Домашнее задание: Расширение функциональности системы управления комментариями

1. **Описание системы:**
    - Название системы: Управление комментариями.
    - Цель системы: Расширение функциональности системы управления комментариями с использованием Spring Boot, JPA, PostgreSQL и миграций Liquibase.
    - Технологии: Spring Boot, Spring Data JPA, Liquibase, PostgreSQL.
2. **Функциональные требования:**
    - Добавление новой сущности "Article":
        - Создать сущность "Article" с полями: id, title, content, creation_time.
        - Создать миграцию для добавления таблицы "article" с соответствующими полями.
    - **Связь "One-to-Many" между "Article" и "Comment":**
        - Установить связь "One-to-Many" между сущностями "Article" и "Comment" (одна статья может иметь много комментариев).
        - Обновить миграции для отражения изменений в связи.
    - **API для работы с новой сущностью "Article":**
        - Создать эндпоинты для выполнения операций CRUD над сущностью "Article".
        - Обеспечить взаимодействие с новой сущностью через контроллер.
    - **Обновление сущности "Comment":**
        - Добавить поле "article_id" в сущность "Comment", которое будет содержать идентификатор связанной статьи.
        - Обновить миграции для отражения изменений в сущности "Comment".

- **Обновленный API для управления комментариями:**
    - Расширить API для управления комментариями так, чтобы можно было указывать к какой статье относится каждый комментарий.

3. **База данных:**
    - Использовать Liquibase для управления миграциями базы данных.
4. **Тестирование:**
    - Провести тестирование вручную с использованием Postman для каждого из новых эндпоинтов и проверки связей между "Article" и "Comment".
5. **Дополнительные требования:**
    - Реализация обработки ошибок с возвратом соответствующих HTTP-кодов и сообщений об ошибках в формате JSON.
    - Реализация логирования событий в приложении.
6. **Среда выполнения:**
    - Java 8 и выше.
    - Apache Maven для сборки и управления зависимостями.
    - PostgreSQL для базы данных.
    - Liquibase для управления миграциями.
    - Postman для тестирования API.
7. **План работ:**
    - Добавить сущность "Article" и соответствующие миграции.
    - Установить связь "One-to-Many" между "Article" и "Comment" с обновлением миграций.
    - Создать эндпоинты для управления "Article" через контроллер.
    - Обновить сущность "Comment" с добавлением поля "article_id" и соответствующим образом изменить миграции.
    - Расширить API для управления комментариями с учетом новых изменений.

- Провести тестирование вручную для проверки корректности работы приложения после внесенных изменений.

Проект с занятия:

https://drive.google.com/file/d/1rODTKz0sV2UjsYR5wJlvUCM1vWtg-x8z/view?usp=sharing