

Федеральное государственное образовательное бюджетное
учреждение высшего образования
«Финансовый университет при Правительстве Российской Федерации»
(Финансовый университет)

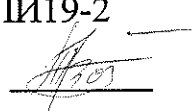
Факультет информационных технологий и анализа больших данных
Департамент анализа данных и машинного обучения

Выпускная квалификационная работа
на тему: «Разработка программного обеспечения статистического анализа
социальных сетей»

Направление подготовки 09.03.03 - Прикладная информатика,
Профиль: ИТ-сервисы и технологии обработки данных в экономике и финансах

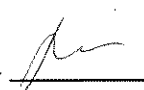
Выполнила студентка группы ПИ19-2

Волкова Татьяна Алексеевна



Руководитель к.ф.-м.н., доцент

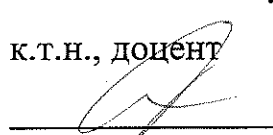
Беспалова Наталья Викторовна



**ВКР соответствует предъявляемым
требованиям**

Руководитель департамента анализа данных
и машинного обучения,

к.т.н., доцент



Д.А.Петросов

«10» мая 2023 г.

Москва 2023

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Анализ предметной области и постановка задачи.....	6
1.1 Выявление особенностей статистического анализа данных социальных сетей.....	6
1.2 Анализ существующих ИТ-решений для анализа социальных сетей	16
1.3 Постановка задачи	20
1.4 Выводы по главе 1.....	22
2. Проектирование приложения.....	23
2.1 Проработка функциональных и нефункциональных требований	23
2.2 Проектирование архитектуры приложения	29
2.3 Выводы по главе 2.....	37
3. Реализация приложения статистического анализа данных социальных сетей	38
3.1 Выбор инструментальных средств разработки программного приложения	38
3.2 Описание реализации приложения	44
3.3 Оценка результатов.....	52
3.4 Выводы по главе 3.....	66
ЗАКЛЮЧЕНИЕ	67
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	70
ПРИЛОЖЕНИЕ	74

ВВЕДЕНИЕ

Социальные сети являются неотъемлемой частью современной жизни. Они не только позволяют людям поддерживать связь между собой, но и являются источником огромного объема пользовательских данных, которые можно мониторить и анализировать. Благодаря социальным сетям люди могут поделиться своими мыслями, фотографиями, видео, музыкой и другой информацией, а также получать обратную связь от других пользователей. Кроме того, социальные сети — это не только место для общения, но и для бизнеса. Большое количество людей использует социальные сети для продажи своих товаров и услуг, привлечения новых клиентов и оценки конкурентов в социальных сетях. Некоторые компании используют социальные сети для маркетинга и рекламы своих продуктов и услуг. Рекламодателям необходимо понимать, какие сообщества подходят для рекламы. В связи с этим возникает потребность в ПО, которое будет отображать метрики по сообществам в социальных сетях (например, среднее количество лайков, репостов, комментариев и т. д.), а также анализировать аудиторию сообществ (пол, возраст, город и т. д.) и визуализировать данную информацию. Такое приложение может оказаться незаменимым инструментом для малого бизнеса и стартапов, которые не могут позволить себе дорогие аналитические решения. Благодаря полученным данным они смогут оценить качество контента конкурентов и своего собственного, а также скорректировать свою продукцию или группу в социальной сети, сделать её более привлекательной для потенциальных клиентов и, в конечном итоге, стать успешнее. Более того, приложение будет полезно и для специалистов, занимающихся рекламой, которые благодаря ему смогут более эффективно размещать свои объявления в сообществах, наиболее интересных для целевой аудитории.

Самой популярной социальной сетью в России на данный момент является ВКонтакте, в связи с чем объектом исследования данной работы являются сообщества социальной сети ВКонтакте, предметом — анализ и визуализация их данных.

Основная цель данной работы – облегчение поиска подходящего для рекламы сообщества за счет создания алгоритмического и программного обеспечения статистического анализа и визуализации статистики сообществ социальной сети Вконтакте.

В рамках достижения поставленной цели необходимо решить следующие задачи:

1. Исследование предметной области;
2. Изучение способов получения данных соцсетей;
3. Аналитический обзор существующих решений по обработке и визуализации данных соцсетей;
4. Определение требований и проектирование приложения;
5. Определение стека технологий и реализация ПО.

Планируется разработать интерфейсную (клиентскую) часть в виде веб-приложения и бэкэнд, на котором будет производиться парсинг данных и их обработка. Далее будет подробно рассмотрена обоснованность выбора технологического стека, а пока лишь перечислю главные технологии, используемые при разработке:

- Python и фреймворк Django для веб-приложения
- VK API и библиотека Python requests для получения данных

Выпускная квалификационная работа состоит из трех глав:

В первой главе описана предметная область, рассмотрены основные характеристики соцсетей; типы данных, производимые ими; способы их анализа и сбора; проанализированы существующие программные продукты для анализа соцсетей, рассмотрены их плюсы и минусы; оценены потребности и поставлены задачи для реализации.

Во второй главе рассматривается важный этап разработки программного продукта - составление бизнес-требований и проектирование архитектуры веб-приложения.

В третьей главе приведено обоснование выбранных технологий для реализации, описан процесс и результат разработки программного продукта - веб-приложения, реализующего базовую функциональность анализа соцсетей.

1. Анализ предметной области и постановка задачи

1.1 Выявление особенностей статистического анализа данных социальных сетей

Под социальной сетью понимается интернет-площадка, сайт, который позволяет зарегистрированным на нем пользователям размещать информацию о себе и коммуницировать между собой, устанавливая социальные связи. Контент на этой площадке создается непосредственно самими пользователями.

Социальные сети — это мощный инструмент для маркетинговых исследований. В них пользователи добровольно делятся своей личной информацией, которая содержит огромное количество данных о них: пол, возраст, профессия, социальный статус, место работы или проживания, интересы, особенности поведения, количество путешествий за определенное время, круг общения, устройства, которыми они пользуются и многое другое. Эта информация может быть использована рекламодателями для более глубокого анализа потребительских предпочтений и адаптации рекламных кампаний к конкретной аудитории. Таким образом, можно определить, какой категории пользователей будет действительно интересно то или иное предложение, что позволит значительно сократить издержки на привлечение потребителей, так как основной рекламный поток будет направлен на целевую аудиторию [10].

Важно отметить, что развитие технологий способствует частичной автоматизации маркетинговых исследований в социальных сетях, что упрощает их проведение и значительно экономит временные и финансовые затраты. Например, данные о пользователях социальных сетей можно легально собирать с помощью парсеров - специализированных сервисов, где доступен удобный поиск и возможность сбора баз пользователей, соответствующих определенным фильтрам.

За последние несколько лет произошел резкий скачок популярности социальных сетей. Начиная с 1996 года, когда появились и получили широкое

распространение социальные сети, количество активных пользователей насчитывало 7,7 миллионов человек. А за последние десять лет - фазу активного развития и роста - эти сервисы увеличили свою аудиторию почти в три раза – с 970 миллионов в 2010 году до 3,81 миллиарда пользователей в 2020 году. Резкий скачок в последние года может быть объяснен пандемией COVID-19: пользователи ушли на удаленку и переместились в онлайн. 2021 год стал паузой, а 2022 — вновь годом резкого роста контента в качестве реакции соцмедиа на общественно-политические события. По данным анализа, проведенного Keros, на начало 2023 года соцсети насчитывают 4,76 миллиарда пользователей, что составляет чуть меньше 60% от общей численности населения мира. В последние месяцы темпы роста аудитории социальных сетей несколько замедлились — прирост новых пользователей за год составил 137 миллионов, то есть всего 3%, но это все ещё выше, чем темп рост численности населения – 0.8% [8].

В России в начале 2023 года насчитывалось 106 миллионов пользователей, что эквивалентно 73,3% от общей численности населения. В целом, в январе 2023 года 83,0% всех пользователей Интернета в России (независимо от возраста) пользовались как минимум одной платформой социальных сетей (рисунок 1.1) [9].

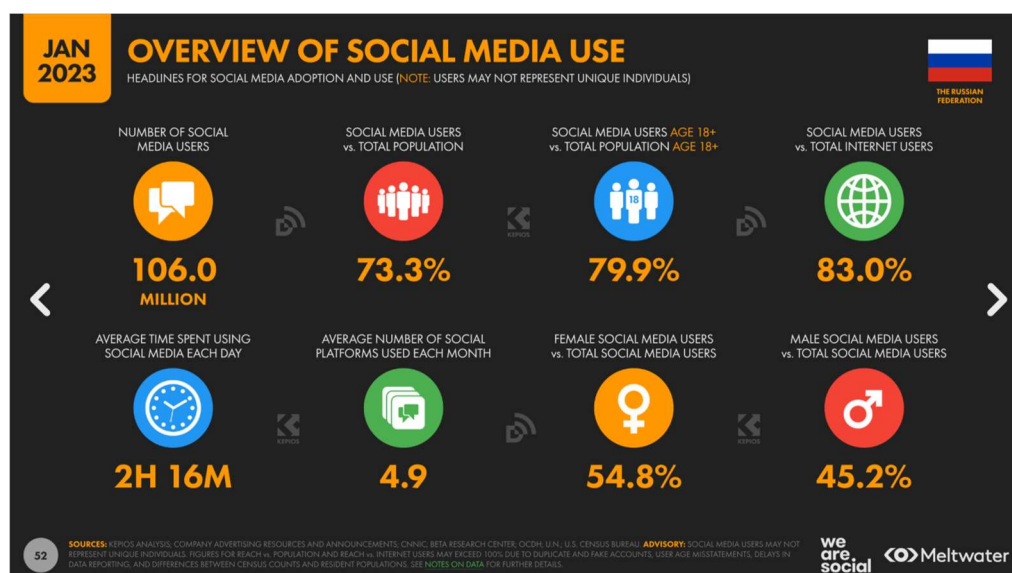


Рисунок 1.1 Отчет Digital 2023: The Russian Federation. Основные цифры использования социальных сетей.

Помимо роста количества пользователей, наблюдается и рост во времени, проводимым людьми в соцсетях. Согласно данным GWI, люди сегодня проводят в социальных сетях больше времени, чем когда-либо. Несмотря на то, что это может показаться незначительным фактом, такой тренд имеет важное значение для маркетологов. Ведь чем больше времени пользователи проводят в соцсетях, тем больше возможностей у рекламодателей привлечь их внимание к своим продуктам и услугам. По результатам последних исследований, среднестатистический пользователь интернета трудоспособного возраста проводит теперь чуть больше 2,5 часов в день на социальные сети.

Говоря о рекламе, исследования Statista показывают, что с 2019 года доля цифровой рекламы в глобальных расходах на рекламу возросла на 27,7%. Если в 2019 году она составляла 57,4%, то в 2022 году она уже достигла 73,3%. Этот рост можно наблюдать на рисунке 1.2, где виден самый большой скачок в 2021 году: глобальные доходы от цифровой рекламы увеличились на треть по сравнению с данными за 2020 год. Несомненно, COVID-19 сыграл значительную роль в перераспределении рекламных бюджетов в мире, и эта ориентация на цифровые каналы, как показывают исследования, сохранилась. В России расходы на охват одного пользователя в 2022 году составили 26,27 доллара.

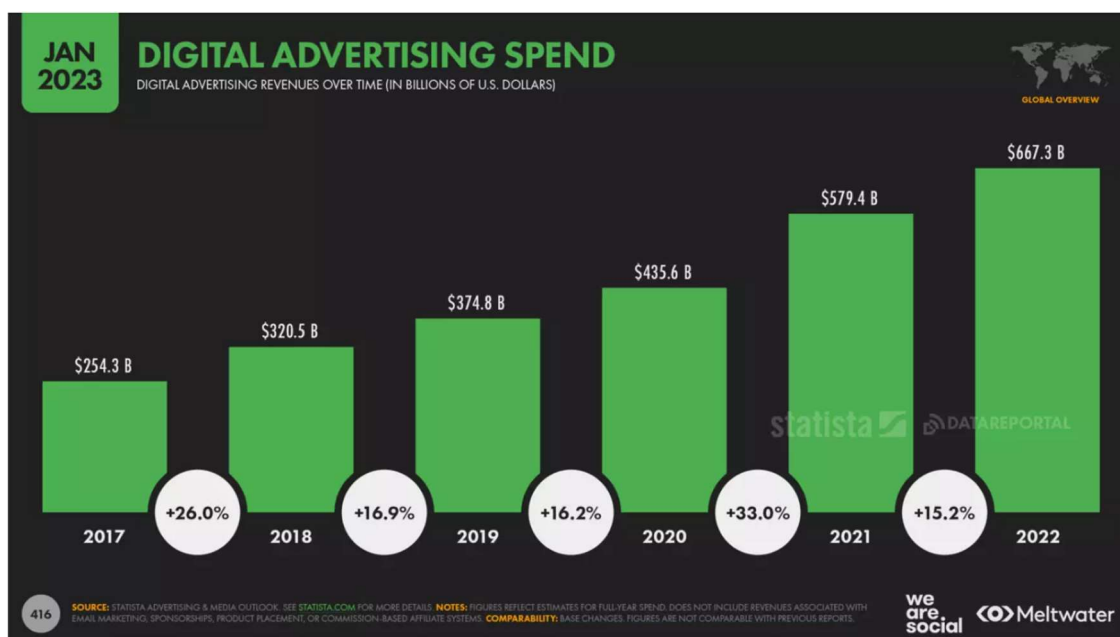


Рисунок 1.2 Отчет Global Digital. Доходы от цифровой рекламы с течением времени в миллиардах долларов США

Таким образом, размещение рекламы в социальных сетях кажется полностью оправданным, что конечно же обуславливает необходимость анализа соцсетей.

Что касается анализа соцсетей, существуют различные его виды. Можно анализировать контент или связи в социальных сетях.

В части контента данные, как правило, являются неструктурированными и представляют собой различные сущности. Социальные сети могут быть условно классифицированы по типу данных следующим образом:

- содержащие преимущественно изображения (например, Instagram¹);
- содержащие преимущественно видео (например, YouTube, Vimeo);
- содержащие преимущественно текстовые данные (например, Twitter, Telegram);
- содержащие данные смешанного типа видео – (например, ВКонтакте, FaceBook¹)

Для анализа различных типов данных существуют различные подходы и методы. Например, одна из самых распространённых задач анализа текстовых данных является анализ тональности — это метод обработки естественного языка (NLP), используемый для определения того, являются ли данные (текст) положительными, отрицательными или нейтральными.

Не менее важными данными являются связи в социальных сетях. Они могут характеризовать знакомство или принадлежность тому или иному сообществу. Исследуя связи в соцсетях, можно решать задачи определения интересов и предпочтений пользователей. Социальные сети, в части связей, чаще всего представлены в виде графов. Аккаунты в социальных сетях сопоставляются с вершинами в графе, а связи – с рёбрами, при этом в зависимости от вида связи (дружба, либо подписка) граф может быть, как неориентированным, так и ориентированным.

¹ Организация Meta и ее продукты Instagram и Facebook признаны экстремистскими организациями и запрещены на территории РФ

В данной работе я буду в основном использовать числовые данные (количество подписчиков, лайков, количество подписчиков определенного возраста и т. д.). Для эффективного представления и передачи результатов я буду использовать несколько методов визуализации данных, включая диаграммы, графики и таблицы. Эти методы не только сделают данные более понятными, но и позволят мне выделить ключевые моменты, что поможет в анализе результатов. Даже самые сложные данные могут быть легко читаемыми благодаря хорошей визуализации.

Теперь рассмотрим несколько наиболее популярных социальных сетей и приведем некоторые цифры для каждой из них.

ВКонтакте - российская социальная сеть, базирующаяся в Санкт-Петербурге. ВКонтакте доступна на нескольких языках, но преимущественно она пользуется популярностью среди русскоговорящих пользователей. В ВКонтакте каждый пользователь может отправлять как личные, так и общедоступные сообщения, создавать мероприятия или группы, играть в игры, слушать музыку, смотреть и выкладывать видео, создавать посты, делиться своими мыслями и идеями в блогах и многое другое. Благодаря удобному интерфейсу и множеству функций, предоставляемых ВКонтакте, пользователи могут легко взаимодействовать друг с другом, находить единомышленников и делиться интересными материалами. Кроме того, ВКонтакте постоянно обновляется и добавляет новые возможности, чтобы удовлетворить потребности своих пользователей.

- Аудитория в России: 81.18 млн пользователей.
- 63,4 % всех пользователей интернета пользуется ВКонтакте.
- Каждый день пользователи соцсети отправляют 15 млрд сообщений.
- В соцсети более 2 млн страниц для бизнеса.
- Активная аудитория VK Mini Apps превысила 34 млн. в месяц.

- Самые популярные сервисы: «Еда ВКонтакте», «Такси ВКонтакте», «Здоровье ВКонтакте», «Объявления ВКонтакте», «VK Работа» и AliExpress, каждым из них пользуется > 1 млн человек в месяц.
- Среднее время, проводимое в социальной сети — 35 минут в день.

Facebook² - американская социальная сеть, базирующаяся в Менло-Парке, штат Калифорния. Facebook² был запущен 4 февраля 2004 года Марком Цукербергом. Facebook² считается одной из четырех крупнейших технологических компаний наряду с Amazon, Apple и Google.

- Аудитория: 2,8 млрд пользователей.
- Аудитория в России: 8,9 млн пользователей.
- Каждый день на Фейсбук² отправляют более 100 млрд сообщений.
- На сайте зарегистрировано более 140 млн компаний.
- В России пользователи в среднем проводят в соцсети 10 минут в день.

Instagram² — это мобильная социальная сеть для обмена фотографиями и видео. Instagram² принадлежит Facebook².

- Аудитория: >2 млрд пользователей.
- Аудитория в России: 61 млн пользователей.
- За минуту пользователи Instagram² делятся 65 тыс. фотографий.
- Всего в соцсеть загружено более 50 млрд фотографий.
- В России пользователи в среднем проводят в Instagram² 26 минут в день.

TikTok — это сервис для создания и просмотра коротких видео, принадлежащий пекинской компании «ByteDance».

- Аудитория: 1.05 млрд пользователей.
- Аудитория в России: 54,86 млн.
- 43,0 % всех пользователей интернета в России пользуются TikTok
- В среднем пользователи TikTok проводят в приложении 1.5 часа в день, в России — 32 минуты.

² Организация Meta и ее продукты Instagram и Facebook признаны экстремистскими организациями и запрещены на территории РФ

- Меньше, чем за четыре года TikTok скачали более 3 млрд раз.
- За неделю с 4 по 10 апреля 2021 года все пользователи TikTok провели в мобильном приложении 13,93 млрд минут.

Telegram — это бесплатный мессенджер для мгновенного обмена аудио- и видеосообщениями, изображениями, GIF, стикерами, документами в разных форматах.

- Аудитория: 700 млн пользователей.
- Аудитория в России: более 55 млн пользователей.
- В русскоязычном Telegram сейчас больше 300 тысяч каналов.
- Суммарная аудитория пятидесяти самых крупных телеграмм-каналов составляет более 100 млн подписчиков.
- Люди проводят в мессенджере в среднем около 4 часов в месяц.
- Активными пользователями Telegram являются более половины (52%) всей аудитории рунета, а 61% как минимум скачали его на свой смартфон.

В России в 2022 году аудитория многих соцсетей упала, это связано с тем, что в настоящее время мировое сообщество накладывает на Россию санкции в связи с проведением РФ спецоперации на Украине. Санкции непосредственно затронули социальные сети: часть из них была заблокирована (Instagram³ и Facebook³), для части ограничен доступ пользователей к контенту. Часть соцсетей ввели самоцензуру (TokTok). Ранее одной из наиболее популярных и самой перспективной площадкой для продвижения считался Instagram³, однако за 3,5 месяца блокировки данная сеть потеряла 55% авторов, часть из которых перешли в сеть ВКонтакте, а многие — в Telegram. Сейчас перспективной социальной сетью для рекламы является Telegram, и многие бренды перенесли свои активы на ее просторы, однако продвигать свой товар через Telegram

³ Организация Meta и ее продукты Instagram и Facebook признаны экстремистскими организациями и запрещены на территории РФ

гораздо сложнее, чем через Instagram⁴ по причине того, что Telegram является мессенджером, а не социальной сетью, что влечет за собой некоторые особенности продвижения: коммуникации должны быть похожи на личную переписку; нет возможности прогнозировать эффективность. Наиболее близкой социальной сетью по продвижению к Instagram⁴ является Вконтакте. Дизайны приложений крайне похожи, аудитории схожи, бренд может вести свою страницу с привычной лентой и функционалом. Контент для Вконтакте, в отличие от Telegram, не нуждается в корректировке, так как попадает под функционал Instagram⁴.

На данный момент ВКонтакте является соцсетью №1 в России как по числу авторов, так и по объёму контента.

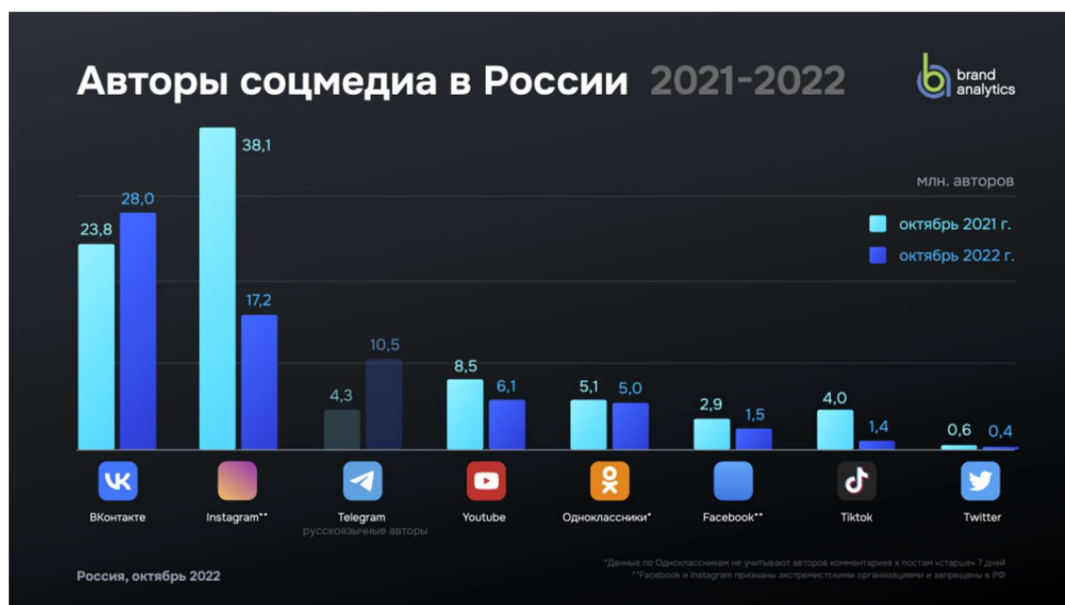


Рисунок 1.3 Изменение числа активных авторов в соцмедиа России с октября 2021 по октябрь 2022 года

На графиках динамики (рисунок 1.3) данные по Telegram представлены без привязки к географии, поскольку в мессенджере нет данных по гео. Стоит отметить, что Telegram и Вконтакте - единственные приложения, которые продолжают демонстрировать впечатляющий рост числа авторов на своей платформе. В то время, как другие социальные платформы, такие как Facebook⁵,

⁴ Организация Meta и ее продукты Instagram и Facebook признаны экстремистскими организациями и запрещены на территории РФ

⁵ Организация Meta и ее продукты Instagram и Facebook признаны экстремистскими

Twitter и Instagram⁵, либо замедляют свой рост, либо терпят снижение. Telegram удивляет своими результатами: он показал рекордный относительный прирост числа авторов — сразу на 144%. ВКонтakte также продемонстрировал рост, опередив лидера последних лет — Instagram⁵, у которого количество авторов снизилось на 55%. Также наиболее сильное снижение числа авторов наблюдается и у TikTok - на 74%. Это может быть связано с тем, что пользователи переключаются на другие социальные сети и мессенджеры, такие как Telegram, которые могут предложить им новые возможности для общения и обмена информацией [19].

В целом, можно сделать вывод, что на фоне общего замедления роста на многих социальных платформах, ВКонтakte опережает многих своих конкурентов и показывает стабильный рост.

Для проведения анализа любых данных необходимо организовать их сбор и дальнейшую обработку, поэтому стоит упомянуть способы извлечения данных из соцсетей. Существует множество различных способов получения данных из социальных сетей. Опишем три способа, которые позволяют собирать данные в автоматическом режиме и рассмотрим преимущества и недостатки каждого способа.

Способ №1 - API социальных сетей

Преимущества:

- Структурированные и документированные выходные данные.
- API, как правило, бесплатны в использовании.
- API обычно содержат SDK и библиотеки для разных языков программирования.

Недостатки:

- API обычно имеют ограничения по скорости.
- API обычно имеют ограничения по данным.

Способ №2 - Синтаксический анализ чистых HTML-страниц

Преимущества:

- Бесплатно.

Недостатки:

- Неструктурированные и недокументированные выходные данные.
- HTML-страницы в настоящее время, как правило, динамичны.
- Существует вероятность получить «бан» из социальной сети.

Способ №3 - Использование сторонних сервисов

Преимущества:

- Простота в использовании.
- Технические навыки не требуются.

Недостатки:

- Обычно неструктурированные и нечитаемые машиной выходные данные.
- Как правило, только частичный доступ к данным социальных сетей.
- Часть сторонних сервисов не бесплатна для использования.

Способ №1

API (интерфейс прикладного программирования) - набор четко определенных методов взаимодействия с приложением. В случае данной работы приложением являются социальные сети.

Все современные социальные сети предоставляют собственный API. Использование API — это законный и корректный способ получения данных из социальных сетей.

Способ №2

Современные HTML-страницы, как правило, динамичны, поскольку они генерируются различными интерфейсными фреймворками, такими как React, Angular, Embers и т. д. Это означает что страницы не имеют стабильной структуры, также страницы имеют теги с динамически генерируемыми атрибутами. Кроме того, если вы используете этот способ, вам необходимо

использовать свою собственную учетную запись в социальной сети или создать специальную техническую учетную запись. Это довольно сложно, и в конце концов вы будете заблокированы социальной сетью.

Способ №3

Существует множество сторонних сервисов, которые могут извлекать данные из социальных сетей, но обычно они являются только прокси-сервером между вами и API социальной сети. Большинство сторонних сервисов не бесплатны в использовании и предоставляют неудобный вывод данных.

Поэтому логичнее всего использовать способ №1, поскольку он является наиболее законным и правильным.

1.2 Анализ существующих ИТ-решений для анализа социальных сетей

В настоящее время существует множество инструментов для мониторинга социальных сетей и анализа пользовательской аудитории. Существует множество веб-сервисов и десктопных приложений, которые могут подсчитывать лайки, пользователей, просмотры и т. д. за день или другой период, уровень вовлеченности (ER) и множество других показателей. Вычисление метрик на самом деле не редкость в наше время.

Исследуем некоторые инструменты для мониторинга социальных сетей и анализа пользовательской аудитории, используя несколько критериев. Удобнее всего это сделать при помощи таблицы (Таблица 1).

Название	Метрики	Аналитика	Бесплатность	Соцсети	Комментарии
Popsters.ru	количество реакций, коммента	ERday, ERpost (анализ	нет	ВКонтакте, Twitter, Telegram, YouTube,	включает в себя множество социальных сетей, также он может

	риев, постов, подписчи ков и др.	активност и)		Одноклассн ики и др.	вычислять ER и показывать множество графиков, основанных на метриках и ER, не нужно регистрироваться [11]
Minter.io	количес во подписчи ков, реакций, прирос аудитори и и др.	пол, возраст, география подписчик ов (анализ аудитории)	нет	Twitter, Facebook ⁶ , LinkedIn, TikTok, Instagram ⁶	имеет хороший пользовательский анализ, но он доступен только в том случае, если у вас есть бизнес- аккаунт Facebook ⁶ [12]
Foller.me	количес во постов, подписчи ков, реакций и др.	тема поста	да	Twitter	анализирует любые Twitter аккаунты, на основании этого подсказывает пользователю темы для публикаций, которые получают хороший отклик [13]

⁶ Организация Meta и ее продукты Instagram и Facebook признаны экстремистскими организациями и запрещены на территории РФ

Publer.io	количество подписчиков, постов, реакций, комментариев, переходов, прирост участников и др.	анализ, как и активность, так и аудитории	с ограничениями	ВКонтакте, Instagram ⁷ , Facebook ⁷	поддерживает также и поиск сообществ по заданным параметрам, однако, к сожалению, с 25 марта 22 г временно приостановил работу [14]
Pur.social (Pur.Ninja)	количество реакций, комментариев, постов, подписчиков и др.	ERday, ERpost (анализ активности)	нет	ВКонтакте, Instagram ⁷ , Facebook ⁷ , Одноклассники и др.	есть функция автопостинга [15]
Borgi.ru [16]	количество подписчиков, постов	активность аудитории	да	ВКонтакте	
Media-vk.ru		анализ аудитории	нет	ВКонтакте	заточен под ВКонтакте,

⁷ Организация Meta и ее продукты Instagram и Facebook признаны экстремистскими организациями и запрещены на территории РФ

		по географии, полу, возрасту, семейному положению			помогает не просто собрать статистику, но и на ее основе составить портрет потребителя (целевой аудитории) [17]
VK Admin	посещаемость, просмотры, количество новых подписчиков и др.	анализ активности и	да	ВКонтакте	разработчиком является сам ВКонтакте. Пользоваться может только администратор сообщества, т.е. статистику чужих сообществ смотреть нельзя [18]

Таблица 1 Сравнение существующих решений в области анализа данных соцсетей

Большинство сервисов имеют общие недостатки:

- Сервисы могут анализировать отдельно: либо пользовательскую аудиторию (по географии, полу, возрасту, семейному положению), либо аналитику активности (уровень вовлеченности и т. д.).
- Большинство платные и достаточно дорогие, что невыгодно малому бизнесу или начинающим маркетологам.
- Все мощные инструменты привезены из-за рубежа, поэтому они не могут работать с vk.com => они не полностью подходят для российских пользователей.

Также существует огромное количество инструментов для анализа страниц пользователей в отдельности, либо для анализа существующих социальных связей, либо для анализа и определения тем текстов из социальных сетей. Предметом исследования же моей работы является анализ сообществ социальной сети ВКонтакте, следовательно, такие сервисы не могут быть использованы.

Результат анализа существующих решений послужит основой для анализа требований, определения метрик и атрибутов пользовательской аудитории.

1.3 Постановка задачи

Задачей данной работы является разработка программного обеспечения для статистического анализа социальных сетей. Цель такого анализа – поиск наиболее подходящих сообществ для последующего размещения в них рекламы.

Оценить сообщество можно с помощью:

1. сбора и структурирования метрик активности. Определим основные метрики:

- Наименование сообщества;
- Количество подписчиков;
- Количество постов;
- Количество лайков в среднем;
- Количество репостов в среднем;
- Количество просмотров в среднем;
- Количество комментариев в среднем;
- Engagement Rate (ER) — это коэффициент вовлеченности аудитории

в публикуемый контент. Или, другими словами, данная метрика отображает процент пользователей, проявивших активность у публикаций/поста;

$$ER = \frac{\text{сумма всех реакций (лайки, комментарии, репосты и т. д.)}}{\text{количество подписчиков} * \text{количество дней в периоде}} * 100$$

- Talk Rate (TR) - коэффициент коммуникабельности. Метрика аналогична Engagement Rate, но учитывает только количество комментариев. Позволяет определить, насколько аудитория вовлечена в коммуникацию;

$$TR = \frac{\text{сумма всех комментариев}}{\text{количество подписчиков} * \text{количество дней в периоде}} * 100$$

- Love Rate (LR) - коэффициент привлекательности. Метрика также аналогична Engagement Rate, но учитывает только количество лайков. Позволяет определить, насколько публикации привлекательны для аудитории;

$$LR = \frac{\text{сумма всех лайков}}{\text{количество подписчиков} * \text{количество дней в периоде}} * 100$$

2. наличия нелегитимных постов;

3. атрибутов пользовательской аудитории:

- Возраст;
- Пол;
- География (страна/город).

Для наглядной демонстрации работы ПО была составлена диаграмма IDEF0 (Рисунок 1.4)

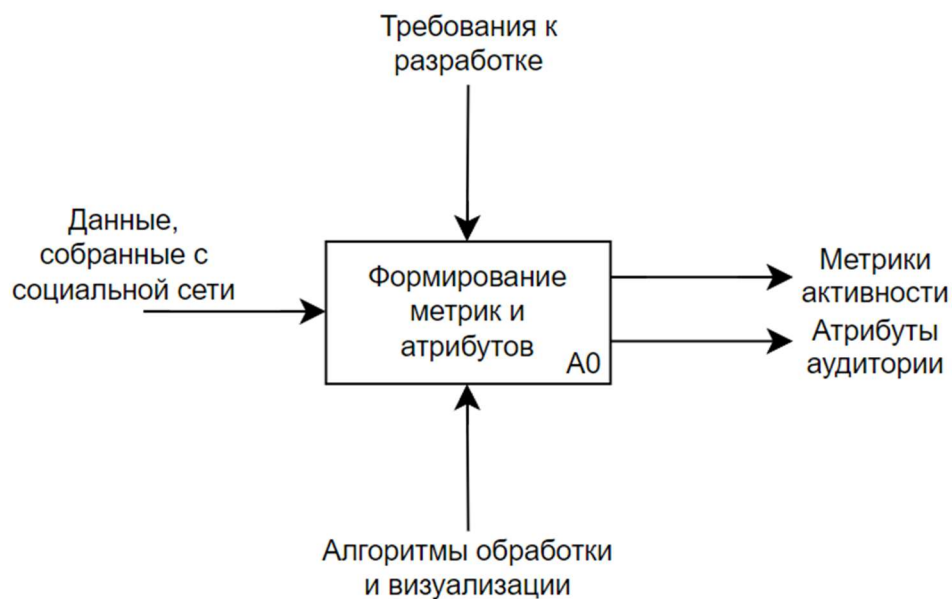


Рисунок 1.4 Модель работы программы

Помимо расчёта основных метрик, приложение должно:

- Визуализировать распределение аудитории по атрибутам.
- Для удобства и доступности со всех устройств программное обеспечение должно быть создано в виде веб-приложения.

Подробнее о функциональных и нефункциональных требованиях к приложению будет написано в следующей главе.

1.4 Выводы по главе 1

В данной главе был проведен анализ предметной области, приведены актуальные тенденции развития соцсетей, рассмотрены способы извлечения данных из соцсетей, проанализированы существующие решения, определена необходимость анализа соцсетей для последующего применения результатов в маркетинге, а конкретно для размещения рекламных постов с том или ином сообществе.

Также определена необходимость разработки программного обеспечения, которое бы выполняло анализ социальных сетей с указанием конкретных показателей, поставлены задачи для проектирования и разработки, что позволяет перейти к этапу проработки бизнес-требований и архитектуры приложения.

2. Проектирование приложения

Проектирование — это важный этап в разработке любого программного продукта. Оно заключается в поиске оптимального способа реализации функциональности системы с учетом имеющихся ресурсов и заданных ограничений. Проектирование является ключевым моментом в разработке программного обеспечения, поскольку позволяет создать качественный и функциональный продукт, который будет удовлетворять потребностям пользователей.

Проектирование включает в себя несколько этапов. Первый этап — это анализ требований к системе. На этом этапе определяются функциональные и нефункциональные требования к системе. Второй этап — это выбор архитектуры и технологий, которые будут использованы при разработке. На этом этапе определяется общая концепция программного продукта, его модули, компоненты и связи между ними. Необходимо выбрать технологии, которые наилучшим образом подходят для решения задач, стоящих перед ПО.

2.1 Проработка функциональных и нефункциональных требований

Проработка функциональных и нефункциональных требований позволяет определить все необходимые функции и характеристики продукта. Важно не только определить эти требования, но и продумать, как они будут взаимодействовать между собой и каким образом будут реализованы.

Проработка требований может включать в себя анализ рынка и конкурентов, определение потребностей пользователей, а также учет технических ограничений и возможностей (первое было проведено в первой главе).

Правильно проработанные требования позволяют сократить количество ошибок и недоработок в разрабатываемом приложении, ведь на каждом этапе процесса разработки приложения будет иметься ясное представление о том, что именно необходимо реализовать. Кроме того, проработка требований может ускорить процесс разработки, ведь чем точнее и подробнее определены

требования, тем легче и быстрее будет их реализовать.

При проработке функциональных требований необходимо учитывать, какой функционал должен быть реализован в приложении, что должно происходить при определенных действиях пользователя и каким образом приложение должно реагировать на них. Кроме того, необходимо определить, какие данные должны быть введены пользователем и как должны быть обработаны приложением.

В свою очередь, проработка нефункциональных требований подразумевает определение технических характеристик приложения, таких как производительность, надежность, безопасность, удобство использования и др. Не менее важно определить, каким образом будет осуществляться поддержка приложения, какие технологии будут использованы и какие требования будут предъявляться к аппаратному обеспечению.

Начнем с проработки функциональных требований. Формируя требования к разработке, нужно понимать, для какого пользователя разрабатываем продукт. Для этого определим Акторов (Actor) и для каждого актора определим набор возможностей.

В проектируемом мною веб-приложении статистического анализа сообществ ВКонтакте определены следующие акторы:

- Авторизованный пользователь — может пользоваться всеми функциями приложения: поиск и добавление сообщества в БД, просмотр основной информации о конкретном сообществе, просмотр метрик активности и атрибутов аудитории и т. д.
- Неавторизованный пользователь — доступна авторизация и регистрация.
- Администратор — лицо, постоянно работающее с этим ПО, опытный пользователь, который имеет доступ к БД пользователей и может их редактировать и удалять. Также ему доступны все те же функции приложения, что и авторизованному пользователю. Сделать пользователя администратором

можно только через панель администратора.

Для понимания ролей ниже представлена диаграмма прецедентов (вариантов использования), которая включает 3 акторов и 7 прецедентов (рисунок 2.1).

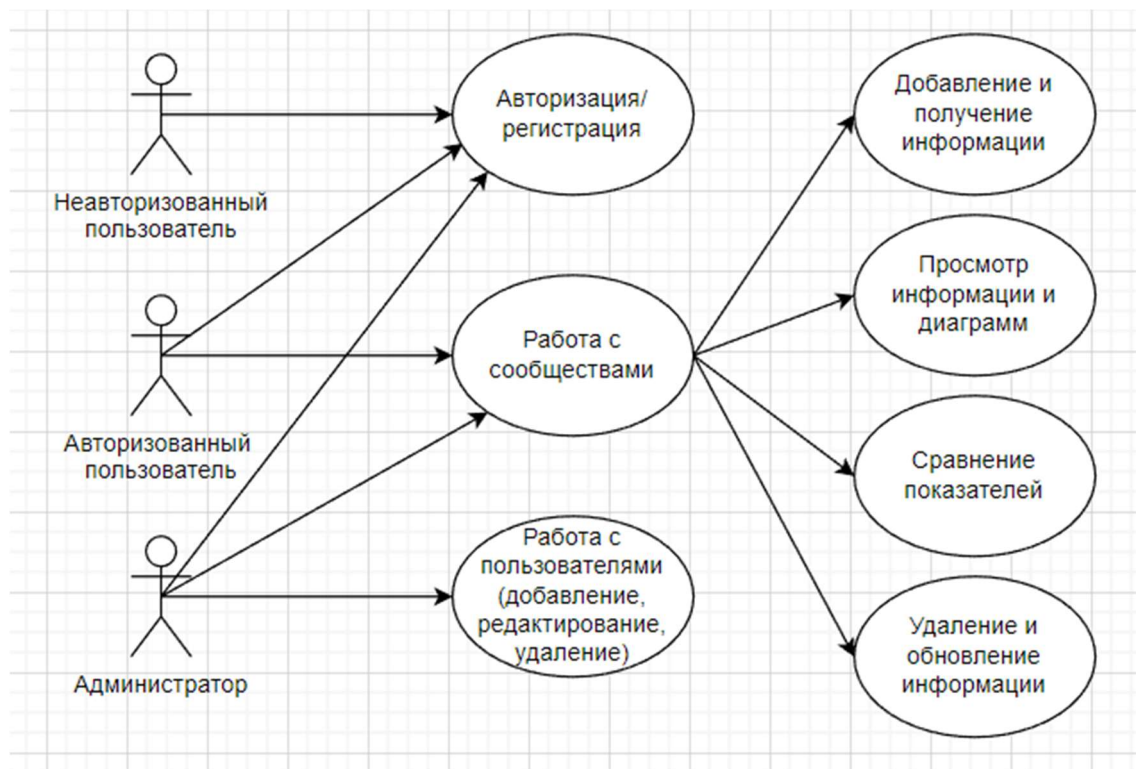


Рисунок 2.1 Диаграмма прецедентов

Далее представлены непосредственно функциональные требования.

Программа должна выполнять следующие функции:

- Предоставлять авторизацию и регистрацию пользователей.
- Обеспечить разделение прав доступа соответственно ролям: предоставлять каждому пользователю системы личный кабинет пользователя, включающий возможность получения информации в зависимости от предоставленных компетенций.
- Обеспечивать добавление сообщества, которое пользователь хочет проанализировать, получение информации о нем, её обновление, а также удаление сообщества.
- Запрашивать подтверждение при удалении сообщества.

- Парсить данные сообщества (количество подписчиков; количество постов; информацию о подписчиках: их возраст, пол, географию; информацию о постах: количество лайков, комментариев, репостов, просмотров), рассчитывать метрики согласно формулам (ER, LR, TR, среднее количество лайков, комментариев, репостов и просмотров), визуализировать атрибуты аудитории (круговые диаграммы: распределение подписчиков по полу, географическому признаку: страна и город; столбчатая диаграмма: распределение подписчиков по возрасту).
- Сохранять спарсенную и рассчитанную информацию в БД.
- Обеспечивать доступ к собранной статистике с целью получения необходимой информации.
- Показывать таблицу с ранее добавленными сообществами с основными характеристиками (количество подписчиков, ER, др.), для сравнения статистики различных сообществ между собой.
- Обеспечивать поиск сообщества по названию сообщества или адресу среди ранее добавленных сообществ, а также сортировку сообществ по названию/количеству участников и другим полям.
- Обновлять информацию при необходимости.
- Уведомлять пользователя, если сообщество содержит нелегитимные посты.
- Уведомлять пользователя об ошибках, возникающих в ходе неправильной работы (неудачная авторизация, сообщество уже добавлено или не существует и др.).

Перейдем к нефункциональным требованиям. Для удобства они разделены на несколько групп:

1. Безопасность:

- Один пользователь не должен иметь доступа к результатам анализа другого пользователя.

- Собираемая информация о подписчиках сообществ должна быть деперсонализированной.
- Пользователь должен иметь контроль над своими данными, такой как возможность удаления своих данных по запросу.
- Защита пользовательских данных, таких как данные авторизации: приложение должно использовать шифрование при хранении данных пользователей для снижения вероятности их утечки [6].

2. Производительность:

- Приложение должно работать без сбоев и аварийных завершений работы.
- Приложение должно обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. В указанных случаях система должна выдавать пользователю соответствующие сообщения, после чего возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.
- Время отклика веб-приложения 99 процентов должно быть не более 2 секунд.

3. Масштабируемость:

- Система должна предусматривать возможность масштабирования по производительности и объему обрабатываемой информации без модификации ее программного обеспечения путем модернизации используемого комплекса технических средств.
- Обновления веб-приложений не должны приводить к повреждению содержимого базы данных.
- Цикломатическая сложность кода не должна превышать 7. Ни один метод ни в одном объекте не может превышать 200 строк кода.
- Код должен иметь содержательные комментарии.

4. Техническое обслуживание:

- Приложение должно быть просто в обслуживании и обновлении.
- Приложение должно использовать современные технологии и платформы, обеспечивающие поддержку системы, как минимум, на горизонте 5 лет.

- База данных приложения должна обрабатываться с применением развитых СУБД (SQLite).

5. Переносимость:

- Приложение должно поддерживать адаптивную верстку.
- Панель мониторинга показателей должна быть доступна на всех устройствах, имеющих веб-браузер.

6. Юзабилити или удобство использования:

- Взаимодействие пользователей с прикладным программным обеспечением, входящим в состав системы должно осуществляться посредством визуального графического интерфейса (GUI).

- Веб-приложение должно использовать принципы UI/UX.
- Интерфейс приложения должен быть выполнен в нейтральных цветах, а также в минималистическом стиле, не должен быть перегружен графическими элементами.

- Интерфейс должен обеспечивать видимость всех элементов управления, которые необходимы для выполнения конкретной задачи.

- Все надписи экранных форм, а также сообщения, выдаваемые пользователю (кроме системных сообщений), должны быть на русском языке.

- Интерфейс должен быть рассчитан как на использование манипулятора типа «мышь», так и на касание экрана, то есть управление системой должно осуществляться с помощью набора экранных меню, кнопок, значков и т. п. элементов. Клавиатурный режим ввода используется главным образом при заполнении и/или редактировании текстовых и числовых полей экранных форм [5].

7. Совместимость:

- Приложение должно быть совместимо с другими системами.

8. Юридические вопросы:

- Приложение должно соответствовать действующим законам и подзаконным актам РФ [7].

Проработанные функциональные и нефункциональные требования позволяют перейти к следующему этапу необходимому перед непосредственной разработкой приложения – проектированию архитектуры.

2.2 Проектирование архитектуры приложения

Архитектурой информационной системы называют концепцию, которая определяет модель, структуру, выполняемые функции элементов архитектуры, а также взаимосвязь и взаимозависимость каких-то компонентов, из которых архитектура в целом состоит.

Основным фреймворком для создаваемого мною приложения был выбран Django. Это свободный высокоуровневый фреймворк для создания веб-приложений на языке Python. Он использует архитектурный паттерн Model-View-Controller (MVC), но с некоторыми отличиями от традиционной реализации этого паттерна. В Django, вместо контроллера, используется понятие URL-маршрутизации и обработчиков запросов (views). Таким образом, в Django архитектурный паттерн называется Model-View-Template (MVT). В MVT, модель (Model) отвечает за хранение и обработку данных, представление (View) отвечает за обработку запросов и взаимодействие с моделью, а шаблон (Template) отвечает за отображение данных пользователю.

В Django сайт строится из одного или нескольких приложений, которые являются отчуждаемыми и подключаемыми. Это одно из существенных архитектурных отличий Django от некоторых других фреймворков. Каждое приложение в Django представляет собой набор моделей, представлений и

шаблонов, которые могут быть использованы в различных проектах. Это позволяет создавать модульные приложения, которые могут быть легко переиспользованы в других проектах. В проектируемом ПО будет два приложения: "user" и "analytics". Приложение "user" будет отвечать за регистрацию и авторизацию пользователей. Оно будет содержать модели для хранения информации о пользователях, представления для обработки запросов, связанных с регистрацией и авторизацией, а также шаблоны для отображения страниц, связанных с этими функциями. Приложение "analytics" будет отвечать за работу с сообществами и содержать соответствующие модели, представления и шаблоны.

По умолчанию Django использует SQLite в качестве базы данных. SQLite — это легковесная, встроенная база данных, которая не требует запущенного сервера и может быть легко перенесена с одного компьютера на другой. Она очень проста в использовании и хорошо подходит для небольших проектов. В проектируемом приложении было решено использовать именно ее.

Для работы с базой данных Django использует собственный ORM (Object-Relational Mapping). Модель данных описывается классами Python, и по ней генерируется схема базы данных. ORM в Django позволяет разработчикам работать с базой данных без необходимости написания SQL-запросов. Вместо этого, они могут использовать объекты Python для создания, чтения, обновления и удаления данных в базе данных. Кроме того, Django также предоставляет множество других инструментов для работы с базами данных: миграции базы данных, административный интерфейс и др. Это делает работу с базами данных в Django очень удобной и эффективной.

Спроектируем саму БД. Сущностей для хранения информации нам требуется всего 2 (рисунок 2.2):

- analytics_group — таблица, которая хранит полученную и рассчитанную информацию по сообществам (ссылка, название, общее и среднее количество лайков/комментариев/репостов/просмотров, количество подписчиков, метрики ER/LR/TR, наличие ненормативной лексики), статус

загрузки, дату и время добавления и обновления информации о сообществе. В age, city, country, sex хранятся пути до генерируемых картинок диаграмм, а в members_info - словарь с информацией о подписчиках (например, {"female": 345, "male": 283, "city": {"Moscow": 350, "Saint Petersburg": 5}, "country": {"Russia": 493, "Kazakhstan": 3}, "age": {"<18": 1, "18-24": 142, "25-34": 41, "35-44": 8, "45-54": 3, ">55": 10}}). Первичный ключ – id, внешний ключ – user_id (наследуется из таблицы auth_user).

- auth_user – таблица, которая хранит информацию о пользователях (логин, пароль в зашифрованном виде, email, является ли пользователь администратором и др.). Первичный ключ – id.

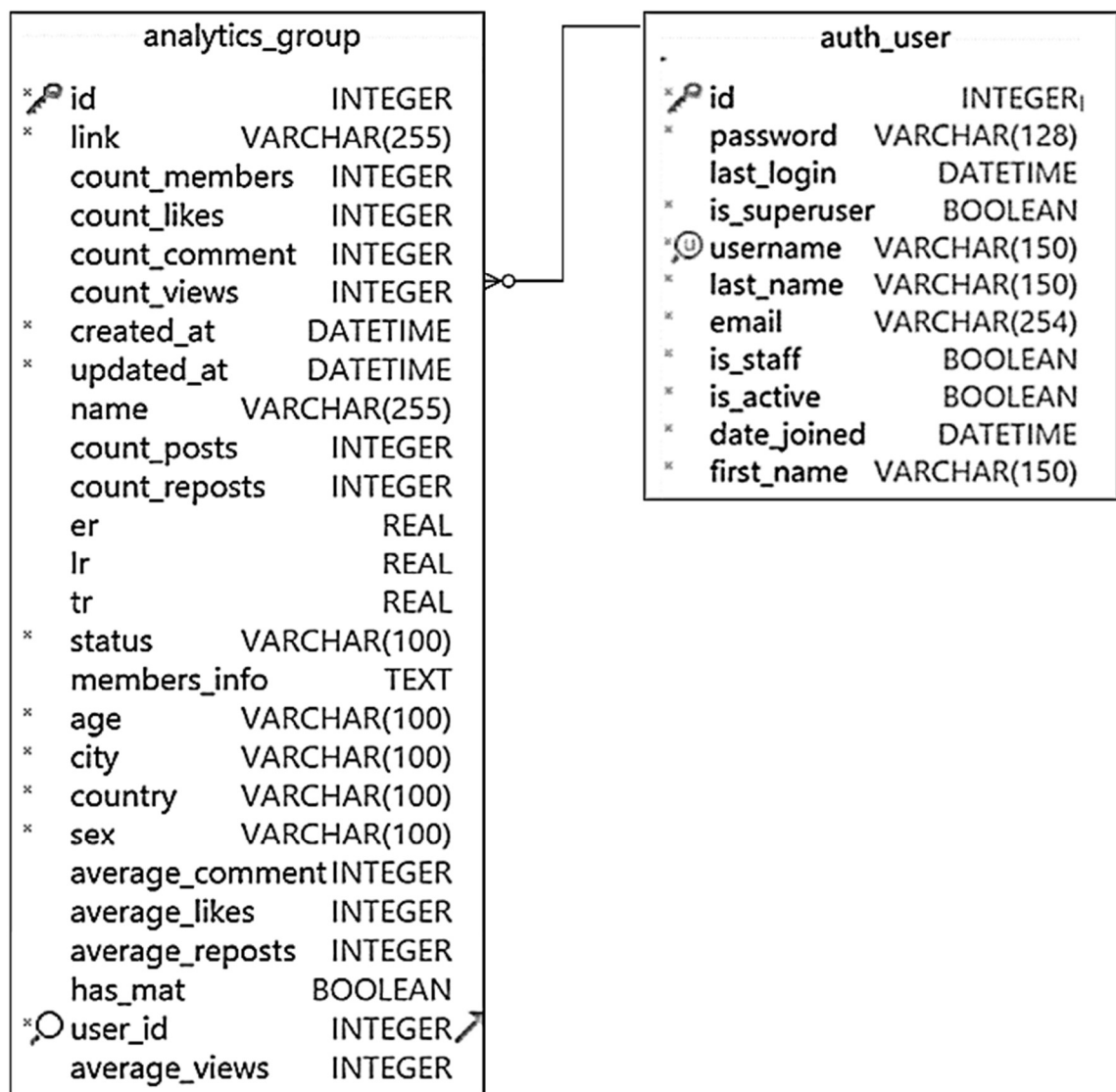


Рисунок 2.2 Модель БД

Также в связи с тем, что парсинг данных сообществ может занять длительное время, особенно если рассматривать большие сообщества с большим количеством подписчиков и постов, было решено использовать библиотеку Celery и БД Redis. Celery — это асинхронная очередь задач для Python, которая позволяет выполнять долгие и трудоемкие задачи в фоновом режиме, позволяя пользователям продолжать работать с интерфейсом приложения без задержек. Использование Celery может значительно повысить производительность и быстродействие приложения для парсинга данных из VK. Celery работает по принципу клиент-сервер. Клиент отправляет задачи в очередь, а сервер получает задачи из очереди и выполняет их [28]. Redis может использоваться в качестве брокера сообщений для Celery, что позволяет отправлять задачи в очередь и получать результаты выполнения задач.

Таким образом, архитектура веб-приложения для анализа сообществ ВКонтакте будет основываться на следующих компонентах:

1. Веб-сервер Django: главный компонент веб-приложения, который обрабатывает пользовательские запросы, отображает HTML-страницы и взаимодействует с базой данных.
2. База данных SQLite: используется для хранения данных приложения, таких как информация о пользователях и сообществах ВКонтакте.
3. Celery: используется для обработки фоновых задач, таких как запросы к API ВКонтакте, обработка статистических данных и т. д.
4. Redis: используется для хранения результатов выполнения задач Celery и обмена сообщениями между приложением и фоновыми задачами [29].

Модель архитектуры разрабатываемого приложения представлена на рисунке 2.3.

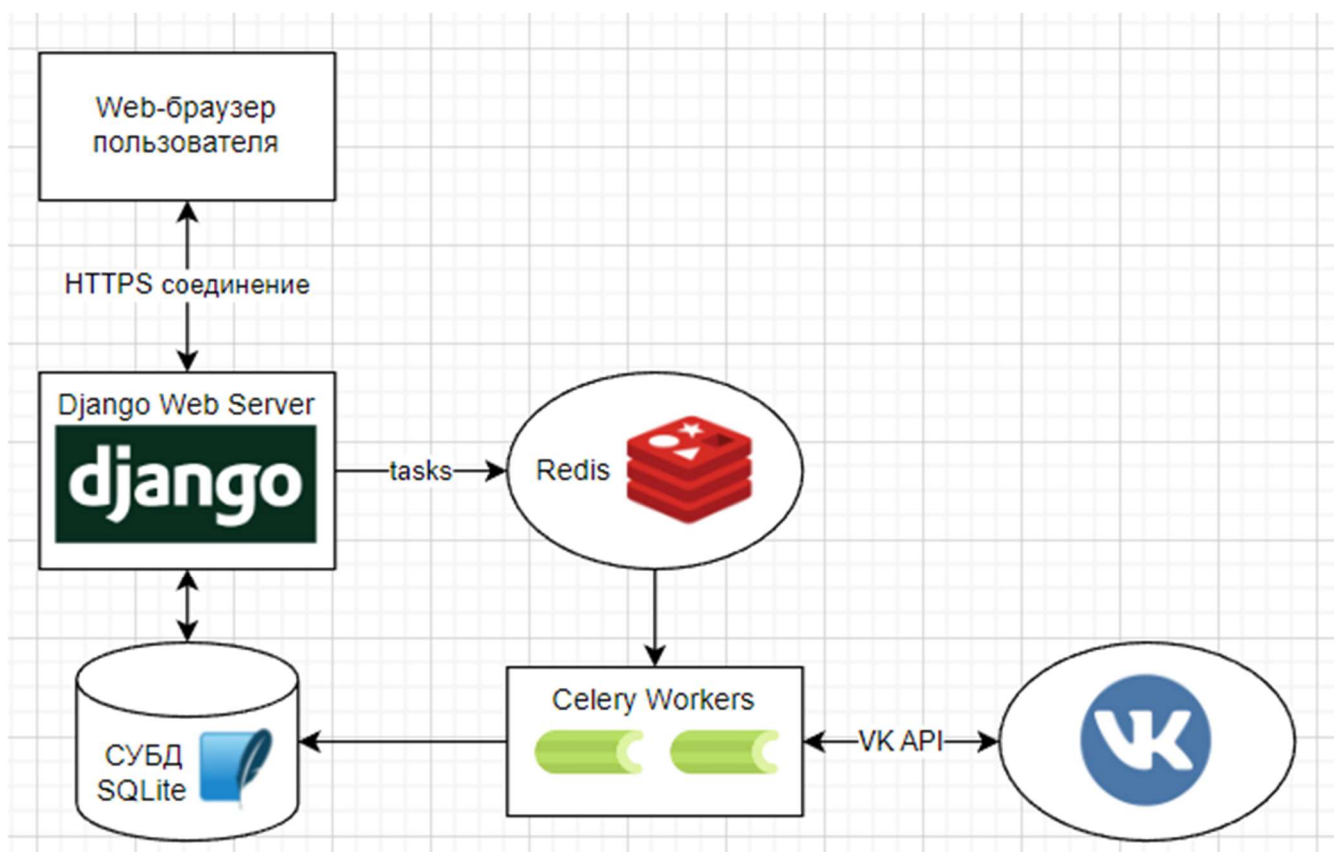


Рисунок 2.3 Модель архитектуры разрабатываемого приложения

Когда пользователь отправляет запрос на анализ сообщества ВКонтакте, веб-сервер Django добавляет задачу в очередь Celery, используя Redis в качестве посредника. Celery получает задачу из очереди, выполняет запросы к API ВКонтакте для получения информации о сообществе и его участниках, обрабатывает статистические данные и сохраняет результат в базу данных SQLite. Пользователь может просматривать результаты анализа сообщества, опрашивая веб-сервер Django, который будет запрашивать результаты из базы данных SQLite.

Как следует из приведенной диаграммы, для развертывания веб-приложения необходим сервер с установленными библиотеками и клиентское автоматизированное рабочее место с установленным браузером.

Полезным будет проектирование бизнес-процесса. Основная цель создания веб-приложения — это анализ конкретного сообщества ВКонтакте. Поэтому рассмотрим алгоритм более подробно. На рисунке 2.8 представлена диаграмма EPC, демонстрирующая бизнес-процесс "Проведение статистического анализа сообщества ВКонтакте". EPC (Event-driven Process Chain) диаграмма — это графическая модель, которая используется для описания и визуализации бизнес-процессов. Она состоит из блоков (событий, функций и процессов) и стрелок, связывающих эти блоки. События являются триггерами, которые запускают процессы, а функции выполняют действия. Процессы описывают последовательность действий и связи между функциями и событиями. На данной диаграмме показаны все этапы обработки и визуализации данных, начиная с постановки цели и заканчивая формированием отчета. В пункте "Обработать данные" входит вычисление метрик согласно формулам: ER, LR, TR, среднее количество лайков, комментариев и просмотров, вычисление процентных соотношений возрастов, полов подписчиков и выделение чаще всего встречающихся городов и стран подписчиков. Визуализация результатов включает в себя построение круговых диаграмм распределения подписчиков по полу и географическому признаку и столбчатой диаграммы распределения по возрасту, а также вывод ранее посчитанных метрик.

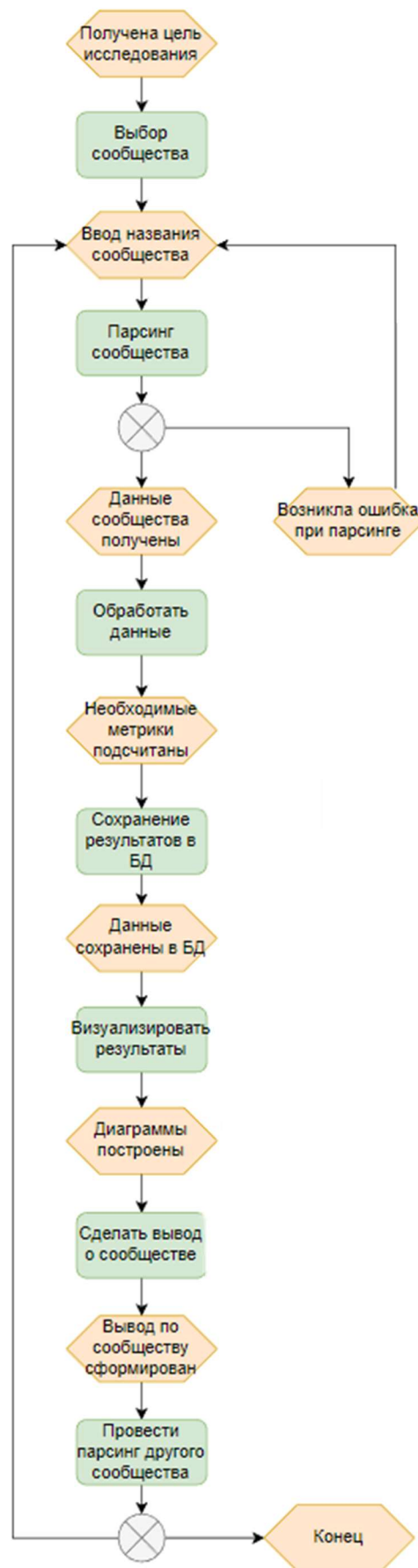


Рисунок 2.8 Диаграмма EPC

Рассмотрим входные и выходные данные данного алгоритма.

Название	Пример	Тип
Ссылка на сообщество	https://vk.com/robotics_laboratory	String (должна содержать vk.com/ в адресе)

Таблица 2 Входные данные

Название	Пример	Тип	Посчитана на основе
Название сообщества	Robotics Laboratory	String	
Количество подписчиков	693	Int	
Количество постов	793	Int	
Количество просмотров в среднем	884	Int	Количество просмотров на постах всего, количество постов
Количество лайков в среднем	4	Int	Количество лайков на постах всего, количество постов
Количество комментариев в среднем	0	Int	Количество комментариев на постах всего, количество постов
Количество репостов в среднем	1	Int	Количество репостов на постах всего, количество постов
ER (%)	0,84	Float	Количество лайков и комментариев в среднем, количество просмотров
LR (%)	0,66	Float	Количество лайков в среднем, количество просмотров

TR (%)	0,02	Float	Количество комментариев в среднем, количество просмотров
Информация о подписчиках	{ "female": 345, "male": 283, "city": {"Moscow": 350, "Saint Petersburg": 5}, "country": {"Russia": 493, "Kazakhstan": 3}, "age": {"<18": 1, "18-24": 142, "25-34": 41, "35-44": 8, "45-54": 3, ">55": 10} }	Dict	Возраста, пол, города и страны подписчиков, количество подписчиков

Таблица 3 Выходные данные

2.3 Выводы по главе 2

Во второй главе работы проанализированы и сформулированы функциональные и нефункциональные требования к разрабатываемому приложению статистического анализа социальных сетей. В результате описана архитектура будущего веб-приложения и обосновано ее использование. Кроме того, были составлены диаграммы прецедентов, ЕРС и модель БД, что помогло уточнить детали и обеспечить более полное понимание проектируемого ПО. В результате, были получены более детальные и точные данные, которые позволят более эффективно реализовать систему и удовлетворить потребности пользователей, что позволяет перейти непосредственно к разработке приложения.

3. Реализация приложения статистического анализа данных социальных сетей

3.1 Выбор инструментальных средств разработки программного приложения

Чтобы достичь максимальной эффективности и рациональности в работе программы, необходимо подобрать наиболее подходящие программные средства разработки, которые будут соответствовать требованиям проекта и обеспечат оптимальное использование ресурсов. Это может включать в себя выбор определенного языка программирования, интегрированных сред разработки, фреймворков, библиотек и других инструментов, которые будут использоваться в процессе разработки. Частично используемые средства описаны в пункте 2.2.

Разработка приложения в основном выполнена на языке **Python**, версии 3.10.7. Python — это высокоуровневый интерпретируемый язык программирования. Python отличается простотой и читаемостью кода, что позволяет повышать производительность разработчика. Python имеет минималистичный синтаксис ядра, но формирует большой объем полезных функций из стандартной библиотеки. Большой плюс Python в том, что он позволяет использовать структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Еще одно преимущество языка заключается в динамической типизации, автоматическом управлении памятью и полной интроспекции. Python поддерживает многопоточные вычисления (что полезно в данном проекте, учитывая количество данных, которые необходимо парсить) и высокоуровневые структуры данных, а также поддерживает разбиение программ на модули, которые могут объединяться в пакеты. Также Python кроссплатформенный, имеет развитую официальную документацию, огромное количество учебных пособий, большое количество библиотек и фреймворков для различных задач и упрощенный экспорт со встроенным диспетчером пакетов Python (pip). Python является одним из самых популярных языков программирования в мире, в

частности он активно используется для анализа данных. Он активно развивается, новые версии с добавлением/изменением языковых свойств выходят примерно раз в два с половиной года 24].

Для разработки на языке программирования Python была выбрана интегрированная среда разработки JetBrains **PyCharm**, которая базируется на платформе IntelliJ IDEA. Это кроссплатформенная среда разработки: может быть установлена и использована на Windows, MacOS и Linux. Плюсом является, что версия PyCharm Community доступна бесплатно.

PyCharm Community предоставляет ряд инструментов, которые упрощают и ускоряют процесс разработки:

- Статический анализ кода, который обеспечивает подсветку синтаксиса, обнаружение ошибок и интеллектуальную подсказку кода.
- Поддержка систем контроля версий, включая Git.
- Инструменты для отладки: поддержка точек останова, пошаговое выполнение кода и многое другое.
- Рефакторинг, в том числе переименование, извлечение метода, введение переменной, введение константы, подъем и опущение метода и т. д.
- Возможность интеграции с плагинами и расширениями, расширяющими функциональность среды разработки.
- Навигация по проекту и исходному коду, включая файловую структуру проекта, быстрый переход между файлами, классами, методами и использованиями методов.
- Поддержка многих популярных фреймворков, в том числе выбранного для разработки Django.
- Инструменты для работы с виртуальными окружениями Python [36].

Кроме того, PyCharm имеет удобный и интуитивно понятный интерфейс, а также обширную документацию и поддержку сообщества пользователей.

Основным фреймворком для разработки сайта, как уже ранее сказано, выбран **Django**. Django — это бесплатный фреймворк для веб-разработки на языке Python3, поддерживаемый Django Software Foundation. Django был создан в 2003 году и с тех пор стал одним из самых популярных фреймворков для веб-разработки.

К его преимуществам можно отнести:

- Множество готовых компонентов, таких как аутентификация, обработка форм, административный интерфейс, которые позволяют управлять данными в базе данных без написания дополнительного кода.
- Встроенный ORM, который позволяет работать с базами данных, используя объектно-ориентированный подход.
- Встроенный механизм для маршрутизации запросов, обработчики URL в Django конфигурируются явно при помощи регулярных выражений.
- Встроенный шаблонизатор, который позволяет создавать динамические страницы.
- Механизмы для повторного использования кода, такие как миксины (mixins) и наследование шаблонов. Миксины позволяют добавлять поведение к классам, не изменяя их исходный код, а наследование шаблонов позволяет создавать шаблоны, которые наследуются от других шаблонов, что позволяет избежать дублирования кода в шаблонах.
- Хорошо структурированная и подробная документация и большое сообщество разработчиков. Это обеспечивает поддержку и постоянное развитие фреймворка [25].

Для хранения данных была выбрана система управления базами данных **SQLite**. Это система, предназначенная для хранения и обработки информации. Комплекс таблиц, взаимосвязанных между собой, для доступа к которым применяется система управления базами данных.

Преимущества:

- Проста в использовании и не требует сложной настройки, можно

использовать в качестве встроенной базы данных в приложениях, что упрощает развертывание и установку приложений.

- Хранит данные в одном файле на диске, что делает её очень компактной и удобной для переноса.
- Обеспечивает высокую производительность при работе с небольшими базами данных.
- Поддерживает стандартный язык SQL, что делает её совместимой с большинством инструментов и библиотек, работающих с реляционными базами данных.
- Бесплатна и имеет открытый исходный код, что делает её доступной для всех.

Недостатки:

Ограниченная поддержка SQL, типов данных, не может обрабатывать огромные объемы данных. Однако данные недостатки не критичны в разрабатываемом приложении [27].

Исходя из этих достоинств и недостатков было принято решение использовать данную СУБД.

В качестве основы для верстки сайта выбран **Bootstrap**. Это набор бесплатных инструментов, предназначенных для создания веб-сайтов и веб-приложений. Он включает в себя готовые HTML- и CSS-шаблоны для типографики, веб-форм, навигационных меню, кнопок, карточек и других интерфейсных компонентов.

Существует несколько преимуществ в использовании Bootstrap:

- Время, необходимое для создания макета сайта, сокращается благодаря готовым шаблонам и решениям в фреймворке.
- Bootstrap предоставляет адаптивный дизайн, который позволяет сайту автоматически подстраиваться под различные размеры экранов, что обеспечивает удобство использования на различных устройствах и браузерах.
- Bootstrap использует Flexbox для управления расположением

элементов на странице, что обеспечивает более гибкий и удобный дизайн.

- Для работы с фреймворком потребуются элементарные навыки верстки [26].

Несмотря на преимущества фреймворка, есть и недостатки. Первый недостаток - при использовании фреймворка кода может получиться больше, чем при написании кода самостоятельно, ведь при разработке с нуля мы реализуем только необходимый функционал. Второй недостаток - может быть проблемой шаблонный дизайн. Тем не менее, это может быть легко решено кастомизацией готовых компонент фреймворка соответственно своим потребностям.

Сами страницы сверстаны, используя язык разметки **HTML** и **CSS** для задания стиля. **HTML** используется для создания и структурирования веб-страниц. С помощью **HTML** можно описать содержание и структуру документа, включая текстовый контент, ссылки, изображения, таблицы, формы ввода и другие элементы веб-страниц. **HTML** является стандартом веб-разработки и поддерживается всеми основными браузерами. Для описания стилей используется **CSS**. Это формальный язык, который позволяет описывать внешний вид документа, размеченного с помощью **HTML** или другого языка разметки. Он позволяет управлять цветом, шрифтом, стилем и расположением элементов веб-страницы. Основной задачей **CSS** было отделить описание логической структуры веб-страницы от ее внешнего вида.

Для визуализации диаграмм используется библиотека **matplotlib**. **Matplotlib** — это библиотека для языка программирования **Python**, которая предоставляет инструменты для создания различных типов графиков и диаграмм (линейные, столбчатые, круговые, точечные и т. д.). **Matplotlib** предоставляет множество настраиваемых параметров для каждого типа графика, таких как цвета, шрифты, размеры; позволяет сохранять графики в различных форматах, таких как **PNG**, **PDF**, **SVG**, **EPS** и другие. **Matplotlib** интегрируется с другими библиотеками **Python**, такими как **NumPy**, **Pandas**. И опять же **Matplotlib** имеет

обширную документацию и множество примеров, что облегчает изучение и использование библиотеки [30].

Для работы с API Вконтакте была выбрана библиотека **requests** - HTTP-библиотека, написанная на Python. С помощью нее мы можем отправлять запросы к API ВКонтакте и получать информацию о сообществах и пользователях социальной сети. Она позволяет отправлять GET, POST, PUT, DELETE и другие типы запросов, работать с параметрами запросов и заголовками, что позволяет настраивать запросы и отправлять дополнительную информацию на сервер.

Также при получении данных о пользователях и постах (помимо библиотеки **celery**, описанной ранее в пункте 2.2) были использованы потоки. **Thread** — это класс из стандартной библиотеки Python **threading**, который позволяет создавать и управлять потоками выполнения в Python. Класс **Thread** позволяет создавать новый поток выполнения, который может выполняться параллельно с основным потоком программы [31].

Для построения диаграмм в прошлой главе мною также использовалось **Draw.io**. Это инструмент для создания и редактирования диаграмм, графиков, потоковых схем, организационных структур и других типов визуальных представлений данных. Draw.io бесплатен и поставляется с библиотеками форм и шаблонов, которые пользователи могут использовать для создания профессионально выглядящих диаграмм. Диаграммы могут быть сохранены на облачный сервис хранения файлов Google Drive и др. или экспортированы в виде изображения или PDF файла. Инструмент доступен онлайн и не требует установки программного обеспечения [32].

3.2 Описание реализации приложения

Основной каталог проекта - `vk_analytics`, в нем содержатся все необходимые компоненты. Рассмотрим директории и файлы, входящие в его состав:

- `db.sqlite3` - файл базы данных SQLite, используемый для хранения информации о сообществах ВКонтакте и пользователях разработанного приложения.
- `manage.py` - файл для управления проектом Django. Это скрипт командной строки, который поставляется с проектом Django и используется для запуска сервера разработки (`runserver`), создания миграций для изменений в моделях базы данных (`makemigrations`), применения миграций к БД (`migrate`), создания суперпользователя для доступа к интерфейсу администратора (`createsuperuser`).
- `requirements.txt` - файл со списком зависимостей проекта, таких как Django, библиотеки для работы с VK API requests и других используемых зависимостей.
- `config/` - директория, которая содержит конфигурационные файлы для проекта. Включает следующие файлы:
 - `celery.py` - файл, который создает экземпляр приложения Celery и настраивает его для работы с Django для асинхронной обработки задач.
 - `settings.py` — это файл настроек проекта, который содержит конфигурацию базы данных, настроек приложений, настроек статических файлов и т.д. (например, настройка `CELERY_BROKER_URL = 'redis://127.0.0.1:6379/0'` - URL-адрес брокера сообщений, который используется для обмена сообщениями между приложением и Celery).
- `user/` - директория содержит файлы и директории, необходимые для работы приложения, связанного с пользователем. Оно использует стандартную модель пользователя Django. Содержит следующие директории и файлы:

- static/user/css/bootstrap.min.css и signin.css - файлы стилей для страниц приложения user.
- templates/user/ - директория, в которой хранятся шаблоны HTML страниц приложения. base.html – файл, содержащий шапку и футер страницы, который используется на обеих страницах и наследуется в следующие 2 файла. auth.html и registration.html – страницы авторизации и регистрации соответственно.
- apps.py - это файл, который содержит конфигурацию приложения user.
- forms.py - файл, в котором определяются формы для ввода данных пользователем: форма авторизации и регистрации пользователя. Формы используются для валидации данных, введенных пользователем, и для создания или обновления соответствующих моделей.
- models.py - файл, который содержит определения моделей для пользователей. Модель пользователя User используется для хранения информации о пользователях, такой как имя пользователя, адрес электронной почты и пароль.
- urls.py - файл маршрутизации, который определяет маршруты для авторизации, регистрации и выхода из системы. Маршруты определяют, какие представления будут вызываться для каждого URL-адреса.
- views.py - файл, в котором определяются обработчики запросов и логика приложения user. Представления обрабатывают запросы от пользователей и взаимодействуют с моделями и формами для выполнения соответствующих действий.
- media/ - директория для хранения картинок сгенерированных диаграмм в формате .png, которые отображают распределение аудитории по возрасту, городам, странам и полу. Директория содержит 4 поддиректории: age/, city/, country/ и sex/, в которых хранятся диаграммы распределения аудитории по соответствующим параметрам. Каждая диаграмма имеет имя файла, состоящее

из короткого адреса сообщества VK (например, диаграмма распределения аудитории по возрасту для группы VK с коротким адресом `itiabd2000` будет храниться в файле `itiabd2000.png` в директории `media/age/`). Для каждой группы VK, для которой были собраны данные, в директориях `age/`, `city/`, `country/` и `sex/` будут храниться соответствующие диаграммы.

- `venv/` - директория с виртуальным окружением Python, которое используется для изоляции зависимостей проекта от глобальных зависимостей Python на компьютере. Она создается автоматически при создании виртуального окружения. Виртуально окружение рекомендуется использовать при загрузке проекта во избежание конфликта версий.

- `analytics/` - директория с приложением для парсинга и анализа данных из VK. Это приложение имеет схожую структуру и логику взаимодействия компонентов с директорией `user/`, так как оба приложения являются частями проекта Django `vk_analytics`. Однако эту директорию хотелось бы рассмотреть поподробнее, ведь она является основной.

- `migrations/` - директория, которая содержит файлы миграций для изменения структуры базы данных без необходимости удаления и создания ее заново. Каждый файл миграции содержит инструкции для изменения базы данных, такие как создание, изменение или удаление таблиц, добавление или удаление столбцов и т.д. Файлы миграций создаются автоматически при изменении моделей в приложении Django с помощью команды `python manage.py makemigrations`.

- `static/analytics/css/bootstrap.min.css`, `sticky-footer-navbar.css` и `style.css` - файлы стилей для страниц приложения `user`

- `templates/analytics/` - директория, в которой хранятся шаблоны HTML страниц приложения `analytics`. `base.html` – файл, содержащий шапку и футер страницы, который используется и наследуется на все следующие страницы. `about.html` – страница с основной информацией о разработанном приложении и контактной информацией. `create.html` и `delete.html` – страницы добавления и удаления сообщества соответственно. `detail.html` – страница с информацией по

конкретному сообществу. `list.html` – страница со всеми добавленными сообществами.

- `admin.py` - файл, который содержит настройки административного интерфейса Django для моделей приложения.

- `apps.py` - это файл, который содержит конфигурацию приложения `user`.

- `forms.py` - файл, в котором определяются формы для ввода данных пользователем. В приложении `analytics` у нас только одна форма, где мы вводим ссылку на сообщество, которое собираемся анализировать.

- `models.py` - содержит определение модели `Group`, которая представляет группу VK, данные которой будут анализироваться. Модель содержит следующие поля: `link` - ссылка на группу VK; `name` - название группы; `count_members`, `count_likes`, `count_comment`, `count_views`, `count_reposts`, `count_posts` - общее количество подписчиков/лайков/комментариев/просмотров/репостов/постов в группе; `average_likes`, `average_comment`, `average_reposts`, `average_views` - среднее количество лайков/комментов/репостов/просмотров на постах группы; `has_mat` - наличие нелегитимных постов в группе (0 или 1); `er`, `lr`, `tr`; `members_info` - информация об участниках группы в формате JSON; `age`, `city`, `country`, `sex` - изображения диаграмм распределения аудитории группы по возрасту участников; `created_at` - дата и время создания записи в базе данных, значение этого которого будет устанавливаться автоматически при создании новой записи в базе данных с помощью параметра `auto_now_add=True`; `updated_at` - дата и время последнего обновления записи в базе данных, значение которого будет автоматически обновляться при каждом сохранении записи в базе данных с помощью параметра `auto_now=True`; `user` – поле, которое связывает запись группы с пользователем, который ее создал, является внешним ключом к модели `User` и определено с помощью метода `ForeignKey`, при удалении пользователя из базы данных, все связанные с ним записи групп также будут удалены; `status` – статус обработки сообщества (готово/в обработке). Каждое поле также имеет

свойство `verbose_name`, которое определяет человекочитаемое имя поля. После определения моделей в файле `models.py` необходимо выполнить миграцию базы данных, чтобы создать соответствующие таблицы в базе данных. Это можно сделать с помощью команды `python manage.py makemigrations` и `python manage.py migrate`.

- `urls.py` - содержит маршруты приложения `analytics`.
- `views.py` - содержит функции-обработчики для отображения страниц и обработки данных, связанных с аналитикой.

Класс `GroupListView` отображает список групп пользователя, в нем также содержится код для сортировки списка групп по различным полям.

`GroupDetailView` - класс-обработчик, который отображает информацию по конкретной группе. В функции-обработчике используется метод `filter` модели `Group`, который выбирает объект `Group` с заданным идентификатором `pk`. Затем, выбранный объект `Group` передается в шаблон `detail.html` в качестве контекста. Таким образом, данный код позволяет пользователю просмотреть информацию по конкретной группе.

`GroupCreateView` отвечает за добавление новой группы. В методе `get` происходит отображение формы для добавления новой группы. В методе `post` происходит обработка данных, полученных из формы. При отправке формы происходит проверка наличия группы с такой же ссылкой у текущего пользователя. Если группа не найдена, то происходит попытка получить информацию о группе с помощью функции `get_common_info`. Если группа найдена, то создается новый объект `Group` и запускается задача `update_group` для обновления данных о группе. Если при выполнении какого-либо из шагов возникает ошибка, то пользователю возвращается страница с ошибкой. В конце метода происходит перенаправление пользователя на главную страницу аналитики.

Класс `GroupUpdateView` отвечает за обновление информации о группе, `GroupDeleteView` - за удаление группы, `AboutPageView` - за отображение страницы с информацией о проекте.

– `services/` - это общая директория для сервисов, которые используются в проекте. Они предоставляют определенный функционал для проекта: предоставляют функционал для работы с VK API, расчета метрик, обновления данных о группе и отрисовки графиков.

Файл `vk.py` содержит функции для работы с VK API с помощью библиотеки `requests`. В данном файле реализованы функции для получения общей информации о группе, получения информации о подписчиках группы, получения информации о постах группы, проверки постов на легитимность.

API (Application Programming Interface) — это набор программных интерфейсов, протоколов и инструментов, которые позволяют различным приложениям взаимодействовать друг с другом. API определяет, как различные компоненты программного обеспечения должны взаимодействовать друг с другом, обеспечивая стандартизированный способ обмена информацией между приложениями. API могут использоваться для доступа к данным, функциям и сервисам, предоставляемым другими приложениями или сервисами. Например, социальные сети предоставляют API для доступа к данным профилей пользователей, а платежные системы - для обработки платежей.

Для загрузки необходимой информации по группам были использованы следующие методы VK API [21]:

- `groups.getById` - используется для получения информации о группе VK по ее идентификатору или короткому имени; возвращает основную информацию о группе, такую как ее название, описание, количество участников, ссылку на группу и т.д.

- `groups.getMembers` - возвращает список идентификаторов пользователей, которые состоят в группе, а также дополнительную информацию о каждом участнике, такую как его имя, фамилия, пол, дата рождения, город, страна и т.д.

- `wall.get` - возвращает список записей со стены, а также дополнительную информацию о каждой записи, такую как ее текст, дата публикации, количество лайков, комментариев и т. д.

Пример использования GET запроса к API VK для получения общей информации о группе:

```
BASE_URL = 'https://api.vk.com/method/' # содержит базовый URL точки доступа API VK
```

```
HEADERS = {'Authorization': 'Bearer 781b939e781b939e781b939ed47b08d25b7781b781b939e1c3ac44c423272b4f874bc02'} # содержит токен авторизации для доступа к API
```

```
params = {'group_id': group_name,
          'v': '5.131',
          'fields': 'activity, city, description, members_count, start_date, status'
} # содержит параметры для запроса: короткий адрес группы, версию API и поля, которые мы запрашиваем
```

```
response = requests.get(f'{BASE_URL}groups.getById', params=params, headers=HEADERS)
```

Функции vk.py:

- `get_common_info` – получает общую информацию о группе VK по ее имени и возвращает словарь с полученной информацией или `None`, если произошла ошибка при запросе.

- `get_members_info` и `get_wall_info` – принимают два аргумента: имя группы VK и информацию о группе и возвращают информацию группе, добавляя информацию об участниках или постах соответственно. Также необходимо отметить следующую особенность VK API: существует ограничения на количество записей и участников, которые могут быть получены за один запрос – 100 и 1000 соответственно. Поэтому чтобы получить информацию обо всех участниках группы или записях со стены, необходимо использовать параметры `offset` и `count`. Параметр `offset` указывает, с какого элемента начинать выборку, а параметр `count` указывает, сколько элементов нужно выбрать. Для ускорения процесса были использованы потоки, каждый из которых отправляет запрос на получение 100 записей / 1000 участников. Потоки отправляют запросы

одновременно, что позволяет получить информацию быстрее, чем если бы запросы отправлялись последовательно.

- `open_url` - используется для отправки запросов к API VK с использованием библиотеки `requests`. Функция принимает следующие аргументы: `params` - параметры запроса, `url` - адрес запроса, `result` - список для сохранения результатов запросов, `index` - индекс текущего запроса в списке, `offset` - смещение для выборки данных из API VK. Функция отправляет запрос методом GET с использованием переданных параметров и заголовков. Если ответ содержит ошибку `KeyError`, то функция будет повторять запрос с экспоненциальной задержкой до 5 раз.

- `thread_processing` - используется для запуска потоков для параллельной обработки запросов.

- `find_ban_words` - используется для проверки текста на наличие запрещенных слов с помощью регулярного выражения и списка запрещенных слов, а `get_ban_words` - для чтения списка запрещенных слов из файла **`ban_words.txt`** и возвращения его в виде списка.

Файл **`metrics.py`** содержит функции для расчета метрик эффективности группы: ER, LR и TR.

Файл **`graph.py`** содержит функции для отрисовки графиков.

В файле **`groups.py`** определена функция `update_group`, которая обновляет данные группы в базе данных. Сначала функция получает объект группы по ее `id`, затем извлекает имя группы из ссылки и получает общую информацию о группе, информацию о постах и участниках с помощью соответствующих функций из `analytics.services.vk`. Далее функция вычисляет метрики с помощью функций из `analytics.services.metrics` и функции для расчета средних значений `count_average` и создает графики с помощью функции `create_graphs` из `analytics.services.graph`. Затем функция обновляет данные группы в базе данных, добавляя вычисленные метрики и графики, а также устанавливает статус группы как "готово".

Полная структура проекта представлена на рисунках 3.1 и 3.2.

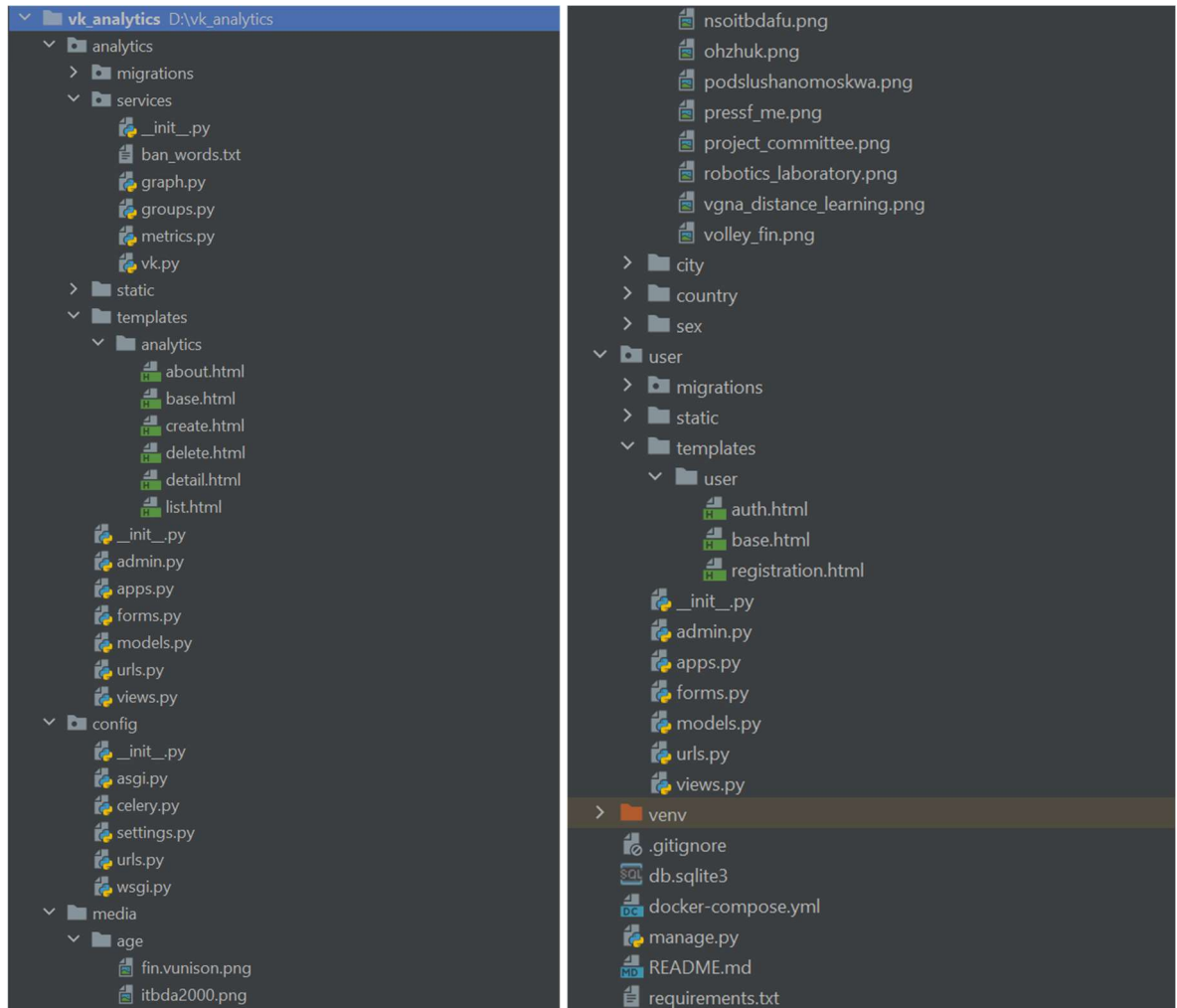


Рисунок 3.1 и 3.2 Структура приложения

3.3 Оценка результатов

Для запуска приложения в первый раз необходимо провести следующие манипуляции:

1. Сначала нужно выполнить команду `docker compose up -d`, чтобы запустить контейнер Redis, который является брокером (для этого должен быть установлен Docker или Redis).
2. Затем нужно создать виртуальную оболочку и запустить ее с помощью команды `.\venv\scripts\activate`.
3. Следующим шагом нужно установить все необходимые пакеты и библиотеки с помощью команды `pip install -r requirements.txt`.
4. После этого нужно провести все необходимые миграции с помощью

команды `python manage.py migrate`.

5. Для создания суперюзера (админа) нужно выполнить команду `python manage.py createsuperuser`.

После подготовки приложения можно запустить его:

1. Для запуска Celery нужно выполнить команду `celery -A config worker --pool solo -l INFO` в одном терминале.

2. Для запуска Django нужно выполнить команду `python manage.py runserver` в другом терминале.

3. После этого открываем в браузере ссылку `http://127.0.0.1:8000/`

Итак, приложение запущено. Рассмотрим его работу от лица администратора, чтобы затронуть весь функционал программы (авторизованному пользователю недоступна панель администратора).

Первая страница, которая открывается – страница авторизации (рисунок 3.3). Запрашивает эл. почту и пароль пользователя. Если эл. почты введена некорректно, ее нет в БД или же пароль не совпадает с паролем в БД, то вход выполнен не будет (рисунок 3.4–3.5). При корректной авторизации происходит переход на главную страницу сайта - страницу с добавленными сообществами пользователя.

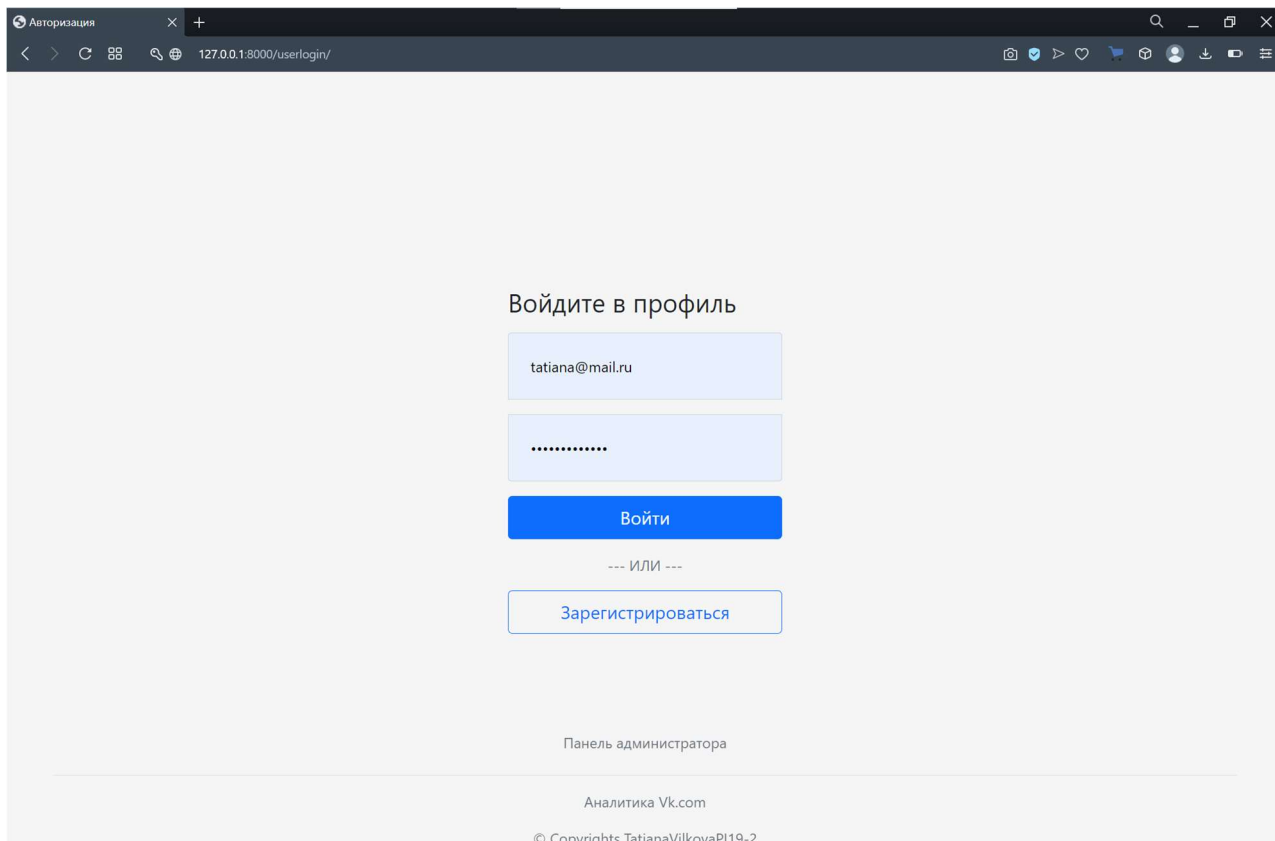


Рисунок 3.3 Страница «Авторизация пользователя»

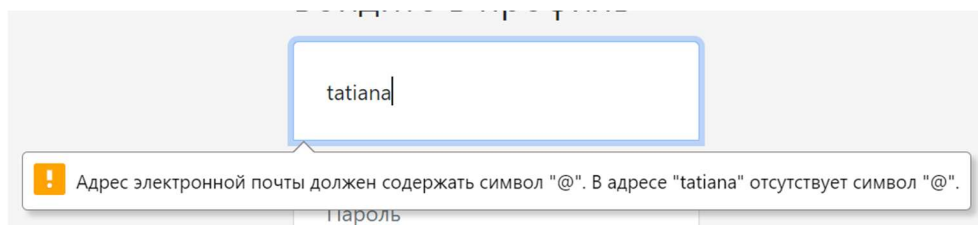


Рисунок 3.4 Реакция приложения на неверный формат почты

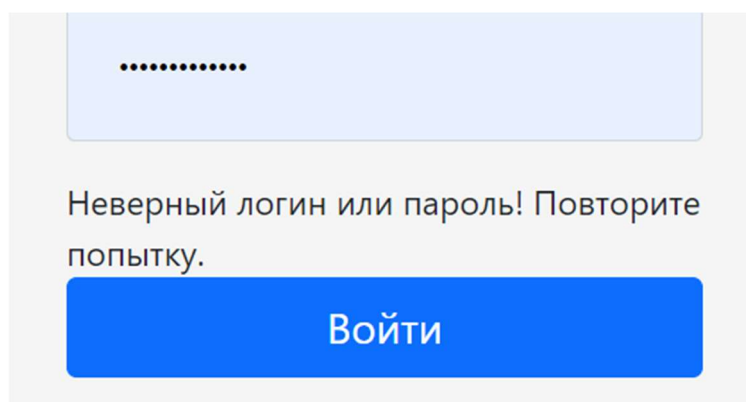
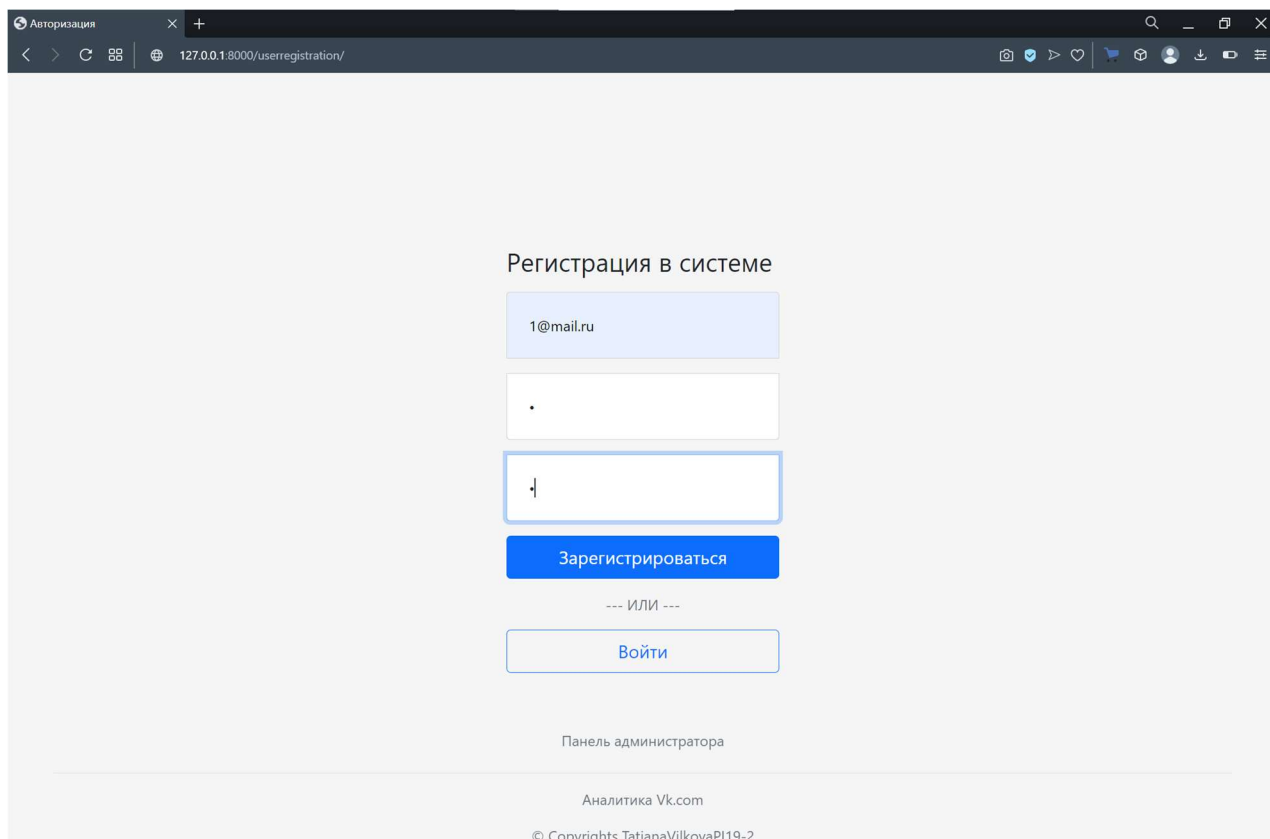


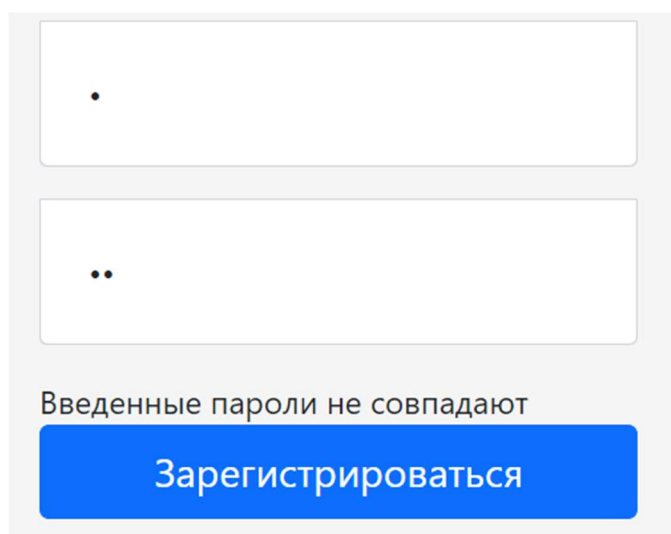
Рисунок 3.5 Неуспешная авторизация

Неавторизованным пользователям доступна регистрация (рисунок 3.6). При регистрации производится проверка на корректность формата эл. почты, на существование пользователя с заданной эл. почтой, сверка введенных паролей (рисунки 3.7–3.8). При успешной регистрации также происходит переход на главную страницу сайта - страницу с добавленными сообществами пользователя.



The screenshot shows a web browser window with the title 'Авторизация' and the address bar displaying '127.0.0.1:8000/userregistration/'. The main content area is titled 'Регистрация в системе'. It contains a form with three input fields: the first contains '1@mail.ru', the second contains a single dot, and the third contains a single dot and a vertical bar. Below the inputs is a blue button labeled 'Зарегистрироваться'. Underneath this button is the text '--- ИЛИ ---' and a light blue button labeled 'Войти'. At the bottom of the page, there is a link 'Панель администратора', a footer with 'Аналитика Vk.com', and a copyright notice '© Copyrights TatianaVilkovaPI19-2'.

Рисунок 3.6 Страница «Регистрация пользователя»



This close-up view of the registration form shows two input fields. The first field contains a single dot, and the second field contains two dots. Below the fields, the text 'Введенные пароли не совпадают' (Entered passwords do not match) is displayed. At the bottom is a blue button labeled 'Зарегистрироваться'.

Рисунок 3.7 Несовпадение введенных паролей при регистрации

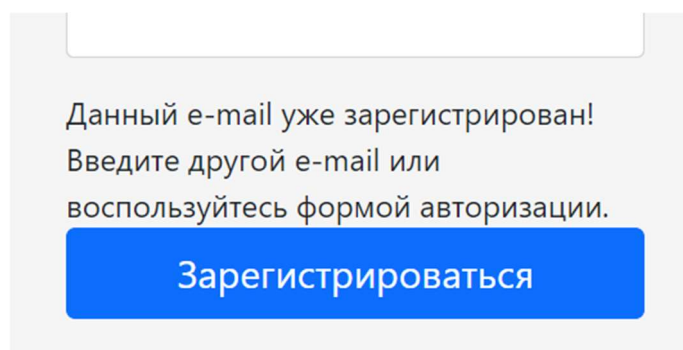


Рисунок 3.8 Существование пользователя с заданной почтой при регистрации

Главная страница – страница с добавленными пользователем сообществами (рисунок 3.9). При открытии данной страницы показывается таблица со всеми ранее добавленными сообществами и их основными характеристиками. Таблицу мы можем отсортировать по названию, ER, LR, TR и количеству подписчиков; перейти по ссылке на страницу сообщества ВКонтакте; обновить информацию (при статусе обновляется эта кнопка становится неактивной); найти в таблице нужное нам сообщество по названию. Некоторые из функций представлены на рисунках 3.10–3.11.

#	Ссылка	Название	ER	TR	LR	Подписчики	Статус	Управление
1	https://vk.com/nsoitbdufu	НСО ИТиАБД Финансовый Университет	0,84	0,02	0,66	693	Обновлено	
2	https://vk.com/ftiabrd_official	ИТиАБД Финансовый университет	1,38	0,02	0,84	1273	Обновлено	
3	https://vk.com/project_committee	Проектный комитет Финансовый университет	10,27	0,47	8,5	1268	Обновлено	
4	https://vk.com/vgna_distance_learning	Дистанционное образование Финансовый Университет	2,14	1,25	0,8	28	Обновлено	
5	https://vk.com/fin.vunison	В Унисон Финансовый университет	3,69	0,16	3,25	2196	Обновлено	
6	https://vk.com/volley_fin	Волейбол Финансовый университет	6,46	0,67	5,22	324	Обновлено	
7	https://vk.com/ftbda2022	1 курс 2022 ИТиАБД Финансовый университет	6,2	0,09	4,21	1201	Обновлено	
8	https://vk.com/pressf_me	PressF Факультет ИТиАБД	2,32	0,08	2,04	539	Обновлено	
9	https://vk.com/ftbda2000	ИТиАБД, вперед, урал	1,65	0,05	1,44	3935	Обновлено	
10	https://vk.com/robotics_laboratory	Robotics Laboratory	10,95	0,11	9,89	66	Обновлено	

3.9 Страница «Список добавленных сообществ»

Обновляется



3.10 Обновление информации о сообществе

+ Добавить		ИТиАБД						Поиск
#	Ссылка	Название	ER	IR	LR	Подписчики	Статус	Управление
1	https://vk.com/nsoitbdafu	НСО ИТиАБД Финансовый Университет	0,84	0,02	0,66	693	Обновлено	
2	https://vk.com/itiabd_official	ИТиАБД Финансовый университет	1,38	0,02	0,84	1273	Обновлено	
3	https://vk.com/itbda2022	1 курс 2022 ИТиАБД Финансовый университет	6,2	0,09	4,21	1201	Обновлено	
4	https://vk.com/pressf_me	PressF Факультет ИТиАБД	2,32	0,08	2,04	539	Обновлено	
5	https://vk.com/itbda2000	ИТиАБД, вперёд, ура!	1,65	0,05	1,44	3935	Обновлено	

3.11 Поиск сообщества по названию

При попытке удалить сообщество открывается страница с подтверждением удаления (рисунок 3.12). После подтверждения вся информация о сообществе удаляется из БД и происходит переход на главную страницу сайта.

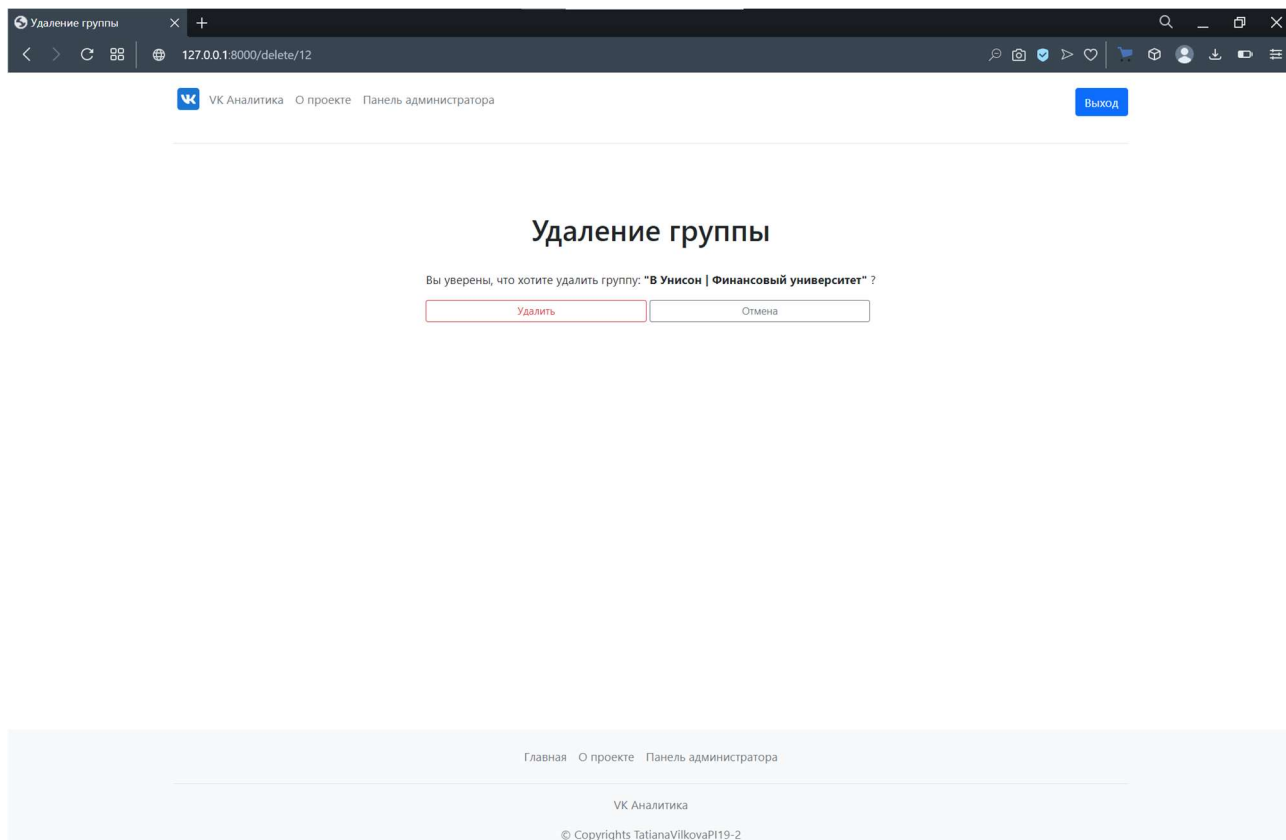


Рисунок 3.12 Страница «Удаление группы»

При нажатии на кнопку «добавить» открывается страница с добавлением сообщества (рисунок 3.13). Производится проверка введена ли ссылка, правильно ли она введена, существует ли сообщество, и не добавлено оно ранее (рисунки 3.14–3.17).

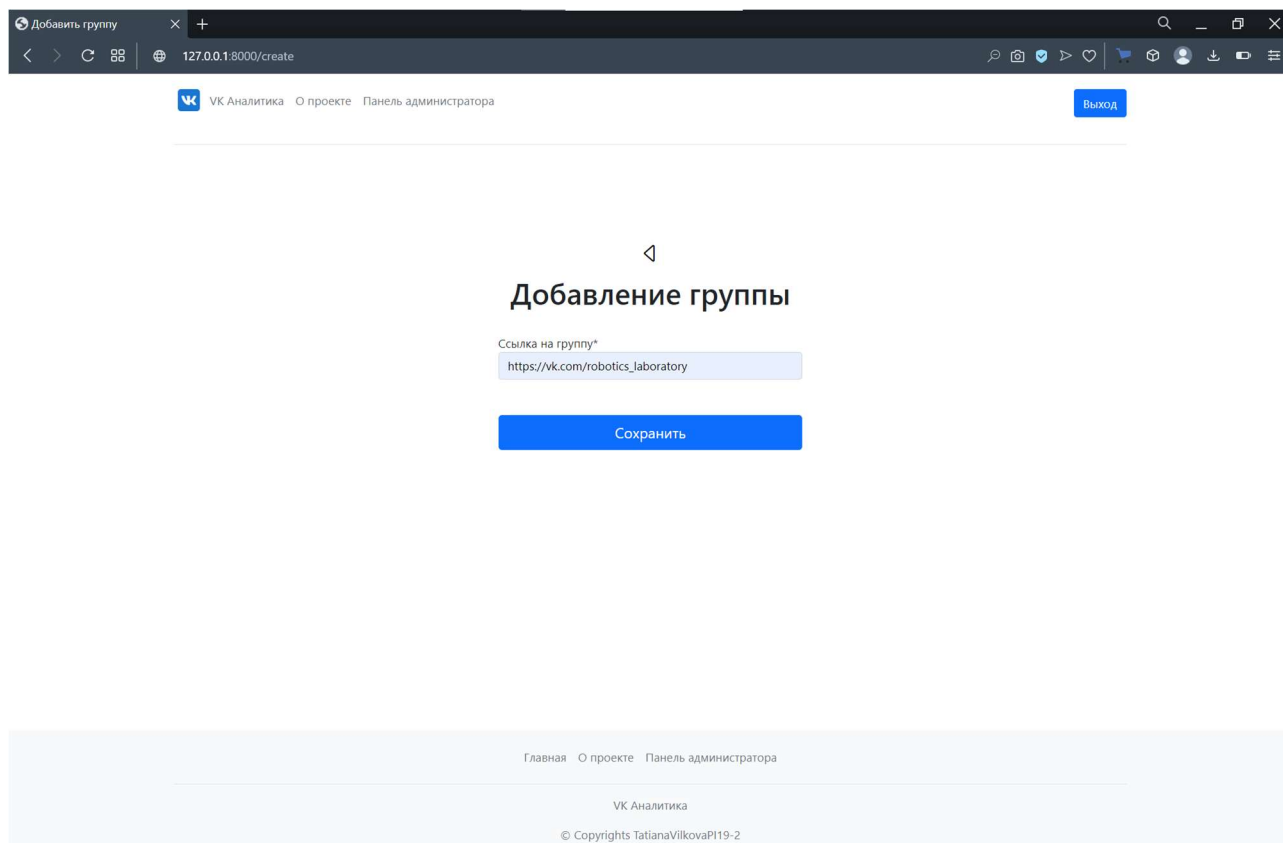


Рисунок 3.13 Страница «Добавление группы»

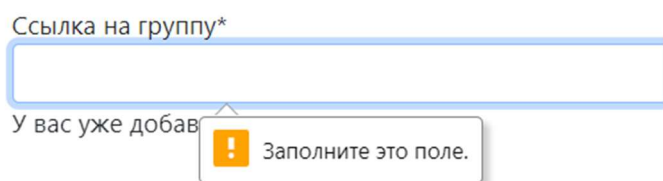


Рисунок 3.14 Реакция приложения на незаполненное поле ссылки

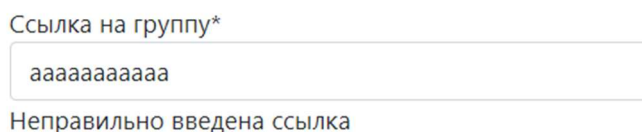


Рисунок 3.15 Реакция приложения на неверный формат ссылки

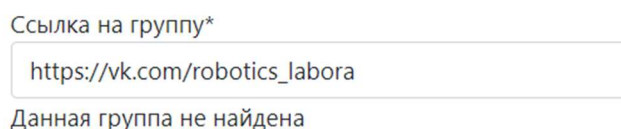


Рисунок 3.16 Проверка на существование группы

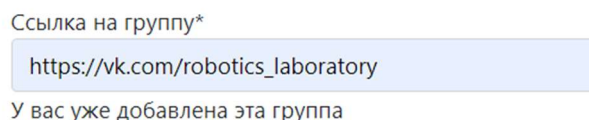


Рисунок 3.17 Проверка на существование в БД

Если нажать на название сообщества, мы переходим на страницу с детальной информацией по конкретному сообществу (рисунок 3.18). Здесь представлены все основные метрики, необходимые для анализа активности аудитории и диаграммы распределения аудитории по полу, городам, странам и возрасту (рисунок 3.19–3.21). Диаграммы автоматически перелистываются каждые 3 секунды. Со страницы с детальной информацией также доступно обновление информации по сообществу и его удаление (после этих действий вы будете перенаправлены на главную страницу)

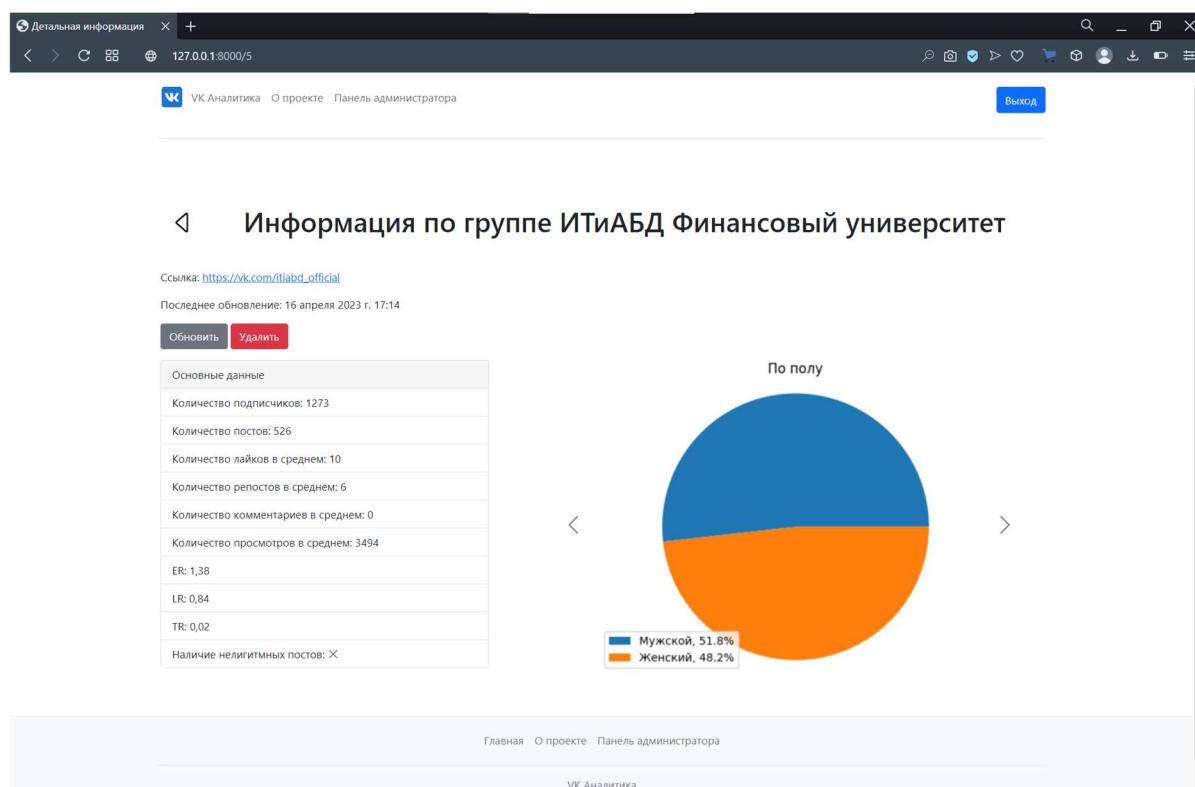


Рисунок 3.18 Страница «Детальная информация по группе»

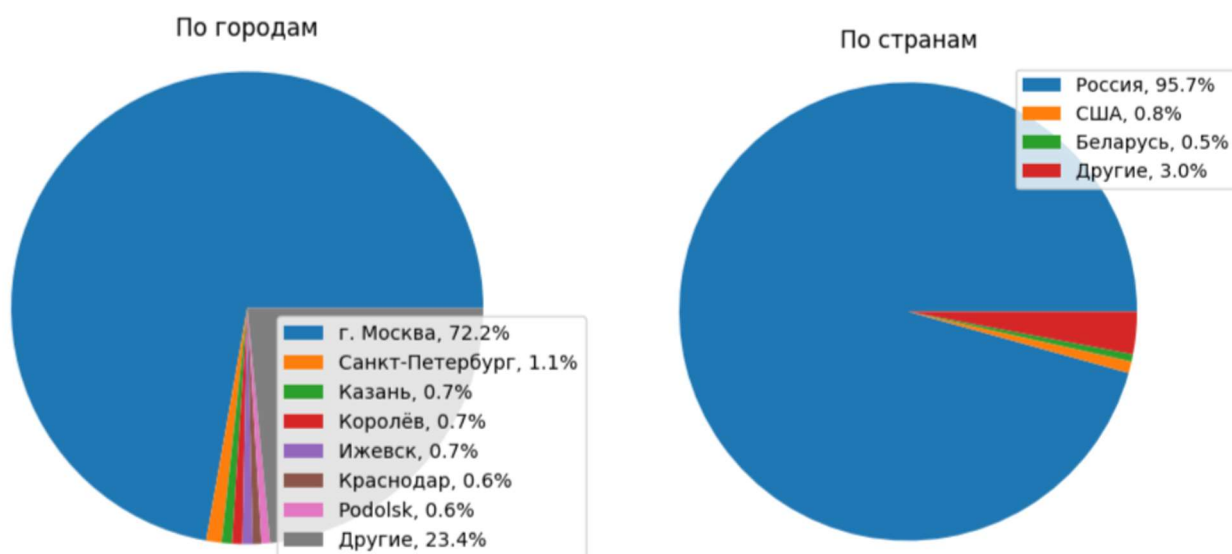


Рисунок 3.19 и 3.20 Распределение аудитории по городам и странам

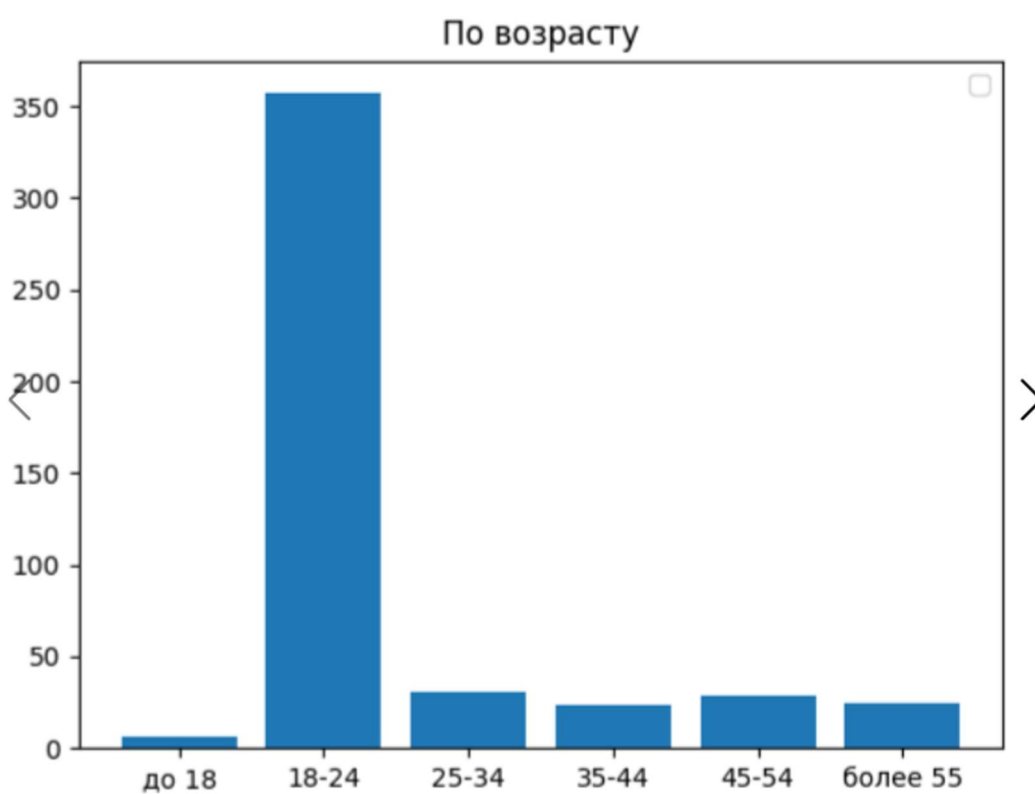


Рисунок 3.21 Распределение аудитории по возрасту

Во кнопкам в шапке и футере сайта можно также спокойно перемещаться между страницами приложения: перейти на главную страницу сайта (список добавленных сообществ), на страницу о проекте (рисунок 3.22), содержащую основную информацию о разработанном приложении и контактную информацию, выйти из личного кабинета (тогда мы попадем на страницу авторизации) и открыть панель администратора (рисунок 3.23).

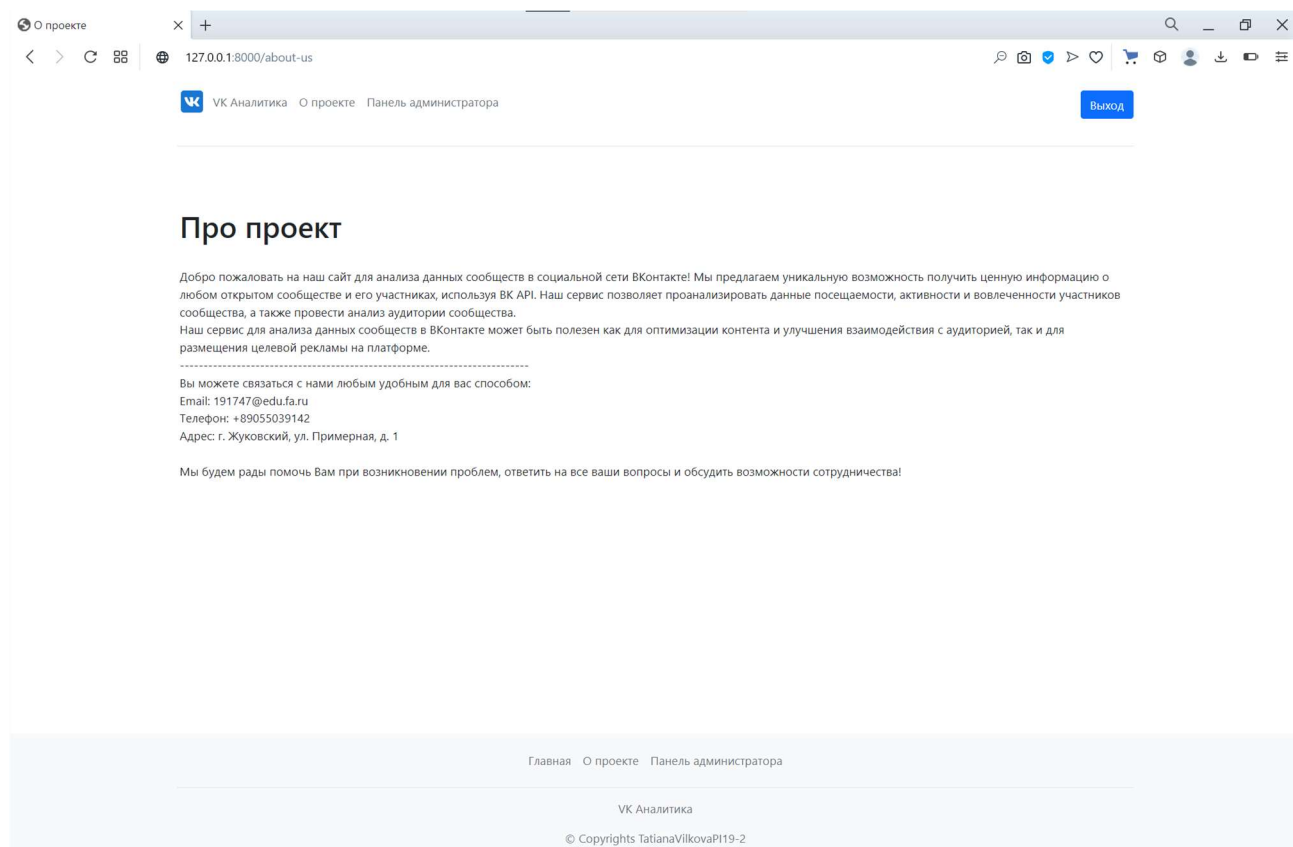


Рисунок 3.22 Страница «О проекте»

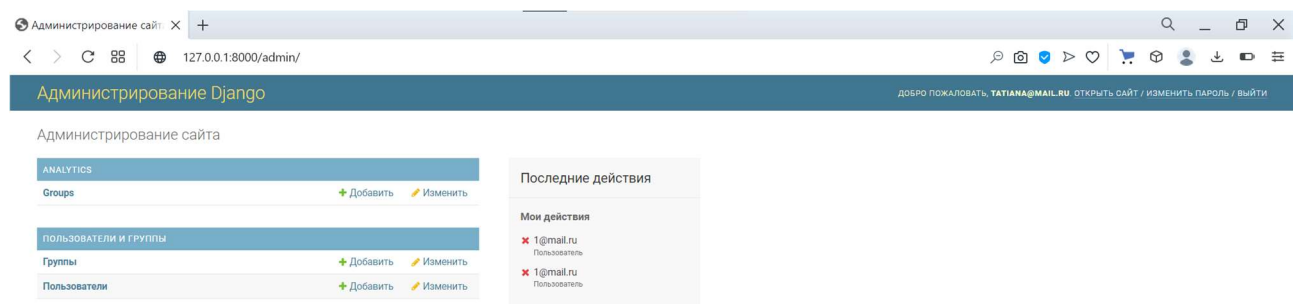


Рисунок 3.23 Страница «Панель администратора»

В качестве панели администратора была использована страница администрирования Django, которая предоставляет удобный интерфейс для управления приложением. Она доступна по адресу /admin и может быть настроена для удовлетворения конкретных потребностей приложения.

Стандартная страница администрирования Django предоставляет возможности для управления пользователями, группами, моделями данных, настройками приложения и многое другое.

Хочется напомнить, что панель администратора доступна только пользователям, имеющим статус суперюзера. Его возможно создать в терминале при помощи команды `python manage.py createsuperuser`, или же уже существующий администратор может наделить этими правами другого пользователя в панели администратора.

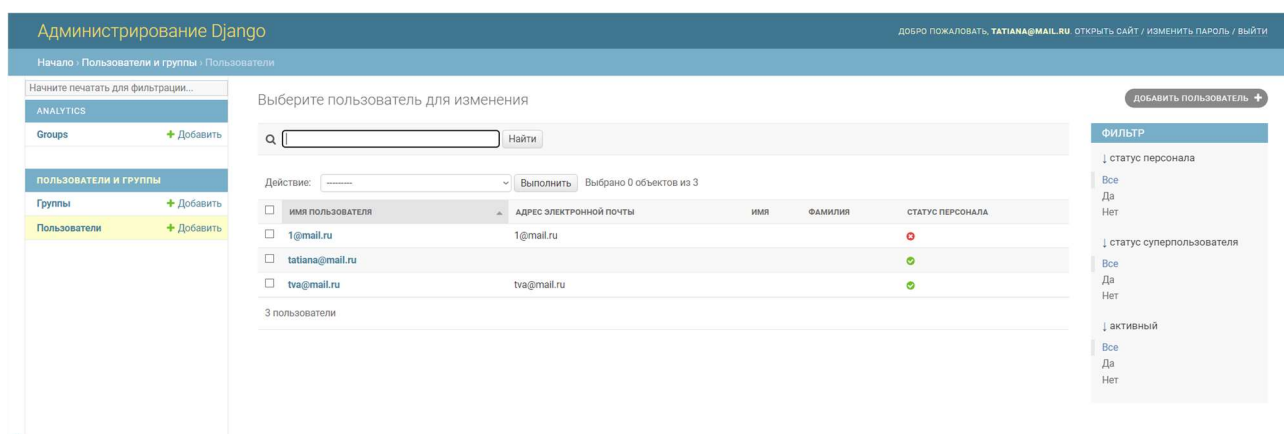


Рисунок 3.24 Редактирование БД пользователей

На рисунке 3.24 представлена страница с БД пользователей. Мы можем добавить или удалить пользователей. Нажимаем на конкретного пользователя открывается страница редактирования пользователя (рисунок 3.25). На данной странице можно изменять логин пользователя, пароль. Можно добавить персональную информацию, настроить права доступа, группы и конкретные права пользователя. Так же есть поля с датой и временем регистрации и последним входом. Пароли пользователей хранятся в зашифрованном виде.

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, TATIANA@MAIL.RU. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > Пользователи и группы > Пользователи > tva@mail.ru

Начните печатать для фильтрации...

ANALYTICS

Groups + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Изменить пользователь

ИСТОРИЯ

tva@mail.ru

Имя пользователя: tva@mail.ru

Обязательное поле. Не более 150 символов. Только буквы, цифры и символы @./!+/_

Пароль: алгоритм: pbkdf2_sha256 итерации: 390000 соль: yRjCv6***** хэш: GM9pzb*****

Пароли хранятся в зашифрованном виде, поэтому нет возможности посмотреть пароль этого пользователя, но вы можете изменить его используя эту форму.

Персональная информация

Имя:

Фамилия:

Адрес электронной почты: tva@mail.ru

Права доступа

☒ Активный

Отметьте, если пользователь должен считаться активным. Уберите эту отметку вместо удаления учетной записи.

☒ Статус персонала

Отметьте, если пользователь может входить в административную часть сайта.

☒ Статус суперпользователя

Указывает, что пользователь имеет все права без явного их назначения.

Рисунок 3.25 Редактирование пользователя

На рисунках 3.26-3.28 представлены страницы с редактированием БД сообщений.

Выберите group для изме

+

127.0.0.1:8000/admin/analytics/group/

Администрирование Django

ДОБРО ПОЖАЛОВАТЬ, TATIANA@MAIL.RU. ОТКРЫТЬ САЙТ / ИЗМЕНИТЬ ПАРОЛЬ / ВЫЙТИ

Начало > Analytics > Groups

Начните печатать для фильтрации...

ANALYTICS

Groups + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Выберите group для изменения

ДОБАВИТЬ GROUP +

q

Найти

Действие: Удалить выбранные groups

Выполнить

Выбрано 0 объектов из 10

10 groups

ФИЛЬТР

Статус

Все

В обработке

Готово

Рисунок 3.26 Редактирование БД сообщений

Group object (23) | Изменить

127.0.0.1:8000/admin/analytics/group/23/change/

Администрирование Django

Начало > Analytics > Groups > Group object (23)

Начните печатать для фильтрации...

ANALYTICS

Groups + Добавить

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

Группы + Добавить

Пользователи + Добавить

Изменить group

Group object (23) ИСТОРИЯ

Ссылка на группу:

Название:

Кол-во подписчиков:

Кол-во лайков:

Кол-во комментариев:

Кол-во просмотров:

Кол-во репостов:

Кол-во постов:

Среднее кол-во лайков:

Среднее кол-во комментариев:

Среднее кол-во репостов:

Среднее кол-во просмотров:

Нелегитимные посты:

Информация об участниках:

```
{
  "female": 22,
  "male": 44,
  "city": {
    "Moscow": 33,
    "Cheboksary": 1,
    "Voronezh": 2,
    "London": 1,
    "Sergiev Posad": 1,
    "Smolensk": 1,
    "Istanbul": 1,
    "Veliky Ustyug": 1,
    "Saint Petersburg": 1,
    "Dallas": 1,
    "Chelyabinsk": 1,
    "Yasikha": 1,
    "country": {
      "Russia": 51,
      "United Kingdom": 1,
      "Turkey": 1,
      "Israel": 1,
      "USA": 1,
      "age": {
        "до 18": 0,
        "18-24": 16,
        "25-34": 1,
        "35-44": 1,
        "45-54": 0,
        "60nee 55": 2
      }
    }
  }
}
```

Рисунок 3.27 Редактирование сообщества

ER:

LR:

TR:

Age: ☐ Очистить
Изменить: Файл не выбран

City: ☐ Очистить
Изменить: Файл не выбран

Country: ☐ Очистить
Изменить: Файл не выбран

Sex: ☐ Очистить
Изменить: Файл не выбран

Статус:

User:

Рисунок 3.28 Редактирование сообщества

При авторизации в качестве обычного пользователя переход к панели администратора невозможен, недоступна кнопка «Панель администратора» (рисунок 3.29).

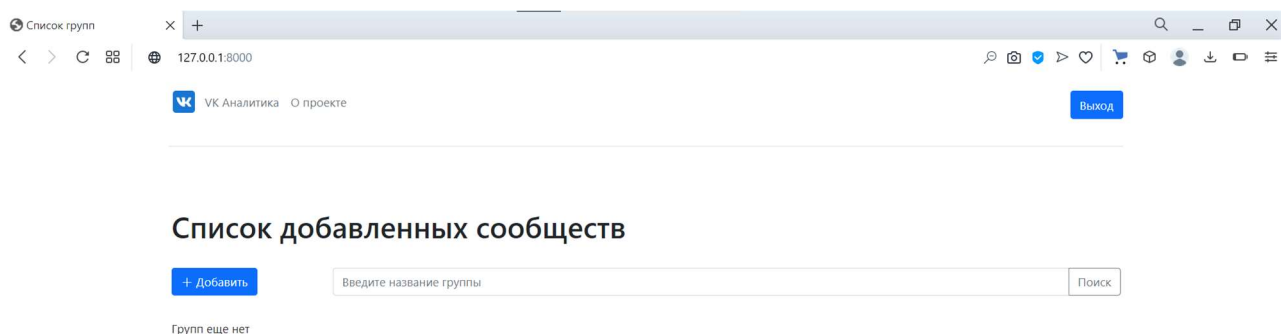


Рисунок 3.29 Главная страница обычного пользователя

Подводя итоги использования веб-приложения, можно с уверенностью сказать, что все функциональные и нефункциональные требования, поставленные в предыдущей главе, были выполнены в полном объеме. Это говорит о том, что разработка была завершена успешно, и приложение готово к использованию. Однако, не стоит забывать, что разработка программного обеспечения — это непрерывный процесс, и приложение может быть доработано и улучшено в будущем, чтобы соответствовать изменяющимся потребностям пользователей и рынка. К счастью, выбранные технологии и структура проекта обеспечивают гибкость и масштабируемость приложения, что позволяет добавлять новые функциональности и улучшать его производительность.

Например, можно добавить функцию получения отчета с собранной информацией и рассчитанными метриками по сообществам в формате .xlsx, .pdf для дальнейшего самостоятельного анализа или для загрузки в другие приложения для анализа данных.

Также же можно добавить модель классификации текста машинного обучения (NLP), которая будет определять наиболее популярные темы постов и их тональность (негативно или позитивные посты в основном). Это будет полезно для более глубокого анализа сообщества и позволит пользователям получать более точную информацию о том, что происходит в сообществе.

3.4 Выводы по главе 3

В данной главе работы были описаны выбранные средства разработки, такие как языки программирования, фреймворки, библиотеки и инструменты, а также обосновано, почему именно они были выбраны. Кроме того, была описана структура разработанного программного приложения, включая использованные модули и компоненты, их взаимосвязь между собой и взаимодействие с пользователем и базой данных. В рамках этой главы также был описан порядок применения приложения, то есть как пользователи будут использовать приложение, какой функционал будет доступен, какие данные будут вводиться и как они будут обрабатываться. Помимо этого, были предложены дальнейшие варианты развития приложения.

ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы было разработано веб-приложение для статистического анализа данных сообществ соцсети ВКонтакте. Разработанная программа позволяет собирать информацию о сообществах в социальной сети ВКонтакте, рассчитывать основные метрики активности аудитории сообществ (среднее количество лайков/комментариев/просмотров/репостов, показатели ER, TR, LR) и получать данные об целевой аудитории (пол, возраст, география), представлять эту информацию пользователю в удобном для него виде, в том числе визуализировать её с помощью круговых и столбчатых диаграмм, что позволяет быстро и наглядно оценить результаты исследований. В целом приложение значительно упрощает процесс сбора и анализа данных ВКонтакте.

В процессе написания выпускной квалификационной работы были изучены различные аспекты, связанные с предметной областью – тенденции развития социальных сетей, принципы анализа данных, содержащихся в них, алгоритмы получения этих данных с помощью различных методов, включая использование API. Кроме того, были изучены новые библиотеки, фреймворки и способы написания кода на языке Python, включая фреймворк Django, библиотеки requests, celery, matplotlib и др.

Для достижения поставленной цели были также решены следующие задачи:

1. Проведен аналитический обзор существующих решений по обработке и визуализации данных социальных сетей, были выделены основные преимущества и недостатки каждого из них, что позволило определить наиболее важные требования;
2. Определены основные требования к функциональности приложения, нефункциональные требования (такие как скорость работы, надежность и безопасность) и проработана архитектура будущего приложения;
3. Определен стек технологий, с обоснованием их выбора и в реализовано программное обеспечение.

Самым сложным и одновременно интересным при реализации программы было получение данных ВКонтакте при помощи API, так как при написании кода программы были обнаружены некоторые особенности методов VK API: они имеют ограничения на количество записей и участников, которые могут быть получены за один запрос и возвращают только первые 1000 участников группы и первые 100 записей со стены группы. Для получения полной информации пришлось искать пути решения этой проблемы: использовать дополнительные параметры в запросах и делать несколько запросов и далее объединять полученные данные. Также оказалось, что VK API имеет ограничения на количество запросов в единицу времени, поэтому при обработке большого количества данных необходимо учитывать эти ограничения и использовать задержки между запросами. Для ускорения работы запросы выполнялись параллельно, для чего были использованы библиотеки `celery` и `threading`.

Отдельно хочу отметить выводы, сделанные в ходе изучения предметной области. Анализ данных соцсетей и разработка приложений для этой цели – это перспективное направление. Популярность соцсетей, а соответственно объемы данных производимыми ими в последние годы стремительно растет. Приложения, работающие с социальными сетями, могут быть полезными для многих областей, включая маркетинг, исследования общественного мнения, социологию и т. д. Поэтому и разработанное в рамках ВКР приложение еще долгие годы будет актуально.

Все требования, поставленные для разработки, реализованы и протестированы. Достоинствами полученного решения можно считать:

- простой, понятный, удобный и приятный взгляду интерфейс;
- приложение анализирует и пользовательскую аудиторию (по географии, полу и возрасту), и аналитику активности (уровень вовлеченности и т. д.);
- приложение работает с одной из самых перспективных социальных сетей в России - ВКонтакте, и удобно для российских пользователей.

Разработанное решение будет полезно для специалистов, занимающихся рекламой, которые благодаря ему смогут более эффективно размещать свои объявления в сообществах, имеющих наиболее активную аудиторию и понимать, в каких сообществах находится их целевая аудитория.

Хотелось бы отметить, что на данный момент веб-приложение работает только с одной социальной сетью - Вконтакте. Однако, для дальнейшего развития и расширения возможностей приложения, можно рассмотреть возможность добавления функций для анализа каналов Telegram. Telegram является одним из наиболее популярных мессенджеров в России и мире, и его пользовательская база постоянно растет, что делает его перспективным объектом для анализа и использования в маркетинговых исследованиях.

Все поставленные задачи выпускной квалификационной работы реализованы, что позволяет сделать вывод о достижении цели работы. Считаю выпускную квалификационную работу успешно завершённой.

Данная работа выполнена мною самостоятельно.

«____» _____ 2023г.

(дата сдачи работы – заполняется от руки)

_____ Волкова Т. А.

(подпись автора)

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Законодательные и нормативные акты

1. ГОСТ 7.82 – 2001. Библиографическая запись. Библиографическое описание электронных ресурсов. Общие требования и правила составления. – М.: ИПК Издательство стандартов, 2001. –21 с.
2. ГОСТ Р 7.0.5-2008 "Библиографическая запись. Библиографическое описание. Общие требования и правила составления"
3. ГОСТ 7.32 – 2001. Система стандартов по информации, библиотечному и издательскому делу. Отчёт о научно-исследовательской работе. Структура и правила оформления. – М.: ИПК Издательство стандартов, 2001. – 21 с.
4. ГОСТ Р 7.0.100-2018 "Библиографическая запись. Общие требования и правила составления"
5. ГОСТ Р ИСО 9241-151-2014 Эргономика взаимодействия человек-система. Часть 151 Руководство по проектированию пользовательских интерфейсов сети Интернет [Текст] / М.: Стандартинформ, 2015. - 28 с.
6. Федеральный закон "О персональных данных" № 152-ФЗ [Электронный ресурс] / Гарант. Режим доступа: http://www.consultant.ru/document/cons_doc_LAW_61801/
7. Техническое задание по ГОСТ 34- разделы 4-8. [Электронный ресурс] / RuGost. Режим доступа: <http://rugost.com/standards/gost-34/>

Интернет-ресурсы

8. Digital 2023 Global Overview Report / We Are Social, Hootsuite [Электронный ресурс]. Режим доступа: <https://datareportal.com/reports/digital-2023-global-overview-report> (дата обращения: 25.02.2023).
9. Digital 2023: Russian Federation / We Are Social, Hootsuite [Электронный ресурс]. Режим доступа: <https://datareportal.com/reports/digital-2023-russian-federation> (дата обращения: 25.02.2023).
10. Роль социальных сетей в интернет-маркетинге [Электронный ресурс]. Режим доступа: <http://elibrary.ru> (дата обращения: 25.02.2023).

11. Popsters - инструмент аналитики социальных медиа [Электронный ресурс]. Режим доступа: <https://popsters.ru/> (дата обращения: 25.02.2023).
12. Minter.io - инструмент аналитики Instagram [Электронный ресурс]. Режим доступа: <https://minter.io/> (дата обращения: 25.02.2023).
13. Foller.me - инструмент аналитики Twitter [Электронный ресурс]. Режим доступа: <https://foller.me/> (дата обращения: 25.02.2023).
14. Publer.io - инструмент управления социальными медиа [Электронный ресурс]. Режим доступа: <https://publer.io/> (дата обращения: 25.02.2023).
15. Pur.social - инструмент управления социальными медиа [Электронный ресурс]. Режим доступа: <https://pur.social/> (дата обращения: 25.02.2023).
16. Borgi.ru - онлайн-инструмент для исследования рынка [Электронный ресурс]. Режим доступа: <https://borgi.ru/> (дата обращения: 25.02.2023).
17. Media-vk.ru - инструмент для анализа групп и сообществ ВКонтакте [Электронный ресурс]. Режим доступа: <https://media-vk.ru/> (дата обращения: 25.02.2023).
18. VK Admin - официальный инструмент управления группами и сообществами ВКонтакте [Электронный ресурс]. Режим доступа: <https://vk.com/vkadmin> (дата обращения: 25.02.2023).
19. Статистика интернета и соцсетей на 2023 год - цифры и тренды в мире и России [Электронный ресурс]. Режим доступа: <https://web-canape.ru> (дата обращения: 25.02.2023).
20. Знакомство с VK API / VK [Электронный ресурс]. Режим доступа: https://vk.com/dev/first_guide (дата обращения: 25.03.2023).
21. VK API Methods / VK [Электронный ресурс]. Режим доступа: <https://dev.vk.com/method> (дата обращения: 25.03.2023).
22. Описание методов API / VK [Электронный ресурс]. Режим доступа: https://vk.com/pages?oid=-1&p=Описание_методов_API (дата обращения: 25.03.2023).

23. VK Developers / VK [Электронный ресурс]. Режим доступа: <https://dev.vk.com> (дата обращения: 25.03.2023).
24. Документация Python 3 / Python [Электронный ресурс]. Режим доступа: <https://docs.python.org/3/index.html> (дата обращения: 15.04.2023).
25. Документация Django / Django [Электронный ресурс]. Режим доступа: <https://docs.djangoproject.com/en/4.2/> (дата обращения: 15.04.2023).
26. Документация Bootstrap / Bootstrap [Электронный ресурс]. Режим доступа: <https://bootstrap-4.ru/> (дата обращения: 15.04.2023).
27. Документация SQLite / SQLite [Электронный ресурс]. Режим доступа: <https://www.sqlite.org/docs.html> (дата обращения: 15.04.2023).
28. Документация Celery / Celery [Электронный ресурс]. Режим доступа: <https://docs.celeryq.dev/en/stable/getting-started/introduction.html> (дата обращения: 15.04.2023).
29. Redis Documentation / Redis [Электронный ресурс]. Режим доступа: <https://redis.io/documentation> (дата обращения: 15.04.2023).
30. Документация Matplotlib / Matplotlib [Электронный ресурс]. Режим доступа: <https://matplotlib.org/stable/index.html> (дата обращения: 25.04.2023).
31. Многопоточность на примерах – модуль threading / Python-Scripts [Электронный ресурс]. Режим доступа: <https://python-scripts.com/threading> (дата обращения: 25.04.2023).
32. Draw.io / Draw.io [Электронный ресурс]. Режим доступа: <https://draw.io> (дата обращения: 25.04.2023).

Учебная и научная литература

33. Барсуков, В.А. Интеграция социальной сети «ВКонтакте» в системы управления сайтами и приложениями. - 2018. - 65с.
34. Дронов В. Django 2.1. Практика создания веб-сайтов на Python. - Москва: BHV, 2019. – 672 с.
35. Гаврилов М. Django. Практика создания веб-сайтов на Python. - Москва: ДМК Пресс, 2019. - 320 с.

36. Шевченко А. PyCharm. Среда разработки на языке Python. - СПб: БХВ-Петербург, 2019. - 336 с.
37. Гриффитс Д. Bootstrap 4. Руководство по созданию веб-сайтов. - Москва: ДМК Пресс, 2019. - 352 с.
38. Мэттингли Д. Python для сложных задач. - Москва: ДМК Пресс, 2019. - 528 с.
39. Вандер Плас Дж. Python для сложных задач. Наука о данных и машинное обучение. - Москва: ДМК Пресс, 2019. - 528 с.
40. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. - Москва: Вильямс, 2016. - 1328 с.
41. Маккинни У., Уайлд Дж. Python и анализ данных. - Москва: ДМК Пресс, 2019. - 496 с.

ПРИЛОЖЕНИЕ

Архив со всеми файлами приложения доступен по ссылке:
<https://disk.yandex.ru/d/RecZqVGVryt5GA>