МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «ВГУ»)**

Математический факультет

СПО «Программирование в компьютерных системах» 09.02.03

Курсовой проект по дисциплине:

Инструментальные средства разработки ПО

«Платформа для покупки и продажи курсовых и дипломных работ «STASP»»

Обучающийся\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Мгалоблишвили А.

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Богатырев И. А.

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Ишкова М. А.

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Петросян В. В.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Зенин К. В

Воронеж 2023

Содержание

[**Введение** 3](#_Toc135179923)

[**1.** **Постановка задачи.** 4](#_Toc135179924)

[**1.1. Требования к разрабатываемой системе.** 4](#_Toc135179925)

[**1.1.1. Функциональные требования.** 4](#_Toc135179926)

[**1.1.2. Технические требования.** 4](#_Toc135179927)

[**1.2. Требования к интерфейсу.** 4](#_Toc135179928)

[**1.3. Задачи, решаемые в процессе разработки.** 4](#_Toc135179929)

[**2. Анализ предметной области.** 6](#_Toc135179930)

[**2.1. Глоссарий.** 6](#_Toc135179931)

[**2.2. Анализ задачи.** 6](#_Toc135179932)

[**2.3. Входные-выходные данные.** 7](#_Toc135179933)

[**2.4. Обзор аналогов.** 8](#_Toc135179934)

[**2.4.1. VSESDAL.** 8](#_Toc135179935)

[**2.4.2. Решатель.** 9](#_Toc135179936)

[**3. Реализация.** 10](#_Toc135179937)

[**3.1. Средства реализации.** 10](#_Toc135179938)

[**3.2. Реализация серверной части приложения.** 11](#_Toc135179939)

[**3.2.1. Панель администратора.** 11](#_Toc135179940)

[**3.2.2. Приложение слайдов.** 12](#_Toc135179941)

[**3.2.3. CRM-система.** 12](#_Toc135179942)

[**3.2.4. Слой доступа к данным пользователей и групп.** 13](#_Toc135179943)

[**3.2.5. Миграции.** 13](#_Toc135179944)

[**3.3. Реализация клиентской части приложения.** 14](#_Toc135179945)

[**3.3.1. Макеты интерфейса.** 14](#_Toc135179946)

[**3.3.2. Макет интерфейса администратора.** 17](#_Toc135179947)

[**Заключение** 18](#_Toc135179948)

[**Список использованных источников** 19](#_Toc135179949)

# **Введение**

Многие студенты сталкиваются с проблемой подготовки курсовых проектов, дипломных работ и различных задач. Различные причины заставляют обучающихся обращаться за помощью к интернет-источникам, знакомым и одногруппникам. И не всегда результат оказывается подходящим для оценки на «высший балл».

Помощь студентам в написании курсовых и дипломных проектов является достаточно востребованной и распространенной услугой в современном образовательном пространстве. Однако, следует помнить, что использование таких услуг может иметь негативные последствия для саморазвития студента и их дальнейшей профессиональной карьеры.

В нашем сервисе вы можете заказать написание курсовых и дипломных проектов по любой дисциплине и уровню сложности. Мы гарантируем полное соответствие техническим требованиям и срокам выполнения работ.

Целью нашей работы является создание удобного сервиса с точки зрения интерфейса и качественного выполненного задания.

1. **Постановка задачи.**

Целью данного проекта является разработка платформы для для покупки и продажи курсовых и дипломных работ, под названием: «STASP».

## **1.1. Требования к разрабатываемой системе.**

### **1.1.1. Функциональные требования.**

К разрабатываемому приложению выдвигаются следующие требования:

- Возможность просмотра тарифов;

- Связь с продавцом через удобный мессенджер;

- Возможность оставить заявку на заказ;

- Поддержка русского языка;

### **1.1.2. Технические требования.**

Приложение должно обеспечивать:

- Переход в телеграмм-бот, после оставления заявки;

- Работающие ссылки на мессенджеры.

## **1.2. Требования к интерфейсу.**

Список требований к интерфейсу:

- Выполнен в едином стиле и цветовой гамме;

- Все надписи должны быть легко читаемы;

- Все элементы управления должны выделяться на фоне содержимого;

- Должен корректно отображаться при изменении размеров экрана;

- Содержать только необходимую для пользователя информацию.

## **1.3. Задачи, решаемые в процессе разработки.**

Были поставлены следующие задачи:

- Анализ предметной области;

- Анализ аналогов;

- Разработка макетов интерфейса;

- Определение используемой платформы;

- Написание технического задания;

- Реализация сервера;

- Реализация интерфейса;

- Реализация телеграм-бота;

- Описание процесса разработки и результата.

# **2. Анализ предметной области.**

## **2.1. Глоссарий.**

**Проект** – разрабатываемое приложение.

**Сервер, серверная часть** – компьютер, обслуживающий другие устройства (клиентов) и предоставляющий им свои ресурсы для выполнения определенных задач.

**Клиент, клиентская сторона** – в данном проекте - сайт, который предоставляет возможности пользователю взаимодействовать со всей системой.

**Front-end** – клиентская часть приложения. Отвечает за получение информации с программно-аппаратной части и отображение ее на устройстве пользователя.

**Back-end** – программно-аппаратная часть приложения. Отвечает за функционирование внутренней части приложения;

**GitHub** – веб-сервис для хостинга IT-проектов и их совместной разработки.

**Пользователь** – авторизованный в системе человек, пользующийся функционалом приложения.

**Администратор** – пользователь, у которого есть привилегии.

**Гость** – человек, не имеющий учетной записи, может только зарегистрироваться или авторизоваться.

**Телеграм-бот** — это компьютерная программа, которая создана для автоматизации определенных задач в мессенджере Telegram. Она может выполнять различные функции, такие как отправка сообщений, рассылка информации, обработка запросов и т. д.

## **2.2. Анализ задачи.**

На Рисунке 1 продемонстрирована диаграмма UseCase, которая указывает какие сценарии использования приложения доступны пользователю. Обычный пользователь имеет 3 сценария работы с приложением.

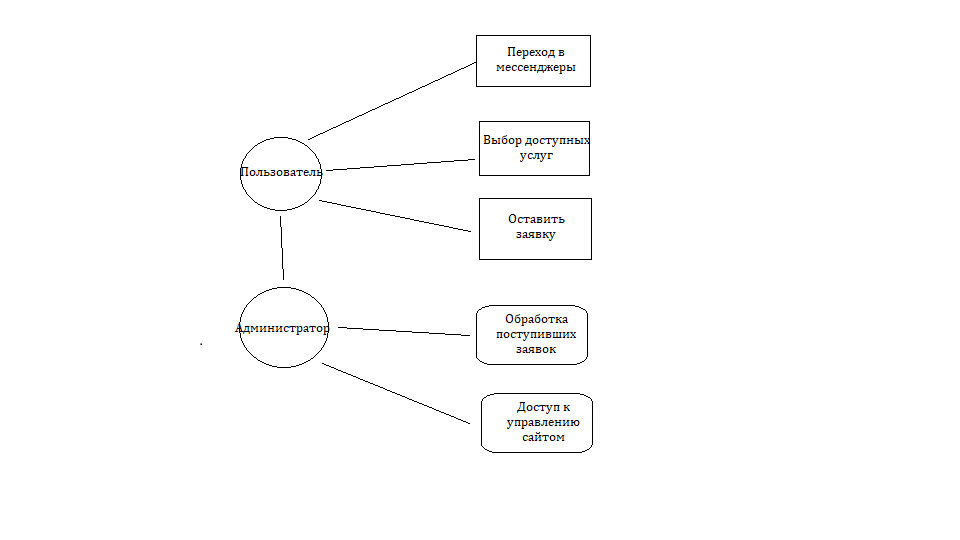


Рисунок 1 - UseCase диаграмма.

## **2.3. Входные-выходные данные.**

Рассмотрим основной бизнес-процесс на примере IDEF0 диаграммы, представленной на Рисунке 2. Данная диаграмма представляет собой общее видение процесса работы веб-приложения.

Работу сервиса регулирует законодательство Российской Федерации. Обеспечивает работу приложения БД с покупателями и продавцами. На вход в систему поступает Пользователь.

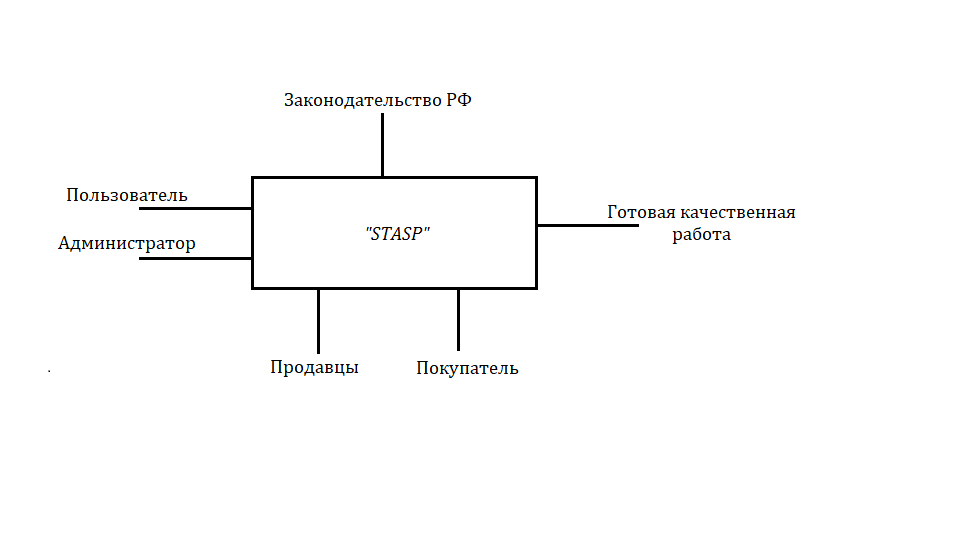


Рисунок 2 - IDEF0 диаграмма.

## **2.4. Обзор аналогов.**

Приступая к разработке, приложения необходимо проанализировать проекты, уже существовавшие до него. Рассмотрим их достоинства, недостатки. Насколько они удобны в использовании, содержат ли необходимую функциональность.

И на основе этого анализа сделаем выводы какие моменты нужно учесть при разработке своего приложения, а именно каким образом будет построен удобно для пользователя интерфейс, и какую функциональность нужно обеспечить.

### **2.4.1. VSESDAL.**

На Рисунке 3 представлен сайт с похожими представленными услугами. В нем есть необходимый функционал для заказа работы. Можно выбрать конкретный предмет. Он эффективен и прост в использовании. В общем, интегрированное решение для написания работы.

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 3 - Внешний вид VSESDAL.

Достоинства:

- Большой выбор готовых работ;

- Разные направления и предметы;

- Простой интерфейс.

Недостатки:

- Нет цен в общем доступе;

- Необходима регистрация;

### **2.4.2. Решатель.**

Сайт для написания различных типов работ, включая: переводы текстов с разных языков, создания презентаций, решения задач, написания контрольных и курсовых, дипломов и т. д.

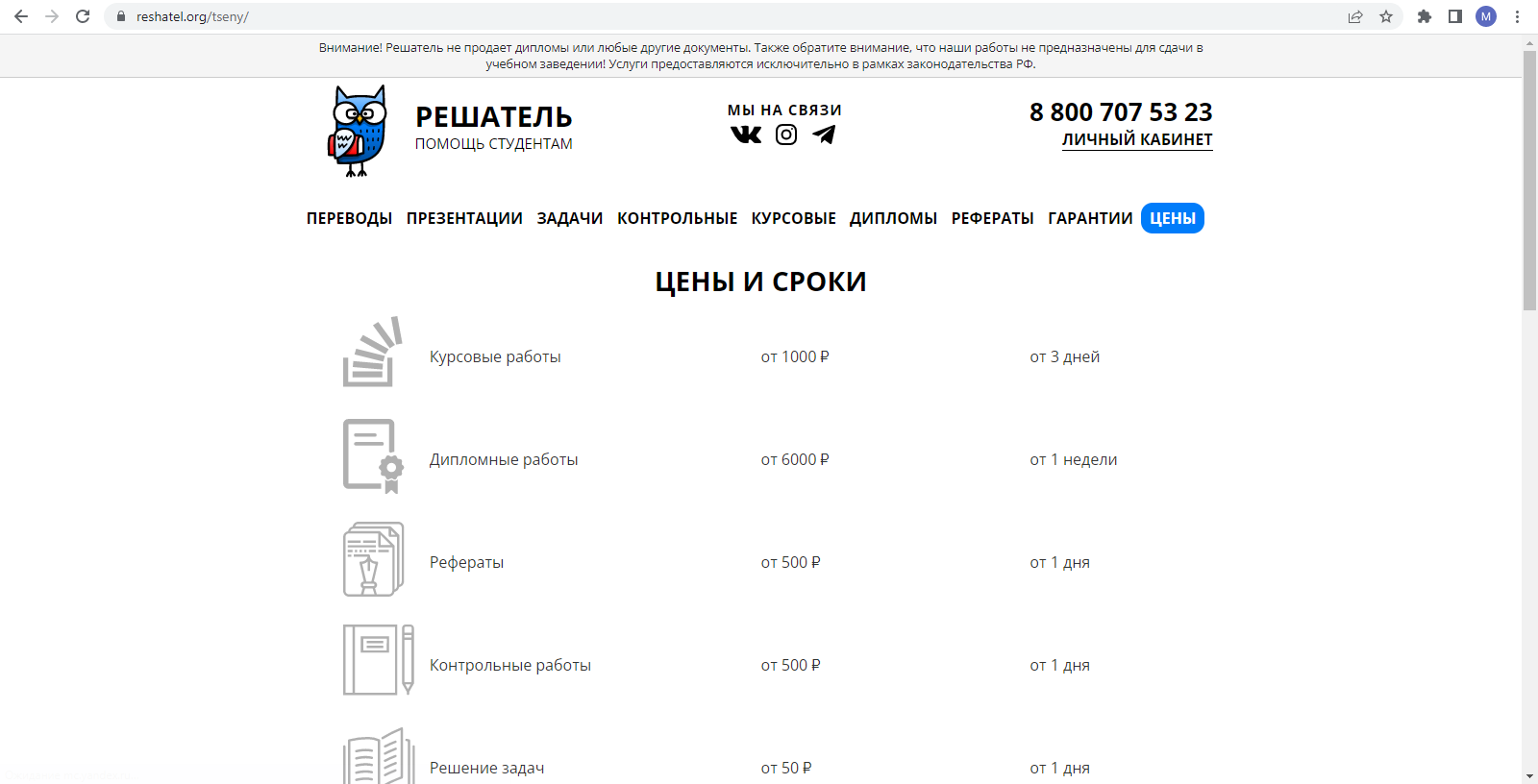
На рисунке 4 представлен интерфейс сайта «Решатель».

Рисунок 4 – сайт «Решатель»

Достоинства:

-Красивый и понятный интерфейс сайта.

-Цены в общем доступе;

-Несколько способов оплаты.

Недостатки:

-Нет технических направлений работ;

-Большая нагрузка и не выполнение работ в срок.

## **3. Реализация.**

## **3.1. Средства реализации.**

В качестве средств реализации приложения были выбраны:

Сервер:

- Windows 10;

- Frameworks – Django 4.2.1;

- Библиотеки: asgiref-3.6.0., Pillow-9.5.0.dist-info,

charset\_normalizer 3.1.0;

- База данных – SQlite3;

- Язык разработки – Python , (html,css);

- PyCharm 2022.3.2 (Proffesional Edition);

- Система контроля версий – git version 2.40.1.windows.1.

Клиент:

- Windows 10,Mac OC;

- PyCharm 2022.3.2 (Proffesional Edition);

- Система контроля версий – git version 2.40.1.windows.1.

Для серверной части была выбрана связка Python - Django так как фреймворк Django имеет большое количество преимуществ, среди них:

- Большое количество доступной документации подробная документация и большое комьюнити. Если у вас появился какой-то вопрос о коде, то вы легко найдёте на него ответ на Stack Overflow или других сайтах;

- DRY (don’t repeat yourself)

- Понятый и простой синтаксис (он написан на Python);

- Большое количество готовых шаблонов и модулей, которые облегчают разработку и позволяют не использовать внешние библиотеки или расширения. Это помогает избегать конфликтов между ними после обновлений;

- Встроенный сервер для развертывания сайта, что существенно облегчает процесс разработки;

- Встроенные ORM, миграции базы данных, аутентификация пользователя, панель администратора, формы

- Django как фреймворк задаёт структуру проекта. Она помогает разработчикам понимать, где и как добавлять новую функциональность.

- REST Framework для создания API

Django REST Framework, который часто сокращают до «DRF», является библиотекой для построения API. Он имеет модульную и настраиваемую архитектуру, которая хорошо работает для создания как простых, так и сложных API.

## **3.2. Реализация серверной части приложения.**

### **3.2.1. Панель администратора.**

На рисунке 4 представлена панель администратора, в ней мы можем вносить правки и изменения на самом сайте.

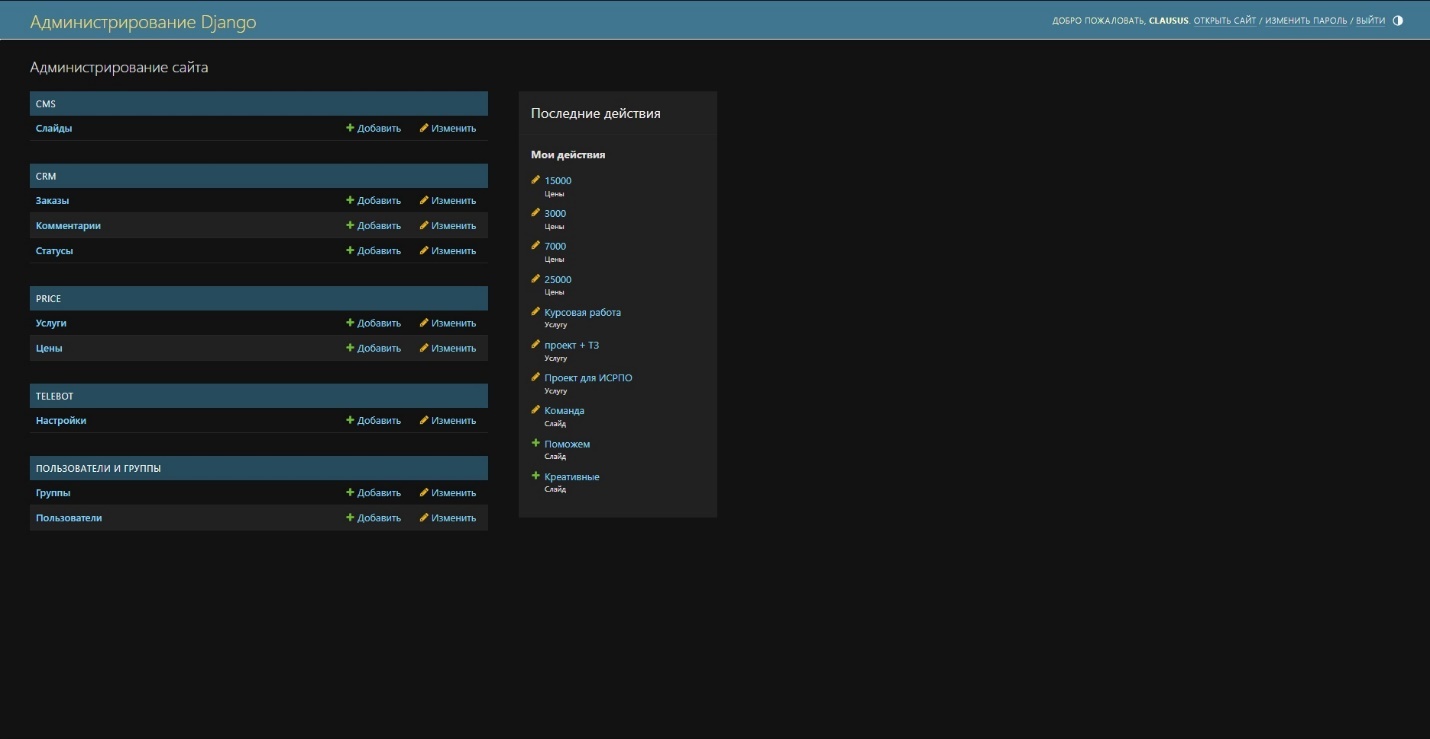


Рисунок 4 – панель администратора.

### **3.2.2. Приложение слайдов.**

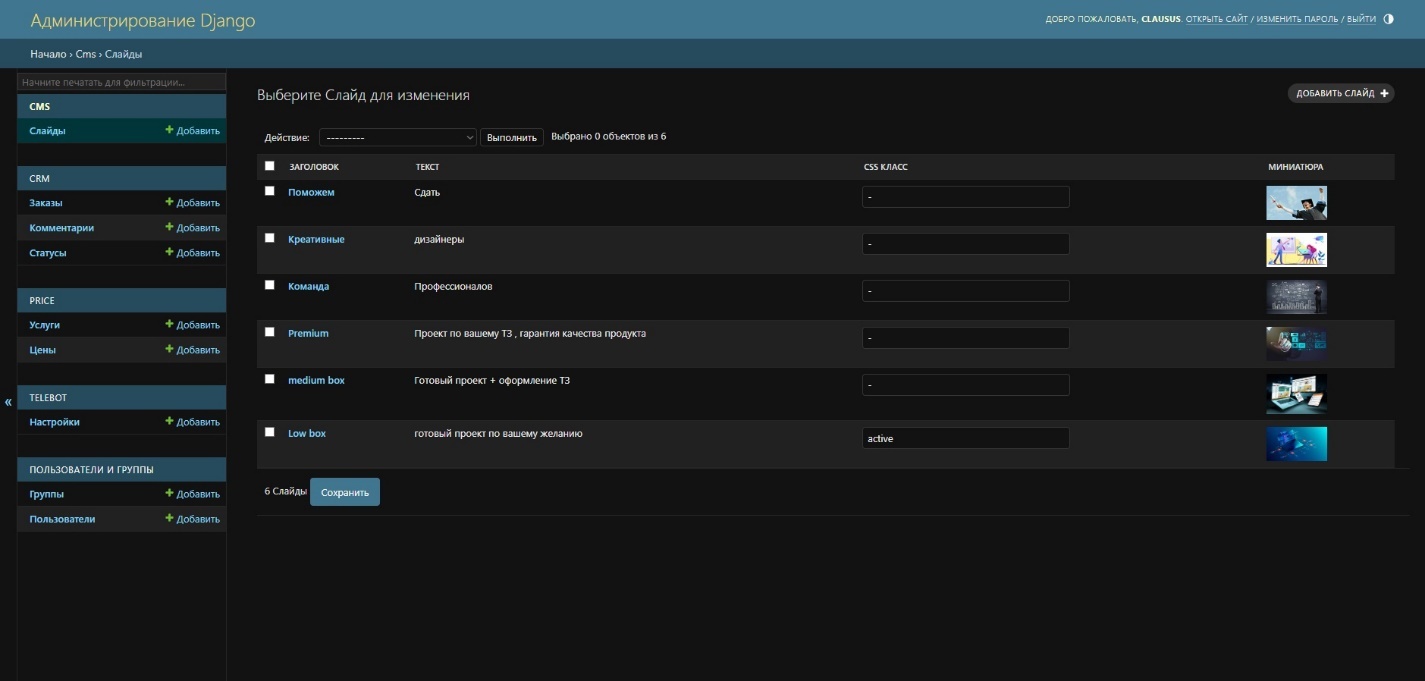


Рисунок 5 – приложение слайдов

На рисунке 5 открыто приложение слайдов, мы можем вносить изменения в данный участок сайта.

### **3.2.3. CRM-система.**

CRM-система — это обработчик форм, заявки поступающие в телеграм-бот, отображаются здесь. Здесь появляется возможность комментировать их и присваивать им статусы стадий выполнения.

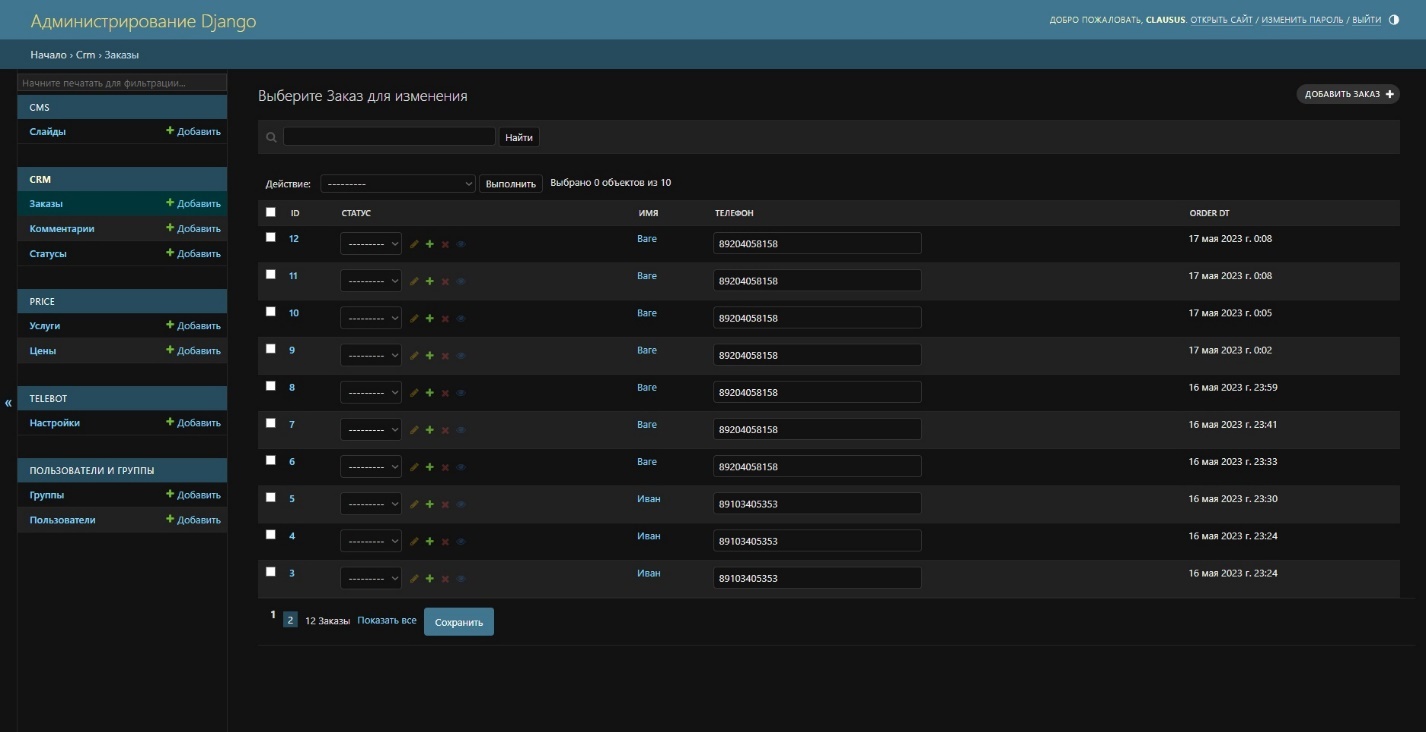


Рисунок 6 - CRM-система.

### **3.2.4. Слой доступа к данным пользователей и групп.**

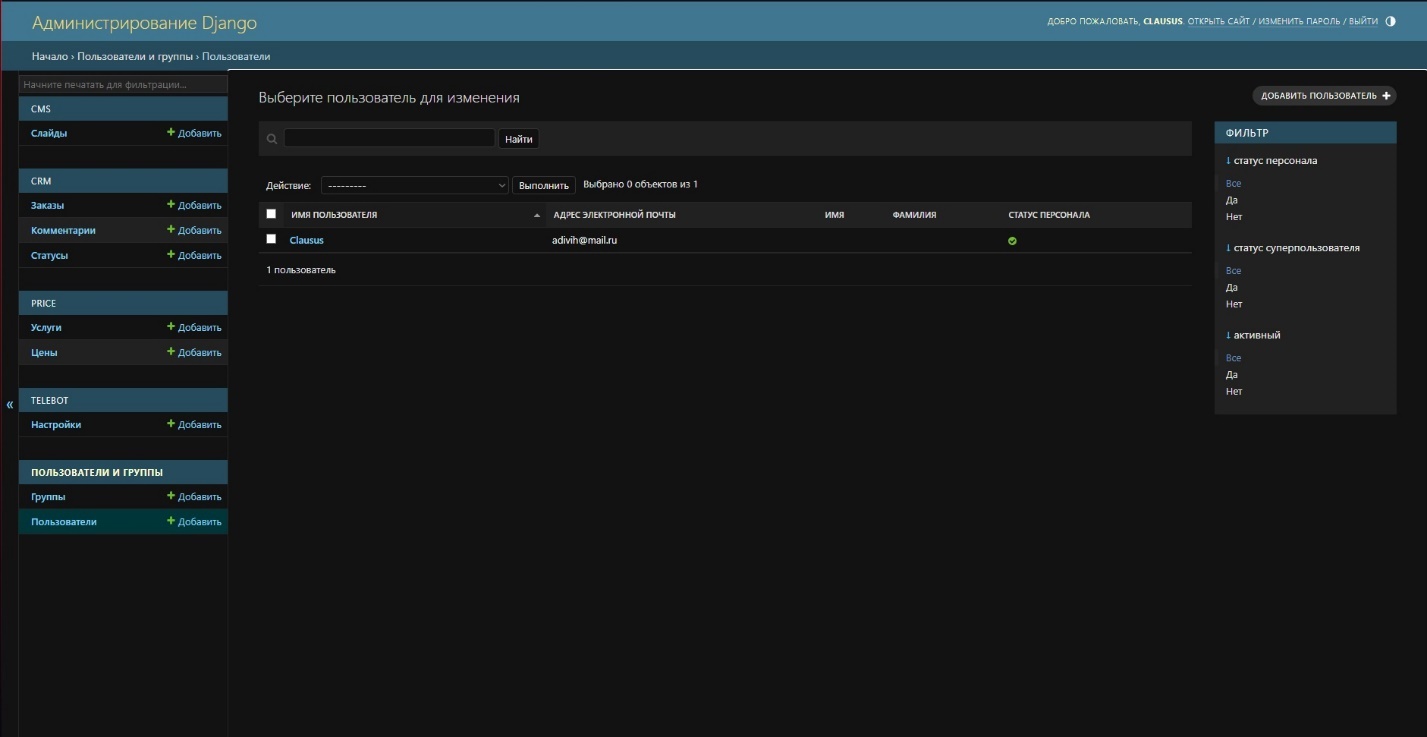


Рисунок 7 – Пользователи и группы.

На рисунке 7 представлен список пользователей и групп и их привилегии. На данном этапе проекта существует один администратор, но в перспективе можно назначать и других модераторов сайтов, выдавая им функционал и авторизовываясь на сайте.

### **3.2.5. Миграции.**

Миграции — это способ Django распространять изменения, которые вы вносите в свои модели (добавление поля, удаление модели и т.д.), в схему вашей базы данных. Они разработаны, чтобы быть в основном автоматическими, но вам нужно знать, когда выполнять миграции, когда их запускать и с какими общими проблемами вы можете столкнуться.

На рисунке 8 представлен пример обработки миграции через CMD-консоль.

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

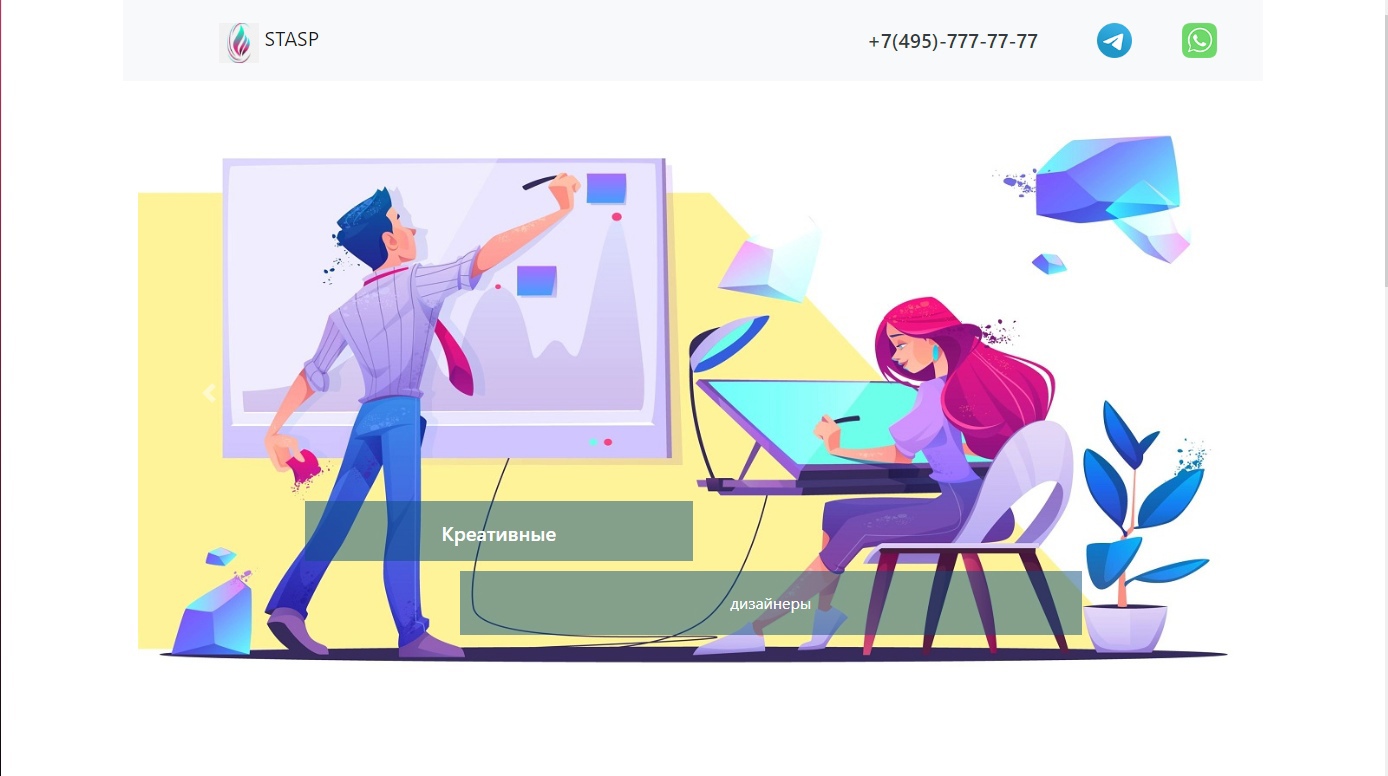
Рисунок 8 – Пример миграции.

## **3.3. Реализация клиентской части приложения.**

### **3.3.1. Макеты интерфейса.**

Сразу после запуска приложения пользователю будет показан экран, на котором ему будут показаны все возможные услуги, предоставляемые нашим сервисом. Номера и доступные мессенджеры для связи.

На Рисунке 9 представлен начальный экран сайта.

Рисунок 9 - Начальный экран сайта.

Пролистав вниз, можно увидеть окно с пакетами услуг (тарифами).



Рисунок 10 – Окно с тарифами.

Рисунок 11 – Описание цен на услуги.



На Рисунке 11 представлена перечень цен и конкретных услуг, самые популярных запросов.

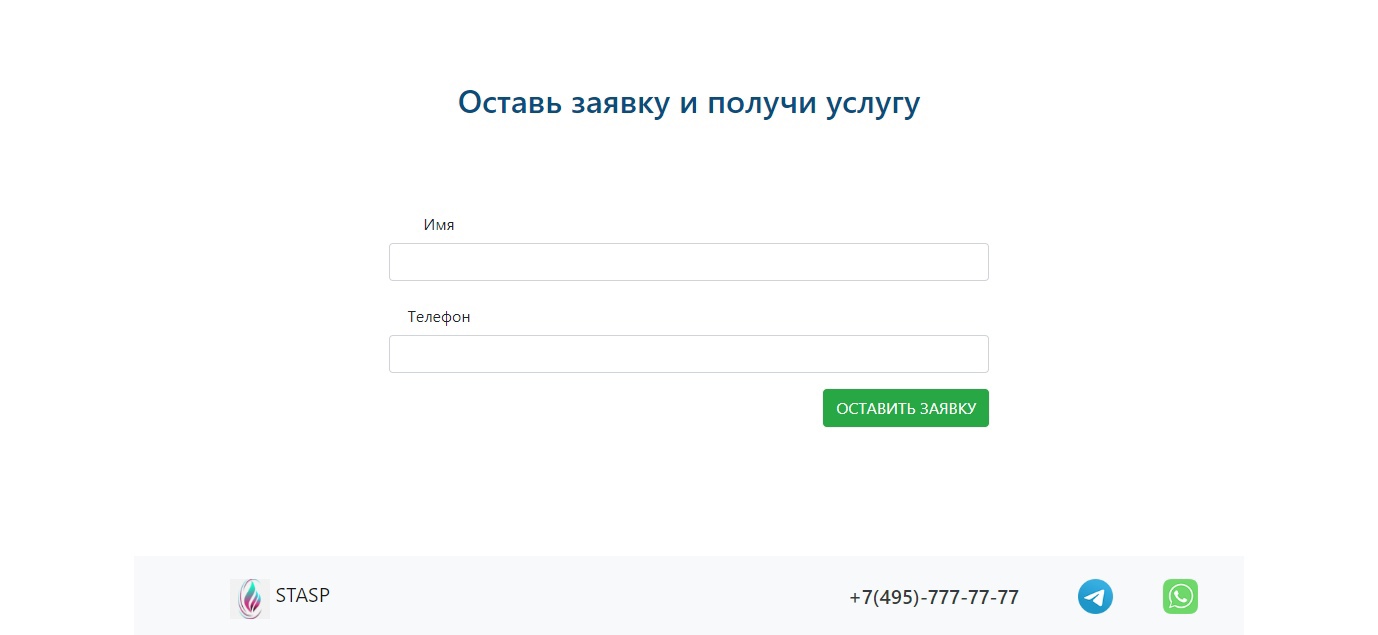
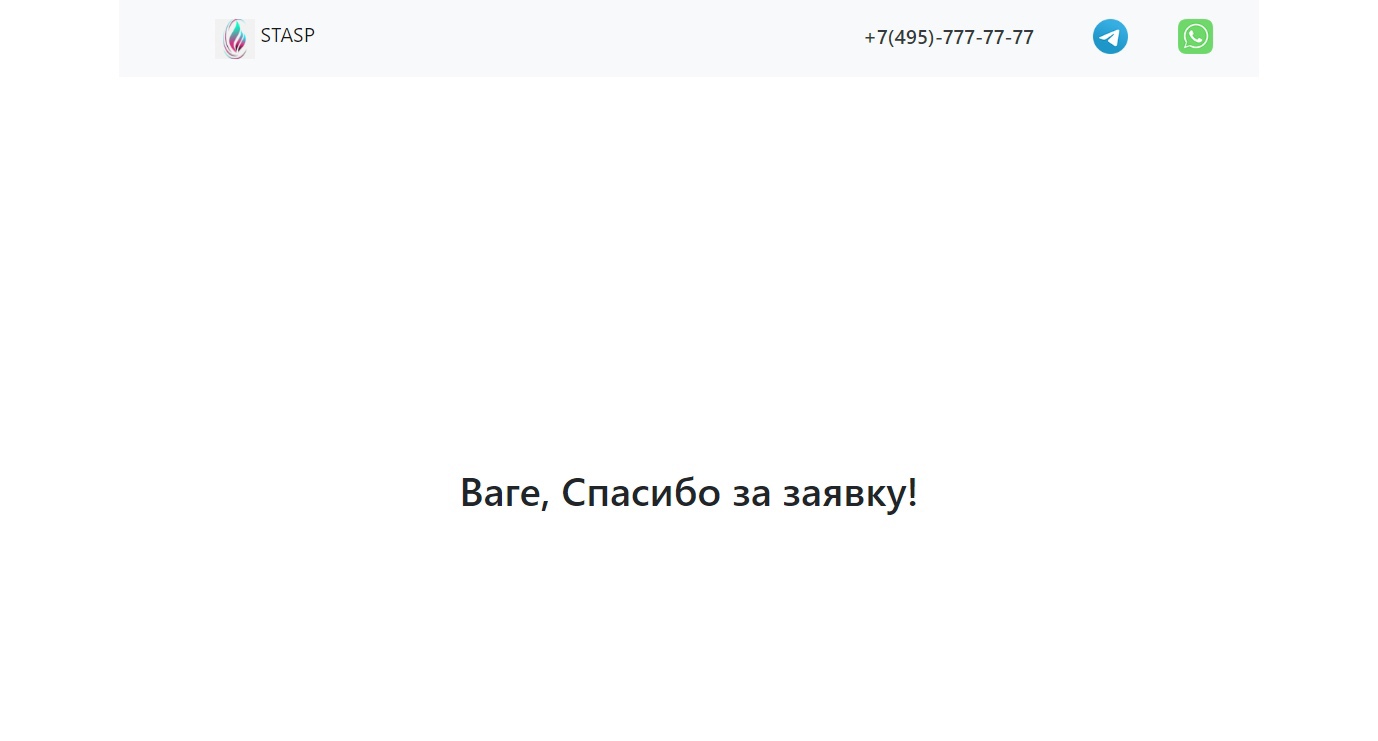


Рисунок 12 - Экран создания заявки.

На Рисунке 12 представлена страница после успешной обработки и приема заявки.

Рисунок 12 – Успешный прием заявки.

На Рисунке 12 представлена страница, видимая после успешной обработки и приема заявки.

### **3.3.2. Макет интерфейса администратора.**

На рисунке 14 отображено сообщение в телеграм-боте, после успешной обработки заявки.



Рисунок 15 – Сообщение от телеграм-бота.

# **Заключение**

В ходе выполнения данного проекта, командой был разработан сервис для продажи и покупки курсовых и дипломных работ. Для прямой связи с платежеспособным и потенциальным клиентом был создан телеграм-бот, в котором, в свою очередь, отображается контактный телефон и имя клиента. Правила продаж тоже были соблюдены, цены обозначены сразу, есть несколько доступных мессенджеров, необходимых для уточнения деталей работы.

Была выполнены следующие задачи:

- Спроектирована система с учётом требований, описанных в техническом задании;

- Разработан Back-end сайта;

- Разработан Front-end сайта;

- Создан телеграм-бот;

- Связана форма оставления заявки и телеграм-бот;

- Описаны возможные услуги;

- Связаны Front-end и Back-end части приложения.

# **Список использованных источников**

1. Oliver Gierke документация Spring Data [Электронный ресурс]. – Режим доступа: https://docs.spring.io/springdata/jpa/docs/current/reference/html/#repositories.query-methods (дата обращения: 11.04.2022).

2. Stéphane Nicoll документация Spring Boot [Электронный ресурс]. – Режим доступа: https://docs.spring.io/springboot docs/current/reference/htmlsingle/#documentation.first-step (дата обращения: 04.05.2022).

3. <https://reshatel.org/tseny/>

4. <https://vsesdal.com/about>

5. Madhura Bhave документация Lombook [Электронный ресурс]. – Режим доступа: https://projectlombok.org/features/ (дата обращения: 05.05.2022).

6. A Brief Timeline of Python.

URL:[http://python–history.blogspot.ru/2009/01/brief–timeline–of–python.html](http://python-history.blogspot.ru/2009/01/brief-timeline-of-python.html) (дата обращения: 16.05.2023).

7. Алгазинов Э. К. Анализ и компьютерное моделирование информационных процессов и систем / Э. К. Алгазинов, А. А. Сирота – М.: Диалог-МИФИ, 2009. – 416 с. 8. Корн Г. Справочник по математике (для научных работников и инженеров) / Г. Корн, Т. Корн. – Наука, 1974 г.