**Слайд 1 – Титульный слайд**

Здравствуйте, уважаемый председатель и члены государственной экзаменационной комиссии, вашему внимаю представляется дипломная работа на тему: «Разработка программного обеспечения статистического анализа социальных сетей», выполненная студенткой группы ПИ19-2 Волкова Татьяна Алексеевна.

**Слайд 2 – Актуальность ВКР**

Актуальность разработки связана с интенсивным развитием социальных сетей и соответственно с ростом объемов данных, производимых ими (пользователи добровольно делятся своей личной информацией, которая содержит огромное количество данных о них: пол, возраст, профессия, социальный статус, место работы или проживания, интересы и др). Большое количество людей использует социальные сети для продажи своих товаров и услуг, привлечения новых клиентов и оценки конкурентов в социальных сетях. 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, место жительства 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 и скорректировать его, чтобы сделать группу более привлекательной для потенциальных клиентов.

В России в 2022 году аудитория многих иностранных соцсетей упала (это можно наблюдать на картинке на слайде), это связано с тем, что в настоящее время мировое сообщество накладывает на Россию санкции в связи с проведением РФ спецоперации на Украине. Эта аудитория перешла в российские соцсети, а конкретно в ВКонтакте и Telegram.

**Слайд 3 – Цели и задачи**

Основная цель работы – облегчение поиска подходящего для рекламы сообщества за счет создания алгоритмического и программного обеспечения статистического анализа и визуализации статистики сообществ социальной сети Вконтакте. Полученная информация может быть использована рекламодателями для адаптации рекламных кампаний к конкретной аудитории и для выбора сообщество для рекламы с наиболее активной аудиторией.

Для достижения цели необходимо решить ряд задач:

1. изучить предметную область (какие соцсети наиболее популярны на данный момент, какие данные производятся соцсетями, для чего их можно использовать);
2. изучить способы получения данных соцсетей (апи, через элементы хтмл страниц, сторонние сервисы);
3. провести сравнительный анализ существующих решений по обработке и визуализации данных соцсетей, выделить их + и - и на его основе этого определить необходимый функционал (оценить потребности и поставить задачи)
4. составить требования и продумать архитектуру приложения;
5. выбрать подходящие для разработки инструментальные средства и непосредственно реализовать ПО.

**Слайд 4 - Объект и предмет исследования**

Самой популярной социальной сеть в России на данный момент является ВКонтакте, в связи с чем объектом исследования моей выпускной работы были выбраны ее сообщества, Предметом является анализ и визуализация их данных.

**Слайд 5 – Проектирование приложения: определение требований**

Начинать разработку ПО необходимо с его проектирования. Первый этап — это составление требований к системе. Проанализировав предметную область и существующие решения для анализа данных соцсетей, можно определить следующие требования – разрабатываемое ПО должно:

1 быть реализовано в виде веб-приложения для удобства и доступности со всех устройств;

2 парсить данные сообществ Вконтакте и возвращать пользователю информацию с помощью, которой он может оценить сообщество: рассчитанные метрики активности (количество подписчиков, постов, лайков/репостов/просмотров/комментариев в среднем, Engagement Rate (ER) — это коэффициент вовлеченности аудитории (данная метрика отображает процент пользователей, проявивших активность у публикаций/поста; Talk Rate (TR) - коэффициент коммуникабельности и Love Rate (LR) - коэффициент привлекательности (аналогичны ER, но учитывает только количество комментариев и лайков соответственно);

3 проверять наличие нелегитимных постов;

4 визуализировать распределение аудитории по возрасту, полу, стране и городу проживания

**Слайд 6 – Проектирование приложения: определение требований**

5 предоставлять авторизацию и регистрацию пользователей.

6 обеспечить разделение прав доступа соответственно ролям:

• 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.

Также были определены и нефункциональные требования, например, безопасность (данные хранятся в зашифрованном виде) или юзабилити (удобный интерфейс)

**Слайд 7 – Проектирование приложения: архитектура приложения**

Основным фреймворком для созданного мною приложения был выбран Django, использующий архитектурный паттерн Model-View-Template. 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. Каждое приложение в Django представляет собой набор моделей, представлений и шаблонов. В проектируемом ПО два приложения: "user" (будет отвечать за регистрацию и авторизацию пользователей) и "analytics" (будет отвечать за работу с сообществами).

На слайде представлена архитектура разработанного приложения, основанная 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 django по умолчанию): 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: временное хранилище задач.

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.

**Слайд 8 – Проектирование приложения: описание БД**

Модели БД имеет всего 2 сущности.

1 auth\_user – таблица, которая хранит информацию о пользователях (логин, пароль в зашифрованном виде, email, является ли пользователь администратором и др.).

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. Связана с таблицей auth\_user с помощью внешнего ключа user\_id.

**Слайд 9 – Реализация приложения: технологический стэк**

На данном слайде представлены средства, использованные для разработки. Некоторые из них я уже упоминала при описании архитектуры. Python – основной язык разработки. PyCharm – среда разработки. Django и Django REST API – основной фреймворк для веб-приложения. SQLite – субд для хранения данных, полученных из ВК. Html, css, Bootstrap – набор инструментов для верстки сайтов. Redis - временное хранилище задач. Библиотеки Celery (для фоновых задач), matplotlib (для визуализации), requests (для работы с VK API), threading (для параллельных процессов).

**Слайд 10 – Реализация приложения: API VK** (серверная часть)

API (Application Programming Interface) — это набор программных интерфейсов, протоколов и инструментов, которые позволяют различным приложениям взаимодействовать друг с другом. Для выгрузки необходимой информации по группам были использованы следующие методы VK API:

- 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.

При помощи библиотеки requests и GET запросов обращалась к этим методам. Пример на слайде.

BASE\_URL - это строка, содержащая базовый URL точки доступа API VK.

HEADERS - это словарь, содержащий токен авторизации для доступа к API VK.

params - это словарь, содержащий параметры для запроса API. Параметр group\_id указывает ID группы, информацию о которой запрашивают. Параметр v указывает версию API VK, которую следует использовать. Параметр fields указывает поля объекта группы, которые следует включить в ответ.

Метод requests.get() отправляет HTTP GET-запрос к API VK с указанными параметрами и заголовками, и сохраняет ответ в переменной response.

Во время разработки была выявлена следующая проблема: имеются 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.

**Слайд 11–18 – Реализация приложения: интерфейс. Страница «…»** (клиентская часть)

Рассмотрим интерфейс разработанного приложения. На картинке видно, что он выполнен в нейтральных цветах и минималистическом стиле.

Первая страница, которая открывается – страница авторизации. 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. При корректной авторизации происходит переход на страницу с добавленными сообществами пользователя. Неавторизованным пользователям доступна регистрация. При регистрации производится проверка на корректность формата эл. почты, на существование пользователя с заданной эл. почтой, сверка введенных паролей.

При открытии страницы «список добавленных сообществ» показывается таблица со всеми ранее добавленными пользователем 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; удалить информацию о сообществе.

При нажатии на кнопку «добавить» открывается страница с добавлением сообщества. Производится проверка введена ли ссылка, правильно ли она введена, существует ли сообщество, и не добавлено оно ранее.

Мы также можем увидеть основную информацию, рассчитанные метрики и диаграммы распределения аудитории, нажав на название сообщества в таблице. Диаграммы автоматически перелистываются каждые 3 секунды.

Также веб-приложение имеет страничку о проекте, где можно найти контактную информацию.

Стандартная страница администрирования Django предоставляет возможности для управления пользователями, группами, моделями данных, настройками приложения и многое другое. Панель администратора доступна, если вы авторизовались как админ.

Во кнопкам в шапке и футере сайта можно также спокойно перемещаться между страницами приложения.

**Слайд 19 – Результаты и выводы**

Разработанная программа позволяет собирать информацию о сообществах в социальной сети ВКонтакте, рассчитывать основные метрики активности аудитории сообществ 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.

(К достоинствам можно отнести) Программа имеет РЯД достоинств и возможностей для дальнейшего развития. К достоинствам можно отнести, что приложение анализирует и пользовательскую аудиторию, и аналитику активности (зачастую существующие решения предоставляют что-то одно); а также то, что оно работает с одной из самых перспективных социальных сетей в России, и подходит для российских пользователей.

Кроме того, хотелось отметить, что приложение имеет структурированный исходный код и архитектуру, которая позволяет добавлять новые функционал. 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. Также, например, модель классификации текста машинного обучения (NLP), которая будет определять наиболее популярные темы постов и их тональность.

Цели и задачи, поставленные при выполнении работы были выполнены с соблюдением всех предъявленных требований в установленные сроки.

**Слайд 20 – Спасибо за внимание**

Доклад окончен, спасибо за внимание.