[ВВЕДЕНИЕ 2](#_Toc59044679)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 3](#_Toc59044680)

[1.1 Описание предметной области 3](#_Toc59044681)

[1.2 Пример разработок проекта для аналогичных систем 3](#_Toc59044682)

[1.3 Обоснование необходимости разработки 5](#_Toc59044683)

[1.4 Постановка задачи 5](#_Toc59044684)

[2. ТЕХНИЧЕСКОЕ ЗАДАНИЕ 7](#_Toc59044685)

[2.1 Назначения, цели создания web-приложения 7](#_Toc59044686)

[2.2 Требования к web-приложению в целом 7](#_Toc59044687)

[2.3 Требования к интерфейсу 8](#_Toc59044688)

[2.4 Требования к задачам и функциям 10](#_Toc59044689)

[2.5 Требования к видам обеспечения 11](#_Toc59044690)

[3. ФУНКЦИОНАЛЬНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ 12](#_Toc59044691)

[3.1 Разработка функциональной модели 12](#_Toc59044692)

[3.2 Моделирование бизнес-процессов 13](#_Toc59044693)

[3.3 Проектирование программного обеспечения 14](#_Toc59044694)

[4. ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ 16](#_Toc59044695)

[4.1 Реализация базы данных 16](#_Toc59044696)

[4.2 Организация сбора и обработки информации 18](#_Toc59044697)

[5. ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ 21](#_Toc59044698)

[5.1 Структура и функции частей web-приложения 21](#_Toc59044699)

[5.2 Архитектура web-приложения 22](#_Toc59044700)

[5.3 Описание программного модуля «Регистрация» 24](#_Toc59044701)

[5.4 Описание программного модуля «Заказ товара» 26](#_Toc59044702)

[5.5 Описание программного модуля «Добавление данных» 28](#_Toc59044703)

[5.6 Описание программного модуля «Удаление данных» 30](#_Toc59044704)

[5.7 Описание программного модуля «Добавление в корзину» 32](#_Toc59044705)

[6. Тестирование ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 33](#_Toc59044706)

[6.1 Модульное тестирование 33](#_Toc59044707)

[6.2 Тестирование интеграции 34](#_Toc59044708)

# ВВЕДЕНИЕ

В настоящее время широко применяется концепция web-приложения для интернет-магазинов, когда пользователю не нужно выходить из дома чтобы приобрести себе какой-либо товар.

На данный момент web-приложения для интернет-магазина представляют собой новую, но уже устоявшуюся систему реализации товаров или услуг. По сути эта система является более эффективной по сравнению со стандартным «физическим» магазином. Этот способ позволяет:

* улучшить качество обслуживания клиентов, так как все действия по покупке товаров производятся в автоматическом режиме, а менеджерам необходимо только уточнить информацию и выдать распоряжения о доставке товаров;
* снизить трудоемкость и напряженность труда персонала, минимизировать ошибки в его действиях;
* снизить издержки, так как отпадает необходимость в аренде здания для магазина.

Цель курсового проекта - разработать программное обеспечение обеспечения web-приложения для издательства книг.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* анализ предметной области;
* обзор аналогов;
* разработка технического задания;
* выполнение функционально-ориентированного проектирования;
* разработка информационного обеспечения;
* разработка программного обеспечения;
* выполнение тестирования программного обеспечения.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1. Описание предметной области

Выбранной предметной областью курсового проекта является издательство книг. Издательство книг работает как интернет магазин. Позволяет пользователям оформлять заказы на покупку и доставку заказов в сети Интернет.

Организация рабочего процесса в книжном магазине является довольно трудоемкой. Продавцы-консультанты должны знать наиболее полную информацию о продаваемых книгах, а также иметь возможность найти ее в кратчайшие сроки. Начальство магазина должно иметь возможность следить за выполнением заказов и т.д.

Предполагается, что сайт будет использоваться клиентами магазина, а также администраторами.

Каждый из клиентов магазина имеет свой профиль, корзину, а также возможность обратиться за помощью (рисунок 1.1).



Рисунок 1.1 – Use Case диаграмма организация рабочего процесса в книжном магазине

1. Пример разработок проекта для аналогичных систем

Прежде чем перейти к реализации web – приложения, рассмотрим примеры готовых сайтов, представленных на рисунках 1.2 и 1.3.

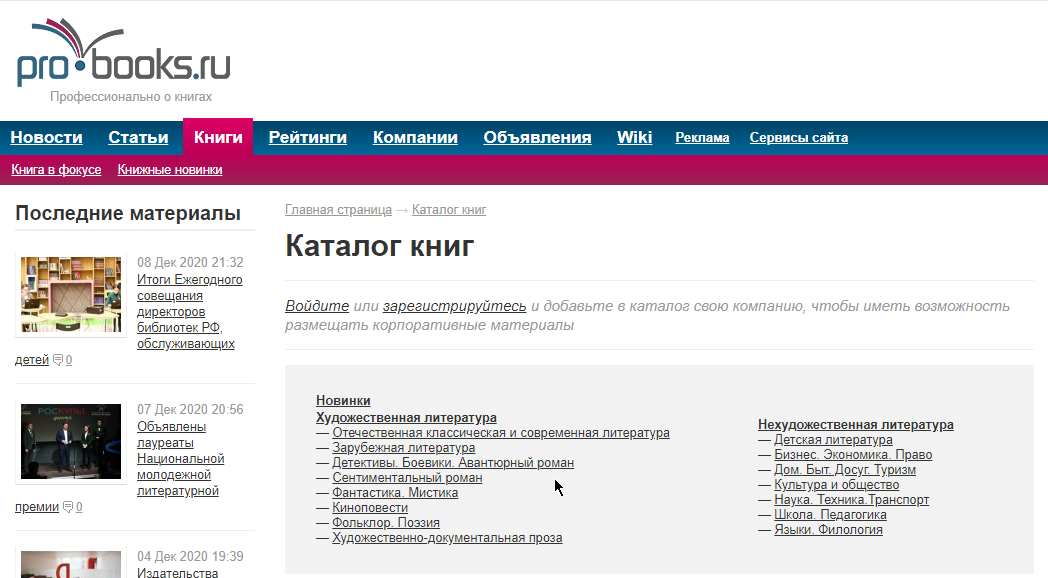


Рисунок 1.2 – Сайт каталога книг

Недостатки данного сайта:

* Нет возможности оформлять заказ;
* Нет регистрации/авторизации;
* Нет корзины заказов.

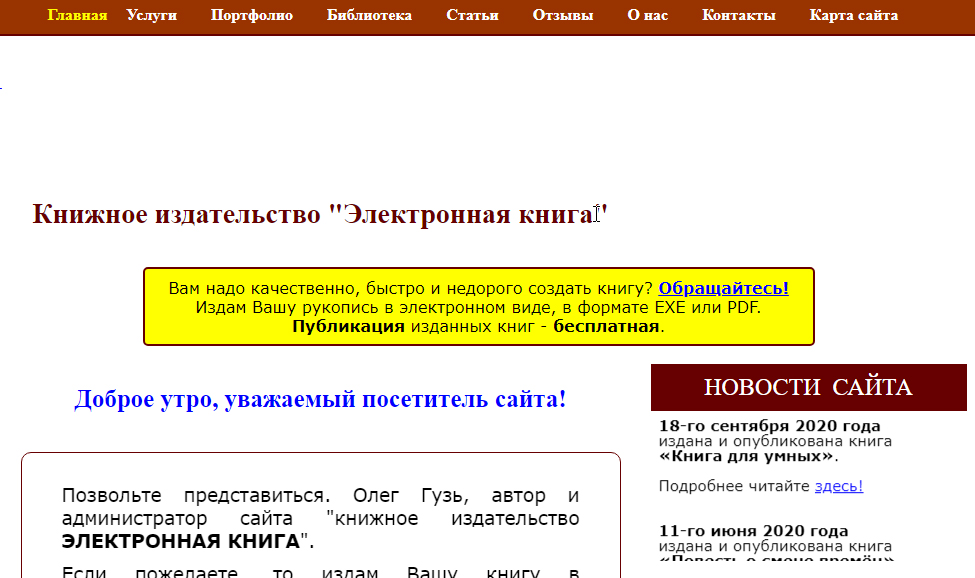


Рисунок 1.3 – Книжное издательство «Электронная книга»

Недостатки данного сайта:

* Нет на главной странице поиска по жанрам;
* Нет описания книги и стоимости;
* Нет административной панели.

1. Обоснование необходимости разработки

Основные причины, по которым издательству книг нужно web – приложение:

* На сайте издательство книг будет размещена вся информация о книгах, ценах, жанрах, подробное описание. Это поможет привлечь больше писателей, а так же покупателей.
* Постоянное размещение на сайте новых книг, информации об акциях – это тоже способ привлечь клиентов, которые захотят купить книгу или издать её, воспользовавшись условиями акции.
* Услуга онлайн заказа и доставки на дом. Наличие сайта даёт такую возможность. Наличие услуги доставки и оформление заказа в режиме онлайн добавит популярности издательству книг и дополнительно увеличит доход.

1. Постановка задачи

Для обеспечения управления интернет-магазином целесообразно разграничить права пользователей. Интернет-магазин имеет два типа пользователей:

1) администратор;

2) пользователь.

«Администратор» имеет возможность добавлять, удалять, изменять книги и категории книг, вносить изменения в дизайн сайта, производить просмотр заказов, в том числе с частично отсутствующими данными для доставки. Также администратор выполняет управление контентом сайта.

«Пользователь» регистрируется или авторизируется просматривает страницы интернет-магазина с помощью веб-браузера, выбирает категории книг, просматривает информацию о конкретной книге, помещает книги в корзину, просматривает корзину, производит поиск книги по названию, оформляет заказ на покупку одной или нескольких книг.

«Продавец-консультант» имеет возможность заказывать книгу, просматривать статистику, просматривать информацию о книгах.

«Начальник» может просматривать статистику, а так же просматривать информацию о книгах.

# ТЕХНИЧЕСКОЕ ЗАДАНИЕ

## Назначения, цели создания web-приложения

Основной целью создания и назначения web-приложения для издательства книг «booKKing» является привлечение клиентов, использующих сеть Internet, чтобы не тратить свое время на поиски книг, знакомить клиентов с жанрами книг и давать возможность оформление заказов.

* улучшить качество обслуживания клиентов, так как все действия по покупке товаров производятся в автоматическом режиме, а продавцу-консультанту необходимо только уточнить информацию и выдать распоряжения о доставке товаров;
* снизить трудоемкость и напряженность труда персонала, минимизировать ошибки в его действиях.
* снизить издержки, так как отпадает необходимость в аренде здания для магазина;

## Требования к web-приложению в целом

Web-приложение должно состоять из двух частей - клиентской и административной. Клиентская часть доступна всем пользователям, которые заходят на сайт. Клиентская часть включает в себя:

* главная страница, посвященная ознакомлению с сайтом;
* контактная информация;
* история магазина;
* помощь в работе с сайтом;
* каталог.

Административная часть включает:

* добавление новых пользователей;
* добавление новых товаров;
* добавление новых страниц сайта;
* обработка заказов.

Web-приложение создано для представительства книжного магазина «Книжный Мир» в сети для потенциальных клиентов под доменным именем booKKing.ru. Web-приложение должен состоять из взаимосвязанных разделов с четко разделенными функциями.

Для поддержания сайта и эксплуатации веб-интерфейса системы управления сайтом от персонала не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб-браузером, например, Yandex версии 20.11.3 или Google Ghrome 87. Информация, размещаемая на сайте, является общедоступной.

Пользователи сайта: покупатель администратор, начальник, продавец - консультант.

Покупатель может проводить регистрацию/авторизацию и заказывать книгу или книги.

Доступ к административной части имеют пользователи с правами администратора.

Администратор может добавлять и изменять категорию, добавлять и изменять книгу, просматривать статистику.

Начальник может просматривать статистику, а так же просматривать информацию о книгах.

Продавец – консультант может заказать книгу, просмотреть статистику, просмотреть информацию о книгах

## Требования к интерфейсу

При разработке сайта должны использоваться цветовые стили такие как тёмно-синий, белый, тёмно-розовый. Основные разделы сайта должны быть доступны с первой страницы. На первой странице не должно быть большого объема текстовой информации.

В дизайне сайта не должны присутствовать:

* мелькающие барьеры;
* много сливающегося текста;
* единообразия;
* слишком мелкого текста;
* слияние текста с фоном;
* перебора с картинками.

На главной странице web – приложения должны быть расположены секции элементов:

* Шапка сайта (Рисунок 1.4);
* Каталог (Рисунок 1.5)
* Подвал сайта (Рисунок 1.6)

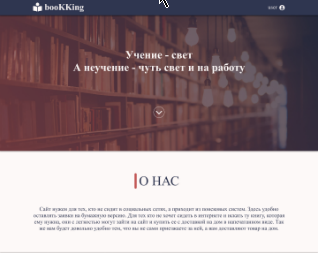


Рисунок 1.4 – Секция элемента «Шапка сайта»

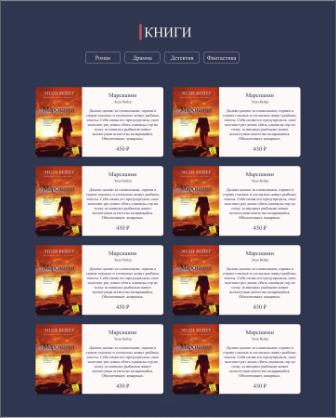


Рисунок 1.5 – Секция элемента «Каталог»

PicasaPhotoViewer_sUuRLx6Ntf

Рисунок 1.6 – Секция элемента «Подвал сайта»

## Требования к задачам и функциям

Функции клиентской части web-приложения:

* Аутентификация и регистрация;
* Просмотр информации о книге;
* Добавление книги в корзину;
* Просмотр корзины;
* Заказ книг из корзины;
* Удаление книги из корзины;

Функции для административной части:

* Добавить / изменить категорию.
* Добавить / изменить книгу.
* Просмотреть статистику.
* Просмотр, добавление, изменение, удаление информации о авторах
* Просмотр, добавление, изменение, удаление информации о книгах
* Просмотр, добавление, изменение, удаление информации о категориях
* Просмотр, добавление, изменение, удаление информации о продажах
* Просмотр, добавление, изменение, удаление информации о пользователях
* Просмотр, добавление, изменение, удаление информации о корзинах

## Требования к видам обеспечения

Для того, что бы клиент смог просматривать информацию на сайте, ему необходимо скачать браузер (Yandex версии 20.11.3 или Google Ghrome 87) и перейти на сайт издательства книг booKKing.ru

Для реализации web-приложения разработчику необходимо установить:

* Photoshop CS5;
* Текстовый редактор Sublime Text 3;
* База данных MySQL;
* PHP;
* Apache.

Для разработки можно использовать локальный сервер Denver (PHP, MySQL, Apache).

# ФУНКЦИОНАЛЬНО-ОРИЕНТИРОВАННОЕ ПРОЕКТИРОВАНИЕ

## Разработка функциональной модели

Перед построением функциональной модели необходимо изучить данную предметную область. В качестве предметной области рассматриваем издательство книг «booKKing». Данное предприятие принимает заказы клиентов и доставляет их по указанному адресу. Основные процедуры:

* Администратор принимает заказы;
* Продавец-консультант просматривает информацию о книгах;
* Курьер доставляет данную книгу на указанный адрес.

Компания использует бухгалтерскую систему, которая позволяет оформить заказ, счет и отследить платежи по счетам

В данной предметной области субъектом будет выступать само предприятие; цель моделирование – воспроизвести бизнес процессы, происходящие на предприятии.

После определения контекста моделирования, можно приступить к построению контекстной диаграммы. Контекстная диаграмма показана на рисунке 3.1.

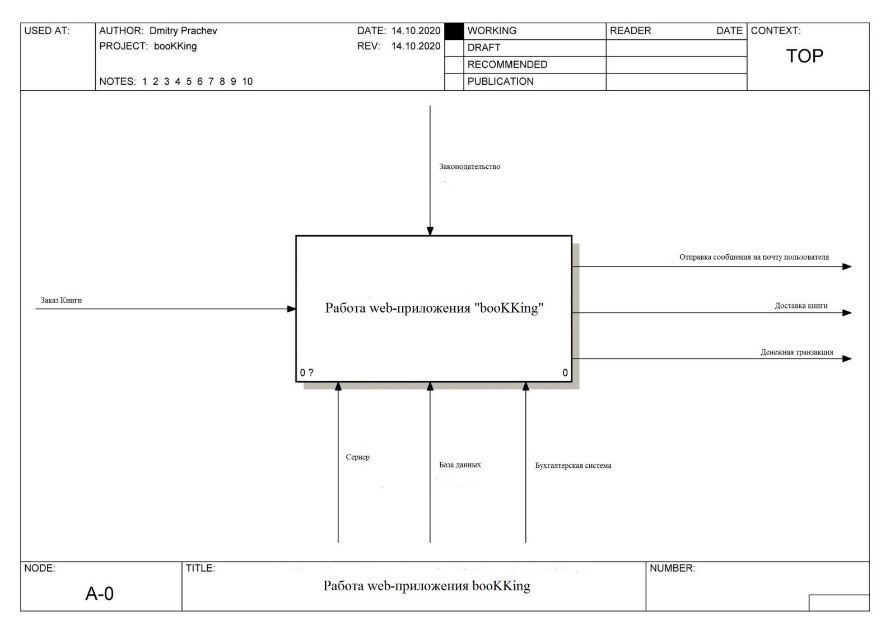


Рисунок 3.1 – Контекстная диаграмма А-0

Для рассматриваемого предприятия входными стрелками будут:

* Запрос книги.

Выходные стрелки:

* Отправление сообщения на почту покупателя;
* Доставка книги;
* Денежная транзакция;

Стрелки управления:

* Законодательство.

Стрелки механизмов:

* Сервер;
* База данных;
* Бухгалтерская система.

## Моделирование бизнес-процессов

После создания дочерней диаграммы соединим граничные стрелки с работами. Стрелки «Заказ книги» и «Аккаунт», «Сервер», «База данных» соединим с работой «Генерация каталога книг». Выходом работы «Генерация каталога книг» будет «Выбранные книги», выходом «Генерация страницы книги» - «Покупка». Стрелки «Законодательство», «Сервер», «База данных», , «Аккаунт» будет входом в механизм в следующих работах: «Обработка покупки» и «Регистрация / авторизация », «Генерация каталога книг», а стрелка «Бухгалтерская система» - «Обработка покупки» , а на выходе с работы «Обработка покупки» будут стрелки «Отправка сообщения на почту пользователя», «Денежная транзакция», «Доставка книги».

После соединения граничных стрелок с работами, приступим к соединениям работ между собой.

Работа «Регистрация / авторизация» получает на входе законодательство, сервер, базу данных, информацию о которых она передает работе «Генерация каталога книг» в качестве входной информации.

Работа «Генерация каталога книг» на входе получает аккаунт и заказ книги, а так же сервер и базу данных , после чего передает информацию работе «Генерация страницы книги» в качестве входной информации.

Работа «Генерация страницы книги» на входе получает выбранную книгу, заказ книги, сервер и базу данных, которую должен оплатить клиент. После оплаты, информация передается работе «Обработка покупки» в качестве входной информации, а на выходе отправка сообщения на почту покупателя, денежная транзакция, доставка книги.

После соединения работ между собой, приступим к построению диаграммы декомпозиции. Диаграмма декомпозиции показана на рисунке 3.2.

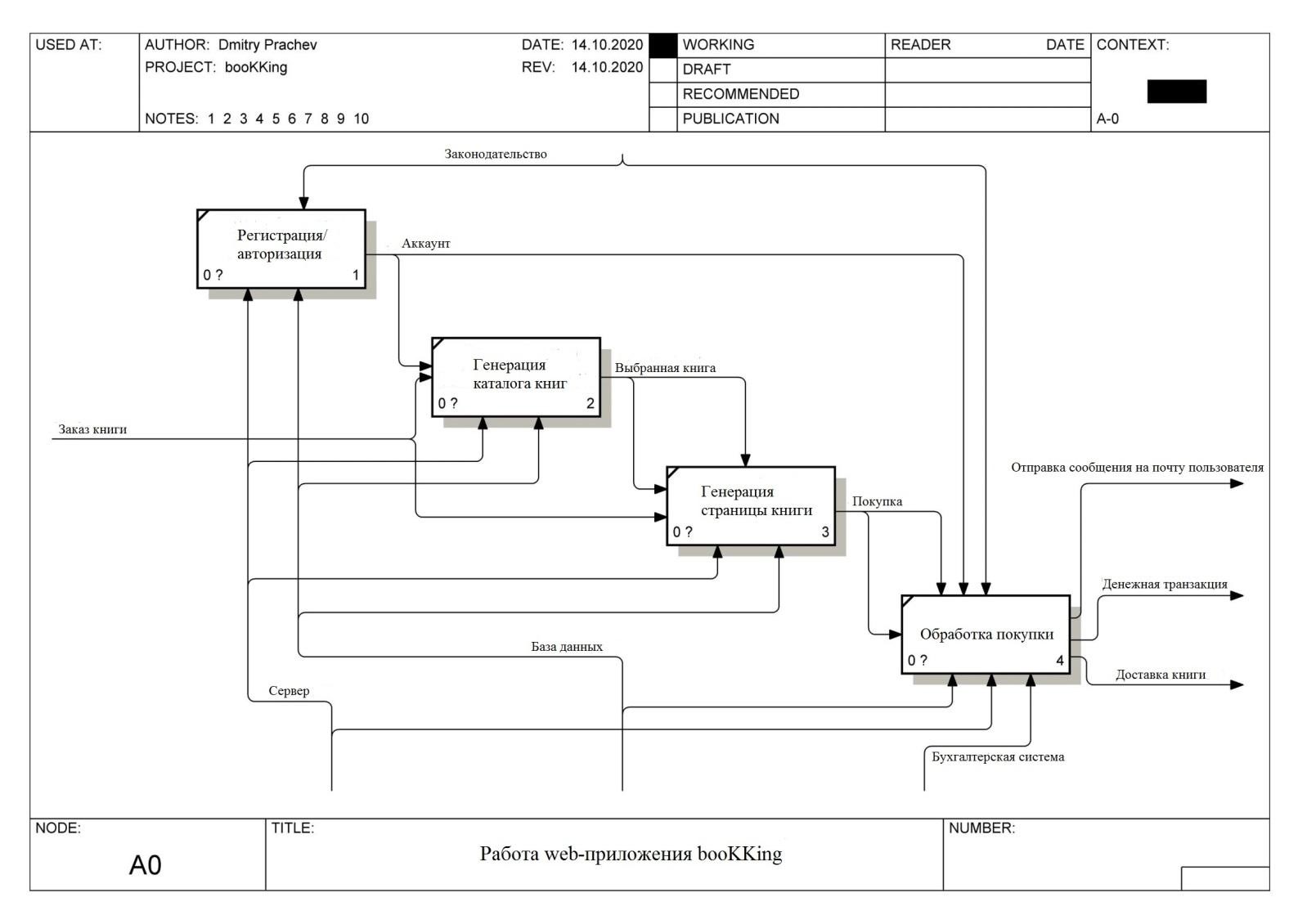
****

Рисунок 3.2 – Диаграмма декомпозиции.

## Проектирование программного обеспечения

Для реализации web-приложения проектирует базу данных booKKing. Для проектирования базы данных выделим базовые сущности предметной области издательство книг «booKKing»:

* сущность authors с атрибутами: id\_authors, surname (фамилия), name (имя), patronymic (отчество);
* сущность books с атрибутами: id\_book, id\_author, name (имя), description (описание), img (изображение), price (цена)
* сущность categories с атрибутами: id\_category, name (имя), code (код);
* сущность books\_categories с атрибутами: id\_book, id\_category;
* сущность users с атрибутами: id\_user , surname (Фамилия), name (имя), email (электронный адрес), password (пароль), address (адрес), phone (телефон);
* сущность sales с атрибутами: id\_sale, id\_user, date (дата), done (сделано), price (цена);
* сущность cart с атрибутами: id\_cart, id\_sale, id\_book, quanity (количество), price (цена);

Логическая модель информационной системы представлена на рисунке 3.3.

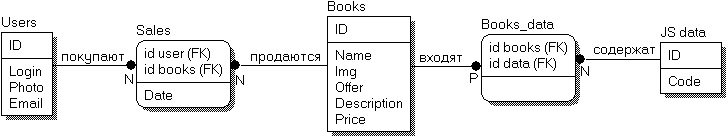


Рисунок 3.3 - Логическая модель информационной системы

Концептуальная модель базы данных показана на рисунке 3.4.

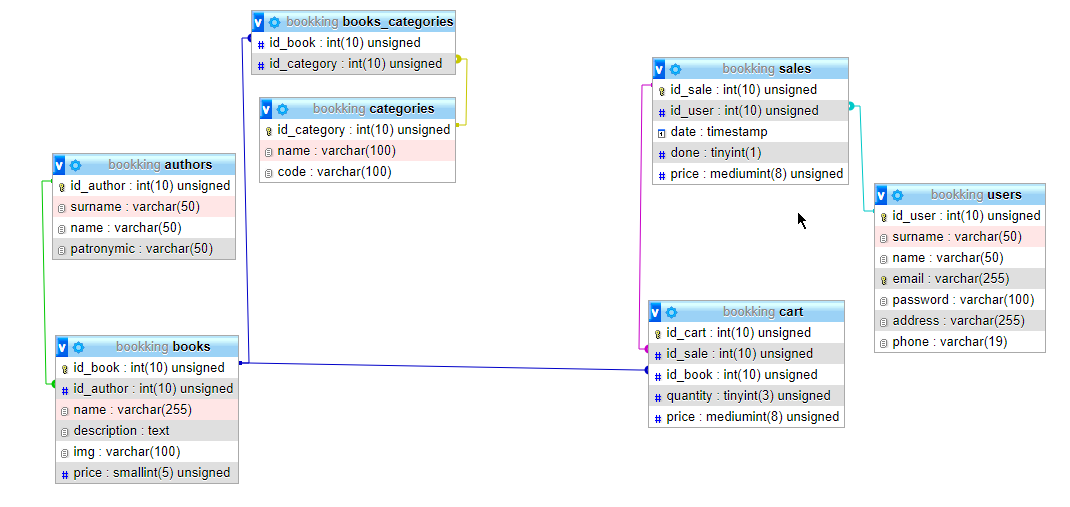


Рисунок 3.4 – Концептуальная модель базы данных

# ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

## Реализация базы данных

База данных издательство книг «booKKing» реализована в PhpMyAdmin в 7 таблиц: authors, books, books\_categories, cart, categories, sales, users. База данных предоставлена на рисунке 4.1.

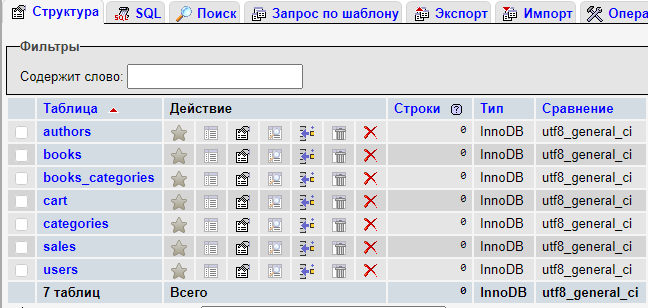


Рисунок 4.1 – База данных «booKKing»

Таблица authors содержит информацию о последних новостях. Таблица содержит следующие поля: id\_authors, surname, name, patronimyc. Пример показан на рисунке 4.2.



Рисунок 4.2 – Таблица «authors»

Таблица books. Содержит информацию о книгах, авторах книг, описание, а также изображение книг и цену книг. Таблица содержит следующие поля: id\_book, id\_author, name, description, img, price. Пример показан на рисунке 4.3.

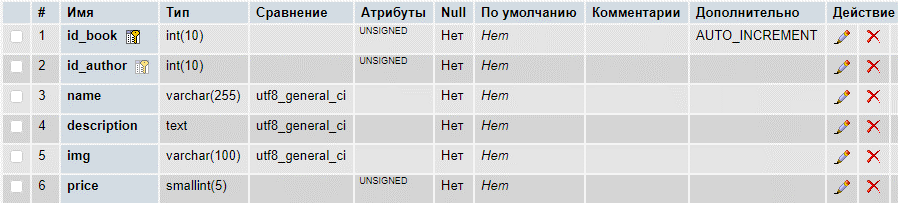


Рисунок 4.3 – Таблица «books»

Таблица books\_categories содержит информацию о категориях книг. В таблице имеются такие поля: id\_categories,id\_book. Пример таблицы показан на рисунке 4.4.



Рисунок 4.4 – Таблица «books\_categories»

Таблица cart содержит id корзины, id продажи, id книги, которую покупает пользователь, количсество. В таблице имеются следующие поля: id\_cart, id\_sale, id\_book, quantity. Пример таблицы показан на рисунке 4.5.



Рисунок 4.5 – Таблица «cart»

Таблица categories имеет id категории, название и код. Таблица содержит следующие поля: id\_category, name, code. Пример таблицы показан на рисунке 4.6.

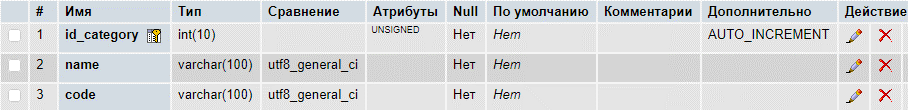


Рисунок 4.6 – Таблица «categories»

Таблица sales содержит информацию о продажах. Она содержит информацию о названии заказанной книги, id пользователя, дату заказа, цену. Таблица имеет следующие поля: id\_sale, id\_user, date, done, price. Пример таблицы показан на рисунке 4.7.

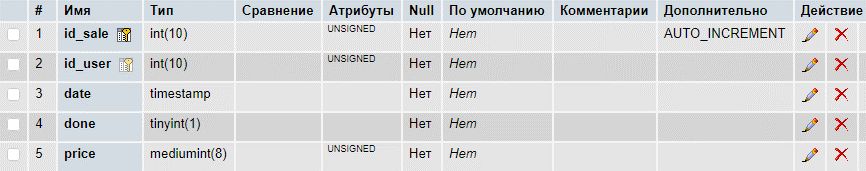


Рисунок 4.7 - Таблица «Sales»

## Организация сбора и обработки информации

Сбор и обработка информации обрабатывается вручную и автоматически.

Пример ручного заполнения информации показан на рисунке 4.8.

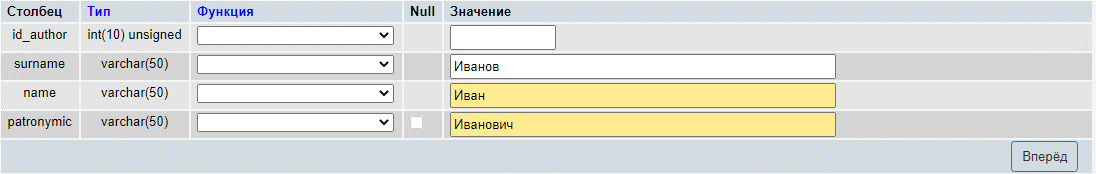


Рисунок 4.8 – Ручное заполнение информации

Для автоматической организации сбора и обработки информации, разработана админ панель управления сайта, доступ к которой осуществляется на сайте при вводе логина и пароля (рисунок 4.9).



Рисунок 4.9 – Панель управления контентом web-приложения

Для добавления, редактирования и удаление записей разработаны соответствующие кнопки. Добавление данных реализовано с помощью формы, показанной на рисунке 4.10

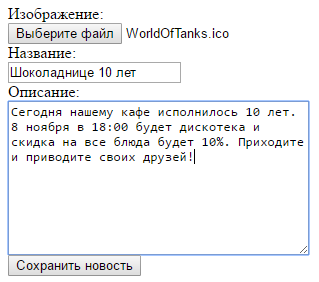


Рисунок 4.10 – Добавление информации на сайт

Для редактирования информации необходимо нажать на кнопку «Редактировать». Редактирование данных реализовано с помощью формы, показанной на рисунке 4.11.

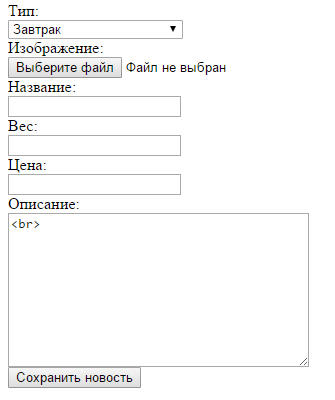


Рисунок 4.11 – Редактирование информации на сайте

Для того, что бы удалить информацию, необходимо во вкладку, в которой хотите удалить информацию и нажать на кнопку «Удалить». Информация автоматически удалится из таблицы и страница автоматически обновится.

# ПРОГРАМНОЕ ОБЕСПЕЧЕНИЕ

## Структура и функции частей web-приложения

Чаще всего web-приложения состоят как минимум из двух основных компонентов:

* Клиентская часть web-приложения;
* Админ часть web-приложения;

Это основные компоненты большинства web-приложений. Схема взаимодействия компонентов web-приложения показаны на рисунках 5.1 и 5.2.

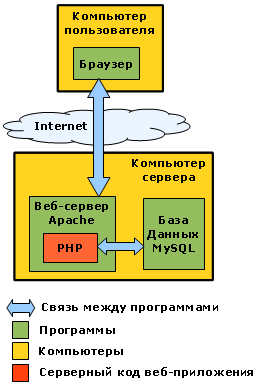


Рисунок 5.1 - Схема взаимодействия компонентов web-приложения



Рисунок 5.2 - Схема взаимодействия компонентов web-приложения

* Браузер через Интернет отсылает HTTP-запросы веб-серверу;
* Веб-сервер вызывает PHP-скрипт;
* PHP-скрипт обращается к базе данных;
* В результате PHP-скрипт возвращает клиенту веб-страницу.

## Архитектура web-приложения

Многоуровневая архитектура

Это одна из самых распространенных архитектур. На её основе построено множество крупных фреймворков — Java EE, Drupal, Express. Пожалуй, самый известный пример этой архитектуры — это сетевая модель OSI.

Система делится на уровни, каждый из которых взаимодействует лишь с двумя соседними. Поэтому запросы к БД, которая обычно располагается в самом конце цепочки взаимодействия, проходят последовательно сквозь каждый «слой». Пример многоуровневой архитектуры показан на рисунке 5.3.

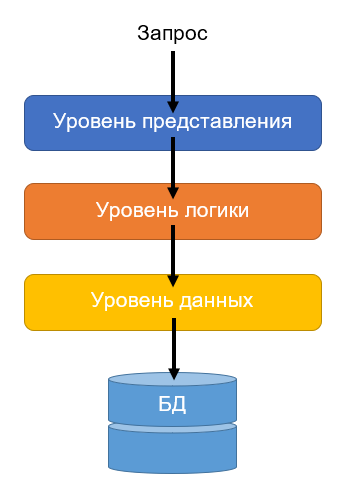


Рисунок 5.3 – Многоуровневая архитектура

Событийно-ориентированная архитектура

В этом случае разработчик прописывает для программы поведение (реакции) при возникновении каких-либо событий. Событием в системе считается существенное изменение её состояния.

Система, управляемая событиями, обычно содержит два компонента: источники событий и их потребители. Типов событий обычно тоже два: инициализирующее событие и событие, на которое реагируют потребители.

В данном курсовом проекте используется трехуровневая архитектура. Пример трехуровневой архитектуры показан на рисунке 5.4.

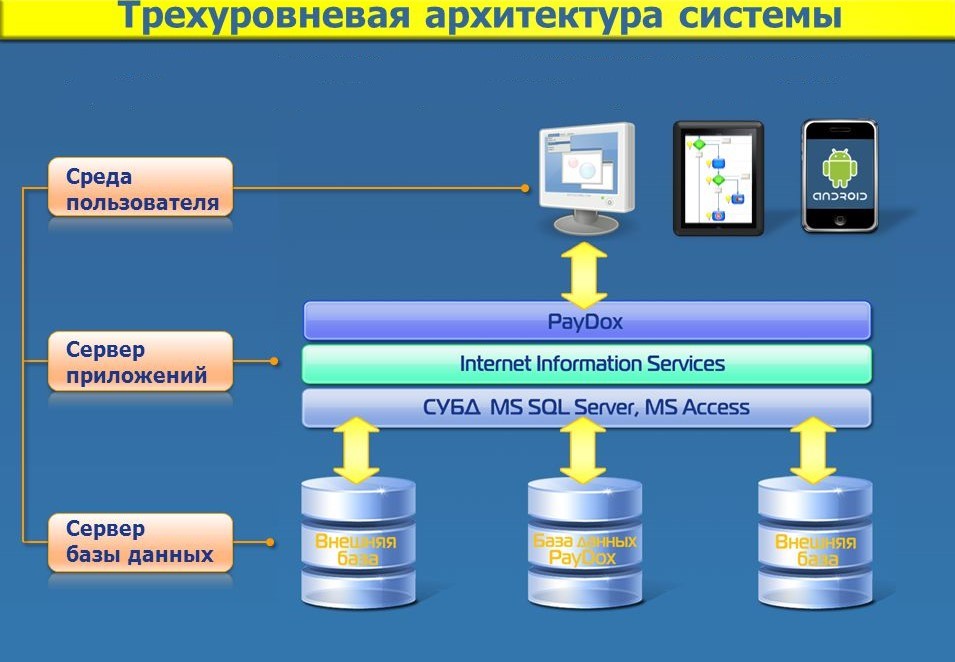


Рисунок 5.4 – Трехуровневая архитектура системы

Локальная схема клиентского интерфейса показана на рисунке 5.5.

Пользовательский интерфейс

add.php

add\_hn.php

add\_kt.php

add\_l.php

add\_p.php

add\_u.php

add\_z.php

delete.php

holod\_napoy.php

index.php

izm.php

karty\_abonement.php

karty\_bonus.php

karyera.php

koffe\_tea.php

kontakty.php

lanch.php

menu.php

MySyte.php

MySyteAdd.php

novosty.php

o\_kompanii.php

pazza.php

ujin.php

zavtrak.php

style

footer.css

glav.css

header.css

header\_glav.css

karty.css

karyera.css

kontakty.css

menu.css

novosty.css

o\_kompanii.css

java

footer.js

jquery.maskidinput.min.js

jquery-2.2.4.js

jquery-latest.min.js

menu.js

menu\_ed.js

menu-glav.js

novosty.js

image

users

Снимок\_экрана\_(2).png

z.png

blocks

footer.php

glav.php

header.php

header-glav.php

karty\_abonement1.php

karty\_abonement2.php

karty\_bonus1.php

karty\_bonus2.php

karyera.php

kontakty.php

menu\_holod\_napoy.php

menu\_koffe\_tea.php

menu\_lanch.php

menu\_pizza.php

menu\_ujin.php

menu\_vse.php

menu\_zavtrak.php

novosty.php

o\_kompanii.php

Рисунок 5.5 – Локальная схема клиентского интерфейса

## Описание программного модуля «Регистрация»

Данный модуль позволяет зарегистрировать клиента, что бы он мог входить на сайт под своей учетной записью. Что бы зарегистрировать клиента, необходимо нажать на кнопку «Зарегистрироваться», после чего внести данные и нажать на кнопку «Зарегистрироваться». Пример показан на рисунке 5.6.

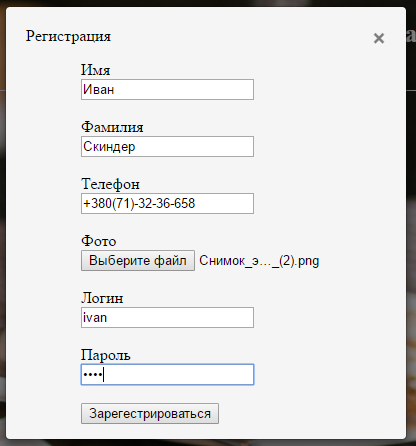


Рисунок 5.6 – Форма регистрации клиента

Данные заносятся в таблицу users. Таблица users показана на рисунке 5.7.

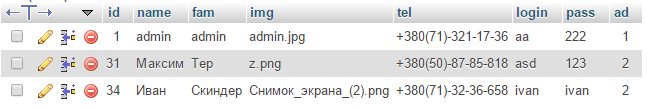


Рисунок 5.7 – Таблица users

Скрипт регистрации клиента (add\_users.php):

// получение данных

$n\_name\_r = $\_POST['n\_name\_r'];

$f\_name\_r = $\_POST['f\_name\_r'];

$kaha = $\_FILES["foto\_r"]["name"];

$telefon\_r = $\_POST['telefon\_r'];

$login\_r = $\_POST['login\_r'];

$pass\_r = $\_POST['pass\_r'];

$ad = 2;

// поместить изображение в папку

if (isset($\_POST["MAX\_FILE\_SIZE"])) {

$tmp\_file\_name = $\_FILES["foto\_r"]["tmp\_name"];

$dest\_file\_name=$\_SERVER['DOCUMENT\_ROOT']."/blocks/users/image/users/".$\_FILES["foto\_r"]["name"];

move\_uploaded\_file($tmp\_file\_name, $dest\_file\_name);}

//запись данных в таблицу

mysqli\_query($link, "INSERT INTO users (name, fam, img, tel, login, pass, ad) VALUES ('$n\_name\_r', '$f\_name\_r', '$kaha', '$telefon\_r', '$login\_r', '$pass\_r', '$ad')");

## Описание программного модуля «Заказ товара»

Данный модель позволяет оформить заказ. Для оформления заказа необходимо выбрать жанр и выбрать книгу. После выбора книги, нужно на него нажать, а потом нажать на кнопку «Заказать». После нажатия кнопки, появится окно, в котором нужно заполнить все поля и нажать на кнопку «Оформить заказ». Форма для оформления заказа представлена на рисунке 5.8.

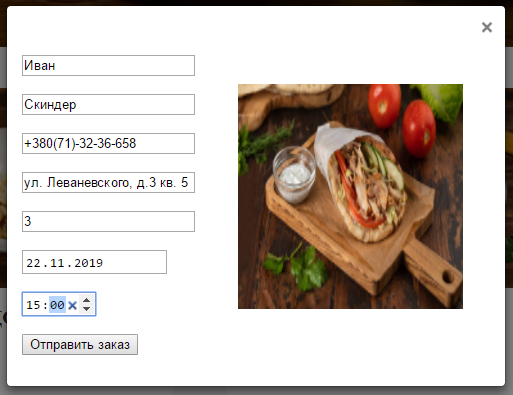


Рисунок 5.8 – Форма оформления заказа

Данные о заказе заносятся в таблицу cart. Таблица cart показана на рисунке 5.9.



Рисунок 5.9 – Таблица cart

Скрипт оформления заказа:

$n\_name = $\_POST['n\_name'];

$f\_name = $\_POST['f\_name'];

$telefon = $\_POST['telefon'];

$data\_z = $\_POST['data\_z'];

$vremy\_z = $\_POST['vremy\_z'];

$data = $\_POST['data'];

$vremy = $\_POST['vremy'];

$cena1 = $\_POST['cena1'];

$nazv = $\_POST['z\_nazv'];

$kol = $\_POST['kol'];

$addres = $\_POST['addres'];

$type = $\_POST['type'];

$cena\_i = $kol\*$cena1;

if ($n\_name != "" and $f\_name != "" and $telefon != "" and $data != "" and $vremy != "" and $kol != "" and $addres != "") {

if ($data < $data\_z) {

echo "Введите дату не раньше сегоднешней";

echo '<br><a href="menu.php">Назад</a><br>'; }

elseif ($data == $data\_z) {

$time = time();

$time\_f = $time + 6000;

$time\_o = date("H:i:s","28800");

$time\_z = date("H:i:s","54000");

if ($vremy >= $time\_f and $vremy >= $time\_o and $time <= $time\_z) {

mysqli\_query($link, "INSERT INTO zakazi (name, fam, tel, addres, type, nazv, kol, cena, cena\_itog, data\_z, data, vremy) VALUES ('$n\_name', '$f\_name', '$telefon', '$addres', '$type', '$nazv', '$kol', '$cena1', '$cena\_i', '$data\_z', '$data', '$vremy')");

mysqli\_close($link);

header('Location: menu.php'); }

else{

echo "Введите время в промежутке от 9:00 до 18:00 или за 2 часа до заказа";

echo '<br><a href="menu.php">Назад</a><br>'; }}

elseif ($data > $data\_z){

$time\_o = date("H:i:s","28800");

$time\_z = date("H:i:s","54000");

if ($vremy >= $time\_o and $time <= $time\_z){

mysqli\_query($link, "INSERT INTO zakazi (name, fam, tel, addres, type, nazv, kol, cena, cena\_itog, data\_z, data, vremy) VALUES ('$n\_name', '$f\_name', '$telefon', '$addres', '$type', '$nazv', '$kol', '$cena1', '$cena\_i', '$data\_z', '$data', '$vremy')");

mysqli\_close($link);

header('Location: menu.php'); }

else{

echo "Введите время в промежутке от 9:00 до 18:00";

echo '<br><a href="menu.php">Назад</a><br>'; }}}

else{

echo "Заполните все поля";

echo '<br><a href="menu.php">Назад</a><br>'; }

## Описание программного модуля «Добавление данных»

Данный модуль позволяет добавлять книги на сайт. Для добавления книг на сайт, необходимо нажать на кнопку добавить. После чего ваш запрос на добавление рассматривает администратор, который может как принять, так и отклонить вашу книгу. Пример добавления книги в раздел где все категории показана на рисунке 5.10.

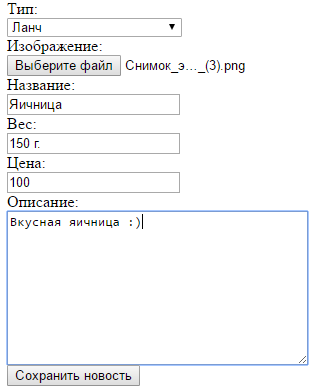


Рисунок 5.10 – Форма добавления информации

Данные заносятся в таблицу books. Пример показан на рисунке 5.11.

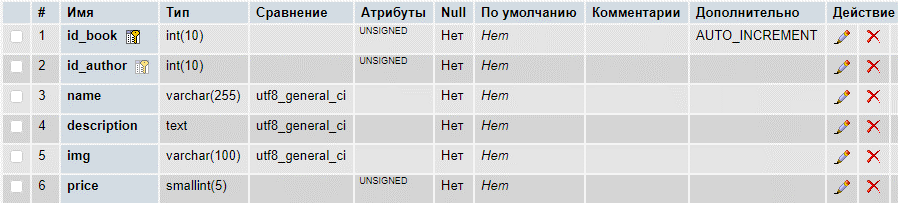


Рисунок 5.11 – Таблица books

Скрипт добавления информации:

if (isset($\_POST["MAX\_FILE\_SIZE"])) {

$tmp\_file\_name = $\_FILES["img"]["tmp\_name"];

$dest\_file\_name = $\_SERVER['DOCUMENT\_ROOT']."/blocks/admin/image/menu/".$\_FILES["img"]["name"];

move\_uploaded\_file($tmp\_file\_name, $dest\_file\_name);}

if (isset($\_POST['save'])){

$type=strip\_tags(trim($\_POST['type']));

$img=strip\_tags(trim($\_FILES["img"]["name"]));

$nazv=strip\_tags(trim($\_POST['nazv']));

$ves=strip\_tags(trim($\_POST['ves']));

$cena=strip\_tags(trim($\_POST['cena']));

$txt=strip\_tags(trim($\_POST['txt']));

mysqli\_query($link, "INSERT INTO menu (type, img, nazv, ves, cena, txt) VALUES ('$type', '$img', '$nazv', '$ves', '$cena', '$txt')");

mysqli\_close($link);

header('Location: menu.php');}

## Описание программного модуля «Удаление данных»

Данный модуль позволяет удалять информацию на сайте. Для этого необходимо войти в панель управления. После нужно перейти во вкладку, в которой хотите изменить информацию и нажать на кнопку «delete». Форма для удаления информации на странице «Категории» показана на рисунке 5.12.

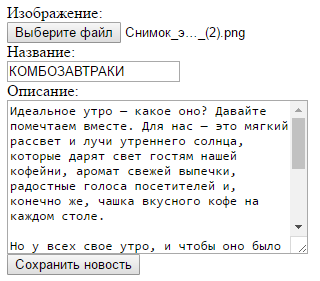


Рисунок 5.12 – Форма удаления информации

Данные редактируются в таблице categories. Таблица categories показана на рисунке 5.13.

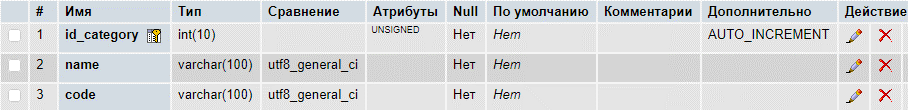


Рисунок 5.13 – Таблица categories

Скрипт редактирования информации:

$id=$\_GET['id'];

$result = mysqli\_query($link, "SELECT id, zagolovok, txt1, txt FROM o\_kompanii WHERE id='$id'");

$row = mysqli\_fetch\_array($result);

if (isset($\_POST['save'])){

$zagolovok=strip\_tags(trim($\_POST["zagolovok"]));

$txt1=strip\_tags(trim($\_POST['txt1']));

$txt=strip\_tags(trim($\_POST['txt']));

mysqli\_query($link, "UPDATE o\_kompanii SET zagolovok='$zagolovok', txt1='$txt1', txt='$txt' WHERE id='$id'");

header('Location: o\_kompanii.php');}

## Описание программного модуля «Добавление в корзину»

Данный модуль позволяет добавить выбранный товар в корзину. Для этого необходимо войти на сайт, выбрать жанр который хотите приобрести, нажать купить и оно автоматически добавляется в корзину. Скрипт добавления в корзину:

Ы$id=$\_GET['id'];

mysqli\_query($link, "DELETE FROM karty\_a2 WHERE id='$id'");

mysqli\_close($link);

header('Location: karty\_abonement.php');

# Тестирование ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

* 1. Модульное тестирование

Модульное тестирование проверяет функциональность и ищет дефекты в частях приложения, которые доступны и могут быть протестированы по-отдельности (модули программ, объекты, классы, функции и т.д.). Все найденные дефекты, как правило, исправляются в коде без формального их описания в системе менеджмента багов (Bug Tracking System).

Протестируем модуль добавление данных. На рисунке 6.1 представлена форма добавления информации.

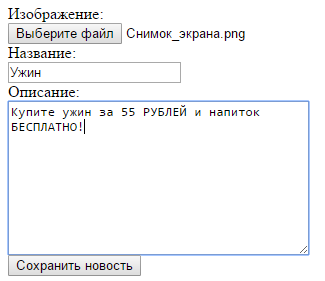


Рисунок 6.1 – Форма добавления информации

Данная форма добавляет новости на сайт. В таблицу novosty заносятся: изображение, название и описание. Таблица novosty показана на рисунке 6.2.

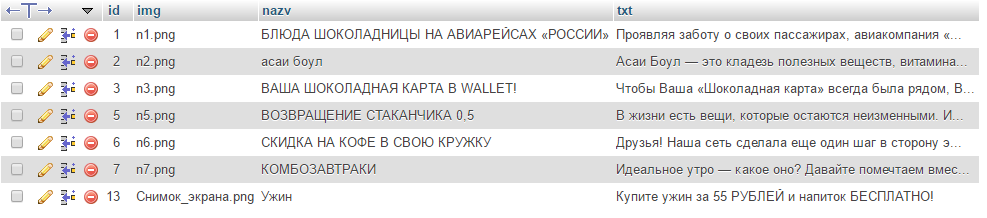


Рисунок 6.2 – Таблица novosty

После занесения данных в таблицу, информация отображается на странице, которую может просматривать клиент. Страница «Новости» показана на рисунке 6.3.

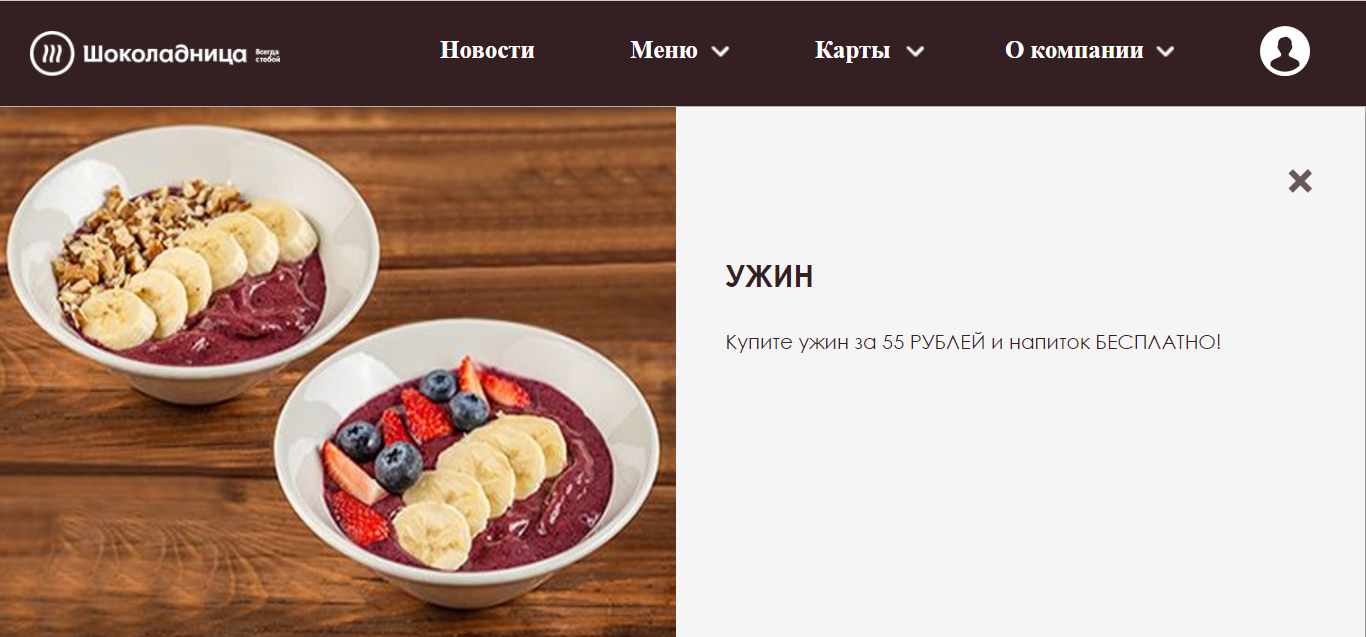


Рисунок 6.3 – Страница «Новости»

Аналогично были проверены и протестированы остальные модули:

* модуль регистрации пользователя;
* модуль редактирования информации на странице;
* модуль удаления информации;
* модуль оформления заказов.
  1. Тестирование интеграции

Интеграционное тестирование предназначено для проверки связи между компонентами, а также взаимодействия с различными частями системы (операционной системой, оборудованием либо связи между различными системами).

Уровни интеграционного тестирования

* Компонентный интеграционный уровень (Component Integration testing) Проверяется взаимодействие между компонентами системы после проведения компонентного тестирования.
* Системный интеграционный уровень (System Integration Testing) Проверяется взаимодействие между разными системами после проведения системного тестирования.

Подходы к интеграционному тестированию:

* Снизу вверх (Bottom Up Integration)
* Сверху вниз (Top Down Integration)
* Большой взрыв ("Big Bang" Integration)

Снизу вверх (Bottom Up Integration) Все низкоуровневые модули, процедуры или функции собираются воедино и затем тестируются. После чего собирается следующий уровень модулей для проведения интеграционного тестирования. Данный подход считается полезным, если все или практически все модули, разрабатываемого уровня, готовы. Также данный подход помогает определить по результатам тестирования уровень готовности приложения.

Сверху вниз (Top Down Integration) Вначале тестируются все высокоуровневые модули, и постепенно один за другим добавляются низкоуровневые. Все модули более низкого уровня симулируются заглушками с аналогичной функциональностью, затем по мере готовности они заменяются реальными активными компонентами. Таким образом мы проводим тестирование сверху вниз.

Большой взрыв ("Big Bang" Integration) Все или практически все разработанные модули собираются вместе в виде законченной системы или ее основной части, и затем проводится интеграционное тестирование. Такой подход очень хорош для сохранения времени. Однако если тест кейсы и их результаты записаны не верно, то сам процесс интеграции сильно осложнится, что станет преградой для команды тестирования при достижении основной цели интеграционного тестирования.

Таблица 6.1 – Тестирование интеграции

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | index.php | holod\_napoi.php | karty\_abonement.php | karty\_bonus.php | karyera.php | koffe\_tea.php | kontakty.php | lanch.php | menu.php | novosty.php | o\_kompanii.php | pizza.php | ujin.php | zavtrak.php |
| index.php |  | + | + | + | + | + | + | + | + | + | + | + | + | + |
| holod\_napoi.php | + |  | + | + | + | + | + | + | + | + | + | + | + | + |
| karty\_abonement.php | + | + |  | + | + | + | + | + | + | + | + | + | + | + |
| karty\_bonus.php | + | + | + |  | + | + | + | + | + | + | + | + | + | + |
| karyera.php | + | + | + | + |  | + | + | + | + | + | + | + | + | + |
| koffe\_tea.php | + | + | + | + | + |  | + | + | + | + | + | + | + | + |
| kontakty.php | + | + | + | + | + | + |  | + | + | + | + | + | + | + |
| lanch.php | + | + | + | + | + | + | + |  | + | + | + | + | + | + |
| menu.php | + | + | + | + | + | + | + | + |  | + | + | + | + | + |
| novosty.php | + | + | + | + | + | + | + | + | + |  | + | + | + | + |
| o\_kompanii.php | + | + | + | + | + | + | + | + | + | + |  | + | + | + |
| pizza.php | + | + | + | + | + | + | + | + | + | + | + |  | + | + |
| ujin.php | + | + | + | + | + | + | + | + | + | + | + | + |  | + |
| zavtrak.php | + | + | + | + | + | + | + | + | + | + | + | + | + |  |