

Факултет инжењерских наука Универзитета у Крагујевцу



Назив студијског програма: Рачунарска техника и софтверско инжењерство

Ниво студија: Основне академске студије

Предмет: Софтверски инжењеринг 2

Број индекса: 568/2015

Александра Д. Нешић

Разој веб апликације за размену коришћених књига

Дипломски рад

Комисија за преглед и одбрану

1. Др Велибор Исаиловић, доцент – ментор

2. _____

3. _____

Датум одбране: _____

Оцена: _____

Задатак дипломског рада

Платформа треба да омогући размену коришћених књига без новца. Вредност књиге се мери у фиктивном новцу, који се добија “продајом” неке књиге сервису који ће користити ову платформу. Платформа треба да омогући регистравање корисника, чију регистрацију одобрава администратор. Регистрацијом корисник стиче рачун са неким иницијалним износом фиктивног новца. На том рачуну се акумулира фиктивни новац од књига које је корисник уступио сервису за размену. Вредност књиге дефинише корисник, али је верификује администратор. Такође, корисник даје опис књиге из којег се може закључити опште стање књиге (да ли недостају листови, да ли је исписана, подвлачена, итд.). Уколико је књига непродата дуже од неког предвиђеног времена, платформа аутоматски шаље кориснику информацију о томе, са предлогом корекције цене. Платформа треба да омогући:

- претрагу књига по насловима, жанровима, ауторима и сл.
- приказ основних информација о одређеној књизи
- случајни приказ одређеног броја књига на почетном екрану приликом логовања
- директну комуникацију између различитих корисника
- оцену корисника чија је књига купљена

Препоручена литература:

- [1] W. Jason Gilmore, Beginning PHP and MySQL: From Novice to Professional, Fourth Edition, компјутерска литература, САД, 2010.
- [2] Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition, компјутерска литература, САД, 2014.
- [3] W3Schools, <https://www.w3schools.com>, преузето 15.09.2019.
- [4] PHP, <https://www.php.net/>, преузето 15.09.2019.

Крагујевац, октобар 2019.

Ментор:

Др Велибор Исаиловић

Abstract:

This thesis describes the development of a web application for exchanging used books. The application is implemented in PHP scripting language, which creates HTML pages on the server before it, filled with dynamic content, is sent to the client. In this way of generating content, the client cannot see the code (script) that generated the content he is currently viewing, but has access to pure HTML code.

The platform would offer a long-lasting solution to students, students whose families getting textbooks is a huge blow to the home budget, and also to collectors and many other book lovers. The project itself was created in order to preserve the "power" of the book and facilitate the procurement of necessary textbooks and other diverse books. The book is a good read, and without it, there is no education, so this project offers a step forward in realizing modernization related to education itself, that is, offering a faster path to the acquisition of the book.

Key words: web application, scripting language, PHP, HTML, server, client

Резиме:

Овај дипломски рад описује развој веб апликације за размену коришћених књига. Апликација је реализована у скриптном језику РНР помоћу кога се креирају HTML странице на серверу пре него што се она, попуњена динамичким садржајем, пошаље клијенту. Овим начином генерисања садржаја клијент не може видети код (скрипту) који је генерисао садржај који тренутно гледа, већ има приступ чистом HTML коду.

Платформа би понудила дуготрајно решење ученицима, студентима чијим породицама набавка уџбеника представља огроман удар на кућни буџет, а такође и колекционарима и многим другим љубитељима књиге. Сам пројекат створен је у циљу да се очува „моћ“ књиге и олакша набавка потребних уџбеника и других разноврсних књига. Књига је добро штиво, а без ње нема образовања, тако да овај пројекат нуди корак напред у реализовању модернизације везане за само школство, то јест нуди бржи пут ка набавци књиге.

Кључне речи: веб апликација, скриптни језик, РНР, HTML, сервер, клијент

Садржај

1. Увод.....	1
2. Теоријске основе.....	2
2.1 Продаја преко мреже.....	2
2.2 Веб апликација.....	3
3. Радно окружење.....	5
3.1 HTML - Hyper Text Markup Language	5
3.2 CSS - Cascading Style Sheets.....	7
3.3 PHP - Hypertext Preprocessor.....	9
3.3.1 Серверски пакет за развој веб апликација.....	10
3.4 MySQL	11
4. Реализација апликације.....	13
4.1 Навигациони панел.....	13
4.2 Регистрација и пријава корисника.....	17
4.3 Корисничка страна.....	19
4.4 Администратор страна.....	22
4.5 База података.....	24
5. Спецификација корисничких захтева.....	26
5.1 Дијаграм активности.....	26
5.2 Дијаграм случајева коришћења.....	27
6. Закључак.....	29
7. Литература.....	30

1. Увод

Са напредовањем технологија повећава се употреба интернета и он постаје један од главних видова комуникације, информисања, складиштења, обраде и размене података, али и система продаје. Продаја преко мреже (engl. Online) је све заступљенија у свету, а у Америци је постала и најпопуларнији начин куповине. Тиме је потиснула многе традиционалне продавнице. У Србији до пре неколико година није био популаран овај начин продаје, јер је мали број људи имао храбрости да се одлучи за куповину преко интернета. Међутим, из године у годину расте број сајтова за куповину и код нас. Главна предност овог начина куповине је могућност куповине од куће, без одласка до продавница које су километрима удаљене, у којима жељена књига можда тренутно није ни расположива.

Живот без мобилних телефона постао је незамислив, и да данас ретко можемо срести особу која не поседује бар један мобилни уређај или таблет. Због оваквих трендова, веб апликације су се временом прилагођавале уређајима, тако да сада на својим телефонима можемо приступити готово свим веб апликацијама и оне ће бити прилагођене мобилном уређају, таблету или десктоп рачунару, односно њиховим екранима. Систем куповине преко интернета је омогућен свакоме ко поседује бар неки од уређаја преко којег се може повезати на интернет.

Циљ овог дипломског рада јесте развој веб апликације за размену коришћених књига. Под појмом веб апликација подразумева се софтвер чији се код налази на веб серверу. Корисник приступа апликацији помоћу веб претраживача, и интернета. Задатак веб апликације је:

- да препозна шта корисник тражи од ње,
- да по потреби препозна корисника,
- да провери да ли корисник има право приступа апликацији, њеном делу или подацима,
- и на крају, да кориснику испоручи тражене податке, документ или информацију у складу са овлашћењима корисника.

У дипломском раду је описан развој апликације за размену коришћених књига, при чему се вредност књиге мери у фиктивном новцу. Омогућена је регистрација корисника при којој корисник стиче рачун са иницијалним износом фиктивног новца (1000дин). Корисник дефинише назив књиге, аутора, жанр, цену, као и опште стање књиге (да ли недостају листови, да ли је исписана, подвлачена, итд.). Платформа је развијена тако да омогућава претрагу књига по насловима, ауторима, ИСБН броју, затим приказ основних информација о књизи, директну комуникацију између корисника, као и оцену корисника чија је књига купљена.

Апликација је писана у **Атому** [1] који служи за уређивање текста и изворног кода. Програмски језици који су коришћени за израду ове апликације јесу **PHP**, **HTML** и **CSS**.

2. Теоријске основе

2.1 Продаја преко мреже

Половином 80-их година се први пут спомиње појам интернет куповине и интернет пословања. Први предмет који је продат преко интернета био је компакт-диск (engl. CD) на коме се налазио музички албум Стинга „Ten Summoner's Tales“, који је продала група студената са Swarthmore колеџа из Америке. Након тога настао је први веб сајт за онлајн продају ког је направио Pjer Omidar 1996, „AuctionWeb “. Он је на свој сајт поставио покварени ласер који је желео да прода. Чувени Пјеров ласер тада је постигао цену од 14.83\$ и први онлајн сајт за продају је заживео. Пјер је свој сајт преименовао у „Echo Bay“ познатији као „eBay“ и временом постао милионер, популаризацијом и развојем интернет продаје. Годинама се трговина преко интернета усавршавала, да би данас постала један од водећих начина трговине[2].

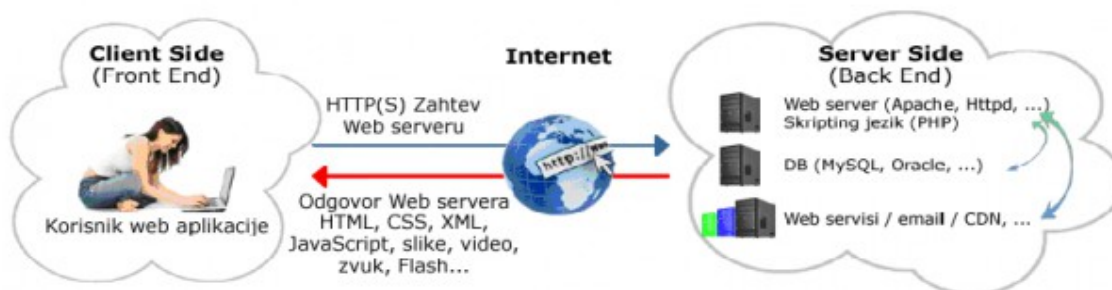
Куповина путем интернета еволуира из дана у дан. Све више фирми жели да прошири своју продају и све су веће потребе за програмерима веб сајтова.

Постоје различите врсте онлајн продаје, а међу најпознатијим су:

- B2C (Business-to-Consumer), вид продаје подразумева да компаније продају робу крајњим потрошачима преко софтвера за продају аутоматски без потребе за додатном интеракцијом између људи. Најпознатији сајт који функционише по овом принципу је Амазон.
- B2B (Business-to-Business) продаја се односи на продају између компанија, када примера ради, велике компаније продају робу продавцима на мало.
- C2C (Consumer-to-Consumer) је врста куповине где купац продаје купцу. Најверодостојнији примера оваквог начина онлајн продаје је eBay.

2.2 Веб апликација

Веб апликација треба да препозна шта корисник жели и да му испоручи одговарајући резултат у складу са задатим критеријумима и овлашћењима за приступ (слика 1) [3].



Слика 1. Веб апликација

Основни принцип рада веб апликације се састоји од следећих корака:

1. Када корисник у веб претраживачу унесе жељену УРЛ адресу (нпр. www.google.com), његов компјутер шаље HTTP захтев преко интернета до веб сервера на коме се налазе HTTP сервер, скриптинг језик и сама веб апликација.
2. Веб апликација анализира упућени захтев и покушава да "схвати" шта корисник жели од ње: којој страници се приступа, да ли се у оквиру захтева налазе параметри које треба препознати, да ли је можда корисник послао неке податке или фајлове преко веб обрасца?
3. Веб апликација је "разумела" шта се од ње тражи. Сада треба да препозна корисника: ко је послао захтев, као и да ли корисник то сме да тражи?
4. Уколико корисник има одговарајућа права за започету акцију, веб апликација може да повлачи и/или да обрађује податке, комуницира са другим веб серверима или веб сервисима, припрема редирекцију на други веб сервер.
5. Резултат рада веб апликације може бити HTML веб страница, CSS, JavaScript, или неки други документ који веб сервер шаље преко интернета до компјутера корисника.

Предност веб апликација је њихова независност од врсте компјутера, оперативног система или платформе. Исту веб апликацију је могуће користити на десктоп компјутеру, таблет рачунару или паметном телефону, без обзира на оперативни систем који је инсталиран на тим уређајима (Windows, Linux, Mac итд). Могу лако да размењују податке са другим апликацијама и, по потреби, да се интегришу са различитим веб сервисима. Веб апликација се инсталира, ажурира и одржава на веб серверу, без потребе да се дистрибуира и/или инсталира на рачунарима крајњих корисника.

Ово је велика предност у компанијама и системима са великим бројем корисника.

Веб апликације имају очигледне предности. Обичном кориснику рачунара у свакодневном раду је свакако једноставније да користи веб апликације: ништа не треба да се инсталира, нове верзије софтвера су аутоматски доступне, цена месечног закупа је само део цене десктоп апликације.

Тим који је развио софтвер може да прати статистичке податке везане за рад апликације у реалном времену. То је велика предност, јер омогућава увид у интеракцију корисника са вашим производом.

И можда најјачи аргумент - новац. Веб апликација функционише на различитим платформама и оперативним системима.

Недостатак веб апликација је Интернет веза. Уколико дође до прекида Интернет везе, веб апликација постаје делимично или у потпуности неупотребљива. Власник апликације може да, барем теоријски, прати све што корисници раде, тако да је изражен и проблем приватности и заштите пословних података.

3. Радно окружење

3.1 HTML - *Hyper Text Markup Language*

HTML није програмски језик, већ језик ознака (енг. Markup - Ознака). За израду HTML странице је довољан обичан текст едитор. HTML користи ознаке (тагове) да опише различите елементе интернет странице. HTML елементи дају структуру HTML документу и говоре веб претраживачу шта ће веб страница садржати [4].

HTML тагови су ознаке за почетак и крај неког елемента. HTML тагови почињу знаком “<” (мање од), а завршавају се знаком “>” (веће од).

Постоје две врсте тагова:

- отварајући тагови, нпр. <html>
- затварајући тагови са косом цртом, нпр. </html>

HTML синтакса прати следећа правила:

- HTML елементи и њихова својства се пишу малим словима,
- Вредности својстава HTML елемената се наводе унутар двоструких наводника,
- HTML елементи треба да буду затворени,
- За парни HTML елемент се пишу и почетни и завршни таг,
- Празни елементи морају бити правилно затворени косом цртом.

Приказ најчешће коришћених тагова и њихов опис дат је у табели 1.

Табела 1. *Списак најчешће коришћених тагова*

Основни тагови	Опис
<i>html</i>	Креира HTML документ.
<i>head</i>	Део у којем се налази наслов странице и остале информације које нису видљиве на самој страници.
<i>body</i>	Главни део странице који је видљив кориснику.
<i>h1, h2, h3, h4, h5, h6</i>	Ознаке за дефинисање наслова, од највећег ка најмањем.
<i>p</i>	Параграф текста.
<i>div</i>	Дели страницу на различите секције/блокове.
<i>br</i>	Прави ред размака у текстуалном блоку.
<i>hr</i>	Прави хоризонталну линију за раздвајање делова садржаја.

Head тагови	Опис
<i>title</i>	Дефинише наслов странице.
<i>meta</i>	Садржи основне информације о страници.
<i>link</i>	Повезује страницу са другим документима.
<i>script</i>	Садржи скрипте које се извршавају на клијентској страни.
<i>style</i>	Садржи стилове који описују изглед HTML елемента.

Атрибути	Опис
<i>id</i>	Јединствени идентификатор елемента.
<i>class</i>	Додељује елементу једну или више подкласа.
<i>title</i>	Дефинише назив елемента.
<i>alt</i>	За дефинисање алтернативног текста.
<i>href</i>	Дефинише са којим документом је линк повезан.
<i>src</i>	Одређује извор слике тј. путању до ње.
<i>style</i>	Дефинише уметнуте стилове елемента.

3.2 CSS - *Cascading Style Sheets*

CSS је језик форматирања помоћу ког се дефинише изглед елемената веб сајта, фонтова, боја размака између параграфа, уређивање табела. Првобитно, HTML је служио да дефинише комплетан изглед, структуру и садржај веб сајта, али је од верзије 4.0 HTML-а уведен CSS који би дефинисао конкретан изглед, док је HTML остао у функцији дефинисања структуре и садржаја [5].

Коришћењем CSS-а постало је могуће одвојити презентацију и податке на презентацији, од самог дизајна. У овом случају HTML код постаје прегледнији и мањи, што значи да га је пуно лакше контролисати, а такође је могуће једноставном променом параметара у стилу променити изглед странице без измене HTML кода.

CSS синтакса се састоји од описа изгледа елемената у документу. Опис може да дефинише изглед више елемената. Такође, више описа може да дефинише изглед једног елемента.

Сваки CSS опис се састоји из три елемента (слика 2):

```
ciljni_element
{
    svojstvo: vrednost;
    svojstvo: vrednost;
    ...
}
```

Слика 2. CSS синтакса

Циљни елементи се дефинишу на три начина:

1. наводећи HTML таг циљних елемената,
2. наводећи класу елемената,
3. наводећи идентификациону вредност (ID) елемента.

Својства се описују низом кључних речи дефинисаних у W3C [6] стандарду, које се могу сврстати у следеће категорије:

- позадина елемента,
- ивица,
- оквир,
- приказ,
- димензије,
- фонт,

- маргине,
- унутрашњи празан простор,
- позиција,
- изглед припадајућег текста.

CSS код се може унети у HTML документ (заглавље или тело странице), на следеће начине:

- као **style** атрибут HTML елемента:

```
<div style="margin: 20px; padding: 5px 10px"></div>
```

- као засебни фајл који се позива помоћу **<link>** тага:

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

- као интерни CSS блок уоквирен **<style>** **</style>** таговима:

```
<style type="text/css">
  table, td, th
  {
    border: 3px solid blue;
  }
</style>
```

3.3 PHP - Hypertext Preprocessor

PHP је скрипт језик направљен за генерисање динамичких веб страница. Синтакса је позапљена из C-а, Java и Perl-а са пар јединствених PHP специфичних додатака. Циљ језика је да омогући веб програмерима да брзо пишу динамички генерисане странице [7].

PHP се уопштено извршава на веб серверу, узимајући PHP код као улаз а креирајући веб странице као излаз. PHP се може користити на већини веб сервера и на готово сваком оперативном систему и платформи без икакве новчане надокнаде.

PHP је скриптни језик помоћу кога се могу креирати HTML странице на серверу пре него што се она, попуњена динамичким садржајем, пошаље клијенту. Овим начином генерисања садржаја клијент не може видети код (скрипту) који је генерисао садржај који тренутно гледа, већ има приступ чистом HTML коду.

Историјски развој PHP-а је почео када је 1995. године Rasmus Lerdorfa објавио PHP/FI. То је био скуп алата (енгл. Personal Home Page Tools) написаних у програмском језику Perl које је аутор користио за личне потребе. Расмус је цео пројекат превео на C и омогућио комуникацију са базом података. Напокон, цео код је објављен 1997. године као бесплатни open source како би постао свима доступан. Тиме су даљи развој и тестирање постали ефикаснији и бржи.

PHP 3.0 су направили Endi Gutmans и Zeev Suraski 1997. године и то је била прва верзија која је добила форму у којој данас познајемо PHP.

PHP 4.0 је настао 1998. године, док је PHP 5.0 објављен 2004. године. У верзији 5, PHP је добио пуну подршку за објектно оријентисано програмирање.

PHP код се налази на веб серверу. Крајњи корисник не може да види PHP код, већ само HTML документе које PHP испоручује.

PHP генерише HTML код тако што:

- Препозна шта корисник тражи,
- Пронађе потребне податке из базе података или документа,
- У HTML код интегрише динамички садржај,
- Испоручи кориснику готову HTML страницу.

PHP код почиње тагом “<?php” и завршава се тагом “?”(слика 3).

```
<?php
echo "Dobar dan! ";
?>
```

Слика 3. PHP синтакса

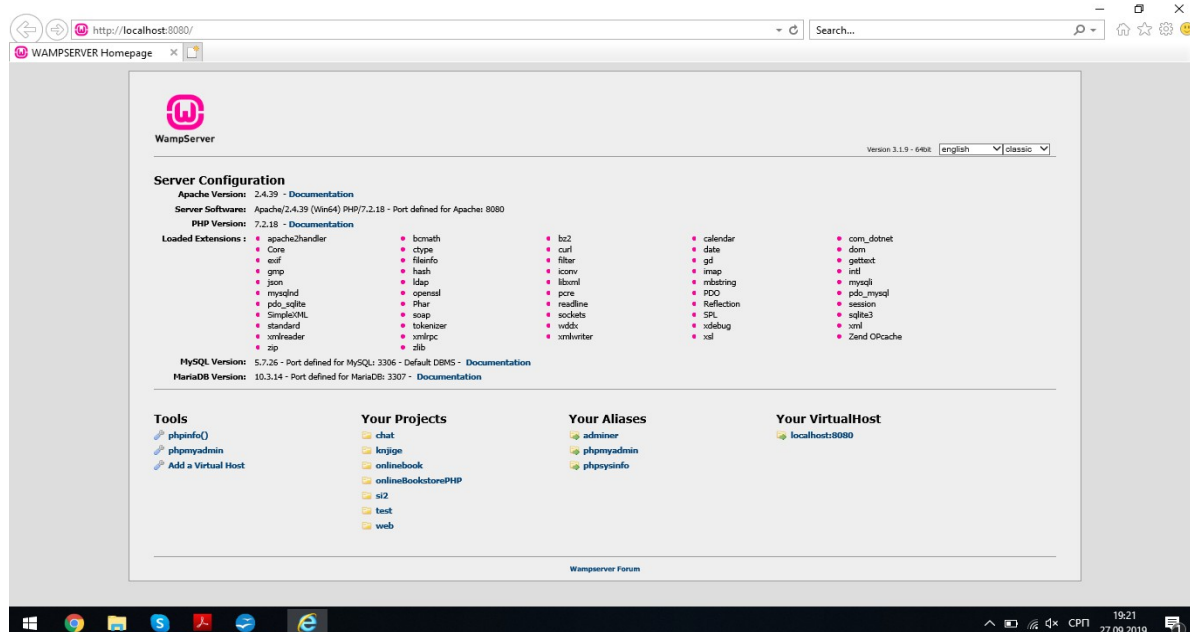
Осетљивост на велика и мала слова (енгл. case sensitive):

- Функције, класе, кључне речи (else, if, while ...) - имена нису осетљива на велика и мала слова,
- Променљиве - имена су осетљива на велика и мала слова,
- Константе - осетљивост на велика и мала слова се одређује приликом дефинисања константе.

Велика предност, али уједно и велики проблем РНР-а је да нема ограничења за улазак у овај свет: бесплатан је, цена хостинга је ниска, поседује изванредну пратећу документацију, на располагању је велики број готових, такође бесплатних решења.

3.3.1 Серверски пакет за развој веб апликација

Пакет под именом **WAMP** [8] (слика 4) и њему слични представљају независно креиране програме који се инсталирају на компјутере који поседују microsoft-ов Windows оперативни систем. WAMP представља акроним састављен од почетних слова оперативног система коме је намењен и компоненти које садржи. Компоненте једног таквог система су Apache сервер, MySQL сервер и PHP, Perl или Python. Apache представља веб сервер, MySQL представља програм за манипулацију базом података док PHP представља скриптни језик који може да манипулише подацима који се налазе у базама података и да генерише динамичке веб странице сваки пут кад дође захтев од корисника. Остали програми могу бити прикључени оваквом пакету као што је phpMyAdmin који омогућава веб програмерима да помоћу дефинисаног интерфејса комуницирају са базом података.



Слика 4. Изглед wamp развојног окружења

3.4 MySQL

База података је скуп међусобно повезаних информација, које су смештене у споредној меморији рачунара које су истовремено доступне разним корисницима и апликацијама. Системи за управљање базама података називају се **Data Base Management Systems – DBMS** [9]. Ови системи физички врше приказивање базе података у складу са траженом логичком структуром и обављају све операције над овим информацијама. Подаци у бази су логички организовани у складу с неким моделом података.

MySQL је SQL систем за управљање релационим базама података. Подржава стандардан SQL, и ради на веб серверу (backend). Погодан је и за мале и за велике веб апликације. Брз је, поуздан и лак за употребу. Бесплатан је за преузимање и коришћење. MySQL је власништво **Oracle** [10] корпорације.

SQL је програмски језик за рад са скуповима информација и односима између њих. SQL се користи за опис скупова података који могу да помогну у добијању одговора на питања. Када се користи SQL, мора се користити исправна синтакса. Синтакса је скуп правила по којима се елементи језика исправно комбинују. SQL синтакса је заснована на синтакси енглеског језика.

Пре него што PHP приступи подацима у бази података, мора да успостави везу са MySQL сервером:

```
$db = new mysqli($host, $korisnicko_ime, $lozinka, $baza);
```

где су:

`$db` – идентификатор успостављене везе

`$host` – име хоста или IP адреса

`$korisnicko_ime` – корисничко име

`$lozinka` - лозинка

`$baza` – име базе података

MySQL упити се у PHP програмском језику извршавају на следећи начин:

```
$rezultat = $db->query($upit);
```

где су:

`$upit` - SQL упит (SELECT, INSERT, UPDATE, DELETE, ...)

`$db` - идентификатор успостављене везе.

При читању резултата MySQL упита, резултат може да се прочита као:

- нумерички низ:

```
$rez = mysqli_fetch_row($rezultat);
```

- или као асоцијативна матрица:

```
$rez = mysqli_fetch_assoc($rezultat);
```

Основне операције SQL упита јесу убацивање података у табелу (INSERT), читање података (SELECT), измена података (UPDATE) и брисање података из табеле (DELETE) (слика 5).

```
INSERT INTO ime_tabele VALUES (v1, v2, v3, ...);  
  
SELECT lista_kolona FROM ime_tabele WHERE uslov;  
  
UPDATE ime_tabele SET kolona=v1, ... WHERE uslov;  
  
DELETE FROM ime_tabele WHERE uslov;
```

Слика 5. SQL упити – основне операције

Затварање везе са MySQL базом података у PHP програмском језику се врши на следећи начин:

```
mysqli_close($db);
```

SQL упитни језик препознаје различите типове података:

- Карактере: CHAR (фиксне дужине), VARCHAR (променљиве дужине), TEXT
- Бројеве: BIGINT, INT, SMALLINT, TINYINT, REAL, FLOAT, MONEY
- Време и датуми: DATE, DATETIME

4. Реализација апликације

4.1 Навигациони панел

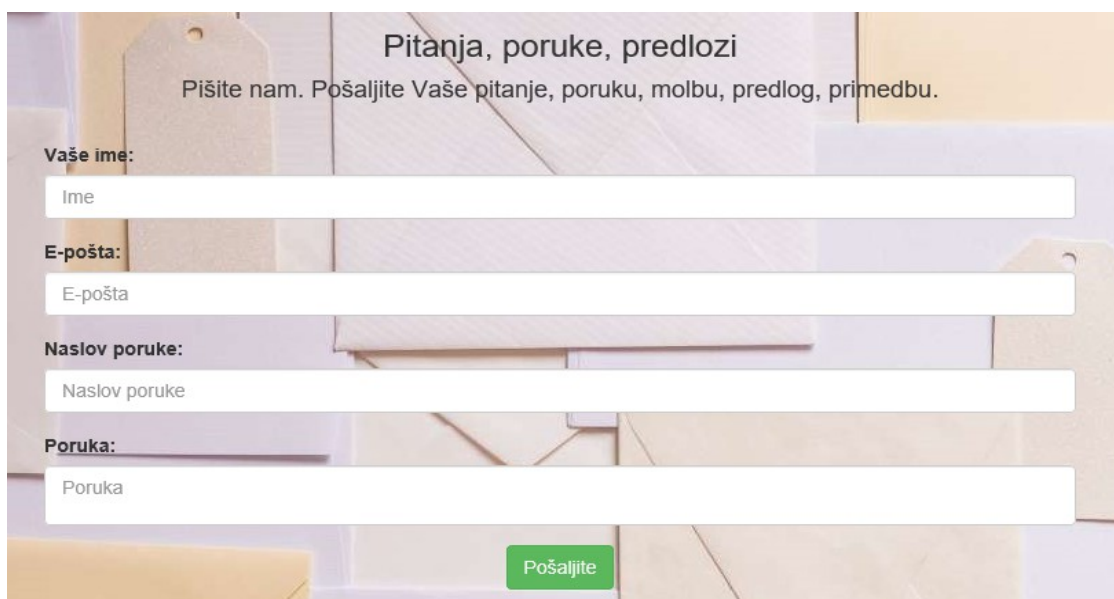
Почетна страна ове веб апликације је *index.php*. Приступом овој страни, корисник улази на сајт за размену коришћених књига. У заглављу ове стране се налази навигациони панел. Навигациони панел врши навигацију ка следећим странама: Почетна, О нама, Контакт, Оцените наше кориснике, Регистрација и Пријава. Ознака **<nav>** дефинише тај скуп навигацијских веза (слика 6). С обзиром да је потребно омогућити да корисник може да иде са једне стране на другу, овај навигациони панел је укључен на свим странама преко наредбе: **include("includes/header1.php");** . Такође, све стране садрже и подножје које се укључује преко следеће наредбе: **include("includes/footer.php");** . За дефинисање подножја докумената или одељака користе се тагови **<footer>** и **</footer>**. Они обично садрже информације о аутрским правима, контакт информације, итд.

```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target="#myNavbar">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="#">Prodaja knjiga</a>
    </div>
    <div class="collapse navbar-collapse" id="myNavbar">
      <ul class="nav navbar-nav">
        <li class="active"><a href="index.php">Početna</a></li>
        <li><a href="about.php">O nama</a></li>
        <li class="active"><a href="kontakt1.php">Kontakt</a></li>
        <li><a href="ocena.php">Ocenite naše korisnike</a></li> -->
      </ul>
      <ul class="nav navbar-nav navbar-right">
        <li><a href="register.php"><span class="glyphicon
glyphicon-user"></span> Registracija</a></li>
        <li><a href="login.php"><span class="glyphicon glyphicon-
log-in"></span> Prijava</a></li>
      </ul>
    </div>
  </nav>
```

Слика 6. Реализација навигацијског панела

Фајл *about.php* представља страну "О нама". На овој страни се налазе опште информације о сајту. Ознака `<div>` сама по себи не представља ништа, па се зато користи за груписање садржаја. Овај елемент се обично користи као конетјнер (engl. container) за проток садржаја. Нема утицај на садржај или изглед док не буде дизајниран CSS-ом користећи class или id атрибуте. Управо на овој страни, опште информације о сајту су стављене између почетног и завршног `<div>` тага.

Уколико постоје питања, примедбе, жалбе или похвале, корисницима је омогућено да се обрате слањем мејла сервису, у овом случају продавници за размену књига. Корисници шаљу своју поруку на мејл продавнице попуњавајући образац са својим подацима (слика 7).



The image shows a contact form with the title "Pitanja, poruke, predlozi". Below the title is a subtitle: "Pišite nam. Pošaljite Vaše pitanje, poruku, molbu, predlog, primedbu." The form contains four input fields: "Vaše ime:" (Your name) with placeholder "Ime", "E-pošta:" (Email) with placeholder "E-pošta", "Naslov poruke:" (Message subject) with placeholder "Naslov poruke", and "Poruka:" (Message) with placeholder "Poruka". At the bottom right of the form is a green button labeled "Pošaljite".

Слика 7. Контакт страна

Постоји много PHP библиотека за слање мејлова, а једна од њих је **PHPMailer** [11]. Генерално, PHPMailer је алтернатива **mail()** функцији која је најчешће коришћена. Међутим, постоји више сценарија када **mail()** функционалност није довољно флексибилна.

Када се креира PHP скрипта (*kontakt1.php*) која ће слати е-пошту посредством SMTP сервера (дефинисаног SMTP хостом, портом, (корисничким) имејл налогом и лозинком), биће потребно урадити извесна подешавања (слика 8). Једно од најважнијих јесте да се наведе да SMTP захтева аутентификацију и унос постојећег имејл налога и његове лозинке. У питању је исти процес као када правимо налог на Gmail-у и шаљемо пошту преко тог сервиса. SMTP је у основи протокол који шаље захтев за е-поштом на мејл сервер и након верификације шаље ту пошту даље на удаљени сервер. Битно је напоменути да уколико се на Gmail-у користи двострука аутентификација, пошту није могуће слати.

Највећа предност коришћења удаљеног SMTP-а у PHPMailer-у је та што уколико је подешен на неко друго име домена, које није локални домен, биће означен као СПАМ на примаочевом серверу.

Приликом рада је направљен и Gmail који је везан ову апликацију. Мејл адреса, која се и наводи у коду, је *prodavnicaknjiga8@gmail.com*. На ову адресу стиже пошта, сва питања, примедбе, похвале и жалбе од стране корисника.

```
require 'class/class.phpmailer.php';
$mail = new PHPMailer;
$mail->IsSMTP(); //Postavljanje SMTP-a
$mail->Host = "smtp.gmail.com"; //Postavljanje hosta
$mail->Port = '465'; //default SMTP server port
$mail->SMTPAuth = true; //SMTP autentikacija. Username i Password
$mail->Username = 'prodavnicaknjiga8@gmail.com'; //SMTP username
$mail->Password = 'prodavnica1234'; //SMTP password
$mail->SMTPSecure = 'ssl'; //Konekcioni prefiks.Opcije su
" ", "ssl" or "tls"
$mail->From = $_POST["email"]; //From email adresa
$mail->FromName = $_POST["name"]; // From name
$mail->AddAddress('prodavnicaknjiga8@gmail.com', 'Email');
$mail->AddCC($_POST["email"], $_POST["name"]);
$mail->WordWrap = 50;
$mail->IsHTML(true); //Tip poruke HTML
$mail->Subject = $_POST["subject"]; //Naslov poruke
$mail->Body = $_POST["message"]; //Telo poruke
```

Слика 8. Подешавање PHPMailer-а посредством SMTP-а

У заглављу веб апликације се налази дугме "Оцените наше кориснике". Кликом на дугме отвара се страница на којој се налазе сви корисници апликације. Сваки корисник има и позитивне и негативне оцене. Приликом оцењивања потребно је изабрати једну од опција: Like или Dislike. Свим корисницима је омогућено да виде оцене које имају остали корисници.

Преко SELECT упита из базе су извучени сви корисници чији је статус одобрен од стране администратора (слика 9). Приказ корисника је у виду засебних картица за сваког корисника на којима се може видети име корисника, е-пошта, позитивне и негативне оцене. Онако како су подаци дефинисани у бази, тако се и наводе у коду, да би се ти подаци извукли директно из базе односно табеле којој припадају. У овом случају, из табеле "users" извлаче се username, email, grade_plus и grade_minus (слика 10). Кликом на дугме "Like" поље са позитивним оценама се ажурира при чему се број оцене повећава за један при сваком клику. Такође, кликом на дугме "Dislike" број негативних оцена се повећава за један при сваком клику и ажурира се у табели.

```
$sql = "SELECT * FROM users WHERE role='user' AND status=1";
$result = mysqli_query($db, $sql);
if ($success != 1) {
    $success = 0;
}
```

Слика 9. Упит за приказ свих корисника из базе

```
<div class="card-body">
  <h4 class="card-title text-center">Korisnik: <?php echo
$row['username']; ?></h4>
  <p class="card-text">E-pošta: <?php echo $row['email']; ?></p>
</div>
<ul class="list-group list-group-flush">
  <li class="list-group-item">Pozitivne ocene: <?php echo
$row['grade_plus']; ?></li>
  <li class="list-group-item">Negativne ocene: <?php echo
$row['grade_minus']; ?></li>
</ul>
```

Слика 10. Извлачење одређених података из табеле "users"

4.2 Регистрација и пријава корисника

Платформа омогућава регистрацију корисника, чију регистрацију одобрава администратор. У горњем десном углу апликације се налази дугме за регистрацију корисника. Кликом на дугме се отвара страна која омогућава регистрацију корисника. Потребни подаци за регистровање корисника су: корисничко име, адреса, е-пошта и лозинка. Ова поља су обавезна и морају се попунити како би регистрација била успешна (слика 11). Уношењем одговарајућих података, у табели за кориснике уносе се подаци, а администратор у пољу “status” мења вредност на "1" и тиме одобрава приступ кориснику да користи апликацију. Регистрацијом корисник стиче рачун са иницијалним износом фиктивног новца у вредности од 1000 динара.

Registracija

Korisničko ime:

Adresa:

E-pošta:

Lozinka:

Ponovite lozinku:

[Registrujte se](#)

Već ste naš član? [Prijavite se](#)

Registracija

Korisničko ime je obavezno.
E-pošta je obavezna.
Lozinka je obavezna.

Korisničko ime:

Adresa:

E-pošta:

Lozinka:

Ponovite lozinku:

[Registrujte se](#)

Слика 11. Форма за регистрацију корисника

Кликом на дугме “Registruјte se” извршава се неколико активности у позадини. Уколико нису унети подаци у обавезна поља, корисник бива обавештен да мора унети све податке. Затим, извршава се провера да ли је корисник унео већ постојеће корисничко име или е-пошту. У том случају, регистрацију није могуће извршити (слика 12). Такође, систем ће обавестити корисника уколико је унео неважећу адресу е-поште како би је исправио и омогућио да се регистрација обави успешно. Након ових провера, подаци се уносе у табелу о корисницима и регистрација је успешно извршена.

```
$user_check_query = "SELECT * FROM users WHERE
username='$username' OR email='$email' LIMIT 1";
$result = mysqli_query($db, $user_check_query);
$user = mysqli_fetch_assoc($result);

if ($user) { // if user exists
    if ($user['username'] === $username) {
        array_push($errors, "Korisničko ime već postoji.");
    }
    if ($user['email'] === $email) {
        array_push($errors, "E-pošta već postoji.");
    }
}
```

Слика 12. Провера да ли постоји корисник са истим подацима у бази

Корисник након регистрације може да се пријави уколико му је налог одобрен од стране администратора, уносећи своје корисничко име и лозинку. Уколико налог није одобрен, корисник неће моћи да се пријави (слика 13). Кликом на дугме “Пријавите се” испитује се да ли је корисник унео све податке у поља која су обавезна да би се пријавио. Такође, уколико је унео погрешну лозинку или корисничко име, биће обавештен да исправи како би се успешно пријавио.

Слика 13. Форма за пријаву корисника

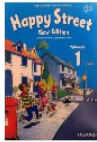
Уколико је корисник сервиса у питању, приликом пријављивања ће приступити страници за куповину књига. Уколико се администратор пријави преко исте форме, биће прослеђен на страну за администратора.

Било да је у питању корисник или администратор, омогућено је одјављивање кликом на дугме “logout”, након чега се корисници поново враћају на страну за пријављивање.

4.3 Корисничка страна

Приликом пријаве корисник приступа почетној страни. Ово је страна на којој се врши куповина књига. Омогућен је преглед свих књига које су доступне за продају, као и приказ основних информација о одређеној књизи (слика 14). Основне информације су: назив књиге, име аутора, жанр, цена, стање књиге и ИСБН број. Приликом пријављивања врши се случајан приказ одређеног броја књига на почетном екрану. То је одрађено убацивањем команде **ORDER BY rand()** [12] у упит за приказ књига .

Књиге и њихове основне информације се извлаче из табеле “books” из базе.

 Elektronika za neupocene Gordon McComb, Earl Boysen	 Happy street Stella Maidment, Lorena Roberts	 Hajduci Branislav Nusic
Žanr: kompjuteri i internet	Žanr: udzbenik	Žanr: knjiga za decu
Cena: 960.00	Cena: 627.20	Cena: 100.00
Stanje: podvlacena	Stanje: koriscena, ispisana	Stanje: fale stranice(15-28)
ISBN: 852-2535363124-24	ISBN: 773-73859685-43	ISBN: 978-86-7346-
Vaša E-pošta: 	Vaša E-pošta: 	Vaša E-pošta:
Kupi	Kupi	Kupi

Слика 14. Приказ књига и основних информација о њима

Поред тога што је дат опис стања књига, корисник може да убади и слику која то може и да докаже. Испод информација о књизи, налази се поље у коме корисник уноси своју е-адресу. Преко мејла бива обавештен о успешној куповини и када ће му књига бити достављена.

Да би кориснику била олакшана куповина, уведена је и претрага књига по наслову, имену аутора или ИСБН броју (слика 15). Када корисник упише податке по којима жели да претражи одређену књигу, кликом на дугме “Тражи” имаће увид у то да ли се та књига уопште налази у бази за продају.

```
if (!empty($author) && !empty($name) && !empty($ISBN)) {  
    $orSql = " OR ";  
}  
$sqlSearch .= "AND ( " . $nameSql . $orSql . $authorSql . $orSql  
    . $isbnSql . " )";
```

Слика 15. Претрага по наслову, аутору и ИСБН броју

Када се корисник одлучи да купи књигу, ту операцију извршава кликом на дугме “Купи” (слика 16). Приликом те активности на основу идентификационог броја књиге, у бази се књига обележава као продата. У случају да корисник нема довољно новца на рачуну, продаја се неће извршити и корисник ће бити обавештен да нема довољно новца за ову операцију. У супротном, прочита се цена књиге из базе, и одузме од новца који корисник има на рачуну. Корисник бива обавештен преко мејла, да је купио књигу и да ће му бити достављена у року од 48 часова.

```
if (isset($_POST['buy_book'])) {  
    $id = mysqli_real_escape_string($db, $_POST['id']);  
    $price = mysqli_real_escape_string($db, $_POST['price']);  
  
    $user = getUserById($_SESSION["user_id"], $db);  
  
    if($user['money'] < $price) {  
        array_push($errors, "Nemate dovoljno novca da kupite  
knjigu.");  
    }  
    if (count($errors) == 0) {  
        $user_id = $user["id"];  
        $query = "UPDATE users SET money = money - '$price' WHERE  
id='$user_id' ";  
        $queryBook = "UPDATE books SET sold = 1 WHERE id='$id'";  
  
        if (mysqli_query($db, $query)) {  
            mysqli_query($db, $queryBook);  
            $success = 1;  
        }  
    }  
}
```

Слика 16. Активности извршене након куповине књиге

Сваки корисник може да објави књигу коју жели да прода. Додавање књиге на листу свих књига за продају се обавља кликом на дугме на навигационом панелу, “Dodaj knjigu”. На овој страни, налази се форма коју треба попунити. Форма садржи наслов књиге, назив аутора, жанр, опис књиге из кога се може закључити опште стање књиге (да ли недостају листови, да ли је исписана, подвучена, итд.), цена и ИСБН број књиге. Корисник уноси све потребне податке и кликом на дугме “Dodaj”, књига се додаје у базу, односно табелу са књигама. У табели се аутоматски уноси датум када је књига објављена за продају, као и који корисник ју је објавио (слика 17).

```
$date = date("Y-m-d");
$user_id = $_SESSION["user_id"];
$sql = "INSERT INTO books (name,
authors,genre,description,price,ISBN,sold,date,user_id) ";
$sql .= " VALUES ('$name',
'$authors','$genre','$description','$price',
$ISBN,0,'$date','$user_id')";
if (mysqli_query($db, $sql)) {
    header('location: knjige.php');
} else {
    echo "Error: " . $sql . "<br>" . mysqli_error($db);
}
```

Слика 17. Додавање књиге у базу

4.4 Администратор страна

Администратор страна је слична корисничкој страни. Приликом пријављивања администратора, омогућен му је увид у све књиге које се налазе у бази података. Пријава администратора се извршава уношењем корисничког имена “admin” и лозинке “adminadmin”. Након пријаве, отвара се страна за администратора. Вредност поља “role” у бази у овом случају је “admin”. Он, такође, може да врши претрагу књига по називу књиге, по аутору и ИСБН броју. Тиме му је олакшано да изврши неку активност за одређену књигу.

Администратор има могућност да обави неколико активности. Једна од активности јесте брисање књига из базе. Кликом на дугме “Obriši”, покренут је упит који брише књиге, са одређеним идентификационим бројем, из табеле “books” (слика 18).

```
if (isset($_POST['delete_book'])) {
    $id = mysqli_real_escape_string($db, $_POST['id']);
    if (count($errors) == 0) {
        $query = "DELETE FROM books WHERE id='$id' ";
        if (mysqli_query($db, $query)) {
            $success = 1;
        }
    }
}
```

Слика 18. Брисање књиге

Уколико књига није продата дуже од неког предвиђеног времена, потребно је кориговати цену (слика 19). Дугме за корекцију цене омогућава администратору да провери када је потребно одрадити корекцију за одређену књигу. Уколико је прошло више од тридесет дана од дана када је књига објављена, онда је потребно одрадити промену цене како би се књига лакше продала. Корекција цене се извршава тако што се цена, након месец дана, смањује за 20 процената. Уколико није прошло више од тридесет дана, књига не захтева промену цене и администратор и корисник бивају обавештени о томе.

Администратор прати стање података у бази. Он има улогу да одобри регистрацију корисника, променом “status” поља на вредност 1. Има увид у листу корисника којима је одобрена регистрација, које је стање на рачуну корисника, као и коју оцену има.

Такође, има увид који корисник је објавио коју књигу, ког датума, као и да ли је књига продата или није.

```
if($dani < (-31)){  
    if (count($errors) == 0) {  
        $query = "UPDATE books SET price = price - (price * 0.2)  
WHERE id='$id' ";  
        if (mysqli_query($db, $query)) {  
            $success1 = 1;  
        }  
    }  
}
```

Слика 19. *Корекција цене за 20%*

4.5 База података

Веома је важно да правилно дизајнирамо базу података пре него што почнемо да је креирамо. У супротном, скоро сигурно ћемо морати да се враћамо и мењамо редове или колоне табела које креирамо. Добра пракса је нацртати табеле и атрибуте, и тако осмислити базу пре него што је креирамо и сместимо податке у њу.

Процес коришћења MySQL-а са PHP-ом:

1. Успостављање везе са MySQL-ом и означавање тренутно активне базе података **mysqli_connect**;
2. Извршавање упита и добијање скупа резултата **mysqli_query**
3. Учитавање реда резултата из скупа резултата **mysqli_fetch_row**
4. Обрада добијених вредности за учитани ред
5. Затварање конекције са базом података **mysqli_close**

Успостављање везе са базом је извршено на следећи начин:

```
$db = mysqli_connect('localhost', 'root', '', 'database');
```

где су параметри:

- localhost – хост,
- root – корисничко име,
- '' - лозинка,
- database – име базе података.

Погодно је да се параметри конекције са базом података чувају у посебној датотеци (connection.php). Овај фајл позивамо у свакој скрипти у којој је потребно вући податке из базе, тако што правилно наведемо путању до ње, **include("includes/connection.php")**; .

Флексибилност подешавања тих параметара на једном, централном месту олакшава:

- тестирање система са резервном или удаљеном копијом података, једноставним мењањем имена базе података и/или сервера у датотеци,
- тестирање система са различитим комбинацијама корисничких имена и лозинки, са различитим овлашћењима.

При употреби MySQL функција за рад с базама података постоји могућност да се позивање неке од функција заврши неуспехом. Узроци неуспеха могу бити различити: задат је

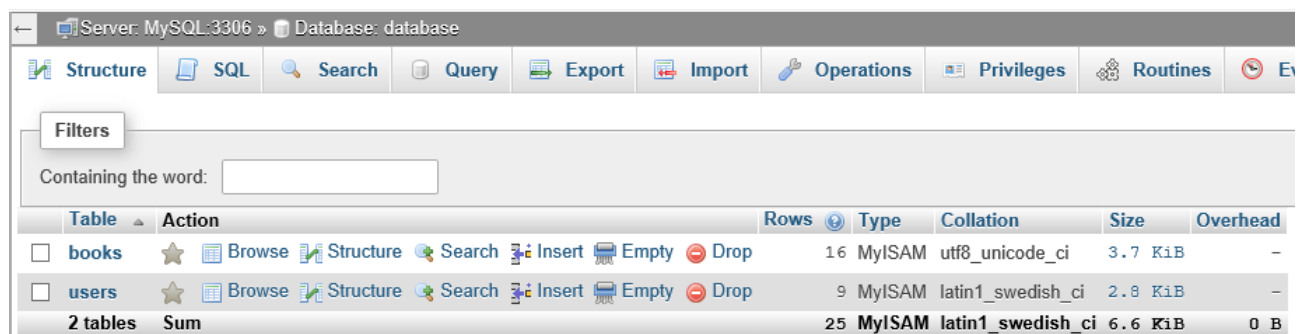
погрешан параметар, корисник је унео погрешну лозинку, упит је погрешно састављен, итд. У том случају PHP ће у већини случајева приказати неку своју грешку (најчешће на нивоу упозорења (engl. Warning)) и наставити са извршавањем скрипта.

База “*database*” садржи две табеле, табелу за кориснике “*users*” и табелу за књиге “*books*” (слика 20). Атрибути табеле о корисницима су: идентификациони број, корисничко име, адреса, е-пошта, лозинка, новац на рачуну, позитивна оцена, негативна оцена, статус и улога.

Лозинка корисника је криптована, тако да се чува приватност, само корисник зна њену вредност. Фиктивни новац на рачуну је 1000 динара, а сваком куповином или продајом се та вредност мења. Статус корисника чија је регистрација одобрена од стране администратора је “1”, уколико није, вредност је “0”. Разликујемо две улоге у систему, корисник и админ.

Атрибути табеле са књигама су: идентификациони број књиге, назив књиге, слика, аутор, цена, жанр, опис стања, ИСБН број, датум, да ли је продата књига или не, и идентификациони број корисника који ју је додао.

Датум књиге се уписује онда када је књига додата у базу. Та вредност нам омогућава да знамо када је потребно извршити корекцију цене књиге. Вредност “sold” може бити нула или један. Уколико је нула значи да књига није продата, а у супротном да јесте. Поље “user_id”, повлачи идентификациону вредност корисника из табеле о корисницима. На основу ње можемо знати који корисник је објавио коју књигу.



The screenshot shows the MySQL Database Structure view for a database named 'database'. It displays two tables: 'books' and 'users'. The 'books' table has 16 rows, MyISAM engine, utf8_unicode_ci collation, and a size of 3.7 KiB. The 'users' table has 9 rows, MyISAM engine, latin1_swedish_ci collation, and a size of 2.8 KiB. The summary row shows 25 rows total for both tables, with a total size of 6.6 KiB.

Table	Action	Rows	Type	Collation	Size	Overhead
books	Browse Structure Search Insert Empty Drop	16	MyISAM	utf8_unicode_ci	3.7 KiB	-
users	Browse Structure Search Insert Empty Drop	9	MyISAM	latin1_swedish_ci	2.8 KiB	-
2 tables	Sum	25	MyISAM	latin1_swedish_ci	6.6 KiB	0 B

Слика 20. База података Database

5. Спецификација корисничких захтева

Ово поглавље има за циљ да опише случајеве коришћења система уз приложене дијаграме. Захтеви корисника се описују преко модела случајева коришћења. Дијаграм активности служи да детаљно прикаже ток контроле кључних функција система.

УМЛ (engl. *Unified Modeling Language*)[13] је стандардни графички језик за моделовање објектно – орјентисаног софтвера.

- **Unified** – уједињује све постојеће нотације
- **Modeling** – користи се за моделовање софтверских елемената
- **Language** – средство комуникације

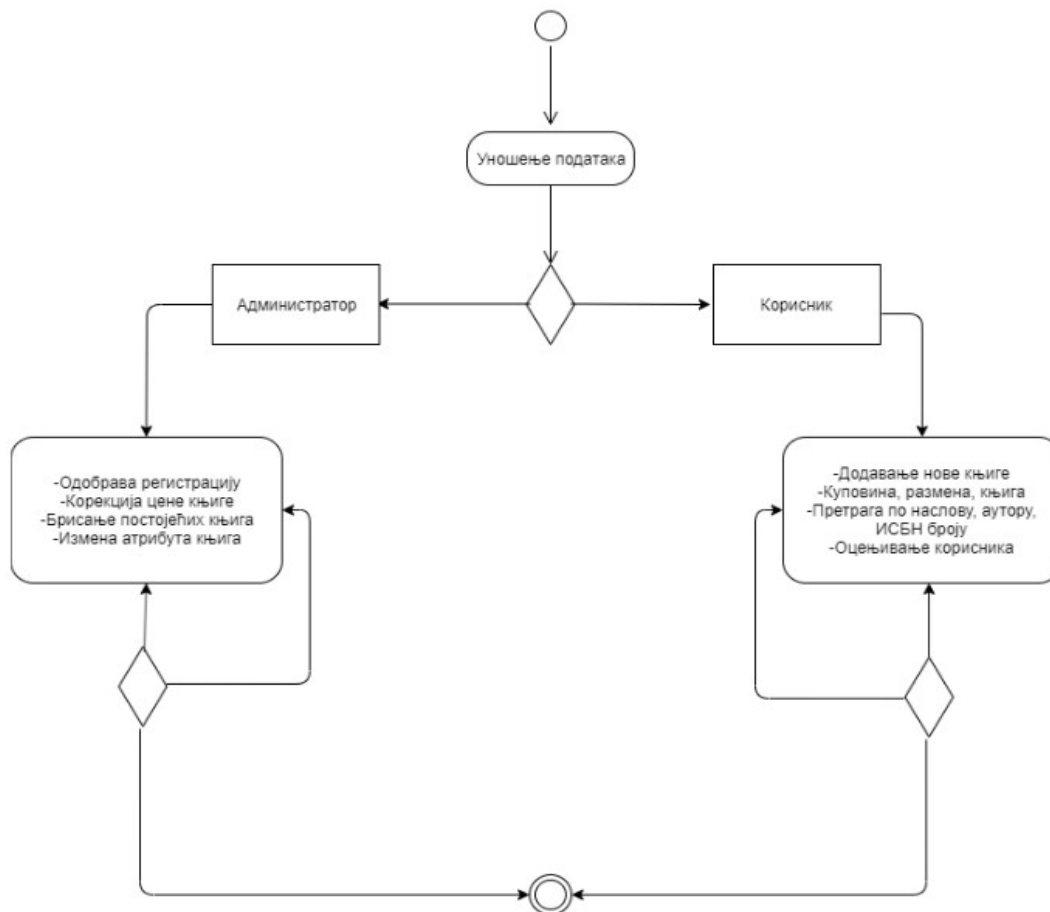
Основни елементи:

- **Елементи** – битни концепти (класе, објекти, поруке)
- **Релације** – повезивање индивидуалних ствари
- **Дијаграми** – груписање међусобно повезаних колекција ствари и релација

5.1 Дијаграм активности

Дијаграм активности (слика 21) представља ток у систему из једне активности у другу. Користи се за репрезентовање динамичког понашања система. Њиме се представљају пословни процеси вишег нивоа, токови података и сложени логички елементи система.

Активности су веће целине у систему које се могу разложити на више акција, а акције представљају операције у систему.



Слика 21. Дијаграм активности

5.2 Дијаграм случајева коришћења

Дијаграм случајева коришћења (engl. *Use case diagram*) је графички приказ садржаја скупа случајева коришћења (слике 22, 23). Указује на границе система и његову интеракцију са спољним светом. Омогућавају крајњим корисницима да разумеју систем, односно, поглед корисника на функционисање система (шта систем ради, а не како систем функционише).

Актери (учесници) су корисници система који изводе случајеве коришћења. Корисник је човек који користи систем, док је учесник специфична улога коју корисник има у комуникацији са системом. Случај коришћења дефинише функционалност система са становишта учесника, односно, шаблон понашања делова система.



Слика 22. Дијаграм случајева коришћења администратора



Слика 23. Дијаграм случајева коришћења корисника

6. Закључак

У раду је приказан процес развоја веб апликације за размену коришћених књига. Уз детаљну анализу захтева, користећи се модерним алатима, направљен је програм који се може користити у мањим продавницама за продају књига у свакодневном обављању послова. Програм даје разне могућности управљања подацима у зависности од статуса корисника (администратор, корисник). Постоји могућност унапређивања апликације додавањем наруџбенице, плаћањем преко банке, омогућавањем комуникације између корисника преко чета (engl. Chat), итд.

Такође, у проблематици развоја апликација у оквиру PHP програмског језика, свакако је значајно изучити доступне развојне оквире (engl. Framework) као што су LARAVEL, ZEND, Simfony итд.

Платформа нуди решење ученицима и студентима чијим породицама набавка уџбеника није занемарљива ставка у кућном буџету, а такође и колекционарима и многим другим љубитељима књиге. Пројекат је осмишљен са циљем да се очува „моћ“ књиге и олакша набавка потребних уџбеника и разних других књига. Књига је и данас неопходно средство у образовању, упркос огромном утицају интернета и електронских садржаја. Овај пројекат нуди решење за брже проналажење старијих издања књига, које је готово немогуће пронаћи у комерцијалним књижарама.

7. Литература

- [1] Атом. Преузето: Август 10, 2019, од <https://atom.io/>
- [2] eBay. Преузето: Септембар 29, 2019, од <https://www.ebayinc.com/company/our-history/>
- [3] Веб апликација. Преузето: Септембар 29, 2019, од <http://sr.magma.rs/blog/web-aplikacije/sta-je-to-web-aplikacija>
- [4] Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition, компијутерска литература, САД, 2014.
- [5] Robin Nixon, Learning PHP, MySQL, JavaScript, CSS & HTML5, Third Edition, компијутерска литература, САД, 2014.
- [6] W3C. Преузето: Септембар 29, 2019, од <https://www.w3.org/standards/>
- [7] W. Jason Gilmore, Beginning PHP and MySQL: From Novice to Professional, Fourth Edition, компијутерска литература, САД, 2010.
- [8] WAMP. Преузето: Август 10, 2019, од <http://www.wampserver.com/en/>
- [9] W. Jason Gilmore, Beginning PHP and MySQL: From Novice to Professional, Fourth Edition, компијутерска литература, САД, 2010.
- [10] Oracle. Преузето: Септембар 29, 2019, од <https://www.oracle.com/index.html>
- [11] PHPMailer. Преузето: Септембар 15, 2019, од <https://github.com/PHPMailer/PHPMailer>
- [12] Синтакса SQL. Преузето: Септембар 20, 2019, од <https://www.w3schools.com/>
- [13] УМЛ. Преузето: Септембар 29, 2019, од <http://moodle.mfkg.rs/>