

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

ИНДИВИДУАЛЬНОЕ ДОМАШНЕЕ ЗАДАНИЕ
по дисциплине «Введение в нереляционные базы данных»
Тема: Каталог всех дворов-колодцев Санкт-Петербурга

Студенты гр. 1384 _____ Белокобыльский И.В.
Преподаватель _____ Мухачева П.Р.
_____ Шаганов В.А.
_____ Заславский М.М.

Санкт-Петербург
2024

ЗАДАНИЕ

Студент Белокобыльский И.В.

Студент Мухачева П.Р.

Студент Шаганов В.А.

Группа 1384

Тема: Каталог всех дворов-колодцев Санкт-Петербурга

Исходные данные:

Задача - составить каталог всех дворов-колодцев в СПБ. Смысл сервиса - дать возможность найти все дворы, а также вести учет персональных достижений (Кто в скольки дворах побывал). Страницы: профиль пользователя, карточка двора, поиск пользователей и дворов, карта.

Содержание пояснительной записи:

«Содержание»

«Введение»

«Сценарии использования»

«Модель данных»

«Разработанное приложение»

«Выводы»

«Приложения»

«Список использованных источников»

Предполагаемый объем пояснительной записи:

Не менее 15 страниц.

Дата выдачи задания: 05.09.2024

Дата сдачи реферата: 24.12.2024

Дата защиты реферата: 24.12.2024

Белокобыльский И.В.

Мухачева П.Р.

Студенты гр. 1384

Шаганов В.А.

Преподаватель

Заславский М.М.

АННОТАЦИЯ

В рамках данного курса предполагалось разработать какое-либо приложение в команде на одну из поставленных тем. Была выбрана тема создания приложения для ведения каталога дворов-колодцев города Санкт-Петербурга на основе СУБД Neo4j. Найти исходный код и всю дополнительную информацию можно по ссылке: <https://github.com/moevm/nosql2h24-wells>

SUMMARY

As part of this course, it was supposed to develop an application in a team on one of the set topics. The topic of creating an application for maintaining a catalog of well yards in St. Petersburg based on the Neo4j database was chosen. You can find the source code and all additional information here: <https://github.com/moevm/nosql2h24-wells>

СОДЕРЖАНИЕ

1.	Введение	6
1.1.	Актуальность проблемы	6
1.2.	Постановка задачи	6
1.3.	Предлагаемое решение	6
1.4.	Качественные требования к решению	6
2.	Сценарии использования	7
2.1.	Макеты UI	7
2.2.	Сценарии использования	12
2.6.	Вывод об операциях	20
3.	Модель данных	21
3.1.	Нереляционная модель данных	21
3.2.	Реляционная модель данных	25
3.2.	Сравнение моделей	28
4.	Разработанное приложение	30
4.1.	Краткое описание приложения	30
4.2.	Использованные технологии	30
5.	Выводы	31
5.1.	Достигнутые результаты	31
5.2.	Недостатки и пути для улучшения	31
5.3.	Будущее развитие решения	31
6.	Приложения	33
6.1.	Документация по сборке и развертыванию приложения	33
6.2.	Инструкция для пользователя	33
7.	Литература	35

1. ВВЕДЕНИЕ

1.1. Актуальность проблемы

Актуальность создания веб-приложения для ведения каталога дворов-колодцев Санкт-Петербурга обусловлена уникальностью этого явления, которое вызывает интерес, как у жителей города, так и у туристов. Однако, из-за сложности доступа и большого количества локаций возникают трудности в поиске наиболее интересных мест из всего многообразия дворов Северной столицы.

1.2. Постановка задачи

Задача проекта заключается в разработке веб-приложения, которое позволяет пользователям:

- найти дворы;
- вести учет персональных достижений;
- просматривать отзывы о дворах от других пользователей;
- просматривать статистику посещений.

1.3. Предлагаемое решение

Для реализации предлагается веб-приложение с использованием HTML, CSS, JS для клиентской части, Python для серверной части, Neo4j для хранения данных, Docker для контейнеризации и docker-compose развертывания.

1.4. Качественные требования к решению

Решение должно быть удобным, производительным, надежным и легко расширяемым, удобным механизмом поиска, а также возможностью развертывания на различных платформах с использованием Docker и docker-compose.

2. СЦЕНАРИИ ИСПОЛЬЗОВАНИЯ

2.1. Макет UI

На рисунках 1-18 представлены макеты приложения.



Логин

Пароль

Рисунок 1 – Страница регистрации и аутентификации

Поиск по дворам

Двор дома-угола
Садовая ул., д. 128, наб. реки Фонтанки, д. 199
★ 4.8

Двор-всемирный
Малый просп. П. С., д. 16
★ 3

Бармалеева, 4
Бармалеева, 4
★ 4.2

Дом-кольцо
Набережная реки Фонтанки, д. 92; Горюховая, д. 59
★ 4.5

Каменноостровский двор
Каменноостровский пр., д. 44б
★ 3

Поиск по пользователям

ivanov aka Иван Иванович
4 двора

petrov aka Петров Петр Петрович
4 двора

sidorov aka Сидоров Николай Николаевич
3 двора

alexander aka Михаилос Александр Андреевич
4 двора

Ефим aka Ефим Ефимович

Рисунок 2 – Главная страница для неаутентифицированного пользователя

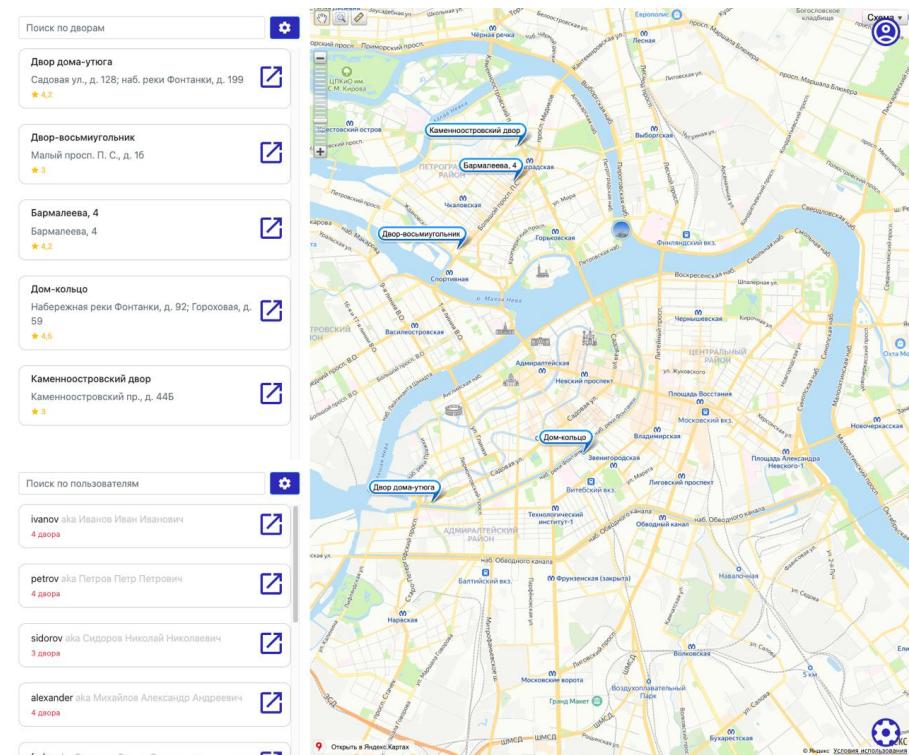


Рисунок 3 – Главная страница для аутентифицированного пользователя

Двор-восьмиугольник
★ 3 / 5

Малый просп. П. С., д. 16

Двор посетили 4 пользователей

Отзывы

Дата посещения	Ваша оценка двора:	
дд. мм . гггг	<input type="radio"/> 1 <input type="radio"/> 2 <input type="radio"/> 3 <input type="radio"/> 4 <input type="radio"/>	Сохранить

Добавить свой отзыв

Отправить

petrov aka Петров Петр Петрович 4 ★ 2024-11-20

Мне нравится

sidorov aka Сидоров Николай Николаевич 2 ★ 2024-11-20

Мне не понравилось

alexander aka Михайлов Александр Андреевич 3 ★ 2024-11-21

Мне нравится

ivanov aka Иванов Иван Иванович 3 ★ 2024-11-29

Мне нравится

Рисунок 4 – Страница двора

ivanov
Иванов Иван Иванович
Дворов посещено: 4
На сайте с 11.12.2024

Посещённые дворы

Поиск по дворам

дд.мм.гггг дд.мм.гггг

Рейтинг, от Рейтинг, до

- Двор-восьмиугольник**
Малый просп. П. С., д. 16
★ 3 Посетил 2024-11-29
- Бармалеева, 4**
Бармалеева, 4
★ 4,2 Посетил 2024-11-21
- Дом-кольцо**
Гороховая, д. 59, Набережная реки Фонтанки, д. 92
★ 4,5 Посетил 2024-11-20
- Двор дома-утюга**
Садовая ул., д. 128, наб. реки Фонтанки, д. 199
★ 4,2 Посетил 2024-11-28

Рисунок 5 – Страница профиля другого пользователя

Avatar **admin**
Дворов посещено: 0
На сайте с 22.12.2024

Посещённые дворы

Поиск по дворам

дд.мм.гггг дд.мм.гггг

Рейтинг, от Рейтинг, до

Рисунок 6 – Страница своего профиля

Редактирование профиля

Никнейм

Фамилия

Имя

Отчество

URL фотографии

Сохранить

Рисунок 7 – Редактирование профиля

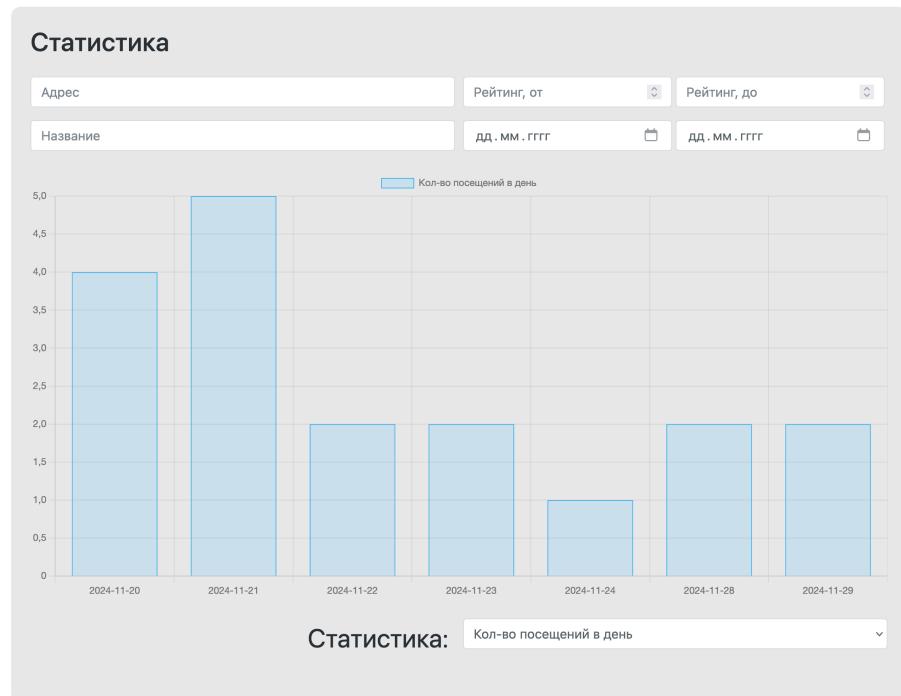


Рисунок 8 – Страница статистики

Расширенный поиск дворов

Адрес	Название	Рейтинг, от	Рейтинг, до
Фильтрация по координатам		Посещений, от Посещений, до	
Найти			
Дом-кольцо Набережная реки Фонтанки, д. 92; Гороховая, д. 59 ★ 4,5			

Рисунок 9 – Страница расширенного поиска дворов

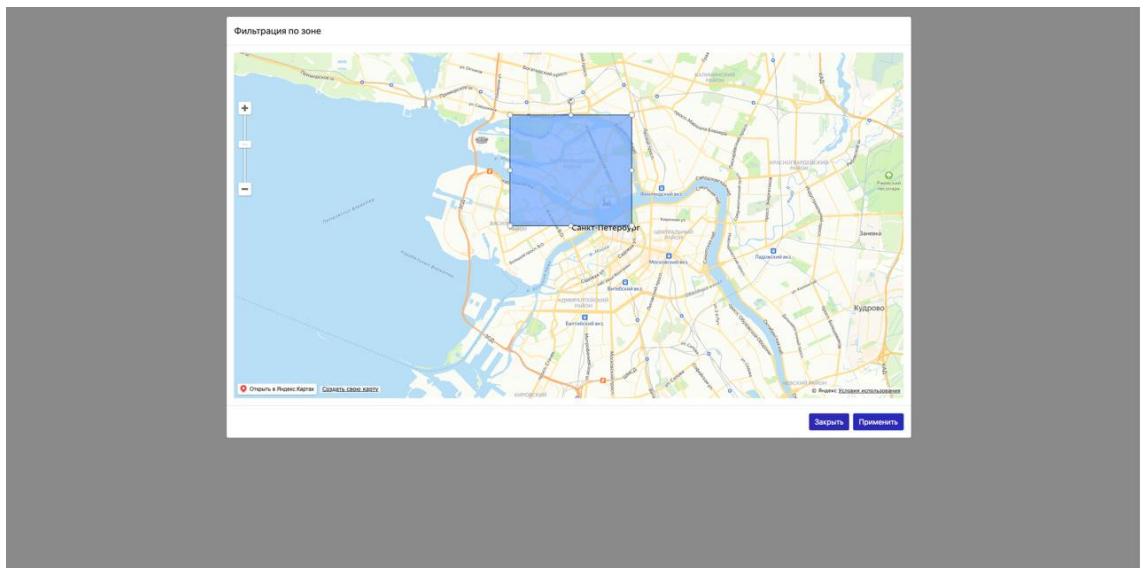


Рисунок 10 – Страница расширенного поиска дворов – фильтрация по зоне (координатам)

Расширенный поиск пользователей

Фамилия	Имя	Отчество	
Никнейм		Посетил, от	Посетил, до
Найти			
ivanov aka Иван Иванович 4 двора petrov aka Петров Петр Петрович 4 двора sidorov aka Сидоров Николай Николаевич 3 двора alexander aka Михайлов Александр Андреевич 4 двора fedor aka Федоров Федор Федорович 3 двора admin aka 0 двора			

Рисунок 11 – Страница расширенного поиска пользователей

Администрирование

Import

Экспорт

Отзывы / Посещения

Статистика

Рисунок 12 – Страница администрирования

Поиск по посещениям / отзывам

По дворам

Адресс Название Рейтинг, от Рейтинг, до

Фильтрация по координатам

По пользователям и посещениям

Фамилия Имя Отчество

Никнейм Посетил, от Посетил, до dd.mm.yyyy dd.mm.yyyy

С комментарием

Комментарий

Найти

ivanov aka Иванов Иван Иванович 5 ★ 2024-11-20
Было очень интересно!

sidorov aka Сидоров Николай Николаевич 4 ★ 2024-11-21

alexander aka Михайлов Александр Андреевич 5 ★ 2024-11-20
Хороший двор!

fedor aka Федоров Федор Федорович 4 ★ 2024-11-21

petrov aka Петров Петр Петрович 4 ★ 2024-11-28

ivanov aka Иванов Иван Иванович 5 ★ 2024-11-28
Круто!

sidorov aka Сидоров Николай Николаевич 3 ★ 2024-11-29

Рисунок 13 – Страница расширенного поиска по посещениям

2.2. Сценарии использования

Страница 1 - Главная страница для неаутентифицированного

Действующее лицо: неаутентифицированный пользователь

Основной сценарий:

1. Пользователь видит главный экран с картой и списком всех дворов
2. Пользователь нажимает на кнопку регистрации и переходит на страницу аутентификации (страница 2)

Альтернативный сценарий 1:

1. Пользователь видит главный экран с картой и списком всех дворов
2. Пользователь нажимает на двор на карте / в списке дворов (возможно, воспользовавшись поиском) и переходит на страницу аутентификации (страница 2)

Альтернативный сценарий 2:

1. Пользователь видит главный экран с картой и списком всех дворов
2. Пользователь нажимает на пользователя из списка ТОП пользователей и попадает на страницу аутентификации (страница 2)

Альтернативный сценарий 3:

2. Пользователь нажимает на кнопку расширенного поиска и попадает на страницу расширенного поиска (страница 9)

Страница 2 - Аутентификация

Действующее лицо: неаутентифицированный пользователь

Основной сценарий:

1. Пользователя редиректит на страницу аутентификации при попытке совершить действие в другом СИ
2. Пользователь вводит свой никнейм в поле "Логин"
3. Пользователь вводит свой пароль в поле "Пароль"
4. Пользователь нажимает на кнопку "Войти / Зарегистрироваться"
5. Осуществляется переход на главную страницу (страница 3)

Альтернативный сценарий 1:

1. Пользователя редиректит на страницу аутентификации при попытке совершить действие в другом СИ

2. Пользователь придумывает новый никнейм и вводит его в поле "Логин"
 3. Пользователь придумывает пароль и вводит его в поле "Пароль"
 4. Пользователь нажимает на кнопку "Войти / Зарегистрироваться"
 5. Осуществляется переход на главную страницу (страница 3)
- Альтернативный сценарий 2:
1. Пользователя редиректит на страницу аутентификации при попытке совершить действие в другом СИ
 2. Пользователь вводит никнейм в поле "Логин"
 3. Пользователь вводит неверный пароль в поле "Пароль"
 4. Пользователь получает сообщение "Неверный логин или пароль"
 5. Переход на шаг 2 основного или альтернативного сценариев
- Страница 3 - Главная страница для аутентифицированного**
- Действующее лицо: аутентифицированный пользователь
- Основной сценарий:
1. Пользователь видит главный экран с картой и списком всех дворов и двумя кнопками: в правом верхнем углу кнопку профиля, в нижнем - кнопку администрирования
 2. Пользователь нажимает на кнопку своего профиля и переходит на страницу профиля (страница 6)
- Альтернативный сценарий 1:
2. Пользователь нажимает на двор на карте / в списке дворов (возможно, воспользовавшись поиском) и переходит на страницу двора (страница 4)
- Альтернативный сценарий 2:
2. Пользователь нажимает на пользователя из списка ТОП пользователей и попадает на страницу пользователя (страница 5)
- Альтернативный сценарий 3:
2. Пользователь нажимает на кнопку администрирования и попадает на администрирования (страница 11)

Альтернативный сценарий 4:

2. Пользователь нажимает на кнопку расширенного поиска в списке дворов и попадает на страницу расширенного поиска дворов (страница 9)

Альтернативный сценарий 5:

2. Пользователь нажимает на кнопку расширенного поиска в списке пользователей и попадает на страницу расширенного поиска пользователей (страница 10)

Страница 4 - Страница двора

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь видит страницу с названием, адресом, координатами, количеством посещений и рейтингом двора, а также отзывы других авторизованных пользователей — текст отзыва, оценку, логин автора и дату публикации — и поле для создания собственного отзыва

2. Пользователь скроллит страницу вниз и читает отзывы

3. Пользователь пишет свой отзыв в отведённом для этого поле рядом с надписью «Отзывы», выбирает значение оценки, указывает дату посещения и нажимает кнопку отправить

Альтернативный сценарий

- Пользователь нажимает на имя другого пользователя — автора отзыва — и попадает на его страницу (Страницу 5)

Страница 5 - Страница пользователя

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь видит страницу профиля пользователя: некоторые его личные данные (ФИО, логин, дату регистрации), количество посещённых дворов и их список

2. Пользователь вводит значения фильтров для полей Адрес и Дата посещения (От и До) или оставляет фильтры пустыми

3. Пользователь скроллит страницу вниз и просматривает список посещённых дворов

Альтернативный сценарий:

- Пользователь кликает на карточку двора из списка посещённых дворов и переходит на Страницу двора (Страницу 4)

Страница 6 - Профиль

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь видит страницу своего профиля: некоторые личные данные (ФИО, логин, дату регистрации), количество посещённых дворов и их список

2. Пользователь вводит значения фильтров для полей Адрес и Дата посещения (От и До) или оставляет фильтры пустыми

3. Пользователь скроллит страницу вниз и просматривает список посещённых дворов

Альтернативный сценарий

2. Пользователь кликает на карточку двора из списка посещённых дворов и переходит на Страницу двора

3. Пользователь нажимает на кнопку "Редактировать" и попадает на страницу редактирования профиля

Страница 7 - Редактирование профиля

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь вводит желаемый никнейм

2. Пользователь вводит свои ФИО

3. Пользователь вставляет ссылку на аватарку

4. Пользователь нажимает "Сохранить" и попадает на страницу своего профиля

Альтернативный сценарий

1. Пользователь вводит желаемый никнейм, но он уже занят
2. Пользователь нажимает "Сохранить"
3. Пользователь видит ошибку, что такой никнейм уже существует

Страница 8 - Статистика

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь переходит на страницу администрирования
2. Пользователь переходит к разделу "Посещение дворов" и видит распределение посещения дворов по датам
3. (Опционально) Пользователь выбирает тип статистики, на которую хочет посмотреть, из выпадающего списка
4. (Опционально) Пользователь вводит числовое значение в поле "Рейтинг"
5. (Опционально) Пользователь вводит адрес или фрагмент адреса в поле "Адрес"
6. (Опционально) Пользователь вводит название или фрагмент названия в поле "Название"
7. (Опционально) Пользователь выбирает диапазон дат в полях "Дата, от" и "Дата, до"
8. График обновляется, пользователь видит распределение выбранной статистики с применёнными фильтрами

Страница 9 - Расширенный поиск дворов

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь переходит на страницу расширенного поиска дворов

2. (Опционально) вводит адрес двора (в т.ч. частично)
3. (Опционально) вводит название двора (в т.ч. частично)
4. (Опционально) вводит рейтинг двора (в т.ч. частично - первые цифры)
5. (Опционально) выбирает прямоугольник на карте, где может лежать двор.

6. Нажимает "Найти"

8. Пользователю показывается список дворов с заданными параметрами

Страница 10 - Расширенный поиск пользователей

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь переходит на страницу расширенного поиска пользователей
 2. (Опционально) вводит имя (в т.ч. частично)
 3. (Опционально) вводит фамилию (в т.ч. частично)
 4. (Опционально) вводит отчество (в т.ч. частично)
 5. (Опционально) вводит никнейм (в т.ч. частично)
 6. (Опционально) вводит количество посещенных дворов
 7. Нажимает "Найти"
8. Пользователю показывается список пользователей с заданными параметрами

Страница 11 - Администрирование

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь переходит на страницу администрирования
2. Пользователь нажимает на кнопку "Импорт"
3. Браузер предлагает пользователю выбрать файл (скорее всего, в json)
4. Пользователь загружает файл

Альтернативный сценарий 1:

1. Пользователь переходит на страницу администрирования
2. Пользователь нажимает на кнопку "Экспорт"

3. Скачивается файл с данными проекта (скорее всего, в json)

Альтернативный сценарий 2:

1. Пользователь переходит на страницу администрирования

2. Пользователь нажимает на кнопку "Отзывы посещения" и переходит на страницу просмотра всех отзывов (посещений) (страница 12)

Альтернативный сценарий 3:

1. Пользователь переходит на страницу администрирования

2. Пользователь нажимает на кнопку "Статистика" и переходит на страницу просмотра статистики (страница 8)

Страница 12 - Расширенный поиск отзывов (посещений)

Действующее лицо: аутентифицированный пользователь

Основной сценарий:

1. Пользователь переходит на страницу расширенного поиска пользователей

2. (Опционально) вводит адрес двора (в т.ч. частично)

3. (Опционально) вводит название двора (в т.ч. частично)

4. (Опционально) вводит рейтинг двора (в т.ч. частично - первые цифры)

5. (Опционально) выбирает прямоугольник на карте, где может лежать двор.

6. (Опционально) вводит имя (в т.ч. частично)

7. (Опционально) вводит фамилию (в т.ч. частично)

8. (Опционально) вводит отчество (в т.ч. частично)

9. (Опционально) вводит никнейм (в т.ч. частично)

10. (Опционально) вводит количество посещенных пользователями дворов

11. (Опционально) вводит дату, от которой смотреть отзывы

12. (Опционально) вводит дату, до которой смотреть отзывы

13. Нажимает "Найти"

14. Пользователю показывается список пользователей с заданными параметрами

2.6. Вывод об операциях

На основе представленных сценариев использования можно заключить, что в данном веб-приложении основная функциональность направлена как на поиск, так и на запись данных:

- Поиск дворов
- Написание отзывов
- Учет пользовательских посещений
- Визуализация статистики

3. МОДЕЛЬ ДАННЫХ

3.1. Нереляционная модель данных

Графическое представление модели на рисунке 14.

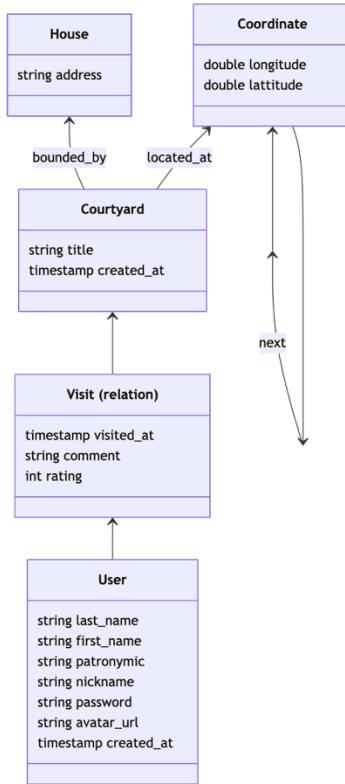


Рисунок 14. Графическое представление модели данных

Описание.

Сущность **Coordinate** представляет собой набор координат для ограничения двора (широта, долгота). Она имеет связь с другой координатой, которая идет следующей, тем самым представляя зацикленный связанный список.

Сущность **House** представляет собой дом для определения адреса двора. По сути адресом двора называем адрес домов, которые его ограничивают.

Сущность **Courtyard** представляет собой двор, название, дату добавления в систему, массив домов и координату, которую считаем точкой отсчета.

Сущность **User** представляет собой пользователя, имеющего ФИО, никнейм, пароль, аватар в виде ссылки и дату регистрации. Он может иметь связь с двором - посещение. В посещении указывается дата и оценка как обязательные поля и optionalный комментарий.

Оценка удельного объема информации, хранимой в модели.

В связах необходимо учитывать также и вес самой связи - два id связываемых объектов. Принимаем вес за 8 байт.

1. visit (между User и Courtyard)

- comment (text, предположим 200 байт в среднем)
- rating (tinyint): 1 байт
- visited_at (timestamp): 8 байт
- Итого для связи visit: $200 + 1 + 8 + 8 = 217$ байт

Для хранения N связей User-Courtyard необходимо $217N$ байт

2. bounded_by (между Courtyard и House)

- Итого для связи bounded_by: 8 байт

Для хранения N связей Courtyard-House необходимо $8N$ байт

2. located_at (между Courtyard и Coordinate)

- Итого для связи located_at: 8 байт

Для хранения N связей Courtyard-Coordinate необходимо $8N$ байт

3. next (между Coordinates)

- Итого для связи next: 8 байт

Для хранения N связей Coordinates-Coordinates необходимо $8N$ байт

Запросы к модели, с помощью которых реализуются сценарии использования.

Добавление пользователя.

```
CREATE (u:User {  
    last_name: 'Фамилия',  
    first_name: 'Имя',  
    patronymic: 'Отчество',  
    nickname: 'Никнейм',  
    password: 'Пароль',  
    avatar_url: 'Ссылка на аватар',  
    created_at: timestamp()  
})  
RETURN u
```

Создание двора.

```
CREATE (c:Courtyard {title: $title, created_at: timestamp() })  
WITH c
```

```

UNWIND $houses AS address
MERGE (h:House {address: address})
MERGE (h)-[:BOUNDS]->(c)
RETURN elementId(c) AS id, c.title AS title, c.created_at AS created_at,
       collect(h.address) AS houses

```

Получение списка пользователей.

```

MATCH (u:User)
OPTIONAL MATCH (u)-[v:VISITED]->(c:Courtyard)
WITH u, collect(DISTINCT c) AS courtyards
WITH u, size(courtyards) AS visited_courtyards
      RETURN elementId(u) AS id, u.nickname AS nickname, u.first_name AS first_name,
u.last_name AS last_name,
      u.patronymic AS patronymic, u.avatar_url AS avatar_url, u.created_at AS
created_at,
      visited_courtyards
SKIP $skip LIMIT $limit

```

Получение списка отзывов.

```

MATCH (u:User)-[v:VISITED]->(c:Courtyard)
MATCH (c)<-[:BOUNDS]-(h:House)
MATCH (c)-[:LOCATED_AT]->(start:Coordinate)
MATCH (start)-[:NEXT*0..]->(coords:Coordinate)
WITH c, u, v, h, coords,
      elementId(u) AS user_id, elementId(c) AS courtyard_id
OPTIONAL MATCH (c)<-[v_all:VISITED]-(:User)
WITH c, u, v, h, coords, user_id, courtyard_id,
      avg(v_all.rating) AS average_rating
OPTIONAL MATCH (u)-[:VISITED]->(unique_courtyard:Courtyard)
WITH c, u, v, h, coords, user_id, courtyard_id, average_rating,
      COLLECT(DISTINCT h.address) AS houses,
      COUNT(DISTINCT unique_courtyard) AS visited_courtyards
WITH v, u, user_id, courtyard_id, average_rating, c,
      collect(DISTINCT h.address) AS houses
WITH v,
{
    id: user_id,
    nickname: u.nickname,
    first_name: u.first_name,
    last_name: u.last_name,
    patronymic: u.patronymic,
    avatar_url: u.avatar_url,
    created_at: u.created_at
} AS user,
{
    id: courtyard_id,
    title: c.title,
}

```

```

        houses: houses,
        average_rating: average_rating,
        created_at: c.created_at
    } AS courtyard
RETURN courtyard, user,
    v.visited_at AS visited_at,
    v.comment AS comment,
    v.rating AS rating
ORDER BY courtyard.id, visited_at
SKIP $skip LIMIT $limit

```

Получение статистики.

```

MATCH (c:Courtyard)<-[v:VISITED]-(:User)
MATCH (c)<-[:BOUNDS]-(h:House)
MATCH (c)-[:LOCATED_AT]->(start:Coordinate)
MATCH (start)-[:NEXT*0..]->(coords:Coordinate)
WITH v.visited_at AS visit_day,
    avg(v.rating) AS average_rating,
    COUNT(DISTINCT v) AS visit_count,
    COUNT(DISTINCT CASE WHEN v.comment IS NOT NULL AND v.comment <> "" THEN v END)
AS review_count
RETURN visit_day,
    average_rating,
    visit_count,
    review_count
ORDER BY visit_day ASC

```

Посещение двора.

```

MATCH (u:User) WHERE elementId(u) = $user_id
MATCH (c:Courtyard) WHERE elementId(c) = $courtyard_id
CREATE (u)-[:VISITED {visited_at: $visited_at, comment: $comment, rating: $rating}]->(c)

```

Обновление профиля пользователя.

```

MATCH (u:User)
WHERE elementId(u) = $user_id
SET u.last_name = COALESCE($last_name, u.first_name),
    u.first_name = COALESCE($first_name, u.first_name),
    u.patronymic = COALESCE($patronymic, u.first_name),
    u.nickname = COALESCE($nickname, u.first_name),
    u.avatar_url = COALESCE($avatar_url, u.first_name)
RETURN elementId(u) AS id, u.nickname AS nickname, u.first_name AS first_name,
u.last_name AS last_name,
    u.patronymic AS patronymic, u.avatar_url AS avatar_url, u.created_at AS
created_at

```

Поиск дворов.

```

MATCH (c:Courtyard)<-[:BOUNDS]-(h:House)

```

```

MATCH (c) -[:LOCATED_AT]->(start:Coordinate)
MATCH (start)-[:NEXT*0..]->(coords:Coordinate)
OPTIONAL MATCH (c)<-[v:VISITED]-(u:User)
WITH c, h, coords, coalesce(avg(v.rating), 0) AS average_rating, count(DISTINCT u)
AS visitors_count
RETURN elementId(c) AS id,
       c.title AS title,
       c.created_at AS created_at,
       collect(DISTINCT {address: h.address}) AS houses,
       collect(DISTINCT {longitude: coords.longitude, latitude: coords.latitude}) AS
coordinates,
       visitors_count,
       average_rating
SKIP $skip LIMIT $limit

```

3.2. Реляционная модель данных

Графическое представление модели на рисунке 15.

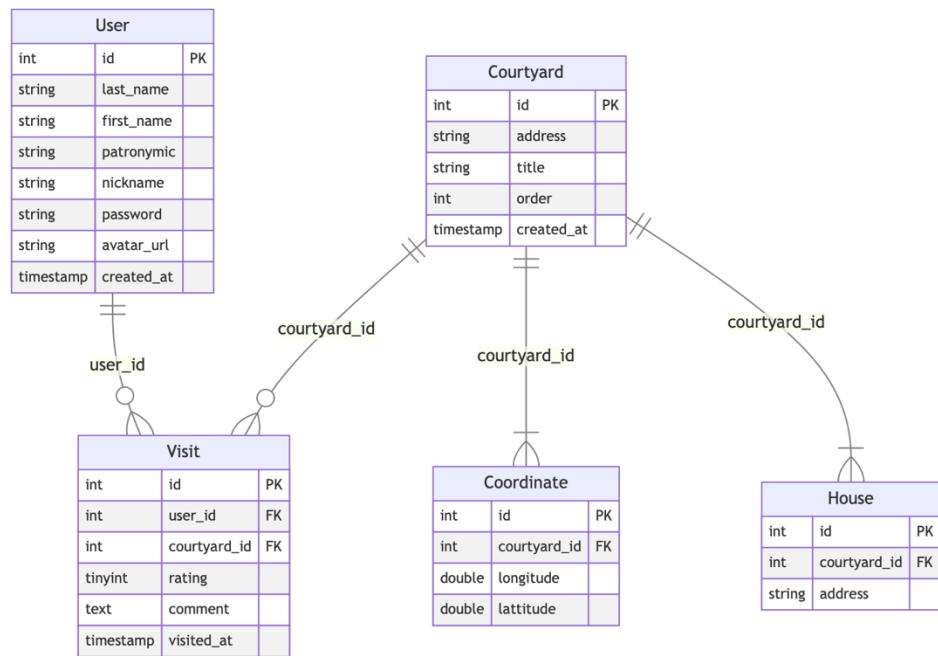


Рисунок 15. Графическое представление модели данных

Описание.

Сущность Coordinate представляет собой координату для ограничения двора (широта, долгота и порядковый номер для обхода). Она относится ко двору по принципу 1:N.

Сущность House представляет собой дом для ограничения двора (адрес).

Она относится ко двору по принципу 1:N.

Сущность Courtyard представляет собой двор, имеющий название, дату добавления в систему. У него имеется несколько координат и домов.

Сущность User представляет собой пользователя, имеющего ФИО, никнейм, пароль, аватар в виде ссылки и дату регистрации. Он может иметь связь с двором M:N - посещение. В посещении указывается дата и оценка как обязательные поля и optionalный комментарий.

Оценка удельного объема информации, хранимой в модели.

1. User

- id (int): 4 байта
- last_name, first_name, patronymic, nickname, password, avatar_url (string, предположим 50 байт в среднем): $6 * 50 = 300$ байт
 - created_at: 8 байтов
 - Итого для User: $4 + 300 + 8 = 312$ байт

Для хранения N пользователей необходимо $312N$ байт

2. Courtyard

- id (int): 4 байта
- address, title (string, предположим по 100 байт в среднем): 200 байт
 - created_at: 8 байтов
 - Итого для Courtyard: $4 + 200 + 8 = 212$ байт

Для хранения N дворов необходимо $212N$ байт

3. Coordinate

- id, order (int): 8 байт
- courtyard_id (int): 4 байта
- longitude, latitude (double): $2 * 8 = 16$ байтов
 - Итого для Coordinate: $4 + 4 + 16 = 24$ байта

Для хранения N координат необходимо $28N$ байт

3. House

- id (int): 4 байта

- courtyard_id (int): 4 байта
- address (string): предположим, 50 байт
- Итого для House: $8 + 50 = 58$ байта

Для хранения N координат необходимо $58N$ байт

4. Visit

- user_id, courtyard_id (int): $2 * 4 = 8$ байтов
- rating (tinyint): 1 байт
- comment (text, предположим 200 байт в среднем)
- visited_at (timestamp): 8 байтов
- Итого для Visit: $8 + 1 + 200 + 8 = 217$ байт

Для хранения N посещений необходимо $217N$ байт

Запросы к модели, с помощью которых реализуются сценарии использования.

Создание пользователя.

```
INSERT INTO User (last_name, first_name, patronymic, nickname, password, avatar_url,
created_at)
VALUES ('Иванов', 'Иван', 'Иванович', 'ivanov_ivan', 'securepassword',
'https://example.com/avatar.jpg', CURRENT_TIMESTAMP);
```

Посещение двора пользователем.

```
INSERT INTO Visit (user_id, courtyard_id, rating, comment, visited_at)
VALUES (
    (SELECT id FROM User WHERE nickname = 'ivanov_ivan'),
    (SELECT id FROM Courtyard WHERE address = 'Улица Примерная, 1'),
    5,
    'Отличный двор!',
    CURRENT_TIMESTAMP
);
```

Создание двора.

```
INSERT INTO Courtyard (title, created_at)
VALUES ('Двор на Примерной', CURRENT_TIMESTAMP);
```

Поиск пользователей по никнейму.

```
SELECT * FROM User WHERE nickname = 'ivanov_ivan';
```

Получение двора по названию.

```
SELECT
    c.title, c.created_at, co.longitude, co.latitude, h.address
FROM
```

```

Courtyard c
    JOIN
Coordinate co ON c.id = co.courtyard_id
    JOIN
House h ON c.id = h.courtyard_id
WHERE
    c.title = 'Двор на примерной'
ORDER BY
    co.order;

```

Получение самых посещаемых дворов.

```

SELECT
    c.id, c.address, c.title, COUNT(v.id) AS visit_count
FROM
    Courtyard c
    JOIN
    Visit v ON c.id = v.courtyard_id
WHERE
    v.created_at >= (CURRENT_TIMESTAMP - INTERVAL '1 month')
GROUP BY
    c.id, c.address, c.title
ORDER BY
    visit_count DESC
LIMIT 10;

```

3.3. Сравнение моделей

Удельный объем информации

Основываясь на расчётах объемов информации для нереляционных и реляционных БД, можно сделать вывод, что больший объём данные занимают в реляционной БД.

Запросы по отдельным сценариям использования

Количество запросов для совершения сценариев использования в нереляционной и реляционной моделях идентично. Во всех случаях, кроме импорта, оно не зависит от размеров БД. Аналогичный результат с количеством коллекций.

Вывод

В результате сравнения двух моделей был получен следующий результат:

- 1.** Удельный объем информации для нереляционной модели оказался меньшим, чем для реляционной, для одних и тех же данных;

2. Количество запросов для нереляционной и реляционной моделей оказалось равным.

Таким образом, нереляционная модель лучше подходит для решения данной задачи, т.к. данные в ней занимают меньше места, а в остальных аспектах модели должны показывать примерно одинаковые результаты.

4. РАЗРАБОТАННОЕ ПРИЛОЖЕНИЕ

1.1. Краткое описание приложения

Код приложения разделен на две части: frontend и backend.

Frontend реализован на основе фреймворка языков HTML, CSS и JS, последний осуществляет взаимодействие с API backend части для основной функциональности проекта. Для отображения карты используется API Яндекс.Карт, графики реализованы с помощью библиотеки Chart.js.

В backend части проекта используется язык Python с фреймворком FastAPI для реализации API. В этой части проекта обеспечивается взаимодействие с БД, обработка запросов от frontend.

1.2. Использованные технологии

Frontend: HTML, CSS, JS, nginx.

Backend: Python.

БД: Neo4j.

Контейнеризация: Docker, docker-compose

5. ВЫВОДЫ

1.1. Достигнутые результаты

В ходе работы было разработано приложение, представляющее собой платформу – каталог дворов-колодцев Санкт-Петербурга, позволяющее пользователям лучше ориентироваться благодаря отзывам, а также вести учет персональных туристических достижений в этой области. Приложение ориентировано на предоставление пользователям удобного инструмента для поиска наиболее интересных дворов по гибким фильтрам.

Приложение также поддерживает функции импорта и экспорта данных, что упрощает управление, и первичную загрузку данных.

1.2. Недостатки и пути для улучшения полученного решения

На данный момент приложение имеет следующие недостатки:

1. Отсутствие авторизации (разграничения прав: администратор, пользователь). Это необходимо для безопасности приложения
2. Отсутствие мобильной версии. Это обязательно для повышения удобства использования. Поиск дворов на улице вероятнее всего будет осуществляться с мобильных телефонов
3. Отсутствие модерации отзывов. Это необходимо для защиты от спама, провокаций, разжигания ненависти и пр.
4. Отсутствие комментариев к отзывам и оценки отзывов со стороны других пользователей. Это позволит другим посетителям отслеживать, какие отзывы полезные, а на какие стоит обращать меньше внимания.

1.3. Будущее развития решения

В будущем в первую очередь необходимо добавление авторизации, позволяющее администраторам устанавливать правила платформы и отслеживать их выполнение – модерировать поведение других пользователей на платформе. После чего обязательно исправление frontend под мобильные версии.

В качестве дополнения на пользовательском опыте положительно скажется внедрение комментариев или оценок к отзывам. Для дальнейшего роста возможно расширение функциональности и на другие туристические места Санкт-Петербурга и не только.

6. ПРИЛОЖЕНИЯ

1.1. Документация по сборке и развертыванию приложения

1. Скачать исходный код проекта или склонировать репозиторий.
2. Перейти в корневую директорию проекта.
3. Собрать контейнеры приложения: docker-compose build --no-cache.
4. Запустить контейнеры: docker-compose up.
5. Открыть приложение в браузере по адресу <http://localhost:8080/>.
6. Нажать на кнопку входа
7. Войти в систему:
 - Логин: ivanov
 - Пароль: password

1.2. Инструкция для пользователя

Регистрация

Создание аккаунта происходит автоматически при вводе нового никнейма, потому для нее достаточно придумать логин и пароль на странице аутентификации.

Поиск дворов

Дворы можно искать как на главной странице – они отображаются в левом списке и на карте, так и на странице расширенного поиска, где можно указать область поиска и более подробные параметры

Написание отзыва

Для написания отзыва необходимо перейти на страницу двора, выбрать оценку, которую можно дополнить необязательным комментарием

Просмотр статистики

Для просмотра статистики требуется перейти на страницу администрирования, откуда перейти на страницу статистики. На ней можно посмотреть основные данные по посещениям дворов за последний месяц с гибкой фильтрацией

Импорт / экспорт

Нажмите на кнопки “Импорт”, “Экспорт” со страницы администратора для импорта или экспорта данных в разделе статистики. Данные сохраняются и принимаются в формате JSON.

2. ЛИТЕРАТУРА

1. Ссылка на GitHub. – [Электронный ресурс]. – URL:
<https://github.com/moevm/nosql2h24-wells>.
2. Документация Python. – [Электронный ресурс]. – URL:
<https://docs.python.org/3/> (дата обращения 22.12.2024).
3. Документация Neo4j. – [Электронный ресурс]. – URL:
<https://neo4j.com/docs/> (дата обращения 22.12.2024).
4. Документация API Яндекс.Карт. – [Электронный ресурс]. – URL:
<https://yandex.ru/dev/jsapi30/doc/ru/?from=mapsapi> (дата обращения 22.12.2024).
5. Документация FastAPI. – [Электронный ресурс]. – URL:
<https://fastapi.tiangolo.com> (дата обращения 22.12.2024).
6. Документация Chart.js. – [Электронный ресурс]. – URL:
<https://www.chartjs.org/docs/latest/> (дата обращения 22.12.2024).