**МИНОБРНАУКИ РОССИИ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**“Воронежский государственный университет”**

Факультет компьютерных наук

Кафедра теории обработки и защиты информации Направление 9.03.02 Информационные системы и технологии

Курсовая работа по дисциплине “Технологии программирования”

Разработка приложения «Социальная сеть»

Зав. Кафедрой \_\_\_\_\_\_\_\_\_\_\_\_\_С.Д. Махортов, д.ф.- м.н., профессор \_\_.\_\_.20\_\_

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.С. Мишанин

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_Д.А. Космовский 3 курс, д/о

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_Н.И. Бережнов 3 курс, д/о

**Воронеж 209**

# ВВЕДЕНИЕ

На сегодняшний день существует много социальных сетей. Каждая из них имеет свои сильные и слабые стороны. Где-то хорошо продуман графический интерфейс, но плохо спроектирована сама система. Некоторые сети предоставляют пользователю ограниченные возможности в плане сбора, отбора и оценки информации. Количество информации увеличивается с каждым годом, становится всё труднее находить нужные и полезные данные. Для создателей контента возникают проблемы защиты авторских прав. Люди, не желающие платить за контент или распространяющие его от своего имени, наносят большой урон авторам контента. В связи с этим возникла необходимость в создании новой социальной сети, отвечающей всем этим требованиям.

**ОБЩИЕ ПОЛОЖЕНИЯ**

В работе рассматриваются общие положения в разработке приложения «Социальная сеть». Дополнительную и более подробную информацию можно найти в техническом задании.

Техническое задание составлено по стандарту IEEE STD 830-998.

# ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В документе используются следующие понятия и термины:

**Социальная сеть** – интернет площадка, сайт, который позволяет зарегистрированным пользователям размещать информацию и общаться между собой.

**Пользователь** – физическое лицо, которое использует социальную сеть.

**Администратор** – определённое физическое лицо, которое следит за соблюдением всех правил социальной сети.

**Модератор –** определенное физическое лицо, которое следит за соблюдением правил сообщества.

**Создатель контента** – автор контента для пользователей.

**Пост** – запись, содержащая текстовое или графическое сопровождение.

**Контент –** информационное содержание поста.

**Сообщество** – это сущность, состоящая из группы пользователей, объединенных общими интересами, и постов в этой сущности.

**Сообщение** – структурная единица обмена информацией между несколькими пользователями.

**Комментарий** – запись, содержащая информацию, как правило, имеющую оценочный характер, являющуюся выражением мнения пользователя на определенную тему.

# ПОСТАНОВКА ЗАДАЧИ

**Цель**: разработать клиент-серверное приложение «Социальная сеть», направленную на создание интересного контента и поддержку его создателей.

Данная система направлена на пользование:

-любым пользователем;

-любым создателем контента, публикующим посты, не нарушающих правил социальной сети.

# Требования:

«Социальная сеть» должна позволять пользователям общаться друг с другом, комментировать посты, создавать новые сообщества. Пользователь, создавший сообщество, автоматически становится его модератором.

Пользователь, опубликовавший пост, является создателем контента.

Социальная сеть должна предоставлять удобный поиск и фильтрацию всего контента.

Контент, обладающий непристойным содержанием, включающий в себя экстремистские и расистские призывы, должен быть удален из сети, а пользователи создавшие - забанены.

Контента, в сети должен быть защищен, путём вставки и сокрытии информации внутри изображения.

Функциональные требования более детально представлены на рисунке 1:

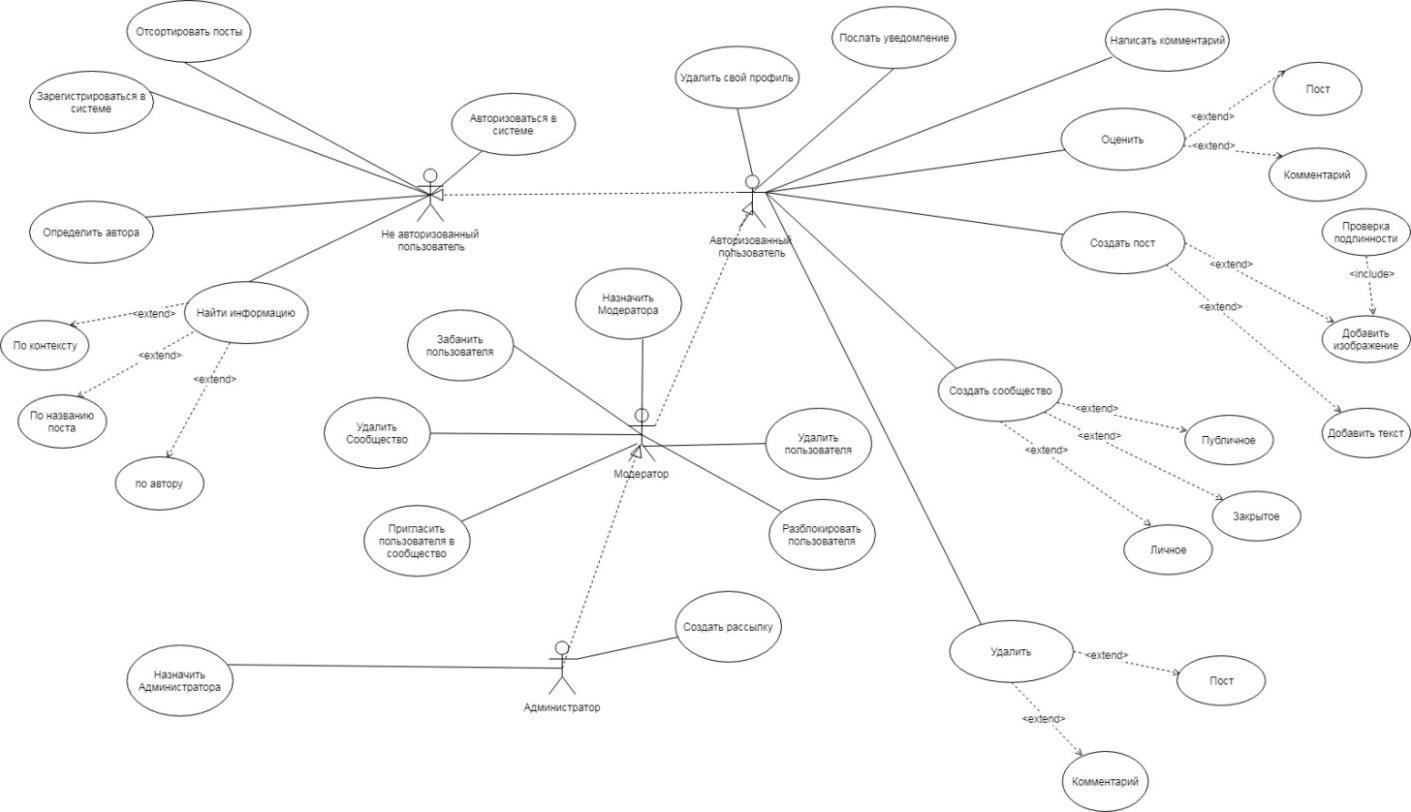


Рисунок 1- диаграмма вариантов использования

Распределение ролей в системе:

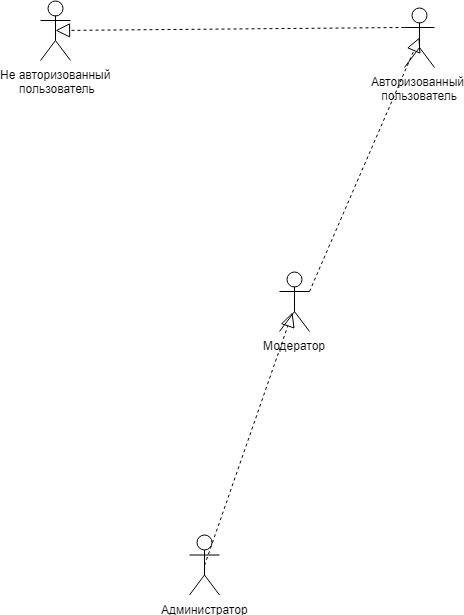


Рисунок 2 - роли в системе

Основные use-case в системе.

# Функциональность неавторизованного пользователя.

* 1. Регистрация:
     + посредством заполнения определенных полей пользователем;
  2. Авторизация:
     + посредством ввода логина и пароля;

3.Возможность отбора и поиска информации для пользователей приложения:

* + - поиск по названию поста;
    - имени пользователя;
    - по теме сообщества;
    - по скрытой в графическом контенте информации.

4. Осуществить сортировку по качеству постов для пользователя:

* + - по новизне;
    - по рейтингу; 5.Определить автора контента

# Функциональность авторизованного пользователя.

* 1. Возможность просмотра:
     + информации о себе и других пользователей;
     + своей страницы;
     + своей подписки.
  2. Возможность редактирования:
     + своей страницы;
     + своей подписки.
  3. Возможность создания сообщества.
     + публичного – просмотр разрешен всем пользователям, возможность добавление контента для всех участников сообщества;
     + личного – просмотр разрешён всем пользователям, добавлять контент могут только модераторы этого сообщества;
     + закрытого – просмотр и возможность добавление контента разрешен только участникам этого сообщества.

4.Возможность создания постов с различными видами информации:

* + - только текст;
    - только изображения;
    - текст и изображения.

1. Возможность защитить свой контент. В каждый графический контент будет встраиваться информация о его создателе;
2. Пользователь может разместить пост:
   * в любом закрытом и публичном сообществе, участником которого он является;
   * в личном сообществе, где пользователь является модератором.
3. Пользователь может оставить комментарий к посту.
4. Пользователь может поставить оценку:
   * посту;
   * комментарию;
5. Послать уведомление любому пользователю
6. Написать администратору:
   * сообщить о проблеме;
   * сообщить о нарушении;
   * пожаловаться на модератора;
7. Отписаться от любого сообщества
8. Подписаться на сообщество:
   * любое публичное;
   * любое личное;
   * на закрытое сообщество по приглашению модераторов;
9. Пользователь может удалить:
   * свой комментарий;
   * свой пост;
   * свой профиль;

# Функциональность модератора.

1. Назначить нового модератора для сообщества. 2.Забанить пользователя за нарушение правил сообщества. 3.Удалить пользователя из закрытого сообщества.

4. Удалить сообщество

* + личное и закрытое при условии, что пользователь является модератором этих сообществ;
  + модератор может послать запрос на удаление публичного сообщества администратору;

5. Разблокировать пользователя.

6. Пригласить пользователя в любое сообщество.

# Функциональность администратора.

1.Удалить любое сообщество в социальной сети 2.Сделать рассылку пользователям:

* + сообщить о технических проблемах;
  + послать предупреждение; 3.Назначить нового администратора

# ЗАДАЧИ

В соответствии с заданной целью были поставлены следующие задачи:

1.Провести анализ требований к разрабатываемой системе. 2.Провести проектирование приложения и отобразить предметную область в виде диаграмм .

3.Реализовать приложение, удовлетворяющее указанным требованиям, описать процесс разработки и итоговый результат.

4.Провести тестирование приложения

5.Развернуть приложение на стороннем сервисе.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Существует большое множество социальных сетей, в которых упор делается не на социальную составляющую (взаимодействие между людьми, их объединение по интересам), а на создание и потребление определенного вида контента. Такие социальные сети чаще всего называются интернет-площадками.

Самые известные площадки в мире:

1. Youtube
2. Pinterest
3. Reddit
4. Tumblr

Самые известные площадки в России:

1. ВКонтакте
2. Пикабу

Относительно недавно ВКонтакте запустила систему

«Прометей», поддерживающую создателей оригинального контента. При простом копировании контента из определенных сообществ под постом будет указан оригинальный автор. Сама система не является совершенной, так как допускает возможность копирования контента по частям. Задача нахождения неоригинального контента ложится на плечи рядовых пользователей.

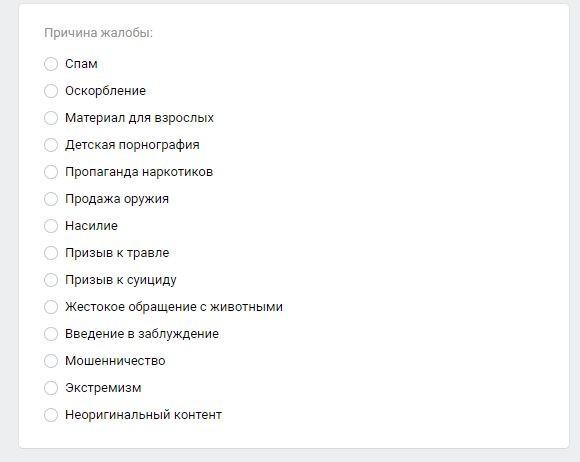


Рисунок 3 - интерфейс возможных причин жалобы пользователя

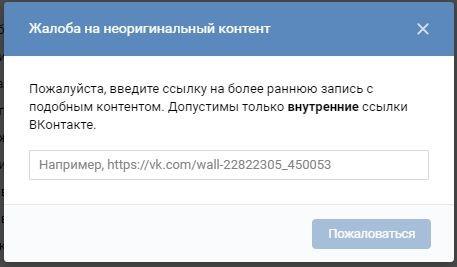


Рисунок 4 - интерфейс для жалобы пользователя

В последнее время практически на каждой иностранной интернет-площадке пользователи поднимают проблемы модерации контента. В русскоязычном сообществе таким же вопросом задаются пользователи сайта Пикабу. На Youtube создатели контента все больше задаются вопросом монетизации контента, а также позицией самой площадки.

В декабре 208 года площадка tumblr изменила свою политику модерации контента, в результате чего за следующий год потеряла до 30% своей аудитории.

Отдельно стоят площадки ВКонтакте и Reddit как хорошие примеры модерации и контроля контента. ВКонтакте выделяется системой «Прометей», а Reddit – открытой двухуровневой системой модерации сообществ – с использованием модераторов сообществ и администраторов, контактирующих с модераторами.

Несмотря на это, ВКонтакте не имеет четко модерированную систему, а Reddit не имеет разграничения сообществ на личные закрытые и публичные.

В результате разработки будет создана новая социальная сеть, ориентированная на поддержку создателей оригинального контента, путем внедрения информации о посте, где он упоминался. Таким образом, любой пользователь приложения сможет определить автора любого контента, созданного в этой сети. При создании контента в приложение будет встроен специальный сервис, который не позволит нарушителю опубликовать чужой контент. Система модерации будет создана по типу двухуровневой системы на сайте Reddit, но с использованием разных уровней доступа для модераторов и администраторов и значительного разграничения их возможностей и ответственности.

Любой пользователь может создать три вида сообщества:

* + личное – просмотр разрешён всем пользователям, добавлять контент может только создатель сообщества;
  + закрытое – просмотр и возможность добавления контента разрешён только участникам этого сообщества.
  + публичное – просмотр разрешён всем пользователям, возможность добавления контента для всех участников сообщества.

**АНАЛИЗ ЗАДАЧ**

**Задача поиска информации в сети**

Ставится задач эффективно находить разнородную информацию в сети.

На любой странице пользовательского интерфейса должен находиться интерфейс пользователя.

Действия пользователя:

1.Пользователь вводит нужную информацию в поле.

2.Приложение показывает результаты.

3.Пользователь может отменить выбор.

4.Пользователь может продолжить поиск.

**Задача поиска информации в сети**

Информация должна быть отсортирована по рейтингу и новизне.

Предполагается только сортировка постов в сети, так как они являются самым важным компонентом в приложении. Пользователю предоставляется интерфейс для сортировки и возможность выбора критерия, по которому сортировать.

**Задача модерации контента в сети**

Модерация контента должна быть осуществлена в соответствии с требованиями. Пользователь должен имеет возможность выбрать тип сообщества при создании. Модерация будет осуществлять исходя из этих условий.

**Задача удаления контента в сети**

Если не накладывается особых ограничений, то контент удаляется каскадно из сети. При удалении сообщества удаляются и посты.

**Задача защиты контента в сети**

Для защиты контента был использован метод Least Significant Bit (сокращенно – LSB). Это один из самых распространенных методов стеганографии, популярность которого обусловлена тем, что в нем учитывается физиология человека, а именно его способность различать изменения яркости.

Порог чувствительности человеческого глаза составляет примерно 1 – 3%, для яркости, получаемой по формуле:

l(p)=0,299r(p)+0,587g(p)+0,114b(p),

Где l(p) – яркость пикселя, r, g, b – красная, зеленая и голубая компоненты. При стеганографической вставке в контейнер чаще всего используется синий канал RGB. По формуле выше видно, что изменение младших бит синего канала изменит общую яркость пикселя менее чем на один процент. Таким образом, пользователь не заметит изменения контента, а сам контент будет защищен стеганографической вставкой. В качестве усовершенствования алгоритма был выбран генератор на основе решета Эратосфена – генератор выбирает пиксели, в которые будет произведена вставка. Такое решение позволит дополнительно защитить контент, так как авторство будет определяться и после минимальных манипуляций с защищенным контентом.

**Задача отправления приглашения на вступление пользователя в сообщество**

Пользователям должна отправляться ссылка для однократного использования. При этом действия пользователя должны быть идентифицируемы. Планируется создать get запрос с шифрованными параметрами.

**Задача оформления интерфейса приложения**

В приложении будут созданы:

* домашняя страница пользователя
* страница сообщества с постами
* страница создания контента
* страница новостей
* страница постов пользователя

**Задача отправления уведомлений пользователями приложения и оценки контента**

Должен быть продуман интерфейс для отправки уведомлений от любого пользователя другим пользователям. Необходимо также создать рассылку всем пользователям от администратора.

## Анализ средств реализации

Приложение имеет front и back часть. Рассмотрим технологии, которые использовались там отдельно.

**Back-end:**

**PostgreSQL -** свободно распространяемая СУБД, имеющая кроссплатформенную реализацию. Сильными сторонами PostgreSQL считаются:

* высокопроизводительные и надежные механизмы транзакций и репликации;
* расширяемая система встроенных языков программирования
* легкая расширяемость.

СУБД постоянно модернизируется, поэтому отвечает последним стандартам Sql и как раз подходит для приложения.

**Flask** версия 1.0.3 – это фреймворк для создания веб-приложений на языке программирования Python. Он относится к категории так

называемых микрофреймворков – минималистичных каркасов веб-приложений, сознательно предоставляющих лишь самые базовые возможности.

**Safrs** версия 2.2.1 - библиотека для развертывания SWAGGER, служащего для облегчения документации предоставляемых API.

**Werkzeug** версия 0.14.1 - модуль flask, обеспечивающий систему маршрутизации. Удобен для конструирования url и отслеживания его уникальности.

**SQLAlchemy** версия 1.3.3 - ORM для работы с реляционными СУБД. Позволяет писать высокопроизводительные запросы к серверу с базой данных, является переносимой и совместимой с популярными СУБД. Обеспечение защиты от sql инъекций, так как параметры запросов экранируются.

**cryptography** версия 2.7 - библиотека предоставляющая функции для шифрования, удобна при использовании алгоритмов симметричного шифрования.

**Flask\_Login** версия 0.4.1 **-** модуль flask, облегчающий работу с сессиями пользователей в системе и их авторизацию.

**Stegano** - стеганографическая библиотека, используется для встраивания информации в графический контент в сети. Удобна и проста в применении.

**Front-end:**

WTForms версия 2.2.1 - модуль flask, позволяющий создавать формы, абстрагируясь от html. Очень удобен для минималистичных интерфейсов.

Jinja2. Этот шаблонизатор используется для языка питон и рекомендуется сообществом flask.

HTML - стандартизированный язык разметки в интернете, необходим приложению для отображения в браузере.

CSS - каскадные таблицы стилей, использующиеся для создания дизайнерской стороны пользовательского интерфейса.

Bootstrap - фраймворк, включающий себя в набор инструментов для создания веб приложений. Включает в себя html и css шаблоны, что и было использовано в приложении. В приложении не используется адаптивная разметка и задействована библиотекам не в полную силу, так как может возникать несовместимость со старыми браузерами.

JavaScript - скриптовый интерпретированный язык, использующийся в приложении для создания диалоговых окон и поиска контента в сети.

Unittest –встроенная библиотека для написания Unit тестов.

**Uml диаграммы**

Далее будет продемонстрировано 8 типов диаграмм. Это необходимо для того, чтобы полностью сложилось представление о разработке, функционировании, тестировании системы. Также позволит определить границы рассматриваемой системы.

**Диаграмма классов**

Ниже приведена диаграмма классов на концептуальном уровне.

В приложение есть 7 основных сущности Community, User, Administrator, Comment, Post, Social object, Notification и сервисы, взаимодействующие с ними.

Диаграмма для пользователя, администратора, и модератора и сервисов, с которыми они взаимодействуют.

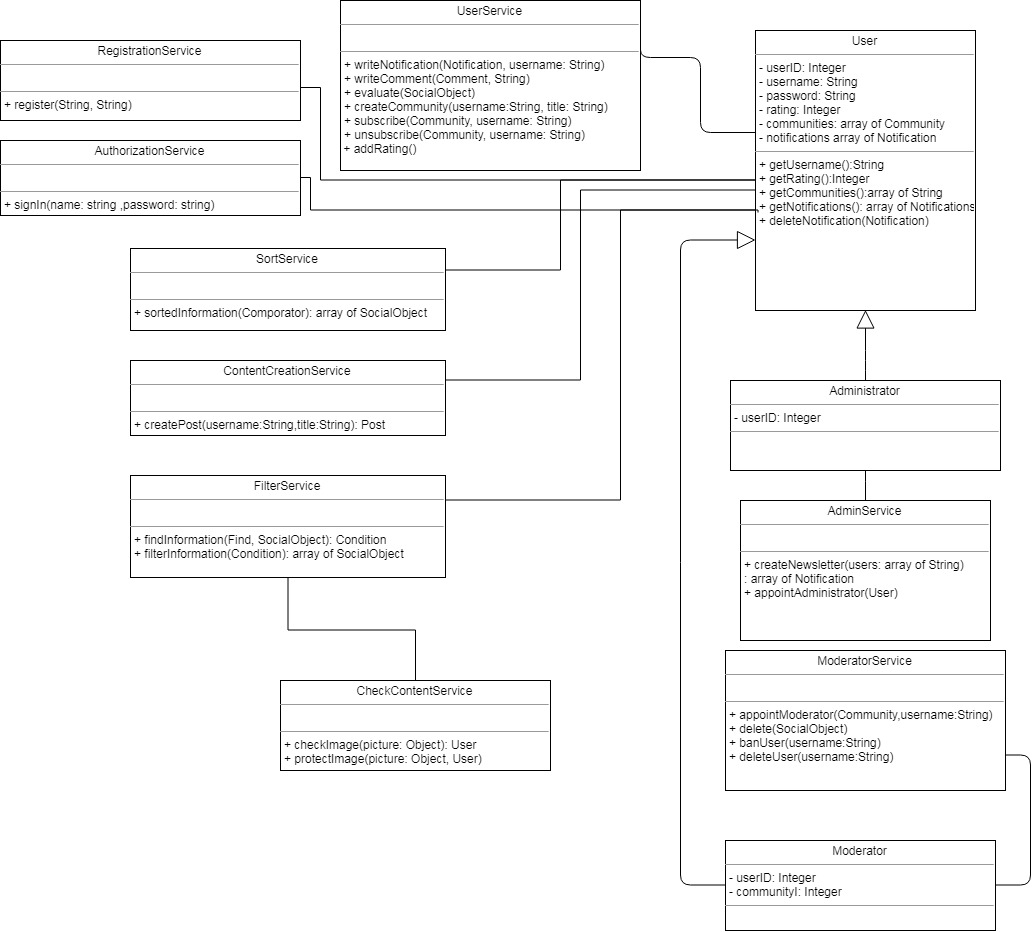
****

Рисунок Диаграмма классов для пользователей в системе

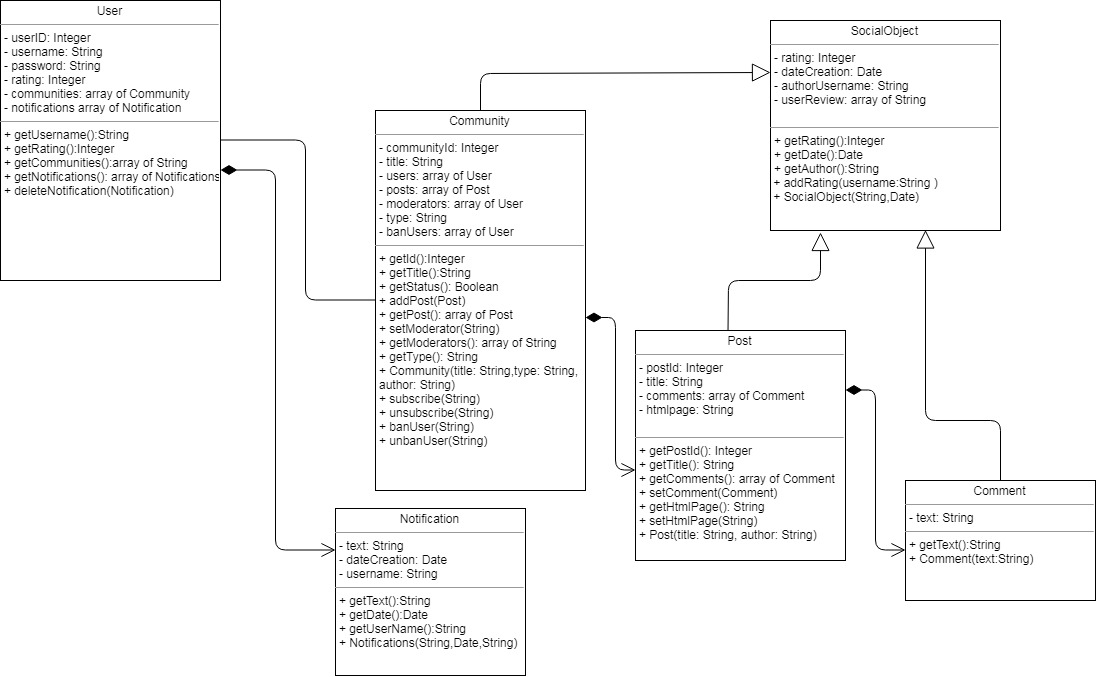


Рисунок диаграмма классов для наиболее важных сущностей в нашей сети

**Диаграмма последовательностей**

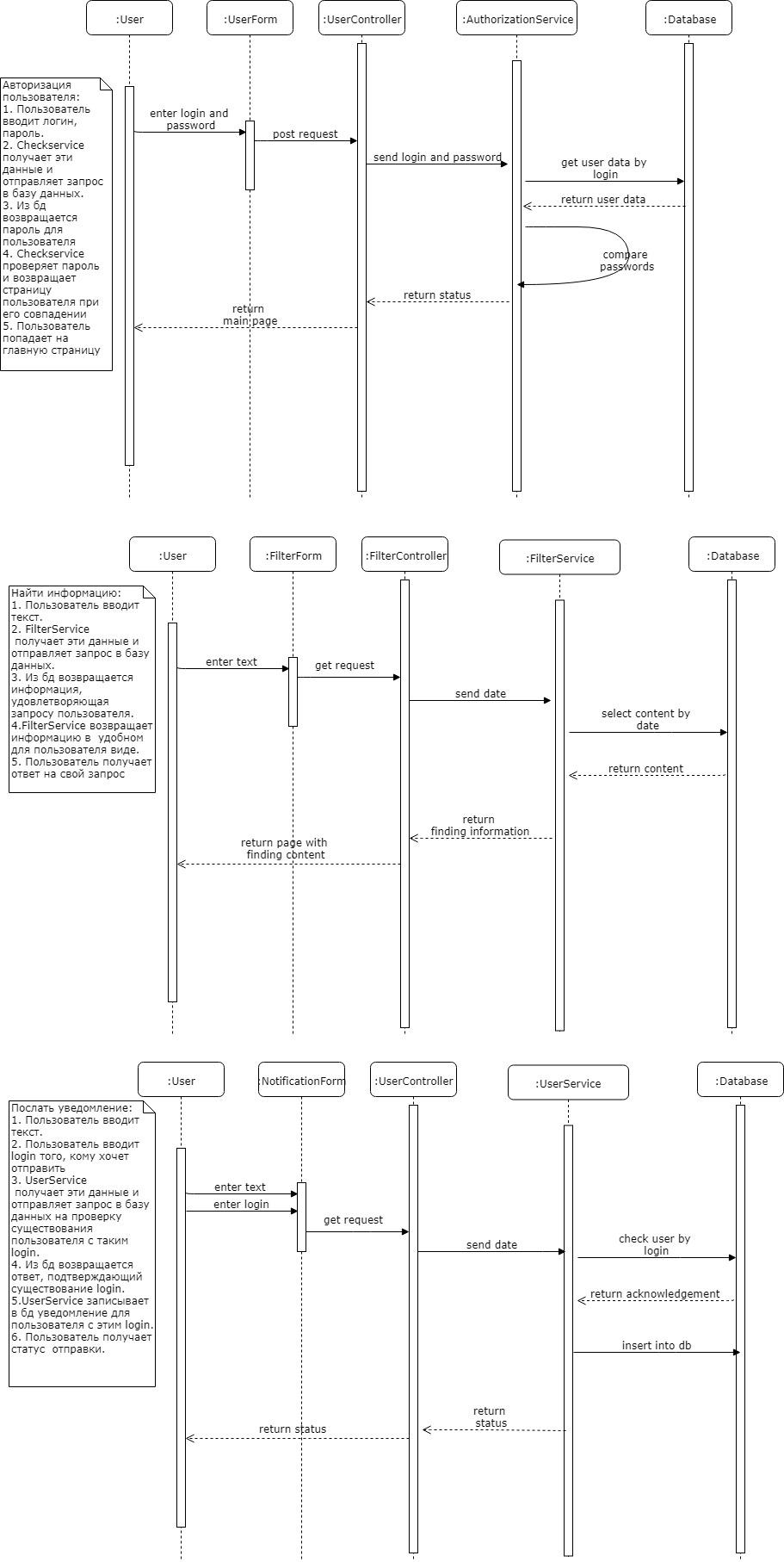
Все диаграммы приведены для успешного сценария пользователя.

Основные виды сообщений:

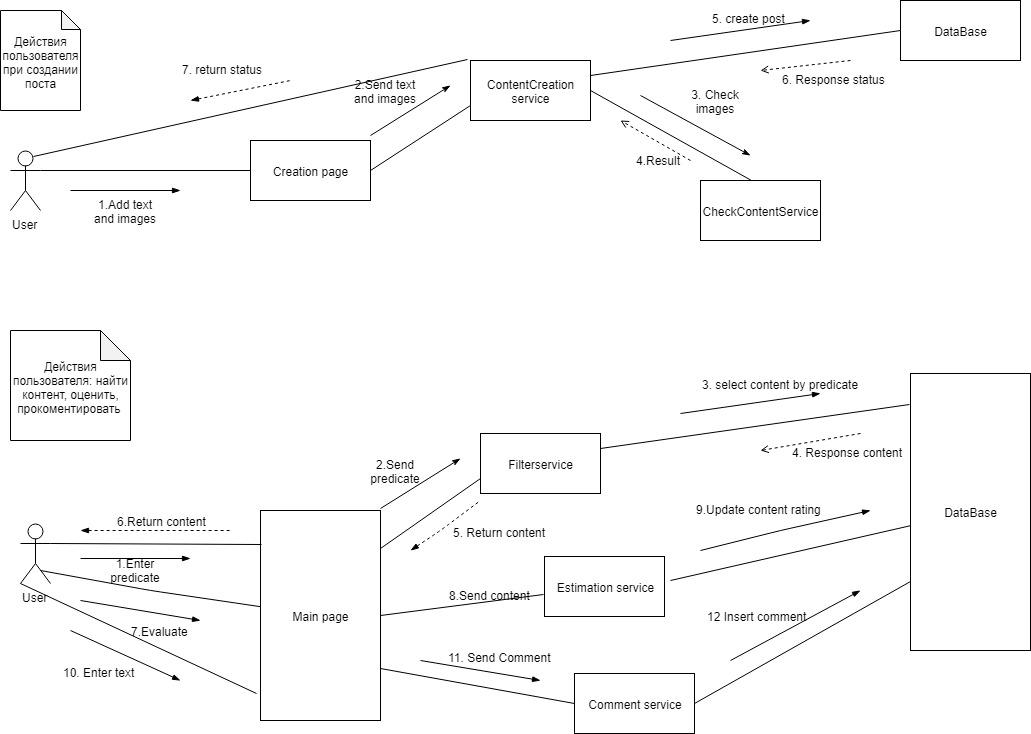
1. Post и get запросы на сервер от браузера, так как в качестве протокола взаимодействия был выбран http.

2. Ответом сервера на запросы браузера является html страница.

3. Select, insert, update, delete запросы сервера к базе данных.

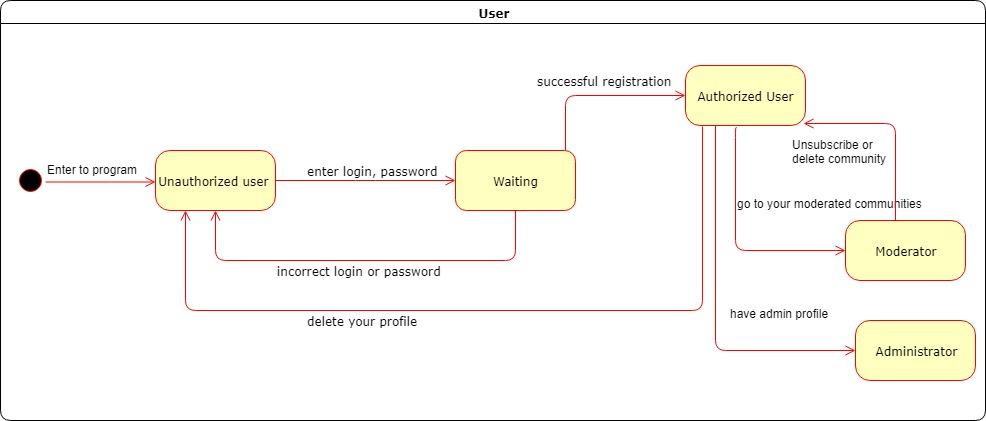


**Диаграмма коопераций**



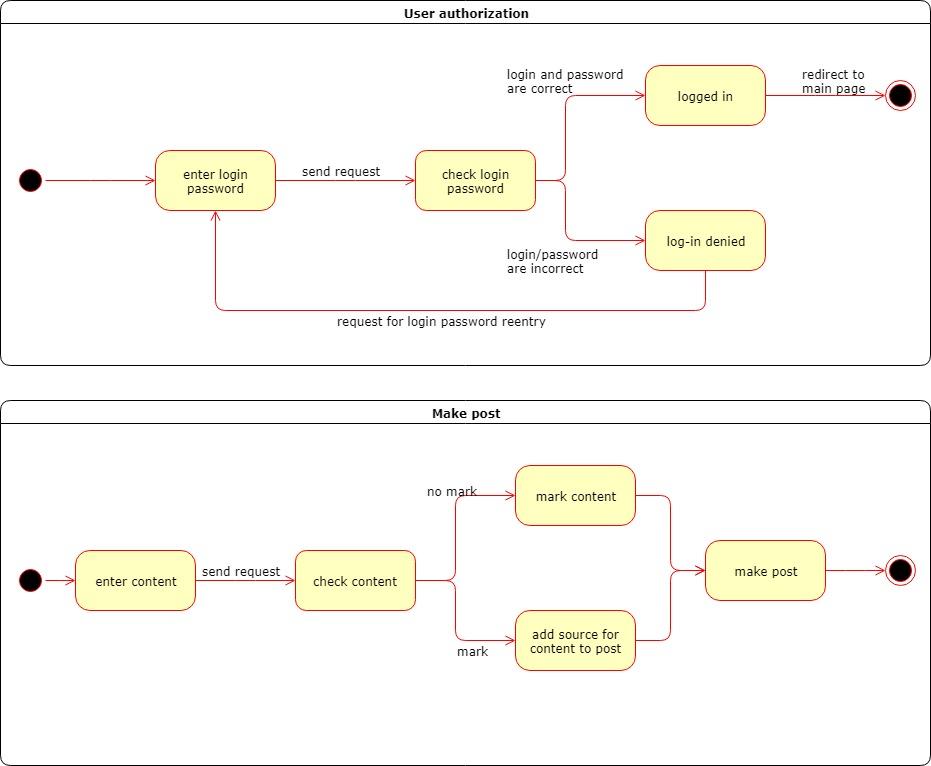
**Диаграмма состояний**

На диаграмме показаны всевозможные состояния пользователя в системе и переходы между ними.



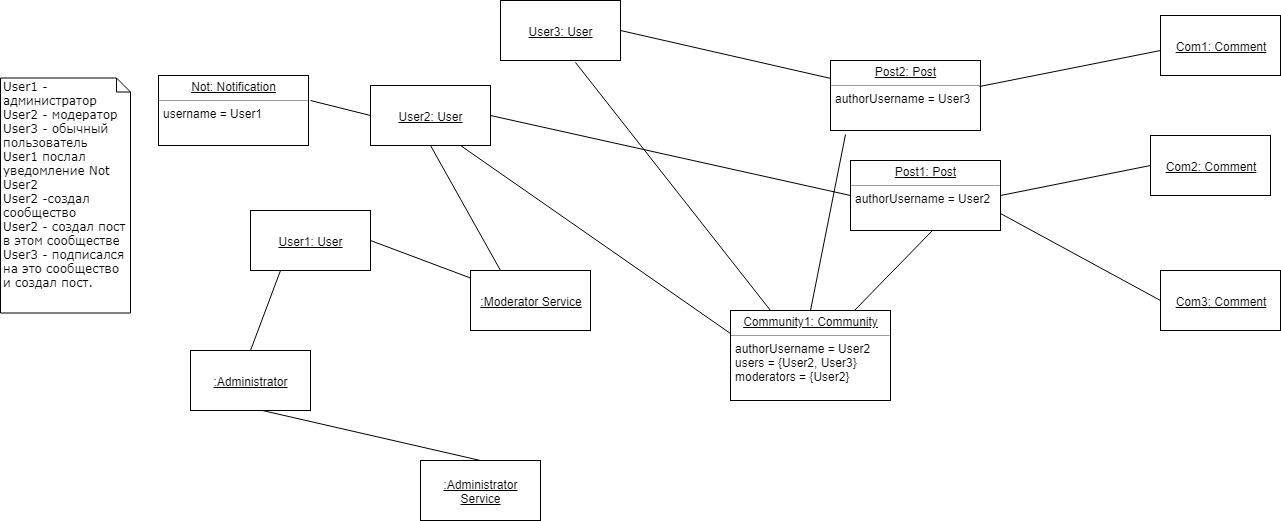
**Диаграмма активности**

Ниже представлено две наиболее важные диаграммы активности для авторизации пользователя и создания поста в сообществе.



**Диаграмма объектов**

На диаграмме представлено три пользователя User1- администратор, User2 – модератор, User3 – обычный пользователь и объекты которые им доступны.

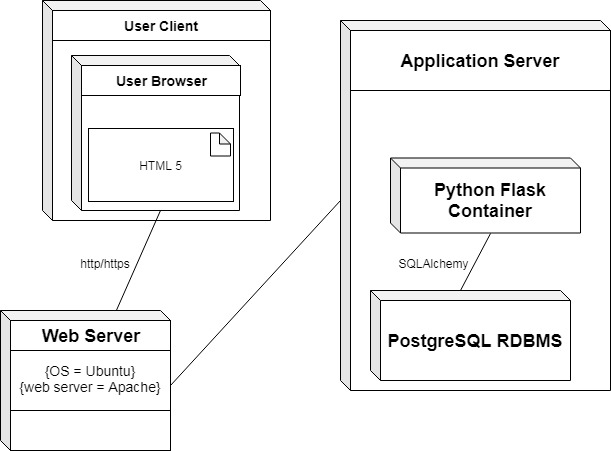


**Диаграмма развертывания**

Приложение будет поднято на apache сервере.

Взаимодействие между клиентом и сервером приложения будет

происходить при помощи REST архитектуры.



# Реализация

**Поиска информации в сети**

Так как в сети содержиться разнородный контент, то было принято решение, что поиск будет проходить на стороне клиента, посредством JS скрипта.

Весь контент в сети, который может подлежать поиску, должен был быть обернут в блоки. Эти дочерние блоки в свою очередь должны содержаться в блоке с id = content. Поиск происходит по блокам. Блоки, в которых информация не была найдена, не отображаются браузером.

**Задача модерации контента**

Для облегчения модерации в сообществе было поле тип, с помощью таблиц модератор и администратор определялись права пользователя и объединялись с ограничениями, связанные с типом сообщества.

Реализация происходила путём добавления условных выражений и механизма вывода сообщений пользователю.(flush)

**Задача удаления контента из сети**

Задача решалась путем добавления ограничения целостности на таблицы Post, User, Community, Comment. Для всех внешних ключей была использована опция ondelete = cascade.

**Задача защиты контента в сети**

Был выбран робастный LSB алгоритм. Его реализация взята из библиотеки stegano. Надо заметить, что встраивание информации происходит незаметно для пользователя. В изображения добавляется имя пользователя, создавшего контент. Алгоритм защищён от помех и незначительных искажений изображения.

**Задача отправления приглашения**

Приглашение, отправленное пользователю должно быть однозначно идентифицировано его именем. Необходима шифрация данных. Для решения этой задачи используется библиотека criptography и встроенный в неё алгоритм блочного шифрования AES. Шифруется токен состоящий из названия сообщества и имени пользователя. Этот токен сохраняется в строке браузера в виде get параметра. При переходе пользователя по этой ссылке происходит дешифрация токена.

**Задача оформления интерфейса приложения**

Для оформления использовались готовые css шаблоны из фраймворка Bootstrap. Были заданы стили текста, кнопок, формы. Адаптивная вёрстка в проекте не применялась.

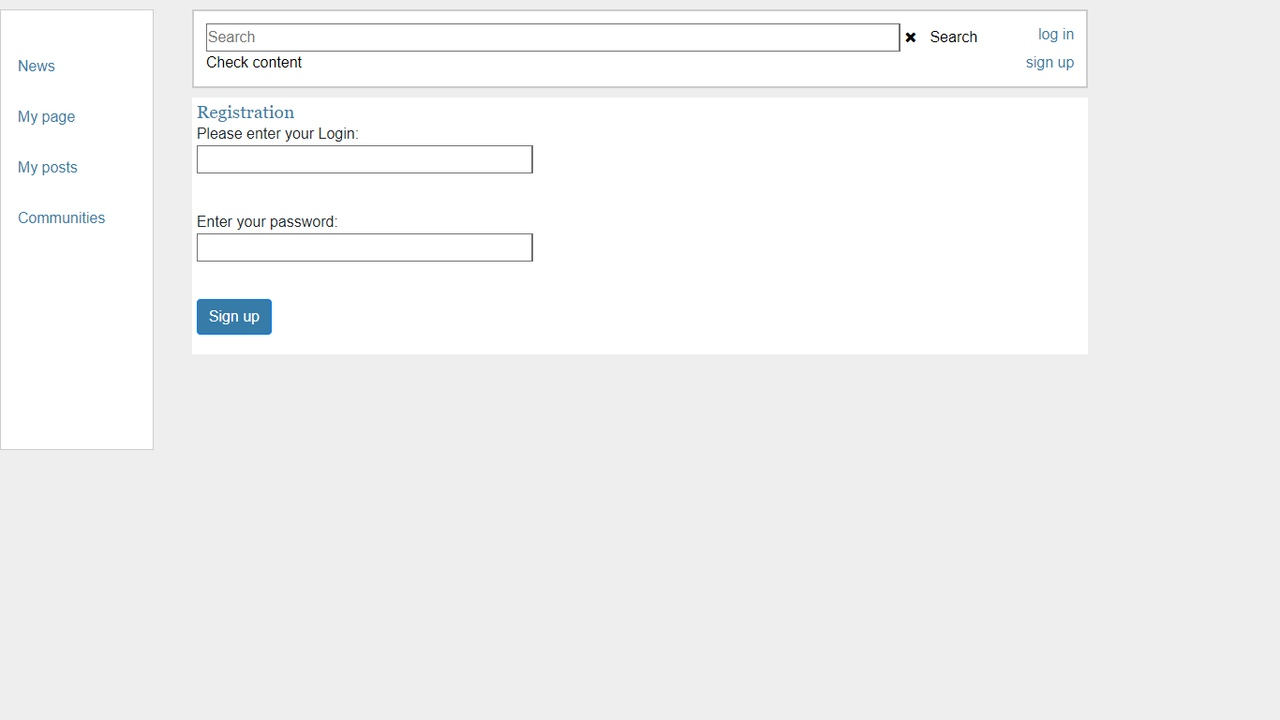
**Задача отправления уведомления и оценки контента**

Необходимо было сделать отправку уведомлений одиночным пользователем, так и сделать рассылку. Для этого была создана функция AddNotification. Уведомления было решено хранить json массивом в атрибуте notifications у пользователя.

Оценка контента проводилось путем создания в посте и в комментарии поля evalutedusers и постоянного его обновления. Для этого поля проводилась бинаризация массива пользователей, которые оценили этот контент.

# Интерфейс

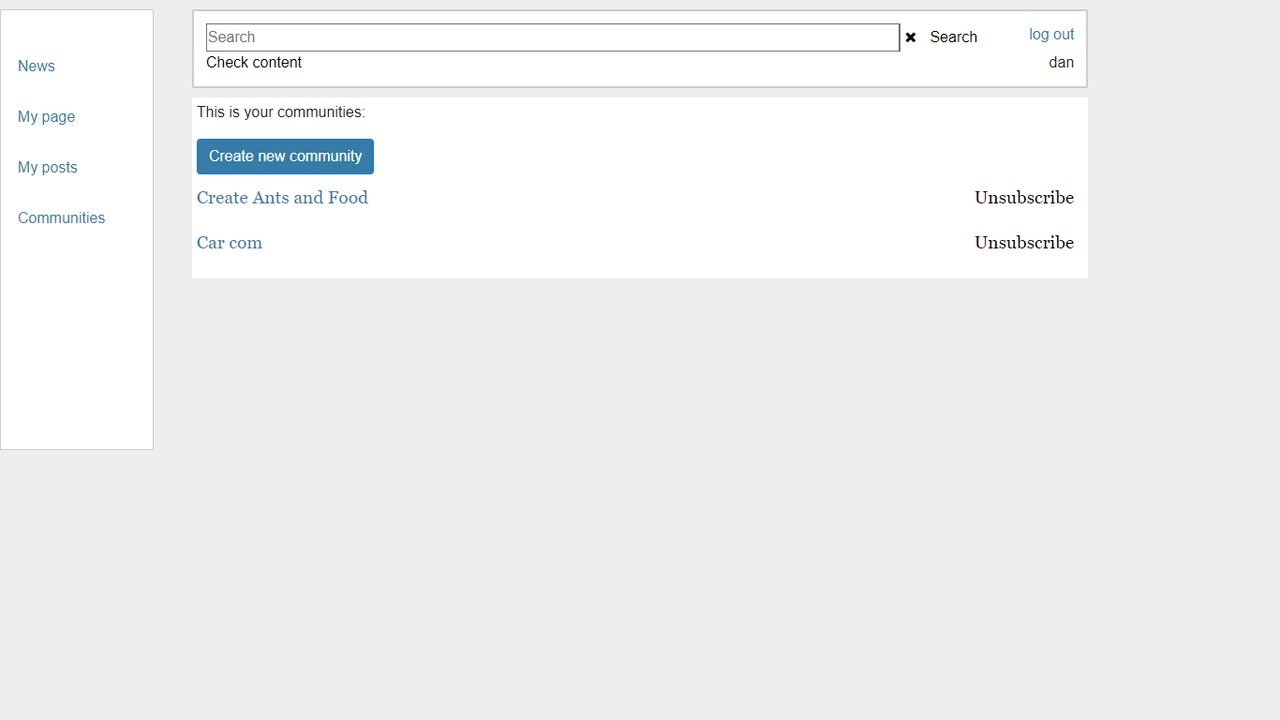
Интерфейс регистрации и авторизации



Новостная страница для пользователя dan.

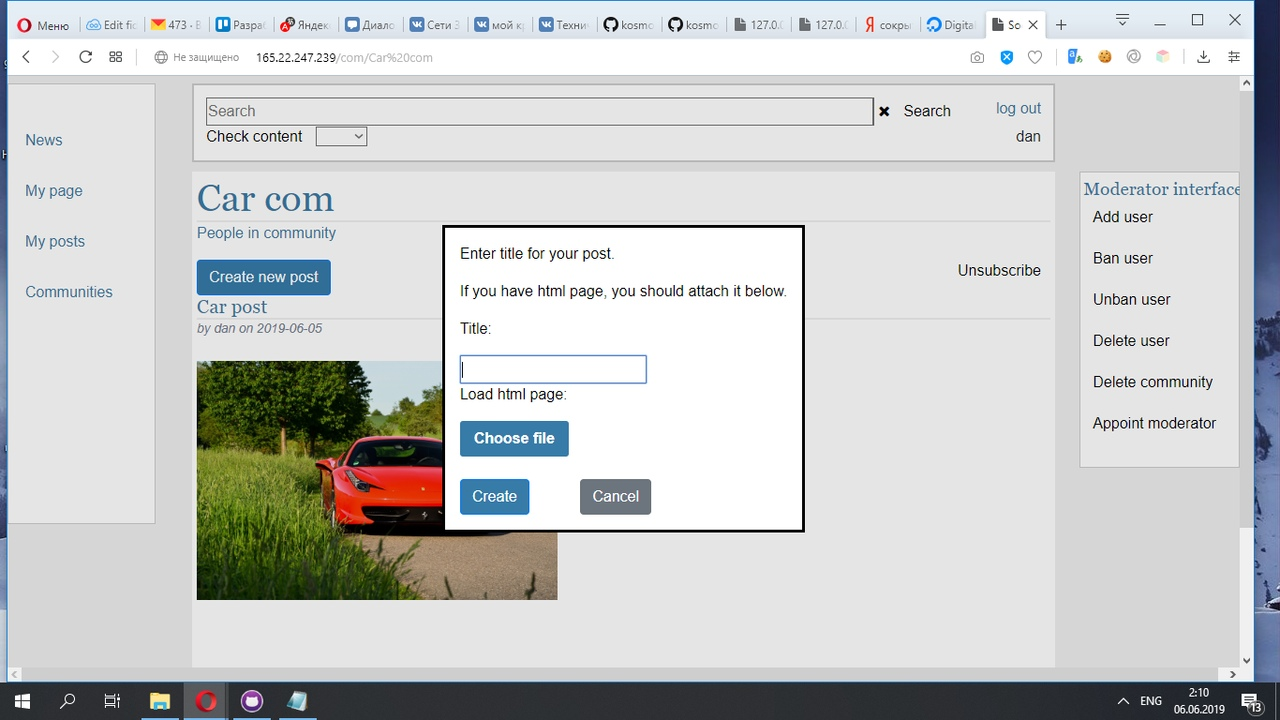
# 

Интерфейс подписки пользователя

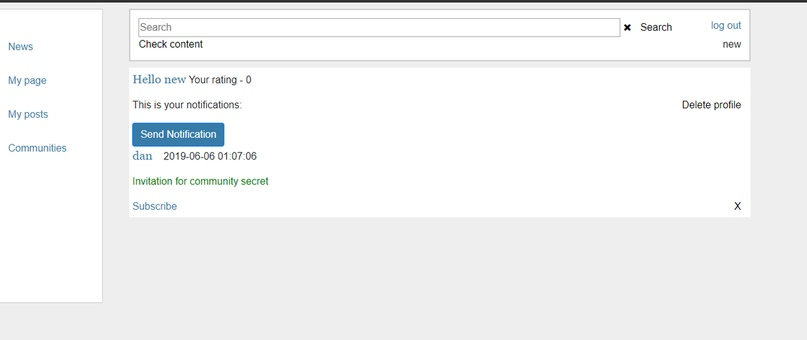


Интерфейс зарегистрированного пользователя, подписанного на сообщество.

# Диалоговое окно на создание поста



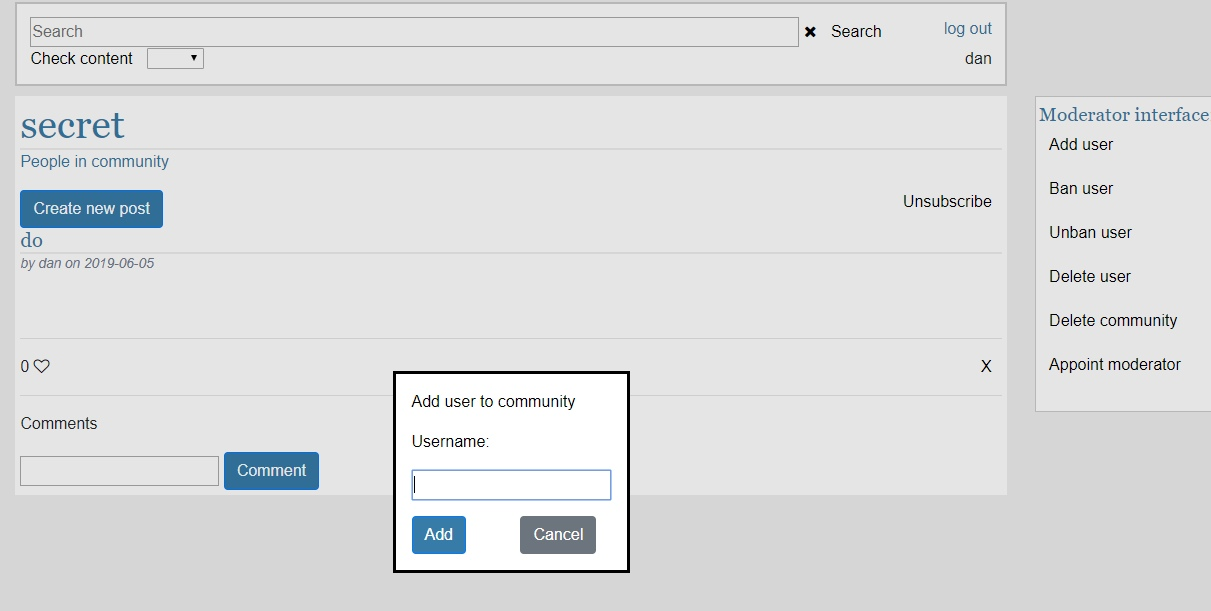
Страница пользователя с уведомлениями. Ниже представлено приглашение вступить в сообщество.



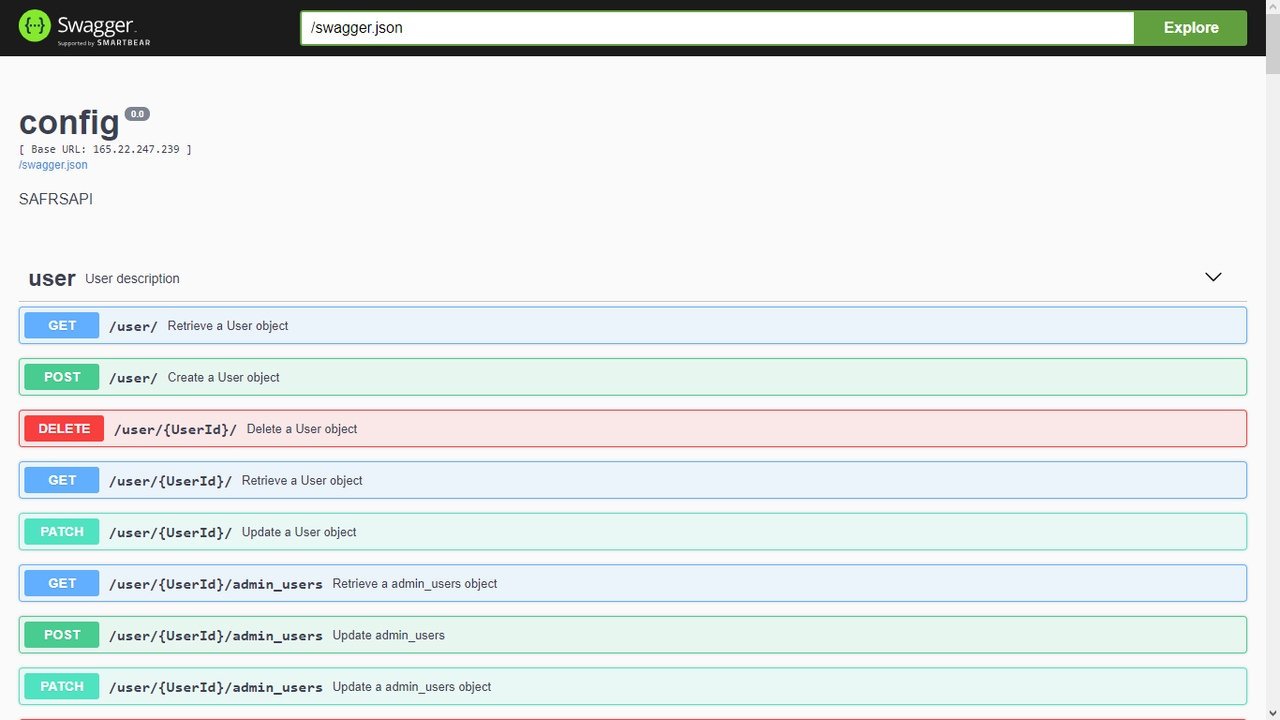
Интерфейс администратора. Для администратора он одинаковый в любом сообществе.

# 

Модальное диалоговое окно. Так представляется большинство модальных диалоговых окон в приложении.



# Swagger api:



# Тестирование

Для проверки приложения были проведены следующие виды тестирований:

* модульное тестирование
* системное тестирование
* end-to-end тестирование

1.Модульное тестирование

- тестирование сервиса администратора

1. Назначить нового администратора.
2. Создать рассылку.

- тестирования сервиса работы с комментариями

1. Создать комментарий
2. Удалить комментарий
3. Оценить комментарий

- тестирование на удаление сущностей

1. Удалить пользователя
2. Удалить сообщество
3. Отписаться от сообщества

- like сервис

1. Оценить пост
2. Оценить и посмотреть кто поставил лайк

- сервис модератора

1. Создать сообщество, проверка что пользователь модератор
2. Назначить модератора
3. Удалить пост

- сервис уведомлений

1. Послать уведомление
2. Удалить уведомление

- сервис регистрации

1. Создать пользователя
2. Проверить шифрацию паролей

- сервис сортировки

1. Сортировать по новизне
2. Сортировать по рейтингу

- int тестирование

1. Проверить на существование пользователя
2. Проверка на существования администратора

При финальном тестировании системы все тесты были пройдены.

2.Системное тестирование.

Проводилось в различных браузерах.

Opera, Yandex, Google Chrome, Edge, Mozilla Firefox.

Во всех браузеров приложение функционировало обычным образом.

Только не весь функционал был доступен в браузере Mozilla Firefox из-за проблемы с java скриптами.

Также было проведено тестирование на на устройствах с различными размерами экранов. Результаты для 17 и 14 дюймовых экранов были удовлетворительны. Приложения не является мобильным и не предназначено для малых экранов.

3. End-to-end тестирование

В браузере opera различными пользователями, разного возраста и пола было проверено 10 сценариев.

Сценарии, которые были проверены:

1. Поиск интересного поста - переход на страницу сообщества - сортировка постов в этом сообществе проверка уникальности картинки.
2. Регистрация пользователя - авторизация пользователя - создание сообщества.

Остальные сценарии для авторизованных пользователей

1. Подписка на публичное сообщество - создание поста в этом сообществе.
2. Поиск старого уведомления - посылка ответного уведомления пользователю - переход в сообщество с постами этого пользователя и постановка лайков.
3. Создание сообщества - добавление поста - добавление к нему картинки - добавление текста - создание поста в сообществе - комментарий к нему.
4. Зайти в сообщество - создать пост - добавить картинку - скачать картинку из сети - зайти в другое сообщество - добавить картинку - получить исключение.
5. Перейти в личное сообщество, в которым пользователь не является модератором - добавить пост - получить исключение - поставить like существующему посту - сделать к нему комментарий
6. Зайти под админом - удалить сообщество - сделать рассылку всем пользователям - назначить нового администратора - посмотреть все сообщества - зайти в любое закрытое.
7. Зайти в сообщество, в котором пользователь является модератором - забанить двух пользователей - разблокировать одного из них - создать пост в этом сообществе.
8. Зайти в сообщество, в котором пользователь является модератором - назначить нового модератора - пригласить двух пользователей в сообщество - удалить сообщество.

# Заключение

Было разработано приложение, отвечающее всем требованиям к системе. Проведен анализ предметной области, проведена реализация поставленных задач. В том числе было сделано:

- сервис защиты контента

- сервис поиска информации в сети

- сервис сортировки информации

- сервис оценки и комментирования постов

- сервис отправки и получения уведомлений пользователями

- сервис модератора

- сервис администратора

- сервис для создания контента

- сервис для регистрации и авторизации

В работе были показаны диаграммы классов, состояния, активности, объектов, прецедентов, развертывания, коммуникации и последовательности.

При помощи html вёрстки были реализованы следующие интерфейсы

1. Интерфейс обыкновенного пользователя
2. Интерфейс модератора
3. Интерфейс администратора
4. Интерфейс для создания и защиты контента
5. Интерфейс для оценки и комментирования сообщества

# Список использованных источников

1. Фаулер М. UML. Основы, 3-е переиздание. - Пер. с англ. - Санкт-Петербург: Символ-Плюс, 2004. - 192с.
2. Flask documentation [Электронный ресурс]. - URL: <http://flask.pocoo.org/docs/1.0/>
3. SQLALchemy documentation [Электронный ресурс]. - URL: <https://flask-sqlalchemy.palletsprojects.com/en/2.x/>
4. PostgreSQL documentation [Электронный ресурс] URL: <https://www.postgresql.org/files/documentation/pdf/11/postgresql-11-A4.pdf>
5. Bootstrap documentation [Электронный ресурс] URL: <https://getbootstrap.com/docs/4.3>