

Финальная работа курса «Разработка на Spring Framework»

О проекте

Вам предстоит разработать бэкенд-составляющую сервиса бронирования отелей с возможностью управлять контентом через административную панель CMS.

Главная задача — обеспечить пользователям возможность забронировать понравившийся отель на определённый период.

Помимо этого, нужно сделать поиск отелей по критериям и рейтингам и систему выставления оценок в приложении (от 1 до 5).

Сервис должен позволять администраторам выгружать статистику по работе в формате CSV-файла.

Задание 1. Подготовка окружения

- Создайте проект с необходимыми зависимостями для разработки приложения. Вам понадобится Spring, Spring Web MVC, Spring Data, Mupstrcut.
- Создайте новую базу данных для приложения.
- Опишите файл docker-compose, который будет содержать контейнер с вашим приложением и контейнер с PostgreSQL.

Задание 2. Работа с отелями

- Опишите сущность «Отель». У каждого объекта должно быть название, заголовок объявления, город, в котором расположен отель, адрес, расстояние от центра города, рейтинг (от 1 до 5) и количество оценок, на основе которых был рассчитан рейтинг.
- Создайте в базе данных таблицу для хранения информации об отелях.
- Настройте отображение данных из таблицы на поля в сущности в Java.
- Создайте интерфейс для работы с сущностью «Отель», используя Spring Data JPA. Интерфейс должен предоставлять возможность выполнять CRUD-операции.
- Создайте сервис для работы с отелями. Сервис должен предоставлять возможность выполнять CRUD-операции.
- Опишите объекты передачи данных DTO (Data Transfer Object), которые будут применяться в контроллерах при запросе пользователя и ответах ему.
 - Опишите DTO HTTP-запроса, который будет использоваться в слое контроллеров при запросе на создание или редактирование сущности «Отель». Обратите внимание, что рейтинг отеля и количество поставленных оценок нельзя изменить при помощи HTTP-запросов.
 - Опишите DTO HTTP-ответа, который будет использоваться в слое контроллеров для возвращения сущности «Отель».
 - Опишите DTO HTTP-ответа, который будет использоваться в слое контроллеров для возвращения списка объектов (других DTO) с пагинацией.

- С помощью библиотеки Mapstruct создайте сервисы для маппинга сущностей в DTO и DTO в сущности:
 - из DTO на создание или редактирование в сущность «ОТЕЛЬ»;
 - из сущности «ОТЕЛЬ» в DTO на редактирование сущности;
 - из сущности «ОТЕЛЬ» в DTO, описывающее сущность «ОТЕЛЬ» (в этой DTO должен быть рейтинг отеля и количество поставленных оценок).
- Создайте контроллер выполнения для CRUD-операций над отелями с помощью REST API. Контроллер должен содержать следующие методы:
 - метод поиска по ID конкретного отеля;
 - создание отеля;
 - редактирование отеля;
 - удаление отеля;
 - получение списка всех имеющихся отелей.

Задание 3. Обработка ошибок

- Опишите DTO, который вернётся в случае ошибок в сервисе.
- Опишите advice, который обработает ошибки и вернёт их пользователям DTO с сообщением, а также корректный HTTP-код ответа с указанием:
 - на отсутствие сущности в БД — возвращение кода 404;
 - клиентские ошибки (неправильный ввод, некорректные данные) — возвращение кода 400;
 - необработанные ошибки — возвращение кода 500.

Задание 4. Работа с комнатами

- Опишите сущность «Комната». Для каждого объекта указываются: название, описание, номер, цена, максимальное количество людей, которое можно разместить, и даты, когда комната недоступна. Каждая комната находится в конкретном отеле.
- Создайте в базе данных таблицу для хранения информации о комнатах.
- Настройте отображение данных из таблицы на поля в сущности в Java.
- Создайте интерфейс для работы с сущностью «Комната», используя Spring Data JPA. Интерфейс должен предоставлять возможность выполнять CRUD-операции.
- Создайте сервис для работы с комнатами. Сервис должен предоставлять возможность выполнять CRUD-операции.
- Опишите объекты передачи данных DTO (Data Transfer Object), которые будут применяться в контроллерах при запросе пользователя и ответах ему.
 - Опишите DTO HTTP-запроса, который будет использоваться в слое контроллеров при запросе на создание или редактирование сущности «Комната».
 - Опишите DTO HTTP-ответа, который будет использоваться в слое контроллеров для возвращения сущности «Комната».
- С помощью библиотеки Mapstruct создайте сервисы для маппинга сущностей в DTO и DTO в сущности:
 - из DTO на создание или редактирование в сущность «Комната»;
 - из сущности «Комната» в DTO на редактирование сущности;
 - из сущности «Комната» в DTO, которое описывает сущность «Комната».

- Создайте контроллер выполнения для CRUD-операций над комнатами с помощью REST API. Контроллер должен содержать следующие методы:
 - метод поиска по ID конкретной комнаты;
 - создание комнаты;
 - редактирование комнаты;
 - удаление комнаты.

Задание 5. Работа с пользователями

- Опишите сущность «Пользователь». У каждого пользователя есть уникальное имя, пароль, электронная почта и роль (пользователь или администратор).
- Создайте в базе данных таблицу для хранения информации о пользователях.
- Настройте отображение данных из таблицы на поля в сущности в Java.
- Создайте интерфейс для работы с сущностью «Пользователь», используя Spring Data JPA. Интерфейс должен предоставлять возможность:
 - выполнять CRUD-операции;
 - искать пользователя по его имени;
 - проверять, существует ли пользователь с указанными именем и адресом электронной почты.
- Создайте сервис для работы с пользователями. Сервис должен предоставлять возможность:
 - выполнять CRUD-операции;
 - создавать нового пользователя;
 - искать пользователя по имени.
- Опишите объекты передачи данных DTO (Data Transfer Object), которые будут применяться в контроллерах при запросе пользователя и ответах ему.
 - Опишите DTO HTTP-запроса, который будет использоваться в слое контроллеров при запросе на создание или редактирование сущности «Пользователь».
 - Опишите DTO HTTP-ответа, который будет использоваться в слое контроллеров для возвращения сущности «Пользователь».
- С помощью библиотеки Mapstruct создайте сервисы для маппинга сущностей в DTO и DTO в сущности:
 - из DTO на создание или редактирование в сущность «Пользователь»;
 - из сущности «Пользователь» в DTO на редактирование сущности;
 - из сущности «Пользователь» в DTO, которое описывает сущность «Пользователь».
- Создайте контроллер выполнения для CRUD-операций над пользователями посредством REST API. Контроллер должен содержать метод создания нового пользователя с указанной ролью (роль принимается как один из параметров при создании пользователя). Перед созданием нового пользователя требуется проверка по логину и почте на отсутствие уже зарегистрированного аккаунта с такими же данными.

Задание 6. Работа с бронированием

- Опишите сущность «Бронирование». Она включает даты заезда и выезда, информацию о забронированной комнате и пользователе, который осуществляет бронирование.

- Создайте в базе данных таблицу для хранения информации о бронированиях.
- Настройте отображение данных из таблицы на поля в сущности в Java.
- Создайте интерфейс для работы с сущностью «Бронирование», используя Spring Data JPA. Интерфейс должен предоставлять возможность выполнять CRUD-операции.
- Создайте сервис бронирования. Сервис должен предоставлять возможность:
 - Пользователям бронировать комнаты на указанные даты. Обратите внимание, что бронируется именно комната. Забронировать комнату могут разные пользователи и несколько раз, однако только на свободные даты.
 - Получать список всех оформленных броней.
- Опишите объекты передачи данных DTO (Data Transfer Object), которые будут применяться в контроллерах при запросе пользователя и ответах ему.
 - Опишите DTO HTTP-запроса, который будет использоваться в слое контроллеров при запросе на бронирование.
 - Опишите DTO HTTP-ответа, который будет использоваться в слое контроллеров для возвращения результата операции бронирования.
- Создайте контроллер для бронирования. Контроллер должен содержать следующие методы:
 - Для бронирования комнаты.
 - Получения всех оформленных броней.

Задание 7. Настройка Spring Security в приложении

Опишите все необходимые сервисы и конфигурации для настройки Spring Security в вашем приложении. Для разработки сервиса используйте Basic Auth.

- Опишите объект UserDetails, который будет содержать нужные поля и роли, а также сервис считывания пользователей из базы данных — UserDetailsService.
- Опишите базовую конфигурацию Spring Security. В вашем сервисе доступ без авторизации пользователя будет возможен только для эндпоинта регистрации.
- Измените созданные контроллеры:
 - чтобы регистрация пользователя была доступна без авторизации;
 - создание, редактирование и удаление отелей было доступно только администратору;
 - создание, редактирование и удаление комнат было доступно только администратору;
 - получение списка броней было доступно только администратору;
 - все остальные методы были доступны и пользователю, и администратору, но только при авторизации.

Задание 8. Создание метода изменения рейтинга отеля

Создайте метод в контроллере сущности «Отель», который позволит менять рейтинг отеля (по шкале оценивания от 1 до 5). Контроллер должен быть доступен обычному пользователю и администратору.

Изменение рейтинга отеля будет происходить по следующей формуле:

Если:

- rating — текущий рейтинг;
- newMark — новая оценка;
- totalRating — сумма всех оценок;
- numberOfRating — общее количество оценок.

Тогда:

- Сначала вычисляется сумма всех оценок (totalRating) путём умножения текущего рейтинга (rating) на общее количество оценок (numberOfRating):

$$\text{totalRating} = \text{rating} \times \text{numberOfRating}$$

- Затем вычитается текущий рейтинг (rating) из полученной суммы (totalRating) и добавляется новая оценка (newMark):

$$\text{totalRating} = \text{totalRating} - \text{rating} + \text{newMark}$$

- Далее вычисляется новый средний рейтинг (новое значение rating) путём деления суммы всех оценок (totalRating) на общее количество оценок (numberOfRating) с округлением до одной десятой (один знак после запятой):

$$\text{rating} = \text{totalRating} / \text{numberOfRating}$$

(округление до одного знака после запятой)

- Далее общее количество оценок (numberOfRating) увеличивается на единицу, так как добавилась одна новая оценка.

$$\text{numberOfRating} = \text{numberOfRating} + 1$$

Задание 9. Разработка метода, отдающего постраничную информацию об отелях с учётом фильтрации от пользователя

- Опишите DTO, который будет отдавать список объектов с числом всех имеющихся записей в БД. Этот DTO должен использоваться при работе с пагинацией.
- Добавьте метод в контроллер для сущности «Отель» с возможностью фильтрации по следующим категориям:
 - ID отеля;
 - название отеля;
 - заголовок объявления;
 - город;
 - адрес;
 - расстояние до центра города;

- рейтинг и число рейтингов.
- Добавьте возможность задавать параметры пагинации на запрос по отелям (количество отелей на странице и номер страницы). Для написания фильтрации используйте JPA Specification.

Задание 10. Разработка метода, отдающего постраничную информацию о комнатах с учётом фильтрации от пользователя

- Добавьте метод в контроллер для сущности «Комната» с возможностью фильтрации по следующим категориям:
 - ID комнаты;
 - заголовок;
 - минимальная и максимальная цена;
 - количество гостей в комнате;
 - дата заезда и дата выезда;
 - ID отеля.
- Необходимо добавить возможность задавать параметры пагинации на запрос по комнатам (количество комнат на странице и номер страницы). Для написания фильтрации используйте JPA Specification.
- Фильтрация по датам заезда и выезда должна учитывать оба поля. Если заполнено только одно из полей, фильтрация не срабатывает. При выборе дат заезда и выезда нужно показывать только те номера, которые свободны в этом временном диапазоне.
- Измените метод получения информации об оформленных бронированиях таким образом, чтобы он поддерживал пагинацию.

Пояснение. В отеле есть номер, который забронирован на следующие даты:

- 01.01.23–10.01.23;
- 20.01.23–30.01.23.

Если в полях фильтра были указаны даты с 03.01.23 (заезд) по 11.01.23 (выезд), то номера отеля, забронированные на даты 01.01.23–10.01.23, не попадут в выборку. Если были выбраны даты с 11.01.23 (заезд) по 19.01.23 (выезд), в выборку попадут номера отеля, забронированные на даты:

- 01.01.23–10.01.23;
- 20.01.23–30.01.23.

Задание 11. Создание слоя, отвечающего за сбор статистики в приложении

Создайте слой статистики, который будет работать с Kafka (получение статистической информации) и MongoDB (хранение информации). Напишите сервис внутри этого слоя с методом выгрузки статистических данных в CSV-файл.

Всего в сервисе должно быть два события, которые будут обрабатываться этим слоем, — регистрация пользователя и бронирование комнаты.

- В событии регистрации должны храниться данные о созданном пользователе (его ID).

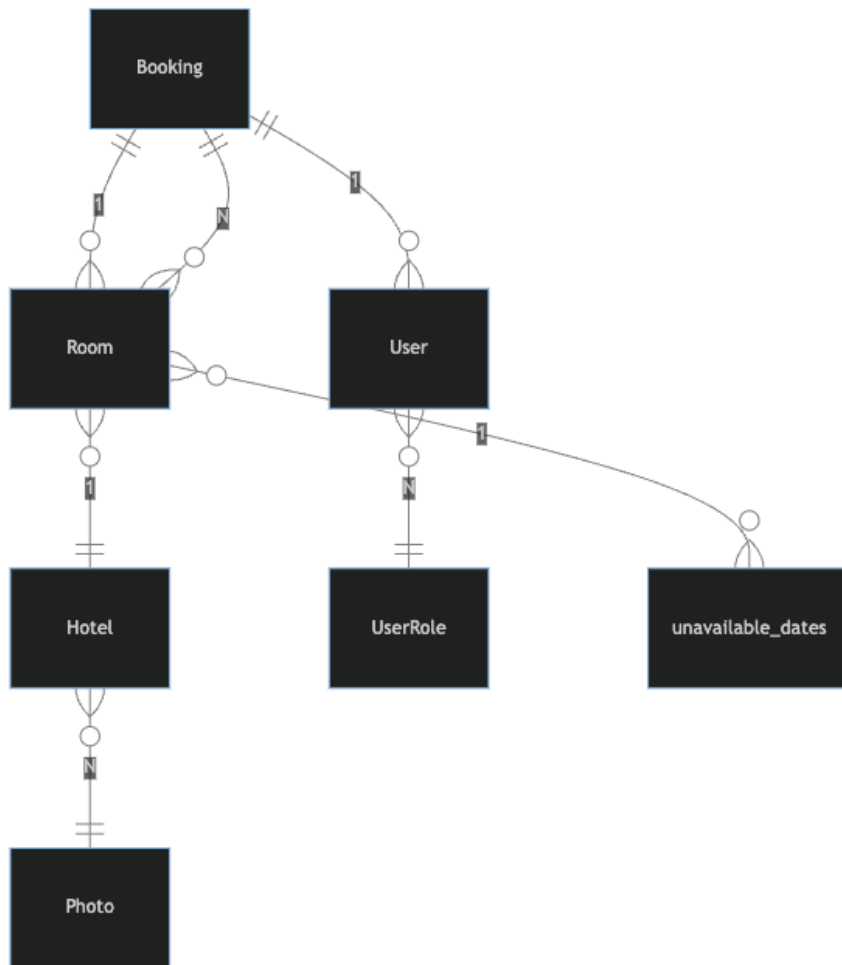
- В событии бронирования комнаты должны храниться данные о пользователе, который оформил бронь (его ID), а также о датах заезда и выезда. Данные по этим событиям нужно хранить в MongoDB в формате JSON.
- Опишите базовые события в сервисе. Создайте модели для отправки и обработки событий.
- Опишите настройки для Kafka. Добавьте сервисы Kafka и ZooKeeper в docker-compose.yml.
- Создайте сущность, репозиторий и сервис для работы со статистикой.
- Создайте сервис для выгрузки всех статистических данных в CSV-файл.
- Добавьте отставку событий при регистрации нового пользователя и бронировании комнаты.
- Добавьте контроллер для работы со статистическими данными. Контроллер должен предоставлять метод для скачивания CSV-файла с накопленными статистическими данными. Метод должен быть доступен только администратору.

Формат сдачи материалов и оценивание

1. Готовую финальную работу и её промежуточные итерации (отдельно выполненные задания) принимаются в формате Git-репозитория с историей коммитов.
2. Промежуточные итерации можно обсуждать с проверяющим куратором, если у вас появятся вопросы.

Рекомендации по выполнению

1. При написании сервиса (основной бизнес-логики) используйте Spring MVC и PostgreSQL.
2. При написании слоя статистики используйте MongoDB.
3. Ниже указаны связи сущностей в БД.



Изображение: предоставлено Павлом Вавиловым

4. При разработке проекта опирайтесь на схему выше.
5. Полностью соблюдайте требования для сервиса, которые подробно описаны в каждом задании.

Как отправить финальную работу на проверку

Отправьте финальную работу и её промежуточные итерации (отдельно выполненные задания) в формате Git-репозитория с историей коммитов через форму на платформе.