ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НАУЧНО-ОБРАЗОВАТЕЛЬНАЯ КОРПОРАЦИЯ ИТМО»

Факультет Программной Инженерии и Компьютерной Техники



Лабораторная работа по информационным системам №1

Вариант 368949

Выполнил:

Степанов Арсений Алексеевич

Группа:

P3318

Преподаватель:

Николаев Владимир Вячеславович

Задание

Реализовать информационную систему, которая позволяет взаимодействовать с объектами класса Organization, описание которого приведено ниже:

```
public class Organization {
        // Поле не может быть null
2
        // Значение поля должно быть больше О
        // Значение этого поля должно быть уникальным
        // Значение этого поля должно генерироваться автоматически
        private Long id;
        // Поле не может быть null
        // Строка не может быть пустой
        private String name;
10
11
        // Поле не может быть null
12
        private Coordinates coordinates;
13
14
        // Поле не может быть null
15
        // Значение этого поля должно генерироваться автоматически
16
        private java.util.Date creationDate;
17
18
        // Поле может быть null
19
        private Address official Address;
20
21
        // Значение поля должно быть больше О
        private float annualTurnover;
23
24
        // Значение поля должно быть больше О
25
        private int employeesCount;
26
27
        // Поле не может быть null
28
        // Значение поля должно быть больше О
29
        private Float rating;
30
31
        // Поле может быть null
32
        private OrganizationType type;
33
34
        // Поле может быть null
        private Address postalAddress;
36
   }
37
```

```
public class Coordinates {
// Значение поля должно быть больше -469
private float x;

// Значение поля должно быть больше -440
// Поле не может быть null
private Long y;
}
```

```
public class Address {
    // Ποπε με μοχεπ δωπь null
    private String street;

// Дπιμα cmpoκι με δοπκια δωπь δοπωμε 24
    // Ποπε μοχεπ δωπь null
    private String zipCode;

// Ποπε με μοχεπ δωπь null
    private Location town;

}
```

```
public class Location {
   private int x;

private float y;

// Πολε με μοχεπ δωπω null
private Double z;
}
```

```
public enum OrganizationType {
    COMMERCIAL,
    PUBLIC,
    GOVERNMENT,
    TRUST,
    OPEN_JOINT_STOCK_COMPANY;
}
```

Разработанная система должна удовлетворять следующим требованиям:

- 1. Основное назначение информационной системы управление объектами, созданными на основе заданного в варианте класса
- 2. Необходимо, чтобы с помощью системы можно было выполнить следующие операции с объектами: создание нового объекта, получение информации об объекте по ИД, обновление объекта (модификация его атрибутов), удаление объекта. Операции должны осуществляться в отдельных окнах (интерфейсах) приложения. При получении информации об объекте класса должна также выводиться информация о связанных с ним объектах
- 3. При создании объекта класса необходимо дать пользователю возможность связать новый объект с объектами вспомогательных классов, которые могут быть связаны с созданным объектом и уже есть в системе
- 4. Выполнение операций по управлению объектами должно осуществляться на серверной части (не на клиенте), изменения должны синхронизироваться с базой данных

- 5. На главном экране системы должен выводиться список текущих объетов в виде таблицы (каждый атрибут объекта отдельная колонка в таблице). При отображении таблицы должна использоваться пагинация (если все объекты не помещаются на одном экране)
- 6. Нужно обеспечить возможность фильтровать/сортировать строки таблицы, которые показывают объекты (по значениям любой из строковых колонок). Фильтрация элементов должна производиться только по полному совпадению
- 7. Переход к обновлению (модификации) объекта должен быть возможен из таблицы с общим списком объектов и из области с визуализацией объекта (при ее реализации)
- 8. При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться в интерфейсах у других пользователей, авторизованных в системе
- 9. Если при удалении объекта с ним связан другой объект, связанные объекты должны удаляться
- 10. Пользователи должны иметь возможность просмотра всех объектов. Для модификации объекта должно открываться отдельное диалоговое окно. При вводе некорректных значений в поля объекта должны появляться информативные сообщения о соответствующих ошибках

В системе должен быть реализован отдельный пользовательский интерфейс для выполнения специальных операций над объектами:

- 1. Вернуть массив объектов, значение поля пате которых начинается с заданной подстроки
- 2. Вернуть массив объектов, значение поля rating которых меньше заданного
- 3. Вернуть массив объектов, значение поля rating которых больше заданного
- 4. Объединить организации, создав новую и зачислив в неё всех сотрудников двух исходных. Пользователь должен указать исходные организации, имя и адрес новой организации
- 5. Реализовать поглощение одной организацией другой без увольнения сотрудников

Представленные операции должны быть реализованы в рамках компонентов бизнес-логики приложения без прямого использования функций и процедур БД

Особенности хранения объектов, которые должны быть реализованы в системе:

1. Организовать хранение данных об объектах в реляционной СУБД (PostgreSQL). Каждый объект, с которым работает ИС, должен быть сохранен в базе данных

- 2. Все требования к полям класса (указанные в виде комментариев к описанию классов) должны быть выполнены на уровне ORM и БД
- 3. Для генерации поля id использовать средства базы данных
- 4. Для подключения к БД на кафедральном сервере использовать хост pg, имя базы данных studs, имя пользователя/пароль совпадают с таковыми для подключения к серверу

При создании системы нужно учитывать следующие особенности организации взаимодействия с пользователем:

- 1. Система должна реагировать на некорректный пользовательский ввод, ограничивая ввод недопустимых значений и информируя пользователей о причине ошибки
- 2. Переходы между различными логически обособленными частями системы должны осуществляться с помощью меню
- 3. При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться на области у всех других клиентов

При разработке ИС должны учитываться следующие требования:

- 1. В качестве основы для реализации ИС необходимо использовать Spring MVC
- 2. Для создания уровня хранения необходимо использовать Hibernate
- 3. Разные уровни приложения должны быть отделены друг от друга, разные логические части ИС должны находиться в отдельных компонентах

Выполнение

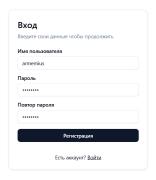
Для реализации фронтенд-составляющей информационной системы был использован фреймворк Svelte Kit и библиотека компонентов ShadCN, система сборки Vite. Для бэкенд-составляющей был использован фреймворк Spring MVC с Hibernate в качестве ORM

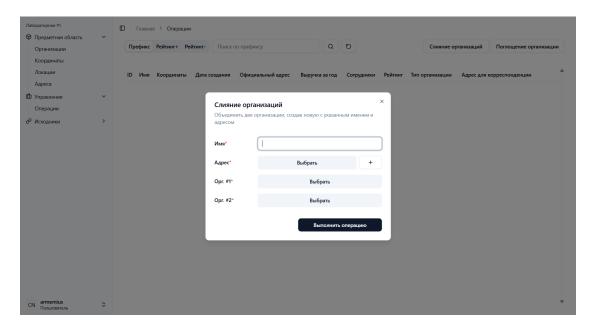
Для локального развёртывания приложения каждый отдельный сервис был обёрнут в Docker-контейнер с последующим использованием в системе развёртывания контейнеров Docker-Compose

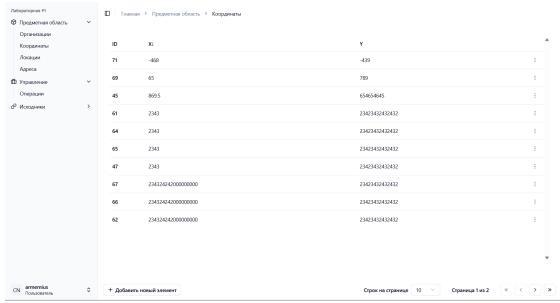
Исходный код содержится в репозитории на GitHub:

- 1. Фронтенд Ссылка на код в репозитории
- 2. Бэкенд Ссылка на код в репозитории
- 3. Диаграмма классов Ссылка на изображение

Скриншоты пользовательского интерфейса







Выводы

В ходе работы была разработана и протестирована информационная система на основе заданного техзадания. Были углублены знания касательно отдельных аспектов работы экосистемы Spring и Svelte Kit. Изучены новые инструменты для работы с фронтендом, бэкендом, базами данных и инфраструктурой