namijenjen različitim platformama (engl. *cross-platform*) za instalaciju Apache distribucije koji sadržava (engl. *X-cross*) različite platforme, Apache HTTP poslužitelj, MySQL, PHP i Perl. Jednostavan je za instalaciju i besplatan te daje podršku za kreiranje i upravljanje bazom podataka u MySQL-u i SQLite-u između ostalih. Ipak XAMPP se najčešće preporučuje jer radi na svim većim operacijskim sustavima uključujući Windows, MAC i Linux te pruža podršku prema Perlu, Filezilli, Mercury mailu i drugim skriptama.

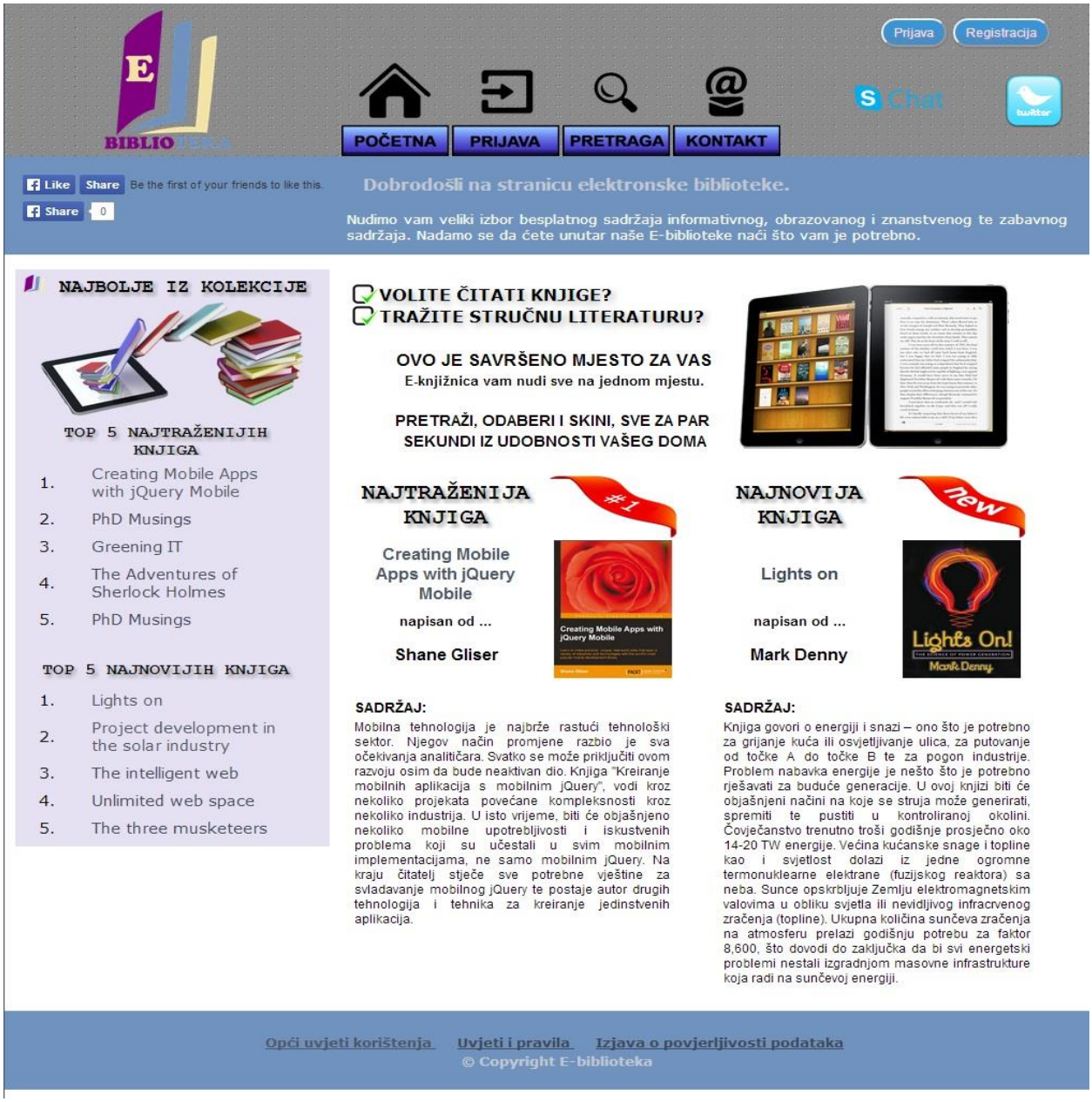
Nakon instalacije, lokalno računalo se može gledati kao udaljeno računalo (engl. *remote host*) spajanjem preko FTP klijenta. FileZilla je besplatan program koji se sastoji od FileZilla klijenta i FileZilla poslužitelja. FTP klijent je platforma podržana od mnogih operacijskih sustava poput Microsoft Windowsa, GNU/Linuxa i Mac OS X-a. Koristi se za spajanje na poslužitelja i ažuriranje promjena na internet stranici nakon što se postavi na internet.

Ako se internet stranica želi smjestiti na stvarnom udaljenom poslužitelju, onda je potrebno odrediti poslužitelj i domenu koja označava novu lokaciju internet stranice. *000webhost.com* je predvodnik u usluzi vodećih poslužitelja jer ne ostavlja bezbroj reklama na internet stranici. Ima dobru brzinu, veliku pouzdanost i odličnu korisničku podršku. Jedna od rijetkih poslužiteljskih tvrtki koje imaju dokazanu garanciju od gotovo neprekidnog 24-osatnog rada. Daje besplatnu internet hosting uslugu, podržava PHP, MySQL, Cron Jobs, Curl, GD i PHP Pear Module, dodatan e-mail račun i upravljačku ploču s dodacima upravljanja stranice.

Kada internet stranica postane dostupna korisniku, potrebno je obaviti testiranje stranice. Svakom korisniku važno je jednostavno snalaženje kroz internet stranicu što omogućava dobra navigacija kroz prethodno isplaniran dizajn stranice. Od svake se internet stranice zahtijeva brzi odziv na korisnikov upit što se mora obaviti brzim učitavanjem stranice, brzim odzivom i brzom pretragom kroz bazu podataka.

Ipak, ovo bi bilo nebitno ako sama internet stranica svojim dizajnom ne privuče korisnika. Internet stranica zato mora imati dobar i jednostavan dizajn, kvalitetan sadržaj te višestruke mogućnosti u kojem korisnik djeluje kao aktivan sudionik što povećava interakciju korisnika te omogućava veću dinamičnost stranice.

6. OPIS APLIKACIJE



Sl. 6.1. Izgled početne stranice internet aplikacije E-biblioteke.

Stranica Elektronske biblioteke, u nastavku E-biblioteka, napravljena je za tri vrste korisnika: korisnika ili takozvanog „gosta“ koji ima pravo pristupa početnoj stranici, pretrazi i pregledu knjiga te kontakt-stranici, prijavljenog korisnika koji može pristupiti svim stranicama i administratora koji ima apsolutni pristup uz dodatno administracijsko sučelje. Link za početnu stranicu E-biblioteke: www.e-knjiznica.net84.net/index.php.

Stranica E-biblioteke podijeljena je u tri dijela prema slici 6.1. Zaglavlje se sastoji od loga biblioteke, četiri gumba za kretanje kroz stranicu te gumba za prijavu i odjavu. Pored toga su i kontakt-oblici za različite društvene i komunikacijske mreže. Tijelo stranice podijeljeno je na središnji dio i rubni stupac. U podnožju početne stranice prikazana su tri dokumenta s pravilima i uvjetima korištenja E-biblioteke.

6.1. BAZA PODATAKA

Za dohvaćanje podataka iz baze, kao što su korisničko ime, lozinka korisnika ili podaci o knjigama, potrebno je prvo napraviti konekcijsku vezu stranice s bazom. Unutar PHP dokumenta *config.php* kreira se konekcija sa slijedećim podacima (tablica 6.1.):

Tab. 6.1. Dio koda za konekcija na bazu podataka, config.php datoteka.

```
<?php
$db_host="mysql3.000webhost.com";
$db_username="a7065913_admin";
$db_password="Jazz9009";
$db_name="a7065913_kdb";
$connect=mysql_connect($db_host,$db_username,"Jazz9009");
?>
```

Potrebno je definirati ime poslužitelja (*engl. host*), korisničko ime, lozinku i ime baze podataka. Potom se radi veza MySQL naredbom *mysql_connect($ime_hosta, $korisničko_ime, $lozinka)* i uključuje se greška ukoliko je veza neuspjela. Ovaj PHP dokument uključuje se u svaki dokument gdje se dohvaćaju podaci, ispisuju ili unose u bazu. Baza podataka "a7065913_kdb" se sastoji od četiri tablice: „knjige“, „korisnik“, „matched“ i „viewed“ kako je prikazano na slici 5.2.

Tablica „knjige“ sadrži slijedeće važne atribute kao što je ID koji ima vrijednost automatskog prirasta (engl. *auto increment*) te je ujedno i primarni (engl. *primary*) ključ. To znači da u tablici može postojati samo jedan atribut s istom vrijednosti ID-a koji je osnovni uvjet pretraživanja knjiga. ISBN knjige veličine je VARCHAR(13) te je ograničen na 13 znakova koliko ISBN sadrži.

Označen je kao jedinstven (engl. *unique*) ključ što znači da samo jedan atribut može imati isti ISBN. To je drugi uvjet po kojem se knjiga može pretraživati. Ostali atributi su isti kao iz sadržaja knjige te pomoćni atributi za izračun broja skidanog materijala ili ocjene knjige.

Tablica „korisnik“ sadržava atribut „user_id“ koji se automatski ispunjava rastućom vrijednosti od prošlog korisnika. Atribut je ujedno i primarni ključ po kojem se razlikuju korisnici. Druga dva važna atributa su „email“ i „korisnickoIme“ koji su jedinstveni ključevi. Tako su označeni kako bi se ograničila višestruka prijava korisnika s istim e-mail računom ili registracija više korisnika s istim korisničkim imenom. Drugi atributi su vezani za aktivaciju računa putem e-maila, zaboravljene lozinke ili korisničkog imena.

Tablice „*matched*“ i „*viewed*“ su međusobno povezane te uspoređuju pregled knjiga jednog korisnika s pregledima drugih korisnika. Tablica „*viewed*“ ima atribute „*user_id*“ iz tablice „*korisnik*“ te „*id*“ iz tablice „knjige“. Za svaku vrijednost „*user_id*“ upisuje se vrijednost "1" ako je korisnik pregledao knjigu ili "0" ako ju nije pregledao.

Kada korisnik *x* otvori određenu knjigu na stranici *knjiga.php?id=y,* u tablici „*viewed*“ se upisuje vrijednost y = "1" na mjestu „*user_id*“ = "x" te se provjerava u tablici „*matched*“ koji korisnici imaju vrijednost "1" na mjestu „*id*“="x". Potom se zbraja koliko je korisnika pregledalo druge knjige. Iz ovoga se radi ponuda korisniku do pet najgledanijih knjiga uz trenutno otvorenu knjigu.

Baza podataka automatski se ažurira prema kodu iz PHP dokumenata. Iako je mogući unos podataka preko stranice http://sql3.000webhost.com/phpMyAdmin, na internet stranici postavljene su forme koje olakšavaju administratoru pristup, pregled i obradu podataka.

6.2. PRIJAVA KORISNIKA

Struktura i razmještaj stranice za prijavu korisnika je identična početnoj stranici. Kada korisnik klikne na gumb „Prijava“ na desnoj gornjoj strani zaglavlja ili na gumb „Prijava“ u središnjem dijelu zaglavlja, automatski mu se otvara stranica za prijavu koja je prikazana na slici 6.2.: www.e-knjiznica.net84.net/prijava.php.

Sl. 6.2. Korisnički panel za prijavu.

S lijeve strane nalazi se korisnički panel (slika 6.2.) gdje se korisnik može prijaviti unosom korisničkog imena i lozinke u naznačena polja te klikom na gumb „Logiraj se“ automatski se prijavljuje ukoliko su uneseni ispravni podaci. Ako korisnik nije prijavljen, klikom na gumb „Registriraj se“, otvara se forma za registraciju novog korisnika na stranici *registracija.php*. Registracija korisnika opisana je u prilogu P.6.1. koji prikazuje greške korisnika tijekom upisa podataka, slanje registracijskog koda na korisnikov račun i uspješnu registraciju korisnika na bazu korisnika. Korisniku je ponuđena dodatna opcija ukoliko je zaboravio svoje korisničko ime ili lozinku. Klikom na poveznicu *korisničko ime* otvara se stranica *recover.php?mode=korisnickoIme*. Korisnik mora unijeti svoju e-mail adresu s kojom je prijavljen te klikom na gumb „Pošalji“ dobiva elektronsku poštu s traženim podacima o korisničkom imenu. Ako korisnik klikne na poveznicu *lozinka* otvara se stranica *recover.php?mode=lozinka*. Budući da je lozinka šifrirana, složeno je, ali i nesigurno korisniku poslati čitljiv oblik stare lozinke. Stoga korisnik na elektronsku poštu dobiva link preko kojeg će promijeniti zaboravljenu lozinku. Nakon unosa korisničkih podataka za prijavu, podaci se provjeravaju u bazi podataka na stranici http://sql3.000webhost.com/phpMyAdmin jer je stranica učitana na njihovom besplatnom poslužitelju odnosno na http://www.000webhost.com kada se stranica prebaci na udaljeni poslužitelj.

Sl. 6.3. Stranica prijave i korisnički panel.

Nakon uspješne prijave korisniku se otvara stranica (slika 6.3.) s korisničkim panelom gdje mu se nude tri opcije za uređenje vlastitog profila. Korisnik može promijeniti svoju lozinku unosom prvo stare lozinke te potom nove koja se provjerava dvostrukim unosom. Korisnik može promijeniti svoje korisničke podatke. Može promijeniti ime, prezime i e-mail adresu koja je označena kao obavezan unos te bez čega se ne može potvrditi promjena.

Korisniku se automatski ispisuju trenutni podaci iz baze podataka, tako da ne mora unositi nove podatke ako su stari podaci ispravni. Na kraju korisniku se nudi opcija trajne odjave iz popisa registriranih korisnika. Korisnik mora potvrditi zahtjev za brisanjem svog korisničkog računa na pop-up prozoru koji se pojavljuje na gornjem dijelu stranice. Ovime se sprječava slučajno brisanje korisnika iz baze podataka i daje mogućnost odustajanja od zahtjeva.

Na kraju korisnik se može ovdje odjaviti sa stranice E-biblioteke kao i preko gumba „Odjavi se“ u desnom gornjem kutu zaglavlja koji se ponavlja kroz sve stranice. Nakon odjave korisnik se automatski prebacuje na početnu stranicu E-biblioteke.

6.3. PRETRAGA KNJIGE

Sl. 6.4. Forma za pretragu knjige i ponuđene knjige s pop-up prozorom, odsječak stranice.

Bilo kojem korisniku, prijavljenom ili neprijavljenom omogućeno je pretraživanje knjiga iz baze podataka na stranici:

www.e-knjiznica.net84.net/pretraga.php.

Pretraživanje baze knjiga podijeljeno je na dvije vrste: osnovno i napredno pretraživanje prema slici 6.4. U osnovnom pretraživanje zadano je pretraživanje ključnih riječi po nazivu knjige, ali korisnik može promijeniti pretraživanje na autora knjige. U kućicu pored „Upiši tekst pretrage“ korisnik upisuje riječi prema koji se pretražuje baza knjiga. Ako korisnik upiše ime autora i označi kružić za pretragu na nazivu knjige, stranica neće naći traženi pojam.

Napredno pretraživanje nije nužno za pretragu pojma. Pod naprednim pretraživanjem korisnik može odabrati vrstu knjige, izdavača, godinu izdanja te ISBN knjige. Pod vrstom knjige korisnik može odabrati tri vrste na koje su knjige podijeljene. Knjige se dijele na računalnu znanost, ostale znanosti i književnost. Pod godinom izdanja korisnik može unijeti samo četiri znaka s maksimalnom vrijednosti do 2013. godine. U kućicu za upis izdavača može se unijeti samo dio ili puni naziv izdavača. Pod ISBN-om korisnik ne mora ništa unijeti ako ne zna podatke.

Korisnik može unijeti oznaku NA (*nepoznat artikl*) što označava novije elektronske knjige koji nemaju ISBN knjige jer nisu namijenjene prodaji. Klikom na gumb „Pretraži“ prolazi se kroz bazu podataka te se prikazuju dobiveni rezultati. Ukoliko je korisnik unio krive podatke, klikom na gumb „Osvježi“ brišu se svi uneseni podaci i korisnik može unijeti nove podatke za pretragu. Kod za pretragu knjiga prikazuje (P.6.2.) preuzimanje unesenih pojmova pretrage, pretragu baze knjiga odnosno tablicu '*knjige*' mysql naredbama i prikaz rezultata na internet stranici.

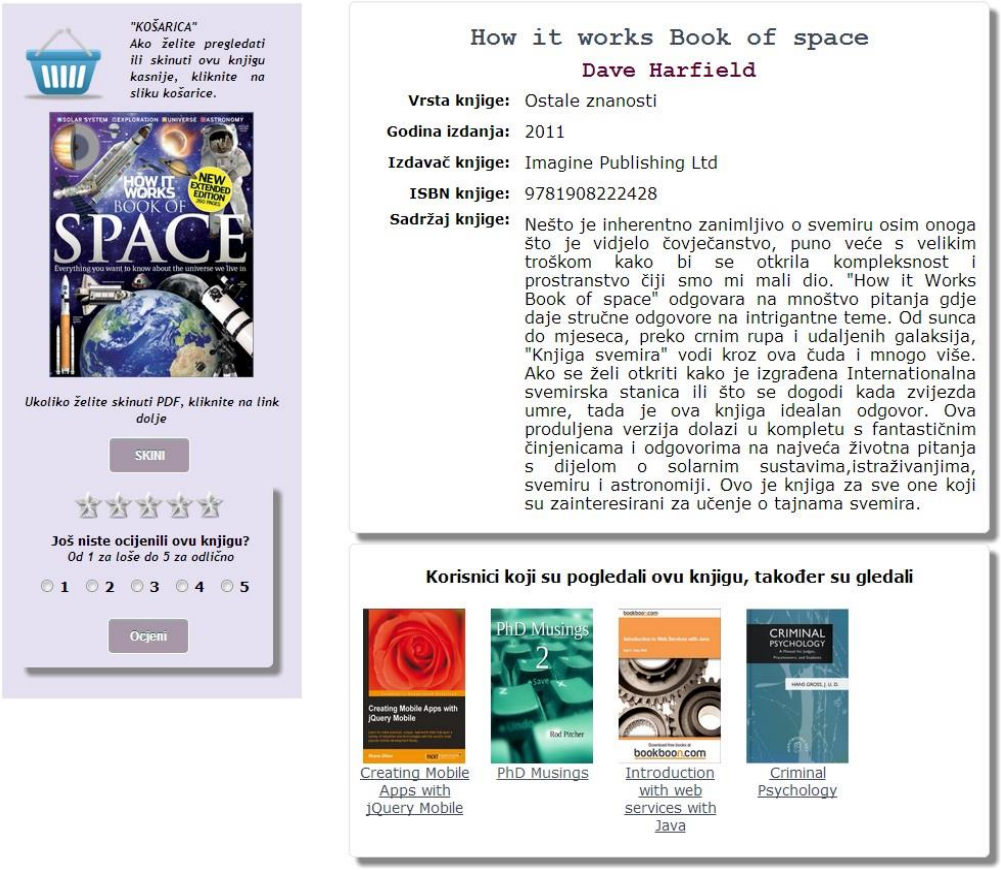
S lijeve strane unutar rubnog stupca nalazi se popis pet najtraženijih i pet najnovijih knjiga. Prema slici 6.4. ako se miš postavi nad naslovom knjige, na stranici će biti izbačen prozor s kratkim opisom knjige. Kratki opis sadržava ime autora, vrstu knjige, izdavača i godinu izdanja sa slikom knjige. Svi naslovi knjiga su aktivni te automatski vode korisnika na pregled cjelokupnog sadržaja odabranog naslova.

Ako se u bazi knjiga ne pronađe traženi rezultat, na stranici će pisati „TRAŽENI POJAM NIJE NAĐEN“, a korisnik se može vratiti na novu pretragu. Ako su rezultati nađeni unutar baze podataka, na stranici će biti ispisani podacima o nazivu knjige, autoru i sadržaj unutar 250 znakova. Obični korisnik može kliknuti na naslov knjige nakon čega odlazi na stranicu *knjiga.php* dok administrator može vidjeti i link na kojem će preuređivati knjigu.

Prema slici 6.5. stranica *knjiga.php* prikazuje opis knjige. U središnjem dijelu prikazana je vrsta knjige, godina izdanja, izdavač knjige te ISBN i sadržaj knjige. U rubnom stupcu s lijeve strane nalazi se slika knjige, prosječna ocjena korisnika i gumb preko kojeg samo registrirani korisnik može skinuti knjigu. Ako trenutno prijavljen korisnik nije ocijenio knjigu, nudi mu se dodatna opcija ocjenjivanja knjige što se potom zbraja u novu prosječnu ocjenu. Klikom na gumb „SKINI“ prijavljenom korisniku se otvara knjiga u PDF dokumentu u novoj kartici ili novom prozoru (ovisno o postavkama preglednika).

Knjiga se može pročitati unutar preglednika ili spremiti na jedno od medija zapisa (računalo, prijenosni disk, USB uređaj i sl.). Iznad slike knjige, prijavljeni korisnik može automatski spremiti knjigu unutar svoje korisničke košarice ako klikne na nju.

Na stranici *prijava.php* nalazi se ista košarica preko koje se otvaraju svi spremljeni podaci. Na stranici: www.e-knjiznica.net84.net/store.php?id=xx se prikazuje slika, ime, autor, godina i izdavač svake knjige koja je spremljena u košaricu. Knjige se mogu zasebno otvoriti, a zapis se može obrisati klikom na crveni gumb ("X") u gornjem desnom kutu. Samo prijavljeni korisnik može ocijeniti knjigu ili ostaviti knjigu u košaricu, dok pretragu knjige može napraviti korisnik i prije prijave na sustav.



Sl. 6.5. Pregled sadržaja knjige i prijedlozi drugih knjiga, odsječak stranice.

Ispod sadržaja knjige korisniku se predlažu dodatne knjige koje bi ga zanimale obzirom na preglede drugih korisnika. U ponudi se prikazuje trenutna knjiga sa svojim indeksom te pet najčešće otvorenih knjiga koje su pregledali drugi korisnici koji su gledali i trenutnu knjigu. Korisnik može otvoriti sadržaj knjige klikom na sliku ili naslov knjige.

6.4. KONTAKT FORMA

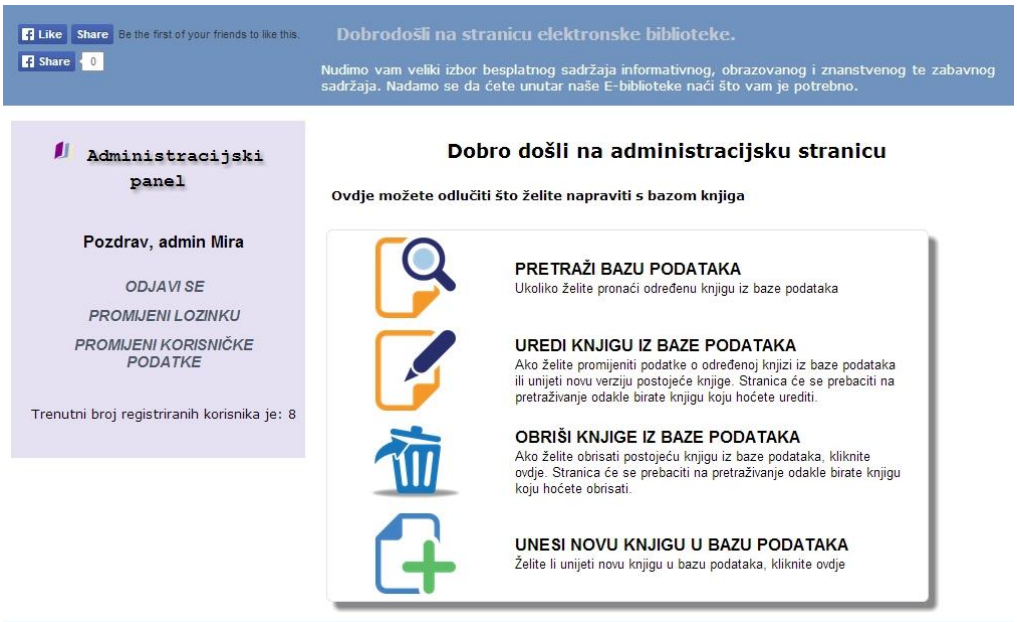
Razlika između korisnika i administratora je u kontakt-stranici koja se prikazuje samo korisniku (www.e-knjiznica.net84.net/contact.php).

Ako korisnik ima pitanja u vezi ponude knjiga, pretraživanja, dodatnih opcija ili ima problema u radu E-biblioteke, može kontaktirati administratora putem zadane forme. U formi je potrebno navesti pitanje koje želi uputiti i e-mail adresu na koju će dobiti odgovor. Osim direktnog slanja e-maila administratora se može dobiti telefonskim putem, upitom na Skype, Twitter i Facebook račun koji su dostupni kroz cijelu internet stranicu unutar zaglavlja.

Nakon što korisnik unese tražene podatke, e-mail se šalje direktno administratoru na e-mail račun: ebiblioteka@e-knjiznica.net84.net, a korisnik dobiva odgovor „Uspješno ste poslali poruku. Odgovor ćete dobiti u što kraćem vremenskom roku. Hvala!“. Time korisnik zna da je upit uspješno poslan.

6.5. ADMINISTRACIJSKO SUČELJE

Stranica (www.e-knjiznica.net84.net/admin.php) omogućava administratoru četiri osnovne opcije za rad s bazom knjiga. Administrator može pretraživati bazu podataka kao i korisnik, može urediti ili obrisati postojeću knjigu te unijeti novu knjigu u bazu podataka kako je prikazano na slici 6.6. Ako administrator želi urediti ili obrisati knjigu, prvo mora pretražiti knjigu u bazi podataka te potom odabrati opciju „Preuredi knjigu: *naziv knjige*“ dok za obični pregled odabire opciju „Pregledaj knjigu: *naziv knjige*“ na stranici *pretraga.php*.



Sl. 6.6. Administracijska stranica i administracijski panel.

Administratoru se otvara sadržaj knjige s mogućnostima promjene. Ukoliko je došlo do krivog upisa ili se traži unos novije verzije, administrator može promijeniti godinu izdanja, izdavača, vrstu knjige te sadržaj knjige, sliku i PDF-dokument. Naziv i autor knjige se ne može mijenjati jer bi značilo da je riječ o različitim knjigama. U tom slučaju, administrator mora obrisati staru knjigu koja mu ne treba odabirom opcije *obriši* te potom unijeti novu knjigu.

U lijevom rubnom stupcu nalazi se opcije za rad s knjigama. Administratoru se uvijek pokazuje trenutno otvorena opcija te dodatne mogućnosti rada. Osim uređivanja administrator može obrisati postojeću knjigu ili unijeti novu. Unos nove knjige se obavlja preko internet aplikacije s formom za unosom knjige. Potrebno je unijeti sve podatke osim slike i PDF dokumenta jer je dovoljno ostaviti podatke za nove knjige koje stižu u ponudi.

7. TESTIRANJE INTERNET APLIKACIJE

DATUM I VRIJEME	Trenutna stranica → Nova stranica
Akcija korisnika → Reakcija sustava	

18.12.2013. 11:36	stranica: *prijava.php*
Upisom krivog korisničkog imena i točne lozinke „Moguća greška prilikom prijave. Korisnik nije nađen. Probajte se registrirati. „	
18.12.2013. 11:39	stranica: *prijava.php*
Upisom ispravnih korisničkih podataka. Korisnik je uspješno prijavljen te se otvara korisnički panel.	
18.12.2013. 11:42	stranica: *prijava.php*
Upisom ispravnih korisničkih podataka. Korisnik je uspješno prijavljen te se otvara korisnički panel	
18.12.2013. 11:44	stranica: *prijava.php → recover.php*
Klik na „Zaboravili ste svoje korisničko ime“ ili „lozinku“. → Korisnik mora unijeti prijavljenu e-mail adresu.	
18.12.2013. 11:46	stranica: *recover.php*
Unos nepostojeće e-mail adrese. Upisana e-mail adresa ne postoji u sustavu.	
18.12.2013. 11:48	stranica: *recover.php → stranica prijava.php*
Unos ispravne e-mail adrese. → Korisnik se automatski prebacuje gdje upisuje podatke dobivene putem e-mail računa.	
18.12.2013. 11:49	stranica: *changepassword.php*
Upisom krivog korisničkog imena i točne lozinke „Moguća greška prilikom prijave. Korisnik nije nađen. Probajte se registrirati. „	
18.12.2013. 11:50	stranica: *changepassword.php*
Unos netočne trenutne lozinke, ispravnih novih lozinki. „Trenutna lozinka je netočna.“	
18.12.2013. 11:52	stranica: *changepassword.php*
Unos netočne trenutne lozinke, ispravnih novih lozinki. „Trenutna lozinka je netočna.“	

18.12.2013. 11:52	stranica: *changepassword.php*
Unos prekratke nove lozinke. „Lozinka mora sadržavati najmanje 6 znakova.“	
18.12.2013. 11:54	stranica: *changepassword.php*
Unos ispravne trenutne lozinke te novih lozinki koje se podudaraju. „Nove lozinke se ne podudaraju“	
18.12.2013. 11:56	stranica: *changepassword.php*
Unos ispravne trenutne lozinke te novih lozinki koje se podudaraju. „Vaša lozinka je promijenjena“	
18.12.2013. 12:12	stranica: *setting.php*
Prazno polje na mjestu imena i e-maila. „Niste unijeli ime“ / „Niste unijeli e-mail“	
18.12.2013. 12:42	stranica: *setting.php*
Prazno polje na mjestu prezimena. „Vaši podaci su ažurirani“ * prezime nije obavezno	
18.12.2013. 11:46	stranica: *setting.php*
Unos neispravne e-mail adrese. „Potrebna je validna e-mail adresa“.	
18.12.2013. 11:48	stranica: *setting.php*
Klik na „ovdje“ ako se korisnik želi iščlaniti. Pop-up prozor: „Jeste sigurni da se želite iščlaniti?“ OK – CANCEL	
18.12.2013. 14:37	stranica: *pretraga.php*
Korisnik ne unosi podatke, klik samo na dugme „Pretraži“. „Ništa nije uneseno. Morate unijeti barem 1 pojam za pretragu“.	
18.12.2013. 14:39	stranica: *pretraga.php*
Korisnik unese samo jedan znak za pretraživanje. Greška na stranici.	
18.12.2013. 14:41	stranica: *pretraga.php*
Korisnik unosi ime postojećeg autora, ali pretražuje prema *naslovu* ili unosi tekst o naslovu, ali pretražuje po autoru. „Traženi pojam nije nađen“.	

18.12.2013. 14:41 stranica: *pretraga.php*
Korisnik unosi ime postojećeg autora, ali pretražuje prema *naslovu* ili unosi tekst o naslovu, ali pretražuje po autoru. „Traženi pojam nije nađen“.
18.12.2013. 14:50 stranica: *pretraga.php*
Korisnik unosi različite kategorije, ukoliko se samo jedna kategorija ne podudara s rezultatom vraća se greška. „Traženi pojam nije nađen“.
18.12.2013. 14:55 stranica: *knjiga.php*
Korisnik želi skinuti knjigu, ali nije prijavljen. Gumb „SKINI“ se prikazuje samo prijavljenim korisnicima.
18.12.2013. 14:57 stranica: *knjiga.php*
Korisnik ocjenjuje knjigu, klik na „Ocijeni“. Korisnik mora osvježiti stranicu radi dobivanja novih informacija.
18.12.2013. 14:58 stranica: *knjiga.php*
Korisnik označava knjigu klikom na košaricu. Korisnik je već unio knjigu u košaricu, sustav unosi knjigu samo jednom.
18.12.2013. 14:59 stranica: *store.php*
Korisnik klikne na gumb **X** za brisanje knjige iz košarice. Sustav automatski briše knjigu iz košarice.
18.12.2013. 15:00 stranica: *contact.php*
Korisnik nije upisao e-mail račun i sadržaj poruke. „Polja označena s zvjezdicom (*) su nužna“.
18.12.2013. 15:04 stranica: *contact.php*
Korisnik unosi podatke za kontakt (e-mail i sadržaj su obavezni, ostalo je izborno). „Uspješno ste poslali poruku. Odgovor ćete dobiti u što kraćem vremenskom roku. Hvala!“
18.12.2013. 15:06 stranica: *input.php*
Administrator unosi novu knjigu, ako se ne upišu polja označena s zvjezdicom (*). „Niste unijeli *ime_polja*…“.
18.12.2013. 15:21 stranica: *input.php*
Unos godine manje od 1000. ili veće od 2013. Pop-up prozor: „Value must be greater or equal then 1000“ / „Value must be less or equal then 2013“

18.12.2013. 15:50	stranica: *input.php*
Unos sadržaja zahtijeva tekst unutar 2500 znakova. Unutar kvadratića korisnik ne može upisivati sadržaj koji sadrži preko 2500 znakova.	
18.12.2013. 15:54	stranica: *edit.php*
Administrator unosi novu knjigu, ako se ne upišu polja označena s zvjezdicom (*). „Niste unijeli *ime_polja*…“	
18.12.2013. 16:20	stranica: *delete.php*
Administrator odabire brisanje knjige gumbom „Obriši“. Knjiga se automatski obriše te se administratoru nudi opciju nove pretrage.	
18.12.2013. 16:28	stranica: *delete.php* → *pretraga.php*
Administrator klikne na gumb „Pretraži drugu knjigu“. → Otvara se stranica *pretraga.php.*	

8. ZAKLJUČAK

U ovom diplomskom radu napravljen je dizajn elektronske biblioteke preko UML dijagrama. Implementacija navedenog dizajna provedena je kroz aplikaciju e-biblioteke kreirane jezicima za izradu internet stranica. Ideja izrade e-biblioteke nastala je od potrebe učenika i studenata za pronalaženjem sve većeg broja potrebnih informacija na digitalnim medijima.

UML dijagrami korišteni su za dizajniranje projekta, a najveći je naglasak bio na klasnom i sekvencijalnom dijagramu. Definirani su najvažniji akteri i slučajevi korištenja prema zahtjevima korisnika te je prikazano međudjelovanje između aktera. Dizajn je implementiran u internet aplikaciju korištenjem HTML, CSS, PHP i Javascript jezika. Aplikacija je prilagođena Google Chrome pregledniku na osobnom računalu gdje je i napravljeno uspješno testiranje.

Najveći dio aplikacije bilo je kreiranje baze podataka i izrada kvalitetnog pretraživanja po zadanim uvjetima pretrage što je smanjilo vrijeme pretraživanja i povećalo kvalitetu rezultata. Registracija korisnika i uređivanje korisničkih podataka samo su neke od dodatnih opcija rada s bazom podataka preko korisničkog sučelja.

Aplikacija ima još prostora za razvoj što se tiče administracijskog sučelja zbog veće mogućnosti upravljanja bazom podataka, dodavanje novih vrsta digitalnih medija poput audio ili video zapisa, bržeg učitavanja internet aplikacije, bržeg pretraživanje baze podataka i veće sigurnosti korisničkih podataka.

LITERATURA

[1] Mirjana Bota: Primjena alata otvorenog pristupa za prepoznavanje znakova, 2011.

[2] Darija Ramljak: Vizualno modeliranje objektno orijentiranih sustava korištenjem uml-a, Zagreb, 2001.

[3] Dan Pilone, Neil Pitman: UML 2.0 in a Nutshell, O'Reilly, lipanj 2005.

[4] Jennifer Niederst Robbins : HTML5 Pocket Reference, 5th Edition, Littlechair, Inc., kolovoz, 2013. godine

[5] Mehdi Achour, Friedhelm Betz, Antony Dovgal i drugi: PHP Manual, PHP Documentation Group, prosinac, 2008

[6] Julie C. Meloni: Sams Teach Yourself PHP, MySQL® and Apache All in One, Pearson Education, Inc., 2012. godina

[7] Tizag.com: www.tizag.com

[8] Robin Nixon, Learning PHP, MySQL, and JavaScript, O'Reilly Media, Inc, srpanj, 2009.

[9] Ivan Vazler, Odjel za matematiku, Osijek: predavanja za predmet: „Web programiranje i primjene"

[10] California state university, odjel za računalne znanosti i inženjerstvo: cse.csusb.edu

SAŽETAK

U ovom radu napravljen je dizajn elektronske biblioteke za skidanje tekstualnog digitalnog sadržaja. Implementacija dizajna napravljena je u obliku internet aplikacije s pripadnom bazom podataka. Korisničko sučelje omogućava registraciju korisnika, promjenu korisničkih podataka te pretragu knjige prema traženim pojmovima. Administracijsko sučelje omogućava ažuriranje baze knjiga dodavanjem novih knjiga ili promjenom postojećih. Za izradu dizajna korišten je StarUML program za crtanje UML dijagrama, a za izradu aplikacije korišteni su programski jezici: HTML, CSS, PHP, MySQL i Javascript. Cilj dizajna i njegove implementacije je pretraga baze podataka i skidanje e-knjiga.

Ključne riječi: elektronska biblioteka, UML dijagram, internet aplikacija, baza podataka, HTML, PHP, skidanje knjige, e-knjiga

ABSTRACT

This master thesis presents a design and implementation of an e-library for purchasing and downloading electronic books and other material in a digital format. Implementation of design is made in the form of web application with the associated database. The user interface provides user registration, change of users data and search for books by the certain terms. Administration interface allows administrator to update the database by adding new books or changing existing ones. StarUML program was used for designing and creating UML diagrams and for creating application the following programming languages were used: HTML, CSS, PHP, MySQL and Javascript. The goal of design and its implementation is search through database and downloads of e-books.

Keywords: electronic library, UML, web application, database, HTML, PHP, download book, e-book

ŽIVOTOPIS

Mirjana Bota rođena je 20.09.1988. godine u Osijeku. Kći Željka i Katice Bota s prebivalištem u Osijeku. Pohađala je osnovnu školu „Tin Ujević“ nakon čega 2003. godine upisuje II. gimnaziju u Osijeku. Srednjoškolsko obrazovanje završava s odličnim uspjehom akademske godine 2006/2007. Iste godine upisuje se na Elektrotehnički fakultet u Osijeku, smjer računarstvo. 2010. godine završava smjer preddiplomskog studija te upisuje prvu godinu diplomskog studija, smjer procesno računarstvo.

Mirjana Bota

PRILOZI

PRILOG 6.1.a (stranica: registracija.php, dio koda)

```
<?php
if(empty($_POST) === false) {
	$required_fields = array('korisnickoIme','lozinka','lozinka_again','ime','email');
	foreach ($_POST as $key=>$value) {
		if(empty($value) && in_array($key,$required_fields) === true) {
			$errors[] = 'Polja označena s zvjezdicom (*) su nužna'; break 1;
	} }
	if(empty($errors) == true) {
		if(user_exists($_POST['korisnickoIme']) === true) {
			$errors[] = 'Korisnicko ime \" .$_POST['korisnickoIme'] . '\' je već zauzeto.';  }
		if(preg_match("/\\s/",$_POST['korisnickoIme']) == true) {
			$errors = 'Vaše korisnicko ime ne smije imati razmak';  }
		if(strlen($_POST['lozinka']) < 6) {
			$errors[]= 'Lozinka mora biti najmanje 6 znakova';  }
		if($_POST['lozinka'] !== $_POST['lozinka_again']) {
			$errors[]= 'Vaše lozinke se ne podudaraju';  }
		if(!filter_var($_POST['email'], FILTER_VALIDATE_EMAIL)) {
			$errros[]= 'Potreban je validna email adresa';  }
		if(email_exists($_POST['email']) == true) {
			$errors[] = 'Email adresa \" . $_POST['email'] . '\' se već koristi.';   }
}   }
if(isset($_GET['success']) && empty($_GET['success'])) {
	echo 'Uspješno ste registrirani. Molimo, provjerite vaš email kako biste aktivirali račun.';
```

PRILOG 6.1.b (stranica: registracija.php, dio koda)

```
        /*$msg="Upravo ste registirani na našoj stranici http://www.e-knjiznica.net84.net <br/> Ukoliko to vi niste zatražili odgovorite na ovaj mail. Nadamo se da će vam se svidjeti naša biblioteka. <br/> Vaša E-biblioteka ";
        $mail=$_POST["email"];
        $subject="Registracija";
        $headers = "From: mbota@gmail.com" ;
        mail($mail,$subject,$msg,$headers);*/
} else {
        if(isset($_POST['agree'])) {
                if  (empty($_POST) === false && empty($errors)  === true){
                        $register_data = array(
                        'korisnickoIme'  => $_POST['korisnickoIme'],
                         'lozinka'        => $_POST['lozinka'],
                         'ime'            => $_POST['ime'] ,
                         'prezime'        => $_POST['prezime'],
                         'email'          => $_POST['email'],
                         'email_code'     => md5($_POST['username']+microtime()) );
                        register_user($register_data);
                        header('Location: registracija.php?success'); exit();
                } else if (empty($errors) === false) {    echo output_errors($errors);
                } else  {    echo output_errors($errors);    }
} }
?>
```

PRILOG 6.2.a (stranica: pretraga.php, dio koda)

```
<?php
if(isset($_POST['submit'])) {
    $text = $_POST['text']; $vrsta = $_POST['vrsta'];
    $godina = $_POST['godina']; $isbn = $_POST['isbn'];
    $izdavac = $_POST['izdavac']; $searchquery = $text; $min_length = 0;
    if (empty($text) && empty($vrsta) && empty($godina) && empty($isbn) && empty($izdavac)) {
        echo "Ništa nije uneseno. Morate unijeti barem 1 pojam za pretragu";
    } else {
    if(strlen($searchquery) >= $min_length){
        $searchquery = htmlspecialchars($searchquery);
        $searchquery = mysql_real_escape_string($searchquery);
        switch($_POST['pojam']) {
            case 'naziv':
            $raw_results1 = mysql_query("SELECT * FROM `knjige` WHERE (`naziv` LIKE '%".$searchquery."%' && `vrsta_knjige` LIKE '%".$vrsta."%' && `godIzdanja` LIKE '%".$godina."%' && `ISBN` LIKE '%".$isbn."%' && `izdavac` LIKE '%".$izdavac."%') "); break;
            case 'autor':
            $raw_results1 = mysql_query("SELECT * FROM knjige WHERE (`autor` LIKE '%".$searchquery."%' && `vrsta_knjige` LIKE '%".$vrsta."%' && `godIzdanja` LIKE '%".$godina."%' && `ISBN` LIKE '%".$isbn."%' && `izdavac` LIKE '%".$izdavac."%')");
            break;
        }  // end of switch
        if(mysql_num_rows($raw_results1) > 0) {
            while($results = mysql_fetch_array($raw_results1)) {
```

PRILOG 6.2.b (stranica: pretraga.php, dio koda)

```
?> <p class="ispisRez">

<img src="slike/vivlia.png" width="25" height="20" align="left">  <?php echo "Pregledaj knjigu:  ";  echo "<a href='knjiga.php?id=".$results['id']."' > ".$results['naziv'] . "</a>" ;

echo "<br />";

echo "<strong>AUTOR: </strong>" . $results['autor'] . "<br />" ;

echo "<strong>SADRŽAJ: </strong>";   /*missing part of cutted content*/

<?php   } } else {   echo "TRAŽENI POJAM NIJE NAĐEN";

          }

}  else  {   echo "Unijeli ste premalo znakova";  }

} } } ?>
```