|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** |
| **Институт ИТ**  **Кафедра Прикладной математики** |
| **Курсовая работа**  **По курсу «Системы управления данными»**  **Тема работы: «***Разработка программного продукта, работающего в среде СУБД по предметной области: электронный магазин книг***»** |
| |  |  | | --- | --- | | Выполнили:  Студент группы | Аникин Артём Сергеевич  ИНБО-05-19 | | Научный руководитель | к.т.н, доцент Скляр А. Я. |   Москва 2022г. |



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**МИРЭА – Российский технологический университет**

**Институт** информационных технологий

**Кафедра** Прикладной математики

|  |  |  |  |
| --- | --- | --- | --- |
| **ЗАДАНИЕ** | | | |
| **на выполнение курсовой работы** | | | |
| **по** **дисциплине** **«Системы управления данными»** | | | |
|  | | | |
| Студент:Аникин Артём Сергеевич  Группа ИНБО-05-19 | | | |
| 1. **Тема**: **«***Разработка программного продукта, работающего в среде СУБД по предметной области: электронный магазин книг***»** | | | |
| 1. **Исходные данные:** | | | |
| Документация для языка программирования Python и среды разработки Django | | | |
|  | | | |
|  | | | |
| 1. **Перечень вопросов, подлежащих разработке, и обязательного графического материала:** | | | |
| Анализ предметной области | | | |
| Проектировании системы управления данными | | | |
| Описание информационной системы, процессы и этапа ее разработки | | | |
| 1. **Срок представления к защите курсовой работы:** **до** «\_\_» \_\_\_\_ 20\_\_ г. | | | |
|  | | | |
| Задание на курсовой проект, (работу) выдал | «\_\_\_»\_\_\_\_\_\_202\_\_г. | *Подпись руководителя проекта* |  |
| Задание на курсовой проект, (работу) получили | «\_\_\_»\_\_\_\_\_\_202\_\_г. | *Подписи студентов – исполнителей проекта* |

# Содержание

[Введение 6](#_Toc105544577)

[1 раздел 7](#_Toc105544578)

[1.1 Анализ предметной области 7](#_Toc105544579)

[1.2 Выводы 8](#_Toc105544580)

[2 раздел 9](#_Toc105544581)

[2.1 Выбор модели жизненного цикла 9](#_Toc105544582)

[2.2 Логическая схема базы данных 10](#_Toc105544583)

[2.3 Выводы 11](#_Toc105544584)

[3 раздел 12](#_Toc105544585)

[3.1 Выбор программных средств 12](#_Toc105544586)

[3.2 Физическая модель базы данных 12](#_Toc105544587)

[3.3 Программная реализация информационной системы 14](#_Toc105544588)

[3.4 Пример работы с сайтом 15](#_Toc105544589)

[3.4.1 Пример создания экземпляра модели «Книга» 15](#_Toc105544590)

[3.4.2 Пример редактирования экземпляра модели «Книга» 17](#_Toc105544591)

[3.4.3 Пример удаления экземпляра модели «Книга» 18](#_Toc105544592)

[3.4.4 Пример добавления книги в избранное 18](#_Toc105544593)

[3.4.5 Пример покупки книги 18](#_Toc105544594)

[3.5 Выводы 20](#_Toc105544595)

[Заключение 21](#_Toc105544596)

[Список используемых источников 22](#_Toc105544597)

[Приложения 23](#_Toc105544598)

[Приложение А 24](#_Toc105544599)

[Приложение Б 26](#_Toc105544600)

# Введение

В настоящее время для различных нужд разработано огромное количество инструментария для работы с данными. В рамках данного курсового проекта рассматривается пример создания интернет магазина книг с помощью одного из инструментов для работы с данными [1].

Целью курсового проектирования является получение практических навыков проектирования систем управления данными для управления производством и анализа экономической деятельности.

Курсовая работа разбита на три раздела, соответствующих различным задачам:

* раздел 1 содержит анализ предметной области, описывает процессы этапа исследования;
* раздел 2 содержит информацию о проектировании системы управления данными, описывает процессы этапа проектирования информационной системы;
* раздел 3 содержит описание информационной системы, процессы и этапа ее разработки.

# раздел

## Анализ предметной области

В настоящее время бумажные книги почти пропали из рук рядовых граждан. Их место заняли электронные аналоги произведений как текстовые их варианты, так и аудио. Как правило, электронные книги дешевле (это связано с отсутствием затрат на печать) и физически компактнее, что является, несомненно, плюсом [6].

Существует множество интернет магазинов и приложений для смартфонов, средствами которых происходит распространение литературных произведений различных жанров. Одним из наиболее известных сервисов можно назвать «Литрес». У каждого пользователя, зарегистрировавшегося в этом сервисе, есть своя электронная библиотека, наполняющаяся по мере покупки новых книг. Широкий ассортимент книг и большой список авторов позволяют охватить широкую аудиторию. Наряду с возможностью скачать купленные книги «в один клик» напрямую с сайта, существуют «избранные» книги и авторы, что упрощает навигацию. Отдельно следует упомянуть о возможности оставлять цитату из книги на ее странице в магазине. Текст цитаты будут видеть все пользователи, заходящие на страницу книги. Это является несомненным плюсом, ведь ничто так не сближает людей, как одна и та же любимая цитата.

Таким образом, хороший сервис состоит из мелочей, которые складываются в удобство использования. То есть можно выделить, что разрабатываемая система должна иметь минимальный порог вхождения. Должна сочетать только лучшее из решений конкурентов, устраняя при этом их недостатки.

## Выводы

На сегодняшний день электронные книги прочно засели в обществе, обретя свою нишу. Одним из основных критериев системы распространения таких книг является удобство использования и отсутствие порога вхождения пользователя.

# раздел

## Выбор модели жизненного цикла

Для разработки курсового проекта выбрана каскадная модель жизненного цикла системы. Она предусматривает последовательное выполнение всех этапов проекта в строго заданном порядке. Переход на следующий этап означает полное завершение работ на предыдущем этапе [5].

На рисунке 2.1 изображена примерная диаграмма потоков данных. При заходе на сайт пользователь проходит авторизацию и при необходимости регистрацию. Информация о пользователях берется из таблицы «Пользователи» соответственно. Далее он отправляется просматривать каталог, где может добавлять в избранное книги и авторов и покупать книги. Информация об этом записывается в соответствующие таблицы.

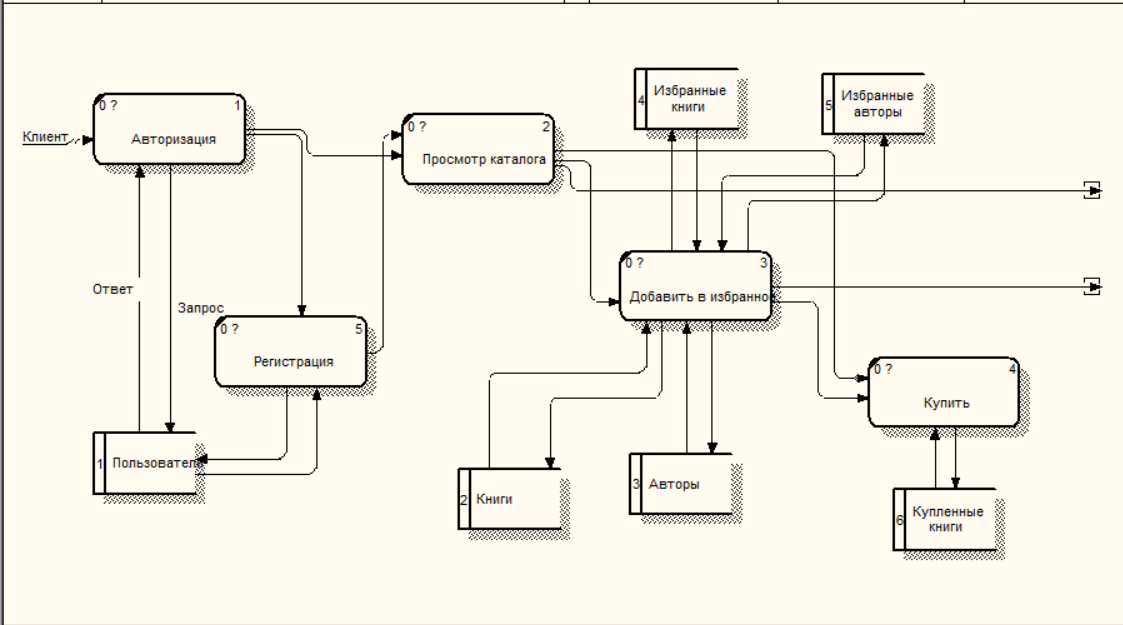


Рисунок . – Диаграмма потоков данных

## Логическая схема базы данных

На рисунке ниже представлена логическая схема базы данных проектируемой системы.

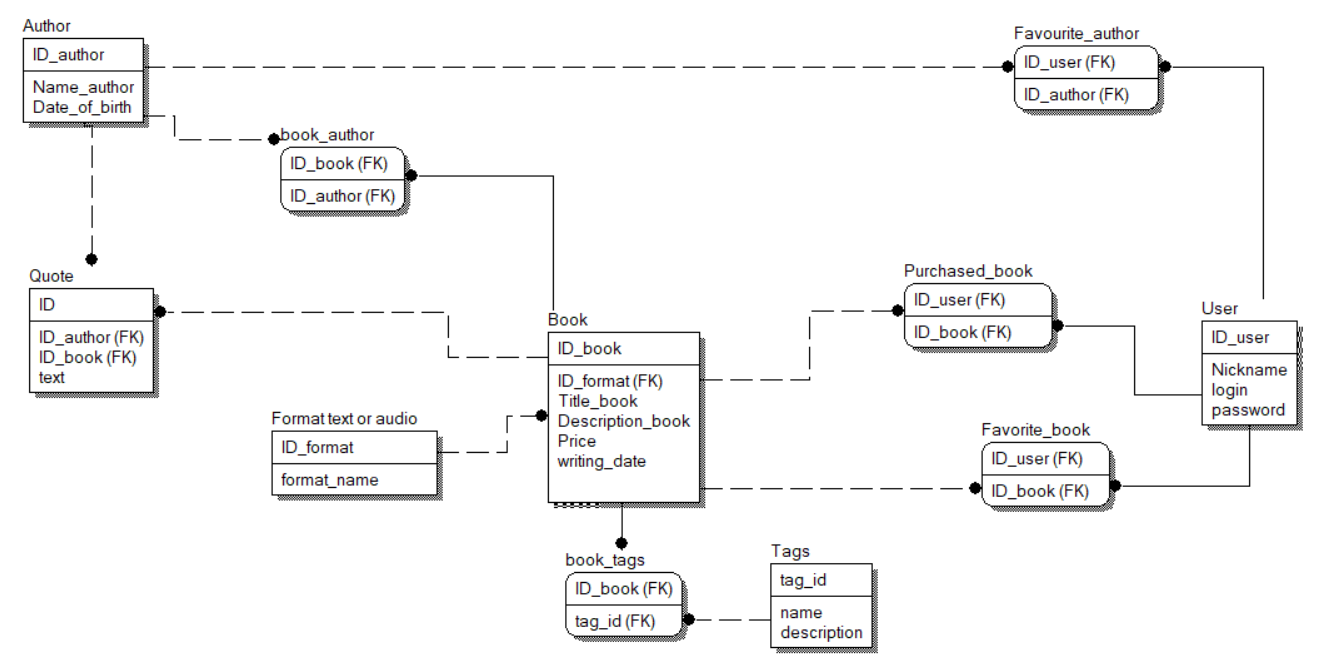


Рисунок . – Логическая схема базы данных

Можно выделить три основные сущности: книга, пользователь и автор.

У таблицы «Книга» помимо уникального идентификатора каждой записи (ID) есть также название книги, ее описание, цена и дата написания. Есть поле с форматом, где будет указано: это аудиокнига или текстовый формат.

Также есть тэги, определяющие принадлежность к жанру, настроению, историческому событию и тд. У каждого тэга есть ID, наименование и краткое описание.

У таблицы «Автор» есть ID, поле с именем или псевдонимом писателя, а также его дата рождения.

Таблица «Цитата» содержит ID каждой цитаты, автора цитаты и книгу, откуда она взята, и сам текст цитаты.

У таблицы «Пользователь» есть ID, никнейм, логин и пароль каждого пользователя.

Из логической схемы базы данных видно, что у пользователя есть личный список избранных книг («Избранные книги») и избранных авторов («Избранные авторы»). Также есть личная библиотека («Купленные книги»), которая формируется посредством приобретения книг пользователем. Каждая из этих таблиц содержит поле ID, а также два поля с внешними ключами. Один ключ принадлежит пользователю, а другой книге или автору соответственно [2].

У таблицы «Формат» есть ID и название формата (текстовый или аудио формат).

## Выводы

Таким образом, из схемы базы данных, приведенной на рисунке 2.2, видно, что разрабатываемый проект имеет базовый функционал, описанный в предыдущем разделе. То есть пользователь способен покупать книги, расширяя тем самым личную библиотеку, а также добавлять авторов и книги в избранное.

# раздел

## Выбор программных средств

Курсовой проект реализован средствами фреймворка Django с использованием языка программирования Python. Создание базы данных происходит с помощью ORM системы и записывается в sqlite файл.

Выбор данного инструмента обусловлен тем, что «из коробки» в Django уже есть механизм для выполнения миграций, удобная возможность настройки всего проекта и иерархия файлов, созданная автоматически.

## Физическая модель базы данных

На рисунке 3.1 представлена физическая модель базы данных. Таблицы сгенерированы средствами ORM Django. Здесь присутствуют все таблицы, описанные в логической схеме БД, кроме таблицы «Пользователь». Она создается автоматически и «из коробки» реализует механизмы авторизации и аутентификации. Готовый sqlite файл открыт с помощью программы DataGrip [3]. Исходный код скрипта создания базы данных представлен в приложении Б.

Фронтэнд часть сайта написана с помощью HTML, CSS и jQuery.

Реализован ряд триггеров для журналирования, представленный в приложении А. В результате их работы в таблицу записывается тип операции, дата, название таблицы и ID строки, над которой происходит операция.

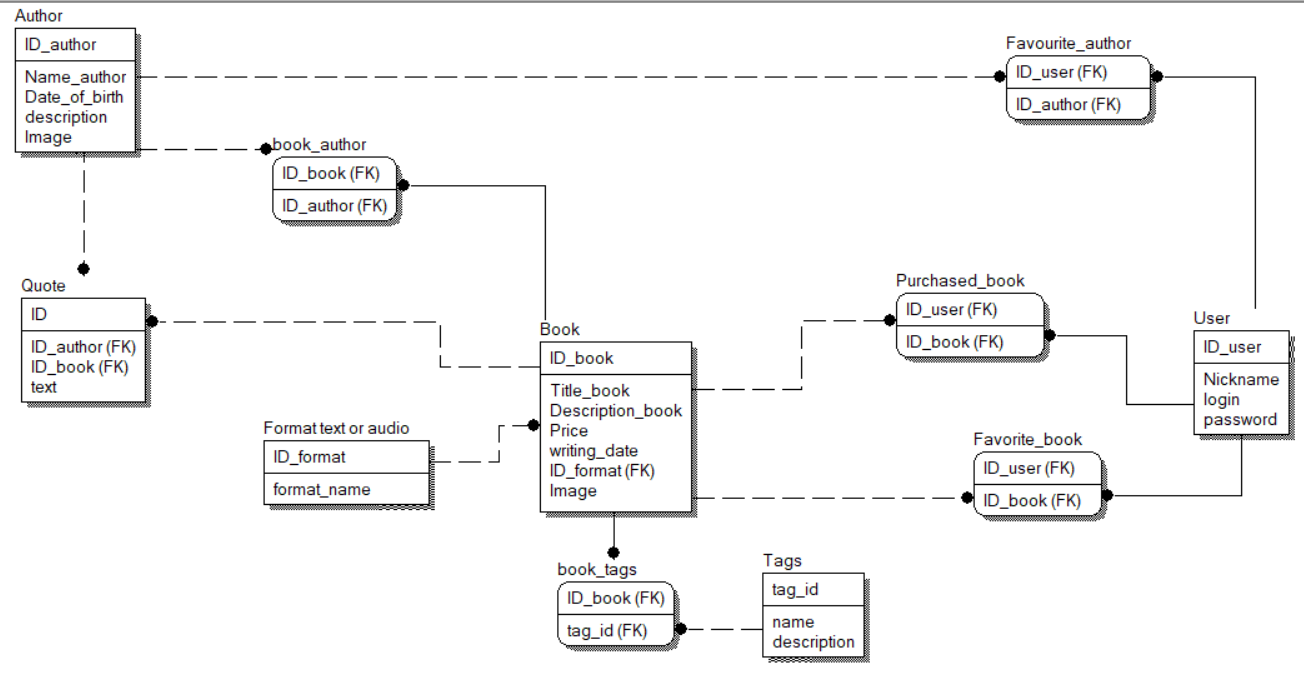


Рисунок . – Физическая модель базы данных

В разрабатываемой системе реализован REST API, который обеспечивает возможность обмена JSON-файлами. Один из который в качестве примера приведен на рисунке 3.2.

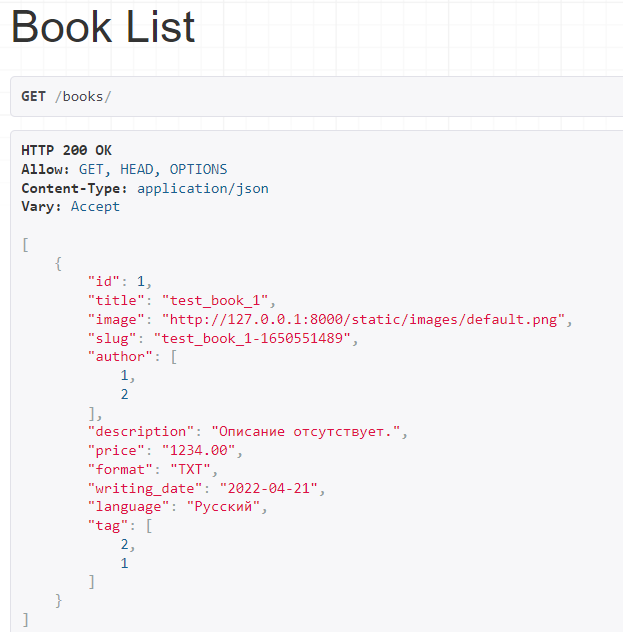


Рисунок . – JSON-файл со списком книг

В структуре приведенного файла видны несколько экземпляров модели «Книга». В левой части указано название столбцов, а в правой – их значения.

## Программная реализация информационной системы

Фреймворк Django реализует модель MVC (Model-View-Controller) или Модель-Представление-Контроллер. Компоненты MVC:

* модель – этот компонент отвечает за данные, а также определяет структуру приложения. Здесь описываются поля таблиц базы данных.
* представление – этот компонент отвечает за взаимодействие с пользователем. То есть код компонента view определяет внешний вид приложения и способы его использования;
* контроллер – этот компонент отвечает за связь между model и view. Код компонента controller определяет, как сайт реагирует на действия пользователя. По сути, это мозг MVC-приложения.

Приложение, разрабатываемое в курсовом проекте, реализует технологию тонкого клиента. То есть человеку, чтобы пользоваться им, не нужно скачивать специальные программы для работы в приложении. Достаточно иметь браузер с подключением к интернету [7].

## Пример работы с сайтом

На сайте есть два типа пользователей: клиент и сотрудник. У каждого из них разные права доступа: у сотрудника есть все права на создание, редактирование и удаление всех записей; клиент же может покупать книги, добавлять в избранное и создавать, редактировать и удалять цитаты. Примеры, приведенные дальше, будут рассмотрены с использованием аккаунта сотрудника.

### Пример создания экземпляра модели «Книга»

Для создания экземпляра книги необходимо перейти к списку всех книг и нажать на кнопку «Добавить». Далее Вас перенесет к странице создание новой книги. Пример страницы показан на рисунке 3.3. В форме необходимо заполнить поля: в поле «Название» введено «Новая книга», выбраны авторы данной книги и тд. После завершения заполнения полей необходимо нажать на кнопку «Готово», чтобы вас перенесло на страницу созданной книги. Данная страница представлена на рисунке 3.4.

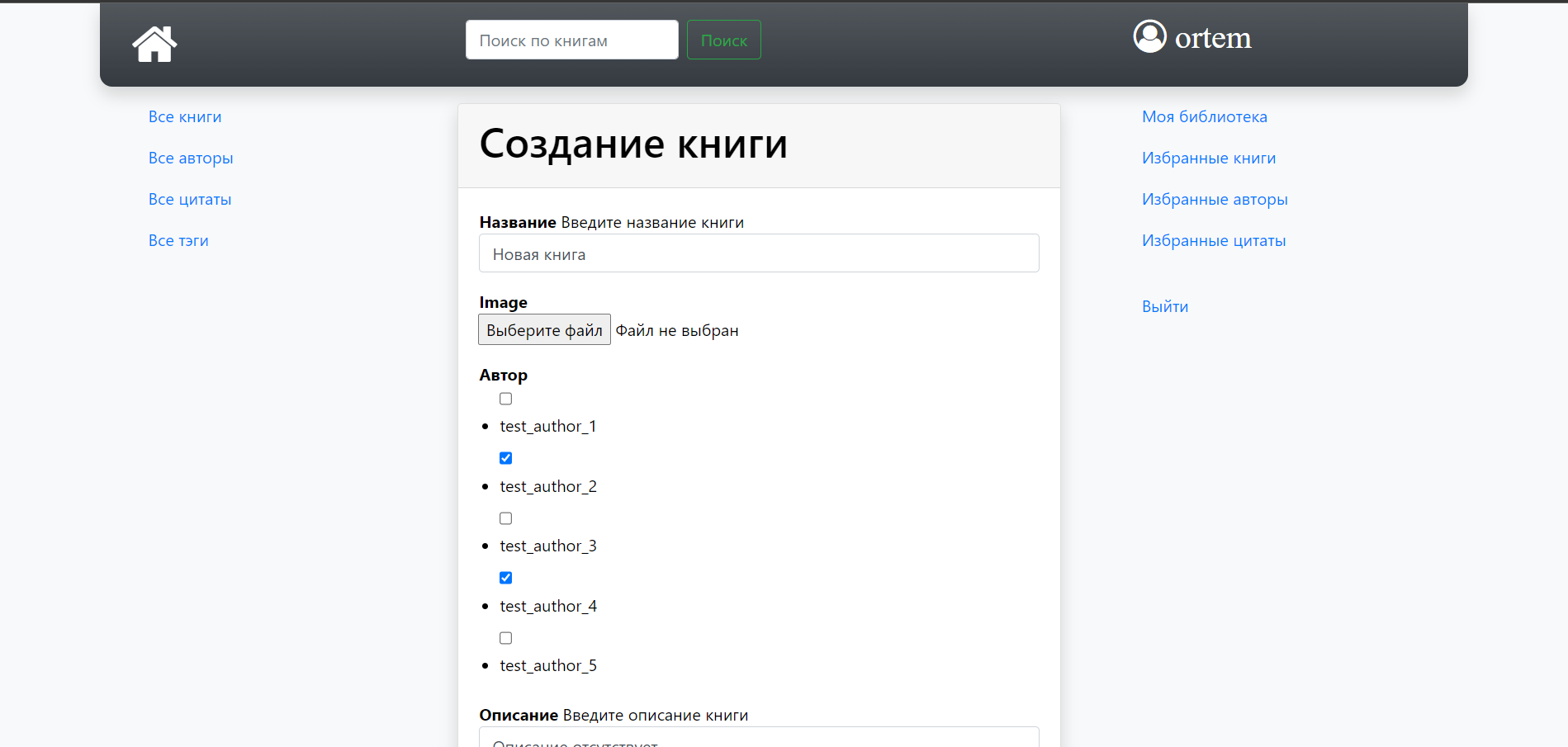


Рисунок . – Пример создания экземпляра модели «Книга»

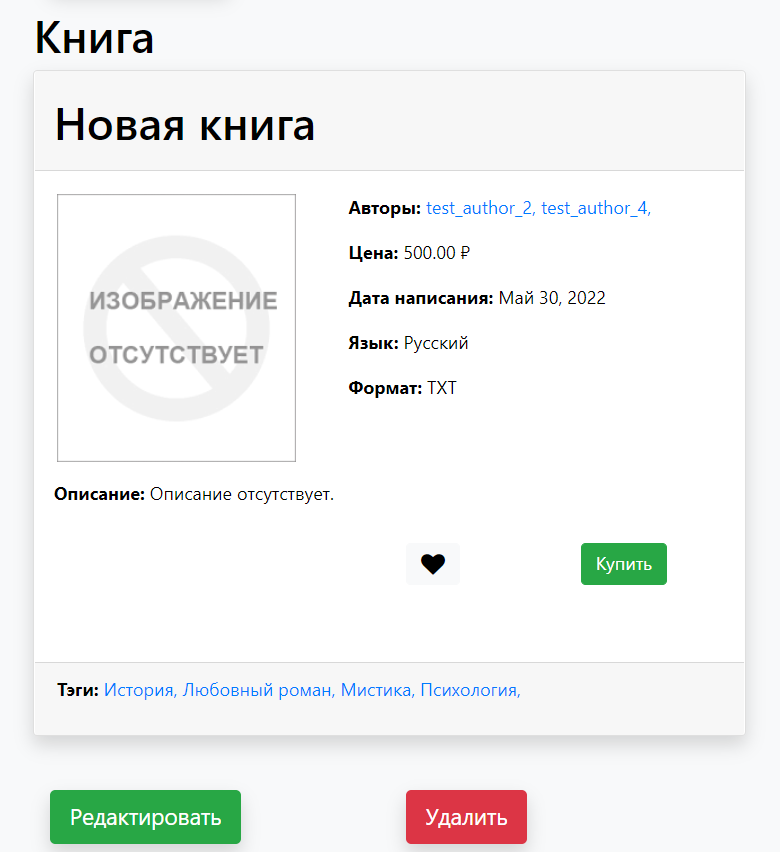


Рисунок . – Созданный экземпляр модели «Книга»

### Пример редактирования экземпляра модели «Книга»

Для редактирования созданной книги на рисунке 3.4 необходимо нажать на кнопку «Редактировать». Вас перенесет на форму редактирования, подобную форме создания нового экземпляра. После ввода необходимых изменений необходимо нажать на кнопку «Готово» для применения изменений.

### Пример удаления экземпляра модели «Книга»

Для удаления созданной книги на рисунке 3.4 необходимо нажать на кнопку «Удалить». Вас перенесет на страницу подтверждения удаления, где необходимо нажать соответствующую кнопку. После чего книга будет удалена.

### Пример добавления книги в избранное

Для добавления созданной книги на рисунке 3.4 в избранное необходимо нажать на кнопку с сердечком, после чего оно изменит свой цвет, и книга появится в разделе «Избранные книги»(рисунок 3.5).

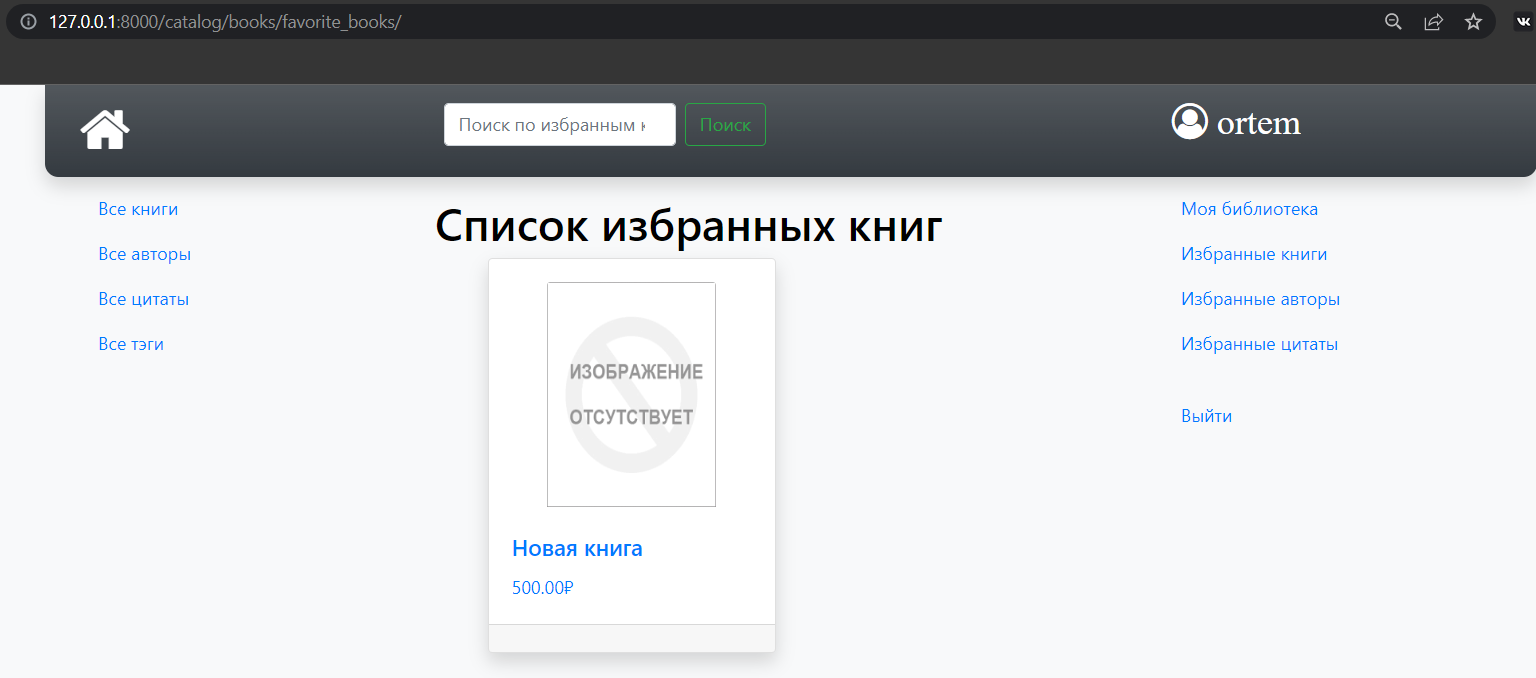


Рисунок . – Результат добавление в избранное

### Пример покупки книги

Для покупки созданной книги на рисунке 3.4 необходимо нажать на кнопку «Купить», после чего она изменится, и книга появится в разделе «Моя библиотека»(рисунок 3.6). Также после покупки появится кнопка «Скачать».

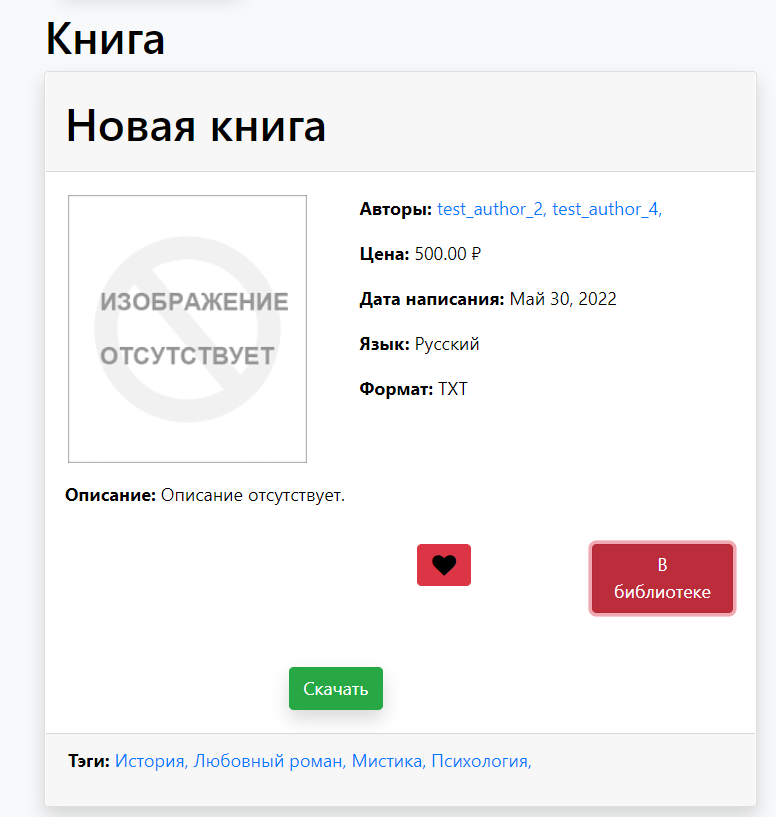


Рисунок . – Купленная книга

Следует отметить, что кнопки для добавления в избранное и покупки используют AJAX. AJAX – это аббревиатура, которая означает Asynchronous Javascript and XML. То есть, это подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером.

Таким образом, нет необходимости обновлять страницу самостоятельно, чтобы увидеть примененные изменения.

## Выводы

Таким образом, видно, что при создании сайта, использовались современные технологии: AJAX чаще всего ассоцириуется с термином Web 2.0 и преподносится как новейшее Web-приложение.

Выбранные инструменты в совокупности обеспечивают удобство пользования сайтом.

Отдельно следует отметить функционал добавления в избранное, потому что при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

# Заключение

В ходе выполнения курсового проекта выполнены следующие задачи:

* исследована и проанализирована предметная область;
* спроектирована система управления данными и описаны этапы проектирования информационной системы;
* выполнена полная разработка проектируемой системы.

Цель курсового проекта – получение практических навыков проектирования систем управления данными для управления производством и анализа экономической деятельности – достигнута.

# Список используемых источников

1. Дейт К. Дж. Введение в системы баз данных — 8-е изд. — М.: Вильямс, 2005. — 1328 с.
2. Грофф Джеймс Р., Вайнберг Пол Н., Опель Эндрю Дж. SQL: полное руководство, 3-е издание - М.:«Вильямс», 2014. - 960 с.
3. Грабер Мартин. SQL. – М.: Лори, 2007. – 672 с.
4. Ковязин А.Н., Востриков С.М. Мир InterBase: Архитектура, администрирование и разработка приложений баз данных в InterBase/Firebird/Yaffil. - М: КУДИЦ-Образ, 2003. – 496 с.
5. Барсегян А., Куприянов М., Холод И., Степаненко В. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004.
6. Спирли Эрик - “Корпоративные хранилища данных. Планирование, разработка и реализация“. - 2001. - 400с.
7. Скляр А.Я. Технология обработки и хранения данных (sql серверы и хранилища данных). Учебное пособие. Электронное издание. М.; МИРЭА. 2017, 181 с.

# Приложения

Приложение А – SQL-скрипт для создания триггеров базы данных.

Приложение Б – SQL-скрипт для создания структуры базы данных.

# Приложение А

create trigger get\_log\_update

after update

on buy\_composition

for each row

begin

INSERT INTO log (log\_operation, log\_row\_id)

VALUES ('update', NEW.buy\_id);

end;

create trigger get\_log\_delete

before delete

on buy\_composition

for each row

begin

INSERT INTO log (log\_operation, log\_row\_id)

VALUES ('delete', OLD.buy\_id);

end;

create trigger get\_log\_insert

after insert

on buy\_composition

for each row

begin

INSERT INTO log (log\_operation, log\_row\_id)

VALUES ('insert', NEW.buy\_id);

end;

# Приложение Б

create table catalog\_author

(

id integer not null

primary key autoincrement,

name varchar(200) not null,

slug varchar(200) not null

unique,

image varchar(100) not null,

date\_of\_birth date not null,

description text not null

);

create table catalog\_book

(

id integer not null

primary key autoincrement,

title varchar(200) not null,

image varchar(100) not null,

description text not null,

price decimal not null,

format varchar(20) not null,

writing\_date date not null,

language varchar(20) not null,

slug varchar(200) not null

unique

);

create table catalog\_book\_author

(

id integer not null

primary key autoincrement,

book\_id bigint not null

references catalog\_book

deferrable initially deferred,

author\_id bigint not null

references catalog\_author

deferrable initially deferred

);

create table catalog\_favoriteauthor

(

id integer not null

primary key autoincrement,

author varchar(255),

user varchar(255)

);

create table catalog\_favoritebook

(

id integer not null

primary key autoincrement,

book varchar(255),

user varchar(255)

);

create table catalog\_favoritequote

(

id integer not null

primary key autoincrement,

user varchar(255),

quote varchar(255)

);

create table catalog\_purchasedbook

(

id integer not null

primary key autoincrement,

book varchar(255),

user varchar(255)

);

create table catalog\_quote

(

id integer not null

primary key autoincrement,

text text not null,

slug varchar(200) not null

unique,

author\_id bigint not null

references catalog\_author

deferrable initially deferred,

book\_id bigint not null

references catalog\_book

deferrable initially deferred

);

create table catalog\_tag

(

id integer not null

primary key autoincrement,

name varchar(100) not null,

slug varchar(100) not null

unique,

description text not null

);

create table catalog\_book\_tag

(

id integer not null

primary key autoincrement,

book\_id bigint not null

references catalog\_book

deferrable initially deferred,

tag\_id bigint not null

references catalog\_tag

deferrable initially deferred

);