|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

**ТЕХНИЧЕСКИЙ ПРОЕКТ**

на разработку

«Telegram-бот для просмотра информации о рыночных активах»

2024

**Содержание**

[1. Требования к системе 2](#_Toc181420795)

[1.1. Функциональные требования 2](#_Toc181420796)

[1.2. Нефункциональными требованиями к ИС являются: 4](#_Toc181420797)

[1.3. Интерфейсные требования 6](#_Toc181420798)

[1.3.1. Пользовательский интерфейс и взаимодействие 6](#_Toc181420799)

[1.3.2. Поддержка мультиязычности 6](#_Toc181420800)

[2. Уточнение структуры данных 7](#_Toc181420801)

[2.1. Выделение сущностей 7](#_Toc181420802)

[2.2. Построение логической ER-диаграммы 8](#_Toc181420803)

[2.3. Построение диаграммы потоков данных 8](#_Toc181420804)

[3. Укрупненный сценарий использования 11](#_Toc181420805)

[4. Диаграмма вариантов использования 12](#_Toc181420806)

[4.1. Просмотр и поиск активов 12](#_Toc181420807)

[4.2. Просмотр информации об активе 12](#_Toc181420808)

[4.3. Подписка на уведомления об изменении цены 13](#_Toc181420809)

[4.4. Управление избранными активами 13](#_Toc181420810)

[4.5. Настройка параметров бота 14](#_Toc181420811)

[5. Уточнение структуры системы 15](#_Toc181420812)

[6. Пользовательские интерфейсы 17](#_Toc181420813)

[7. Определение языка разработки 20](#_Toc181420814)

[8. Определение системы управления базами данных 22](#_Toc181420815)

[9. Определение библиотеки для создания пользовательского интерфейса 23](#_Toc181420816)

1. Требования к системе

Основной процесс в виде диаграммы IDEF0 показан на рисунке 1.

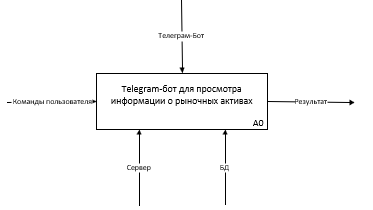


Рисунок 1 – Диаграмма IDEF0

* 1. Функциональные требования

Telegram-бот должен предоставлять возможность:

* Приветствие и предоставление информации:

Бот должен приветствовать пользователя при первой активации и предоставлять информацию о доступных функциях.

* Просмотр и поиск активов:

Пользователь может просматривать список доступных активов (акции, валюты, криптовалюты) с возможностью поиска и фильтрации по различным параметрам (цена, объем и т.д.).

* Просмотр текущей информации об активе:

Пользователь может запрашивать актуальную информацию об активе, включая цену, объем торгов, рыночную капитализацию и изменение цены за выбранный период.

* Просмотр графиков актива:

Пользователь может получать графики изменения цены актива за различные периоды (час, день, неделя, месяц, год).

* Управление избранными активами:

Пользователь может добавлять активы в избранное или удалять их для быстрого доступа.

* Подписка на уведомления об изменении цены:

Пользователь может подписываться на уведомления при изменении цены актива на определенный процент или получать периодические оповещения.

* Управление параметрами бота:

Пользователь может настраивать параметры бота, такие как часовой пояс, базовая валюта и частота уведомлений.

* Получение помощи и справки:

Пользователь может запросить справочную информацию о доступных командах бота или связаться с поддержкой.

* Уведомления о достижении ценового уровня:

Пользователь может настраивать уведомления при достижении определенной цены актива.

* Отмена подписок и уведомлений:

Пользователь может отменять или изменять параметры активных подписок и уведомлений.

* Проверка корректности данных при вводе:

Бот проверяет корректность введенных данных (цена, объем, тикер) перед выполнением операций.

* Формирование графиков для аналитики:

Система создает графики или диаграммы для анализа данных по активам за выбранные периоды и предоставляет их в формате PNG или PDF.

Пользователю должны предоставляться следующие возможности:

* Поиск и фильтрация активов:
  + Возможность искать активы по названию или тикеру.
  + Фильтрация активов по различным параметрам (цена, объем, изменения и т.д.).
* Получение подробной информации:
  + Доступ к детальной информации об активе, включая исторические данные.
* Управление подписками:
  + Настройка и управление подписками на уведомления.
* Персонализация настроек:
  + Изменение настроек бота для удобного отображения информации и уведомлений.

## Нефункциональными требованиями к ИС являются:

Требования к производительности:

* + Бот должен отвечать на команды пользователей в течение 1-3 секунд. Это включает обработку команды и отправку сообщения в ответ.

Требования к надежности:

* + Telegram-бот должен работать стабильно и без сбоев.
  + Сервер, на котором размещен бот, должен обеспечивать бесперебойное электропитание и стабильное подключение к интернету.

Требования к безопасности:

* + Защита пользовательских данных и предотвращение несанкционированного доступа.
  + Шифрование данных при передаче между ботом и базой данных.

Требования к масштабируемости:

* + Возможность масштабирования системы для обработки увеличивающегося числа пользователей и запросов без существенного ухудшения производительности.

Требования к удобству использования:

* + Интуитивно понятный интерфейс с четкими и понятными сообщениями.
  + Логичная структура меню и навигации.

1.3. **Интерфейсные требования**

1.3.1. Пользовательский интерфейс и взаимодействие

Интерфейс должен быть удобным для пользователей, интуитивно понятным и иметь четкую структуру. Все сообщения, которые выдает бот, должны быть понятными и лаконичными. Интерфейс включает в себя:

* Чат с ботом:

Центральная часть экрана занимает чат, в котором происходит взаимодействие между пользователем и ботом.

* Кнопки и меню:

Кнопки для быстрого доступа к основным функциям бота, расположенные снизу чата.

* Ответы и уведомления:

Текстовые сообщения, графики и другие медиафайлы, предоставляемые ботом.

1.3.2. Поддержка мультиязычности

Интерфейс должен поддерживать только русский язык, мультиязычность не предусмотрена.

1. Уточнение структуры данных

В ходе уточнения данных были выполнены следующие задачи, показанные в Таблице 1.

Таблица 1 – Задачи для уточнения структуры данных.

|  |  |  |  |
| --- | --- | --- | --- |
| **№ задачи** | **Название** | **Содержание** | **Результат** |
| 1 | Определение атрибутов сущностей | Определение и описание атрибутов для каждой сущности, таких как типы данных и ограничения | Таблица с атрибутами сущностей и их описанием |
| 2 | Представление атрибутов сущностей | Создание таблицы с атрибутами для каждой сущности | Готовая таблица с атрибутами сущностей |
| 3 | Построение диаграммы потоков данных | Определить, как будут передвигаться данные в программе; построить DFD диаграмму | Готовая диаграмма потоков данных (DFD) |

* 1. Выделение сущностей

Таблица с атрибутами и описанием сущностей показано в Таблице 2.

Таблица 2 – Сущности и их описание

|  |  |  |
| --- | --- | --- |
| № п/п | Название сущности | Описание сущности |
| 1 | Пользователь | Лица, взаимодействующие с ботом для просмотра активов, подписок и настроек. |
| 2 | Избранные активы | Список активов, добавленных пользователем в избранное. |
| 3 | Подписки | Информация о подписках пользователя на уведомления об изменениях цен. |

* 1. Построение логической ER-диаграммы

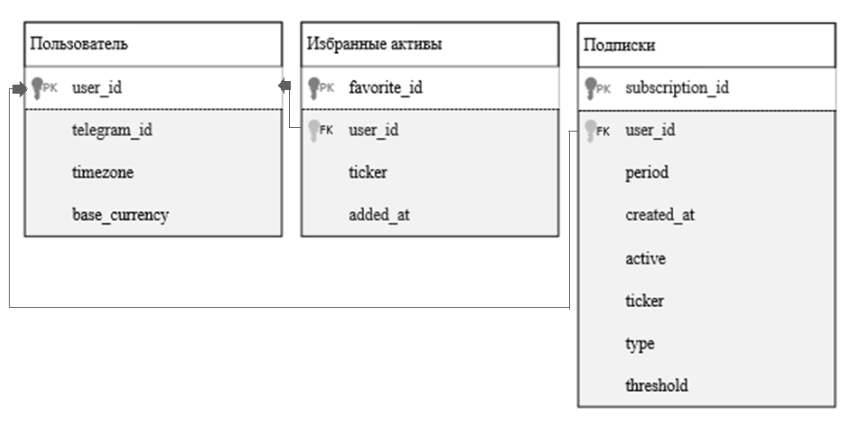
Логическая ER-диаграмма показана на рисунке 2. 

Рисунок 2 – Таблицы

* 1. Построение диаграммы потоков данных

DFD (Data Flow Diagram) – это графическое представление потоков данных в системе, показывающее, как информация перемещается через различные процессы. DFD используется для моделирования и анализа информационных систем, позволяя представить их структуру, функции и потоки данных между различными компонентами.

На рисунке 3 представлена DFD-диаграмма Телеграмм-бота.

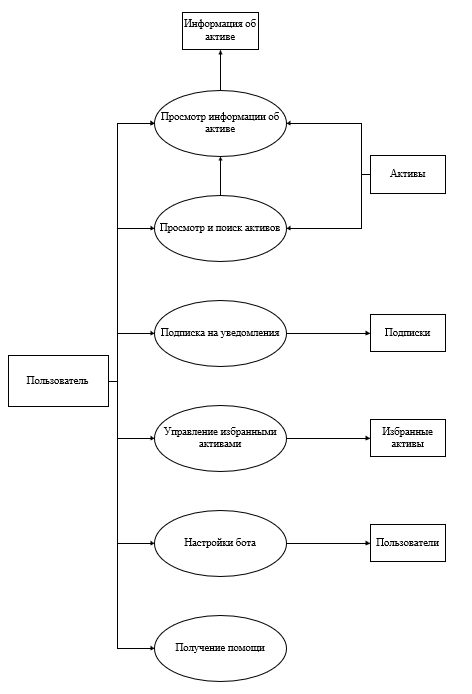


Рисунок 3 – DFD диаграмма'

На данной диаграмме присутствуют следующие блоки работы:

* Просмотр и поиск активов – процесс, в котором пользователь запрашивает список активов или ищет конкретный актив.
* Просмотр информации об активе – процесс предоставления детальной информации об активе.
* Подписка на уведомления – процесс настройки уведомлений о изменениях цены или достижении ценового уровня.
* Управление избранными активами – процесс добавления или удаления активов из избранного.
* Настройки бота – процесс изменения параметров работы бота.
* Получение помощи – процесс предоставления справочной информации пользователю.

Также имеются хранилища данных:

* Пользователи – хранение информации о пользователях.
* Активы – хранение информации о доступных активах.
* Избранные активы – хранение информации об избранных активах пользователей.
* Подписки – хранение информации о подписках пользователей на уведомления.

1. **Архитектура системы**
   1. **Выбранный архитектурный стиль**

Для разработки Telegram-бота выбран клиент-серверный архитектурный стиль. Этот выбор обусловлен следующими причинами:

* Разделение обязанностей: Клиентская часть (бот) отвечает за взаимодействие с пользователем, а серверная часть обрабатывает запросы, управляет данными и взаимодействует с внешними сервисами.
* Масштабируемость: Клиент-серверная архитектура позволяет легко масштабировать серверную часть при увеличении числа пользователей.
* Безопасность: Разделение клиентской и серверной частей упрощает реализацию механизмов безопасности и контроля доступа.
  1. **Компоненты системы**

Основные компоненты системы:

1. Telegram-бот (Клиентская часть):
   * Отправка и получение сообщений.
   * Обработка пользовательских команд и взаимодействий.
   * Отображение информации и графиков.
2. Серверная часть:
   * Обработка запросов от бота.
   * Взаимодействие с базой данных.
   * Интеграция с внешними API для получения финансовых данных.
   * Управление подписками и уведомлениями.
3. База данных (PostgreSQL):
   * Хранение информации о пользователях, избранных активах и подписках.
   * Обеспечение целостности и безопасности данных.
4. Внешние сервисы:
   * API для получения актуальных данных о рыночных активах (например, Yahoo Finance).
   1. **Диаграмма компонентов**

*Описание диаграммы компонентов:*

* **Telegram-бот** взаимодействует с **Серверной частью** через API.
* **Серверная часть** обращается к **Базе данных** для хранения и получения информации.
* **Серверная часть** также взаимодействует с **Внешними сервисами** для получения актуальных данных о рыночных активах.
* **База данных** обеспечивает хранение данных пользователей, избранных активов и подписок.

1. Укрупненный сценарий использования

Пользователь запускает Telegram-бота, отправляя команду `/start`. Бот приветствует пользователя и предлагает три основных действия: «Просмотр активов», «Избранное» и «Настройки». Пользователь выбирает опцию «Просмотр активов», после чего бот отображает список доступных категорий активов, таких как акции, валюты и криптовалюты. Пользователь выбирает одну из категорий, например, «Криптовалюты», и бот выводит список популярных активов этой категории с краткой информацией о каждом, включая текущую цену и изменения за последние 24 часа. В конце списка бот предлагает дополнительные опции, такие как «Поиск по названию или тикеру» и «Обновить список». Если пользователь выбирает поиск, бот запрашивает название или тикер актива, после чего отображает результаты, соответствующие запросу. Пользователь может выбрать актив для получения более подробной информации, включая цену, объем торгов, рыночную капитализацию, а также максимум и минимум за последние 24 часа.

Просматривая информацию об активе, пользователь может выбрать опцию «Посмотреть график». Бот предложит выбрать период графика — день, неделю, месяц или год, — после чего сгенерирует и отправит график изменения цены актива за выбранный период. После просмотра графика пользователь может изменить период, подписаться на уведомления об изменении цены или добавить актив в избранное для быстрого доступа. Если пользователь решает подписаться на уведомления, бот предложит два типа уведомлений: при изменении цены на определенный процент или периодические оповещения. Пользователь может указать процент изменения цены или выбрать периодичность оповещений, после чего бот подтвердит настройку подписки и сохранит её в базе данных. При достижении заданных условий бот отправит уведомление с актуальной информацией.

Пользователь также может управлять списком избранных активов, выбирая их из подробной информации об активе или открывая раздел «Избранное» в главном меню. В этом разделе бот отображает список всех активов, добавленных в избранное, с краткой информацией, и позволяет быстро перейти к подробной информации об активах или удалить их из избранного. В разделе «Настройки» пользователь может изменить основные параметры бота, такие как часовой пояс, базовую валюту для отображения финансовых данных и частоту уведомлений. В процессе взаимодействия пользователь может в любой момент воспользоваться командой `/help`, чтобы получить справочную информацию о доступных командах и возможностях бота.

Таким образом, Telegram-бот обеспечивает удобный доступ к информации об активах, настройке уведомлений и управлению избранным, предоставляя пользователю возможность быстро получать актуальные данные и персонализировать взаимодействие с ботом.

1. Диаграмма вариантов использования

## 4.1. Просмотр и поиск активов

**Описание:** Пользователь может просматривать список доступных активов и искать конкретные активы по названию или тикеру.

**Предварительные условия:** Наличие активного подключения к боту и доступ к базе данных для получения информации об активах.

**Нормальное направление:**

* Пользователь запускает бота, отправляя команду /start.
* Бот приветствует пользователя и предлагает главное меню.
* Пользователь выбирает "Просмотр активов".
* Бот предлагает выбрать категорию активов.
* Пользователь выбирает категорию (например, "Криптовалюты").
* Бот отображает список активов в выбранной категории.
* Пользователь может искать актив по названию или тикеру.
* Бот выводит результаты поиска.
* Пользователь выбирает актив для просмотра подробной информации.

**Выходные условия:** Пользователь получил список активов или результаты поиска, возвращается в главное меню.

## 4.2. Просмотр информации об активе

**Описание:** Пользователь может получать детальную информацию об активе, включая текущую цену, объем торгов и рыночную капитализацию.

**Предварительные условия:** Пользователь выбрал актив для просмотра подробной информации.

**Нормальное направление:**

* Пользователь выбирает актив из списка.
* Бот предоставляет детальную информацию об активе.
* Пользователь может выбрать дополнительные опции: посмотреть график, подписаться на уведомления, добавить в избранное или вернуться назад.

**Выходные условия:** Пользователь получил подробную информацию об активе или перешел к дополнительным опциям.

## 4.3. Подписка на уведомления об изменении цены

**Описание:** Пользователь может настраивать уведомления о изменении цены актива на определенный процент или получать периодические оповещения.

**Предварительные условия:** Пользователь находится в меню информации об активе.

**Нормальное направление:**

* Пользователь выбирает "Подписаться на уведомления".
* Бот предлагает выбрать тип уведомления.
* Пользователь выбирает тип уведомления и настраивает параметры (например, процент изменения цены).
* Бот подтверждает настройку подписки.
* Подписка сохраняется в базе данных.

**Выходные условия:** Пользователь подписался на уведомления, бот подтвердил настройку.

## 4.4. Управление избранными активами

**Описание:** Пользователь может добавлять активы в избранное для быстрого доступа и управления ими.

**Предварительные условия:** Пользователь находится в меню информации об активе.

**Нормальное направление:**

* Пользователь выбирает "Добавить в избранное".
* Бот подтверждает добавление актива в избранное.
* Пользователь может просматривать список избранных активов через главное меню.
* Пользователь может удалять активы из избранного.

**Выходные условия:** Актив добавлен или удален из избранного, пользователь может просматривать список избранных активов.

## 4.5. Настройка параметров бота

**Описание:** Пользователь может изменять настройки бота, такие как часовой пояс, базовая валюта и частота уведомлений.

**Предварительные условия:** Пользователь находится в меню настроек.

**Нормальное направление:**

* Пользователь выбирает "Настройки" в главном меню.
* Бот отображает доступные параметры для настройки.
* Пользователь выбирает параметр для изменения (например, часовой пояс).
* Бот предлагает варианты выбора и пользователь делает выбор.
* Бот сохраняет изменения и подтверждает настройку.

**Выходные условия:** Настройки бота обновлены, пользователь возвращается в главное меню.

.

1. Уточнение структуры системы

При уточнении структуры системы была описана подробная информация об основных объектах с их описанием, показана в Таблице 3.

Таблица 3 – Информация об объектах

|  |  |  |  |
| --- | --- | --- | --- |
| Сущность | Атрибут | Тип данных | Описание |
| Пользователь | user\_id | INTEGER | Уникальный идентификатор пользователя (первичный ключ). |
|  | telegram\_id | BIGINT | Telegram ID пользователя (уникальный). |
|  | timezone | VARCHAR(10) | Часовой пояс пользователя (например, 'UTC+3'). |
|  | base\_currency | VARCHAR(3) | Базовая валюта пользователя (например, 'USD', 'EUR'). |
| Избранные активы | favorite\_id | INTEGER | Уникальный идентификатор записи (первичный ключ). |
|  | user\_id | INTEGER | Идентификатор пользователя (внешний ключ). |
|  | ticker | VARCHAR(10) | Тикер избранного актива (например, 'BTC', 'ADA'). |
|  | added\_at | TIMESTAMP | Время добавления актива в избранное. |
| Подписки | subscription\_id | INTEGER | Уникальный идентификатор подписки (первичный ключ). |
|  | user\_id | INTEGER | Идентификатор пользователя (внешний ключ). |
|  | ticker | VARCHAR(10) | Тикер актива для подписки. |
|  | type | VARCHAR(20) | Тип подписки (например, 'price\_change', 'periodic'). |
|  | threshold | DECIMAL(10,2) | Пороговое значение (например, процент изменения цены). |
|  | period | VARCHAR(10) | Период уведомлений (например, 'hour', 'day'). |
|  | created\_at | TIMESTAMP | Дата создания подписки. |
|  | active | BOOLEAN | Статус подписки (активна или нет). |

1. Пользовательские интерфейсы

Интерфейс Telegram-бота для управления активами включает в себя несколько функциональных элементов:

Главное меню

После запуска бота (/start) отображается главное меню с крупными кнопками:

1. Просмотр активов — Переход в раздел, где можно просмотреть и искать активы.
2. Избранное — Быстрый доступ к избранным активам.
3. Настройки — Открывает параметры бота.

Раздел «Просмотр активов»

В разделе «Просмотр активов» доступны следующие кнопки и команды:

* Категории активов:
* Акции
* Валюты
* Криптовалюты
* Функции для работы с выбранной категорией:
  + - Поиск по названию или тикеру — Запрашивает ввод для поиска конкретного актива.
    - Обновить список — Обновляет данные о выбранных активах.
    - Назад — Возвращает в главное меню.

Раздел «Информация об активе»

В этом разделе можно получить подробные сведения по каждому активу, доступные через команды и кнопки:

* /info [тикер] — Показывает текущие данные об активе, с доступом к следующим функциям:
  + - * Посмотреть график — Отображает график изменения цены актива.
      * Подписаться на уведомления — Настройка уведомлений о изменении цены актива.
      * Добавить в избранное — Добавляет актив в список избранного.
      * Назад — Возвращает в предыдущее меню.

Раздел «Просмотр графиков»

Для получения графика по активу:

* /chart [тикер] — Команда для показа графика.
* Выбор периода графика:
  + - За 1 день
    - За 1 неделю
    - За 1 месяц
    - За 1 год
* Дополнительные опции:
  + - Изменить период — Позволяет изменить временной промежуток.
    - Подписаться на уведомления — Настройка уведомлений по изменению цен.
    - Добавить в избранное — Добавляет актив в избранное.
    - Назад — Возвращает к информации об активе.

Раздел «Подписки и уведомления»

Настройка уведомлений:

* /subscribe [тикер] — Подписка на уведомления по активу.
* Тип уведомлений:
  + - При изменении цены на определенный процент
    - Периодические уведомления
* Настройка параметров уведомления:
  + Процент изменения цены — Ввод процента для активации уведомления.
* Подтверждение подписки:
  + - Отменить подписку
    - Просмотреть мои подписки
    - Назад
* /alert [тикер] — Устанавливает ценовое уведомление.

Раздел «Избранное»

* /favorites — Отображает список избранных активов.
* Удаление из избранного:
  + Удалить из избранного — Кнопка удаления актива из списка избранного.

Раздел «Настройки»

Основные параметры настройки:

* Часовой пояс — Автоматическое определение или ручной выбор.
* Базовая валюта — Выбор среди валют USD, EUR, RUB и т. д.
* Управление уведомлениями — Настройка и управление подписками на уведомления.

1. Определение языка разработки

Для разработки Telegram-бота и серверной части будет использоваться Python — высокоуровневый, универсальный язык, подходящий как для работы с финансовыми данными, так и для создания ботов и веб-приложений. Python выбран за его простоту, гибкость и широкий выбор библиотек для работы с API, анализа данных и взаимодействия с базами данных.

Основной стек технологий:

1. Telegram-бот:
   * aiogram — библиотека для создания асинхронных Telegram-ботов на Python. Aiogram поддерживает все возможности Telegram API и позволяет легко обрабатывать команды, кнопки и взаимодействия с пользователем в режиме реального времени.
2. Серверная часть:
   * yfinance — библиотека для загрузки финансовых данных из Yahoo Finance. Подходит для получения исторических данных о стоимости акций, валют и криптовалют.
   * investpy и investiny — библиотеки для работы с данными с сайта Investing.com, которые помогают получать актуальную информацию о различных активах, таких как акции, индексы, валюты и криптовалюты.
3. База данных:
   * SQLAlchemy — ORM для работы с базами данных в Python, предоставляющая удобный способ взаимодействия с PostgreSQL и упрощающая создание, чтение, обновление и удаление данных.
   * PostgreSQL — реляционная СУБД для надежного хранения и обработки данных. PostgreSQL выбрана за ее надежность, поддержку транзакций и масштабируемость.

Почему Python?

1. Гибкость и универсальность: Python позволяет легко интегрировать бота, сервер и базу данных в единое приложение, что ускоряет разработку и упрощает поддержку.
2. Асинхронное программирование: Использование aiogram и других асинхронных библиотек позволяет обрабатывать несколько запросов одновременно, что улучшает производительность бота.
3. Богатая библиотека для работы с данными и API: Python предлагает множество готовых решений, таких как yfinance, investpy и investiny, которые упрощают доступ к финансовым данным в режиме реального времени.
4. Простота разработки и сопровождения: Лаконичный синтаксис Python ускоряет процесс создания прототипа и делает приложение легко поддерживаемым.

Таким образом, Python и указанные библиотеки обеспечивают надежное, масштабируемое и гибкое решение для создания Telegram-бота, позволяющего пользователю удобно просматривать и отслеживать данные о рыночных активах.

.

.

1. Определение системы управления базами данных

Для хранения данных в Telegram-боте выбрана PostgreSQL — мощная реляционная СУБД, поддерживающая сложные запросы и хорошо подходящая для работы с большими объемами информации и многопользовательскими приложениями.

Почему PostgreSQL?

* Надежность и устойчивость: PostgreSQL обеспечивает высокую стабильность, что важно для работы бота без сбоев.
* Поддержка ACID-транзакций: Транзакционная обработка данных обеспечивает их целостность.
* Масштабируемость: Поддержка масштабирования позволяет адаптировать базу данных при росте числа пользователей.
* Безопасность: Поддержка шифрования и контроля доступа защищает данные пользователей.
* Гибкость в моделировании данных: Структура базы данных легко изменяется при изменении требований.

Интеграция с Python:

PostgreSQL отлично сочетается с Python через библиотеку SQLAlchemy, что упрощает обработку запросов к базе данных.

Инструменты для работы с PostgreSQL:

* pgAdmin — визуальный инструмент для администрирования базы данных.
* DBeaver — универсальный инструмент, который поддерживает работу с различными СУБД, включая PostgreSQL.

1. Определение библиотеки для создания пользовательского интерфейса

Для разработки пользовательского интерфейса Telegram-бота будет использоваться aiogram — мощная библиотека для создания ботов на Python, поддерживающая асинхронное программирование.

Основные преимущества aiogram:

* Простота использования: aiogram обеспечивает лаконичный синтаксис, что упрощает разработку.
* Асинхронное программирование: Поддержка асинхронности позволяет обрабатывать множество запросов одновременно, улучшая производительность.
* Полная поддержка Telegram API: Библиотека поддерживает все функции Telegram, включая отправку сообщений, управление кнопками и команды.
* Активное сообщество: Широкое сообщество пользователей aiogram помогает решать возникающие вопросы и обеспечивает развитие библиотеки.

Использование aiogram позволяет создать Telegram-бот с богатым функционалом для работы с активами, предоставляя пользователям гибкий и удобный инструмент для получения информации.