Техническое задание

на разработку чат-бота «Курс криптовалют»

в мессенджере в «Telegram»

Оглавление

[Цель проекта 3](#_Toc96224967)

[Целевая аудитория 3](#_Toc96224968)

[Задачи проекта 3](#_Toc96224969)

[Функциональные требования: 3](#_Toc96224970)

[Нефункциональные требования: 4](#_Toc96224971)

[Критерии готовности: 5](#_Toc96224972)

[User Stories 5](#_Toc96224973)

[Job Stories 6](#_Toc96224974)

[Список возможных валют 7](#_Toc96224975)

[Use Cases 8](#_Toc96224976)

[Архитектурная схема 17](#_Toc96224977)

[BPMN 18](#_Toc96224978)

[UML БД 22](#_Toc96224979)

[Бизнес-метрики приложения: 23](#_Toc96224980)

[Список возможных команд 24](#_Toc96224981)

[Описание триггеров 25](#_Toc96224982)

[Допущения и ограничения: 25](#_Toc96224983)

[Предполагаемое развитие 25](#_Toc96224984)

# Цель проекта

Требуется разработать программное обеспечение Чат-бот мессенджера Телеграм для оперативного оповещения пользователей о событиях курсов криптовалют.

# Целевая аудитория

Целевая аудитория - пользователи Телеграм, связанные с финансовыми биржами, заинтересованные в оперативном получении информации о криптовалютах.

# Задачи проекта

* Создать ядро системы, обеспечивающее функционал управления подписками на события о курсе валют.
* К ядру системы реализовать интерфейс бота Телеграм.
* Реализовать механизм взаимодействия с источником курсов криптовалют

# Функциональные требования:

1. Система должна приветствовать пользователя при начале работы
2. В системе должны быть подсказки по использованию системы
3. Интерфейс системы должен быть интуитивно понятным
4. Система должна поддерживать текстовые команды взаимодействия, а также интерактивный режим через нажатия выводимых на экран кнопок
5. Пользователь должен иметь возможность
   1. создать подписку,
   2. отписаться от выбранной подписки,
   3. удалить подписки,
   4. посмотреть созданные подписки.
6. Должны быть реализованы подписки:
   1. Периодическое отображение курса выбранной пары валют. Пара валют задается пользователем через текстовые команды или интерактивный режим.
   2. Периодическое отображение среднего курса по выбранной паре валют
   3. Однократное отображение целевого значения курса выбранной пары валют. Целевое значение задается отклонением в процентах относительно текущего значения
   4. Однократно посмотреть текущий курс валют
7. При выполнении недопустимых пользователем действий должно выдаваться сообщение о невозможности такого действия, работоспособность приложения не должна нарушаться
8. Должны быть разработаны метрики работы приложения для оценки работы приложения. Виды метрик:
   1. бизнес-метрики
   2. системные метрики
9. Вывести Дашборд в Графана о состоянии сервера и сервисов.
10. Настроить алерт-сервис Телеграм для уведомления о критичных событиях.
11. Реализовать CI/CD для приложения с помощью кластера на Kubernetes на Digital Ocean с помощью сервера непрерывной интеграции GitLab.
12. Дополнительным модулем реализовать
    1. подготовку данных по криптовалютам из выделенного источника для ML-модели
    2. создать продюсер по генерации данных по криптовалюте
    3. создать консьюмер по получению данных по криптовалюте
    4. вычислить почасовой минимум и максимум каждой валюты

# Нефункциональные требования:

Язык разработки: Java

БД: PostgreSQL

Время работы: 24х7

Количество ОЗУ: не менее 1 ГБ

ЦПУ: 2 ядра по 2 ГГЦ

ПЗУ: 1 Гб

# Критерии готовности:

В чат-бот можно зайти по предоставленной ссылке.

Функционал по 1-7 пунктам функциональных требований подтвержден.

Предоставлен материал по результатам реализации метрик (снимки экрана).

Предоставлен исходный код дополнительно приложения на оценку.

# User Stories

1. Я, как трейдер, хочу получать информацию о колебаниях курса как можно чаще, чтобы выбрать выгодный момент для проведения транзакции.
2. Я, как человек, который планирует сделать долгосрочное вложение в криптовалюту, хочу получить уведомление, когда курс интересующей меня криптовалюты упадет до нужного мне уровня, чтобы я мог купить криптовалюту по выгодной цене.
3. Я, как человек, планирующий продать криптовалюту, хочу получить уведомление, когда курс криптовалюты вырастет до оптимального для меня уровня, чтобы продать свою криптовалюту по выгодной цене.
4. Я, как инвестиционный консультант, хочу получить информацию о средней цене ряда основных криптовалют на рынке, чтобы помочь моим клиентам выбрать правильный проект для инвестиций.
5. Я, как человек, который планирует заняться майнингом, хочу получать уведомления о росте или падении курса ряда криптовалют, чтобы понять, какая валюта более выгодна для майнинга.
6. Я, как человек, которому предстоит общаться с людьми, занимающимися инвестициями, хочу узнать курс основных криптовалют быстро и без подписки, чтобы суметь поддержать разговор.
7. Я, как человек, который не имеет возможности тратить время на запоминание текстовых команд и перечитывание инструкций бота, хочу управлять ботом с помощью кнопок, чтобы сэкономить свое время.

# Job Stories

1. Когда я вывожу свою криптовалюту на площадку чтобы начать торги, я хочу получать уведомления об изменении курса как можно чаще, чтобы не пропустить момент для удачной операции
2. Когда я прекращаю свою деятельность на торгах, я хочу полностью отключить бота и удалить все подписки, чтобы не получать не актуальную для меня больше информацию.
3. Когда я готовлюсь к встрече со своим клиентом, я хочу быстро получить информацию о среднем курсе основных криптовалют, чтобы я мог проконсультировать клиента и помочь ему выбрать проект для инвестиций.
4. Когда я планирую купить криптовалюту, я хочу получить уведомление, когда курс интересующей меня криптовалюты упадет до нужного мне уровня, чтобы я мог купить криптовалюту по выгодной цене.
5. Когда я планирую продать криптовалюту, я хочу получить уведомление, когда курс интересующей меня криптовалюты вырастет до оптимального для меня уровня, чтобы я мог продать криптовалюту по выгодной цене.
6. Когда я уезжаю в отпуск, я хочу иметь возможность приостановить оповещения по подпискам, чтобы у меня была возможность полноценно отдохнуть.
7. Когда я возвращаюсь из отпуска, я хочу иметь возможность быстро восстановить оповещения по моим подпискам, чтобы не тратить время на их повторное создание.

# Список возможных валют

| **Отображаем пользователю** | **Код в источнике** | **Название для отображения пользователю** | **Флаг криптовалюта** |
| --- | --- | --- | --- |
| BTC | BTC | Bitcoin | true |
| ETH | ETH | Ethereum | true |
| USDT | USDT | Tether | true |
| BNB | BNB | Binance Coin | true |
| XRP | XRP | Ripple | true |
| ADA | ADA | Cardano | true |
| RUB | RUB | Russian rouble | false |
| EUR | EUR | Euro | false |

# Use Cases

**Bot-APP**

===============================================================

**UC1. Регистрация пользователя**

1. Пользователь написал **/start**
2. Проверить наличие этого пользователя в базе данных.
3. В случае отсутствия пользователя добавить пользователя в **users.**
4. Вывести сообщение “Приветствуем в нашем боте курсов криптовалют, $Ник пользователя. Ознакомьтесь с возможными командами нашего бота”
5. Отобразить кнопку **В главное меню**

===============================================================

**UC2. Запросить инструкции использования бота**

1. Пользователь написал **/help** или нажал кнопку **Помощь.**
2. Вывести [список возможных команд](https://docs.google.com/document/d/1d9vByeXk1cC4-2G-HRC5RxQDmgGp1_GZhEU3-E3oJAQ/edit#heading=h.dnztho55lnw7) страницу
3. Отобразить кнопку **Назад**

===============================================================

**UC3. Кнопка Назад**

1. Пользователь нажал кнопку Назад
2. Перейти в предыдущее меню

===============================================================

**UC4. Кнопка в главное меню. Интерактивный режим**

1. Пользователь нажал кнопку в главное меню
2. Отправить команду **/MAIN**

**UC4. Обработка команды /MAIN**

1. Получили команду **/MAIN**
2. Отобразить кнопки **Доступные валюты || Операции с валютами || Пауза || Возобновить все подписки || Стоп || Помощь**

===============================================================

**UC5. Отображение всех подписок. Интерактивный режим**

1. Пользователь нажал кнопку **Операции с валютами**
2. Отобразить кнопки **Подписаться, Отписаться, Список**
3. Пользователь нажал кнопку **Список**
4. Отправить команду **/SHOWALL**. Примечание: здесь и далее по тексту фраза Отправить текстовую команду означает, что нажатие кнопки пользователем эмулирует набор команды пользователем и отправку команды на сервер. Обработка отправленных команд описана в соответствующем UC.

**UC5. Отображение всех подписок. Обработка команды /SHOWALL**

1. Ищем записи в **user\_triggers** по **user\_id**
2. Выводим все подписки на экран в текстовом формате **№ - XXX/YYY/Тип триггера/Период/Искомое значение**
3. Отобразить кнопку **В главное меню**

===============================================================

**UC6. Вывести все доступные валюты. Интерактивный режим**

1. Пользователь нажал кнопку **Доступные валюты**
2. Отправляем текстовую команду **/ALL\_CUR**

**UC6. Вывести все доступные валюты. Обработка команды /ALL\_CUR**

1. Выбрать все записи **currency\_figi\_user** и **currency\_description** из **currency**
2. Вывести результат в виде списка в формате **currency\_figi\_user (upper case) - currency\_description (lower case)**
3. Отобразить кнопку **В главное меню**

===============================================================

**UC7. Подписаться на рассылку курса валюты. Интерактивный режим.**

1. Пользователь нажал кнопку **Операции с валютами,**
2. Отобразить кнопки **Подписаться, Отписаться, Список**
3. Пользователь нажал кнопку **Подписаться**
4. Вывести сообщение “Выберите первую валюту”,
5. Показать кнопки валют, названия кнопок вывести из **currency.currency\_name,** список валют
6. Пользователь нажал кнопку валюты,
7. Вывести сообщение “Выберите вторую валюту”
8. Показать кнопки валют, названия кнопок вывести из **currency.currency\_name,** [список валют](https://docs.google.com/document/d/1d9vByeXk1cC4-2G-HRC5RxQDmgGp1_GZhEU3-E3oJAQ/edit#heading=h.z8okkk7deney)
9. Пользователь нажал кнопку валюты,
10. Вывести сообщение “Выберите тип отслеживания”
11. Показать кнопки **Текущий курс || Рассылка || Среднее за минуту || Будильник**
12. При нажатии кнопки **Текущий курс** перейти к **UC8. Настройка типа триггера** **Текущий курс.**
13. При нажатии кнопки **Рассылка** перейти к **UC8.** **Настройка типа триггера** **Рассылка.**
14. При нажатии кнопки **Среднее** перейти к **UC8.** **Настройка типа триггера** **Среднее.**
15. При нажатии кнопки **Будильник** перейти к **UC8. Настройка типа триггера Будильник.**

**UC8. Настройка типа триггера Текущий курс.**

1. Перешли из **UC7.**
2. Отправить текстовую команду **/once/cur1/cur2.** Перейти к **UC9. Обработка триггера типа Текущий курс.**

**UC8. Настройка типа триггера Рассылка**

1. Перешли из **UC7.**
2. Отправить текстовую команду **/simple/cur1/cur2.** Перейти к **UC9. Сохранение триггера типа Рассылка.**

**UC8. Настройка типа триггера Среднее**

1. Перешли из **UC7.**
2. Отправить текстовую команду **/average/cur1/cur2/period.** Перейти к **UC9. Сохранение триггера типа Среднее.**

**UC8. Настройка типа триггера Будильник**

1. Перешли из **UC7.**
2. Вывести сообщение “Выберите искомое направление изменения”
3. Показать кнопки **Повышение ||** **Понижение,**
4. Пользователь выбрал направление изменения. Повышение соответствует UP, Понижение соответствует DOWN,
5. Вывести сообщение “Настройте величину изменения. Текущее значение $value”. Значение $value по умолчанию 5.
6. Показать кнопки “+” || “-“ || Создать.
7. При нажатии кнопок “+” || “-“ произвести инкремент или декремент $value, перейти к пункту 5.
8. При нажатии кнопки “Создать” отправить текстовую команду **/target/cur1/cur2/dir/val.** Перейти к **UC9. Сохранение триггера типа Будильник.**

**UC9. Обработка триггера типа Текущий курс**

1. Получили команду **/once/cur1/cur2**
2. Ищем запись в **currency\_pair** с currency\_1 = cur1 и currency\_2 = cur2.
3. Если не находим, выводим сообщение “Выбранный набор валют недоступен”. Конец UC.
4. Запрашиваем текущее значение **$current\_value** пары **cur1/cur2**
5. Выводим сообщение на экран “**cur1/cur2** = **$current\_value”**
6. Отобразить кнопку **В главное меню**

**UC9. Сохранение триггера типа Рассылка**

1. Получили команду **/simple/cur1/cur2**
2. Ищем запись в **currency\_pair** с currency\_1 = cur1 и currency\_2 = cur2.
3. Если не находим, выводим сообщение “Выбранный набор валют недоступен”. Конец UC.
4. Запрашиваем **trigger\_type\_id** из **trigger\_type** по значению **currency\_pair**.
5. Проверяем, есть ли запись в **user\_triggers** с данными **user\_id**, **pair\_id, trigger\_type\_id, trigger\_period=period.**
6. Если запись есть, выводим сообщение “У вас уже есть такая подписка”. Конец UC.
7. Если записи нет, сохраняем в **user\_triggers** данные **user\_id**, **pair\_id, trigger\_type\_id, trigger\_period, trigger\_targetvalue**
8. Выводим сообщение “Подписка успешно создана”
9. Отобразить кнопку **В главное меню**

**UC9. Сохранение триггера типа Среднее**

1. Получили команду **/average/cur1/cur2/period**
2. Ищем запись в **currency\_pair** с currency\_1 = cur1 и currency\_2 = cur2.
3. Если не находим, выводим сообщение “Выбранный набор валют недоступен”. Конец UC.
4. Запрашиваем **trigger\_type\_id** из **trigger\_type** по значению **average**.
5. Проверяем, есть ли запись в **user\_triggers** с данными **user\_id**, **pair\_id, trigger\_type\_id, trigger\_period=period.**
6. Если запись есть, выводим сообщение “У вас уже есть такая подписка”. Конец UC.
7. Если записи нет, сохраняем в **user\_triggers** данные **user\_id**, **pair\_id, trigger\_type\_id, trigger\_period, trigger\_targetvalue**
8. Выводим сообщение “Подписка успешно создана”
9. Отобразить кнопку **В главное меню**

**UC9. Сохранение триггера типа Будильник**

1. Получили команду **/target/cur1/cur2/dir/val**
2. Ищем запись в **currency\_pair** с currency\_1 = cur1 и currency\_2 = cur2.
3. Если не находим, выводим сообщение “Выбранный набор валют недоступен”. Конец UC.
4. Запрашиваем **trigger\_type\_id** из **trigger\_type** по значению **target**.
5. Запрашиваем текущее значение **$current\_value** пары **cur1/cur2**.
6. В случае dir=up вычисляем **$target\_value**=**$current\_value\*(100+val)/100**
7. В случае dir=down вычисляем **$target\_value**=**$current\_value\*(100-val)/100**
8. Проверяем, есть ли запись в **user\_triggers** с данными **user\_id**, **pair\_id, trigger\_type\_id, trigger\_targetvalue=target\_value.**
9. Если запись есть, выводим сообщение “У вас уже есть такая подписка”. Конец UC.
10. Если записи нет, сохраняем в **user\_triggers** данные **user\_id**, **pair\_id, trigger\_type\_id, trigger\_period=null, trigger\_targetvalue=target\_value**
11. Отправить триггер в **TriggerAPP**
12. Выводим сообщение “Вы будете оповещены при достижении пары cur1/cur2 значения $target\_vallue”
13. Отобразить кнопку **В главное меню**

===============================================================

**UC10. Отписаться от рассылки курса валюты**

1. Пользователь нажал кнопку **Операции с валютами,**
2. Отобразить кнопки **Подписаться, Отписаться, Список**
3. Пользователь нажал кнопку **Отписаться**
4. Выполнить действия **UC5. Отображение всех подписок.**
5. Вывести сообщение “Выберите подписку для удаления“
6. Показать кнопки с “Id” подписок, полученных в п 4.
7. Пользователь нажал кнопку с соответствующим Id.
8. Отправляем текстовую команду **/unsubscribe/sub\_id**.
9. Переходим к **UC10. Обработка команды** **/unsubscribe/sub\_id**

**UC10. Обработка команды** **/unsubscribe/sub\_id**

1. Получили команду **/unsubscribe/sub\_id**
2. Ищем запись в **user\_triggers** с id = **sub\_id**
3. Если не находим, выводим сообщение “такой подписки нет”. переходим к пункту 5. конец UC.
4. Если находим, удаляем запись в **user\_triggers** с id = **sub\_id**.
5. Выводим сообщение подписка “**№ - XXX/YYY/Тип триггера/Период/Искомое значение** удалена”
6. Отобразить кнопку **В главное меню**

===============================================================

**UC11. Прекратить оповещения по всем созданным подпискам**

1. Пользователь нажал кнопку **Пауза** или вводит команду **/pause**
2. Архивируем все записи в **user\_triggers** по **user\_id**
3. Отправляем сообщение “все подписки остановлены. вы можете просмотреть свои подписки и возобновить их в любой момент”
4. Отобразить кнопку **В главное меню**

===============================================================

**UC12. Возобновить оповещения по всем созданным подпискам**

1. Пользователь нажал кнопку **Возобновить подписки** или вводит команду **/resume**
2. Возвращаем из архива все записи в **user\_triggers** по **user\_id**
3. Отправляем сообщение “все подписки восстановлены”
4. Отобразить кнопку **В главное меню**

===============================================================

**UC 13. Прекратить использование бота. Интерактивный режим**

1. Пользователь нажал кнопку **Стоп**
2. Выводим сообщение “Вы действительно хотите удалить все подписки? Действие необратимо”
3. Выводим кнопки **Да, Назад**.
4. Отправляем текстовую команду **/stop**

**UC13. Прекратить использование бота. Обработка команды /stop**

1. Получили команду **/stop**.
2. Удаляем все записи в **user\_triggers** по **user\_id**
3. Отправляем сообщение “все подписки удалены. вы можете посмотреть новые в меню Операции с валютами”
4. Отобразить кнопку **В главное меню**

===============================================================

**UC14. Получение неизвестной команды.**

1. Получили от пользователя сообщение.
2. Проверили сообщение на полное соответствие одной из [команд](https://docs.google.com/document/d/1d9vByeXk1cC4-2G-HRC5RxQDmgGp1_GZhEU3-E3oJAQ/edit#heading=h.dnztho55lnw7)
3. При несоответствии полученного сообщение какой-нибудь команде из списка вывести сообщение “Команда не опознана”.
4. Отобразить кнопку **В главное меню**

===============================================================

**UC15. Оповещение клиента**

1. Модуль оповещения регулярно (при получении данных со стрима) читает таблицу **user\_triggers**
2. Читает триггеры, на которые подписан пользователь.
3. Если **trigger\_type\_name = simple** переходим в **UC16. Оповещение клиента. Обработка триггера simple**
4. Если **trigger\_type\_name = average** переходим в **UC16. Оповещение клиента. Обработка триггера average**
5. Если **trigger\_type\_name = target** переходим в **UC16. Оповещение клиента. Обработка триггера target**
6. Если **$result** = null, Конец UC.
7. Если **$result** != null формирует ответ по каждой строке.
8. Отправляет пользователю **$result**
9. Переходит к следующей строке **user\_triggers**

===============================================================

**UC16. Оповещение клиента. Обработка триггера simple**

1. Получаем значение значение курса от **Trigger-APP** (стрим 1)
2. Сохраняем в **$result**
3. Возвращаем **$result** в **UC15. Оповещение клиента**

**UC16. Оповещение клиента. Обработка триггера average**

1. Получаем среднее значение пары от **Trigger-APP** (стрим 2)
2. Сохраняем в **$result**
3. Возвращаем **$result в UC15. Оповещение клиента**

**UC16. Оповещение клиента. Обработка триггера target**

1. Проверяем, пришел ли Триггер от **Trigger-APP**
2. Если пришел - **$result** = искомое значение пары валют.
3. Если нет - **$result = null**
4. Возвращаем **$result в UC15. Оповещение клиента**

===============================================================

**Trigger-App**

===============================================================

**UC17. Загрузка информации от поставщика информации.**

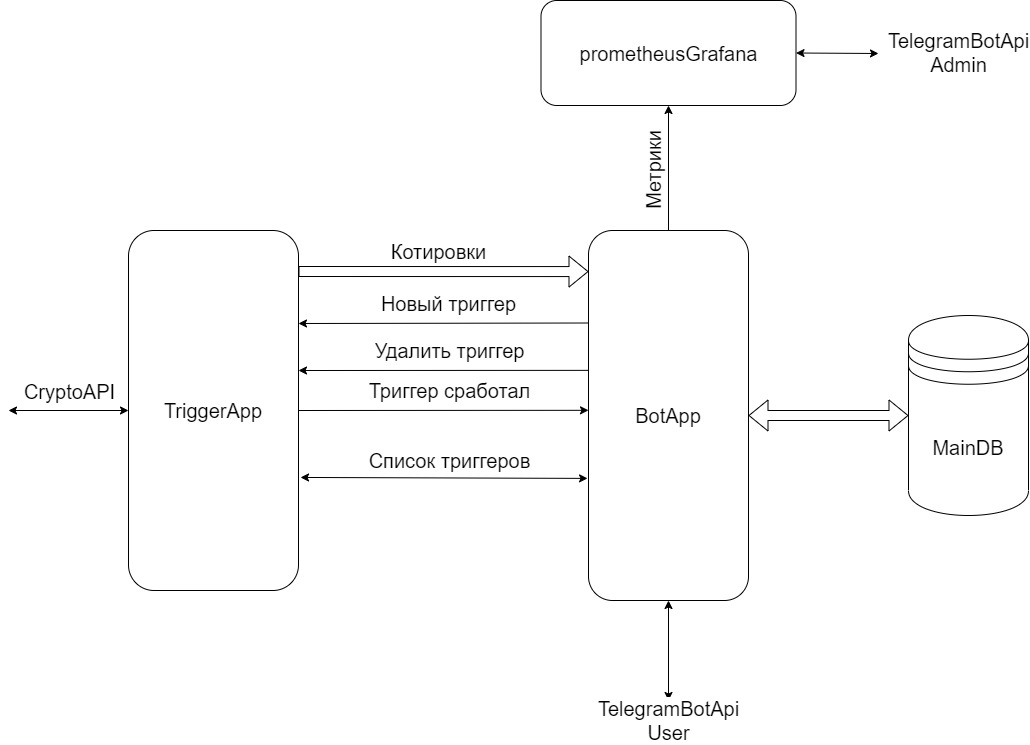
1. **Trigger-APP** при старте получает информацию о необходимых парах валют от **Bot-APP,** которые содержатся в **currency**, также получает список всех **Триггеров** типа **target**.
2. Модуль связи с поставщиком информации о курсе криптовалюты периодически загружает информацию по всем криптовалютам от источника.
3. Парсит ответ и забирает только пары валют, которые содержатся в **currency**.
4. Вычисляет min/avg/max значения за все время работы приложения
5. Вычисляет скользящее min/avg/max за 1, 5, 10 минут
6. Определяет, сработали ли **Триггеры target (UC17. Обработка триггера target)**. При срабатывании готовит к отправке информацию в виде **Id сработавшего триггера.**
7. Информацию отправляет модулю **Bot-APP**.

**UC17. Обработка триггера target**

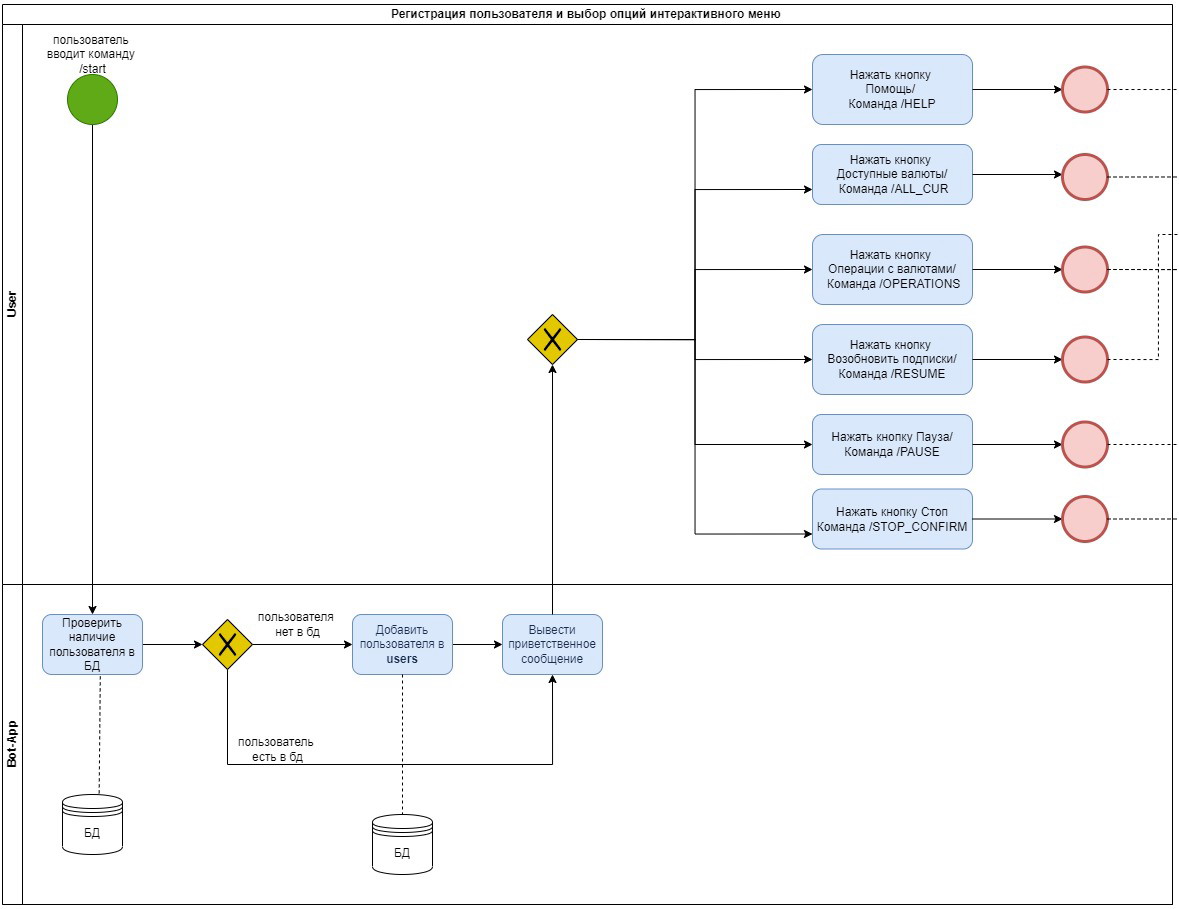
1. Выбираем последнее значение значение курса пары в триггере
2. Если значение пары == **trigger\_targetvalue,** Сохраняем **$result = trigger\_id.**
3. Иначе **$result = null**
4. Возвращаем **$result в UC15. Загрузка информации от поставщика информации.**

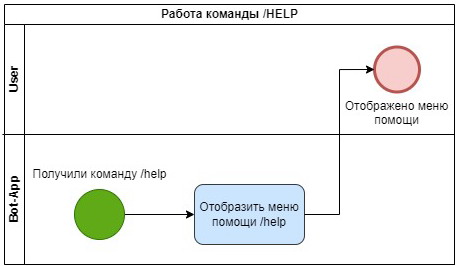
===============================================================

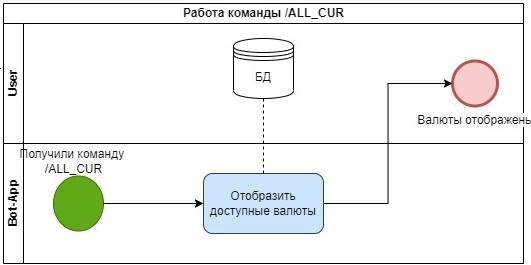
# Архитектурная схема

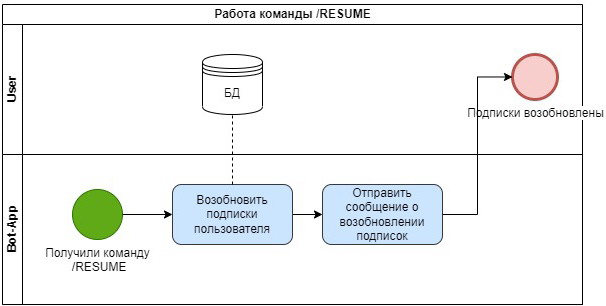


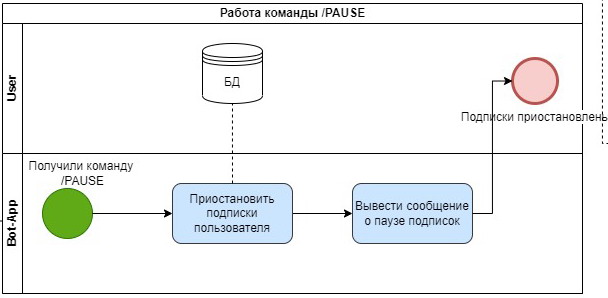
# BPMN

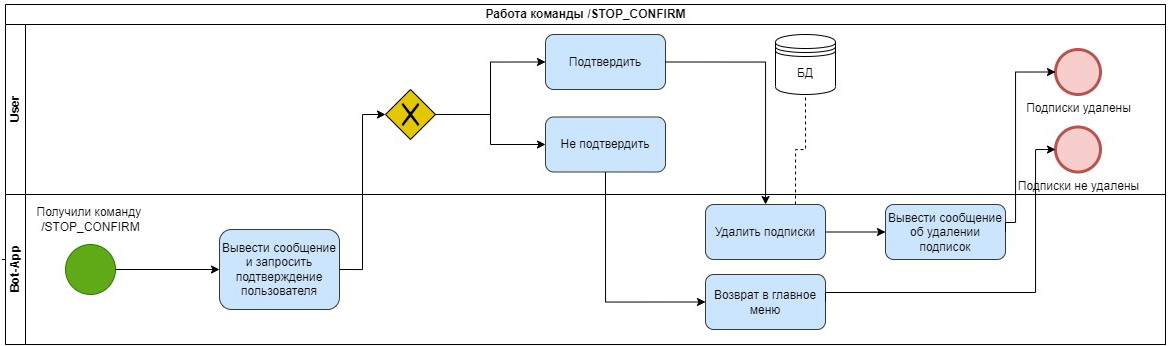
****

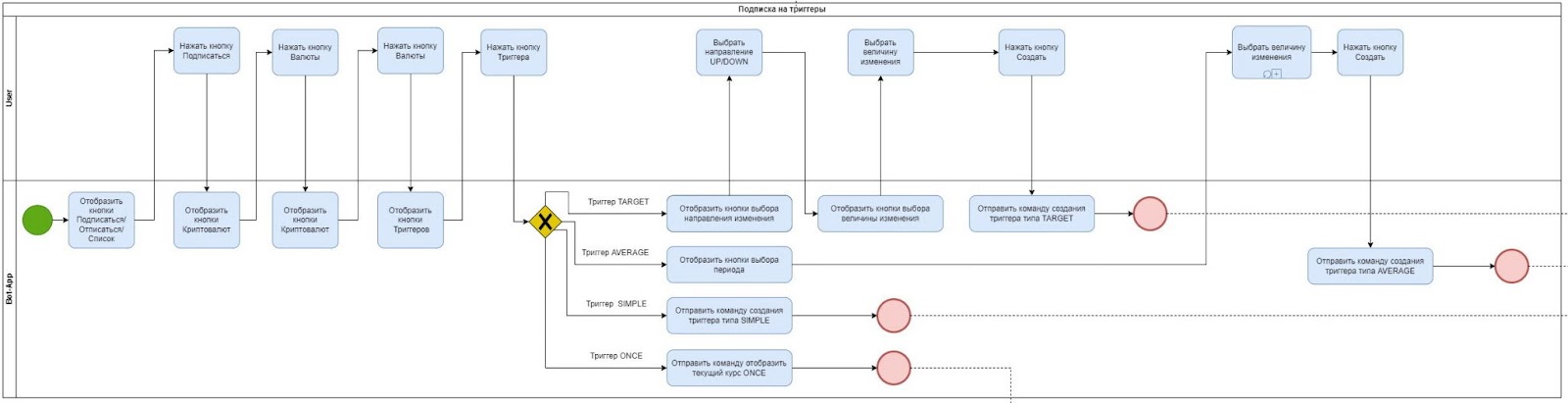
****

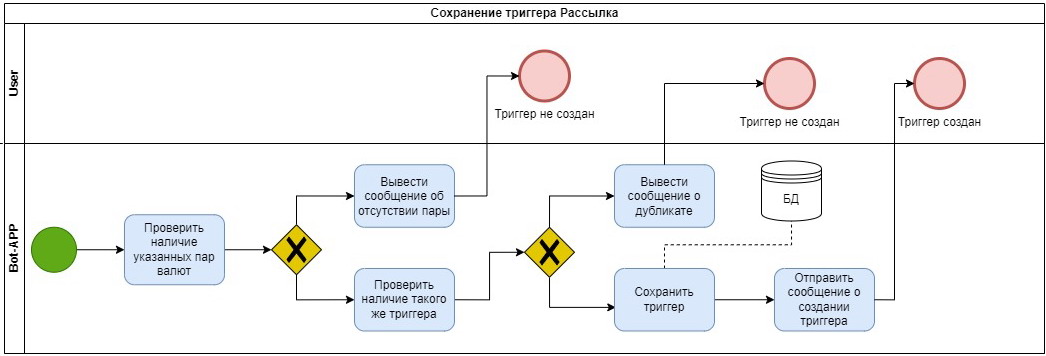
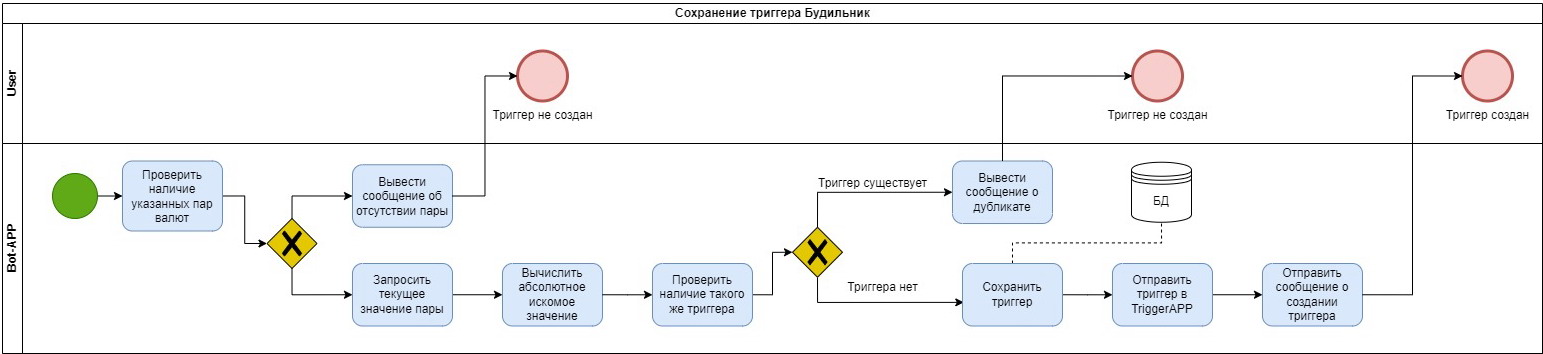
****

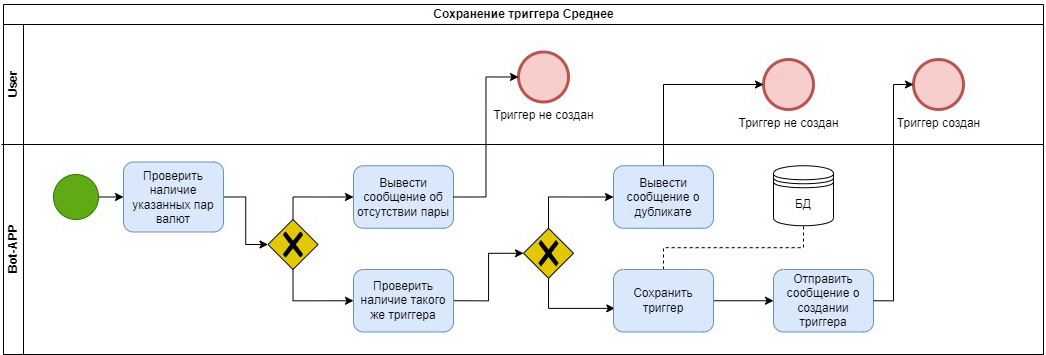
****

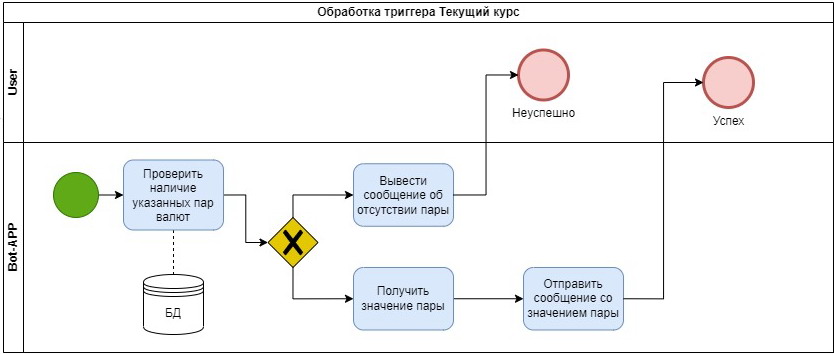


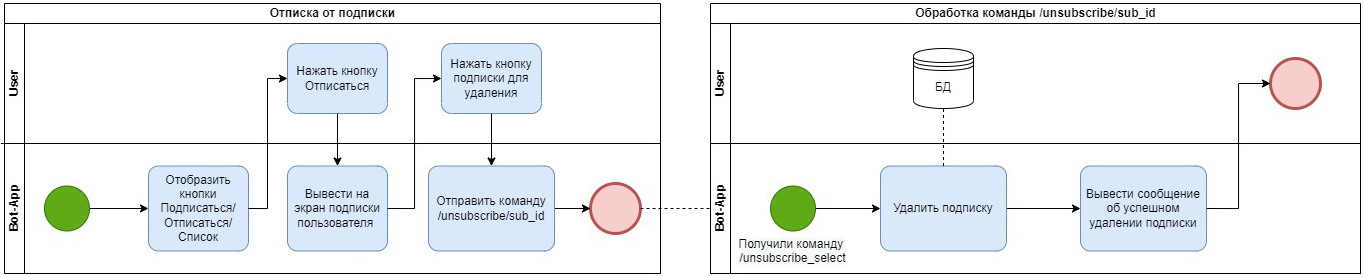
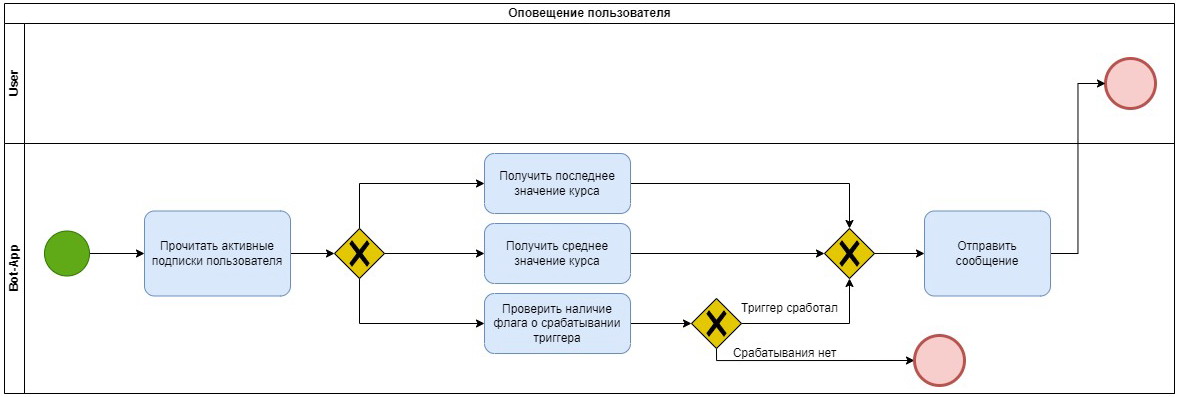
****

****

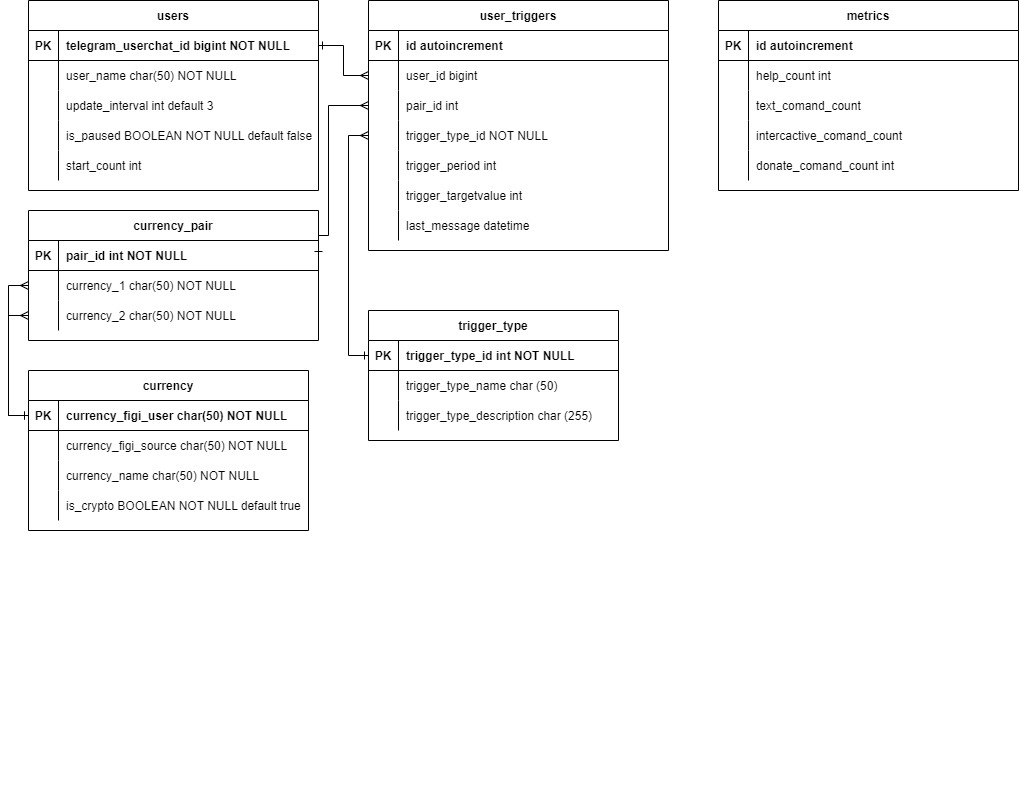
****

****

****

****

# UML БД

****

# Бизнес-метрики приложения:

Количество пользователей

* всего зарегистрированных - показывает популярность робота
* активных/архивных - достаточность функционала
* повторных - функциональность относительно конкурентов
* количество новых пользователей за месяц - показывает популярность робота

Количество подписок

* всего - популярность робота
* количество по типам - для оценки наиболее требуемого функционала, анализ расширения функционала

Количество валют

* доступные из возможных - функциональность

Интуитивность интерфейса

* количество запросов страницы помощи - показывает удобство интерфейса
* ·количество использования текстовых команд относительно интерактивного режима - показывает удобство интерфейса

Количество донатов - функциональность.

# Список возможных команд

Все команды должны начинаться со знака “/”. Например /start, /help, /ONCE/BTC/USD

1.     **/start** - начать работу с ботом

2.     **/help** - показать список возможных команд

3.     **/stop** - удалить все созданные подписки пользователя

4.     **/resume** - восстановить оповещения по всем созданным подпискам

5.     **/pause** - прекратить оповещения по всем созданным подпискам

6.     **/main** - перейти в главное меню

7.     **/subscriptions/showall** - показать все созданные подписки. Рекомендация: пользуйтесь командой в режиме **/pause** для удобства использования.

8.     **/unsubscribe/sub\_id** - удалить подписку с идентификационным номером sub\_id. Рекомендация: пользуйтесь командой в режиме **/pause** для удобства использования.

9.     **/target/cur1/cur2/dir/val** - создать подписку на достижение целевого изменения (повышение или понижение) от текущего курса на заданный процент, где cur1 и cur2 - коды доступных валют, dir - целевое направление изменения UP || DOWN, val - значение изменения в %. Пример /target/btc/rub/up/10. Соответствует кнопке **Будильник**.

10.  **/average/cur1/cur2/period** - создать подписку на среднее значение за период времени period в минутах. Соответствует кнопке **Среднее**.

11.  **/simple/cur1/cur2** - создать подписку на курс валют, где cur1 и cur2 - коды доступных валют. Соответствует кнопке **Рассылка**.

12.  **/all\_cur** - отобразить доступные валюты. Рекомендация: пользуйтесь командой в режиме **/pause** для удобства использования.

13.  **/once/cur1/cur2** - однократно узнать стоимость пары валют, где cur1 и cur2 - коды доступных валют. Например /ONCE/BTC/USD. Соответствует кнопке **Текущий курс**.

14.  Рекомендуем пользоваться интерактивным режимом работы с роботом, последовательно нажимая кнопки на экране.

# Описание триггеров

|  |  |  |
| --- | --- | --- |
| **Название кнопки** | **Код триггера** | **Описание триггера** |
| Рассылка | simple | Периодический вывод данных пары валют |
| Среднее | average | Среднее значение пары валют за выбранный период |
| Будильник | target | Сообщение о достижении необходимого значения пары валют |
| Текущий курс | once | Разовый запрос текущего значения |

# Допущения и ограничения:

Для реализации выбрано 8 валют

Функционал модуля обработки Триггеров ограничен.

# Предполагаемое развитие

Настраиваемое время периодического оповещения

Выбор источника курса

Функционал торговли валютами.

Если пользователь долго сидит без подписок, напомнить о себе сообщением.

Настройка суточного графика паузы и возобновления оповещения. Например, в 00:00 автоматически ставить на паузу, в 10:00 автоматически возобновлять оповещения.

Функционал материальной благодарности разработчиком – donate.

Оповещение о появлении новой валюты на рынке.