

# ExpressReport

- Лёвин Александр Александрович, МГУ ВМК
- Солодова Надежда Алексеевна, МГУ Мехмат
- Ефремова Ольга Игоревна, МГУ ВМК



# Описание проекта

---

**ExpressReport** - сигнальный Telegram-бот, собирающий дневные торговые настроения по активам для предоставления краткой ежедневной рекомендации на покупку или продажу актива.

## Реализованная функциональность

---

В рамках использования Telegram-бота пользователь сможет:

1. Подписываться на интересующие активы из имеющегося списка;
2. Изменять выбор интересующих активов;
3. Получать ежедневный отчёт о торговых настроениях по активам, на которые была совершена подписка.

# Получение торговых настроек

---

Торговые настройки собираются с сайта **TradingView** - это аналитическая платформа и социальная сеть, в которой трейдеры могут делиться своими идеями об активах.

По каждому популярному активу ежедневно выкладывается большое количество идей от трейдеров, и эти идеи могут быть следующих видов:

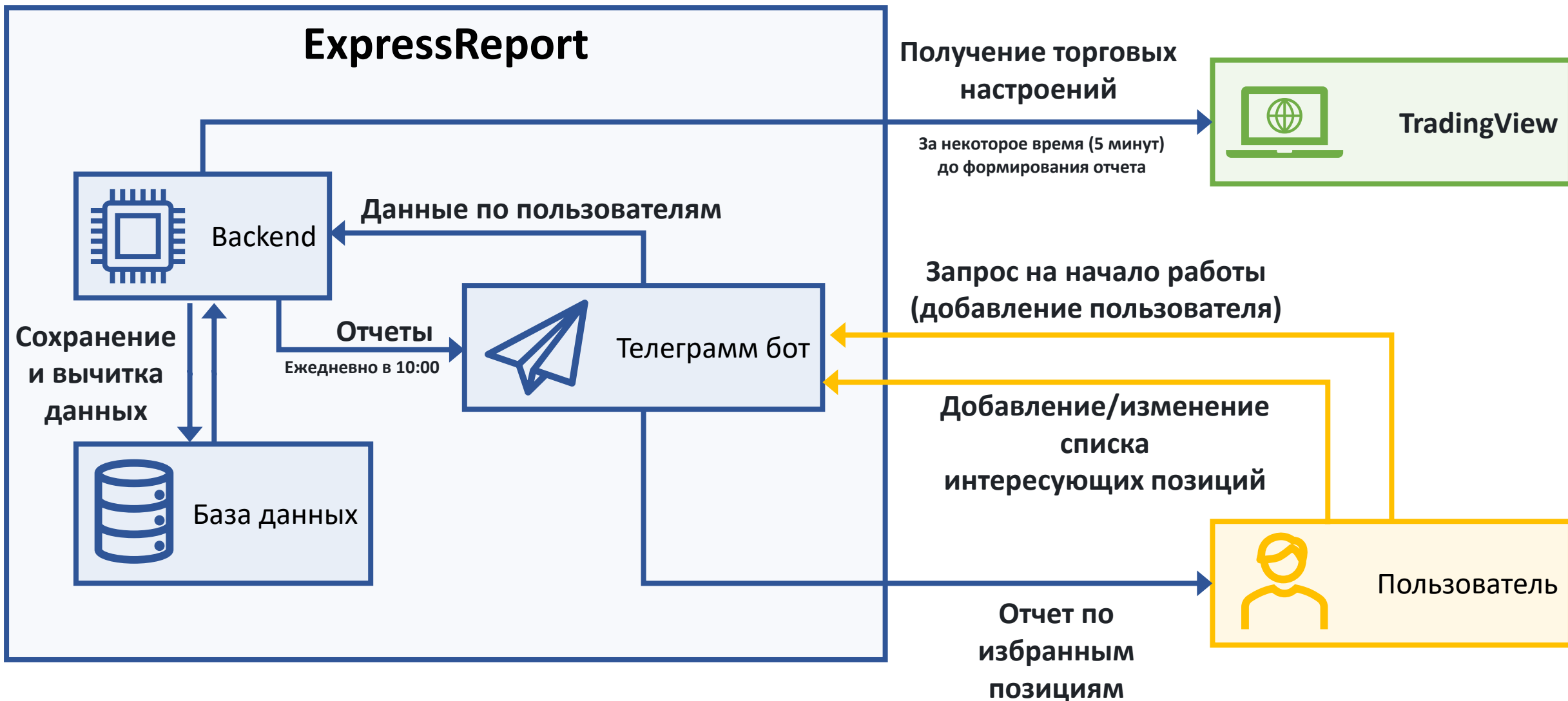
Положительными  
(рекомендация к покупке)

Отрицательными  
(рекомендация к продаже)

Нейтральными  
(рекомендация не трогать актив)

Имея количество идей по каждому из видов, можно определить общее настроение рынка по выбранному активу и принять более взвешенное решение при торговле. С этой целью и был разработан данный проект.

# Архитектурная схема



# Схема базы данных

В рамках проекта была спроектирована База Данных на основе СУБД PostgreSQL

user_log_info	
123 log_id	serial4 NOT NULL
123 user_id	int4 NOT NULL
subscribes_before	_bool NOT NULL
subscribes_after	_bool NOT NULL
change_date	timestamp NOT NULL

user	
123 user_id	serial4 NOT NULL
ABC user_name	varchar(100) NOT NULL
create_date	timestamp NOT NULL
subscribes	_bool NOT NULL

asset_update	
123 update_id	serial4 NOT NULL
ABC asset_name	varchar(32) NOT NULL
123 n_buy	int4 NOT NULL
123 n_sell	int4 NOT NULL
123 n_neutral	int4 NOT NULL
ABC recommendation	varchar(32) NOT NULL
123 open_price	int4 NOT NULL
123 close_price	int4 NOT NULL
123 low_price	int4 NOT NULL
123 high_price	int4 NOT NULL
update_date	timestamp NOT NULL

## USER

содержит информацию о всех зарегистрированных в системе пользователях (написали */start*).

## USER\_SER\_LOG\_INFO

содержит историю изменений подписок пользователя.

При изменении пользователем активных подписок поле **USER.SUBSCRIBES** автоматически обновляется, а предыдущий набор подписок заносится в **USER\_LOG\_INFO**.

## ASSET\_UPDATE

содержит дневную информацию по всем поддерживаемым активам. Добавление информации происходит раз в сутки.

# Инструкция работы с ботом

---

№	Действие	Описание
1	Зарегистрироваться в системе	Необходимо отправить “/start” в диалоговое окно
2	Выбрать интересующие активы	С помощью динамических кнопок
3	Подтвердить выбор	С помощью кнопки «Подтвердить выбор». Бот отправит сообщение со списком всех выбранных активов
4	Получить отчет по активам	Отчет отправляется автоматически ежедневно в установленное время

# Инструкция по локальному запуску бота

---

0

Создание окружения  
`python3 -m virtualenv .venv`

Linux: Активация окружения

`source .venv/bin/activate`

Windows: Активация окружения

`.venv/Scripts/activate`

1

Загрузка зависимостей проекта  
`pip install -r ./src/requirements.txt`

2

Создание Базы данных  
`python3 -m ./src/database/db_orm.py`

*Перед первым запуском необходимо раскомментировать 47 строку в db\_orm.py для создания БД.*

3

Запуск бота  
`python3 -m ./src/interface/bot.py`

# Пути дальнейшего роста

---

- Перенос бота на сервер для постоянного функционирования;
- Отладка всех ошибок при параллельной обработке запросов;
- Расширение списка активов за счёт более компактного интерфейса;
- Добавление выбора времени для присылаемого отчёта;
- Добавление недельных отчётов по активам.