

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«**СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**»

Институт космических и информационных технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О. В. Непомнящий

подпись

« _____ » _____ 2023г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 – Информатика и вычислительная техника

код – наименование направления

Телеграм-бот «Мосбиржа-информатор»

тема

Руководитель _____ старший преподаватель К.В. Коршун
подпись, дата

Выпускник _____ С.А. Ману
подпись, дата

Нормоконтролер _____ старший преподаватель К.В. Коршун
подпись, дата

Красноярск 2023

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«**СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ**»

Институт космических и информационных
технологий

институт

Вычислительная техника

кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

_____ О. В. Непомнящий

подпись

« _____ » _____ 2023г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Красноярск 2023

Студенту Ману Сергею Александровичу
фамилия, имя, отчество

Группа КИ19-08Б Направление (специальность) 090301
номер код

Информатика и вычислительная техника
наименование

Тема выпускной квалификационной работы: Телеграм-бот
“Мосбиржа-информатор”

Утверждена приказом по университету № _____ от _____

Руководитель ВКР: К.В. Коршун, Старший преподаватель кафедры ВТ
инициалы, фамилия, учёная степень, должность, место работы

ИКИТ СФУ

Исходные данные для ВКР:

Задание на ВКР.

Перечень разделов ВКР:

1) Анализ предметной области;

2) Разработка чат-бота;

3) Тестирование и руководство пользователя.

Перечень графического материала: Презентация в формате

Microsoft PowerPoint.

Руководитель ВКР _____ К.В. Коршун
подпись инициалы, фамилия

Задание принял к исполнению _____ С.А. Ману
подпись инициалы, фамилия

« » Г.
дата

РЕФЕРАТ

Выпускная квалификационная работа по теме «Telegram-бот с возможностью получения данных о торгах с Мосбиржи» содержит 40 страниц текстового документа, 11 рисунков, 8 использованных источников и 4 приложения.

ТЕЛЕГРАМ-БОТ С ВОЗМОЖНОСТЬЮ ПОЛУЧЕНИЯ ДАННЫХ О ТОРГАХ С МОСБИРЖИ.

Цель работы: разработать чат-бота для мессенджера Telegram для отслеживания хода торгов акциями и валютными парами.

Задачи:

- провести анализ системы чат-ботов и инструментов для реализации чат-бота и анализ получаемых данных;
- разработать интерфейс и функционал чат-бота – модуль получения данных, модуль обработки, модуль аналитики;
- провести тестирование чат-бота и написание руководства пользователя.

В первой главе проведен анализ предметной области, приведен анализ системы чат-ботов как в общем виде, так и конкретно для Telegram'a, представлено подробное описание используемых технологий и продуктов для разработки.

Во второй главе показан процесс разработки как модуля, получающего данные с сайта Мосбиржи, так и модуля, обрабатывающего данные. Приведена общая архитектура системы, подкрепленная диаграммами.

В третьей главе представлено тестирование чат-бота на базе мессенджера Telegram, а также описано руководство пользователя.

В результате работы над ВКР был разработан и реализован чат-бот на базе мессенджера Telegram для отслеживания торгов на Мосбирже.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Анализ предметной области	4
1.1 Анализ задания и обоснования выбора принимаемого решения.....	4
1.2 Анализ системы ботов	5
1.3 Анализ требований к аппаратной и программной частям.....	8
1.3.1 Программная часть	8
1.3.2 Аппаратная часть	11
1.4 Вывод по первой главе	13
2 Разработка чат-бота.....	14
2.1 Регистрация и настройка чат-бота.....	14
2.2 Получение данных с сайта Мосбиржи.....	16
2.3 Построение БД на основе SQLite	18
2.4 Описание прецедентов.....	20
2.5 Диаграмма классов.....	23
2.6 Вывод по второй главе	24
3 Тестирование и руководство пользователя	25
3.1 Тестирование работы телеграм-бота.....	25
3.2 Интерпретация сообщений телеграм-бота	27
3.3 Вывод по третьей главе	29
ЗАКЛЮЧЕНИЕ	30
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	31
ПРИЛОЖЕНИЕ А	32
ПРИЛОЖЕНИЕ Б.....	33
ПРИЛОЖЕНИЕ В	36
ПРИЛОЖЕНИЕ Г.....	39

ВВЕДЕНИЕ

Биржевые торги – основа рыночной развитой экономики, где огромное количество участников совершает сделки с различными видами активов. Вид активов с самой высокой ликвидностью и спросом – акции. В 2022 году частные инвесторы вложили в акции на Московской бирже 243,5 млрд рублей [1].

В настоящее время торговля на бирже и инвестирование продолжает активно набирать популярность. Число физических лиц, имеющих брокерские счета на Московской бирже, за 2022 год увеличилось на 6,1 млн и достигло 22,9 млн [2]. Увеличение участников торгов ведет к увеличению волатильности на рынке. Волатильность – это рост или спад, то есть изменение цены на тот или иной актив. Особенно важен сегодня вопрос колебания цен на те или иные активы встал после открытия торгов Московской биржи. Колебания цен внутри торгового дня сессии позволяет инвестору попробовать заработать на разнице купленного актива и проданного – такое называют трейдингом.

Также остро становится вопрос о способах получения информации. Все больше трафика, а потому и внимания, получает мессенджер Telegram. Telegram занимает первое место в рейтинге iOS App Store в категории “социальные сети” и третье место в рейтинге Google Play по всем категориям.

Сегодня у каждого брокера имеется телеграм-канал, в котором человек может оперативно получить актуальную информацию, как о биржевых торгах в целом, так и информацию по отдельному эмитенту, в частности.

Ежедневные публикации о состоянии как мировых фондовых рынках, так и об отечественном, стали уже больше данностью и необходимой частью, чем специализированной информацией.

1 Анализ предметной области

1.1 Анализ задания и обоснования выбора принимаемого решения

Разработка телеграм-бота с возможностью получения данных о торгах с сайта Мосбиржи.

Разрабатываемое приложение предназначено для оперативного получения информации об изменении цены того или иного биржевого актива.

Приложение должно работать в системе мессенджера Telegram в виде специального аккаунта – бота. Приложение состоит из серверной части, обработчика получаемых данных с сайта Мосбиржи, и части, отвечающей за взаимодействие с пользователем.

Чат-бот обладает следующим перечнем функций, которые он может предложить пользователю:

- выдача данных пользователю о торгах акциями из котировального списка индекса акций Мосбиржи;
- выдача данных пользователю о ходе торгов на валютном рынке. Валютные пары вида USD/RUB, EUR/RUB, CNY/RUB.

Со списком акций индекса Мосбиржи (IMOEX) можно ознакомиться на любой платформе по торговой и биржевой информации, к примеру, сайт блога Смарт-Лаб [3].

1.2 Анализ системы ботов

Бот – автоматизированный алгоритм, который состоит из заданной последовательности действий. Главная задача бота – автоматизация процессов и решения задач пользователя.

Прообразом чат-ботов стал тест Тьюринга 1950 года. Он заключался в том, что человек должен был определить в диалоге между машиной и другим человеком, где компьютер, а где живой собеседник.

В 1966 году Джозеф Вейценбаум создал программу-психиатра Элизу, которая строила диалог по ключевым фразам и могла каждый раз создавать новую ветку беседы по слову из реплики пользователя. Продолжил тему психиатрических экспериментов робот Парри, который имитировал поведение больного шизофренией и служил для обучения студентов медицинских вузов.

С 1988 года благодаря Ролло Карпентеру и его JabberWacky диалоговые программы стали частью развлечений. В 1991 году в Сингапуре создали компьютерного психолога Dr.Sbaitso, а в 1995 году появился первый проект с открытым исходным кодом — алгоритм с женским образом A.L.I.C.E.

2010 год стал стартом для развития виртуальных голосовых помощников от ведущих мировых IT-брендов. Один за другим в сети появлялись Siri, Cortana, Alexa, Алиса, ассистенты Салют. А с 2016 года простые диалоговые программы стали набирать популярность в бизнесе.

Telegram-боты – специализированные программы, написанные для платформы Telegram. Спектр функций ботов разнообразен: получение новостей, поиск информации, переводы сообщений пользователей. Сами боты делятся на несколько направлений:

- Чат-боты. Представляют из себя чат, имитирующий общение с пользователем на заданную тематику.

- Боты-информаторы. Бот, информирующий пользователя о тех или иных событиях, будь то новости, анонсы мероприятий, интересующие пользователя публикации.
- Бот-ассистент. Боты, разработанные онлайн-сервисами, как дополнение к основной веб-версии.
- Игровые боты. Пользователь с помощью такого бота может играть в HTML-5 игры в группах или частных чатах.

Техническая составляющая системы работы чат-ботов в Telegram базируется на использовании API (Application Programming Interface) Telegram. API Telegram — это набор инструментов, предоставляемых разработчикам, для создания и интеграции программных приложений с Telegram.

Чат-боты в Telegram работают на основе протокола HTTP (HyperText Transfer Protocol). При получении сообщения от пользователя, Telegram отправляет запрос с данными о сообщении на сервер бота. Бот обрабатывает запрос и формирует ответ, который отправляется обратно на сервер Telegram. Затем Telegram передает ответ пользователю в виде сообщения.

Ниже представлена схема системы работы чат-ботов в Telegram (Рисунок 1).

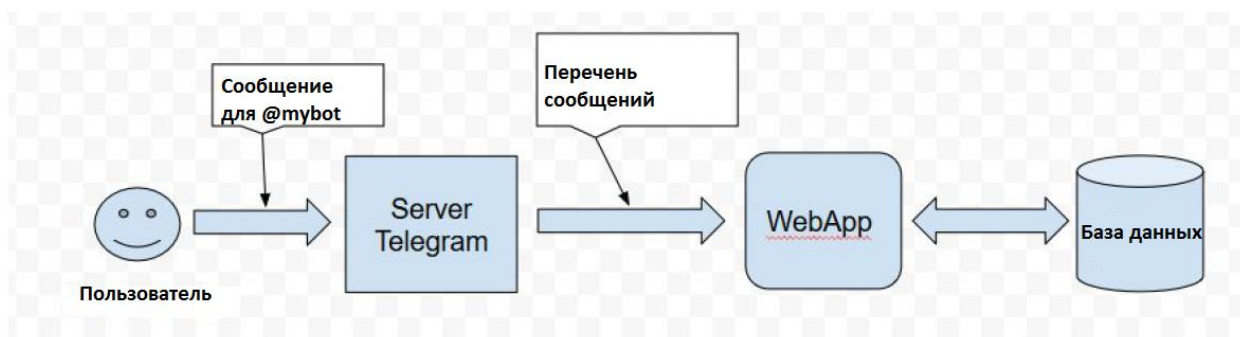


Рисунок 1 – Система работы ботов в Telegram

Для понимания работы чат-бота в Telegram следует разобрать работу существующего чат-бота «URL Shortener».

URL Shortener позволяет сокращать длинные URL-адреса. URL – унифицированный указатель ресурса – система унифицированных адресов электронных ресурсов, или единообразный определитель местонахождения ресурса. Достаточно просто написать ему ссылку на интернет-ресурс, URL которой нужно сократить. Пример работы продемонстрирован ниже (Рисунок 2).



Рисунок 2 – пример работы телеграм-бота «URL Shortener»

Для работы чат-бота в Telegram нужны следующие компоненты и действия.

1. Создание и регистрация бота в системе Telegram.
2. Разработка сценария работы бота.
3. Сервер для развертывания бота, в котором будут обрабатываться запросы пользователя и храниться данные с сайта Мосбиржи.

1.3 Анализ требований к аппаратной и программной частям

1.3.1 Программная часть

Программная часть состоит из выбора языка программирования, на котором будет произведена разработка бота. Нужен язык, позволяющий выполнять серверное программирование, а также имеет библиотеки для работы с Telegram.

В качестве языка программирования был выбран Python. Python является мощным и универсальным языком программирования, который позволяет быстро и эффективно создавать приложения в различных областях.

Основные преимущества Python можно описать так:

- Простота и легкость изучения: Python имеет простой синтаксис и легок в освоении, что делает его привлекательным для новичков.
- Обилие библиотек: Python имеет большое количество библиотек, которые делают его универсальным языком программирования. NumPy для научных вычислений, Pandas для работы с данными, Django для веб-разработки и многие другие.
- Переносимость: Python может работать на различных операционных системах, таких как Windows, MacOS и Linux. Кроме того, программы, написанные на Python, могут быть легко перенесены на другие платформы.

Его простота и доступность делают его привлекательным для начинающих программистов, а его мощь и гибкость - для профессиональных разработчиков.

Из недостатков языка Python можно выделить динамическую типизацию, то есть отсутствие статической типизации, что может привести к некоторым проблемам при работе с классами. Ещё один недостаток Python – скорость. Python – скриптовый язык программирования. Для задач высокой производительности он выступает лишь для общения с API низкоуровневых библиотек. Одна из причин его медлительности – проверка типа объекта, перед взаимодействием с ним.

Aiogram — это фреймворк на языке Python, предоставляющий простой в использовании интерфейс для разработки ботов Telegram [4]. Он построен на базе Telegram Bot API и призван упростить процесс разработки для создателей ботов. Фреймворк предоставляет ряд возможностей, включая поддержку встроенных запросов, работу с клавиатурой и редактирование сообщений.

Одним из главных преимуществ использования Aiogram является его простота. Библиотека предоставляет удобный интерфейс, позволяющий легко создавать и управлять ботами Telegram. Она также поддерживает широкий спектр функциональных возможностей, включая отправку сообщений, получение обновлений и обработку пользовательского ввода.

Еще одно преимущество использования Aiogram – гибкость при настройке для работы. Фреймворк разработан таким образом, что его можно настроить под свои нужды. Это делает ее идеальным выбором для разработчиков, которые хотят создавать сложные Telegram-боты с расширенной функциональностью.

Aiogram также включает в себя ряд инструментов и утилит, которые облегчают отладку и тестирование вашего бота. Например, фреймворк включает встроенный логгер, который позволяет легко отслеживать ошибки и

исключения. Он также предоставляет ряд инструментов для тестирования, которые помогут вам убедиться, что ваш бот работает правильно, прежде чем развернуть его на производстве.

Telegram Bot API – HTTP интерфейс для подключения и настройки бота в системе Telegram. Боты в Telegram – учетные записи, с одним лишь исключением – они не требуют номер телефона для настройки записи.

Существует два способа получения/отправки данных от бота.

- Long Polling – постоянно с определенным промежутком времени опрашивает сервера Telegram о наличии обновлений.
- Webhook – сервер сам опрашивает и оповещает приложение об обновлениях.

Боту при создании присваивается уникальный токен вида 123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11. Получить его можно, зарегистрировав чат-бота в чат-боте «Bot-Father» от Telegram. Управление ботом происходит с помощью HTTP – запросов. Подробнее ознакомиться с работой и взаимодействием с Telegram Bot API можно на сайте с документацией Telegram [8].

1.3.2 Аппаратная часть

Аппаратная часть состоит из выбора сервера для хранения и обработки данных. У хостинга для чат-бота должна быть поддержка нескольких языков программирования. Хостинг для телеграм-бота на Python должен поддерживать этот язык.

Для развертывания чат-бота в Telegram лучше всего подходит VDS/VPS. VDS/VPS — это виртуальный выделенный сервер, который хостинг-провайдер предоставляет в аренду пользователям.

- VPS (Virtual Private Server) — виртуальный приватный сервер.
- VDS (Virtual Dedicated Server) — виртуальный выделенный сервер.

У провайдера есть дата-центр, то есть помещение, где находятся сервера. Ресурсы каждого такого сервера делят на части и запускают несколько независимых друг от друга виртуальных серверов — VPS/VDS. По функционалу они практически не отличаются от физических серверов. Они не требуют наличия домена в обязательном порядке, и к нему можно будет обратиться по выделенному IP-адресу, что удобно при развертывании бота. К тому же такие серверы лучше поддерживают увеличение масштабов проекта, так что при увеличении потока посетителей не придется переходить на другой сервер — достаточно будет просто сменить тариф или его настройки.

Выделенные IP-адреса бывают двух типов: статические и динамические.

Статический выделенный IP-адрес — это адрес, который назначается устройству постоянно и не меняется со временем. Такой тип IP-адреса удобен для использования в случаях, когда необходимо постоянное и надежное соединение с Интернетом, например, для хостинга сайта или сервера.

Динамический выделенный IP-адрес — адрес, который назначается устройству на определенный период времени и может меняться при каждом подключении к Интернету. Такой тип IP-адреса удобен для использования в случаях, когда необходимо иметь доступ к Интернету с разных устройств, но

при этом не требуется постоянного адреса. Динамический выделенный IP-адрес часто используется провайдерами Интернет-услуг, чтобы оптимизировать использование доступных адресов.

Так как информация, передаваемая от бота, состоит исключительно из текста, то есть передаваться будут только текстовые сообщения, то мощности сервера не будут столь велики.

Проанализировав рынок хостинга VPS/VDS в России, был выбран следующий тарифный план облачного VPS:

- операционная система – Ubuntu 20.04. LTS. Эта версия Ubuntu будет поддерживаться минимум до конца 2025 года, что делает ее наиболее актуальной ОС;
- SSD – 80 ГБ NVME SSD. Подходит для быстрого доступа к данным на сервере;
- RAM – 4 ГБ DDR4 SDRAM. Отличается повышенными частотными характеристиками и пониженным напряжением питания;
- vCPU - Intel Xeon E5-2660v3. Отлично подходит для работы на базе VPS.

1.4 Вывод по первой главе

В первой главе выпускной квалификационной работы поставлены цели и задачи, которые должны быть выполнены при реализации выпускной квалификационной работы. Обзор системы устройства как ботов в общем, так и чат-ботов, в частности, позволило понять, какие требования нужно выставить к разрабатываемому телеграм-боту. В результате выделены общие возможности и особенности рассмотренного программного обеспечения. Схематичное представление системы чат-бота позволило оценить объем работы.

2 Разработка чат-бота

2.1 Регистрация и настройка чат-бота

Первое, что нужно сделать для создания бота – пройти регистрацию у специализированного чат-бота «BotFather».

Требуется одна команда «/newbot» для регистрации бота, после чего указывается имя нашего бота. После этого произойдет генерация уникального токена авторизации для чат-бота. Результаты взаимодействия с «BotFather» представлены ниже (Рисунок 3).

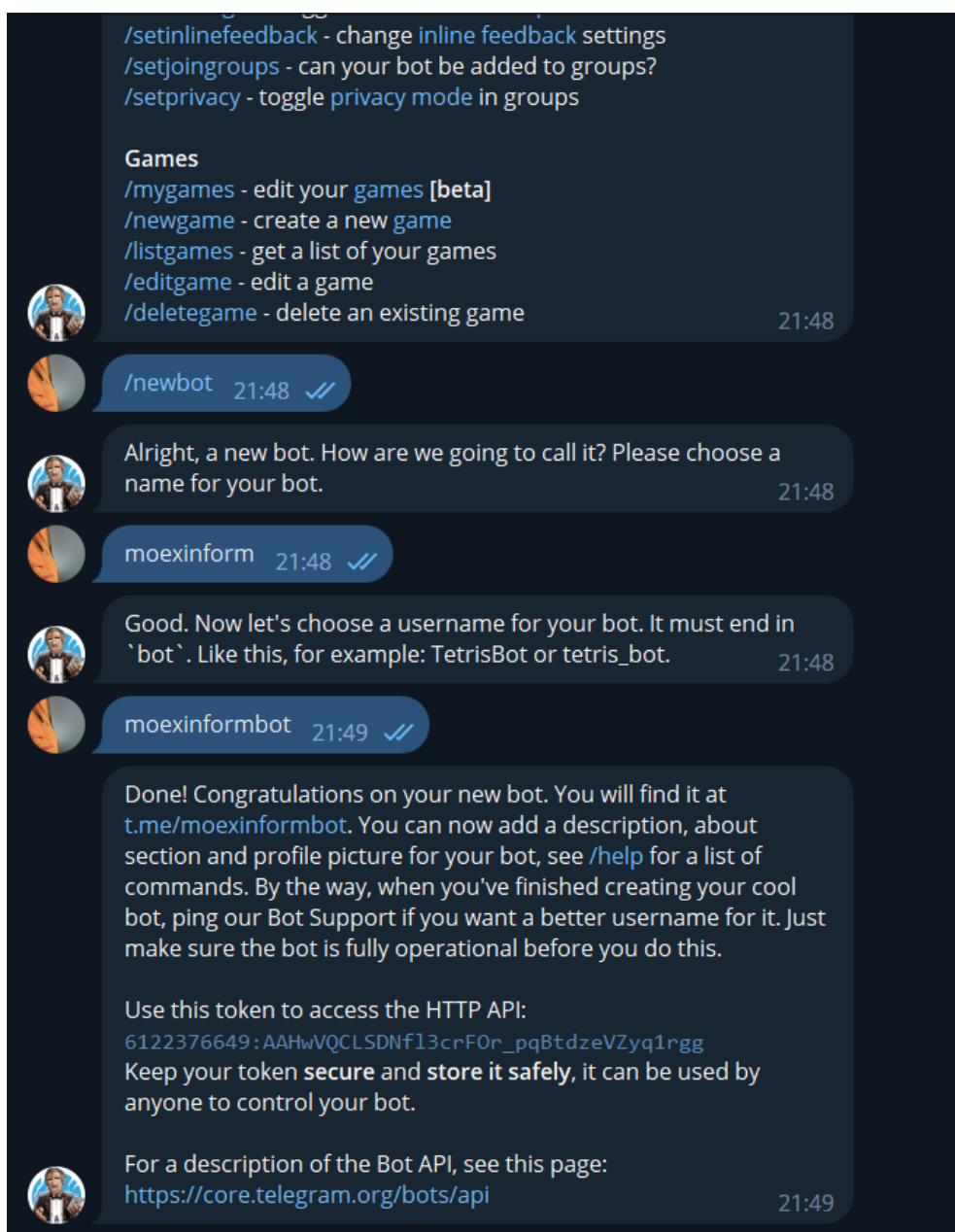


Рисунок 3 – Регистрация чат-бота в боте Telegram «BotFather»

Токен авторизации для подключения к HTTP API имеет вид «6122376649:AAHwVQCLSDNfl3crFOr_pqBtdzeVZyq1rgg» и защищается лично разработчиками. В случае, если доступ к токену смогли получить из вне или разработчик потерял его, существует возможность создать новый токен, введя команду «/token».

У чат-ботов есть пострегистрационные настройки, приведенные и описанные в таблице 1.

Таблица 1 – Команды для конфигурации чат-ботов

Команда	Описание
/setname	Изменение имени чат-бота
/setdescription	Изменение описания чат-бота, текст до 512 символов, описывающий бота. Пользователи увидят этот текст в начале разговора с ботом, озаглавленный "What can this bot do?".
/setabouttext	Изменение информации о боте, более короткий текст до 120 символов. Пользователи увидят этот текст на странице профиля бота.
/setuserpic	Изменение фото профиля чат-бота.
/setcommands	Изменение списка команд, поддерживаемых чат-ботом. Пользователи будут видеть список команд всякий раз, когда они набирают '/' в разговоре с вашим ботом.
/deletebot	Удаление чат-бота и освобождение его логина.
/setinline	Переключение встроенного режима запросов для бота.
/setjoiningroups	Переключатель, который отвечает за то, можно ли добавлять бота в группы или нет.

2.2 Получение данных с сайта Мосбиржи

Чат-боту необходимо получать данные о торгах с сайта Мосбиржи. Данные предоставлены в виде следующей таблицы (Рисунок 4):

Ход торгов, 23.04.2023
T+: Основной режим - безадрес.

Код	Кратк. наим.	Последняя	Изм., %	Первая	Мин.	Макс.	Ср.взвеш.	Объем	Время
SBER	Сбербанк	235,17	-0,47 %	237,4	233,52	238,85	236,14	27 370 731 655	23:50:06
LKOH	ЛУКОЙЛ	4 729,5	-0,35 %	4 759,5	4 711	4 776,5	4 742	4 055 912 113	23:50:06
GAZP	ГАЗПРОМ ао	181,72	-0,01 %	182	181,21	183,49	182,12	3 814 045 320	23:50:06
YNDX	Yandex clA	2 086,8	-0,56 %	2 106,6	2 062	2 128,6	2 087,4	2 574 409 843	23:50:06
GMKN	ГМКНорНик	15 798	-0,03 %	15 740	15 420	15 970	15 738	2 119 306 856	23:50:06
SBERP	Сбербанк-п	235,47	-0,37 %	237,95	233	238,35	236,05	2 089 954 061	23:50:06
VTBR	ВТБ ао	0,020195	+0,82 %	0,020045	0,01986	0,02031	0,0201	2 083 710 435	23:50:06
MTLR	Мечел ао	196,94	+1,32 %	195	190,63	197,84	194,11	1 843 989 095	23:50:06
AFLT	Аэрофлот	38,97	-0,61 %	39,2	38,38	39,9	39,06	1 729 189 474	23:50:06
ALRS	АЛРОСА ао	68,17	+1,46 %	67,36	66,93	68,78	67,82	1 285 694 636	23:50:06
NVTK	Новатэк ао	1 321,4	+0,29 %	1 317,8	1 306,4	1 326,6	1 317,2	1 278 791 170	23:50:06
PLZL	Полюс	10 382,5	-0,27 %	10 380	10 250	10 437,5	10 359	1 238 584 969	23:50:06
SELG	Селигдар	52,13	+4,49 %	50,14	49,6	53,89	52,55	1 138 771 183	23:50:06
MAGN	ММК	41,175	-0,64 %	41,555	41	41,8	41,295	883 771 040	23:50:06
ROSN	Роснефть	389,75	+0,57 %	387,8	384,55	401	388,55	730 787 270	23:50:06

Рисунок 4 – Данные о ходе торгов с сайта Мосбиржи

Данные, которые находятся в таблице можно описать следующим образом:

- Код – идентификатор финансового инструмента. По нему можно быстро найти финансовый инструмент в терминале брокера;
- Кратк. наим. – краткое наименование ценной бумаги. Указывает на название эмитента, выпустившего ценную бумагу;
- Последняя – цена последней сделки. Торги начинаются в 10:00 по московскому времени и заканчиваются в 23:50 по московскому времени;
- Изм., % – изменение цены последней сделки к последней цене предыдущего дня, указанное в процентах. Цена последней сделки учитывается на момент времени последнего обновления данных;
- Первая – цена первой сделки после начала торговой сессии;
- Мин. – минимальная цена сделки за текущую торговую сессию;
- Макс. – максимальная цена сделки за текущую торговую сессию;

- Ср.взвеш. – средневзвешенная цена. Определяется путем деления суммы произведений цен каждой из сделок на количество ценных бумаг в соответствующей сделке;
- Объем – объем совершенных сделок в валюте расчетов. В нашем случае – это российский рубль;
- Время – время последнего обновления данных о ходе торговой сессии.

Из всех данных, описанных выше, для получения понадобятся: код, краткое наименование, последняя, первая, минимальная, максимальная, объем и время.

Ниже можно увидеть блок-схему алгоритма получения данных с сайта Мосбиржи (Рисунок 5).

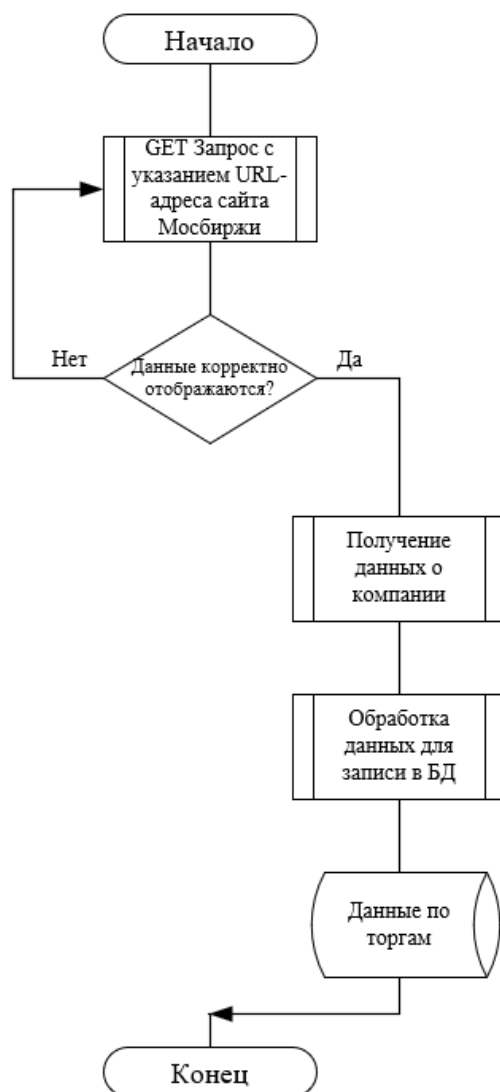


Рисунок 5 – алгоритм получения данных с сайта Мосбиржи.

Программа отправляет GET-запрос на сервер Мосбиржи для получения содержимого страницы. После этого происходит проверка содержимого страницы: в случае, если данные на сайте в виде таблицы не отображаются, что крайне часто бывает, программа снова отправляет GET-запрос для получения данных. Если данные корректно отображены, далее происходит получение этих данных обработчиком, который приводит эти данные к удобному для базы данных виду.

Следующий шаг – сбор таких данных о каждой компании из индекса Мосбиржи за последние 90 торговых дней. Из этих данных считается среднее изменение цены каждой акции за последние 90 дней. Далее собираются данные реального времени раз в 15 минут, то есть сегодняшнего дня. Модуль аналитики смотрит, произошел ли за 15 минут торгов рост/падение цены акции. Если такое происходит, то чат-бот высылает сообщение пользователю.

2.3 Построение БД на основе SQLite

SQLite - это популярная система управления реляционными базами данных (PCУБД) с открытым исходным кодом, которая широко используется во многих приложениях.

Разработчики могут использовать команды SQL для создания, обновления, удаления и запроса данных в таблицах SQLite. SQLite поддерживает подмножество команд SQL (Structured Query Language), которые аналогичны тем, что используются в других реляционных системах баз данных, таких как MySQL, PostgreSQL и Oracle [6].

Одним из ключевых преимуществ SQLite является его небольшой размер. Размер библиотеки SQLite составляет менее 1 МБ, что делает ее идеальной для приложений с ограниченным пространством для хранения данных, например, для мобильных приложений. Несмотря на небольшой размер, SQLite очень быстр и эффективен. Он может обрабатывать большие

наборы данных и сложные запросы без каких-либо проблем с производительностью.

SQLite также очень надежен. Она разработана таким образом, чтобы быть очень устойчивой к сбоям и повреждению данных. Он использует транзакционную модель, что означает, что изменения в базе данных фиксируются только после того, как они были проверены и сохранены на диске. Если во время транзакции происходит отключение электричества или сбой системы, SQLite может автоматически восстановить базу данных до целостного состояния.

SQLite также отличается высокой степенью переносимости. Он может работать на самых разных платформах, включая Windows, macOS, Linux, Android и iOS. Базы данных SQLite можно легко использовать совместно с различными приложениями и платформами, что делает ее популярной среди разработчиков мобильных приложений.

Ниже представлена созданная в SQL база данных приложения (Рисунок 6).

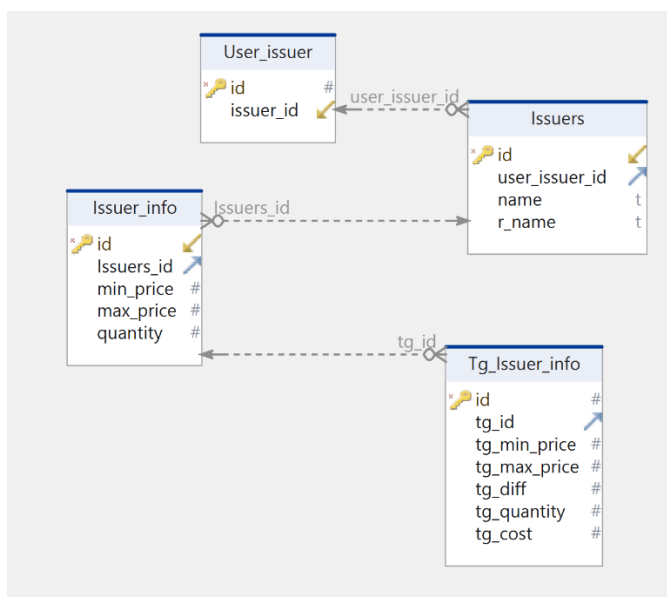


Рисунок 6 – Диаграмма базы данных приложения

Структуру базы данных можно описать следующим набором функций и переменных:

а) Users – данные пользователей, которые мы будем хранить:

1) Id – пользовательский идентификатор, типа данных integer, выдаваемый каждому пользователю, который будет взаимодействовать с ботом;

б) User_Issuer – эмитент, выбранный пользователем. Информация, выдаваемая ботом будет исключительно по этому эмитенту:

1) Id – идентификатор эмитента;

2) User_id – идентификатор пользователя;

в) Issuers – данные эмитентов:

1) Id – идентификатор эмитента;

2) Name – название эмитентов;

3) R_name – тикер эмитентов;

г) Issuer_info – данные эмитентов, собираемые с сайта Мосбиржи:

1) Id – ключ;

2) Issuer_id – идентификатор эмитента;

3) Price – рыночная цена эмитента в сделке;

4) Quantity – количество лотов эмитента в сделке;

д) TG_Issuer_info – данные по сделке, которые пользователь будет получать в чат-боте:

1) Tg_id – идентификатор эмитента;

2) Tg_price – рыночная цена, по которой произошла сделка;

3) Tg_quantity – количество лотов, которые участвовали в сделке;

4) Tg_cost – стоимость всех ценных бумаг эмитента, участвовавших в сделке;

5) Tg_diff – изменение рыночной цены эмитента после совершения сделки, указанная в процентах.

2.4 Описание прецедентов

На рисунке 7 представлена диаграмма прецедентов

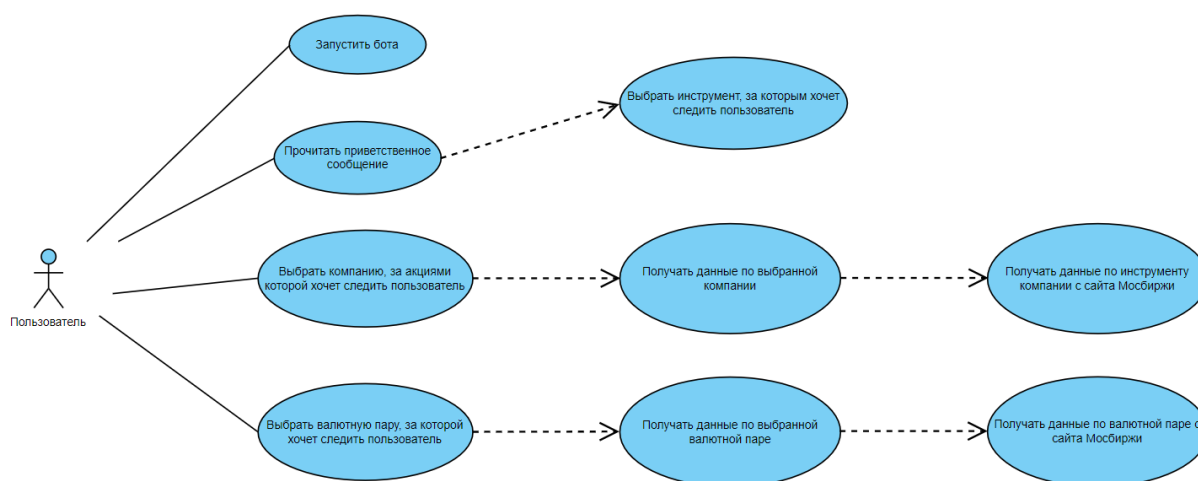


Рисунок 7 – Диаграмма прецедентов

Текстовое описание прецедентов будет следующим.

Прецедент «Запустить бота» активируется командой «/start» и позволяет пользователям ознакомиться с чат-ботом и выбрать дальнейшие действия.

Прецедент «Прочитать приветственное сообщение» включает в себя взаимодействие с чат-ботом. Из этого прецедента пользователь выбирает, за акциями какой компании он собирается следить.

Прецедент «Выбрать инструмент, за которым хочет следить пользователь» активируется после того, как пользователь прочитал приветственное сообщение. На выбор доступно отслеживание валютных пар вида USD/RUB или акции компаний индекса Мосбиржи.

Прецедент «Выбрать компанию, за акциями которой хочет следить пользователь» позволяет пользователю выбрать компанию, акции которой он будет отслеживать. Это происходит после того, как пользователь выбрал вид инструмента в приветственном сообщении. Этот прецедент переходит к прецеденту «Получать данные по выбранной компании», что заставляет чат-бота сформировать GET-запрос ISS-серверу Мосбиржи, в котором будет указано название компании. Прецедент «Получить данные по инструменту

компании с сайта Мосбиржи» – система получает данные с сайта Мосбиржи и формирует ответное сообщение пользователю.

Прецедент «Выбрать валютную пару, за которой хочет следить пользователь» позволяет пользователю выбрать валютную пару, информацию по которой он хочет получать. Это происходит после того, как пользователь выбрал вид инструмента в приветственном сообщении. Этот прецедент переходит к прецеденту «Получать данные по выбранной валютной паре», что заставляет чат-бота сформировать GET-запрос серверу Мосбиржи, в котором будет указано название и идентификатор валютной пары. Прецедент «Получить данные по валютной паре с сайта Мосбиржи»: система получает данные с сайта Мосбиржи и формирует ответное сообщение пользователю.

2.5 Диаграмма классов

Диаграммы классов – графический инструмент моделирования объектно-ориентированных систем и имеют важное значение в различных аспектах разработки программного обеспечения. Они используются для визуализации структуры классов и их взаимосвязей, а также для представления атрибутов и методов, определенных в каждом классе. Является типом диаграммы UML (унифицированный язык моделирования) [7].

Компоненты диаграммы включают в себя классы, атрибуты, методы и связи. Каждый класс представляет собой абстракцию реального или концептуального объекта и определяет его свойства и поведение. Атрибуты описывают состояние объекта, в то время как методы представляют его операции или функциональность.

Классы представлены в виде прямоугольников с именем класса, атрибутами и методами класса. Связи между классами показаны линиями, соединяющих их. Другие элементы, которые могут быть включены в диаграмму классов, включают отношения наследования, которые показаны в виде стрелок, указывающих от производного класса к базовому классу, и интерфейсы, которые показаны в виде кружков с именем интерфейса внутри. Диаграмма классов представлена на рисунке 8.

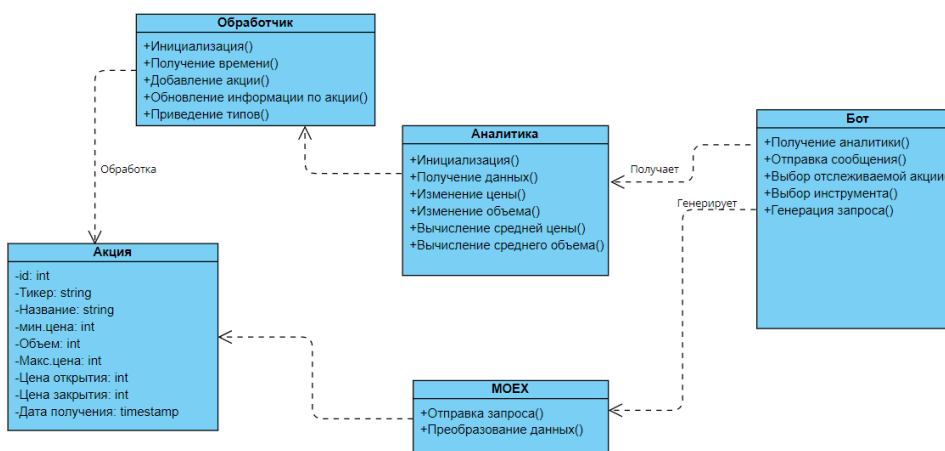


Рисунок 8 – Диаграмма классов

2.6 Вывод по второй главе

В результате выполнения этапа проектирования и разработки была описана модель получения данных с сайта Мосбиржи, их последующая обработка для взаимодействия и сравнения данных с данными реального времени, которая является фундаментом работы.

Подробно рассмотрен основной функционал чат-бота на основе структурной схемы, а также диаграммы прецедентов. Были выделены два вида пользователей, и четко разграничены их возможности при работе с сервисом.

Рассмотрена схема БД со всеми её связями между таблицами, что позволило понять, как приложение ассоциирует процессы.

3 Тестирование и руководство пользователя

3.1 Тестирование работы телеграм-бота

Для того, чтобы начать пользоваться ботом, необходимо написать стартовую команду - /start. После этого пользователю будет доступен выбор для отслеживания из двух инструментов: акции индекса Мосбиржи и валютных пар. Результат продемонстрирован на рисунке 9.

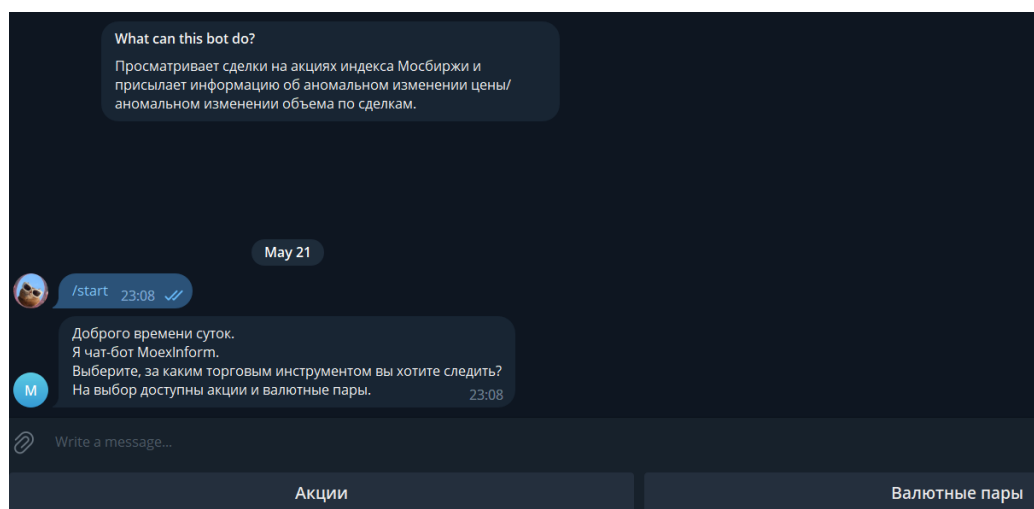


Рисунок 9 – Начало работы с чат-ботом

Выбирая акции, чат-бот покажет список тикеров – специальных идентификаторов финансовых инструментов, в этом случае, с акциями. Список тикеров компаний индекса Мосбиржи показано ниже (Рисунок 10).

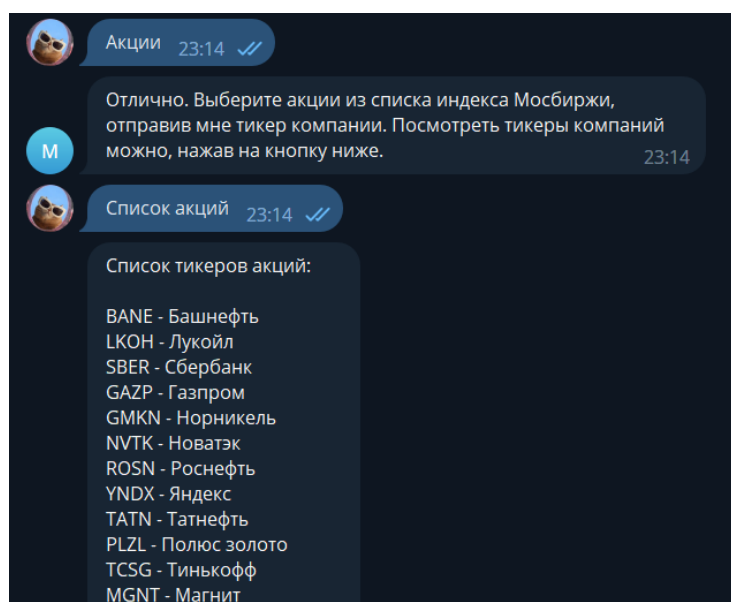


Рисунок 10 – Список тикеров и наименований компаний

Список акций индекса Мосбиржи также можно получить, написав команду – «список». Полный список акций индекса Мосбиржи можно посмотреть в приложении А.

Выбрав компанию, за ходом торгов которой хочет следить пользователь, написав чат-боту ее тикер, пользователь будет получать сообщения, как только чат-бот заметит аномальное изменение цены или аномальный объем. Пример работы такого функционала представлен ниже (Рисунок 11).

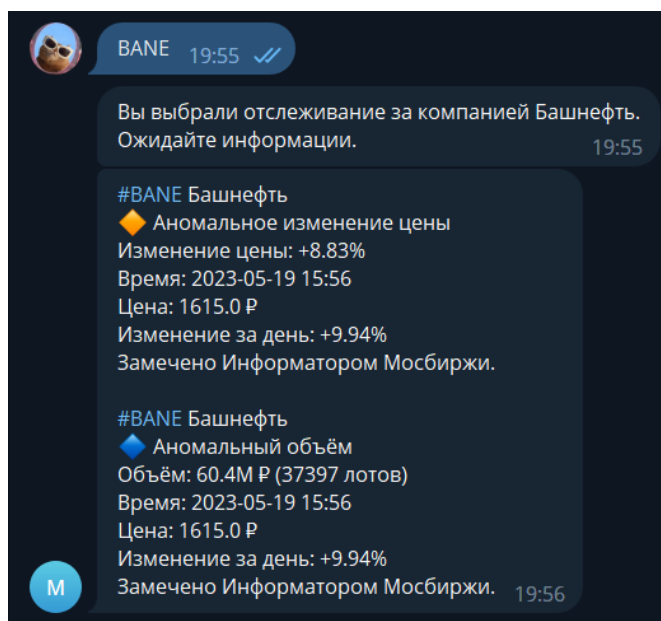


Рисунок 11 – Пример сообщения о торгах чат-бота

Ниже описано, что представляет из себя сообщение, отправленное пользователю чат-ботом и как это работает.

Аномальное изменение цены – программа получает данные с Мосбиржи раз в 15 минут. В случае с аномальной ценой, программа сравнивает полученные только что данные с теми данными, которые были собраны за последние 90 торговых дней. Из прошлых данных программа высчитывает среднее отклонение цены за торговый день, после чего считается среднее отклонение цены от начальной за 90 дней.

В случае, если за 15 минут, цена акции, в момент получения данных, отклоняется от цены открытия больше, чем она делала это за последние 90

торговых дней, пользователю отправляется сообщение об аномальном изменении цены. В этом же сообщении пользователь получит информацию об изменении цены с начала торгов.

Аномальный объем – программа получает данные с Мосбиржи раз в 15 минут. Происходит сравнение полученных данных с данными по последним 90 торговым дням. Высчитывается средний объем сделок, сделанный за каждый день из 90 последних. В случае, если объем сделок, проведенных за 15 минут больше среднего объема, пользователь получает сообщение от чат-бота об аномальном объеме. В этом же сообщении пользователь получит сообщение об изменении цены с начала торгов.

3.2 Интерпретация сообщений телеграм-бота

Может быть 3 вида сообщений: аномальное изменение цены с аномальным объемом сделок, аномальное изменение цены без аномального объема сделок, аномальный объем сделок без аномального изменения цены.

Аномальное изменение цены с аномальным объемом сделок.

Что это может значить: ведется покупка или продажа большим участником рынка. Это может быть покупка или продажа инсайдеров, фондов или крупных частных инвесторов, либо массовое закрытие торговых позиций.

Возможные причины такого поведения участников рынка:

- Инсайдерам стало известно о корпоративном событии, таком как утверждение выплаты дивидендов, хорошая или плохая отчетность;
- Один или несколько участников рынка набирают позицию по бумаге, так как видят в ней потенциал;
- Массовое закрытие или открытие позиций.

Аномальный объем сделок без аномального изменения цены.

Что это может значить: произошла большая сделка на торговой сессии.

Возможные причины таких сделок:

- Сделку совершили покупатель и продавец по предварительной договоренности;
- На торгах появился крупный покупатель, и продавец продал ему большое количество акций;
- На торгах появился крупный продавец, и покупатель купил у него большое количество акций.

Аномальное изменение цены без аномального объема.

Что это может значить: произошла сделка в условиях малого количества участников рынка на торгах.

Возможные причины изменения цены:

- Покупка или продажа произошла, когда на торгах было малое количество участников, поэтому сделка с небольшим объемом привела к значительному изменению цены акции;
- Иногда такое происходит в дни, которые объявлены официальными выходными в стране, но при этом торги не закрывают на время праздничного дня.

3.3 Вывод по третьей главе

В третьей главе была представлена работа телеграм-бота. Показаны сообщения, которые отправляет и получает бот, в том числе сообщения об аномальном поведении акции на торгах.

Также в третьей главе было продемонстрировано руководство пользователя, которое помогает пользователю понять, что могут обозначать сообщения об аномальных движениях в акциях.

ЗАКЛЮЧЕНИЕ

Таким образом, при помощи языка программирования Python, библиотеки для работы с Telegram Aiogram и базы данных SQLite был создан функционирующий чат-бот на базе мессенджера Telegram, отвечающий поставленным задачам и первоначальным задумкам. В процессе работы над чат-ботом удалось разграничить виды инструментов, информацию по которым может давать бот, автоматически отправлять сообщения об аномальных движениях в акциях индекса Мосбиржи и валютных парах.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Московская Биржа | Московская биржа подвела итоги торгов в 2022 году [Электронный ресурс]. URL: <https://www.moex.com/n53917/?nt=106> (дата обращения: 15.03.2023).
2. Московская Биржа | Более 6 млн человек в 2022 году открыли брокерские счета на Московской бирже [Электронный ресурс]. – 2023. URL: <https://www.moex.com/n53950/?nt=106> (дата обращения: 15.03.2023).
3. Индекс Московской Биржи / ММВБ (IMOEX) - состав и компоненты. Сайт блога о трейдинге и инвестировании «sMart-lab.ru» [Электронный ресурс]. – 2023. URL: https://smart-lab.ru/q/index_stocks/IMOEX/ (дата обращения: 12.03.2023).
4. Welcome to aiogram's documentation! — aiogram 2.25.1 documentation [Электронный ресурс]. URL: <https://docs.aiogram.dev/en/latest/> (дата обращения: 12.03.2023).
5. Система менеджмента качества Общие требования к построению, изложению и оформлению документов учебной деятельности. // Стандарт организации СТО СТУ 7.5–07–2021 [Электронный ресурс]. – 2021. URL: <https://about.sfu-kras.ru/docs/8127/pdf/237025> (дата обращения: 19.12.2022).
6. About SQLite | SQLite Documentation [Электронный ресурс]. 2023. URL: <https://www.sqlite.org/about.html> (дата обращения: 05.05.2023).
7. Диаграммы классов UML — Блог программиста [Электронный ресурс]. – 2017. URL: <https://pro-prof.com/archives/3212> (дата обращения: 12.03.2023).
8. Telegram Bot API – Telegram Documentation [Электронный ресурс]. – 2023. URL: <https://core.telegram.org/bots/api> (дата обращения: 12.03.2023).

ПРИЛОЖЕНИЕ А

Список акций индекса Мосбиржи

Таблица А.1 – список компаний индекса Мосбиржи

Тикер компании	Название компании
LKOH	Лукойл
SBER	Сбербанк
GAZP	Газпром
GMKN	Норникель
NVTK	Новатэк
ROSN	Роснефть
YNDX	Яндекс
TATN	Татнефть
PLZL	Полюс золото
TCSG	Тинькофф
MGNT	Магнит
SNGS	Сургутнефтегаз
MTSS	МТС
PHOR	ФосАгро
POLY	Полиметалл
CHMF	Северсталь
IRAO	ИнтерРАО
NLMK	НЛМК
ALRS	Алроса
PIKK	ПИК
FIVE	X5 Retail Group
MOEX	Московская биржа
RUAL	Русал
VTBR	ВТБ
OZON	Озон
MAGN	ММК
RTKM	Ростелеком
TRNFP	Транснефть
CBOM	МКБ
AFKS	АФК Система
VKCO	VK
AFLT	Аэрофлот
HYDR	Русгидро
ENGP	En+
AGRO	Globtrans
FIXP	Fix Price

ПРИЛОЖЕНИЕ Б

Файл «Моех.ру»

```
import json
import requests
from datetime import datetime, timedelta
import pytz
import schedule
import sqlite3 as sl

def querys(company_name):
    queryinitial(company_name)
    while True:
        # Постоянное обновление значений в бд
        schedule.every().day.at("09.30",
"Europe/Moscow").do(update(company_name))

# Первое получение данных с апи и запись в бд
def queryinitial(company_name):
    dates = findDates()

    con = sl.connect("timeSliceData.db")
    cur = con.cursor()

    cur.execute(
        f"""CREATE TABLE IF NOT EXISTS monthData{company_name} (
"date" TEXT NOT NULL UNIQUE,
"name" TEXT NOT NULL,
"low" INTEGER NOT NULL,
"high" INTEGER NOT NULL,
```

```

PRIMARY KEY("date"))
"""
)

try:
    URL =
f"https://iss.moex.com/iss/history/engines/stock/markets/shares/boards/TQBR/secu
rities/{company_name}.json?from={dates[3]}-{dates[4]}-
{dates[5]}&till={dates[0]}-{dates[1]}-{dates[2]}.json"
    response = requests.get(URL)
    data = response.json()["history"]["data"]

    for i in range(len(data)):
        obj = (
            data[i][1],
            company_name,
            data[i][7],
            data[i][8],
        )
        cur.execute(
            f"""INSERT INTO monthData{company_name} VALUES(?, ?, ?,
?);""", obj
        )
        con.commit()
    except Exception as e:
        print(f"{e}")

def update(company_name):
    # Код для обновления значений в бд
    try:

```

```

URL_NOW =
f"https://iss.moex.com/iss/engines/stock/markets/shares/boards/TQBR/securities/{
company_name}.json"
    response = requests.get(URL_NOW)
    except Exception as e:
        print(f"{e}")

# Нахождение нужной даты от которой вести отсчет
def findDates():
    start_date = datetime.now(pytz.timezone("Europe/Moscow"))
    start_day = start_date.day
    start_month = start_date.month
    start_year = start_date.year

    start_date = datetime(start_year, start_month, start_day)
    end_date = start_date - timedelta(days=30)
    end_day = end_date.day
    end_month = end_date.month
    end_year = end_date.year

    return [start_year, start_month, start_day, end_year, end_month, end_day]

queryinitial("SBER")

```

ПРИЛОЖЕНИЕ В

Файл «Analytics.py»

```
from datetime import datetime, timedelta
import pandas as pd
from inc.Db import Db

class Analytics:
    def __init__(self, db : Db):
        self.df = db.get_df()
        self.df['matdays'] = self.df['matdate'] - datetime.now()

    def get_main_stats(self):
        date_2022 = datetime.strptime("2022-01-01", "%Y-%m-%d")
        date_2021 = datetime.strptime("2021-01-01", "%Y-%m-%d")
        date_2020 = datetime.strptime("2020-01-01", "%Y-%m-%d")
        date_2019 = datetime.strptime("2019-01-01", "%Y-%m-%d")
        delta365 = timedelta(days=365)
        df = self.df

        stats = {
            'всего облиг' : len(df.index),
            'торгуемых' : len(df[ df['is_traded'] == 1].index),
            'для квалов' : len(df[ df['isqualifiedinvestors'] == True].index),

            'выпущенных в 2021' : len(df[ (df['issuedate'] >= date_2021) &
            (df['issuedate'] <= date_2022)].index),
            'выпущенных в 2020' : len(df[ (df['issuedate'] >= date_2020) &
            (df['issuedate'] <= date_2021)].index),
```

'выпущенных в 2019' : len(df[(df['issuedate'] >= date_2019) & (df['issuedate'] <= date_2020)].index),

'с доходностью > 1%' : len(df[df['effectiveyield'] >= 1].index),

'с доходностью > 8%' : len(df[df['effectiveyield'] >= 8].index),

'с доходностью > 11%' : len(df[df['effectiveyield'] >= 11].index),

'листинг 1' : len(df[(df['is_traded'] == 1) & (df['listlevel'] == 1)].index),

'медианная доходность, листинг 1, %' : round(df[(df['is_traded'] == 1) & (df['listlevel'] == 1)]['effectiveyield'].median(),2),

'листинг 2' : len(df[(df['is_traded'] == 1) & (df['listlevel'] == 2)].index),

'медианная доходность, листинг 2, %' : round(df[(df['is_traded'] == 1) & (df['listlevel'] == 2)]['effectiveyield'].median(),2),

'листинг 3' : len(df[(df['is_traded'] == 1) & (df['listlevel'] == 3)].index),

'медианная доходность, листинг 3, %' : round(df[(df['is_traded'] == 1) & (df['listlevel'] == 3)]['effectiveyield'].median(),2),

'медианная цена, %' : round(df['price'].mean(),2),

'медианная цена, с дох >= 11, %' : round(df[df['effectiveyield'] >= 11]['price'].median(),2),

'медианная цена, с дох >= 8 & < 11, %' : round(df[(df['effectiveyield'] >= 8) & (df['effectiveyield'] < 11)]['price'].median(),2),

'медианная цена, с дох >= 1 & < 8, %' : round(df[(df['effectiveyield'] >= 1) & (df['effectiveyield'] < 8)]['price'].median(),2),

'медианная цена, листинг 1, %' : round(df[(df['is_traded'] == 1) & (df['listlevel'] == 1)]['price'].median(),2),

'медианная цена, листинг 2, %' : round(df[(df['is_traded'] == 1) & (df['listlevel'] == 2)]['price'].median(),2),

'медианная цена, листинг 3, %' : round(df[(df['is_traded'] == 1) & (df['listlevel'] == 3)]['price'].median(),2),


```

        'медианная цена, matday < 365, %' : round(df[df['matdays'] <
delta365]['price'].median(),2),
        'медианная доходность, matday < 365, %' : round(df[df['matdays'] <
delta365]['effectiveyield'].median(),2),
    }
    return stats

def report_lowest_price(self, min_normal=90):
    return self.df[self.df['price'] < min_normal].sort_values(by=['effectiveyield'],
ascending=False)

def report_365_cheap_1121(self):
    delta365 = timedelta(days=365)
    df = self.df[ (self.df['is_traded'] == 1) & (self.df['listlevel'] == 2) &
(self.df['matdays'] < delta365)]
    med = df['price'].median()
    return df[df['price'] < med].sort_values(by=['price'], ascending=True)

def report_365_yielddest(self):

    delta365 = timedelta(days=365)
    df = self.df[(self.df['matdays'] < delta365) & (self.df['listlevel'] <= 2)]
    med = df['effectiveyield'].median()
    return df[df['effectiveyield'] > med].sort_values(by=['effectiveyield'],
ascending=False)

```

ПРИЛОЖЕНИЕ Г

Файл «Db.py»

```
from datetime import datetime, timedelta
from importlib import resources
from sqlite3 import create_engine, func, desc, and_, or_
from sqlite3 import sessionmaker

from inc.Models import
import pandas as pd

class Db:
    def __init__(self):
        with resources.path("_db", "db.db") as path:
            engine = create_engine(f"sqlite:/// {path}")
            _session = sessionmaker()
            _session.configure(bind=engine)
            self.session = _session()

    def get_df(self):
        return pd.read_sql(self.session.query(Bond).statement, self.session.bind)

    def add_bond(self, j):
        """
        Добавляю новую акцию
        или обновляю ту что уже в базе
        :param j:
```

```

:return:
"""
o = self.session.query(Bond).filter_by(secid=j['secid']).first()
if not o: o = Bond()

o.from_json(j)
self.session.add(o)

def update_bond_from_json(self, bond : Bond, j:dict):
    """
    Обновление облиги
    запись спеков и доходностей
    :param bond:
    :param j:
    :return:
    """
    bond.from_json(j)
    bond.updated = datetime.now()
    self.session.add(bond)

def get_random_bond(self) -> Bond:
    return
self.session.query(Bond).filter_by(is_traded=True).order_by(func.random()).first()

def get_next_bond(self, seconds = 18000 ) -> Bond:
    before = (datetime.now() - timedelta(seconds=seconds))
    return self.session.query(Bond).filter(and_( or_(Bond.updated == None,
Bond.updated < before), Bond.is_traded ==
True)).order_by(desc(Bond.updated)).first()

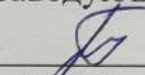
```

Министерство науки и высшего образования РФ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий
институт
Вычислительная техника
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой


подпись О. В. Непомнящий

« 22 » 06 2023г.

БАКАЛАВРСКАЯ РАБОТА

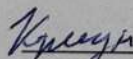
09.03.01 – Информатика и вычислительная техника

код – наименование направления

Телеграм-бот «Мосбиржа-информатор»

тема

Руководитель


подпись, дата

доцент кафедры ВТ

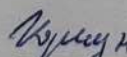
К.В. Коршун

Выпускник


подпись, дата

С.А. Ману

Нормоконтролер


подпись, дата

доцент кафедры ВТ

К.В. Коршун

Красноярск 2023