

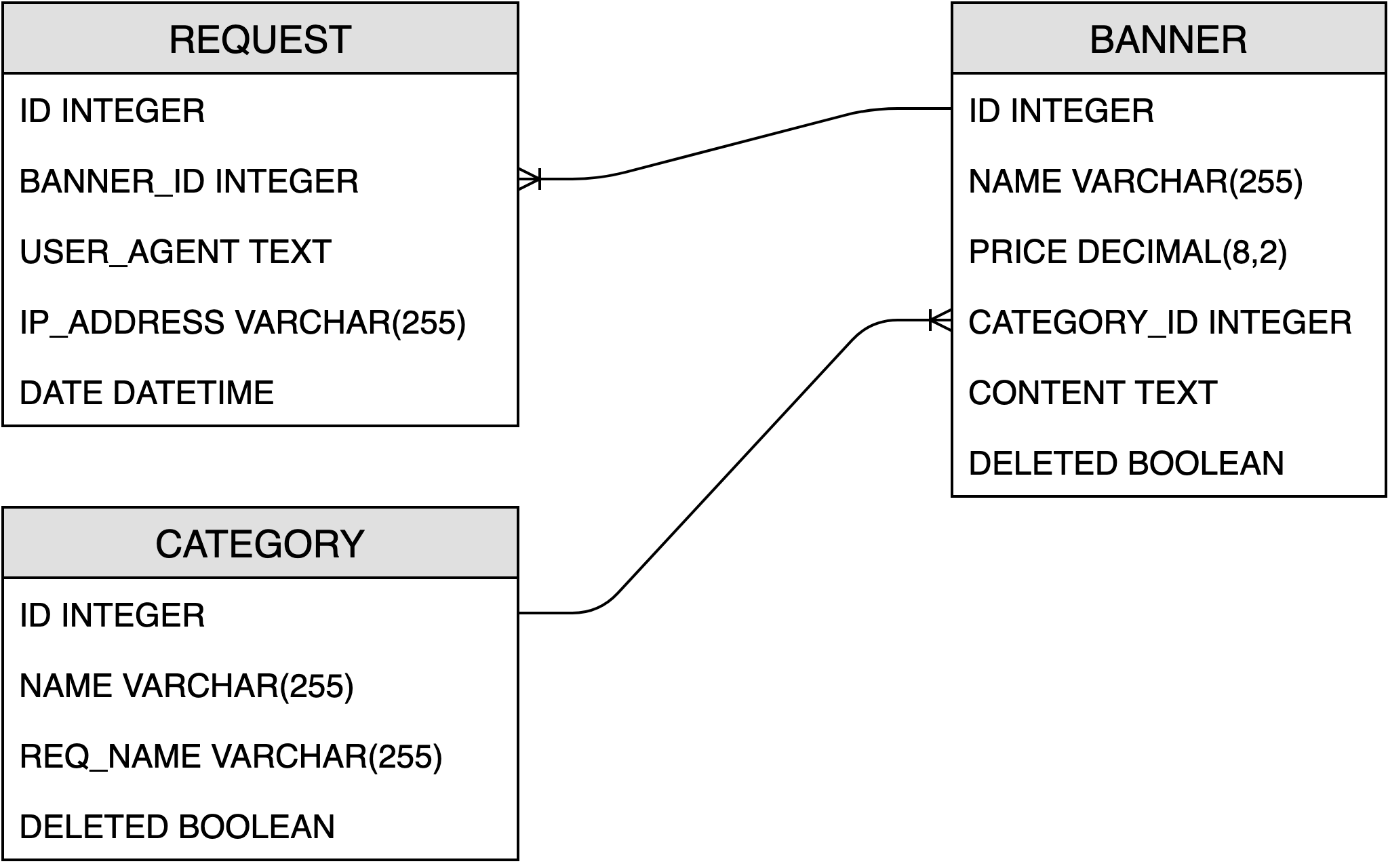
Тестовое задание java developer

## **Общее описание**

Разработка web приложения для управления рекламными баннерами и категориями. Баннеры и категории можно создавать, редактировать и удалять. Выборка баннера происходит через стандартный HTTP запрос, с указанием категории в параметрах запроса.

## **Модель данных**

Всего должно быть минимум 3 таблицы: category, banner, requests. Базовый вариант таблиц и связей:

[](https://www.draw.io/?page-id=pmiWAZ5jIgT0fuCQI1lM&scale=auto#G1SZhNlz6yd64aEm3vC91BJZdbRGUuLhU4)

## **Требования к функционалу**

Удаление баннера или категории должно производиться посредством пометки записи как “deleted”. Все удалённые записи остаются в таблицах, но перестают отображаться на UI. Удалённые баннеры не показываются в рекламных запросах.

Категорию нельзя удалить если с ней связаны не удалённые баннеры. При удалении категории, используемой в каких-либо баннерах, должно показываться сообщение со списком идентификаторов баннеров, в которых используется категория.

Для страниц баннера и категории должен работать поиск по имени, то есть, если в поиске написать “ban”, то на странице должны отображаться только те баннеры и категории, которые содержат этот текст, например, “firstban” или “somebannner”. Поиск должен игнорировать регистр символов.

Все данные для создания и редактирования записей должны валидироваться на сервере в соответствии с ограничениями полей в базе данных. Также все поля в таблицах banner и category должны быть обязательны для заполнения.

Имена баннеров должны быть уникальны также как name и req\_name в категории. Если запрос на создание/редактирование/удаление сущности не прошел валидацию, то в нижней части страницы должно выводиться сообщение об этом.

Для получения текста баннера используется URL вида: [http://host:port/bid?category=<REQ\_NAME](about:blank)> . В ответ приложение возвращает текст баннера из указанной в запросе категории. При наличии нескольких баннеров с такой категорией должен выбираться баннер с самой высокой ценой. Если присутствует несколько баннеров с самой высокой ценой, то среди них выбирается случайный. Если по данному запросу не нашлось ни одного баннера, то сервер должен вернуть HTTP status 204.Информация о каждом запросе должна записываться в таблицу requests.

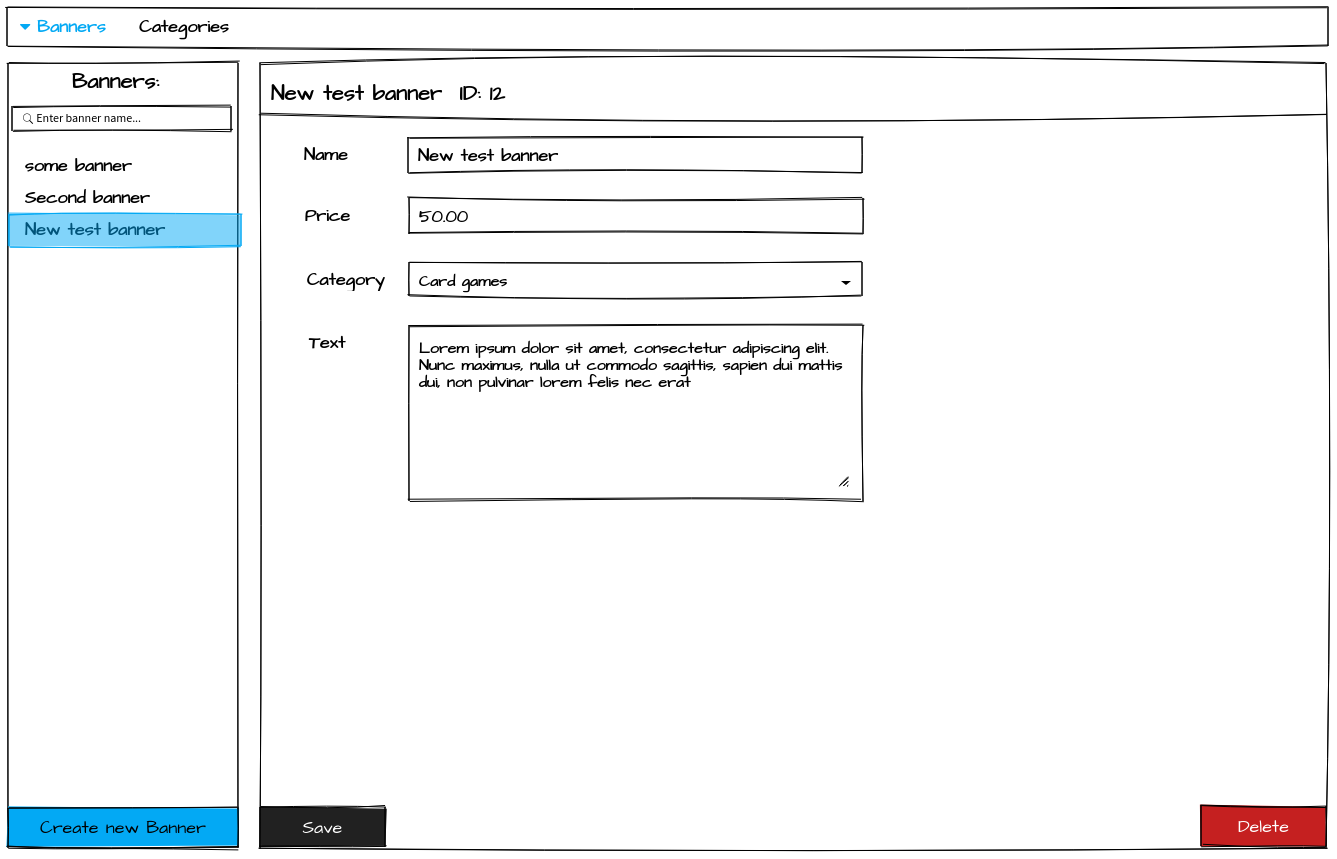
При повторном обращении на протяжении одного дня пользователь с одним и тем же ip-адресом и user-agent-ом не должен увидеть показанную ранее рекламу.

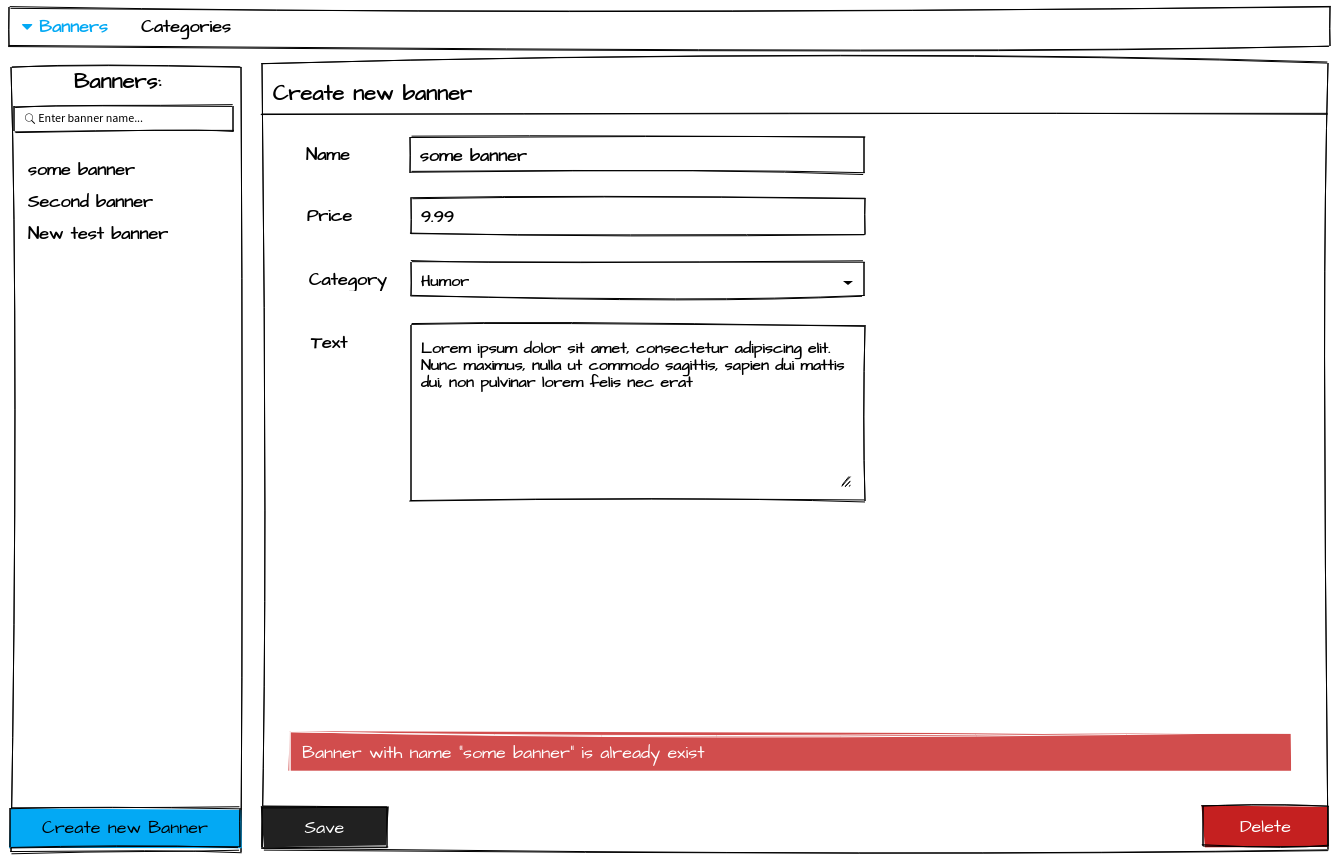
## **Используемые инструменты**

1. Back-end: Java + Spring Boot + Hibernate ORM + Maven/Gradle
2. Front-end: Любой фреймворк (использование Rest API + React.js - большой плюс)
3. Database: MySQL
4. Version control: GIT

После выполнения тестового задания мы бы хотели получить ссылку на репозиторий github/gitlab/bitbucket с выполненным проектом и md файлом с описанием установки и запуска.

## Макет редактирования баннера



Макет создания баннера с сообщением ошибки валидации

Макет редактирования категории