Постановка задачи:

Необходимо реализовать приложение для организации совещаний или собраний в организации. В организации есть несколько офисов, в каждом офисе несколько переговорных. Сотрудники могут посещать собрания в разных офисах.

Стек-технологий:

1. UI: Thymeleaf или любые другие технологии.
2. Java + Spring + доп. библиотеки (*lombok, liquibase, swagger*) + Тестирование (*Junit, Mockito*).
3. PostgreSQL; GitHub; Maven.

Требования к системе:

1. Приложение должно работать и взаимодействовать по правилам из предметной области;
2. Необходимо реализовать REST-приложение, в котором реализованы необходимые CRUD-запросы для взаимодействия с базой;
3. Необходимо добавить механизм авторизации/аутентификации;

*Вопросы*:

1. Необходимо определить ограничения, связанные с предметной областью (что допустимо, что нет);
2. Придумать модель предметной области – определить сущности и их взаимосвязи;
3. Изучить стек-технологий и связанные с системой понятия.

Предметная область:

Предметная область представляет собой систему, обладающую следующими сущностями:

* Офисы
  + Адрес, телефон, предметы инвентаря на складе
* Кабинеты в офисах, где происходят встречи
  + Вместимость, офис, предметы инвентаря
* Сотрудники
  + Имя, должность, офис, аккаунт в приложении, почта
* Собрания
  + Место, время, тема собрания

Особенности сущностей:

* Офисы могут находиться в разных городах
* Сотрудники могут обладать разными ролями (admin, user)
* Кабинеты могут быть наполнены различными предметами инвентаря

Ограничения сущностей находятся в разделе – База данных.

Приложение:

Описание:

Веб-приложение для планирования и организации локальных собраний в офисах.

Пользовательские особенности:

* Удобный пользовательский интерфейс
* Регистрация и авторизация для пользователей, сотрудник запрашивает создание аккаунта у администратора и получает логин и пароль
* Хранение необходимой информации о пользователях, офисах, собраниях
* Планирование собраний – создание, изменение, отмена
* Отправка уведомлений пользователям

Внутренняя структура:

* Бэкенд приложения: PostgreSQL, Liquibase, Java (Spring, Lombok, Swagger, Junit, Mockito)
* Фронтенд приложения: Angular?

Программные особенности:

* Безопасность – хэширование пароля, spring security
* Тестирование
* Кроссплатформенность

База данных:

Первичный ключ – так отмечаются поля, которые в купе являются первичным ключом для таблицы и должны быть уникальными.

Уникально – так отмечаются поля, которые должны быть уникальными для таблицы, но не являются первичным ключом.

Внешний ключ (главная таблица) – так отмечаются поля, которые являются внешним ключом для таблицы, в скобках указывается главная таблица.

Двумя цветами могут отмечаться поля, которые являются и внешним, и первичным ключом одновременно.

Структура базы данных:

* Таблица офисов
  + ID офиса
  + Город
  + Адрес
  + Телефон
* Таблица кабинетов
  + ID кабинета
  + ID офиса (Таблица офисов)
    - В одном офисе может быть несколько кабинетов для собраний
  + Вместимость
    - Нельзя позвать на собрание больше человек, чем вмещает комната
* Таблица для комнатного инвентаря – таблица для связи кабинетов и предметов инвентаря
  + ID кабинета (Таблица кабинетов)
  + ID предмета инвентаря (Таблица предметов инвентаря)
  + Количество предметов в комнате
* Таблица предметов инвентаря
  + ID предмета
  + ID офиса (Таблица офисов)
    - Если необходимо запросить в кабинет предметы, они должны быть на складе в соответствующем офисе
  + Количество предметов на складе
    - Нельзя запросить больше предметов, чем есть на складе
* Таблица сотрудников
  + Логин
  + Пароль
  + ID сотрудника
  + Имя
  + Должность
  + Email
  + ID офиса (Таблица офисов)
* Таблица ролей сотрудников (Таблица для связи сотрудника и роли)
  + ID сотрудника (Таблица сотрудников)
  + ID роли (Таблица ролей)
* Таблица ролей
  + ID роли
  + Название роли
* Таблица собраний
  + ID собрания
  + Название собрания (Для указания общей темы)
  + ID комнаты (Таблица комнат)
    - В одной комнате в одно и то же время не может проходить два собрания
  + ID офиса (Таблица офисов)
  + Дата и время начала
  + Дата и время окончания
* Таблица приглашений (Таблица для связи сотрудников и собраний)
  + ID сотрудника (Таблица сотрудников)
  + ID собрания (Таблица собраний)

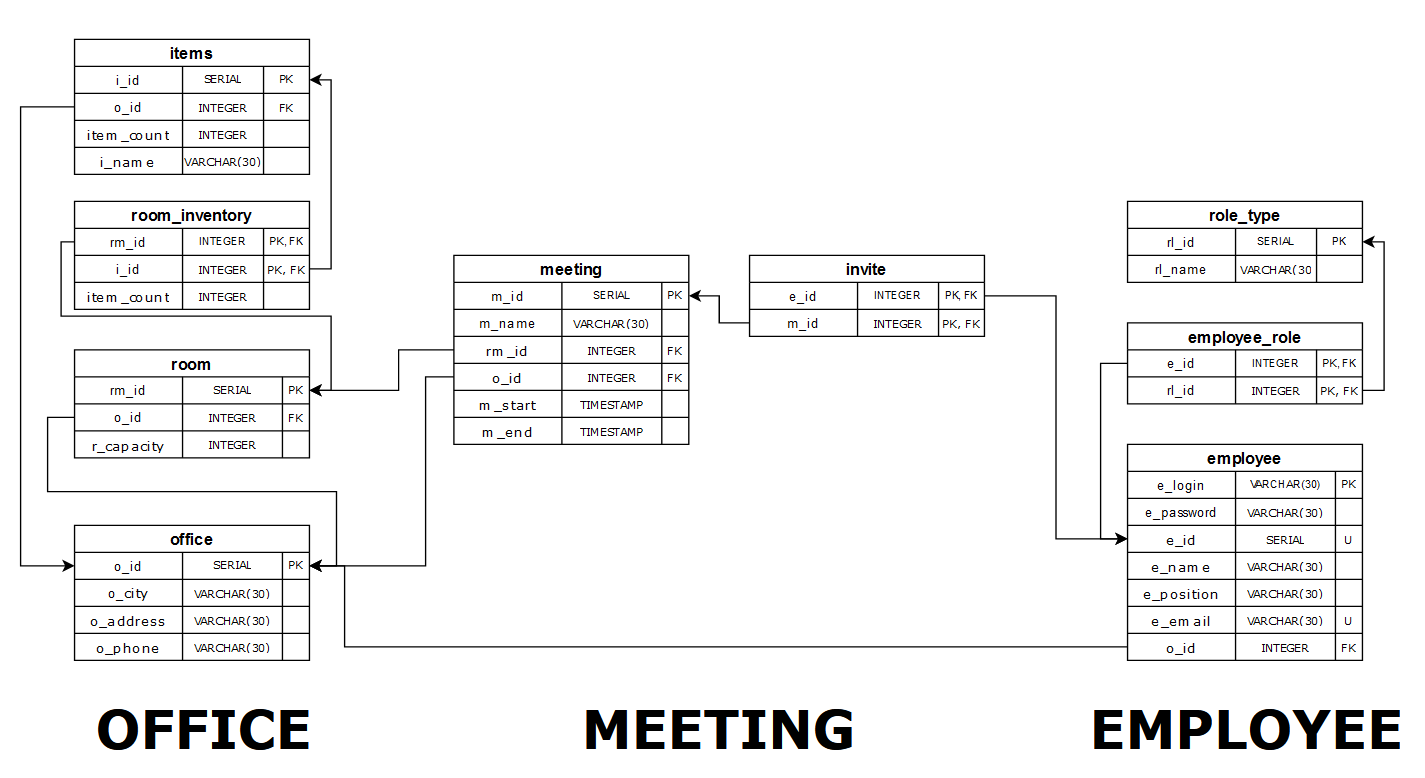


Рисунок 1. UML