

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные СУБД»
Тема: Каталог жилого фонда СПб

Студент гр. 9304	_____	Ковалёв П.Д.
Студент гр. 9304	_____	Кузнецов Р.В.
Студент гр. 9304	_____	Прокофьев М.Д.
Преподаватель	_____	Заславский М.М.

Санкт-Петербург
2022

ЗАДАНИЕ НА ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ

Студент Ковалёв П.Д.

Студент Кузнецов Р.В.

Студент Прокофьев М.Д.

Группа 9304

Тема работы: Каталог жилого фонда СПб

Содержание пояснительной записки:

- Аннотация
- Содержание
- Введение
- Качественные требования к решению
- Сценарии использования
- Модель данных
- Разработанное приложение
- Заключение
- Список использованных источников

Предполагаемый объем пояснительной записки:

Не менее 40 страниц.

Дата выдачи задания: 01.09.2022

Дата сдачи реферата: 15.12.2022

Дата защиты реферата: 22.12.2022

Студент	_____	Ковалёв П.Д.
Студент	_____	Кузнецов Р.В.
Студент	_____	Прокофьев М.Д.
Преподаватель	_____	Заславский М.М.

АННОТАЦИЯ

В данной работе разработано веб-приложение, представляющее каталог жилого фонда СПб, позволяющая пользователям просматривать список объектов жилого фонда, фильтровать их по множеству параметров, просматривать статистику жилого фонда, а также искать объекты на карте. В системе используется следующий стек технологий: язык JavaScript, библиотека React для клиентской части; nodejs в качестве серверной части; Docker Compose для сборки проекта.

SUMMARY

In this work, a web application has been developed that represents the catalog of the housing stock of St. Petersburg, allowing users to view a list of housing stock objects, filter them by a variety of parameters, view housing stock statistics, and also search for objects on the map. The system uses the following technology stack: JavaScript language, React library for the client side; nodejs as a back end Docker Compose to build the project.

СОДЕРЖАНИЕ

Введение	5
1. Качественные требования к решению	6
1.1. Текущие требования	6
2. Сценарии использования	7
2.1. Макет пользовательского интерфейса	7
2.2. Сценарии использования	7
2.3. Преобладающие операции	12
3. Модель данных	13
3.1. Нереляционная модель данных	13
3.2. Реляционный аналог модели данных	19
3.3. Сравнение моделей	25
4. Разработанное приложение	26
4.1. Описание приложения	26
4.2. Используемые технологии	26
Заключение	27
Список использованных источников	28
Приложение А. Макет пользовательского интерфейса	29
Приложение Б. Документация по сборке и развёртыванию приложения	36

ВВЕДЕНИЕ

При работе с объектами жилого фонда, работникам помимо адреса данного фонда, необходимо знать его местоположение на карте, а также подробную информацию о его состоянии. Создание единого каталога жилого фонда позволит собрать в одном месте всю возможную информацию о каждом объекте жилого фонда, что позволит работникам выполнять свою работу быстрее и качественнее.

Цель работы – разработать ИС, представляющую из себя каталог жилого фонда.

Основные задачи:

1. Сформулировать основные сценарии использования приложения.
2. Разработать макет пользовательского интерфейса.
3. Разработать схему базы данных.
4. Подготовить прототип приложения.

1. КАЧЕСТВЕННЫЕ ТРЕБОВАНИЯ К РЕШЕНИЮ

1.1. Текущие требования

Текущие требования к решению выглядят следующим образом:

1. Имеется страница авторизации. Используется только администратором для того, чтобы была возможность импортировать в систему набор данных с объектами жилого фонда.
2. Есть возможность как импорта данных в систему, так и экспорта из системы в .csv формате.
3. Модули предоставляют следующие возможности:
 - a. Cathalog – модуль, позволяющий просматривать набор данных. Представляет из себя таблицу, по которой можно производить поиск и фильтрацию.
 - b. Map – модуль, позволяющий производить поиск объектов на карте, а также возможность просмотра информации о каждом объекте на отдельной странице.
 - c. Statistics – модуль, позволяющий просмотреть статистику по каталогу.
4. Данные лежат по адресу https://classif.gov.spb.ru/irsi/7840013199-Tehniko-ekonomicheskie-pasporta-mnogokvartirnyh-domov/structure_version/207/.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. Макет пользовательского интерфейса

Разработан макет пользовательского интерфейса, представленный в приложении А.

2.2. Сценарии использования

Сценарий использования «Авторизация»:

- Действующее лицо: Администратор
- Пререквизиты: Администратор открыл браузер.
- Основной сценарий:
 1. Администратор переходит на страницу /auth
 2. Администратор попадает на страницу “Авторизация”
 3. В случае ввода корректных данных, Администратор успешно авторизуется, после чего попадает на страницу “Каталог жилищного фонда”
- Альтернативный сценарий:
 1. Администратор ввёл некорректный ключ в соответствующее поле
 2. Веб-сайт сообщает об ошибке, поле для ввода ключа загорается красным цветом

Сценарий использования «Выход из аккаунта»:

- Действующее лицо: Администратор
- Пререквизиты: Администратор зашёл на веб-сайт и авторизовался.
- Основной сценарий:
 1. Администратор нажимает на кнопку “Выход” в левом нижнем углу страницы
 2. Администратор попадает на страницу с авторизацией

Сценарий использования «Просмотр каталога жилого фонда»:

- Действующее лицо: Пользователь.
- Пререквизиты: Пользователь зашёл на веб-сайт.
- Основной сценарий:
 1. Пользователь при заходе на сайт попадает в раздел сайдбара “Каталог” - на страницу “Каталог жилищного фонда”
 2. Пользователь имеет возможность просмотреть таблицу, представляющую собой каталог жилищного фонда
 3. Пользователь нажимает на кнопку “Фильтровать” в результате чего пользователь видит модальное окно
 4. В модальном окне пользователь задает значения некоторым параметрам, по которым будут фильтроваться данные. Поддерживается возможность ввода нескольких значений для строковых полей и диапазонов значений для числовых полей.
 5. Пользователь может нажать на значок вопроса в правом верхнем углу модального окна, в результате чего всплывает подсказка по работе с фильтрами.
 6. Пользователь может сортировать таблицу, нажав на кнопку “Сортировать:” и выбрав в появившемся списке столбец, по которому будет произведена сортировка.
 7. Пользователь может произвести поиск данных по ключевому слову, используя форму поиска (на макете представлен как форма-input Search), написав туда слово, в результате чего в таблицу будут данные, у которых значение хотя бы одного столбца совпадает с заданным словом

Сценарий использования «Просмотр страницы объекта каталога»:

- Действующее лицо: Пользователь.
- Пререквизиты: Пользователь зашёл на веб-сайт.

– Основной сценарий:

1. Пользователь при заходе на сайт попадает в раздел сайдбара “Каталог” - на страницу “Каталог жилищного фонда”.
2. Пользователь нажал на раздел сайдбара “Карта”
3. Пользователь попадает на страницу “Карта жилищного фонда”
4. При нажатии на иконку глаза в строке с адресом жилого дома пользователь перенаправляется на страницу, содержащую подробную информацию по данному объекту и его местоположение на карте.

Сценарий использования «Просмотр геолокации»:

– Действующее лицо: Пользователь.

– Пререквизиты: Пользователь зашёл на веб-сайт.

– Основной сценарий:

1. Пользователь при заходе на сайт попадает в раздел сайдбара “Каталог” - на страницу “Каталог жилищного фонда”.
2. Пользователь нажал на раздел сайдбара “Карта”
3. Пользователь попадает на страницу “Карта жилищного фонда”
4. Слева от карты находится список всех жилых домов из каталога
5. Пользователь нажимает на интересующий его жилой дом из списка и на карте появляется метка, показывающая местонахождение выбранного дома
6. Пользователь может нажать на кнопку фильтр, в результате чего появляется модальное окно, в котором пользователь задает значения некоторым параметрам, по которым будут фильтроваться данные. Поддерживается возможность ввода нескольких значений для строковых полей и диапазонов значений для числовых полей.

7. Пользователь может нажать на значок вопроса в правом верхнем углу модального окна, в результате чего всплывает подсказка по работе с фильтрами.

Сценарий использования «Просмотр статистики»:

- Действующее лицо: Пользователь со стандартными правами.
- Пререквизиты: Пользователь зашел на веб-сайт.
- Основной сценарий:
 1. Пользователь при заходе на сайт попадает в раздел сайдбара “Каталог” - на страницу “Каталог жилищного фонда”.
 2. Пользователь нажал на раздел сайдбара “Статистика”
 3. Пользователь попадает на страницу “Статистика жилищного фонда”
 4. Пользователь может наблюдать несколько графиков со статистикой по набору данных жилищного фонда.
 5. Пользователь может нажать на кнопку “Фильтры”, в результате чего появится всплывающее окно с фильтрами.
 6. Пользователь может нажать на значок вопроса в правом верхнем углу модального окна, в результате чего всплывает подсказка по работе с фильтрами.
 7. Пользователь может задать значения полям фильтра, в результате чего для построения графиков со статистикой будут отобраны данные, соответствующие значения которых удовлетворяют выбранным пользователем значениям.

Сценарий использования «Импорт НД»:

- Действующее лицо: Администратор.
- Пререквизиты: Администратор зашёл на веб-сайт и успешно авторизовался.
- Основной сценарий:

1. После успешной авторизации Администратор попадает в раздел сайдбара “Каталог” - на страницу “Каталог жилищного фонда”
2. Администратор нажимает на кнопку “Импорт НД”, которая находится в сайдбаре
3. В появившемся Администратор выбирает файл, который будет импортирован

Сценарий использования «Экспорт НД»:

- Действующее лицо: Пользователь.
- Пререквизиты: Пользователь зашёл на веб-сайт.
- Основной сценарий:
 1. Пользователь при заходе на сайт попадает в раздел сайдбара “Каталог” - на страницу “Каталог жилищного фонда”.
 2. Пользователь нажимает на кнопку “Экспорт НД”, которая находится в сайдбаре
 3. Текущий набор данных скачивается пользователю на ПК в виде файла

Сценарий использования «Просмотр больничных дней»:

- Действующее лицо: Пользователь.
- Пререквизиты: Пользователь зашёл на веб-сайт и успешно авторизовался.
- Основной сценарий:
 1. Пользователь при заходе на сайт попадает в раздел сайдбара “Каталог” - на страницу “Каталог жилищного фонда”.
 2. Пользователь нажимает на ссылку, где находятся имя и фамилия, в правом верхнем углу любой страницы веб-сайта
 3. Пользователь попадает на страницу с профилем Администратора, где указаны его имя и роль (права доступа)

2.3. Преобладающие операции

В модуле Cathalog фигурируют только операции на чтение, т.к. в данном модуле данные можно просмотреть и отфильтровать.

В модуле Map фигурируют только операции на чтение.

В модуле Statistics фигурируют только операции на чтение, т.к. в данном модуле данные для построения графиков можно отфильтровать.

3. МОДЕЛЬ ДАННЫХ

3.1. Нереляционная модель данных

Разработана схема нереляционной базы данных (рисунок 1).

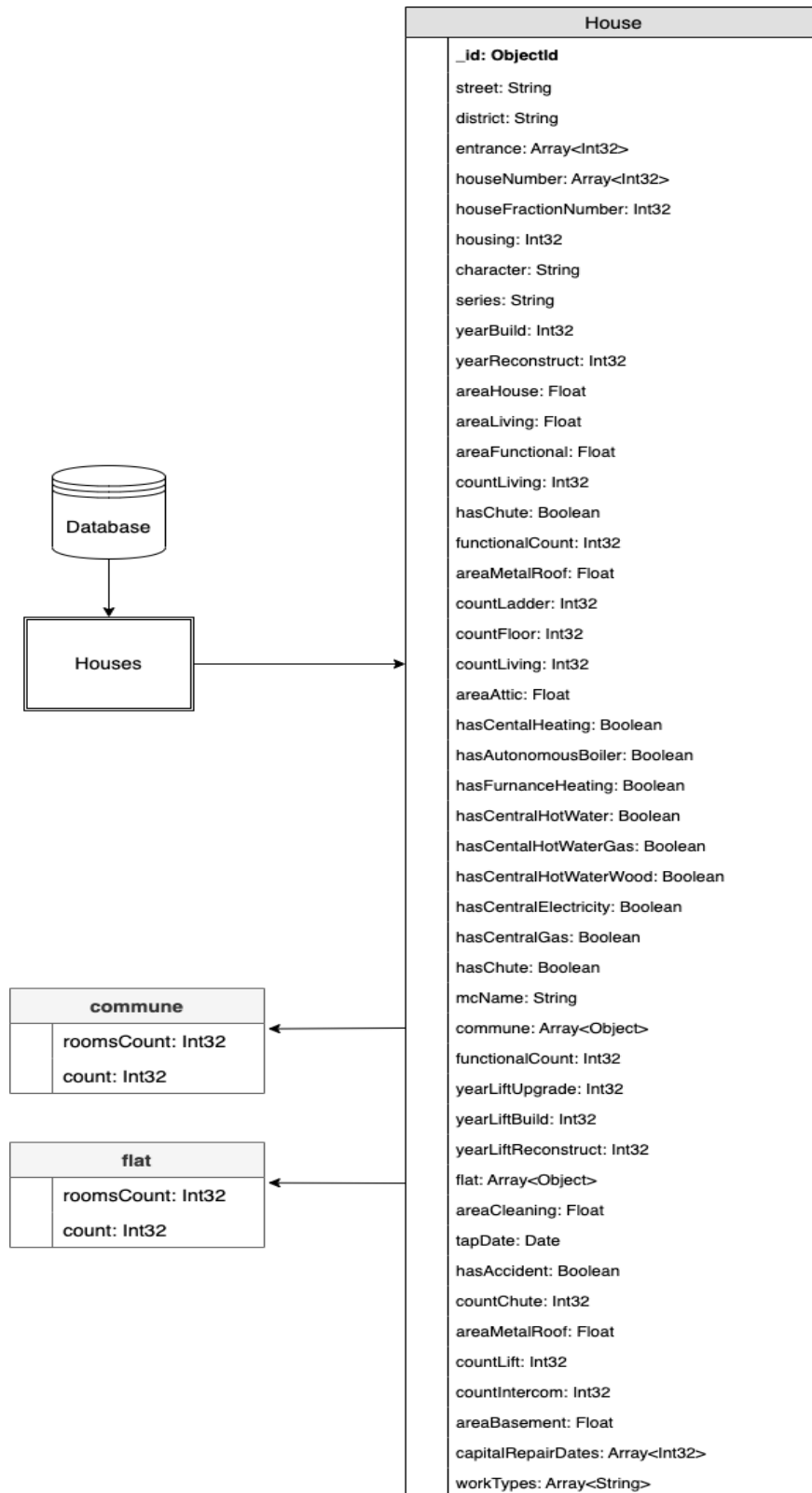


Рисунок 1 – Графическое представление нереляционной базы данных

БД содержит одну коллекцию – Houses. Поля данной коллекции следующие:

- `_id`: ObjectId – Номер (12 байт)
- `street`: UTF-8 String – Улица (128 байт, т.к. используются русские символы)
- `houseNumber`: Array[Int32]– Номер дома (в среднем 12 байт, т.к. в среднем 1 номер)
- `houseFractionNumber`: Int32 – Дробный номер дома (4 байта)
- `housing`: Int32 – Корпус (4 байта)
- `character`: UTF-8 String – Литера (64 байта)
- `district`: UTF-8 String – Район (64 байта)
- `commune`: Array[Object] – Коммунальные (в среднем 75 байт)
 - `roomsCount` – Количество комнат
 - `count` – Количество квартир с данным количеством комнат
- `series`: UTF-8 String – Серия, тип проекта (32 байта)
- `yearBuild`: Int32 – год постройки (4 байта)
- `yearReconstruct`: Int32 – год реконструкции (4 байта)
- `areaHouse`: Float – Общая площадь здания, м2 (4 байта)
- `areaLiving`: Float – Площадь жилых помещений, м2 (4 байта)
- `areaFunctional`: Float – Площадь нежилых помещений функционального назначения, м2 (4 байта)
- `countLadder`: Int32 – Число лестниц (4 байта)
- `countFloor`: Int32 – Число этажей (4 байта)
- `countLiving`: Int32 – Количество проживающих (4 байта)
- `areaAttic`: Float – Площадь мансард, м2 (4 байта)
- `hasCentralHeating`: Boolean – Центральное отопление (1 байт)
- `hasAutonomousBoiler`: Boolean – Автономная котельная (1 байт)
- `hasFurnaceHeating`: Boolean – Печное отопление (1 байт)

- hasCentralHotWater: Boolean – Центральное горячее водоснабжение (1 байт)
- hasCentralHotWaterGas: Boolean – Горячее водоснабжение от газовых колонок (1 байт)
- hasCentralHotWaterWood: Boolean – Горячее водоснабжение от дровяных колонок (1 байт)
- hasCentralElectricity: Boolean – Центральное электроснабжение (1 байт)
- hasCentralGas: Boolean – Центральное газоснабжение (1 байт)
- hasChute: Boolean – Мусоропроводы (1 байт)
- flat: Array[Object] – квартиры (75 байт в среднем, среднее количество 2)
 - roomsCount – количество комнат
 - count – Количество квартир с данным количеством комнат
- functionalCount: Int32 – Количество встроенных нежилых помещений (4 байта)
- yearLiftBuild: Int32 – Год ввода лифтов в эксплуатацию (4 байта)
- yearLiftReconstruct: Int32 – Год реконструкции лифтов (4 байта)
- yearLiftUpgrade: Int32 – Год модернизации лифтов (4 байта)
- areaCleaning: Float – общая площадь уборки придомовых территорий, м2 (4 байта)
- tapDate: Date – Дата составления ТЭП (8 байт)
- mcName: UTF-8 String – Полное наименование управляющей компании (128 байт)
- hasAccident: Boolean – Аварийность (1 байт)
- capitalRepairDates: Array[Int32] – Год проведения капитального ремонта (26 байт в среднем)
- workTypes: Array[UTF-8 String[64]] – Виды работ (549 байт в среднем, среднее количество 3)

- countChute: Int32 – Количество стволов мусоропровода (4 байта)
- areaMetalRoof: Float – Площадь металлической кровли (4 байта)
- countLift: Int32 – Общее количество лифтов (4 байта)
- countIntercom: Int32 – Количество ПЗУ (4 байта)
- areaBasement: Float – Площадь подвалов, м2 (4 байта)

Оценим объем базы данных. Средний объем документа с домом: $V = 1349$ bytes. Количество многоквартирных домов в Санкт-Петербурге: $N = 23123$. Объем базы данных - $N * V = 29,7$ MB.

Избыточность выражается в дубликатах mcName, street, workTypes -

$$\frac{23123 * 1349 \text{ bytes}}{(23123 * 1349 \text{ bytes} - ((23123 - 3751) * 128 \text{ bytes} + (23123 - 1617) * 128 \text{ bytes} + (23124 * 3 - 938) * 128 \text{ bytes}))}$$

1.8

Общая формула избыточности (при 83.8% дубликатов mcName, 93% дубликатов street, 98.65% дубликатов типов работ) -

$$\frac{N * 1349}{(N * 1349 - N * 0.838 * 128 + N * 0.93 * 128 + 3 * N * 0.9865 * 128)} = 1.8$$

Так как всего существуют 3751 управляющая компания, 1617 улиц и 938 уникальных типов работ.

Запросы к БД представлены на рисунках 2-6.


```

db.houses.insert_many([
    {
        "_id": 1,
        "street": "Дунайский пр.",
        "houseNumber": 36,
        "housing": 1,
        "character": "А",
        "district": "Фрунзенский",
        "commune": Null,
        "series": "600.11",
        "yearBuild": 2001,
        "yearReconstruct": null,
        "areaHouse": 10425.4,
        "areaLiving": 9017.7,
        "areaFunctional": null,
        "countLadder": 4,
        "countFloor": 10,
        "countLiving": 288,
        "areaAttic": null,
        "hasCentralHeating": true,
        "hasAutonomousBoiler": false,
        "hasFurnanceHeating": false,
        "hasCentralHotWater": true,
        "hasCentralHotWaterGas": false,
        "hasCentralHotWaterWood": false,
        "hasCentralElectricity": true,
        "hasCentralGas": false,
        "hasChute": true,
        "flat": [{roomsCount: 1, count: 40}, {roomsCount: 2, count: 42}, {roomsCount: 3, count: 78}],
        "functionalCount": null,
        "yearLiftBuild": 2001,
        "yearLiftReconstruct": null,
        "yearLiftUpgrade": null,
        "areaCleaning": 3558,
        "tapDate": datetime(2012, 12, 27),
        "mcName": "ЮжДомСтрой",
        "hasAccident": false,
        "capitalRepairDates": null,
        "workTypes": workTypes,
        "countChute": 4,
        "areaMetalRoof": areaMetalRoof,
        "countLift": 4,
        "countIntercom": 4,
        "areaBasement": 1206.5
    }
])

```

Рисунок 2 – Запрос на добавление узла

```

db.houses.update_one(
    {"_id": 1},
    {
        "$set": {"hasAccident": True}
    }
)

```

Рисунок 3 – Запрос на обновление данных об доме (добавление информации о наличии аварийности)

```
db.houses.update_one({
  '_id': 1,
  {
    "$set": {"yearReconstruct": 2022}
  }
})
```

Рисунок 4 – Запрос на обновление данных о доме по его id (добавление информации о реконструкции)

```
db.houses.find({
  'yearBuild': {'$gt': 1998}
})
```

Рисунок 5 - Запрос для поиска домов, построенных после 1990 года.

```
db.houses.count_documents({
  '_id': {'$ne': 1},
  'yearBuild': 1998,
  'hasAccident': True
})
```

Рисунок 6 - Запрос для подсчета кол-ва аварийных домов с id не равным определенному, имеющих определенный год строения.

3.2. Реляционный аналог модели данных

Разработана схема реляционной базы данных (рисунок 2).

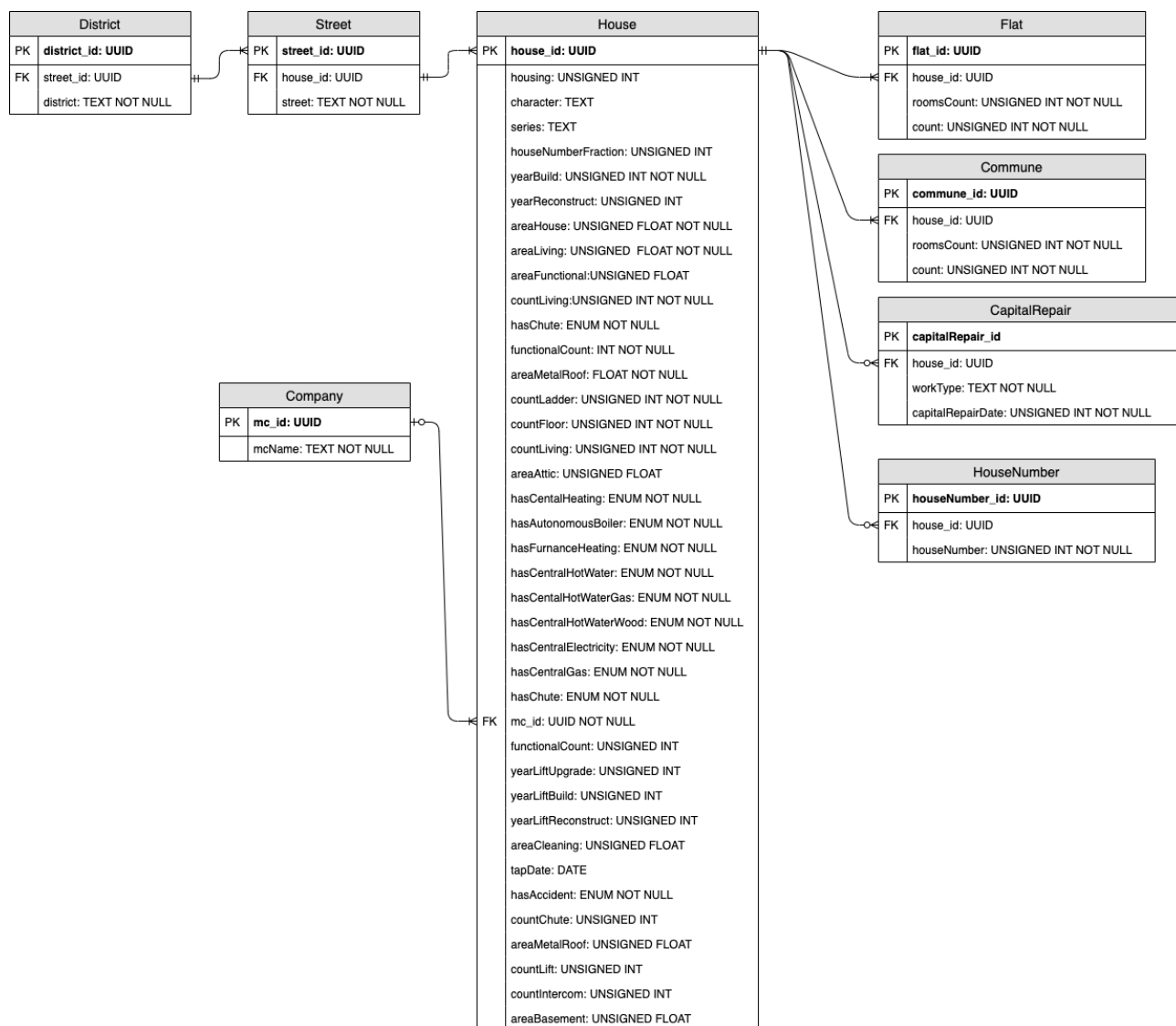


Рисунок 7 – Графическое представление реляционной базы данных

Ниже приведены описания таблиц.

Таблица “House”:

- `house_id: UUID` – ID дома (16 байт)
- `housing: INT` – Корпус (4 байта)
- `character: TEXT` – Литера (64 байта)
- `series: TEXT` – Серия, тип проекта (32 байта)
- `houseFractionNumber: INT` – Дробный номер дома (4 байта)
- `yearBuild: UNSIGNED INT` – Год постройки (4 байта)

- yearReconstruct: INT - Год проведения реконструкции (4 байта)
- areaHouse: FLOAT – Общая площадь здания, м2 (4 байта)
- areaLiving: FLOAT – Площадь жилых помещений, м2 (4 байта)
- areaFunctional: FLOAT - Площадь нежилых помещений функционального назначения, м2 (4 байта)
- countLiving: INT – Количество проживающих (4 байта)
- hasChute: ENUM – Мусоропроводы (1 байт)
- functionalCount: INT – Количество встроенных нежилых помещений (4 байта)
- areaMetalRoof: Float – (4 байта)
- countLadder: INT – Число лестниц (4 байта)
- countFloor: INT – Число этажей (4 байта)
- areaAttic: FLOAT - Площадь мансард, м2 (4 байта)
- hasCentralHeating: ENUM – Центральное отопление (1 байт)
- hasAutonomousBoiler: ENUM – Автономная котельная (1 байт)
- hasFurnaceHeating: ENUM – Печное отопление (1 байт)
- hasCentralHotWater: ENUM – Центральное горячее водоснабжение (1 байт)
- hasCentralHotWaterGas: ENUM – Горячее водоснабжение от газовых колонок (1 байт)
- hasCentralElectricity: ENUM – Центральное электроснабжение (1 байт)
- hasCentralGas: ENUM – Центральное газоснабжение (1 байт)
- hasChute: ENUM – Мусоропроводы (1 байт)
- mc_id: UUID – ID управляющей компании
- functionalCount: INT – Количество встроенных нежилых помещений, (4 байта)
- yearLiftUpgrade: INT – Год модернизации лифтов (4 байта)
- yearLiftBuild: INT – Год ввода лифтов в эксплуатацию (4 байта)

- yearLiftReconstruct: INT – Год реконструкции лифтов (4 байта)
- areaCleaning: FLOAT – Общая площадь уборки придомовых территорий, м2 – (4 байта)
- tapDate: DATE – Дата составления ТЭП (8 байт)
- hasAccident: ENUM – Аварийность (1 байт)
- countChute: INT – Количество стволов мусоропровода (4 байта)
- areaMetalRoof: FLOAT – Площадь металлической кровли (4 байта)
- countLift: INT – Общее количество лифтов (4 байта)
- countIntercom: INT – Количество ПЗУ (4 байта)
- areaBasement: FLOAT – площадь подвалов, м2 (4 байта)

Таблица “Company”:

- mc_id: UUID – ID управляющей компании (16 байт)
- mcName: TEXT – Название управляющей компании (128 байт)

Таблица “District”:

- district_id: UUID – ID района (16 байт)
- street_id: UUID – ID улицы, находящейся в этом районе (16 байт)
- district: TEXT – Название района (64 байта)

Таблица “Street”:

- street_id: UUID – ID улицы (16 байт)
- house_id: UUID – ID дома (16 байт)
- street: TEXT – Название улицы (128 байт)

Таблица “Flat”:

- flat_id: UUID – ID квартиры, (16 байт)
- house_id: UUID – ID дома (16 байт)
- roomsCount: INT – Количество комнат (4 байта)
- count: INT – Количество квартир с данным количеством комнат (4 байта)

Таблица “Commune”

- commune_id: UUDI – ID коммунальной квартиры (16 байт)

- house_id: UUID – ID дома (16 байт)
- roomsCount: INT – Количество комнат (4 байта)
- count: INT – Количество квартир с данным количеством комнат (4 байта)

Таблица “CapitalRepair”:

- capitalRepair_id: UUID – ID капитального ремонта (16 байт)
- house_id: UUID – ID дома (16 байт)
- workType: TEXT – Виды работ (128 байт)
- capitalRepairDate: INT – Год проведения капитального ремонта (4 байта)

Таблица “HouseNumber”:

- houseNumber_id: UUID – ID номера дома (16 байт)
- house_id: UUID – ID дома (16 байт)
- houseNumber: INT – Номер дома (4 байта)

Средний объём таблиц:

Таблица District:

- Количество объектов: 19
- Вес объекта: 96 bytes
- Вес таблицы: 1536 bytes

Таблица Street:

- Количество объектов: 1618
- Вес объекта: 160 bytes
- Вес таблицы: 258880 bytes

Таблица Company:

- Количество объектов: 3752
- Вес объекта: 144 bytes
- Вес таблицы: 540288 bytes

Таблица House:

- Количество объектов: 23123
- Вес объекта: 358 bytes
- Вес таблицы: 8278034 bytes

Таблица Flat:

- Количество объектов: $23123 * 2 = 46246$
- Вес объекта: 40 bytes
- Вес таблицы: 1849840 bytes

Таблица Commune:

- Количество объектов: $23123 * 2 = 46246$
- Вес объекта: 40 bytes
- Вес таблицы: 1849840 bytes

Таблица CapitalRepair:

- Количество объектов: $23123 * 3 = 69372$
- Вес объекта: 164 bytes
- Вес таблицы: 11377008 bytes

Таблица HouseNumber:

- Количество объектов: $23123 * 1 = 23123$
- Вес объекта: 36 bytes
- Вес таблицы: 832428 bytes

Итого: 24447566 bytes (23,3 MB)

Избыточность выражается дублированием проведенных работ:

$$\frac{24447566}{(24447566 - (69372 - 938) * 128 \text{ bytes})} = 1.5$$

Количество многоквартирных домов:

$$N = 23123$$

Средний вес записи о многоквартирном доме:

$$V = 1057 \text{ байт}$$

Вес базы данных:

$$N * V = 23.3 \text{ MB}$$

Общая формула избыточности (при дублировании проведенных работ 98.64%):

$$\frac{N * 1057}{N * 1057 - 3 * N * 0.9864 * 128} = 1.5$$

Запросы к БД представлены на рисунках 8-13.

```
INSERT into Company values (1, 'ЮжДомСтрой');
INSERT into District values (1, 'Фрунзенский');
INSERT into Street values (1, 1, 'Дунайский пр. ');
INSERT into House values (1, 1, 1, 'А', 'Индивидуальный', 1970, 1980,
                          500.0, 400.0, 0.0, 2, 2, 45,
                          1, 34, true, false, false, true,
                          false, false, true, true, false, 1,
                          1980, 1970, 1975, 200.0, '2004-12-12', true,
                          4, 585.4, 4, 2, 170.0);

INSERT into Flat values (1, 1, 2, 36);

INSERT into CapitalRepair values (1, 1, 'Очистка и антисептирование древесины', 2008);

INSERT into HouseNumber values (1, 1, 38);
```

Рисунок 8 – Запрос на ввод данных о доме

```
SELECT district, street, housing, `character` FROM House RIGHT JOIN Street ON House.street_id = street.street_id RIGHT
```

Рисунок 9 – Запрос на вывод адресов домов

```
UPDATE House SET hasAccident = true WHERE house_id=3;
```

Рисунок 10 - Запрос на обновление данных об доме (добавление информации о наличии аварийности)

```
UPDATE House SET yearReconstruct = 2022 WHERE house_id=3;
```

Рисунок 11 - Запрос на обновление данных о доме по его id (добавление информации о реконструкции)

```
SELECT district, street, housing, `character`, yearBuild FROM House JOIN Street ON House.street_id = street.street_id
```

Рисунок 12 - Запрос для поиска домов, построенных после 1990 года.

```
SELECT district, street, housing, `character`, yearBuild, house_id FROM House JOIN Street ON House.street_id = street.
```

Рисунок 13 - Запрос для подсчета кол-ва аварийных домов с id не равным определенному, имеющих определенный год строения.

3.3. Сравнение моделей

Для данной БД реализация в MongoDB "выигрывает" реализацию в SQL по скорости запросов: несмотря на то, что каждый из примеров требует 1 запрос, из-за медленных JOIN'ов в SQL, NoSQL реализация оказывается быстрее.

Вес реляционной базы данных - 23.3 MB, вес нереляционной базы данных - 29,7 MB.

Избыточность реляционной базы данных - 1.5, избыточность нереляционной базы данных - 1.8.

Анализируя модели данных и полученные результаты, приходим к выводу, что обе модели растут с линейной скоростью.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

4.1. Описание приложения

Серверная часть в основе своей использует nodejs и взаимодействует с mongodb.

Клиентская часть в основе своей использует библиотеку React.

Для просмотра информации о каталоге жилого фонда достаточно просто зайти на сайт. Загружать данные (осуществлять импорт НД) имеет возможность только администратор, который может зарегистрироваться через отдельную страницу /entrance.

Снимки экранов разработанного приложения представлены в приложении Б.

4.2. Используемые технологии

Для клиентской части используется следующий стек технологий:

- язык программирования JavaScript;
- библиотека React;

Для серверной части используется следующий стек технологий:

- язык программирования JavaScript
- платформа nodejs
- база данных MongoDB.

Для сборки проекта используется Docker Compose.

ЗАКЛЮЧЕНИЕ

В ходе данной работы был разработан макет веб-приложения каталога жилого фонда СПб, на основании которого был реализован прототип такого приложения, содержащий основной функционал. Приложение позволяет просматривать список жилого фонда, фильтровать его, производить поиск объектов фонда на карте, смотреть статистику по каталогу.

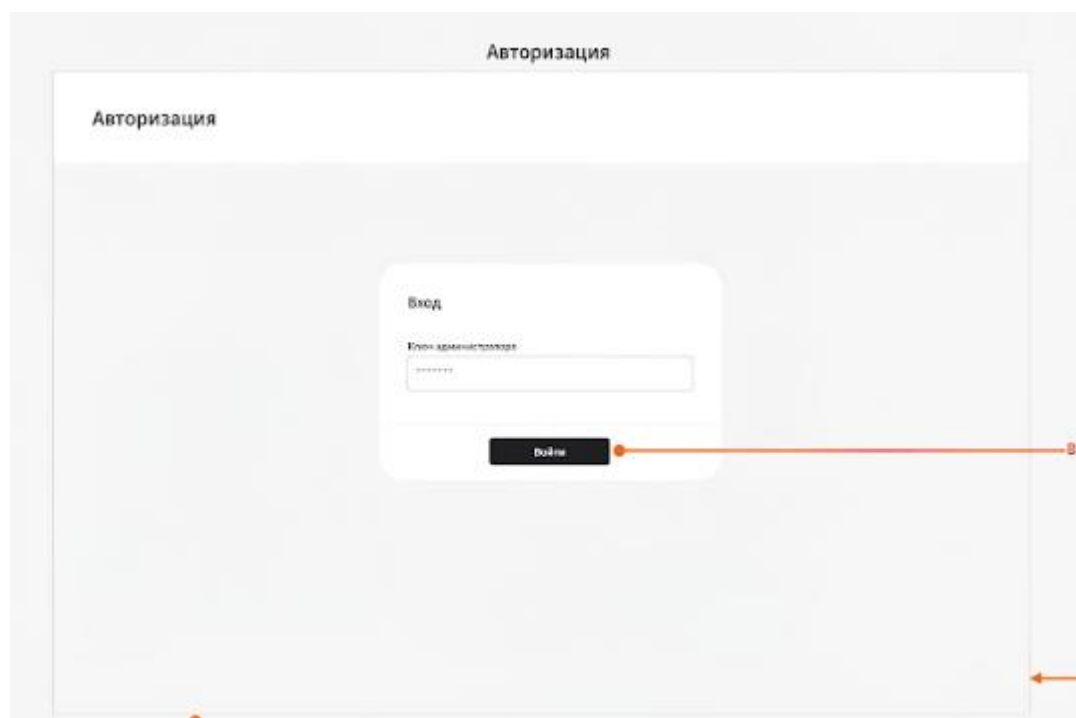
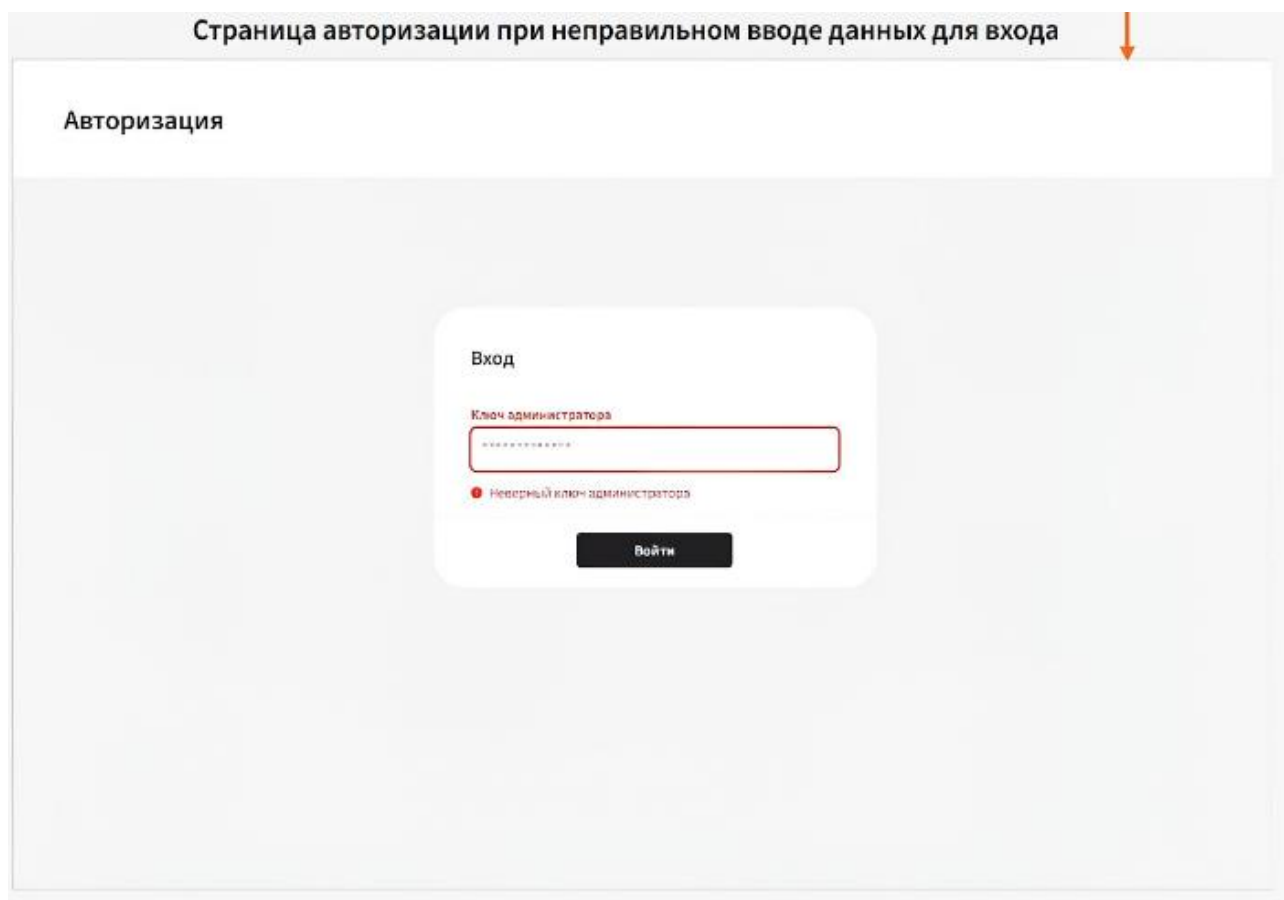
В дальнейшем данное приложение можно расширить, добавив новые модули.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. NodeJS [Электронный ресурс] URL: <https://nodejs.org/en/>
2. JavaScript [Электронный ресурс] URL:
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
3. MongoDB Documentation [Электронный ресурс] URL:
<https://www.mongodb.com/docs>
4. React Documentation [Электронный ресурс]. URL:
<https://reactjs.org/docs/getting-started.html> (дата обращения
25.10.2022)
5. Docker Documentation [Электронный ресурс]. URL:
<https://docs.docker.com>

ПРИЛОЖЕНИЕ А

МАКЕТ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА



Модальное окно для фильтрации статистики жилищного фонда

Каталог Жилья

Статистика жилищного фонда

Нижче представлені графіки со статистикою о жилых домах

Павел Ковалев

Фильтры

Фильтрация домов для отображения статистики

Улица: Брянская, Брестская, Минская

Дом: 8/30, 7/30

Литера: П, А

Район: Красный молот, Петроградский

Тип коммунальной квартиры: -

Количество коммунальных комнат: 1-2

Общее количество комнат: 1,2

Серия/Тип Проекта: -

Применить

Справка по взаимодействию с фильтрацией

Каталог Жилья

Статистика жилищного фонда

Нижче представлені графіки со статистикою о жилых домах

Павел Ковалев

Фильтры

Фильтрация домов для отображения статистики

Улица: Брянская, Брестская, Минская

Дом: 8/30, 7/30

Литера: П, А

Район: Красный молот, Петроградский

Тип коммунальной квартиры: -

Количество коммунальных комнат: 1-2

Общее количество комнат: 1,2

Серия/Тип Проекта: -

Применить

Параметры поддерживают ввод нескольких значений и диапазонов значений.
 Чтобы задать несколько значений, вводите их через запятую, например «Пушкинская, Фрунзенская»
 Чтобы задать диапазон значений, вводите его через тире, например «1-2».
 Можно задать несколько диапазонов, например «1-3, 5-6».
 Если значение не задано (в поле стоит «-»), то по умолчанию будет использоваться все возможные значения данного поля.

Вход на Карту

Каталог Жилья

Каталог

Карта

Статистика

Импорт НД

Экспорт НД

Выход

Каталог жилищного фонда

Павел Ковалев

Ниж представлена таблица с информации о жилых домах

Search

Фильтры

Сортировать: Литера

Улица	Дом	Литера	Район	Тип коммунальной квартиры	Количество коммунальных	Общее количество комнат	Серия, тип Проекта
Брянская	8/30	П	Красный мост	2 комнаты	2	2	Индивидуальный
Минская	7/30	А	Красный мост	1 комната	4	1	Индивидуальный
Восточная	6/30	А	Красный мост	3 комнаты	2	3	Индивидуальный
Люблинская	5/30	Б	Красный мост	1 комната	4	1	Индивидуальный
Варшавская	5/30	Б	Красный мост	1 комната	4	1	Индивидуальный

1 2 3 4 5 ... 20

Нажатие на фильтрацию

Вход в статистику

Каталог Жилья

Каталог

Карта

Статистика

Импорт НД

Экспорт НД

Выход

Статистика жилищного фонда

Павел Ковалев

Ниж представлены графики со статистикой о жилых домах

Фильтры

Площадь всех зданий для каждого района, кв. м

Количество квартир

Площадь незанятых жилых помещений, кв. м

Количество проживающих

31

Каталог Жилья

Каталог

Карта

Статистика

Импорт НД

Экспорт НД

Выход

Каталог жилищного фонда

Нижне представлена таблица с информации о жилых домах

Search

Фильтры

Сортировка: Литера

Улица	Дом	Литера	Район	Тип коммунальной квартиры	Количество коммунальнх	Общее количество комнат
Брянская	8/30	П	Красный холот	2 комнаты	2	2
Имская	7/30	А	Красный холот	1 комната	4	1
Врестская	6/30	А	Красный холот	3 комнаты	2	3
Лобинская	5/30	Б	Красный холот	1 комната	4	1
Варшавская	5/30	Б	Красный холот	1 комната	4	1

12345...20

Каталог жилищного фонда

Нижче представлена таблиця з інформації про житлові будинки

Павел Ковалев

Фильтрация домов по параметрам

Улица: Брянская

Дом: 8/30

Литера: П

Район: Красный молот

Тип коммунальной квартиры: 2 комнаты

Количество коммунальных комнат: 2

Общее количество комнат: 2

Серия/Тип Проента: Индивидуальный

Применить

Справка по взаимодействию с фильтрацией

Каталог жилищного фонда

Павел Ковалев

Параметры поддерживают ввод нескольких значений и диапазонов значений.
Чтобы задать несколько значений, введите их через запятую, например «Пушкинская, Брусиловская»
Чтобы задать диапазон значений, введите его через тире, например «1-3». Можно задать несколько диапазонов, например «1-3, 5-6».

Фильтрация домов по параметрам

Улица: Брянская, Брестская, Минская

Дом: 8/30, 7/30

Литера: П, Л

Район: Красный молот, Петроградский

Тип коммунальной квартиры: -

Количество коммунальных комнат: 1-2

Общее количество комнат: 1, 2

Серия/Тип Проекта: -

Применить

Карта жилищного фонда

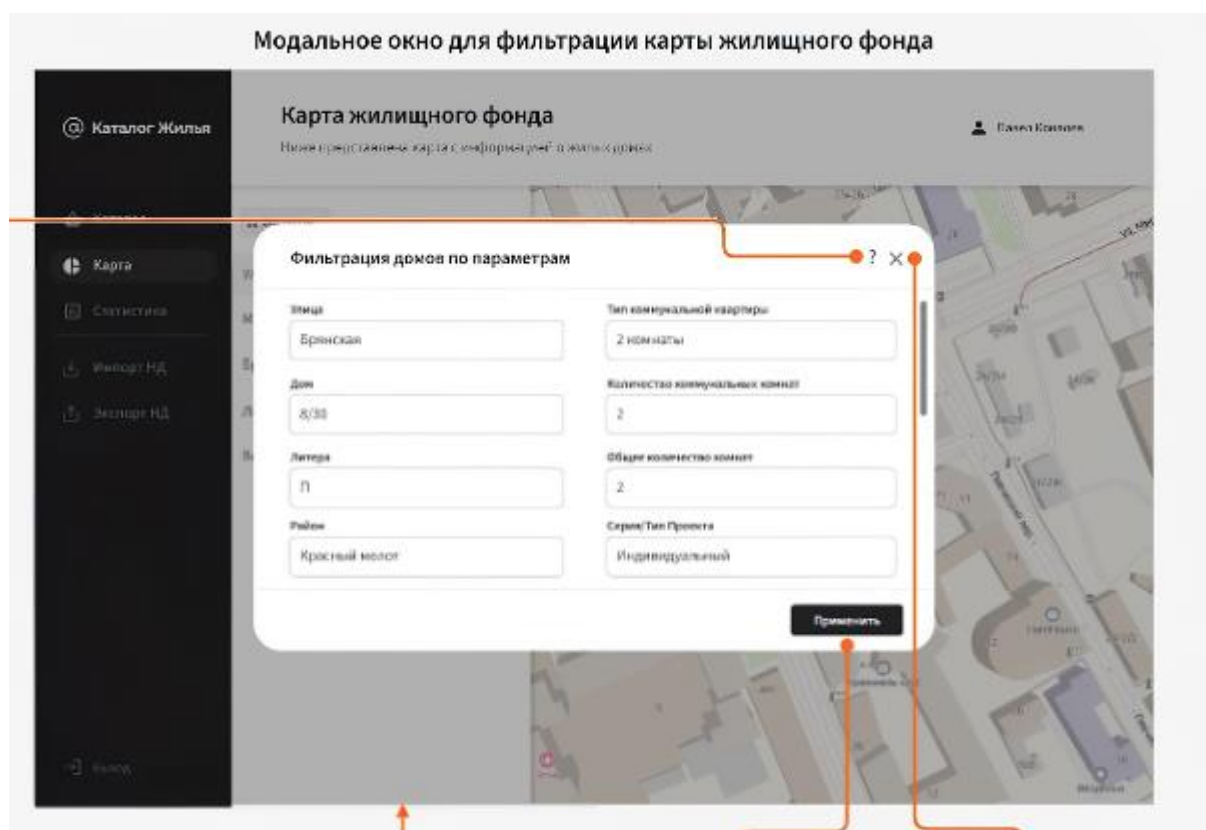
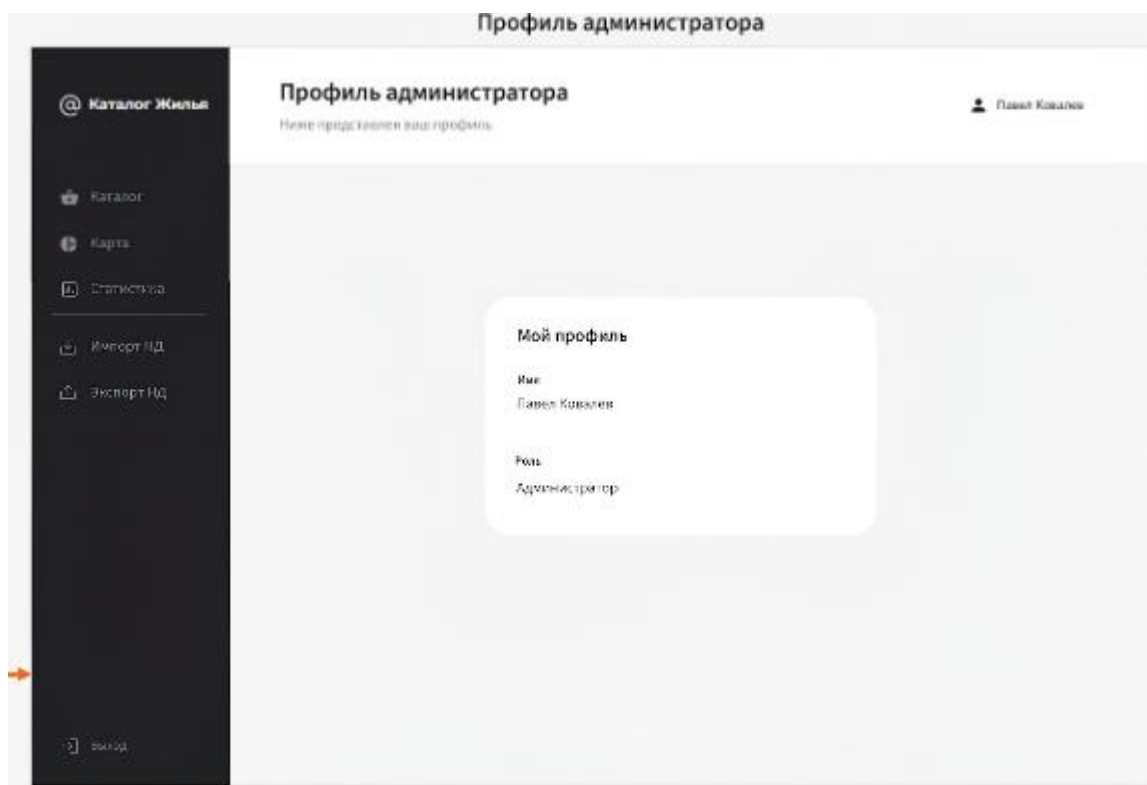
Павел Ковалев

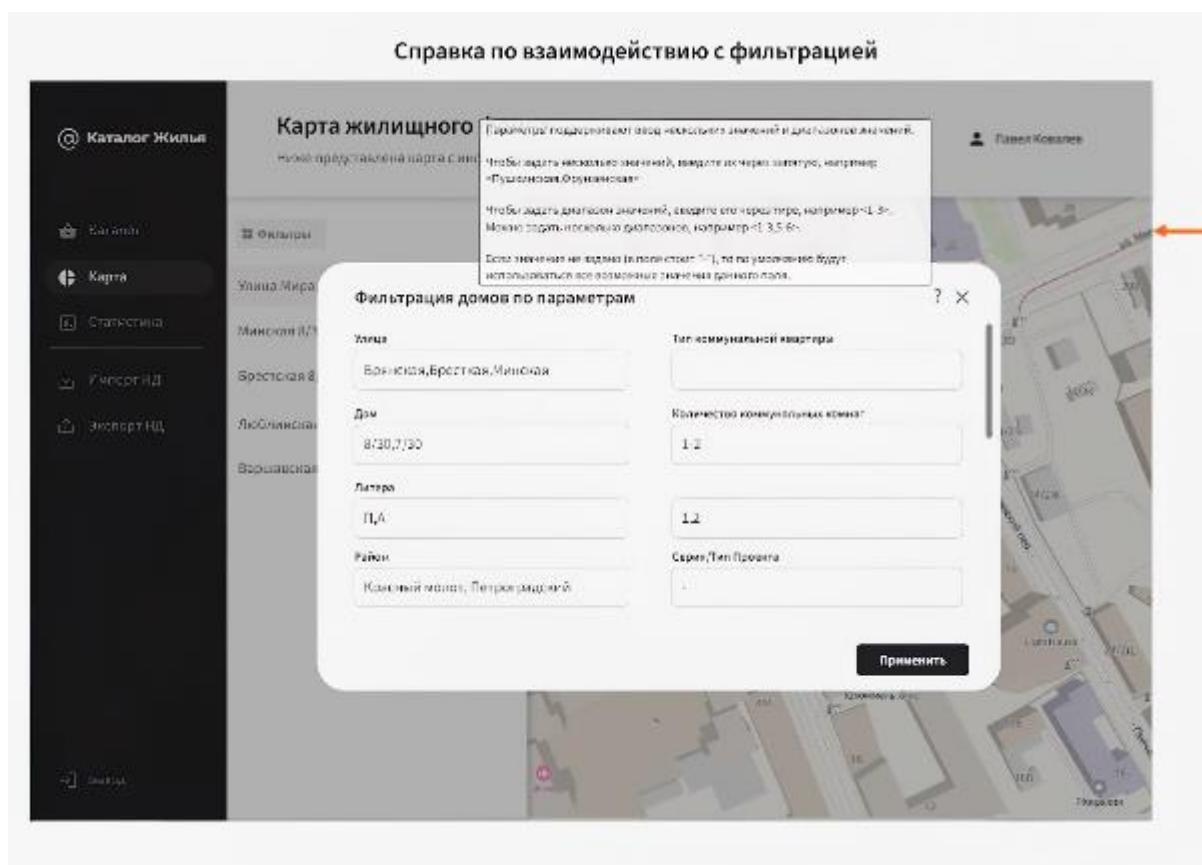
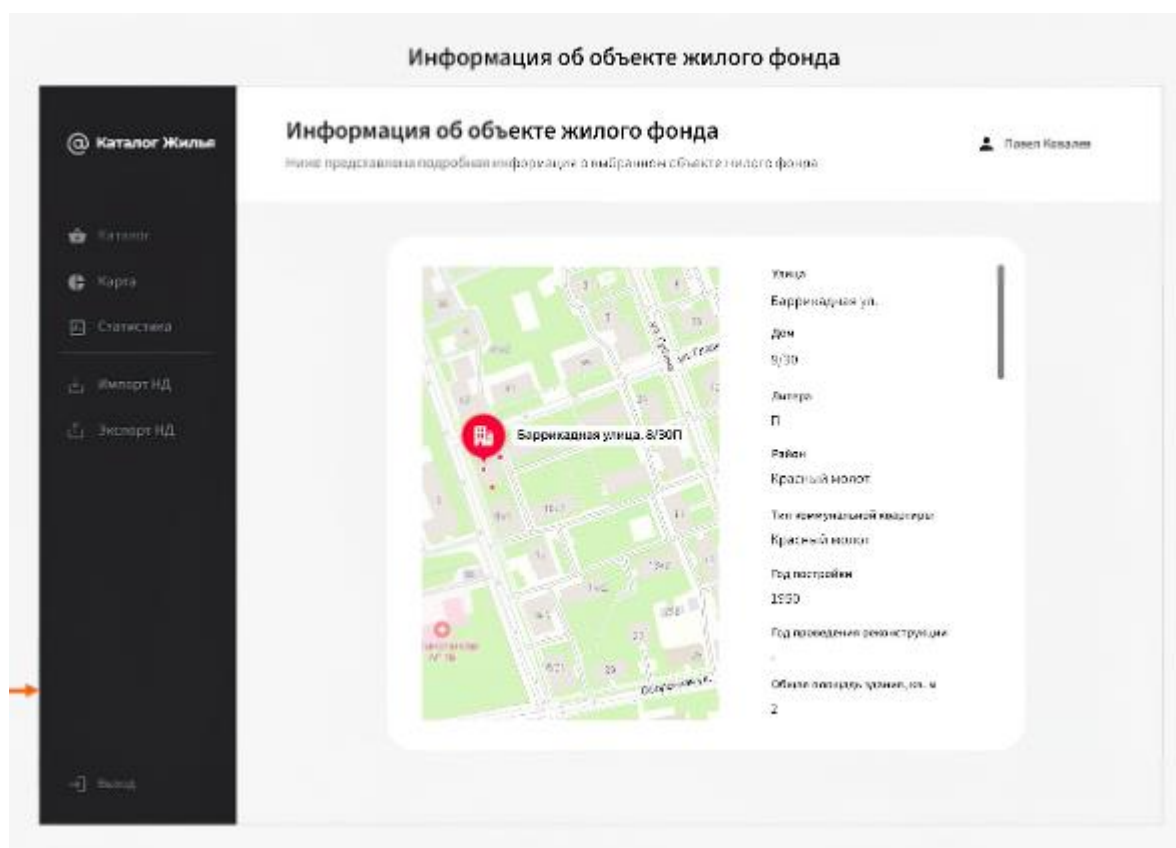
Ниже представлена карта с информацией о жилых домах

Фильтры

- Улица Мира, д.17, Санкт-Петербург
- Минская 8/30, Красный молот
- Брестская 8/30, Красный молот
- Лобановская 8/30, Красный молот
- Варшавская 8/30, Красный молот

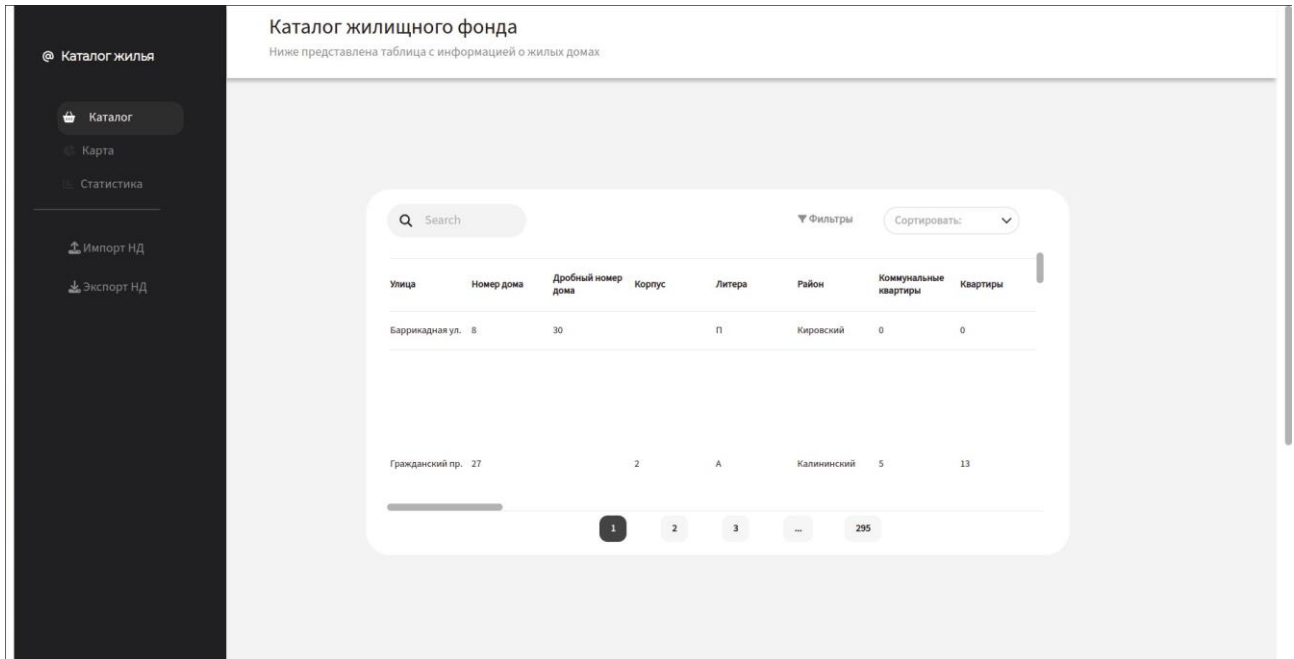
Карта показывает район с улицей Мира, д. 17, отмеченной красной меткой.



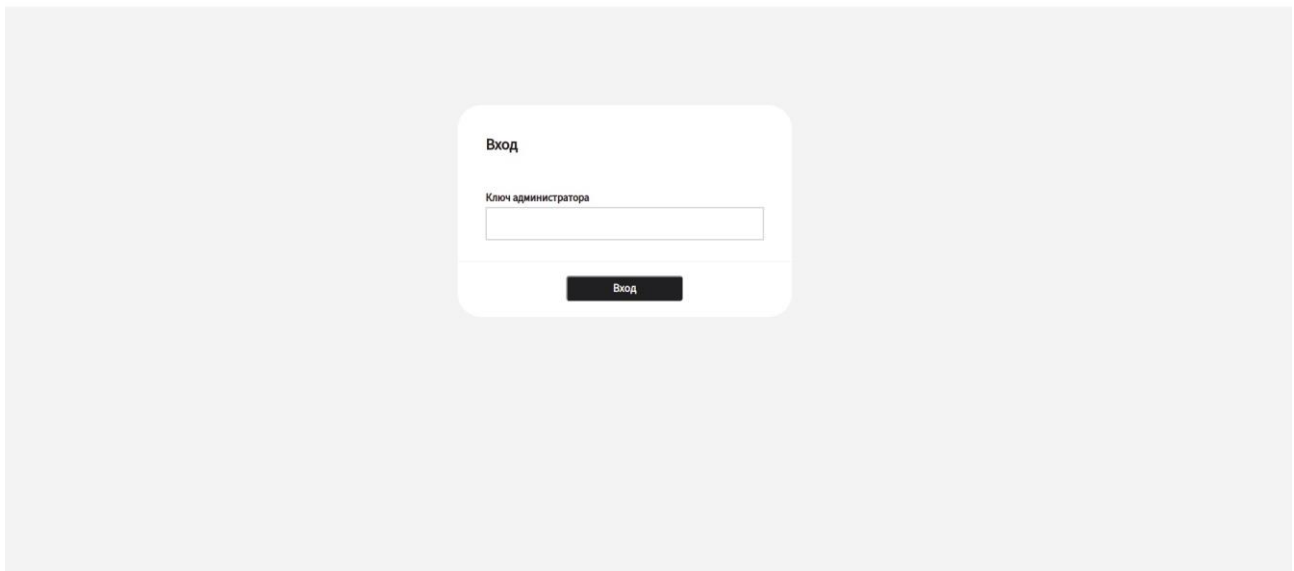


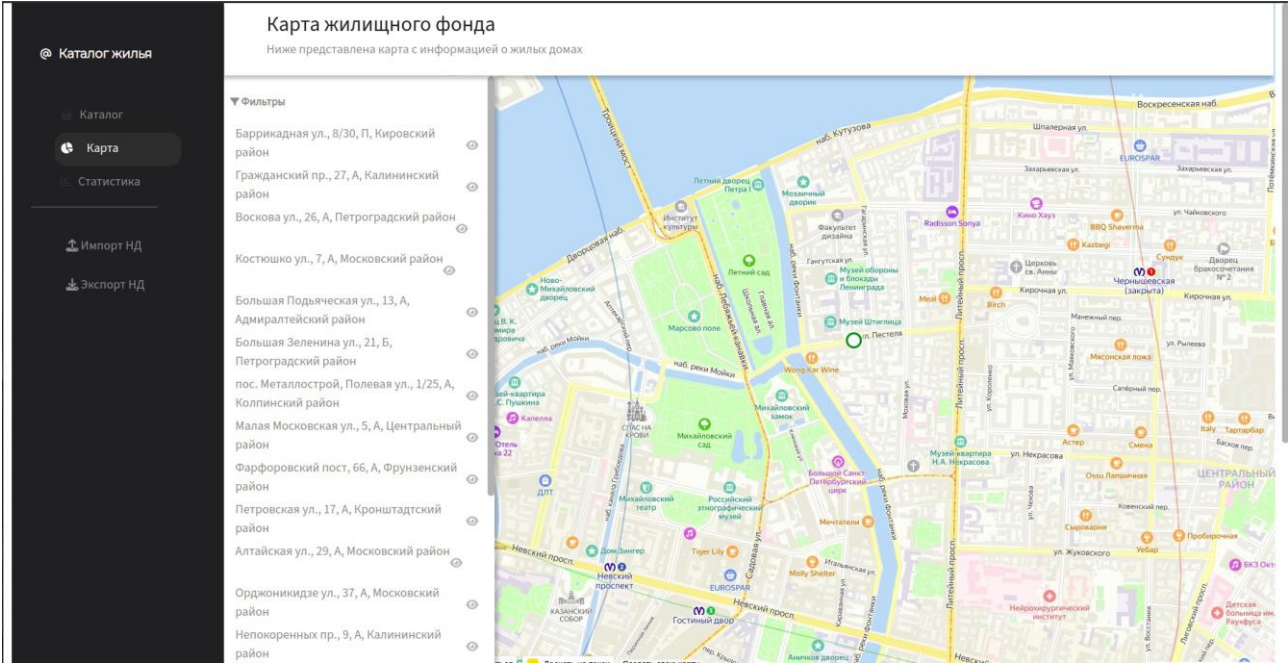
ПРИЛОЖЕНИЕ Б

СНИМКИ ЭКРАНА ПРИЛОЖЕНИЯ



Авторизация





Информация об объекте жилого фонда

Ниже представлена подробная информация о выбранном объекте жилого фонда



Улица
Баррикадная ул.

Номер дома
8

Дробный номер дома
30

Корпус
-

Литера
П

Район
Кировский