**Техническое задание**

Реализовать веб-сайт для статей на разные темы. Автор статьи может редактировать и удалять статью. Админ может удалять статьи. Все авторизованные пользователи могут просматривать статью, добавлять комментарии, ставить оценку статье.

У авторизованного пользователя есть доступ к его персональной странице, на которой он может просматривать список статей, которые он опубликовал, перейти на страничку добавления статьи. Так же на странице профиля отображаются его основные данные: никнейм, имейл, дата регистрации. Также на странице пользователя есть дополнительные данные о пользователе. Это может быть номер телефона, пол и т.д. Типы таких данных может указывать (создавать), а также удалять администратор. Пользователь имеет возможность редактирования дополнительных данных о себе.

Авторизованные пользователи имеют возможность добавлять, удалять и редактировать комментарии. Администраторы имеют возможность удалять комментарии любых пользователей.

В базе данных хранится количество посещений для каждой статьи, оно увеличивается при посещении статьи как аутентифицированным, так и не аутентифицированным пользователем.

Для каждой статьи необходимо загрузить картинку, которая, в последствии, будет расположена в заголовке к статье. Картинки могут храниться как на облаке, так и в базе данных, в зависимости от выбора администратора.

Авторизованные пользователи имеют возможность оценивать статью при помощи кнопок. Оценка может иметь значение “LIKE” и “DISLIKE”.

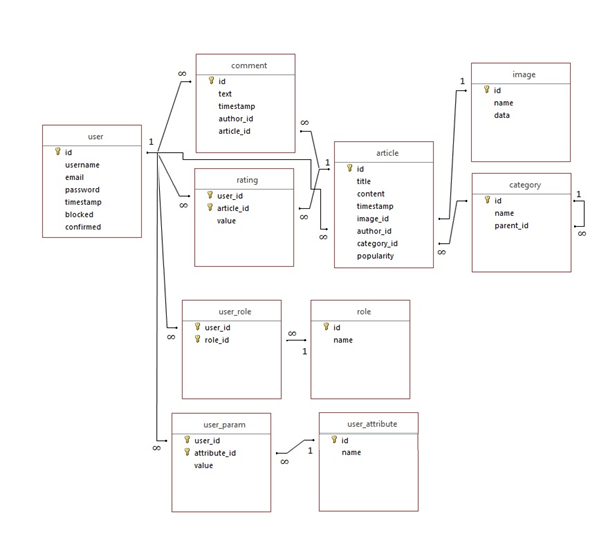
При создании статьи необходимо указать категорию, к которой она относится. Категории могу образовывать древовидную иерархию. Админ может создавать и удалять категории.

Админам доступна страница «Администрирование», где можно удалить, заблокировать, сделать админом и разжаловать любого пользователя. Также админам доступен SQL-терминал.

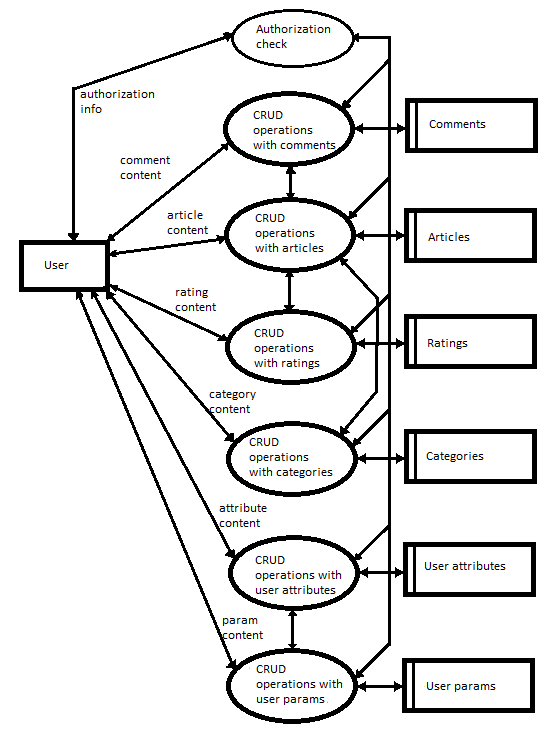
Статью можно скачивать в формате pdf, нажав на соответствующую кнопку.

Ссылка на проект: https://github.com/yatskevichfyodor/articles

**ER – диаграмма**

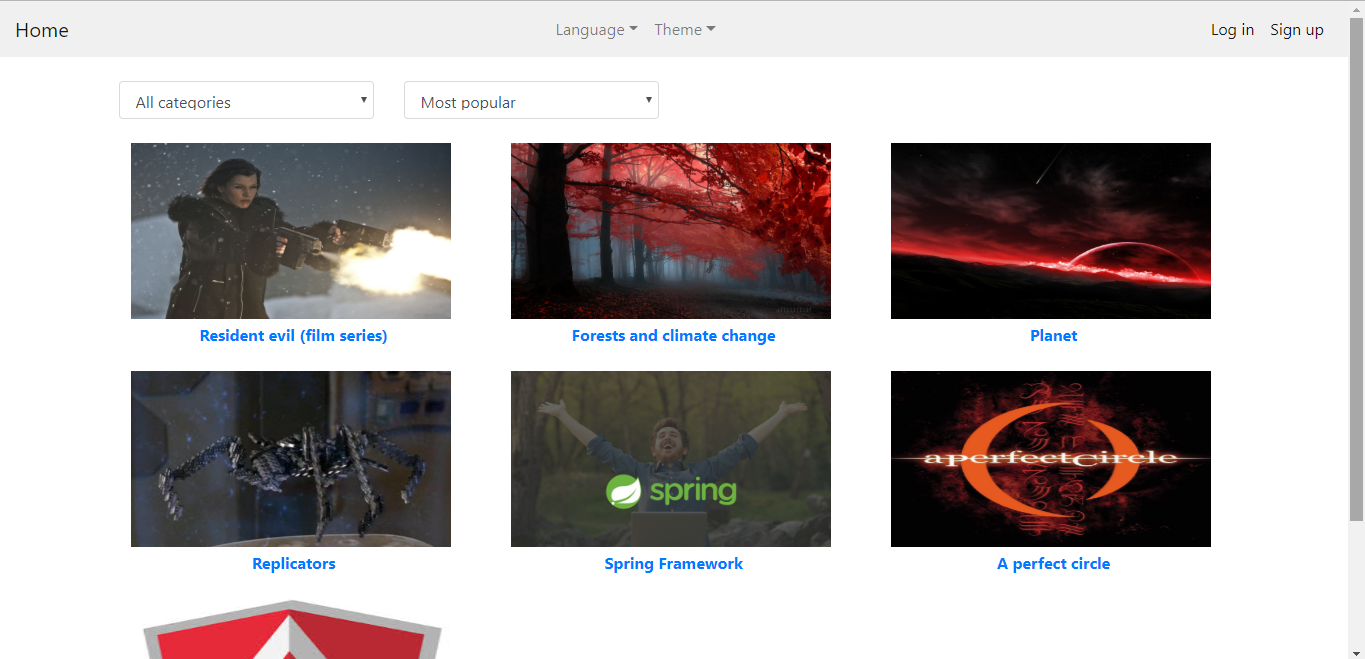


**DF – диаграмма**



**Описание основной бизнес-логики**

**Главная страница**

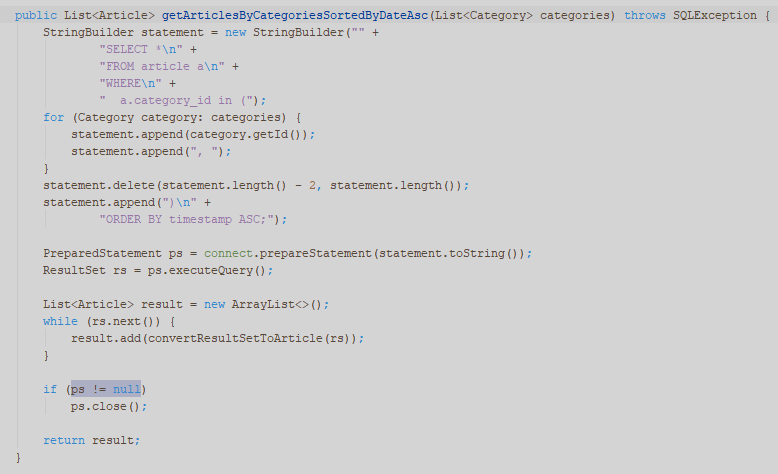


Здесь имеется возможность отфильтровать статьи, а также выбрать порядок вывода статей (По популярности – по убыванию, по дате добавления – по возрастанию и убыванию). При выборе категории выводятся статьи для выбранной категории, а также для всех подкатегорий.

Код, отвечающий за получение статей в базе данных отличается в зависимости от выбора порядка вывода и может быть представлен одним из следующих методов:







Каждый из методов принимает в качестве параметров список категорий, по которым выбираются статьи для вывода. Данные категории получается при помощи вызова двух методов.

Первый получает иерархию использованных категорий, а второй преобразует иерархию в лист.

Для формирования иерархии использованных категорий используется класс UsedCategriesHirarchyBuilder. Формирование происходит при помощи следующего кода:

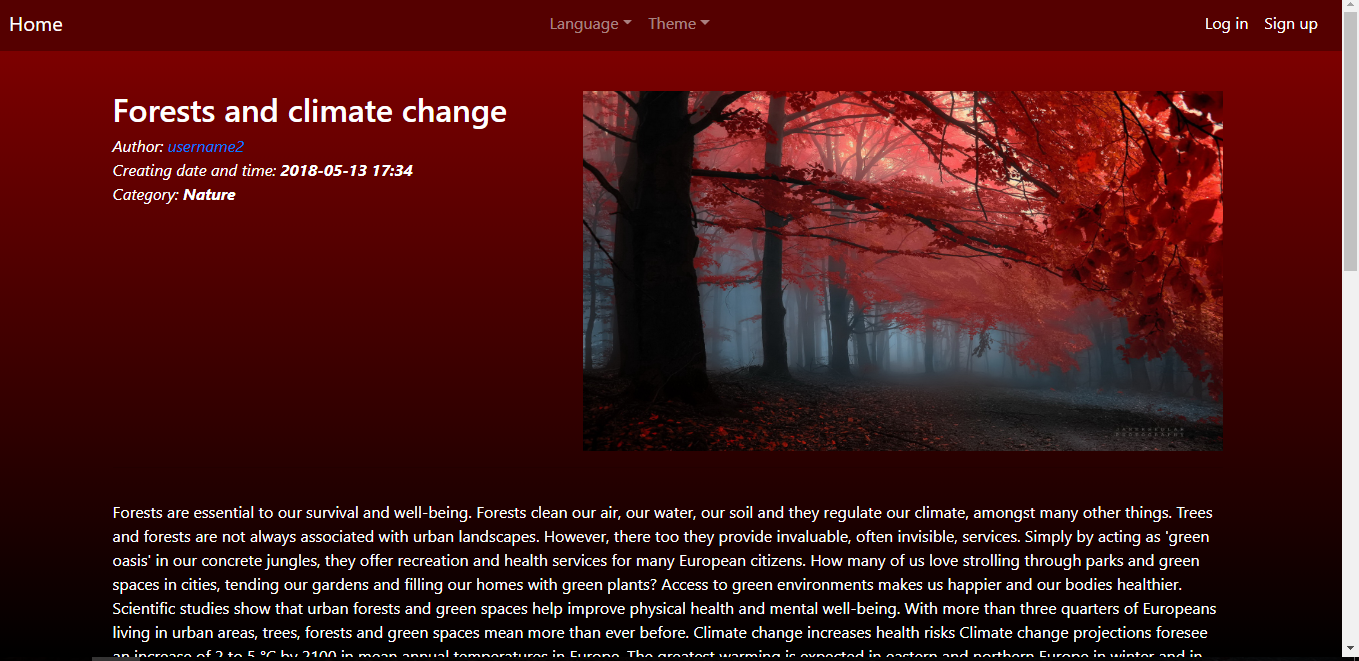


В данном методе, сначала из БД считываются используемые категории, при помощи следующего класса:



**Страница просмотра статьи**

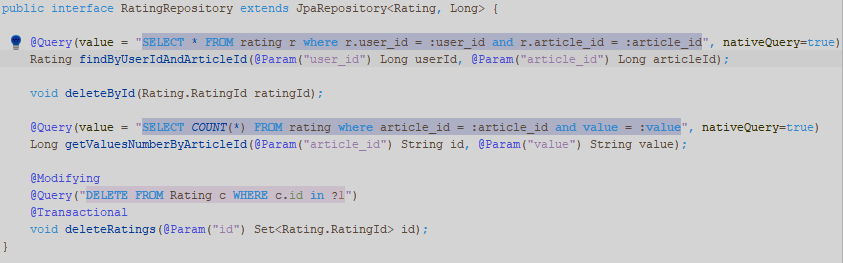
Заголовок статьи:



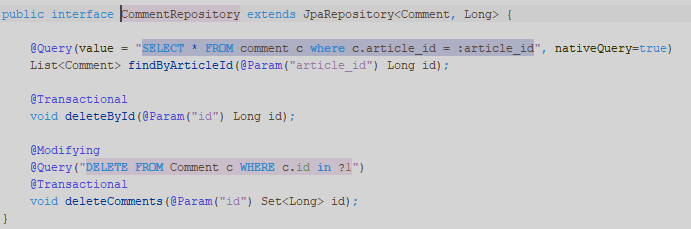
Нижняя часть: рейтиг, комментарии, кнопка получения статьи в pdf формате.



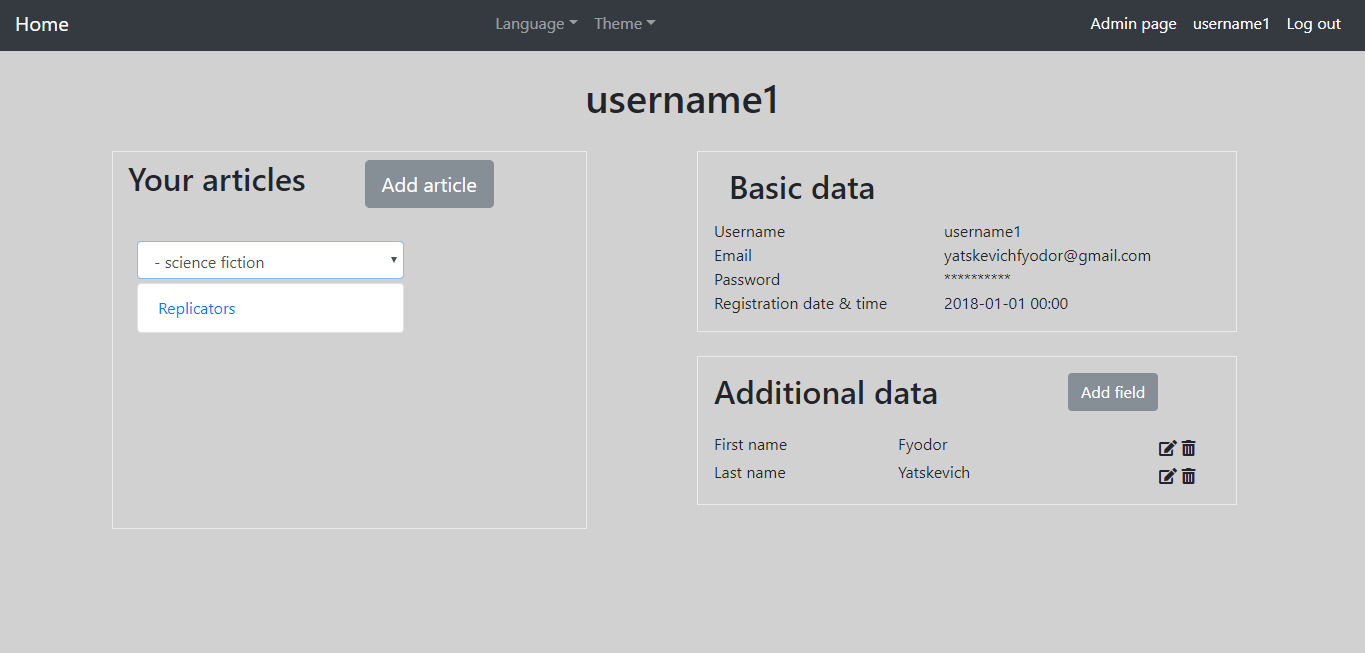
Для получения при загрузке страницы оценки пользователя и суммарной оценки статьи используется следующий класс:



Для получения списка комментариев для статьи, а также других операций с комментариями используется следующий класс:



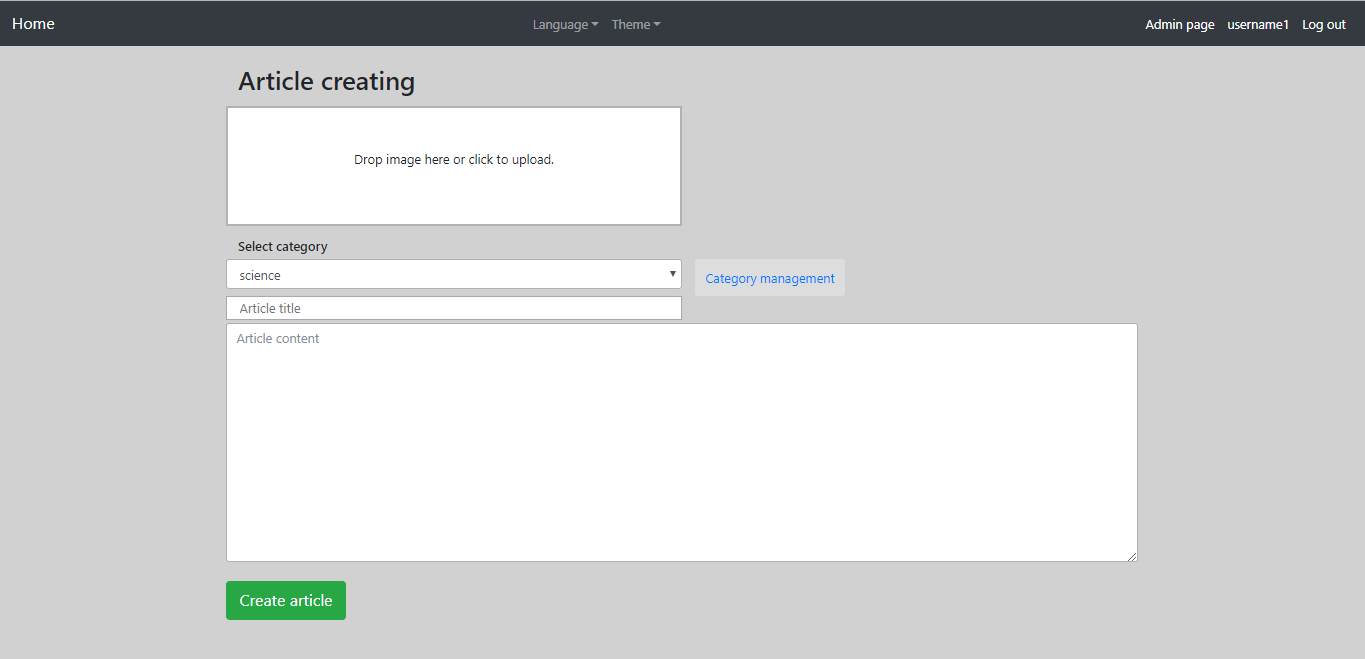
**Страница пользователя**



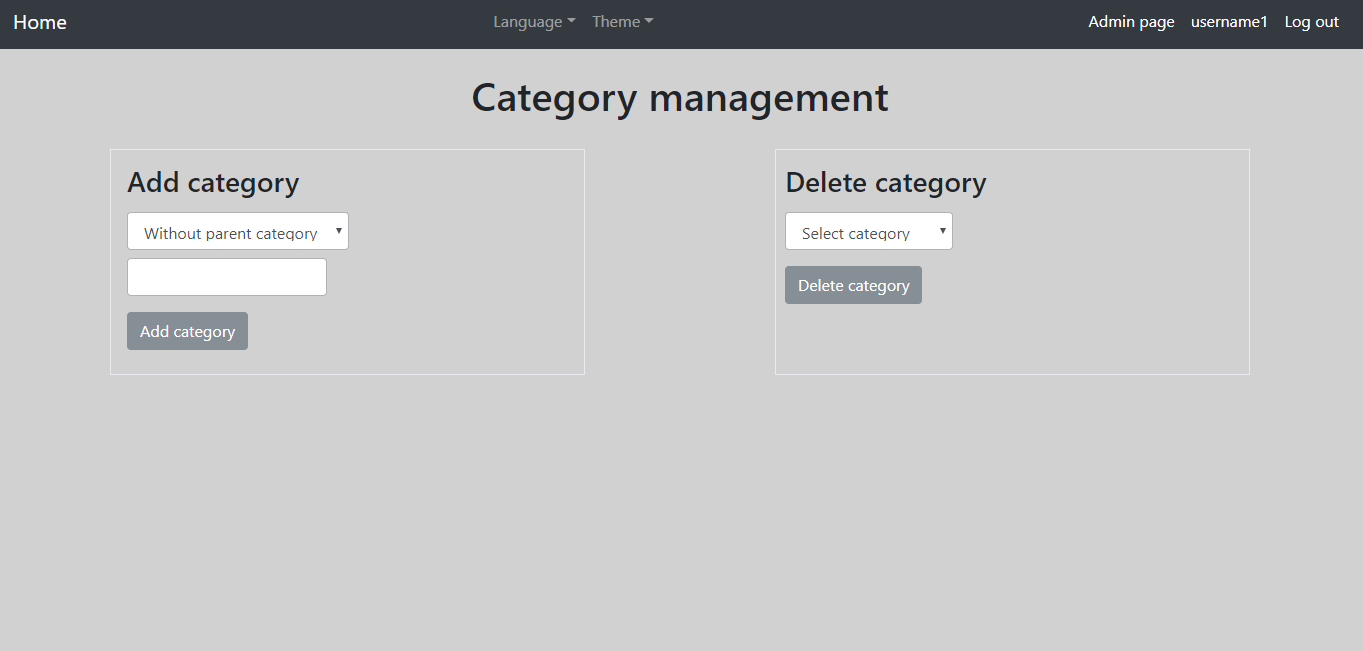
Для формирования списка категорий для элемента dropdown используются классы UserCatgoriesHierarchyBuilder и HierarchicalCategoryHierarchyToListConverter. Принцип формирования иерархии использованных категорий в UserCatgoriesHierarchyBuilder аналогичен методу в UsedCatgoriesHierarchyBuilder:



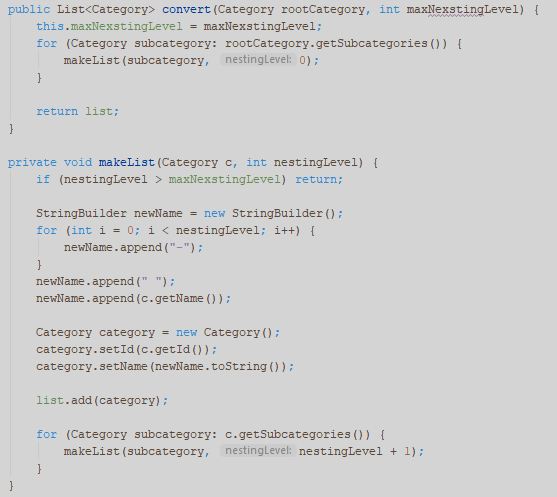
**Страница добавления статьи**



**Страница управления категориями**

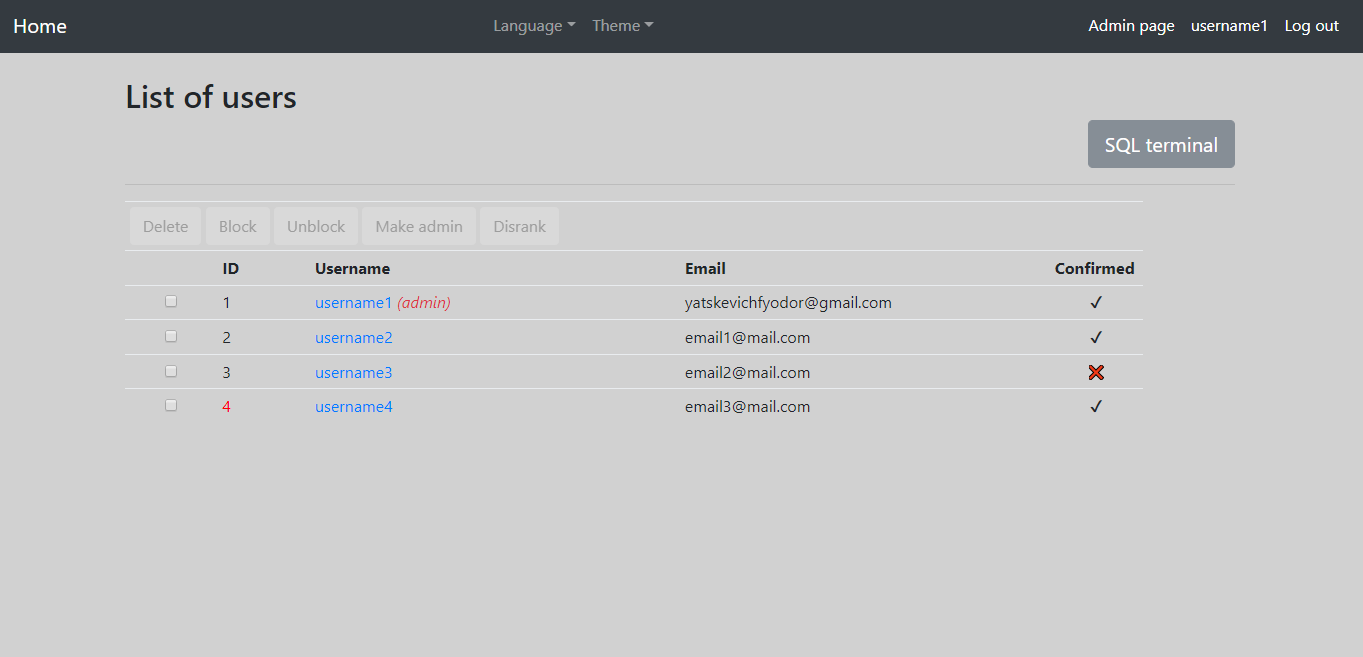


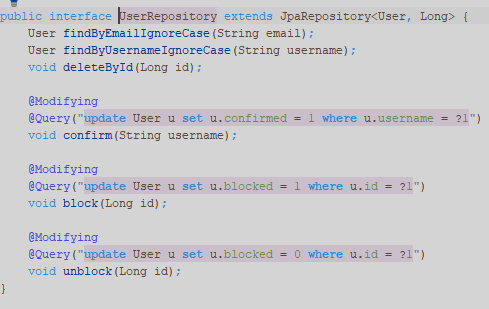
Здесь для формирования категорий для левого элемента dropdown используются классы UsedCategoryHierarchyBuilder и HierarchicalCategoryHierarchyToListConverter. При помощи HierarchicalCategoryHierarchyToListConverter, иерархия не только преобразуется в лист, но еще и обрезается, чтобы максимальный уровень вложенности выводимых категорий не превышал 4. Т.к. максимальный уровень вложенности категории 5, а данный dropdown служит для отображения списка возможных родительских категорий.



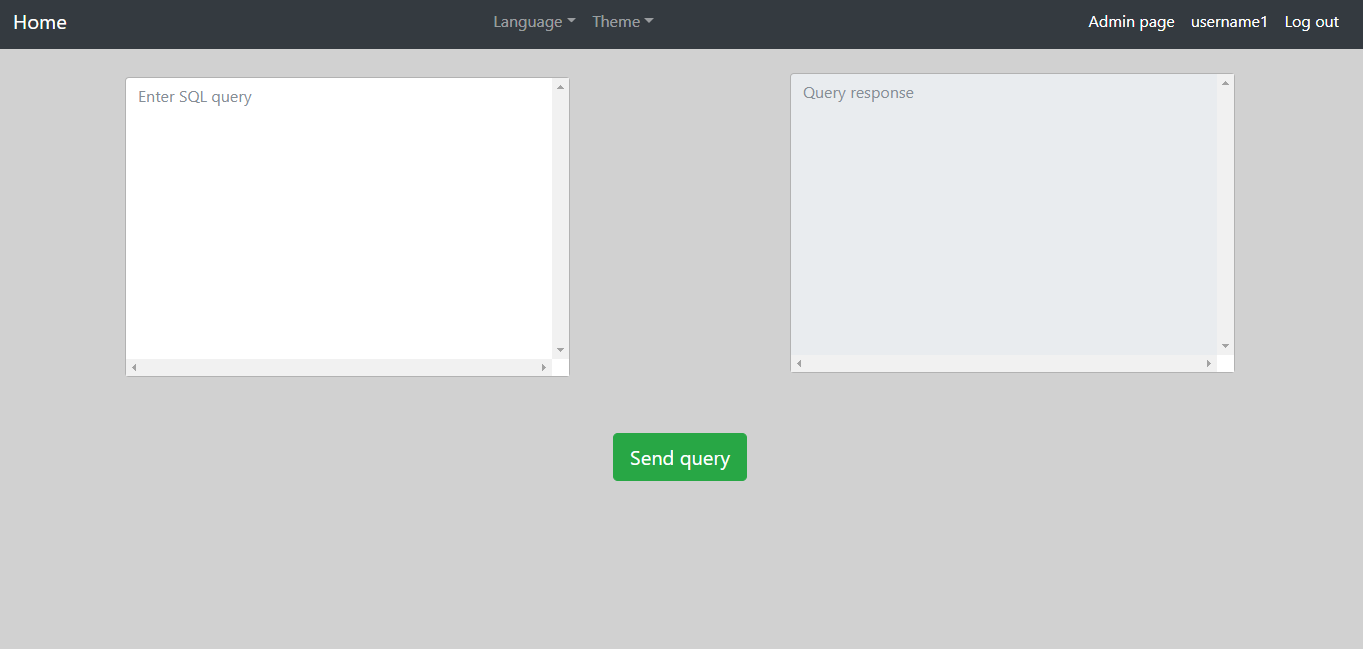
При удалении категории происходит валидация на наличие подкатегорий и статей связанных с удаляемой категорией. Если такие существует, то пользователю сообщается об ошибке, иначе категория удаляется.

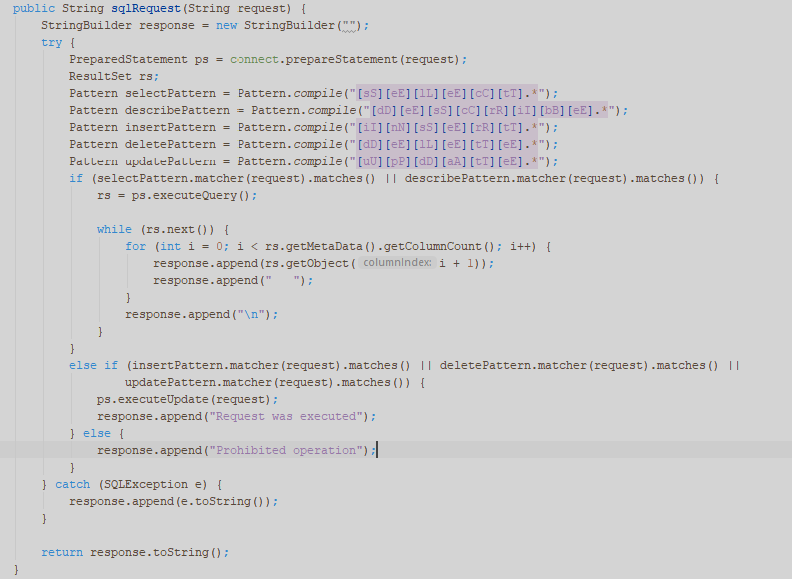
**Страница администрирования**

****



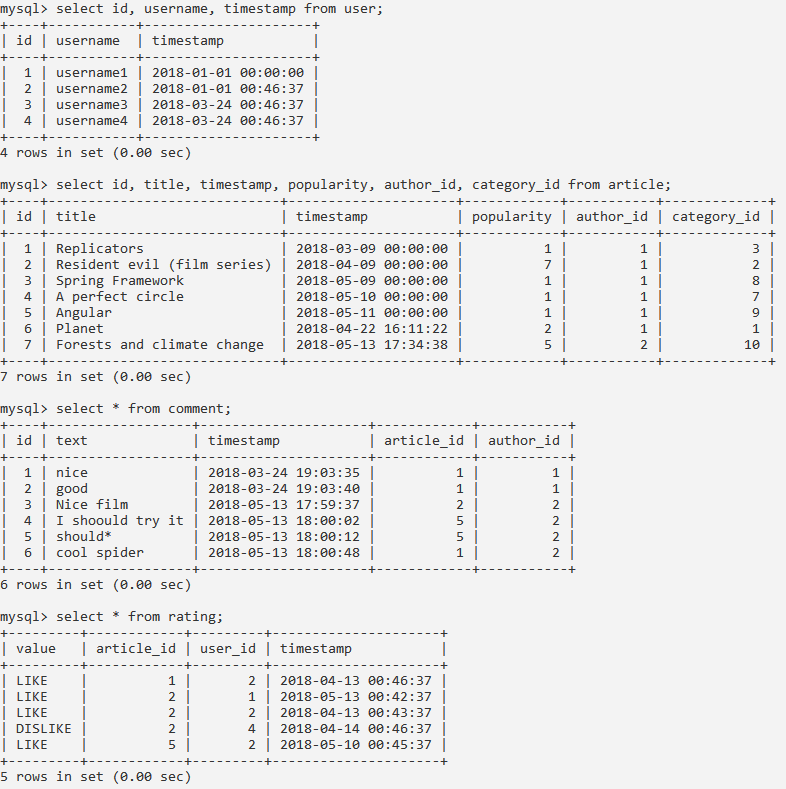
**SQL-терминал**



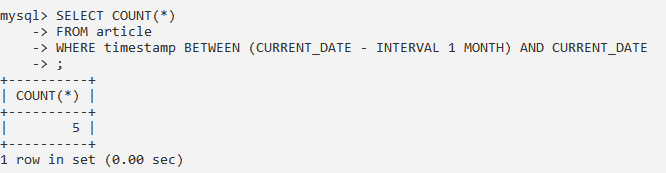


**SQL-запросы**

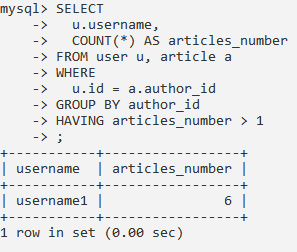
Данные из БД для таблиц user, article, comment, rating:



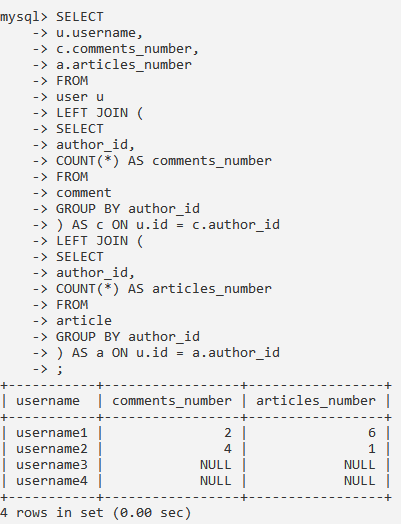
Статистика: вывести количество статей, добавленных за последний месяц.



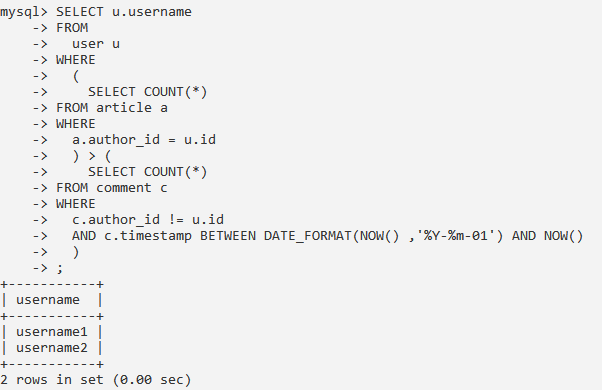
Having: вывести всех пользователей, количество статей которых больше 1.



Множественное сравнение: вывести всех пользователей из таблицы user, а также количество комментариев и количество статей для них с учетом того что их может и не быть.



Множественное сравнение: вывести пользователей, количество статей которых больше количества комментариев, оставленных всеми другими пользователями за текущий месяц



Множественное сравнение: вывести информацию о статьях, рейтинг которых более N (лайков больше 1) был набран в течение месяца со дня публикации.

