

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

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ О.В. Непомнящий  
«\_\_» \_\_\_\_\_ 2024 г.

**БАКАЛАВРСКАЯ РАБОТА**

090301 Информатика и вычислительная техника

Анализатор биржевого рынка для Telegram Bot

Руководитель	_____	_____	старший преподаватель	С.Л. Верхошенцева
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		П.С. Соин
	<i>подпись</i>	<i>дата</i>		
Нормоконтролёр	_____	_____	старший преподаватель	С.Л. Верхошенцева
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	

Красноярск 2024

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

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ О.В. Непомнящий

« \_\_\_\_ » \_\_\_\_\_ 2023 г.

**ЗАДАНИЕ  
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ  
в форме бакалаврской работы**

Красноярск 2023

Студенту Соину Павлу Сергеевичу  
фамилия, имя, отчество

Группа КИ20-08Б Направление (специальность) 090301  
номер код

Информатика и вычислительная техника  
наименование

Тема выпускной квалификационной работы: Анализатор биржевого рынка для Telegram Bot

Утверждена приказом по университету № 4858/С от 18.03.2024

Руководитель ВКР: С.Л. Верхошенцева, старший преподаватель кафедры ВТ  
инициалы, фамилия, учёная степень, должность, место работы

ИКИТ СФУ

Исходные данные для ВКР:

- 1) Провести анализ существующих решений.
- 2) Разработать функционал и интерфейс чат-бота.
- 3) Реализовать получение актуальных данных с сайта Мосбиржи.
- 4) Разработать механизм предоставления прогнозов по ценам акций.
- 5) Провести тестирование разработанного чат-бота.

Перечень разделов ВКР:

- 1) Анализ предметной области.
- 2) Проектирование чат-бота
- 3) Разработка и тестирование чат-бота

Перечень графического материала: Презентация в формате Power Point

Руководитель ВКР С.Л. Верхошенцева  
подпись инициалы, фамилия

Задание принял к исполнению П.С. Соин  
подпись инициалы, фамилия

«20» 12 2023 г.  
дата

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Анализатор биржевого рынка для Telegram Bot» содержит 48 страниц текстового документа, 26 рисунков, 1 таблицу и 17 использованных источников.

АНАЛИЗ ДАННЫХ, ПРОГНОЗИРОВАНИЕ, SARIMA, ЧАТ-БОТ.

Цель работы – разработка инструмента для доступа к актуальным финансовым данным, аналитической информации и прогнозов по ценам акций на Московской бирже.

Задачи, решенные в процессе разработки:

- провести анализ существующих решений;
- разработать функционал и интерфейс чат-бота;
- реализовать получение актуальных данных с сайта Мосбиржи;
- разработать механизм предоставления прогнозов по ценам акций;
- провести тестирование разработанного чат-бота.

В первой главе был проведен анализ предметной области, рассмотрены основные концепции по функционированию и взаимодействию с биржей, переход к цифровому типу торговли на бирже, современные инструменты анализа рыночных данных, а также был проведен анализ мессенджеров как платформы для взаимодействия с пользователями.

Во второй главе был выбран язык разработки, составлены основные требования к разрабатываемому чат-боту, построены диаграмма прецедентов и диаграммы последовательностей, описывающие функционал разрабатываемого чат-бота.

В третьей главе рассмотрены метод загрузки исторических данных и реализация базы данных, процесс построения прогнозирования при помощи модели SARIMA, а также произведена разработка и тестирование чат-бота.

В результате выпускной квалификационной работы был разработан функциональный чат-бот на базе мессенджера Telegram для получения ознакомительной информации и прогноза по акциям, торгуемым на московской бирже.

## СОДЕРЖАНИЕ

Введение.....	3
1. Анализ предметной области .....	5
1.1. Цифровая биржа .....	6
1.2. Существующие аналоги.....	8
1.2.1. Инструменты торговли у брокеров .....	8
1.2.2. Мессенджеры.....	9
1.3. Выводы по главе 1 .....	11
2. Проектирование чат-бота.....	12
2.1. Выбор инструментов для реализации .....	12
2.1.1. Конструкторы .....	12
2.1.2. Язык программирования Python .....	12
2.1.3. Среда разработки.....	13
2.2. Спецификация требований к разрабатываемому чат-боту .....	13
2.3. Разработка UML диаграмм.....	14
2.3.1. Диаграмма прецедентов.....	15
2.3.2. Диаграмма последовательностей.....	16
2.4. Выводы по главе 2 .....	20
3. Разработка и тестирование чат-бота .....	21
3.1. Реализация базы данных и метод загрузки исторических данных .....	21
3.2. Модель для прогноза.....	24
3.3. Создание чат-бота.....	29
3.4. Тестирование работы чат-бота в Telegram.....	31
3.5. Выводы по главе 3 .....	36
Заключение .....	37
Список использованных источников .....	38
ПРИЛОЖЕНИЕ А predict_function.py.....	40
ПРИЛОЖЕНИЕ Б handlers.py .....	44
ПРИЛОЖЕНИЕ В main.py .....	48

## ВВЕДЕНИЕ

Торговля на бирже является одной из основ экономики. Ежедневно происходит огромное количество сделок с различными активами. Суммарно объем торгов за 2023 год составил 1,3 квадрилн рублей, что на 0,2 квадрилн рублей больше, чем в 2022 году, что является максимальным показателем за всю историю биржевых торгов [1].

Инвестирование и биржевая торговля активно набирает популярность среди физических лиц. Чем больше участников, участвующих в торгах, тем больше происходит изменение цены на различные активы. Трейдинг – деятельность участника финансового рынка, в ходе которого он совершает торговые операции, направленные на извлечение прибыли [2]. В связи с тем, что все больше людей хотят иметь дополнительный заработок, многие из них обращаются именно к трейдингу. Но для извлечения прибыли из данной деятельности нужны некоторые знания в области экономики и финансовой грамотности, что не является доступным для всех людей.

Не у каждого есть возможность для самостоятельного поиска новостей и для этого многие брокеры публикуют финансовые новости в своих социальных сетях. На сегодняшний день наиболее популярной социальной сетью является Telegram. Он занимает первое и второе место в рейтинге приложений для связи в Google Play и App Store соответственно, а значит на данную социальную сеть делается упор при публикации различных новостей.

В связи с этим, в рамках работы была поставлена цель: разработать инструмент для доступа к актуальным данным о финансовых инструментах, аналитической информации и прогнозов по ценам акций, валюты и другим финансовым инструментам на Московской бирже.

Для достижения обозначенной цели были поставлены следующие задачи:

1. Провести анализ существующих решений.
2. Разработать функционал и интерфейс чат-бота.

3. Реализовать получение актуальных данных с сайта Мосбиржи.
4. Разработать механизм предоставления прогнозов по ценам акций.
5. Провести тестирование разработанного чат-бота.

Ожидаемым результатом работы является разработанная система в мессенджере в виде чат-бота, способная предоставлять актуальную информацию об изменениях цен на акции, а также осуществлять прогнозирование на основе исторических данных. Такой инструмент может быть полезен для упрощения работы с биржей как профессиональным инвесторам, так и начинающим. Опираясь на полученные данные прогноза, можно анализировать стоимость ценных бумаг и своевременно покупать, либо продавать их.

## 1. Анализ предметной области

В первую очередь необходимо разобраться с понятием «Биржа», ее вариациями и как она работает. Биржа – это площадка, на которой люди заключают сделки и производят обмен деньгами, товарами, ценными бумагами или контрактами. Существует несколько типов бирж: товарная, валютная, фондовая и другие. Мы будем рассматривать фондовую биржу или же фондовый рынок. Фондовый рынок, по своей сути, похож на обычный продуктовый рынок, в котором есть как спрос, так и предложение, соответственно, чем выше спрос, тем выше предложение, и наоборот.

Важной особенностью биржи является официальность и организованность. Весь процесс устроен таким образом, чтобы торговля велась по правилам, поэтому к торгам допускаются лишь профессиональные участники фондовой биржи:

1. Управляющие компании. Такая компания совершает сделки от своего имени. Существует два варианта работы с управляющими компаниями: когда компания работает согласно стратегии клиента для получения дохода, либо когда инвестор вкладывается в паевые инвестиционные фонды, которые инвестируют в большое количество различных активов, что помогает получить оптимальную активность с минимальными рисками [3].

2. Брокер. Брокером также является компания, но она совершает сделки на рынке от имени клиента и на его средства. Работая с брокером, инвестор составляет свой инвестиционный портфель самостоятельно. Брокер выполняет пожелания клиента и берет за это свою комиссию [3].

3. Государственный регулятор. Наблюдает за работой управляющих компаний и брокеров. Необходим для соблюдения ряда требований, таких как: регулярная публикация отчета о сделках и финансовом положении, а также набор только квалифицированного персонала.

## 1.1. Цифровая биржа

Раньше биржи находились в специальных помещениях, где встречались представители продавцов и покупателей и договаривались о сделках. На сегодняшний день, большая часть торгов происходит в цифровом формате, где нет необходимости ни продавцу, ни покупателю встречаться лично. Все операции можно совершать из любой точки мира, а для сделки можно использовать мобильный телефон, компьютер или другие цифровые устройства [4].

Пример графика для отслеживания изменения цен на активы на примере ценной бумаги «Лукойл» (LKOH) представлен на рисунке 1.

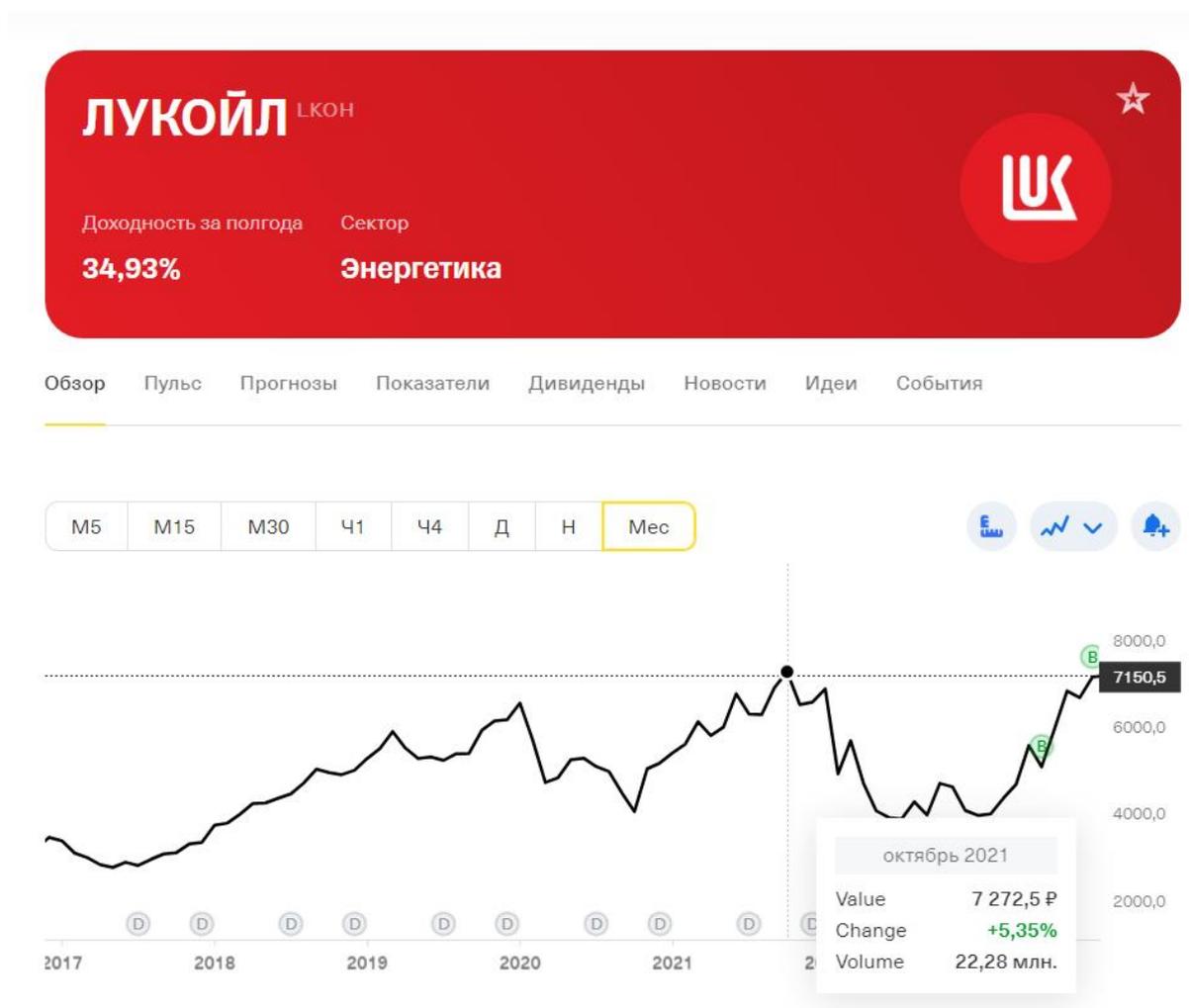


Рисунок 1 – Пример страницы ценной бумаги

Также данные о цене актива можно представлять посредством таблиц в Microsoft Excel или Google Sheets, но при таком отображении таблицу придется создавать вручную с использованием API или HTTP-запросов. Пример представления данных в Google Sheets представлен на рисунке 2 [5].

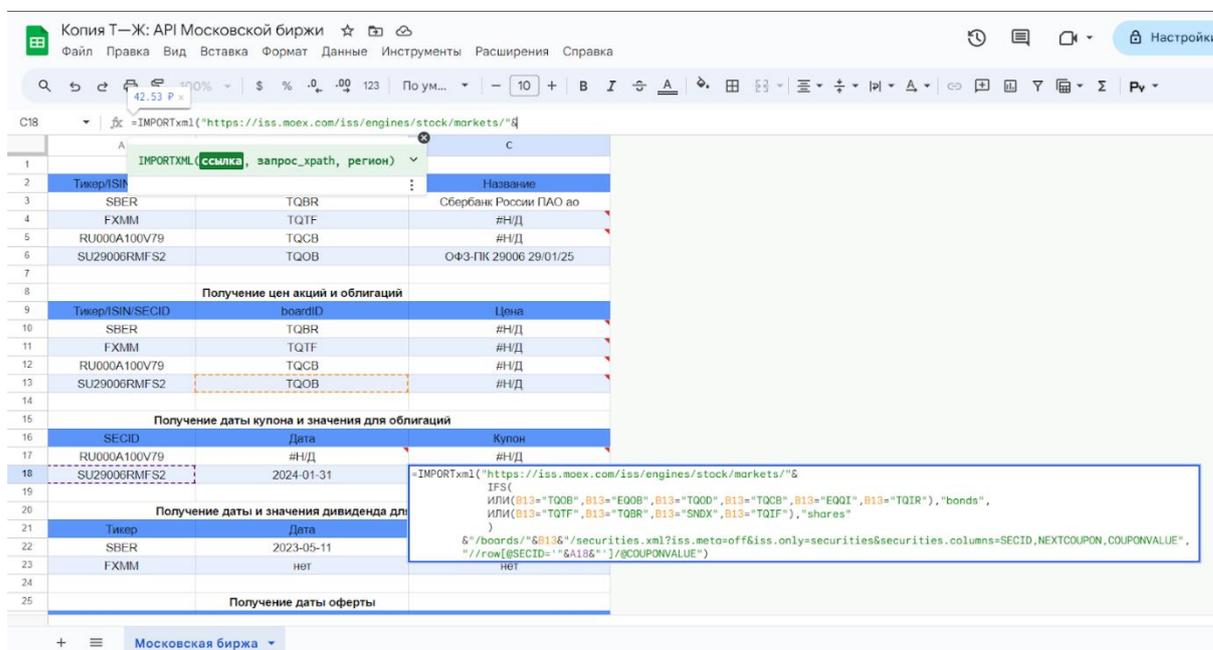


Рисунок 2 – Представление отслеживания ценных бумаг в Google Sheets

Для самостоятельного анализа и представления графиков с котировками рынка пользователь может использовать различные площадки, но лучше всего использовать официальную площадку, через которую проходят торги различных активов. В нашей стране официальной площадкой, для проведения операций с ценными бумагами, является Мосбиржа.

Московская биржа – крупнейший российский биржевый холдинг, который является организатором торгов акциями российских компаний и стран СНГ, облигациями, производственными инструментами, валютой, инструментами денежного рынка, углеродными единицами, драгоценным металлом, зерном и сахаром [6].

С сайта Мосбиржи можно получать огромное количество данных, таких как: список торгуемых бумаг, спецификации инструмента, обобщенные данные ОТС ПФИ и РЕПО, и другие данные.

## **1.2. Существующие аналоги**

В мире информационных технологий все чаще обращаются к автоматизированным анализаторам. С их помощью можно обрабатывать большой объем данных и делать более точные прогнозы используя технологии искусственного интеллекта.

Автоматизированные анализаторы дают возможность инвесторам следить за рыночными тенденциями и получать рекомендации оптимального времени покупки, продажи позиции. Это позволяет сделать процесс инвестирования доступным и понятным для человека, не имеющего глубоких знаний в сфере финансов.

### **1.2.1. Инструменты торговли у брокеров**

Существует множество различных инструментов для торговли ценными бумагами. Самыми популярными являются следующие инструменты:

1. Экспертные советники. Программы, написанные для торговли на платформе MetaTrader. С их помощью возможно проводить технический анализ, определять сигналы входа и выхода из актива.

2. Автоматизированные инвестиционные роботы. Встроенные инструменты для автоматизации торговли, например робот от банка «ВТБ» [7]. Автоматически управляет инвестиционным портфелем и оптимизирует его. Минусом такого робота является то, что все сделки происходят автоматически без уведомления инвестора, что может как увеличить стоимость портфеля, так и уменьшить.

3. Системы копирования сделок. Такие системы автоматически копируют сделки более опытных инвесторов, подбирая количество бумаг под ваши свободные деньги. Главным минусом такой системы является ежедневная небольшая комиссия за копирование сделок.

Использование вышеперечисленных платформ не всегда является доступным, так как их использование возможно только на персональном компьютере или ноутбуке. Человек в наше время стал более мобильным, а следовательно персональный компьютер или ноутбук не всегда есть под рукой. Несмотря на это, у большинства людей есть мобильные телефоны, с возможностью использования мессенджеров. Именно поэтому было предложено разработать инструмент для анализа и прогнозирования данных биржевого рынка для мобильного устройства.

### **1.2.2. Мессенджеры**

Концепция взаимодействия большого количества людей, передача информации, личной переписки ежесекундно, с помощью одного приложения или устройства, уходит своими корнями в разработки прошлого века и находит свое отражение в мессенджерах. Мессенджер – это программа для мгновенного обмена текстовыми сообщениями и мультимедиа между зарегистрированными пользователями через интернет [8].

Мессенджеры и социальные сети занимают первое место среди сервисов, используемых людьми в интернете. Данные GWI свидетельствуют о том, что люди сегодня проводят в соцсетях больше времени, чем когда-либо. По результатам последних исследований, среднестатистический интернет-пользователь тратит чуть больше 2,5 часов в день на социальные сети, а на мессенджеры приходится самая большая доля времени в интернете (4 минуты из 10) [9].

Существуют разные виды мессенджеров, некоторые из них рассчитаны на определенную аудиторию, задачу или функцию. Самыми распространёнными в России является WhatsApp, Viber и Telegram. Далее рассмотрим каждый из них.

WhatsApp – это общепризнанный, не теряющий своей популярности мессенджер. Позволяет отправлять текстовые и голосовые сообщения,

создавать беседы, а также совершать видео и аудио звонки. Также доступна версия WhatsApp Business, которая предназначена для бизнес-маркетинга. С помощью такого приложения можно запускать рассылки, настраивать автоматическую отправку ответов на часто задаваемые вопросы. В России WhatsApp пользуется спросом среди людей среднего возраста. К основным недостаткам можно отнести то, что отсутствуют некоторые функции, которые есть у конкурентов. Кроме того, данное приложение имеет принадлежность к компании Meta, которая признана экстремистской и запрещена в РФ [10].

Viber является не менее популярным мессенджером. Основными пользователями данного приложения являются люди среднего возраста и старше. Оно предлагает стандартные функции – звонки, конференции, текстовые сообщения, обмен медиа контентом и документами. Основным недостатком является то, что периодически возникают технические неполадки. Например, при работе с приложением Viber происходят «вылеты», что вынуждает пользователя заново открывать приложение и проходить повторную авторизацию. Кроме того, наблюдаются проблемы с уведомлениями.

Telegram – самая популярная площадка среди молодого населения страны. Приложение с почти неограниченной коллекцией функций для взаимодействия с миром вокруг, а именно звонки и видеоконференции, обмен фотографиями и видеоматериалами. Мессенджер с легкостью подстраивается под любые запросы аудитории. К недостаткам можно отнести доступность некоторых функций по платной подписке. Однако в приложении в открытом доступе находятся чат боты, которые способны выполнять команды и решать определенные задачи. Мессенджер способен заменить браузер и сэкономить время на выполнении рутинных задач [11].

Исходя из всего вышеизложенного, при разработке собственного биржевого анализатора для взаимодействия с пользователем, предлагается использовать мессенджер Telegram, так как данным приложением пользуется большое количество людей. Кроме того, в нем простой, удобный и понятный

интерфейс, как для мобильного телефона, так и для персонального компьютера.

### **1.3. Выводы по главе 1**

В данной главе были сформулированы основные понятия, проанализированы существующие аналоги и возможные площадки для реализации поставленной задачи, в рамках выпускной квалификационной работы.

Основой проекта послужит Telegram Bot с получением данных с сайта Московской биржи. С помощью чат-бота пользователь сможет узнать текущую стоимость актива, просматривать историю цены на актив, а также получать прогноз об изменениях в стоимости ценных бумаг.

## **2. Проектирование чат-бота**

### **2.1. Выбор инструментов для реализации**

Для реализации данного проекта необходимо выбрать язык программирования и инструмент для написания кода. Чат-бот Telegram может быть реализован множеством способов, в том числе и без программирования с помощью конструкторов, но такой метод сильно ограничен в функционале.

#### **2.1.1. Конструкторы**

Создание ботов при помощи конструкторов значительно облегчает разработку, но функциональность таких решений крайне ограничена. Чат-боты, созданные на конструкторах чаще всего, подойдут для кафе и продуктовых магазинов с доставкой еды, магазинов одежды, служб технической поддержки. Но если есть задачи, которые нельзя решить при помощи конструкторов, лучше написать функционал самостоятельно [12].

#### **2.1.2. Язык программирования Python**

Python – высокоуровневый язык программирования, динамически типизированный с автоматическим управлением памятью, созданный для повышения производительности разработчика, улучшения читаемости и качества самого кода [13].

Данный язык является одним из самых популярных среди разработчиков. Он используется в различных сферах деятельности начиная от математических вычислений и веб-интерфейсов, заканчивая разработкой игр и машинным обучением [13].

Также Python имеет огромное количество библиотек, над созданием которых трудились, как и энтузиасты, так и крупные компании. Например, библиотеки NumPy и SciPy созданные энтузиастами, необходимы для работы

с числами и сложными математическими операциями. Библиотеки TensorFlow и Keras были созданы компанией Google для машинного обучения и создания искусственного интеллекта [14].

Исходя из вышеперечисленного, Python будет отличным языком для реализации проекта.

### **2.1.3. Среда разработки**

В связи с тем, что для разработки проекта был выбран язык Python, то для разработки необходимы IDE с поддержкой именно этого языка. Самыми популярными IDE для Python являются VS Code и PyCharm.

Visual Studio Code является бесплатной открытой и легкой IDE, предназначенной для различных языков программирования. Свою популярность получил за счет легкости, настраиваемому интерфейсу, легкому дизайну и большим библиотекам расширений.

PyCharm имеет более сложный интерфейс с большим количеством различных панелей, таких как быстрое переключение между проектами, быстрый доступ к терминалу, консоли языка программирования, а также к окну управления версиями. К тому же данный IDE был разработан специально для языка Python, что дает более мощный инструмент для разработки на Python.

Сравнив все «за» и «против» обоих IDE можно сделать вывод, что PyCharm выглядит более интересной средой разработки за счет большей адаптации для языка Python [15].

## **2.2. Спецификация требований к разрабатываемому чат-боту**

Существует три основных метода взаимодействия с ботом

1. Сообщения в чат. Пользователь вводит сообщение, бот принимает это сообщение и отвечает заготовленным действием;

2. Клавиатура под полем ввода. Сообщения в чате, с заранее заготовленным запросом;

3. Инлайн-клавиатура. Данный вид клавиатуры является набором функциональных кнопок под сообщениями бота, при нажатии на которые, бот будет реагировать в соответствии с заготовленным действием, связанным с данной кнопкой.

Оптимальное взаимодействие с ботом возможно путем сочетания сообщений от пользователя и инлайн-клавиатуры. Ручная отправка сообщения чат-боту дает возможность получить краткую информацию об активе. После чего пользователь может воспользоваться инлайн-клавиатурой для выбора дальнейших действий:

- получить прогноз об изменении цены: при нажатии данной кнопки пользователю отобразится график с возможным изменением цены на бумагу;
- получить график изменения цен: при нажатии этой кнопки пользователь увидит график с историческими данными о цене актива за последний месяц с среднесуточной частотой изменения цены.

### **2.3. Разработка UML диаграмм**

UML (Universal Modeling Language) – универсальный язык моделирования, необходимый для создания описания взаимодействия как предметной области, так и конкретной задачи. Визуальное моделирование в UML можно представить как процесс роста от общей идеи и абстрактной модели, до логической, а в последующем и физической модели.

Диаграмма – один из способов графического представления множества элементов [16].

### 2.3.1. Диаграмма прецедентов

Диаграмма прецедентов – диаграмма поведения, на которой показано множество прецедентов и актеров, а также отношение между ними [16]. На рисунке 3 представлена диаграмма прецедентов разрабатываемого чат-бота.



Рисунок 3 – Диаграмма прецедентов

В таблице 1 представлено описание блоков диаграммы прецедентов.

Таблица 1 – Описание прецедентов

Название прецедента	Предусловие	Основной сценарий	Альтернативный сценарий
Запустить бота	Пользователь открыл диалог с чат-ботом и ввел команду «/start» или нажал кнопку «Запустить».	В чат пользователю выводится сообщение с описанием бота.	Нет
Прочитать приветственное сообщение	Пользователь начал диалог с чат-ботом.	Пользователю выводится приветственное сообщение с краткой информацией о возможностях чат-бота, описание принципа взаимодействия и предложение прочитать полное описание по взаимодействию с чат-ботом.	Нет

## Окончание таблицы 1

Название прецедента	Предусловие	Основной сценарий	Альтернативный сценарий
Вызвать справку о работе с чат-ботом	Пользователь начал диалог с чат-ботом.	Пользователю выводится полное описание по взаимодействию с чат-ботом.	Нет
Выбрать ценную бумагу	Пользователь начал диалог с чат-ботом.	Пользователь вручную вводит тикер необходимого актива и получает краткую информацию о выбранном инструменте.	Пользователь ничего не ввел.
Получить прогноз об изменении цена на актив	Пользователь выбрал интересующий его инструмент.	Пользователю выводится график с возможным изменением цены на инструмент.	Пользователь ничего не выбрал.
Получить график с изменением цен	Пользователь выбрал интересующий его инструмент.	Пользователю выводится график с историческими данными за последний месяц.	Пользователь ничего не выбрал.

### 2.3.2. Диаграмма последовательностей

Диаграмма последовательностей – диаграмма поведений, на которой показано взаимодействие и структурная организация объектов, посылающих и принимающих сообщения [16]. В данном разделе описаны диаграммы последовательности, которые показывают наиболее значимые процессы в системе. На рисунке 4 изображена диаграмма последовательности для варианта использования «Запуск бота», представляющая процесс пользователя с чат-бота.

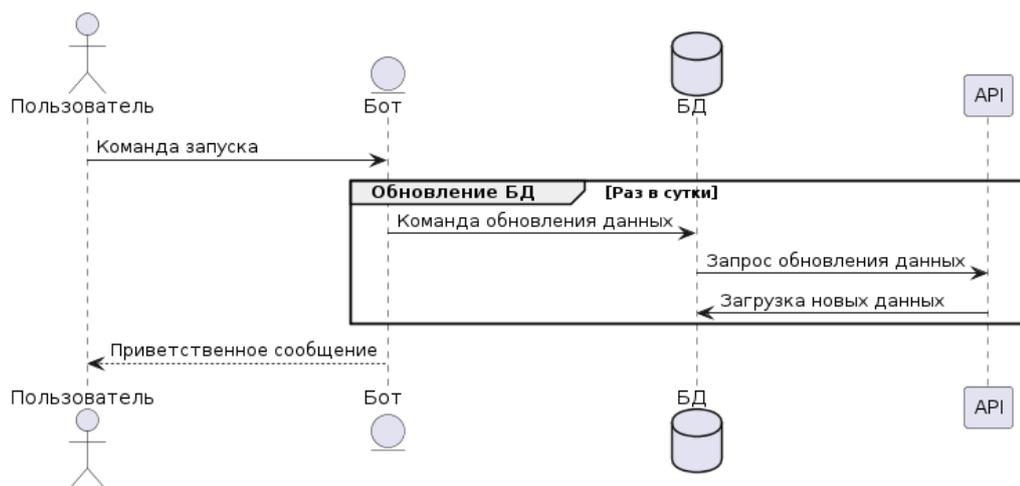


Рисунок 4 – Диаграмма последовательности «Запуск бота»

Пользователь нажимает кнопку «Запустить» или вводит «/start», после чего происходит запуск диалога между пользователем и чат-ботом. Подтверждением запуска бота является приветственное сообщение с кратким описанием о взаимодействии с ботом.

На рисунке 5 приведена диаграмма последовательности для варианта использования «Вызвать справку по работе бота» представляющая процесс вывода полного описания по взаимодействию с чат-ботом.

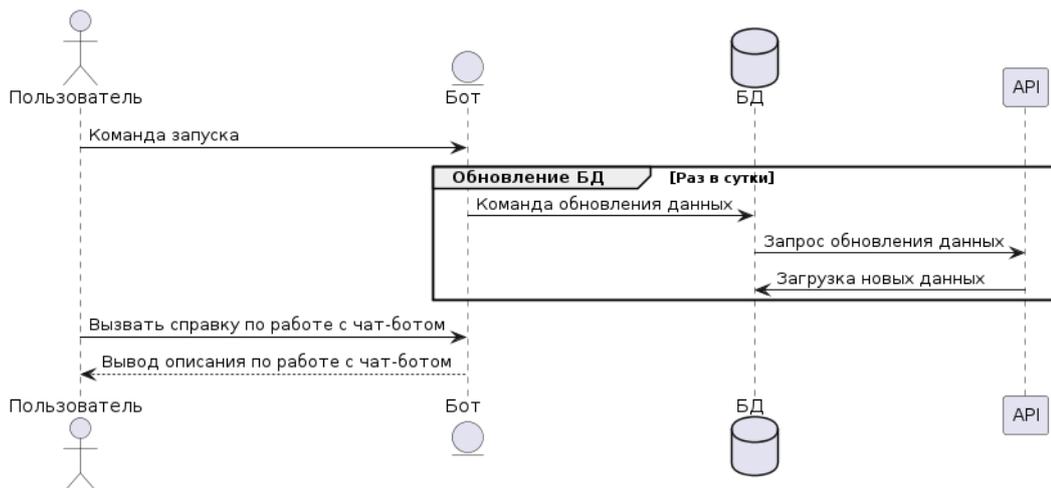


Рисунок 5 – Диаграмма последовательности «Вызвать справку по работе с ботом»

Пользователь вводит команду «/help» или нажимает на кнопку «Что могу?», после чего ему выводится сообщение с полным описанием по возможностям взаимодействия с чат-ботом.

На рисунке 6 изображена диаграмма последовательности «Ввод названия инструмента», представляющая процесс запроса данных об интересующем инструменте.

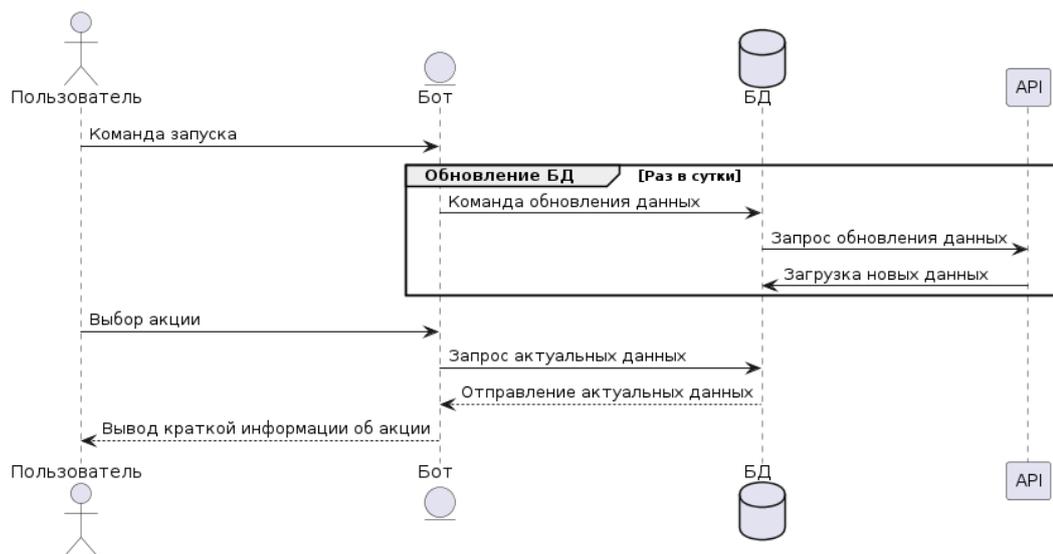


Рисунок 6 – Диаграмма последовательности «Ввод названия инструмента»

Выбор инструмента происходит за счет ввода пользователем тикера инструмента, после чего пользователю выводится краткая информация об активе.

На рисунке 7 представлена диаграмма последовательности, описывающая процесс «Получить прогноз об изменении цены».

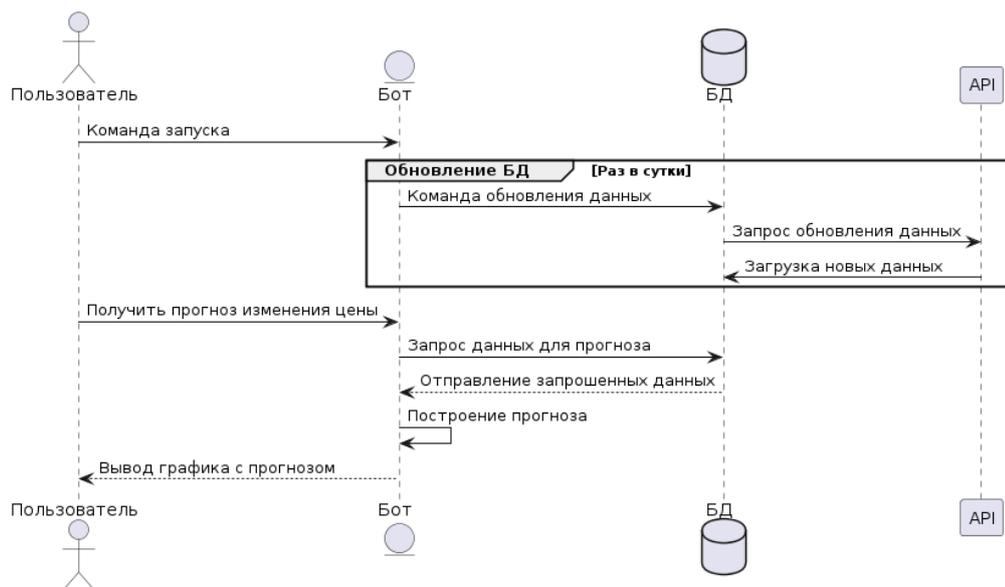


Рисунок 7 – Диаграмма последовательности «Получить прогноз изменения цены»

Процесс прогнозирования цены на актив происходит за счет аналитики исторических данных с биржи по выбранному инструменту, после чего строится график с возможным изменением цены и выводится пользователю. Предполагается, что пользователь выбрал интересующий его инструмент.

На рисунке 8 изображена диаграмма последовательности, описывающая процесс «Получить график изменения цены».

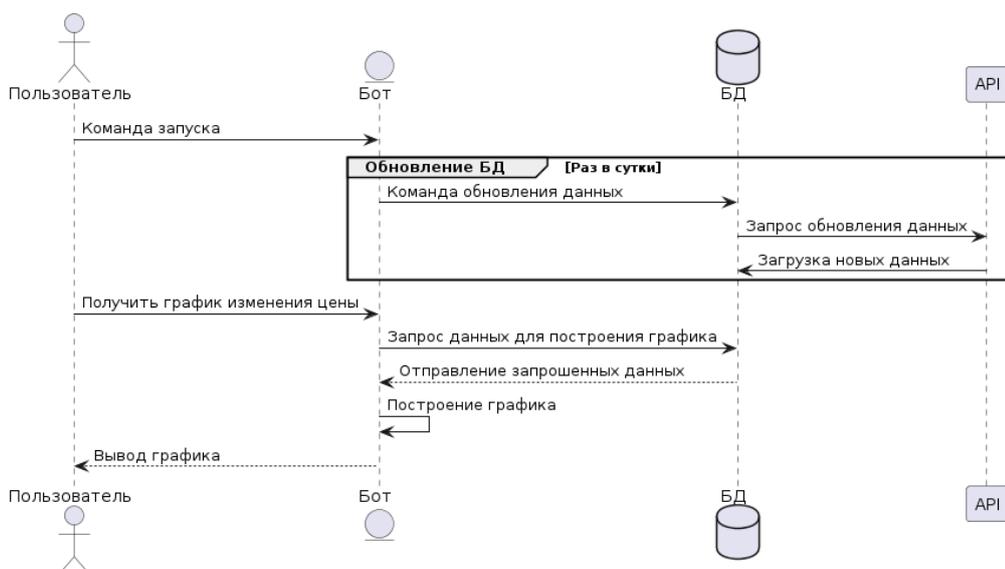


Рисунок 8 – Диаграмма последовательности «Получить график изменения цены»

Процесс построение графика с изменением цены на актив происходит за счет сбора исторических данных с биржи по выбранному инструменту, после чего строится график с изменением цены за последний месяц и выводится пользователю. Предполагается, что пользователь выбрал интересующий его инструмент.

#### **2.4. Выводы по главе 2**

В результате выполнения этапа проектирования была описана модель получения данных с сайта Мосбиржи, дальнейшая их обработка и возможности взаимодействия с этими данными.

Также были рассмотрены основные функции чат-бота на основе диаграмм прецедентов. Были выделены два вида отображения итоговых данных и их функционал.

### 3. Разработка и тестирование чат-бота

#### 3.1. Реализация базы данных и метод загрузки исторических данных

Загрузка данных происходит при помощи библиотеки MoeXAlgo, разработанной Московской биржей специально для алгоритмической торговли. С помощью библиотеки можно получить исторические данные, которые можно использовать для тестирования торговых стратегий, проверок гипотез и бэктестов, а также онлайн данные для алгоритмической торговли. В разрабатываемом проекте используются свечи по тикеру, для загрузки данных за необходимый период.

Загрузка происходит при помощи команды: «`Ticker().candles()`». Чтобы получить данные по интересующей акции необходимо указать тикер компании в команду «`Ticker()`», а в команду «`candles()`» указывается период, за который интересуют свечи, частотность и лимит загружаемых данных. После загрузки данные преобразуются и загружаются в базу данных. Всего база данных содержит 977 таблиц. На рисунке 9 представлена часть созданной базы данных.

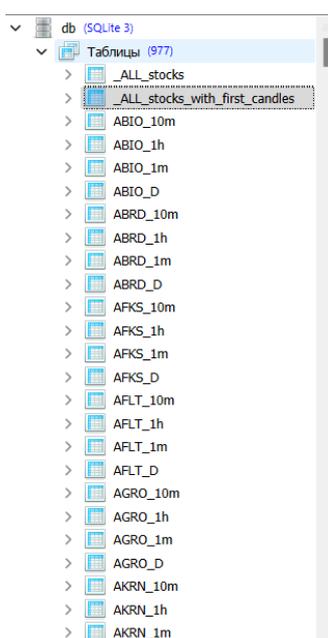


Рисунок 9 – Часть созданной базы данных

Изначально в базу данных в таблицу «ALL\_stocks» загружаются все доступные акции на Московской бирже, с полной информацией о них. Часть таблицы «ALL\_stocks» представлена на рисунке 10.

ticker	shortname	lotsize	decimals	minstep	issuesize	isin	regnumber	listlevel	information	
1	ABIO	iAPTEH ao	10	2	0.02	92645451	RU000A0JNAB6	1-01-08902-A	2	Аргтен биотех (ранее Институт Стволовых Клеток Человека ПАО «ИСКЧ» — биотехнологическая...
2	ABRD	АбрауДюрсо	10	1	0.2	98000184	RU000A0JS5T7	1-02-12500-A	3	Завод шампанских вин «Абрау-Дюрсо» — крупнейший в России производитель, выпускающий и...
3	AFKS	Система ao	100	3	0.001	9650000000	RU000A0DQZE3	1-05-01669-A	1	АФК «Система» является одной из самых крупных российских публичных финансовых корпорац...
4	AFLT	Аэрофлот	10	2	0.01	3975771215	RU0009062285	1-01-00010-A	1	ПАО «Аэрофлот» — крупнейший авиаперевозчик России по пассажиропотоку и пассажирообор...
5	AGRO	AGRO-гдр	1	1	0.2	0	US7496552057	None	1	Группа Компаний «РУСАГРО» - это крупнейший вертикальный агрохолдинг России. В настоящее...
6	AKRN	Акрон	1	0	2	36757156	RU0009028674	1-03-00207-A	2	ПАО «Акрон» — один из ведущих вертикально интегрированных производителей минеральных...
7	ALRS	АЛРОСА ao	10	2	0.01	7364965630	RU0007252813	1-03-40046-N	1	ПАО «АЛРОСА» — лидер алмазодобывающей отрасли мира, российская горнорудная компания с...
8	AMEZ	АшинскийМЗ	100	2	0.02	498454822	RU000A0B88G6	1-03-45219-D	3	ПАО «Ашинский металлургический завод» — крупнейший производитель аморфных и ...
9	APTK	АптекиЗбиб	10	3	0.002	7630433826	RU0008081765	1-01-07335-A	2	ПАО «Аптечная сеть 36,6» — крупная российская аптечная сеть, основанная в 1991 году. Аптечна...
10	AQUA	ИНАРКТИКА	1	1	0.5	87876649	RU000A0JQTS3	1-01-04461-D	1	ПАО «ИНАРКТИКА» (ранее Русская Аквакультура) является управляющей Компанией группы, ...
11	ARSA	Арсаргера	100	2	0.01	123817165	RU000A0JP0Q7	1-01-03163-D	3	ПАО «Управляющая компания «Арсаргера» — первая в России управляющая компания, акции кот...
12	ASSB	АстрЭнСб	1000	3	0.005	773900078	RU000A0D8MM8	1-01-55064-E	3	ПАО «Астраханская энергосбытовая компания» осуществляет централизованное энергоснабжени...
13	ASTR	Астра ao	1	2	0.05	210000000	RU000A106T36	1-01-01286-G	2	Группа Астра — один из лидеров российского IT-рынка и ведущий производитель защищенного...
14	AVAN	Авангрд-ao	1	0	1	80700000	RU000A0DM7B3	10102879B	3	АКЦИОНЕРНЫЙ КОММЕРЧЕСКИЙ БАНК "АВАНГАРД" - ПУБЛИЧНОЕ АКЦИОНЕРНОЕ ОБЩЕСТВО я...
15	BANE	Башнефт ao	1	1	0.5	147846489	RU0007976957	1-01-00013-A	3	ПАО АНК «Башнефт» является российским предприятием нефтегазовой промышленности. Комп...
16	BANEP	Башнефт ап	1	1	0.5	29788012	RU0007976965	2-01-00013-A	3	ПАО АНК «Башнефт» является российским предприятием нефтегазовой промышленности. Комп...
17	BELU	Новабев ao	1	0	1	15800000	RU000A0HLSM1	1-01-55052-E	1	Novabev Group — один из ведущих игроков алкогольного сектора российской экономики, лидер...
18	BISVP	БашИнСв ап	100	2	0.01	36165270	RU0009100176	2-01-00011-A	3	ПАО «Башинформсвязь» — один из крупнейших операторов связи в уральском регионе и стране,
19	BLNG	Белон ao	100	3	0.005	1150000000	RU000A0J2QG8	1-06-10167-F	3	ОАО «Белон» — компания, занимающаяся производством и продажей угля в России. Она являетс...
20	BRZL	БурЗолото	1	0	1	7027270	RU0009288658	1-06-20577-F	3	ОАО Бурятзолото — крупная золотодобывающая компания с долгосрочными планами развития,
21	BSPB	БСП ao	10	2	0.01	457544031	RU0009100945	10300436B	1	ПАО «Банк Санкт-Петербург» предоставляет кредитование, расчетно-кассовое обслуживание, ...
22	BSPBP	БСП ап	100	2	0.05	201000000	RU000A0JP0U9	20100436B	3	ПАО «Банк Санкт-Петербург» предоставляет кредитование, расчетно-кассовое обслуживание, ...
23	CARM	СТГ	100	4	0.0005	2203330301	RU000A105NV2	1-01-03536-G	3	CarMoney — российский финтех-сервис, действующий в форме микрофинансовой организации, ...

Рисунок 10 – Часть таблицы «ALL\_stocks»

После, из созданной таблицы, считываются тикеры доступных акций и начинается поиск первых данных, начиная с 01.01.2022. После того как даты первых свечек будут найдены и загружены в таблицу «ALL\_stocks\_with\_first\_candles» происходит чтение только что созданной таблицы и загрузка информации об изменениях цены каждой акции. Часть таблицы «ALL\_stocks\_with\_first\_candles» представлена на рисунке 11.

	ticker	listlevel	shortname	first_date
1	ABIO	2	iАРТГЕН ао	2022-01-03 09:59:00
2	ABRD	3	АбрауДюрсо	2022-01-03 10:00:00
3	ACKO	3	ACKO ао	<i>NULL</i>
4	AFKS	1	Система ао	2022-01-03 07:00:00
5	AFLT	1	Аэрофлот	2022-01-03 06:59:00
6	AGRO	1	AGRO-гдр	2022-01-03 09:59:00
7	AKRN	2	Акрон	2022-01-03 09:59:00
8	ALRS	1	АЛРОСА ао	2022-01-03 06:59:00
9	AMEZ	3	АшинскийМЗ	2022-01-03 09:59:00
10	APTK	2	АптекиЗбиб	2022-01-03 09:59:00
11	AQUA	1	ИНАРКТИКА	2022-01-03 09:59:00
12	ARSA	3	Арсагера	2022-01-03 10:04:00
13	ASSB	3	АстрЭнСб	2022-01-03 10:12:00
14	ASTR	2	Астра ао	2023-10-13 15:09:00
15	AVAN	3	Авангрд-ао	2022-01-03 10:16:00
16	BANE	3	Башнефт ао	2022-01-03 09:59:00
17	BANEP	3	Башнефт ап	2022-01-03 09:59:00
18	BELU	1	НоваБев ао	2022-01-03 10:00:00
19	BISVP	3	БашИнСв ап	2022-01-03 09:59:00
20	BLNG	3	Белон ао	2022-01-03 09:59:00
21	BRZL	3	БурЗолото	2022-01-03 10:02:00
22	BSPB	1	БСП ао	2022-01-03 09:59:00
23	BSPBP	3	БСП ап	2022-01-03 11:06:00
24	CARM	3	СТГ	2023-07-03 09:59:00

Рисунок 11 – Содержимое таблицы «ALL\_stocks\_with\_first\_candles»

Таким образом, данные загружаются последовательно по всем акциям за различные таймфреймы. Для каждой акции со своим таймфреймом создается отдельная таблица, в которой указаны даты начала и конца изменения цены, цена открытия и закрытия позиции, максимальная и минимальная цена полученных данных, объем последней сделки и количество доступных акций. На рисунке 12 представлена таблица «SBER\_10m», в которой содержатся данные о свечках с интервалом в 10 минут.

	begin	end	open	high	low	close	volume	value
1	2022-01-03 07:00:00	2022-01-03 07:09:59	295.9	296.1	294.5	294.97	440880	130166483.99999997
2	2022-01-03 07:10:00	2022-01-03 07:19:59	294.97	295.2	294.34	295	196510	57926156.8
3	2022-01-03 07:20:00	2022-01-03 07:29:59	295	295.2	294.75	295.17	88250	26038794
4	2022-01-03 07:30:00	2022-01-03 07:39:59	295.2	295.2	294.76	295.08	145320	42871543.99999999
5	2022-01-03 07:40:00	2022-01-03 07:49:59	294.92	295.12	294.61	294.7	71160	20980652.100000005
6	2022-01-03 07:50:00	2022-01-03 07:59:59	294.7	295.1	294.52	295.1	111610	32895975.600000005
7	2022-01-03 08:00:00	2022-01-03 08:09:59	295.09	295.39	294.9	295.22	200810	59273623.2
8	2022-01-03 08:10:00	2022-01-03 08:19:59	295.32	295.5	295.18	295.47	133870	39546138
9	2022-01-03 08:20:00	2022-01-03 08:29:59	295.46	295.67	295.42	295.55	116150	34329589.699999996
10	2022-01-03 08:30:00	2022-01-03 08:39:59	295.55	295.67	295.54	295.64	75600	22347381.999999996
11	2022-01-03 08:40:00	2022-01-03 08:49:59	295.6	296.7	295.6	296.35	528920	156630173.2
12	2022-01-03 08:50:00	2022-01-03 08:59:59	296.23	296.8	296.23	296.55	226700	67234089
13	2022-01-03 09:00:00	2022-01-03 09:09:59	296.55	296.95	296.55	296.7	242640	72012226.600000001
14	2022-01-03 09:10:00	2022-01-03 09:19:59	296.7	296.72	295.88	295.89	178970	53009537.899999998
15	2022-01-03 09:20:00	2022-01-03 09:29:59	295.89	296.62	295.87	296.62	137590	40759224.599999999
16	2022-01-03 09:30:00	2022-01-03 09:39:59	296.62	296.74	296.1	296.15	116260	34457703.5
17	2022-01-03 09:40:00	2022-01-03 09:49:59	296.22	296.3	295.86	296.12	155880	46153062.899999999
18	2022-01-03 09:50:00	2022-01-03 09:59:59	296.1	296.8	296	296.6	332720	98594650.799999998
19	2022-01-03 10:00:00	2022-01-03 10:09:59	296.57	297.29	296.45	297.14	942510	279904262.900000004
20	2022-01-03 10:10:00	2022-01-03 10:19:59	297.14	298.98	297.1	298.46	2394590	714002724.49999999
21	2022-01-03 10:20:00	2022-01-03 10:29:59	298.45	298.85	298.23	298.35	990600	295731563.9
22	2022-01-03 10:30:00	2022-01-03 10:39:59	298.36	300.77	298.36	300.47	2554880	765605097.39999999
23	2022-01-03 10:40:00	2022-01-03 10:49:59	300.47	301.2	300.11	300.51	1261480	379235776.399999986
24	2022-01-03 10:50:00	2022-01-03 10:59:59	300.51	300.62	299.64	300.25	1393190	418110599.200000005
25	2022-01-03 11:00:00	2022-01-03 11:09:59	300.2	302.87	299.11	302.75	3879080	1169891926.00000002

Рисунок 12 – Часть таблицы «SBER\_10m»

### 3.2. Модель для прогноза

Существует несколько различных статистических моделей, используемых для прогнозирования временных рядов. Самыми популярными среди них являются модели семейства ARIMA.

Сама модель ARIMA это скользящая средне-интегрированная авторегрессия. В её состав входят:

- авторегрессионная модель (AR), в которой будущие значения прогнозируются на основе комбинации прошлых значений, для этого используется параметр  $p$ ;

- интегрирование (I), показывающее количество операций разности для достижения стационарности временного ряда, для этого используется параметр  $d$ ;

– модель скользящего среднего (MA), где при прогнозировании используются ошибки прошлых прогнозов, для этого используется параметр  $q$ .

Данная модель используется, когда данные не имеют сезонности. Если сезонность присутствует, то лучше использовать модель SARIMA, которая состоит из двух разных моделей ARIMA, где одна модель отвечает за сезонные компоненты, а вторая за несезонные.

В ходе анализа данных была зафиксированная сезонность, в связи с чем будет использоваться модель SARIMA. Сезонность заключается в том, что во втором и девятом месяцах каждого года цены на акции заметно падали. На рисунках 14, 15 изображены графики среднего роста цены в зависимости от месяца.

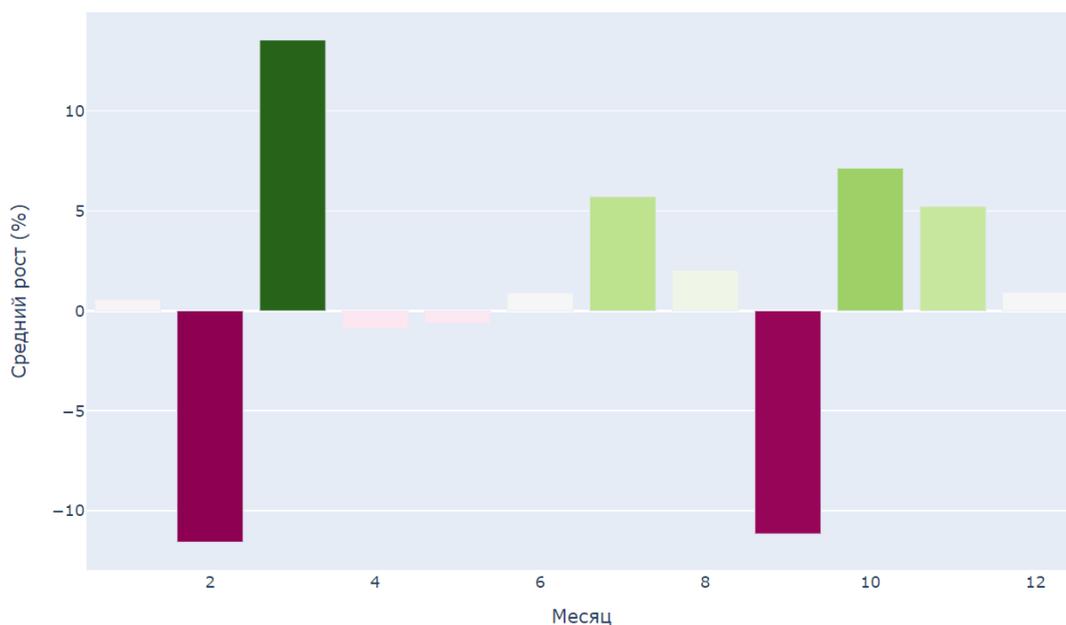


Рисунок 13 – График среднего роста цены акций Сбербанка

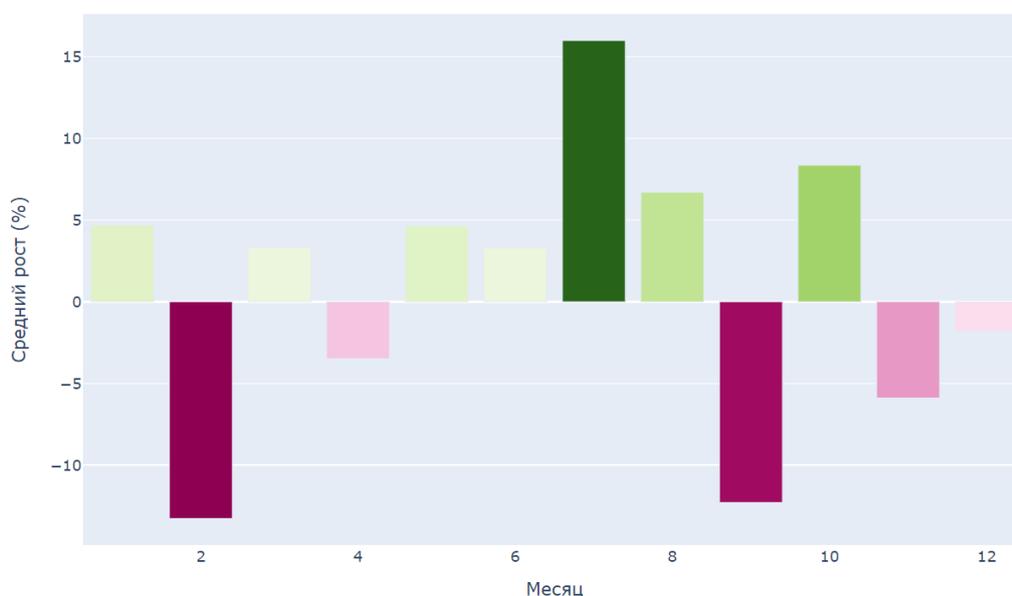


Рисунок 14 – График среднего роста цены акций Яндекса

Для правильного прогнозирования необходимо определить параметры модели. Определить их можно двумя способами: самостоятельный анализ и автоматический подбор. В алгоритме используется автоматический выбор, так как на выходе можно получить наиболее точные и подходящие для прогноза параметры.

Использование автоматического метода происходит за счет библиотеки «`rdmarima.arima`». Вызывая данную библиотеку необходимо передать данные, для которых будет происходить поиск параметров, указать, что данные имеют сезонную зависимость, а также указать период, в течение которого возможны сезонные различия. На рисунке 15 представлен результат поиска подходящих параметров для модели.

```

=====
SARIMAX Results
=====
Dep. Variable:          y      No. Observations:          579
Model:                 SARIMAX(2, 2, 1)  Log Likelihood             -1806.346
Date:                 Tue, 14 May 2024  AIC                        3620.691
Time:                 14:19:08         BIC                        3638.123
Sample:               0              HQIC                       3627.489
                        - 579
Covariance Type:      opg
=====
              coef      std err          z      P>|z|      [0.025      0.975]
-----
ar.L1         -0.0216      0.027      -0.787      0.431      -0.075      0.032
ar.L2          0.1869      0.013     14.199      0.000      0.161      0.213
ma.L1         -0.9871      0.013    -75.714      0.000     -1.013     -0.962
sigma2         30.4913      0.672     45.407      0.000     29.175     31.807
=====
Ljung-Box (L1) (Q):          0.00  Jarque-Bera (JB):          39662.42
Prob(Q):                    0.96  Prob(JB):                   0.00
Heteroskedasticity (H):     0.12  Skew:                       -3.23
Prob(H) (two-sided):        0.00  Kurtosis:                   43.10
=====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

```

Рисунок 15 – Результат поиска подходящих параметров для модели

После нахождения параметров можно начать обучение модели. Для этого необходимо передать в модель сам временной ряд, а также только что полученные параметры. Результат обучения модели продемонстрирован на рисунке 16.

```

=====
SARIMAX Results
=====
Dep. Variable:          Price    No. Observations:      579
Model:                 SARIMAX(2, 2, 1)x(2, 2, 1, 12)  Log Likelihood         -1834.425
Date:                  Tue, 14 May 2024              AIC                    3682.850
Time:                  15:25:09                      BIC                    3713.057
Sample:                0                            HQIC                   3694.652
                                     - 579
Covariance Type:      opg
=====
              coef    std err          z      P>|z|      [0.025    0.975]
-----+-----
ar.L1          0.0214    0.027        0.794    0.427    -0.032    0.074
ar.L2          0.1536    0.018        8.383    0.000     0.118    0.189
ma.L1         -1.0000   14.790       -0.068    0.946   -29.988   27.988
ar.S.L12       -0.7510    0.018   -42.087    0.000    -0.786   -0.716
ar.S.L24       -0.4258    0.016   -27.160    0.000    -0.457   -0.395
ma.S.L12       -0.9996    6.554     -0.153    0.879   -13.844   11.845
sigma2         38.1545   582.027     0.066    0.948  -1102.598  1178.907
=====
Ljung-Box (L1) (Q):      0.04  Jarque-Bera (JB):      9924.38
Prob(Q):                 0.85  Prob(JB):              0.00
Heteroskedasticity (H):  0.15  Skew:                  2.00
Prob(H) (two-sided):    0.00  Kurtosis:              23.36
=====

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

```

Рисунок 16 – Результат обучения модели

После обучения необходимо выполнить сам прогноз. Для этого используется метод «`gridect()`», в который передается размерность набора данных и период, на который необходимо произвести прогноз. В результате работы метода есть начальные данные и полученный прогноз, на основе начальных данных. Далее необходимо сформировать график для пользователя по полученным данным.

В результате проведенных вычислений и обучения модели, был получен график с прогнозом акций компании «Сбербанк» (Рисунок 17). Данный график показывает прогноз на ближайшие 60 дней.



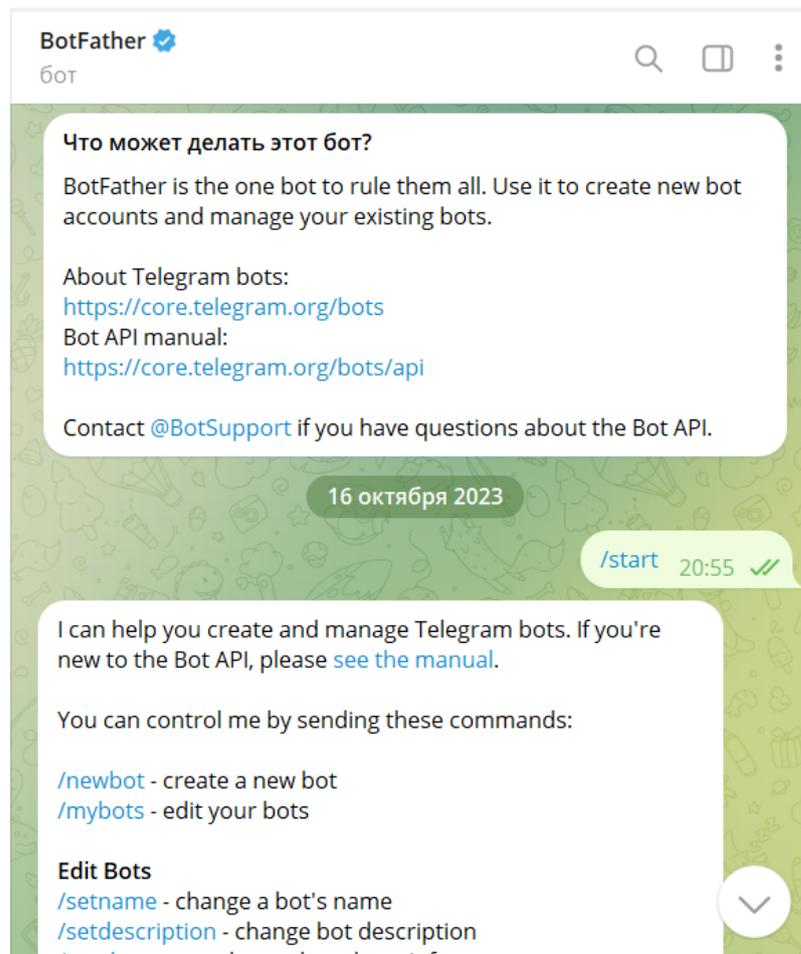


Рисунок 18 – Создание собственного чат-бота

Далее BotFather предлагает придумать имя и тег создаваемого бота, по которому у пользователей мессенджера будет возможность его найти. После успешного создания чат-бота, бот BotFather отправит в чат токен только что созданного бота. Получаемый токен выглядит следующим образом: «0123456789:ABC6DefgHI51jKlm24noPqR7stUv8wxYzAb». На рисунке 19 изображены диалог с BotFather с именем, тегом и частью токена, созданного бота.

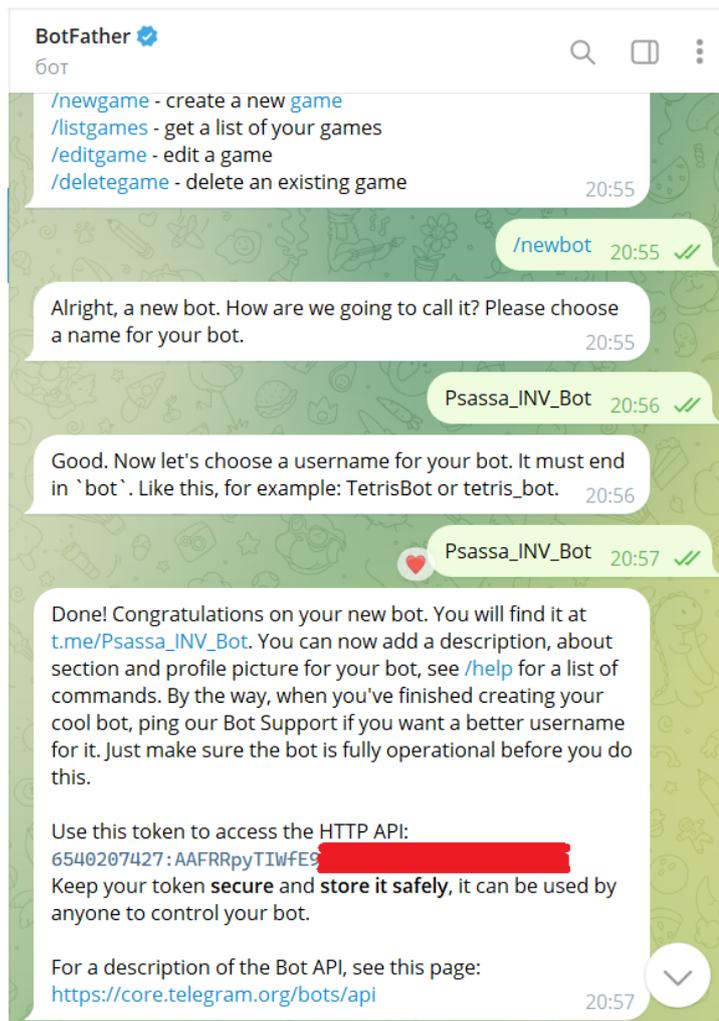


Рисунок 19 – Создание имени, тега и токена чат-бота

Также при создании бота есть возможность настроить информацию о созданных ботах, добавить различные команды для взаимодействия, загрузить картинку для аватара и добавить описание.

### 3.4. Тестирование работы чат-бота в Telegram

При первом использовании чат-бота, пользователю необходимо ввести команду «/start» или нажать на кнопку «Запустить», которая автоматически отправит команду «/start». После чего пользователю выводится приветственное сообщение. Пример первого запуска чат-бота изображен на рисунке 20.

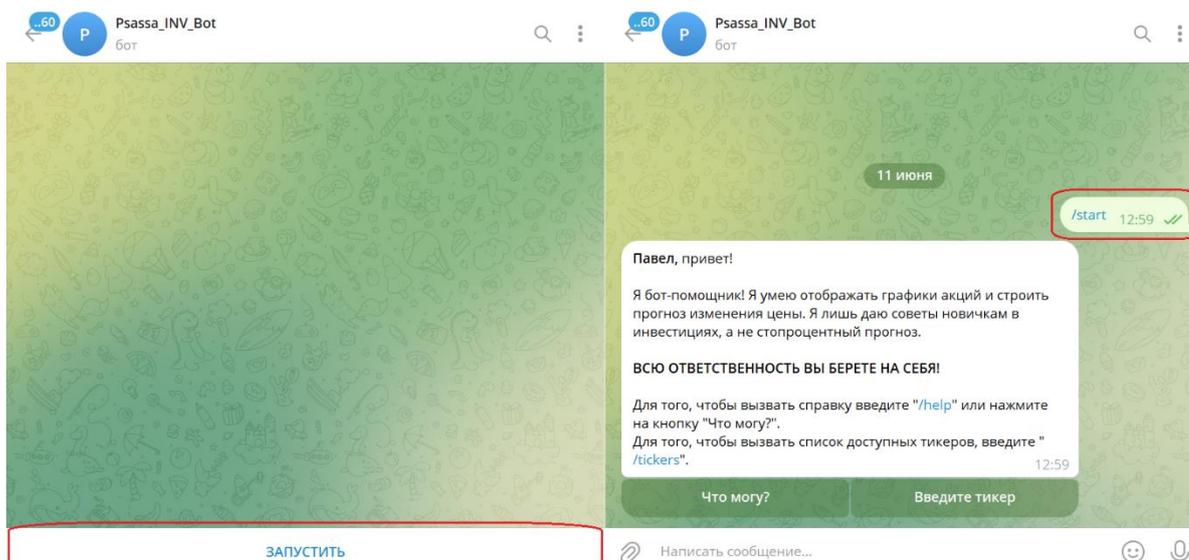


Рисунок 20 – Запуск чат-бота

После чего пользователь может ввести команду «/help», которая отвечает за краткую справку по взаимодействию с чат-ботом. Кнопка под сообщением «Что могу?» также отвечает за команду «/help». Результат ввода команды «/help» и нажатия кнопки «Что могу?» продемонстрированы на рисунке 21.

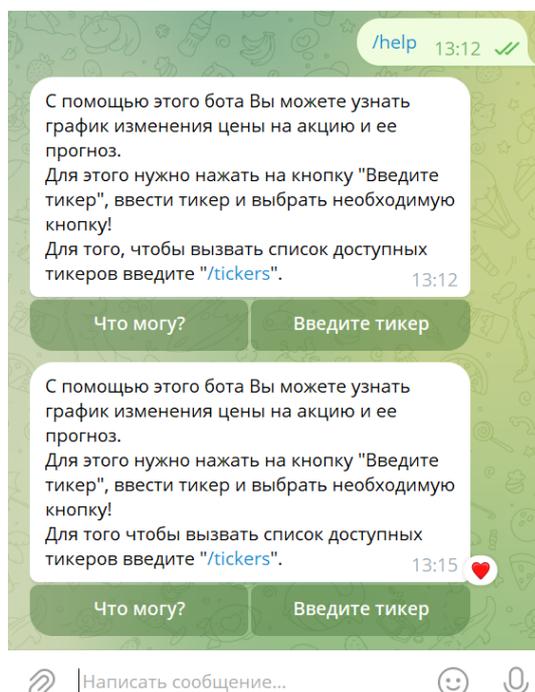


Рисунок 21 – Результат ввода команды «/help» и нажатия кнопки «Что могу?»

Также пользователю становятся доступны команда «/tickers», которая выводит список доступных тикеров для просмотра, и кнопка «Введите тикер», нажимая на которую пользователю будет предложено ввести тикер интересующей его компании. Тикеры в свою очередь – «это кодовое обозначение актива на бирже, которое состоит не более чем из шести латинских букв или цифр (либо их сочетания) и является уникальным на конкретной бирже по отношению к конкретной акции» [17]. На рисунках 22, 23 изображены результаты отправления команды «/tickers» и нажатия кнопки «Введите тикер» соответственно.

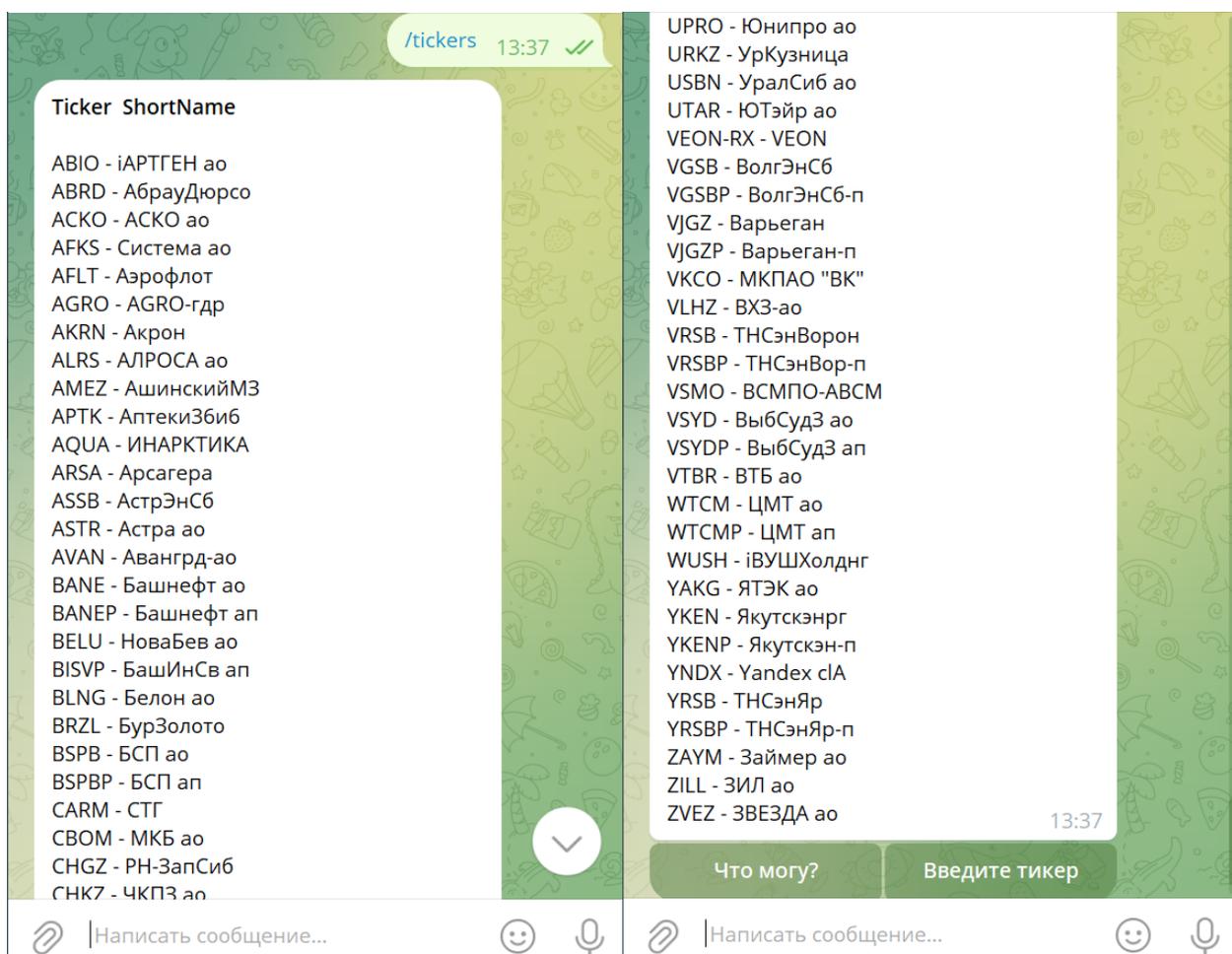


Рисунок 22 – Результат ввода команды «/tickers»

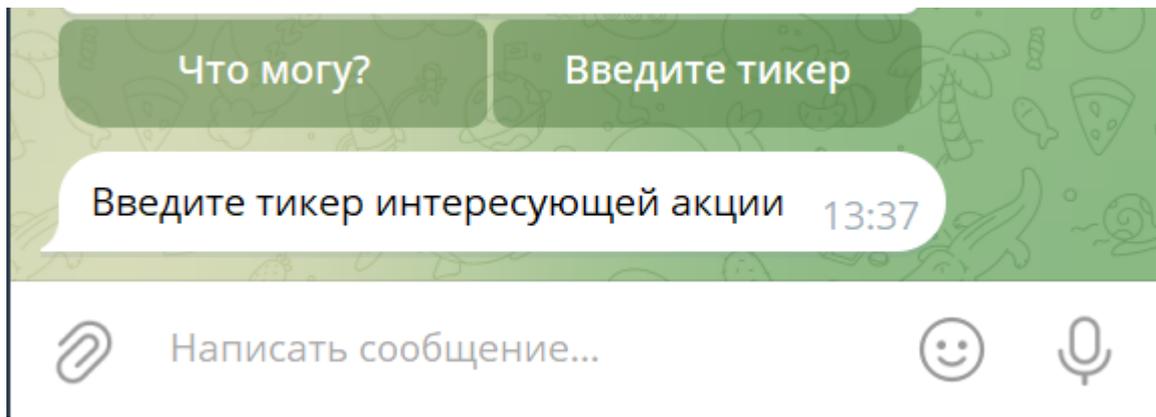


Рисунок 23 – Результат нажатия на кнопку «Введите тикер»

При вводе интересующего тикера пользователем, выводится сообщение, в котором указывается введенный тикер, название компании и краткая информация о компании, которой принадлежит данный тикер. Ниже сообщения появляется три кнопки «Прогноз», «График» и «Выбрать другой тикер». Внешний вид представлен на рисунке 24.

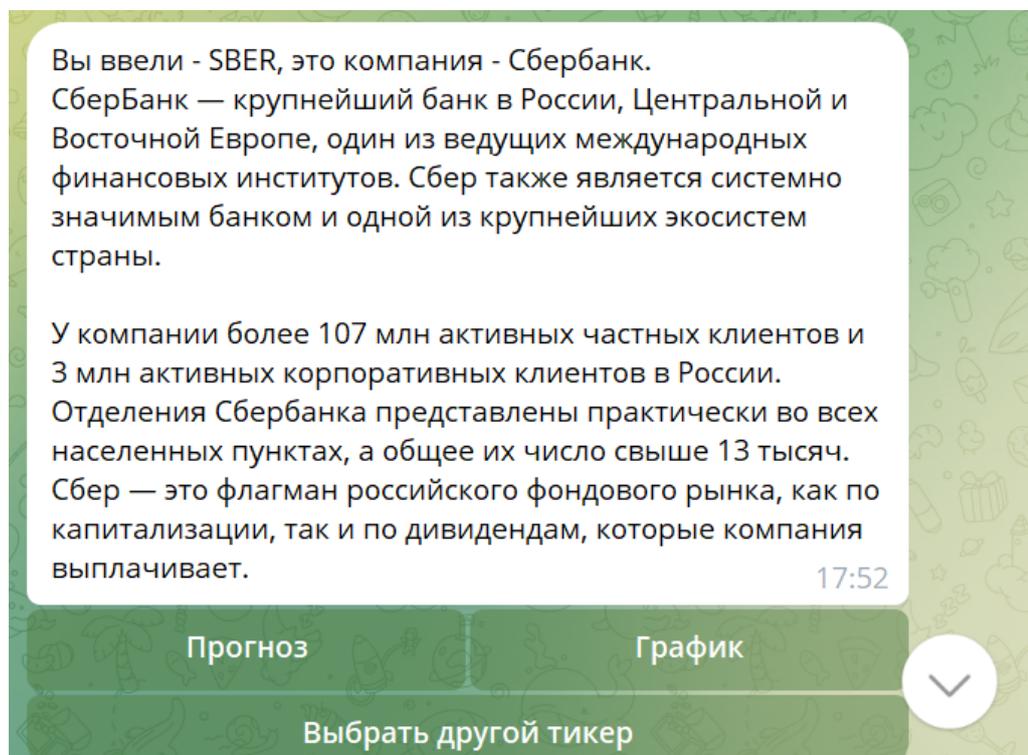


Рисунок 24 – Вывод информации после ввода тикера

Кнопка «Прогноз» выводит пользователю информационное сообщение о том, что происходит генерация прогноза, далее отображается прогноз с периодом в два месяца по акции, которая была выбрана ранее. Пример отображения представлен на рисунке 25.



Рисунок 25 – Отображение прогноза

При нажатии на кнопку «График» пользователю выводится сообщение о генерации графика, происходит отображение графика изменения цены. Пример представлен на рисунке 26.



Рисунок 26 – Отображение графика изменения цены

### 3.5. Выводы по главе 3

В результате выполнения этапа была разработана база данных, которая содержит в себе информацию об изменении цены на акции с различной частотностью за последние два года.

Для прогнозирования временных рядов используется модель SARIMA. Данная модель обеспечивает хорошую точность прогнозирования, что является ключевым фактором в области финансовых инвестиций на ближайший период.

Прогноз, полученный при помощи модели SARIMA, показывает высокую точность при сравнении с реальными показателями на текущий период.

Разработан функциональный чат-бот в Telegram, с помощью которого пользователю могут ориентироваться на рынке акций.

## ЗАКЛЮЧЕНИЕ

В первой главе были проанализированы существующие аналоги и возможные площадки для реализации проекта, в результате чего был выбран Telegram Bot в качестве инструмента взаимодействия с пользователем.

Во второй главе был разработан метод получения данных, их последующая обработка, а также сформулированы основные функции чат-бота.

В третьей главе была разработана база данных, метод построения графиков изменения цены, прогнозирование на ближайший период, а также реализован чат-бот.

В результате выполнения выпускной квалификационной работы был разработан функциональный чат-бот на базе мессенджера Telegram, в котором пользователи могут получить описание интересующей компании, график изменения цены акции выбранной компании, а также прогноз на ближайшие два месяца с возможным изменением цены акции.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Московская биржа подвела итоги торгов в декабре 2023 года // Московская биржа: официальный сайт. – URL: <https://www.moex.com/n66707> (дата обращения 25.02.2024).
2. Что такое трейдинг и как на нем заработать // ФИНАМ: официальный сайт. – URL: <https://www.finam.ru/publications/item/chto-takoe-treiyding-i-kak-na-nem-zarabotat-20221215-143200/#title0> (дата обращения 15.02.2024).
3. Чем брокер отличается от управляющей компании // ААА Управление капиталом: сайт. – URL: <https://aaacapital.ru/journal/misc/upravlyayushchiy-i-broker-v-chem-raznitsa/> (дата обращения 15.02.2024).
4. Что такое биржа // Тинькофф: официальный сайт. – URL: <https://www.tinkoff.ru/invest/help/educate/how-it-works/what-is-exchange/exchange-types/?card=q3> (дата обращения 06.03.2024).
5. Как я слежу за акциями в гугл-таблице: три простых способа // Тинькофф журнал: сайт. – URL: <https://journal.tinkoff.ru/list/spreadsheets-for-investor/> (дата обращения 06.03.2024).
6. Московская биржа // Википедия: официальный сайт. – URL: [https://ru.wikipedia.org/wiki/Московская\\_биржа](https://ru.wikipedia.org/wiki/Московская_биржа) (дата обращения 06.03.2024).
7. Инвестируйте с роботом-советником // ВТБ: официальный сайт. – URL: <https://www.vtb.ru/personal/investicii/robot/#strategy> (дата обращения 15.02.2024).
8. Мессенджер // ELAMA: официальный сайт. – URL: <https://elama.ru/glossary/messendzher> (дата обращения 06.03.2024).
9. Россияне тратят на социальные сети до пяти часов в день // ВЦИОМ: официальный сайт. – URL: <https://rg.ru/2023/08/10/vciom-polzovateli-socsetej-tratjat-na-nih-do-piati-chasov-v-den.html> (дата обращения 06.03.2024).

10. Рейтинги, интернет // Media Score: официальный сайт. – URL: <https://mediascore.net/data/> (дата обращения 06.03.2024).

11. До 16 и дольше: как россияне читают Telegram и WhatsApp. Инфографика // РБК: официальный сайт. – URL: [https://www.rbc.ru/technology\\_and\\_media/21/05/2023/6467807d9a79475cae170214](https://www.rbc.ru/technology_and_media/21/05/2023/6467807d9a79475cae170214) (дата обращения 15.02.2024).

12. Боты в Telegram: какими они бывают, как работают и как сделать своего // SkillBox Media: сайт. – URL: <https://skillbox.ru/media/marketing/boty-v-telegram-kakimi-oni-byvayut-kak-rabotayut-i-kak-sdelat-svoego/> (дата обращения 15.02.2024).

13. Python // Википедия: официальный сайт. – URL: <https://ru.wikipedia.org/wiki/Python> (дата обращения 28.03.2024).

14. Полезные библиотеки для Python: чем пользуются разработчики // SkillFactory Media: сайт. – URL: <https://blog.skillfactory.ru/top-29-bibliotek-dlya-python-chem-polzuyutsya-razrabotchiki/> (дата обращения 28.03.2024).

15. VSCode vs PyCharm // Python.Engineering: сайт. – URL: <https://python.engineering/pycharm-vs-vscode/> (дата обращения 28.03.2024)

16. Диаграммы прецедентов // НГТУ НЭТИ: официальный сайт. – URL: [https://ciu.nstu.ru/kaf/persons/1914/page47048/diagramm\\_precedentov](https://ciu.nstu.ru/kaf/persons/1914/page47048/diagramm_precedentov) (дата обращения 18.02.2024).

17. Что такое тикеры акций и зачем они нужны инвестору // ФИНАМ: официальный сайт. – URL: <https://www.finam.ru/publications/item/chto-takoe-tikery-akcii-i-zachem-oni-nuzhny-investoru-20200929-11440/> (дата обращения 30.05.2024)

## ПРИЛОЖЕНИЕ А

### **predict\_function.py**

```
import sqlite3
import pandas as pd
import plotly.graph_objs as go

from pmdarima.arima import auto_arima
from statsmodels.tsa.statespace.sarimax import SARIMAX

def graphic(user_ticker):
    #Загрузка данных из БД в датафрейм
    with sqlite3.connect("db.db") as conn:
        cur = conn.cursor()
        query = f"SELECT end, close FROM {user_ticker}_D"
        cur.execute(query)
        tuple_list = cur.fetchall()
        df = pd.DataFrame(tuple_list, columns=['Date', 'Price'])
        split_index = len(df) // 2
        half_df = df.iloc[split_index:]
        half_df.set_index('Date', inplace=True)

        fig = go.Figure()
        #Построение графика
        fig.add_trace(go.Scatter(
            x=half_df.index,
            y=half_df['Price'],
            mode='lines',
            line=dict(color='orange')
        ))

        fig.update_layout(
            title=f"Price change schedule {user_ticker} from {half_df.iloc[0].name} to
{half_df.iloc[-1].name}",
            xaxis_title="Date",
            yaxis_title="Price",
            legend_title="Legend",
            width=900,
            height=600
        )
    #Сохранение графика как файла
```

## ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

```
file_path = f"{user_ticker}_graphic.png"  
fig.write_image(file_path)  
  
return file_path
```

```
def info_tick(): #Получение списка тикеров и шортнейма  
with sqlite3.connect("db.db") as conn:  
    cur = conn.cursor()  
    query = "SELECT ticker, shortname FROM _ALL_stocks"  
    cur.execute(query)  
    tuple_list = cur.fetchall()  
    df = pd.DataFrame(tuple_list, columns=['Ticker', 'ShortName',])  
    return df
```

```
def information_tick(): #Получение тикера, шортнейма и кратк. информ  
with sqlite3.connect("db.db") as conn:  
    cur = conn.cursor()  
    query = "SELECT ticker, shortname, information FROM _ALL_stocks"  
    cur.execute(query)  
    tuple_list = cur.fetchall()  
    df = pd.DataFrame(tuple_list, columns=['Ticker', 'ShortName', 'Information'])  
    return df
```

```
def search_in_db(user_ticker): #Поиск существует ли интересующий тикер  
with sqlite3.connect("db.db") as conn:  
    cur = conn.cursor()  
    cur.execute("SELECT ticker FROM _ALL_stocks")  
    tuple_list = cur.fetchall()  
  
    # Преобразуем список кортежей в список строк  
    tickers = [item[0] for item in tuple_list]  
  
    if user_ticker in tickers:  
        return True  
    else:  
        return False
```

```
def prediction(user_ticker): #Функция прогнозирования  
with sqlite3.connect("db.db") as conn:  
    cur = conn.cursor()
```

## ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

```
query = f"SELECT end, close FROM {user_ticker}_D"
cur.execute(query)
tuple_list = cur.fetchall()
df = pd.DataFrame(tuple_list, columns=['Date', 'Price'])
#Преобразование даты
df['Date'] = pd.to_datetime(df['Date'])
df['date'] = df['Date'].dt.date
df['time'] = df['Date'].dt.time
df.drop(['time', 'date'], axis=1, inplace=True)
df["Date"] = pd.to_datetime(df["Date"], format='%Y-%m-%d')
df['Year'] = df['Date'].dt.year
df["Month"] = df["Date"].dt.month
df["Day"] = df["Date"].dt.day
#Поиск параметров модели и построение модели
parametr_arima = auto_arima(df['Price'], seasonal=True, m=52,
suppress_warnings=True)
order = parametr_arima.order
pdq = list(order)
seasonal_order = pdq[0], pdq[1], pdq[2], 52
model_sarimax = SARIMAX(df['Price'], order=order,
seasonal_order=seasonal_order)
fitted_model = model_sarimax.fit()
#Построение прогноза
predictions = fitted_model.predict(len(df['Price']), len(df) + 60)
split_index = len(df) // 4
quat_df = df.iloc[split_index:]
quat_df.set_index('Date', inplace=True)
df.set_index('Date', inplace=True)
start_date = pd.Timestamp(df.index[-1])

# Создаем диапазон дат для прогноза, начиная со следующего дня после
последней даты
forecast_dates = pd.date_range(start=start_date + pd.Timedelta(days=1),
periods=len(predictions), freq='D')

# Создаем DataFrame с прогнозами и добавляем столбец с датами
predictions_with_dates = pd.DataFrame({'Date': forecast_dates, 'Predictions':
predictions})
predictions_with_dates.set_index('Date', inplace=True)
fig = go.Figure()
```

## ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ А

```
# Представления графиков
fig.add_trace(go.Scatter(
    x=quat_df.index,
    y=quat_df['Price'],
    mode='lines',
    name=f'Price {user_ticker}',
    line=dict(color='orange')
))

# Add predictions line plot
fig.add_trace(go.Scatter(
    x=predictions_with_dates.index,
    y=predictions_with_dates['Predictions'],
    mode='lines',
    name=f'Forecast {user_ticker}',
    line=dict(color='green')
))

fig.update_layout(
    title=f"{user_ticker} - Price forecast",
    xaxis_title="Date",
    yaxis_title="Price",
    legend_title="Legend"
)

file_path = f"{user_ticker}_prediction.png"
fig.write_image(file_path)

return file_path
```

## ПРИЛОЖЕНИЕ Б

### handlers.py

```
import os
import asyncio

from aiogram import types, html, Dispatcher, F
from aiogram.filters import CommandStart, Command
from aiogram.fsm.context import FSMContext
from aiogram.types import FSInputFile

from state import Ticker
from kb import start_kb, main_kb, main_kb_graph, main_kb_predict
from predict_function import prediction, graphic, info_tick, search_in_db,
information_tick

user_ticker = None
async def start_command(message: types.Message, state: FSMContext):
    await message.answer(f'<b>{html.quote(message.from_user.full_name)},</b>
привет!\n\n'
        f'Я бот-помощник! Я умею отображать графики акций и
строить прогноз изменения цены. '
        f'Я лишь даю советы новичкам в инвестициях, а не
стопроцентный прогноз.\n'
        f'\n<b>ВСЮ ОТВЕТСТВЕННОСТЬ ВЫ БЕРЕТЕ НА
СЕБЯ!</b>\n\n'
        f'Для того, чтобы вызвать справку введите "/help" или нажмите
на кнопку "Что могу?". \n'
        f'Для того, чтобы вызвать список доступных тикеров, введите
"/tickers"!',
        reply_markup=start_kb)
    await state.clear()

async def help_command(message: types.Message, state: FSMContext):
    await message.answer(f'С помощью этого бота Вы можете узнать график
изменения цены на акцию и ее прогноз.\n'
        f'Для этого нужно нажать на кнопку "Введите тикер", ввести
тикер и выбрать необходимую кнопку!\n'
        f'Для того, чтобы вызвать список доступных тикеров введите
"/tickers".\n',
        reply_markup=start_kb)
```

## ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
await state.clear()

async def ticker_info(message: types.Message, state: FSMContext):
    inform = info_tick()
    header = "<b>Ticker ShortName</b>\n\n"
    # Преобразуем DataFrame в список строк с нужным форматированием
    inform_list = [f"{row['Ticker']} - {row['ShortName']}" for index, row in
inform.iterrows()]
    # Разделяем список на две части
    mid_point = len(inform_list) // 2
    part1 = inform_list[:mid_point]
    part2 = inform_list[mid_point:]
    # Объединяем строки в одну длинную строку с разделением на новые
строки
    inform_str1 = header + '\n'.join(part1)
    inform_str2 = header + '\n'.join(part2)
    # Отправляем каждую часть отдельно
    await message.answer(inform_str1, reply_markup=start_kb)
    await message.answer(inform_str2, reply_markup=start_kb)
    await state.clear()

async def get_help(c: types.CallbackQuery, state: FSMContext):
    await c.message.answer(f"С помощью этого бота Вы можете узнать график
изменения цены на акцию и ее прогноз.\n'
        f"Для этого нужно нажать на кнопку "Введите тикер", ввести
тикер и выбрать необходимую кнопку!\n'
        f"Для того чтобы вызвать список доступных тикеров введите
"/tickers".\n',
        reply_markup=start_kb)
    await c.answer()
    await state.clear()

async def get_ticker(c: types.CallbackQuery, state: FSMContext):
    await c.message.answer(f"Введите тикер интересующей акции")
    await c.answer()
    await state.set_state(Ticker.name)

async def name_ticker(message: types.Message, state: FSMContext):
    global user_ticker
    user_ticker = message.text
```

## ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
if search_in_db(user_ticker) == True:
    try:
        tickers = information_tick()
        match_ticker = tickers[tickers['Ticker'] == user_ticker]
        ticker = match_ticker.iloc[0]['Ticker']
        short_name = match_ticker.iloc[0]['ShortName']
        inf_tick = match_ticker.iloc[0]['Information']
        await message.answer(f"Вы ввели - {ticker}, это компания -
{short_name}.\n\n{inf_tick}", reply_markup=main_kb)
        await state.clear()
    except:
        pass
else:
    await message.answer(f"Вы ввели неизвестный тикер, попробуйте еще
раз!")

async def generate_predict(user_ticker, chat_id, message_id, bot):
    try:
        file_predict = prediction(user_ticker)
        photo = FSInputFile(file_predict)
        await bot.send_photo(chat_id, photo=photo, reply_markup=main_kb_predict,
reply_to_message_id=message_id)
    except Exception as e:
        await bot.send_message(chat_id, f"Произошла ошибка: {str(e)}")
    finally:
        # Удаление файла, если он существует
        if os.path.exists(file_predict):
            os.remove(file_predict)

async def get_predict(c: types.CallbackQuery, state: FSMContext):
    global user_ticker
    await c.message.answer("Пожалуйста, подождите, идет генерация прогноза
\n Процесс может занять продолжительное время!")
    await c.answer()
    await state.clear()
    asyncio.create_task(generate_predict(user_ticker, c.message.chat.id,
c.message.message_id, c.bot))
```

## ПРОДОЛЖЕНИЕ ПРИЛОЖЕНИЯ Б

```
async def get_graph(c: types.CallbackQuery, state: FSMContext):
    global user_ticker
    await state.clear()
    try:
        await c.message.answer("Пожалуйста, подождите, идет генерация
графика")
        # Генерация графика
        file_graphic = graphic(user_ticker)
        # Отправка графика пользователю
        photo = FSInputFile(file_graphic)
        await c.message.answer_photo(photo=photo, reply_markup=main_kb_graph)

        # Отправка ответа на callback запрос
        await c.answer()
    except Exception as e:
        # Обработка возможных исключений и отправка сообщения об ошибке
        await c.message.answer(f"Произошла ошибка: {str(e)}")
    finally:
        # Удаление файла, если он существует
        if os.path.exists(file_graphic):
            os.remove(file_graphic)

def register_user_messages(dp: Dispatcher):
    dp.message.register(start_command, CommandStart())
    dp.message.register(help_command, Command('help'))
    dp.message.register(ticker_info, Command('tickers'))
    dp.callback_query.register(get_help, F.data == 'help_user')
    dp.callback_query.register(get_ticker, F.data == 'ticker_user')
    dp.message.register(name_ticker, Ticker.name)
    dp.callback_query.register(get_predict, F.data == 'predict')
    dp.callback_query.register(get_graph, F.data == 'graph')
```

## ПРИЛОЖЕНИЕ В

### main.py

```
import asyncio
import logging
import schedule
import time

from aiogram.client.bot import DefaultBotProperties
from aiogram import Bot, Dispatcher
from config import BOT_TOKEN

from handlers import register_user_messages
from Update_db import stocks_in_db, download_candels

logger = logging.getLogger(__name__)

async def main():
    logging.basicConfig(
        level=logging.INFO,
        format="%(asctime)s - %(levelname)s - %(name)s - %(message)s",
    )
    logger.error("Starting bot")

    bot = Bot(BOT_TOKEN, default=DefaultBotProperties(parse_mode='HTML'))
    dp = Dispatcher()

    register_user_messages(dp)
    await dp.start_polling(bot)

if __name__ == '__main__':
    asyncio.run(main())
    #Обновление БД каждые сутки
    schedule.every(24).hours.do(download_candels)

    while True:
        schedule.run_pending()
        time.sleep(1)
```

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

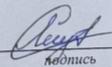
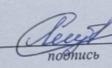
Институт космических и информационных технологий  
Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
О.В. Непомнящий  
«17» 06 2024 г.

### БАКАЛАВРСКАЯ РАБОТА

090301 Информатика и вычислительная техника

Анализатор биржевого рынка для Telegram Bot

Руководитель	 подпись	17.06.2024 дата	старший преподаватель должность, ученая степень	С.Л. Верхошенцева
Выпускник	 подпись	17.06.2024 дата		П.С. Соин
Нормоконтролёр	 подпись	17.06.2024 дата	старший преподаватель должность, ученая степень	С.Л. Верхошенцева

Красноярск 2024