

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ** **РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Уральский государственный экономический университет»**

**(УрГЭУ)**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
(БАКАЛАВРИАТ)**

**Тема: Оптимизация логистики на торговом предприятии**

Кафедра информационных технологий и статистики

Институт менеджмента и информационных технологий

Направление подготовки

09.03.01 Информатика и вычислительная техника

Направленность (профиль)

Автоматизированные системы управления производством

Дата защиты: 20.06.2022

Оценка:

Студент

Сёмкин Сергей Константинович

Группа ИВТ-18-2

Руководитель: канд. экон. наук, доцент Кислицын Евгений Витальевич

Нормоконтролер: Старший преподаватель Панова Марина Валерьевна

Екатеринбург

2022 г.

[ВВЕДЕНИЕ 3](#_Toc105344487)

[1. ОПИСАНИЕ И ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc105344488)

[1.1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ 5](#_Toc105344489)

[1.1.1 ТЕХНИКО-ЭКОНОМИЧЕСТКАЯ ХАРАКТЕРИСТИКА 5](#_Toc105344490)

[2. ПРОЕКТНАЯ ЧАСТЬ 6](#_Toc105344491)

[2.1 ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ 6](#_Toc105344492)

[2.1.1 ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ И СХЕМА ДАННЫХ 6](#_Toc105344493)

[2.1.2 ВХОДНАЯ ИНФОРМАЦИЯ 10](#_Toc105344494)

[2.1.2.1 ФОРМА «ДОБАВИТЬ ОБЪЕКТ» 12](#_Toc105344495)

[2.1.2.2 Форма «Редактирование объекта» 20](#_Toc105344496)

[2.1.3 КЛАССИФИКАТОРЫ И НОРМТИВНО-СПРАВОЧНАЯ ИНФОРМАЦИЯ 21](#_Toc105344497)

[2.1.4 ВЫХОДНАЯ ИНФОРМАЦИЯ 22](#_Toc105344498)

[2.2 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ 24](#_Toc105344499)

[2.3 ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ 24](#_Toc105344500)

[2.3.1 СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ 24](#_Toc105344501)

[2.3.2 СПЕЦИФИКАЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ 27](#_Toc105344502)

[2.3.2.1 Backend 28](#_Toc105344503)

# ВВЕДЕНИЕ

В наше время наличие информационной системы, автоматизирующей некоторые бизнес-процессы, является одним из важных показателей конкурентоспособности на рынке. Информационная система значительно облегчает и уменьшает объём работы, выполняемой непосредственно человеком, исключая человеческие погрешности, которые могут оказаться в итоге фатальными.

В сфере продажи недвижимости, каждый день появляются новые предложения о продаже жилой недвижимости: квартир, комнат, загородных участков, так же появляется спрос на подобные предложения.

Агентства недвижимости (далее АН) представляют из себя посредника между продавцом и покупателем давая гарантию обоим сторонам, что если сделка состоится, то не принесёт никаких проблем. Так же АН предоставляют услуги по поиску, как покупателей, так и продавцов, снимая тем самым хлопоты с клиентов.

Хранение данных об объектах, собственниках, а также дальнейшая обработка данных значительно облегчит работу риелторов, так же руководство сможет отслеживать объём заказов и эффективность сотрудников.

Предметом исследования является процессы создания, обработки, изменения и удаления объектов недвижимости (далее ОН), а также работа с доступами к системе и разделение ролей для различных по должностям сотрудников.

Целью дипломной работы является создание и сопровождение «Системы для работы с объектами недвижимости».

Для достижения цели поставлены следующие задачи:

* + Выполнить сравнительны анализ готовых решений на рынке;
  + Спроектировать будущую информационную систему;
  + Разработать информационную систему;
  + Протестировать получившийся продукт;
  + Произвести оценку системных требований для серверного оборудования и возможные затраты.

# ОПИСАНИЕ И ИССЛЕДОВАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

# 1.1 ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

# 1.1.1 ТЕХНИКО-ЭКОНОМИЧЕСТКАЯ ХАРАКТЕРИСТИКА

Практика проходила в ООО «Ареон», в отделе разработки и внедрения ПО, находящемся по адресу: Свердловская область, г. Екатеринбург, ул. Розы Люксембург 19.

Директор: Сбродов В.И.

Руководитель группы разработки ПО: Никонов А.С.

Сайт: <https://www.areon.pro>

# ПРОЕКТНАЯ ЧАСТЬ

# ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ

# ИНФОЛОГИЧЕСКАЯ МОДЕЛЬ И СХЕМА ДАННЫХ

В данной части выпускной квалификационной работы демонстрируется роль базы данных (далее БД) в информационной системе. Для описания БД используется инфологическая модель, иными словами модель в сущность-связь. Данная модель согласуется с парадигмой объектно-ориентированного программирования, которая в настоящий момент, безусловно, является основой для разработки сложных программных систем.

В начале, необходимо, продемонстрировать весь список имеющихся таблица в БД. Структура и описание таблиц представлена ниже.

Таблица 1 Физическая модель таблицы «Role»

|  |  |
| --- | --- |
| Название колонки | Тип данных |
| Id | int |
| Name | Varchar (15) |

Таблица Role в данной таблице перечислены все возможные роли, которые могут иметь пользователи. Заполняется заранее.

Таблица 2 Физическая модель таблицы «Token»

|  |  |
| --- | --- |
| Название колонки | Тип данных |
| Id | Int |
| Token | Varchar (255) |
| isActive | Boolean |
| sellByUTC | DateTime |

Таблица Token хранятся токены полученные пользователями:

* Первичный ключ;
* Строковое значение токена;
* Состояние токена (активный/неактивный);
* Время жизни токена.

Записи в таблицу добавляются при авторизации пользователя.

Данные из данной таблицы используются при работе с API.

Таблица 3 физическая модель таблицы «User»

|  |  |
| --- | --- |
| Название колонки | Тип данных |
| Id | Int |
| fullName | Varchar (255) |
| Login | Varchar (50) |
| Password | Varchar (20) |
| isActive | Boolean |
| Role\_id | Int |
| Token\_id | Int |

Таблица User хранит данные пользователей:

* Первичный ключ;
* ФИО;
* Логин пользователя;
* Пароль в зашифрованном виде;
* Состояние пользователя (активный/неактивный);
* Внешний ключ к таблице Role;
* Внешний ключ к таблице Token.

Пользователи представляют из себя сотрудников агентства недвижимости и добавляются в данную таблицу через встроенную админ панель Django.

Таблица 4 физическая модель таблицы «Owner»

|  |  |
| --- | --- |
| Наименование колонки | Тип данных |
| Id | Int |
| fullName | Varchar (255) |
| Phone | Varchar (12) |
| Email | Varchar (320) |

Таблица Owner хранит данные собственника:

* Первичный ключ;
* ФИО собственника;
* Номер телефона;
* Адрес электронной почты.

Данные в данную таблицу добавляются при добавлении объекта недвижимости в форме добавления объекта в шаге, связанном с данными собственника.

Данные из этой таблицы используются при отображении списка недвижимости.

Таблица 5 Физическая модель таблицы «Realty»

|  |  |
| --- | --- |
| Наименование колонки | Тип данных |
| Id | Int |
| Owner\_id | Int |
| User\_id | Int |
| typeRealty | Varchar (20) |

Таблица Realty хранит основные данные объекта недвижимости:

* Первичный ключ;
* Внешний ключ к таблице Owner;
* Внешний ключ к таблице User;
* Тип недвижимости.

Данные в таблицу добавляются при создании объекта.

Данные из данной таблицы используются при отображении списка объектов.

Таблица 6 физическая модель таблицы RealtyData

|  |  |
| --- | --- |
| Наименование колонки | Тип данных |
| Id | Int |
| Realty\_id | Int |
| Name | Varchar (25) |
| Value | Varchar (200) |

Таблица RealtyData хранит полное описание объекта недвижимости:

* Первичный ключ;
* Внешний ключ к таблице Realty;
* Наименование свойства (поля);
* Значение свойства (поля).

Данные в таблицу добавляются при создании таблицы, так же данные могут изменяться. При удалении записи в таблице Realty все связные записи в данной таблице так же удалятся.

Данные из этой таблицы используются при отображении списка недвижимости.

Таблица 7 Физическая модель таблицы ImageRealty

|  |  |
| --- | --- |
| Наименование колонки | Тип данных |
| Id | Int |
| Realty\_id | Int |
| Path | Varchar (max) |

Таблица ImageRealty хранит ссылки на фотографии относящиеся к объекту недвижимости

Таки образом, спроектированную БД информационной системы для агентства недвижимости можно отобразить схемой, которая приставлена ниже на рисунке 1.

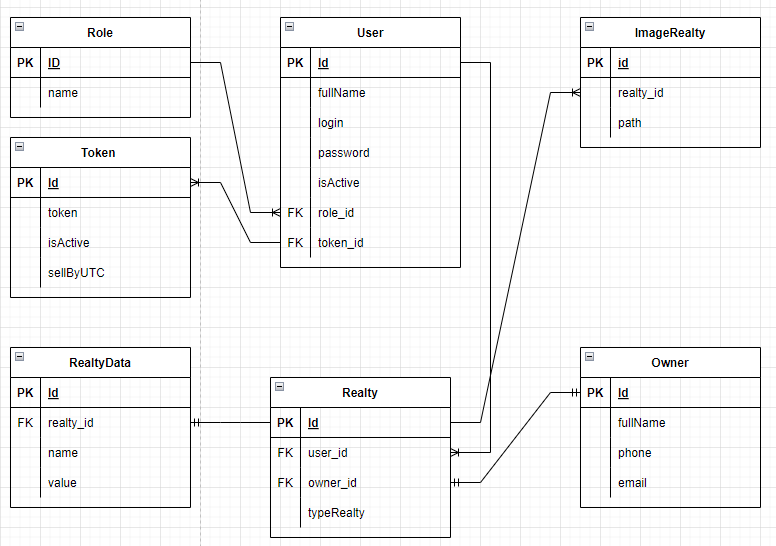


Рисунок 1 схема базы данных

В данной работы была использована СУБД SQLite. SQLite – компактная встраиваемая кроссплатформенная СУБД, которая поддерживает весь необходимый список команд SQL, исходный код находящийся в общественном доступе. Данная СУБД разработана по принципу всё необходимое есть, ничего лишнего нет.

# ВХОДНАЯ ИНФОРМАЦИЯ

Входными данными являются:

1. Форма «Добавить объект», позволяет каждому пользователю, имеющему активную учётную запись и авторизованного на сайте добавлять объекты недвижимости.

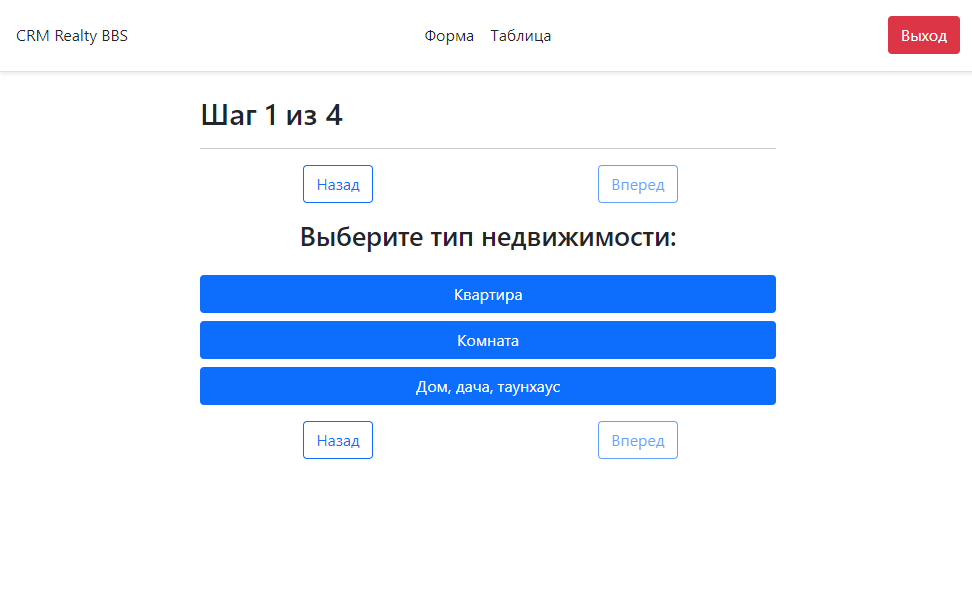


Рисунок 2 Форма «добавить объект». Шаг 1 из 4

1. Форма «Редактирование объекта», позволяет изменить свойства объекта недвижимости, добавленные в форме «добавить объект», а также добавить значения к незаполненным необязательным свойствам объекта недвижимости.
2. Админ панель для работы с таблицей пользователи, через неё можно добавлять, удалять и изменять данные риелторов и собственников.

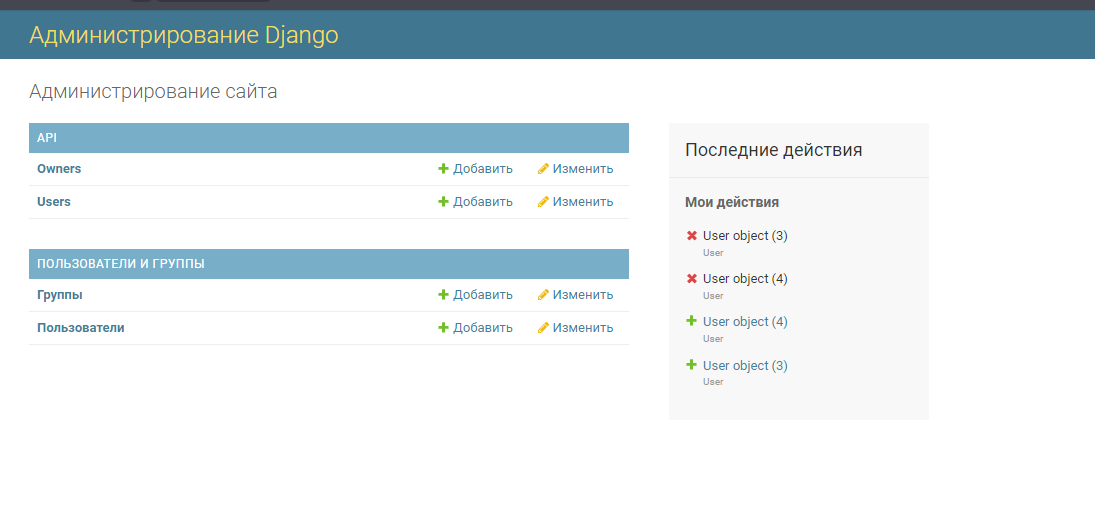


Рисунок 3 Админ панель

# ФОРМА «ДОБАВИТЬ ОБЪЕКТ»

В форма добавления объекта состоит из четырёх смысловых этапов, разделённых кнопка «вперёд» и «назад» (рисунок 2).

На первом этапе пользователю необходимо выбрать тип недвижимости, которую он добавляет. Данное значение в дальнейшем будет использоваться при получении списка объектов, только определённого типа, а также на следующем этапе предлагаемые поля для ввода варьируются в зависимости от выбранного типа.

На втором этапе пользователю выводятся следующие параметры для ввода относящиеся к типу недвижимости из предыдущего этапа. Красной звёздочной справа сверху у названия поля сигнализирует о том, что данное поле является обязательным к заполнению. Так же у опциональных полей пишется «опц.», которая говорит о том, что поле не обязательное.

Для типа недвижимости «квартира» (рисунок 4) предоставляется следующий список параметров:

* Цена объекта;
* Количество комнат;
* Тип квартиры;
* Планировка;
* Высота потолка;
* Площадь общая;
* Площадь жилая;
* Площадь комнат;
* Площадь кухни;
* Этаж;
* Всего этажей;
* Лифт;
* Тип дома;
* Новый дом;
* Год постройки;
* Капитальных ремонт;
* Вид из окон;
* Ремонт;
* Балкон/Лоджия;
* Ванна;
* Санузел;
* Водоснабжение;
* Отопление;
* Мебель;
* Загрузка фото объекта.

Данные поля после заполнения всей формы передаются на сервер через API и обрабатываются.

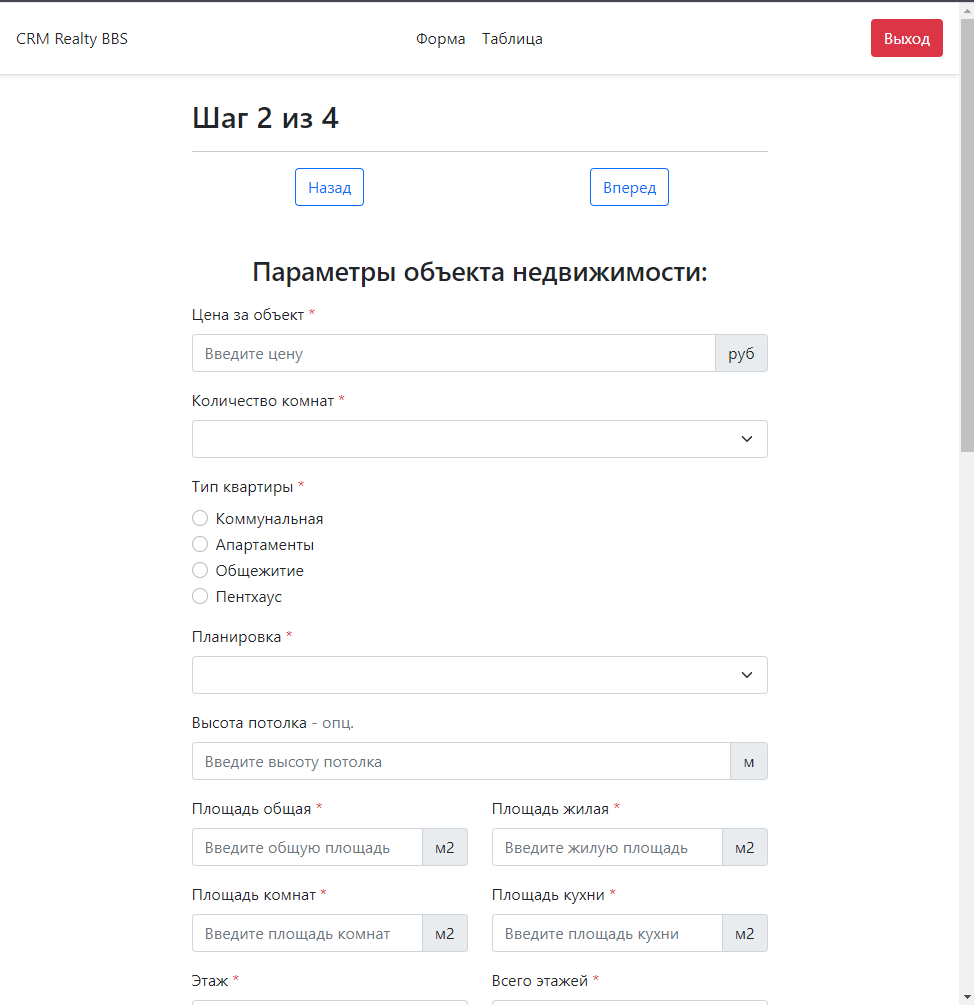


Рисунок 4 Форма «Добавить объект» (часть). Тип недвижимости «Квартира» Шаг 2 из 4

Для типа недвижимости «Комната» (рисунок 5) предоставляется следующий список параметров:

* Цена объекта;
* Количество комнат;
* Тип квартиры;
* Планировка;
* Высота потолка;
* Площадь общая;
* Площадь жилая;
* Площадь комнат;
* Площадь кухни;
* Этаж;
* Всего этажей;
* Лифт;
* Тип дома;
* Новый дом;
* Год постройки;
* Капитальных ремонт;
* Вид из окон;
* Ремонт;
* Балкон/Лоджия;
* Ванна;
* Санузел;
* Водоснабжение;
* Отопление;
* Мебель;
* Загрузка фото объекта.

Данные поля после заполнения всей формы передаются на сервер через API и обрабатываются.

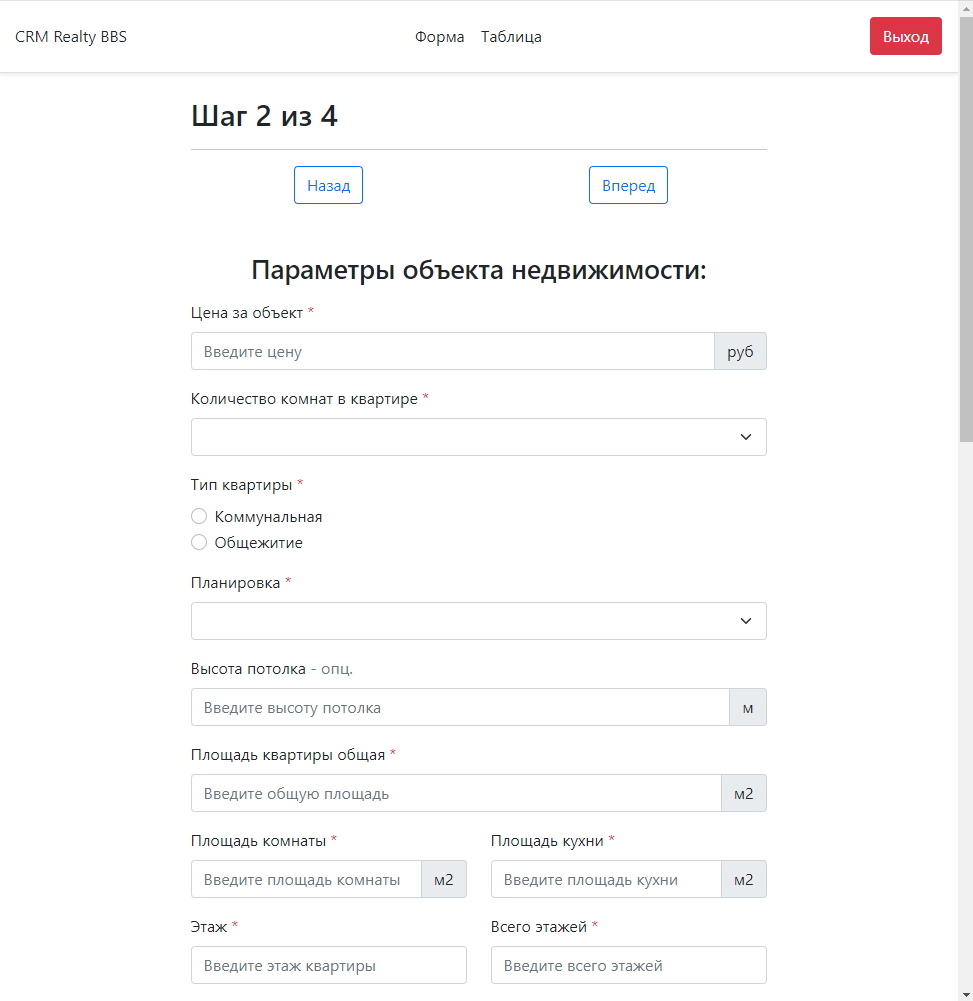


Рисунок 5 Форма «Добавить объект» (часть). Тип недвижимости «Комната» Шаг 2 из 4

Для типа недвижимости «Дом, дача, таунхаус» (рисунок 6) предоставляется следующий список параметров:

* Тип дома;
* Цена за объект;
* Площадь общая;
* Площадь жилая;
* Площадь комнат;
* Площадь кухни;
* Площадь участка;
* Высота потолка;
* Всего этажей;
* Тип дома;
* Новый дом;
* Год постройки;
* Ремонт;
* Юридический статус земли;
* Отопление;
* Канализация;
* Вода;
* Электричество;
* Газ;
* Сауна/Баня;
* Мебель;
* Загрузка фото объекта.

Данные поля после заполнения всей формы передаются на сервер через API и обрабатываются.

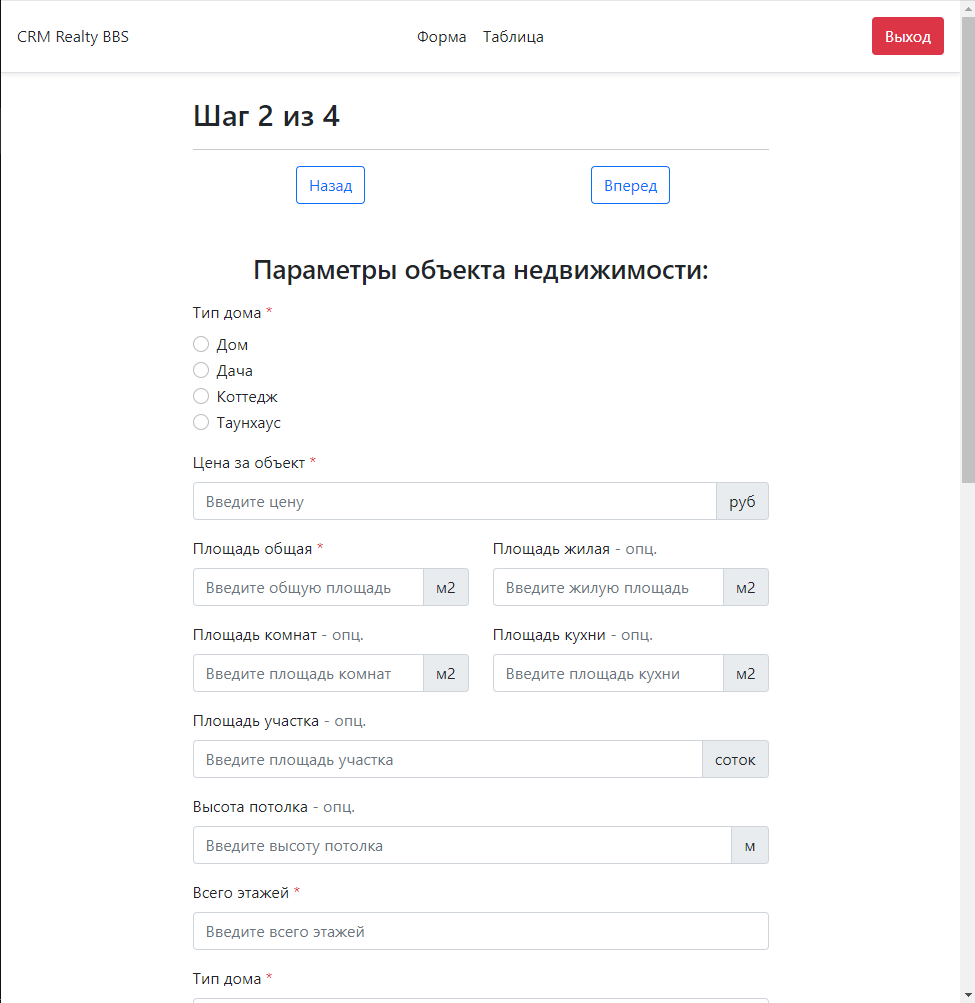


Рисунок 6 Форма «Добавить объект» (часть). Тип недвижимости «Дом, дача, таунхаус» Шаг 2 из 4

На третьем этапе (рисунок 7) пользователю необходимо заполнить данные собственника:

* ФИО собственника;
* Телефон собственника;
* Электронная почта собственника.

Так же, как и на предыдущем этапе данные будут отправлены на сервер через API. Обязательные поля помечены звёздочкой, а необязательные надписью «опц.». В отличии от предыдущих данных, данные собственника записываются в отдельную таблицу.

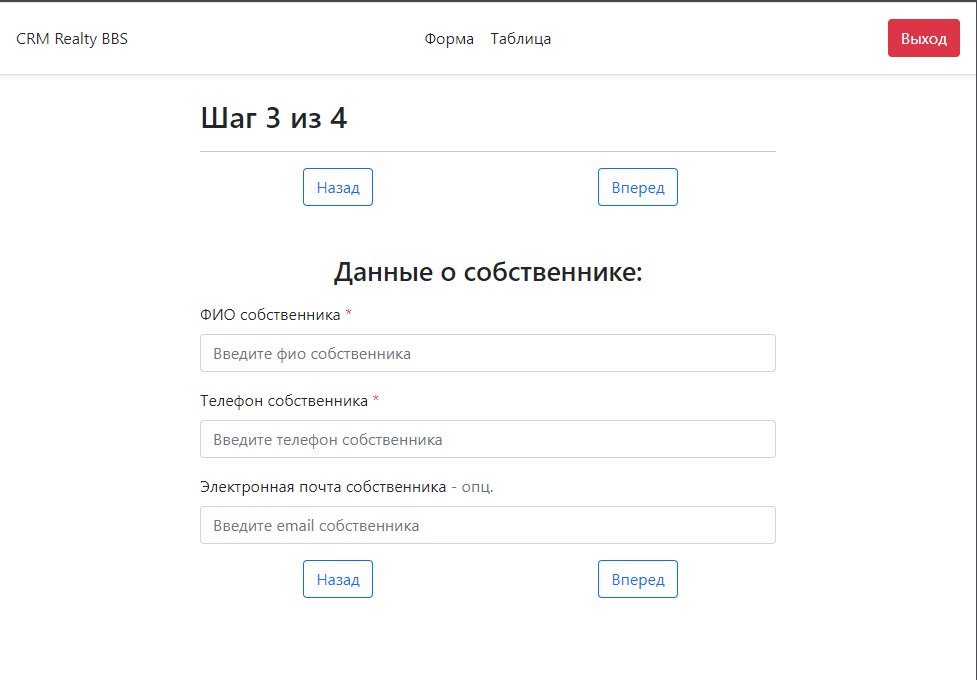


Рисунок 7 Форма «Добавить объект». Шаг 3 из 4

На четверном этапе, представленном на рисунке ниже (рисунок 8) пользователю необходимо заполнить данные, относящиеся к расположению объекта недвижимости:

* Расположение;
* Населённый пункт;
* Район;
* Улица;
* Дом;
* Корпус;
* Квартира;
* Метро;
* До метро;
* Величина.

Так же, как и на предыдущем этапе данные будут отправлены на сервер через API. Обязательные поля помечены звёздочкой, а необязательные надписью «опц.».

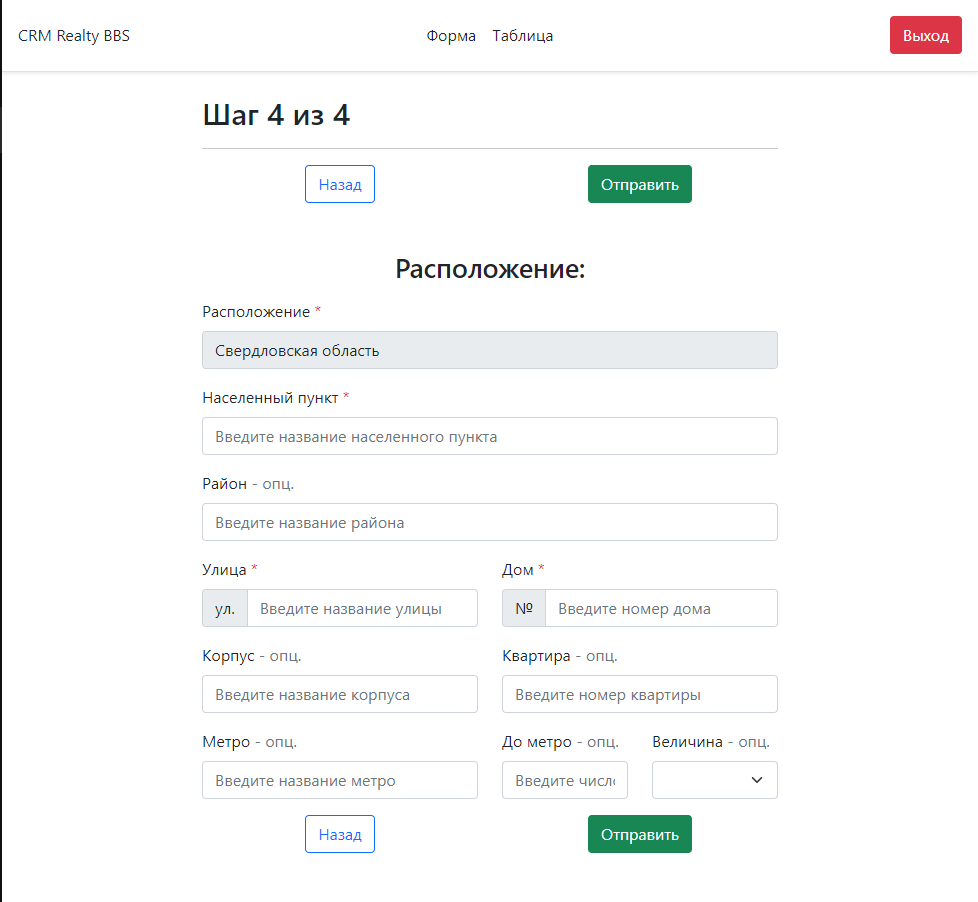


Рисунок 8 Форма «Добавить объект». Шаг 4 из 4

# Форма «Редактирование объекта»

Данная форма предназначена для редактирования уже существующего объекта. С помощью неё помимо того, что можно изменить данные, так же добавить данные для необязательных полей, которые не были запенены при создании объекта недвижимости. Переход к данной форме находится на личной странице объекта (рисунок 9). Для перехода необходимо нажать на кнопку «Редактировать».



Рисунок 9 Личная страница объекта

ЗДЕСЬ БУДЕТ ПОИСАНИЕ ФОРМЫ РЕДАКТИРОВАНИЯ С ФОТО

# КЛАССИФИКАТОРЫ И НОРМТИВНО-СПРАВОЧНАЯ ИНФОРМАЦИЯ

В БД присутствует два справочника:

* Пользователи;
* Роли.

Справочник «Пользователи» содержит информацию о пользователях использующий информационную систему (таблица 8).

Таблица 8 Состав таблицы «Пользователи»

|  |  |
| --- | --- |
| Название колонки | Тип данных |
| Id | Int |
| fullName | Varchar (255) |
| Login | Varchar (50) |
| Password | Varchar (20) |
| isActive | Boolean |
| Role | Int |
| Token | Int |

Справочник «Роли» содержит список ролей, которые выдаются пользователям (таблица 9).

Таблица 9 Состав таблицы «Роли»

|  |  |
| --- | --- |
| Название колонки | Тип данных |
| Id | Int |
| Name | Varchar (25) |

В данном разделе была представлена нормативно-справочная информация.

# ВЫХОДНАЯ ИНФОРМАЦИЯ

Пользователь получает данные, которые хранятся в информационной системе к нескольким местах.

При переходе на вкладку «Таблица» пользователь видит список объектов недвижимости выбранного типа (рисунок 10).

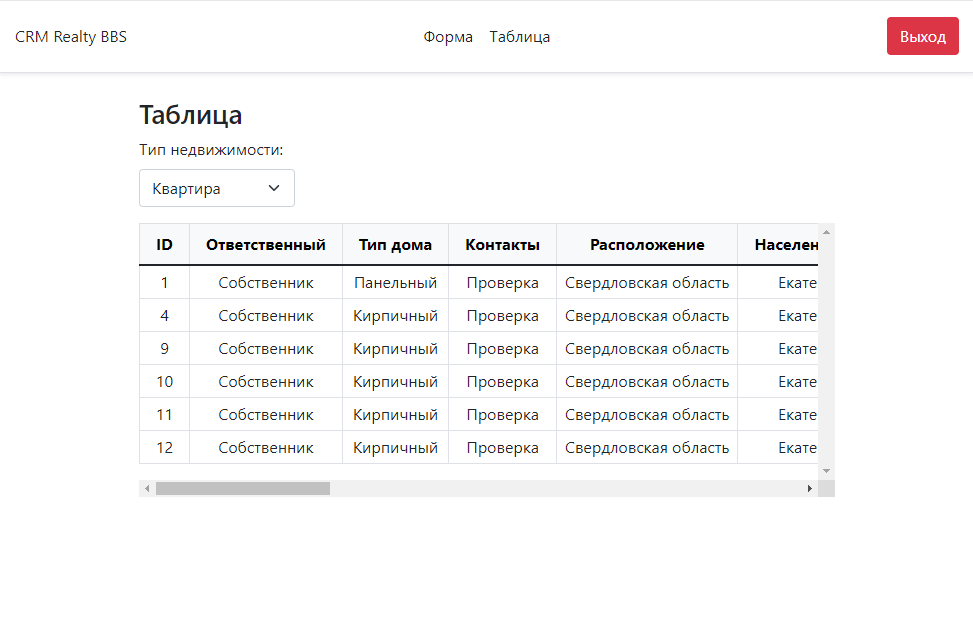


Рисунок 10 Вкладка Таблица

Таблица является адаптивной и выводит только те столбцы, которые относятся к данному типу недвижимости.

Так же можно посмотреть более наглядное описание каждого конкретно взятого объекта недвижимости нажав на строку в таблице объектов. При нажатии на появляется личный кабинет объекта (рисунок 9). Данные разделены на три логические блока:

* Контактные данные;
* Расположение;
* Параметры.

Выходные данные сервер отправляет при редактировании объекта недвижимости. Перейдя на форму «Редактирования объекта» для пользователя в поля, соответствующие свойству объекта, поругаются имеющиеся данные.

Рисунок

# МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

# ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

# СТРУКТУРА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Главная страница сайта содержит таблицу бля вывода имеющихся объектом, но с одним нюансом. Привилегированные сотрудники могут видеть абсолютно все объекты недвижимости, но для обычных сотрудников выводится только информация об их объектах недвижимости.

Смежная станицей является

Структура сайта информационной системы представлена в соответствии с рисунком 11.

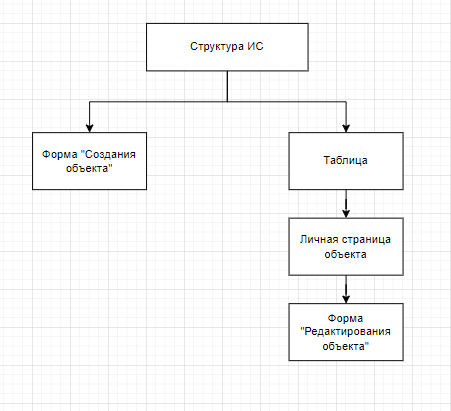


Рисунок 11 Структура сайта

Для доступа к данным на сайте необходимо авторизоваться, используя личный логин и пароль. Структура станицы авторизации рисунок 12.



Рисунок 12 Структура страницы авторизации

Для создания объекта недвижимости в информационной системе существует отдельная страница «Добавить объект». Для перехода на страницу необходимо выбрать её в навигационной панели.

Структура страницы добавления объекта представлена на рисунке 13.

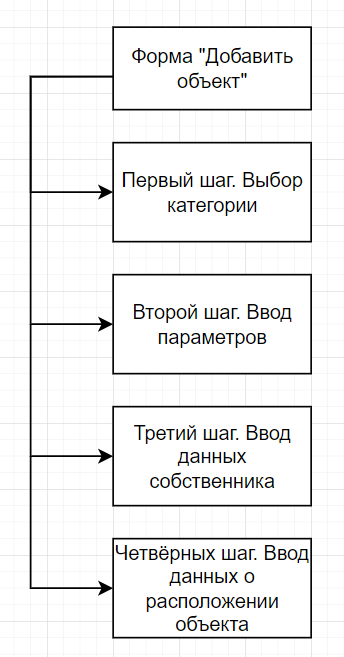


Рисунок 13 Структура формы «Добавить объект»

Для редактирования объекта недвижимости в информационной системе существует отдельная страница «Редактировать проект». Для перехода на страницу необходимо выбрать объект недвижимости в таблице, а потом нажать на кнопку редактировать расположенной в вверху у названия.

Структура страницы «Редактировать объект» представлена на рисунке 14.

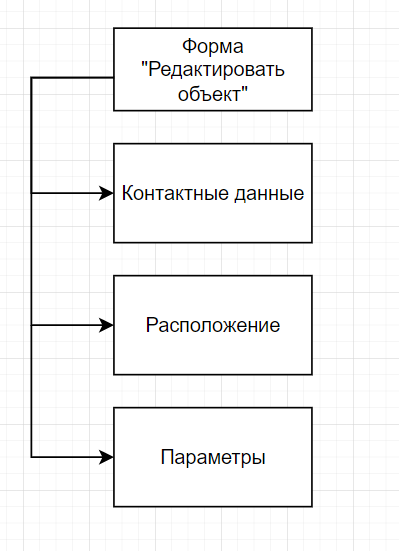


Рисунок 14 Структура страницы «Редактировать объект»

В данном разделе были рассмотрены структуры основных страниц.

# СПЕЦИФИКАЦИЯ ПРОГРАММНЫХ МОДУЛЕЙ

Данный проект состоит из двух смысловых частей:

* Backend. Отвечает за функционал, выполняемый на стороне сервера;
* Frontend. Отвечает за функционал, видимый пользователю, иными словами, за клиентскую сторону проекта.

# Backend

Реализацию серверной части информационной системы была произведена с помощью фреймворка Django написанной на языке программирования python.

Серверный функционал можно разделить на несколько смысловых частей:

* Роутинг входящих запросов;
* Запросы к базе данных;
* Отправка ответов на запросы.

За роутинг запросов в Django происходит за счёт файлов urls.py. Данные файлы располагаются в папке проекта и в папках приложений. Внутри описывается массив, состоящий из функций *path* принимающих в качестве аргумента:

* Запрещённый адрес URL;
* Функцию, которая обрабатывает запрос по этому адресу.

Так же к этому массиву в главном файле urls.py отдельно добавляются обработчики запросов медиа файлов (рисунок 15).

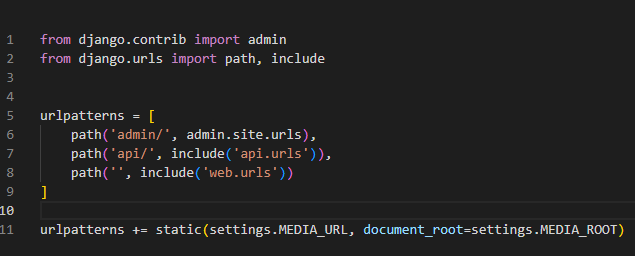


Рисунок 15 Листинг главного файла urls.py

Запросы к базе данных реализованы в Django через всторенную ORM.

ORM – технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную БД». Для запроса к базе данных необходимо указать название описанной модели, по которой Django создаёт в используемой БД таблице и воспользовавшись необходимой функцией сделать запрос.

Для выборки данных в информационной системе чаще всего использовалась функция *filter*. Она позволяет добавить условие выборки указав столбец и условие выборки. Данная функция возвращает в качестве результата *QuerySet,* который с помощью функции *values* модно отфильтровать, взявтолько определённые столбцы, а также преобразовать в массив словарей.

Также для выборки одной записи используется функция *get,* в качестве параметров может принимать поля таблицы и условия выборки. В отличии от *filter* функция возвращает объект, с полями аналогичными полям модели.

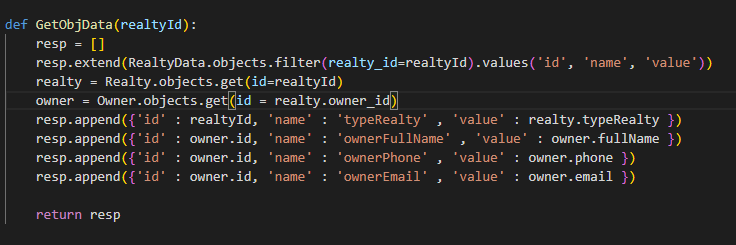


Рисунок 16 Функция для получения данных объекта недвижимости

На рисунке 16 демонстрируются использование вышеописанных функций *filter* и *get*.

Для ответа на поступивший запрос необходимо использовать встроенную функцию *Response*. В качестве параметров функция принимает:

* *status* (по умолчанию 200);
* *data* (возвращаемые данные);
* *template\_name* (имя шаблона);
* *headers* (заголовки http ответа);
* *content\_type* (явное определение типа).

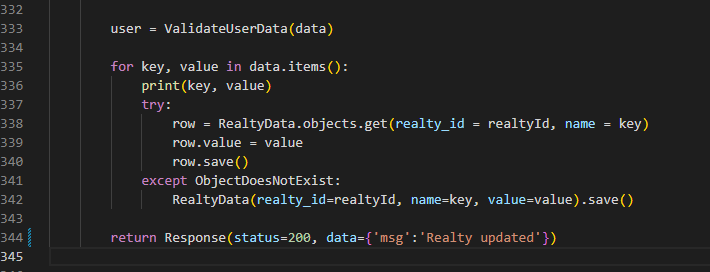


Рисунок 17 Обновление данных невидимости (часть).

# Frontend