**Министерство образования Иркутской области**

Государственное бюджетное профессиональное образовательное учреждение Иркутской области

«Иркутский авиационный техникум»

(ГБПОУИО «ИАТ»)

|  |  |
| --- | --- |
| **ДП.09.02.03.22.182.11.ПЗ** | УТВЕРЖДАЮ  Зам. директора по УР, к.т.н.  \_\_\_\_\_\_\_\_\_\_\_Е.А. Коробкова |

ИНФОРМАЦИОННАЯ СИСТЕМА «СЕРВИС ОБЪЯВЛЕНИЙ»

|  |  |  |
| --- | --- | --- |
| Нормоконтролер: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (О.Ю. Безносова) |
| Консультант по экономической части: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (А.А. Белова) |
| Руководитель: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (Е.А. Матвеева) |
| Студент: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись, дата) | (В.А. Ильин) |

Иркутск 2022

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc96083502)

[1 Предпроектное исследование 5](#_Toc96083503)

[1.1 Исследование предметной области 5](#_Toc96083504)

[1.2 Анализ инструментов разработки 6](#_Toc96083505)

[1.3 Обоснование выбора программного продукта 8](#_Toc96083506)

[2 Проектирование 11](#_Toc96083507)

[2.1 Архитектура программного обеспечения 11](#_Toc96083508)

[2.2 Функциональное проектирование 13](#_Toc96083509)

[2.3 Проектирование базы данных 19](#_Toc96083510)

[2.4 Проектирование пользовательского интерфейса 24](#_Toc96083511)

[3 Реализация программного обеспечения 28](#_Toc96083512)

[3.1 Кодирование программного обеспечения 28](#_Toc96083513)

[4 Документирование программного обеспечения 31](#_Toc96083514)

[4.1 Руководство пользователя программного обеспечения 31](#_Toc96083515)

[5 Стоимость разработки и внедрения программного продукта 38](#_Toc96083522)

[5.1 Организационно-экономическое обоснование проекта 38](#_Toc96083523)

[5.2 Расчет затрат на разработку программного продукта 38](#_Toc96083524)

[6.1 Расчет затрат на внедрение программного продукта 41](#_Toc96083525)

[6.2 Основные выводы 42](#_Toc96083526)

[Заключение 43](#_Toc96083527)

[Список использованных источников 44](#_Toc96083528)

[Приложение А - Листинг маршрутов 46](#_Toc96083529)

[Приложение Б - Листинг javascript 50](#_Toc96083530)

Введение

Темой дипломного проекта является информационная система «Сервис объявлений».

В современном мире множество людей пытаются завести пассивный доход, открывая свой личный бизнес, или быть самозанятыми. В любом из случаев им необходима реклама, какое-то объявление, которое поможет привлечь новых клиентов или сотрудников.

Объявлением называют официальное сообщение, уведомление о чём-либо, сделанное в устной (реже письменной) форме.

Самое главное при работе с объявлениями – это правильное хранение всех данных. Все необходимые данные удобно хранить в таблицах. Информация в базе данных (далее БД) тоже представляется в виде таблиц, что позволяет организовать хранение всех необходимых данных объявлений и пользователей. Качество и надежность результатов вышеназванных действий служит основой эффективной работы информационной системы. Актуальным является выставление любых объявлений, а не только определенной тематики. Данную базу можно применить в любом сервисе объявлений, также имеется возможность её доработать. Для этого нужна общая база данных, включающая всю необходимую информацию. И программа, которая будет понятна и автоматизирует работу с базой данных.

Для реализации была поставленная следующая цель: разработать информационную систему «Сервис объявлений».

Для достижения цели были поставлены следующие задачи:

* + Провести предпроектное исследование.
  + Составить техническое задание на разработку программного продукта в соответствии с ГОСТ.
  + Провести проектирование программного продукта.
  + Реализовать программный продукт.
  + Разработать документы для программного продукта.
  + Рассчитать стоимость разработки и внедрения программного продукта.

База данных должна быть спроектирована так, чтобы обеспечить хранение всех необходимых данных, имея при этом максимально упрощенную структуру. Структура базы данных должна обеспечивать целостность и корректность информации.

1. Предпроектное исследование
   1. Исследование предметной области

Предметной областью дипломного проекта является информационная система «Сервис объявлений» (далее ИС).

Рекламное объявление — это краткое изложение основной информации о товаре или услуге, которое содержит ключевые фразы и способствует привлечению новых клиентов. Рекламное объявление распространяет тексты с предложением товаров и услуг.

Сервис объявлений (электронная доска объявлений) — сайт, на котором размещаются объявления. Сервис объявлений функционально подобна обыкновенной: это сайт, где каждый желающий может вывесить своё объявление, а все посетители сайта — прочитать его. Электронная доска объявлений, как правило, поделена на несколько тематических разделов, согласно содержанию объявлений.

Информационная система — система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Все пользователи информационной системы желающие выставить объявление, должны пройти регистрацию в этой ИС, но просматривать объявления может и не зарегистрированный пользователь. Также пользователь имеет право просматривать, редактировать и удалять свои объявления, поэтому в ИС имеется личный кабинет пользователя.

Таким образом, в функционирование сервиса объявлений входит:

- регистрация пользователей;

- данные об объявлениях;

- манипулирования данными объявлений;

- панель администратора;

- личный кабинет пользователя.

* 1. Анализ инструментов разработки

Для разработки приложения можно использовать следующие программные продукты: SQL, Django, PyCharm, Python, РНР, Ruby, JavaScript, JetBrains PhpStorm, JetBrains WebStorm, Notepad++, NodeJS, MySQL, Oracle, PostgreSQL, Microsoft Visual Studio, MySQL Workbench, PhpMyAdmin, Microsoft Visio, Draw IO, Figma.

SQL — «Язык структурированных запросов» декларативный язык программирования, применяемый для создания, модификации и управления данными в реляционной базе данных, управляемой соответствующей системой управления базами данных.

Python — это высокоуровневый язык программирования, который используется в различных сферах IT, таких как машинное обучение, разработка приложений и другие.

PyCharm — интегрированная среда разработки для языка программирования Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django.

Django — свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования МVС.

Microsoft Visual Studio — линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств.

РНР: Hypertext Preprocessor — «Инструменты для создания персональных веб-страниц» -серверный язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений.

Notepad++ — свободный текстовый редактор с открытым исходным кодом для Windows с подсветкой синтаксиса большого количества языков программирования и разметки, а также языков описания аппаратуры VНDL и Verilog.

JetBrains PhpStorm — представляет собой интеллектуальный редактор для РНР, НТМL и JavaScript, с возможностями анализа кода на лету, предотвращения ошибок в коде, автоматизированными средствами рефакторинга для РНР и JavaScript.

JetBrains WebStorm — интегрированная среда разработки на JavaScript, CSS & НТМL от компании JetBrains, разработанная на основе платформы IntelliJ IDEA. WebStorm обеспечивает автодополнение, анализ кода на лету, навигацию по коду, рефакторинг, отладку, и интеграцию с системами управления версиями

JavaScript — это полноценный динамический язык программирования, который применяется к НТМL документу, и может обеспечить динамическую интерактивность на веб-сайтах.

NodeJS — программная платформа, превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.j s добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API (написанный на С++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кoдa.

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle

PhpMyAdmin — веб-приложение с открытым кодом, написанное на языке РНР и представляющее собой веб-интерфейс для администрирования CYБД MySQL.

PostgreSQL — свободная объектно-реляционная система управления базами данных. Существует в реализациях для множества UNIХ-подобных платформ, включая AIX, различные ВSD-системы, НР-UХ, IRIX, Linux, macOS, Tru64, QNX, а также для Microsoft Windows.

Oracle Database или Oracle RDBMS — объектно-реляционная система управления базами данных компании Oracle

MySQL Workbench — инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию БД в единое бесшовное окружение для системы баз данных MySQL.

Draw.io — это сервис, предназначенный для формирования диаграмм и схем. Сервис разделён на три части — меню, панель объектов и сам документ.

Figma — кросс-платформенный онлайн-сервис для дизайнеров интерфейсов и веб-разработчиков. Разработка интерфейсов происходит в онлайн-приложении.

У Figma две ключевые особенности: доступ к макету прямо из окна браузера и возможность совместной работы над документами.

* 1. Обоснование выбора программного продукта

Для разработки ИС было решено использовать PHPStorm, PHPMyAdmin и MySQL. Для проектирования ИС были выбраны Draw.io, Figma, Microsoft Visio.

Для выбора инструментов разработки были произведены сравнения, которые показаны в таблице 1, таблице 2, таблице 3, и таблице 4.

Таблица 1 — Сравнение IDE для разработки программного продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название IDE | Visual studio code | Notepad++ | JetBrains PhpStorm | JetBrains  WebStorm |
| Стоимость | Свободное ПО | Свободное ПО | Студенческая лицензия | Студенческая лицензия |
| Автосохранение | + | - | + | + |
| Автодополнение | + | - | + | + |
| Интеграция с системами управления версиями (git) | -  (возможно добавить плагин) | -  (возможно добавить плагин) | + | + |
| Поиск по функции в коде | - | - | + | + |
| Встроенные языки программирования | -  (возможно добавить плагин) | -  (возможно добавить плагин) | PHP, HTML, JavaScript, CSS, SQL | HTML, JavaScript, CSS, SQL |

Таким образом, из представленных IDE можно выделить несколько инструментов, способных облегчить и ускорить разработку программного обеспечения, а именно JetBrains PhpStorm, JetBrains WebStorm, так как программный продукт будет реализован с помощью web - технологий. Исходя из того, что для разработки необходим язык программирования PHP, то в качестве IDE был выбран JetBrains PhpStrorm.

Таблица 2 — Сравнение языков программирования для разработки программного продукта

|  |  |  |  |
| --- | --- | --- | --- |
| Название языка программирования | PHP | Ruby | Python |
| Простота обучаемости | + | - | + |
| Читабельность | + | - | + |
| Динамическая типизация | + | + | + |
| Интеграция баз данных | + | + | - |
| Объектно-ориентированные  возможности | + | + | + |

Вывод, из представленных языков программирования можно выделить РНР, так как именно этот язык более перспективный в разработке web -приложения.

Таблица 3 — Сравнение СУБД для разработки программного продукта.

|  |  |  |  |
| --- | --- | --- | --- |
| Название СУБД | Oracle | MySQL | PostgreSQL |
| Язык программирования | С, С++, Java, Ruby, Objective | Delphi, С, С++, Java, Perl, РНР, Pvthon, Ruby, SQL | pgSQL, РНР, Python, Ruby и др. |
| Скорость обработки | Быстрая | Быстрая | Медленная |
| Операционная система Windows | Поддерживается | Поддерживается | Поддерживается |
| Исходный код | Закрытый | Открытый | Открытый |
| Лицензия | Коммерческая | Свободная | Свободная |

В результате сравнения, из представленных СУБД можно выделить MySQL, так как оно имеет свободную лицензию и подходит для создания web – приложения.

Таблица 4 — Сравнение средств проектирования

|  |  |  |  |
| --- | --- | --- | --- |
| Название средства проектирования | Draw.io | Visio | NClass |
| Поток данных диаграмм | + | + | - |
| Er - диаграмма | + | + | - |
| Проверка связей | + | + | - |
| Удобство использования | + | + | - |
| Лицензия | свободная | коммерческая | свободная |

Следовательно, из представленных средств проектирования, было принято решение использовать Draw.io, так как оно имеет свободную лицензию и большое количество UМL средств.

1. Проектирование

Перед проектированием информационной системы было разработано техническое задание, на основе которого производилось проектирование информационной системы «Система объявлений» Техническое задание приложено к пояснительной записке.

* 1. Архитектура программного обеспечения

Архитектура программного обеспечения – совокупность важнейших решений об организации программной системы. Архитектура включает:

1. выбор структурных элементов и их интерфейсов, с помощью которых составлена система, а также их поведения в рамках сотрудничества структурных элементов;
2. соединение выбранных элементов структуры и поведения во всё более крупные системы;
3. архитектурный стиль, который направляет всю организацию – все элементы, их интерфейсы, их сотрудничество и их соединение.

Информационная система использует клиент-серверную архитектуру, что говорит о том, что все действия на клиентской части пользователя над данными обрабатываются сервером на серверной части.

Клиент-серверная архитектура позволяет значительно снизить нагрузку у пользователя, что позволяет запускать информационную систему практически на любом устройстве.

На рисунке 1 демонстрируется архитектура веб-приложений, которая соответствует разрабатываемой информационной системе.

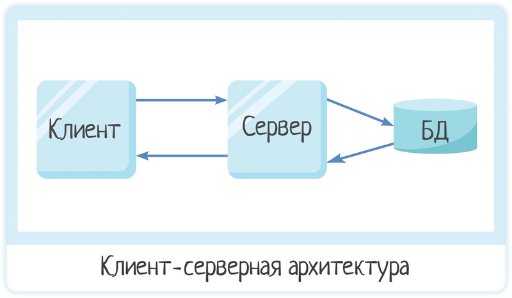


Рисунок 1 – Клиент-серверная архитектура веб-приложения

Браузер клиента отправляет запрос на подключение к странице веб-сервер. Веб-сервер отправляет запрос в веб-приложение, которое запрашивает обработку PHP. PHP может запросить подключение к MySQL, на что Apache отправляет запрос к MySQL, который по запросу берет данные из БД и отправляет обратно на веб-сервер, после чего PHP продолжает обработки. Когда обработка закончит, Apache генерирует необходимую URL и отправляет ответ клиенту, где браузер выполняет скрипты.

Информационная система использует архитектурную модель MVC (Model View Controller) представленную на рисунке 2.

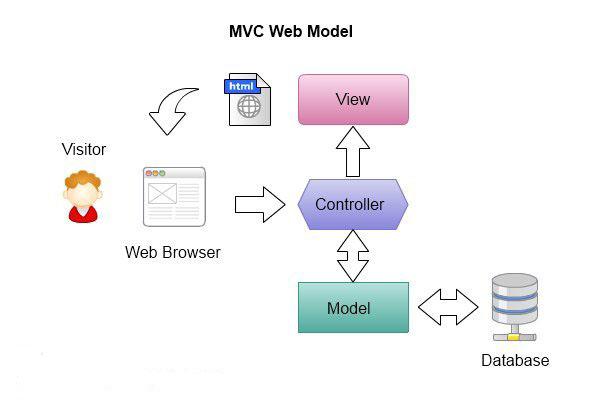


Рисунок 2 – Архитектурная модель MVC

Архитектура MVC описывает простой способ построения структуры программного продукта, целью которого является отделение бизнес-логики от пользовательского интерфейса. В результате, программный продукт легче масштабируется, тестируется, сопровождается и конечно же реализуется.

В архитектуре MVC модель предоставляет данные и правила бизнес-логики, представление отвечает за пользовательский интерфейс, а контроллер обеспечивает взаимодействие между моделью и представлением.

Модель — содержит бизнес-логику приложения и включает методы выборки, обработки и предоставления конкретных данных.

Вид — используется для задания внешнего отображения данных, полученных из контроллера и модели.

Контроллер — связующее звено, соединяющее модели, виды и другие компоненты в рабочее приложение.

* 1. Функциональное проектирование

Функциональное проектирование – это подход к проектированию, который нацелен, прежде всего, на создание эффективно работающего объекта. Функциональное проектирование позволяет раскрыть все функции системы.

Выполнение требуемой функции – главная цель и основа разработки объекта. Во внимание принимаются, прежде всего, функциональные показатели качества и показатели надёжности.

Контекстная диаграмма – это модель, представляющая систему как набор иерархических действий, в которой каждое действие преобразует некоторый объект или набор объектов.

На контекстной диаграмме на рисунке 3 показаны входные данные, управление механизм, выходные данные и функция.



Рисунок 3 – Контекстная диаграмма

Для демонстрации деталей в процессе необходимо спроектировать диаграмму декомпозиции.

Диаграмма декомпозиции – это разбиение функции, т.е. работа информационной системы «Сервис объявлений».

На рисунке 4 показана диаграмма декомпозиции, которая расписывает работу раздела объявления информационной системы из контекстной диаграммы. На этой диаграмме показаны 3 функций — это «Авторизация», «Список объявлений», «Манипулирование объявлениями». А также на этой диаграмме присутствуют входные данные, выходные данные, управления и механизмы.

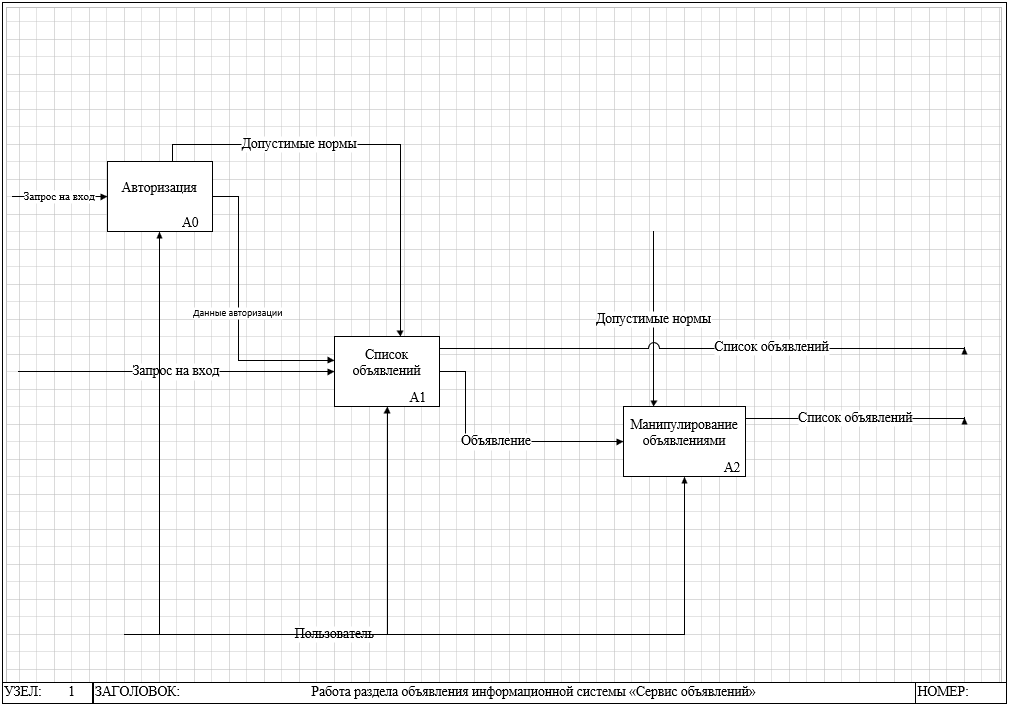


Рисунок 4 – Диаграмма декомпозиций работы раздела объявления

На рисунке 5 показана диаграмма декомпозиции, которая расписывает работу раздела города информационной системы из контекстной диаграммы. На этой диаграмме показаны 3 функций — это «Авторизация», «Список городов», «Манипулирование данными модели города».

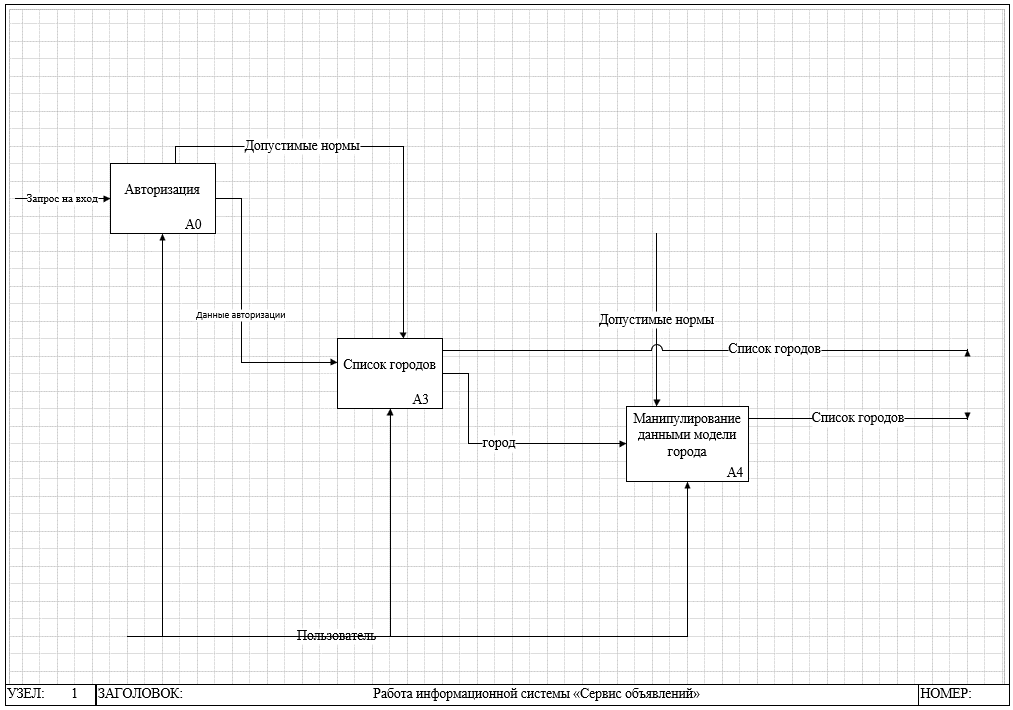


Рисунок 5 – Диаграмма декомпозиций работы раздела города

На рисунке 5 показана диаграмма декомпозиции, которая расписывает работу раздела города информационной системы из контекстной диаграммы. На этой диаграмме показаны 3 функций — это «Авторизация», «Список пользователей», «Манипулирование данными пользователей».

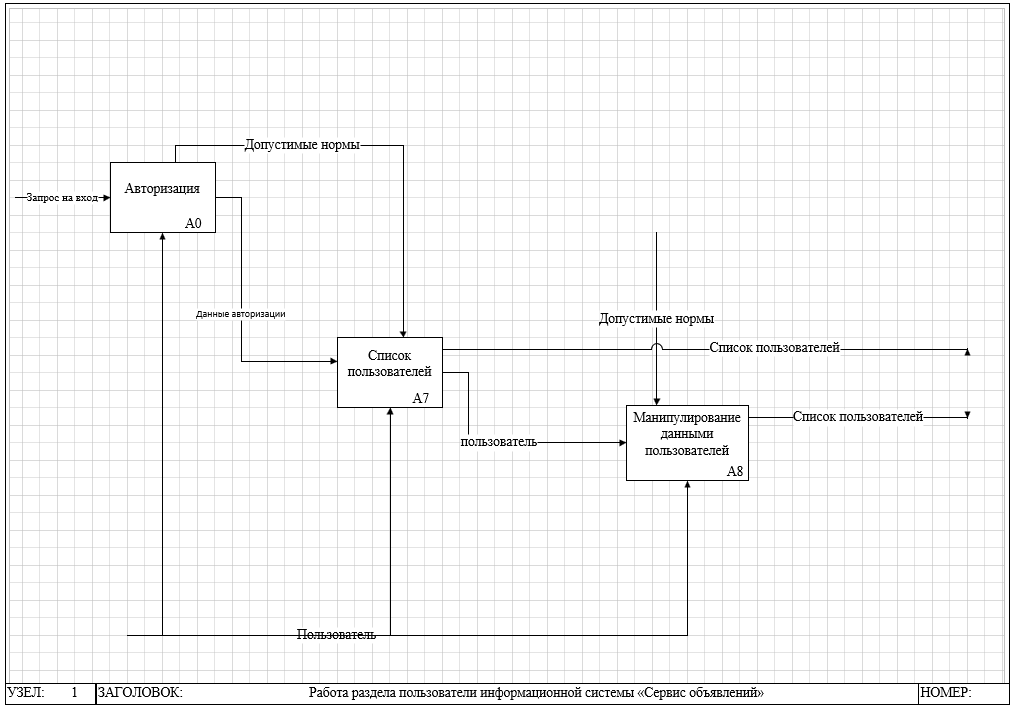


Рисунок 6 – Диаграмма декомпозиций работы раздела пользователи

На рисунке 7 показана диаграмма IDEF3, которая расписывает работу приложения из рисунка 4.

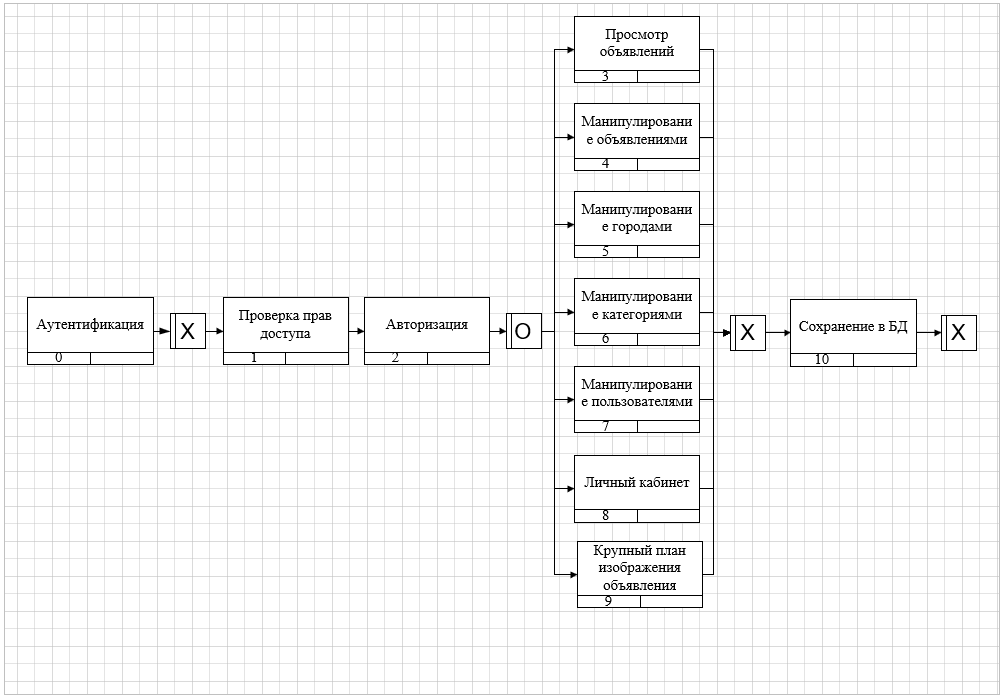


Рисунок 7 –IDEF3 диаграмма

На рисунке 8 показана диаграмма потоков данных, которая описывает передачу данных между разными функция и объектами приложения.

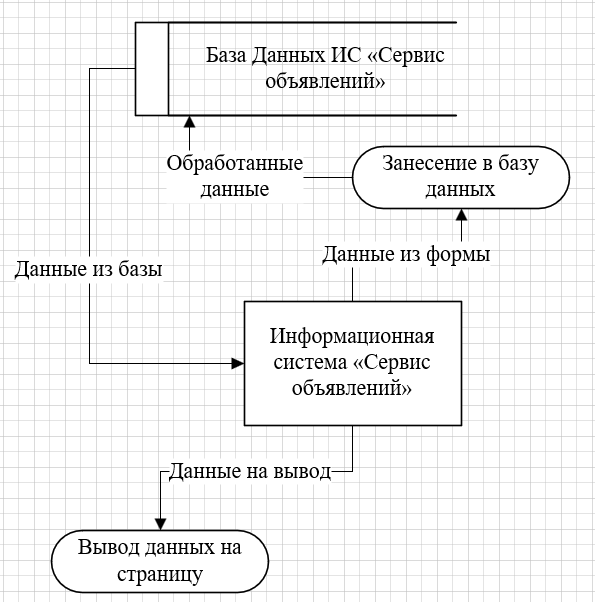


Рисунок 8 – Диаграмма потоков данных

Также во время функционального проектирования была построенна диграмма прецендентов, или как ещё называют диграмма вариантов использования, для выявления ролей пользователей и их права доступа. На рисунке 9 представлена диаграмма прецендентов информационной системы «Сервис объявлений».

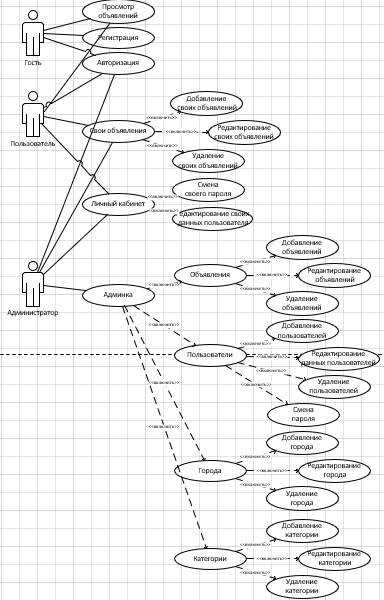


Рисунок 9 – Диаграмма прецендентов

Роль пользователя «Гость» имеет доступ только к таким действиям как: просмотр объявлений, регистрация и авторизация (рисунок 10).

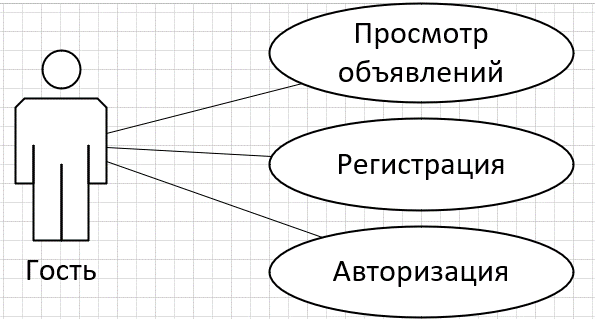


Рисунок 10 – Диаграмма прецендентов - Гость

Роль пользователя «Пользователь» имеет доступ к более широкому спектру лействий, чем «Гость», а именно к таким как: просмотр объявлений, авторизация, свои объявления, личный кабинет. Свои объявления включает в себе и другие действия, а именно: добавление своих объявлений, редактирование своих объявлений, удаление своих объявлений. Личный кабинет также включает в себя другие действия такие как: смена своего пароля и редактирование своих данных пользователя (рисунок 11).

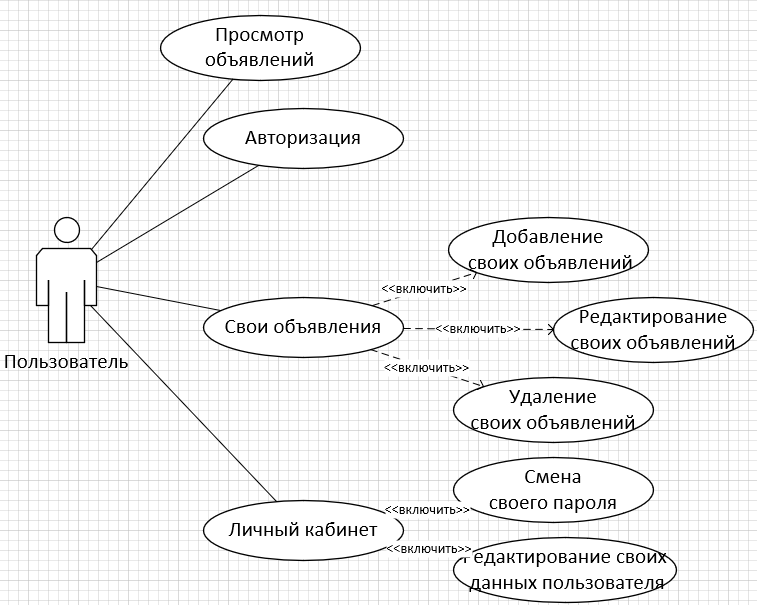


Рисунок 11 – Диаграмма прецендентов - Пользователь

Роль пользователя «Администратор» имеет доступ к тем же действия, что и «Пользователь», но добавляется группа действий «Админка». «Админка» включает в себя 4 групповых действия: объявления, пользователи, города и категории. Каждое из этих групп включает в себя действия, связанные с манипулированием данными модели группы (рисунок 12).

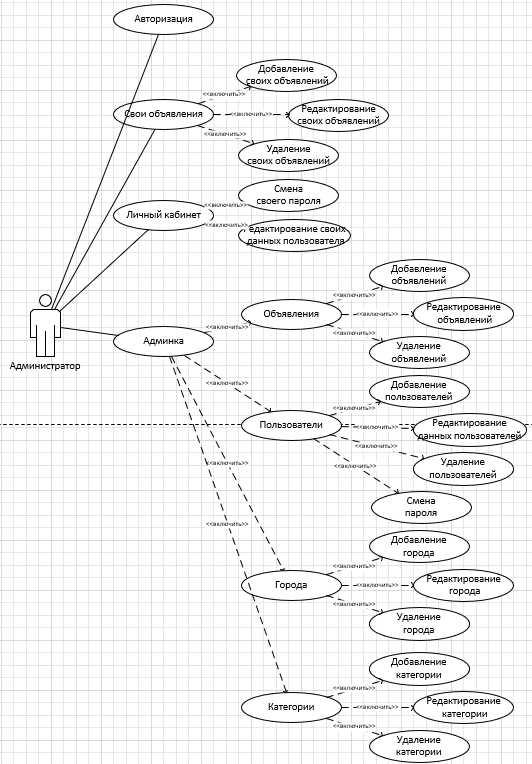


Рисунок 12 – Диаграмма прецендентов - Администратор

Подробное функциональное проектирование позволяет глубже исследовать проектную область и цель проекта, что может позволить сэкономить время на разработке информационной системы, а также позволит избежать перенаписание кода.

* 1. Проектирование базы данных

Проектирование информационной системы происходит при помощи CASE средств, которые позволяют быстро создавать схемы и реляционные модели программ.

Для лучшего изучения предметной области и проектирования базы данных. Были построены 3 схемы модели базы данных: концептуальная, логическая и физическая схемы.

Концептуальная модель — это отражение предметной области, для которой разрабатывается база данных. На рисунке 13 представлена концептуальная модель базы данных ИС. Для построения модели были выявлены 5 сущностей: пользователи, посты (объявления), картинки постов (объявлений), города и категории.

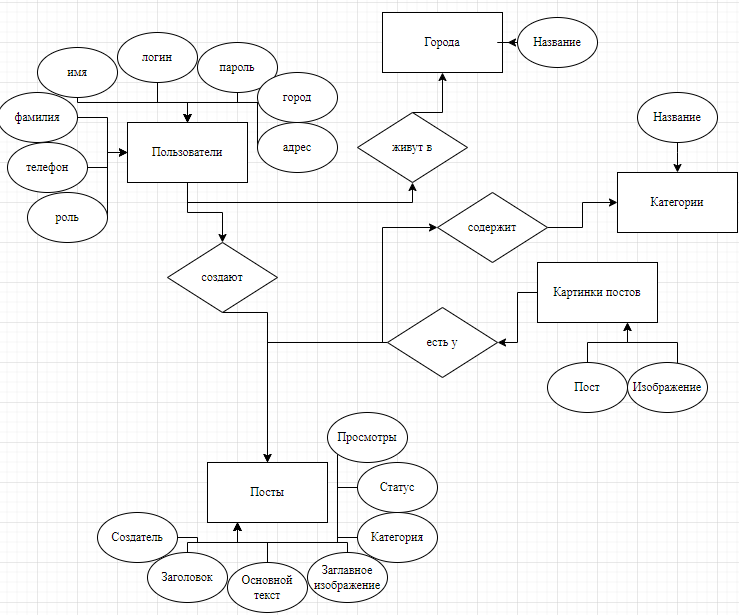


Рисунок 13 – концептуальная модель базы данных

Логическая (даталогическая) модель представляет собой модель базы данных, которая не привязана к конкретной СУБД. В ней выделяют основные объекты БД и определяют связи между этими объектами. Полученная логическая модель представлена на рисунке 14.

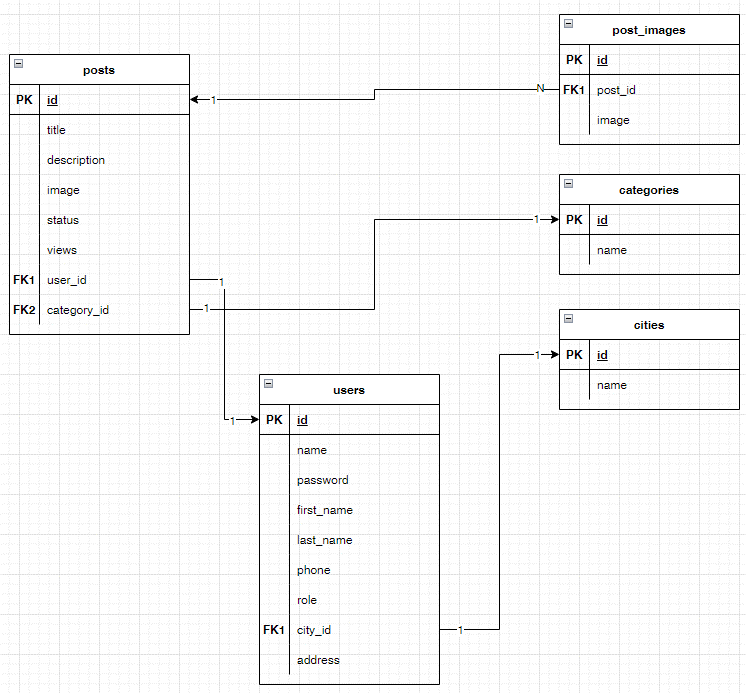


Рисунок 14 – логическая модель базы данных

Модель сущность-связь или схема баз данных – модель данных, позволяющая описывать концептуальные схемы предметной области.

Физическая модель базы данных — это модель данных, которая определяет, каким образом представляются данные, и содержит все детали, необходимые СУБД для создания базы данных. На рисунке 15 изображена физическая модель базы данных ИС.

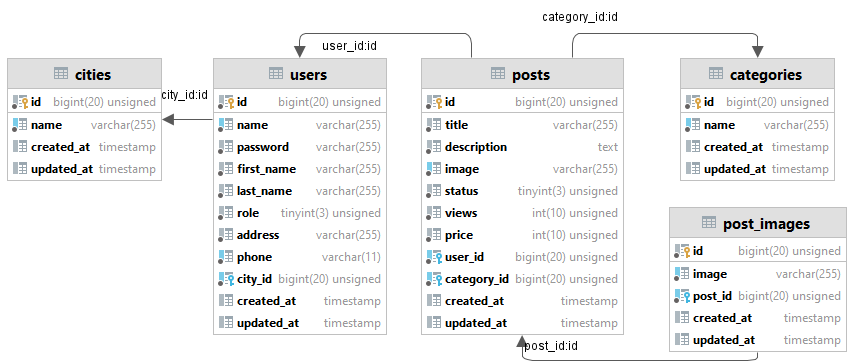


Рисунок 15 – физическая модель базы данных

База данных представляет собой 5 связанных таблиц. Для создания связей между таблицами используются внешние ключи.

Были выявлены следующие сущности и их атрибуты представленные в таблицах от 5 до 9.

Таблица 5 — Атрибуты сущности posts

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint(20) | Идентификатор объявления |
| title | varchar (255) | Заголовок объявления |
| description | varchar (255) | Описание объявления |
| image | varchar (255) | Главное изображение объявления |
| status | tinyint (3) | Статус объявления |
| views | int(10) | Количество просмотров |
| price | int(10) | Стоимость товара по объявлению |
| user\_id | bigint(20) | Идентификатор пользователя |
| category\_id | bigint(20) | Идентификатор категории |

Таблица 6 — Атрибуты сущности categories

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор категории |
| name | varchar (255) | Название категории |

Таблица 7 — Атрибуты сущности cities

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор города |
| name | varchar (255) | Название города |

Таблица 8 — Атрибуты сущности post\_images

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint | Идентификатор изображений объявления |
| post\_id | bigint | Идентификатор объявления |
| image | varchar (255) | Путь до изображения |

Таблица 9 — Атрибуты сущности users

|  |  |  |
| --- | --- | --- |
| Атрибут | Тип данных | Описание атрибута |
| id | bigint(20) | Идентификатор пользователя |
| name | varchar(255) | Логин пользователя |
| password | varchar(255) | Пароль пользователя |
| first\_name | varchar(255) | Имя пользователя |
| last\_name | varchar(255) | Фамилия пользователя |
| role | tinyint(3) | Роль пользователя |
| address | varchar(255) | Адрес пользователя |
| phone | varchar(11) | Телефон пользователя |
| city\_id | bigint(20) | Идентификатор города |

Нормальная форма — свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Третья нормальная форма, которую придерживается база данных, содержит в себе следующее требование:

– Отношение находится в 3НФ, когда находится во 2НФ и каждый не ключевой атрибут нетранзитивно зависит от первичного ключа. Проще говоря, второе правило требует выносить все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы в отдельные таблицы.

Исходя из всего выше сказанного, уровень нормализации базы данных равен 3.

* 1. Проектирование пользовательского интерфейса

Интерфейс приложения – это одна из важнейших частей проектирования программного обеспечения.

Интерфейсом информационной системы «Сервис объявлений» разделяется на 2 части: меню навигации, содержащее разделы приложения, и рабочую область. Проектирование интерфейса выполнено в Figma.

На рисунке 16 показан прототип страницы «Вход», который позволяет авторизоваться пользователю ИС.

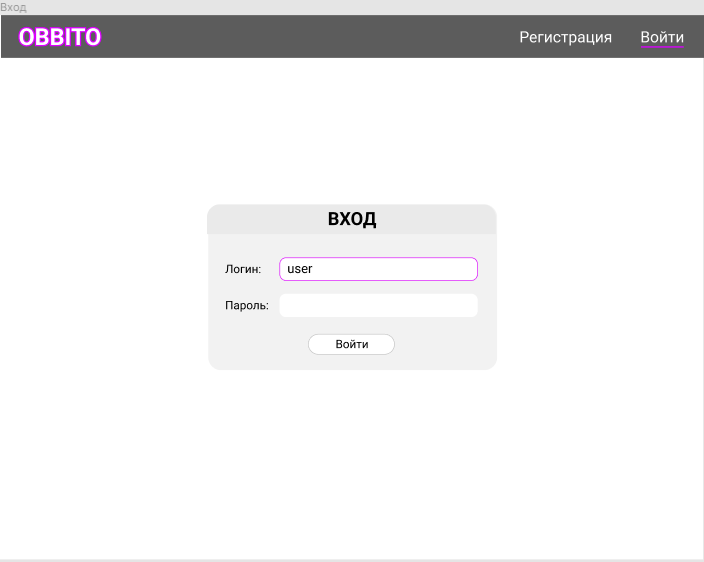


Рисунок 16 – Прототип страницы «Вход»

На рисунке 17 демонстрируется прототип главной страницы, на которой пользователь может увидеть недавно выставленные объявления. На эту страницу может войти и не авторизованный пользователь ИС.

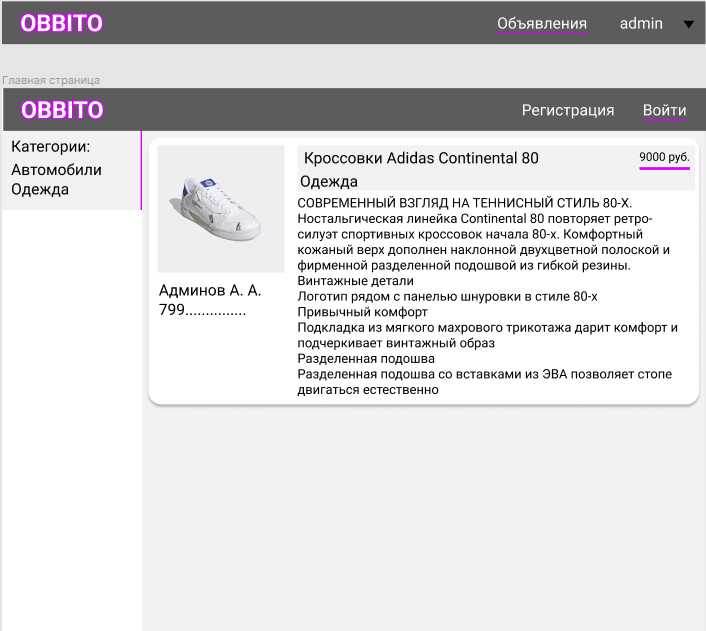


Рисунок 17 – Прототип главной страницы

На рисунке 18 демонстрируется прототип страницы «Добавление объявления» у администратора. На рабочем пространстве размещена форма для заполнения.

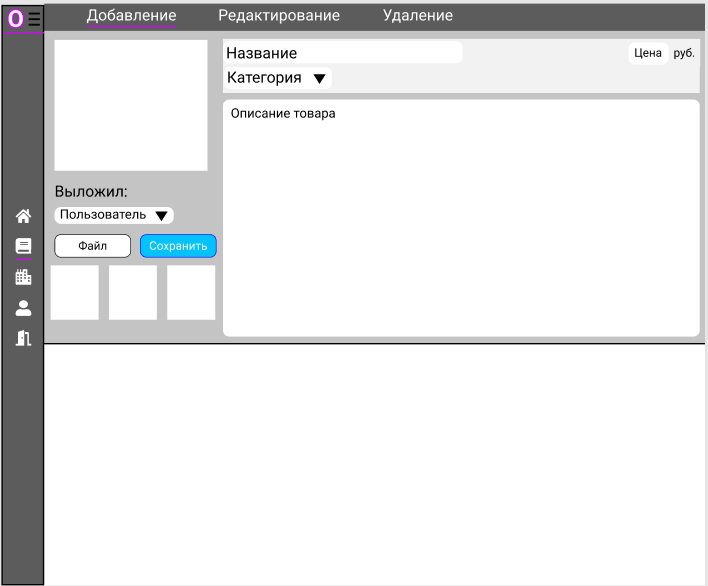


Рисунок 18 – Прототип страницы «Добавление объявления» у администратора

На рисунке 19 демонстрируется прототип страницы «Личный кабинет». На рабочем пространстве размещена заполненная форма с данными пользователя.

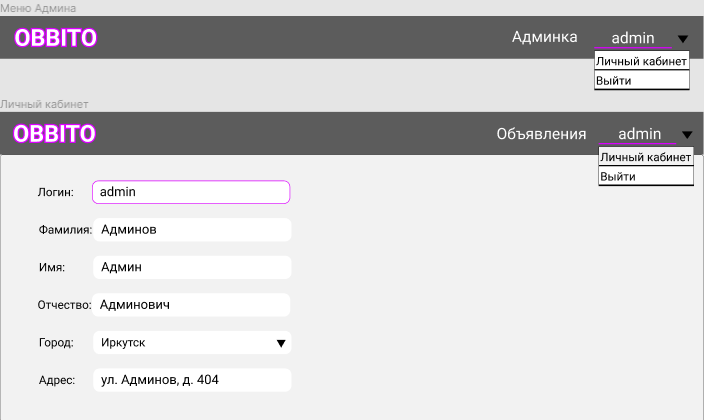


Рисунок 19 – Прототип страницы «Личный кабинет»

На рисунке 20 демонстрируется прототип страницы «Регистрация», которая позволяет зарегистрировать нового пользователя ИС.

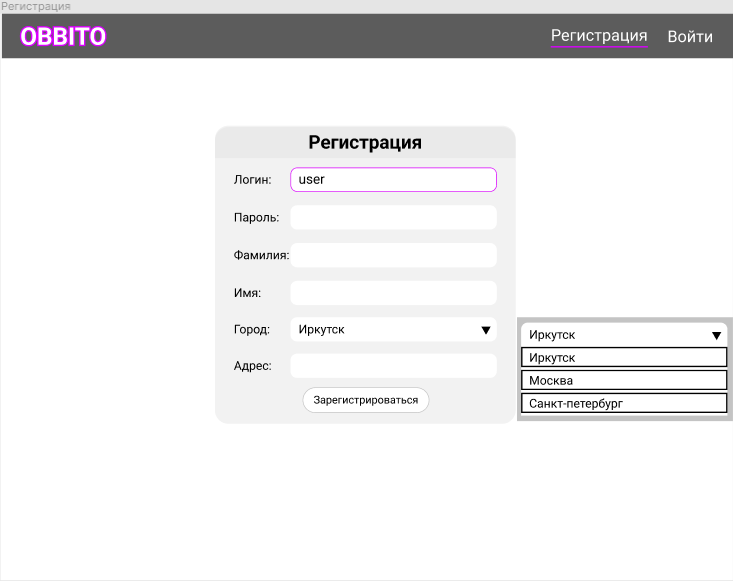


Рисунок 20 – Прототип страницы «Регистрация»

На рисунке 21 демонстрируется прототип страницы «Редактирование объявления» у администратора. Данная страница доступна только пользователям с соответствующими правами доступа. На рабочем пространстве размещены заполненная форма для редактирования данных объявлений.

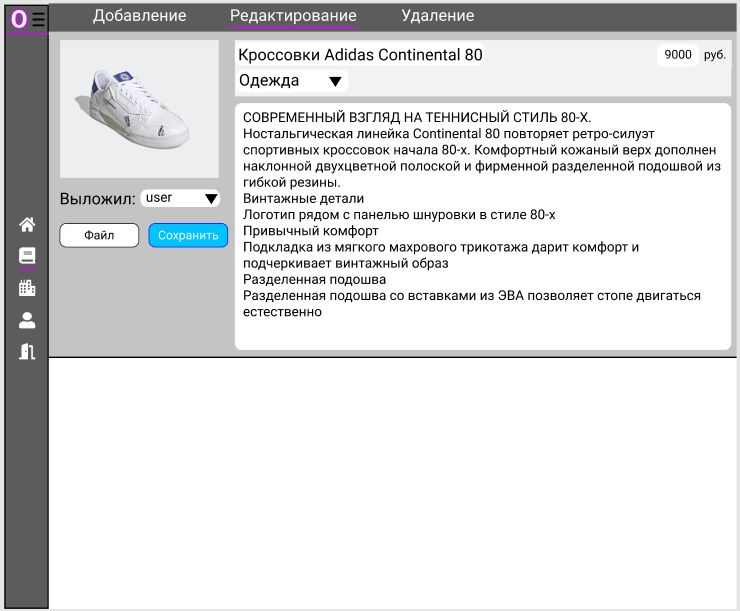


Рисунок 21 – Прототип страницы «Редактирование объявления» у администратора

Проектирование интерфейса пользователя позволяет определиться с расположение объектов и данных на рабочем пространстве. Также позволяет определиться с количеством страниц и избежать конфликтов с заказчиков.

1. Реализация программного обеспечения
   1. Кодирование программного обеспечения

Суть реализации программного обеспечения заключается в кодировании – процесс написания программного кода. В связи с тем, что разрабатываемая информационная система основано на веб-технологиях, в ходе разработки было принято решение использовать:

* каскадную таблицу стилей CSS - для клиентской части;
* язык программирования JavaScript – для клиентской части;
* язык программирования PHP- для серверной части.

Благодаря каскадной таблицы стилей CSS информационная система «Сервис объявлений» имеет простой, но стильный внешний вид, который продемонстрирован на рисунке 22.

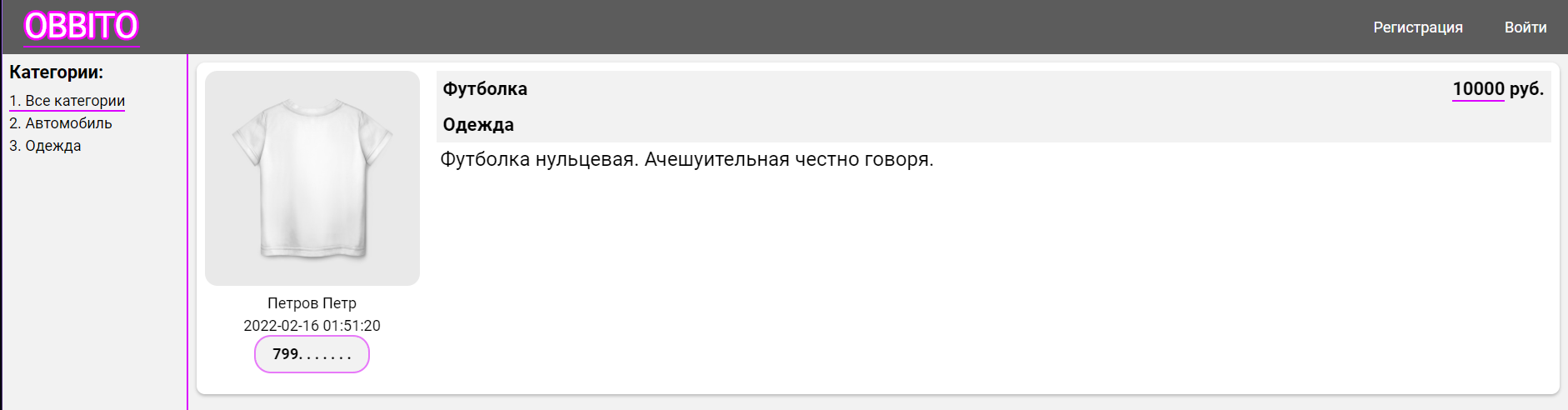


Рисунок 22 –Главная страница

JavaScript позволил организовать модальные окна в ИС, выпадающий список, предпросмотр загружаемых изображений. Листинг написанного кода представлен в приложении Б Листинг javascript.

Реализованная на PHP архитектура MVC позволила разделить логику от вывода информации пользователю. В архитектуре логика реализуется в моделях и контроллерах, а вывод информации в представлениях.

Контроллеры отвечают за обработку запросов пользователя и вызов соответствующих представлений. Контроллер представляет собой класс, который содержит методы обработки функций. Все контроллеры проекта находятся в папке /app/controllers. Самый крупный контроллер — это PostController. Он отвечает за работу с данными об объявлениях. На рисунке 23 представлен код функции insert из контроллера PostController.

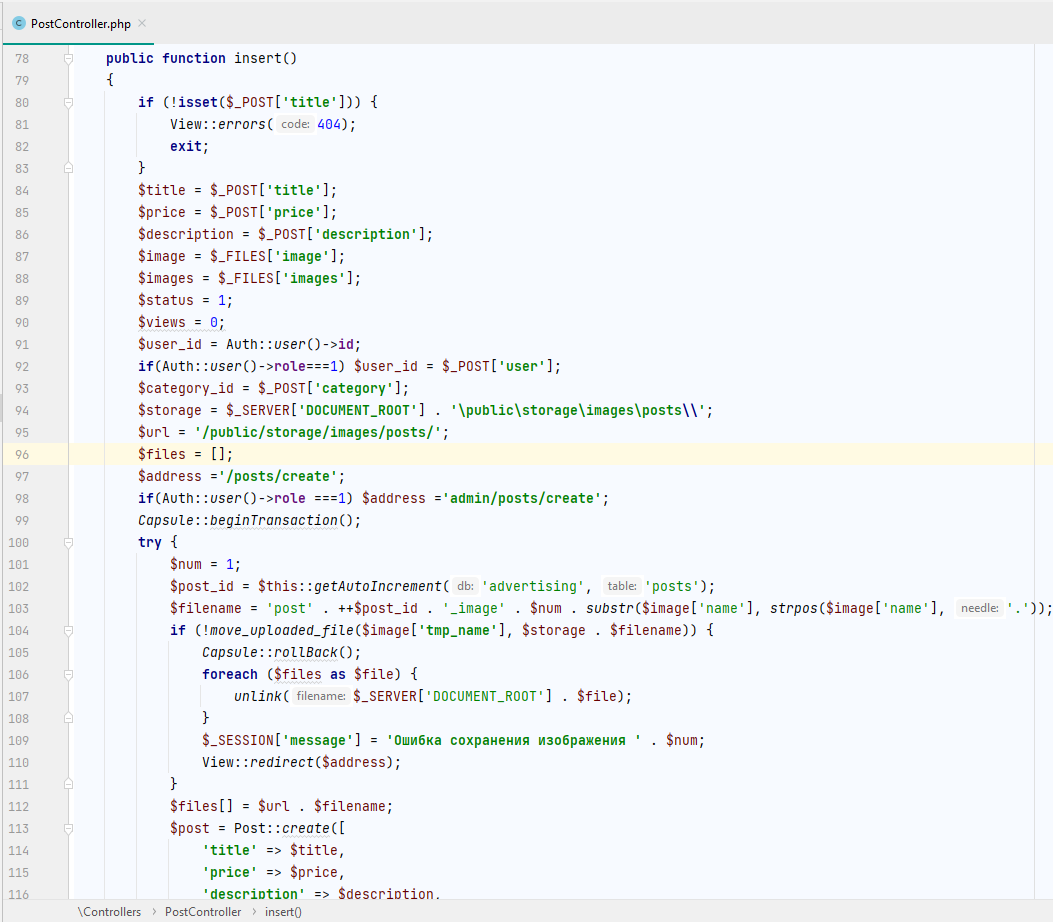


Рисунок 23 – Фрагмент контроллера PostController

Представления отвечают за показ результатов запроса пользователю. Каждое представление – это отдельная страница, содержащая html код, а также php вставки. Все представления находятся в папке /app/views. Представление главной страницы является одной из крупнейших представлений ИС. В ней данные из контроллера вставляются в шаблон, где формируется список категорий и объявлений. На рисунке 24 изображен фрагмент кода этого представления.

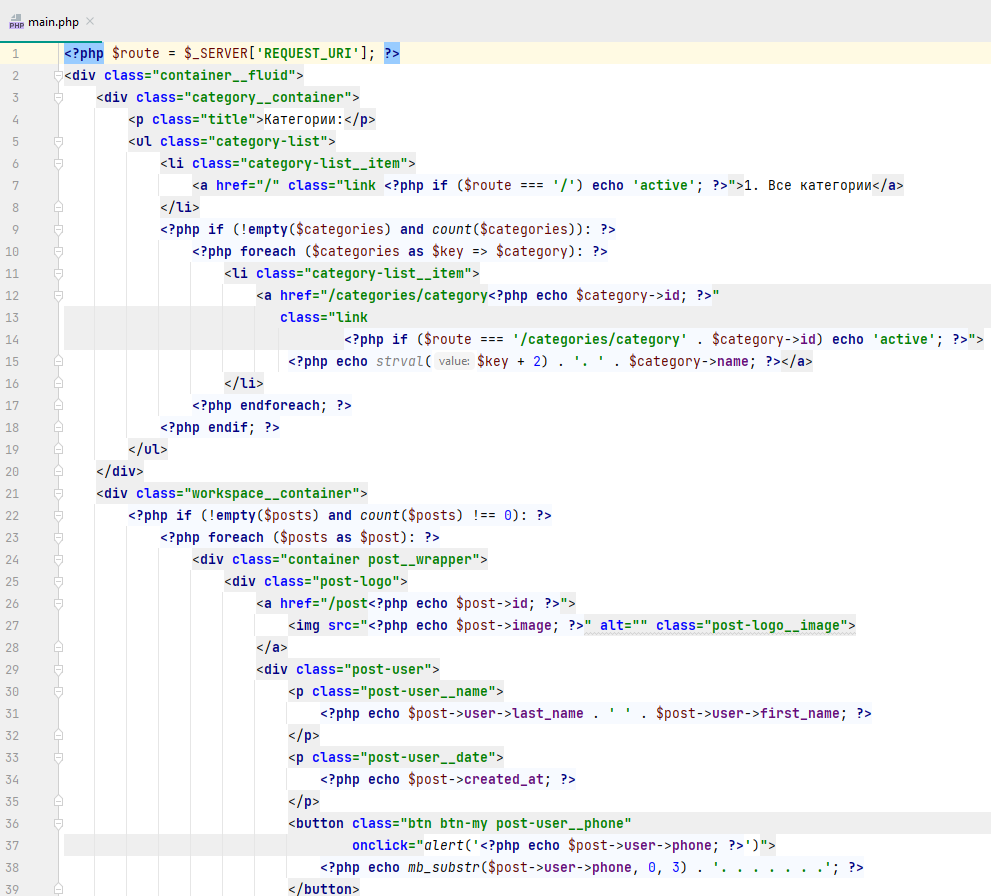


Рисунок 24 – Фрагмент представления главной страницы

В работе приложения не малую роль играет система маршрутов, которые связывают файл представления с нужным ему контроллером. Все маршруты ИС представлены в виде массива routes в папке app/config. Часть маршрутов проекта можно просмотреть в приложении А Листинг маршрутов.

1. Документирование программного обеспечения
2. Руководство пользователя программного обеспечения
   * 1. Описание системы

Информационная система «Сервис объявлений» позволяет не авторизированным пользователям увидеть объявления, но ничего большего. Связаться с пользователем, который выложил это объявление не получится, так как номер скрыт для всех и открывается только авторизированным пользователям.

Авторизированный пользователь может выставлять свои объявления. Редактировать их и удалять. Так же позволит видеть номера в объявлениях, чтобы связаться с пользователем.

Если у авторизированного пользователя роль администратора, то у него открывается возможность к выдачи этой роли другим пользователям, манипулировать данными объявлений, пользователей и другими.

* + 1. Страницы входа и учетные записи

Страница входа позволяет войти в одну из учетных записей информационной системы. Страница представляет собой форму, которая демонстрируется на рисунке 25.

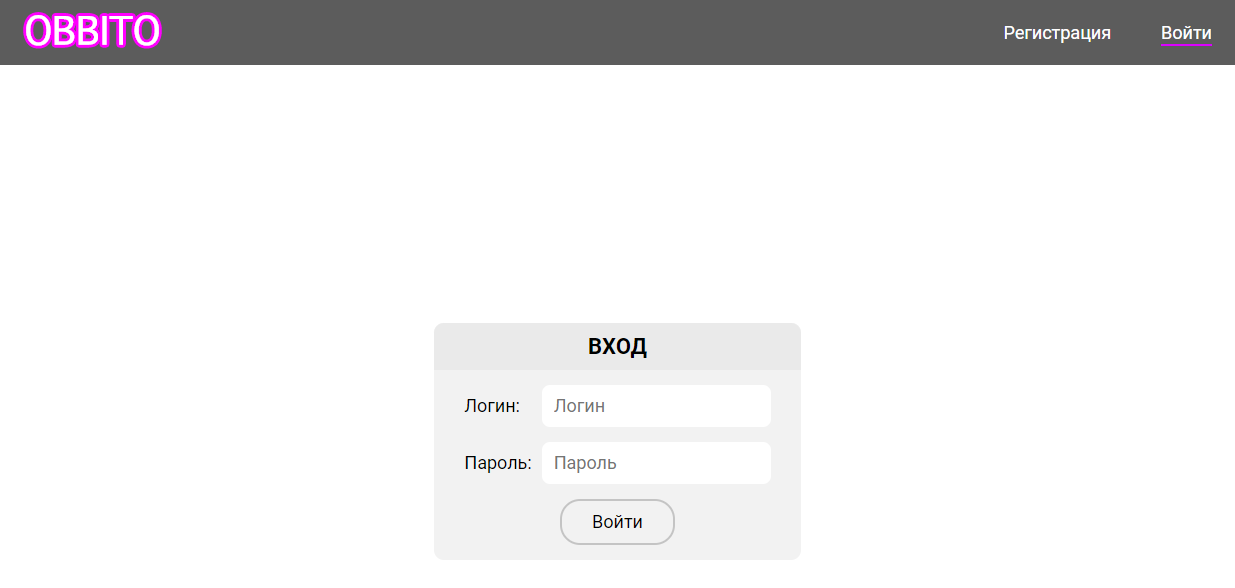


Рисунок 25 – Страница входа

В информационной системе существует 3 вида пользователей: неавторизированный, авторизированный пользователи и администратор.

При первой распаковке информационной системы существует только одна учетная запись со всеми правами доступа (логин: root, пароль: qweqwe123). После входа рекомендуется создать новую учетную запись с таким уровнем доступа, войти в нее и удалить стандартную учетную запись, так как её данные доступны любому прочитавшему это руководство пользователя.

* + 1. Главная страница

Главная страница позволяет увидеть последние добавленные объявления и выбрать объявления определенной категории. Внешний вид главной страницы неавторизированного и авторизированного пользователей на рисунке 26.

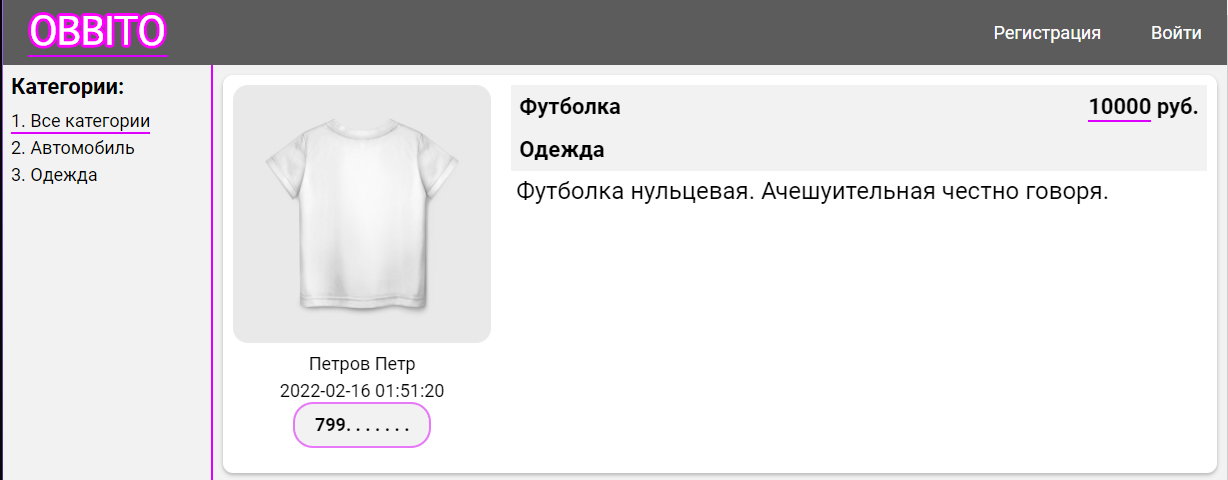


Рисунок 26 – Главная страница

На главной странице администратора имеется только навигация по панели администратора. Администратор также имеет доступ к главной странице обычных пользователей. Главная страница представлена на рисунке 27.

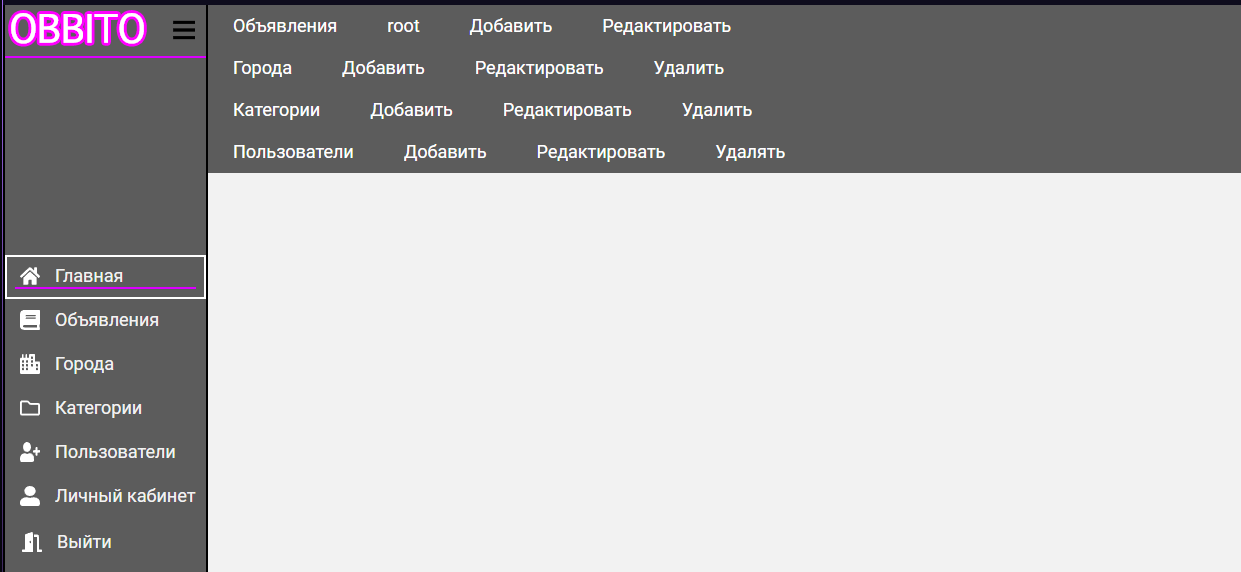


Рисунок 27 – Главная страница администратора

* + 1. Манипулирование объявлениями

Раздел «Объявления» в панели администратора позволяет добавлять, изменять и удалять объявления. В форме добавления объявлений нет ничего не обычно. Всё интуитивно понятно. Заполняется форма: указать заголовок объявления, стоимость товара или услуги, категорию объявления, описание и выбрать главное изображение. Загрузка дополнительных изображений необязательно, но рекомендуется, что пользователь лучше понял объявление. Внешний вид раздела представлен на рисунке 28.

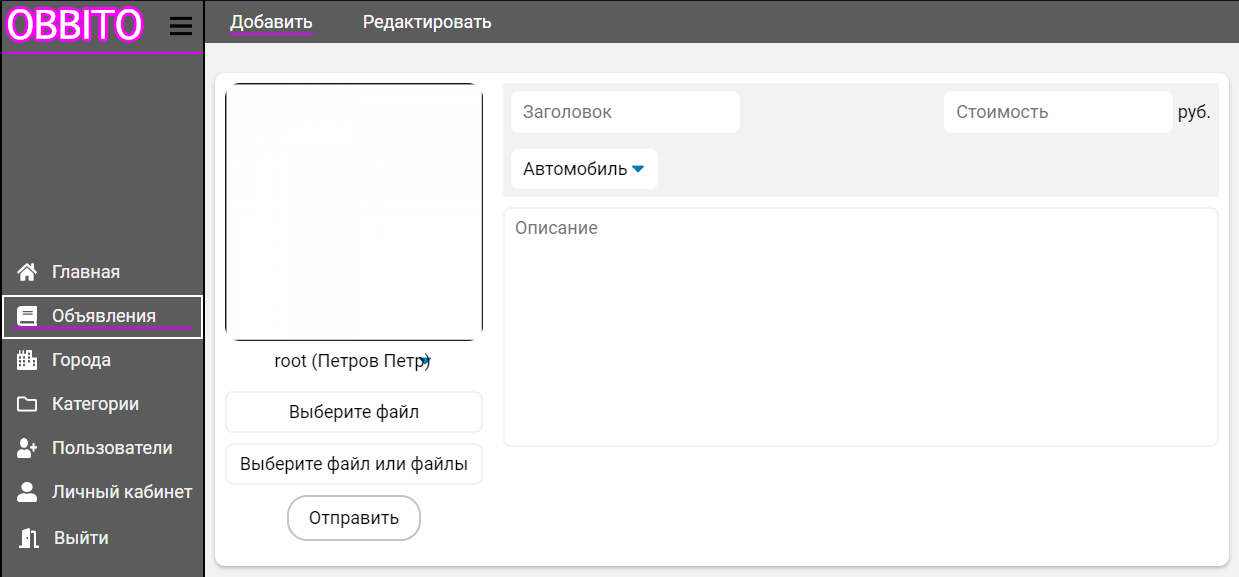


Рисунок 28 – Добавление объявления администратором

Отличием от версии для просто авторизированных пользователей в том, что администратор может зарегистрировать объявление за любым пользователем.

Все выбранные дополнительный изображения появляются ниже формы. Также все изображения можно просмотреть в крупном плане, в модальном окне (рисунок 29).

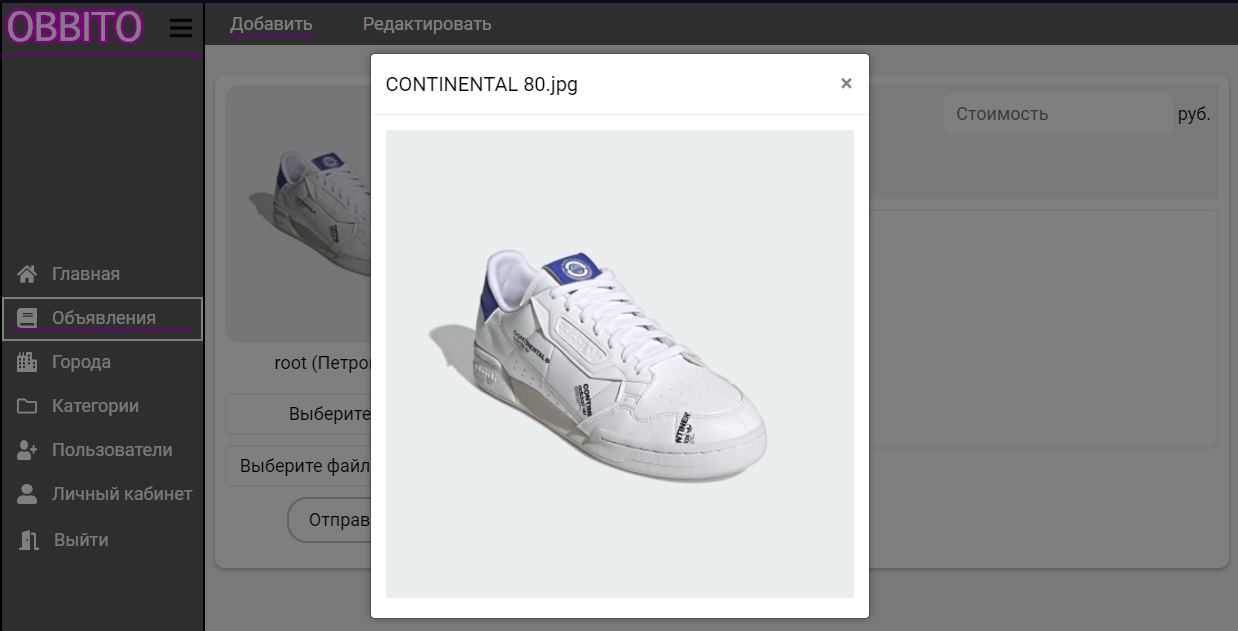


Рисунок 29 – Крупный план изображений

Отличие редактирования от добавления лишь в том, что форма по умолчанию заполнена и необходимо лишь изменить информацию и сохранить (Рисунок 30).

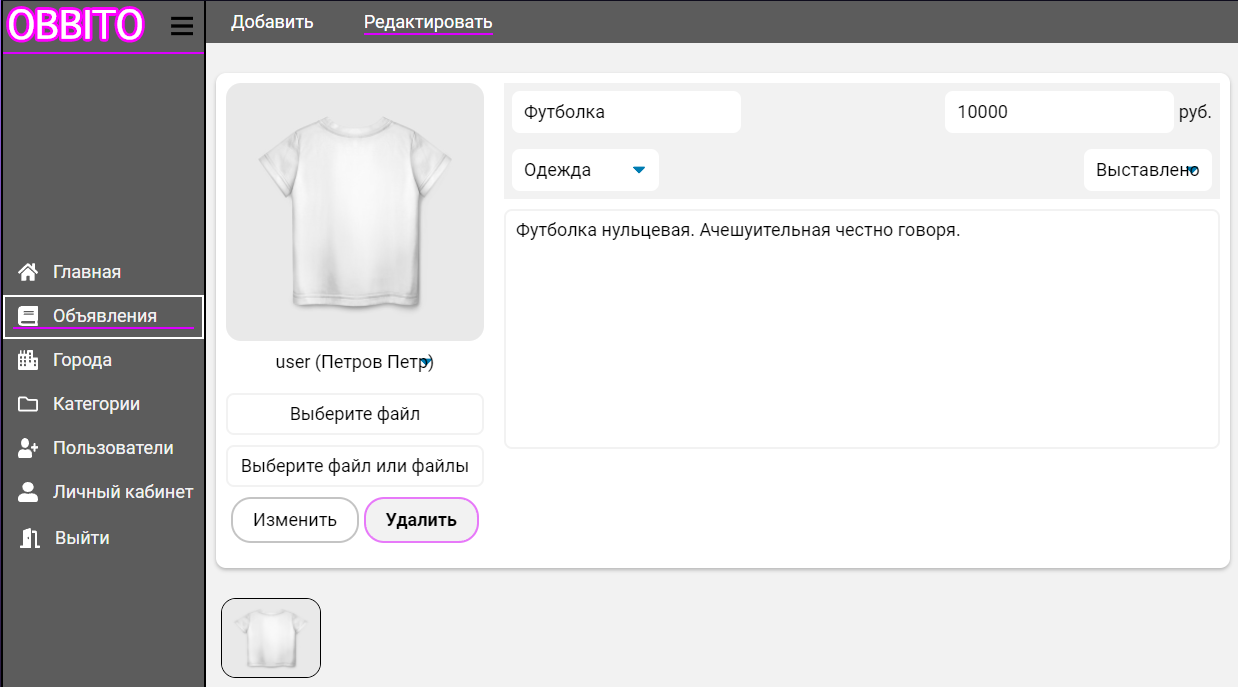


Рисунок 30 – Страница редактирования и удаления объявлений

В окно редактирования добавляется возможность изменить статус объявления и удалить его. При нажатии на кнопку «Удалить», информационная система потребует подтвердить удаление и если удалить не получается сообщит по какой причине.

* + 1. Манипулирования статическими данными

К статическим данным относятся названия городов и категорий, которые используются в информационной системе. Раздел «Города» и «Категории» в панели администратора позволяют добавлять, изменять и удалять города и категории соответственно.

Так как интерфейсы этих разделов идентичны, рассмотрим только один из них. В страницы добавления входит форма, в которой можно указать название нового города или категории. Внешний вид страницы добавления продемонстрирован на рисунке 31.

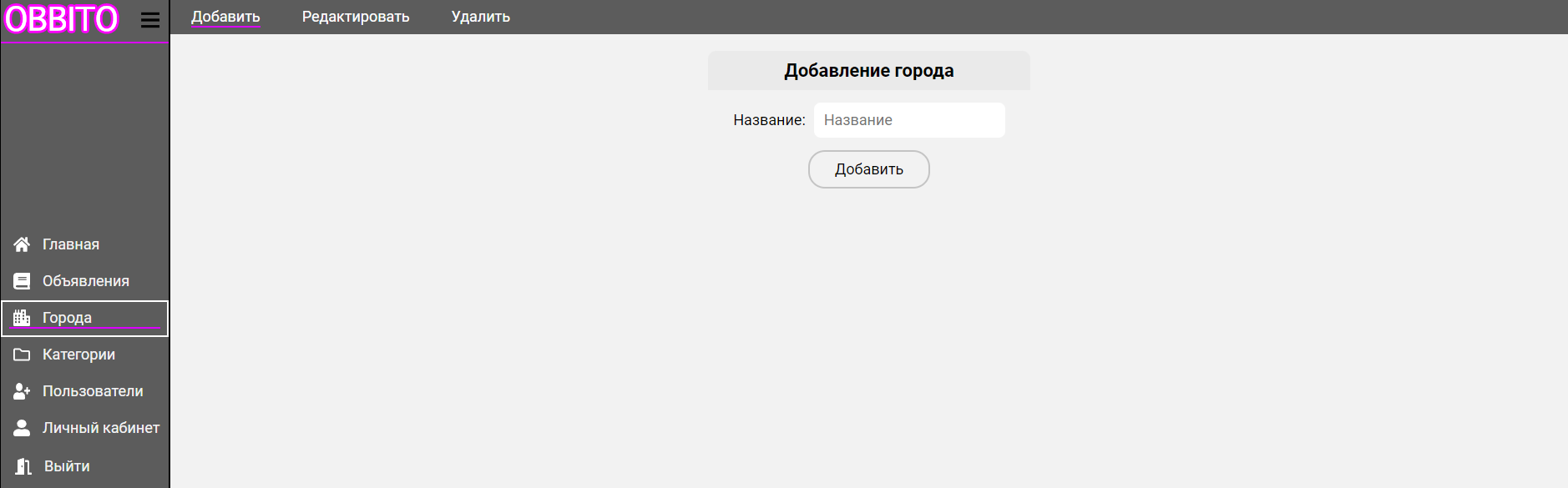


Рисунок 31 –Добавление города

На страницах редактирования выводится список имеющихся городов или категорий в виде заполненных форм. Сохранение происходит при нажатии на кнопку «Изменить». Внешний вид страниц редактирования представлен на рисунке 32.

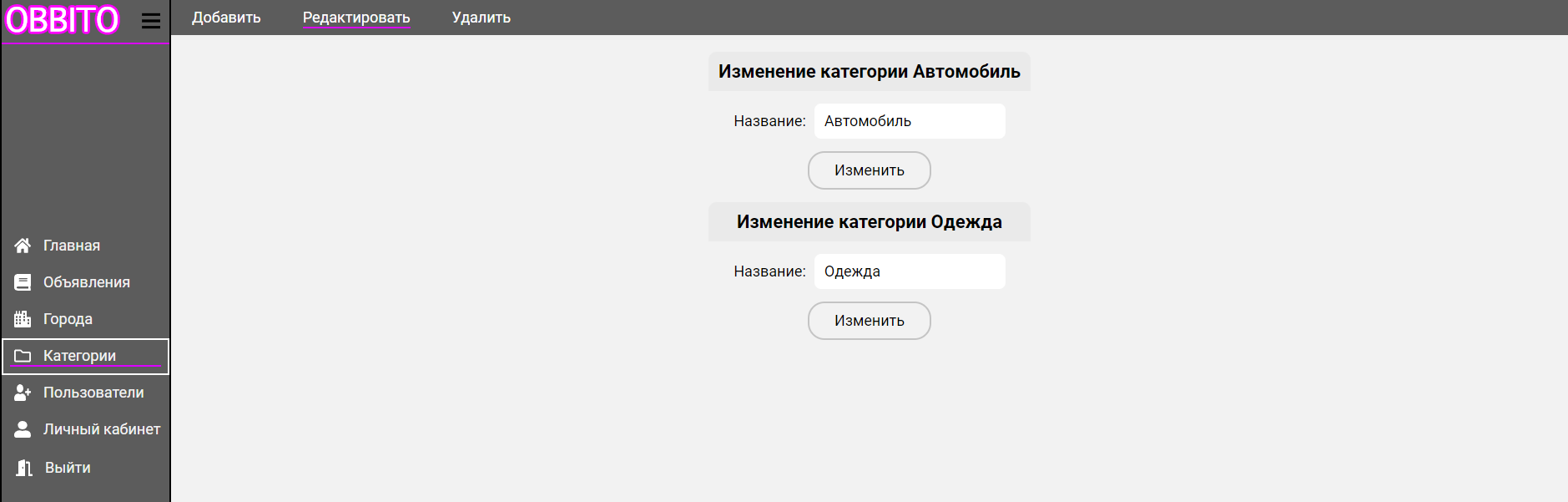


Рисунок 32 –Редактирование категорий

Страницы удаления аналогичны редактированию. Выводится список городов или категорий, и кнопка «Удалить». Внешний вид на рисунке 33.

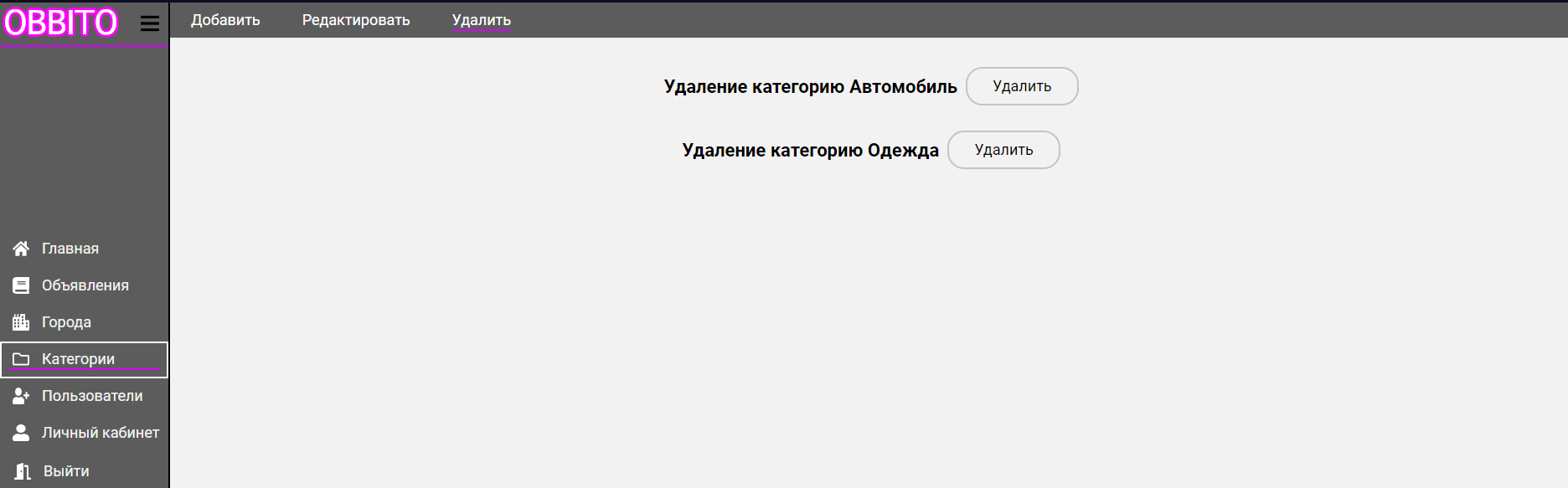


Рисунок 33 –Удаление категорий

* + 1. Манипулирование пользователями

Раздел «Пользователи» в панели администратора позволяет регистрировать новых пользователей с различными ролями, редактировать личные данные и пароли пользователям, а также удалять учетные записи. Внешний вид страницы добавления на рисунке 34.

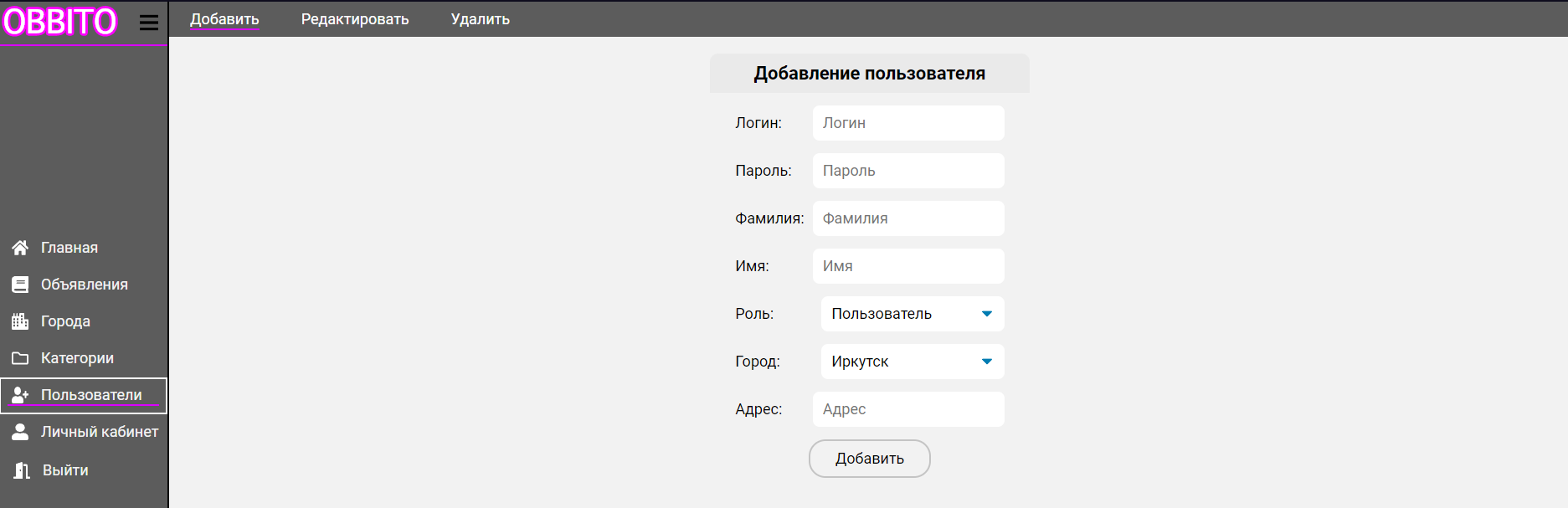


Рисунок 34 –Добавление пользователя

Страница редактирования позволяет изменять данные пользователей. Выводятся данные в виде 2 форма: одна для редактирования личных данных, другая для изменения пароля. Внешний вид на рисунке 35.

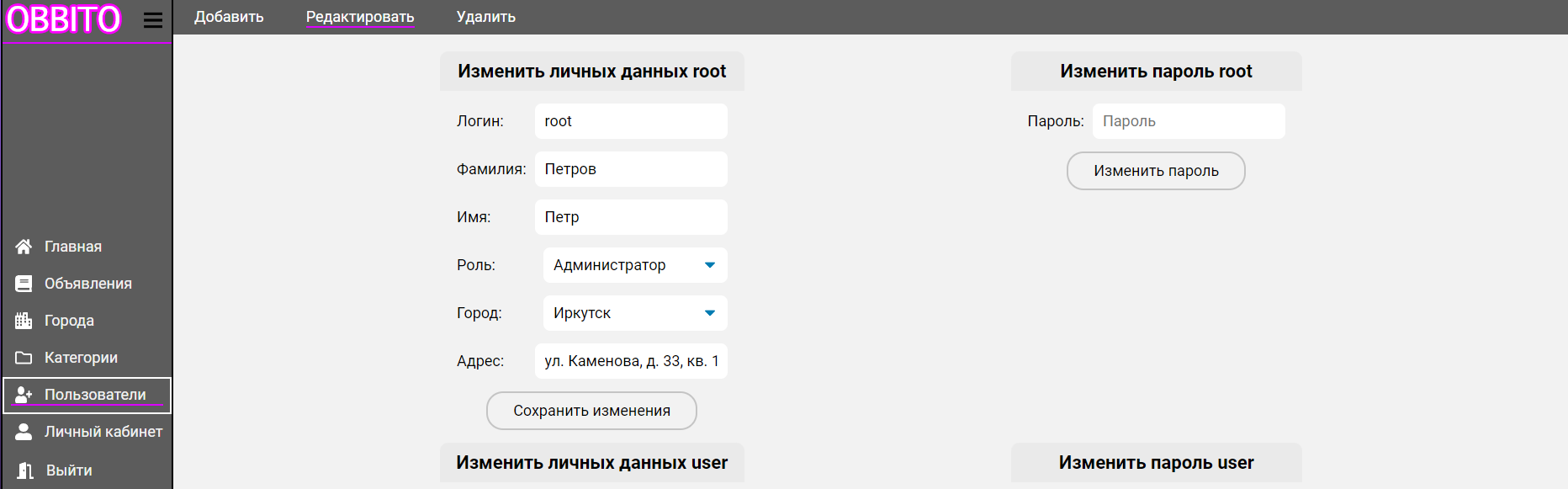


Рисунок 35 –Редактирование пользователей

Удаление пользователей аналогично разделу «Категории». Выводится список пользователей, и кнопка удалить (рисунок 36.)

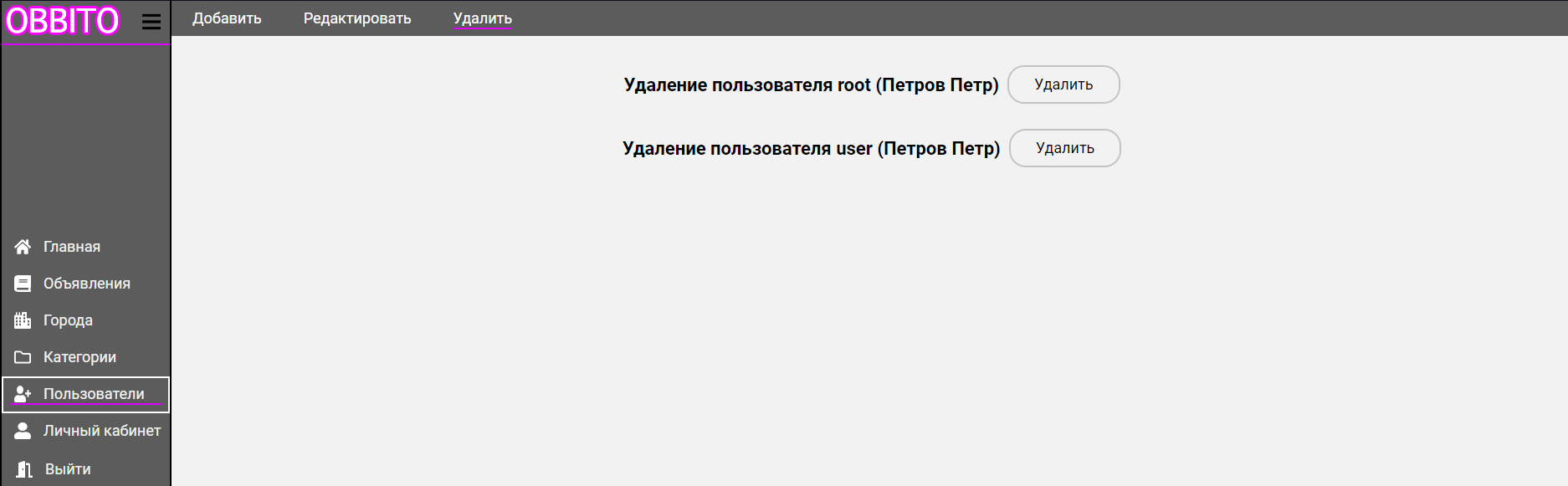


Рисунок 36 –Удаление пользователей

1. Стоимость разработки и внедрения программного продукта
   1. Организационно-экономическое обоснование проекта

Разрабатываемая информационная система будет использоваться для упрощения процесса оказания услуг, накопления клиентской базы, связанного c выставлением объявлений.

Разработанный программный продукт распространен в крупных сервисах. Стоимость разработки сервиса объявлений составляет от 80 000 рублей, в зависимости от функционала.

* 1. Расчет затрат на разработку программного продукта

Расчет полных затрат на разработку проектного решения (КРПР) осуществляется по формуле 6.1:

(6.1)

КРПР=ЗОТР+ЗЭВМ+ЗСПП+ЗХОН+Е+А,

гдеЗОТР – сумма оплаты труда разработчика/разработчиков ПП;

ЗЭВМ – затраты, связанные с эксплуатацией техники;

ЗСПП – затраты на специальные программные продукты, необходимые для разработки ПП;

ЗХОН – затраты на хозяйственно-операционные нужды (бумага, литература, носители информации и т.п.);

Е – затраты на электроэнергию, руб.;

А – амортизация ПК, руб.

Для подсчета фонда оплаты труда разработчика необходимо определить общее время разработки (таблица 10). Время, затрачиваемое на разработку проектного решения j-м разработчиком, определяется методом экспертных оценок или хронометража. Итоговое значение рассчитывается на основании приведенных исходных данных по формуле 6.2:

(6.2)

где tβ– время β-го этапа разработки проектного решения, дн.

Таблица 10 - Затраты времени на создание программного продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Этап создания | | Затраты времени  (в днях) | Затраты времени (в часах) | Машинное время работы над ПП  (в часах) |
| Разработка | Обследование объекта автоматизации | 3 | 24 | 16 |
| Анализ и уточнение требований | 3 | 24 | 8 |
| Разработка технического задания | 4 | 32 | 32 |
| Проектирование структуры | 5 | 40 | 40 |
|  | Программная реализация | 58 | 464 | 450 |
| Тестирование программного продукта | 5 | 40 | 40 |
| Отладка программного продукта | 5 | 40 | 40 |
| Разработка описания | 5 | 40 | 40 |
| ИТОГО | 88 | 704 | 666 |

Рабочий день принимается равным 8 часам.

Сумму оплаты труда разработчика за время работы над программным продуктом рассчитаем исходя из часовой тарифной ставки и фонда фактического времени, затраченного на разработку программного продукта (по формуле 6.3):

(6.3)

Зотр=Ст1\*Фвр,

где Ст1 - часовой тарифной ставки (принимаем равной 21 руб./час);

Фвр – фонд фактического времени, затраченного на разработку программного продукта, час.

Разработка программного продукта проводилась бесплатно, так как связывающий договор не требует оплаты труда.

Затраты, связанные с использованием вычислительной и оргтехники, рассчитываются по формуле 6.4:

,

(6.4)

где ТМРПР – машинное время работы над программным продуктом, час;

kГ– коэффициент готовности ЭВМ,  ;

n – количество единиц техники, равно 1;

СМ-Ч – себестоимость машино-часа, СМ-Ч=6руб.

Затраты на электроэнергию рассчитываются по следующей формуле 6.5:

Е=W×t×T,

(6.5)

где W – мощность, потребляемая ПК, кВт/час;

t – время работы ПК, час;

T – тариф электроэнергии, руб.

Тариф 1,23 рубля за киловатт.

Амортизация ПК рассчитывается по следующей формуле 6.6:

(6.6)

где – первоначальная стоимость ПК, руб;

– процент амортизации в год.

При разработке программного продукта использовались следующие свободно распространяемые специальные продукты PhpStorm, Draw.IO, Figma.

Затраты на хозяйственно-организационные нужды приводятся в таблице 11 и вычисляются по формуле 6.7:

,

(6.7)

где Цτ – цена τ-го товара, руб.;

Кτ – количество τ-го товара.

Таблица 11 - Затраты на хозяйственно-организационные нужды

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена за единицу (руб.) | Кол-во (шт.) | Всего (руб.) |
| Диск CD-RWDigitex | 35 | 1 | 35 |
| Бумага | 1 | 120 | 120 |
| Чернила для принтера | 700 | 2 | 1400 |
| Итого | | | 1555 |

Результаты выполненных расчетов сводятся в общей таблице (таблица 12).

Таблица 12 – Затраты на разработку

|  |  |  |
| --- | --- | --- |
| Наименование затрат | Условное обозначение | Значение |
| Оплата труда разработчика программного продукта |  | 14784 |
| Затраты, связанные с эксплуатацией техники |  | 3796,2 |
| Затраты на электроэнергию | Е | 655,35 |
| Затраты на амортизацию ПК | А | 1250 |
| Затраты на хозяйственно-операционные нужды |  | 1555 |
| Итого затрат на разработку |  | 22040,55 |

* 1. Расчет затрат на внедрение программного продукта

Информационная система не внедряется в предприятие, поэтому затрат на внедрение нет. Также все используемые программные средства бесплатны и свободно распространяются.

* 1. Основные выводы

Исходя из расчетов стоимости разработки, можно прийти к выводу, что информационная система обойдется предприятию (без учета оплаты труда) в размере 7256,55 рублей. Такие затраты значительно меньше стоимости рассматриваемых аналогов.

# Заключение

В ходе дипломного проекта было разработано информационная система «Сервис объявлений», в котором реализовано система регистрации и авторизации, хранение информации об объявлениях и пользователях, добавление, удаление и редактирование данных, распределение прав учётных записей и подсчет количества просмотра объявления, и изменение статуса объявления.

Информационная система «Сервис объявлений» упростить распространения объявлений пользователей и обновлять базу данных, накапливать клиентскую базу. Это ускорит процесс оказания услуг, что позволит оказывать большее количество услуг, что приведет к росту предприятия.

Все поставленные задачи, а именно: система регистрации и авторизации, хранение информации об объявлениях и пользователях, добавление, удаление и редактирование данных, распределение прав учётных записей и подсчет количества просмотра объявления, и изменение статуса объявления.

Все цели, поставленные для разработки, были достигнуты.

В информационной системе были реализованы следующие функции:

* регистрация и авторизация;
* хранение информации об объявлениях и пользователях;
* добавление, удаление и редактирование данных;
* распределение прав учётных записей;
* подсчет количества просмотра объявления, и изменение статуса объявления.

# Список использованных источников

1. BizzApps [Электронный ресурс] – Режим доступа: https://bizzapps.ru/p/draw-io/ – Документация (Дата обращения: 21.04.2022).
2. Calltouch [Электронный ресурс] – Режим доступа: https://www.calltouch.ru/glossary/reklamnoe-obyavlenie/ – Рекламное объявление. (Дата обращения: 16.02.22).
3. CyberForum [Электронный ресурс] – Режим доступа: https://www.cyberforum.ru/php/thread1397412.html – Удалить переменную из сессии после ее вывода (Дата обращения: 18.02.22).
4. Envatotuts+ [Электронный ресурс] – Режим доступа: https://code.tutsplus.com/ru/tutorials/using-illuminate-database-with-eloquent-in-your-php-app-without-laravel--cms-27247 – Используем Illuminate Database вместе с Eloquent в приложении (Дата обращения: 28.04.2022).
5. Habr [Электронный ресурс] – Режим доступа: https://habr.com/ru/post/150267/ – Реализация MVC паттерна на примере создания сайта-визитки на PHP (Дата обращения: 17.02.2022).
6. Hostinger Руководства [Электронный ресурс] – Режим доступа: https://www.hostinger.ru/rukovodstva/web-server-apache/ – Что такое Apache? (Дата обращения: 16.03.22).
7. Life-prog [Электронный ресурс] – Режим доступа: https://life-prog.ru – Документация (Дата обращения: 21.04.2022).
8. Lucidchart [Электронный ресурс] – Режим доступа: https://www.lucidchart.com/pages/ru/visio-что-это – Что такое Visio? (Дата обращения: 18.04.2022).
9. Mr. Кодер работает [Электронный ресурс] – Режим доступа: http://code.mu/ru/php/book/oop/ – Учебник по ООП и MVC на PHP (Дата обращения: 07.02.22).
10. PHP.NET [Электронный ресурс] – Режим доступа: https://www.php.net/ – Документация по PHP (Дата обращения: 02.02.2022).
11. PHPMyAdmin [Электронный ресурс] – Режим доступа: https://php-myadmin.ru/doc/ability.html – Руководство. Возможности PhpMyAdmin (Дата обращения: 08.03.22).
12. SQL. Полное руководство. Грофф Дж. Р., Вайнберг П.Н., Оппелъ Э. Дж. 2015 – 959 с.
13. WebForMyself [Электронный ресурс] – Режим доступа: https://webformyself.com/peredacha-znachenij-peremennyx-iz-javascript-v-php-i-naoborot/ – Документация (Дата обращения: 27.03.2022).
14. Web-технологии [Электронный ресурс] – Режим доступа: https://htmlweb.ru/java/example/image\_preview3.php – Предпросмотр картинки перед загрузкой на сервер на javascript (Дата обращения: 18.02.2022).
15. Wikipedia [Электронный ресурс] – Режим доступа: https://ru.wikipedia.org/wiki/Информационная\_система – Информационная система (Дата обращения: 16.02.22).

# Приложение А - Листинг маршрутов

<?php

return [

'' => [

'controller' => 'MainController',

'action' => 'index'

],

'lk'=>[

'controller' => 'MainController',

'action' => 'lk'

],

'admin'=>[

'controller' => 'AdminController',

'action' => 'index'

],

'categories/category\d{1,}' => [

'controller' => 'MainController',

'action' => 'view'

],

'login' => [

'controller' => 'AuthController',

'action' => 'login'

],

'registration' => [

'controller' => 'AuthController',

'action' => 'registration'

],

'post\d{1,}' => [

'controller' => 'PostController',

'action' => 'view'

],

'posts'=>[

'controller'=>'PostController',

'action'=>'index'

],

'posts/create'=>[

'controller'=>'PostController',

'action'=>'create'

],

'posts/edit'=>[

'controller'=>'PostController',

'action'=>'edit'

],

'admin/posts/create'=>[

'controller'=>'AdminController',

'action'=>'createPost'

],

'admin/posts/edit'=>[

'controller'=>'AdminController',

'action'=>'editPost'

],

'admin/cities/create'=>[

'controller'=>'AdminController',

'action'=>'createCity'

],

'admin/cities/edit'=>[

'controller'=>'AdminController',

'action'=>'editCity'

],

'admin/cities/delete'=>[

'controller'=>'AdminController',

'action'=>'deleteCity'

],

'admin/categories/create'=>[

'controller'=>'AdminController',

'action'=>'createCategory'

],

'admin/categories/edit'=>[

'controller'=>'AdminController',

'action'=>'editCategory'

],

'admin/categories/delete'=>[

'controller'=>'AdminController',

'action'=>'deleteCategory'

],

'admin/users/create'=>[

'controller'=>'AdminController',

'action'=>'createUser'

],

'admin/users/edit'=>[

'controller'=>'AdminController',

'action'=>'editUser'

],

'admin/users/delete'=>[

'controller'=>'AdminController',

'action'=>'deleteUser'

],

'api/login' => [

'controller' => 'AuthController',

'action' => 'auth'

],

'api/registration' => [

'controller' => 'AuthController',

'action' => 'reg'

],

'api/logout' => [

'controller' => 'AuthController',

'action' => 'logout'

],

'api/user/create' => [

'controller' => 'UserController',

'action' => 'create'

],

'api/user/update' => [

'controller' => 'UserController',

'action' => 'update'

],

'api/user/delete' => [

'controller' => 'UserController',

'action' => 'delete'

],

'api/user/changePassword' => [

'controller' => 'UserController',

'action' => 'changePassword'

],

'api/posts/create'=>[

'controller'=>'PostController',

'action'=>'insert'

],

'api/posts/update'=>[

'controller'=>'PostController',

'action'=>'update'

],

'api/posts/delete'=>[

'controller'=>'PostController',

'action'=>'delete'

],

'api/cities/create'=>[

'controller'=>'CityController',

'action'=>'insert'

],

'api/cities/update'=>[

'controller'=>'CityController',

'action'=>'update'

],

'api/cities/delete'=>[

'controller'=>'CityController',

'action'=>'delete'

],

'api/categories/create'=>[

'controller'=>'CategoryController',

'action'=>'insert'

],

'api/categories/update'=>[

'controller'=>'CategoryController',

'action'=>'update'

],

'api/categories/delete'=>[

'controller'=>'CategoryController',

'action'=>'delete'

],

'admin/migrate/fresh' => [

'controller' => 'MigrateController',

'action' => 'fresh'

],

'admin/migrate/freshAndSeed' => [

'controller' => 'MigrateController',

'action' => 'freshAndSeed'

],

'admin/migrate' => [

'controller' => 'MigrateController',

'action' => 'migrate'

],

'admin/migrateAndSeed' => [

'controller' => 'MigrateController',

'action' => 'migrate'

],

];

# Приложение Б - Листинг javascript

const files = document.querySelectorAll('.form-item\_\_file')

files.forEach(function (el) {

el.addEventListener('change', function (ev) {

const input = ev.target

const main = document.getElementById("mainImage")

const images = document.querySelector('.post-images')

const label = input.parentElement

const text = label.querySelector('div')

// text.innerHTML = ''

if (input.files.length === 0) {

text.innerHTML = 'Выберите файл'

if(input.id ==='image'){

main.setAttribute('src','/public/storage/images/template.jpg')

main.setAttribute('title','Template')

main.dataset.toggle=''

}

if(input.id ==='images'){

images.innerHTML=''

}

return false;

}

for (let i =0,f; f=input.files[i];i++){

if (!f.type.match('image.\*')) {

continue;

}

let reader = new FileReader();

reader.onload=(function (file){

return function (e){

if(input.id ==='image'){

main.setAttribute('src',e.target.result)

main.dataset.toggle='modal'

main.dataset.title=file.name

main.setAttribute('title',file.name)

// text.innerText +=file.name

} else{

let span = document.createElement('span');

span.innerHTML = ['<img class="thumb" src="', e.target.result,

'" title="', file.name, '" data-toggle="modal" ',

'data-title="',file.name,'"/>'].join('');

images.insertBefore(span, null);

// text.innerText +=file.name+'\n'

}

}

})(f)

reader.readAsDataURL(f)

}

})

})

const modal = $modal()

document.addEventListener('click', function (e) {

if (e.target.dataset.toggle === 'modal') {

const img = e.target

modal.setTitle(img.dataset.title)

modal.setContent(`<img style="width: 100%;" src="${img.src}" title="${img.title}">`)

modal.show();

}

if(e.target.id==='formDelete'){

e.preventDefault()

if(confirm('Вы действительно хотите удалить объявление?')){

const formData = new FormData()

formData.append('id',e.target.dataset.id)

fetch('/api/posts/delete',{

method: "POST",

credentials: "same-origin",

body: formData

})

.then(()=>location.reload())

}

}

})