**Техническое задание**

**на реализацию системы отслеживания котировок криптовалют**

**Введение**

Реализовать систему, которая будет отслеживать курс криптовалют через Telegram-бота.

Список отслеживаемых криптовалют представлен в Таблице 1.

`figi` – единый глобальный идентификатор ценной бумаги (является уникальным для каждой криптовалюты).

Таблица 1. Список отслеживаемых ботом криптовалют:

|  |  |  |
| --- | --- | --- |
| **Наименование криптовалюты** | **Код криптовалюты** | **Идентификатор `figi`** |
| Bitcoin | BTC | "BTC" |
| Ethereum | ETH | "ETH" |
| Binance Coin | BNB | "BNB" |
| Dogecoin | DOGE | "DOGE" |
| Dotcoin | DOT | "DOT" |
| Cardano | ADA | "ADA" |

**Цели создания системы:**

1. Создание продукта с унифицированным доступом к котировкам криптовалют;
2. Формирование клиентской базы системы;
3. Автоматизация процесса слежения за котировками криптовалют;
4. Повышение клиентской базы системы;

**Задачи:**

1. Для предоставления пользователю информации о котировках криптовалют, система (Telegram-бот) должна поддерживать следующие команды:

1. Подписка на стрим котировок криптовалют;
2. Отписка от стрима котировок криптовалют;
3. Подписка на изменения курса валюты от текущего значения.

2. С целью привлечения новых пользователей и повышения конкурентной способности, в системе должна быть реализована система контроля метрик:

1. Число активных триггеров;
2. Число ошибок;
3. Курс конкретной криптовалюты.

Это также позволит измерять эффективность развития и корректность работы системы на определенных этапах.

3. Для повышения стабильности работы системы и предотвращения некорректной работы, необходимо реализовать:

1. Тестирование телеграмм бота;
2. Проверку команд на выдачу информации по курсу криптовалют;
3. Проверку корректной работы мониторинга с указанным интервалом времени.

4. Для оперативного контроля, визуализации и мониторинга за работой системы, необходимо реализовать:

- поддержку визуализации временных рядов и показателей системы (дашборд в grafana);

- поддержку обмена сообщениями по различным событиям в системе(интегрировать slack, zabbix и prometheus).

**Технические решения:**

Система (Telegram-бот) должна поддерживать следующие команды:

**1. Подписка на стрим (/addsubscr).**

Бот должен принимать от пользователя идентификатор `figi` валюты, за котировками которой хочет следить пользователь. В результате, бот каждые 3 секунды должен отправлять пользователю сообщение с данными о текущей котировке, запрашиваемой криптовалюты.

**2. Отписка от стрима (/delsubscr).**

При получении сообщения в формате “Stop\_`figi`”, бот должен прекратить отправку данных пользователю о значениях курса валюты `figi`.

При получении сообщения в формате “Stop\_all”, бот должен прекратить отправку данных пользователю о значениях котировок всех запрашиваемых ранее криптовалют.

**3. Подписка на события вида "падение курса валюты `figi` от текущего на X%" (/eventsubscr).**

При получении подобного вида сообщения, бот должен начать следить за курсом `figi` и при необходимости уведомлять пользователя о событии.

Также бот должен поддерживать сервисные команды. Сообщая сервисную информацию о своей работе и список поддерживаемых стандартных команд для работы.

4. Создать **механизм для обработки данных** криптовалют и передать обработанные данные в хранилище для бота (реализовать Pipeline).

**Требования к системе:**

1. Разрабатываемая Система (Telegram-бот) должна соответствовать принципу расширяемости, т.е. иметь возможность наращивания своей функциональности.
2. Разрабатываемая система должна работать с мобильными и десктопными устройствами.
3. Интерфейс пользователя должен быть мультиязычным, и поддерживать два языка: русский и английский.
4. Система не может первым начинать общение с пользователем (первичное общение, рассылка спама должна быть запрещена Системой).
5. Система должна стабильно работать при 100 одновременно подключенных пользователей.

**Модель данных**

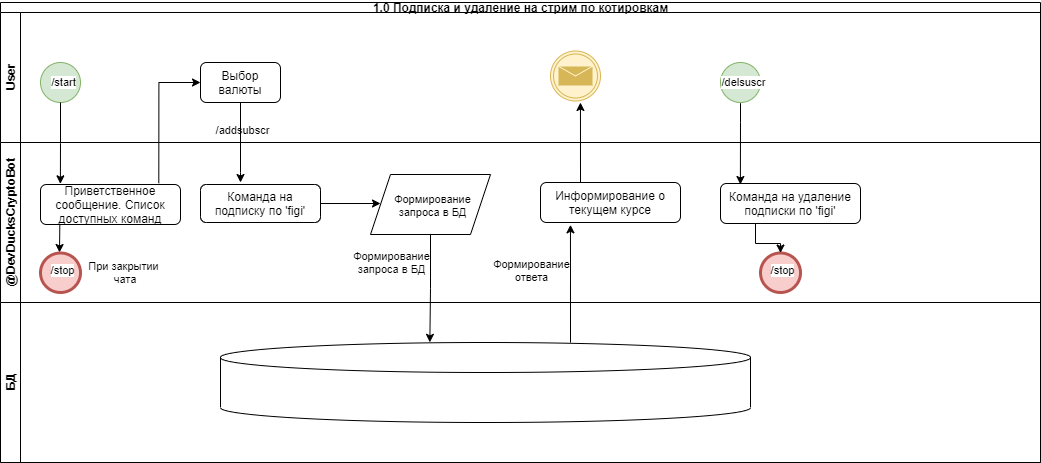
На Диаграммах показаны основные сущности системы и их взаимосвязь между собой. 

Рисунок 1. BPMN схема процессов подписки и отмены подписки.

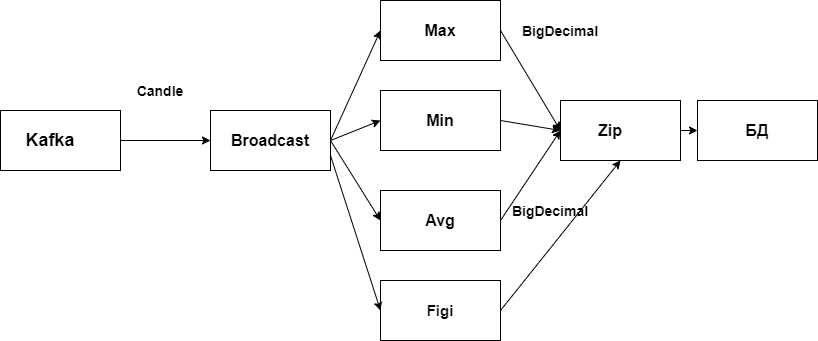


Рисунок 2. Схема реализации и связей в Pipeline.

**Порядок разработки**

- разобраться с зависимостями;

- создать класс бота и реализовать обработку текстовых сообщений пользователя, не являющихся командами;

- создать классы команд;

- прописать запуск приложения.,

**Требования к приемке-сдаче проекта**

1. Исполнитель должен предоставить следующий комплект поставки при сдаче проекта:

- Техническое задание;

- Исходный код Системы;

- Тестовые сценарии.

2. Приемо-сдаточные испытания должны проводиться по каждому этапу отдельно на сервере Заказчика в сроки, оговоренные договором.

3. Ход проведения приемо-сдаточных испытаний Заказчик и Исполнитель документируют в Протоколе проведения испытаний.

4. На основании Протокола проведения испытаний Исполнитель совместно с Заказчиком подписывает Акт приемки-сдачи программы в эксплуатацию.