

Федеральное государственное автономное
образовательное учреждение высшего образования
«Научно-образовательная корпорация ИТМО»

**ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И
КОМПЬЮТЕРНОЙ ТЕХНИКИ**

ЛАБОРАТОРНАЯ РАБОТА № 1

по дисциплине

«Информационные системы»

Вариант № 521596

Работу выполнил:
студент группы Р3209
Черноморов К.А
Преподаватель:
Райла Мартин

г. Санкт-Петербург
2024 г.

Текст задания:

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

```
public class Flat {
    private long id; //Значение поля должно быть больше 0, Значение этого поля должно быть уникальным, Значение этого поля должно генерироваться автоматически
    private String name; //Поле не может быть null, Строка не может быть пустой
    private Coordinates coordinates; //Поле не может быть null
    private java.time.LocalDate creationDate; //Поле не может быть null, Значение этого поля должно генерироваться автоматически
    private double area; //Максимальное значение поля: 521, Значение поля должно быть больше 0
    private Double price; //Максимальное значение поля: 648728965, Значение поля должно быть больше 0
    private Boolean balcony; //Поле может быть null
    private float timeToMetroOnFoot; //Значение поля должно быть больше 0
    private Long numberOfRooms; //Максимальное значение поля: 11, Значение поля должно быть больше 0
    private Furnish furnish; //Поле может быть null
    private View view; //Поле может быть null
    private Transport transport; //Поле не может быть null
    private House house; //Поле может быть null
}

public class Coordinates {
    private Float x; //Поле не может быть null
    private Integer y; //Максимальное значение поля: 965, Поле не может быть null
}

public class House {
    private String name; //Поле может быть null
    private long year; //Максимальное значение поля: 681, Значение поля должно быть больше 0
    private long numberOfFloors; //Максимальное значение поля: 80, Значение поля должно быть больше 0
}

public enum Furnish {
    DESIGNER,
    NONE,
    FINE,
    LITTLE;
}

public enum View {
    YARD,
    PARK,
    TERRIBLE;
}

public enum Transport {
    FEW,
    NONE,
    LITTLE,
    NORMAL,
    ENOUGH;
}
```

Разработанная система должна удовлетворять следующим требованиям:

- Основное назначение информационной системы - управление объектами, созданными на основе заданного в варианте класса.
- 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.
- 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.
- Выполнение операций по управлению объектами должно осуществляться на серверной части (не на клиенте), изменения должны синхронизироваться с базой данных.
- На главном экране системы должен выводиться список текущих объектов в виде таблицы (каждый атрибут объекта - отдельная колонка в таблице). При отображении таблицы должна использоваться пагинация (если все объекты не помещаются на одном экране).
- Нужно обеспечить возможность фильтровать/сортировать строки таблицы, которые показывают объекты (по значениям любой из строковых колонок). Фильтрация элементов должна производиться по неполному совпадению.
- Переход к обновлению (модификации) объекта должен быть возможен из таблицы с общим списком объектов и из области с визуализацией объекта (при ее реализации).
- При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться в интерфейсах у других пользователей, авторизованных в системе.
- Если при удалении объекта с ним связан другой объект, связанные объекты должны удаляться.
- Пользователю системы должен быть предоставлен интерфейс для авторизации/регистрации нового пользователя. У каждого пользователя должен быть один пароль. Требования к паролю: пароль должен быть содержать не менее n символов. В системе предполагается использование следующих видов пользователей (ролей): обычные пользователи и администраторы. Если в системе уже создан хотя бы один администратор, зарегистрировать нового администратора можно только при одобрении одним из существующих администраторов (у администратора должен быть реализован интерфейс со

списком заявок и возможностью их одобрения).

- Редактировать и удалять объекты могут только пользователи, которые их создали, и администраторы (администраторы могут редактировать все объекты).
- Зарегистрированные пользователи должны иметь возможность просмотра всех объектов, но модифицировать (обновлять) могут только принадлежащие им (объект принадлежит пользователю, если он его создал). 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:

- Удалить один (любой) объект, значение поля view которого эквивалентно заданному.
- Вернуть массив объектов, значение поля name которых начинается с заданной подстроки.
- Вернуть массив объектов, значение поля furnish которых меньше заданного.
- Найти самую дешёвую квартиру с балконом.
- Вернуть список всех квартир, отсортированный по времени, которое требуется, чтобы добраться от них до метро пешком в порядке возрастания

Представленные операции должны быть реализованы в качестве функций БД, которые необходимо вызывать из уровня бизнес-логики приложения.

Особенности хранения объектов, которые должны быть реализованы в системе:

- Организовать хранение данных об объектах в реляционной СУБД (PostgreSQL). Каждый объект, с которым работает ИС, должен быть сохранен в базе данных.
- Все требования к полям класса (указанные в виде комментариев к описанию классов) должны быть выполнены на уровне ORM и БД.
- Для генерации поля id использовать средства базы данных.
- Пароли при хранении хэшировать алгоритмом SHA-384.
- При хранении объектов сохранять информацию о пользователе, который создал этот объект, а также фиксировать даты и пользователей, которые обновляли и изменяли объекты. Для хранения информации о пользователях и об изменениях объектов нужно продумать и реализовать соответствующие таблицы.
- Таблицы БД, не отображающие заданные классы объектов, должны содержать необходимые связи с другими таблицами и соответствовать ЗНФ.
- Для подключения к БД на кафедральном сервере использовать хост pg, имя базы данных - studs, имя пользователя/пароль совпадают с таковыми для подключения к серверу.

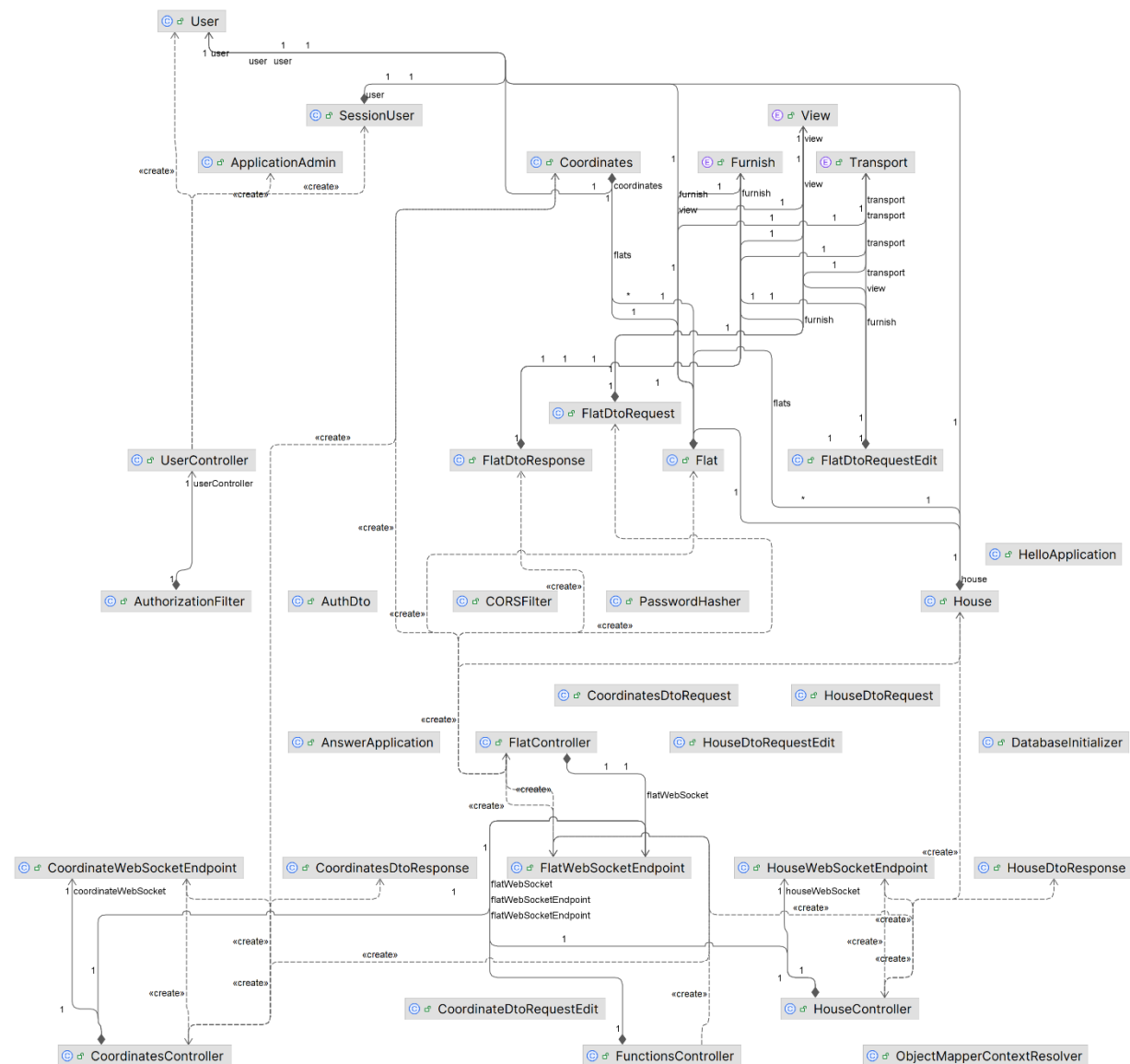
При создании системы нужно учитывать следующие особенности организации взаимодействия с пользователем:

- Система должна реагировать на некорректный пользовательский ввод, ограничивая ввод недопустимых значений и информируя пользователей о причине ошибки.
- Переходы между различными логически обособленными частями системы должны осуществляться с помощью меню.
- Во всех интерфейсах системы должно быть реализовано отображение информации о текущем пользователе (кто авторизован) и предоставляться возможность изменить текущего пользователя.
- [Опциональное задание - +2 балл] В отдельном окне ИС должна осуществляться визуализация объектов коллекции. При визуализации использовать данные о координатах и размерах объекта. Объекты от разных пользователей должны быть нарисованы разными цветами. При нажатии на объект должна выводиться информация об этом объекте.
- При добавлении/удалении/изменении объекта, он должен автоматически появиться/исчезнуть/измениться на области у всех других клиентов.

При разработке ИС должны учитываться следующие требования:

- В качестве основы для реализации ИС необходимо использовать Java EE, Managed Beans.
- Для создания уровня хранения необходимо использовать JPA + EclipseLink.
- Разные уровни приложения должны быть отделены друг от друга, разные логические части ИС должны находиться в отдельных компонентах.

Диаграмма классов



Исходный код: <https://github.com/Gallade901/is-lab-1>

Вывод: Я разработал web приложение и изучил новые технологии такие как WebSocket, Docker.