

Федеральное государственное бюджетное образовательное учреждение высшего образования

Факультет

Καφεόρα

вычислительной техники

«Вычислительная техника»

Специальность:	09.05.01 «Применение и эксплуатация автоматизированных систем специального назначения»
Специализация:	«Эксплуатация вычислительных машин, комплексов, систем и сетей специального назначения»

НА ТЕМУ

АНАЛИЗ ПОВЕДЕНЧЕСКИХ РЕАКЦИЙ ПОЛЬЗОВАТЕЛЕЙ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ

Студент 20.11.2025 Беляков Евгений Николаевич
(подпись, дата) (ФИО полностью)

Руководитель _____ Слепцов Н.В.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр
(подпись, дата) (фамилия, инициалы) 23.06.2025 Бычков А.С.

Рецензент главный естественн. науки Васильев А.В.
(должность, место работы) (подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 28.05.2025 г. № 16)

Заведующий кафедрой _____ Митрохин М.А.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой удовлетворительно
(протокол заседания ГЭК от 13 июня 2015 № 13)

Секретарь ГЭК

Гурин Е.И.
(фамилия, инициалы)

Пенза 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет

Вычислительной техники

Кафедра

Вычислительная техника

«Утверждаю»
Заведующий кафедрой



Митрохин М.А.

«6» 05 2025 г.

ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ

1. Студент Беляков Евгений Николаевич гр. 20ВОЭ1 факультета вычислительной техники.

2. Тема работы: «Анализ поведенческих реакций пользователей интернет-приложений».

Тема утверждена приказом ПГУ № 601/о от 06.05.2025 г.

3. Руководитель работы к.т.н., доцент Слепцов Н.В.

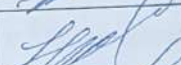
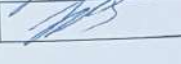
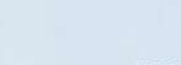
4. Задание на работу (назначение разработки, исходные данные и т.п.).

Разработать приложение для анализа поведенческих реакций пользователя. Приложение должно выполнять следующие функции: анализ отметок нравится, комментариев, репостов пользователей социальной сети ВКонтакте; анализ тематик сообществ пользователя; построение графиков для администратора системы и хранение архивных данных. Приложение для анализа поведенческих реакций пользователей должно разрабатываться с помощью среды разработки Visual Studio Code. База данных должна разрабатываться для СУБД «SQLite3».

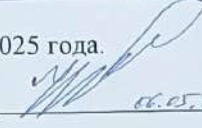
5. Перечень подлежащих разработке вопросов.

Требуется произвести анализ предметной области, определить функциональный состав программы, выбрать стек технологий для реализации программы, разработать алгоритм работы программы, разработать диаграммы: развертывания, вариантов использования, классов, с последующим написанием программы.

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1.	Анализ предметной области	10	08.05.2025 - 09.05.2025	
2.	Проектирование программного приложения	10	10.05.2025 - 13.05.2025	
3.	Разработка базы данных	20	14.05.2025 - 18.05.2025	
4.	Отладка и тестирование	10	19.05.2025 - 22.05.2025	
5.	Разработка экономической и бжд части	10	23.05.2025 - 27.05.2025	
6.	Оформление графической части	10	28.05.2025	
7.	Оформление выпускной работы	30	29.05.2025 - 10.06.2025	

Дата выдачи задания « 6 » 05 2025 года.

Руководитель дипломного проекта  06.05.2025 Слепцов Н.В.
(подпись, дата)

Задание к исполнению принял студент  06.05.2025 Беляков Е.Н.
(подпись, дата)

Работу к защите допустить
Декан факультета  25.06.2025 Фионова Л.Р.
(подпись, дата)

Реферат

Пояснительная записка 80 листов, 12 рисунков, 8 таблиц и 6 приложений

ПРИЛОЖЕНИЕ, АНАЛИЗ, VK, PYTHON, API, SQLITE

Объектом исследования является процесс анализа поведенческих реакций пользователей.

Цель работы – разработка приложения для анализа поведенческих реакций пользователей интернет-приложений.

В процессе работы необходимо выполнить анализ предметной области, сформулировать техническое задание для разрабатываемого приложения, разработать логическую и физическую модели данных для разработки базы данных.

В результате проделанной работы разработано приложение для поведенческих реакций пользователей интернет-приложений, состоящее из базы данных и приложения для работы с ней.

База данных спроектирована с использованием CASE-средства StarUML. База данных разработана для СУБД SQLite3, приложение реализовано в среде разработки Visual Studio Code.

					ПГУ1.090501.02.001 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	Анализ поведенческих реакций пользователей интернет- приложения Пояснительная записка	Лит.	Лист	Листов	
Разраб.		Беляков Е.Н.		23.02.25				2	80
Провер.		Слепцов Е.Н.		23.02.25					
Н. Контр.		Бычков А.С.		23.02.25					
Утверд.		Митрохин М.А.		23.02.25					
						ФВТ, каф. ВТ гр.20ВОЭ1			

Содержание

Перечень сокращений и обозначений.....	5
Введение.....	6
1 Анализ предметной области.....	9
1.1 Описание предметной области.....	9
1.2 Анализ аналогов.....	11
...1.3 Цели и задачи разрабатываемого программного приложения.....	14
1.4 Анализ целевой аудитории.....	15
2 Техническое задание.....	16
2.1 Назначение разработки.....	16
2.2 Требование к программе.....	16
2.2.1 Требования к функциональным характеристикам.....	16
2.2.2 Требование к надёжности.....	17
2.2.3	

					ПГУ1.090501.02.001 ПЗ	Лист
						3
Изм.	Лист	№ докум.	Подпись	Дата		

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

БД – база данных

ГОСТ – государственный стандарт

ЕСПД – единая система программной документации

ЕСКД – единая система конструкторской документации

МРОТ – минимальный размер оплаты труда

НФ – нормальная форма

ОС – операционная система

ПО – программное обеспечение

ПЭВМ – персональная электронно-вычислительная машина

ЦНС – центральная нервная система

ЭВМ – электронно-вычислительная машина

					ПГУ1.090501.02.001 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

FK (Foreign Key) – внешний ключ

ID (Identifier) – идентификатор объекта

PK (Primary Key) – первичный ключ

LLM (Large Language Model) – большая языковая модель

SQL (Structured Query Language) – язык структурированных запросов

UML (Unified Modeling Language) – унифицированный язык моделирования

Введение

SMM специалисты занимаются широким кругом задач, направленных на продвижение бренда в интернете: администрированием сообществ, производством контента, аналитикой (сбор и анализ метрик, составление отчетов) и запуском рекламы. По мнению некоторых исследователей, применение нейронных сетей позволяет заметно упростить работу SMM-специалиста, в частности копирайтинг [1].

Нейронные сети являются частью понятия искусственный интеллект и предназначены для имитации работы отдельной функции человеческого мозга, например распознавания образов. В последние годы большую популярность получили LLM (Large Language Model, большие языковые модели) – нейронные сети с большим количеством параметров. Для их развертывания используются специализированные дата-центры, построенные с помощью

					ПГУ1.090501.02.001 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

промышленных видеокарт, оптимизированных под инференс (запуск) нейронных сетей [2]. Примером больших языковых моделей является GPT4o от OpenAI, Mistral от одноименной компании, Grok, DeepSeek R1 и другие.

Приложение для анализа поведенческих реакций пользователей интернет-приложений включает в себя базу данных и клиент-серверное приложение обработки данных. При проектировании базы данных использовалось CASE-средство StarUML для СУБД SQLite3, для реализации приложения была применена среда разработки Visual Studio Code.

Также SMM специалисты очень хорошо могут настроить таргетированную рекламу, делая продукт более узнаваемым. Бизнес становится более прибыльным, реклама становится продвижением продукта. Это улучшает и само качество продукта, повышает конкуренцию, и прогресс в разных технологиях.

Нейронные сети, это инструмент будущего, он легко интегрируется в разные сферы деятельности, улучшает работоспособность производства, автоматизирует бизнес-процессы, оптимизирует изготовление, выпуск разной продукции. Многие думаю, что нейронные сети вытеснят множество людей, но это не так. Нейронные сети дополняют многие профессии, облегчая рутинные циклы производства, давая больше времени больше творчеству и креативности. У людей появляются большее идей, нейронные сети тут выступают как говорящий помощник [3].

Нейронные сети трансформируют мир, но их внедрение требует баланса между инновациями и ответственностью. Будущее — за «симбиозом» человека и ИИ: технологии усилят интеллектуальные возможности, сохраняя контроль над критическими решениями. По прогнозам PwC, к 2030 г. ИИ добавит \$15.7 трлн в мировую экономику, а нейросети станут «цифровой нервной системой» цивилизации [4].

Нейронные сети — вычислительные системы, имитирующие структуру и функции биологического мозга. Они состоят из взаимосвязанных «нейронов», обрабатывающих информацию слоями, что позволяет решать

					ПГУ1.090501.02.001 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

задачи, недоступные традиционным алгоритмам. Их революционное применение охватывает медицину, финансы, автономные системы и творческие индустрии [5].

Нейронные сети (НС) обладают рядом ключевых преимуществ, которые сделали их фундаментом современного искусственного интеллекта и машинного обучения.

Автоматическое извлечение признаков (Feature Learning): нейросети могут самостоятельно находить и извлекать сложные, иерархические признаки из сырых данных (изображений, текста, звука, чисел). Это избавляет человека от необходимости вручную проектировать и подбирать признаки (feature engineering), что часто является самым трудоемким и сложным этапом в традиционном машинном обучении [6].

Глубокие сети (с множеством слоев) учатся абстрактным представлениям: от простых краев на изображении до сложных объектов или смысла текста.

Высокая точность в сложных задачах: нейросети, особенно глубокие, демонстрируют наилучшие результаты в задачах, где данные имеют сложную структуру и взаимосвязи. В качестве примеров можно привести.

Компьютерное зрение:

- распознавание объектов;
- классификация изображений;
- обнаружение лиц;
- сегментация изображений;
- генерация изображений.

Обработка естественного языка (NLP):

- машинный перевод;
- анализ тональности текста;
- распознавание речи;
- генерация текста;
- чат-боты.

					ПГУ1.090501.02.001 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Анализ временных рядов (прогнозирование (финансы, погода), обнаружение аномалий)) [7].

Рекомендательные системы (персонализированные рекомендации товаров, фильмов, музыки).

Универсальность (гибкость архитектур):

Существует множество архитектур НС, специально разработанных для разных типов данных и задач:

- сверточные НС (CNN): Идеальны для изображений и любых данных с пространственной структурой (например, спектрограммы звука);
- рекуррентные НС (RNN) и Трансформеры: Оптимальны для последовательностей (текст, речь, временные ряды);
- генеративно-сопоставительные сети (GAN): Генерация новых данных (изображения, текст, музыка);
- автокодировщики (Autoencoders): Сжатие данных, поиск аномалий, понижение размерности.

Возможность комбинировать разные типы слоев для решения комплексных задач (мультимодальное обучение).

Устойчивость к шуму и неполным данным: нейросети, особенно при достаточном количестве данных для обучения, способны научиться игнорировать незначительный шум в данных и делать предсказания даже при наличии некоторых пропущенных значений или небольших искажений (например, частично загороженный объект на фото) [8].

Масштабируемость и параллелизация: вычислительные графы НС хорошо поддаются массово параллельной обработке на специализированном оборудовании (GPU, TPU). Это позволяет эффективно обучать очень большие модели на огромных объемах данных, что часто напрямую ведет к увеличению точности [9].

Адаптивность и непрерывное обучение (в определенных сценариях).

Хотя базовое обучение обычно требует много ресурсов, существуют методы, позволяющие дообучать уже готовые модели на новых данных без

					ПГУ1.090501.02.001 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

полного переобучения с нуля. Это делает их применимыми в динамически меняющихся средах.

Техники трансферного обучения (transfer learning) позволяют использовать знания, полученные на одной задаче (например, распознавание объектов на большом наборе ImageNet), для решения другой, но схожей задачи (например, классификация медицинских снимков), значительно экономя время и ресурсы. Способность моделировать сложные нелинейные зависимости, благодаря использованию нелинейных функций активации и множеству слоев, НС могут аппроксимировать чрезвычайно сложные, нелинейные функции, описывающие взаимосвязи в данных, с которыми не справляются простые линейные модели [10].

Следует отметить, что нейросети:

- не панацея: нейросети не всегда лучше традиционных алгоритмов (например, для простых задач с небольшими данными или где важна интерпретируемость).
- требуют данных: для эффективного обучения глубоких НС нужны очень большие размеченные наборы данных;
- вычислительная сложность: Обучение сложных моделей требует значительных вычислительных ресурсов (GPU/TPU) и времени.

"Черный ящик": Интерпретировать, почему сложная НС приняла то или иное решение, часто очень трудно или невозможно (проблема интерпретируемости) [11].

Риск переобучения: Сложные НС склонны к переобучению (overfitting) на тренировочных данных, если не применять методы регуляризации и валидации [12].

Требуют экспертизы:

- подбор архитектуры;
- гиперпараметров и обучение модели требуют глубоких знаний и опыта.

					ПГУ1.090501.02.001 ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

Подводя итог, можно отметить, что главные преимущества нейронных сетей — это их способность автоматически находить сложные закономерности в неструктурированных данных (изображения, текст, звук) и достигать высочайшей точности в задачах, которые ранее были недоступны компьютерам. Их универсальность и масштабируемость сделали их мощнейшим инструментом современного ИИ.

					ПГУ1.090501.02.001 ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

1 Анализ предметной области

1.1 Описание предметной области

Актуальностью создания приложения для анализа поведенческих реакций пользователей интернет-приложений является широкая популярность различных интернет-приложений среди россиян всех возрастов. Данная работа концентрируется на интернет-приложении Вконтакте, как на одном из самых популярных не только в России, но и в мире.

В 1991 году Тим Бьернс Ли изобрел Интернет для проекта ЦЕРН. Интернет тогда сильно отличался от сегодняшнего. Скорость передачи данных и мощность компьютеров того времени не позволяла передавать и обрабатывать большие объемы информации, поэтому интернет представлял собой статические страницы из текста с минимальным количеством изображений.

С развитием компьютерного оборудования и интернета начали появляться интернет-приложения – довольно широкое понятие, появившееся в эпоху Web 2.0. В отличие от интернета начала 90х годов прошлого века, Интернет сейчас представляет собой гораздо более децентрализованную систему, которая способна передавать большие объемы информации. Устройства так же сильно изменились. Практически все популярные сайты в Интернете сегодня адаптированы под мобильные версии, которые по своей популярности обогнали десктопные. В последние 5 лет наметился еще один тренд: превращение некоторых интернет-приложений в супераппы.

Супер-апп – это приложение, которое совмещает в себе множество функций из нескольких сервисов. В последние годы крупные корпорации, включая российские Сбербанк, Вконтакте, Яндекс, китайский WeChat инвестируют большие средства в развитие супераппов. Супераппы позволяют компаниям вовлекать пользователей в свою так называемую экосистему. В итоге пользователь больше времени проводит в сервисах одной компании, увеличивая ее прибыль. В последние годы в некоторых супераппах

					ПГУ1.090501.02.001 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

появляются функции, связанные с государственными услугами: хранением электронных версий документов, получением государственных услуг в режиме онлайн (в России это ГосУслуги и сервис МАХ, в Китае – уже упомянутый ранее WeChat).

Рассматриваемый в этом дипломе Вконтакте, созданный Павлом и Николаем Дуровым, изначально представлял собой сайт для знакомств студентов и выпускников вузов. Позже он превратился в социальную сеть, а в последние годы стал супераппом, что позволило ему расширить свою аудиторию и ее вовлеченность, а с уходом иностранных сервисов в начале 2022 года популярность Вконтакте среди рекламодателей сильно возросла.

Целью данной работы является разработка приложения со встроенной базой данных. Приложение будет обеспечивать получение информации о пользователе: его отметках «нравится», репостах и комментариях для дальнейшего анализа специалистом. Полученную информацию будет хранить база данных.

1.2 Анализ аналогов

Для того чтобы определиться с функционалом будущего приложения необходимо провести анализ аналогов. Рассмотрим каждый из них.

Аналитика Вконтакте — Вконтакте предоставляет инструменты для аналитики, которые отражают множество показателей, но не имеют интеграции с нейросетевыми технологиями, не позволяют имитировать получение чужой ленты пользователя (и анализ тематики лайков).

После проведения анализа аналогов разрабатываемого приложения для анализа поведенческих реакций пользователей интернет-приложений можно сделать вывод, что данное приложение будет иметь следующие преимущества перед конкурентами: интеграция с нейросетями и адаптация на использование в российских реалиях: (русскаяязычный интерфейс).

					ПГУ1.090501.02.001 ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

1.3 Цели и задачи разрабатываемого программного приложения

Цели создания приложения:

- увеличение продаж;
- упрощение работы SMM-специалиста;
- улучшение качества обслуживания клиентов.

В результате внедрения приложения планируется достичь ряда задач, а именно:

- повысить эффективность работы SMM-специалистов;
- снизить лишнюю нагрузку на маркетологов.

1.4 Анализ целевой аудитории

Целевая аудитория приложения включает в себя маркетологов и SMM-специалистов. Одной из первоочередных задач SMM-специалиста является определение интересов и потребностей потенциальных клиентов. Внедрение приложения позволит упростить агрегацию данных и дальнейший анализ. Так же приложение может быть полезно социологам, блогерам (для понимания своей аудитории) и маркетинговым агентствам – для сегментации аудитории и персонализации рекламных кампаний.

					ПГУ1.090501.02.001 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

2 Техническое задание

2.1 Назначение разработки

Приложение для анализа поведенческих реакций пользователей интернет-приложений должно быть предназначено для автоматизации сбора информации о предпочтениях пользователя.

2.2 Требования к программе

2.2.1 Требования к функциональным характеристикам

Программное обеспечение должно выполнять следующие функции:

- анализ отметок нравится, репостов, комментариев;
- анализ тематик сообществ;
- построение графиков для администратора.

Выходными данными приложения должны быть:

- графики предпочтений пользователей;
- отчет в формате png.

2.2.2 Требования к надежности

Приложение для анализа поведенческих реакций пользователей интернет-приложений должно работать без перебоев, контролировать вводимые данные и в случае некорректного ввода данных должна сообщать об этом пользователю.

Входная информация должна проверяться на соответствие типу данных и не противоречить законам логики.

2.2.3 Требования к составу и параметрам технических средств

Приложение должно быть предназначено для использования на локальном компьютере. Он должен иметь следующие характеристики:

- ОС Windows 10 и выше или Linux 4.19 и выше;
- процессор Intel Core 5 поколения и выше;
- свободное место на жестком диске 2 ГБ и выше;
- оперативную память 8 ГБ и выше;
- клавиатура;

					ПГУ1.090501.02.001 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

— мышь.

2.3 Требования к программной документации

Приложение для анализа поведенческих реакций пользователей интернет-приложений должно сопровождаться документацией, содержащей:

- техническое задание;
- описание логического проектирования БД;
- описание физического проектирования БД;
- описание программы;
- программу и методику испытаний;
- описание применения.

Структура и содержание программной документации должны соответствовать требованиям стандартов Единой системы программной документации (ЕСПД).

2.4 Стадии и этапы разработки

Этапы разработки приложения для анализа поведенческих реакций пользователей интернет-приложений представлены в таблице 1.

Таблица 1 — Стадии и этапы разработки

Стадии	Этапы	Объем работы в процентах	Срок выполнения	Исполнитель
Анализ предметной области	Анализ предметной области	10		Беляков Е.Н.
Проектирование программного приложения	- проектирование БД - проектирование приложения - проектирование модуля для составления врачебных заключений	40		Беляков Е.Н.
Разработка	Разработка базы	10		Беляков Е.Н.

					ПГУ1.090501.02.001 ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

базы данных	данных			
Стадии	Этапы	Объем работы в процентах	Срок выполнения	Исполнитель
Отладка и тестирование	Отладка и тестирование	10		Беляков Е.Н.
Разработка экономической части	Оценка затрат на разработку и внедрение программного приложения контроля выполнения задач	5		Беляков Е.Н.
Разработка раздела БЖД	Оценка тяжести труда эксплуататора программного приложения	5		Беляков Е.Н.
Оформление графической части	Оформление графической части	10		Беляков Е.Н.
Оформление Выпускной работы	Оформление пояснительной записки	10		Беляков Е.Н.

2.5 Порядок контроля и приемки

Для приемки приложения необходимо разработать тестовые наборы данных и убедиться, что документация на проект соответствует требованиям ЕСПД. Тестовые наборы данных должны позволить проверить каждую функцию приложения. При составлении тестовые наборы следует опираться на соответствующие нормативные документы такие как ГОСТ Р 56920-2024 «Системная и программная инженерия. Тестирование программного обеспечения. Общие положения».

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16

3 Обоснование целесообразности использования выбранных средств разработки

На основании анализа требований к функциональным характеристикам приложения для анализа поведенческих реакций пользователей интернет-приложений были выбраны средства разработки.

3.1 Язык программирования Python

Язык программирования – это набор правил, предназначенный для записи компьютерных программ. Далее полученный код компилируется либо интерпретируется, т. е переводится с языка, понятного человеку на язык, понятный компьютеру (в частности процессору).

Python – интерпретируемый язык программирования общего назначения. Является одним из самых популярных инструментов для анализа данных, дата-майнинга и веб-скрапинга, при этом являюсь простым в освоении и популярным (ссылка на tiobe). Старые версии Python критиковались за низкую производительность, но в последнее время его авторы сосредоточились на решении проблем с производительностью. Так в Python появились режим работы без GIL (Global Interpreter Lock, глобальная блокировка интерпретатора), JIT компилятор и оптимизированная стандартная библиотека.

Преимущества Python:

- легкий в чтении и написании;
- большое количество обучающего материала;
- высокая скорость разработки
- большое количество качественных сторонних модулей.

3.2 CASE-средство StarUML

Для разработки моделей баз данных (логических, физических) удобно применять case-средства. CASE-средства позволяют частично

					ПГУ1.090501.02.001 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

автоматизировать процесс разработки баз данных, выполняя в том числе такие задачи как генерацию SQL кода из физического описания БД.

К преимуществам StarUML можно отнести:

- большое сообщество пользователей;
- систему дополнений;
- возможность создания всех видов UML диаграмм.

К недостаткам можно отнести следующие:

- часть функций доступна только в платной версии;
- закрытый исходный код.

3.3 СУБД SQLite3

Для функционирования приложения необходимо выбрать СУБД.

СУБД делятся на категории: SQL и NOSQL, реляционные и нереляционные. Так как объем БД предполагается сравнительно небольшим, а данные имеют четко определенную структуру и легко представимы в виде таблицы, было принято решение использовать реляционную и компактную СУБД Sqlite. К преимуществам Sqlite также можно отнести:

- встраиваемость;
- открытый исходный код;
- лицензия (общественное достояние);
- полное покрытие тестами.

3.4 Библиотека MistralAI

Для анализа комментариев пользователей интернет-приложения «ВКонтакте» было решено использовать большую языковую модель (БЯМ).

Результаты тестирования самых популярных и производительных представлены на рисунке 1.

					ПГУ1.090501.02.001 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

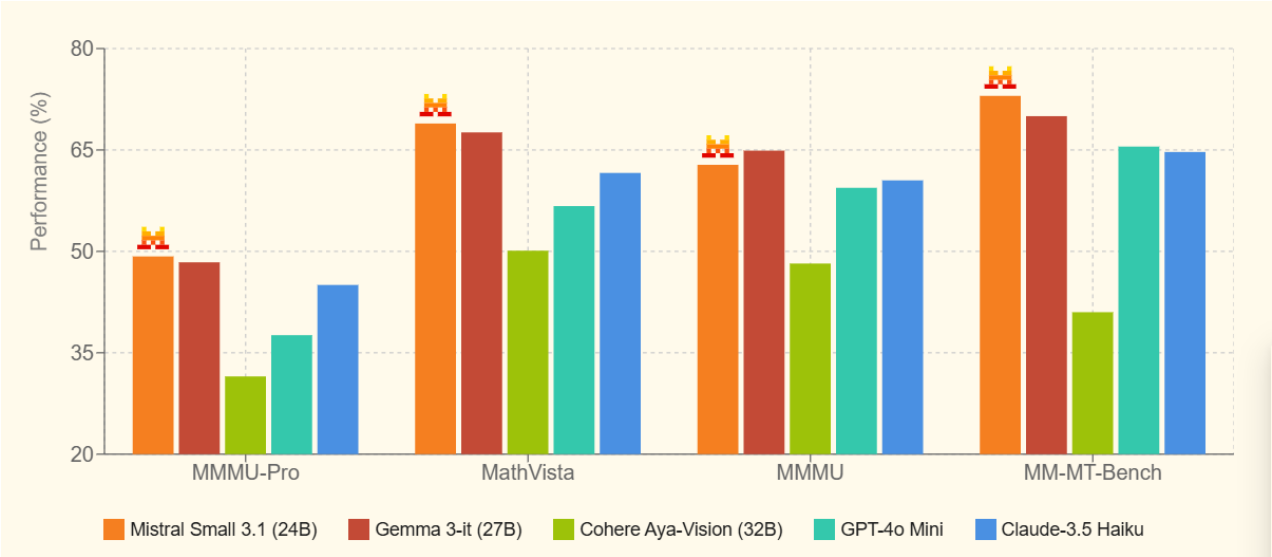


Рисунок 1 – Бенчмарк БЯМ

Целью бенчмарка MMMU является проверка способности модели понимать сложные мультимодальные данные. Тест охватывает 6 основных категорий: искусство, бизнес, наука, здоровье, инженерия, гуманитарные науки и включает около 11 тысячи вопросов.

Бенчмарк MathVista концентрируется на решении математических задач. Содержит 6500 задач из алгебры, геометрии, статистики и других разделов математики. Задачи требуют логического вывода на основе текста и изображений.

Тест ММ-МТ Bench является комплексным и проверяет возможности модели в задачах разных типов.

По результатам тестирования Mistral обходит своих конкурентов. Модель так же единственная из списка имеет бесплатное API, а открытый исходный код и веса делают данное решение более устойчивым к возможным новым антироссийским санкциям.

3.5 Среда разработки Visual Studio Code

Выбор среды разработки должен основываться на стеке разработки, финансовых возможностях пользователя (разработчика) и юридических особенностях (некоторые иностранные среды разработки невозможно купить из-за санкций).

Visual Studio Code – современная среда разработки, разработанная корпорацией Майкрософт. Изначально рассчитанная на веб-разработчиков, сейчас данная IDE занимает ведущие позиции среди своих конкурентов не только по веб-разработке, но и вообще среди всех сред разработки. Visual Studio Code так же имеет открытый исходный код, отличную оптимизацию и отлично поддерживает стек разработки (Python, Sqlite, Qt5). К преимуществам Visual Studio Code так же можно отнести:

- систему расширений;
- большое количество поддерживаемых языков программирования;
- инструменты работы с системой контроля версий.

					ПГУ1.090501.02.001 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

4 Разработка приложения

Диаграмма развертывания приложения для анализа поведенческих реакций пользователей интернет-приложений представлена на рисунке 2.

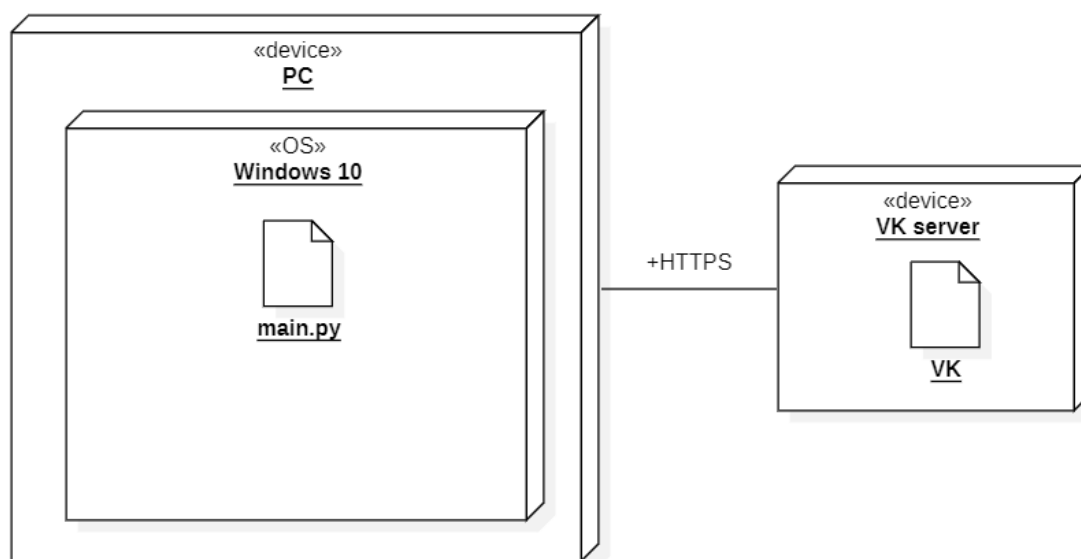


Рисунок 2 – Диаграмма развертывания

4.1 Логическое проектирование базы данных

Логическая модель базы данных приведена на рисунке 3 и содержит три

comments_analysis		
PK	lid	INTEGER
	post_url	TEXT
	analysis	TEXT
	timestamp	DATETIME

group_analysis		
PK	id	INTEGER
	user_id	INTEGER
	group_id	INTEGER
	group_activity	TEXT
	timestamp	DATETIME

reposts_analysis		
PK	id	INTEGER
	user_id	INTEGER
	group_id	INTEGER
	group_activity	TEXT
	timestamp	DATETIME

likes_analysis		
PK	id	INTEGER
	user_id	INTEGER
	group_id	INTEGER
	group_activity	TEXT
	timestamp	DATETIME

Рисунок 3 – Логическая модель данных

Логическая модель данных

Описание атрибутов сущностей логической модели приведено в таблицах 2 – 5.

Таблица 2 – Описание атрибутов сущности «АнализКомментариев»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
Id	ID комментария	РК	
post_url	URL поста		
analysis	Анализ Mistral		
timestamp	Время анализа		

Таблица 3 – Описание атрибутов сущности «АнализСообществ»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
Id	ID анализа	РК	
user_id	ID пользователя		
group_id	ID группы		
group_activity	Тематика группы		
timestamp	Время анализа		

Таблица 4 – Описание атрибутов сущности «АнализЛайков»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
Id	ID анализа	РК	
user_id	ID пользователя		
group_id	ID группы		
group_activity	Тематика группы		
timestamp	Время анализа		

Таблица 5 – Описание атрибутов сущности «АнализРепостов»

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
Id	ID анализа	РК	
user_id	ID пользователя		
group_id	ID группы		
group_activity	Тематика группы		
timestamp	Время анализа		

Процесс нормализации – это последовательный процесс приведения базы

данных к эталонному виду, то есть переход от одной нормальной формы к следующей.

Отношение находится в 1 НФ, если все строки таблицы различные и все атрибуты являются атомарными.

Отношение находится во 2 НФ, если оно находится в 1 НФ и каждый не ключевой атрибут таблицы функционально полно зависит от первичного ключа.

Отношение находится в 3 НФ, если оно находится во 2 НФ и между не ключевыми атрибутами не существует транзитивных зависимостей.

Все атрибуты четырех сущностей атомарны и находятся в 3 НФ. Для подтверждения данного фактора необходимо рассмотреть каждую сущность:

– сущность «АнализКомментариев» имеет ключевой атрибут «ID комментария». Атрибуты «post_url», «analysis», «timestamp» данной сущности являются простыми и неделимыми;

– сущность «АнализСообществ» имеет ключевой атрибут «id». Атрибуты «user_id», «group_id», «group_activity», «timestamp» данной сущности являются простыми и неделимыми;

– сущность «АнализРепостов» имеет ключевой атрибут «id». Атрибуты «user_id», «group_id», «group_activity», «timestamp» данной сущности являются простыми и неделимыми;

– сущность «АнализЛайков» имеет ключевой атрибут «id». Атрибуты «user_id», «group_id», «group_activity», «timestamp» данной сущности являются простыми и неделимыми.

Таким образом, была разработана логическая модель данных.

4.2 Физическое проектирование

На основании разработанной логической модели данных была построена физическая модель данных. Физическая модель данных разработана для СУБД Sqlite и представлена на рисунке 3.

comments_analysis		
PK	lid	INTEGER
	post_url	TEXT
	analysis	TEXT
	timestamp	DATETIME

group_analysis		
PK	id	INTEGER
	user_id	INTEGER
	group_id	INTEGER
	group_activity	TEXT
	timestamp	DATETIME

reposts_analysis		
PK	id	INTEGER
	user_id	INTEGER
	group_id	INTEGER
	group_activity	TEXT
	timestamp	DATETIME

likes_analysis		
PK	id	INTEGER
	user_id	INTEGER
	group_id	INTEGER
	group_activity	TEXT
	timestamp	DATETIME

Физическая модель данных

Физическая модель данных содержит четыре таблицы. Описание таблиц физической модели приведено в таблицах 6 — 9.

Таблица 6 – Описание полей таблицы «comments_analysis»

					ПГУ1.090501.02.001 ПЗ	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
Id	id	Не допустимо	PK	INTEGER
post_url	post_url	Не допустимо		TEXT
analysis	analysis	Не допустимо		TEXT
timestamp	timestamp	Не допустимо		DATETIME

Таблица 7 – Описание полей таблицы «group_analysis»

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
Id	id	Не допустимо	PK	INTEGER
user_id	user_id	Не допустимо		INTEGER
group_id	group_id	Не допустимо		INTEGER
group_activity	group_activity	Не допустимо		TEXT
timestamp	timestamp	Не допустимо		DATETIME

Таблица 8 – Описание полей таблицы «reposts_analysis»

					ПГУ1.090501.02.001 ПЗ	Лист
						25
Изм.	Лист	№ докум.	Подпись	Дата		

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
Id	id	Не допустимо	PK	INTEGER
user_id	user_id	Не допустимо		INTEGER
group_id	group_id	Не допустимо		INTEGER
group_activity	group_activity	Не допустимо		TEXT
timestamp	timestamp	Не допустимо		DATETIME

Таблица 9 – Описание полей таблицы «likes_analysis»

Имя атрибута	Физическое имя	Нулевое значение	Ключевое поле	Тип данных
Id	id	Не допустимо	PK	INTEGER
user_id	user_id	Не допустимо		INTEGER
group_id	group_id	Не допустимо		INTEGER
group_activity	group_activity	Не допустимо		TEXT
timestamp	timestamp	Не допустимо		DATETIME

Таким образом, была разработана физическая модель данных для СУБД Sqlite3.

4.3 Описание архитектуры

					ПГУ1.090501.02.001 ПЗ	Лист
						26
Изм.	Лист	№ докум.	Подпись	Дата		

Программа реализована с помощью языка программирования Python в среде разработки Visual Studio Code и имеет модульную структуру, представленную на рисунке 4.

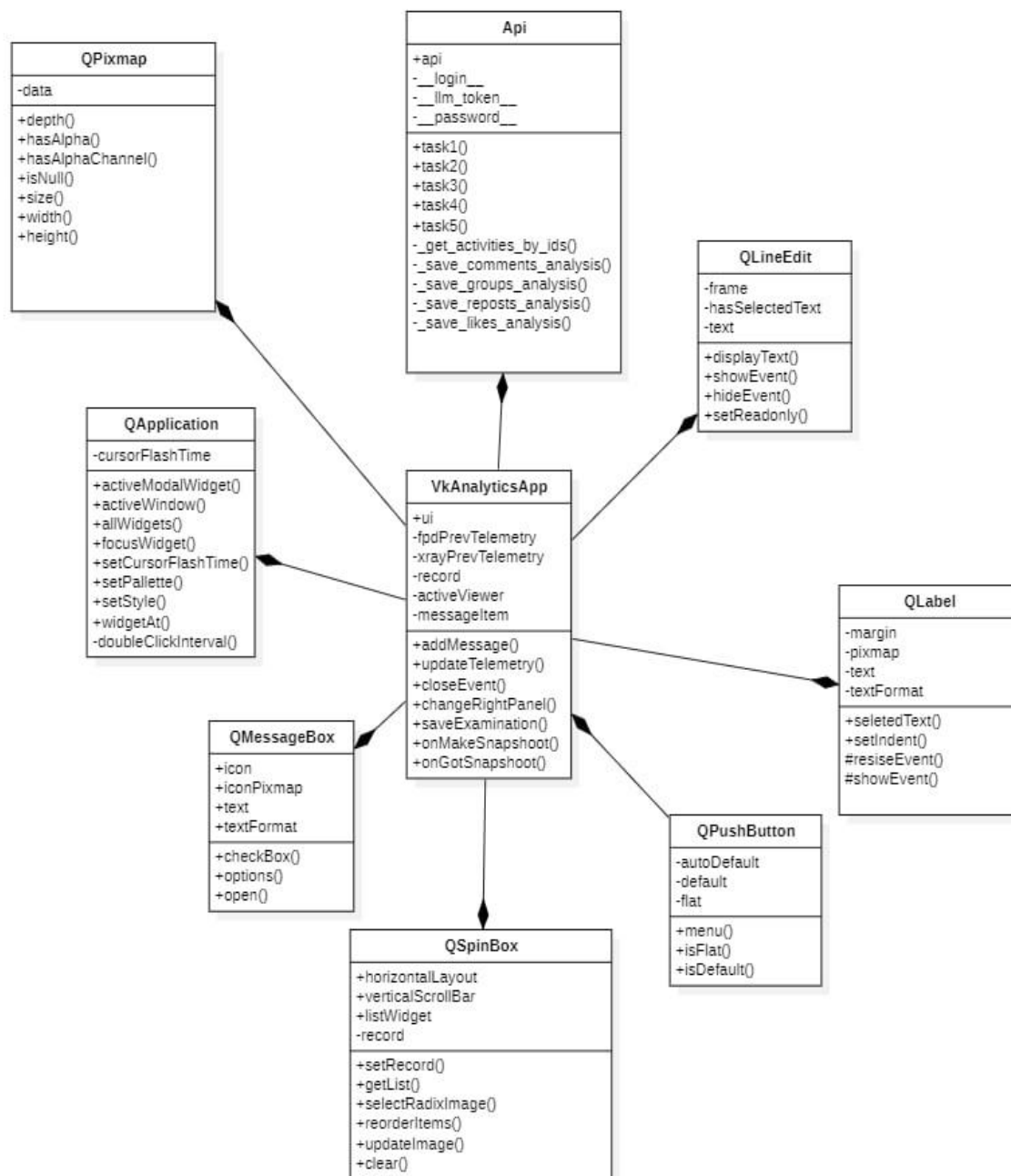


Рисунок 4 — Диаграмма классов

Описание классов представлено в таблице 8.

Таблица 8 – Описание классов программы

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

Название класса	Описание класса
VkAnalyticsApp	Модуль для отображения главного окна приложения.
Api	Модуль, реализующий функции для взаимодействия с апи Вконтакте и Mistral
QApplication	Модуль для создания графического приложения на Qt
Название класса	Описание класса
QPixmap	Изображение Qt
QMessageBox	Модуль для отображения модального окна
QLabel	Модуль для отображения текстовой заметки.
QLineEdit	Модуль для ввода текстовых данных пользователем.
QPushButton	Модуль для реализации кнопки.
QSpinBox	Модуль обеспечивающий вывод прогресс бара.

Диаграмма вариантов использования представлена на рисунке 5.

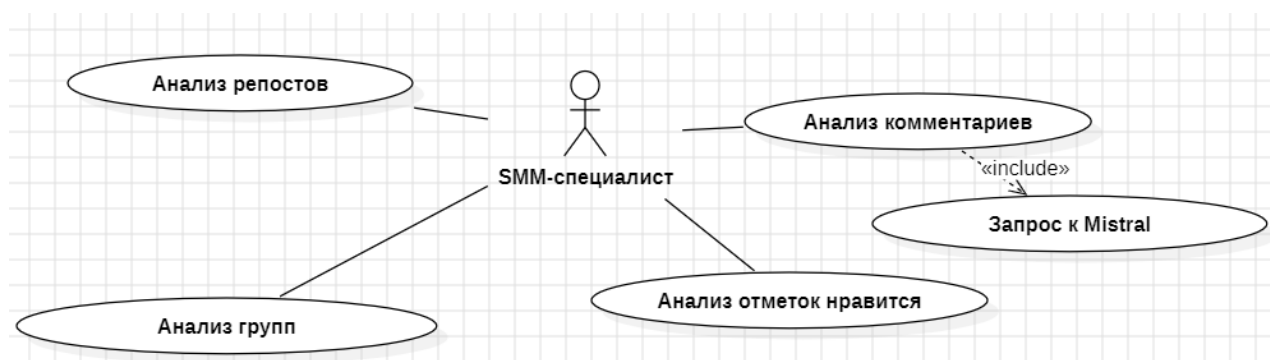


Рисунок 5 — Диаграмма вариантов использования

4.4 Описание программы

Код приложения хранится в папке src.

Файл src/api.py реализует взаимодействие с API VK и Mistral.

В начале импортируются модули – vk, time, sys, os, sqlite3, matplotlib.pyplot, datetime, mistralai. Модуль vk нужен для работы с api Вконтакте. Модуль time обеспечивает задержку при отправке запросов к api Вконтакте, модуль sys используется для организации выхода из программы, os

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		28

– для чтения переменных окружения, `sqlite3` – для работы с БД в формате `Sqlite3`. `Matplotlib` представляет собой широко распространенную библиотеку для рисования графиков, модуль `datetime` используется для получения текущей даты и наконец `mistralai` отвечает за взаимодействие с большой языковой моделью. В самом начале программы объявляется класс `API`.

Применение объектного ориентированного программирования позволяет скрыть (инкапсулировать) от пользователя класса детали реализации, такие как данные для входа в аккаунт Вконтакте и токен от `Mistral API`.

Метод `__init__` класса `Api` представляет собой конструктор. В нем происходит аутентификация на сайте с помощью создания объекта класса `Api` модуля `vk`. В качестве параметров выступают логин и пароль от аккаунта пользователя и версия `API` (5.199). Далее вызывается метод `_init_db`.

Метод `_init_db` создает базу данных. Для этого используется функция `connect` модуля `sqlite3`. Далее для открытого соединения с БД получается объект типа `cursor` и вызывается его метод `execute`, в который передается SQL строка — создание таблицы. Вызов метода производится 4 раза (для 4 таблиц). Таблицы создаются только если они не существуют. При этом используется SQL команда `CREATE TABLE IF NOT EXISTS`. В конце производится запись изменений в БД при помощи метода `commit` курсора `cursor`.

Метод `_save_reposts_analysis` сохраняет анализ репостов в базу данных. При этом используется функция `connect` модуля `sqlite3`, которая создает соединение `conn`. Далее в цикле проходимся по списку активностей и добавляем информацию в БД при помощи SQL команды `INSERT`. В конце транзакция выполняется при помощи метода `commit`.

Метод `_save_likes_analysis` сохраняет анализ отметок нравится в базу данных. При этом используется функция `connect` модуля `sqlite3`, которая создает соединение `conn`. Далее в цикле проходимся по списку лайков и добавляем информацию в БД при помощи SQL команды `INSERT`. В конце транзакция выполняется при помощи метода `commit`.

					ПГУ1.090501.02.001 ПЗ	Лист
						29
Изм.	Лист	№ докум.	Подпись	Дата		

Метод `_save_comments_analysis` сохраняет анализ комментариев в базу данных. При этом используется функция `connect` модуля `sqlite3`, которая создает соединение `conn`. Далее в цикле проходимся по списку комментариев и добавляем информацию в БД при помощи SQL команды `INSERT`. В конце транзакция выполняется при помощи метода `commit`.

Метод `_save_groups_analysis` сохраняет анализ тематик групп в базу данных. При этом используется функция `connect` модуля `sqlite3`, которая создает соединение `conn`. Далее в цикле проходимся по списку тематик и добавляем информацию в БД при помощи SQL команды `INSERT`. В конце транзакция выполняется при помощи метода `commit`.

Метод `_get_activities_by_ids` класса `Api` принимает в качестве аргумента строку `ids` – id сообществ через запятую. В начале функции объявляется пустой список `activities`, который в дальнейшем будет возвращаться. Далее информация о группах получается с помощью вызова метода `vk api groups.getById`. Для получения информации о тематике группы необходимо дополнительно запросить ее, передав строку `fields='activity'`. Далее необходимо в цикле по `group_info` отфильтровать список групп, исключив из него события. Далее получаем тематику группы и добавляем ее в массив `activities`, если группа не является Публичной страницей или Закрытым сообществом. Для таких групп тематики нет. В конце список `activities` возвращается.

Метод `task1` класса `Api` реализует анализ репостов. В качестве аргумента принимает 1 целое число — `user_id`. В начале метода объявляется `count` – количество постов, получаемых за раз. Далее в бесконечном цикле получаем стену переданного пользователя `user_id`. При этом используется метод `vk api wall.get`. Методу передаются параметры: `owner_id=user_id`, `count=100` и `offset` (смещение от начала). Смещение увеличивается на `count` каждую итерацию цикла. Выход из цикла происходит когда метод `wall.get` не вернул данные, т.е постов больше нет. В теле цикла в строку `group_ids` добавляем `owner_id`. После цикла из `groups_ids` убирается последняя, лишняя запятая.

					ПГУ1.090501.02.001 ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

Для этого используется срез :-1. Далее список тематик групп получается с помощью вызова метода класса `_get_activities_by_ids`, которому передается `groups_ids`. В конце вызывается функция `_save_reposts_analysis` и возвращается значение тематик групп.

Метод `task2` класса `Api` реализует анализ ленты пользователя. В качестве аргумента принимает 2 целых числа — `user_id` и `cnt` (насколько глубоко искать в истории). В начале получаем подписки пользователя с помощью вызова метода `vk api users.getSubscriptions`. В качестве аргументов передаются `user_id` и параметр `extended=0`. Далее получаем новости, используя метод `newsfeed.get`. Нас интересуют новости от сообщества, поэтому передаем `source_ids` и `filters='posts'`. Далее мы перебираем посты и проверяем какой из нас понравился. Для этого вызываем метод `vk likes.isLiked` и получаем поле `isLiked`. Если пост понравился, добавляем `id` группы в список `group_ids`. В конце вызывается функция `_save_likes_analysis` и возвращается значение тематик групп, полученное с помощью вызова приватного метода `_get_activities_by_ids`.

Метод `task3` класса `Api` реализует анализ комментариев записи. В качестве аргумента принимает 1 строку — `post_url`. В начале получаем комментарии к записи, используя `api` метод `wall.getComments`. Значения `owner_id` и `post_id` берутся из `post_url` – первая часть это `owner_id`, вторая `post_id`. Затем для каждого комментария получаем поле `text` и склеиваем все в одну строку текста. Далее создаем объект класса `Mistral`. В конструктор передаем токен от LLM. Затем получаем ответ от модели, используя метод `complete`. Модель и сообщение передаются как аргументы функции. Модель `mistral-small-2503` обеспечивает хорошее время обработки запроса (не содержит ненужной для такой задачи мыслительной функции) и при этом доступна бесплатно. Затем мы получаем ответ от модели и сохраняем его в БД с помощью вызова метода `_save_comments_analysis`. В конце функция возвращает текст, который мы получили от модели.

					ПГУ1.090501.02.001 ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

Метод `task4` класса `Api` реализует анализ тематик сообществ пользователя. В качестве аргумента принимает 1 целое число — `user_id`.

В начале получаем список групп пользователя. Для этого используется метод `groups.get`, которому передаем `user_id` и `fields='activity'` (для получения тематик групп). Далее `id` групп сохраняются в строковую переменную `group_str`. Затем получаем тематики групп, вызвав метод `_get_activities_by_ids`, передав `group_str`. В конце вызываем `_save_groups_analysis` для записи данных в БД.

Метод `task5` класса `Api` реализует вывод аналитического отчета на экран. В начале устанавливаем соединение с базой данных. Для этого используется функция `connect` модуля `sqlite3`, которая создает соединение `conn`. Далее получаем объект курсора с помощью вызова метода `cursor` объекта `conn`. Затем переходим к созданию графика. Для этого используем функцию `subplot` модуля `matplotlib.pyplot`, которой передаем аргументы `(2,2)` — количество графиков по оси `x` и `y`. Далее получаем данные из таблицы анализа репостов. При этом используется `sql` оператор `SELECT` с группировкой по тематике группы. Если данные есть, добавляем их на график. Затем получаем данные из таблицы анализа лайков. При этом используется `sql` оператор `SELECT` с группировкой по тематике группы. Если данные есть, добавляем их на график. Далее получаем данные из таблицы анализа тематик групп. При этом используется `sql` оператор `SELECT`. Если данные есть, добавляем их на график. Заключительный график — время анализа. Для получения времени анализа используется оператор `SELECT` с объединением данных из 3 таблиц (`reposts_analysis`, `likes_analysis`, `groups_analysis`). Далее добавляем данные на график и сохраняем его в `png` файл с помощью метода `savefig` объекта `plt`. В конце созданный график демонстрируется с помощью метода `show`.

Для проверки надежности используем модульное тестирование. Модульное тестирование для файла `api.py` реализовано в функции `main`. Эта функция вызывается только при запуске непосредственно файла `api.py` и не вызывается при импорте `api`. Для этого используется проверка значения

					ПГУ1.090501.02.001 ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подпись	Дата		

переменной `__name__`. В функции `main` создается объект `Api`, которому передается логин и пароль пользователя. Логин и пароль получаются из соответствующих переменных среды, что обеспечивает возможность безопасного распространения исходного кода по Интернету. Далее просим пользователя выбрать действие (анализ репостов, анализ ленты, анализ групп, анализ комментариев или построение графиков). Если выбрано несуществующее действие, пользователю будет отображена ошибка.

Далее для каждого действия просим ввести дополнительную информацию (`id` пользователя, `url` поста или количество проверок). Для построения графика ввод дополнительной информации не требуется.

Файл `src/main.py` реализует графический интерфейс пользователя. В начале импортируются модули – `sys`. Из модуля `PyQt5.QtWidgets` импортируются следующие сущности: `QApplication` (родительский класс для графического приложения Qt), `QMainWindow` (главное окно приложения), `QWidget` (главный класс графического виджета), `QVBoxLayout` (вертикальный компоновщик для создания адаптивного интерфейса), `QHBoxLayout` (горизонтальный компоновщик), `QLabel` (надпись), `QComboBox` (инструмент выбора из вариантов), `QLineEdit` (поле ввода текстовых данных), `QPushButton` (кнопка), `QTextEdit` (многострочное текстовое поле), `QGroupBox` (для создания группы виджетов), `QSpinBox` (для ввода числовых данных) и `QMessageBox` (диалоговое окно). Далее из модуля `PyQt5.Gui` импортируется `QPixmap` – класс для поддержки работы с изображениями и модуль `api`.

Программа начинается с объявления класса `VkAnalyticsApp`, который наследуется от `QMainWindow`.

Метод `__init__` представляет собой конструктор класса. В нем производим вызов конструктора родительского класса при помощи строки `super().__init__()`. Далее вызываем метод `init_ui`.

Метод `init_ui` предназначен для создания графического интерфейса. В нем задается названия окна и его размеры. При этом используются методы `setWindowTitle` и `setGeometry`. Далее создаются главный виджет и

					ПГУ1.090501.02.001 ПЗ	Лист
						33
Изм.	Лист	№ докум.	Подпись	Дата		

выполняется компоновка. Поля и текстовые метки для ввода логина и пароля хранятся в `QHBoxLayout`, что обеспечивает адаптивность интерфейса под разные размеры экранов. Выбор задачи реализован при помощи `QComboBox`. Варианты добавляются при помощи вызова метода `addItem`. Задача так же компоуется при помощи `QHBoxLayout`. Далее осуществляется вызов функции `connect`, который привязывает изменение задачи к обновлению графического интерфейса (вызову функции `update_input_fields`). Функция `connect` реализует один из краеугольных механизмов Qt – сигналы и слоты, что позволяет отделить логику обработки и реализацию интерфейса. Далее создаем стековый виджет, который нужен для хранения виджетов для разных задач: при выборе определенной задачи будет виден только один виджет.

Виджеты добавляются в стек при помощи вызова метода `addWidget`. Далее создаем кнопку «Выполнить» и группу виджетов «Результаты». Текстовое поле Результаты должно быть помечено как только для чтения. Далее вызывается метод `update_input_fields`.

Метод `update_input_fields` обеспечивает обновление поля ввода в зависимости от выбранной позиции (показывает нужный виджет из стека).

Метод `validate_apr` проверяет авторизацию. Если пользователь не ввел логин или пароль, то на экране отображается сообщение с ошибкой. Далее создается объект `Api` модуля `api`, которому передается логин и пароль.

Метод `execute_task` нужен для выполнения выбранной задачи. Для начала получаем индекс комбобокса задачи и далее в зависимости от задачи получаем данные из текстовых полей. В конце каждого условия вызывается нужный метод класса `Api` и показывает результат пользователю. Для этого используется метод `show_text_result`. В случае возникновения исключения показываем пользователю сообщение с ошибкой.

Метод `show_image_result` показывает графический отчет пользователю. Для этого создаем `QPixmap` и задаем его отображение в `image_result`.

4.5 Руководство пользователя

					ПГУ1.090501.02.001 ПЗ	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		

Для запуска программы необходимо при необходимости установить Python и необходимые библиотеки. Установку Python производить только на ОС Windows, Linux уже содержит Python. После завершения процесса открыть командную строку и прописать:

```
py -m pip install vk, matplotlib, mistralai
```

Далее запустить программу двойным щелчком мыши по файлу main.py.

До начала необходимо ввести логин и пароль от Вконтакте, используя соответствующие текстовые поля. Далее выбрать требуемую задачу — анализ репостов, анализ лайков в ленте, анализ комментариев к посту, анализ групп или построение аналитического отчета. После этого необходимо заполнить данные: ID пользователя для 1 и 4 задачи, для анализа лайков количество проверок, для анализа комментариев — url поста. Для построения аналитического отчета дополнительные данные не требуются. Далее необходимо нажать на кнопку «Выполнить». Результат появится в поле «Результаты». Для графического отчета дополнительно сохраняется файл в формате png.

4.6 Методика испытаний программы

Системное тестирование предполагает проверку всех реализованных функций на предмет соответствия их техническому заданию.

Для перехода к системному тестированию для начала необходимо провести модульное. Файл api.py содержит модульный тест.

Для начала необходимо ввести логин и пароль от Вконтакте (рисунок 6).

Если пропустить ввод, данные для входа будут взяты из переменных окружения VK_LOGIN и VK_PASSWORD.

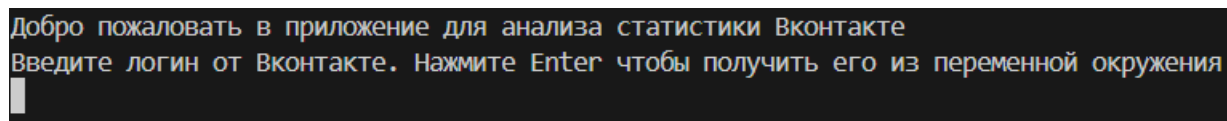


Рисунок 6 — Ввод логина и пароля

					ПГУ1.090501.02.001 ПЗ	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

Двухфакторная авторизация не поддерживается. В случае ошибки входа (рисунок 7) необходимо отключить ее в настройках VK ID и повторить попытку.

Ошибка входа. Пожалуйста, отключите двухфакторную авторизацию.

Рисунок 7 — Ошибка входа

После ввода правильных логина и пароля откроется меню (рисунок 8).

Выберите действие
1 - анализ репостов
2 - анализ отметок нравится в ленте
3 - анализ комментариев
4 - анализ тематик сообществ пользователя
5 - построение графиков
Ваш выбор: █

Рисунок 8 — Выбор действия

Необходимо последовательно протестировать каждый пункт меню. Выбрав пункт 1, далее вводим id пользователя (рисунок 9).

Выберите действие
1 - анализ репостов
2 - анализ отметок нравится в ленте
3 - анализ комментариев
4 - анализ тематик сообществ пользователя
5 - построение графиков
Ваш выбор: 1
Введите ID пользователя: 230190268
Результат анализа репостов: ['Дом культуры', 'Музыкальная группа', 'Государственная организация', 'Юмор', 'Фан-сообщество', 'Музыкальная группа']

Рисунок 9 — Анализ репостов

Далее необходимо визуально сравнить данные с сайтом Вконтакте. Открываем страницу пользователя с указанным id и видим искомые музыкальные группы и государственные сообщества (рисунок 10).

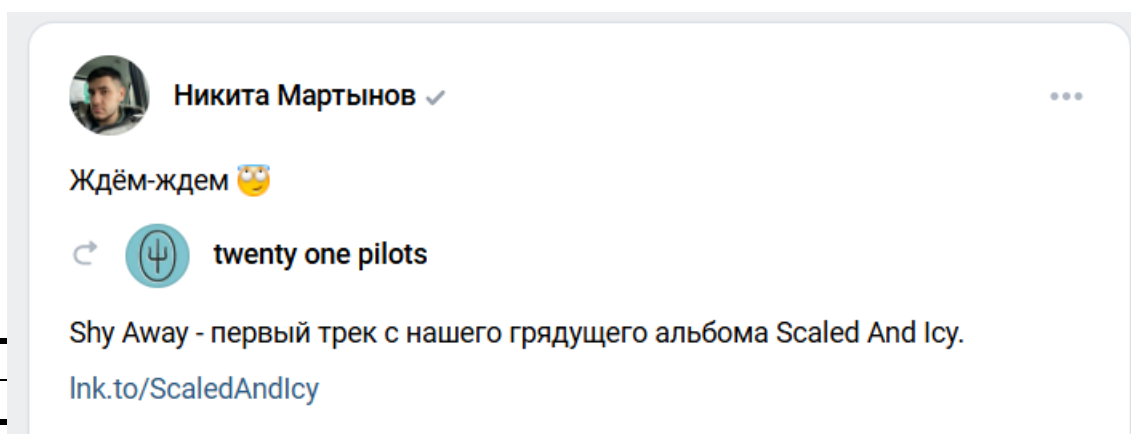


Рис
уно
к 10

— Репосты пользователя

Далее необходимо протестировать анализ отметок нравится в ленте. Для этого необходимо открыть ленту Вконтакте и оценить произвольную запись, запомнив тематику группы (рисунок 11).

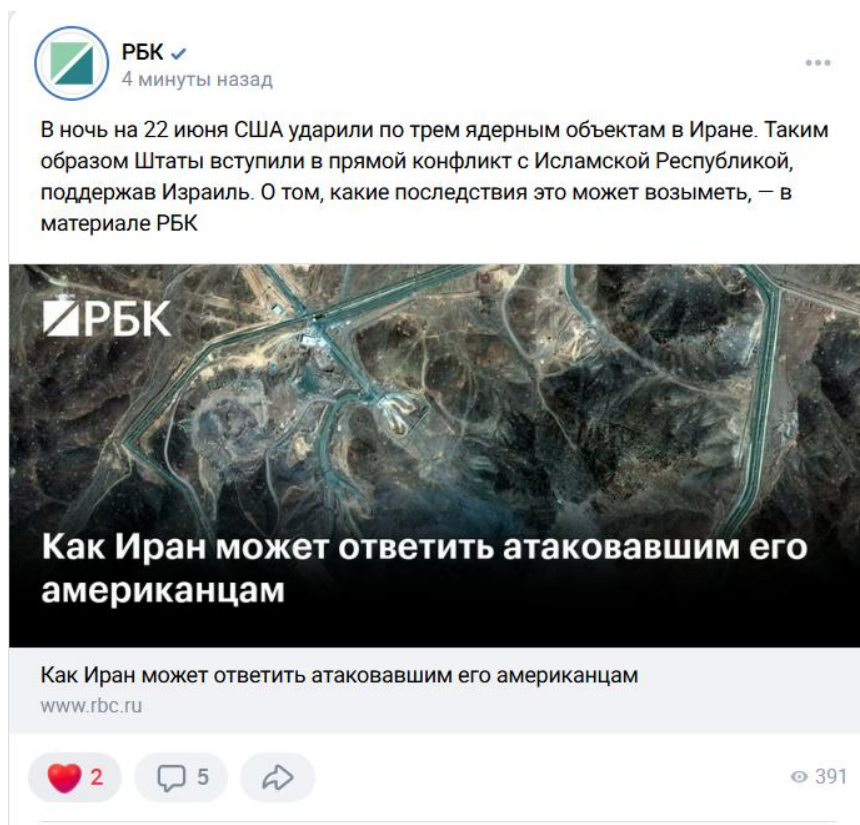


Рисунок 11 — Лента пользователя

После этого сравнить тематику сообщества, запись которого понравилась с тем что выводит программа (рисунок 12).

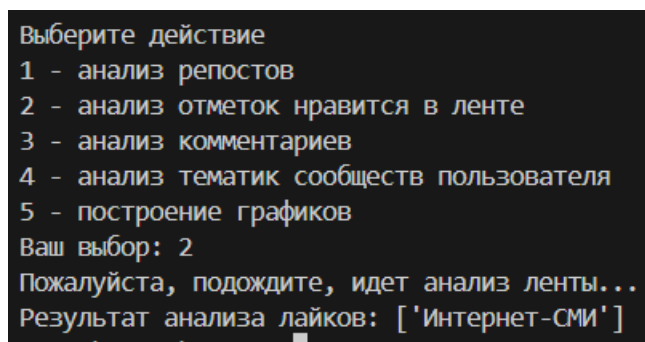


Рисунок 12 — Тематика лайков ленты пользователя

					ПГУ1.090501.02.001 ПЗ	Лист
						37
Изм.	Лист	№ докум.	Подпись	Дата		

Для анализа комментариев необходимо выбрать пункт 3 и ввести url поста (рисунок 13).

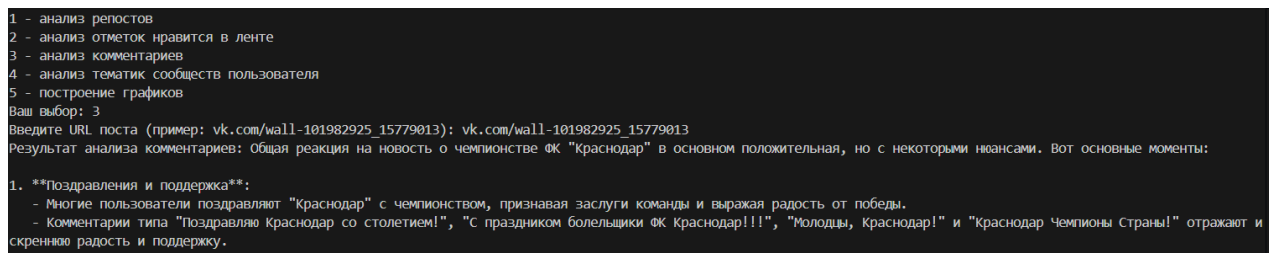


Рисунок 13 — Анализ комментариев

Убедимся, что анализ верный. Для этого откроем запись и посмотрим на самые популярные комментарии (рисунок 14). Анализ выполнен верно.



Рисунок 14 — Популярный комментарий

Наконец проверим функцию построению графиков. Для этого в меню необходимо выбрать пункт 5. Откроется график (рисунок 15).

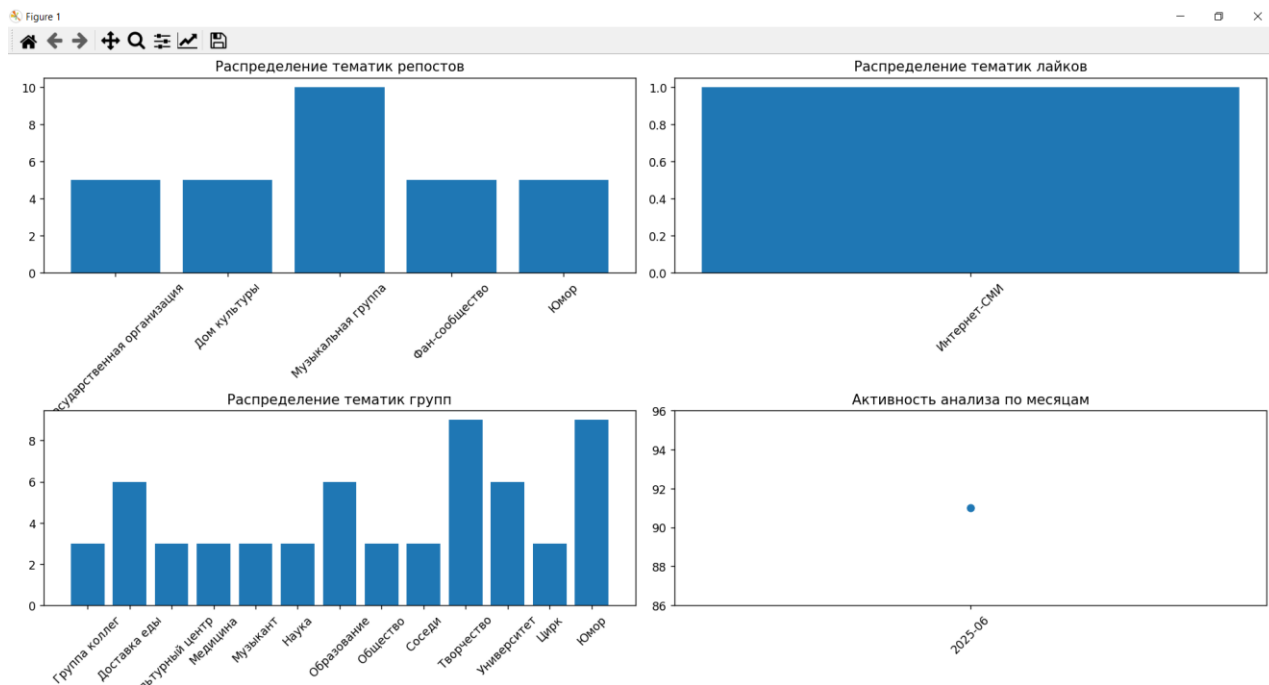


Рисунок 15 — Графики

Графики построены правильно, модуль апи работает корректно.

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		38

Далее после успешного тестирования модуля получения данных необходимо протестировать основную программу.

После запуска приложения будет открыта главная форма (рисунок 16).

Аналитика ВКонтакте

Настройки ВКонтакте

Логин: belyakovpnz@gmail.com

Пароль: ●●●●●●●●

Выбор задачи

Выберите задачу: Анализ репостов пользователя

ID пользователя: 230190268

Выполнить

Результаты

Анализ репостов:
Дом культуры
Музыкальная группа
Государственная организация
Юмор
Фан-сообщество
Музыкальная группа

Рисунок 16 — Главная форма

Далее необходимо ввести логин, пароль и нажать кнопку «Выполнить». Будет выполнен анализ репостов (рисунок 17).

Аналитика Вконтакте

Настройки ВКонтакте

Логин: belyakovpnz@gmail.com

Пароль:

Выбор задачи

Выберите задачу: Анализ репостов пользователя

ID пользователя: 230190268

Выполнить

Результаты

Анализ репостов:
Дом культуры
Музыкальная группа
Государственная организация
Юмор
Фан-сообщество
Музыкальная группа

Рисунок 17 — Анализ репостов. Далее необходимо выбрать

задачу №2 (анализ лайков в ленте) и нажать на кнопку «Выполнить». Результаты соответствуют результатам тестирования модуля, функция работает верно.

Аналитика Вконтакте

Настройки ВКонтакте

Логин: belyakovpnz@gmail.com

Пароль:

Выбор задачи

Выберите задачу: Анализ лайков в ленте

Количество проверок: 1

Выполнить

Результаты

Анализ лайков:
Интернет-СМИ

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		40

Рисунок 18 — Анализ репостов

Затем необходимо выбрать задачу №3 (анализ комментариев), ввести url поста и нажать на кнопку «Выполнить». Полученный результат совпадает с результатом модульного тестирования, функционал работает корректно (рисунок 19).

Рисунок 19 — Анализ комментариев для тестирования

Аналитика Вконтакте

Настройки Вконтакте

Логин: belyakovpnz@gmail.com

Пароль: ••••••••

Выбор задачи

Выберите задачу: Анализ комментариев к посту

URL поста: vk.com/wall-101982925_15779013

Выполнить

Результаты

Анализ комментариев:
Комментарии пользователей к новости о чемпионстве ФК "Краснодар" можно разделить на несколько категорий:

- **Поздравления и поддержка**:**
- Многие пользователи поздравляют "Краснодар" с чемпионством, выражая уважение и поддержку. Примеры: "Поздравляю Краснодар со столетием!", "С праздником болельщики ФК Краснодар!!!", "Поздравляю, Краснодар! Наконец-то!"
- **Критика и негатив**:**
- Некоторые комментарии содержат негативные высказывания, направленные как на "Краснодар", так и на других клубов, особенно "Зенит". Примеры: "Позор краснодар", "Несмотря на то, что Газпромнит тянули за уши, не всё"

рования тематик групп, на которых подписан пользователь необходимо выбрать соответствующий пункт меню и ввести ID пользователя. Полученный тематики соответствуют ранее полученный результат, функция работает корректно (рисунок 20).

Аналитика ВКонтакте

Настройки ВКонтакте

Логин: belyakovpnz@gmail.com

Пароль: ••••••••

Выбор задачи

Выберите задачу: Анализ групп пользователя

ID пользователя: 386107338

Выполнить

Результаты

Анализ групп:
 Группа коллег
 Университет
 Образование
 Творчество
 Наука
 Юмор
 Цирк
 Юмор
 Культурный центр
 Юмор
 Творчество
 Творчество
 Общество

Рисунок 20 — Анализ тематик групп

Для тестирования создания графиков необходимо выбрать нужный пункт в меню (рисунок 21). График построен корректно.

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		42

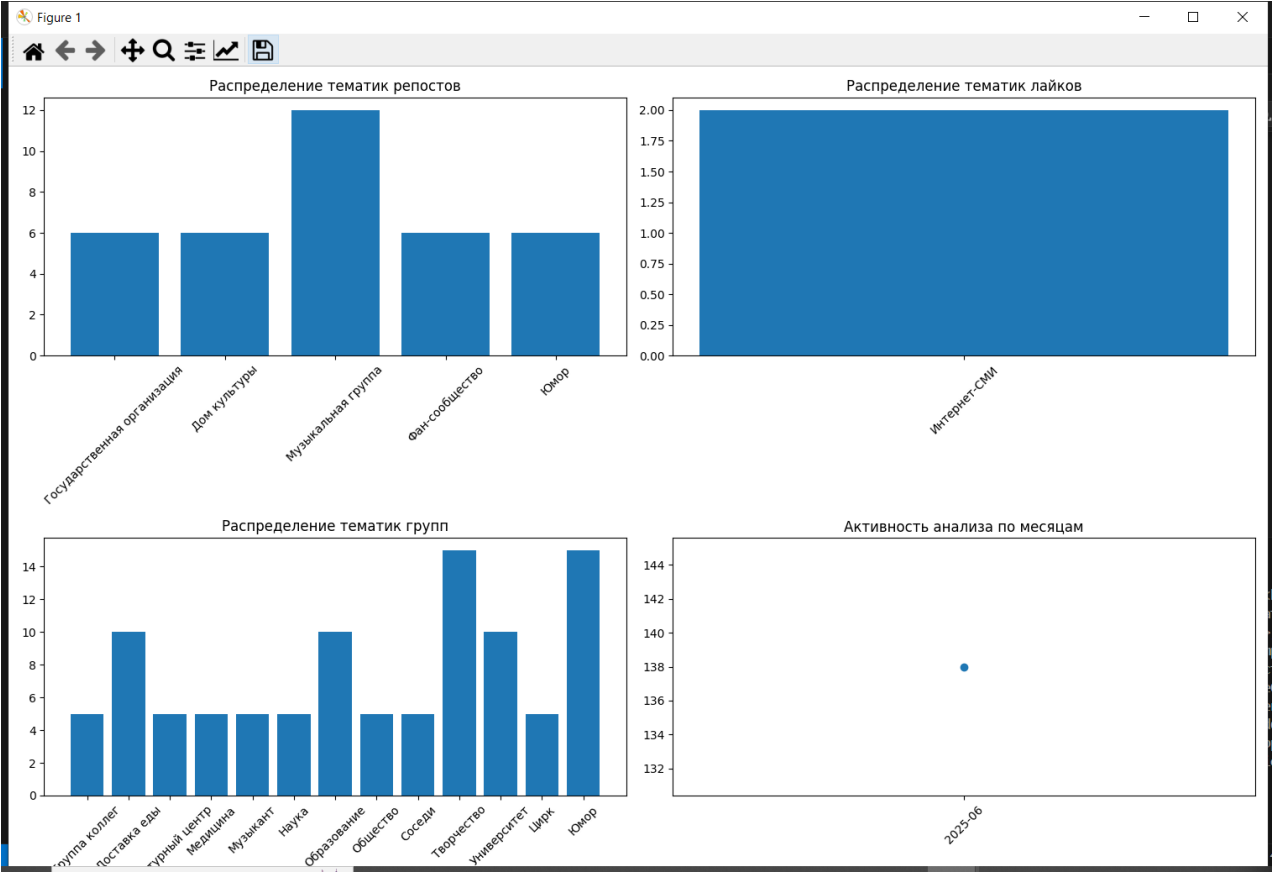


Рисунок 21 — Аналитический отчет

5 Оценка затрат на разработку и внедрение приложения для анализа поведенческих реакций пользователей интернет-приложений

5.1 Организация разработки приложения

Проект был разработан командой разработчиков, представленной в таблице 9. В современной индустрии разработки приложений дизайнер часто трудоустроен по ГПХ, так как написание и отладка программного кода зачастую занимают больше времени, а дизайнер выполняет свою работу за небольшой заранее известный срок в начале разработки проекта.

Для организации труда руководителя проекта и программиста было закуплено оборудование, представленное в таблице 9.

Таблица 9 – Оборудование

Название	Количество, штук	Магазин	Стоимость, рублей
Стол компьютерный	2	DNS	9900
Компьютер	2	DNS	72100
Монитор	2	DNS	5000
Мышь	2	DNS	4500
Клавиатура	2	DNS	3000
Сетевой фильтр	1	DNS	4000

Состав коллектива разработчиков отражен в таблице 10.

Таблица 10 – Состав коллектива разработчиков

Специалист	Количество, человек	Описание занятости
Руководитель проекта	1	– управление проектом; – набор команды исполнителей; – управление ресурсами проекта; – контроль соблюдения требований заказчика; – приемка и внедрение проекта; – анализ предметной области.
Программист	1	– создание базы данных; – написание, отладка и тестирование программного кода – разработка и утверждение программной документации; – приемка и внедрение программного кода.
Дизайнер	1	– разработка интерфейса приложения; – разработка оформления пользовательской документации.

Данные по выплате заработной платы сотрудников в штате представлены в таблице 10, а дизайнера, трудоустроенного по договору ГПХ в таблице 11. Стоит отметить, что заработанная плата в таблицах не включает отчисления на страховые взносы, которые будут рассчитаны позже.

Таблица 10 – Сведения о заработной плате

Специалист	Количество, человек	Сумма заработной платы, рублей
Руководитель проекта	1	100000
Программист	1	80000

Таблица 11 – Сведения о выплатах по ГПХ

Специалист	Количество, человек	Сумма выплаты, рублей
Дизайнер	1	100000

5.2 Расчет себестоимости разработки приложения

Расчет затрат на разработку приложения можно выполнить по формуле:

$$K_{\text{п}} = \Phi_{\text{з/п}}[(1+\beta_{\text{д}})(1+\beta_{\text{с}})+\beta_{\text{н}}+\beta_{\text{пр}}]+t_{\text{ЭВМ}}C_{\text{м-ч}},$$

где $\Phi_{\text{з/п}}$ – фонд основной заработной платы разработчиков и других исполнителей работ, р.; $\beta_{\text{д}}$ – коэффициент дополнительной зарплаты, можно принимать равным 0,1; $\beta_{\text{с}}$ – коэффициент отчислений на социальные нужды от основной и дополнительной заработной платы, равен 0,3; $\beta_{\text{н}}$ – коэффициент накладных расходов организации, разрабатывающей проект, можно принимать 0,7; $\beta_{\text{пр}}$ – коэффициент прочих расходов, принимать 0,1; $t_{\text{ЭВМ}}$ – машинное время, затраченное для отладки модулей скрипта, ч., $C_{\text{м-ч}}$ – стоимость машино-часа работы ЭВМ, руб.

Укрупненный расчет фонда основной заработной платы исполнителей работ по разработке приложения производится по формуле:

$$\Phi_{\text{з/п}} = \sum_{i,j} t_{ij} C \text{ где, } \sum_{i,j} t_{ij} -$$

где это суммарная трудоемкость работ по разработке приложения, чел.-ч. (чел.-дн.), например, 21 чел.-дн. C – тарифная ставка часовая (дневная) разработчиков и других исполнителей работ, рассчитывается по формуле: $C = \frac{ЗП_{\text{м}}}{Д}$, где $ЗП_{\text{м}}$ – среднемесячная заработная плата разработчиков и других исполнителей работ, равна 105000 руб.; $Д$ – количество рабочих дней в месяце, равно 21 дню. Дневная тарифная ставка C численно равна: $C = 105000/21 = 5000$ руб.

Себестоимость машино-часа работы ПК определяется по формуле:

$$C_{\text{м-ч}} = \frac{З_{\text{п}} + А + З_{\text{э}} + З_{\text{р}} + З_{\text{н}} + З_{\text{д}}}{\Phi},$$

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		46

где $Z_{\text{п}}$ – затраты на заработную плату обслуживающего персонала с учетом всех отчислений, руб.;

A – годовая сумма амортизации, руб.; $Z_{\text{э}}$ – затраты на силовую электроэнергию, руб.; $Z_{\text{р}}$ – затраты на ремонт и обслуживание оборудования в год, руб.; $Z_{\text{м}}$ – затраты на материалы в год, руб.; $Z_{\text{н}}$ – накладные расходы, руб.; $\Phi_{\text{д}}$ – действительный годовой фонд времени работы КСА, ч. Так как нет необходимости в обслуживании ПК, то расчет затрат на заработную плату обслуживающего персонала проводить не надо ($Z_{\text{п}} = 0$). Годовые амортизационные отчисления по КСА считаются по формуле: $A = \text{СКСА} \times \text{На} / 100$ Где, СКСА— стоимость ПК, используемого при отладке; Стоимость ПК, необходимой для разработки АС составляет 50000 руб. На – норма амортизации, 25%. Таким образом, сумма годовой амортизации составит: $A = 50000 \times 0,25 = 12500$ руб. Затраты на электроэнергию в год $Z_{\text{э}}$ определяются следующим образом:

$$Z_{\text{э}} = P \times \Phi_{\text{В}} \times \text{ЦЭ}$$

где, P — установленная мощность, кВт; ЦЭ — стоимость силовой электроэнергии, р/кВт; $\Phi_{\text{В}}$ — время, в течение года, когда КСА потребляет электроэнергию, ч. 30 Затраты на текущие ремонты $Z_{\text{р}}$ и на материалы $Z_{\text{м}}$ в год берутся по данным бухгалтерии. При укрупненном расчете их сумма может быть принята от 4,5 до 10 % от стоимости ЭВМ. $Z_{\text{р}} + Z_{\text{м}} = 50000 \times 0,1 = 5000$ руб. В накладные расходы включаются затраты на оплату труда АУП, затраты на амортизацию и содержание площадей, затраты на отопление, освещение и прочие. Примем их равными 20% от стоимости ЭВМ. Годовой фонд времени $\Phi_{\text{д}}$ устанавливается, исходя из номинального фонда времени и времени профилактики оборудования и ремонтов:

$$\Phi_{\text{д}} = S \times h \times D - \text{ТПР}$$

где, S — продолжительность смены, ч.; h — количество смен; D — число рабочих дней в году, дн.; ТПР— время ремонтов и профилактики оборудования в год, ч. Исходные данные для расчета затрат на разработку приведены в таблице 11.

					ПГУ1.090501.02.001 ПЗ	Лист
						47
Изм.	Лист	№ докум.	Подпись	Дата		

Таблица 11 – Исходные данные для расчета затрат на разработку

Параметр	Значение
Коэффициент дополнительной заработной платы - β_d	0,1
Коэффициент отчислений на социальные нужды - β_c	0,3
Коэффициент накладных расходов - β_n	0,7
Коэффициент прочих расходов - $\beta_{пр}$	0,1
Дневная ставка разработчика - С (руб.)	5000
Суммарная трудоемкость -1 (чел-дней)	21
Продолжительность смены - S (ч)	6
Количество смен - h	1
Число рабочих дней в году	247
Параметр	Значение
Время ремонта и профилактики оборудования в год - $T_{пр}$	26
Срок эксплуатации ПЭВМ – $t_{экспл}$ (мес)	48
Стоимость ПЭВМ - $C_{кса}$ (руб)	50000
Потребляемая мощность ПЭВМ - W_y (кВт~ч)	0,3
Стоимость электроэнергии $C_э$ (р/кВт-ч)	4.24

Минимальная цена разработки Z_{min} складывается из полных затрат на разработку КП и минимально необходимой суммы прибыли P_{min} , размер которой позволял бы на минимальном уровне осуществить

самофинансирование организации-разработчика после всех обязательных платежей и выплаты налогов.

$$Z_{\min} = КП + П_{\min},$$

где сумма прибыли $П_{\min}$ рассчитывается исходя из планируемого минимального уровня рентабельности затрат организации-разработчика.

$$П_{\min} = К П \times R_{\min} 100,$$

где $R_{\min}$ — минимальный уровень рентабельности, 20%. Таким образом, минимальная сумма прибыли: $П_{\min} = 2826000 \times 0,20 = 565200$ руб. Следовательно, $Z_{\min} = 2826000 + 565200 = 3391200$ руб. Так как система разрабатывается под заказчика, то цена продажи будет договорной. Учитывая, что затраты на разработку будут эффективными, если доходы покроют их полностью, и будет получена минимально необходимая сумма прибыли, цену продажи с учётом НДС 20% установили равной 2712960 руб. Единовременные затраты на внедрение системы $К$ определяются по формуле:

$$К = З_{\text{пр}} + КК, 32,$$

где $З_{\text{пр}}$ – затраты на приобретение модуля с учетом затрат на адаптацию и обучение персонала; $КК$ – величина капитальных затрат; δ – коэффициент участия КСА. Так как на подготовку к разработке данной системы не требуются дополнительные затраты, примем $З_{\text{пр}} = 0$. Коэффициент участия КСА используется только для данной регистратуры, поэтому коэффициент участия δ равен 1. Комплекс средств автоматизации используется для одной регистратуры и коэффициент участия равен единице. Величина капитальных затрат определяется по формуле:

$$Кк = К_{\text{ктс}} + К_{\text{м}} + К_{\text{инв}} + К_{\text{зд}} + К_{\text{ос}} + К_{\text{тр}} + К_{\text{соп}} - К_{\text{выс}},$$

где, $К_{\text{ктс}}$ — сметная стоимость КТС, 50000 руб.; $К_{\text{м}}$, $К_{\text{инв}}$, $К_{\text{зд}}$ — затраты на установку, монтаж и запуск КТС в работу, на производственный инвентарь, на строительство и реконструкцию зданий для размещения КТС. Так как для установки КТС не требуется дополнительной реконструкции здания, примем $К_{\text{зд}} = 0$, а $К_{\text{м}}$ равным 10% от стоимости КТС, т.е. 5000 р, $К_{\text{инв}}$ — 1000000 р; $К_{\text{ос}}$ — сумма оборотных средств, 0 р.; $К_{\text{тр}}$ — транспортно-

					ПГУ1.090501.02.001 ПЗ	Лист
						49
Изм.	Лист	№ докум.	Подпись	Дата		

заготовительные расходы, примем равными 10% от стоимости КТС, т. е. 5000 р.; Ксоп — сметная стоимость системы стандартного обеспечения применения КТС, примем равной 15% от стоимости КТС, т. е. 7500 р.; Квыс ~ сумма высвобожденных средств в результате ввода в действие КСА, 1000000 р. Таким образом, единовременные затраты на внедрение АС составят: $K_{вн} = 339120 + 50000 + 5000 + 1000000 + 5000 + 7500 - 1000000 = 411620$ руб.

Для расчета текущих затрат на функционирование АС воспользуемся методом, который предполагает определение текущих затрат посредством основных составляющих:

$$ЗТЕК = ЗСОП + ЗКСА$$

,где Зкса— годовые текущие затраты на эксплуатацию КСА, руб/ год; Ззп — годовые затраты на заработную плату специалистов в условиях функционирования модуля с начислениями, руб/ год. Внедрение АС подразумевает высвобождение специалиста, занимающегося обработкой данных в больнице и, соответственно, не требует затрат на заработную плату специалиста. Поэтому $Ззп = 0$. Затраты ЗКСА определяются по формуле:

$$ЗКСА = ЗКТС + ЗЗД$$

,где ЗКТС— годовые затраты на эксплуатацию КТС без учета заработной платы персонала, руб/ год; ЗСОП — годовые затраты на поддержание и актуализацию системы обеспечения применения КТС (хранение, обновление, контроль данных, программ и другие операции), руб/ год; ЗЗД — годовые затраты на содержание и ремонт производственных помещений, руб/ год. $ЗКТС = 2000 * 7,85 = 15700$ руб. $ЗЗД = 10000$ руб. $ЗСОП = 5000$ руб. $ЗКСА = 15700 + 10000 = 25700$ руб. Таким образом, годовые затраты на функционирование АС: $ЗТЕК = 5000 + 25700 = 30700$ руб. При внедрении АС высвобождается специалист, занимающийся обслуживанием лаборатории. Годовая экономия на зарплате с учетом страховых взносов ЭЗ при ставке зарплаты специалиста 90000 руб. составит: $ЭЗ = 90000 \text{ руб.} * 12 \text{ мес.} * 2 = 2160000$ руб. 34 Чтобы получить чистую годовую экономию, необходимо вычесть затраты на функционирование АС: $ЭГ = ЭЗ - ЗТЕК =$

					ПГУ1.090501.02.001 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

2160000 руб. – 30700 руб. = 2129300 руб. Для оценки эффективности затрат на создание (внедрение) АС можно применять следующие методы: метод расчета чистой дисконтированной стоимости; метод полного возмещения инвестиций. Метод расчета чистой дисконтированной стоимости предполагает, что предприятие заранее задает минимально допустимую ставку процента, при которой капитальные вложения могут считаться эффективными. Такая ставка процента называется расчетной ставкой процента ("субъективная" ставка процента фирмы). Базисом для установления расчетной ставки может быть ставка процента на заемный капитал, который предприятие должно выплачивать кредитору. Если процентная ставка не учитывает инфляцию, то ее называют номинальной ставкой процента i_n . Реальная ставка процента i_r учитывает уровень инфляции ln . Реальная ставка процента i_r рассчитывается по формуле Фишера:

$$i_r = (i_n - ln) / (1 + ln),$$

где. i_r , i_n , ln заданы в десятичных дробях. При низких процентных ставках и низкой инфляции реальная ставка процента может быть найдена по формуле: $i_r = i_n - ln$ Чистая дисконтированная стоимость — это суммарный эффект за период функционирования инвестиций с учетом приведения всех результатов и затрат к начальному году (дисконтирование с помощью расчетной ставки процента).

Затраты на разработку системы – 2826000 руб. С учетом того, что систему можно тиражировать, а число потенциальных потребителей 10, то затраты на одно лицо без НДС и дополнительных расходов равна 282600 руб. без учета расходов на внедрение АС. Стоимость аналогичных АС на рынке 300000 руб. Следовательно, разработка АС целесообразна, потому что затраты на покупку АС на рынке на 10 лиц составляет 3000000 без учета расходов на внедрение.

5.3 Оценка эффективности приложения

					ПГУ1.090501.02.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		51

Эффективность приложения зависит от материальных и нематериальных эффектов от его внедрения.

К материальным эффектам можно отнести улучшение производительности труда SMM специалиста (ориентировочно на 10%), что позволит экономить около 8000 рублей в месяц.

К нематериальным эффектам улучшение условий труда за счет уменьшения количества рутинной работы.

Для улучшения экономической эффективности можно оценивать данный параметр. Оценка даёт возможность посмотреть на изъяны эффективности и даёт перспективы развития для данного критерия. Можно улучшить систему показателей, соотнести их значения, сопоставить их между собой. Приложение показывает, какие данные более востребованы у пользователей и даёт возможность улучшить покупательскую способность клиентов, за счёт внедрения более таргетированной рекламы. Приложение хорошо анализирует данные, помогает продукту лучше распространяться и масштабировать. Даёт больше притока клиентов. Такие профессии как блогер становятся более высокооплачиваемыми, а государство получает больше налогов.

5.4 Анализ рисков при разработке и внедрении приложения

Риски взаимодействия между заказчиком и исполнителем – это риски, связанные с отсутствием коммуникации между исполнителем и заказчиком или их представителями.

Риски планирования – это риски, связанные с отсутствием навыков планирования сроков выполняемых работ, как менеджером, так и исполнителями. Часто слишком мало времени закладывается на этап тестирования и исправления ошибок в исходном коде, предпроектного обследования. Для минимизации риска, проектный руководитель следил за четким выполнением сроков на каждом этапе проекта.

Недостаточное обсуждение задач или архитектуры может негативно сказаться на разрабатываемом приложении. Для уменьшения последствий от

					ПГУ1.090501.02.001 ПЗ	Лист
						52
Изм.	Лист	№ докум.	Подпись	Дата		

таких рисков на начальном этапе проекта разработано соглашение и обозначена

в договоре частота проведения встреч по проекту, куратора проекта со стороны заказчика и исполнителя, а также состав рабочей группы.

Функциональные риски – это риски, связанные с неверным выполнением функций, возложенных на систему. Для минимизации данных рисков необходим тщательный анализ требований заказчика, разработка моделей и привлечение к работе грамотных специалистов.

Уход из компании ключевых сотрудников является ситуацией с повышенным риском. Для предотвращения этого необходимо инвестировать время и средства в обучение кадров чтобы сгладить возможные последствия.

Риски использования нелицензионного ПО должны предотвращаться на этапе проектирования приложения. По возможности следует выбирать программные модули, которые являются решениями с открытым исходным кодом и лицензией, одобренной OSI.

					ПГУ1.090501.02.001 ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подпись	Дата		

6 Общие принципы обеспечения безопасности на рабочем месте оператора

6.1 Актуальность вопросов безопасности человека на рабочем месте

С каждым годом человек все больше исследует и использует IT технологии в своей жизни. Это стало возможно благодаря распространению ПЭВМ (персональных электронных вычислительных машин). Они стали употребляться почти на каждом рабочем месте, именно поэтому необходимо помнить про безопасность оператора ПЭВМ, сохранении его здоровья и работоспособности.

Использование ЭВМ связано с распознаванием печатного и рукописного текста, анализ изображений на экране, долгим нахождением в одной позе, и др., что может пагубно сказаться на здоровье человека, если не придерживаться нормами использования ЭВМ.

На основании ГОСТ 12.0.003-74 вредные производственные факторы подразделяются на 4 категории:

- физические, к ним относятся недостаточная освещенность рабочего места, повышенная яркость света, отклонения от нормы температуры воздуха, высокая влажность воздуха, повышенный уровень шума, и др.;
- психофизиологические, к ним относятся нервно-психические и физические перегрузки, перенапряжение ушей и глаз;
- биологические, к ним относятся микроорганизмы и макроорганизмы, которые могут вызвать различные заболевания;
- химические, к ним относятся превышения в рабочей зоне аммиака, фенола, озона, оксида углерода и др.

					ПГУ1.090501.02.001 ПЗ	Лист
						54
Изм.	Лист	№ докум.	Подпись	Дата		

6.2 Техника безопасности при работе с электроприборами

Персональная электронно-вычислительная машина – это электроприбор, следовательно, работа с ним требует знаний правил техники безопасности:

- при работе с электрооборудованием следует использовать заземленные розетки;
- после выключения электроприборов из сети, нельзя тянуть их за шнур и находиться рядом с мокрыми руками;
- неисправность у прибора нельзя чинить до момента отключения от сети;
- до включения прибора в сеть необходимо проверить сохранность шнура (наличие заломов и повреждения изоляции);
- не оставлять включенные приборы без присмотра.

В состав ПЭВМ входят монитор, блок питания, процессор – все они создают электромагнитные поля. Такие поля создают еще распределительные щиты, электропроводка, линии электропередач.

6.3 Влияние ПЭВМ на физиологию человека

Работа за компьютером подразумевает длительное напряжение на зрительные органы и опорно-двигательный аппарат. Если не контролировать время пребывания за ЭВМ, то это может привести к нарушению зрения, искривлению позвоночника, а также нарушению кровообращения в организме.

Нагрузка на зрительный аппарат может вызвать нарушения в работе ЦНС (центральной нервной системы). При длительной работе за компьютером снижается фокусировка и острота зрения, увеличивается чувствительность глаз, они могут слезиться и краснеть.

Длительное нахождение человека в одной позе приводит к быстрому развитию утомляемости.

При работе за компьютером стоит обратить внимание на:

- высоту рабочей зоны и стула;

					ПГУ1.090501.02.001 ПЗ	Лист
						55
Изм.	Лист	№ докум.	Подпись	Дата		

- наличие опорной спинки и подлокотников;
- угол наклона головы при работе;
- угол наклона экрана компьютера.

Огромное влияние на здоровье человека оказывает уровень шума в помещении. Согласно СанПиН 2.2.3670-20 "Санитарно-эпидемиологические требования к условиям труда", эквивалентный уровень шума не должен превышать 50 дБА. Если шум выше нормы будет длительно присутствовать на рабочем месте сотрудника, то будут ухудшены его условия труда и нанесен вред здоровью. А именно:

- снижается концентрация внимания;
- повышается усталость;
- снижается работоспособность сотрудника.

Для того, чтобы снизить уровень шума в помещении используется шумоизоляция.

6.4 Техника безопасности при работе с персональным компьютером

На основе типовой инструкции по охране труда при работе на ПК (персональном компьютере) ТОИ Р-45-084-01, которая была утверждена Приказом Минсвязи Российской Федерации №162 от 02.07.2001, деятельность на ЭВМ делится на 3 группы:

- группа А – чтение информации с монитора подразумевает два перерыва по 15 минут, без учета обеденного перерыва;
- группа Б – набор текста подразумевает перерывы по 10 минут раз в час;
- группа В – творческий труд подразумевает перерывы по 15 минут раз в час.

Выделенное время рекомендуется уделять физической нагрузке, такой как зарядка или гимнастика для глаз.

					ПГУ1.090501.02.001 ПЗ	Лист
						56
Изм.	Лист	№ докум.	Подпись	Дата		

На основании всего вышесказанного, можно сделать вывод, что для сохранения здоровья сотрудников и ограничения от негативного влияния ПЭВМ необходимо использовать только сертифицированную и качественную технику. Обеспечивать сотрудникам рациональный режим труда и перерывов.

Необходимо отметить, что сертифицированная техника не является полной гарантией безопасности условий труда, необходимо регулярно проводить проверку всей используемой техники.

					ПГУ1.090501.02.001 ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

Заключение

В ходе выполнения дипломного проекта был изучен процесс анализа поведенческих реакций пользователей интернет-приложений. В начале были выбраны средства разработки, подходящие для реализации поставленной задачи. Затем были разработаны логическая и физическая модели базы данных и комплект UML диаграмм: развертывания, классов и вариантов использования.

В разделе экономики был произведен подсчет стоимости разработки программного комплекса, включающий кроме фонда оплаты труда амортизацию и закупку компьютерной техники. Так же оценено внедрение программного комплекса и сроки окупаемости проекта.

В разделе БЖД был оценен вред ПЭВМ на физиологию человека, приведена техника безопасности при использовании системы.

В полученном приложении были реализованы все заявленные функции, составлена документация и проведено тестирование программы. Дипломный проект реализован с учетом требований стандартов ГОСТ, ЕСПД, ЕСКД. Техническое задание выполнено в полном объеме.

					ПГУ1.090501.02.001 ПЗ	Лист
						58
Изм.	Лист	№ докум.	Подпись	Дата		

Список использованных источников

1. Как автоматизация контента с LLM меняет правила игры в копирайтинге и делает вас звездой маркетинга / [Электронный ресурс] // Яндекс. Дзен : [сайт]. — URL: <https://dzen.ru/a/aEQ1kU32L0xbNqWs> (дата обращения: 02.04.2025).
2. Как автоматизация контента с LLM меняет правила игры в копирайтинге и делает вас звездой маркетинга / [Электронный ресурс] // IBM : [сайт]. — URL: <https://www.ibm.com/think/topics/mistral-ai> (дата обращения: 02.04.2025).
3. Программирование на Python для начинающих [Текст] / Васильев А.Н.— . — Москва: ДМК Пресс, 2019 — 506 с.
4. Искусственный интеллект в столичной медицине будет использоваться более чем в 50 направлениях лучевых исследований / [Электронный ресурс] // mos.ru : [сайт]. — URL: <https://www.mos.ru/mayor/themes/8991050/> (дата обращения: 02.04.2025).
5. TIOBE Index for May 2025 / [Электронный ресурс] // TIOBE : [сайт]. — URL: <https://www.tiobe.com/tiobe-index/> (дата обращения: 11.05.2025).
6. The Top Programming Languages 2024 / [Электронный ресурс] // IEEE Spectrum : [сайт]. — URL: <https://spectrum.ieee.org/top-programming-languages-2024> (дата обращения: 12.05.2025).
7. StarUML2 documentation / [Электронный ресурс] // StarUML : — URL: <https://staruml.readthedocs.io/en/latest/index.html> обращения: 28.05.2025).
8. Шлее Макс Qt 5.10. Профессиональное программирование на C++ [Текст] / Шлее Макс — . — Санкт-Петербург: БХВ, 2018 — 1072 с.
9. Генеративное глубокое обучение. Творческий потенциал нейронных сетей [Текст] / Дэвид Фостер — 2-е изд. — Берлин: Springer, 2009 — 417 с.
10. ГОСТ 12.0.0003-74 ССБТ. Библиографическая запись. Опасные и вредные производственные факторы. Классификация (с Изменением № 1). Изд-во стандартов, 2002. — 5 с.

					ПГУ1.090501.02.001 ПЗ	Лист
						59
Изм.	Лист	№ докум.	Подпись	Дата		

11. СанПиН 2.2.3670-20. Санитарно-эпидемиологические требования к условиям труда. Изд-во стандартов, 2012. – 47с.

12. Типовая инструкция по охране труда при работе на персональном компьютере. – 2001. – 3 с.

					ПГУ1.090501.02.001 ПЗ	Лист
						60
Изм.	Лист	№ докум.	Подпись	Дата		

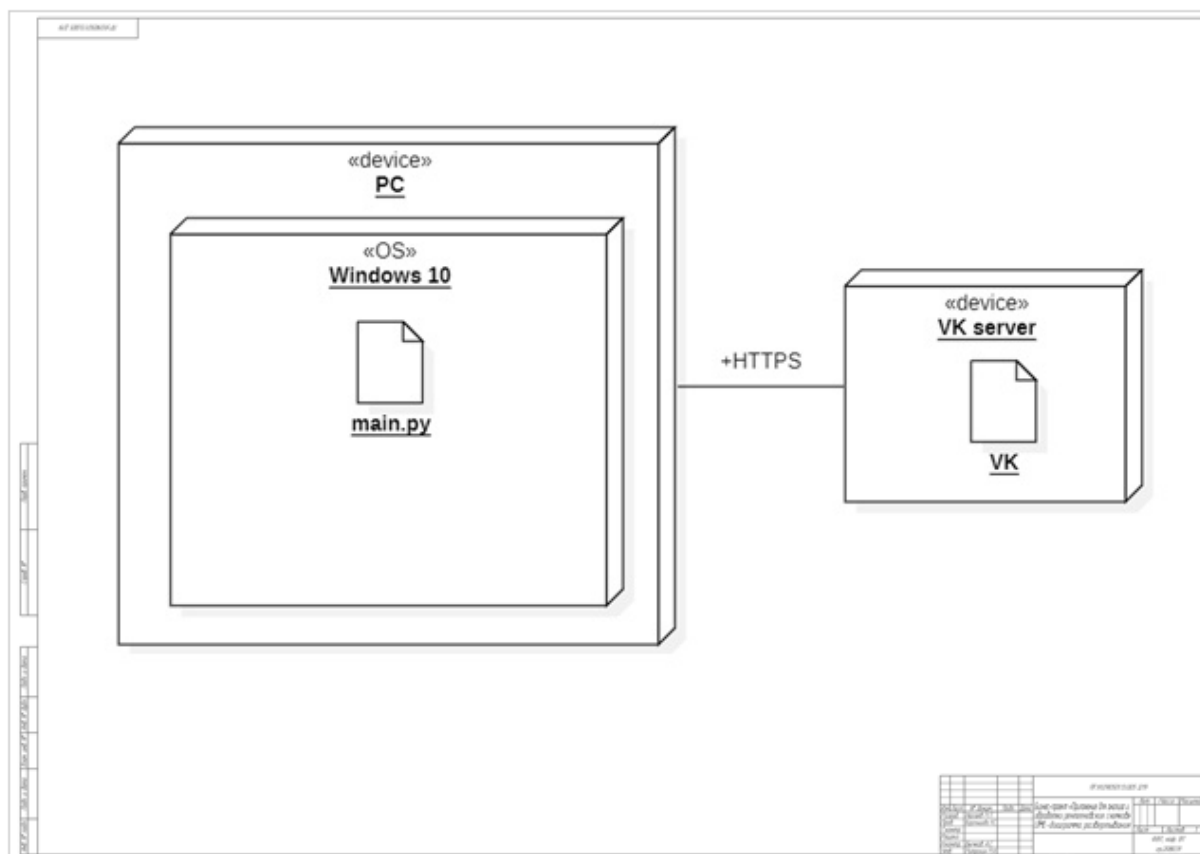
Приложение А

(обязательное)

Анализ поведенческих реакций пользователей интернет-приложений
UML-диаграммы

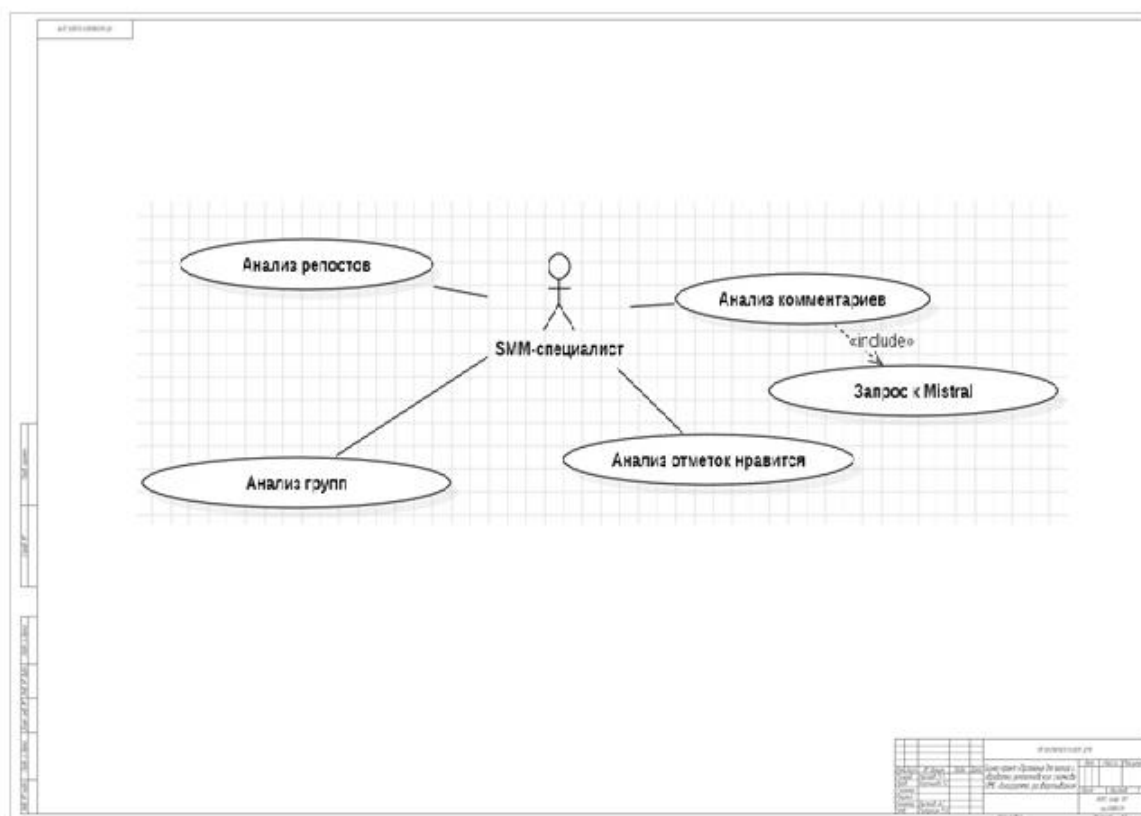
					ПГУ1.090501.02.001 ПЗ	Лист
						61
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение А.1 – Диаграмма развертывания

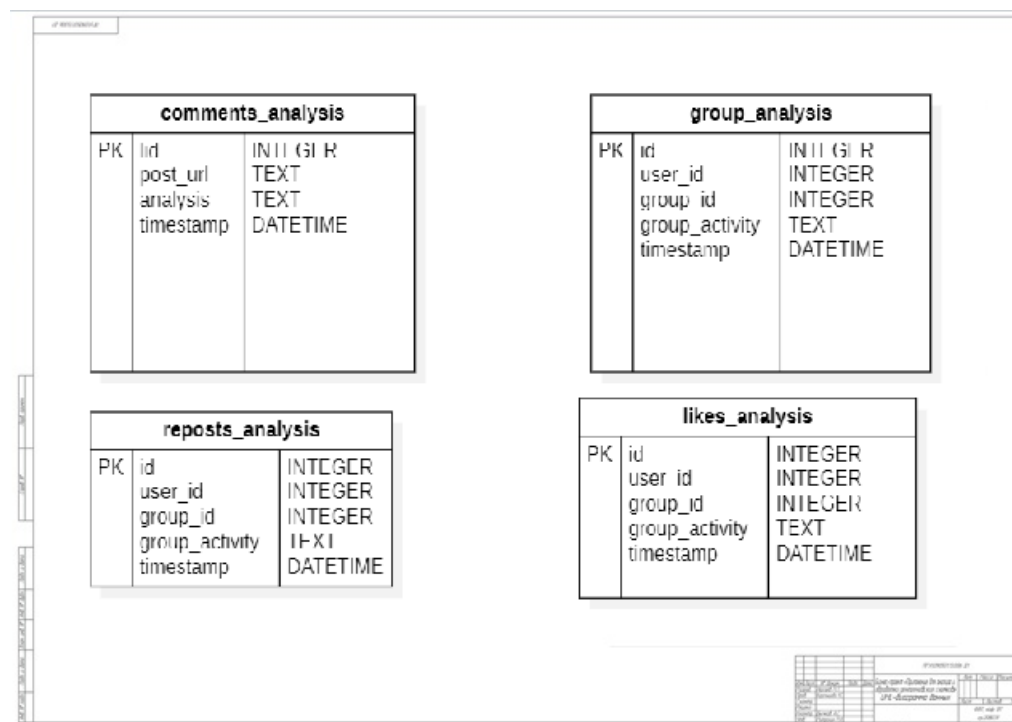


					ПГУ1.090501.02.001 ПЗ	Лист
						62
Изм.	Лист	№ докум.	Подпись	Дата		

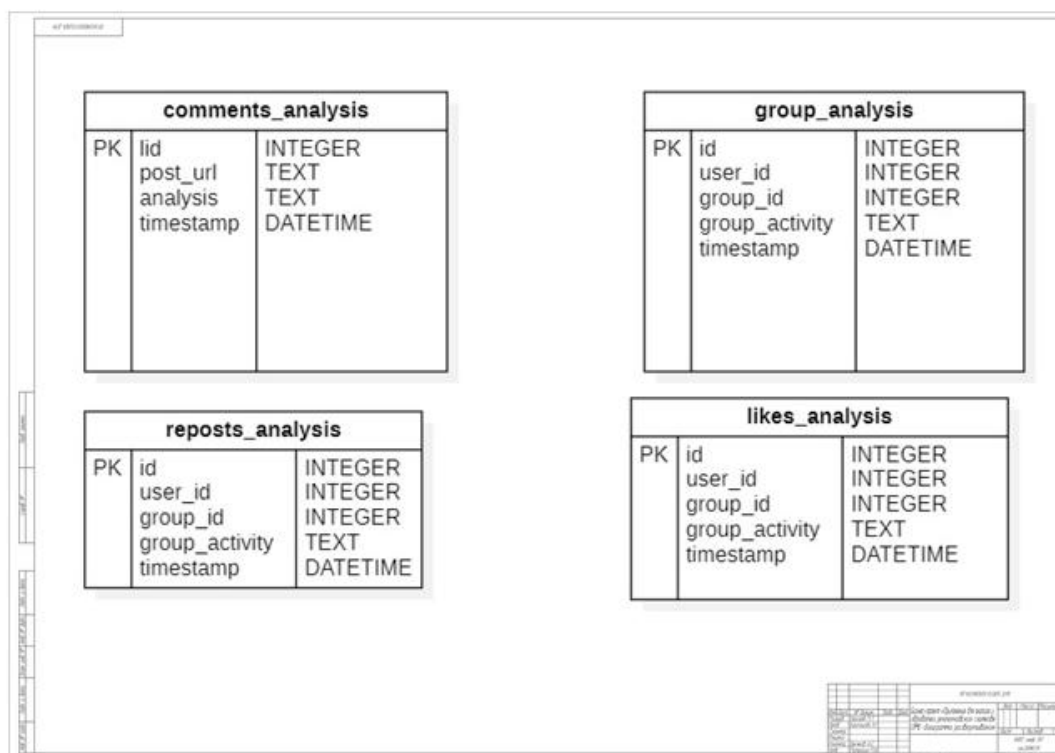
Приложение А.2 – Диаграмма вариантов использования



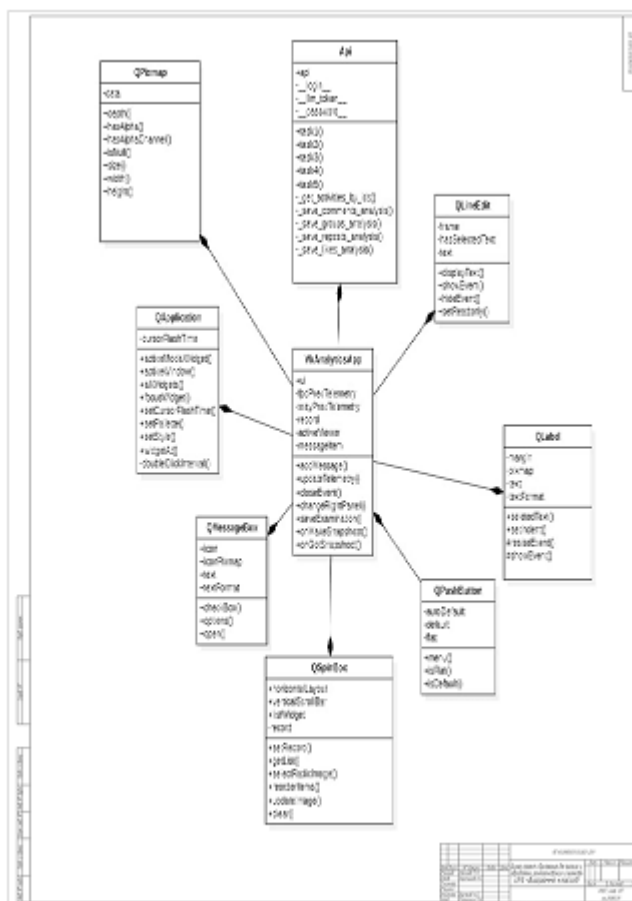
Приложение А.3 – Логическая модель данных



Приложение А.4 – Физическая модель данных



Приложение А.5 – Диаграмма классов



Приложение Б

(обязательное)

Анализ поведенческих реакций пользователей интернет-приложений
SQL-скрипт

					ПГУ1.090501.02.001 ПЗ	Лист
						66
Изм.	Лист	№ докум.	Подпись	Дата		

```
PRAGMA foreign_keys=on;
```

```
CREATE TABLE IF NOT EXISTS reposts_analysis (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    user_id INTEGER,  
    group_id INTEGER,  
    group_activity TEXT,  
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP)
```

```
CREATE TABLE IF NOT EXISTS likes_analysis (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    user_id INTEGER,  
    group_id INTEGER,  
    group_activity TEXT,  
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP)
```

```
CREATE TABLE IF NOT EXISTS comments_analysis (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    post_url TEXT,  
    analysis TEXT,  
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP)
```

```
CREATE TABLE IF NOT EXISTS groups_analysis (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    user_id INTEGER,  
    group_id INTEGER,  
    group_activity TEXT,  
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP)
```

					ПГУ1.090501.02.001 ПЗ	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение В

(обязательное)

Анализ поведенческих реакций пользователей интернет-приложений

Листинг программы

					ПГУ1.090501.02.001 ПЗ	Лист
						68
Изм.	Лист	№ докум.	Подпись	Дата		

```

import vk
import time
import sys
import os
import sqlite3
import matplotlib.pyplot as plt
from datetime import datetime
from collections import Counter
from mistralai import Mistral

# Конфигурация
DB_NAME = "vk_analytics.db"

class Api:
    def __init__(self, login, password, mistral_token =
'culR2kHCBHMn88DYJnd7Ciq3GtOXm4kr'):
        self.__token__ =
'vk1.a.PO9WUx8gJzUQIG2TT03PJSy94s2lNTkfQm9nPqFlVMGFL5oAA3A_bJ
GGbkS8j7uRCBtMFiYLA2hPhzu4rGqzS-kmpcJ5ZHgUCW-
auFytM2shlqvMBxr7Xo1X_Nqt35k6VIKjDcOs_8l6lLjEFQNsATPQA7nf-
qQzFh0L6D2k8kxhoe80PzIUk8hqyXFLt85zm5gZvVHthIb4OZHZeQQXg'
        self.llm_key = mistral_token
        self.api = vk.API(access_token=self.__token__,
v='5.199')
        self._init_db()

    def _init_db(self):
        """Инициализация базы данных"""
        with sqlite3.connect(DB_NAME) as conn:
            cursor = conn.cursor()

            # Таблица для хранения анализа репостов
            cursor.execute('''CREATE TABLE IF NOT EXISTS
                                reposts_analysis (
                                    id INTEGER PRIMARY KEY AUTOINCREMENT,
                                    user_id INTEGER,

```

					ПГУ1.090501.02.001 ПЗ	Лист
						69
Изм.	Лист	№ докум.	Подпись	Дата		

```

        group_id INTEGER,
        group_activity TEXT,
        timestamp DATETIME DEFAULT
CURRENT_TIMESTAMP)'''

# Таблица для хранения анализа лайков
cursor.execute('''CREATE TABLE IF NOT EXISTS
        likes_analysis (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        user_id INTEGER,
        group_id INTEGER,
        group_activity TEXT,
        timestamp DATETIME DEFAULT
CURRENT_TIMESTAMP)''')

# Таблица для хранения анализа комментариев
cursor.execute('''CREATE TABLE IF NOT EXISTS
        comments_analysis (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        post_url TEXT,
        analysis TEXT,
        timestamp DATETIME DEFAULT
CURRENT_TIMESTAMP)''')

# Таблица для хранения анализа групп
cursor.execute('''CREATE TABLE IF NOT EXISTS
        groups_analysis (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        user_id INTEGER,
        group_id INTEGER,
        group_activity TEXT,
        timestamp DATETIME DEFAULT
        CURRENT_TIMESTAMP)''')

conn.commit()

```

					ПГУ1.090501.02.001 ПЗ	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

```

def _save_reposts_analysis(self, user_id, activities):
    """Сохраняет анализ репостов в базу данных"""
    with sqlite3.connect(DB_NAME) as conn:
        cursor = conn.cursor()
        for activity in activities:
            # Для репостов у нас нет group_id,
            сохраняем только тематику
            cursor.execute('''INSERT INTO
reposts_analysis (user_id, group_activity)
VALUES (?, ?)''', (user_id,
activity))
            conn.commit()

def _save_likes_analysis(self, user_id, activities):
    """Сохраняет анализ лайков в базу данных"""
    with sqlite3.connect(DB_NAME) as conn:
        cursor = conn.cursor()
        for activity in activities:
            cursor.execute('''INSERT INTO
likes_analysis (user_id, group_activity)
VALUES (?, ?)''', (user_id,
activity))
            conn.commit()

def _save_comments_analysis(self, post_url, analysis):
    """Сохраняет анализ комментариев в базу данных"""
    with sqlite3.connect(DB_NAME) as conn:
        cursor = conn.cursor()
        cursor.execute('''INSERT INTO comments_analysis
(post_url, analysis)
VALUES (?, ?)''', (post_url, analysis))
        conn.commit()

def _save_groups_analysis(self, user_id, activities):
    """Сохраняет анализ групп в базу данных"""
    with sqlite3.connect(DB_NAME) as conn:

```

					ПГУ1.090501.02.001 ПЗ	Лист
						71
Изм.	Лист	№ докум.	Подпись	Дата		

```

        cursor = conn.cursor()
        for activity in activities:
            cursor.execute(''INSERT INTO groups_analysis
(user_id, group_activity)
                                VALUES (?, ?)'', (user_id,
activity))
        conn.commit()

def _get_activities_by_ids(self, ids: str)->list:
    time.sleep(1.0)
    activities = []
    group_info = self.api.groups.getById(group_ids=ids,
fields='activity', filter='groups,publics')['groups']
    for group in group_info:
        if group['type'] == 'event':
            continue
        activity = group.get('activity')
        bad = (activity is None) or (activity == 'Публичная
страница') or (activity == 'Закрытое сообщество') or (activity ==
'Частное сообщество')
        if not bad:
            activities.append(activity)
    return activities

def task1(self, user_id):
    count = 100
    offset = 0
    group_ids = ''

    while True:
        response = self.api.wall.get(owner_id=user_id,
count=count, offset=offset)
        if 'items' not in response:
            break
        items = response['items']
        if not items:

```

					ПГУ1.090501.02.001 ПЗ	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		

```

        break
    for item in items:
        if 'copy_history' in item:
            for copy in item['copy_history']:
                group_id = abs(copy['owner_id'])
                group_ids += f'{group_id},'
    offset += count

    group_ids = group_ids[:-1]
    activities = self._get_activities_by_ids(group_ids)
    self._save_reposts_analysis(user_id, activities)
    return activities

def task2(self, user_id: int, cnt: int) -> list[str]:
    group_ids = ''
    subscriptions_list =
self.api.users.getSubscriptions(user_id=user_id,
extended=0)['groups']['items']
    groups_list = ['- ' + str(x) for x in subscriptions_list]
    posts = {}
    newsfeed = self.api.newsfeed.get(
        filters='post',
        source_ids=', '.join(groups_list),
        count=100, timeout=10)

    for x in newsfeed['items']:
        if x.get('source_id') is None:
            continue
        posts[x['post_id']] = x['source_id']

    if cnt != 1:
        for c in range(cnt - 1):
            next_from = newsfeed['new_from']
            kwargs = {
                'from': next_from,
                'filters': 'post',

```

					ПГУ1.090501.02.001 ПЗ	Лист
						73
Изм.	Лист	№ докум.	Подпись	Дата		

```

        'source_ids': ', '.join(groups_list),
        'count': 100,
        'timeout': 10
    }
    newsfeed = self.api.newsfeed.get(**kwargs)
    for x in newsfeed['items']:
        if x.get('source_id') is None:
            continue
        posts[x['post_id']] = x['source_id']
    time.sleep(1)

liked_posts = []
for post in posts.items():
    try:
        itemID, ownerID = post
        isLiked = self.api.likes.isLiked(
            user_id=user_id,
            item_id=itemID,
            type='post',
            owner_id=ownerID,
            timeout=5)['liked']
    except Exception:
        isLiked = False

    if isLiked:

liked_posts.append(f'vk.com/wall{ownerID}_{itemID}')
        group_ids += str(abs(ownerID)) + ","
        time.sleep(1)

if group_ids == '':
    return []

group_ids = group_ids[:-1]
activities = self._get_activities_by_ids(group_ids)
self._save_likes_analysis(user_id, activities)

```

					ПГУ1.090501.02.001 ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		

```

        return activities

def task3(self, post_url)->str:
    sep = post_url.index('_')
    post_id = post_url[sep+1:]
    owner_id = post_url[post_url.index('wall')+4:sep]

    comments = []
    response = self.api.wall.getComments(owner_id=owner_id,
post_id=post_id, count=100)

    if 'items' in response:
        comments = [item['text'] for item in
response['items']]
    else:
        print("Ошибка при получении комментариев:", response)

    comments_text = "\n".join(comments)
    prompt = f"Проанализируй следующие комментарии
пользователей и дай общую реакцию на
новость:\n{comments_text}\n\nАнализ:"

    try:
        client = Mistral(api_key=self.llm_key)
        chat_response = client.chat.complete(
            model="mistral-small-2503",
            messages=[{"role": "user", "content": prompt}]
        )
        analysis = chat_response.choices[0].message.content
        self._save_comments_analysis(post_url, analysis)
        return analysis
    except Exception as e:
        print("Ошибка при обращении к Mistral API:", str(e))
        return 'Ошибка'

def task4(self, user_id):

```

					ПГУ1.090501.02.001 ПЗ	Лист
						75
Изм.	Лист	№ докум.	Подпись	Дата		

```

        groups_info = self.api.groups.get(user_id=user_id,
fields='activity')
        groups = groups_info['items']
        group_str = ','.join(map(str, groups))
        activities = self._get_activities_by_ids(group_str)
        self._save_groups_analysis(user_id, activities)
        return activities

def task5(self):
    """Построение графиков на основе данных из базы"""
    try:
        with sqlite3.connect(DB_NAME) as conn:
            cursor = conn.cursor()

            # Создаем фигуру с несколькими графиками
            fig, axs = plt.subplots(2, 2, figsize=(15, 10))

            # 1. График распределения тематик репостов
            cursor.execute("SELECT group_activity, COUNT(*)
FROM reposts_analysis GROUP BY group_activity")
            reposts_data = cursor.fetchall()
            if reposts_data:
                activities, counts = zip(*reposts_data)
                axs[0, 0].bar(activities, counts)
                axs[0, 0].set_title('Распределение тематик
репостов')

                axs[0, 0].tick_params(axis='x', rotation=45)

            # 2. График распределения тематик лайков
            cursor.execute("SELECT group_activity, COUNT(*)
FROM likes_analysis GROUP BY group_activity")
            likes_data = cursor.fetchall()
            if likes_data:
                activities, counts = zip(*likes_data)
                axs[0, 1].bar(activities, counts)

```

					ПГУ1.090501.02.001 ПЗ	Лист
						76
Изм.	Лист	№ докум.	Подпись	Дата		

```

        axs[0, 1].set_title('Распределение тематик
лайков')

        axs[0, 1].tick_params(axis='x', rotation=45)

# 3. График распределения тематик групп
cursor.execute("SELECT group_activity, COUNT(*)
FROM groups_analysis GROUP BY group_activity")
groups_data = cursor.fetchall()
if groups_data:
    activities, counts = zip(*groups_data)
    axs[1, 0].bar(activities, counts)
    axs[1, 0].set_title('Распределение тематик
групп')

    axs[1, 0].tick_params(axis='x', rotation=45)

# 4. Общая активность по времени
cursor.execute("""
    SELECT strftime('%Y-%m', timestamp), COUNT(*)
    FROM (
        SELECT timestamp FROM reposts_analysis
        UNION ALL
        SELECT timestamp FROM likes_analysis
        UNION ALL
        SELECT timestamp FROM groups_analysis
    )
    GROUP BY strftime('%Y-%m', timestamp)
    ORDER BY strftime('%Y-%m', timestamp)
""")
timeline_data = cursor.fetchall()
if timeline_data:
    months, counts = zip(*timeline_data)
    axs[1, 1].plot(months, counts, 'o-')
    axs[1, 1].set_title('Активность анализа по
месяцам')

    axs[1, 1].tick_params(axis='x', rotation=45)

```

					ПГУ1.090501.02.001 ПЗ	Лист
						77
Изм.	Лист	№ докум.	Подпись	Дата		

```

plt.tight_layout()
plt.savefig('vk_analytics_report.png')
plt.show()
print("Отчет сохранен как
'vk_analytics_report.png'")

except Exception as e:
    print(f"Ошибка при построении графиков: {str(e)}")

# Основной блок
if __name__ == "__main__":
    print("Ошибка входа. Пожалуйста, отключите двухфакторную
авторизацию.")
    print("Добро пожаловать в приложение для анализа статистики
Вконтакте")
    print("Введите логин от Вконтакте. Нажмите Enter чтобы
получить его из переменной окружения")
    login = input()
    if len(login)==0:
        login = os.environ.get('VK_USER')

    print("Введите пароль от Вконтакте. Нажмите Enter чтобы
получить его из переменной окружения")
    password = input()
    if len(password)==0:
        password = os.environ.get('VK_PASSWORD')

    print("Выберите действие")
    print("1 - анализ репостов")
    print("2 - анализ отметок нравится в ленте")
    print("3 - анализ комментариев")
    print("4 - анализ тематик сообществ пользователя")
    print("5 - построение графиков")

    try:
        variant = int(input("Ваш выбор: "))

```

					ПГУ1.090501.02.001 ПЗ	Лист
						78
Изм.	Лист	№ докум.	Подпись	Дата		

```

except ValueError:
    print("Ошибка: введите число от 1 до 5")
    sys.exit(1)

api = Api(login, password)

match variant:
    case 1:
        #Мартынов Никита 230190268
        user_id = int(input("Введите ID пользователя: "))
        print("Результат анализа репостов:",
api.task1(user_id))
    case 2:
        #252168745 Marat
        user_id = int(input("Введите ID пользователя: "))
#494676191
        cnt = int(input("Количество проверок (1-5): "))
        print("Пожалуйста, подождите, идет анализ ленты...")
        print("Результат анализа лайков:", api.task2(user_id,
min(max(cnt, 1), 5)))
    case 3:
        url = input("Введите URL поста (пример: vk.com/wall-
101982925_15779013): ")
        print("Результат анализа комментариев:",
api.task3(url))
    case 4:
        #Яна Бурыкина 386107338
        user_id = int(input("Введите ID пользователя: "))
        print("Результат анализа групп:", api.task4(user_id))
    case 5:
        api.task5()
    case _:
        print('Неизвестная операция')

import sys

from PyQt5.QtWidgets import (QApplication, QMainWindow, QWidget,
QVBoxLayout, QHBoxLayout, QLabel,

```

					ПГУ1.090501.02.001 ПЗ	Лист
						79
Изм.	Лист	№ докум.	Подпись	Дата		

```

        QComboBox, QLineEdit, QPushButton,
QTextEdit, QStackedWidget, QGroupBox,

        QSpinBox, QFileDialog, QMessageBox)

from PyQt5.QtGui import QPixmap
from PyQt5.QtCore import Qt

# Импорт API
from api import Api

class VkAnalyticsApp(QMainWindow):
    def __init__(self):
        super().__init__()
        self.api = None
        self.init_ui()

    def init_ui(self):
        """Инициализация графического интерфейса"""
        self.setWindowTitle("Аналитика Вконтакте")
        self.setGeometry(100, 100, 800, 600)

        # Главный виджет и компоновка
        main_widget = QWidget()
        main_layout = QVBoxLayout()

        # --- Секция настроек ВК ---
        api_group = QGroupBox("Настройки Вконтакте")
        api_layout = QVBoxLayout()

        # Поля для логина от ВК
        login_layout = QHBoxLayout()
        login_layout.addWidget(QLabel("Логин:"))
        self.vk_login_input = QLineEdit()
        self.vk_login_input.setPlaceholderText("Введите ваш логин
от Вконтакте")
        login_layout.addWidget(self.vk_login_input)

```

					ПГУ1.090501.02.001 ПЗ	Лист
						80
Изм.	Лист	№ докум.	Подпись	Дата		

```

# Поле для ввода пароля от ВК
vk_password_layout = QHBoxLayout()
vk_password_layout.addWidget(QLabel("Пароль:"))
self.vk_password_input = QLineEdit()
self.vk_password_input.setPlaceholderText("Введите ваш
пароль от Вконтакте")
self.vk_password_input.setEchoMode(QLineEdit.Password)
vk_password_layout.addWidget(self.vk_password_input)

api_layout.addLayout(login_layout)
api_layout.addLayout(vk_password_layout)
api_group.setLayout(api_layout)

# --- Секция выбора задачи ---
task_group = QGroupBox("Выбор задачи")
task_layout = QVBoxLayout()

# Выбор задачи
task_select_layout = QHBoxLayout()
task_select_layout.addWidget(QLabel("Выберите задачу:"))
self.task_combo = QComboBox()
self.task_combo.addItem([
    "Анализ репостов пользователя",
    "Анализ лайков в ленте",
    "Анализ комментариев к посту",
    "Анализ групп пользователя",
    "Построение аналитического отчета"
])

self.task_combo.currentIndexChanged.connect(self.update_input_fiel
ds)

task_select_layout.addWidget(self.task_combo)

task_layout.addLayout(task_select_layout)

# Стековый виджет для разных полей ввода

```

					ПГУ1.090501.02.001 ПЗ	Лист
						81
Изм.	Лист	№ докум.	Подпись	Дата		

```

self.input_stack = QStackedWidget()

# Виджет для задач 1, 2, 4 (user_id)
self.user_id_widget = QWidget()
user_id_layout = QHBoxLayout()
user_id_layout.addWidget(QLabel("ID пользователя:"))
self.user_id_input = QLineEdit()
self.user_id_input.setPlaceholderText("Например:
123456789")
user_id_layout.addWidget(self.user_id_input)
self.user_id_widget.setLayout(user_id_layout)

# Виджет для задачи 2 (дополнительный параметр)
self.likes_params_widget = QWidget()
likes_params_layout = QHBoxLayout()
likes_params_layout.addWidget(QLabel("Количество
проверок:"))
self.check_count_input = QSpinBox()
self.check_count_input.setRange(1, 5)
self.check_count_input.setValue(3)
likes_params_layout.addWidget(self.check_count_input)
self.likes_params_widget.setLayout(likes_params_layout)

# Виджет для задачи 3 (URL поста)
self.post_url_widget = QWidget()
post_url_layout = QHBoxLayout()
post_url_layout.addWidget(QLabel("URL поста:"))
self.post_url_input = QLineEdit()
self.post_url_input.setPlaceholderText("Например:
vk.com/wall-101982925_15779013")
post_url_layout.addWidget(self.post_url_input)
self.post_url_widget.setLayout(post_url_layout)

# Пустой виджет для задачи 5
self.empty_widget = QWidget()

```

					ПГУ1.090501.02.001 ПЗ	Лист
						82
Изм.	Лист	№ докум.	Подпись	Дата		

```

        # Добавление виджетов в стек
        self.input_stack.addWidget(self.user_id_widget)      #
Индекс 0 - для задач 1,4
        self.input_stack.addWidget(self.likes_params_widget) #
Индекс 1 - для задачи 2
        self.input_stack.addWidget(self.post_url_widget)     #
Индекс 2 - для задачи 3
        self.input_stack.addWidget(self.empty_widget)        #
Индекс 3 - для задачи 5

task_layout.addWidget(self.input_stack)
task_group.setLayout(task_layout)

# --- Кнопка выполнения ---
self.execute_btn = QPushButton("Выполнить")
self.execute_btn.clicked.connect(self.execute_task)

# --- Область результатов ---
result_group = QGroupBox("Результаты")
result_layout = QVBoxLayout()

self.result_stack = QStackedWidget()

# Текстовые результаты
self.text_result = QTextEdit()
self.text_result.setReadOnly(True)

# Отображение для графиков
self.image_result = QLabel()
self.image_result.setAlignment(Qt.AlignCenter)

self.result_stack.addWidget(self.text_result)  # Индекс 0
- текст
        self.result_stack.addWidget(self.image_result) # Индекс 1
- изображение

```

					ПГУ1.090501.02.001 ПЗ	Лист
						83
Изм.	Лист	№ докум.	Подпись	Дата		

```

result_layout.addWidget(self.result_stack)
result_group.setLayout(result_layout)

# --- Компоновка главного окна ---
main_layout.addWidget(api_group)
main_layout.addWidget(task_group)
main_layout.addWidget(self.execute_btn)
main_layout.addWidget(result_group)

main_widget.setLayout(main_layout)
self.setCentralWidget(main_widget)

# Инициализация полей ввода
self.update_input_fields(0)

def update_input_fields(self, index):
    """Обновляет поля ввода в зависимости от выбранной
задачи"""
    # Задачи: 0-репосты, 1-лайки, 2-комментарии, 3-группы, 4-
графики

    # Показываем соответствующий набор полей ввода
    if index in [0, 3]: # Анализ репостов или групп
        self.input_stack.setCurrentIndex(0)
    elif index == 1:    # Анализ лайков
        self.input_stack.setCurrentIndex(1)
    elif index == 2:    # Анализ комментариев
        self.input_stack.setCurrentIndex(2)
    else:               # Построение отчета
        self.input_stack.setCurrentIndex(3)

def validate_api(self):
    """Проверяет и инициализирует API с введенными токенами"""
    vk_login = self.vk_login_input.text().strip()
    vk_password = self.vk_password_input.text().strip()

```

					ПГУ1.090501.02.001 ПЗ	Лист
						84
Изм.	Лист	№ докум.	Подпись	Дата		

```

        if not vk_login or not vk_password:
            QMessageBox.warning(self, "Ошибка", "Пожалуйста,
введите корректные данные для входа")
            return False

        try:
            self.api = Api(vk_login, vk_password)
            return True
        except Exception as e:
            QMessageBox.critical(self, "Ошибка", f"Не удалось
инициализировать API: {str(e)}")
            return False

    def execute_task(self):
        """Выполняет выбранную задачу"""
        if not self.validate_api():
            return

        task_index = self.task_combo.currentIndex()

        try:
            # Задача 0: Анализ репостов
            if task_index == 0:
                user_id = int(self.user_id_input.text())
                result = self.api.task1(user_id)
                self.show_text_result("Анализ репостов:\n" +
"\n".join(result))

            # Задача 1: Анализ лайков
            elif task_index == 1:
                user_id = int(self.user_id_input.text())
                cnt = self.check_count_input.value()
                result = self.api.task2(user_id, cnt)
                self.show_text_result("Анализ лайков:\n" +
"\n".join(result))

```

					ПГУ1.090501.02.001 ПЗ	Лист
						85
Изм.	Лист	№ докум.	Подпись	Дата		

```

        # Задача 2: Анализ комментариев
        elif task_index == 2:
            post_url = self.post_url_input.text().strip()
            result = self.api.task3(post_url)
            self.show_text_result("Анализ комментариев:\n" +
result)

        # Задача 3: Анализ групп
        elif task_index == 3:
            user_id = int(self.user_id_input.text())
            result = self.api.task4(user_id)
            self.show_text_result("Анализ групп:\n" +
"\n".join(result))

        # Задача 4: Построение отчета
        elif task_index == 4:
            self.api.task5()
            self.show_image_result('vk_analytics_report.png')

    except Exception as e:
        QMessageBox.critical(self, "Ошибка выполнения",
f"Произошла ошибка: {str(e)}")

    def show_text_result(self, text):
        """Отображает текстовые результаты"""
        self.result_stack.setCurrentIndex(0)
        self.text_result.setText(text)

    def show_image_result(self, filename):
        """Отображает графический отчет"""
        self.result_stack.setCurrentIndex(1)
        pixmap = QPixmap(filename)
        self.image_result.setPixmap(pixmap.scaled(
            self.image_result.width(),
            self.image_result.height(),
            Qt.KeepAspectRatio,

```

					ПГУ1.090501.02.001 ПЗ	Лист
						86
Изм.	Лист	№ докум.	Подпись	Дата		

```

        Qt.SmoothTransformation
    ))

if __name__ == "__main__":
    app = QApplication(sys.argv)
    window = VkAnalyticsApp()
    window.show()
    sys.exit(app.exec_())

```

					ПГУ1.090501.02.001 ПЗ	Лист
						87
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение Д

(обязательное)

Приложение для анализа и обработки рентгеновских снимков

Презентация

					ПГУ1.090501.02.001 ПЗ	Лист
						88
Изм.	Лист	№ докум.	Подпись	Дата		

Анализ поведенческих реакций пользователей интернет-приложений

Выполнил: Беляков Евгений 20ВОЭ1
Руководитель: к.т.н, доцент Слепцов Н.В.

Слайд Г.1 – Тема дипломного проекта



2

Цель и теоретическая база исследования

Проект направлен на разработку модели анализа поведенческих реакций через современные методы обработки данных. Анализируются лайки, репосты, тематики групп и комментарии ВКонтакте с её разнообразной аудиторией.

Слайд Г.2 – Цель дипломного проекта

					ПГУ1.090501.02.001 ПЗ	Лист
						89
Изм.	Лист	№ докум.	Подпись	Дата		

3		
Сравнение инструментов анализа пользовательского поведения	1/ Google Analytics, Yandex.Metrica, предлагают разные функции, включая сбор данных о кликах и поведении, значимы для анализа пользовательских реакций в соцсетях.	3/ Преимущества и ограничения: одни инструменты ориентированы на веб-аналитику, другие — на визуальную аналитику, но их возможности недостаточны для социальной сети с сложными взаимодействиями.
	2/ Функционал охватывает анализ лайков, комментариев, тематик групп.	4/ Пробелы включают отсутствие интеграции анализа поведения с нейросетями, отсутствие функции имитации получения чужой ленты
...		

Слайд Г.3 – Сравнение инструментов анализа пользовательского поведения

← →

Выбор технологий для разработки аналитической системы

+



4

Для обработки данных используются Python; для нейросетевого анализа - Mistral; визуализация осуществляется через Matplotlib, что обеспечивает мощный и гибкий аналитический стек.

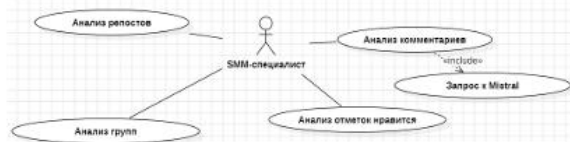
Хранение структурированных данных организовано на Sqlite3

...

Слайд Г.4 – Выбор технологий для разработки аналитической системы

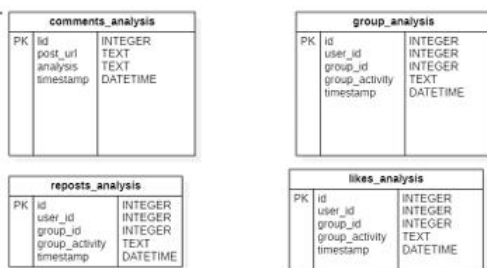
+ Диаграмма вариантов использования

аналитической системы



Слайд Г.5 – Диаграмма вариантов использования

+ Физическая модель



Слайд Г.6 – Физическая модель данных

Изм.	Лист	№ докум.	Подпись	Дата

Заключение

Реализовано приложение для анализа поведения пользователей ВК, что позволило оптимизировать работу SMM специалиста

Слайд Г.9 - Заключение

					ПГУ1.090501.02.001 ПЗ	Лист
						93
Изм.	Лист	№ докум.	Подпись	Дата		