

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский политехнический университет»

Кафедра «Инфокогнитивные технологии»
Образовательная программа «Веб-технологии»

Отчет по курсовому проекту
по дисциплине «Инженерное проектирование»

Тема: «Коворкинг»

Выполнил:

Студент группы 191-321

Перфильев А. О.

подпись, дата

Принял:

Старший преподаватель

Данышина М.В.

подпись, дата

Москва 2021

Оглавление

ВВЕДЕНИЕ	3
ОСНОВНАЯ ЧАСТЬ	4
Анализ конкурентов.....	4
Инфологическая схема базы данных.....	5
Физическая схема базы данных.....	6
Кейсы использования приложения.....	7
Этапы реализации проекта.....	8
Современные CSS решения	10
Типовые запросы к базе данных	11
ЗАКЛЮЧЕНИЕ	12
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	13

ВВЕДЕНИЕ

Модель разработки – каскадная. Проект решает задачу создания административной части проекта по тематике «Коворкинг». При разработке я ориентировался на успешные аналоги, такие как:

<https://space1.ru/>

<https://synergyspace.ru/>

<https://www.wework.com/>

<https://www.denswap.com/>

<https://www.showcase.com/>

ОСНОВНАЯ ЧАСТЬ

Анализ конкурентов

Имя	URL	Особенности сайта
Space 1	https://space1.ru/	Немного пространств и краткая информация о них
Synergy Space	https://synergyspace.ru/	Мало пространств, но подробное описание преимуществ каждого
WEWORK	https://www.wework.com/	Множество рабочих пространств с подробным описанием и указанием местоположения на карте
DENSWAP	https://www.denswap.com/	Несколько пространств в США с удобным описанием и указанием преимуществ
showcase	https://www.showcase.com/	Очень много мест по всему миру

Объединяющие факторы: все сайты очень похожи функционалом, однако у сайтов с большим количеством объектов присутствует поиск, сортировка и фильтрация.

Инфологическая схема базы данных

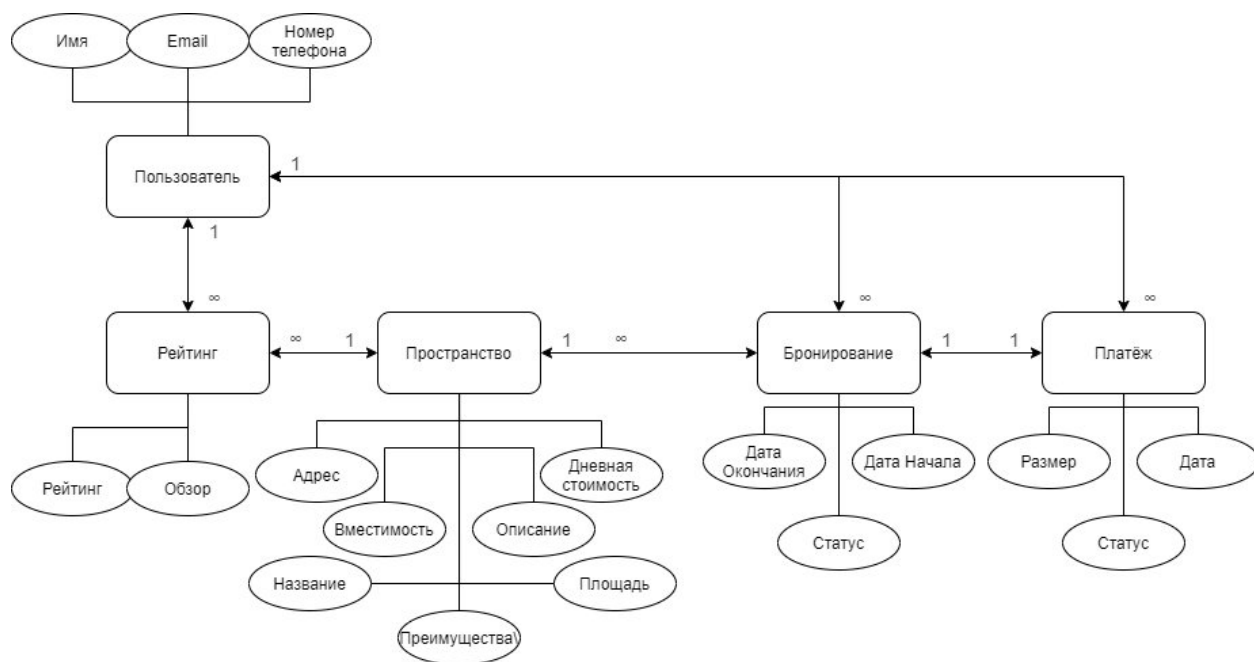


Рис. 1 “Инфологическая схема базы данных”

Физическая схема базы данных

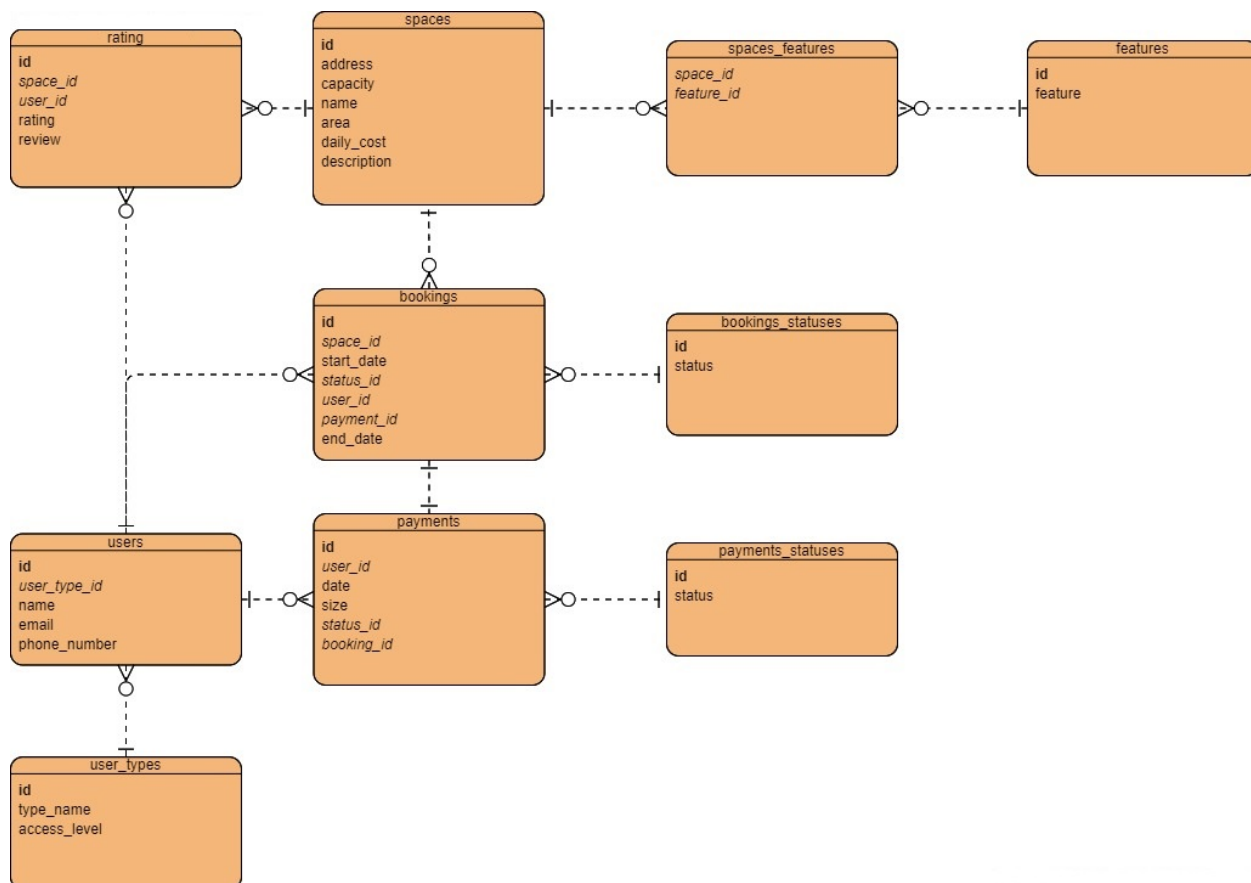


Рис. 2 “Физическая схема базы данных”

Кейсы использования приложения

1. Администратор добавляет новое пространство
2. Администратор добавляет новые преимущества для пространств
3. Администратор удаляет пользователя
4. Администратор экспортирует список пространств
5. Администратор импортирует список пользователей из другого приложения
6. Оператор следит за статистикой в дашборде
7. Пользователь регистрируется
8. Пользователь выбирает понравившееся пространство
9. Пользователь бронирует пространство на определенный промежуток времени
10. Пользователь оплачивает свою бронь
11. Пользователь ставит оценку пространству и пишет отзыв

Этапы реализации проекта

1. Анализа конкурентов

Я нашел пять сайтов по тематике коворкинга из выдачи поисковых систем, среди которых как отечественные проекты, так и зарубежные.

Проанализировал структуру и функциональность каждого из них. Сделал вывод о том, что их структура и функциональность по большей части одинаковы, а значит скорее всего это эталон и стоит его придерживаться.

2. Проектирование инфологической модели предметной области

На основе проделанного ранее анализа я составил схему «Сущность связь», в которое отразил, какие основные сущности присутствуют в проекте, какие у них есть свойства и поля, а так же как они связаны между собой.

3. Проектирование физической структуры

На основе инфологической схемы базы данных я создал ER-диаграмму, в которой отразил конкретные таблицы, столбцы этих таблиц, а так же связи разных типов между ними, такие как «Один к одному», «Один ко многим» и «Многие ко многим».

4. Создание репозитория проекта

На github я создал репозитори для проекта, связал его с локальным репозиторием, создал 2 ветки «master» и «dev».

5. Создание Django-приложения

Я изучил материалы по теме в сети интернет и приступил к созданию приложение, прежде всего я создал модели, которые мигрировал в базу данных, а так же настроил роутинг и файл settings.py.

6. Настройка административного интерфейса Django

Я добавил возможность редактирования данных в базе через административную панель. Дополнительно я добавил возможность поиска, сортировки и фильтрации данных в каждой таблице.

7. Наполнение базы данных

Я наполнил базу данных различной информацией о пространствах, бронированиях, оплатах, пользователях, отзывах, а так же различных преимуществах.

8. Реализация REST API

Я добавил возможность управления данными приложения со стороны, реализовал RESTfull API, а так же разобрался с тестированием API через сервис POSTMAN.

9. Реализация экспорта и импорта данных

С помощью django-import-export я реализовал возможности импорта и экспорта данных в таблицах.

10. Документирование

Задokumentировал всю проделанную работу.

11. Создание фронтенда

Создал много различных страниц, вывел на них информацию из БД. Добавил формы для добавления бд, реализовал GRUD. С помощью препроцессора SASS написал стили для сайта. Подключил несколько сторонних JS и CSS библиотек.

12. Заполнение оценочного листа

Заполнил оценочный лист в соответствии с проделанной работой и функциональными возможностями приложения.

Современные CSS решения

```

1. .single-space_container {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 20px 20px;
    grid-auto-flow: row;
    grid-template-areas:
        "fullSpace_img fullSpace_info"
        "fullSpace_reviews fullSpace_reviews";

```

Используется технология Grid.

```

2. .single-space-card__img {
    width: 100%;
    height: 500px;
    border-radius: 5px;
    object-fit: cover;
    @media screen and (max-width: 600px) {
        height: 300px;
    }
}

```

Для обрезания картинок используется свойство object-fit.

```

3. .textarea {
    resize: none;
    height: 200px !important;
    padding: 15px;
}

```

Для текстового поля используется свойство resize.

```

4. .space-card-wrap {
    width: calc(100% / 4);
    margin-bottom: 20px;
    @media screen and (max-width: 1200px) {
        width: calc(100% / 3);
    }
    @media screen and (max-width: 800px) {
        width: calc(100% / 2);
    }
    @media screen and (max-width: 400px) {
        width: 100%;
    }

    a {
        color: $dark;
        text-decoration: none;
    }
}

```

Для вычисления размеров используется функция calc()

```

5. .space-card_description {
    overflow: hidden;
    display: -webkit-box;
    -webkit-line-clamp: 3;
    line-height: 160%;
    -webkit-box-orient: vertical;
    padding-bottom: 5px;
    border-bottom: 1px solid $grey;
    margin-bottom: 5px;
}

```

Используется свойство display со значением -webkit-box, чтобы обрезать текст до 3-х строк с «...» в конце.

Типовые запросы к базе данных

- 1) `SELECT address FROM spaces_space WHERE capacity > 5 ORDER BY id DESC;`
- 2) `SELECT daily_cost FROM spaces_space WHERE area > 200 ORDER BY id DESC;`
- 3) `INSERT INTO spaces_space (address, capacity, name, area, daily_cost, description) VALUES ('Some adress', 20, 'Some name', 400, 5000, 'Some description');`
- 4) `UPDATE spaces_space SET capacity = 10 WHERE id = 2;`
- 5) `DELETE FROM main_user WHERE username = 'Vlad';`

ЗАКЛЮЧЕНИЕ

Результатом является готовое django-приложение, в котором реализован CRUD, есть административная часть, разделение на пользователей, dashboard и много другого.

Доступы:

Аккаунт админа:

Login: admin

Password: admin

Аккаунт оператора:

Login: Operator

Password: oper1234

Макет в figma:

<https://www.figma.com/file/33mkts24QAwdKq4LnzR4Om/Coworking?node-id=0%3A1>

Лендинг: <http://lexa070301.bhuser.ru/coworking/landing/>

Сайт на хостинге: <http://coworking.std-944.ist.mospolytech.ru>

GitHub репозиторий: <https://github.com/Lexa070301/coworking>

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

<https://docs.djangoproject.com/en/3.1/>

<https://developer.mozilla.org/ru/docs/Learn/Server-side/Django>

<https://djbook.ru/>

<https://realpython.com/get-started-with-django-1/>

<https://django-import-export.readthedocs.io/en/latest/>

<https://www.django-rest-framework.org/>