

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	4
1 Постановка задачи	5
2 Глоссарий	6
3 Анализ предметной области	7
3.1 Анализ существующих решений	7
4 Анализ задачи	10
4.1 Варианты использования системы	10
4.2 Общая структура системы	12
4.3 Конфигурация системы	13
4.4 Взаимодействие компонентов системы	15
4.5 Взаимодействие в системе	16
4.6 Варианты состояния системы	17
4.7 Действия с системой	19
4.8 Развертывание системы	20
4.9 IDEF0	21
4.10 ER-диаграмм	22
5 Анализ средств реализации	23
6 Сценарии воронок конверсии	24
7 Графический интерфейс мобильного приложения	25
7.1 Стартовые экраны	25
7.2 Экраны авторизации	25
7.3 Экран ленты	26
7.4 Экран поиска	27
7.5 Экран профиля	28
7.6 Экран создания твитта	29
7.7 Экран профиля пользователя	30
7.8 Экран модерации твиттов	31
8 Тестирование	33
8.1 UI тестирование	33

8.2	Дымовое тестирование	34
8.3	Юзабилити тестирование	35
Заключение		37

Введение

Потребности людей в получении и распространении информации увеличиваются с каждым годом. Ежедневно происходит огромное количество различных событий: от котят, бегающих за лазерной указкой, до полетов людей в космос. Сейчас становится очень тяжело уследить за тем, что важно и интересно конкретному пользователю, так как эта информация разбросана по всему интернету. 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 твитты, комментировать их и иметь постоянный доступ с мобильного телефона, значительно упрощает и делает процесс получения информации намного удобнее.

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

- размещение своих “твиттов”
- просмотр “твиттов” других пользователей
- взаимодействие с “твиттами” 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

Данный проект является приложением, позволяющим пользователям осуществлять обмен информацией.

Система автоматизирует процесс размещения, получения и взаимодействия с информацией.

Система предназначена для:

- Получения информации
- Размещения информации
- Взаимодействия 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

Твитт – 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 твиттов 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 нем публикуется 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. Также в нашем постоянно меняющемся мире информация обновляется каждую секунду и актуальная информация может стать нерелевантной, а возможность редактировать твитты 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).

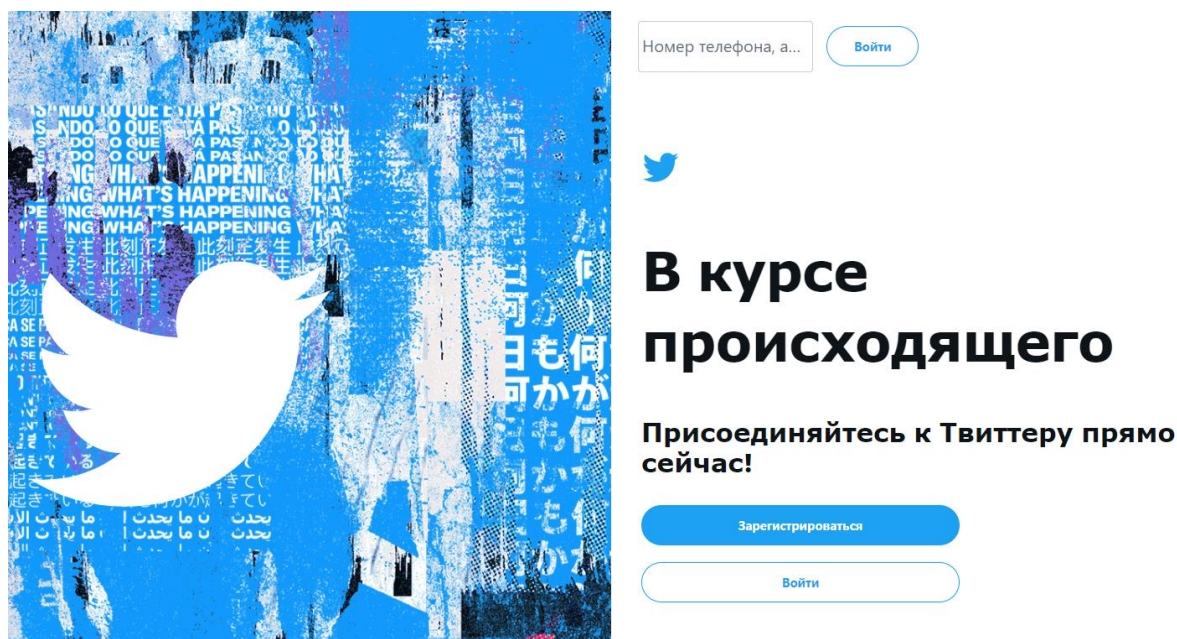


Рисунок 1 – Домашняя страница социальной сети Twitter

Реддит (Reddit)

Является крупным информационно-новостным сайтом. Имеет обширную, сплоченную аудиторию. Также 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.

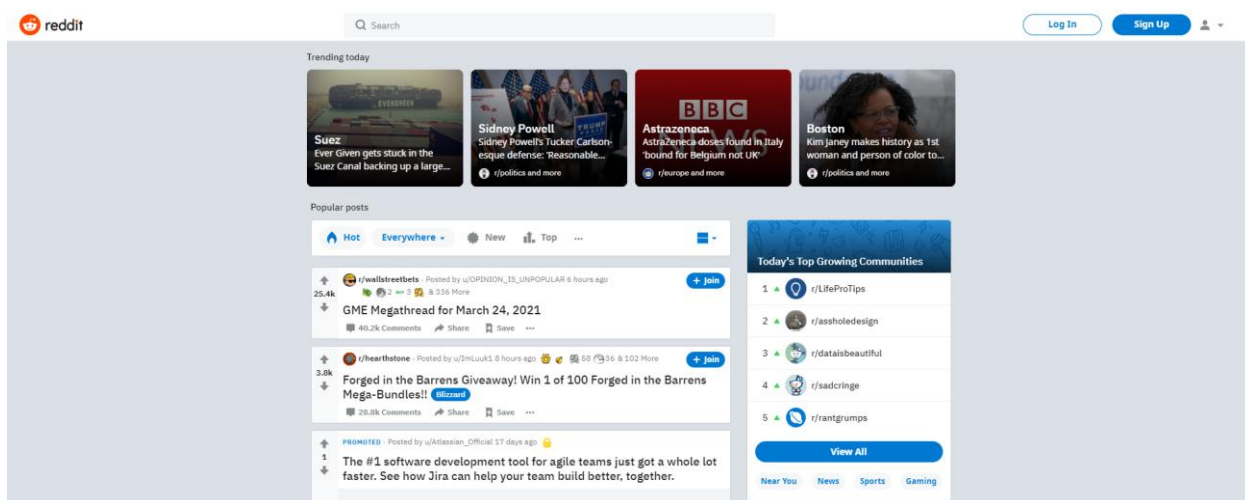


Рисунок 2 – Домашняя страница сервиса Reddit

Инстаграм (Instagram)

Является приложением для обмена фотографиями и видеозаписями с элементами социальной сети, позволяющим 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. Также, как и Твиттер, Инстаграм не имеет возможности выражать недовольство

материалами путем выставления дизлайков. Кроме того, также требуется регистрация для начала использования приложения.

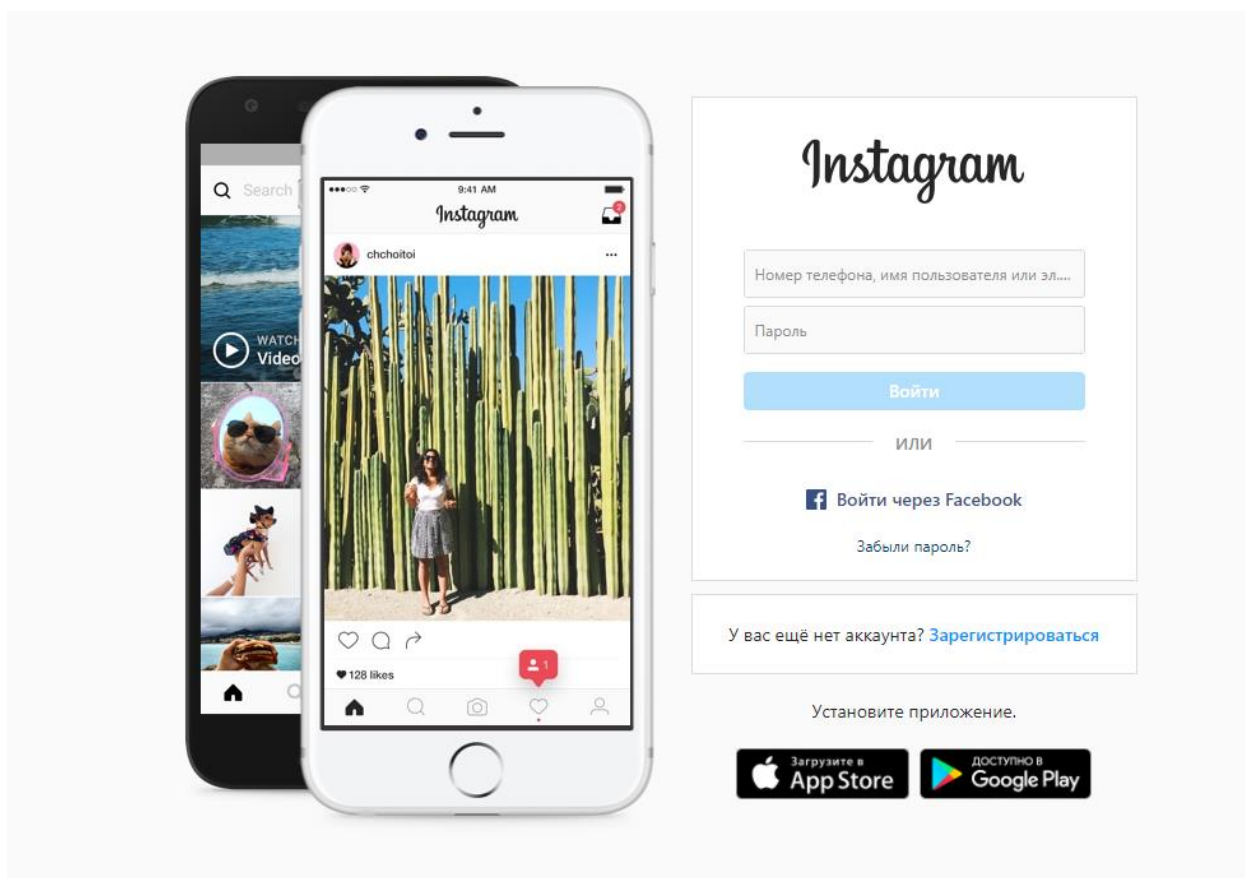


Рисунок 3 – Домашняя страница сервиса Instagram

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

4 Анализ задачи

4.1 Варианты использования системы

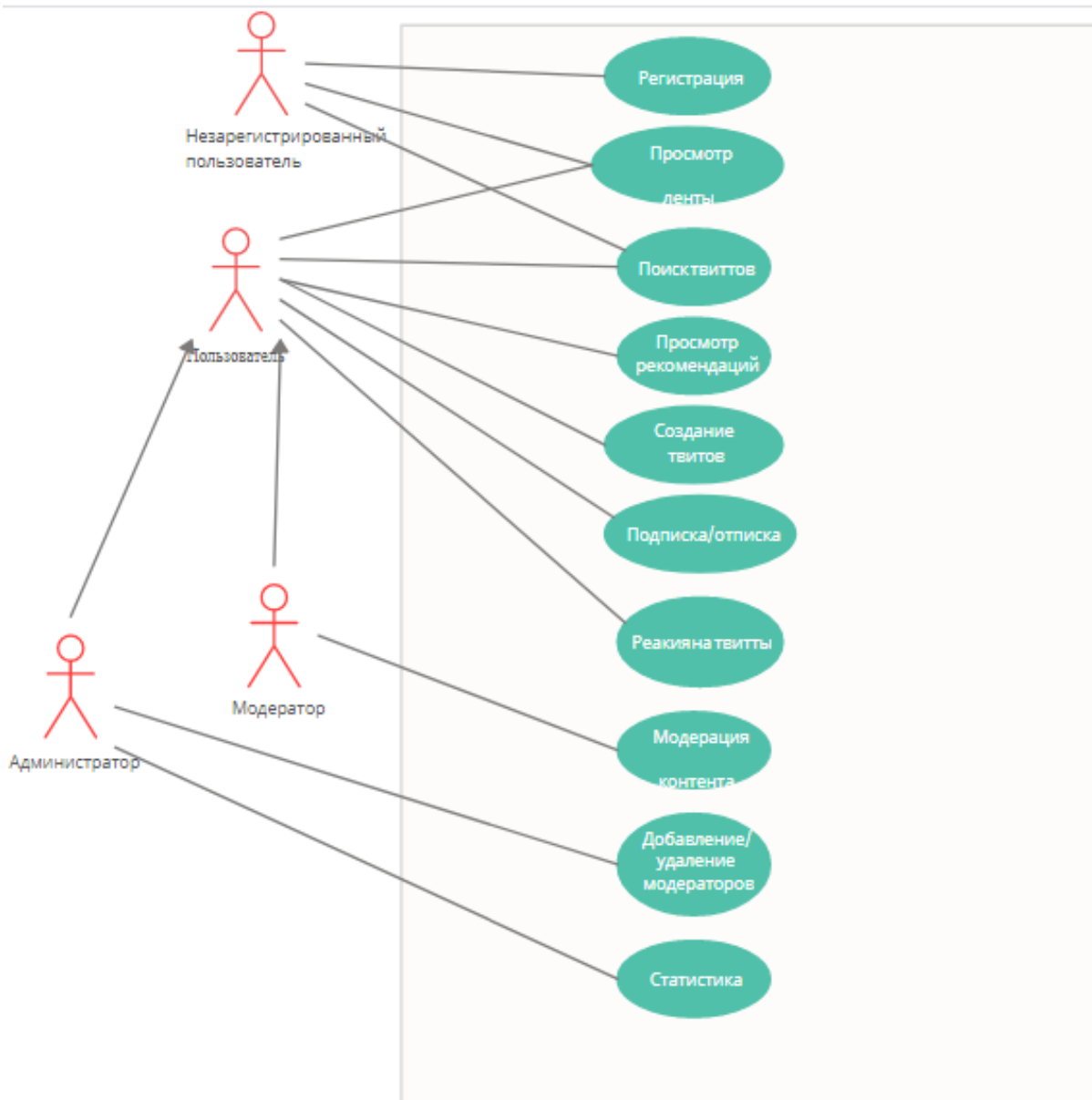


Рисунок 4 – Диаграмма прецедентов

Диаграмма прецедентов предназначена 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 твиттов.

Функции зарегистрированного пользователя:

- просмотр ленты,
- поиск твиттов,
- просмотр рекомендаций,
- создание твиттов,
- подписка/отписка,
- реакция на твитты.

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

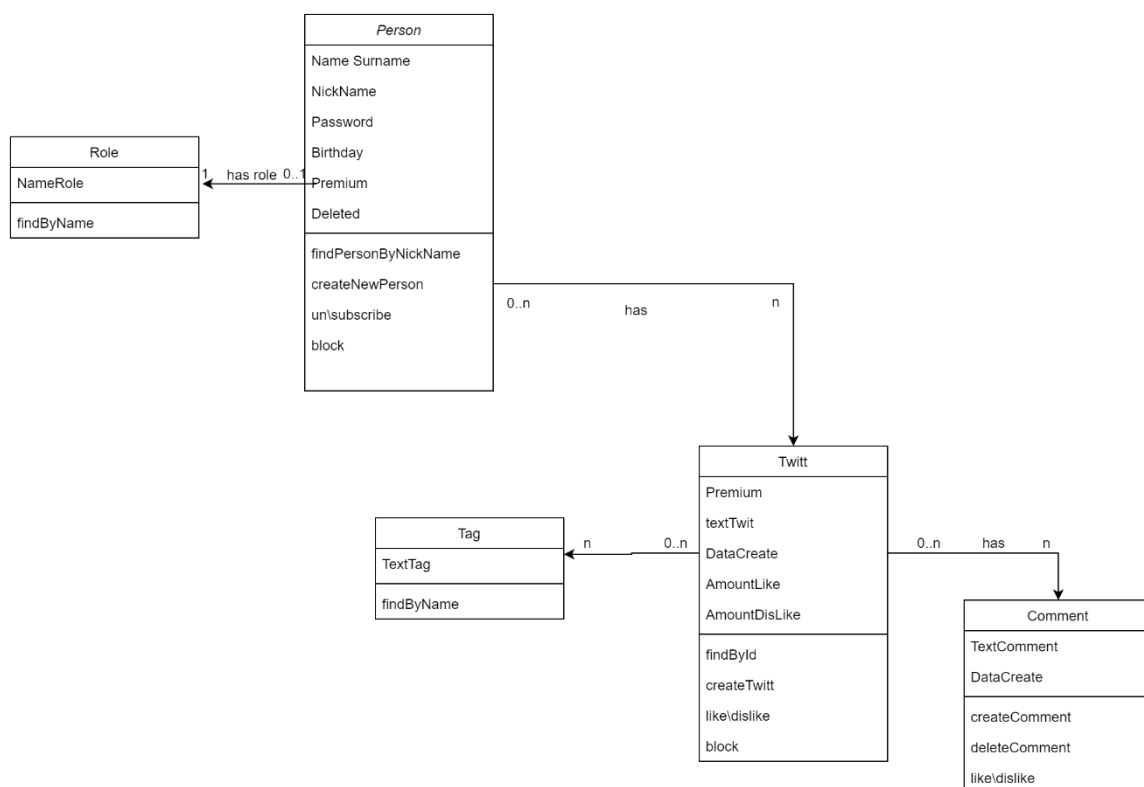


Рисунок 5 – Диаграмма классов.

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 твитт. Класс имеет следующие свойства:

— «Premium» – статус твитта,

— «textTwit» – текст твитта,

— «DataCreate» – дата создания твитта,

— «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

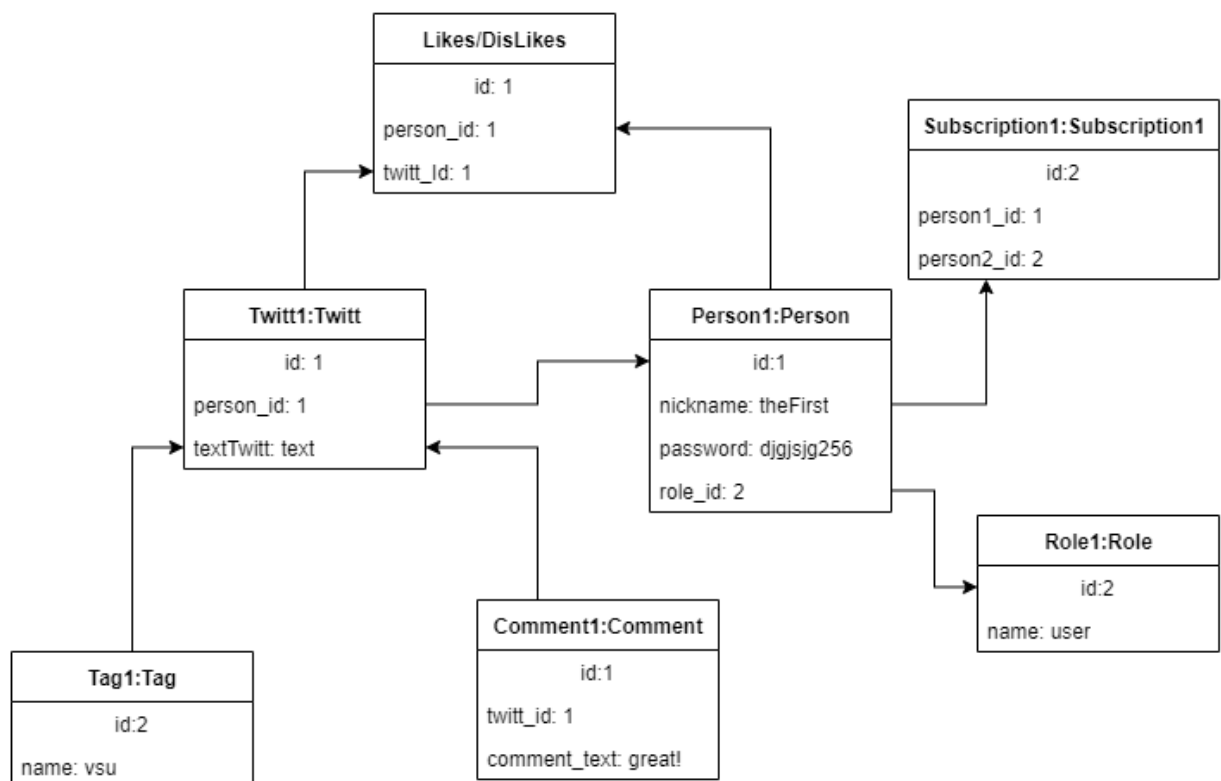


Рисунок 6 – Диаграмма объектов.

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 “Твитт”, экземпляр класса “Тег”, экземпляр класса “Лайк”, экземпляр класса “Комментарий”. На диаграмме показана их связь.

4.4 Взаимодействие компонентов системы

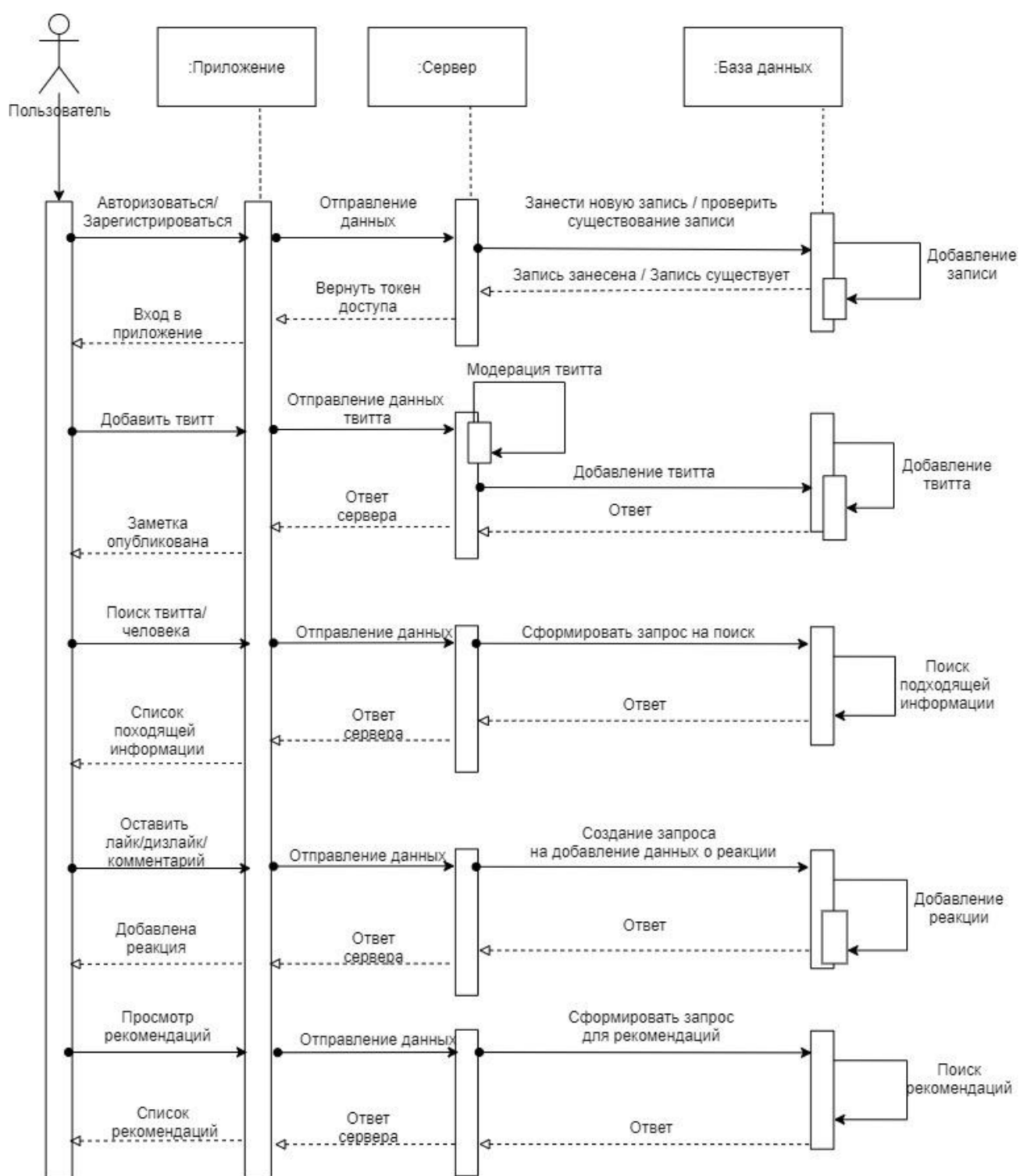


Рисунок 7 – Диаграмма последовательностей гостя.

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

4.5 Взаимодействие в системе

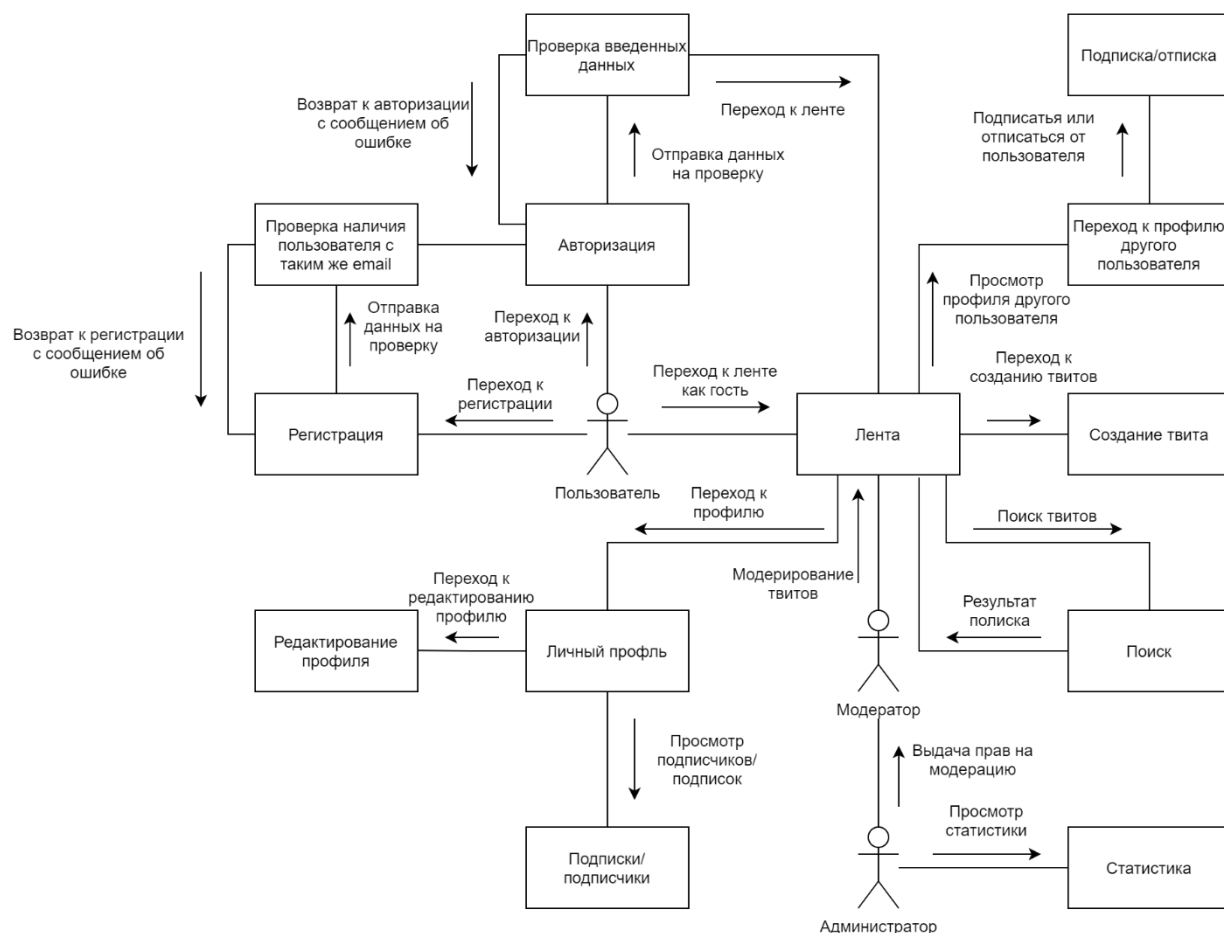


Рисунок 8 – Диаграмма взаимодействия.

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 твиттов, просматривать персонализированную ленту твиттов, создавать новые твитты, осуществлять поиск твиттов. Также пользователь может перейти в свой профиль, где может отредактировать его и просмотреть своих подписчиков и подписки. Модератор просматривает твитты и в случае нарушения правил удаляет нарушившие правила твитты. Администратор просматривает статистику и также 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 может либо зарегистрироваться, либо 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.

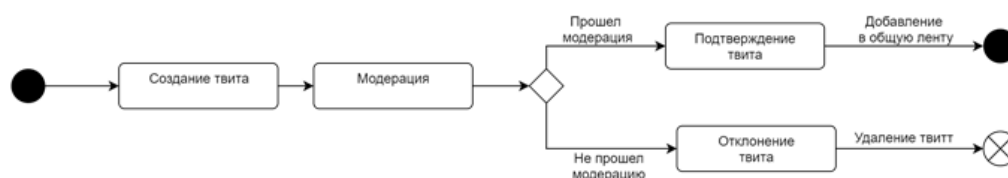


Рисунок 11 – Диаграмма состояний.

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

4.7 Действия с системой

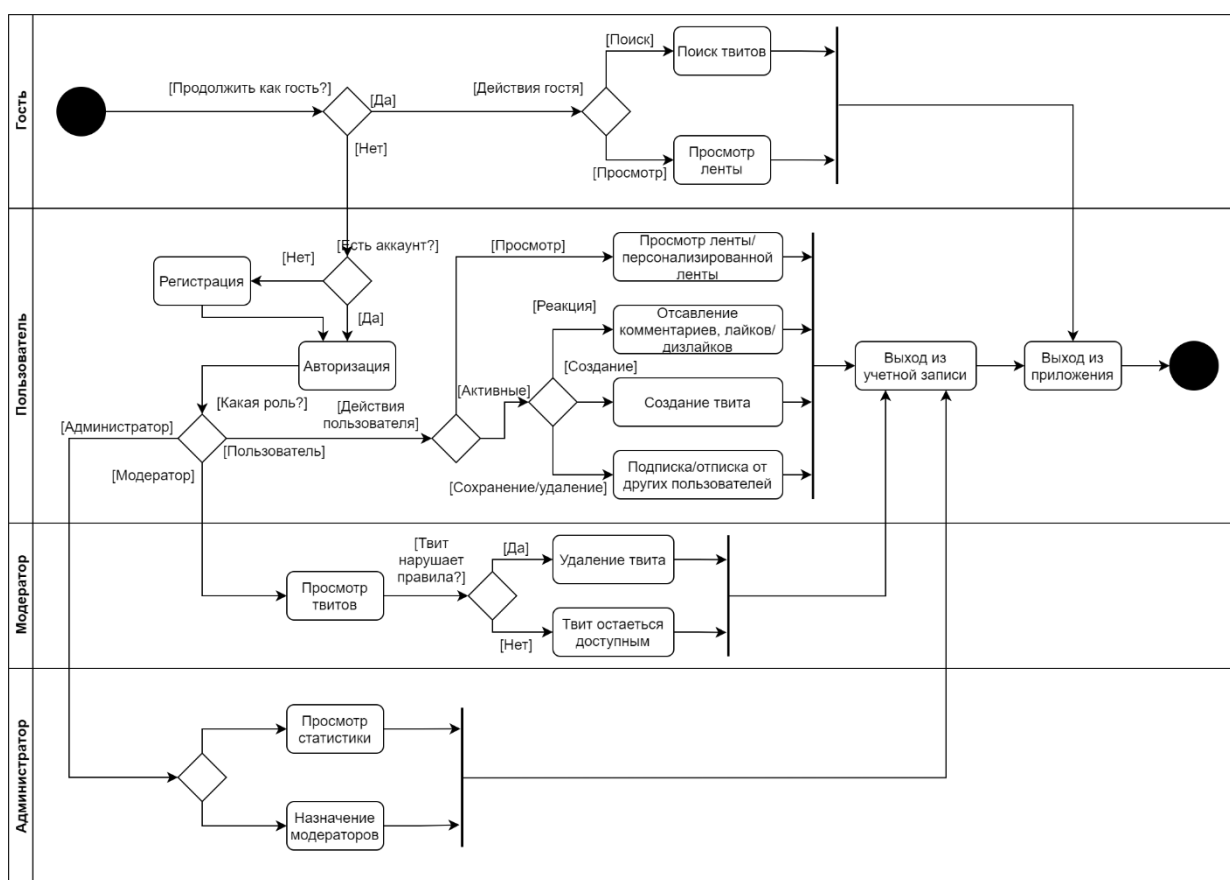


Рисунок 12 – Диаграмма активностей.

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 твиттов 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 твиттов, а также персонализированную ленту твиттов. Также пользователь может создавать новые твитты, ставить лайки твиттам и оставлять комментарии, подписываться и отписываться от других пользователей. Модератор может

просматривать твитты пользователей и в случае нарушения правил удалять некоторые. Администратор может просматривать статистику, а также давать и удалять права модераторов.

4.8 Развертывание системы

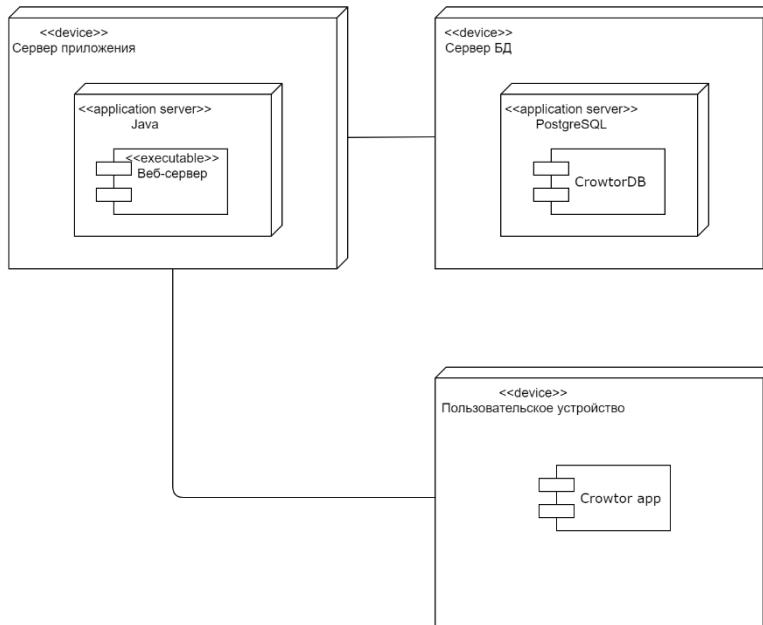


Рисунок 13 – Диаграмма развертывания.

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

4.9 IDEF0

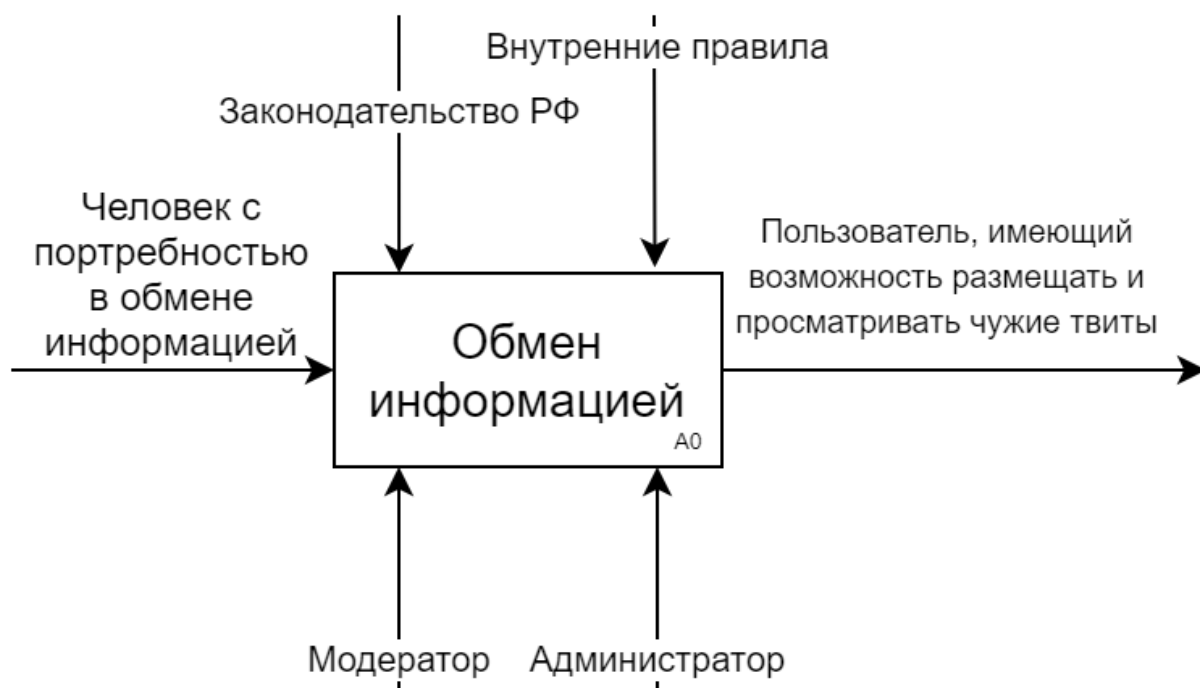


Рисунок 14 – IDEF0

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 твитты и просматривать чужие.

4.10 ER-диаграмм

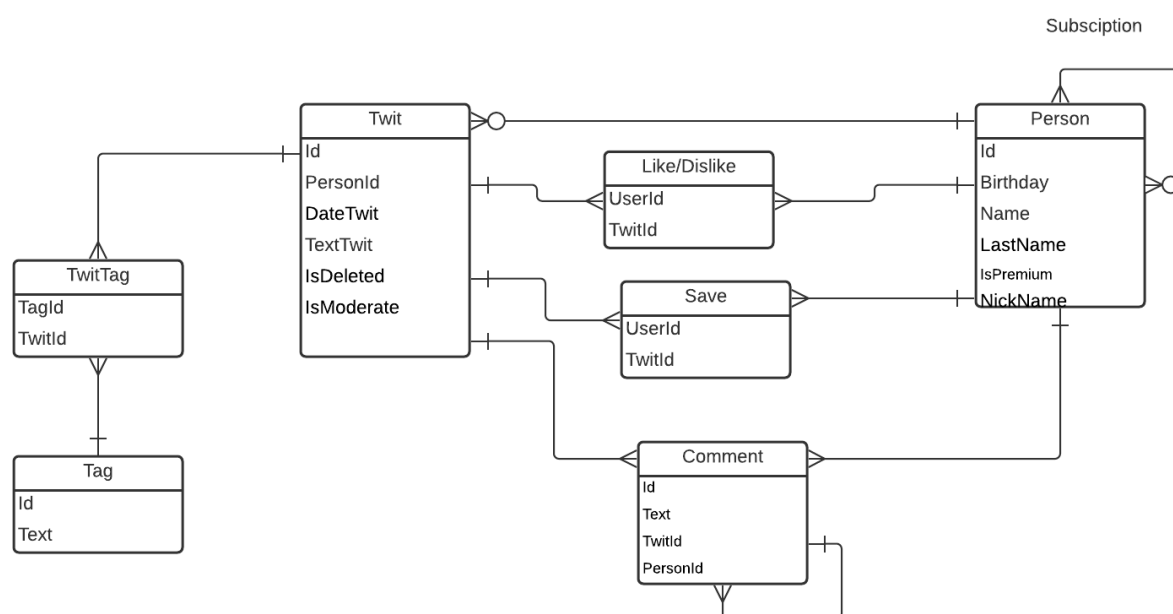


Рисунок 15 – ER-диаграмма

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. Кроме того, важным является 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 и благодаря наличию большого количества модулей значительно ускоряет разработку.

6 Сценарии воронок конверсии

- 1) Зашел в приложение – авторизовался – переход в ленту – создание твитта – публикация
- 2) Зашел в приложение – авторизовался – переход в ленту – переход в профиль
- 3) Зашел в приложение – авторизовался – переход в ленту – оставил реакцию на твитт

7 Графический интерфейс мобильного приложения

7.1 Стартовые экраны

При входе в приложение, пользователи встречают первоначальные приветствия с общей информацией о приложении.

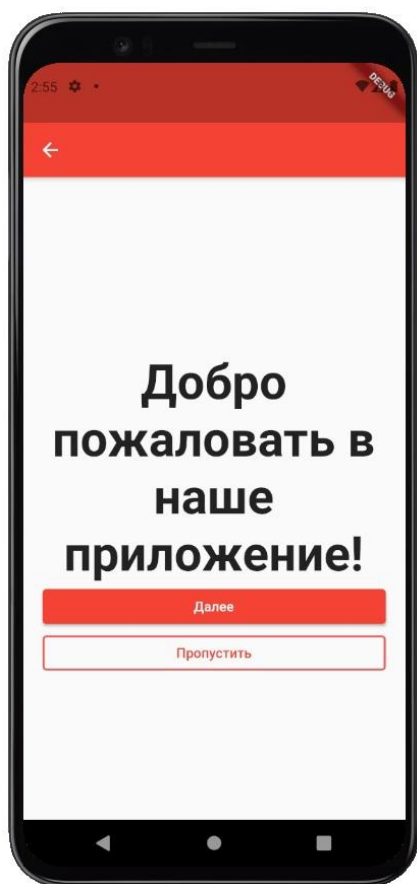


Рисунок 16 – Первый экран информации



Рисунок 17 – Экран

7.2 Экраны авторизации

При авторизации у пользователя есть выбор, либо войти в существующий аккаунт, пройти регистрацию или продолжить пользоваться приложением как гость.

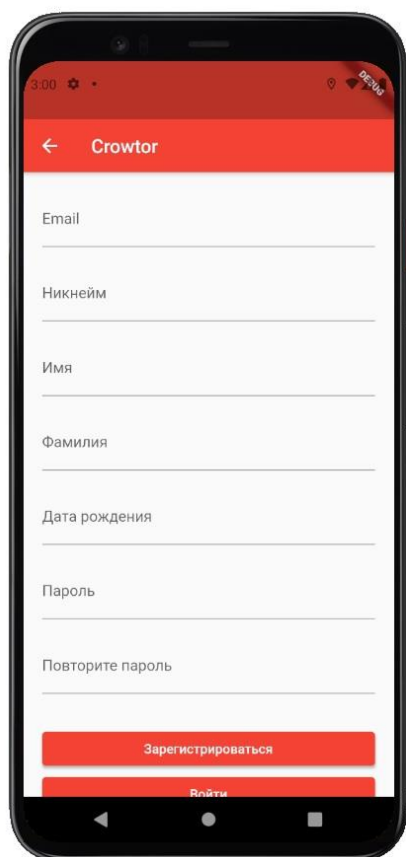


Рисунок 19 – Экран регистрации

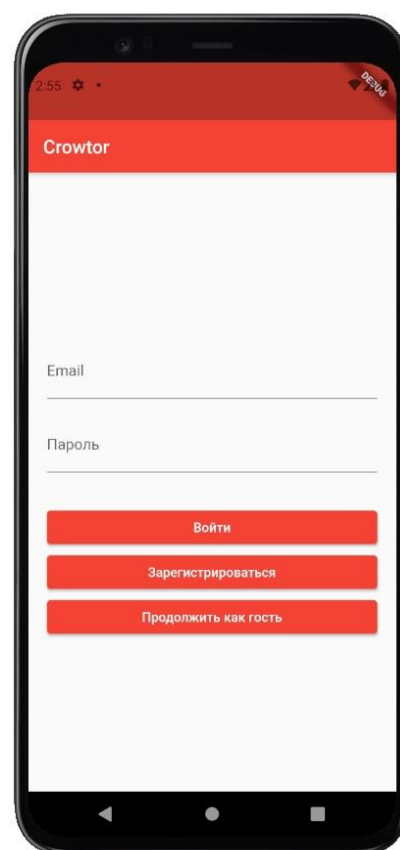


Рисунок 20 – Экран авторизации

7.3 Экран ленты

После прохождения этапа входа, пользователь попадает в ленту твиттов. Она отличается у авторизованного и неавторизованного пользователя тем, что у первого твитты только тех людей, на которых он подписан (если подписок нет, выводится общая лента). Каждый твитт характеризуется следующими полями:

- Имя создателя
- Никнейм создателя
- Текст самого твитта

Авторизованный может проводить следующие действия:

- Поставить лайк
- Поставить дизлайк
- Прокомментировать
- Переслать себе твитт
- Добавить твитт в сохраненные

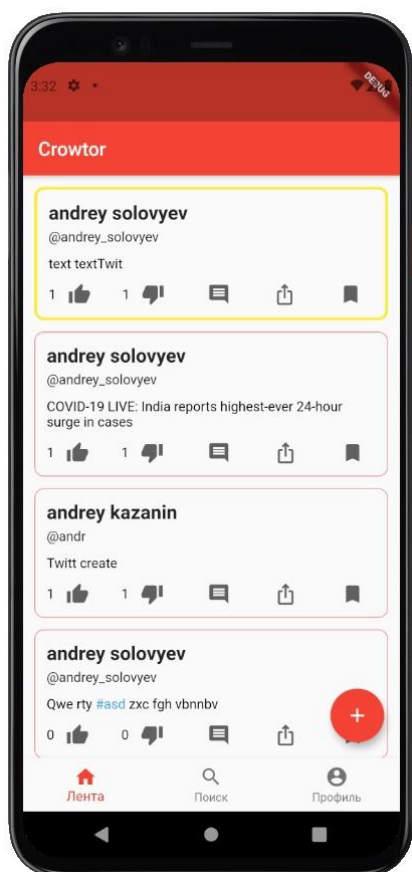


Рисунок 21 – Экран ленты

7.4 Экран поиска

На данном экране пользователь может произвести поиск людей по их никнейму, либо найти твитты по названию. Весь найденный материал будет отображаться ниже формы поиска. Если производится поиск по людям, можно нажать на выбранного пользователя и перейти на его профиль. Форма отображения твиттов, описана в пункте 6.3.

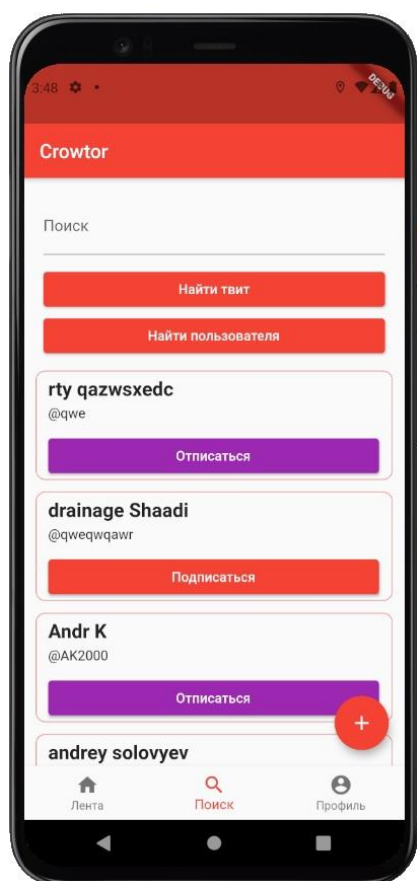


Рисунок 22 – Экран поиска

7.5 Экран профиля

При переходе в свой профиль, эта функция доступна только авторизованным пользователям, потребитель приложения увидит информацию, введенную при регистрации:

- Имя и фамилию
- Никнейм
- Дата регистрации
- Дата рождения
- Количество подписчиков
- Количество подписок

Также пользователь имеет следующие возможности:

- Просмотр своих твиттов в формате, отображенном в пункте 6.3.
- Просмотр твиттов, где он поставил лайк
- Просмотр сохраненных твиттов

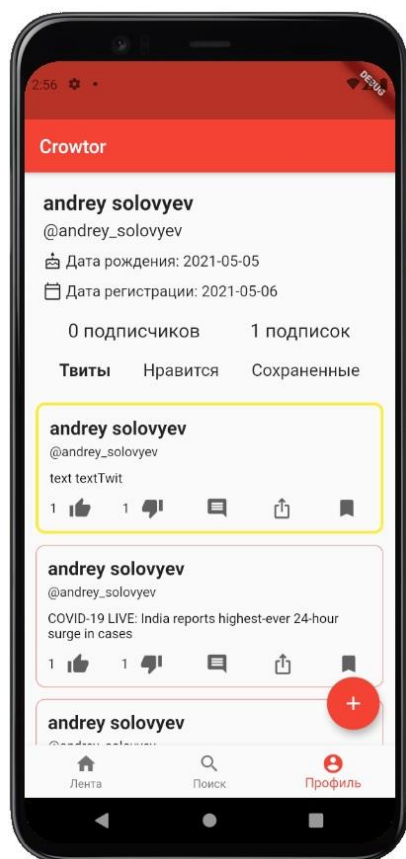


Рисунок 23 – Экран профиля пользователя

7.6 Экран создания твитта

При нажатии на иконку плюс в правой нижней части, пользователь попадет на экран создания твитта. Там имеется текстовое поле, в которое и нужно ввести твитт.

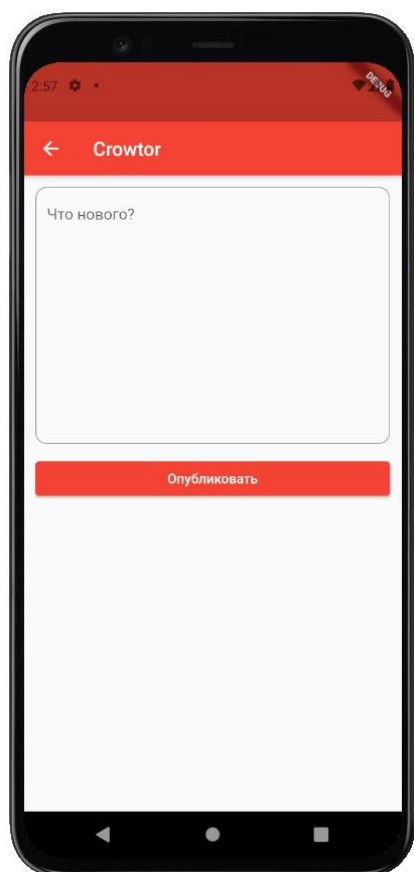


Рисунок 24 – Экран создания твитта

7.7 Экран профиля пользователя

Данный экран показывает нам, как выглядит профиль стороннего пользователя при переходе на него в приложении.

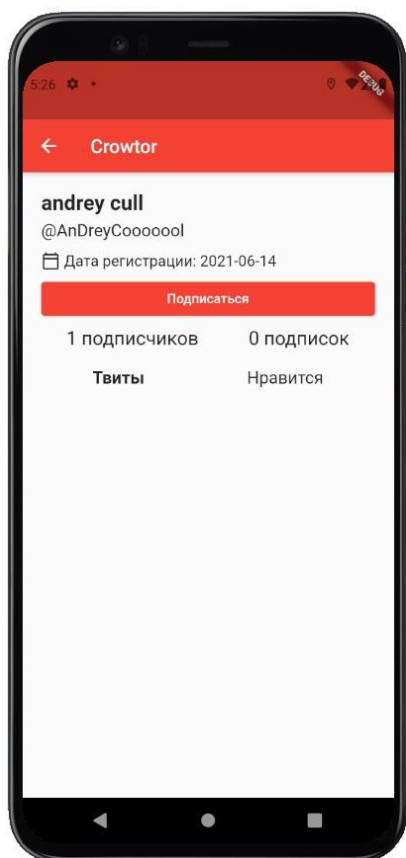


Рисунок 25 – Профиль стороннего пользователя

7.8 Экран модерации твиттов

К данному экрану имеют доступ только модераторы и администраторы. Они могут разрешать публикацию постов или отклонять ее. Все твитты даже без модерации сохраняются.

8 Тестирование

8.1 UI тестирование

Таблица 1 – UI тесты

Шаги теста	Ожидаемый результат	Статус
1. Нажать далее на стартовом экране	Переход к следующему стартовому экрану	Пройден
2. Нажать пропустить на стартовом экране	Переход к экрану авторизации	Пройден
3. Нажать начать на последнем стартовом экране	Переход к экрану авторизации.	Пройден
4. Нажатие кнопки войти на экране авторизации	Вход в приложение и переход к ленте	Пройден
5. Нажатие кнопки регистрации на экране авторизации	Переход к экрану регистрации	Пройден
6. Нажатие кнопки продолжить как гость на экране авторизации	Переход к ленте с ограниченными возможностями	Пройден
7. Нажатие кнопки зарегистрироваться на экране регистрации	Регистрация нового пользователя, сообщение об успешной регистрации	Пройден
8. Нажатие кнопки вход на экране регистрации	Переход к экрану авторизации	Пройден
9. Нажатие кнопки плюс в правом углу экрана	Переход к экрану создания твитта	Пройден
10. Нажатие кнопки опубликовать на экране создания твитта	Публикация твитта	Пройден
11. Нажатие кнопки опубликовать на экране создания твитта (гость)		Пройден
12. Нажатие стрелочки в левой верхней части экрана	Переход к предыдущему экрану	Пройден
13. Нажатие кнопки лента в нижней части экрана	Переход к экрану ленты	Пройден
14. Нажатие кнопки поиск в нижней части экрана	Переход к экрану поиска	Пройден
15. Нажатие кнопки профиль	Переход к профилю пользователя	Пройден
16. Нажатие кнопки профиль (гость)	Переход к экрану авторизации	Пройден

17. Нажатие кнопки найти твитт на экране поиска	Поиск твиттов	Пройден
18. Нажатие кнопки найти пользователя на экране поиска	Поиск пользователей	Пройден
19. Нажатие кнопки подписаться	Подписка на пользователя, изменение кнопки подписаться на кнопку отписаться	Пройден
20. Нажатие кнопки отписаться	Отписка от пользователя, изменение кнопки подписаться на кнопку подписаться	Пройден
21. Нажатие на никнейм пользователя	Переход к профилю пользователя	Пройден
22. Нажатие на твитт	Переход к твитту	Пройден
23. Нажатие кнопки лайк	Добавление лайка к твитту, изменение цвета иконки	Пройден
24. Нажатие кнопки дизлайк	Добавление дизлайка к твитту, изменение цвета иконки	Пройден
25. Нажатие кнопки комментариев	Переход к твитту	Пройден
26. Нажатие кнопки сохранить	Сохранение твитта	Пройден
27. Нажатие кнопки модерация	Переход к экрану модерация	Пройден
Нажатие кнопки выход	Выход из аккаунта пользователя, переход к экрану авторизации	Пройден

8.2 Дымовое тестирование

Таблица 2 - Результаты дымового тестирования

Сценарий	Результат
Регистрация	Пройден
Авторизация	Пройден
Создание заметки	Пройден
Одобрение/удаление твитта	Пройден
Добавление лайка/дизлайка к твитту	Пройден

Добавление комментария к твитту	Пройден
Подписаться на / отписаться от пользователя	Пройден
Получение профиля авторизованного пользователя	Пройден
Поиск пользователя по никнейму	Пройден
Поиска твитта по тексту	Пройден

8.3 Юзабилити тестирование

Для того, чтобы провести юзабилити тестирование, создадим 3 новых профиля. Целью данного тестирования является проверка основных сценариев использования приложения. Отообразим данные в таблице.

Таблица 1 - Юзабилити тестирования

Сценарий	Пользователь 1	Пользователь 2	Пользователь 3
Регистрация	Пройден	Пройден	Пройден
Авторизация	Пройден	Пройден	Пройден
Переход в профиль	Пройден	Пройден	Пройден
Создание твитта	Пройден	Пройден	Пройден
Добавление лайка/дизлайка	Пройден	Пройден	Пройден
Поиск по никнейму	Пройден	Пройден	Пройден

Подписаться на другого пользователя	Пройден	Пройден	Пройден
Отписаться от пользователя	Пройден	Пройден	Пройден
Поиск твитта	Пройден	Пройден	Пройден
Выход из аккаунта	Пройден	Пройден	Пройден

Заключение

В ходе реализации проекта был произведен анализ сервисов, предоставляющий похожий функционал с приложением твиттер. На рынке существует несколько больших приложений, предоставляющий подобный функционал. По сути, каждый из них занимает определенную нишу. Пользователь на основе своих личных предпочтений, отзывов и функционала выбирает наиболее комфортный для него сервис.

В результате были проделаны следующие шаги:

- Разработана база данных, расположенная на удаленном сервере.
- Разработана Back-end часть приложения, которая была развернута на удаленном сервере. Она предоставляет REST-API для работы.
- Разработано мобильное приложение с помощью Flutter.
- Создана связь между Back-end и мобильным приложением.
- Разработаны и пройдены тесты