

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ

**АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА БИБЛИОТЕЧНОГО
ФОНДА ПОКРОВСКОЙ БИБЛИОТЕКИ ФИЛИАЛ №2**

Выпускная квалификационная работа
обучающегося по направлению подготовки
02.03.03 Математическое обеспечение и администрирование
информационных систем
очной формы обучения,
группы 07001402
Сероусова Сергея Владимировича

Научный руководитель:
к.т.н., доцент Румбешт В.В.

БЕЛГОРОД 2018

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	3
ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	6
1.1 Особенности организации	6
1.2 Обзор и анализ существующих систем	8
1.3 Требования к автоматизированной системе	16
ГЛАВА 2 ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ	20
2.1. Применяемые средства разработки	20
2.1 Проектирование и разработка базы данных	22
2.3. Проектирование структуры и разработка веб-приложения	28
ГЛАВА 3 ТЕСТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ	36
3.1 Программа и методика испытаний	36
3.2 Испытание информационной системы	38
ЗАКЛЮЧЕНИЕ	53
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	54
ПРИЛОЖЕНИЕ	55

ВВЕДЕНИЕ

Основной чертой человеческой цивилизации является производство, потребление и накопление информации во всех деятельности человека. Вся жизнь человек получает, обрабатывает и хранит информацию в разном ее виде. Какой бы деятельностью он не занимался он постоянно и непрерывно получает и обрабатывает информацию.

Рост объемов информации, перерабатываемой в процессе человеческой деятельности, сопровождается постоянным перераспределением трудовых ресурсов из сферы материального производства в информационную сферу. Таким образом, наряду с материальными, трудовыми и финансовыми ресурсами возник новый вид – информационные ресурсы, которые стали играть доминирующую роль.

Увеличение количества информации и спрос на нее, способствовали появлению отрасли, связанной с автоматизацией обработки информации – информационной технологии.

Информационные технологии применяются во всех сферах деятельности людей, это позволяет сократить время на обработку большого объема информации, поэтому целесообразно использовать информационные технологии в сферах деятельности с большим объемом информации.

Целью данной дипломной работы является разработка автоматизированной системы учета библиотечного фонда Покровской библиотеки филиал №2. Основными задачами в данной работе будет создание совокупности правил и процедур, обеспечивающих:

- 1) возможность регистрации пользователей;
- 2) регистрацию данных, сведений об общей величине, составе и движении фонда;
- 3) удаленную работу с читателями и данными о них;

Данное приложение упростит работу не только работникам, но и читателям библиотеки. Для этого будет создан кабинет читателя, в котором он сможет посмотреть свой личный формуляр, в нем содержится информация о книгах и их статусе, а также продлить срок хранения книги. Также пользователь сможет с помощью поиска на сайте узнать, если ли интересующая его книга в каталоге библиотеки.

Актуальность создания и внедрения данной системы в библиотеку состоит в том, что в настоящее время сотрудники библиотеки осуществляют процесс учета библиотечного фонда без использования каких-либо средств автоматизации. Работникам библиотеки приходится вручную вести учет книжного фонда, заполнять множество документов и составлять отчеты вручную, что есть трудоемким и время затратным делом.

В результате выполнения поставленной задачи будет разработано веб-приложение, способное реализовать цели преследуемые библиотекой и способное отвечать на множество вызовов, возникающих перед организациями в современную информационную эру.

Для того чтобы в полной мере достигнуть вышеперечисленных задач, следует выделить ряд ключевых компонентов, вторые включают в себя:

- 1) анализ деятельности организации;
- 2) анализ существующих систем, выполняющих схожие задачи;
- 3) постановка требований к разрабатываемой системе;
- 4) проектирование информационной системы;
- 5) реализация спроектированной системы;
- 6) разработка методики испытаний и ее реализация.

Данная выпускная квалификационная работа состоит из введения, трех глав, заключения, списка литературы и приложения.

Введение содержит общие сведения о работе, ее актуальность, цели, задачи.

Первая глава содержит описание организации, в которую данная система будет внедряться, анализ ее деятельности и всех аспектов которые нужно будет автоматизировать. Обзор и анализ существующих систем, что поможет убедиться в необходимости написания своей, а не использования готовой. Требования к автоматизированной системе, постановку задач, требующих решения.

В главе «Проектирование и разработка автоматизированной системы» будет описано выбор средств разработки и проектирование основных составных частей: базы данных, структуры веб-приложения, а также последовательность основных этапов разработки.

В главе «Тестирования автоматизированной системы » в соответствии с указанными требованиями будет разработан алгоритм проведения тестирования. По итогу будут приведены данные тестирования автоматизированной системы.

Выпускная квалификационная работа состоит из 55 страниц, 41 рисунка, 1 таблицы и приложения включающего 14 страниц.

ГЛАВА 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Особенности организации

Работа Покровской сельской библиотеки всегда была направлена на создание уютной атмосферы, где библиотекари прислушиваются к мнениям разных возрастных и социальных групп общины села, а также имеют индивидуальный подход к каждому пользователю, который имеет потребность не только в информации, но и в психологической поддержке. Поэтому одной из главных задач библиотеки, сегодня есть необходимость помочь пользователю справиться с потоком информации, распознать и отобрать самую качественную и предоставить ее в любое время в любом месте.

Основными направлениями работы любой библиотеки являются: комплектование и организация книжного фонда; обслуживание читателей.

Комплектование фондов библиотеки состоит из систематического выявления (путем просмотра библиографических источников и литературы) нужных для данной библиотеки изданий и приобретения их. От своевременности и полноты комплектования библиотеки в значительной мере зависит уровень обслуживания читателей.

Организация книжного фонда включает вопросы учета, расстановки, хранения литературы и доставки ее читателю. Правильная организация фонда облегчает читателю пользование литературой, библиотекаря – быстрое выполнение читательских требований, а также обеспечивает сохранность фондов как общественной собственности.

Обслуживание читателей библиотеки осуществляется различными путями:

- выдача литературы, как в читальный зал, так и за пределы библиотеки;
- помощь отдельным читателям и учреждениям в подборе необходимой им литературы;
- раскрытие книжных фондов библиотеки через систему библиотечных каталогов;
- составление информационно-библиографических пособий различного типа;
- пропаганда наиболее ценной литературы;
- репродуцирование текстов по заказам читателей и т.д.

Библиотечная деятельность связана с учетом большого количества операций, множество книг и читателей серьезно замедляют работу библиотекарей. Сложность поиска нужной книги в каталоге занимает длительное время, и целиком опирается на компетентность работников библиотеки.

В библиотеке ведется картотека читателей. Для ведения библиотечных каталогов, организации поиска требуемых изданий и библиотечной статистики в базе должны храниться сведения, большая часть которых размещаются в аннотированных каталожных карточках

В библиотеку приходит много книг из различных издательств. Каждой книге в библиотеке присваивают номер, а затем передают в различные отделы.

Также в библиотеке ведется картотека читателей. Каждому читателю присваивается номер читательского билета.

В случае выдачи экземпляра книги в библиотеке остается вкладыш, в котором указана дата выдачи, дата предполагаемого возврата и номер читательского билета.

Можно продлить книгу по телефону книгу назвав номер читательского билета и уникальный номер книги.

При возврате книги во вкладыше отмечается срок возврата. При просрочке возврата книги, читатель получает предупреждение. Когда предупреждений у читателя накапливается больше определенного предела, он лишается права пользования библиотекой на определенный срок.

При потере книги читатель лишается права пользования библиотекой на определенный срок, независимо от количества предупреждений.

1.2 Обзор и анализ существующих систем

На данный момент существует большое множество разнообразных программ для учета книг библиотечного фонда (1С:Библиотека Колледжа, АС-Библиотека-3, Библиотека MD 4.3, Автоматизированная информационно-библиотечная система OPAC-Global, Система ИРБИС). Данные программы охватывают все аспекты данной деятельности и представляют собой очень громоздкий функционал. Для многих, небольших учебно-образовательных учреждений такие решения являются очень дорогими и сложными в использовании. В таких случаях встает вопрос о создании специализированных приложений, обладающих только тем функционалом, который требуется для данной сферы деятельности. Такие приложения обладают положительными аспектами:

- они просты в эксплуатации;
- стоимость такой разработки гораздо ниже коммерческих проектов;
- не содержат лишний функционал;
- имеют возможность модернизации.

Поэтому целью данной работы является создание веб-приложения, которое будет содержать автоматизированную систему работы с библиотечным фондом.

Основными преимуществами веб-приложений перед десктопными являются:

- 1) пользователь не нуждается в установке дополнительного программного обеспечения;
- 2) веб-приложение не требовательно к ресурсам;
- 3) не предъявляет никаких требований к аппаратной платформе;
- 4) позволяет своим пользователям быть по-настоящему мобильными.

На данный момент большое количество библиотек имеет свой личный сайт. Однако все эти сайты в той или иной мере не отвечают тем требованиям, которые необходимо выполнять сайту, чтобы подобающим образом упростить ряд аспектов работы библиотеки, требующих автоматизации. Чтобы убедиться в этом, можно рассмотреть несколько таких сайтов и оценить то, как они справляются с поставленными задачами. При анализе главными критериями будут являться наличие электронного каталога и личного кабинета пользователя, так как непосредственно эти аспекты в первую очередь нуждаются в автоматизации.

Первой рассмотренной системой будет сайт Луганской областной универсальной библиотеки им. М. Горького, расположенный по следующей ссылке, <http://library.lg.ua/ru> . Главная страница рассматриваемого сайта представлена на рис. 1.1. На ней мы видим ленту новостей и виджет страницы в facebook.

Первый Code Club в Луганской области поделился своими достижениями

07.06.2018



Code Club в Старобельске – клуб программирования для детей от 8 до 13 лет – присоединился к всеукраинской и всемирной

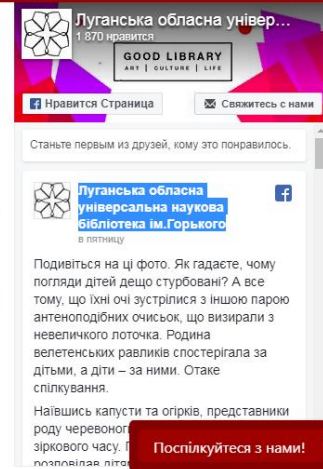


Рис. 1.1 Главная страница сайта

Рассмотрим то, как сайт реализует приведенные выше аспекты, требующие автоматизации.

На сайте имеется электронный каталог, он представлен на рис. 1.2. Также пользователь может посмотреть новые поступления и периодические издания.

rus ukr eng

Луганская областная универсальная научная библиотека им. М. Горького

- [Периодические издания](#)
- [Электронный каталог](#)
- [Новые поступления](#)
- [ЧЗВ](#)
- [Вход](#)

Поиск

Вид ресурса: Электронная библиотека

Шифр зберігання	Інвентарний номер	Автор	Назва	Місце видання
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Видавництво	Рік видання	ISBN/ISSN	Предметна рубрика	Класифікаційні індекси
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

© Ніццок Сергій 2018

Рис. 1.2. Электронный каталог

Хотя личный кабинет пользователя на сайте отсутствует, вы можете зарегистрироваться в данной библиотеке, заполнив соответствующую форму, представленную на рис. 1.3.

Рис. 1.3. Регистрация в библиотеки

Также на сайте вы можете задать интересующий вас вопрос. Для этого служит специальное диалоговое окно, расположенное в нижнем правом углу. Чтобы начать разговор необходимо заполнить имя и фамилию. Диалоговое окно изображено на рис. 1.4.

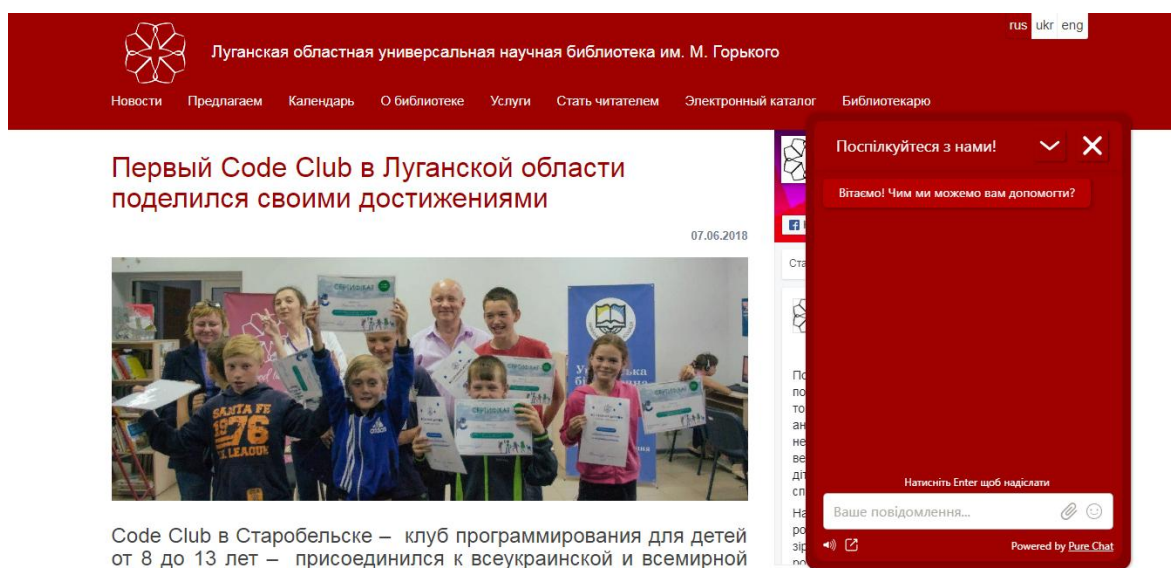


Рис. 1.4. Диалоговое окно

Рассмотрев данную систему и оценив все возможности, можно прийти к выводу, что данный сайт не в полной мере организует все аспекты деятельности библиотеки, подлежащие к автоматизации.

Рассмотрим сайт Троицкой ЦБС, который располагается по адресу: <https://sites.google.com/site/troitsklib/>. Главную страницу сайта можно увидеть на рис. 1.5. Дизайн сайта очень устарел. Цветовая гамма и шрифт текста подобраны неправильно из-за чего глаза быстро устают.

Рассмотрим то, как сайт реализует приведенные выше аспекты, требующие автоматизации.



Рис. 1.5. Главная страница сайта

На главной странице мы видим приветствие, и график работы. На боковой панели много не рабочих ссылок.

На сайте отсутствует электронный каталог, единственной страницей, выполняющей малую часть необходимого функционала, является раздел «информационные ресурсы» на которой находятся новые поступления и периодические издания. Данный раздел представлен на рис. 1.6.

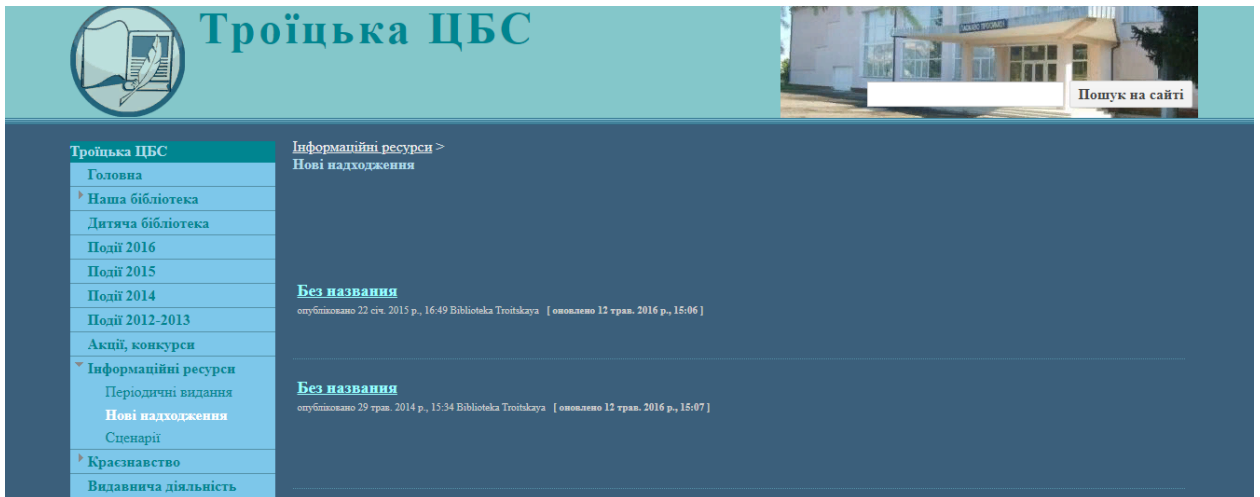


Рис. 1.6. Раздел «информационные ресурсы»

Возможность зарегистрироваться, либо отправить заявку, на сайте отсутствует. На сайте нет обратной связи, чтобы пользователь мог задать интересующий его вопрос.

Рассмотрев данную систему и оценив все возможности, можно прийти к выводу, что данный сайт не организует аспекты деятельности библиотеки, подлежащие к автоматизации.

Последней системой мы рассмотрим, будет сайт Национальной исторической библиотеки Украины, который располагается по адресу: <http://www.nibu.kiev.ua>. На рис. 1.7 изображена главная страница сайта.

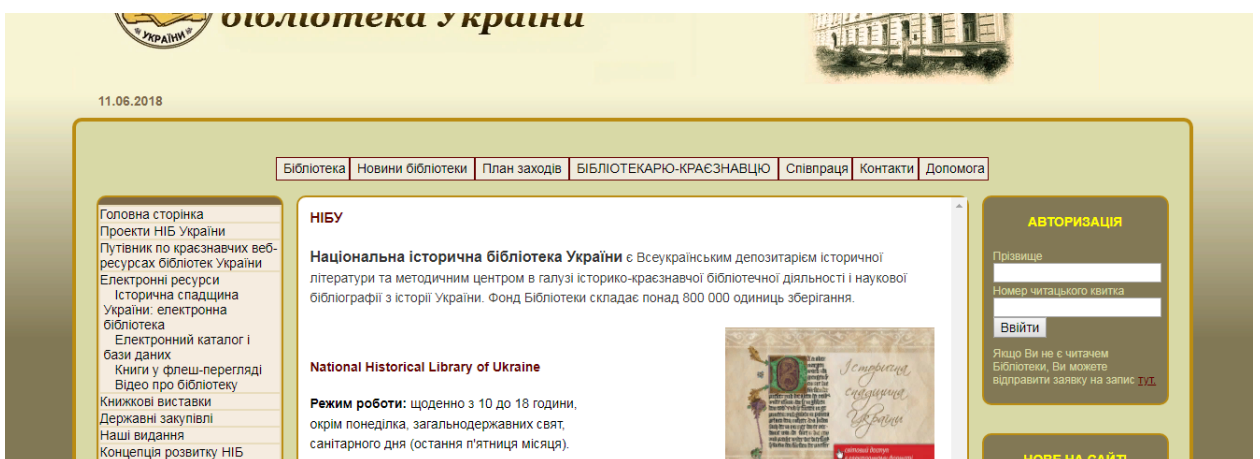


Рис. 1.7. Главная страница сайта

Рассмотрим то, как сайт реализует приведенные выше аспекты, требующие автоматизации.

На данном сайте имеется электронный каталог, который изображен на рис. 1.8. В нем можно совершить поиск по автору, названию и ключевым словам. Также на странице содержатся советы по формированию запроса для поиска интересующей книги.

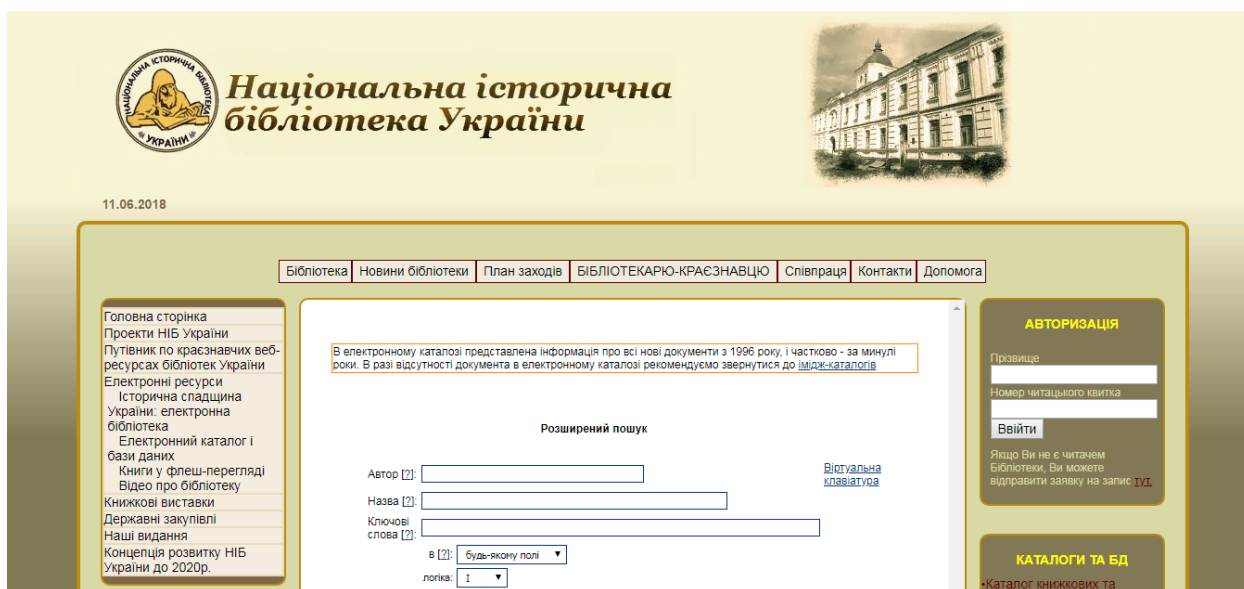


Рис. 1.8. Электронный каталог

Также на данном сайте есть страница регистрации. Для того чтобы зарегистрироваться, необходимо заполнить форму и отправить запрос на регистрации. После чего ваша заявка отправится на обработку работникам библиотеки. После одобрения регистрации на почту приходит письмо о том, что вы стали пользователем библиотеки и номер вашего читательского билета. Регистрационная форма приведена на рис. 1.9.

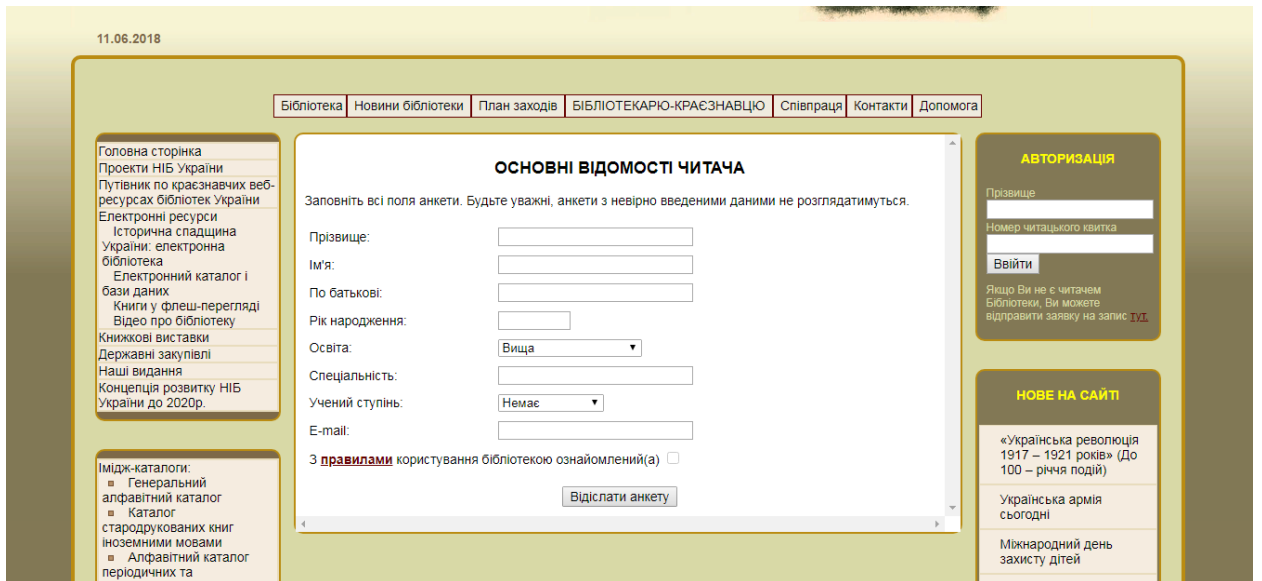


Рис. 1.9. Страница регистрации

По инструкции для того чтобы войти в личный кабинет нужно ввести фамилию и номер читательского билета, но после правильно введенных данных появляется ошибка что данные введены не верно. Данная информация приведена на рис. 1.10.



Рис. 1.10. Страница с информацией после регистрации

Таким образом, рассмотрев ряд систем, три из которых представлены выше, можно прийти к выводу, что те системы, которые уже существуют – не способный в полной мере удовлетворить требования, представленные к сайту потребностями библиотеки. Анализ этих систем основывался на трех критериях, указанных как аспекты деятельности библиотеки, подлежащие автоматизации. Для большей наглядности результаты анализа представлены в таблице 1.1.

Таблица 1.1

Анализ Существующих систем

	Электронный каталог	Личный кабинет	Удаленная работа с читателями
Луганская ОУБ им. М. Горького	+	–	+/-
Троицкая ЦБС	+/-	–	–
НИБ Украины	+	+/-	+

Исходя из анализа, возникает потребность разработать собственную автоматизированную систему. Которая будет реализовать приведенные выше аспекты, требующие автоматизации. Требования и весь функционал к автоматизированной системе мы определим в пункте 2.3. Требования к автоматизированной системе.

1.3 Требования к автоматизированной системе

Приведем основные требования для автоматизированной системы учета библиотечного фонда Покровской библиотеки филиал №2.

В автоматизированной системе должен быть реализован определенный ряд необходимого функционала.

1. Возможность регистрации пользователей. При регистрации должны присутствовать следующие поля для заполнения:

- логин;
- пароль;
- имя;
- фамилия;
- отчество;
- номер и серия паспорта;
- место работы (если есть);
- адрес почты.

2. Личный кабинет читателя. В нем пользователь должен иметь возможность просмотреть свой личный формуляр, в котором хранятся книги. Пользователь должен иметь возможность заказать книгу, продлить время хранения и задать интересующий вопрос администратору через соответствующую форму связи;

3. электронный каталог, в котором будут храниться все книги библиотеки и информация о них. Администратор должен иметь возможность добавить новую книгу, удалить либо изменить информацию о ней. При добавлении книги должны присутствовать следующие поля:

- инвентарный номер;
- заголовок книги;
- автора;
- марку автора;
- индекс отдела;
- место и год издания;
- количество страниц;

- международный номер;
- стоимость.

Требования к временным характеристикам

Требования к временным характеристикам зависят от количества запрашиваемых запросов и пропускной способности сервера. Автоматизированная система должна осуществлять запросы к серверу только по необходимости.

Требования к надежности

Надежная работа автоматизированной системы должна обеспечиваться выполнением следующих организационно-технических мероприятий:

- обеспечением бесперебойного питания сервера;
- обеспечением устойчивости системы путем разделения на независимые части (модули);
- обеспечением защищенности персональной информации пользователей путем хеширования паролей и ограничением доступа к данной информации неуполномоченных сотрудников;

Отказы из-за некорректных действий оператора

Отказы программы вследствие некорректных действий оператора (пользователя) при взаимодействии с автоматизированной системой. Во избежание возникновения всяческих отказов автоматизированной системы, по вышеперечисленной причине, необходимо обеспечить работу пользователя без предоставления ему административных привилегий.

Климатические условия эксплуатации

Климатические эксплуатационные условия, которые должны обеспечивать заданные характеристики, должны соответствовать требованиям, предъявляемым к техническим средствам в части условий их непосредственной эксплуатации.

Требования к численности и квалификации персонала

Для управления автоматизированной системой требуется как минимум один человек, который будет заносить актуальную информацию. Так же требуется персонал, который будет обслуживать сервер, на котором система будет размещена.

Требования к составу и параметрам технических средств

Для работы автоматизированной системы на стороне клиента требуется персональный компьютер с доступом в интернет. Для работы приложения на стороне сервера требуется персональный компьютер или специализированное оборудование, которые способны выполнять задачи сервера, а также имеющие доступ к сети интернет

Требования к информационным структурам и методам решения

Пользовательский интерфейс должен быть предельно ясным и содержать подсказки в случаях, когда это необходимо. Должен существовать программный доступ к управлению содержимым. Новые страницы с данными должны отображаться по следующему принципу – последняя запись отображается первой в списке.

Требования к исходным кодам и языкам программирования

Языком программирования, реализующим исходные коды стороне сервера, должен быть язык программирования PHP с использованием Codeigniter Framework. Интегрированной средой разработки автоматизированной системы должна быть использована среда PHP Storm. В качестве СУБД должна использоваться база данных MySQL. Языки программирования на стороне клиента: HTML и Javascript , включая библиотеку JQuery [8].

Требования к программным средствам, используемым программой

У клиента, для работы с автоматизированной системой должна быть установлена любая операционная система, поддерживающая взаимодействие с всемирной сетью интернет и работу любого современного веб-браузера.

ГЛАВА 2 ПРОЕКТИРОВАНИЕ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1. Применяемые средства разработки

В настоящее время существует множество средств, помощью которых можно написать сайт. Однако для реализации системы было принято решение использовать framework CodeIgniter, язык программирования PHP, СУБД MySQL. Эти инструменты выбраны не случайно, дело в том, что в последнее время стал популярным такой шаблон проектирования как «Модель–Вид–Контроллер», который является идеальной основой для разработки сложного веб-приложения [3].

CodeIgniter – инструмент для создания веб-приложения на PHP. Его основная задача ускорить создание проектов, предоставляя богатый комплект готовых библиотек для общих задач, с легким интерфейсом и логической структурой. CodeIgniter позволяет творчески сосредоточиться на проекте благодаря минимизированному объему кода для достижения поставленных задач. Он распространяется под лицензией Apache/BSD-style с открытым исходным кодом. Также данный framework тщательно документирован [7].

Исходя из этого можно выделить основные преимущества данного фреймворка:

- простота;
- низкие требования к ресурсам;
- скорость работы;
- функциональность;
- оптимизация.

PHP – скриптовый язык общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из

лидеров среди языков, применяющихся для создания динамических веб-сайтов [5]. Язык и его интерпретатор (Zend Engine) разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL. Язык довольно легок в написании, так как его синтаксис схож с синтаксисом C или perl. В PHP нет строгой типизации и совсем отсутствует необходимость по освобождению памяти или выделению памяти. [4]

Выбор данного языка обусловлен пятью важными характеристиками:

- традиционностью;
- простотой;
- эффективностью;
- безопасностью;
- гибкостью.

MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Является наиболее приспособленной СУБД для применения в среде web [6].

Основные преимущества MySQL:

- многопоточность, поддержка нескольких одновременных запросов;
- быстрая поддержка транзакций через механизм InnoDB.
- оптимизация связей с присоединением многих данных за один проход;
- записи фиксированной и переменной длины;
- гибкая поддержка форматов чисел, строк переменной длины и меток времени;
- интерфейс с языками C и Perl, PHP;

- быстрая работа, масштабируемость;
- совместимость с ANSI SQL;
- бесплатна в большинстве случаев;
- хорошая поддержка со стороны провайдеров услуг хостинга.

2.1 Проектирование и разработка базы данных

Основой для любого компьютерного приложения являются данные, которыми оно манипулирует, поэтому проектирование программы стоит начать именно с них. Для того чтобы реализовать требуемый функционал, нужна база данных, которая будет хранить в себе всю необходимую для работы приложения информацию. Ранее мы уже определили СУБД MySQL, именно с ней мы и будем работать через phpmyadmin – веб-интерфейс для администрирования базы данных.

Покровская библиотека представила документы, которые нужно хранить в базе данных, а именно: формуляр читателя, каталог, журнал посещения, инвентарная книга. База данных будет строиться исходя из этих документов.

Инфологическая модель – построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Цель данной модели обеспечить естественный для человека способ сбора и представления информации, которая будет храниться в базе данных. Инфологическая модель БД представлена на рис. 2.1.



Рис. 2.1. Инфологическая модель БД

Даталогическое проектирование – модель логического уровня, представляющая собой отображение логических связей между элементами данных, независимо от их содержания и среды хранения.

Исходными данными для даталогического проектирования является инфологическая модель предметной области. Конечным результатом даталогического проектирования является описание логической структуры базы данных [10].

На данном этапе даталогического проектирования строится логическая структура базы данных нашего веб-приложения. При этом происходит преобразование исходной инфологической модели (см. рис. 2.1) в модель данных, которая поддерживается конкретной СУБД. После этого производится проверка адекватности даталогической модели, отображаемой предметной области. Ниже на рис. 2.2 представлена даталогическая модель базы данных.



Рис. 2.2. Дatalogическая модель БД

Разработка физической модели БД – процесс подготовки описания реализации базы данных на вторичных запоминающих устройствах. Данная модель изображена на рис. 2.3.

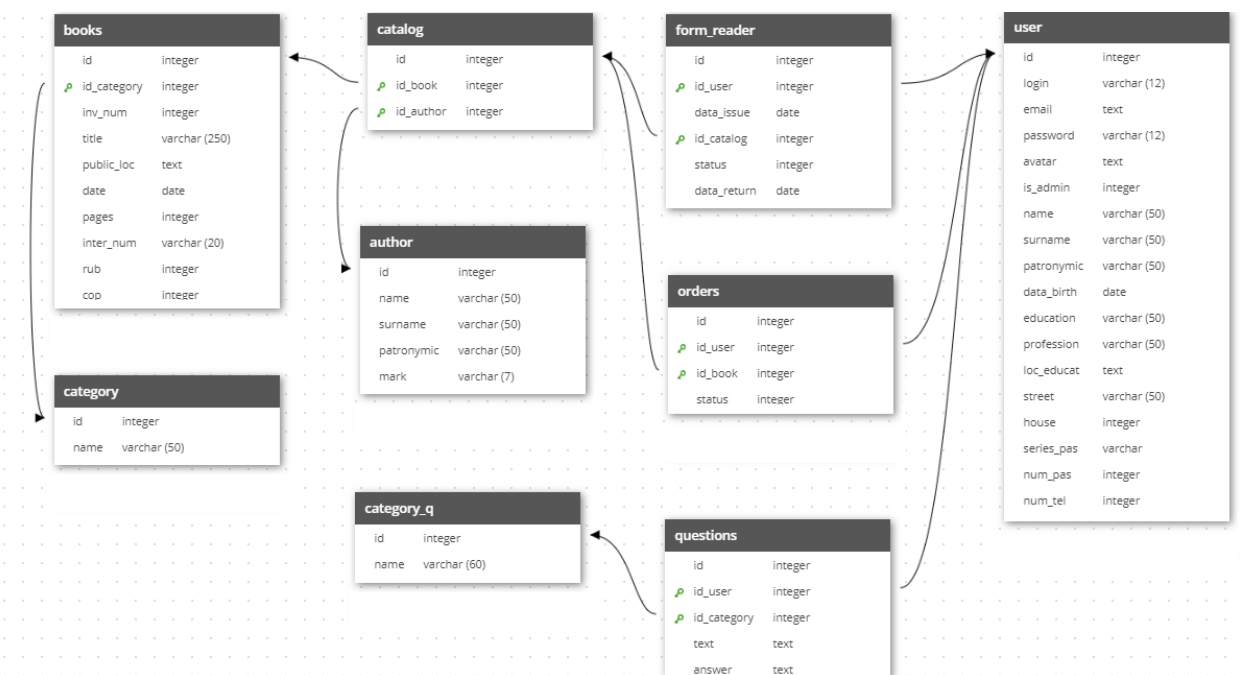


Рис. 2.3. Физическая модель БД

Физическая модель БД отображает таблицы и их названия, поля таблиц, их типы и размеры, и связи между таблицами. [6]

Ниже следует перечень таблиц, на основе которых реализуется база данных.

Таблица «category» предназначена хранить категории книг. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«name» – название категории, типа «varchar», размерностью 50 символов;

Таблица «books» предназначена хранить книги. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«id_category» – внешний ключ, ссылающийся на таблицу «category» и содержит категорию книги, типа «int»;

«inv_num» – инвентарный номер, типа «varchar»;

«title» – заголовок, типа «varchar», размерностью 250 символов;

«public_loc» – место публикации, типа «text»;

«date» – дата публикации, типа «date»;

«pages» – количество страниц, типа «int»;

«inter_num» – международный номер, типа «varchar», размерностью 20 символов;

«rub» – количество рублей, типа «int»;

«cop» – количество копеек, типа «int»;

Таблица «author» хранит всех авторов. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«name» – имя, типа «varchar», размерностью 50 символов;

«surname» – фамилия, типа «varchar», размерностью 50 символов;

«patronymic» – отчество, типа «varchar», размерностью 50 символов;

«mark» – марка автора, типа «varchar», размерностью 7 символов;

Таблица «catalog» хранит автора и написанные им книги. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«id_author» – внешний ключ, ссылающийся на таблицу «author» и содержит идентификатор автора, типа «int»;

«id_book» – внешний ключ, ссылающийся на таблицу «books» и содержит идентификатор книги, типа «int»;

Таблица «form_reader» предназначена для хранения книг, которые находятся у пользователя. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«id_user» – внешний ключ, ссылающийся на таблицу «user» и содержит идентификатор пользователя, типа «int»;

«data_issue» – дата выдачи, типа «date»;

«id_catalog» – внешний ключ, ссылающийся на таблицу «catalog» и содержит идентификатор книги из каталога, типа «int»;

«status» – статус книги, типа «int»;

«data_return» – дата возврата, типа «date»;

Таблица «user» содержит пользователей и информацию о них, так же используется при регистрации. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«login» – логин, типа «varchar», размерностью 12 символов;

«email» – почтовый адрес, типа «varchar»;

«password» – пароль, типа «varchar», размерностью 12 символов;

«avatar» – фотография, типа «varchar»;

«is_admin» – статус администратора, типа «int»;

«name» – имя, типа «varchar», размерностью 50 символов;

«surname» – фамилия, типа «varchar», размерностью 50 символов;

«patronymic» – отчество, типа «varchar», размерностью 50 символов;
«data_birth» – дата рождения, типа «data»;
«education» – образование, типа «varchar», размерностью 50 символов;
«profession» – профессия, типа «varchar», размерностью 50 символов;
«loc_education» – место обучения, типа «varchar», размерностью 50

СИМВОЛОВ;

«street» – название улицы, типа «varchar», размерностью 50 символов;
«house» – номер дома, типа «int»;
«series_pas» – серия паспорта, типа «varchar»;
«num_pas» – номер паспорта, типа «int»;
«num_tel» – номер телефона, типа «int»;

Таблица «question» содержит все заданные вопросы пользователей и ответы на них. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«id_user» – внешний ключ, ссылающийся на таблицу «user» и содержит идентификатор пользователя, типа «int»;

«id_category» – внешний ключ, ссылающийся на таблицу «category_q» и содержит категорию вопроса, типа «int»;

«text» – текст вопроса типа «text»;

«answer» – ответ на вопрос типа «text»;

Таблица «category_q» предназначена хранить категории вопросов. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«name» – название категории, типа «varchar», размерностью 60 символов;

Таблица «order» хранит все заказанные книги и заказавших пользователей. Она имеет следующую структуру:

«id» – первичный ключ, типа «int»;

«id_user» – внешний ключ, ссылающийся на таблицу «user» и содержит идентификатор пользователя, типа «int»;

«id_book» – внешний ключ, ссылающийся на таблицу «books» и содержит идентификатор книги, типа «int»;

«status» – хранит текущий статус заказа, типа «int»;

2.3. Проектирование структуры и разработка веб-приложения

Шаблон проектирования «Модель–Представление–Контроллер»

«Model–View–Controller» – является схемой разделения данных приложения, интерфейса пользователя и управляющей логики на три компонента: модель, представление, и контроллер – таким образом, модификация отдельного компонента осуществляется независимо. [1]

Особенности архитектурного шаблона MVC

При использовании архитектурного шаблона MVC предполагается, разделение приложения MVC, как минимум, на три части, которые описаны ниже.

Модель (Model) представляет структуру данных. Обычно классы моделей содержат функции помогающие получить, вставить и обновить информацию из БД;

Представление (View) информация, которая будет представлена пользователю. Вид обычно является страницей, но в CodeIgniter, может также быть и фрагментом страницы таким как header или footer. Это также может быть RSS страница, или любой другой тип страницы;

Контроллер (Controller) служит как посредник между Моделью и Видом, и любых других ресурсов необходимых для HTTP запроса и формирования веб-страницы.

Все составляющие архитектуры MVC четко определены и самодостаточны – они реализуют принцип разделения ответственности. Логика, управляющая данными модели, содержится только в ней, логика, которая отображает данные, располагается исключительно в представлении, а тот код, который обрабатывает пользовательский ввод и запросы находится в контроллере. Таким образом, четко разграничивая все части в приложении, сопровождение и расширение в течение срока его существования происходит значительно проще, вне зависимости от его увеличения и усложнения [9].

CodeIgniter имеет довольно свободный подход к MVC, т.к. Модели не являются необходимыми. Если не нужно добавлять разделение (между логикой и представлением), или находите что содержание Моделей требует большей гибкости, можно их не использовать и создать свое приложение используя лишь Контроллеры и Виды. CodeIgniter также позволяет подключать собственные скрипты и даже разрабатывать библиотеки ядра для системы [7]. Взаимодействия между моделью, представлением и контроллером показаны на рис. 2.4.

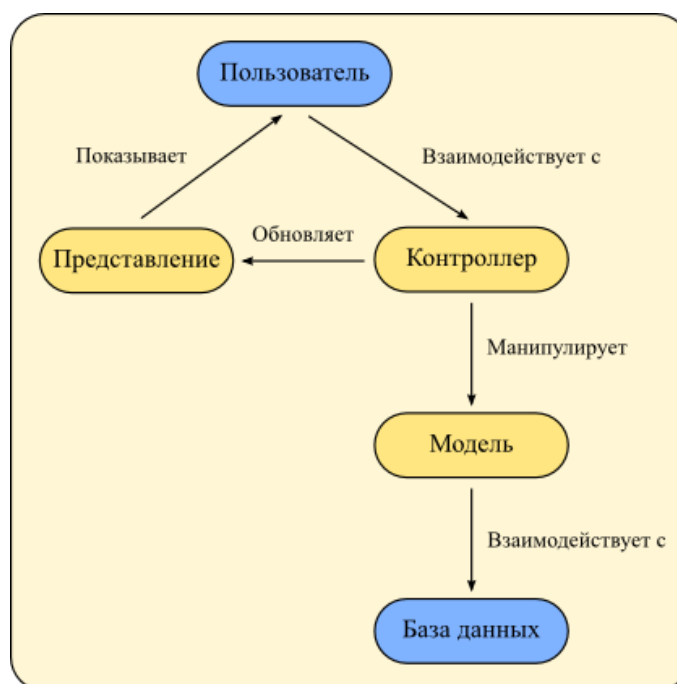


Рис. 2.4. Модель MVC

Итак, определившись с архитектурой клиентской части приложения, начнем проектировать серверную часть. Она будет отвечать за организацию маршрутизации (routing), а также обращаться к базе данных.

Так как написанное приложение использует большой объем кода, то нет возможности прокомментировать весь код целиком. Однако следует описать те разработки, которые являются ключевыми и представляют особый интерес для рассмотрения. Как и было заявлено в требованиях,

Описание нашей разработки приложения начнем с авторизации и регистрации пользователей. За эти функции отвечает контроллер «User». Рассмотрим функцию регистрации пользователя, которая называется «register». Вначале идет проверка полей, если одно из полей заполнено неправильно либо не осталось пустым, то функция вернет ошибку, фрагмент кода приведен в листинге 2.1.

Листинг 2.1. Проверка заполненных полей

```
if ($this->form_validation->run() === false) {  
$this->load->view('header');  
$this->load->view('user/register/register', $data);  
$this->load->view('footer');  
}
```

Конец листинга 2.1.

Далее функция присваивает всем значениям соответствующие переменные.

После проверки на удачное присвоение значений, вся информация заносится в базу данных и создается новая сессия, в которой содержится идентификатор пользователя, логин, идентификатор сессии, статус

администратора, статус верификации. Соответствующий фрагмент кода представлен в листинге 2.2.

Листинг 2.2. Создание нового пользователя

```
$user_id = $this->user_model->get_user_id_from_username($username);  
$user = $this->user_model->get_user($user_id);  
    $_SESSION['user_id'] = (int)$user->id;  
    $_SESSION['username'] = (string)$user->username;  
    $_SESSION['logged_in'] = (bool>true;  
    $_SESSION['is_confirmed'] = (bool)$user->is_confirmed;  
    $_SESSION['is_admin'] = (bool)$user->is_admin;  
    redirect($redirect);
```

Конец листинга 2.2.

После удачного занесения в БД пользователя перенаправит в личный кабинет. Если по какой, либо причине новая запись не занесется в БД, пользователю выведется соответствующая ошибка.

Перейдем к авторизации, за нее отвечает функция «login». Рассмотрим отдельные фрагменты кода. Вначале идет проверка, заполнены ли поля логин и пароль. Если все поля заполнены, то значениям полей присваиваются переменные. Фрагмент кода представлен в листинге 2.3.

Листинг 2.3. Функция авторизации пользователя

```
$this->load->helper('form'); $this->load->library('form_validation');  
$this->form_validation->set_rules('username', 'Username');  
$this->form_validation->set_rules('password', 'Password', 'required');  
if ($this->form_validation->run() == false) {  
    $this->load->view('header');
```

Продолжение листинга 2.3.

```
$this->load->view('user/login/login');
    $this->load->view('footer');
} else {
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $redirect = $this->input->post('redirect');}
```

Конец листинга 2.3.

Далее идет проверка на существование данного логина в БД и проверка соответствия паролей. Если логин будет не найден либо пароль будет не соответствовать, пользователю выдаст соответствующую ошибку. Фрагмент кода представлен в листинге 2.4.

Листинг 2.4. Проверка логина и пароля

```
if ($this->user_model->resolve_user_login($username, $password)) {
    $user = $this->user_model->get_user($user_id);
    $_SESSION['user_id'] = (int)$user->id;
    $_SESSION['logged_in'] = (bool>true;
    $_SESSION['is_confirmed'] = (bool)$user->is_confirmed;
    $_SESSION['is_admin'] = (bool)$user->is_admin;
    redirect($redirect);}
```

Конец листинга 2.4.

При успешной проверке будет создана новая сессия, которая будет хранить идентификатор пользователя, логин, идентификатор сессии, статус администратора, статус верификации.

Последняя функция, которая находится в данном классе это функция конец сеанса «logout». Данная функция предназначена для завершения сеанса пользователя. Код функции представлен в листинге 2.5.

Листинг 2.5. Функция завершения сеанса

```
$data = new stdClass();  
if (isset($_SESSION['logged_in']) && $_SESSION['logged_in'] === true) {  
    foreach ($_SESSION as $key => $value) {  
        unset($_SESSION[$key]);  
    }  
    redirect('/');  
} else {  
    redirect('/');}  
}
```

Конец листинга 2.5.

Здесь идет проверка на существование сессии, если сессия существует, то с помощью функции `unset` сессия "забывает" значение заданной сессионной переменной.

На этом этап создания авторизации и регистрации пользователей закончен. Перейдем к следующему этапу.

Далее рассмотрим класс реализующий функционал каталога. За это отвечает класс «Catalog». Ниже рассмотрим фрагменты кода, реализующие основной функционал.

Для вывода всех записей каталога отвечает функция «catalog». Данная функция обращается к таблице «catalog» в базе данных и выводит все записи данной таблицы. Записи выводятся в порядке убывания идентификатора. Количество записей на одной странице равно десяти. Также эта функция содержит пагинацию. Код данной функции представлен в листинге 2.6.

Листинг 2.6. Вывод всех записей

```
$config['base_url'] = base_url('admin/catalog/index');
$config['total_rows'] = $this->db->count_all('catalog');
$config['per_page'] = 9;
$config['num_links'] = 3;
$config['uri_segment'] = 4;
$this->pagination->initialize($config);
$data['pager']=$this->pagination->create_links();
$from=intval($this->uri->segment(4));
$this->db->limit(9,$from);
$this->db->order_by("id", "desc");
$data['query']=$this->db->get('catalog');
$this->load->view('header');
$this->load->view('admin/catalog/index',$data);
$this->load->view('footer');
```

Конец листинга 2.6.

Перейдем к функции добавления записи в каталог. Функция «add_catalog» реализует добавление новой книги. Вначале функция принимает все значения формы, если значения были приняты, присваивает их соответствующим локальным переменным. Далее все данные заносятся в таблицу. Если запись занесена без ошибок, пользователю показывается соответствующее сообщение об удачном завершении операции. Если же при добавлении произошла ошибка, пользователя проинформирует об этом всплывающее сообщение. После будет предложено повторить попытку добавления новой записи. Фрагмент кода представлен в листинге 2.7.

Листинг 2.7. Присвоение переменных

```
$data = array('title' => $title, 'author' => $author,
             'category' => $category, 'author_mark' => $mark,
             'data' => $data, 'invent_num' => $inventnum,
             'loc_public' => $map, 'rub' => $rub, 'cop' => $kop, 'eu_number' => $eunum,
             'pages' => $pages );
if( $this->db->insert('catalog', $data)) {
    echo'<div class="on" id="on" role="alert">Запись добавлена.';
    }else{
    echo'<div class="err" id="err" role="alert">Ошибка при
добавлении.</div>';}
```

Конец листинга 2.7.

Удаления записи из таблицы каталога производится с помощью функции «del_catalog». Код данной функции представлен в листинге 2.8.

Листинг 2.8. Удаление записи

```
$delId = trim($_POST['delId']);$this->db->where('id', $delId);
if($this->db->delete('catalog')) {
    echo'<div class="on" id="on" role="alert">Запись удалена. <a
href="">Нажмите, чтобы отменить удаление.</a></div>';
    }else{
    echo'<div class="err" id="err" role="alert">Ошибка при
удалении.</div>';}
```

Конец листинга 2.8.

ГЛАВА 3 ТЕСТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ СИСТЕМЫ

3.1 Программа и методика испытаний

Объектом испытаний, рассматриваемым в данной главе, является ранее спроектированная и разработанная «Автоматизированная система учета библиотечного фонда Покровской библиотеки филиал №2».

Целью испытаний информационной системы являются проверка ее работоспособности, а так же ее соответствие указанным в первой главе требованиям к автоматизированной системе.

Полный перечень требований к автоматизированной системе перечислен в третьем разделе первой главы, однако ключевыми пунктами, требующими установки соответствия, являются следующие:

- регистрация пользователей;
- личный кабинет;
- электронный каталог;
- заказ и продление срока хранения книги;

В соответствии с указанными требованиями был разработан следующий алгоритм проведения тестирования «Автоматизированной системы учета библиотечного фонда Покровской библиотеки филиал №2».

1. Протестировать работоспособность системы для незарегистрированного пользователя:

- 1.1. Возможность регистрации.
- 1.2. Поиск по каталогу.

2. Протестировать работоспособность системы для зарегистрированного пользователя в роли обычного пользователя библиотеки.

- 2.1. Возможность перехода в личный кабинет.
- 2.2. Вывод формуляра пользователя и анкеты пользователя.

- 2.3. Возможность продления срока хранения книги.
- 2.4. Возможность заказать книгу.
- 2.5. Возможность задать интересующий вопрос администратору, через форму обратной связи.

3. Протестировать работоспособность системы для авторизованного пользователя в роли администратора.

- 3.1. Добавление записей в каталоге.
- 3.2. Удаление записей в каталоге.
- 3.3. Обновление записей в каталоге.
- 3.4. Поиск записей в каталоге.
- 3.5. Добавление записи в формуляр пользователя.
- 3.6. Изменение статуса книги.
- 3.7. Изменения статуса заказа.
- 3.8. Обновление данных о пользователе.
- 3.9. Удаления пользователя.
- 3.10. Возможность ответить на вопрос пользователя.
- 3.11. Возможность изменить статус заказа.

3.2 Испытание информационной системы

Испытание автоматизированной системы следует начать с тестирования регистрации пользователей. На рис. 3.1 изображена заполненная форма регистрации. В данной форме все поля обязательны к заполнению. Стоит подметить, что проверка полей «Электронная почта», «Дата рождения», «Номер телефона» происходит во время ввода данных в соответствующие поля.

Добро пожаловать на страницу Регистрации
пожалуйста, заполните все поля.

Логин:	Электронная почта:	
<input type="text" value="username"/>	<input type="text" value="username@mail.ru"/>	
Пароль:	Повторите пароль:	
<input type="password" value="....."/>	<input type="password" value="....."/>	
Имя:	Фамилия:	Отчество:
<input type="text" value="Виктор"/>	<input type="text" value="Смелый"/>	<input type="text" value="Александрович"/>
Дата рождения:	Адрес проживания:	
<input type="text" value="22.12.1996"/>	<input type="text" value="Крупская, дом 99"/>	
Образование:	Место обучения:	
<input type="text" value="Высшее"/>	<input type="text" value="Луганский университет имени Тараса Ште"/>	
Номер телефона:	Профессия:	
<input type="text" value="+38997056546"/>	<input type="text" value="нет"/>	

Рис. 3.1. Страница регистрации

При нажатии на кнопку «Зарегистрироваться», происходит проверка полей, на корректность заполнения. После успешной регистрации пользователь попадает в личный кабинет. Личный кабинет представлен на рис. 3.2.

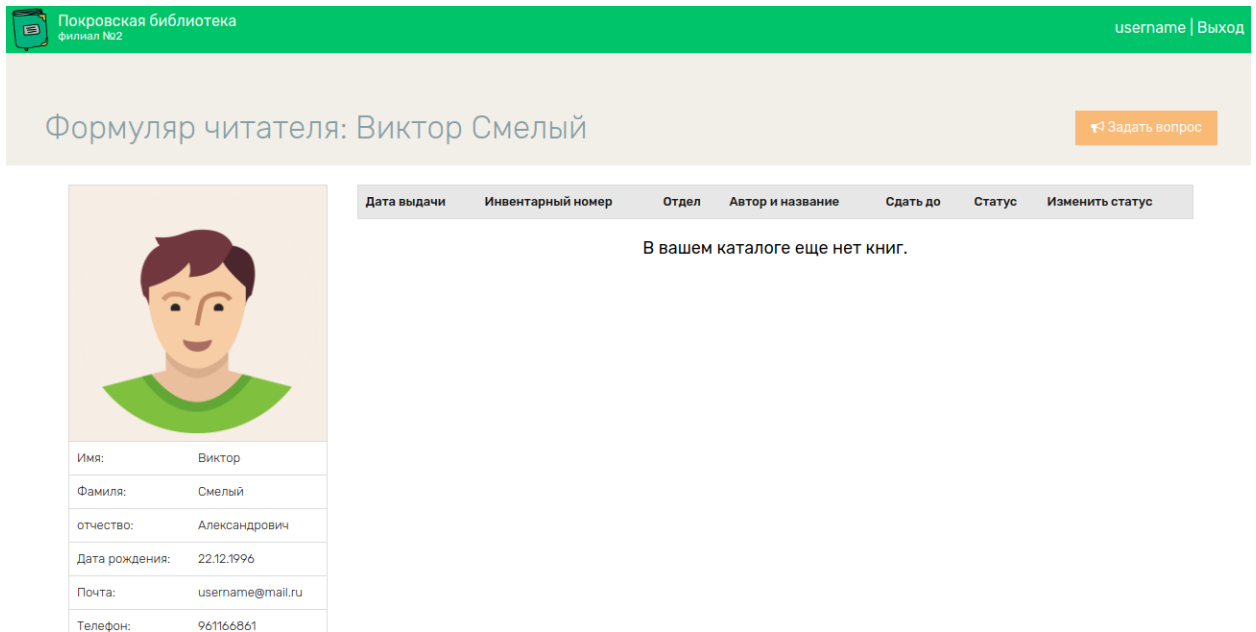


Рис. 3.2. Личный кабинет пользователя

Продолжим тестирование личного кабинета пользователя. Протестируем вывод формуляра читателя и правильность значения поля «Статус». В этом поле должен отображаться текущий статус книги. Вывод формуляра представлен на рис. 3.3.

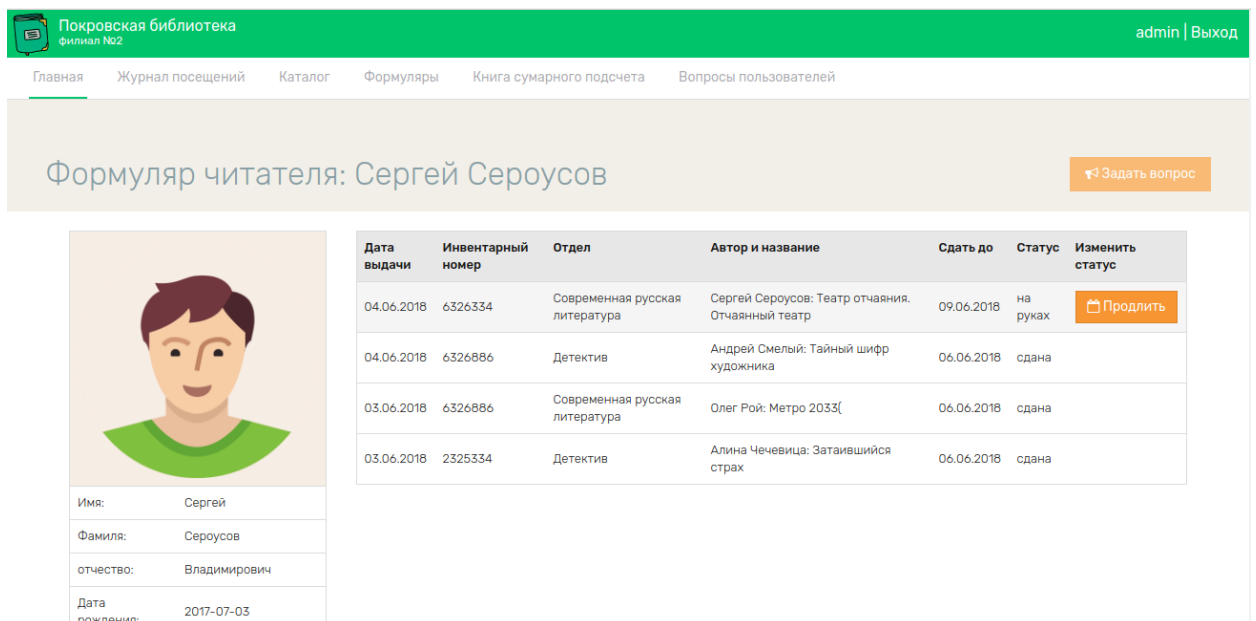


Рис. 3.3. Вывод формуляра читателя

Протестируем возможность продления книги. Нажмем на кнопку «Продлить» напротив первой записи. Результат представлен на рис. 3.4.

Покровская библиотека
филиал №2

admin | Выход

Главная Журнал посещений Каталог Формуляры Книга сумарного подсчета Вопросы пользователей

Формуляр читателя: Сергей Сероусов [Задать вопрос](#)

Дата выдачи	Инвентарный номер	Отдел	Автор и название	Сдать до	Статус	Изменить статус
04.06.2018	6326334	Современная русская литература	Сергей Сероусов: Театр отчаяния. Отчаянный театр	15.06.2018	на руках	
04.06.2018	6326886	Детектив	Андрей Смелый: Тайный шифр художника	06.06.2018	сдана	
03.06.2018	6326886	Современная русская литература	Олег Рой: Метро 2033[06.06.2018	сдана	
03.06.2018	2325334	Детектив	Алина Чечевица: Затаившийся страх	06.06.2018	сдана	

Имя: Сергей
Фамилия: Сероусов
отчество: Владимирович
Дата рождения: 2017-07-03

Дата возврата книги обновлена.

Рис. 3.4. Продления даты хранения книги

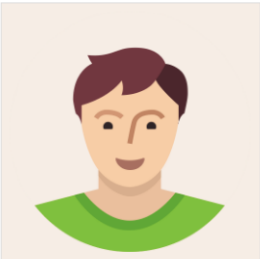
После нажатия на кнопку дата возврата книги увеличилась на 5 дней, о чем нас информирует всплывающее сообщение (см. рис. 3.4).

Продолжим тестирование личного кабинета, но уже от имени администратора. Протестируем возможность добавления записи в формуляр пользователя. Для того чтобы добавить новую запись в формуляр пользователя, нужно перейти в его личный кабинет, нажать на кнопку «Добавить книгу». Как показано на рис. 3.5.

Покровская библиотека филиал №2 admin | Выход

Главная Журнал посещений Каталог Формуляры Книга сумарного подсчета Вопросы пользователей

Формуляр читателя: Камиль Пуля



Дата выдачи	Инвентарный номер	Отдел	Автор и название	Сдать до	Статус	Изменить статус
04.06.2018	6326334	Современная русская литература	Сергей Сероусов: Театр отчаяния. Отчаянный театр	09.06.2018	сдана	
04.06.2018	6326886	Детектив	Андрей Смелый: Тайный шифр художника	06.06.2018	сдана	
03.06.2018	6326886	Современная русская литература	Олег Рой: Метро 2033(06.06.2018	сдана	
03.06.2018	2325334	Детектив	Алина Чечевица: Затаившийся страх	06.06.2018	сдана	

Имя:	Камиль
Фамилия:	Пуля
отчество:	Пуляевич
Дата рождения:	22.12.1893

Рис. 3.5. Заполнение поля и добавление записи

После нажатия появится дополнительное поле, в которое необходимо ввести инвентарный номер книги (см. рис.3.5).

Покровская библиотека филиал №2 admin | Выход

Главная Журнал посещений Каталог Формуляры Книга сумарного подсчета Вопросы пользователей

Формуляр читателя: Камиль Пуля



Дата выдачи	Инвентарный номер	Отдел	Автор и название	Сдать до	Статус	Изменить статус
08.06.2018	9999999	Современная русская литература	Сергей Сероусов: Книга для проверки	13.06.2018	на руках	<input type="button" value="Подтвердить возврат"/>
04.06.2018	6326334	Современная русская литература	Сергей Сероусов: Театр отчаяния. Отчаянный театр	09.06.2018	сдана	
04.06.2018	6326886	Детектив	Андрей Смелый: Тайный шифр художника	06.06.2018	сдана	
03.06.2018	6326886	Современная русская литература	Олег Рой: Метро 2033(06.06.2018	сдана	
03.06.2018	2325334	Детектив	Алина Чечевица: Затаившийся страх	06.06.2018	сдана	

Имя:	Камиль
Фамилия:	Пуля
отчество:	Пуляевич

Рис. 3.7. Результат добавления книги в формуляр

После выполнения вышеперечисленных действий, в формуляре пользователя появилась книга с соответствующим инвентарным номером.

Далее протестируем возможность изменения статуса книги. Текущей статус книги указан в колонке «Статус». На рис. 3.8 приведен личный формуляр читателя со всеми книгами и их статуса.

Покровская библиотека филиал №2 admin | Выход

Главная Журнал посещений Каталог Формуляры Книга сумарного подсчета Вопросы пользователей

Формуляр читателя: Камиль Пуля + Добавить книгу

Дата выдачи	Инвентарный номер	Отдел	Автор и название	Сдать до	Статус	Изменить статус
04.06.2018	6326334	Современная русская литература	Сергей Сероусов: Театр отчаяния. Отчаянный театр	09.06.2018	на руках	<input type="button" value="Подтвердить возврат"/>
04.06.2018	6326886	Детектив	Андрей Смелый: Тайный шифр художника	06.06.2018	сдана	
03.06.2018	6326886	Современная русская литература	Олег Рой: Метро 2033[06.06.2018	сдана	
03.06.2018	2326334	Детектив	Алина Чечевица: Затаившийся страх	06.06.2018	сдана	

Имя: Камиль

Рис. 3.8. Статус книги до нажатия

Чтобы изменить статус «на руках» на статус «сдана» тем подтвердив, что читатель принес книгу, нужно нажать на кнопку «Подтвердить возврат» напротив соответствующей записи. Результат данных действий представлен на рис. 3.9.

Покровская библиотека филиал №2 admin | Выход

Главная Журнал посещений Каталог Формуляры Книга сумарного подсчета Вопросы пользователей

Формуляр читателя: Камиль Пуля + Добавить книгу

Дата выдачи	Инвентарный номер	Отдел	Автор и название	Сдать до	Статус	Изменить статус
04.06.2018	6326334	Современная русская литература	Сергей Сероусов: Театр отчаяния. Отчаянный театр	09.06.2018	сдана	
04.06.2018	6326886	Детектив	Андрей Смелый: Тайный шифр художника	06.06.2018	сдана	
03.06.2018	6326886	Современная русская литература	Олег Рой: Метро 2033[06.06.2018	сдана	
03.06.2018	2326334	Детектив	Алина Чечевица: Затаившийся страх	06.06.2018	сдана	

Имя: Камиль

Рис. 3.9. Изменение статуса книги после нажатия

Перейдем к тестированию каталога. Протестируем вывод записей каталога. На рис. 3.10 приведен каталог со всеми записями, которые он содержит.

Категория	Марка автора	Инвентарный номер	Автор	Название	Место публикации	Дата	Страниц	EU-номер	Руб/Коп	Действия
<input type="radio"/> Детектив	ОР 44	6326221		Бойтесь данайцев, дары приносящих	Белгородский учебная типография	24.08.2007	422	578-5-04-014859-2	390 0	<input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/> Детектив	ЕГ 42	2325334	Алина Чечевица	Затаившийся страх	Белгородская национальная библиотека	11.02.2017	480	543-1-389-14762-6	250 17	<input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/> Современная русская литература	ДА 44	6326886	Олег Рой	Метро 2033[Луганская типография имени Дзюбы	28.12.2014	700	578-5-04-011459-2	450 0	<input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/> Детектив	ОР 12	6326886	Андрей Смелый	Тайный шифр художника	Луганская типография имени Дзюбы	30.01.2018	360	978-5-04-092459-2	120 0	<input type="checkbox"/> <input type="checkbox"/>
<input type="radio"/> Современная русская литература	ЕГ 42	6326334	Сергей Сероусов	Театр отчаяния. Отчаянный театр	Белгородская национальная библиотека	12.06.2018	1280	978-5-389-14762-1	589 0	<input type="checkbox"/> <input type="checkbox"/>

Рис. 3.10. Вывод каталога

Далее протестируем добавление новой записи в каталог. Чтобы добавить запись, нужно нажать на кнопку «Добавить запись». После нажатия появится окно с полями для ввода. Всплывающее окно приведено на рис. 3.11.

Добавить книгу в каталог

Рис. 3.11. Заполнение формы и добавление записи

В появившемся окне нужно заполнить поля, стоит учесть, что все поля обязательны к заполнению. После заполнения нажать на кнопку «Добавить в каталог». Новая запись появится в самом начале таблицы. Результат действий приведет на рис. 3.12.

Покровская библиотечка филиал №2 admin | Выход

Главная Журнал посещений Каталог Формуляры Книга сумарного подсчета Вопросы пользователей

Каталог книг + Добавить книгу Удалить отмеченные Поиск

Категория	Марка автора	Инвентарный номер	Автор	Название	Место публикации	Дата	Страниц	EU-номер	Руб/Коп	Действия
Современная русская литература	СС 42	1234567	Сергей Сероусов	Книга для проверки	Белгородский национальный исследовательский университет	08.06.2018	240	999-8-77-654321-0	6000	
Детектив	ОР 44	6326221	Алексей Смелый	Бойтесь данайцев, дары приносящих	Белгородский учебная типография	24.08.2007	422	578-5-04-014859-2	3900	
Детектив	ЕГ 42	2325334	Алина Чечевица	Затаившийся страх	Белгородская национальная библиотека	11.02.2017	480	543-1-389-14762-6	25017	
Современная русская литература	ДА 44	6326886	Олег Рой	Метро 2033	Луганская типография имени Дзюбы	28.12.2014	700	578-5-04-011459-2	4500	
Детектив	ОР 12	6326886	Андрей Смелый	Тайный шифр художника	Луганская типография имени Дзюбы	30.01.2018	360	978-5-04-092459-	1200	

Рис. 3.12. Результат добавления новой записи

Продолжим тестирование функциональности каталога. Протестируем удаление из каталога. Чтобы удалить запись из каталога, нужно нажать на соответствующую кнопку напротив записи. После нажатия всплывающее сообщение проинформирует нас о том что запись удалена, но в течении нескольких секунд мы можем отменить удаление нажав на соответствующую ссылку в появившемся сообщении. Данные действия приведены на рис. 3.13.

Каталог книг											+ Добавить книгу	Удалить отмеченные	Поиск
Категория	Марка автора	Инвентарный номер	Автор	Название	Место публикации	Дата	Страниц	EU-номер	Руб/Коп	Действия			
<input type="radio"/> Современная русская литература	СС 42	1234567	Сергей Сероусов	Книга для проверки	Белгородский национальный исследовательский университет	08.06.2018	240	999-8-77-654321-0	6000	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Детектив	ОР 44	6326221	Алексей Смелый	Бойтесь данайцев, дары приносящих	Белгородский учебная типография	24.08.2007	422	578-5-04-014859-2	3900	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Детектив	ЕГ 42	2325334	Алина Чечевица	Затаившийся страх	Белгородская национальная библиотека	11.02.2017	480	543-1-389-14762-6	25017	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Современная русская литература	ДА 44	6326886	Олег Рой	Метро 2033{	Луганская типография имени Дзюбы	28.12.2014	700	578-5-04-011459-2	4500	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Детектив	ОР 44	6326886	Андрей Смелый	Тайный шифр художника	Луганская типография имени Дзюбы	30.01.2018	360	978-5-04-092459-2	1200	<input type="checkbox"/> <input type="checkbox"/>			

Ожидание загрузки расширения AdGuard Антибаннер... Запись удалена. [Нажмите что бы отменить удаление.](#)

Рис. 3.13. Удаление записи

Также мы можем удалять несколько записей сразу. Стоит выбрать только нужные нам записи. Результат выбора представлен на рис. 3.14.

Каталог книг											+ Добавить книгу	Удалить отмеченные	Поиск
Категория	Марка автора	Инвентарный номер	Автор	Название	Место публикации	Дата	Страниц	EU-номер	Руб/Коп	Действия			
<input checked="" type="radio"/> Детектив	СВ 4в2	2325334	Алина Чечевица	Для проверки 2	Место публикации	22.01.1994	75	543-1-389-14762-6	25099	<input type="checkbox"/> <input type="checkbox"/>			
<input checked="" type="radio"/> Современная русская литература	С 42	6326111	Сергей Сероусов	Для проверки	Место публикации	22.01.1994	75	578-5-04-011459-2	5599	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Детектив	ОР 44	6326221	Алексей Смелый	Бойтесь данайцев, дары приносящих	Белгородский учебная типография	24.08.2007	422	578-5-04-014859-2	6000	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Детектив	ЕГ 42	2325334	Алина Чечевица	Затаившийся страх	Белгородская национальная библиотека	11.02.2017	480	543-1-389-14762-6	25017	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Современная русская литература	ДА 44	6326886	Олег Рой	Метро 2033{	Луганская типография имени Дзюбы	28.12.2014	700	578-5-04-011459-2	4500	<input type="checkbox"/> <input type="checkbox"/>			
<input type="radio"/> Детектив	ОР 12	6326886	Андрей Смелый	Тайный шифр художника	Луганская типография имени Дзюбы	30.01.2018	360	978-5-04-092459-2	1200	<input type="checkbox"/> <input type="checkbox"/>			

Рис. 3.14. Выбор записей для удаления

После выбора записей нужно нажать на кнопку «Удалить отмеченные», стоит учесть, что пока не выбрана хотя бы одна запись, кнопка будет некликабельна. После нажатия, выбранные записи удалятся. Результат удаления нескольких записей приведет на рис. 3.15.

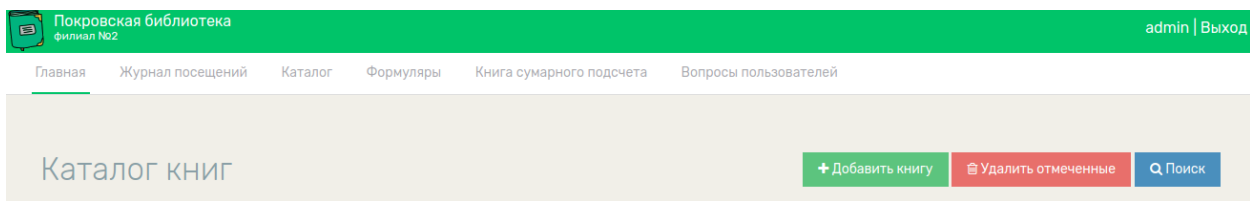


Рис. 3.15. Результат удаления нескольких записей

Следующей функцией тестирования будет изменение данных. Для того чтобы изменить данные записи, нужно нажать на рядом расположенную кнопку. После нажатия на кнопку появится окно, в котором можно редактировать запись. Результат действий приведет на рис. 3.16

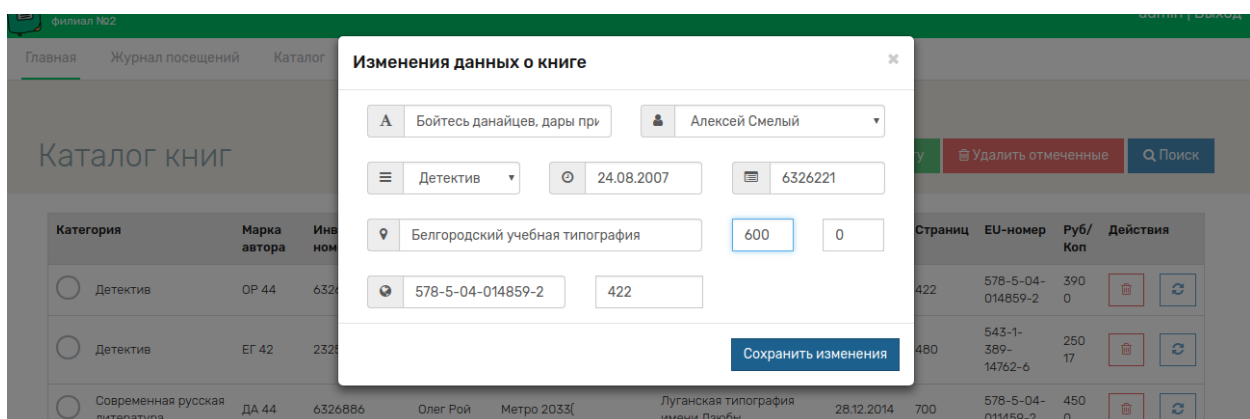


Рис. 3.16. Окно редактирования записи

После редактирования нужно нажать на «Сохранить изменения». В данном случае мы изменяем цену книги с 390 на 600. Результат редактирования данных представлен на рис. 3.17.

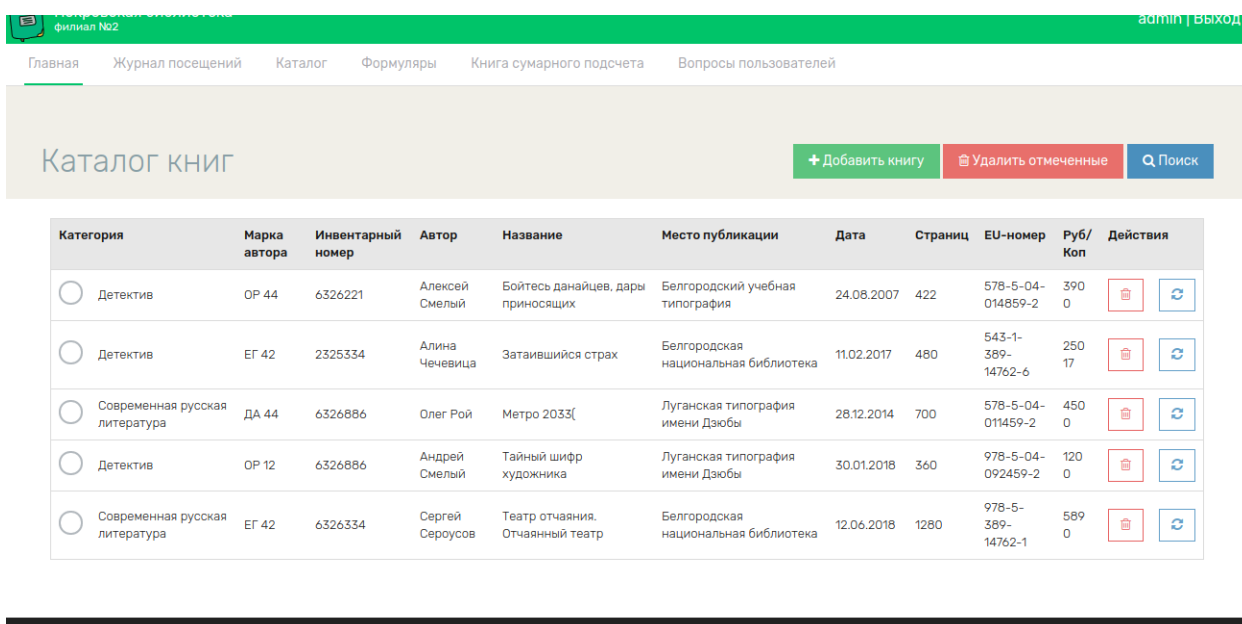


Рис. 3.17. Результат редактирования записи

Далее перейдем к тестированию поиска по каталогу. Чтобы открыть окно поиска нужно нажать на кнопку «Поиск». В появившемся окне выбрать поле, по которому нужно производить поиск. Всего доступно два варианта поиска, по автору и названию. При поиске по автору выводятся все книги автора, которого вы ищите. Окно поиска по автору приведено на рис. 3.18.

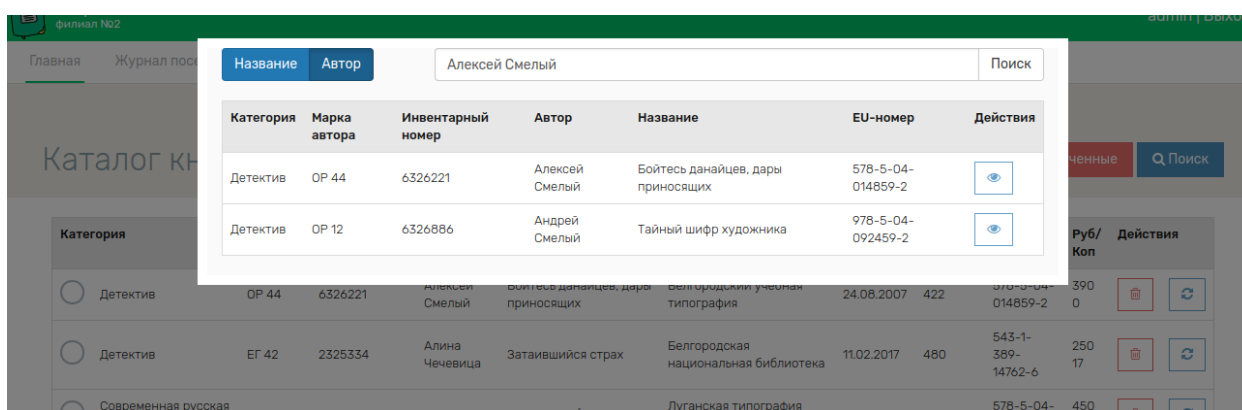


Рис. 3.18. Поиск по автору

При поиске по названию выводятся все совпадения, введенного вами текста с названиями книг в каталоге. Чтобы перейти к записи, нужно нажать

на соответствующую кнопку расположенную рядом. После чего данная запись ненадолго подсветится в основной таблице. На рис. 3.19 представлен поиск по названию.

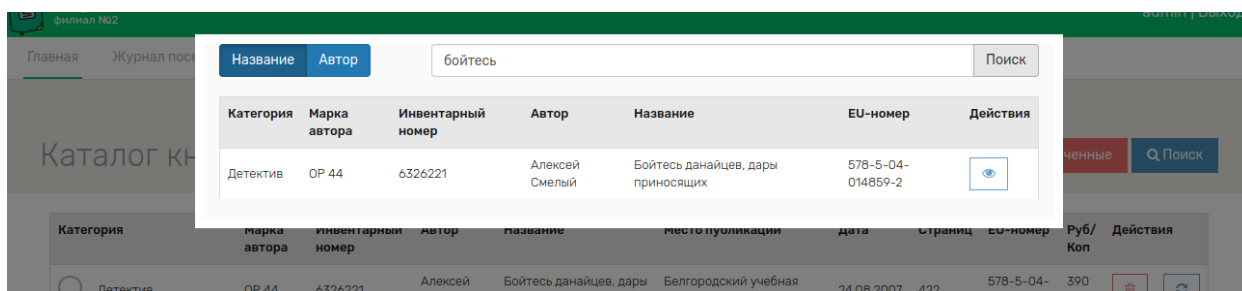


Рис. 3.19. Поиск по названию

Также протестируем поиск по каталогу, который находится на главной странице. Чтобы узнать, есть ли нужная вам книга в каталоге, нужно ввести ее название. Результат поиска приведен на рис. 3.20.

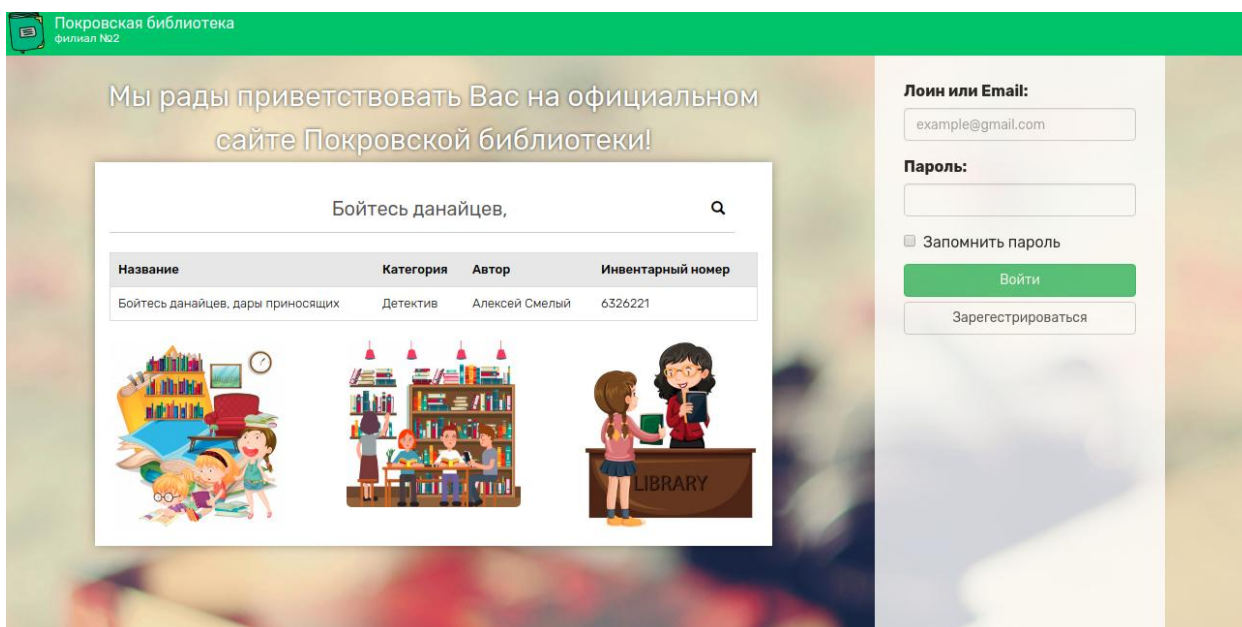
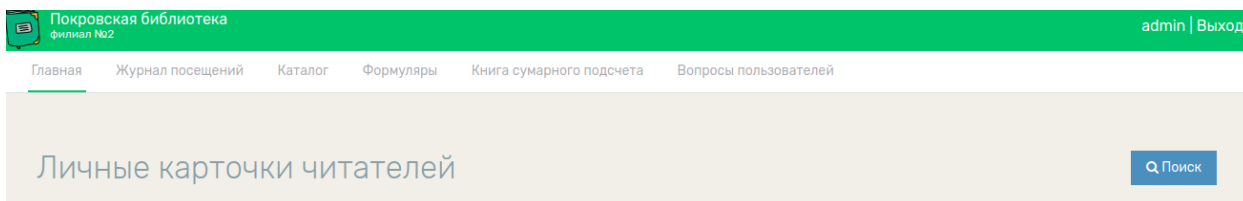


Рис. 3.20. Результат поиска на главной странице

Перейдем к тестированию раздела с личными данными пользователей. На рис. 3.21 Представлен вывод всех пользователей.



Логин	Почта	Имя	Фамилия	Отчество	Дата рождения	Профессия	Место учебы	Улица / Дом	Серия номер паспорта	Номер телефона	Место работы	Действия	
	username	username@mail.ru	Виктор	Смелый	Александрович	22.12.1996	нет	Луганский университет имени Тараса Шевченка	ул.Набережная, дом 12	EN 903112	961166861	не работаю	
	adam	kama@mail.ru	Камиль	Пуля	Пуляевич	22.12.1893	Гангстер	БелГУ имени Тараса Шевченка	Крупская,99	EN 444333	961166861	не работаю	
	admin	adam.carraway@mail.ru	Сергей	Сероусов	Владимирович	2017-07-03	программист	БелГУ	Крупской,69	EN 904304	961166861	не работаю	

Рис. 3.21. Вывод всех пользователей

Протестируем поиск пользователей. Чтобы появилось окно поиска, нужно нажать на кнопку «Поиск». В появившемся окне в поле поиска нужно ввести имя, фамилию или отчество пользователя, после чего ниже появятся все совпадения. Результат поиска представлен на рис. 3.22.

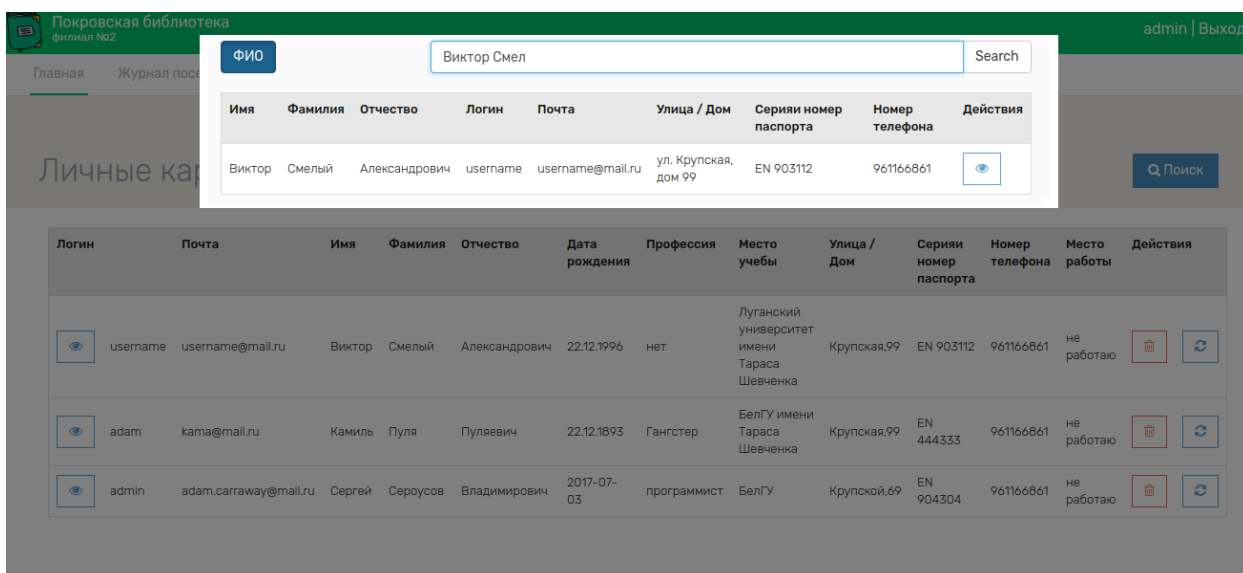


Рис. 3.22. Результат поиска пользователя

Далее протестируем возможность изменения данных пользователя. Чтобы изменить какие-либо данные нужно нажать на соответствующую

кнопку напротив пользователя. После нажатия появится окно, изображенное на рис. 3.23 .

The screenshot shows a web application interface for editing a user profile. The form is titled "Смелый Виктор" and contains the following fields:

- Логин:** username
- Электронная почта:** username@mail.ru
- Серия паспорта:** EN
- Номер паспорта:** 903112
- Имя:** Виктор
- Фамилия:** Смелый
- Отчество:** Александрович
- Дата рождения:** 22.12.1996
- Адрес проживания:** ул.Набережная, дом 12
- Образование:** Высшее
- Место обучения:** Луганский университет имени Тараса Шевченка
- Номер телефона:** 961166861
- Профессия:** нет

At the bottom right of the form is a blue button labeled "Подтвердить изменения".

Рис. 3.23. Форма изменения данных

Чтобы изменить данные нужно отредактировать необходимое нам поле и нажать на кнопку «Подтвердить изменения». Об успешном результате проинформирует всплывающее сообщение. Результат изменения данных приведен на рис. 3.24 .

Личные карточки читателей

Логин	Почта	Имя	Фамилия	Отчество	Дата рождения	Профессия	Место учебы	Улица / Дом	Серия номер паспорта	Номер телефона	Место работы	Действия
username	username@mail.ru	Виктор	Смелый	Александрович	22.12.1996	нет	Луганский университет имени Тараса Шевченка	ул.Набережная, дом 12	EN 903112	961166861	не работаю	
adam	kama@mail.ru	Камиль	Пуля	Пуляевич	22.12.1893	Гангстер	БелГУ имени Тараса Шевченка	Крупская,99	EN 444333	961166861	не работаю	
admin	adam.carraway@mail.ru	Сергей	Сероусов	Владимирович	2017-07-03	программист	БелГУ	Крупской,69	EN 904304	961166861	не работаю	

Рис. 3.24. Результат изменения данных

Перейдем к тестированию системы поддержки пользователей. Протестируем возможность задать вопрос простым пользователем и возможность ответить на вопрос администратором.

Для того что бы задать вопрос нужно нажать на кнопку «Задать вопрос». В появившемся окне отображается поле для ввода вопроса и история вопросов пользователя. Всплывающее окно показано на рис. 2.25.

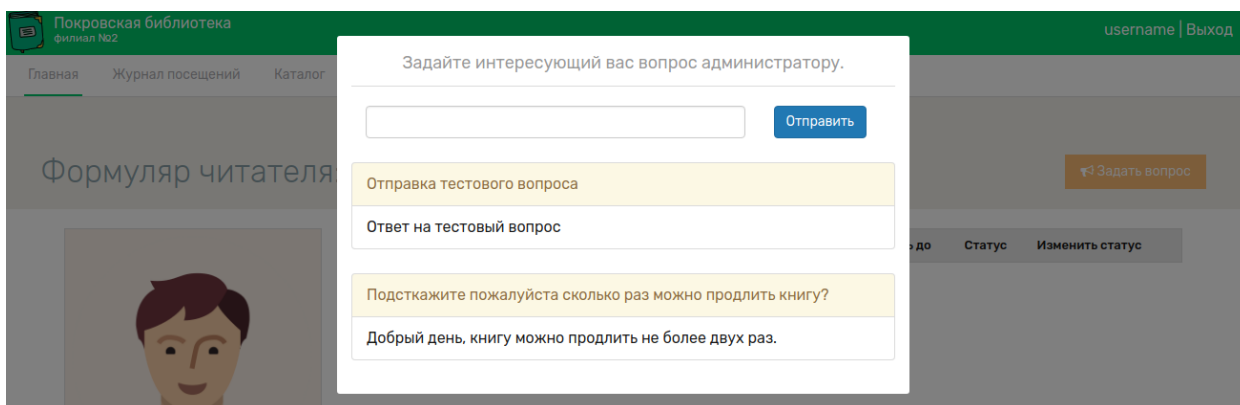


Рис. 2.25. Окно отправки вопроса

Что бы ответить на вопрос нужно перейти в раздел «Вопросы пользователей» и напротив вопроса нажать на соответствующую кнопку. В появившееся поле нужно ввести ответ на вопрос и подтвердить отправку. Данные действия приведены на рис. 2.26.

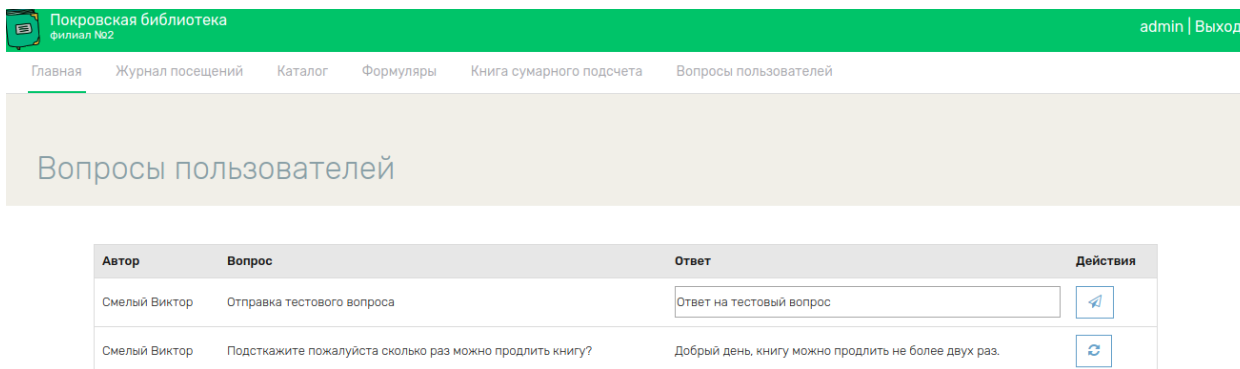


Рис. 2.26. Отправка ответа на вопрос

Что бы изменить ответ на вопрос нужно нажать на соответствующую кнопку напротив вопроса, у которого уже есть ответ. В появившемся поле отредактировать ответ и подтвердить изменения. Данные действия приведены на рис. 2.27.

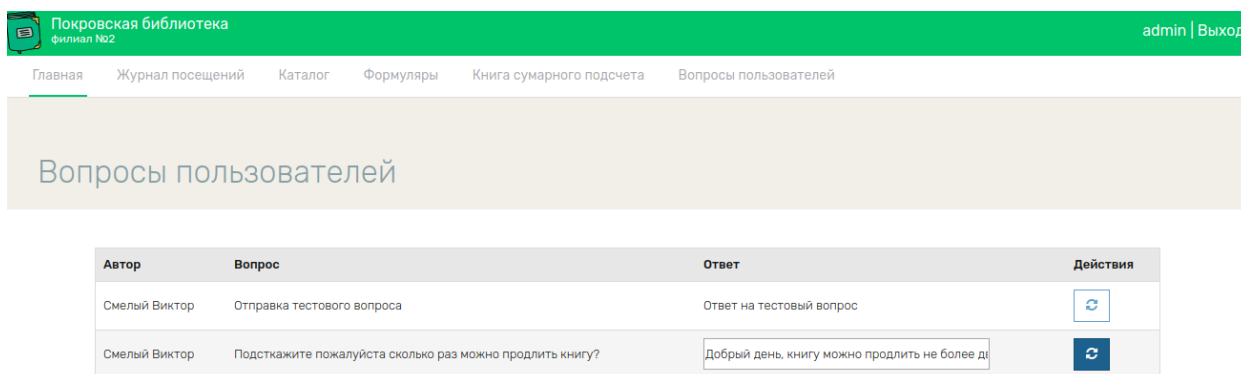


Рис. 2.27. Изменение ответа на вопрос

Подводя итог, по результатам проведения испытаний разработанной автоматизированной информационной системы было установлено, что система полностью работоспособна и готова к введению в эксплуатацию.

Функциональные возможности разработанной системы полностью удовлетворяют по всем пунктам требований к системе, поставленным в начале ее проектирования.

Можно с уверенностью сказать, что данная автоматизированная система в полной мере упростит ряд аспектов работы библиотеки.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была спроектирована и разработана автоматизированная система учета библиотечного фонда Покровской библиотеки. В результате чего были решены следующие задачи:

1. Анализ деятельности организации.
2. Выявление основных направлений деятельности организации, требующих автоматизации.
3. Обзор и анализ существующих систем, выполняющих схожие задачи.
4. Постановка требований к разрабатываемой автоматизированной системе.
5. Проектирование автоматизированной системы.
6. Реализация автоматизированной системы.
7. Проведение испытаний.

. Данная система реализована в виде веб-приложение, которое упрощает ряд аспектов работы библиотеки. Сотрудникам библиотеки предоставлен удобный и понятный интерфейс для ведения учета библиотечного фонда, поддержки и удаленной работы с читателями и их данными. Читателям доступна удаленная регистрация в библиотеке и возможность поиска и заказа книг. Данная автоматизированная система будет передана для внедрения Покровской библиотеке филиал №2.

По итогам тестирования разработанной системы учета, можно сделать следующий вывод – веб приложение полностью удовлетворяет всем требованиям, предъявленным к системе на этапе постановки задачи.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Архитектурный паттерн MVC // Около программирования - тьюториалы и статьи по веб-программированию. - 2011 [Электронный ресурс]. – Режим доступа: <http://artanovy.com/2011/03/arhitekturnyj-pattern-mvc/> 09.03.2016.
2. Документация Bootstrap [Электронный ресурс]. – Режим доступа: <http://getbootstrap.ru/docs/> , 21.04.2018
3. Документация Codeigniter [Электронный ресурс]. – Режим доступа: <https://codeigniter.com/manual>, 21.04.2018
4. Дунаев, Вадим HTML, скрипты и стили / Вадим Дунаев. - М.: БХВ-Петербург, 2015. - 816 с.
5. Дунаев, Вадим Сценарии для Web-сайта. PHP и JavaScript / Вадим Дунаев. - М.: БХВ-Петербург, 2008. - 804 с.
6. Колисниченко, Д. Н. PHP 7 и MySQL 6. Разработка Web-приложений / Д.Н. Колисниченко. - М.: БХВ-Петербург, 2011. - 528
7. Макаров, Александр Codegniter. Сборник рецептов / Александр Макаров. - М.: ДМК Пресс, 2013. - 372 с.
8. Флэнаган Д. JavaScript. Подробное руководство. – Пер. с англ. – СПб: Символ Плюс, 2008. – 992 с.
9. Шаблон проектирования «Модель Представление Контроллер» [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Model-View-Controller>, 07.03.2018.
10. Этапы проектирования базы данных [Электронный ресурс]. – Режим доступа: <https://studfiles.net/preview/719825/page/2/>, 07.03.2018.

ПРИЛОЖЕНИЕ

Исходный код файла User.php

```
defined('BASEPATH') OR exit('No direct script access allowed');

class User extends CI_Controller {

public function __construct() {

parent::__construct();

$this->load->library(array('session'));

$this->load->helper(array('url'));

$this->load->model('user_model');

}

public function index() {

}

public function register() {

$data = new stdClass();

$this->load->helper('form');

$this->load->library('form_validation');

$this->form_validation->set_rules('username', 'Username',
'trim|required|alpha_numeric|min_length[4]|is_unique[users.username]',
array('is_unique' => 'This username already exists. Please choose another one.));

$this->form_validation->set_rules('email', 'Email',
'trim|required|valid_email|is_unique[users.email]');

$this->form_validation->set_rules('password', 'Password',
'trim|required|min_length[6]');

$this->form_validation->set_rules('password_confirm', 'Confirm Password',
'trim|required|min_length[6]|matches[password]');
```

```

$this->form_validation->set_rules('name', 'name', 'trim|required');
$this->form_validation->set_rules('surname', 'surname', 'trim|required');
$this->form_validation->set_rules('patronymic', 'patronymic', 'trim|required');
$this->form_validation->set_rules('data_birth', 'data_birth', 'trim|required');
$this->form_validation->set_rules('education', 'education', 'trim|required');
$this->form_validation->set_rules('profession', 'profession', 'trim|required');
$this->form_validation->set_rules('loc_educat', 'loc_educat', 'trim|required');
$this->form_validation->set_rules('street', 'street', 'trim|required');
$this->form_validation->set_rules('house', 'house', 'trim|required');
$this->form_validation->set_rules('series_pas', 'series_pas', 'trim|required');
$this->form_validation->set_rules('num_pas', 'num_pas', 'trim|required');
$this->form_validation->set_rules('num_tel', 'num_tel', 'trim|required');
if ($this->form_validation->run() === false) {
    $this->load->view('header');
    $this->load->view('user/register/register', $data);
    $this->load->view('footer');
} else {
    $username = $this->input->post('username');
    $email = $this->input->post('email');
    $password = $this->input->post('password');
    $redirect = $this->input->post('redirect');
    $name = $this->input->post('name');
    $surname = $this->input->post('surname');
    $patronymic = $this->input->post('patronymic');

```



```

$data_birth = $this->input->post('data_birth');
$education = $this->input->post('education');
$profession = $this->input->post('profession');
$loc_educat = $this->input->post('loc_educat');
$street = $this->input->post('street');
$house = $this->input->post('house');
$series_pas = $this->input->post('series_pas');
$num_pas = $this->input->post('num_pas');
$num_tel = $this->input->post('num_tel');
if ($this->user_model->create_user($username, $email,
$password,$name,$surname,$patronymic,$data_birth,$education
$education, $profession, $loc_educat, $street, $house, $series_pas, $num_pas,
$num_tel)) {
    $user_id = $this->user_model->get_user_id_from_username($username);
    $user = $this->user_model->get_user($user_id);
    $_SESSION['user_id'] = (int)$user->id;
    $_SESSION['username'] = (string)$user->username;
    $_SESSION['logged_in'] = (bool>true;
    $_SESSION['is_confirmed'] = (bool)$user->is_confirmed;
    $_SESSION['is_admin'] = (bool)$user->is_admin;
    redirect($redirect);
} else {
    $data->error = 'There was a problem creating your new account. Please try again.';
    $this->load->view('header');
    $this->load->view('user/register/register', $data);

```

```

$this->load->view('footer');

}

}

}

public function login() {
    $data = new stdClass();

    $this->load->helper('form');

    $this->load->library('form_validation');

    $this->form_validation->set_rules('username', 'Username',
    'required|alpha_numeric');

    $this->form_validation->set_rules('password', 'Password', 'required');

    if ($this->form_validation->run() == false) {

        $this->load->view('header');

        $this->load->view('user/login/login');

        $this->load->view('footer');

    } else {

        $username = $this->input->post('username');

        $password = $this->input->post('password');

        $redirect = $this->input->post('redirect');

        if ($this->user_model->resolve_user_login($username, $password)) {

            $user_id = $this->user_model->get_user_id_from_username($username);

            $user = $this->user_model->get_user($user_id);

            $_SESSION['user_id'] = (int)$user->id;

            $_SESSION['name'] = (string)$user->name;

            $_SESSION['surname'] = (string)$user->surname;

```

```

$_SESSION['username'] = (string)$user->username;
$_SESSION['email'] = (string)$user->email;
$_SESSION['avatar'] = (string)$user->avatar;
$_SESSION['logged_in'] = (bool>true;
$_SESSION['is_confirmed'] = (bool)$user->is_confirmed;
$_SESSION['is_admin'] = (bool)$user->is_admin;
redirect($redirect);
} else {
$data->error = 'Wrong username or password.';
$this->load->view('header');
$this->load->view('user/login/login', $data);
$this->load->view('footer');
}
}
}

public function logout() {
$data = new stdClass();
if (isset($_SESSION['logged_in']) && $_SESSION['logged_in'] === true) {
foreach ($_SESSION as $key => $value) {
unset($_SESSION[$key]);
}
redirect('/');
} else {
redirect('/');
}
}

```

Исходный код файла Catalog.js

```
<script>

    $("#add ").click(function() {
var title = $("#title").val();
var author = $("#author").val();
    var category = $("#category").val();
var data = $("#data").val();
var inventnum = $("#inventnum").val();
var map = $("#map").val();
var rub = $("#rub").val();
var kop = $("#kop").val();
var eunum = $("#eunum").val();
var pages = $("#pages").val();
$.ajax({
url: "<?php echo base_url(); ?>" + "index.php/admin/add_catalog",
type: "POST",
data: {
"title": title,
"author": author,
"category": category,
"data": data,
"inventnum": inventnum,
"map": map,
"rub": rub,
```

```

"kop": kop,
"eunum": eunum,
"pages": pages
},
cache: false,
success: function (response) {
$("#info").append(response);
setTimeout ('obn()', 6500);
window.location.reload();
}
});
});
function inf(obj) {
var chbox = document.getElementById(obj.id);
var checkId = obj.id;
var id='pole'+checkId.substr(8);
if (chbox.checked) {
document.getElementById(id).style.background = 'rgba(200, 230, 201, 0.66)';
}else {
document.getElementById(id).style.background = '#fff';
}
}
function obn() {
$('.err').slideUp(300, function() {

```

```

$(this).remove();

});

$('.on').slideUp(300, function() {

$(this).remove();

});

};

function del(obj) {

var delId = obj.id.substr(6);

var poleId='pole'+delId.substr(5);

$.ajax({

url: "<?php echo base_url(); ?>" + "index.php/admin/del_catalog",

type: "POST",

data: {"delId": delId},

cache: false,

success: function (response) {

("#info").append(response);

setTimeout ('obn()', 6500);

window.location.reload();

}

});

}

```

Исходный код файла Catalog.php

```
<div class="panel-heading">
  <div class="row">
    <div class="col col-md-3">
      <h3 class="panel-title">Каталог книг </h3>
    </div>
    <div class="col-md-4 text-center" >
      <div class="input-group">
<!--      <span class="input-group-addon" id="basic-addon1">--><?//echo $this-
>db->count_all('catalog').' книг в каталоге';?><!--</span>-->
      </div>
    </div>
    <div class="col col-md-5 text-right">
      <button type="button" class="btn btn-success" data-toggle="modal" data-
target="#myModal_add" ><i class="fa fa-plus" aria-hidden="true"></i> Добавить
книгу</button>
      <button type="button" class="btn btn-danger"><i class="fa fa-trash-o" aria-
hidden="true"></i> Удалить отмеченные</button>
      <button type="button" class="btn btn-primary"><i class="fa fa-search" aria-
hidden="true"></i> Поиск</button>
    </div>
  </div>
</div>
<div class="table-cont">
<table class="table table-hover ">
  <tr id="head-table">
    <th colspan="2">Категория</th>
    <th>Марка автора</th>
```

```

<th>Инвентарный номер</th>
<th>Автор</th>
<th>Название</th>
<th>Место публикации</th>
<th>Дата</th>
<th>Страниц</th>
<th>EU-номер</th>
<th>Руб/Коп</th>
<th colspan="2">Действия</th>
</tr>
<?foreach ($query->result() as $row) {
    $query = $this->db->query('SELECT * FROM author WHERE id="'. $row->author.'" LIMIT 1');
    $author = $query->row_array();
    $query2 = $this->db->query('SELECT * FROM category WHERE id="'. $row->category.'" LIMIT 1');
    $category = $query2->row_array();
    ?>
<tr id='pole<? echo $row->id?>' >
    <td>
        <label class="gtr">
            <input type="checkbox" onclick="inf(this)" id="checkbox<?echo $row->id;?>" name="" value="" class="rtg" >
            <span class="mrt"></span>
            <span class="lst"></span>
        </label>
    </td>
    <td > <? echo $category['name']. ' '. $row->id;?></td>
}

```



```

<td> <? echo $row->author_mark;?></td>
<td> <? echo $row->invent_num;?></td>
<td> <? echo $author['name']. ' '.$author['surname'];?> </td>
<td> <? echo $row->title;?></td>
<td> <? echo $row->loc_puplic;?></td>
<td> <? echo $row->data;?></td>
<td> <? echo $row->pages;?></td>
<td> <? echo $row->eu_number;?> </td>
<td> <? echo $row->rub." ".$row->cop;?></td>
<td>
    <a class="btn btn-danger" id="delete<? echo $row->id?>" onclick="del(this)"
aria-label="Удалить">
        <i class="fa fa-trash-o" aria-hidden="true"></i>
    </a>
</td>
<td>
    <a class="btn btn-primary" id="<? echo $row->id?>" onclick="" aria-
label="Изменить">
        <i class="fa fa-refresh" aria-hidden="true"></i>
    </a>
</td>
</tr>
<?}?>
</table>
<div class='pager pull-right'><? echo $pager;?></div>
<div class="clear"></div>
</div>
<?

```

```

$category=$this->db->get('category');
$author=$this->db->get('author');
?>
<!-- Модаль -->
<div class="modal fade" id="myModal_add" tabindex="-1" role="dialog" aria-
labelledby="myModalLabel">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
label="Close"><span aria-hidden="true">&times;</span></button>
        <h4 class="modal-title" id="myModalLabel">Добавить книгу в
каталог</h4>
      </div>
      <div class="modal-body">
        <div class="row">
          <div class="col-md-6">
            <div class="input-group">
              <span class="input-group-addon" id="basic-addon1"><i class="fa fa-
font" aria-hidden="true"></i></span>
              <input type="text" id='title' name="title" class="form-control"
placeholder="Название" aria-describedby="basic-addon1">
            </div>
          </div>
          <div class="col-md-6">
            <div class="input-group">
              <span class="input-group-addon" id="basic-addon1"><i class="fa fa-
user" aria-hidden="true"></i></span>

```

```

<select name="author" id="author" class="form-control" >
  <option value="0">Автор</option>
  <?foreach ($author->result() as $aut) {?>
    <option value="<?echo $aut->id;? "><?echo $aut->name.' '.$aut-
>surname;?></option>
    <?}?>
  </select>
</div>
</div>
</div>
<div class="row mr-t25">
  <div class="col-md-4">
    <div class="input-group">
      <span class="input-group-addon" id="basic-addon1"><i class="fa fa-
bars" aria-hidden="true"></i></span>
      <select id="category" name="category" class="form-control">
        <option value="0">Категория</option>
        <?foreach ($category->result() as $cat) {?>
          <option value="<?echo $cat->id;? "><?echo $cat->name;?></option>
          <?}?>
        </select>
      </div>
    </div>
    <div class="col-md-4">
      <div class="input-group">
        <span class="input-group-addon" id="basic-addon1"><i class="fa fa-
clock-o" aria-hidden="true"></i></span>

```

```

        <input type="text" id="data" name="data" class="form-control
datepicker-here" placeholder="Дата" aria-describedby="basic-addon1">
    </div>
</div>
<div class="col-md-4">
    <div class="input-group">
        <span class="input-group-addon" id="basic-addon1"><i class="fa fa-
list-alt" aria-hidden="true"></i></span>
        <input type="text" id="inventnum" name="inventnum" class="form-
control" placeholder="Инвентарный номер" aria-describedby="basic-addon1">
    </div>
</div>
</div>
<div class="row mr-t25">
    <div class="col-md-8">
        <div class="input-group">
            <span class="input-group-addon" id="basic-addon1"><i class="fa fa-
map-marker " aria-hidden="true"></i></span>
            <input type="text" id="map" name="map" class="form-control"
placeholder="Место публикации" aria-describedby="basic-addon1">
        </div>
    </div>
</div>
<div class="col-md-2">
    <div class="input-group">
        <input type="text" id="rub" name="rub" class="form-control"
placeholder="Руб." aria-describedby="basic-addon1">
    </div>
</div>

```

```

<div class="col-md-2">
  <div class="input-group">
    <input type="text" id="kop" name="kop" class="form-control"
placeholder="Коп." aria-describedby="basic-addon1">
  </div>
</div>
  <span class="input-group-addon" id="basic-addon1"><i class="fa fa-
globe " aria-hidden="true"></i></span>
  <input type="text" id="eunum" name="eunum" class="form-control"
placeholder="Европейский номер" aria-describedby="basic-addon1">
</div>
</div>
<div class="col-md-3">
  <div class="input-group">
    <input type="text" id="pages" name="pages" class="form-control"
placeholder="Страниц" aria-describedby="basic-addon1">
  </div>
</div>
<div class="col-md-4">
</div>
</div>
<div class="modal-footer">
  <button type="button" id='add' class="btn btn-primary">Save
changes</button></div> </div> </div></div></div>

```