

МИНОБРНАУКИ РОССИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ВГУ»)

Факультет Компьютерных наук
кафедра программирования и информационных технологий

Курсовой проект

Аналог твиттера «Crowter»

09.03.02 Информационные системы и технологии
ПИВИС

Обучающийся _____ *Е. А. Бродская, 3 курс*

Обучающийся _____ *А. А. Казанин, 3 курс*

Обучающийся _____ *А. С. Соловьёв, 3 курс*

Воронеж 2021

Содержание

Содержание	2
Введение	3
1 Постановка задачи	4
2 Глоссарий	5
3 Анализ предметной области	6
3.1 Анализ существующих решений.....	6
4 Анализ задачи.....	9
4.1 Варианты использования системы.....	9
4.2 Общая структура системы	11
4.3 Конфигурация системы	12
4.4 Взаимодействие компонентов системы	14
4.5 Взаимодействие в системе.....	15
4.6 Варианты состояния системы.....	16
4.7 Действия с системой.....	17
4.8 Развертывание системы	19
4.9 IDEF0.....	20
4.10 ER-диаграмм.....	21
5 Анализ средств реализации	21

Введение

Потребности людей в получении и распространении информации увеличиваться с каждым годом. Ежедневно происходит огромное количество различных событий, от котят, бегающих за лазерной указкой, до полетов людей в космос. Так как все эта информация разбросана по всему интернету становится очень тяжело уследить за тем что важно и интересно. Можно использовать различные тематические приложения, искать новости в поисковых сервисах или просто следить за тем, что попадается. Объединение пользователей в одном приложении, где они смогут наблюдать за информацией от интересующих их людей, создавать твиты, комментировать твиты и доступность с мобильного телефона значительно упрощает и делает процесс получения информации намного комфортнее.

Приложение должно позволить пользователям осуществлять обмен информацией. Основная функциональность приложения должна включать в себя:

- размещение своих “твитов”
- просмотр “твитов” других пользователей
- взаимодействие с “твитами” других пользователей

Данный групповой проект посвящен разработке такой системы: простой для восприятия человеком, но в то же время выполняющей все необходимые функции.

1 Постановка задачи

Данное проект является приложением позволяющее пользователям осуществлять обмен информацией.

Данная система автоматизирует процесс размещения, получения, взаимодействия с информацией.

Система предназначена для:

- Получения информации
- Размещении информации
- Взаимодействию с получаемой информацией

Для достижения данной цели были выделены следующие подзадачи:

1. Разработка Front-end части сервиса;
2. Разработка Back-end части сервиса;
3. Создание связи между Front-end и Back-end частями приложения;
4. Разработка базы данных.

2 Глоссарий

Твит – текстовое сообщение, размещенное пользователем.

Лайк – условное выражение одобрения материалу.

Дизлайк – условное выражение неодобрения материалу.

Тег – метка для облегчения процесса поиска.

Java – строго типизированный объектно-ориентированный язык программирования общего назначения

Фреймворк – программная платформа, определяющая структуру программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.

Spring Boot – фреймворк с открытым исходным кодом для Java-платформы.

PostgreSQL – объектно-реляционная система управления базами данных.

SDK (software development kit) - набор средств разработки, позволяющий специалистам по программному обеспечению создавать приложения для определённого пакета программ, программного обеспечения базовых средств разработки, аппаратной платформы, компьютерной системы, игровых консолей, операционных систем и прочих платформ.

Flutter - SDK с открытым исходным кодом для создания мобильных приложений.

Аватар - графическое или фотоизображение пользователя.

Никнейм - сетевое имя, псевдоним, используемый пользователем в Интернете.

Гость – незарегистрированный пользователь

Лента – формат отображение твитов для пользователей.

API (программный интерфейс приложения) - описание способов которыми одна компьютерная программа может взаимодействовать с другой программой.

REST - архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

Бэкенд (backend) — программно-аппаратная часть сервиса. Бэкенд отвечает за осуществление функционирования внутренней части приложения.

3 Анализ предметной области

3.1 Анализ существующих решений

Твиттер (Twitter)

Является одной из крупнейших социальных сетей для публичного обмена сообщениями в мире. В твиттере зарегистрированы сотни миллионов пользователей из разных уголков планеты. Каждый день в твиттере публикуются огромное количество новой информации. Но не вся информация подойдет для каждого пользователя, пользователь может захотеть выразить свое недовольство, поставив дизлайк, но в твиттере такой функции нет. Так же в нашем постоянно меняющемся мире информация обновляется каждую секунду, и актуальная информация может стать нерелевантной, а возможность редактировать твиты отсутствует.

При попытке зайти на сайт твиттера у пользователя имеется только два варианта: войти и зарегистрироваться, но не каждый пользователь хочет сразу регистрироваться (рис.1).

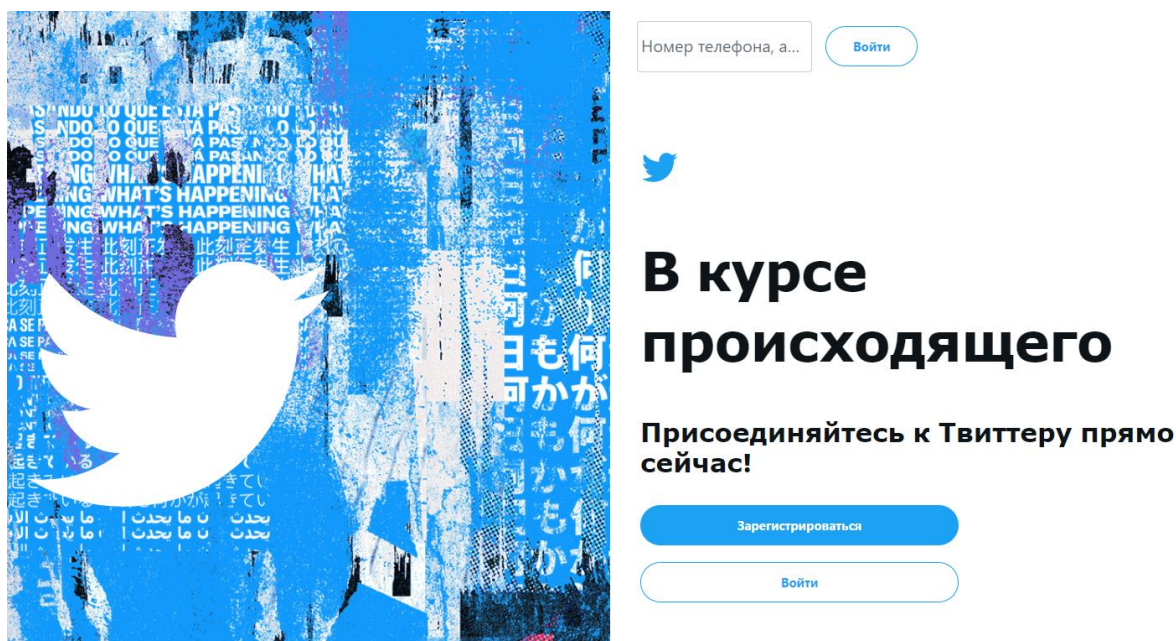


Рисунок 1 – Домашняя страница социальной сети Twitter

Реддит (Reddit)

Является крупным информационно-новостным сайтом. Имеет обширную, сплоченную аудиторию. Так же реддит поддерживает систему голосования за понравившееся сообщение, позволяющее ему тем самым продвигаться.

Данный сайт не имеет обширной аудитории в странах СНГ, более ориентирован на западные страны. Так же имеется слишком суровая модерация сообщений.

Доступ к просмотру информации о фильмах и сериалах также предоставляется только по платной подписке, отметка о просмотре серии только после просмотра ее на сервисе.

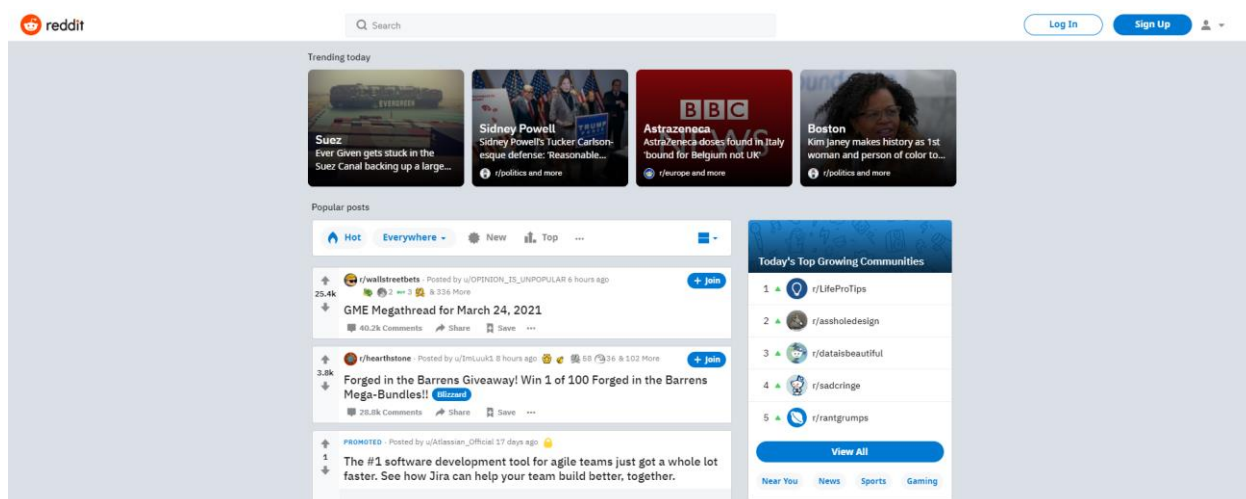


Рисунок 2 – Домашняя страница сервиса Reddit

Инстаграм (Instagram)

Является приложением для обмена фотографиями и видеозаписями с элементами социальной сети, позволяющее снимать фотографии и видео, применять к ним фильтры, а также распространять их через свой сервис и ряд других социальных сетей.

Данное приложение сфокусировано на фотографиях и видеозаписях, тем самым большой объем текстовой информации остаётся в стороне. Так же, как и Твиттер данное приложение не имеет возможности выражать

недовольство материалами путем выставления дизлайков. А также требуется регистрация для начала использования приложения.

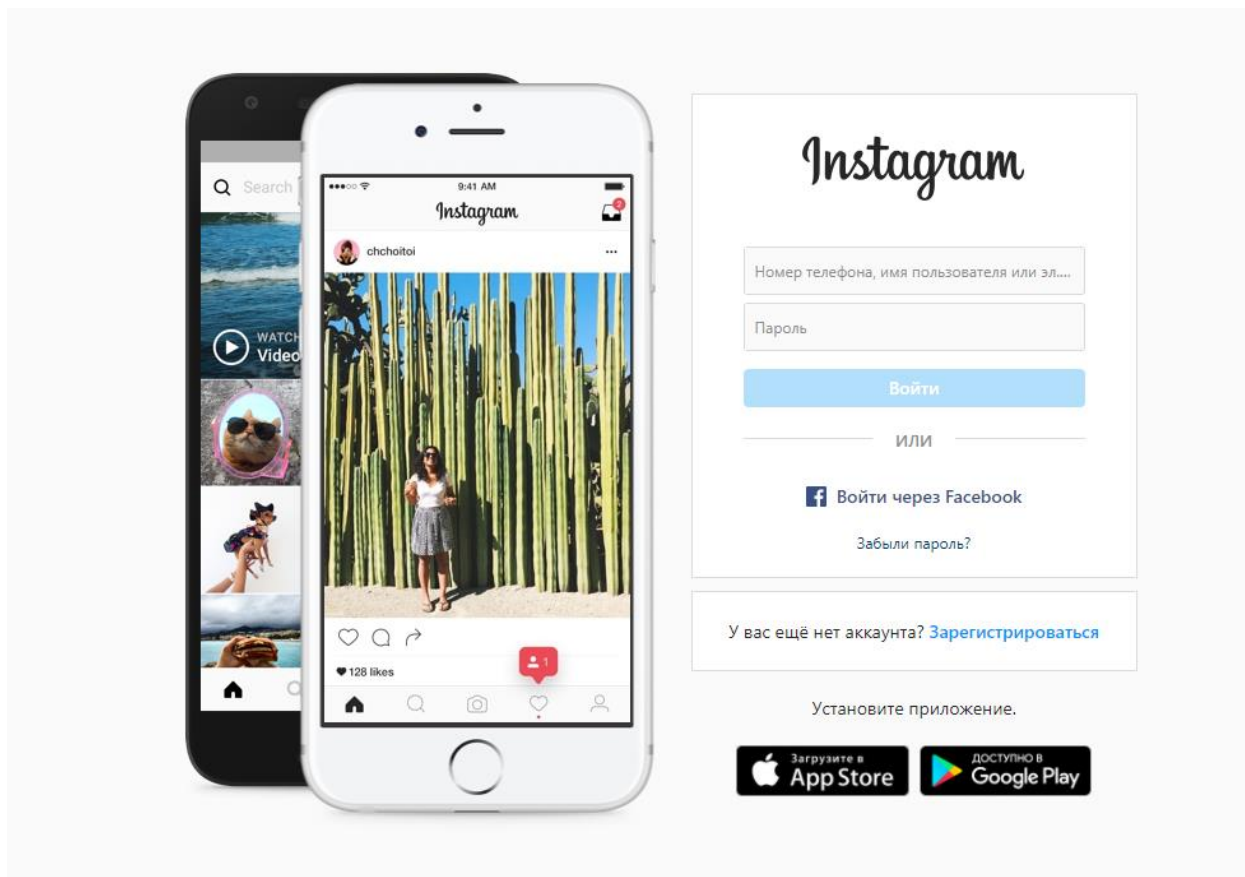


Рисунок 3 – Домашняя страница сервиса Instagram

Существует большое количество интернет сервисов, позволяющих размещать, публиковать, и взаимодействовать с информацией, но почти никаких из них не позволяют выставлять негативные оценки материалам, размещенным на сайтах. Так же не везде реализована система для быстрого, анонимного сохранения полученной информации, что является важным для некоторых пользователей.

4 Анализ задачи

4.1 Варианты использования системы

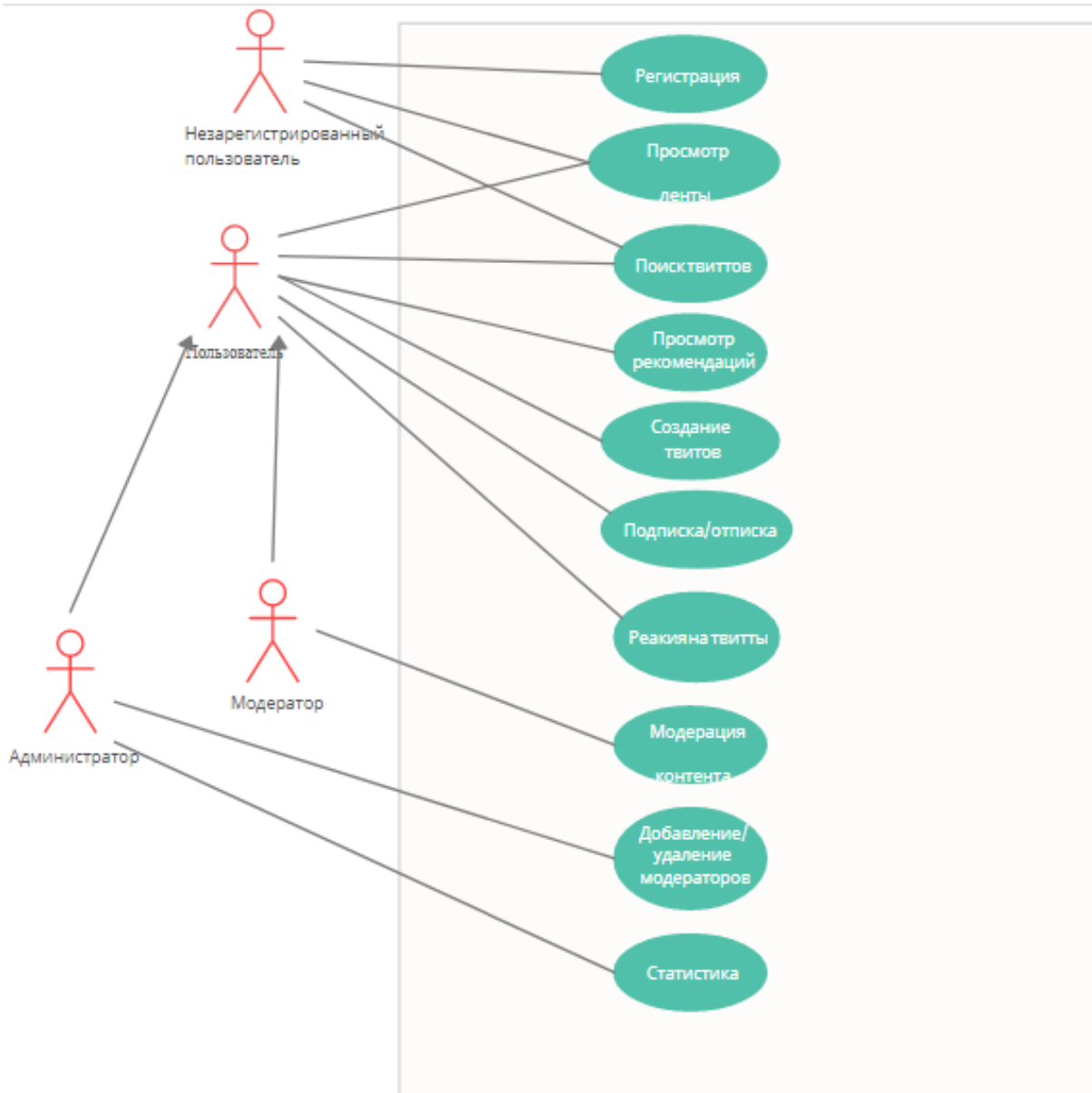


Рисунок 4 – Диаграмма прецедентов

Диаграммы прецедентов предназначены для показа функций действующих лиц системы (актеров). В данной системе есть четыре типа актеров — незарегистрированный пользователь, зарегистрированный пользователь, модератор и администратор.

Функции незарегистрированного пользователя:

— регистрация,

- просмотр ленты,
- отклик на заявку,
- поиск твитов.

Функции зарегистрированного пользователя:

- просмотр ленты,
- поиск твитов,
- просмотр рекомендаций,
- создание твитов,
- подписка/отписка,
- реакция на твиты.

Функция модератора (который является пользователем с расширенными правами доступа):

- модерация контента.

Функция администратора (который является пользователем с расширенными правами доступа):

- добавление/удаление модераторов,
- просмотр статистики.

4.2 Общая структура системы

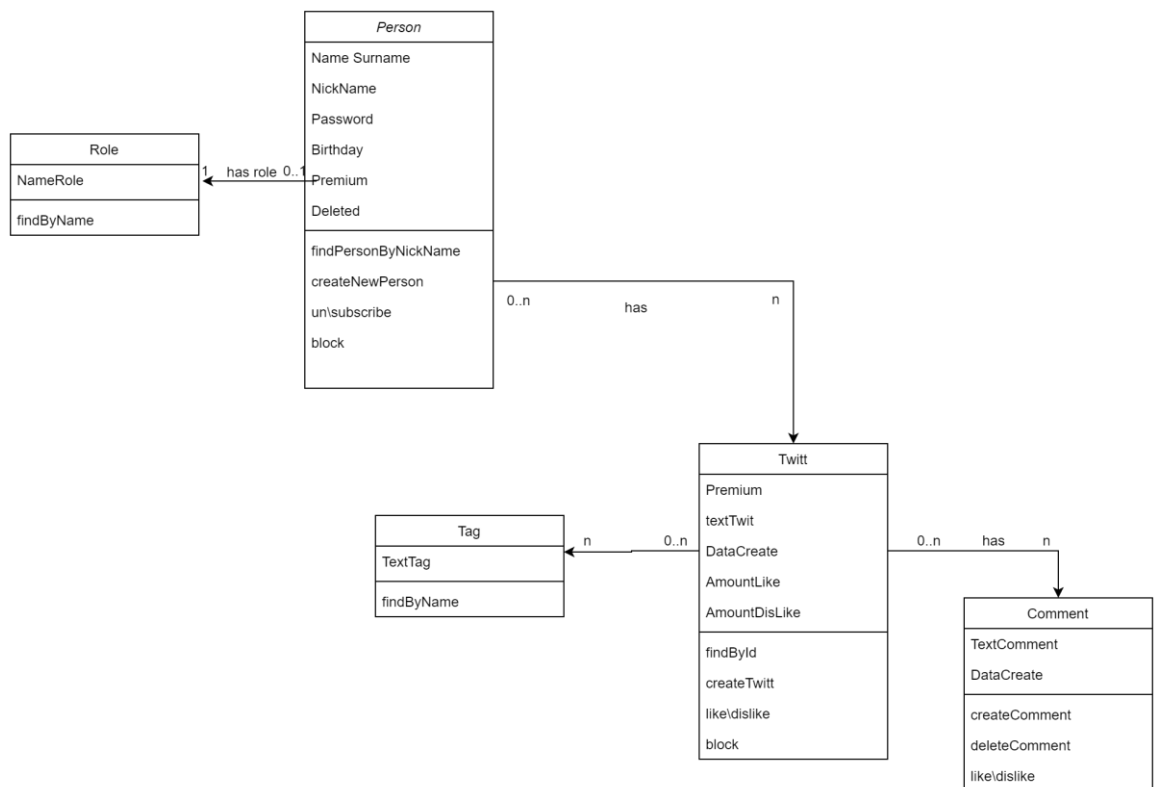


Рисунок 5 – Диаграмма классов.

На рисунке 5 изображена диаграмма классов. Она показывает набор классов, методов и связи между ними.

1. Класс «Role» – представляет собой отражение сущности роль. Класс имеет следующие свойства:

— «NameRole» – название роли.

2. Класс «Person» – представляет собой отражение сущности человек.

Класс имеет следующие свойства:

— «Name Surname» – имя и фамилия,

— «NickName» – никнейм,

— «password» – пароль,

— «Birthday» - день рождения пользователя,

— «Premium» - статус пользователя,

— «Deleted» - удален ли данный пользователь.

3. Класс «Tag» – представляет собой отражение сущности тег. Класс имеет следующие свойства:

— «TextTag» – текст тега.

4. Класс «Twitt» – представляет собой отражение сущности твит. Класс имеет следующие свойства:

— «Premium» – статус твита,

— «textTwit» – текст твита,

— «DataCreate» – дата создания твита,

— «AmountLike» – количество лайков,

— «AmountDisLike» – количество дизлайков.

5. Класс «Comment» – представляет собой отражение сущности заявка группы. Класс имеет следующие свойства:

— «TextComment» – текст комментария,

— «DateCreate» – дата создания комментария.

4.3 Конфигурация системы

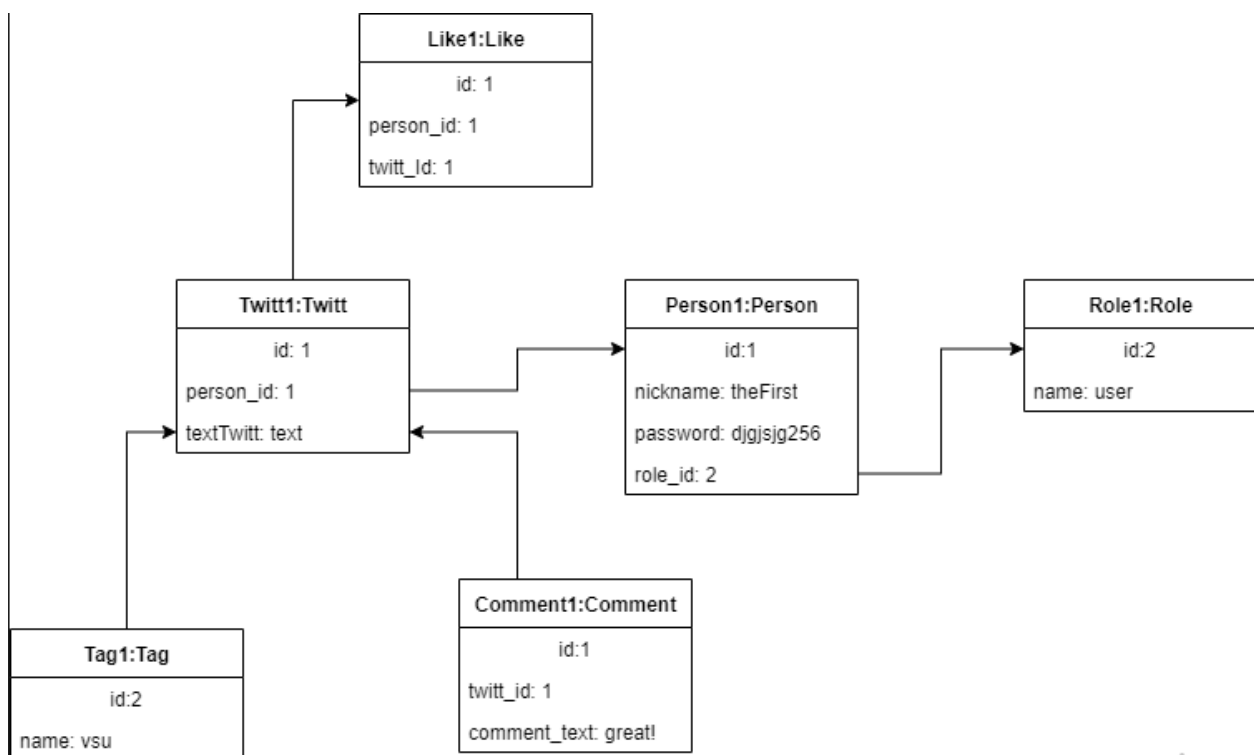


Рисунок 6 – Диаграмма объектов.

На Рисунке 6 изображена диаграмма объектов, которая отражает множество экземпляров классов и отношений между ними в некоторый момент времени. На ней изображён экземпляр класса “Человек”, экземпляр класса “Роль”, экземпляр класса “Твит”, экземпляр класса “Тег”, экземпляр класса “Лайк”, экземпляр класса “Комментарий”. На диаграмме показано их связь.

4.4 Взаимодействие компонентов системы

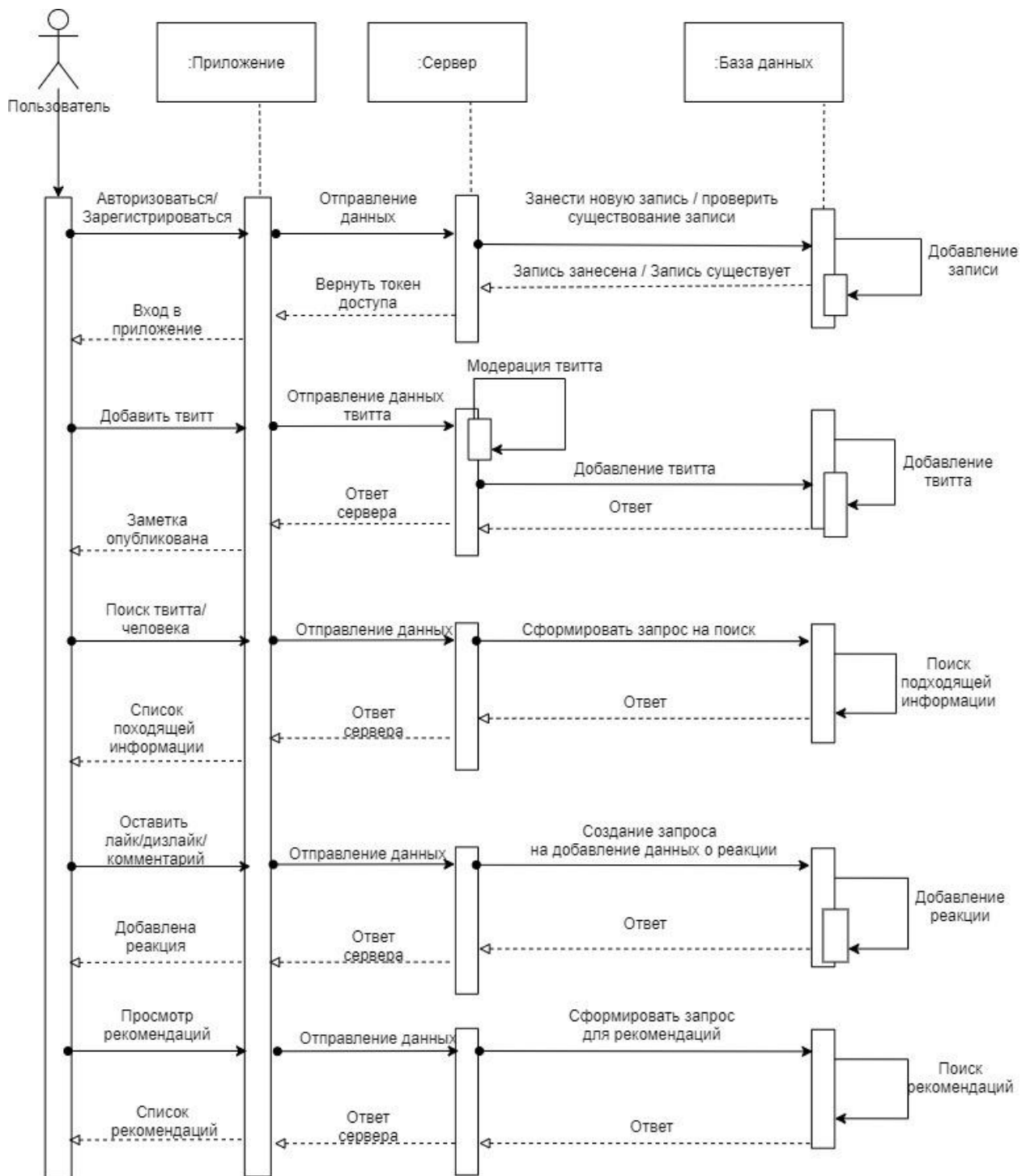


Рисунок 7 – Диаграмма последовательностей гостя.

На следующих рисунках изображены диаграммы последовательности для ролей пользователей системы, на которых изображены упорядоченные во времени взаимодействия объектов.

4.5 Взаимодействие в системе

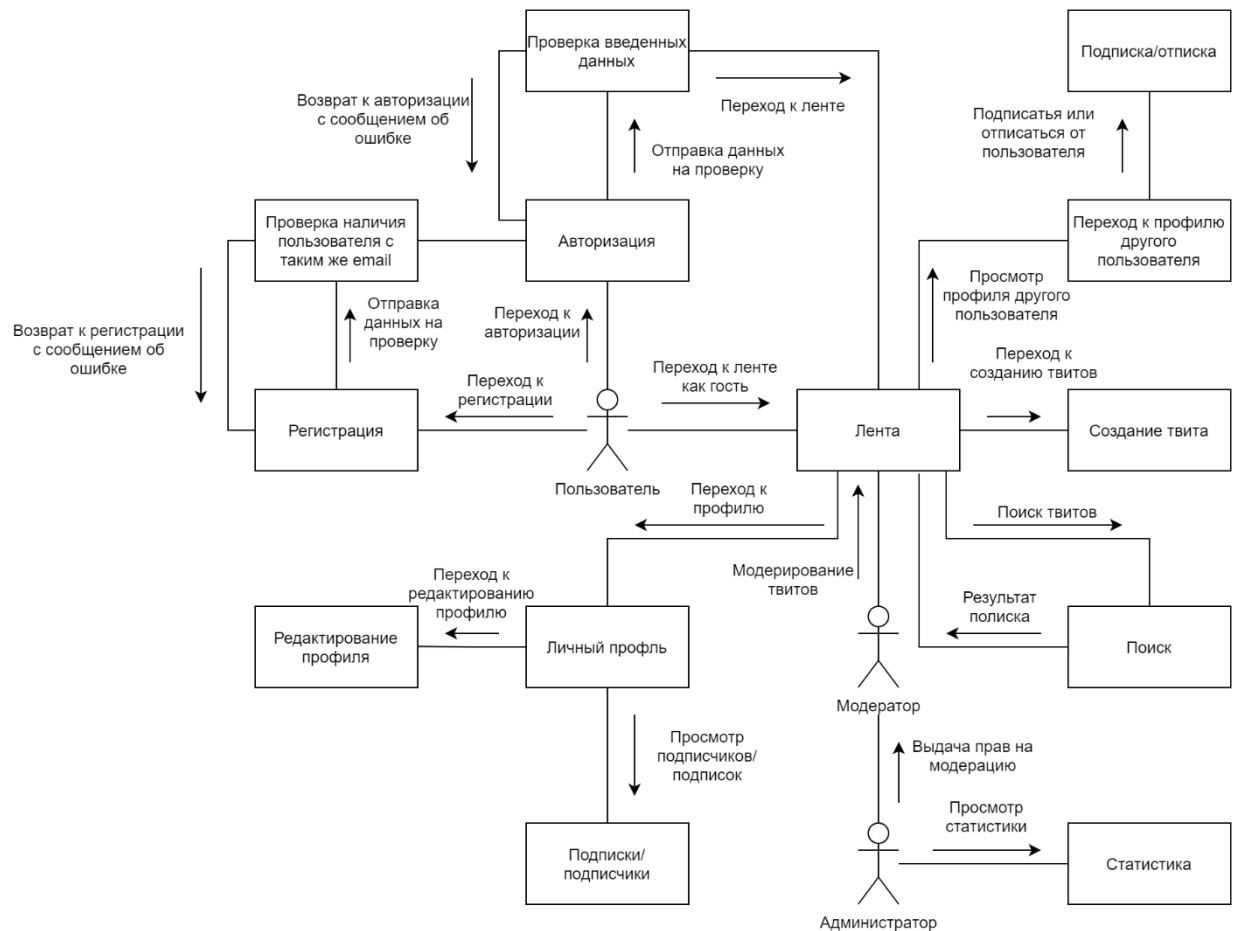


Рисунок 8 – Диаграмма взаимодействия.

На Рисунке 8 представлена диаграмма взаимодействий. Она отражает возможные действия пользователя и системы. Зайдя в систему, пользователь может продолжить как гость и перейти к просмотру ленты. Или пройти авторизацию, и получить все доступные пользователю возможности. Если пользователь не зарегистрирован он может пройти регистрацию. После успешной авторизации пользователь может просматривать ленту твитов, просматривать персонализированную ленту твитов, создавать новые твиты, осуществлять поиск твитов. Так же пользователь может перейти в свой профиль, где может отредактировать свой профиль и просмотреть своих подписчиков и подписки. Модератор просматривает твиты, в случае нарушения правил удаляет нарушившие правила твиты. Администратор просматривает статистику и так же выдает права модераторам.

4.6 Варианты состояния системы



Рисунок 9 – Диаграмма состояний.

Диаграмма состояний, изображенная на Рисунке 9, отражает возможные состояния при входе в приложение. При входе в приложение у пользователя иметься две возможности: зарегистрироваться или авторизоваться. После этих действий происходит проверка полученных данных и возвращается ответ.

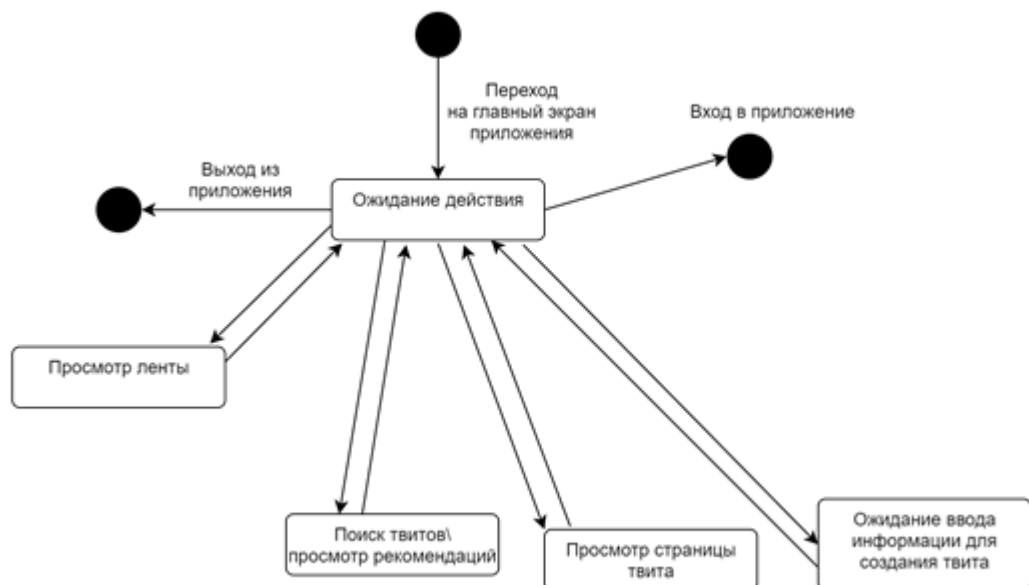


Рисунок 10 – Диаграмма состояний.

Диаграмма состояний, изображенная на Рисунке 10, отражает возможные состояния пользователя на главном экране приложения.

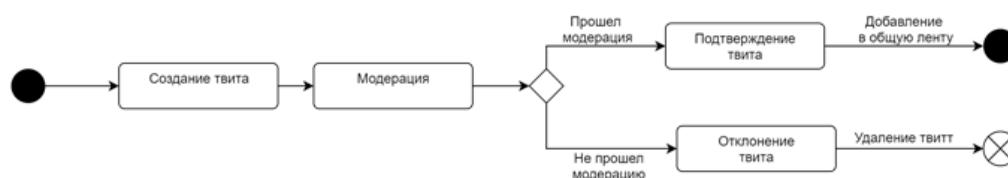


Рисунок 11 – Диаграмма состояний.

Диаграмма состояний, изображенная на Рисунке 11, отражает возможные состояния твита. После создания твит попадает на модерацию. Далее твит может перейти в два состояния: подтверждённый твит и затем добавление его в общую ленту и отклоненный твит, что приведет к его удалению.

4.7 Действия с системой

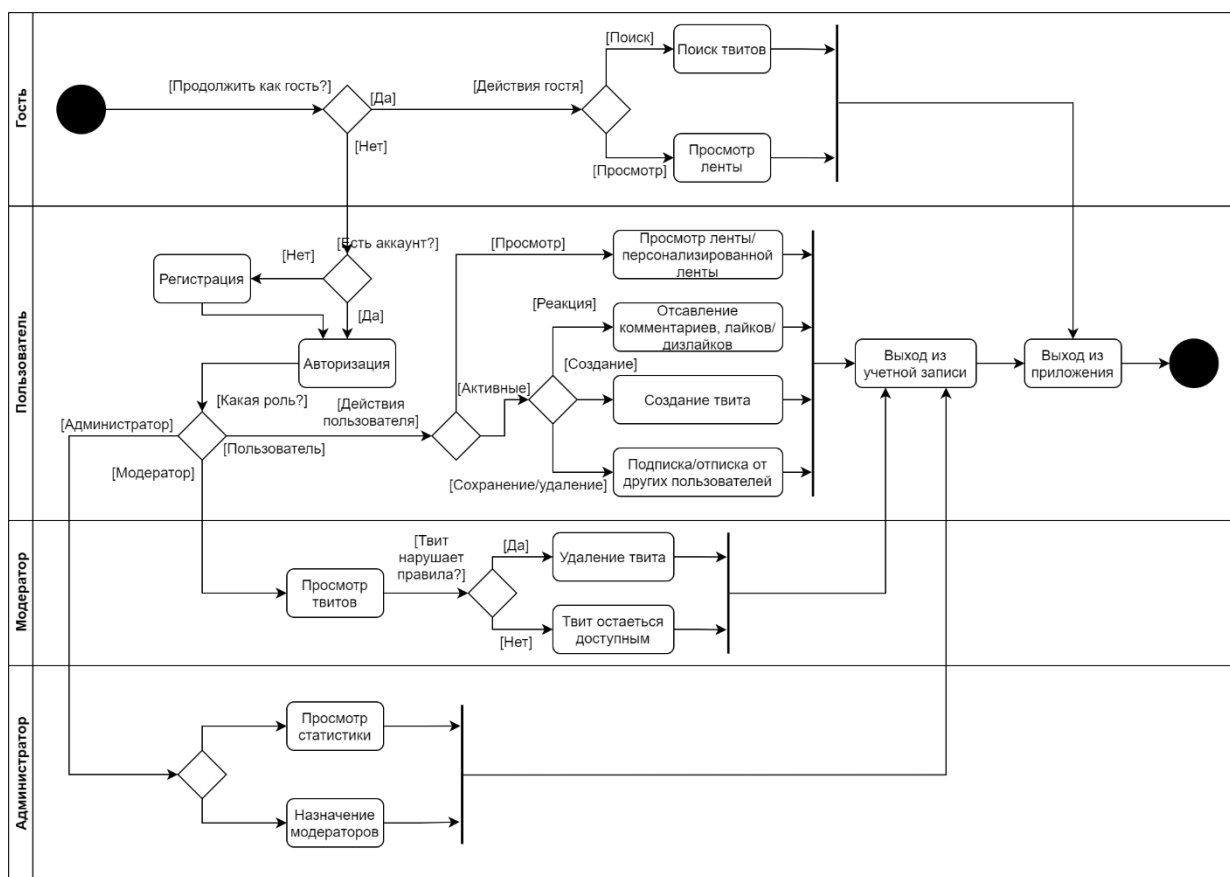


Рисунок 12 – Диаграмма активностей.

Диаграмма активности, изображенная на Рисунке 12 отражает действия пользователей в системе. При входе в систему пользователь может пройти авторизацию и продолжить пользование со всеми доступными для его роли возможностями или продолжить как гость с возможностью просмотра твитов и их поиска, без возможности взаимодействовать с ними. В случае если пользователь еще не зарегистрирован он может пройти регистрацию. При авторизации пользователи разделяются на три роли: пользователь, модератор, администратор. Каждая роль обладает своими возможностями и правами. Пользователь имеет право просматривать ленту твитов, а также персонализированную ленту твитов. Так же пользователь может создавать новые твиты, ставить лайки твитам и оставлять комментарии, подписываться и отписываться от других пользователей. Модератор может просматривать твиты пользователей и в случае нарушения правил удалить этот твит.

Администратор может просматривать статистику, а также давать и удалять права модераторов.

4.8 Развертывание системы

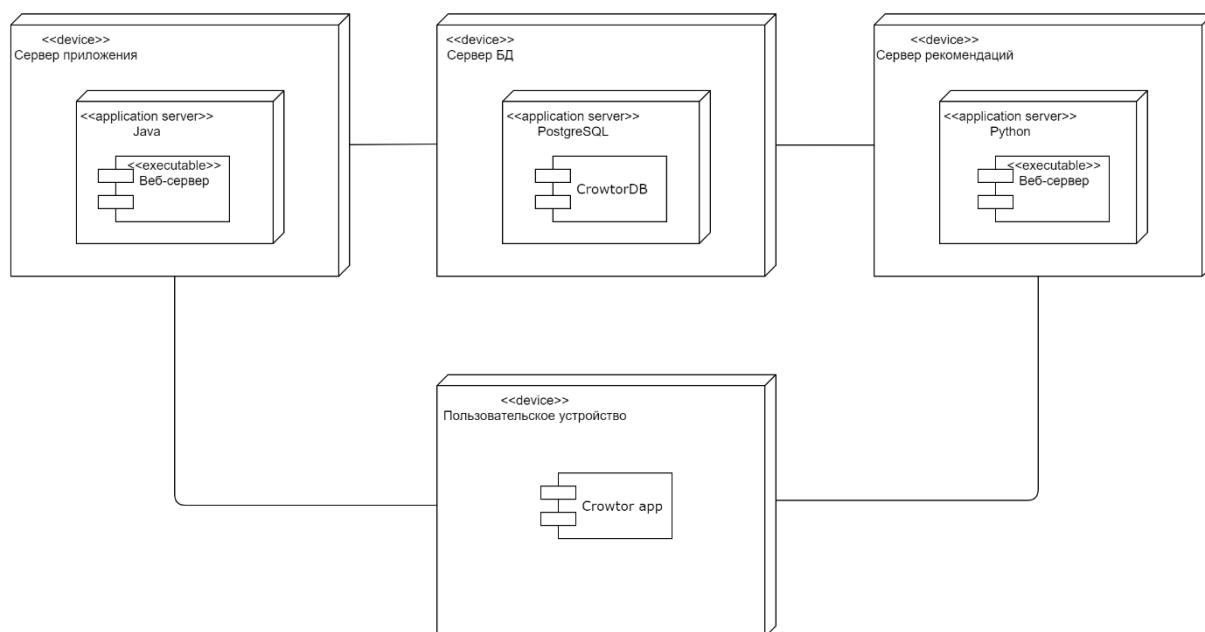


Рисунок 13 – Диаграмма развертывания.

Диаграмма развертывания, изображенная на рисунке 13 показывает топологию системы и распределение компонентов по ее узлам, а также соединения – маршруты передачи информации.

4.9 IDEF0

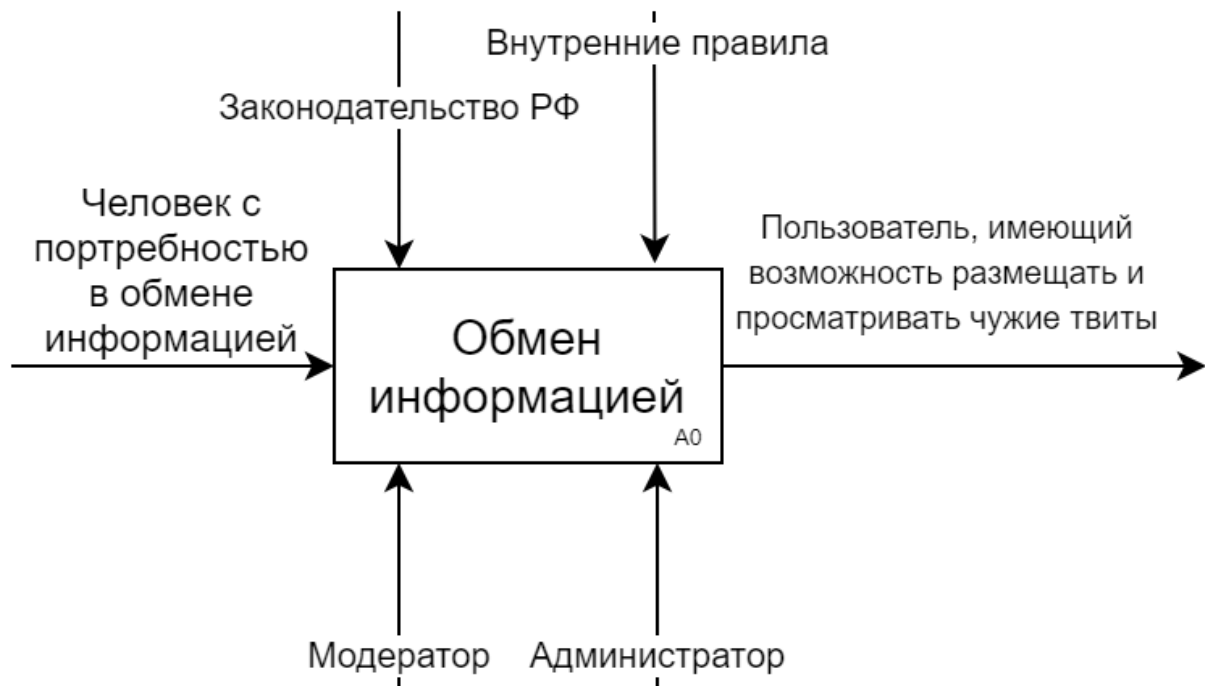


Рисунок 14 – IDEF0

На рисунке 14 изображена IDEF0 диаграмма. Работу системы регулируют законодательство РФ и внутренние правила. Работу системы обеспечивают Модератор и Администратор. На входе в систему поступает человек с потребностью в обмене информацией. На выходе пользователь имеющий возможность размещать свои твиты и просматривать чужие.

4.10 ER-диаграмм

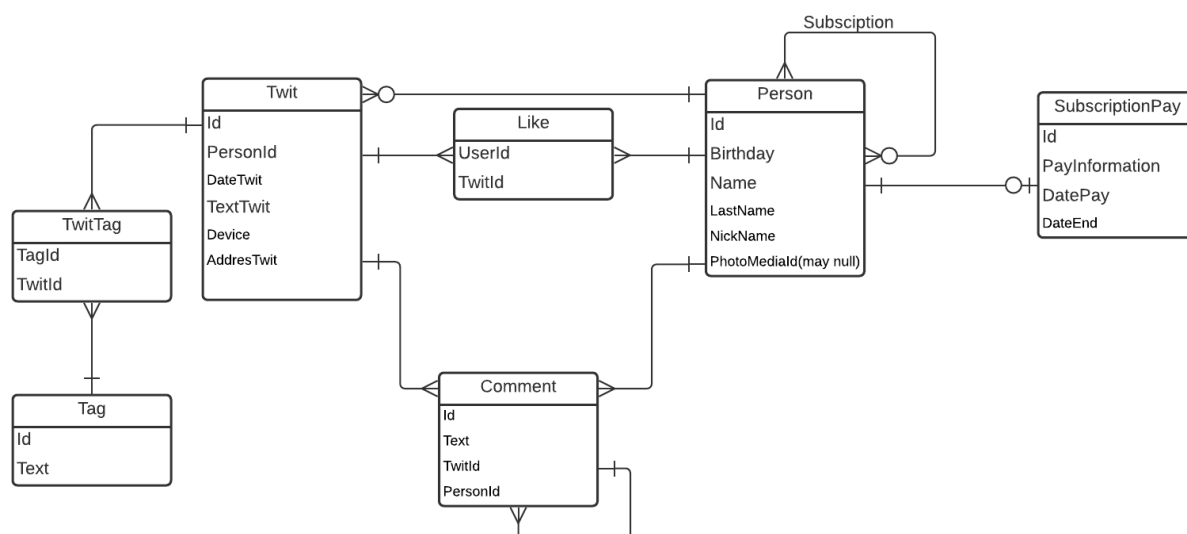


Рисунок 15 – ER-диаграмма

На рисунке 15 изображена ER-диаграмма, схема «сущность-связь», показывающая, как связаны между собой «сущности» внутри системы.

5 Анализ средств реализации

В качестве средств реализации приложения были выбраны следующие технологии:

- В качестве языка разработки в front-end использовался язык Dart и фреймворк Flutter. Язык Dart является достаточно современным языком программирования, получившим современные возможности. Фреймворк Flutter позволяет создавать кроссплатформенные приложения для IOS и Android. Так же данный фреймворк позволяет создавать эффективные приложения, схожие по производительности с нативными приложениями.

- В качестве СУБД была выбрана PostgreSQL. Данная СУБД поддерживает неограниченные размеры БД, что является важным для системы с потенциально большими объемами данных. Так же важным являться ее надежность и отказоустойчивость.

– В качестве языка разработки в back-end использовался язык Java и фреймворк Spring. Java является одним из самых популярных языков программирования в back-end разработке. Так же позволяет создавать высоконагруженные и отказоустойчивые системы. Spring является самым популярным фреймворком для Java, благодаря наличию большого количества модулей значительно ускоряет разработку.