

**МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
“ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ”**

Факультет компьютерных наук
Кафедра программирования и информационных технологий

Сервис для обмена коллекционными карточками и наборами
Курсовой проект

09.03.04 Программная инженерия
Информационные системы и сетевые технологии

Зав. Кафедрой _____ С.Д. Махортов, д-р физ.-мат. наук, профессор
Обучающийся _____ В.А. Винокуров, 3 курс, очное
Обучающийся _____ Д.Ю. Шевченко, 3 курс, очное
Руководитель _____ В.С. Тарасов, ст. преподаватель __. __.20__

Воронеж 2024

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ	4
ВВЕДЕНИЕ	7
1 Постановка задачи	8
1.1 Требования к функциональной части	8
1.2 Технические требования	8
1.2.1 Платформа и ОС	9
1.3 Требования к интерфейсу	9
2 Анализ предметной области	10
2.1 Анализ существующих решений	10
2.1.1 LastSticker	10
2.1.2 TOPDeck	12
2.1.3 Avito	13
2.1.4 Mymagic	14
2.1.5 Mtgtrade	16
2.2 Итоги анализа	17
3 Графическое описание работы системы	19
3.1 Диаграммы прецедентов	19
3.1.1 Диаграмма прецедентов неавторизованного пользователя	19
3.1.2 Диаграмма прецедентов авторизованного пользователя	19
3.2 Диаграмма разворачивания	20
3.3 Диаграммы сотрудничества	21
3.4 Диаграммы последовательности	23
3.4.1 Диаграмма последовательности для неавторизованного пользователя	23
3.4.2 Диаграмма последовательности для авторизованного пользователя	25
3.4.3 Диаграмма последовательности для администратора	27
4 Реализация	29

4.1 Средства реализации.....	29
4.2 Реализация базы данных	30
4.3 Реализация серверной части приложения	32
4.4 Реализация клиентской части приложения	33
4.4.1 Общая информация.....	33
4.4.2 Графический интерфейс	34
4.5 Методология разработки	41
5 Тестирование	43
5.1 Дымовое тестирование	43
5.2 UI-тестирование	44
ЗАКЛЮЧЕНИЕ	46
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	47

ОПРЕДЕЛЕНИЯ, СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

Таблица 1 - Определения, сокращения, обозначения

Термин	Определение термина
API	Программный интерфейс приложения. Описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой
PostgreSQL	Свободно распространяемая объектно-реляционная система управления базами данных (СУБД) с открытым исходным кодом
Rest	Архитектурный стиль взаимодействия компонентов распределенного приложения в сети
Аватарка	Изображение, которое идентифицирует пользователя
Авторизация	Предоставление определенному лицу прав на выполнение определенных действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий
Автоцены	Функция, позволяющая продавать сразу много товаров по установленной цене
Авторизованный пользователь	Пользователь, прошедший процесс авторизации
Архитектура приложения	Способ организации и структурирования программного кода, который обеспечивает работу приложения

Термин	Определение термина
База Данных (БД)	Упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе
Боковое меню	Меню, которое представляет собой панель, которая находится (или открывается, при помощи, каких-либо кнопок или жестов) снизу, слева или справа от области основного контента приложения, содержащая вертикальную, независимую от основного контента приложения прокрутку, и служит основным инструментом навигации в приложении
Интерфейс	Среда, которую пользователь видит и с которой взаимодействует при использовании приложения. Он включает в себя все элементы управления, кнопки, меню, диалоговые окна и другие элементы, которые помогают пользователю управлять приложением и выполнять с его помощью различные задачи
Клиент-серверное взаимодействие	Способ обмена информацией между двумя устройствами, где одна сторона (клиент) запрашивает данные у серверной части. Последняя формирует ответ, направляя его в обмен
Клиентская часть	Часть программного обеспечения, которая взаимодействует непосредственно с пользователем через интерфейс на стороне пользователя
Неавторизованный пользователь	Пользователь, не прошедший процесс авторизации

Термин	Определение термина
Реляционная база данных	Набор данных с predetermined связями между ними. Эти данные организованы в виде набора таблиц, состоящих из столбцов и строк
Реляционная СУБД	Система управления базами данных. Комплекс программ, позволяющих создать реляционную базу данных (БД) и управлять данными
Серверная часть	Программно-аппаратная часть сервиса, которая хранится на сервере, обрабатывает полученные данные и отправляет ответ обратно
Фреймворк	Заготовка, готовая модель в программировании для быстрой разработки, на основе которой можно дописать собственный код

ВВЕДЕНИЕ

В условиях роста популярности коллекционирования и обмена картами важность создания удобных платформ для поиска, обмена и покупки карт становится всё более актуальной для людей, которые интересуются картами. Это решение особенно важно для тех, кто ищет эффективные и безопасные способы взаимодействия с другими участниками сообщества коллекционеров.

Данный курсовой проект посвящен разработке веб-приложения для обмена картами, где владельцы карт могут создавать объявления, оставлять свои контактные данные, а другие пользователи — искать и просматривать предложения с использованием удобного поиска или фильтров по различным критериям. Целью данного проекта является создание платформы, которая упростит взаимодействие коллекционеров карт и обеспечит удобные механизмы поиска и обмена.

Приложение предназначено для создания объявлений о продаже или обмене карт с указанием контактов владельца и возможностью оценки от других пользователей, добавления описаний, изображений и характеристик карт.

В ходе работы проведена разработка и тестирование предложенного веб-приложения.

Данное приложение может быть использовано коллекционерами карт для пополнения коллекции или получения прибыли от редких карт.

1 Постановка задачи

Целью данного курсового проекта является разработка веб-приложения для упрощения процесса обмена и покупки коллекционных карт. Приложение должно предоставлять пользователям возможность создавать объявления, добавлять описания и изображения карт, а также использовать фильтры для поиска предложений по заданным критериям.

1.1 Требования к функциональной части

Неавторизованным пользователям должна быть предоставлена возможность:

- регистрации;
- поиска объявлений и использования фильтров по категориям и характеристикам;
- просмотра объявлений.

Авторизованным пользователям должна быть предоставлена возможность:

- создания объявлений с описанием и изображением коллекционных карт;
- редактирования и удаления собственных объявлений;
- просмотра контактной информации владельцев карт;
- просмотра профилей других пользователей;
- оставления отзывов о сделках с другими пользователями.

1.2 Технические требования

Данные требования необходимы для обеспечения стабильной работы веб-приложения на целевых браузерах и устройствах, а также для

удовлетворения потребностей конечных пользователей в удобстве и эффективности использования приложения.

1.2.1 Платформа и ОС

Веб-приложение должно корректно отображаться в браузерах Google Chrome версии Google Chrome 129.0.6668.89 и выше, Mozilla Firefox 130.0.1 и выше, Safari 5.1.7.

1.3 Требования к интерфейсу

Приложение должно быть выполнено в одной цветовой палитре с использованием ограниченного набора шрифтов. Экраны приложения должны быть оформлены в едином стиле.

2 Анализ предметной области

Данный раздел содержит анализ предметной области.

2.1 Анализ существующих решений

Анализ существующих решений приведен на основе сервисов, указанных в таблице 2.

Таблица 2 - Примеры существующих решений

Название	Ссылка
LastSticker	https://naira-arina.ru/calculate
TOPDeck	https://topdeck.ru/
Avito	https://avito.ru/
Mymagic	https://mymagic.ru/
Mtgtrade	https://mtgtrade.net/

2.1.1 LastSticker

Данный сервис представляет из себя форум для коллекционеров. На сайте можно посмотреть различные коллекции и производителей карт, найти объявления по фильтрам, поучаствовать в аукционах. Функционал сервиса представлен на рисунке 1.

Преимущества:

- большая активная аудитория;
- множество коллекций в фильтрах;
- наглядность статистики;
- другие товары, интересующие коллекционеров.

Недостатки:

- малый функционал для незарегистрированных пользователей;

- перенасыщение страниц информацией;
- неудобная навигация;
- устаревший дизайн.

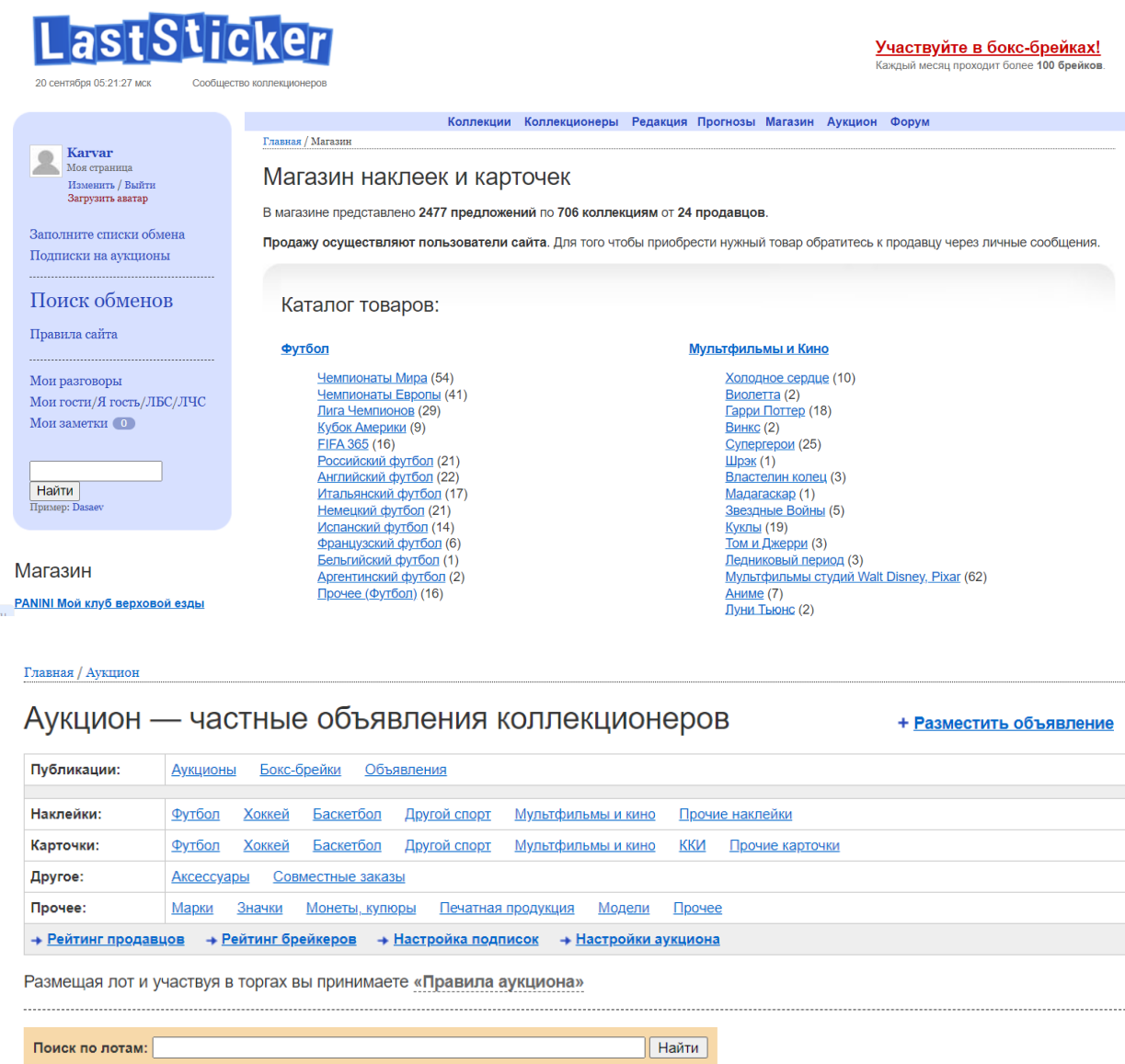


Рисунок 1 - Функционал сайта LastSticker

2.1.2 TOPDeck

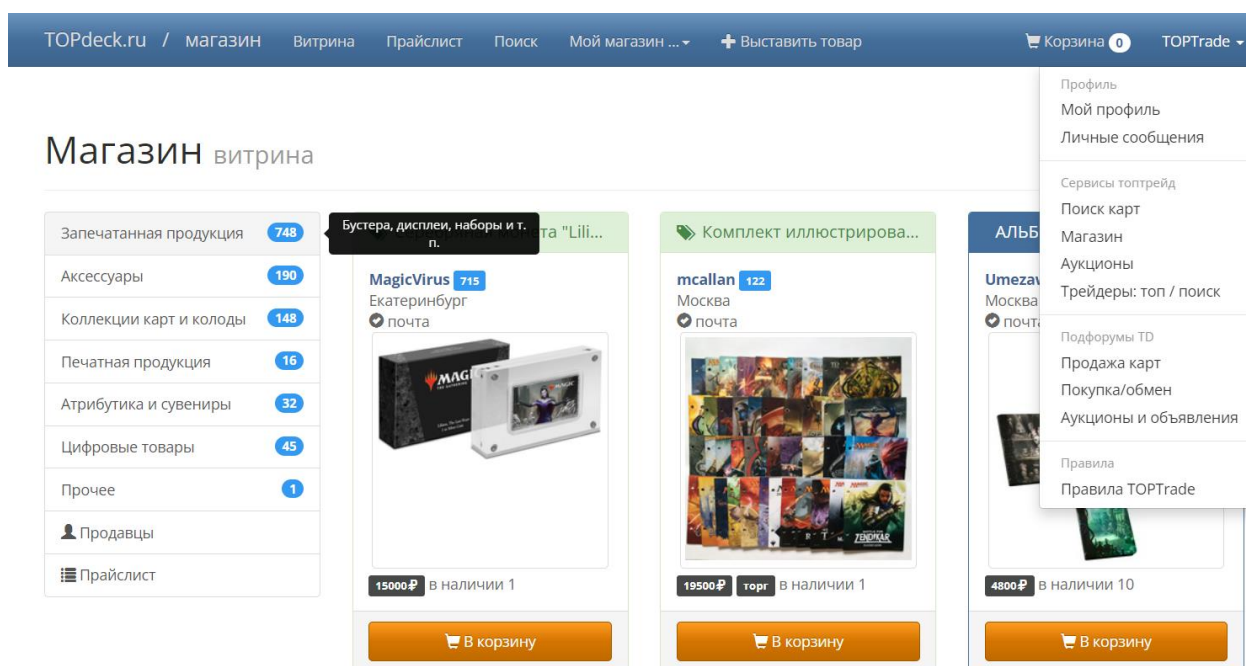
Данный сервис является сайтом для общения людей, которые интересуются картами. Для покупки или обмена картами нужно перейти во вкладку «Магазин». Продукцию можно отсортировать по фиксированным тегам или по поиску. Также присутствует раздел «Аукционы», где можно разместить товары на торги. Функционал сервиса представлен на рисунке 2.

Преимущества:

- узкая тематическая направленность;
- простота навигации по верхнему меню;
- присутствуют аксессуары для карт;
- наглядность статистики.

Недостатки:

- перегруженность функционала;
- мало действующих пользовательских объявлений;
- недостаточное количество фильтров;
- устаревший дизайн.



Поиск карт

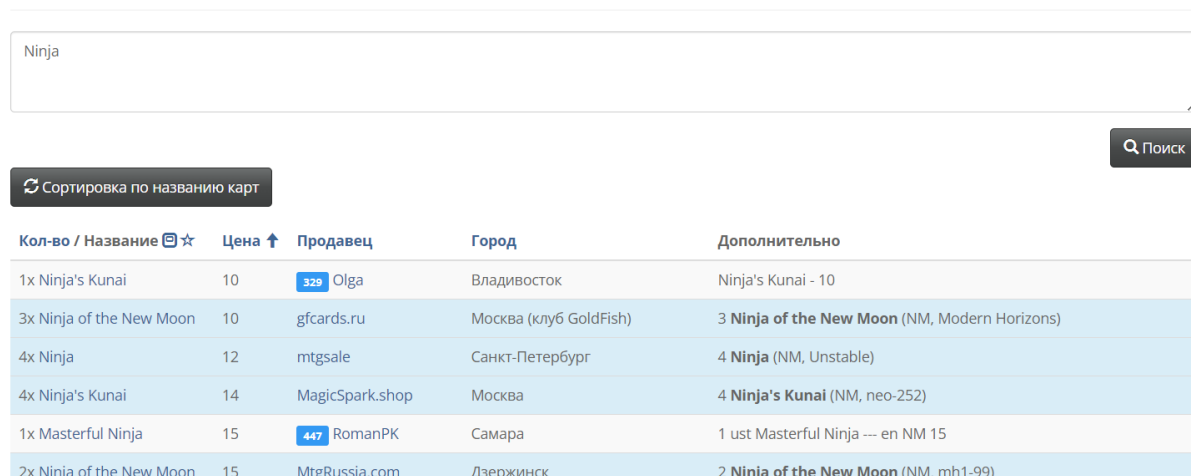


Рисунок 2 - Функционал сайта TOPDeck

2.1.3 Avito

Данный сервис предназначен для продажи разных товаров, в число которых входят и карты. Благодаря большому количеству пользователей, сайт имеет множество предложений для коллекционеров. Функционал представлен на рисунке 3.

Преимущества:

- много предложений;
- есть оценка продавца;

- интегрирована доставка;
- современный интерфейс.

Недостатки:

- не ориентирован на коллекционеров;
- мало фильтров.

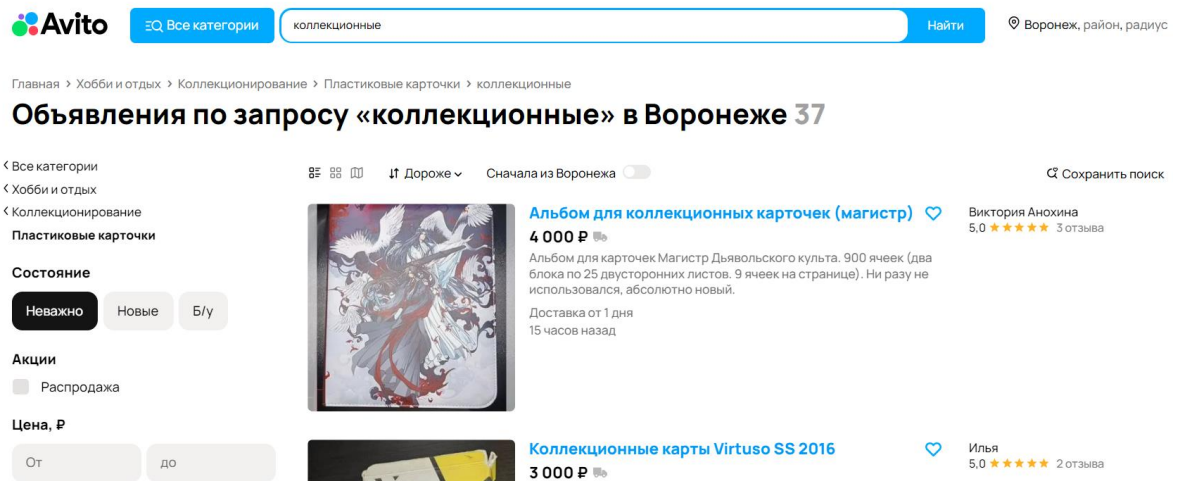


Рисунок 3 - Функционал Avito

2.1.4 Mymagic

Данный сайт является платформой по перепродаже карт. Сервис выкупает карты у одних пользователей и предлагает их к покупке другим. Функционал приложения представлен на рисунке 4.

Преимущества:



- большая база объявлений;
- отсутствие рисков;
- много разнообразных фильтров, делающих поиск легче;
- внятные названия товаров;
- легкая продажа карт.

Недостатки:

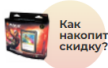
- включение процента посредника в цену;
- отсутствие взаимодействия с сообществом;

— отсутствие обратной связи по товарам и оценки продавцов.

[О магазине](#) [Оплата и доставка](#) [Частые вопросы](#)

МУМАГИК
МАГАЗИН КАНАЛ МАГИЯ

 Как
накопить
скидку?














Каталог

Поиск синглов

Профиль

Избранное

Корзина

-  The Brothers' War Promos PBRO
-  The Brothers' War Retro Artifacts BRR
-  The Brothers' War Tokens TBRO
-  The Dark DRK
-  The List PLIST
-  The Lord of the Rings: Tales of Middle-earth LTR
-  The Lost Caverns of Ixalan LCI
-  The Lost Caverns of Ixalan Commander LCC
-  The Lost Caverns of Ixalan Promos PLCI
-  Theros THS
-  Theros Beyond Death THB
-  Theros Beyond Death Promos PTNB
-  Theros Beyond Death Tokens TTNB

Kamahl, Pit Fighter

15th Anniversary Cards



English - Near Mint - Foil

199 руб.

Кол-во: 0

Подписаться

#2 - Rare - R
Kev Walker

Char

15th Anniversary Cards



English - Near Mint - Foil

252 руб.

Кол-во: 0

Подписаться

#1 - Rare - R
Dave Dorman

Покупка Синглов

Состояние ▾

Фойл ▾

Язык ▾

Редкость ▾

Цвет ▾

Фильтр

Здесь вы сможете сами посмотреть, сколько мы предложим за ваши карты. А если нет времени или просто лень, пишите нам на cards@mymagic.ru с темой письма «Хочу продать карты» - сделаем всё сами.

Товары 1 - 10 из 48627

Начало | Пред. | 1 2 3 4 5 | След. | Конец

Рисунок 4 - Функционал сайта Mymagic

2.1.5 Mtgtrade

Данный сайт представляет из себя сервис для покупки и продажи карт. Также доступен аукцион. Возможность покупать или продавать карты определяется типом аккаунта: «покупатель» - может покупать карты или выставлять их на аукцион, «продавец» - платный тип с возможностью продать карты, «трейдер» - тип аккаунта, позволяющий пользоваться автоценами и продавать запечатанную продукцию. Функционал приложения приведен на рисунке 5.

Преимущества:

- множество фильтров;
- большая пользовательская база.

Недостатки:

- для возможности продавать карты нужно заплатить;
- подтверждение аккаунта требует пользователя позвонить по телефону;
- устаревший дизайн.

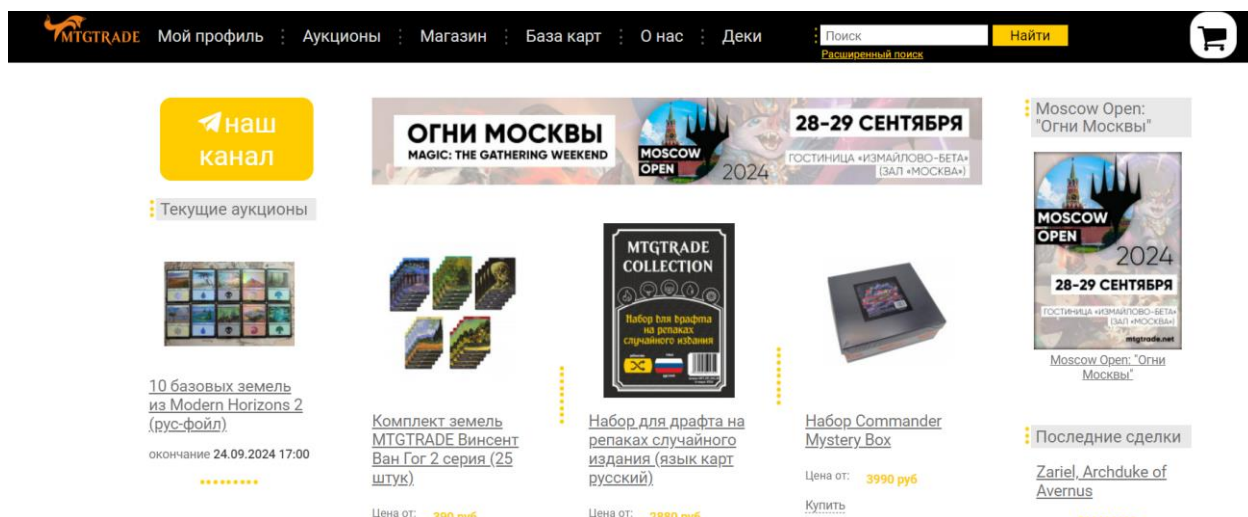


Рисунок 5 - Функционал сайта Mtgtrade

2.2 Итоги анализа

В процессе анализа предметной области было установлено, что при разработке приложения следует придерживаться следующих аспектов:

- широкий набор фильтров;
- возможность обратной связи;
- отсутствие платных функций.

Также было установлено, что необходимо избегать таких ошибок, как:

- перегруженный функционал;
- устаревший дизайн;
- неясные фильтры.

Сравнительная таблица аналогов приведена в таблице 3.

Таблица 3 - Анализ аналогов

Критерий/Название	LastSticker	TOPDeck	Avito	Mymagico	Mtgtrade	CardTrader
-------------------	-------------	---------	-------	----------	----------	------------

Специализирован под карты	Нет	Да	Нет	Да	Да	Да
Ориентация на обмен	Да	Нет	Нет	Нет	Нет	Да
Большое количество пользователей	Да	Нет	Да	Да	Нет	Нет
Множество фильтров	Да	Нет	Нет	Да	Да	Да
Премодерация	Да	Да	Да	Да	Да	Нет

3 Графическое описание работы системы

В разделе находится графическое описание системы.

3.1 Диаграммы прецедентов

Диаграмма прецедентов (Use Case) — это графическое представление функциональных требований к системе, показывающее, как различные актеры взаимодействуют с системой для достижения конкретных целей. Она помогает четко определить и описать основные сценарии использования системы и взаимодействие пользователей с ее функционалом.

3.1.1 Диаграмма прецедентов неавторизованного пользователя

Диаграмма прецедентов для неавторизованного пользователя приведена на рисунке 6.

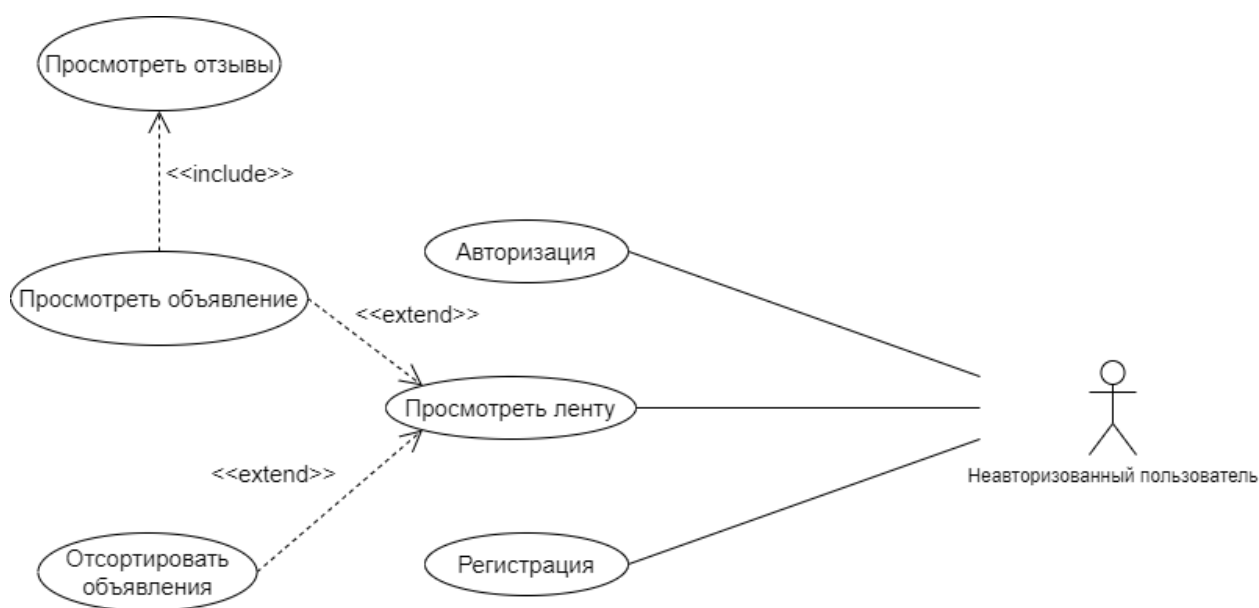


Рисунок 6 - Диаграмма прецедентов неавторизованного пользователя

3.1.2 Диаграмма прецедентов авторизованного пользователя

Диаграмма прецедентов для авторизованного пользователя приведена на рисунке 7.

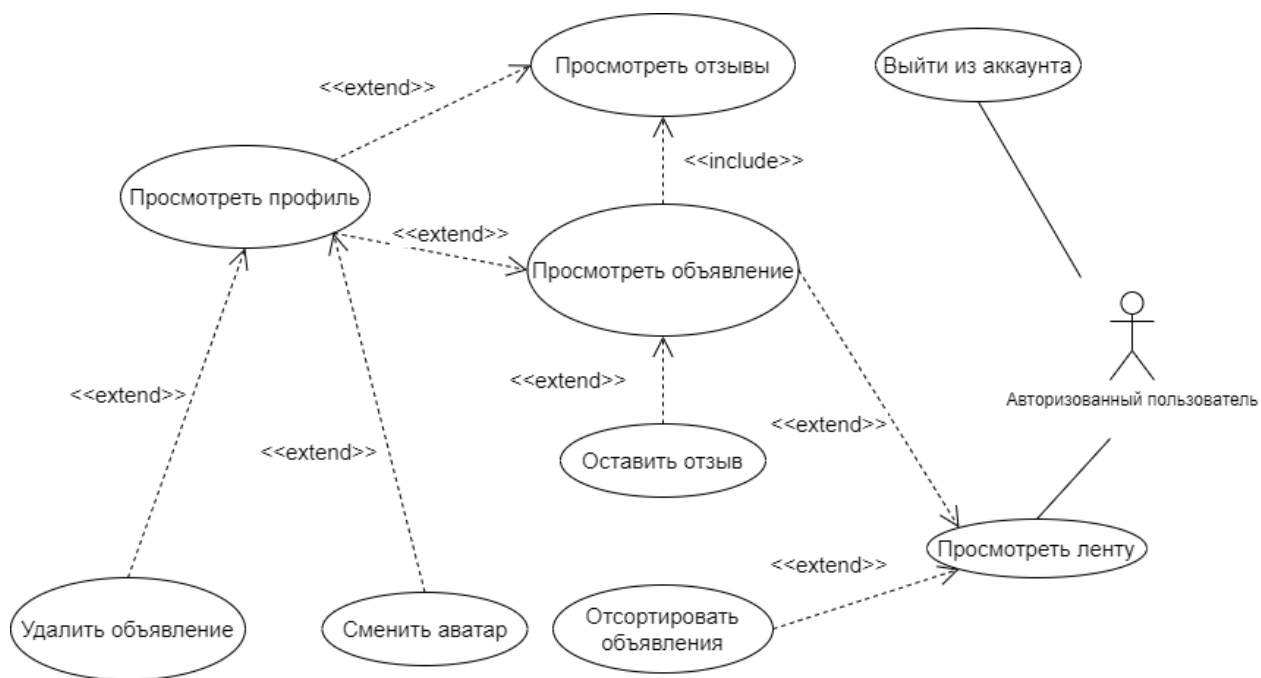


Рисунок 7 - Диаграмма прецедентов для авторизованного пользователя часть

3.2 Диаграмма развертывания

Проект разделен на 3 архитектурные составляющие: веб-приложение, серверная часть и база данных. Приложение выполнено в архитектурном стиле REST API.

Для хранения и извлечения данных сервер взаимодействует с базой данных.

С диаграммой развертывания приложения можно ознакомиться на рисунке 8.

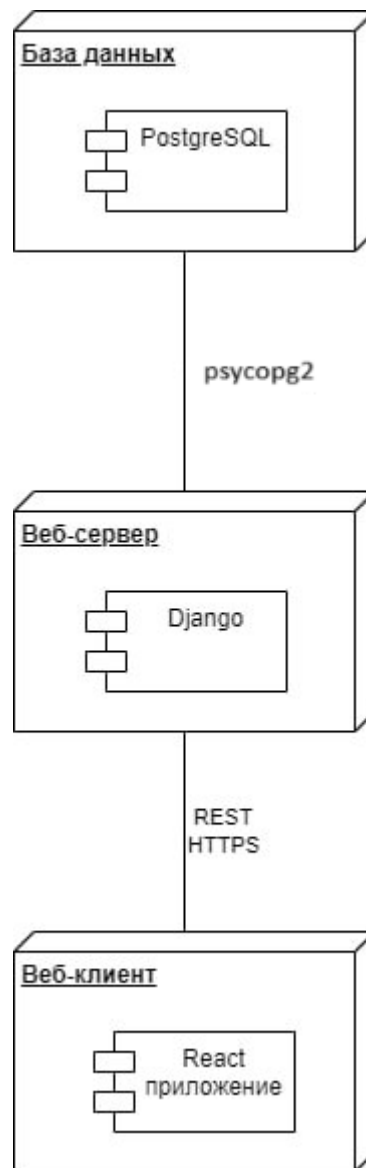


Рисунок 8 - Диаграмма развертывания приложения

3.3 Диаграммы сотрудничества

Диаграмма представляет собой графическую модель, которая показывает взаимодействие объектов в системе и сообщения между ними.



Рисунок 9 - Регистрация пользователя



Рисунок 10 - Создание объявления



Рисунок 11 - Добавление отзыва

3.4 Диаграммы последовательности

Диаграмма последовательности позволяет описать взаимодействие между объектами в системе в виде последовательности сообщений, действий и операций, отображая порядок выполнения действий и обмена информацией между объектами во времени.

3.4.1 Диаграмма последовательности для неавторизованного пользователя

Диаграммы последовательности для неавторизованного пользователя приведены на рисунках 12-13

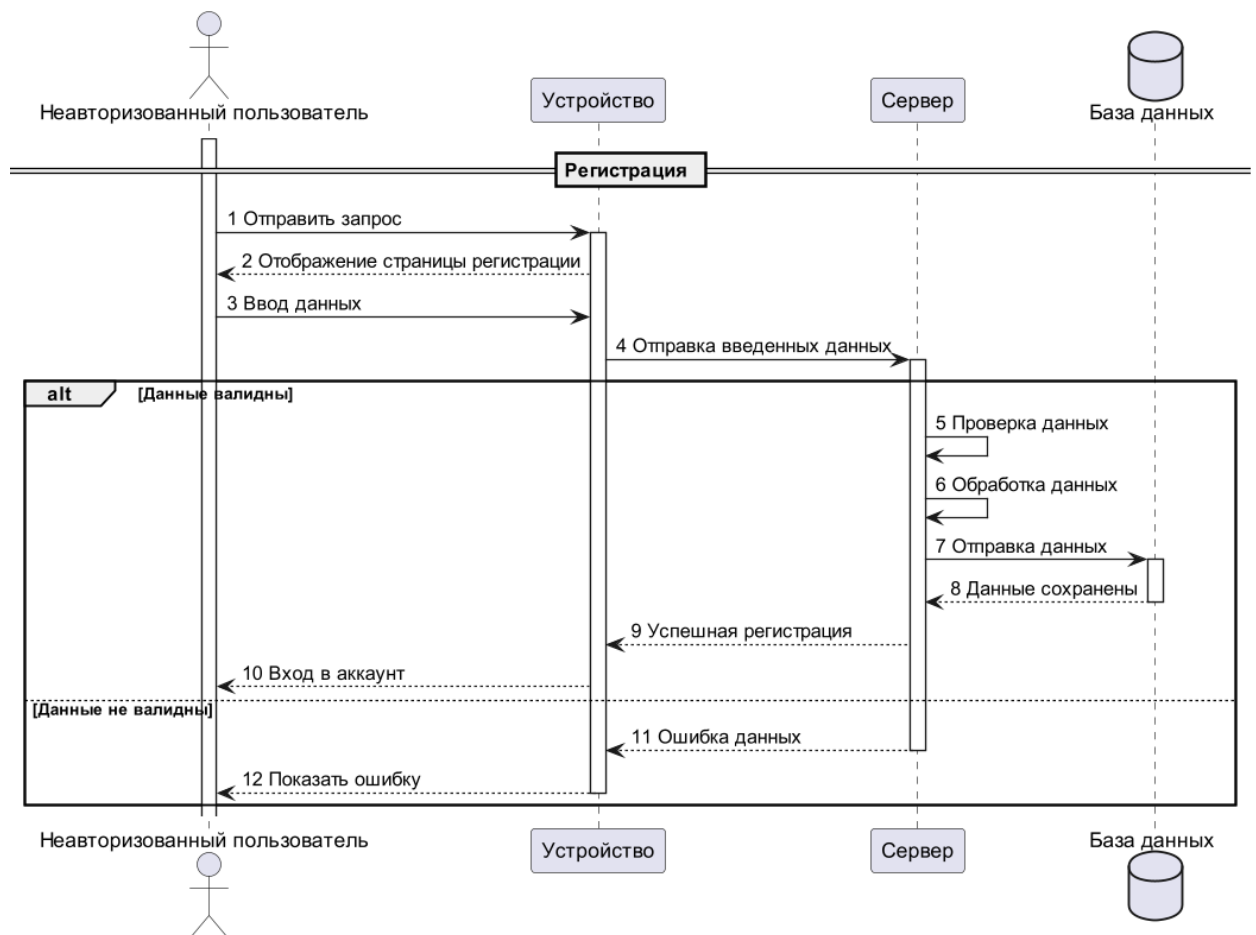


Рисунок 12 - Диаграмма последовательности для регистрации пользователя

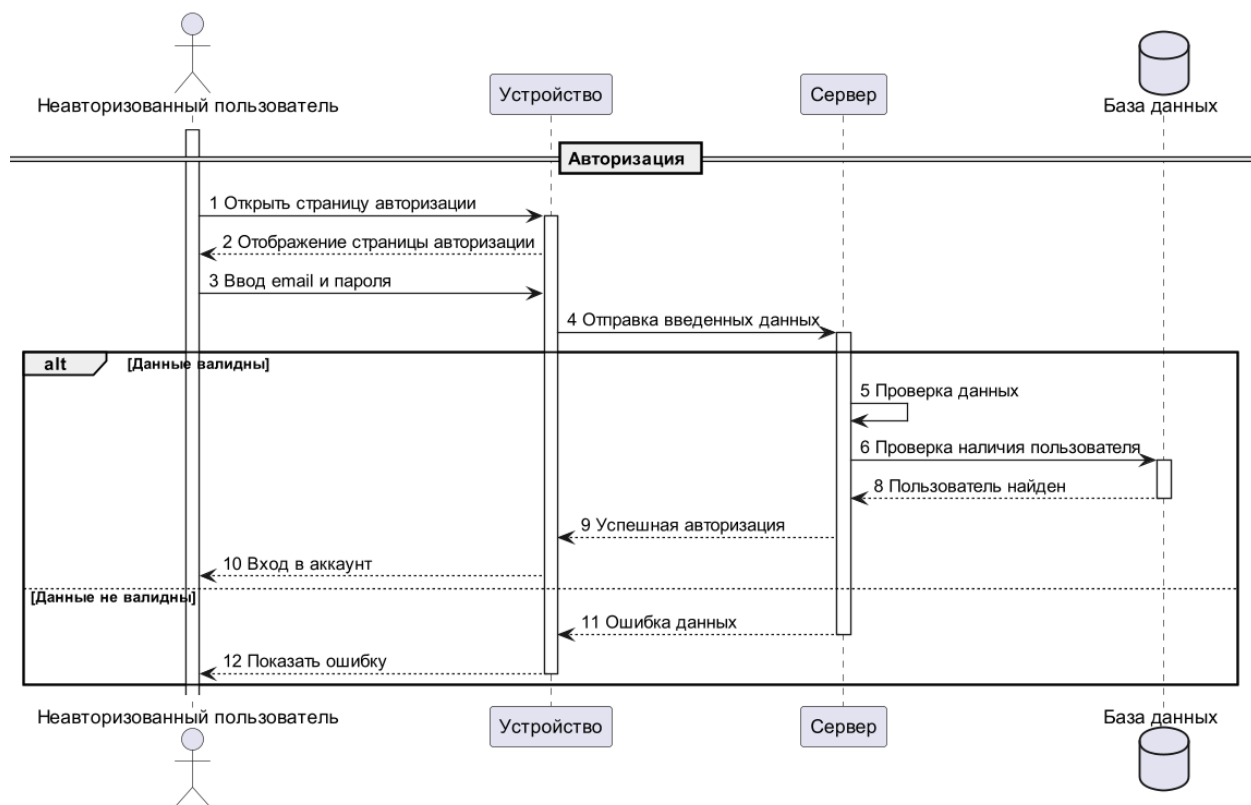


Рисунок 13 - Диаграмма последовательности для авторизации пользователя

3.4.2 Диаграмма последовательности для авторизованного пользователя

Диаграммы последовательности для авторизованного пользователя приведены на рисунках 14-**Ошибка! Источник ссылки не найден..**

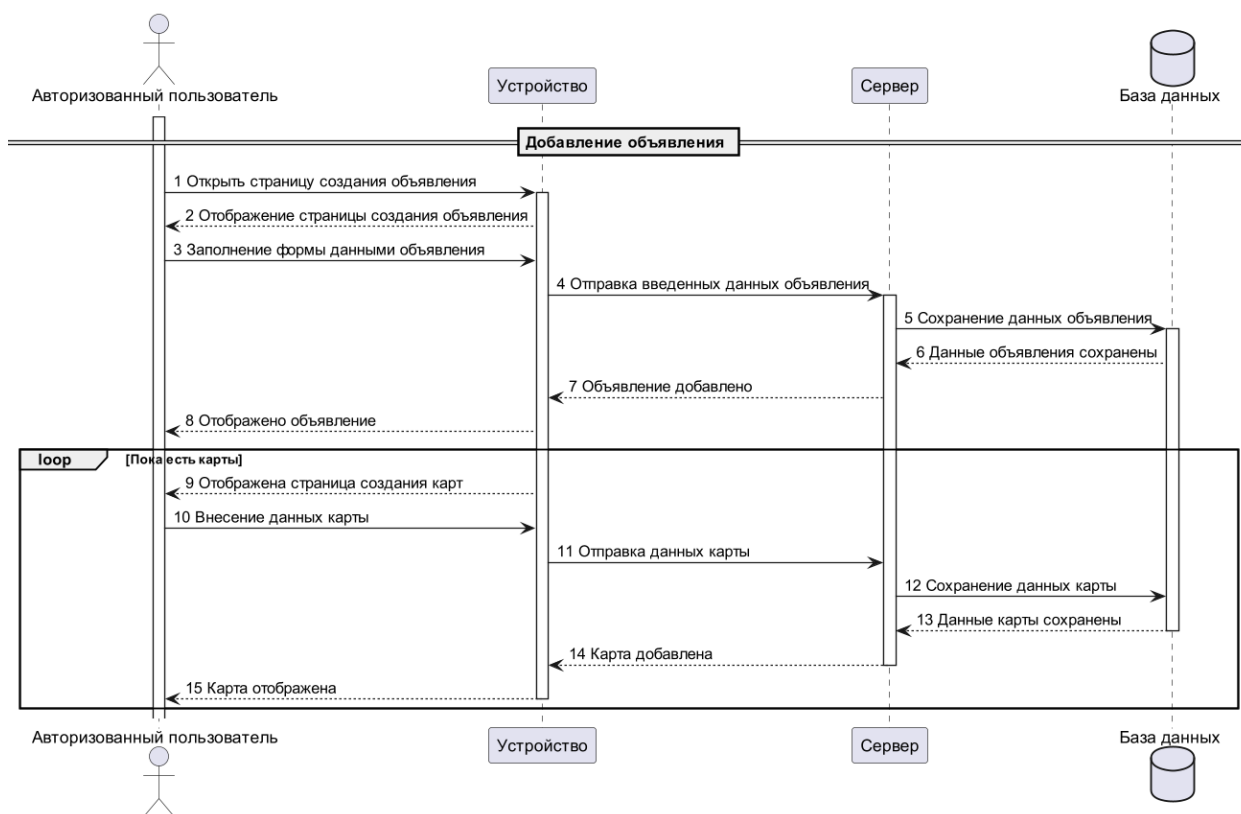


Рисунок 14 - Диаграмма последовательности для добавления объявления

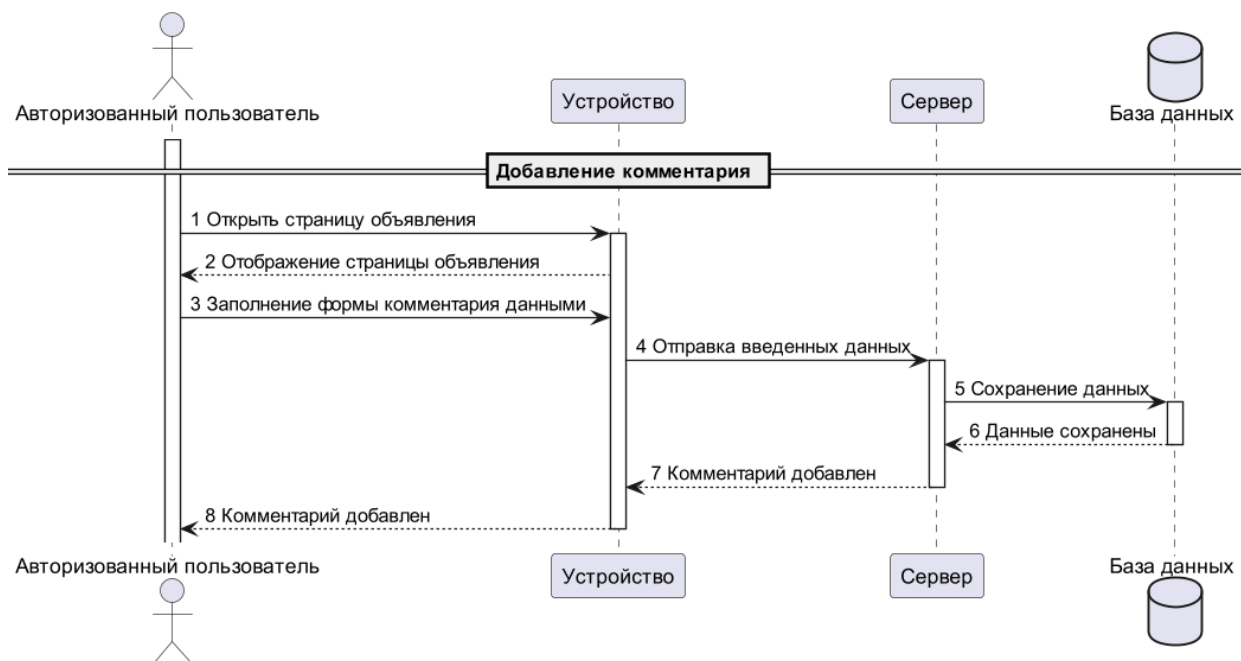


Рисунок 15 - Диаграмма последовательности для добавления комментария

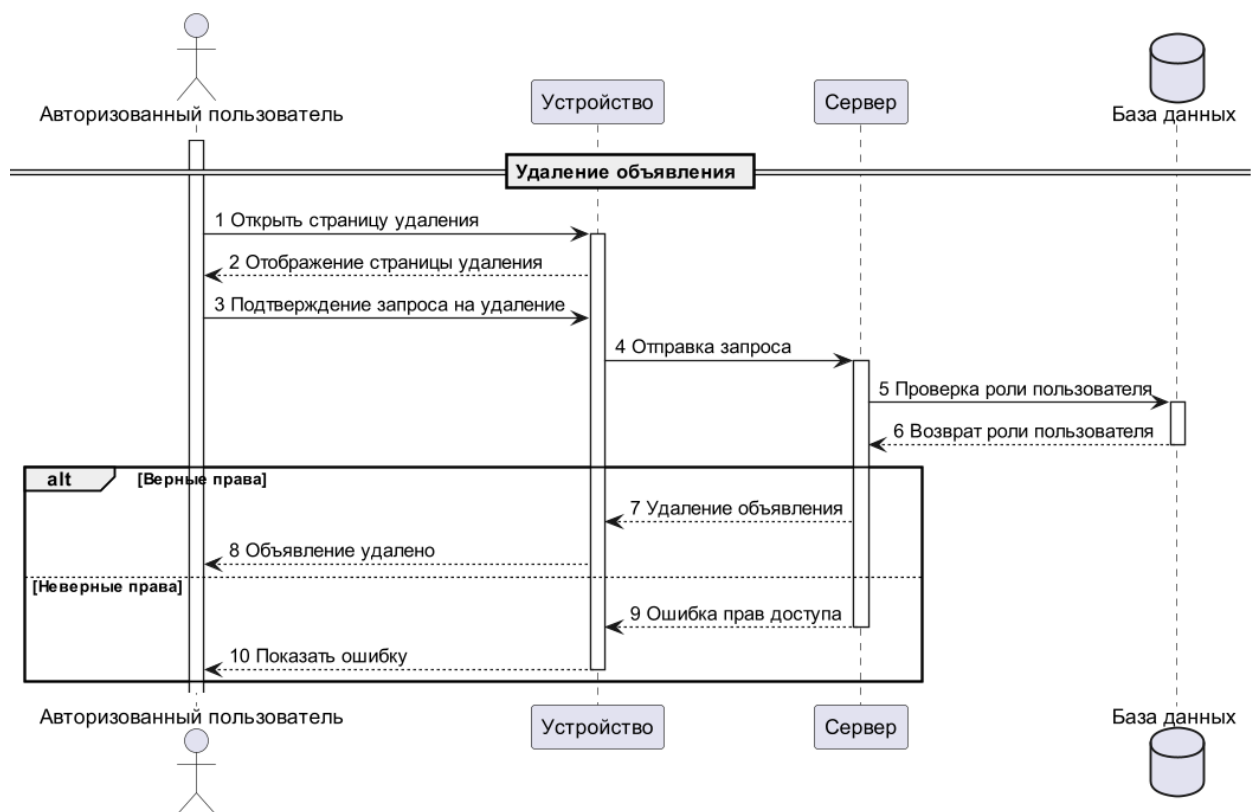


Рисунок 16 - Диаграмма для удаления объявления

3.4.3 Диаграмма последовательности для администратора

Диаграммы последовательности для администратора приведены на рисунках 14-**Ошибка! Источник ссылки не найден..**

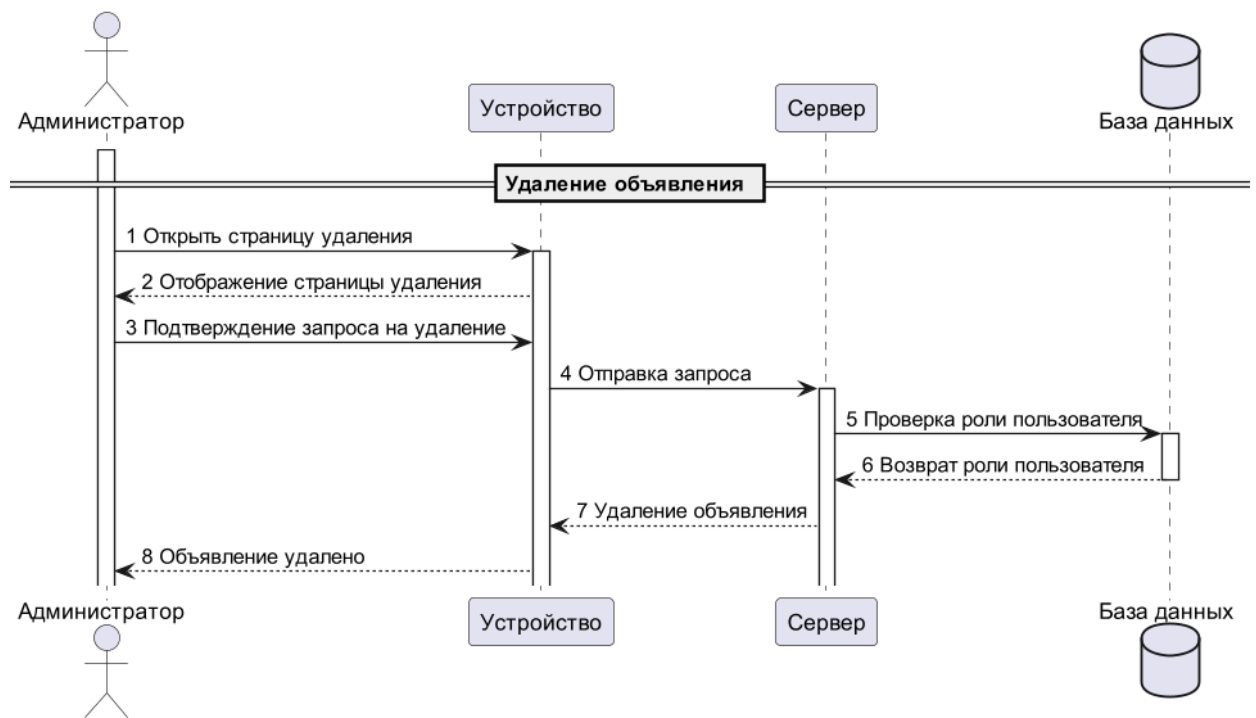


Рисунок 17 - Диаграмма для удаления объявления

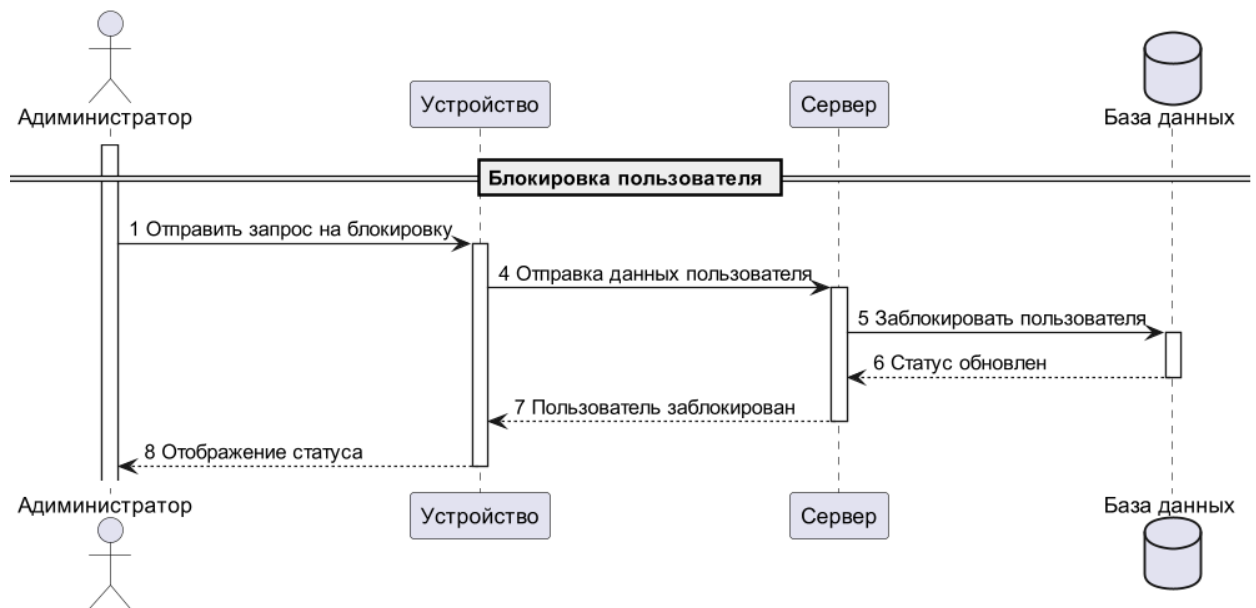


Рисунок 18 - Диаграмма для блокировки пользователя

4 Реализация

В данном разделе приведена реализация приложения.

4.1 Средства реализации

Приложение соответствует архитектуре клиент-сервер. Для реализации серверной части были выбраны следующие средства:

- ОС Windows 10 / Ubuntu 22.10;
- фреймворк Django 5.0.9 [3];
- Python версии 3;
- СУБД PostgreSQL 16 [4];
- объектное хранилище данных MinIO [5];
- система контроля версий Git 2.42.0 [6].

Для серверной части нашего проекта был выбран язык программирования Python в связке с фреймворком Django и СУБД PostgreSQL. Для хранения изображений было выбрано объектное хранилище данных MinIO.

Этот выбор обусловлен несколькими факторами:

- фреймворк Django предоставляет мощные инструменты для быстрого прототипирования и разработки веб-приложений, также имеет встроенные средства для аутентификации, работы с базами данных и управления пользователями;
- выбор PostgreSQL в качестве системы управления базами данных обусловлен его надежностью, производительностью и расширяемостью;
- MinIO выбрано для хранения изображений из-за его высокой производительности и совместимости с S3 API для быстрой и надежной работы с большими объемами данных.

В качестве средств реализации клиентской части были выбраны:

- язык верстки HTML 5
- язык описания внешнего вида документа CSS3
- Javascript версии 1.8.5
- фреймворк React версии 18.2.0
- IntelliJ Idea 2024.1
- система контроля версий Git 2.42.0.

Javascript был выбран для разработки веб-приложения благодаря его современному синтаксису, распространенности и большой базе технической документации.

4.2 Реализация базы данных

На рисунке 19 представлена ER диаграмма используемой на проекте базы данных.

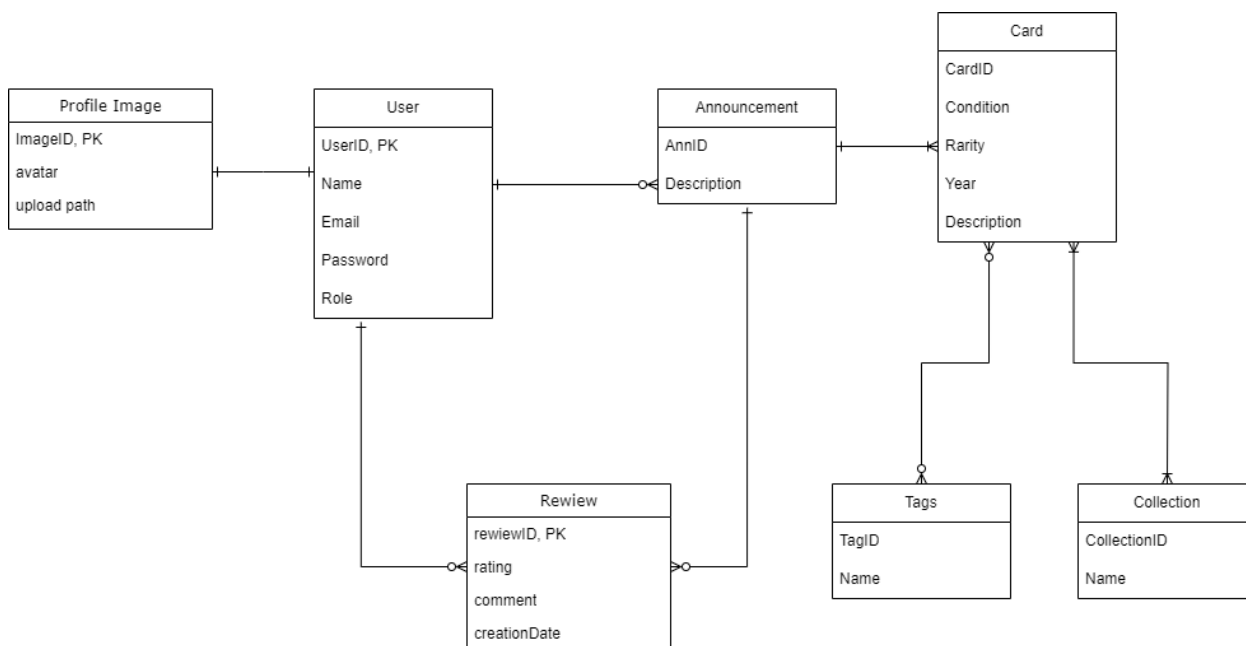


Рисунок 19 - ER диаграмма

Как видно из диаграммы, база данных содержит восемь сущностей. Далее приведена краткая информация о каждой сущности.

`authorization_user` — это пользователь, который может быть обычным пользователем (`user`) или администратором (`admin`). Сущность

authorization_user связана с сущностями announcements_announcement (объявления) и announcements_comments (отзывы) связями 1, 1: 0, n. Поскольку у пользователя может быть несколько объявлений и отзывов, либо не быть вовсе, но у каждого объявления и отзыва должен быть создавший его пользователь.

cards_card – это карты, которые содержатся в объявлении. В объявлении обязательно есть одна или несколько карт, а у каждой карты есть одно объявление.

cards_collection – это коллекции, которые являются одним из фильтров для карт. В одной коллекции может быть много карт, но у карты может быть лишь одна коллекция.

cards_tags – это теги, фильтры создаваемые пользователем при размещении объявления. У каждой карты может быть много тегов, у каждого тега есть одна или более карт.

cards_card_tags – сущность созданная Django для реализации связи многие-ко-многим. Содержит в себе id тегов и карт в качестве внешних ключей.

announcements_announcement – объявления, созданные пользователем. Содержат в себе информацию об объявлении, контактные данные владельца.

На рисунке 20 представлена физическая модель базы данных.

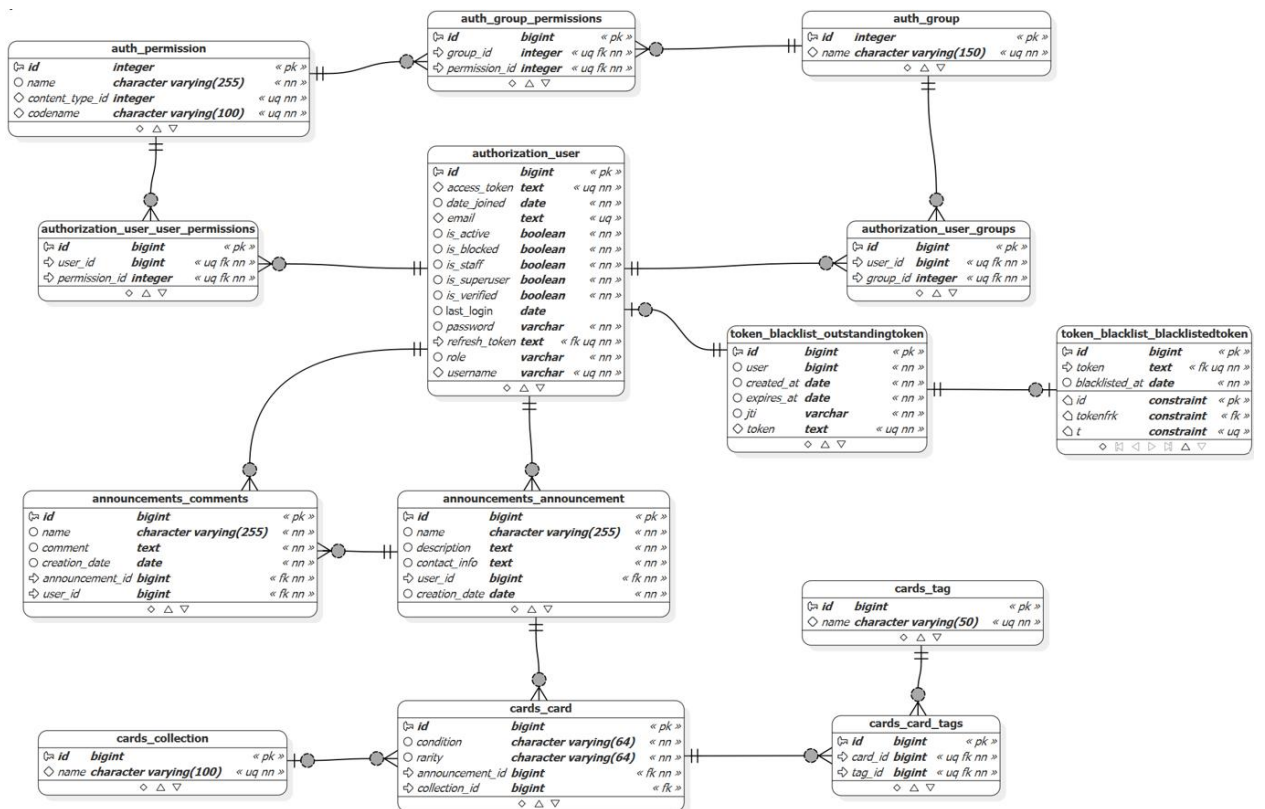


Рисунок 20 - Физическая модель базы данных

4.3 Реализация серверной части приложения

Серверная часть приложения реализована на основе Spring Boot и включает следующие компоненты:

- модели (model) сущностей базы данных для определения полей таблицы и связей между таблицами;
- сериализаторы (serializers), с помощью которых модели преобразуются в JSON и обратно;
- представления (views) которые принимают HTTP запросы и обеспечивают CRUD-операции;
- url адреса, используемые для маршрутизации приложения;
- конфигурационный файл (Settings), используемая для настройки параметров приложений, таких как CORS, соединение с базой данных, подключение библиотек;

- `minio_backend`, используемый как компонент для работы с MinIO;
- `Email Component`, служащий для отправки электронных писем, используемый для подтверждения пользователя;
- `Dockerfile` и `docker-compose.yaml`, с помощью которых проект на сервере можно развернуть в среде `docker`;
- конфиг `nginx`, который позволяет запустить `nginx` для проксирования запросов HTTPS.

4.4 Реализация клиентской части приложения

Данный раздел содержит описание реализации клиентской части.

4.4.1 Общая информация

Клиентская часть приложения реализует пользовательский интерфейс, формирует запросы к серверу и обрабатывает ответы от него. Клиентская часть содержит в себе следующие структурные компоненты:

- пакет со всеми компонентами. В нём содержатся компоненты: авторизация, аватар, контейнер комментариев, фильтры, навигационная панель, превью объявления, пользовательские действия, регистрация а также все компоненты, использующиеся для передачи данных в запросах и получении данных (`Components`);
- пакет, содержащий все страницы в которых используются различные компоненты и обрабатываются запросы к эндпоинтами (`api`) в соответствии с особенностями для конкретной страницы. В числе страниц: страница объявления, авторизации и регистрации, создания объявления, страница удаления объявлений для администратора, главная, поиска результатов, страница профиля пользователя (`Pages`);
- пакет с логикой обрабатывающей навигацию пользователя по сайту и отслеживающий состояние авторизации (`src`);

4.4.2 Графический интерфейс

Далее представлены экраны приложения.

Главная страница содержит навигационную панель, кнопку меню для перехода на основные страницы, а именно регистрацию, авторизацию, личный кабинет, выход, переход на главную страницу, объявления, поиск объявлений, кнопки «показать больше» и «Вернуться» (по умолчанию скрыто на первой странице объявлений). Главная страница представлена на рисунке 21.

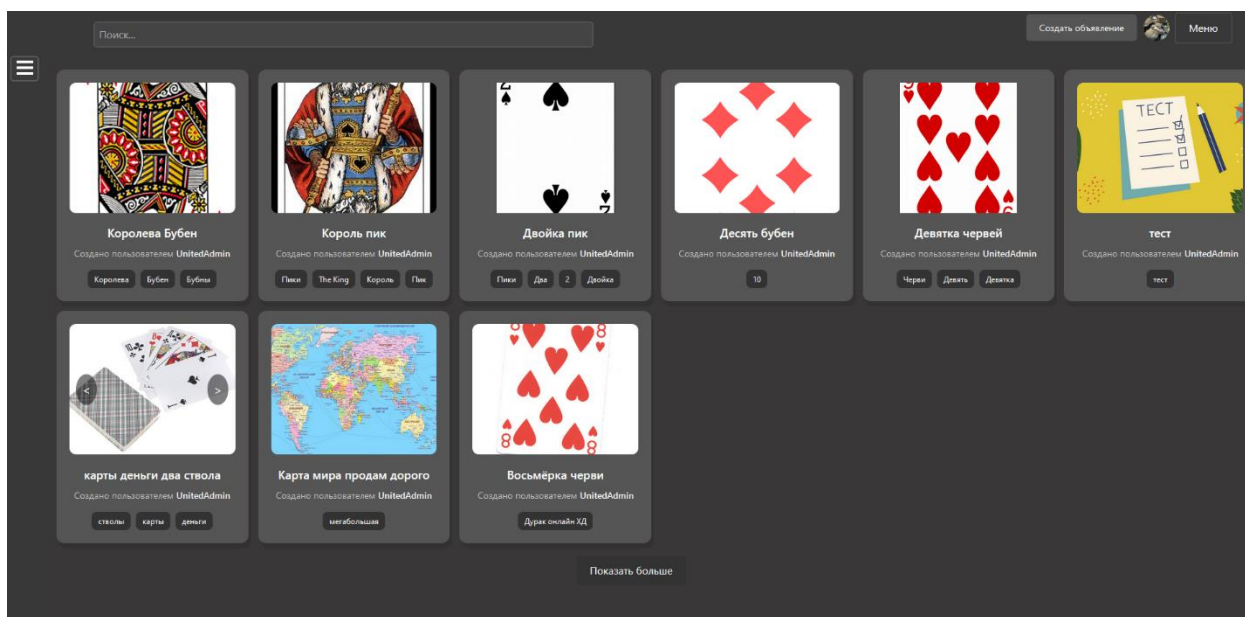


Рисунок 21 - Главная страница

При нажатии на иконку меню слева на главной странице появляется боковое меню фильтров. Возможно выбрать одно состояние, одну редкость и тэги для фильтрации показа объявлений на главной. Боковое меню представлено на рисунке 22.

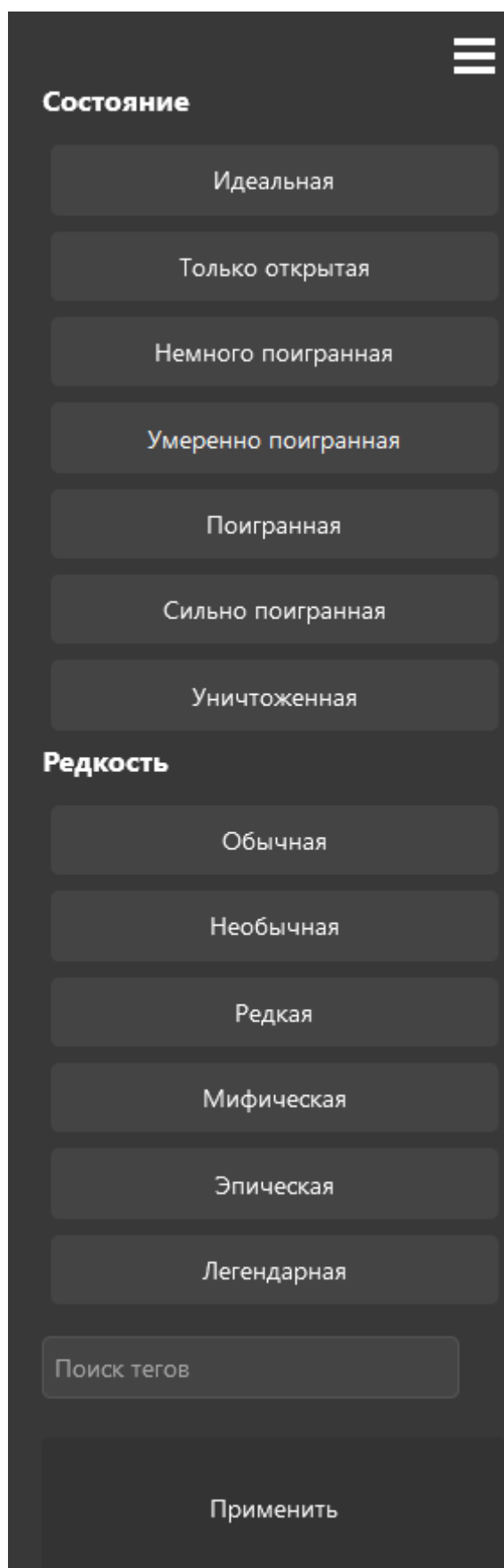


Рисунок 22 - Боковое меню

При нажатии на кнопку «Создать объявление» пользователь перейдет на страницу для создания объявления. Она состоит из нескольких вкладок для карт в объявлении, количество которых задаёт пользователь, создавая новые вкладки карт, и вкладки для общей информации объявления. На вкладке карты пользователю доступна загрузка картинки или фотографии своей карты и ввод тэгов для неё. На вкладке общей информации объявления для создания он должен ввести название объявления, описание, контактную информацию, редкость и качество. Страницы создания объявления представлены на рисунке 23.

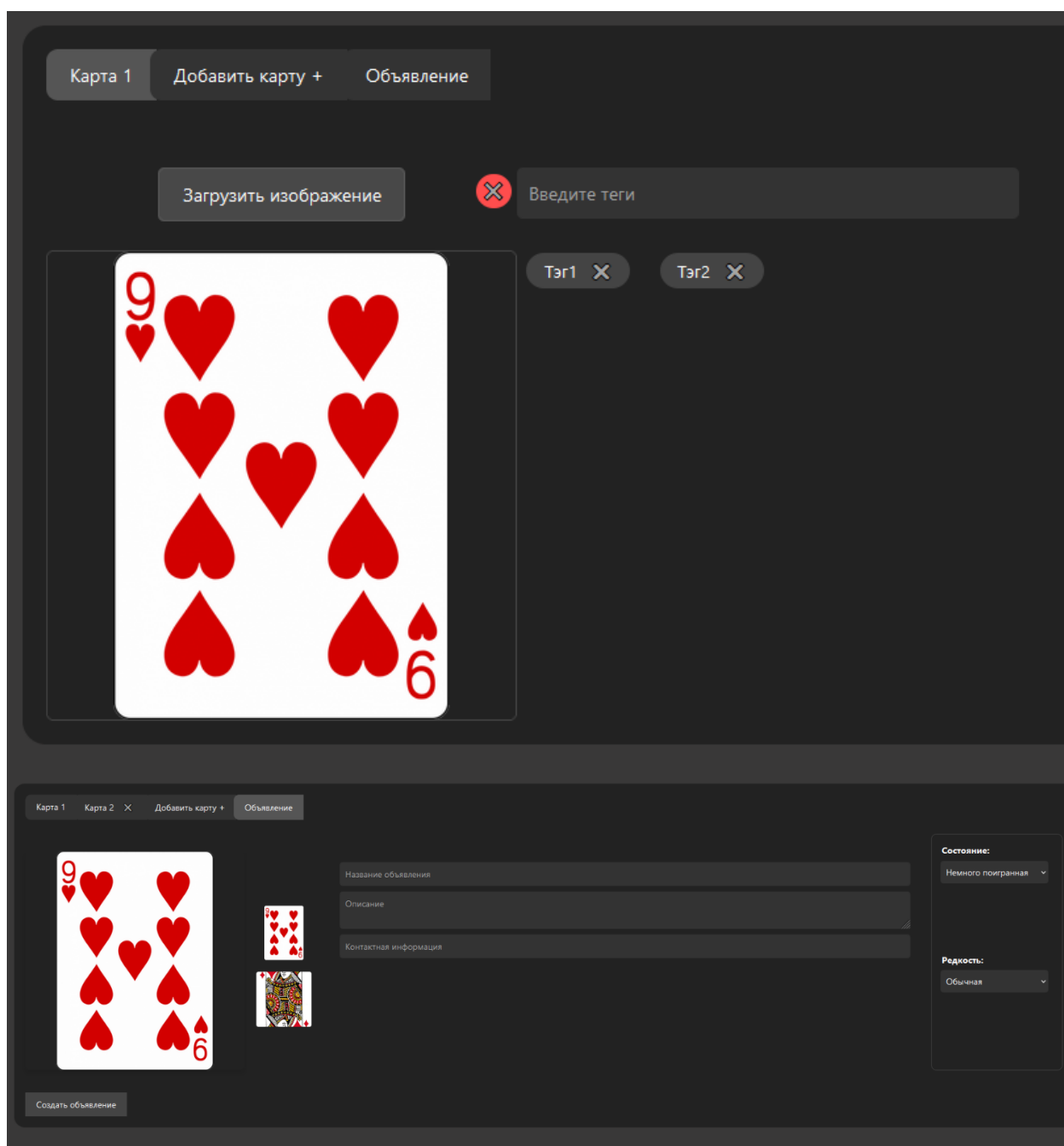


Рисунок 23 – Страницы создания объявлений

На странице «Личный кабинет» пользователь может посмотреть свои объявления, удалить их, сменить или удалить изображение на своём аватаре, там отображается количество отзывов под его объявлениями, последние комментарии к ним. Также администратор может заблокировать пользователя. Страница для просмотра профиля другого пользователя выглядит аналогично, за исключением отсутствия возможности взаимодействовать с аватаром и удалять объявления. Страницы пользователей представлены на рисунке 24.

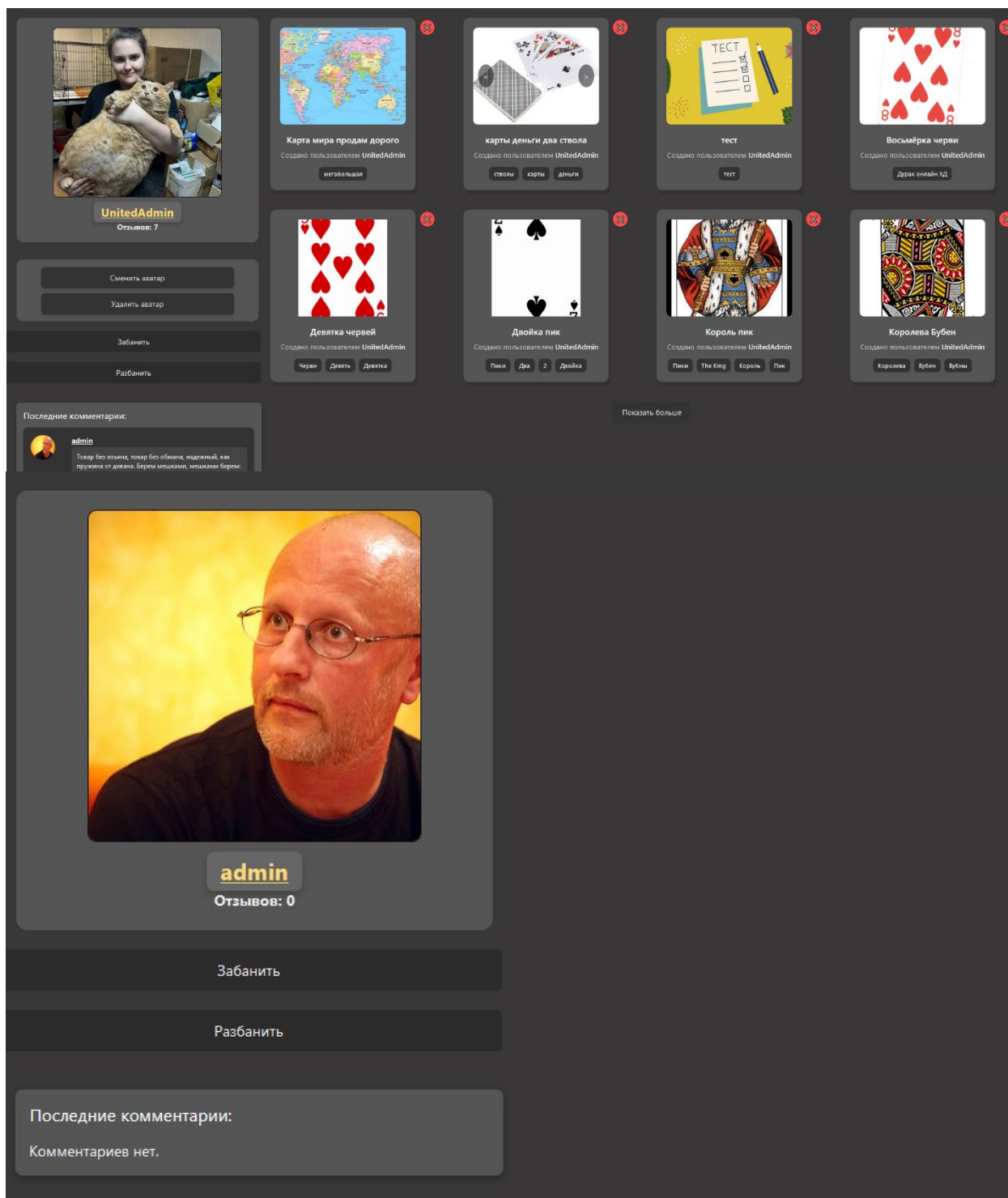


Рисунок 24 - Страницы пользователей

На странице объявления отображается основная информация, изображения карт, информация об авторе объявления, возможность редактировать его если пользователь просматривает своё объявление. Снизу страницы отображаются комментарии к объявлению. Страница объявления представлены на рисунке 25.

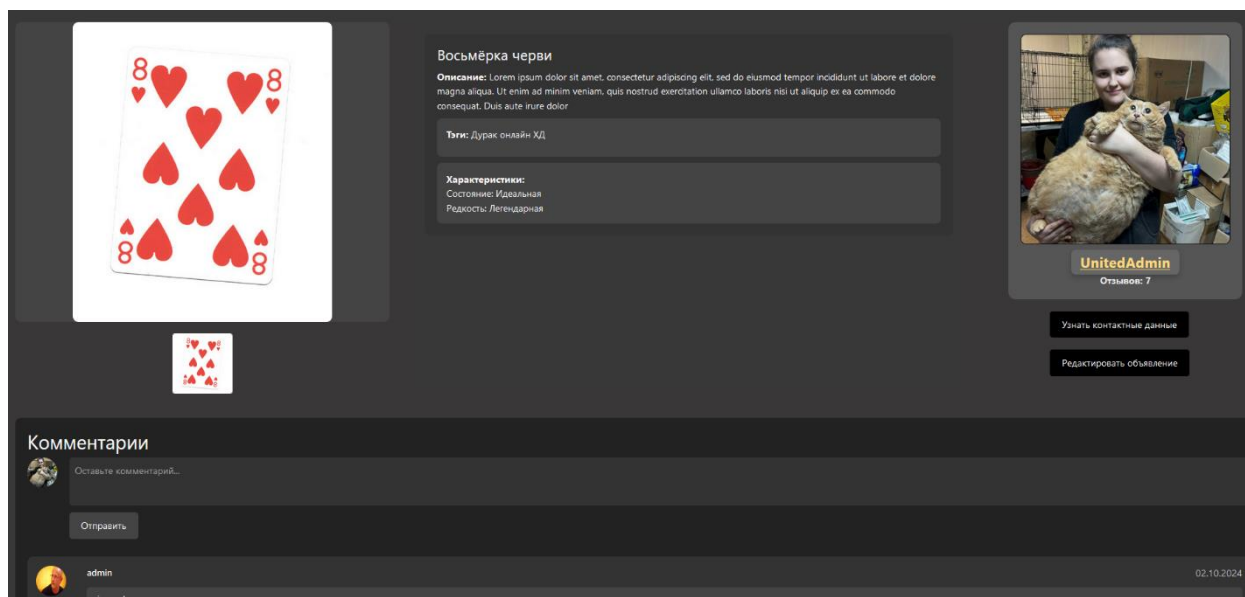


Рисунок 25 - Страница объявления

На странице регистрации пользователь должен ввести запрошенные данные. После этого на его почту отправляется письмо для подтверждения аккаунта. Страница авторизации выглядит аналогично аз исключением ввода логина. После авторизации пользователя перебрасывает на главную страницу. Страницы регистрации и авторизации представлена на рисунке 26.

Регистрация

Регистрация

Регистрация прошла успешно!

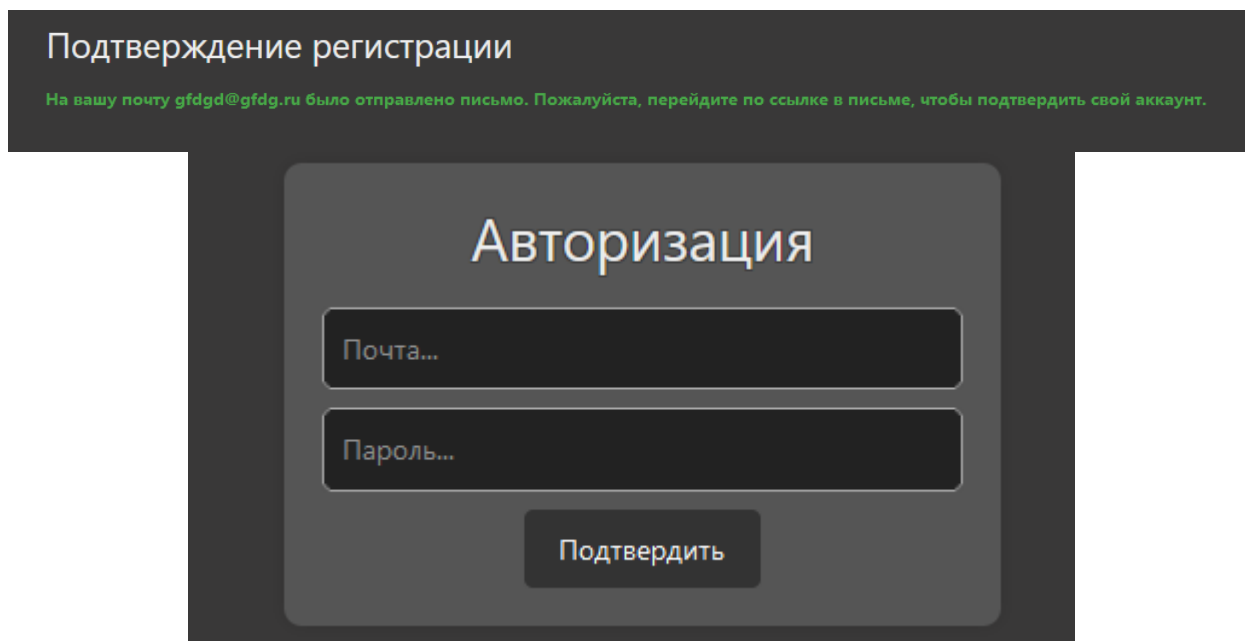


Рисунок 26 - Страницы регистрации и авторизации

4.5 Методология разработки

При реализации проекта была использована каскадная модель (Waterfall) представленная на рисунке 27.

Переход между фазами возможен только после полного и успешного завершения предыдущей. Сначала полностью завершается первый этап ("Анализ требований"), в результате которого появляется полный и исчерпывающий список требований к информационной системе. Только после этого возможен переход к проектированию, в ходе которого разрабатываются дизайн, прототип и отдельные элементы приложения. Далее начинается этап реализации требований в виде написания программного кода системы. После того как перечисленные этапы завершены, производится тестирование и последующая отладка продукта. Тут устраняются все недочеты и ошибки, появившиеся ранее. После этого программный продукт внедряется и осуществляется его поддержка - разработка новой функциональности, устранение ошибок и т.д.

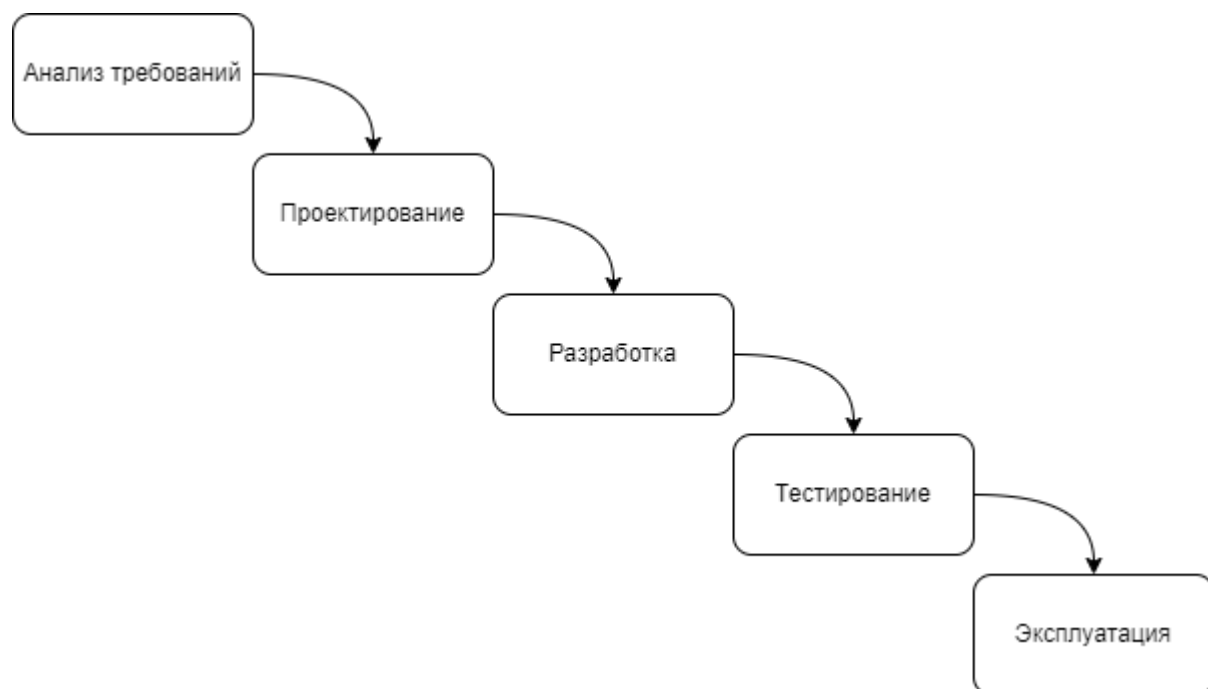


Рисунок 27 - Каскадная модель

5 Тестирование

В разделе находится информация о тестировании системы.

5.1 Дымовое тестирование

Для дымового тестирования необходимо проверить работоспособность основных сценариев. Основные сценарии:

- регистрация;
- авторизация;
- открытие бокового меню;
- добавление объявления;
- добавление комментария;
- поиск объявления по фильтрам.

В таблице 4 представлены результаты тестирования.

Таблица 4 - Результаты тестирования

Сценарий	Результат
Регистрация	Пройден
Авторизация	Пройден
Открытие бокового меню	Пройден
Поиск объявления по фильтрам	Пройден

5.2 UI-тестирование

В таблице 5 приведены результаты UI-тестирования.

Таблица 5 - результаты UI-тестирования

Шаги теста	Ожидаемый ответ	Результат
Попытка входа при некорректных входных данных	Вход не удался	Пройден
Попытка входа при корректных входных данных	Открытие главной страницы	Пройден
Попытка регистрации при корректных данных	Открытие страницы входа	Пройден
Открытие бокового меню если пользователь авторизован	Открытие бокового меню	Пройден
Нажатие на кнопку «Главная»	Откроется главная страница	Пройден
Попытка просмотра профиля неавторизованным пользователем	Откроется ошибка доступа	Пройден
Попытка просмотра профиля неавторизованным пользователем	Откроется страница профиля	Пройден

Шаги теста	Ожидаемый ответ	Результат
Попытка открыть страницу удаления объявлений не администратором	Откроется ошибка доступа	Пройден

В результате все сценарии UI-тестирования были пройдены успешно.

ЗАКЛЮЧЕНИЕ

В ходе реализации приложения были достигнуты поставленные задачи,
а именно:

- регистрация и авторизация;
- создание, редактирование, удаление и просмотр объявлений;
- возможность поиска с применением фильтров;
- просмотр отзывов для авторизованных пользователей;
- возможность удалять чужие объявления и блокировать пользователей для администраторов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. ГОСТ 7.32-2017. Система стандартов по информации, библиотечному и издательскому делу. Структура и правила оформления: [сайт]. – URL: https://cs.msu.ru/sites/cmc/files/docs/2021-11gost_7.32-2017.pdf – Текст: электронный.
2. В.С. Тарасов, Д.И. Шмойлов, А.В. Москаленко «Методическое пособие Проектирование технического задания». - Воронеж Издательский дом ВГУ, 2024 – Текст: электронный.
3. Документация Django: [сайт]. – URL: <https://docs.djangoproject.com/en/5.1/> – Текст: электронный.
4. Документация PostgreSQL: [сайт]. – URL: <https://postgrespro.ru/docs/postgresql/16/index> – Текст: электронный.
5. Документация MinIO: [сайт]. – URL: <https://min.io/docs/minio/windows/index.html> – Текст: электронный.
6. Система контроля версий Git: [сайт]. – URL: <https://ru.wikipedia.org/wiki/Git> – Текст: электронный.
7. Документация React: [сайт]. – URL: <https://react.dev/learn> – Текст: электронный.