

**Министерство науки и высшего образования Российской Федерации**  
федеральное государственное автономное образовательное учреждение высшего  
образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»**

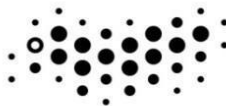
**Отчет**

по проектной работе по теме: “Форум по программированию”  
по дисциплине «Анализ и проектирование на UML»

Автор: Юрпалов С. Н.

Факультет: ИТиП

Группа: М33051



**УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург 2023

## **Словарь предметной области:**

- Форум по программированию - веб-приложение, позволяющее пользователям создавать и участвовать в дискуссионных темах, задавать вопросы и отвечать на них, а также обмениваться знаниями по темам, связанным с программированием.
- Пользователь - лицо, зарегистрировавшее учетную запись на платформе и имеющее доступ к полному спектру функциональных возможностей, таких как создание и управление проектами, участие в обсуждениях, задавание и ответы на вопросы, оценка комментариев и пользователей.
- Модератор - пользователь, которому предоставлены специальные привилегии для модерирования платформы и обеспечения качества контента. Они могут иметь дополнительные функции, такие как возможность редактировать или удалять контент, предупреждать или блокировать пользователей, нарушающих правила сообщества.
- Незарегистрированный пользователь - человек, который не вошел в систему и имеет ограниченный доступ к платформе, например, может просматривать контент, но не может создавать или управлять проектами или участвовать в обсуждениях.
- Проект - набор информации, описывающий проект по программированию, включая его название, описание, участников, используемые инструменты и другие детали, которые автор проекта считает важными.
- Метка проекта - метка или ключевое слово, которое присваивается проекту для его классификации на основе предметной области, технологического стека или других соответствующих атрибутов.
- Тема обсуждения - набор сообщений, относящихся к определенному проекту или теме, где пользователи могут получать ответы на вопросы или обсуждать какие-либо темы.

- Вопрос - сообщение, созданное пользователем, в котором задается конкретный вопрос по программированию, обычно связанный с проектом или темой. Является корнем темы обсуждения.
- Ответ - сообщение, сделанное пользователем в ответ на вопрос, содержащее решение, объяснение или дополнительную информацию.
- Система репутации - система, которая вознаграждает пользователей за положительный вклад, например, за предоставление полезных ответов или отзывов. Эта система может включать такие функции, как повышение или понижение рейтинга ответов или выделение пользователей с высокой репутацией.
- Роль пользователя - уровень полномочий, присвоенный пользователю, который определяет его разрешения и обязанности на платформе. На данный момент включает в себя 2 уровня доступа – обычный пользователь и модератор.
- Система уведомлений - система, которая отправляет уведомления пользователям на основе их предпочтений или активности на платформе. Например, пользователь может получить уведомление, когда кто-то отвечает на один из его вопросов или комментариев, или когда проект, за которым он следит, обновляется.
- Система интеграции Git: система, которая связывает проект на форуме с репозиторием в системе Git для быстрого отображения внесенных изменений (например, при закрытии pull-запроса).

### **Одностраничное описание проекта:**

Проект направлен на разработку платформы форума, которая автоматизирует и облегчает обмен знаниями, совместную работу над проектом, а также задавание и ответы на вопросы, связанные с программированием. Платформа будет реализована в виде веб-приложения с использованием NestJS в качестве бэкенда, HTML + CSS + JS в качестве фронтенда и TypeORM для связи с базой данных PostgreSQL. Платформа позволит пользователям

зарегистрироваться и пройти аутентификацию, чтобы получить доступ к полному спектру функциональных возможностей, таких как создание и управление проектами, участие в обсуждениях, а также задавать и отвечать на вопросы. Пользователи смогут создавать новые проекты, управлять содержанием и разрешениями для своих проектов, а также участвовать в обсуждениях, связанных с конкретными проектами. Платформа также будет включать систему вопросов и ответов, которая свяжет вопросы, связанные с программированием, с конкретными проектами или темами. Платформа будет иметь систему репутации, которая вознаграждает пользователей за положительный вклад, различные роли пользователей для обеспечения хорошего содержания и сообщества, а также теги и фильтрацию для организации и поиска проектов и обсуждений на основе определенных критериев. Пользователи будут получать уведомления, основанные на их предпочтениях или активности на платформе, а платформа будет иметь Git-интеграцию для связи проектов с репозиториями в системе Git для быстрого отображения внесенных изменений. Наконец, платформа будет иметь функцию поиска, которая позволит пользователям легко находить соответствующий контент на основе названия проекта, тега или ключевого слова.

### **Классы и их атрибуты:**

- Участник форума: ID, имя пользователя, пароль, электронная почта, личная информация, репутация.
- Пользователь и модератор: наследуются от Участник форума.
- Проект: ID, название, описание, ссылка на гит-репозиторий.
- Тег проекта: ID, имя, описание.
- Вопрос: ID, содержание, ID автора.
- Ответ: ID, содержание, ID автора.
- Уведомление: ID, ID адресата, сообщение, метка времени.
- Поставщик уведомлений

- Поставщик email уведомлений, поставщик SMS уведомлений: наследуются от Поставщик уведомлений.
- Поисковая система: реализует интерфейс Фильтр проектов.
- Git API: реализует интерфейс Интеграция репозитория и проекта.
- Статус репутации (enum): хороший, плохой.

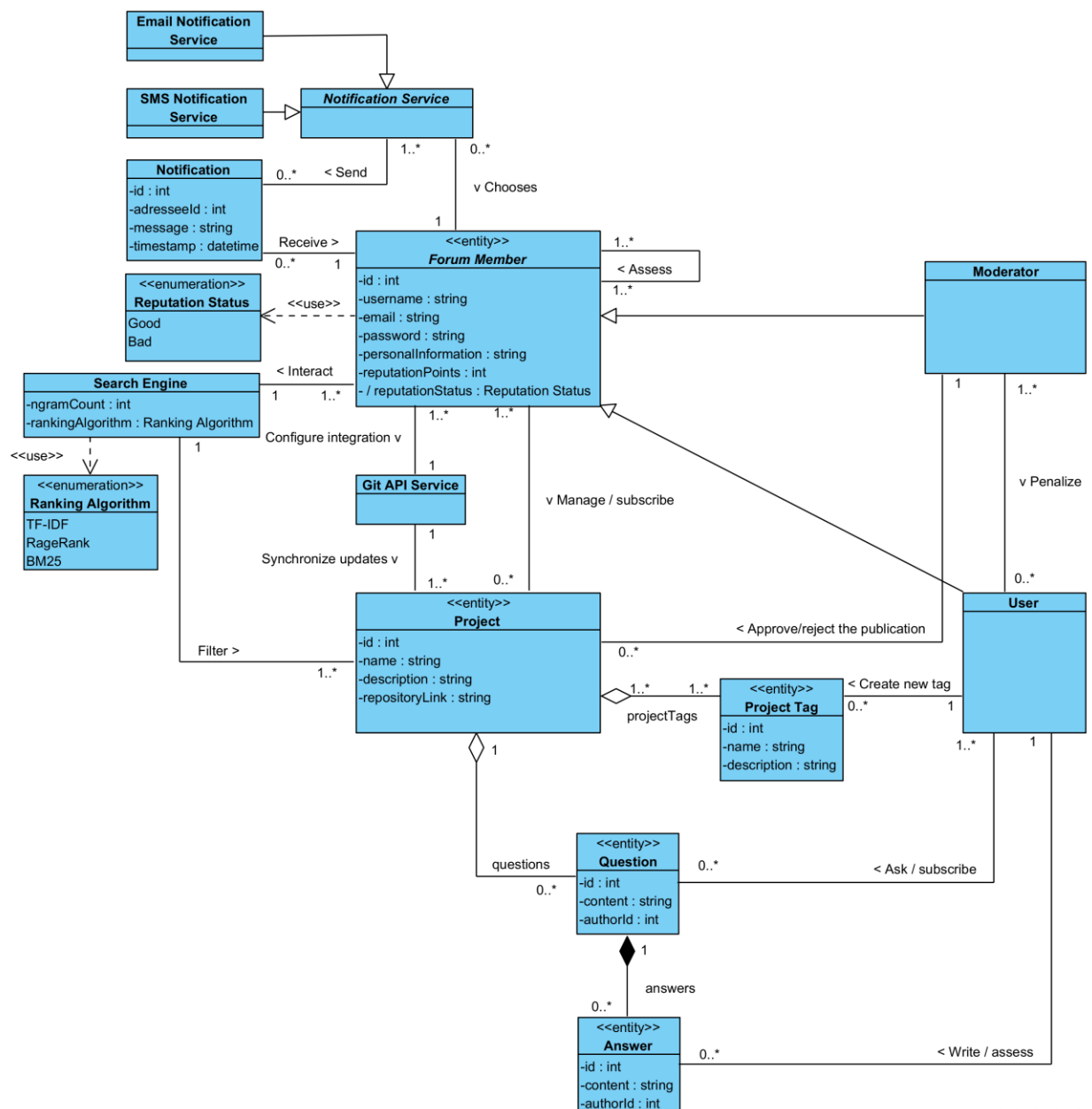
### **Отношения между классами:**

- Пользователь и проект имеют отношения многие-ко-многим через таблицу Проекты пользователей, которая содержит ID автора и ID проекта.
- Пользователь и Проект также имеют отношения многие-ко-многим через таблицу Проекты, на которые подписаны пользователи, которая содержит ID пользователя и ID проекта.
- Проект и Тег проекта имеют отношение многие-ко-многим через таблицу Теги проектов, которая содержит ID проекта и ID тега.
- Проект и Вопрос имеют отношение один-ко-многим, то есть проект может иметь несколько вопросов, но вопрос принадлежит только одному проекту.
- Пользователь и Вопрос имеют связь многие-ко-многим через таблицу Вопросы, на которые подписаны пользователи, которая содержит ID пользователя и ID вопроса.
- Пользователь и Вопрос также имеют связь один-ко-многим, как автора к вопросу. Таблица Вопрос содержит ID автора в качестве внешнего ключа.
- Ответ и Пользователь имеют отношение многие-к-одному, где отклик принадлежит одному пользователю, но у пользователя может быть несколько откликов. Таблица Ответ содержит ID автора в качестве внешнего ключа.
- Ответ и Вопрос имеют связь один ко многим, где вопрос может иметь несколько ответов, но ответ принадлежит только одному вопросу.

Таблица Ответ содержит идентификатор вопроса в качестве внешнего ключа.

- Пользователь и Роль имеют связь многие-к-одному, где пользователь может иметь только одну роль, но роль может быть назначена нескольким пользователям.
- Пользователь и Уведомление имеют связь один-ко-многим, где пользователь может иметь несколько уведомлений, но уведомление должно быть направлено только одному пользователю. Таблица Уведомление содержит ID пользователя в качестве внешнего ключа.
- Поставщик уведомлений и Уведомление имеют связь один-ко-многим, так как поставщик может направить несколько разных уведомлений, но уведомление может быть отправлено лишь одним поставщиком.
- Поставщик уведомлений и пользователь имеют связь один-к-одному, так как пользователь может получать не более одного типа уведомлений.

## Диаграмма классов:



## Компоненты (артифакты) системы и их интерфейсы:

### Компоненты:

- Frontend - HTML, CSS и JS код, который запускается в браузере пользователя. Взаимодействует с Backend API посредством HTTP-запросов и ответов.
- Backend API - сервер NestJS, который предоставляет фронтенду конечные точки для доступа к базе данных и выполнения операций

CRUD над объектами системы. Взаимодействует с базой данных через TypeORM.

- База данных PostgreSQL - база данных, в которой хранятся все данные, связанные с объектами системы. Взаимодействует с API бэкенда посредством SQL-запросов.

#### Интерфейсы:

- Интерфейс Frontend - frontend взаимодействует с API бэкенда посредством HTTPS-запросов и ответов. API предоставляет endpoint для выполнения frontend операций CRUD над объектами системы.
- Интерфейс API backend - backend API взаимодействует с базой данных через SQL-запросы, а с Frontend - через HTTPS-запросы и ответы.
- Интерфейс базы данных - база данных взаимодействует с Backend API посредством SQL-запросов.

#### Физические вычислительные ресурсы:

- Веб-сервер - сервер, способный запускать API backend NestJS и обрабатывать HTTPS-запросы.
- Сервер базы данных - сервер, способный запускать PostgreSQL и хранить базу данных для системы.
- Клиентские устройства - устройства, способные запускать современный веб-браузер, такие как настольные компьютеры, ноутбуки, планшеты и смартфоны.



## Диаграмма размещения:

