

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное
образовательное учреждение высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической теории
упругости и биомеханики

Профилирование деятельности пользователей ВКонтакте

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы
направления 09.03.03 Прикладная информатика

механико-математического факультета

Шагина Романа Вадимовича

Научный руководитель
к.ю.н., доцент

Р. В. Амелин

Зав. кафедрой
д.ф.-м.н., профессор

Л. Ю. Коссович

Введение. Развитие компьютерных технологий и средств связи оказывает большое влияние на социум. Появляются новые его составляющие, новые виды обмена информацией и взаимодействия людей друг с другом. Это касается не только межличностных отношений, но и экономических. Хотя изначально социальные сети создавались для установления отношений между людьми с последующим их переносом из онлайн-общения в личную встречу, сейчас социальные сети не ограничиваются только знакомствами. На данный момент они предоставляют широкий спектр возможностей для получения и распространения информации: прослушивание музыки, ознакомление с новостями и погодой, покупка товаров.

Последнее фактически означает, что социальные сети представляют собой ценные площадки для реализации товаров и получения ответов на главные экономические вопросы: что производить и для кого. Сложность представляет выбор из большого количества пользователей сети тех, кто заинтересован в определённом товаре, то есть выделении целевой аудитории.

Желанием расширить клиентскую базу путём получения представительства в социальной сети ВКонтакте, а также улучшить качество обслуживания существующих руководствовались ИТ-подразделение ООО «Энди», когда производило заказ на проектирование и разработку сервиса профилирования данных пользователей. Компания ожидает получить возможность извлекать данные пользователей, в первую очередь подписчиков конкретного сообщества, выделять подгруппы среди пользователей и осуществлять рассылку сообщений и уведомлений среди её клиентов.

Конечной целью проекта является информационная система для получения, хранения и обработки открытых данных пользователей ВКонтакте определённым образом.

Для достижения поставленной цели должны быть выполнены следующие задачи:

1. Исследовать предметную область, выявить требуемую функциональность сервиса

2. Спроектировать информационную систему
3. Выбрать оптимальные технологии с точки зрения разработки, производительности и дальнейшей поддержки
4. Реализовать информационную систему в соответствии с её проектом

Структурное содержание работы:

1. Краткая характеристика социальной сети ВКонтакте
2. API ВКонтакте
3. Обоснование необходимости разработки веб-сервиса
4. Существующие аналоги
5. Проектирование базы данных
6. Архитектура приложения
7. Краткое описание используемых для реализации технологии
8. MVP
9. Регистрация и авторизация с помощью JWT-токена
10. Выгрузка данных

Основное содержание работы. Социальная сеть ВКонтакте была создана 10 октября 2006 года как социальная сеть для студентов и выпускников российских ВУЗов, но со временем пользоваться социальной сетью начали и другие пользователи. В апреле 2022 года, ежедневная аудитория сайта составила 47,2 миллиона пользователей в России, а ежемесячная аудитория составила 73,4 миллиона. Каждый день ВКонтакте посещают 52% российских пользователей интернета, а ежемесячный охват рунета достигает 84%.

ВКонтакте постоянно расширяет количество внутренних сервисов и функций социальной сети, тесно взаимодействуя как с пользователями, так и создателями контента, маркетологами и разработчиками. Для последних открыта возможность создания и интеграции сторонних приложений, а также взаимодействие с соцсетью посредством API (Application Programming Interface) — интерфейс прикладного программирования.

Для работы с API ВКонтакте необходимо передавать в запросе токен доступа (`access_token`), для получения которого используется протокол OAuth 2.0. Согласно документации, получить его можно тремя различными способами. Наиболее простым из них является так называемый `implicit flow`, который подходит для использования API с клиентской части.

Веб-сервисы — это программные системы, предоставляющие функциональность через интернет. Они позволяют доступ к данным и функциям удаленных приложений, автоматизацию бизнес-процессов и обмен данными между приложениями. В социальных сетях веб-сервисы используются для хранения и обработки данных пользователей, анализа аудитории и разработки социальных приложений. Они обладают преимуществами, такими как доступность, гибкость, масштабируемость, безопасность, экономичность и легкая интеграция, что делает их востребованными в современном информационном обществе.

Веб-сервисы как часть IT-инфраструктуры имеют ряд существенных преимуществ, наиболее важными из которых являются доступность и универсальность. Веб-сервисы обеспечивают доступ к API через стандартные

протоколы, такие как HTTP. Это означает, что любое устройство с подключением к интернету может взаимодействовать с API социальной сети через веб-сервис. Это обеспечивает универсальность и широкую доступность для различных клиентов, включая веб-приложения, мобильные приложения и другие интеграции.

Также важными критериями являются управление и масштабируемость: Веб-сервисы предоставляют удобный способ управления и масштабирования API. Они могут обрабатывать большое количество запросов от разных клиентов одновременно, обеспечивая высокую отзывчивость и производительность. Это особенно важно для социальных сетей, где активность пользователей может быть очень высокой.

Веб-сервисы способны обеспечить достаточно уровень безопасности. Они позволяют применять различные меры безопасности для защиты API и данных пользователей. Они могут использовать аутентификацию и авторизацию для контроля доступа к API и защиты конфиденциальной информации. Также могут применяться различные методы шифрования и другие техники безопасности для обеспечения защиты данных.

Немаловажным фактором считается гибкость интеграции: Веб-сервисы позволяют легко интегрировать API социальной сети с другими системами и сервисами. Например, веб-сервисы могут использоваться для рассылки уведомлений через различные каналы, такие как электронная почта, SMS или пуш-уведомления. Они также могут обеспечивать функциональность профилирования деятельности пользователей, анализа данных и создания персонализированных рекомендаций.

Таким образом, именно веб-сервис, предоставляющий свой API, наилучшим образом решает проблемы компании «Энди», связанные со сложными соединениями сущностей, рассылкой промо-информации и уведомлений, а также гибкой выгрузки данных в требуемом формате.

Существуют платные аналоги подобного сервиса, но часть из них не в состоянии удовлетворить всех потребностей заказчика, а другие слишком

дороги либо неудобны в работе, поскольку не адаптируемы к уже существующим сервисам и инфраструктуре заказчика.

Проектирование информационной системы начинается с анализа предметной области и представления её в виде сущности базы данных с помощью ER-диаграммы (Entity-Relationship diagram, диаграмма сущность-связь) — графической модели данных, используемой в области проектирования баз данных для описания схемы базы данных. ER-диаграмма позволяет визуализировать структуру базы данных, определить связи между сущностями и оптимизировать структуру базы данных, чтобы она соответствовала требованиям бизнеса и была максимально эффективной. Кроме того, ER-диаграмма может использоваться в качестве инструмента для коммуникации между разработчиками, заказчиками и другими участниками проекта, так как она позволяет представить структуру базы данных в понятном и доступном виде.

Всего выделено 11 основных сущностей: Пользователь, Группа, Тип пользователя, Запись, Комментарий, Отношения пользователь-пользователь, Рассылка, Продукт, Заказ, Клиент, Компания.

Запись представляет собой отдельный пост, который пользователи могут «лайкать», делиться с другими и комментировать. В процессе анализа предметной области выяснилось, что отдельные лайки и распространение записей не представляют интереса сами по себе, поэтому они хранятся только в численно значении, а не в виде отдельных сущностей.

Информация о комментарии и его авторе, напротив, могут представлять интерес, поэтому выделены в сущность.

Отношение пользователь-пользователь отражает направление подписки пользователя на другого пользователя или группу.

Сущность Рассылка обеспечивает хранение уведомлений, которые были и должны быть разосланы пользователям через некоторое время.

Продукт вполне очевидным образом представляет собой отдельный продукт или логически цельный набор продуктов компании, отдельную позицию.

Назначение Заказа и Клиента вполне очевидно по их названию.

Сущность Компания предназначена для хранения данных компании, берущую на себя обязательства по обслуживанию конкретного клиента.

В качестве СУБД выбрана PostgreSQL из-за широкого набора предоставляемых ею возможностей, высокую производительность, а также того, что она является свободным программным обеспечением.

Существует два основных подхода при создании веб-сервисов: SOAP и REST.

SOAP (Simple Object Access Protocol) — это протокол обмена сообщениями на основе XML, который предоставляет стандартный способ обмена данными между различными системами. Он обеспечивает защиту данных, поддержку транзакций и обработку ошибок, но является более сложным в использовании и имеет больший объем передаваемых данных. SOAP использует структурированные сообщения, передаваемые через различные транспортные протоколы.

REST (Representational State Transfer) — это архитектурный стиль, использующий протокол HTTP для передачи данных. Он представляет ресурсы в виде URI и выполняет операции над ними с помощью HTTP-методов. REST-сервисы просты в использовании, легковесны, масштабируемы и поддерживают кеширование. Они обеспечивают безопасное взаимодействие и могут быть использованы для обмена данными между приложениями и системами, а также для создания различных типов приложений.

Основные различия между SOAP и REST заключаются в формализации обмена сообщениями, использовании WSDL и протоколах передачи данных. SOAP предоставляет более строгую модель с большей надежностью и безопасностью, в то время как REST имеет более гибкий и простой интерфейс.

REST получил широкую поддержку и популярность, а множество разработчиков уже знакомы с его интерфейсами и форматами данных.

Из-за своих преимуществ REST-архитектура выбирается для многих приложений. Она основана на простых принципах, поддерживает масштабируемость и распределённую архитектуру, а также легко интегрируется с различными языками программирования и фреймворками. SOAP, в свою очередь, более тяжеловесен и требует сложной конфигурации, поэтому предпочтении было отдано REST-архитектуре.

Для реализации веб-сервиса в качестве языка программирования выбран Python версии 3.10, FastAPI в качестве фреймворка для разработки API, библиотека Pydantic для валидации и обмена данными и ORM для упрощения работы с базой данных SQLAlchemy.

Объёмные части программного кода выделяются в отдельные модули по принципу логической связанности. В пакете `api` помещаются все эндпоинты (маршруты приложения), которые делегируют бизнес-логику в пакет `services`, в котором описывается внутренняя логика запросов данных и их обработка. В пакете `models` вынесены Pydantic-модели, а в пакет `tables` классы SQLAlchemy для работы с таблицами базы данных.

Все зависимости вместе с конкретными версиями библиотек записываются в файл `requirements.txt` в формате `название_библиотеки==номер_версии`.

В файл `.env` сохраняются настройки веб-сервиса, которые нежелательно хранить вместе с исходным кодом приложения, например, секретный ключ.

Для авторизации пользователей сервиса в целях контроля и разграничения доступа к различным функциям сервиса используется JSON Web Token или JWT-токен. Он представляет собой JSON объект, состоящий из трёх частей: заголовка (`header`) с общей информацией по токenu, полезные данные (`payload`) с такой информацией, как ID пользователя и т.д., и подписи (`signature`).

В качестве алгоритма хэширования используется HS256, время валидности токена 1 час.

При регистрации пользователей пароли хэшируются по алгоритму bcrypt из библиотеки passlib.hash.

В файле settings.py прописываются основные настройки веб-сервиса, как то: адрес хоста, номер порта, путь к базе данных, а также параметры JWT-токена (секретный ключ, алгоритм шифрования, срок действия токена).

Также имеется механизм выгрузки данных из сервиса в формате CSV и JSON.

Экспорт данных в формате CSV реализован с помощью стандартной библиотеки csv, а именно классов DictReader и DictWriter. При экспорте используется класс StreamingResponse fastapi.responses. Этот класс принимает асинхронный или обычный генератор и передаёт тело ответа. Он подходит для файлоподобных объектов, поскольку не хранит его целиком в памяти, загружая строки по мере необходимости.

Для импорта используется встроенный механизм FastAPI для обработки фоновых задач. Это полезно для таких операций, которые должны выполняться после запроса, но при этом клиенту не обязательно ждать завершения операции перед получением ответа. Например, при получении файла, обработка которого занимает длительное время, можно вернуть ответ с кодом 202 Accepted и обработать его в фоновом режиме. Для этого существует класс BackgroundTasks, в который через метод add_task передаётся метод класса и его аргументы.

Заключение. Главный результат представленной работы заключается в проектировании и реализации веб-сервиса, удовлетворяющего потребностям заказчика в получении, хранении и обработке данных пользователей социальной сети ВКонтакте. Обработка данных включает в себя как простые операции получения данных, так и сложные соединения, такие как поиск общих групп для нескольких пользователей.

Наряду с основным результатом, в рамках выпускной работы рассмотрены примеры работы и основные особенности таких технологий, как FastAPI, Pydantic и SQLAlchemy, регистрация пользователей и их авторизация посредством JWT-токенов, а также вспомогательные механизмы FastAPI для работы с фоновыми задачами.