

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФГБОУ ВО «Брянский государственный технический университет»

Кафедра «Компьютерные технологии и системы»

Утверждаю

Зав. каф. «КТС», д.т.н., доц.

_____ А.В. Аверченков

«__» февраля 2023 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ
РАБОТА**

по специальности

10.05.04 – «Информационно-аналитические системы безопасности»

Специализация – «Автоматизация информационно-аналитической
деятельности»

на тему:

«Моделирование информационно-аналитической системы анализа фондового
рынка»

Документы текстовые: _____ страниц

Автор работы:
Студент группы О-17-ИАС-айд-С
Андронов Михаил Павлович
«__» _____ 2023 г.

Руководитель работы:
к.т.н., доц., доцент каф. «КТС»
Леонов Юрий Алексеевич
«__» _____ 2023 г.

ЗАДАНИЕ

на выпускную квалификационную работу

по специальности

10.05.04 – «Информационно-аналитические системы безопасности»

Специализация – «Автоматизация информационно-аналитической
деятельности»

Студент: Андронов М.П. группа: О-17-ИАС-айд-С

Тема работы: «Моделирование информационно-аналитической системы
анализа фондового рынка»

Дата выдачи задачи: «__» _____ 2022 г.

Дата сдачи работы: «__» _____ 2022 г.

Утверждено приказом по БГТУ № _____ от «__» _____ 2022 г.

Руководитель работы: _____ / Ю.А. Леонов

Заведующий кафедрой

«Компьютерные технологии и системы»

д.т.н., доц. _____

/ А.В. Аверченков

АННОТАЦИЯ

Разработка информационно-аналитической системы для моделирования информационно-аналитической системы анализа фондового рынка.

В выпускной квалификационной работе был проведен анализ фондового рынка, исследованы существующий аналоги и сформулированы функциональные требования к разрабатываемому программному продукту.

На основе выдвинутых требований, была спроектирована и реализована база данных, разработана функциональная схема работы системы и диаграмма потоков данных. Также была создана диаграмма объектов и описана схема взаимодействия серверной и клиентской части ПО.

Был реализован модуль автоматического обновления данных. Для анализа котировок акции были разработаны модули технического и фундаментального анализа, а также модуль анализа новостей. В рамках работы были реализованы интерфейсы пользователя для десктопной версии продукта, а также для Android-приложения

В работе представлены результаты тестирования программного продукта и оценка производительности системы.

Также был проведен экономический анализ разработанной информационно-аналитической системы, в рамках которого был произведен расчет трудозатрат и материальных затрат на обеспечение разработки и внедрения продукта. На основе анализа было представлено экономическое обоснование разработки данного проекта.

ANNOTATION

Development of an information-analytical system for modeling an information-analytical system for analyzing the stock market.

In the final qualifying work, an analysis of the financial market was carried out, existing analogues and compositions of functional requirements for the developed software product were investigated.

Based on the requirements put forward, a database was implemented, a functional diagram of the system operation and a data flow diagram were developed. An object diagram was also created and a scheme of interaction between the server and client parts of the software was described.

The automatic data update module has been enabled. To analyze stock quotes, technical and fundamental analysis modules, as well as a news analysis module, were implemented. As part of the work, user interfaces for the desktop version of the product, as well as for Android applications, were implemented.

The results of testing the quality of the product and the quality of system performance are obtained in the work.

An economic analysis of the developed information and analytical system was also carried out, within the framework of which the calculation of labor costs and material costs for the development and production of products was made. Based on the analysis, a feasibility study for this project was presented.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	7
1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ.....	9
1.1 Анализ предметной области	9
1.2 Анализ существующих решений.....	12
1.3 Описание используемых технологий.....	16
1.4 Анализ источников данных	17
1.5 Выбор методов формализованного анализа.....	24
1.6 Техническое задание.....	27
2 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОЕКТА	33
2.1 Информация о проекте	33
2.2 Расчет трудозатрат и заработной платы исполнителей	34
2.3 Расчет затрат на материальное обеспечение разработки.....	38
2.4 Общие затраты на разработку системы	38
2.5 План маркетинга	40
2.6 Управленческие и коммерческие расходы.....	41
2.7 Финансовый план.....	42
3 ПРОЕКТИРОВАНИЕ АНАЛИТИЧЕСКОЙ СИСТЕМЫ	46
3.1 Разработка архитектуры аналитической системы.....	46
3.2 Описание структурных элементов ИАС.....	47
3.3 Функциональная схема работы программы.....	50
3.4 Разработка модуля технического анализа фондового рынка	53
3.5 Разработка модуля фундаментального анализа фондового рынка....	55
3.6 Реализация модуля анализа влияния новостей на фондовый рынок.	58
3.7 Агрегирование результатов анализа	61

3.8	Проектирование диаграммы потоков данных	62
3.9	Проектирование структуры базы данных	63
3.10	Разработка API серверной части аналитической системы	66
3.11	Разработка пользовательского интерфейса для web-сайта и мобильного приложения.....	69
4	ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ.....	72
4.1	Тестирование информационно-аналитической системы.....	72
4.2	Оценка скорости работы системы.....	75
5	ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ	77
5.1	Анализ уязвимостей и угроз для разрабатываемой ИАС	78
5.2	Реализация средств обеспечения безопасности в клиентских приложениях	80
5.3	Реализация средств обеспечения безопасности в серверных приложениях	81
5.4	Технические средства обеспечения безопасности	84
5.5	Описание организационных мер обеспечения безопасности.....	85
	ЗАКЛЮЧЕНИЕ	86
	СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	87
	Приложение	89
	Приложение 1. Листинг	89

ВВЕДЕНИЕ

В наше время фондовый рынок стремительно развивается. Инвесторам для эффективного вложения средств нужна оперативная информация о том, где купить акции, по какой стоимости, как котировки ценных бумаг (например, стоимость акций Газпрома, Сбербанка, ВТБ, Роснефти) меняются в течение торговой сессии. Также вопрос инвестиций в России становится всё более популярным для обычных граждан.

Понимание поведения инвесторов на финансовом рынке в целом и на фондовом рынке, в частности, всегда было и остается важной и актуальной проблемой для стабильности современной глобальной экономики. Существует немало примеров, вошедших в историю, когда неправильное представление или игнорирование тех или иных событий влекло к значительным убыткам не только для участников торгов, но и для компаний, акции которых обращаются на биржевом рынке, и даже для людей, которые никаким прямым образом не участвуют в данной системе. Примером может послужить «крах Уолл-стрит» 1929 года, ставший началом Великой депрессии. Другой пример: глобальный финансовый кризис 2007-2008 годов, который привел к спаду на рынке жилья, падению бизнеса и безработице.

Целью данной работы является моделирование информационно-аналитической системы анализа фондового рынка.

Для достижения данной цели необходимо решить ряд **задач**, таких как:

- анализ существующих систем анализа фондовых рынков;
- анализ источников данных, определение необходимых математических методов, наиболее подходящих для решения данной задачи;
- организация базы данных и заполнение её достаточным объемом информации;
- разработка аналитического модуля, позволяющего прогнозировать изменения на фондовых биржах;

- разработка программного модуля для актуализации имеющихся данных о котировках акций на фондовых биржах;
- разработка удобного и интуитивно понятного интерфейса для взаимодействия с информационно-аналитической системой.

Объектом исследования является изменения показателей акций различных компаний на фондовых рынках.

Предметом исследования являются подходы к автоматизации анализа изменения показателей акций компаний на фондовых рынках.

Методы и средства исследования. При реализации данной задачи используются методы системно-структурного анализа и декомпозиции предметной области, объектно-ориентированное программирование, использование методологии проектирования реляционных баз данных, формализованные методы анализа данных, работы с большими данными.

Практическую значимость работы составляет:

- разработанный программный модуль для актуализации данных о котировках акций на фондовых биржах;
- разработанный модуль анализа влияния новостей на изменения котировок акций;
- разработанный модуль для выполнения фундаментального и технического анализов фондового рынка на основе финансовой отчетности компаний;
- Выявленные в процессе анализа котировок зависимости изменения показателей акций на фондовых биржах от событий, происходящих внутри компаний, отрасли и в мире в целом.

1 АНАЛИТИЧЕСКИЙ РАЗДЕЛ

1.1 Анализ предметной области

Финансовый рынок – организованная или неформальная система торговли финансовыми инструментами. На финансовом рынке происходит мобилизация капитала, предоставление кредита, осуществление обменных денежных операций и размещение финансовых средств в производстве. А совокупность спроса и предложения на капитал кредиторов и заёмщиков разных стран образует мировой финансовый рынок.

Фондовый рынок – это организованный и регулируемый финансовый рынок, где продаются и покупаются ценные бумаги (деньги, акц и, имущество и другие ресурсы). Сама биржа является лишь торговой площадкой и ее задача – проведение сделок. Если раньше было необходимо физическое взаимодействие трейдеров, то современная биржа оперирует в онлайн-режиме, что зародило понятие высокочастотной алгоритмической торговли. Но, несмотря на интеграцию компьютерных технологий в данный процесс, основные концепции и стратегии трейдеров остались прежними за тем лишь исключением, что торги происходят автоматизировано и быстро.

Существует множество способов классификации фондовых рынков. Некоторые из возможных критериев:

- По виду ценных бумаг (рынок производных финансовых инструментов, рынок акций, рынок облигаций).
- По территориальному принципу (региональные, национальные и международные рынки).

Акция представляет собой ценную бумагу, выпускаемую акционерными обществами, коммерческими банками, биржами, концернами, корпорациями, фирмами, другими предприятиями разных форм собственности, без установленного срока обращения, удостоверяющую внесение средств на цели их развития или членство в акционерном обществе и дающую право её владельцу на получение части прибыли в виде дивиденда.

К выпуску акций различные организации прибегают, когда альтернативные возможности привлечения средств исчерпаны, а также при возникновении предприятий, во вновь зарождающихся перспективных отраслях, сферах, предпринимательства. Но акции не являются только долговыми финансовыми обязательствами, инструментов кредитования. Это один из весьма эффективных способов непосредственно заинтересовать работающих в конечных результатах их труда, активизировать участие трудящихся в управлении предприятием, развить творческую предпринимательскую жилку в каждом. Акционер уже не наёмный работник, а собственник, получивший возможность активно участвовать с правом решающего голоса в общих собраниях, на которых решаются вопросы управления предприятиями, стратегии его развития, распределения и использования дохода и прибыли.

Достаточно указать, что в США насчитывается примерно 50 миллионов акционеров, то есть собственников акций. По масштабу распространения акции в странах с рыночной экономикой стоят на том же уровне, что вклады в сберегательные банки экономике нерыночного типа. Акционерная форма организации производства характерна практически для всех организационных структур развитой рыночной экономики. Она используется и малыми, и средними, и крупными предприятиями. Преимущества и перспективность этой формы определяется, прежде всего, тем, что она интегрирует в себе различные формы собственности, сочетает коллективную собственность с частной. Владельцами акций могут быть и индивидуальное лицо, и коллектив, и организация, и государство. Есть основания утверждать, что в акционерной форме органично сплетаются в единое целое все формы собственности, благодаря чему удаётся преодолеть определённые противоречия между ними и использовать преимущества каждой формы в отдельности, хотя при этом возникают новые противоречия и трудности.

Специфика акционирования состоит в частном отчуждении собственности, а именно, имущественной собственности, вложенной в акцию,

от собственника и передачи её в распоряжение акционерного общества, представленного в качестве юридического лица. Акционер – это только владелец ценной бумаги, непосредственно имуществом он не распоряжается, он лишь в праве получать дивиденды и использовать своё право голоса на собраниях акционеров, которая ему даёт каждая акция. Имущественная же собственность акционерного предприятия обезличена, ею распоряжаются все акционеры, а точнее их собрания, правление, наёмная администрация. В этом и состоит единение личной (частной) и коллективной собственности, тем более что владельцами акций могут быть не только индивидуумы, но и организации.

Объектом собственности акционера является только акция. Только по отношению к ней он может реализовать отношения собственности: продать, подарить, передать по наследству. Однако акционер практически не является собственником акционерного капитала, имущества акционерного предприятия, поскольку непосредственно не распоряжается тем, как используется пай, вложенный им в виде денег, заплаченных за купленные акции. Акционерный капитал, имущество акционерного общества, являются объектом распоряжения, владения и ответственности со стороны акционерного общества как юридического лица. Средства, полученные от распространения акций, фактически безвозвратны, а, значит, могут быть вложены в наиболее долгосрочные и медленно окупаемые активы, под которые достаточно трудно найти другие заёмные средства.

Посредством акций осуществляется как перемещение самой собственности, так и контроль над ней от одних групп собственников к другим. Одновременно имеет место концентрация капитала. Владение контрольным пакетом акций позволяет тому, кто вложил в акции определяющую часть капитала, реализовать практически весь спектр отношений собственности по отношению к акционерному капиталу. По мере роста размера фирм, акционерная форма организации переживает определённую эволюцию. На малых и средних фирмах возможно

осуществление прямого контроля над деятельностью акционерного общества со стороны акционеров и налицо реальная персонификация собственника. Налицо и тенденция к распылению акционерного капитала. В этом случае зачастую контроль над деятельностью фирмы переходит из области прямого контроля в область косвенного регулирования, осуществляемого через рынок капитала (фондовые биржи) посредством купли-продажи акций. Развитие акционерной формы современной рыночной экономики наглядно демонстрирует тенденцию деперсонификации собственника как индивидуума, частного при сосредоточении её у иных субъектов. Особенно это характерно для нашего времени, когда концентрация контроля над акционерной собственностью сосредотачивается в руках

1.2 Анализ существующих решений

Сайт www.macrotrends.net

«Macrotrends» – это сайт, предоставляющий возможность проводить анализ показателей на фондовом рынке в режиме On-line. На сайте представлен перечень компаний, акции которых размещены на фондовой бирже, и финансовые показатели этих компаний (рисунок 1.1).

Stock Name	Ticker	Industry	Market Cap	Closing Price	1 Year % Change	P/E Ratio	Dividend Yield
Agilent Technologies	A	Electrical Test Equipment	\$34,734	116.28	-18.54%	25.01	0.72%
Alcoa	AA	Metal Production & Distribution	\$9,193	49.84	34.00%	5.48	0.80%
Ares Acquisition	AAC	Unclassified	\$1,229	9.83	0.31%	0.00	0.00%
ATA	AACG	Schools	\$33	1.04	-67.40%	0.00	0.00%
Armada Acquisition I	AACI	Unclassified	\$204	9.86		0.00	0.00%
Aadi Bioscience	AADI	Medical - Biomedical and Genetics	\$268	12.80	-58.78%	0.00	0.00%
Arlington Asset Investment Corp.	AAIC	REIT - Mortgage Trusts	\$96	3.24	-21.74%	16.20	0.00%
American Airlines Group	AAL	Transportation - Airlines	\$8,652	13.32	-43.39%	0.00	0.00%
Allsource Asset Management Corp.	AAMC	Finance - Investment Management	\$23	24.51	21.19%	4.85	0.00%
Atlantic American	AAME	Insurance - Accident & Health	\$60	2.96	-30.55%	22.74	0.68%
Aaron's	AAN	Consumer Products - Miscellaneous Staples	\$526	17.14	-52.36%	5.12	2.63%
Applied Optoelectronics	AAOI	Semiconductors	\$55	1.98	-77.47%	0.00	0.00%
AAON	AAON	Building Products - Air Conditioning & Heating	\$2,757	51.95	-19.10%	44.03	0.73%
Advance Auto Parts	AAP	Retail & Wholesale Auto Parts	\$10,442	172.19	-11.81%	14.06	3.48%
Apple	AAPL	Computers - Minicomputers	\$2,134,502	131.88	4.15%	21.41	0.70%
Accelerate Acquisition	AAQC	Unclassified	\$489	9.78	0.51%	0.00	0.00%
American Assets Trust	AAT	REIT - Retail Equity Trusts	\$1,762	29.44	-21.39%	13.44	4.35%
Autoscope Technologies	AATC	Technology Services	\$29	5.37	-21.23%	25.55	8.94%
Almaden Minerals	AAU	Gold Miners	\$34	0.41	-55.94%	0.00	0.00%
Advantage Energy	AAVF	Other Alternate Energy	\$1,585	2.50		16.35	0.00%

Рисунок 1.1 – Интерфейс главной страницы сайта www.macrotrends.net

К плюсам данной системы можно отнести лаконичный дизайн интерфейса, использованный минимализм позволяет сосредоточиться на анализируемых данных и дольше сохранять концентрацию. В системе хранятся исторические данные об изменениях стоимости акций компаний на фондовой бирже за период более, чем 10 лет, что обеспечивает возможность проведения более точного анализа. Большинство данных, представленных на сайте визуализированы в виде диаграмм и графиков, благодаря чему появляется возможность визуального анализа данных.

Также стоит отметить ряд минусов системы. Одним из минусов системы является отсутствие русского и других языков, что ограничивает круг лиц, способных пользоваться функционалом данной системы. Также можно заметить, что дизайн сайта плохо адаптирован под различные разрешения экранов и различных устройств, в связи с чем появляются ограничения на использование данной системы на мобильных устройствах. Наиболее важным минусом системы является медленный процесс актуализации данных, данные о ценных бумагах на бирже обновляются с задержкой от 7 до 15 дней, в редких случаях до месяца. Такая задержка влияет на способность принимать своевременные решения, при анализе фондового рынка. Кроме того, на сайте не представлены компании из России, что не позволяет в полной мере производить анализ фондовых бирж.

Сайт www.tradingview.com

TradingView – это ресурс для осуществления технического анализа ценных бумаг на фондовом рынке. Наибольший интерес вышеупомянутый ресурс представляет с точки зрения визуального анализа данных. На сайте представлено большое количество различных диаграмм, графиков и других графических средств предоставления информации о состоянии ценной бумаги на фондовом рынке (рисунок 1.2).

Кроме того, на сайте реализовано множество инструментов для работы с графиками, что позволяет аналитику быстрее получать необходимые сведения. Также стоит отметить, что на сайте представлена информация в том

числе и о российских компаниях, что позволяет производить исследования актуальные для отечественного бизнеса.

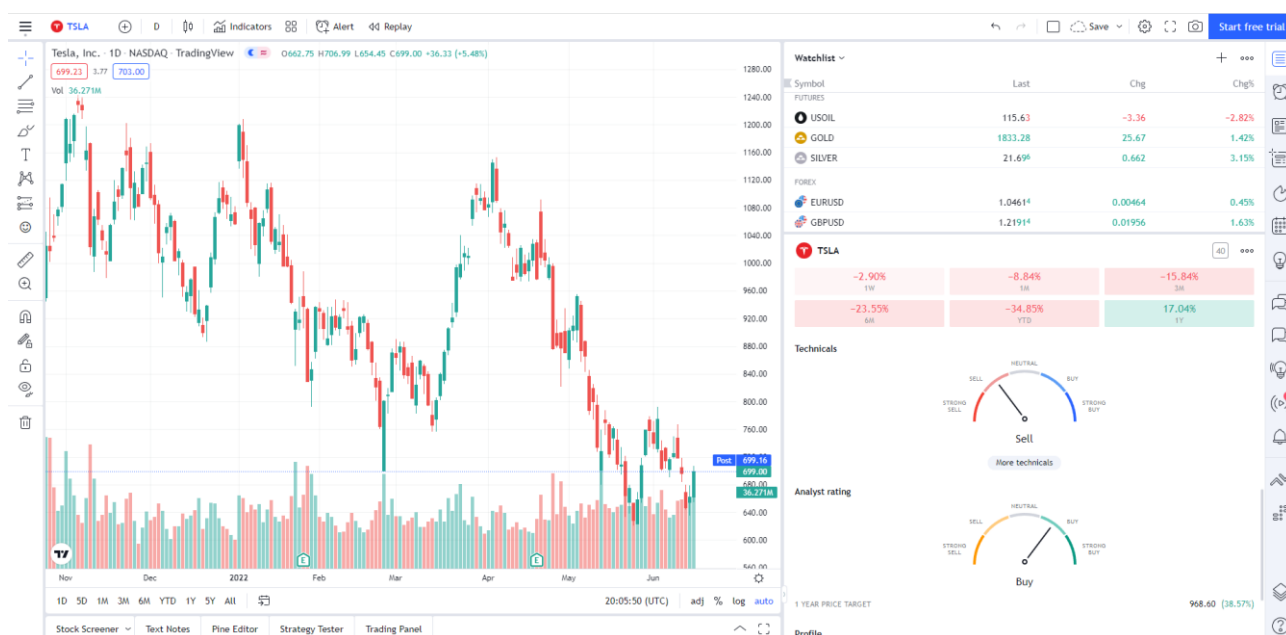


Рисунок 1.2 – Интерфейс сайта www.tradingview.com

Доступ к ресурсам сайта предоставляется в нескольких режимах: свободный доступ, платная подписка и оформление премиум-аккаунта. В зависимости от выбранного режима доступа пользователь получает возможность пользоваться различными аналитическими инструментами. Однако стоит отметить, что бесплатный доступ предоставляет достаточно большой перечень возможностей анализа, которого хватает для осуществления базовых исследований фондового рынка.

Сайт www.finviz.com

Finviz – сайт для проведения анализа ценных бумаг на фондовых биржах. Одной из главных особенностей данного сайта является наличие удобного инструмента для сравнения различных акций на фондовых биржах. Пользователю предоставляется большой перечень параметров для настройки фильтрации и сортировки списка акций с возможностью последующего экспорта полученной сравнительной таблицы (рисунок 1.3).

Из минусов сайта можно отметить отсутствие в перечне ценных бумаг акций российских компаний. Также стоит отметить неудачно подобранный

шрифт без засечек. Так как на данном сайте используется достаточно маленький размер шрифта совместно с небольшим межстрочным интервалом, при долгом чтении таблиц и больших текстов глаза быстрее устают.

The screenshot shows the Finviz website interface. At the top, there's a search bar and navigation links. Below that, a filter section allows users to refine their search by various criteria like Exchange, Market Cap, Earnings Date, etc. The main table displays a list of stocks with columns for No., Ticker, Company, Sector, Industry, Country, Market Cap, P/E, Price, Change, and Volume. The table is sorted by Market Cap in descending order.

No.	Ticker	Company	Sector	Industry	Country	Market Cap	P/E	Price	Change	Volume
1	A	Agilent Technologies, Inc.	Healthcare	Diagnostics & Research	USA	35.68B	27.97	116.47	0.40%	1,873,338
2	AA	Alcoa Corporation	Basic Materials	Aluminum	USA	9.46B	13.07	51.11	4.48%	9,658,295
3	AAAU	Goldman Sachs Physical Gold ETF	Financial	Exchange Traded Fund	USA	-	-	18.19	1.28%	1,232,088
4	AAC	Ares Acquisition Corporation	Financial	Shell Companies	USA	1.23B	98.20	9.83	0.10%	101,397
5	AACG	ATA Creativity Global	Consumer Defensive	Education & Training Services	China	32.27M	-	1.01	-2.88%	16,224
6	AACJ	Armada Acquisition Corp. I	Financial	Shell Companies	USA	203.99M	-	9.85	0.00%	413
7	AACIW	Armada Acquisition Corp. I	Financial	Shell Companies	USA	-	-	0.22	0.00%	0
8	AADI	Aadi Bioscience, Inc.	Healthcare	Biotechnology	USA	285.80M	-	13.46	2.91%	98,226
9	AADR	AdvisorShares Dorsey Wright ADR ETF	Financial	Exchange Traded Fund	USA	-	-	51.66	1.93%	5,711
10	AALC	Arlington Asset Investment Corp.	Real Estate	REIT - Mortgage	USA	94.38M	-	3.18	1.92%	62,406
11	AAL	American Airlines Group Inc.	Industrials	Airlines	USA	8.85B	-	13.31	2.78%	49,127,230
12	AAME	Atlantic American Corporation	Financial	Insurance - Life	USA	58.31M	8.53	2.89	1.40%	5,794
13	AAN	The Aaron's Company, Inc.	Industrials	Rental & Leasing Services	USA	532.98M	5.80	16.49	-0.90%	333,446
14	AAOI	Applied Optoelectronics, Inc.	Technology	Semiconductors	USA	57.46M	-	1.98	2.06%	351,710
15	AAON	AAON, Inc.	Industrials	Building Products & Equipment	USA	2.93B	46.23	52.21	0.38%	134,759
16	AAP	Advance Auto Parts, Inc.	Consumer Cyclical	Specialty Retail	USA	10.70B	19.09	172.29	0.47%	835,618
17	AAPL	Apple Inc.	Technology	Consumer Electronics	USA	2235.08B	21.58	135.43	2.01%	90,303,565
18	AAQC	Accelerate Acquisition Corp.	Financial	Shell Companies	USA	390.80M	-	9.77	-0.05%	30,009
19	AAT	American Assets Trust, Inc.	Real Estate	REIT - Diversified	USA	1.76B	45.93	29.53	2.53%	299,507
20	AATC	Autoscope Technologies Corporation	Technology	Scientific & Technical Instruments	USA	29.32M	12.63	5.41	-0.18%	6,779

Рисунок 1.3 – Страница для сравнения акций на сайте www.finviz.com

К плюсам данной системы можно отнести автоматическое построение линий тренда и линий поддержки курса на графике изменения стоимости акции компании, что позволяет аналитику не тратить дополнительное время на самостоятельное построение данных линий.

Также стоит отметить наличие на сайте тепловых карт индексов акций (рисунок 1.4). Данный инструмент позволяет аналитику в удобной наглядной форме быстро оценить состав рынка, для принятия инвестиционного решения или для решения аналитических задач.

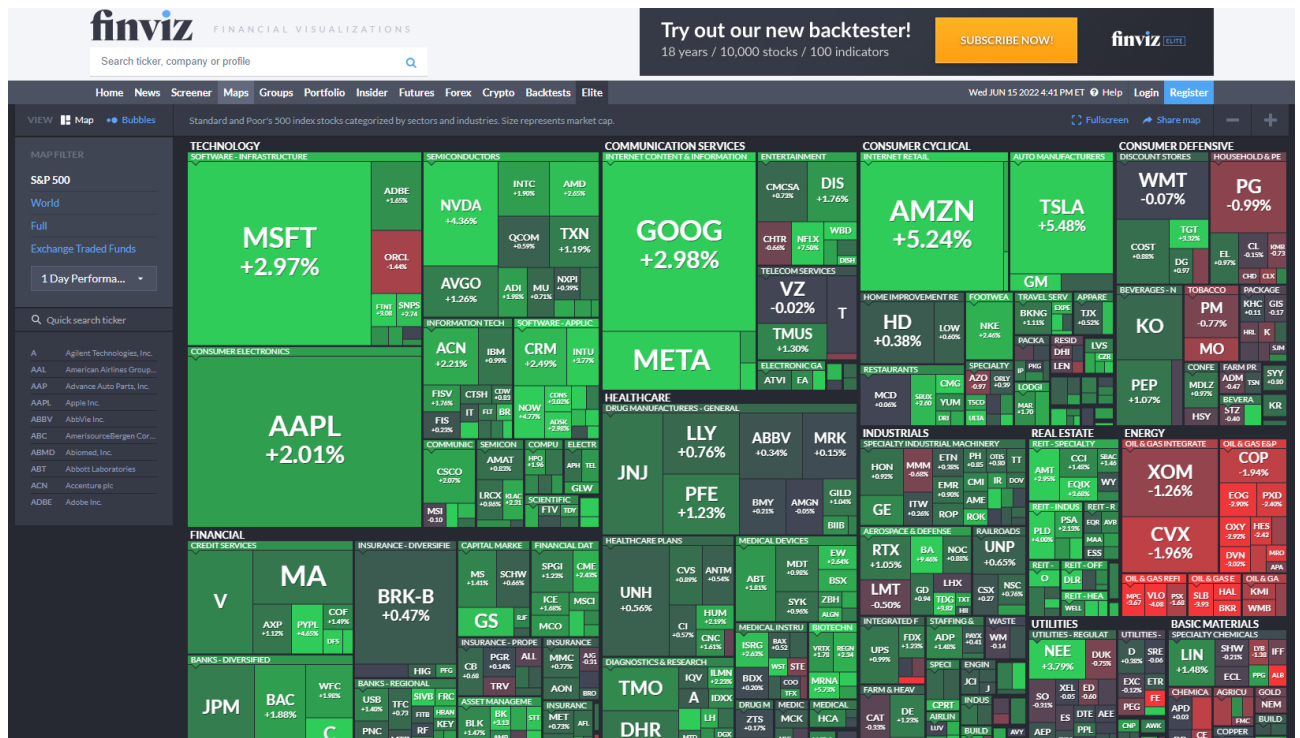


Рисунок 1.4 – Страница с тепловой картой индексов акций на сайте www.finviz.com

1.3 Описание используемых технологий

Одним из важных принципов при выполнении курсового проекта является использование современных технологий и методов. При разработке серверной части программного продукта было принято решение использовать платформу ASP.NET. Данная платформа имеет компонентно-ориентированную архитектуру, что обеспечивает широкий возможности для масштабирования разрабатываемого программного продукта. Кроме того, платформа поддерживает использование строго типизированных языков программирования, благодаря чему обеспечивается более удобная отладка программного кода и скорость работы системы. Также стоит отметить удобное взаимодействие с СУБД.

При проектировании системы важно придерживаться технологии декомпозиции. Декомпозиция подразумевает под собой использование подхода, при котором задачи разделяются на меньшие подзадачи. Кроме того, использование принципа декомпозиции при разработке информационно-аналитической системы позволит разделить процесс проектирования одной

большой системы, на проектирование маленьких и узконаправленных подсистем, взаимодействующих друг с другом.

При реализации программной части ИАС необходимо руководствоваться принципами объектно-ориентированного программирования. Из плюсов данного подхода можно выделить сокращение количества данных, передаваемых между компонентами системы; возможность легко модифицировать и расширять программный код; уменьшение количества дублированного программного кода.

Также было принято решение в качестве объекта для хранения данных использовать реляционную базу данных. Использование реляционной базы данных позволяет обеспечить структуру хранимых данных таким образом, чтобы данные не дублировались и оставались согласованными. Это достигается благодаря использованию нормальных форм. Другим преимуществом использования реляционной базы данных является возможность легкого преобразования данных к форме OLAP-куба, что облегчает работу при большом объеме хранимых данных.

В виду того, что в разрабатываемой системе подразумевается сбор, обработка и хранение больших данных, при их анализе целесообразно использовать методы Data Mining. В частности, в системе данные методы применяются для работы с данными о котировках акций.

Кроме того, в системе используются алгоритмы машинного обучения. Данная технология нашла свое применение в первую очередь в задачах классификации больших объемов данных, а также в задачах анализа текстовой информации.

1.4 Анализ источников данных

В качестве возможных источников данных были рассмотрены несколько порталов, агрегирующих информацию о фондовых биржах. Более приоритетными в данном случае являлись порталы, которые предоставляют

арі для взаимодействия с информационной системой, содержащей данные о фондовых биржах.

Сайт www.moex.com

Данный портал включает в себя данные московской биржи (рисунок 1). На данном портале доступны статические данные о рынках (режимы торгов и их группы, финансовые инструменты и их описание), данные для построения графиков («свечей»), сделки (анонимно), котировки, итоги торгов.

Преимуществом данного портала является наличие программного интерфейса для доступа к информационно-статистическому серверу, благодаря чему становится возможным осуществлять взаимодействия с сервером по протоколу http. Данные с сервера возвращаются в формате xml, а примеры запросов, а также перечень всех возможных запросов удобно оформлен в руководстве для разработчиков.

Недостатком данного сайта является необходимость получения разрешения на действия связанные с получением информации с сайта, для использования полученных данных необходимо оставить заявку и дождаться её одобрения. Также одним из ключевых недостатков данного портала является наличие данных всего одной биржи, а следовательно, при использовании данного источника полученные данные будут ограничены.

ММОСКОВСКАЯ БИРЖА

пятница, 09.07.2021 01:06

Поиск

#	Инструмент	Наименование цб	ISIN	Номинал	Валюта номинала	Объем выпуска	Дата погашения
1	AAL-RM	AmAirlines	US02376R1023	0,0100	USD	641 383 123	-
2	AAPL-RM	Apple	US0378331005	0,0000	USD	16 687 631 000	-
3	ABBV-RM	AbbVie	US00287Y1091	0,0100	USD	1 766 222 336	-
4	ABRD	АбрайДюпко	RU000A0JS5T7	1,0000	RUB	98 000 184	-
5	ABT-RM	Abbott	US0028241000	0,0000	USD	1 776 820 148	-
6	ACKO	ACKO ao	RU000A0JXS91	1,0000	RUB	536 000 000	-
7	ADBE-RM	Adobe	US00724F1012	0,0001	USD	476 400 000	-
8	AFKS	Система ao	RU000A0DQZE3	0,0900	RUB	9 650 000 000	-
9	AFLT	Аэрофлот	RU0009062285	1,0000	RUB	2 444 535 448	-
10	AGRO	AGRO-гдр	US7496552057	0,0000	USD	0	-
11	AKCH	ETF AKCH	RU000A102FK8	0,0000	RUB	-	-
12	AKEU	AKEU ETF	RU000A100Q43	0,0000	EUR	-	-
13	AKMB	ETF AKMB	RU000A101PN3	0,0000	RUB	-	-
14	AKMD	ETF AKMD	RU000A1036A4	0,0000	RUB	-	-
15	AKME	ETF AKME	RU000A102E78	0,0000	RUB	-	-
16	AKNX	AKNX ETF	RU000A0ZZTQ4	0,0000	USD	-	-
17	AKRN	Акрон	RU0009028674	5,0000	RUB	40 534 000	-
18	AKSP	AKSP ETF	RU000A1006V3	0,0000	USD	-	-
19	ALBK	БестЭфБ ao	RU000A0JS2M9	10,0000	RUB	56 490 000	-
20	ALNU	АЛРОСА-Нюр	RU000A0JP468	25,0000	RUB	800 000	-
21	ALRS	АЛРОСА ao	RU0007252813	0,5000	RUB	7 364 965 630	-
22	AMAT-RM	AppliedMtr	US0382221051	0,0100	USD	913 980 756	-
23	AMCC	AMCC ETF	RU000A102XW6	0,0000	USD	-	-
24	AMD-RM	AMD Inc.	US0079031078	0,0100	USD	1 215 020 976	-
25	AMDG	AMDG ETF	RU000A1030P5	0,0000	USD	-	-
26	AMEM	AMEM ETF	RU000A102XZ9	0,0000	USD	-	-
27	AMEZ	АшинскийМЗ	RU000A0B88G6	1,0000	RUB	498 454 822	-
28	AMGF	AMGF ETF	RU000A102Y17	0,0000	USD	-	-
29	AMGM	AMGM ETF	RU000A102X08	0,0000	USD	-	-

Рисунок 1.5 – Скриншот сайта www.moex.com

Сайт www.finam.ru

Данный портал является агрегатором многих бирж (рисунок 2). Однако он создан для информирования пользователей о ситуации на фондовой бирже и рекомендаций по поводу эффективности вложений. В связи с чем, данный сайт не имеет доступного программного интерфейса для удобного получения данных. Также сложность при получении данных заключается в том, что в основе сайта лежат javascript-скрипты. В результате чего необходимо затрачивать дополнительное время на поиск скрипта с необходимыми данными, а также написание сложного регулярного выражения для получения этих данных.

Данный сайт предоставляет возможность скачать данные о котировках определенной акции в формате csv, однако этих данных крайне недостаточно для полноценного анализа и их необходимо дополнять. Из преимуществ данного портала можно отметить отсутствие CAPTCHA, в результате чего

появляется возможность более быстро осуществлять парсинг переходя на разные страницы сайта.

Кроме того, преимуществом сайта является наличие большого объема данных о котировках на разных фондовых биржах, а также информация о событиях, происходящих внутри компаний, новостях, с которыми связаны данные компании, а также в целом события, происходящие в отрасли, частью которой является конкретная компания. В результате чего при анализе становится возможным не только отслеживать значения котировок в зависимости от даты, а также учитывать события, происходившие в данное время и их влияние на изменение курсов.

Финам

Финансы

Облигации

Аналитика

Приложения

Comon

Whotrades

Вебинары

Еврооблигации

Итоги 2020

Ещё

Про рынок

Брокерские услуги

Банк

Управление активами

Форекс

Обучение

О компании

Инвестировать

Демо

+7 (495) 1-346-346

ещё

Новости

Котировки

Главное

Аналитика: обзор рынка и прогнозы

Аналитика: обзор рынка и прогнозы - Комментарии

Анонсы

Важные даты

Единая лента

Ежедневный обзор мировых рынков

Календарь статистики

Календарь статистики

Комментарии по облигациям

Корпоративные новости

Криптоновости и комментарии

Неделя на рынке

Новости и комментарии

Новости компании

Новости компании Финам Форекс

Новости компаний

Новости международных рынков

Новости по облигациям

Публикации в прессе

Разбор полетов

Специалисты и подопытные

Задержка — 15 минут. Регистрируйтесь и получайте данные в реальном времени!

Инструмент	Посл. сделка	% Изм. цены	Открытие	Макс.	Мин.	Закрытие	Объем, шт.	Время обн.
☆ TCS-гдр	6673.0	-0.11%	6719.4	6719.4	6551.2	6680.6	237212	08.07, 23:49
☆ Авангрд-ао	1135P	+0.44%	1120P	1135P	1115P	1130P	43	08.07, 18:38
☆ Арсагера	6.06P	-1.14%	6.18P	6.18P	6.03P	6.13P	8000	08.07, 17:58
☆ АСКО ао	5.54P	0.00%				5.7P		01.01, 03:00
☆ БестЭфБ ао	65.0P	+1.56%	63.5P	69.5P	63.5P	64.0P	6410	08.07, 18:33
☆ БСП ао	65.41P	-1.00%	66.00P	66.35P	64.74P	66.07P	259050	08.07, 18:45
☆ БСП ап	39.78P	0.00%	38.8P	39.78P	38.55P	38.8P	14 520	01.01, 03:00
☆ ВТБ ао	0.048695P	-1.43%	0.049100P	0.049165P	0.048400P	0.049400P	45351200000	08.07, 23:49
☆ ГАЗ-сервис	293P	0.00%				293P		18.05, 00:00
☆ ГАЗКОН-ао	295P	0.00%				295P		08.07, 00:00
☆ ИКРУСС-ИНВ	59.5P	+0.68%	59.5P	60.1P	59.5P	59.1P	600	08.07, 18:15
☆ КузнецкийБ	0.02704P	+9.25%	0.02530P	0.02970P	0.02505P	0.02475P	1444440000	08.07, 18:46
☆ МКБ ао	6.589P	-0.33%	6.606P	6.612P	6.510P	6.611P	67117700	08.07, 23:48
☆ МосБиржа	172.48P	+0.56%	171.00P	172.90P	170.55P	171.52P	6364660	08.07, 23:49
☆ МосОблБанк	0.698P	0.00%				0.698P		08.07, 00:00
☆ ОКС ао	1070P	-1.83%	1070P	1080P	1060P	1090P	53	08.07, 18:13
☆ Приморье	50000P	+0.40%	50000P	50600P	49800P	49800P	11	08.07, 16:37
☆ PFC CK ао	0.2172P	-0.46%	0.2168P	0.2224P	0.2168P	0.2182P	662000	08.07, 18:46
☆ РДБанк ао	222P	0.00%	224P	224P	220P	222P	160	08.07, 17:58
☆ Росбанк ао	78.2P	0.00%	78.6P	80.0P	77.4P	78.2P	1170	08.07, 17:52
☆ САФМАР ао	470.0P	-1.76%	481.2P	481.4P	470.0P	478.4P	2550	08.07, 18:13
☆ Сбербанк	302.63P	-0.47%	303.00P	304.30P	300.62P	304.06P	30771300	08.07, 23:49
☆ Сбербанк-п	280.50P	-0.36%	280.95P	282.00P	278.45P	281.50P	3363250	08.07, 23:49
☆ Система ао	31.155P	-1.63%	31.820P	31.820P	31.014P	31.670P	34628500	08.07, 23:49
☆ УралСиб ао	0.0863P	-4.43%	0.0895P	0.0907P	0.0850P	0.0903P	23940000	08.07, 18:48
☆ Alliance Data Systems Corporation	100.1200	-3.15%	99.9500	104.4700	99.4000	103.3800	33454	08.07, 22:59
☆ Alliance Data Systems Corporation	100.00\$	-3.40%	102.38\$	104.26\$	99.24\$	103.52\$	1028	09.07, 01:20

Рисунок 1.6 – Скриншот сайта www.finam.ru

Платформа Yahoo! Finance

Данная платформа является агрегатором финансовых данных, разработанным компанией «Yahoo!». В частности, данный портал предоставляет данные о котировках акций, экономические характеристики компаний на биржевых рынках, а также различные новости, оказывающие влияние на экономическую ситуацию в мире (рисунок 1.7).

Информация, расположенная на сайте, позволяет проводить полноценный технический и фундаментальный анализ котировок акций. Из плюсов можно отметить наличие открытого API. Сервер возвращает запрашиваемые данные в формате JSON, что делает их удобными для обработки и последующего сохранения в базе данных.

Главным минусом данной платформы является её направленность на международные финансы, т.е. данные предоставляются на английском языке, кроме того, на платформе представлены только международные компании и международные биржи. Данный факт ограничивает полноту данных и ограничивает возможности полного анализа.

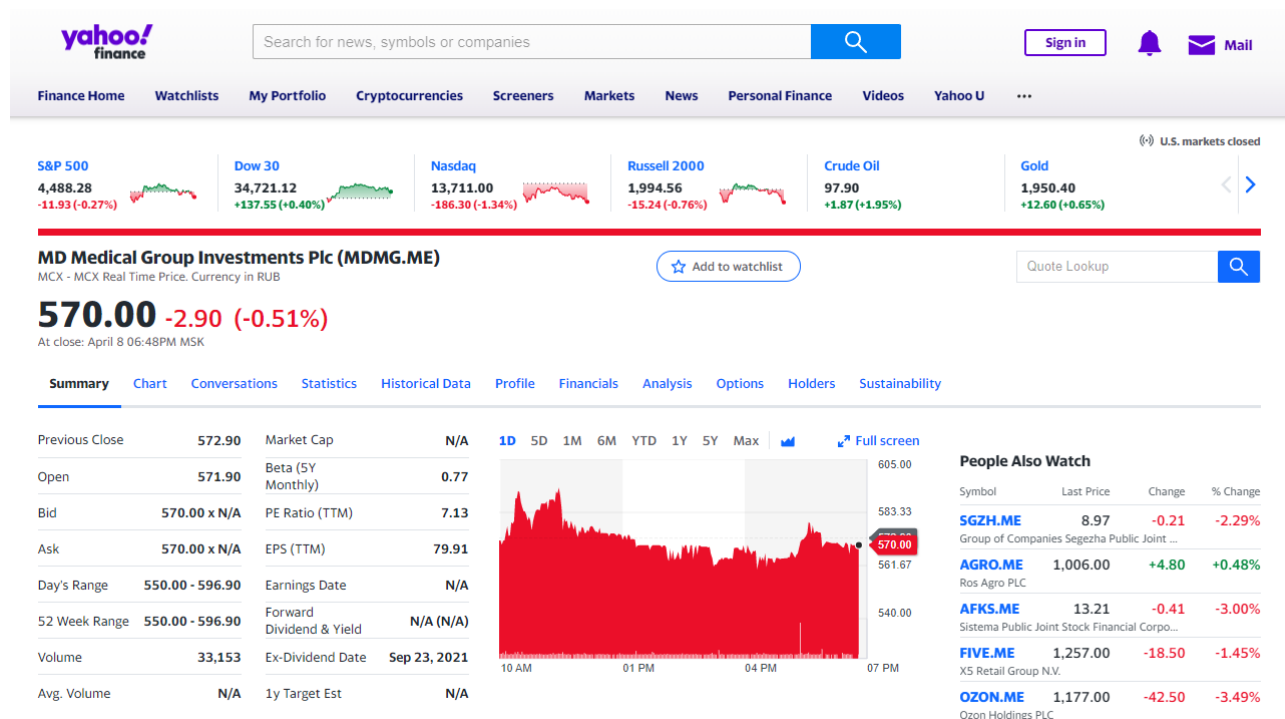


Рисунок 1.7 – Скриншот платформы Yahoo! Finance

Платформа Tinkoff

Платформа Tinkoff была разработана для работы с ценными бумагами, посредством различных финансовых инструментов. Поэтому кроме того, что данная платформа позволяет получать данные о котировках акций, фондовых биржах и т.д., она также предоставляет доступ к инструментам по управлению финансами, например покупки и продажи акций. Кроме того, важным

аспектом является наличие на данной платформе данных не только о международных компаниях и их акциях, но и данных об отечественных компаниях, что могло бы дополнить набор данных для исследований и увеличить качество анализа.

Для взаимодействия с платформой существует API, данный интерфейс не предоставляет документацию по использованию. Однако путем анализа web-сайта Tinkoff-инвестиции можно получить достаточную информацию для использования данного программного интерфейса.

Кроме того, разработчиками Tinkoff созданы и активно поддерживаются библиотеки для работы с платформой, написанные для различных языков программирования, в том числе для C#. Наличие библиотек является самым оптимальным инструментом для получения и обработки данных.

Из минусов можно отметить тот факт, что библиотеки имеют узкую область применения и направлены главным образом на получения сведения о котировках акций и работе с ними. При этом экономические характеристики компаний, а также прочие события внутри компаний, оказывающие влияние на поведение её акций на фондовых биржах, не могут быть получены при помощи данной библиотеки.

Сайт www.rbc.ru

«РБК Новости» – это проект группы компаний РБК, предназначенный для публикации мировых и региональных новостей. Новости и статьи на сайте разделены на категории, например новости спорта, общественные новости, финансовые новости и т.д. Также материалам, размещенным на сайте присвоены ключевые слова, так называемые теги, благодаря чему расширяются возможности поиска информации, а также её группировки и классификации.

Минусом данного портала является то, что он не предоставляет библиотеки или API для взаимодействия с информацией, размещенной на сайте. В связи с этим, для получения данных в автоматическом режиме необходимо обрабатывать html-код страниц. Стоит также отметить, что ввиду

особенностей реализации html-код сайта имеет сложную структуру, что усложняет написание регулярных выражений для получения необходимой информации из html-кода страниц.

Интернет-издание «Лента.ру»

Лента.ру – это российское новостное интернет-издание. Данный портал входит в пятерку самых посещаемых европейских новостных сайтов, а также занимал первые места в номинации «Информационный сайт года». Информация на данном сайте сгруппирована по категориям, например «Культура», «Спорт», «Экономика» и др.

К минусам данного сайта можно отнести то, что он не предоставляет API для взаимодействия с контентом, поэтому получения текста новостей может производиться путем обработки html-кода страниц. Также новостям и статьям на портале не проставляются теги, в связи с чем определение ключевых слов текста должно быть реализовано на программном уровне.

Плюсами данного сайта является наличие архива новостей, в котором новости сгруппированы по дате публикации, что облегчает автоматический сбор новостей. Кроме того, стоит отметить большую информационную базу портала, а именно в архиве сайте присутствуют тексты новостей за период более, чем 20 лет. Также в виду особенностей реализации сайт имеет простую структуру html-кода, что позволяет обрабатывать код страницы и получать необходимые данные за короткое время, используя меньше вычислительных ресурсов.

Проведя анализ источников данных, был сделан вывод, что наиболее оптимальным способом получения данных является агрегация информации из различных источников.

Было принято решение, что в качестве инструмента для получения данных о котировках акций, оптимальным вариантом является библиотека от разработчиков Tinkoff. Так как данная методика будет иметь наименьшее затраты на интеграцию с программным кодом и при этом появляется возможность получать данные не только о международных компаниях, но и о

российских, что даст возможность анализировать фондовые рынки, ориентируясь на специфику России, как отдельного субъекта международных финансовых отношений.

Так как для полноты анализа данных только о котировках акций недостаточно, для получения вспомогательных данных о компаниях было принято решения использовать открытое API платформы Yahoo! Finance. Такой выбор был сделан, потому что данная платформа имеет самый большой набор экономических характеристик компаний, в сравнении с вышеперечисленными источниками данных и при этом имеет удобный в использовании программный интерфейс, передающий данные в структурированном виде.

В качестве источника мировых новостей было выбрано новостное интернет-издание «Лента.ру». Такой выбор был сделан в виду того, что на сайте представлено большое количество новостных статей в период с 31 августа 1999 года, что позволит получить большой объем новостных текстов, который в последствии может быть использован как материал для формирования набора данных для обучения нейронной сети. Кроме того, данные на портале сгруппированы по дате публикации, что обеспечит удобный поиск необходимых новостей при автоматическом получении данных. Полученные данные будут представлять из себя список новостей, включающих заголовки и текст новости, а также дату и время её публикации.

1.5 Выбор методов формализованного анализа

Одной из основных задач, которые должна решать разрабатываемая ИАС, является задача прогнозирования стоимости акции. При декомпозиции задачи прогнозирования были выделены три под задачи: прогнозирование на базе технического анализа, фундаментального анализа и анализа новостей.

Технический анализ акций компании представляет собой систему прогнозирования цен, основанную на информации, полученной в результате рыночных торгов. Иными словами, в основе технического анализа лежит

выделение и изучение определенных закономерностей в движении графика котировок, то есть принятое решение основывается только на графическом изображении линии тренда [1].

Технический анализ инвестиционных свойств ценных бумаг включает множество инструментов, но основные факторы, от которых он зависит – это объем торгов, динамика цен и исторические данные.

Одним из инструментов для формализации технического анализа фондового рынка является использование методов машинного обучения, а именно обученной нейронной сети. Так как существует множество разновидностей нейронных сетей, необходимо определить наиболее подходящий вид нейронных сетей для решения поставленной задачи.

Для определения типа используемой нейронной сети необходимо определить тип входных и выходных данных. Данные, на которых основывается технический анализ, представляют из себя непрерывную линию тренда, состоящую из набора связанных точек, которые соответствуют цене акции конкретной компании в единицу времени. Результатом работы нейронной сети должно являться число, соответствующее прогнозируемой цене акции компании. Иными словами, решение поставленной задачи можно свести к решению задачи прогнозирования временных рядов.

Для задачи прогнозирования временных рядов необходимо использовать рекуррентные или сверточные нейронные сети. Их особенность заключается в том, что на каждой итерации работы нейронной сети учитываются результаты анализа, полученные на предыдущих итерациях. При этом для прогнозирования временных рядов стоимости акций компании наилучшим образом подойдут рекуррентные нейронные сети типа LSTM (Long Short-Term Memory). Потому что данные нейросети при работе позволяют учитывать более ста предыдущих значений временного ряда [2].

Данные, полученные на предыдущей итерации нейронной сети, принято обозначать как h_{t-1} , входные данные на текущем шаге – x_t , функция активации обозначается как σ . В качестве расчетной функции используется

гиперболический тангенс $\tanh()$. Таким образом значение текущей итерации (h_t) нейронной сети рассчитывается как:

$$h_t = \tanh (\delta(h_{t-1}, x_t)) \quad (1)$$

Для обучения рекуррентной нейронной сети необходимо сформировать обучающий набор данных. Для решения задачи технического анализа в качестве исходным данных используются исторические данные о стоимости акции компании. При этом базовой характеристикой чаще всего выбирают стоимость акции на момент закрытия торгов, однако оперировать можно также стоимостью акции на момент открытия торгов или средней стоимостью акции за определенный период. Для формирования обучающей выборки необходимо разделить данные временного ряда на равные блоки, при этом результирующим значением будет являться следующее после блока значение временного ряда.

Для осуществления анализа новостей и оценки их влияния на стоимость акции необходимо решить задачи, связанные с анализом естественного языка. В частности, необходимо в тексте определить перечень компаний, упоминаемых в новости, а также оценить тональность текста новости относительно каждой из обнаруженных компаний.

Для определения тональности текста необходимо произвести предварительную обработку. В данном случае на первом этапе необходимо выполнить лемматизацию текста. Лемматизация – процесс приведения словоформы к лемме – её нормальной (словарной) форме.

Затем необходимо произвести очистку текста от «шума». В данном этапе происходит удаление стоп-слов. Стоп-слова – это набор нерелевантных слов, обычно это артикли, междометия, союзы и т.д., которые не несут смысловой нагрузки [5].

В результате предыдущих шагов текст новости приводится к виду векторов слов. Затем необходимо все слова объединить в единый словарь и отсортировать, тем самым получить вектор-словарь.

Определение тональности текста представляет из себя классификацию текста. Для решения данной задачи наилучшим образом подойдут алгоритмы машинного обучения. Реализация заключается в формировании модели-классификатора, обучении её на тренировочной выборке и использования для классификации на реальных примерах, в данном случае в качестве обучающего набора данных был использован размеченный датасет финансовых новостей [6].

1.6 Техническое задание

1. Общие сведения

1.1. Наименование программы

Наименование программы: «Информационно-аналитическая система анализа фондового рынка».

1.2. Назначение и область применения

Программа предназначена для прогнозирования изменения показателей акций различных компаний на фондовых рынка.

2. Требования к программе

2.1. Требования к функциональным характеристикам

1. необходимо реализовать компоненты для сбора и обработки данных об акциях на фондовом рынке;
2. необходимо обеспечить возможность просмотра данных с различных бирж;
3. необходимо обеспечить возможность сбора и обработки информации о мировых новостях и новостях, связанных с компаниями, а также учитывать их при анализе биржевых рынков;
4. необходимо обеспечить возможность просмотра истории показателей фондовых бирж;
5. система должна взаимодействовать с обученной нейронной сетью по прогнозированию показателей фондового рынка;

6. необходимо реализовать вывод рекомендаций при работе с акциями на фондовом рынке;

7. необходимо реализовать взаимодействие с базой данных.

8. Должен быть разработан интерфейс клиентских приложений для компьютеров под управлением ОС Windows и смартфонов под управлением ОС Android.

2.2. Требования к надёжности

2.2.1. Требования к обеспечению надёжного функционирования программы

Надёжное функционирование программного обеспечения может быть обеспечено только при условии выполнения заказчиком совокупности организационно-технических мероприятий:

- 1) Следование приложенной к программному обеспечению инструкции.
- 2) Обеспечение бесперебойной работы виртуального / выделенного сервера с серверной частью программного обеспечения с помощью обеспечения бесперебойного питания сервера и планового резервирования данных.

2.2.2. Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технического оборудования или ошибочным завершением работы операционной системы, не должен превышать 5 минут.

Время восстановления после отказа, вызванного сбоем технических средств, не должно превышать времени, требуемого на устранение неисправности.

2.2.3. Отказы из-за некорректных действий пользователей системы

Отказ программы вследствие некорректных действий администратора (ввод некорректных исходных данных) при взаимодействии пользователя с программой недопустим.

3. Условия эксплуатации

3.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых программа должна продолжать функционировать, должны удовлетворять требованиям, предъявляемым к техническим средствам по их эксплуатации.

3.2. Требования к квалификации и численности персонала

Для обеспечения работы серверной части программного обеспечения достаточно одного человека - системного администратора, проверяющего настройки, журналы ПО и БД.

3.3. Требования к составу и параметрам технических средств

Требования к клиенту:

1. Процессор Intel Pentium G4500 или новее.
2. Оперативная память объёмом 2 Гб или больше.
3. Постоянный накопитель объёмом от 120 Гб.
4. Операционная система Windows 10.
5. Net Framework 4.7 или новее.

Требования к серверу:

1. Процессор Intel Xeon E-2246G или новее.
2. Оперативная память объёмом 8 Гб или больше.
3. Постоянный накопитель объёмом от 256 Гб.
4. Операционная система Windows Server 2016 или новее.
5. Net Framework 4.7 или новее.

3.4. Требования к информационной и программной совместимости

3.4.1 Требования к информационным структурам и методам решения

Пользовательский интерфейс программного обеспечения должен быть интуитивно понятен и иметь классический стиль оформления меню программы.

Данные о котировках акций должны храниться в базе данных и при необходимости дополняться актуальными данными из интернета.

3.4.2. Требования к исходным кодам и языкам программирования

Исходные коды программного обеспечения должны быть созданы на языке программирования C#. Используемая СУБД - MySQL.

3.4.3. Требования к программным средствам, используемым программой

Системные программные средства, реализованные программой, должны быть представлены лицензионной версией операционной системы Windows 10.

3.4.4 Требования к защите информации и программ

Разрабатываемое программное обеспечение должно отвечать требованиям, утвержденным в следующих нормативно-правовых актах:

- ФЗ №149 от 14 июля 2006 г. "Об информации, информационных технологиях и о защите информации".
- ФЗ №3523-1 от 23 сентября 1992 г. "О правовой охране программ для электронных вычислительных машин и баз данных".
- ФЗ №110 от 19 июля 1995 г. "Об авторском праве и смежных правах" (изменения внесены по постановлению N 207-СФ от 7 июля 2004 года).

3.5. Специальные требования

Специальные требования к программному обеспечению не предъявляются.

4. Требования к программной документации

4.1. Предварительный состав программной документации.

Состав программной документации должен включать в себя:

1. Техническое задание.
2. Текст программы.
3. Описание программы.
4. Руководство пользователя.
5. Руководство по решению возможных проблем.

5. Стадии и этапы разработки

5.1. Стадии разработки

Разработка должна быть проведена в 3 этапа:

1. Разработка технического задания.
2. Рабочее проектирование.
3. Внедрение.

5.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. Разработка программы.
2. Разработка программной документации.
3. Испытание программы.

На стадии внедрения должен быть выполнен этап разработки, подготовка и передача программы.

5.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. Постановка задачи.
2. Определение и уточнение требований к техническим средствам.
3. Определение и уточнение требований к программе.
4. Определение стадий, этапов и сроков разработки программы и

программной документации.

5. Согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. Разработка, согласование и утверждение методики испытаний.
2. Проведение приёмо-сдаточных испытаний.
3. Корректировка программы и программной документации по результатам испытаний.

На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на оборудовании заказчика.

6. Порядок контроля и приёмки

6.1. Виды испытаний

Приёмо-сдаточные испытания должны проводиться согласно разработанной исполнителем и согласованной с заказчиком программой, и методикой испытаний.

Ход проведения приёмо-сдаточных испытаний заказчик и исполнитель документируют в протоколе проведения испытаний.

6.2. Общие требования к приёмке работы

На основе протокола проведения испытаний исполнитель совместно с заказчиком подписывают акт приёмки-сдачи программы в эксплуатацию.

2 ЭКОНОМИЧЕСКОЕ ОБОСНОВАНИЕ РАЗРАБОТКИ ПРОЕКТА

Важным этапом разработки информационно-аналитической системы является оценка ее экономической эффективности. Необходимо определить целесообразность данного проекта с точки зрения финансовой привлекательности и произвести расчёт затрат на проектирование, внедрение и поддержку разрабатываемого программного продукта.

2.1 Информация о проекте

Разрабатываемая система представляет из себя платформу для агрегации данных фондового рынка для их последующего анализа. Платформа позволяет за короткое время получить информацию об акциях компаний, аналитические данные для проведения самостоятельного анализа, а также результаты автоматического анализа, включающие в себя прогнозирование изменения стоимости акции, оценка экономической эффективности компании и рекомендации к покупке или продаже акций.

Информационная-аналитическая система имеет несколько пользовательских интерфейсов. Наиболее полный функционал реализован в web-версии продукта, данная версия в первую очередь рассчитана на пользователей ПК, однако она также удобна в случае использования браузера на мобильном устройстве.

Также проект представлен в виде мобильного приложения для операционной системы на базе Android. Приложение позволяет пользователю получить быстрый доступ к необходимой информации о результатах анализа фондового рынка.

Данный программный продукт будет пользоваться спросом экономических специалистов, предоставляя им достаточно полный объем данных для проведения самостоятельного анализа фондового рынка, а также получение результатов автоматического анализа в короткие сроки, позволяющие уменьшить время на принятие решения.

Данный продукт также будет полезен для пользователей, которые обладают небольшим опытом и знаниями в экономической сфере, но при этом являются участниками фондового рынка. Благодаря встроенному модулю автоматического анализа, пользователь разрабатываемой системы сможет в короткие сроки получить достаточно точные результаты прогнозирования изменения стоимости акций компаний, тем самым у компаний появляется возможность привлечь новых инвесторов и увеличить количество участников на фондовом рынке.

На данный момент аналоги разрабатываемой информационно-аналитической системы реализуют только автоматическую агрегацию данных фондового рынка, при этом анализ данных или отсутствует полностью, или представлен в виде статей, написанных профессиональными экономистами, поэтому пользователь может увидеть результаты анализа только по истечению какого-то промежутка времени и данные могут быть уже неактуальными. Поскольку в разрабатываемой системе анализ проводится в автоматическом режиме, то такой проблемы не возникает.

2.2 Расчет трудозатрат и заработной платы исполнителей

2.2.1 План производства

Для определения затрат на реализацию проекта необходимо разработать план создания программного продукта и выделить основные этапы. На основании перечня этапов необходимо рассчитать трудозатраты, а также сроки создание проекта в целом и каждого этапа в частности.

В рамках разработки данной информационно-аналитической системы необходимо выполнить следующие подзадачи:

- анализ предметной области;
- Исследование существующих аналогов;
- формирование перечня функциональных требований, к разрабатываемому продукту;
- проектирование базы данных и реализация в СУБД;

- анализ уязвимостей и описание средств информационной безопасности;
- создание диаграммы объектов, описание схемы взаимодействия серверной и клиентской части ПО;
- реализация модуля обновления данных;
- разработка модуля анализа котировок акций;
- проектирование интерфейса для клиентской части ПО;
- реализация интерфейса пользователя для десктопного приложения;
- реализация интерфейса пользователя для Android приложения;
- оптимизация, тестирование и отладка приложения.

Для расчёта трудозатрат на реализацию каждого из этапов необходимо определить количество специалистов, участвующих в разработке проекта. В данном случае разработка ведётся одним человеком и расчёты производились исходя из данного факта (таблица 2.1).

Таблица 2.1 – План разработки проекта

№ этапа	Этап работ	Начало периода	Конец периода	Длительность, рабочих дней
1	Анализ предметной области	01.09.2022	09.09.2022	10
2	Исследование существующих аналогов	12.09.2022	23.09.2022	10
3	Формирование перечня функциональных требований, к разрабатываемому продукту	26.09.2022	26.09.2022	1
4	Проектирование базы данных и реализация в СУБД	27.09.2022	28.09.2022	2
5	Анализ уязвимостей и описание средств информационной безопасности	29.09.2022	05.10.2022	5
6	Создание диаграммы объектов, описание схемы взаимодействия серверной и клиентской части ПО	06.10.2022	06.10.2022	1

Окончание таблицы 2.1

№ этапа	Этап работ	Начало периода	Конец периода	Длительность, рабочих дней
7	Реализация модуля обновления данных	07.10.2022	20.10.2022	10
8	Разработка модуля анализа котировок акций	21.10.2022	08.12.2022	35
9	Проектирование интерфейса для клиентской части ПО	09.12.2022	13.12.2022	3
10	Реализация интерфейса пользователя для десктопного приложения	14.12.2022	15.12.2022	2
11	Реализация интерфейса пользователя для Android-приложения	16.12.2022	19.12.2022	2
12	Оптимизация, тестирование и отладка приложения	20.12.2022	21.12.2022	2
Итого:				83

Исходя из разработанного плана производства, разработка программного продукта займёт четыре месяца включая выходные дни, без учета выходных разработка займёт 83 дней. Для визуализации плана разработки используется диаграмма Ганта (рисунок 2.1).

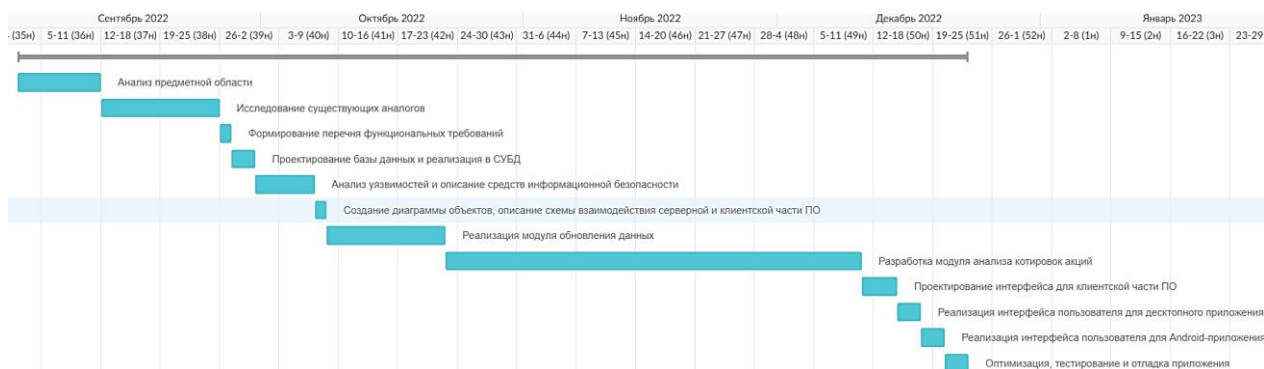


Рисунок 2.1 – Диаграмма Ганта для календарного плана

2.2.2 Расчет заработной платы

Заработная плата рассчитывается по следующей формуле:

$$ЗП_{\text{мес}} = \frac{M \cdot T}{\tau_p \cdot t_{p.d.}} \left(1 + \frac{\pi}{100} \right) \text{ рублей,} \quad (2)$$

где M – зарплата за месяц (рублей), T – общие трудозатраты (чел.ч), $Ч_p$ – количество рабочих дней в месяц, $t_{p.д.}$ – продолжительность рабочего дня в часах, $П$ – процент премии. В данной работе $t_{p.д.} = 8$ ч., $П = 0$, $Ч_p$ рассчитывается в зависимости от календарного месяца.

На основании анализа рынка труда, было определено, что средняя заработная плата программистов в г. Брянске в 2022 году составила 42 000 рублей. На основе представленных данных следует произвести расчёт заработной платы для разработчика проекта (таблица 2.2).

Таблица 2.2 – Расчет заработной платы за разработку проекта

№	Месяц	Исполнитель	Месячная заработная плата, руб.	Рабочих дней в месяц	Трудозатраты, человеко-часов в месяц	ЗП, руб.
1	Сентябрь 2022	Разработчик	42 000	22	176	42 000
2	Октябрь 2022	Разработчик	42 000	21	168	42 000
3	Ноябрь 2022	Разработчик	42 000	21	168	42 000
4	Декабрь 2022	Разработчик	42 000	22	120	30 000
	Итого:					156 000

Общие затраты на заработную плату разработчика составили 156 000 рублей.

Исходя из полученной суммы заработной платы, необходимо рассчитать НДФЛ:

$$\text{НДФЛ} = \frac{\text{ЗП}}{87} * 13 = 23\,310,345 \text{ рублей.} \quad (3)$$

Сумма НДФЛ для заработной платы разработчика на весь период разработки составил 23 310,345. Далее необходимо рассчитать сумму социальных взносов:

$$\text{Соц. выплаты} = (\text{ЗП} + \text{НДФЛ}) * 31\% = 55\,586. \quad (4)$$

Социальные выплаты составляют 55 586 рублей от суммы заработной платы разработчика. Суммарные затраты на заработную плату составляют 234 896 рублей.

2.3 Расчет затрат на материальное обеспечение разработки

2.3.1 Амортизационные расходы

При разработке программного продукта был использован персональный компьютер стоимостью 75 000 рублей. В постановление Правительства РФ от 01.01.2022 №1 (ред. от 18.11.2022) «О Классификации основных средств, включаемых в амортизационные группы» указано, что персональные компьютеры относятся к ОКОФ 330.28.23.23 «Машины офисные прочие» и имеют срок амортизации 2-3 года.

Исходя из этих данных, можно рассчитать амортизационные отчисления на срок разработки проекта:

$$A = M * \frac{P}{36} = 4 * \frac{75\,000}{36} = 8\,333 \text{ рублей}, \quad (5)$$

где M – кол-во месяцев использования оборудования, P – стоимость оборудования.

При создании информационно-аналитической системы использовалось бесплатное ПО, в связи с этим расходы на программное обеспечение равны нулю.

2.4 Общие затраты на разработку системы

Для расчёта финансовых затрат на разработку и внедрение информационно-аналитической системы необходимо учитывать следующие составляющие:

- заработная плата программиста – ЗП;
- страховые взносы – $H_{зп}$;
- амортизация используемых активов – A ;
- модернизацию и приобретение основных средств – $P_{\text{мод}}$.

Для определения себестоимости суммируем показатели, описанные выше (таблица 2.3).

Таблица 2.3 – Себестоимость проекта

№	Элементы себестоимости	Сумма, руб	Доля, %
1	Заработная плата разработчика	156 000	63
2	Отчисления на социальные нужды	78 896	32
3	Амортизация активов	8 333	4
4	Расходы на Интернет	3 120	1



Рисунок 2.2 – Диаграмма себестоимости

Таким образом себестоимость разрабатываемого продукта: $C_{п.п.} = 246\,349$ рублей. Диаграмма себестоимости (рисунок 2.2) позволяет сделать выводы, о том, что основной статьёй расхода является заработная плата разработчика, а также страховые и налоговые отчисления, что составляет 95% всей стоимости программного продукта.

2.5 План маркетинга

2.5.1 Стратегия маркетинга

Данный программный продукт позволяет автоматизировать работу финансовых аналитиков, исходя из этого можно сделать вывод, что данное ПО будет востребовано как для специалистов в финансовой сфере, так и у начинающих инвестор.

Исходя из анализа существующих аналогов, можно сделать вывод, что системы анализа фондового рынка в проектах применяются довольно редко, зачастую они реализованы посредством статей от финансовых аналитиков. В свою очередь разрабатываемая система обеспечивает гораздо более быстрый анализ фондового рынка, основываясь на большом количестве разнообразны данных.

Разрабатываемый продукт будет представлен в общем доступе, также будет возможность получить доступ к арі платформы для интеграции со сторонним ПО. Существует возможность оформления подписки, для получения доступа к арі, после оплаты пользователь получает токен, который необходимо передавать при каждом запросе к арі. Подписка доступна в течении месяца, после чего необходимо доступ прекращается и необходимо осуществить продление подписки для дальнейшего использования арі.

Для определения стоимости был проведён анализ платформ-агрегаторов данных о фондовых рынках, предоставляющих арі по подписке, средняя стоимость подписки составила 9 000 рублей в месяц. С целью увеличения конкурентоспособности разрабатываемого ПО, было принято решение снизить стоимость относительно средней стоимости подписки конкурентов. Цена подписки составила 7 500 рублей в месяц.

2.5.2 План продаж

План продаж необходим для отображения предполагаемой выручки от продаж на определенный период (таблица 2.4).

Таблица 2.4 – План продаж

Показатели	Квартал				Всего
	1	2	3	4	
Ожидаемый объем продаж, ед.	15	20	25	30	90
Цена с НДС, тыс. руб.	9	9	9	9	36
Выручка с НДС, тыс. руб.	135	180	225	270	810
Нетто выручка, тыс. руб.	112,5	150	187,5	225	675
Сумма НДС, тыс. руб.	22,5	30	37,5	45	135

2.6 Управленческие и коммерческие расходы

Для поддержания функционирования разрабатываемого продукта необходимо наличие виртуального сервера, на котором будет размещена серверной часть приложения. Затраты на аренду виртуального сервера составляют 2 300 рублей в месяц.

Для обслуживания выделенного сервера необходимо иметь обученный персонал в лице системного администратора, так же для взаимодействия с клиентами и осуществления продаж необходимо наличие в штате менеджера по продажам. Результаты расчета управленческих и коммерческих расходов приведены в таблице 2.4.

Таблица 2.5 – Управленческие затраты

Затраты	Квартал				Всего
	1	2	3	4	
Зарплата системного администратора	30 000	30 000	30 000	30 000	120 000
Зарплата менеджера по продажам	25 000	25 000	25 000	25 000	100 000
Социальные отчисления	27 830	27 830	27 830	27 830	111 320
Аренда сервера	2 300	2 300	2 300	2 300	9 200
Итого:					340 520

Общие затраты управленческие и коммерческие затраты составили 340 520 рублей за 4 квартала.

2.7 Финансовый план

Бизнес-план любой компании включает в себя финансовое планирование. В данном разделе происходит агрегация рассчитанных данных из других разделов и определяется прибыльность реализуемого проекта.

2.7.1 План прибылей и убытков

Для отображения доходов и расходов необходимо составить план прибылей и убытков на период реализации проекта (таблица 2.6).

Таблица 2.6 – План прибылей и убытков

Показатели	Квартал				Всего
1. Выручка нетто	135	180	225	270	810
2. Постоянные затраты					
2.1. Постоянные общепроизводственные затраты	2,3	2,3	2,3	2,3	9,2
2.2. Постоянные управленческие и коммерческие затраты	82,83	82,83	82,83	82,83	331,32
3. Прибыль до налогообложения	49,87	94,87	139,87	184,87	469,48
4. Налог на прибыль	9,97	18,97	27,97	36,97	93,88
5. Чистая (нераспределённая) прибыль	39,9	75,9	111,9	147,9	375,6
6. Капитализируемая прибыль					
6.1 Резервы	39,9	75,9	111,9	147,9	375,6

Из плана прибылей и убытков видно, что чистая прибыль проекта за 4 квартала составит 375,6 тысяч рублей.

2.7.2 План движения денежных средств

План движения денежных средств отражает потоки инвестиционной, финансовой и операционной деятельности в денежном эквиваленте, благодаря чему появляется возможность рассчитать размер финансовых затрат и произвести прогнозирование объём получаемых денежных средств.

На основе плана движения денежных средств (таблица 2.7) можно рассчитать базовые показатели эффективности проекта, такие как:

- рентабельность инвестиций;
- чистая текущая стоимость;
- внутренняя норма рентабельности проекта;
- срок окупаемости инвестиций.

Таблица 2.7 – План движения денежных средств

Показатели	Квартал				Всего
1. Остаток денежных средств на начало периода	0	75,9	111,9	147,9	335,7
2. Поступление денежных средств					
2.1. Поступление от продажи продукции	49,87	94,87	139,87	184,87	469,48
3. Всего наличие денежных средств	49,87	170,77	251,77	332,77	805,18
4. Выбытие денежных средств					
4.2. Оплата общепроизводственных расходов	2,3	2,3	2,3	2,3	9,2
4.3. Оплата управленческих и коммерческих расходов	82,83	82,83	82,83	82,83	331,32
5. Уплата налогов					
5.1. НДС	22,5	30	37,5	45	135
5.2. Отчисления на соц. нужды	27,83	27,83	27,83	27,83	111,32
5.3. Налог на прибыль	14,196	14,27	14,49	14,64	57,596
6. Всего оттоки	42,026	42,1	42,32	42,47	168,916
7. Чистый денежный поток	7,844	128,67	209,45	290,3	636,264

Отношения среднегодовой прибыли к суммарным инвестиционным затратам в проект называется рентабельностью инвестиций (ROI) и рассчитывается по формуле:

$$ROI = \frac{\sum_{t=1}^T P_t}{I}, \quad (6)$$

где P_t – чистая прибыль от проекта в году t , T – количество лет в периоде, I – величина инвестиционных затрат. В данном случае под инвестиционными затратами будем понимать полную стоимость разработанного программного продукта. Рассчитав рентабельность инвестиций $ROI = 375,6/246,349 = 1,5247$, получим значение рентабельности в 52,47%. Инвестиция считается рентабельной, если процент рентабельности превышает процент по вкладам в банках.

Чистая текущая стоимость проекта NPV (Net Present Value) рассчитывается как разность дисконтированных денежных потоков поступлений и платежей, производимых в процессе реализации проекта за весь инвестиционный период:

$$NPV = \sum_{t=1}^T \frac{NCF_t}{(1+R)^t} - I_0, \quad (7)$$

где I_0 – единовременные затраты на реализацию проекта, NCF_t – чистый денежный поток, R – ставка дисконтирования. Ставку дисконтирования возьмем равную 10%. Рассчитаем значение чистой стоимости проекта:

$$NPV = \frac{636,264}{(1 + 0,1)} - 246,349 = 389,915.$$

Финансирование проекта считается целесообразным, если значение чистой стоимости проекта является положительным числом. Так как значение чистой стоимости проекта не равно нулю, то рассчитывать внутреннюю норму рентабельности нет необходимости.

Также необходимо определить дисконтированный период окупаемости. Он представляет собой денежный поток превысит сумму вложений. В нашем случае он составляет 3 квартала.

Таким образом основные рассчитанные коэффициенты представлены в таблице 2.8.

Таблица 2.8 – Показатели эффективности проекта

№	Название	Значение
1	Рентабельность инвестиций	52,47%
2	Чистая стоимость проекта	389,915 тыс. руб.
3	Период окупаемости	3 квартала

Рассчитанные показатели показывают высокую экономическую выгоду от инвестиций в разработку данного проекта.

На этапе проектирования системы была разработана структура информационно-аналитической системы (рисунок 3.1).

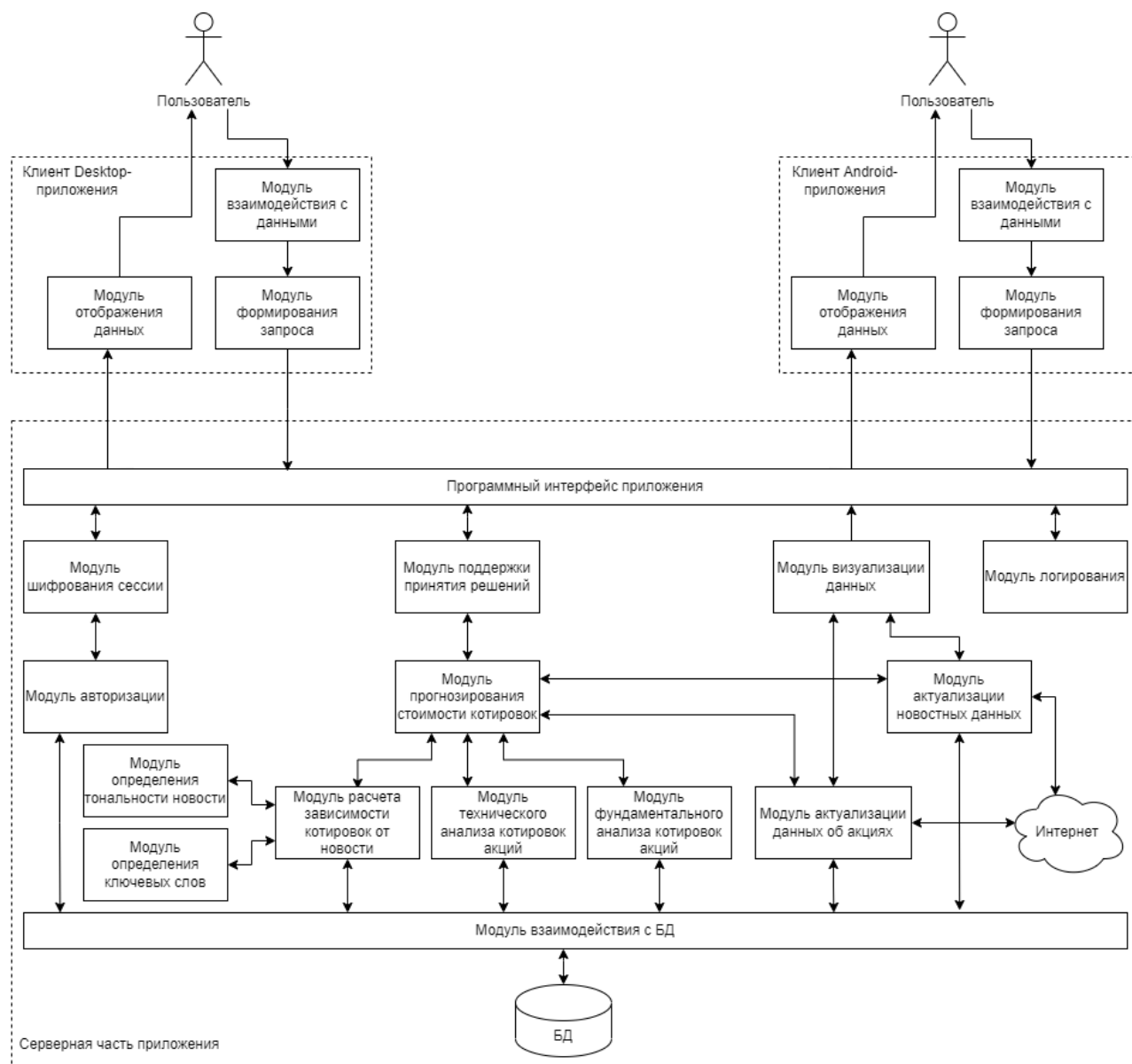


Рисунок 3.1 – Структура ИАС

В качестве архитектуры программного решения была выбрана архитектура «клиент-сервер». Данная архитектура подразумевает наличие серверного устройства, которое способно поддерживать одновременную работу с несколькими клиентами.

В рамках данной архитектуры на серверной части реализуется бизнес-логика приложения, модуль анализа данных, а также модуль взаимодействия с базой данных. В свою очередь модуль взаимодействия с базой данных взаимодействует с сервером СУБД, которая производит манипуляции с данными в базе.

3.2 Описание структурных элементов ИАС

Информационная система должна иметь модульную структуру. Это позволит производить манипуляции внутри отдельных модулей, не внося изменения в другие модули. Также это обеспечит возможность расширения системы путем добавления новых модулей.

3.2.1 Описание серверной части ИАС

Программный интерфейс приложения – это модуль, который служит промежуточным звеном между клиентской и серверной частями приложения. В данном случае реализован универсальный интерфейс для взаимодействия с сервером. Благодаря чему, существует возможность обеспечить доступ к серверной части приложения, не зависимо от структуры и вида клиентской части.

Модуль авторизации реализует возможность идентификации пользователя, для предоставления ему персональной информации и его сохраненных данных. Так как при авторизации используется ввод пароля, то сессия пользователя должна быть зашифрована.

Модуль шифрования необходим для того, чтобы конфиденциальные данные пользователя не передавались по сети в открытом виде. Данный модуль позволяет при необходимости шифровать данные перед их отправкой и дешифровать при получении.

Модуль расчета зависимости котировок акций от новостей служит для выявления степени влияния новости на изменения стоимости акции. Функционирование данного модуля напрямую связана с модулями анализа

текста новости. Данные, полученные в результате расчетов, фиксируются в базе данных.

Для анализа текста новости применяются модуль определения тональности текста и модуль определения ключевых слов. Под определением тональности текста подразумевается поиск в тексте новости обособленных сущностей и определение эмоционального отношения, с которым данные сущности упоминаются. Модуль определения ключевых слов позволяет выделить ключевые слова из текста новости, для проведения более полноценного анализа.

Модули технического и фундаментального анализа необходимы для автоматизации соответствующих видов финансового анализа. Результаты работы данных модулей представляют их себя рассчитанные коэффициенты, значения мультипликаторов, а также значения индикаторов, влияющие на стоимость акции.

Задача модуля прогнозирования стоимости котировок акции заключается в расчете будущего значения стоимости акции исходя из данных и показателей, полученных от вышеупомянутых модулей. Кроме того, данный модуль обеспечивает взаимодействие с модулями актуализации данных, что позволяет гарантировать актуальность результатов работы данного модуля.

Модуль поддержки принятия решений реализует формирование комплекса рекомендаций, влияющих на конечный выбор пользователя. Исходя из данных прогноза модуль предлагает алгоритм действий пользователю, подкрепляя данные рекомендации результатами расчётов и анализа.

Модули актуализации новостных данных и данных об акциях обеспечивают поддержание актуального состояния базы данных. За счет автоматического мониторинга данных в сети Интернет. Кроме того, данные модули являются источниками данных для модуля визуализации данных. Являясь промежуточным звеном для информации из базы данных, они при

необходимости дополняют данные актуальной информацией и затем сохраняют ей в базе данных.

Модуль визуализации данных необходим для формирования структуры и формата данных для последующей передачи их клиенту.

Модуль логирования осуществляет мониторинг работы системы и фиксацию в лог-файл аномальных состояний, а также состояний, при которых появляется угроза нормального функционирования программного продукта. Записи в лог-файле необходимы для определения причин перехода системы в аномальные или аварийные состояния, с целью как можно быстрее внести соответствующие корректировки в настройки система или программный код.

3.2.2 Описание клиентской части ИАС

Клиентская часть разрабатываемой системы подразумевает универсальную структуру вне зависимости от вида реализации клиентской части разрабатываемого продукта. Структура клиентской части включает в себя три модуля.

Модуль взаимодействия с данными определяет возможность пользователя управлять данными системы посредством понятных для пользователя действий. Тем самым данный модуль реализуют интерфейс взаимодействия пользователя с системой.

Модуль формирования запросов позволяет на основе действий пользователя сформировать запрос на получение или изменение данных. Данный модуль осуществляет связь клиентской и серверной частей программного продукта.

Для демонстрации данных используется модуль отображения данных. Функция данного модуля заключается в интерпретации данных с сервера и представлении их в удобном для пользователя виде. В частности, данный модуль обеспечивает возможность пользователя взаимодействовать с различными диаграммами, таблицами и графиками.

3.3 Функциональная схема работы программы

При создании структурной схемы необходимо использовать стандарт IDEF0 (ICAM Definition – integrated computer aided manufacturing definition).

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

Данный стандарт позволяет представить программный комплекс в виде набора функциональных блоков, каждый из которых может осуществлять взаимодействие с другими функциональными блоками посредством четырех видов интерфейса (входа, управления, механизма, выхода).

Интерфейс входа (слева) описывает исходные данные или объекты для выполнения функций. Интерфейс управления (сверху) описывает правила и ограничения. Интерфейс механизма (снизу) описывает ресурсы, используемые в процессе выполнения функции (ресурсы не должны изменяться). Интерфейс выхода (справа) описывает данные или объекты, являющиеся результатом выполнения функции.

Схема IDEF0 делится на несколько уровней, первый уровень (рисунок 3.2) представляет из себя один блок «Прогноз стоимости акции». На вход поступают данные от пользователя, а именно название выбранной компании и период, на который будет производится анализ. Выходными данными является результат прогноза стоимости акции.

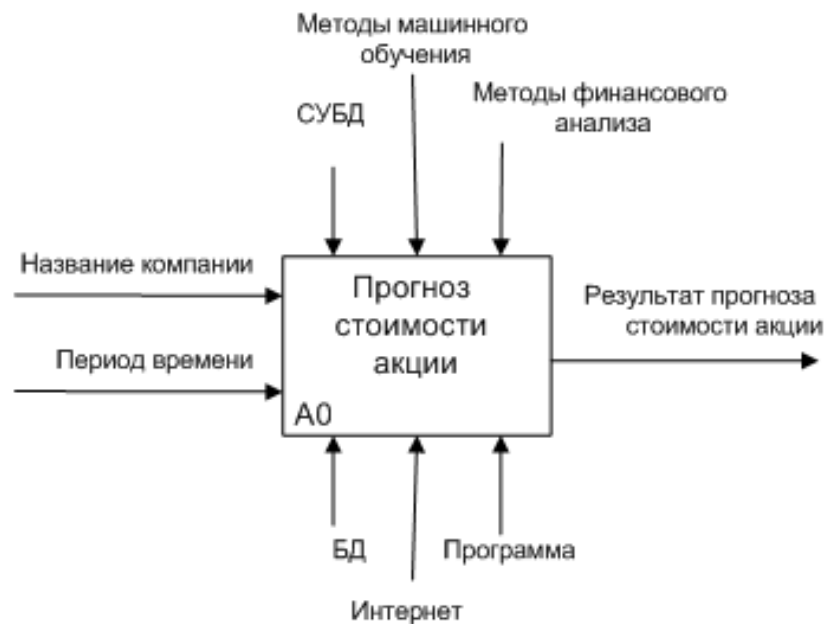


Рисунок 3.2 – Функциональная диаграмма системы (уровень 1)

На втором уровне детализации функциональной диаграммы процессы работы программы отображены более детально (рисунок 3.3). На данном уровне представлены блок по формированию данных для анализа, блок с расчета влияния новостей на стоимость акции, блок технического анализа котировок, блок фундаментального анализа компании и блок прогнозирования стоимости акции.

На этапе формирования данных для анализа программа получает входные данные от пользователя и необходимые данные из базы данных и из интернета.

На следующих этапах проводится комплексный анализ данных, а именно происходит расчет изменения стоимости акции в зависимости от новостей за определенный временной период. Оценка проводится с использованием методов машинного обучения, в данном случае эти методы реализованы в виде обученной нейронной сети.

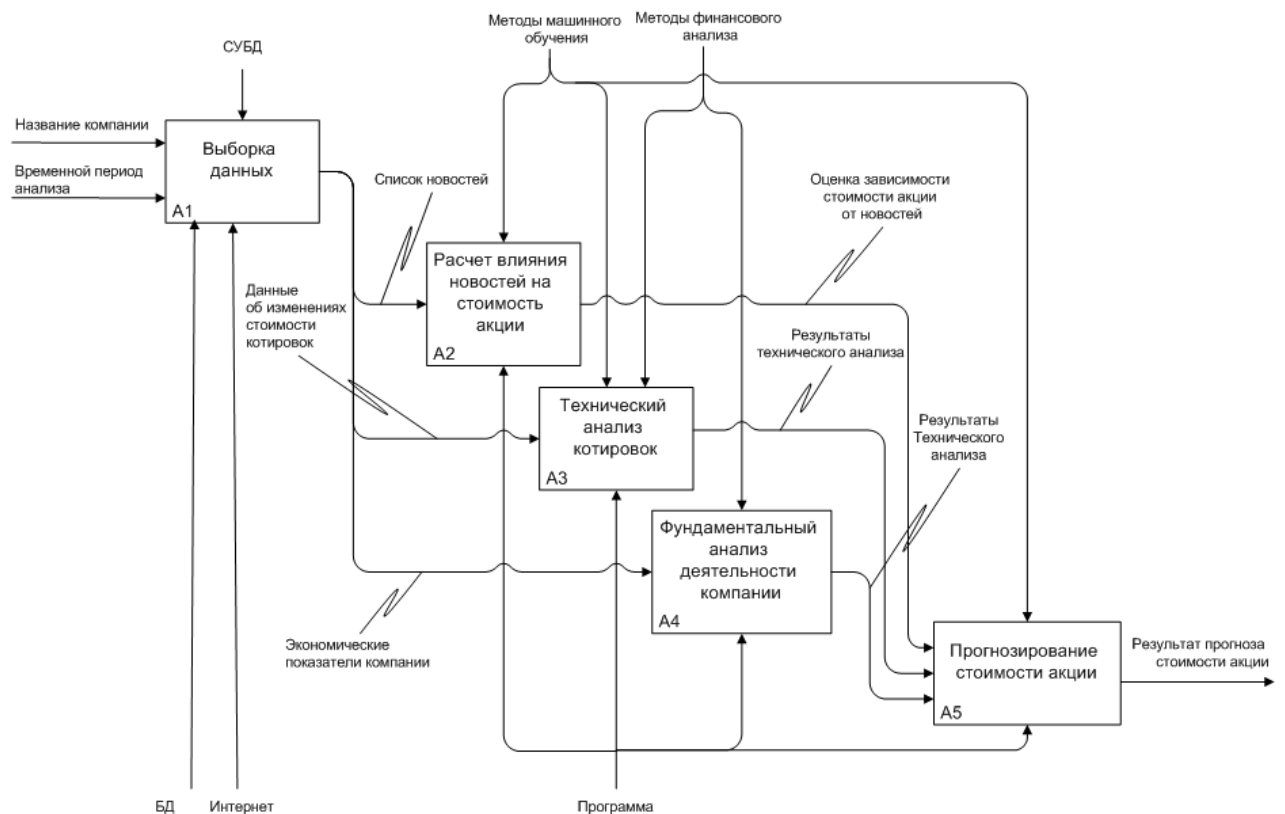


Рисунок 3.3 – Функциональная диаграмма системы (уровень 2)

Кроме того, данные проходят этап технического анализа. Анализ проводится на основе методов финансового анализа с применением методов машинного обучения. В качестве входных данных для анализа используются исторически данные об изменении стоимости котировок акции за определенный период времени.

В блоке фундаментального анализа деятельности компании производится оценка экономических показателей и расчет мультипликаторов для определения финансового состояния компании. В качестве входных данных используются экономические показатели компании, полученные на основе финансовой отчетности компании.

Блок прогнозирования стоимости акции агрегирует в себе результаты анализа, полученные из вышеуказанных блоков, и на основе этих данных формирует результирующую оценку будущей стоимости акции.

3.4 Разработка модуля технического анализа фондового рынка

Для отбора наиболее подходящей нейронной сети было проведено сравнительное тестирование различных типов сетей. В качестве тестового набора данных были взяты данные о стоимости акций компании Лукойл.

Полносвязная нейронная сеть

Наилучшие результаты были получены при использовании четырех полносвязных слоев из 100 нейронов с линейной функцией активации. Исследование проводилось для прогнозируемого периода от 10 до 60 часов. В зависимости от длительности прогнозируемого периода точность изменялась от 0.997 до 0.982 (рисунок 3.4 **Ошибка! Источник ссылки не найден.**).

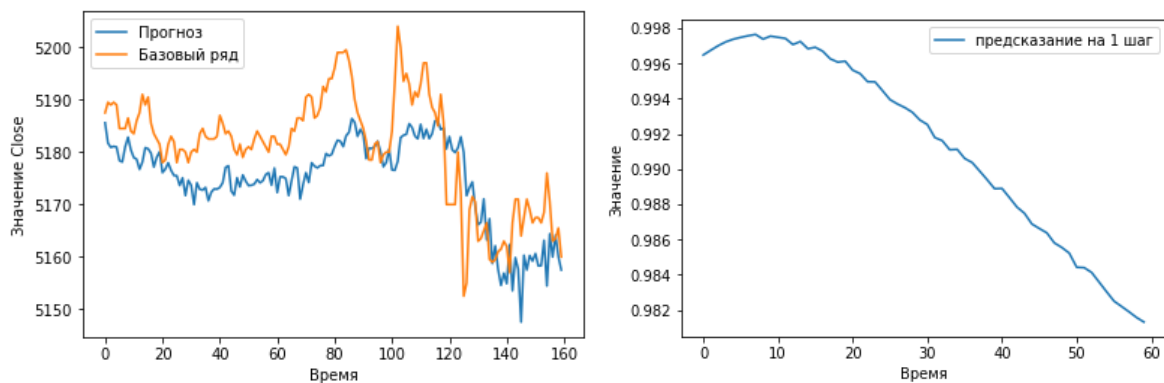


Рисунок 3.4 – Результат прогнозирования полносвязной нейронной сети и изменение точности прогнозирования

Сверточная нейронная сеть

Наилучшие результаты были получены при использовании трех сверточных слоев первый слой из 50 нейронов, остальные из 100 нейронов с линейной функцией активации исследование проводилось для прогнозируемого периода от 0 до 10 часов, т.к. точность работы данной сети на порядок отличается от точности полносвязной сети и не имеет смысла производить расчеты для большего периода. В зависимости от длительности прогнозируемого периода точность изменялась от 0.9997 до 0.997 (рисунок 3.5).

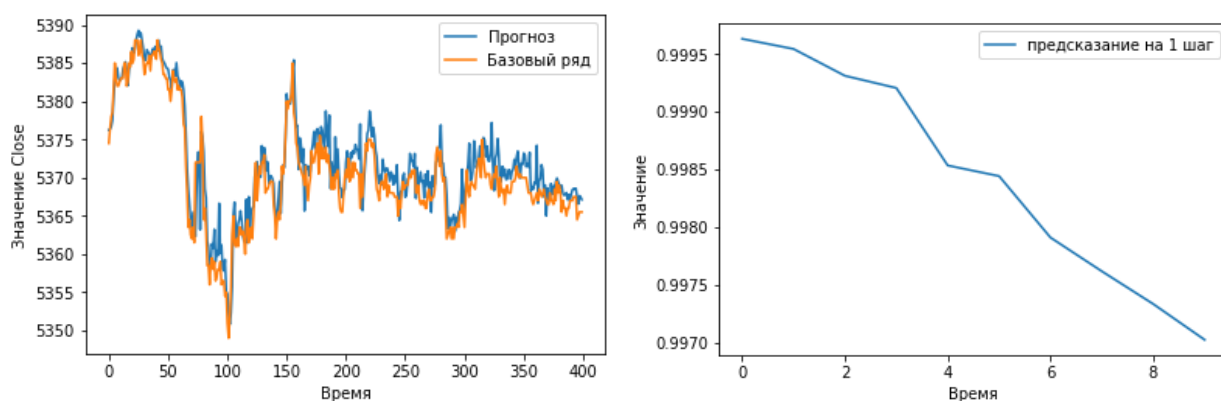


Рисунок 3.5 – Результат прогнозирования сверточной нейронной сети и изменение точности прогнозирования

Рекуррентная нейронная сеть

Наилучшие результаты были получены при использовании трех слоев из 100 нейронов, в качестве функции активации был использован гиперболический тангенс. Исследование проводилось для прогнозируемого периода от 0 до 10 часов. В зависимости от длительности прогнозируемого периода точность изменялась от 0.9996 до 0.998 (рисунок 3.6).

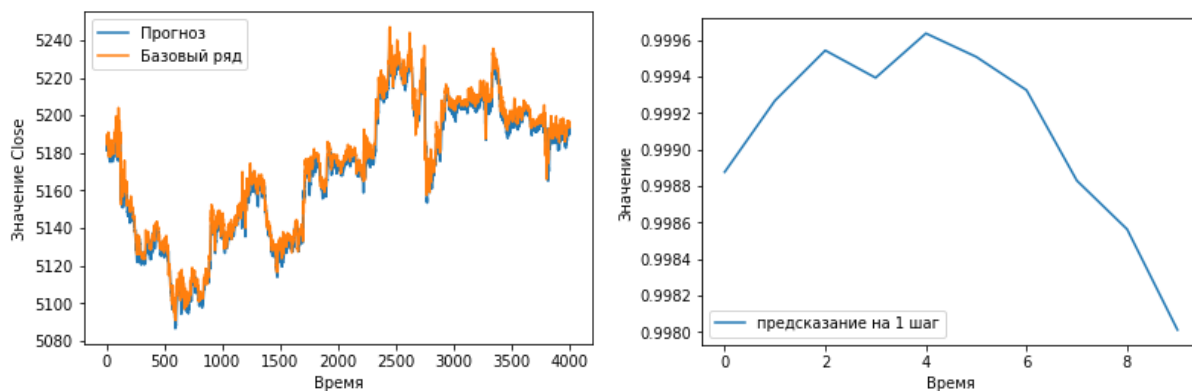


Рисунок 3.6 – Результат прогнозирования рекуррентной нейронной сети и изменение точности прогнозирования

На основе проведенных экспериментов можно сделать вывод, что для задачи прогнозирования временных рядов необходимо использовать рекуррентные или сверточные нейронные сети. Их особенность заключается в том, что на каждой итерации работы нейронной сети учитываются результаты анализа, полученные на предыдущих итерациях. При этом для прогнозирования временных рядов стоимости акций компании наилучшим образом подойдут рекуррентные нейронные сети типа LSTM (Long Short-Term

Memory). Потому что данные нейросети при работе позволяют учитывать более ста предыдущих значений временного ряда.

3.5 Разработка модуля фундаментального анализа фондового рынка

Фундаментальный анализ – метод оценки ценных бумаг, принадлежащих компании. Их можно рассмотреть, изучая бухгалтерские и финансовые отчеты организации. Таким образом получится определить выгоду при покупке бумаг. Целесообразно покупать те бумаги, реальная стоимость которых превышает текущую.

Отчетность – это зеркало финансового состояния компании. С ее помощью оценивают, насколько бизнес стабилен и как он будет развиваться дальше. Финансовые показатели отражают все результаты работы компании, которые влияют на котировки ее ценных бумаг. Если компания получила за квартал убыток, это можно увидеть в отчете. Так же просто увидеть, что увеличились долги, а от операционной деятельности нет дохода. По подобным сигналам решают, стоит вкладывать деньги в этот бизнес или нет [3].

Одним из методов фундаментального анализа является анализ финансового положения компании. Чаще всего при анализе финансового положения обращают внимание на финансовую устойчивость, платежеспособность и оборачиваемость активов предприятия.

Оценка финансовой устойчивости компании позволит определить её платежеспособность в долгосрочной перспективе, а именно способность устойчиво финансировать свою деятельность и развиваться. Для оценки финансовой устойчивости необходимо рассчитать следующие коэффициенты: коэффициент независимости, коэффициент текущей ликвидности, коэффициент средней (срочной) ликвидности, деловая активность (оборотность) предприятия, коэффициент оборачиваемости оборотных активов.

Для определения характеристик компании чаще всего используют мультипликаторы. Это помогает сравнить деятельность и инвестиционную привлекательность предприятий разных размеров.

По своей сути мультипликаторы – это производные финансовые показатели для оценки бизнеса. Их используют приверженцы системного подхода к выбору компаний для вложений. По этой стратегии инвестор покупает акции предприятий, которые недооценены. В этом случае больше шансов получить высокий доход на инвестированный капитал, а риск, что стоимость таких бумаг сильно снизится, соответственно, ниже.

Мультипликатор EV/EBITDA отражает соотношение стоимости предприятия (Enterprise Value, EV) и заработанной прибыли. Причем прибыль берут до вычета амортизации, износа, процентов и налога на прибыль EBITDA (Earnings before interest, taxes, depreciation and amortization) [4].

Значение EV/EBITDA показывает:

- насколько по сравнению с другими недооценена или переоценена акция;
- за какой срок приобретение компании окупится прибылью, если ее не тратить на амортизацию, проценты и налоги.

Price to earnings (P/E) – отношение капитализации к прибыли или рыночной цены предприятия к чистой прибыли на одну акцию. Этот мультипликатор позволяет сравнить компании из различных сфер деятельности. Он показывает, за сколько лет предприятие себя окупает.

Значение P/E от 0 до 5 говорит, что компания недооценена. Если оно больше – скорее всего, переоценена, а если показатель меньше 0 – компания принесла убыток. Не стоит сравнивать компании только по P/E. Они могут быть на разных стадиях развития. Молодые компании иногда показывают мультипликатор хуже, так как имеют большие капитальные расходы, которые уменьшают прибыль, а у некоторых при невысокой прибыли низкие капитальные расходы улучшают P/E.

Мультипликатор P/S – это отношение капитализации компании к выручке, или рыночной цены акции к выручке на одну акцию. С его помощью сравнивают компании из одной отрасли, у которых приблизительно одинаковая маржинальность.

Нормальное значение мультипликатора меньше двух. Если он меньше одного, то компании недооценена. Удобно, что рассчитать P/S можно для каждой компании, ведь выручка всегда больше нуля, и значение его будет положительным.

Кроме того, практика показывает, что определённая совокупность мультипликаторов и некоторых параметров экономических отчетов позволяет сделать вывод о поведении акции на фондовом рынке в целом. Например, к акциям с высоким потенциалом можно относить акции, параметры которых соответствуют следующим условиям:

- рыночная капитализация (Market Cap) от 10 млрд \$;
- мультипликатор P/E от 15%;
- рост стоимости акции за год от 10%.

К акциям, которые имеют стабильный рост, можно отнести акции, которые соответствуют следующим условиям:

- рыночная капитализация (Market Cap) от 10 млрд \$;
- отношение общего долга к собственному капиталу (Debt/Equity) до 300%;
- мультипликатор P/E от 15% до 35%;
- отношение чистой прибыли к выручке (Net Profit Margin) от 15% до 35%;
- рост стоимости акции за 5 лет от 20%.

К быстрорастущим акциям, можно отнести акции, которые соответствуют следующим условиям:

- отношение общего долга к собственному капиталу (Debt/Equity) до 300%;

- мультипликатор P/E от 15%;
- отношение чистой прибыли к выручке (Net Profit Margin) от 5%;
- рост стоимости акции за год от 30%.

Благодаря рассчитанным условиям, появляется возможность проводить быструю группировку акций, относя их к определенной категории в зависимости от их экономических параметров.

3.6 Реализация модуля анализа влияния новостей на фондовый рынок

Для осуществления анализа новостей и оценки их влияния на стоимость акции необходимо решить задачи, связанные с анализом естественного языка. В частности, необходимо в тексте определить перечень компаний, упоминаемых в новости, а также оценить тональность текста новости относительно каждой из обнаруженных компаний (рисунок 3.7 **Ошибка! Источник ссылки не найден.**).

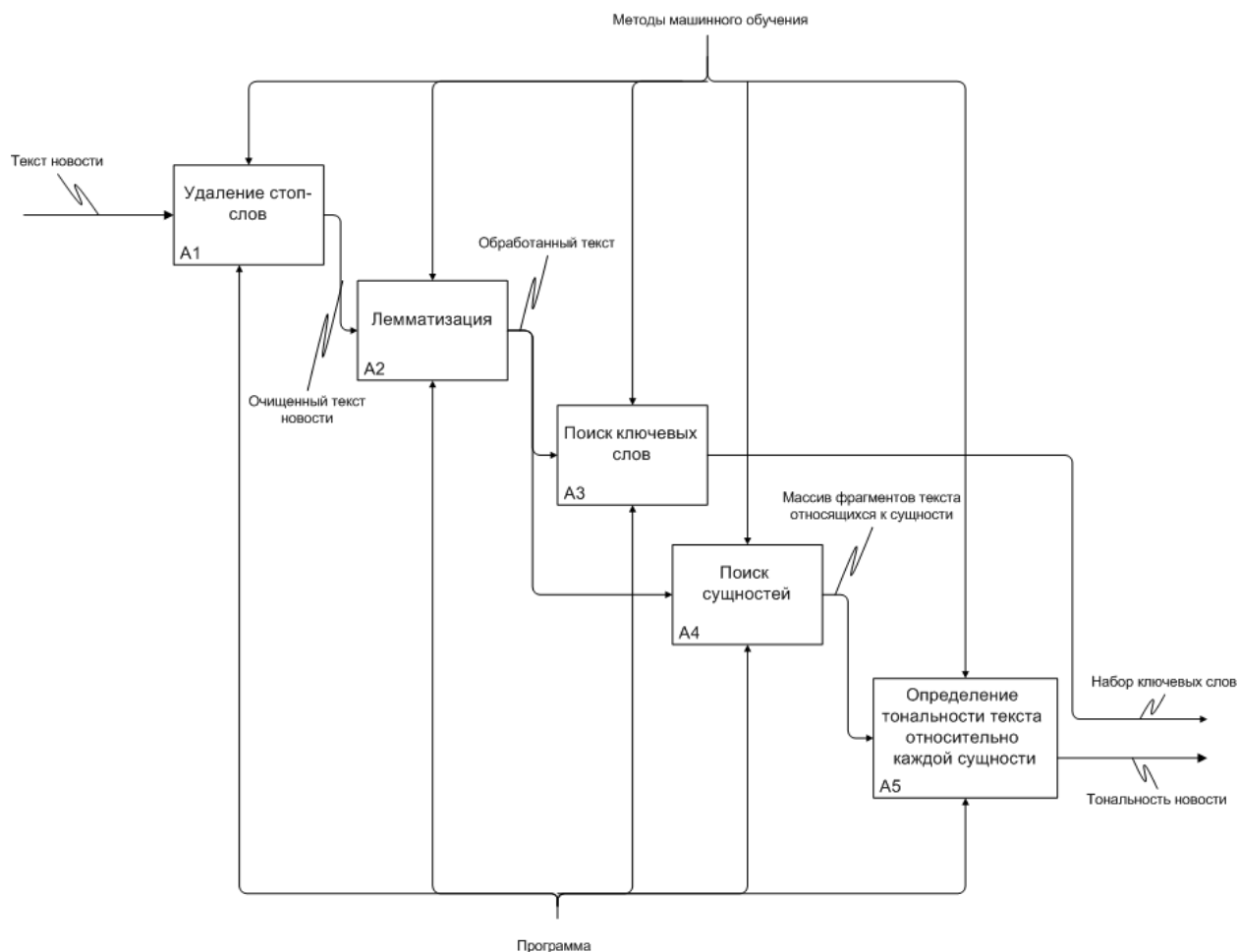


Рисунок 3.7 – Функциональная схема получения тональности текста новости

Однако при определении влияния новости на изменения стоимости акции конкретной компании недостаточно определить тональность текста новости. Зачастую одна и та же новость может содержать информацию, оказывающую различное влияние на перечень компаний. Например, оценивая тональность новости «... компания Apple фиксирует увеличение прибыли на фоне пожара, произошедшего на заводе Samsung ...» невозможно дать однозначный ответ является ли данный текст положительным или отрицательным.

Для решения данной проблемы необходимо при анализе данных реализовать выявление в тексте ключевых слов, а также различных сущностей. Далее для каждой обнаруженной ключевой сущности необходимо формировать собственный вектор-предложения, заменяя само ключевое слово на специальный символ.

Таким образом при рассмотрении примера выше сущности «Apple» и «Samsung» необходимо последовательно выделять как ключевую сущность (таблица 9).

Таблица 9 – Пример набора данных для обучения лингвистической модели

Ключевая сущность	Текст новости	Тональность текста
Apple	компания X фиксирует увеличение прибыли на фоне пожара, произошедшего на заводе Samsung	Позитивно
Samsung	компания Apple фиксирует увеличение прибыли на фоне пожара, произошедшего на заводе X	Негативно

Благодаря анализу новостей можно не только определить направление тренда стоимости акции компании, но и приблизительную степень изменения стоимости. Это достигается за счет использования полученной при анализе тональности текста новости и полученного набора ключевых слов. Производя анализ новостей для уже имеющихся данных, можно обнаружить закономерности между набором ключевых слов текста новости и степени изменения стоимости акции. Основываясь на данной закономерности, можно определить степень изменения стоимости акции в зависимости от полученного набора ключевых слов. Расчёт влияния искомой новости на стоимость акций компании можно представить в виде следующей формулы:

$$impact = \sum_{n=1}^N \frac{\sum_{i=1}^K (-1^{tn} * impact_n * w_{kn})}{K}, \quad (8)$$

где N – количество новостей, влияющих на компанию, K – количество пересекающихся ключевых слов, $impact_n$ – влияние новости n на стоимость акции, w_{kn} – влияние ключевого слова k на стоимость компании в рамках новости n , tn – тональность новости по отношению к компании (1 – негативная, 2 – позитивная).

3.7 Агрегирование результатов анализа

Агрегирование результатов анализа является необходимым этапом прогнозирования стоимости акции компании. В виду того, что анализ осуществляется с применением нескольких независимых видов оценки, результаты, полученные при использовании различных методов, могут отличаться друг от друга. В связи с этим необходимо обеспечить согласованность результатов и разработать формулу для получения прогноза на основе полного многофакторного анализа. Для получения совокупного результата можно применять формулу среднего арифметического [7]:

$$\bar{x} = \frac{\sum_{i=1}^n k_i x_i}{n} = \frac{1}{3} \sum_{i=1}^3 k_i x_i, \quad (9)$$

где n – количество методов анализа, k_i – коэффициент i -го метода анализа, x_i – результат анализа i -го метода анализа.

Важным этапом при расчете совокупного результата анализа является проверка полученных данных на согласованность.

Для определения согласованности результата необходимо рассчитать отклонение результатов анализа определенного метода от среднего совокупного результата. Расчет может быть произведен по формуле среднего квадратичного отклонения [8]. Кроме того, необходимо произвести расчет коэффициента вариации, для выражения согласованности в процентах:

$$V = \frac{\sigma}{\bar{x}} * 100\%, \quad (10)$$

где σ - результат среднего квадратичного отклонения.

Результаты можно считать согласованными в случае, если коэффициент больше 75%, в данном случае результат агрегирования является достоверным. В случае, если коэффициент меньше 75%, то это говорит о невозможности сделать однозначный прогноз. Лучшим решением в данном случае является предоставление аналитику результатов анализа по каждому из методов, для самостоятельного принятия решения.

3.8 Проектирование диаграммы потоков данных

При разработки информационной системы необходимо учитывать какие данные и каким образом будут перемещаться между модулями и блоками системы. Диаграмма потоков данных (DFD – Data Flow Diagram) подразумевает визуализацию хранилищ данных, процессов, происходящих в информационной системе, соединенных потоками данных, а также внешние сущности, взаимодействующие с системой.

Ниже представлен фрагмент диаграммы потоков данных разрабатываемой ИАС, на ней продемонстрированы процессы, связанные с анализом стоимости акций и прогнозирования их стоимости в будущих периодах (рисунок 3.8).

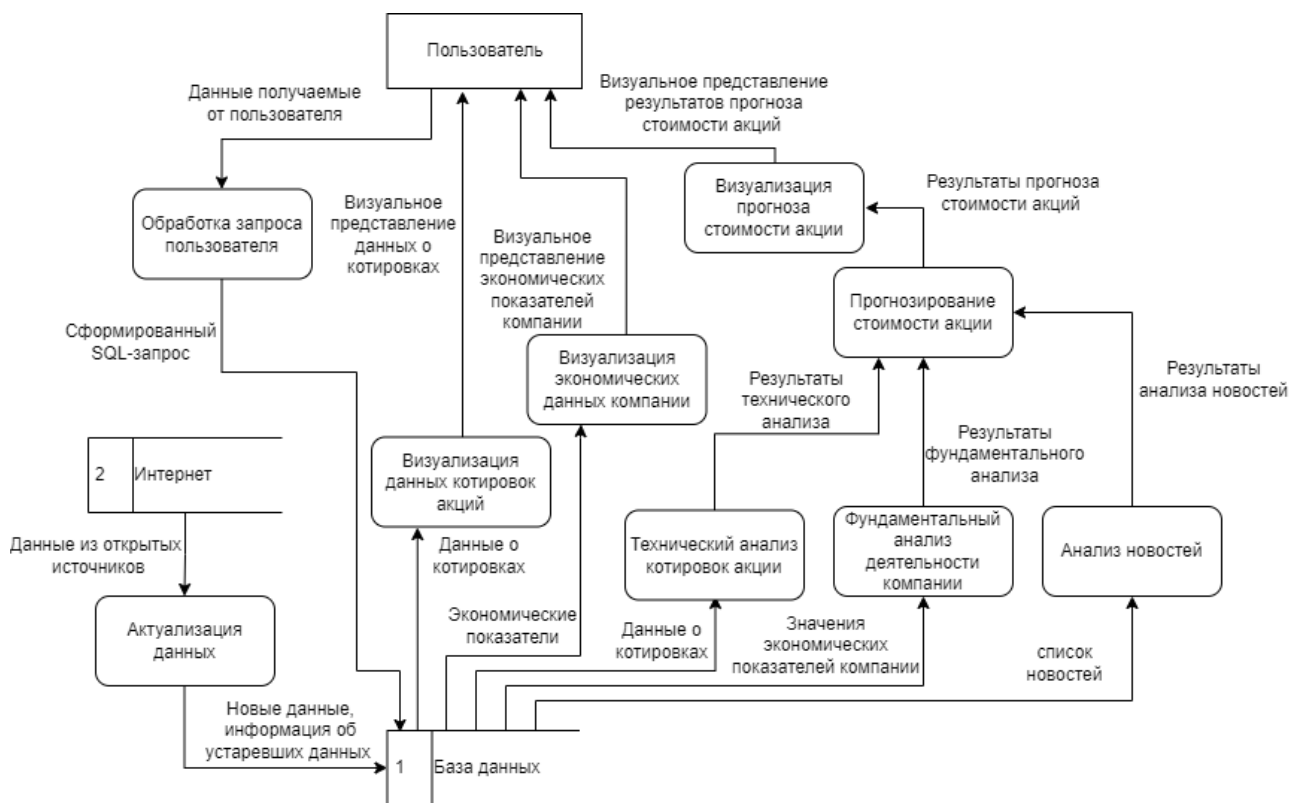


Рисунок 3.8 – Диаграмма потоков данных (фрагмент)

В качестве хранилищ данных в системе используются база данных и сеть Интернет. При этом основным хранилищем является база данных, получение новой информации происходит из сети Интернет в процессе актуализации данных. Важно отметить, что актуализация данных подразумевает под собой

не только добавление новой информации, но и редактирование уже существующих данных.

Внешней сущностью системы является пользователь. Он является инициатором отправки данных на сервер, и он же является конечной точкой для отправляемых данных.

Данные от пользователя проходя процесс обработки, при котором происходит их преобразование в набор необходимых SQL-запросов. Полученные запросы обрабатываются базой данных. В результате обработки на выходе формируется набор данных, который в зависимости от цели перенаправляется в модули визуализации информации или в модули анализа информации. К визуализации информации можно отнести процессы визуализации данных о котировках акций и визуализации экономических данных компании.

В случае выполнения анализа данных, сформированные данные из базы данных используются в процессах технического и фундаментального анализов, а также в процессе анализа новостей. Полученные на выходе данные используются для составления прогноза стоимости акции. Данные полученные в процессе прогнозирования отправляются в модуль визуализации для представления пользователю в удобном для восприятия виде.

3.9 Проектирование структуры базы данных

На основе проведенного анализа предметной области была спроектирована база данных, в которой будет храниться вся необходимая информация для разрабатываемой системы. Данная структура представлена в виде отдельных взаимосвязанных таблиц (рисунок 3.9).

Таблица «companies» содержит в себе информацию о компании, эмитенте акций. Данная информация включает в себя общую информацию о компании, такую как название компании, описание её деятельности, контакты, численность работников и т.д. Также в данной таблице содержатся

экономические характеристики компании, например доходы и расходы компании, её стоимость, выручку и прочие экономические параметры. Таблица «companies» связана с таблицами «company_filings» и «company_events» связями один-ко-многим, так как внутри одной компании публикуется множество документов и происходит множество событий.

Таблица «company_filings» содержит данные об экономических отчетах компании. В таблице содержится информация о различных бухгалтерских отчетах, например бухгалтерские балансы, отчеты об изменении капитала, движении денежных средств и т.д. Для этого таблица содержит такие поля как дата публикации, тип отчета, ссылка на сам документ и т.д.

В таблице «company_events» хранится информация о событиях, происходящих в компании, например выплаты дивидендов и даты их проведения. Учитываются такие свойства события как его типа, дата наступления события, название и ссылку для перехода к дополнительной информации.

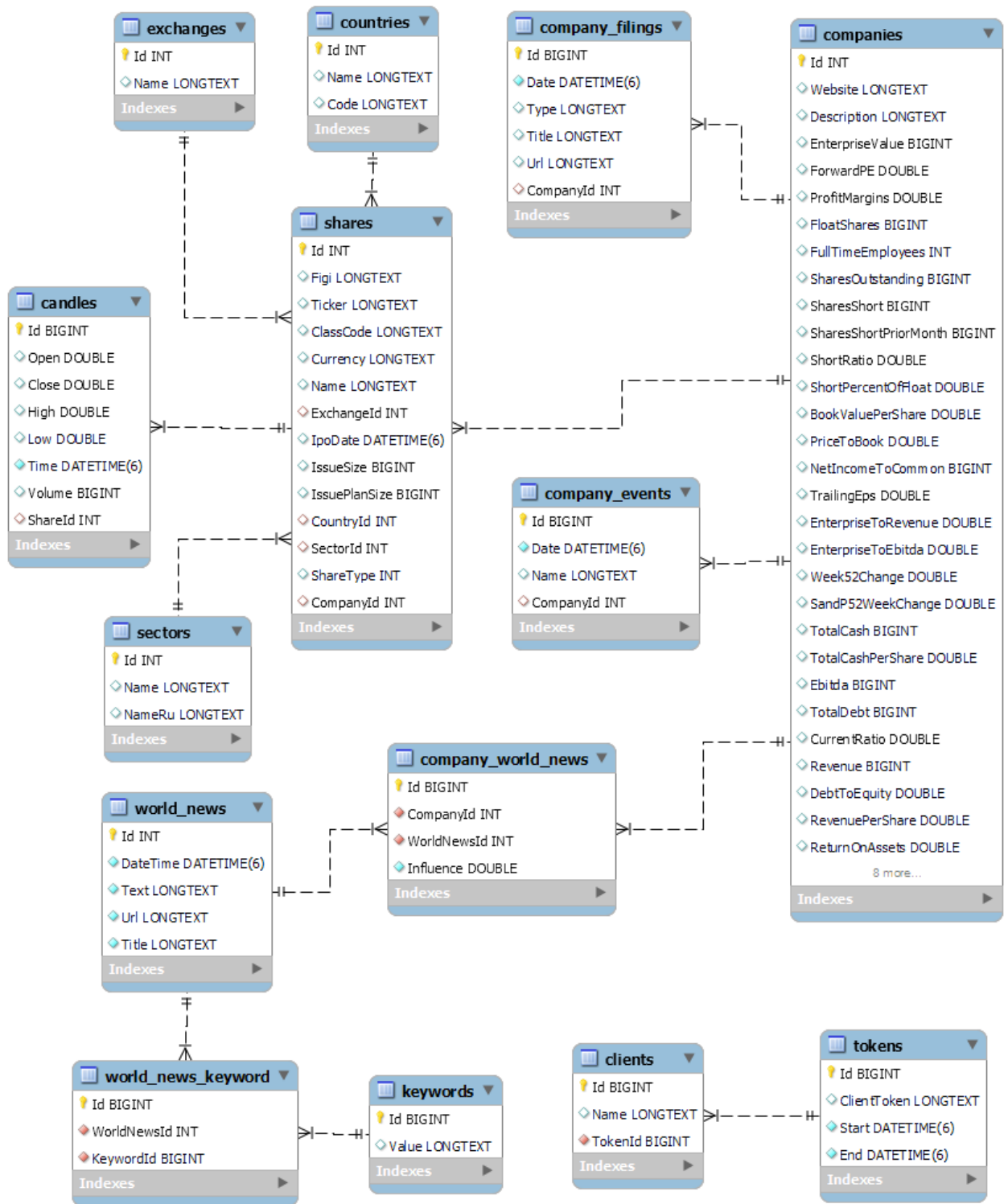


Рисунок 3.9 – ER-диаграмма (фрагмент)

Для хранения данных об акциях была использована таблица «shares». Данная таблица является промежуточной между компаниями и биржами. Она необходима, так как акции одной и той же компании могут размещаться на различных биржах. Таблица содержит информацию о дате IPO, количестве выпущенных акций и т.д. Кроме того, таблица содержит внешние ключи на

таблицы «countries», «exchanges» и «sectors», которые хранят информацию о стране, фондовая биржа и отрасли, к которым относится конкретная акция. Данные таблицы необходимы для фильтрации и группировки акций компаний.

Таблица «candles» хранит в себе исторические данные о котировках. Данные включают в себя количество проданных акций, стоимость акции на момент открытия, закрытия, а также наибольшая и наименьшая стоимости акции, за конкретную дату. Также в данной таблице хранится ссылка на таблицу «shares» реализуя связь один-ко-многим.

Таблица «company_world_news» является связующей таблицей между компаниями и мировыми новостями, которые влияют на показатели изменения котировок акций. Она содержит в себя ссылки на таблицы «companies» и «world_news», также данная таблица хранит в себе оценку влияния мировых новостей на изменения котировок акций.

Таблица «tokens» хранит токены для доступа клиентов к API платформы, данная таблица связана с таблицей «clients», которая хранит данные о клиенте, купившем месячную подписку.

3.10 Разработка API серверной части аналитической системы

Так как разрабатываемая информационная система построена по принципам клиент-серверной архитектуры, то необходимо реализовать возможность обмена информацией между клиентской и серверной частями.

Одним из способов такого обмена является разработка программного интерфейса приложения или API (Application Programming Interface). Данный подход подразумевает набор URL-адресов (эндпоинтов), которые при обращении возвращают необходимые данные в универсальном формате или запускают определенный код обработки данных на сервере.

Главным плюсом использования API является его универсальность. Благодаря тому, что передача данных происходит посредством протокола HTTP, обращение к API можно производить независимо от конфигурации клиентского приложения. Это может быть как web-сайт или мобильное

приложение, так и внешний сервис, которые взаимодействует с разрабатываемой системой.

В разрабатываемой ИАС программный интерфейс можно разделить на несколько блоков в зависимости от обрабатываемой информации (Рисунок 3.10).

Блок «Auth» представляет собой набор эндпоинтов для осуществления регистрации пользователя, авторизации и процесса разлогирования. Также в данном блоке содержатся эндпоинты, необходимые для работы личного кабинета пользователя, например получения списка избранных акций.

Эндпоинты из блока «World news» используются для получения списка мировых новостей, а также информации о конкретной новости.

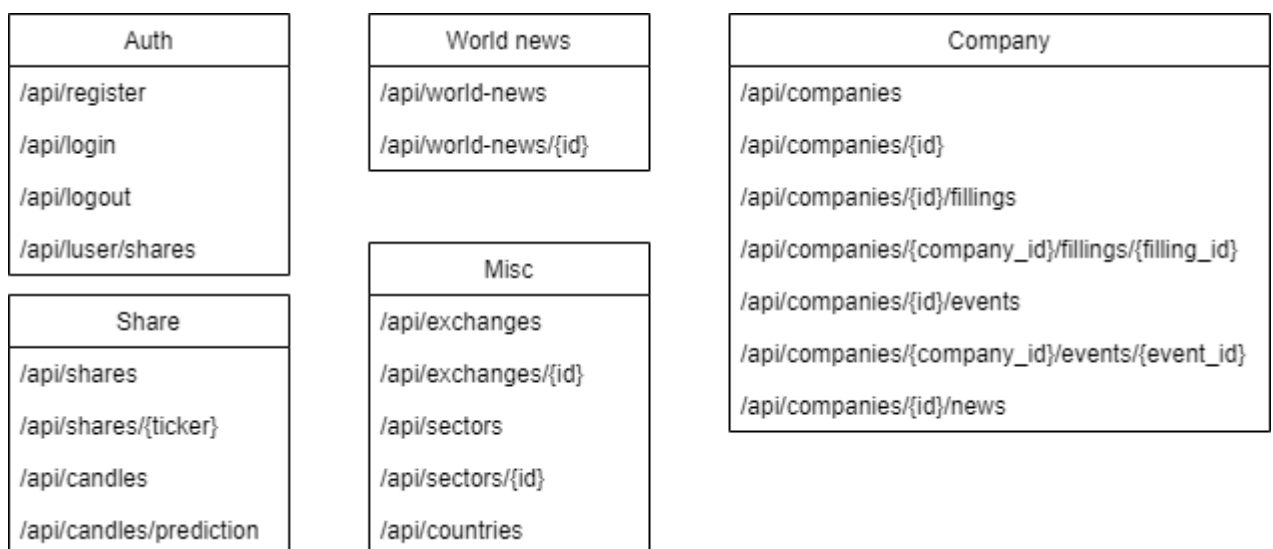


Рисунок 3.10 – Структура программного интерфейса приложения

Блок «Share» содержит в себе набор эндпоинтов для работы с акциями и их котировками. Например, получения списка акций, получение детальной информации об акции, получения котировок акций или получения результатов анализа и прогноза стоимости котировок акций.

В блоке «Company» объединены эндпоинты для получения данных о компании. В частности, это эндпоинты для получения списка компаний, детальной информации о компании, включающую экономические показатели,

получение финансовой отчетности компании, получение списка новостей и событий, происходящих в компании.

Различные сервисные эндпоинты сгруппированы в блоке «Misc». В данном блоке перечислены эндпоинты для получения списка фондовых бирж и перечня акций, представленных на конкретной бирже; получения списка секторов экономики и списка компаний, входящих в их состав; а также списка стран, которые используются для фильтрации и группировки компаний и акций.

Реализуемые эндпоинты доступны по протоколу HTTP и возвращают ответ от сервера в формате JSON. Так как данный формат данных является одним из часто используемых форматов, используемых при построении API. Это обеспечит возможность интеграции разрабатываемой системы с различными сервисами.

Также для обеспечения возможности интеграции в разрабатываемом API используются HTTP-коды, как индикаторы успешности обработки запроса. Ответы сервера могут быть возвращены с одним из перечисленных в списке кодом:

200 OK – данный код будет возвращен в случае успешной обработки запроса;

400 Bad Request – данный код будет возвращен в случае неверного тела запроса. Например, если в параметрах запроса содержится ошибка или запрос содержит недостаточного количество данных;

401 Unauthorized – данный код будет возвращен в случае, если произойдет попытка доступа к данным, которые доступны только авторизованным пользователям;

404 Not Found – данный код будет возвращен при попытке обратиться к несуществующему ресурсу;

405 Method Not Allowed – данный код будет возвращен, когда используемый метод HTTP-запроса не соответствует требуемому. Например,

если при попытке обратиться к эндпоинту, обрабатывающему GET-запросы, использовать при отправке запроса метод POST.

422 Unprocessable Entity – данный код будет возвращен, в случае ввода невалидных данных. Например, если пользователь при авторизации указывает неверный логин или пароль;

500 Server Error– данный код будет возвращен, в случае, когда ошибка происходит на стороне сервера. Например, если сервер вышел из строя или по другой причине не может нормально функционировать.

3.11 Разработка пользовательского интерфейса для web-сайта и мобильного приложения

При разработке информационно-аналитической системы необходимо реализовать интерфейс взаимодействия пользователя с системой. При этом пользовательский интерфейс должен удовлетворять определенным критериям:

- Естественность интерфейса – свойство интерфейса, которое означает, что сообщения и результаты, выдаваемые приложением, не должны требовать дополнительных пояснений.
- Согласованность интерфейса позволяет пользователям переносить имеющиеся знания на новые задания, осваивать новые аспекты быстрее, и благодаря этому фокусировать внимание на решаемой задаче, а не тратить время на уяснение различий в использовании тех или иных элементов управления, команд и т. д. Обеспечивая преемственность полученных ранее знаний и навыков, согласованность делает интерфейс узнаваемым и предсказуемым.
- Принцип «обратной связи» – принцип, который означает, что каждое действие пользователя должно получать визуальное, а иногда и звуковое подтверждение того, что программное обеспечение восприняло введенную команду; при этом вид реакции, по возможности, должен учитывать природу выполненного действия.

- Простота интерфейса – представление на экране информации, минимально необходимой для выполнения пользователем очередного шага задания.

Базовым элементом, вокруг которого строится анализ и прогнозирование стоимости акции, является график изменения её стоимости во времени. Однако кроме на стоимость могут также влиять экономические показатели компании, мировые новости, а также изменения стоимости других акций.

С целью расширить возможности анализа было принято решение использовать построение диаграммы с возможностью добавления на нее графиков других компаний. При этом для удобства реализована функция, которая при клике на определенную дату графика выводит список новостей за этот день (рисунок 3.11), что позволяет пользователю одним нажатием перейти от технического анализа к анализу новостей.

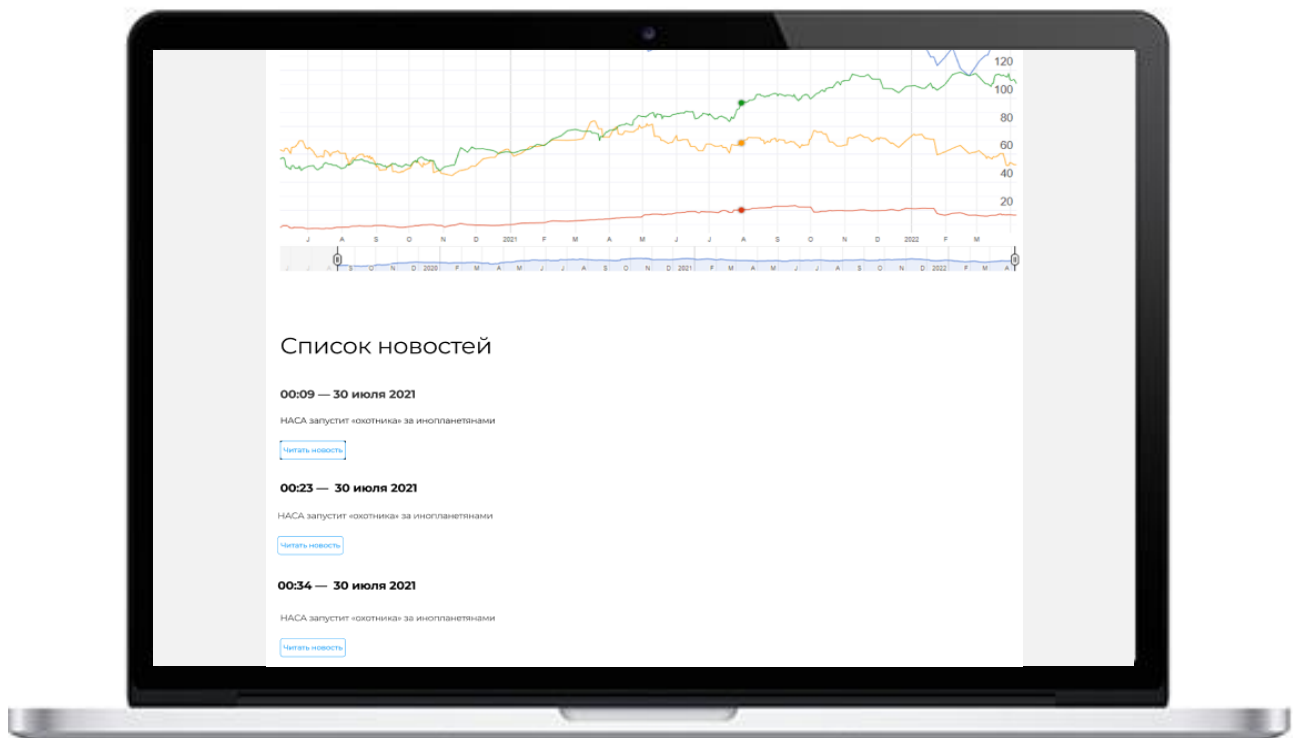


Рисунок 3.11 – Интерфейс программного продукта (фрагмент)

При переходе на страницу компании, пользователь может получить информацию о её деятельности, экономическом состоянии, а также оценить тренд изменения стоимости акции на основе графика.

Кроме того, пользователь должен иметь возможность управлять анализом и прогнозом стоимости акции. Для этого на странице был реализован блок «Прогнозирование стоимости». Данный блок включает в себя реализацию используемых типов анализа: технический анализ, фундаментальный анализ и анализ на основе новостей (рисунок 3.12).

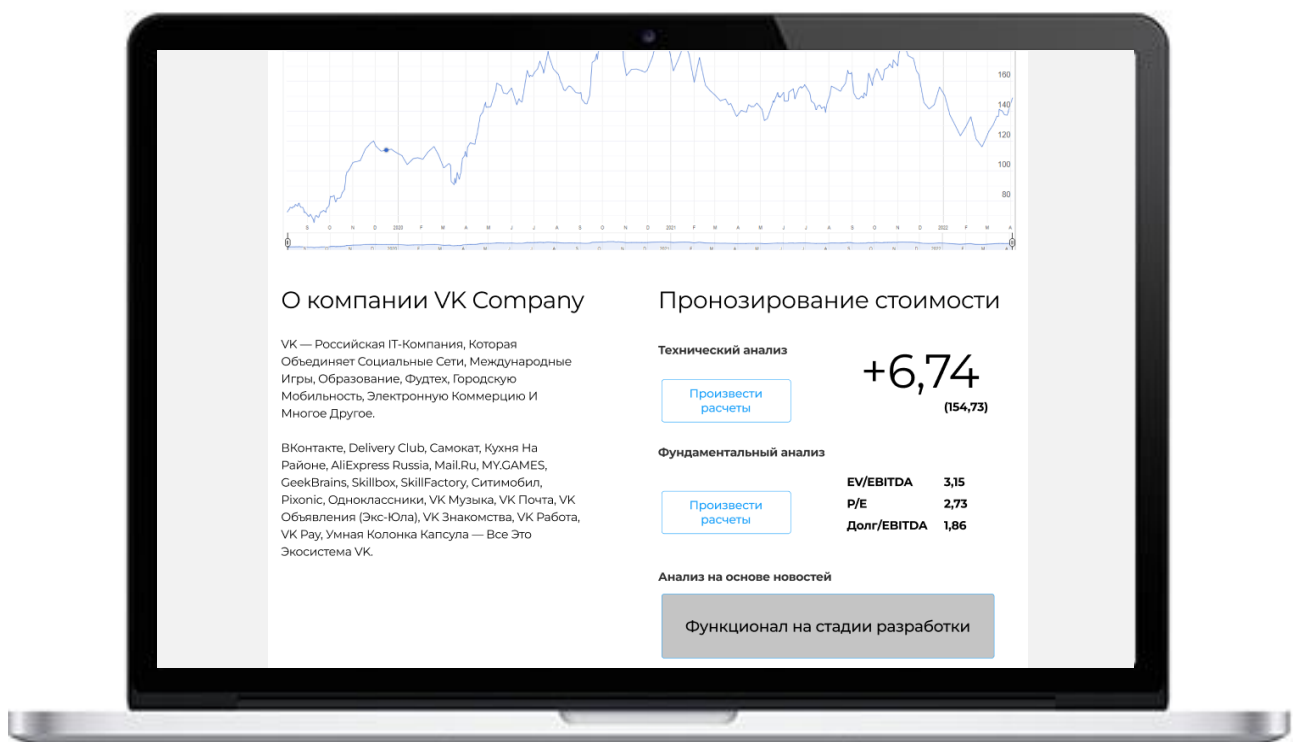


Рисунок 3.12 – Интерфейс программного продукта (фрагмент)

Для получения результатов анализа пользователю необходимо нажать кнопку «Произвести расчеты» в требуемом блоке анализа. После чего на сервер отправляется запрос для проведения анализа стоимости указанной акции. После завершения анализа результаты отправляются на клиентскую часть программного продукта, где выводятся в удобном для восприятия виде.

4 ЭКСПЕРИМЕНТАЛЬНАЯ ЧАСТЬ

4.1 Тестирование информационно-аналитической системы

Задача тестирования ПО нуждается в больших трудозатратах и тщательном применении выбранных подходов. Для успешного внедрения программы в рамках рабочего процесса организации-эксплуатанта необходимо обеспечение полной работоспособности системы в рамках предусмотренных задач, включающее отсутствие критических уязвимостей в процессах, обработку потенциальных ошибок, конфигурацию системы в соответствии с требованиями среды её эксплуатации.

В тестировании системы применимы следующие виды контроля:

1. Компонентное/модульное тестирование (Component/Unit Testing).
2. Интеграционное тестирование (Integration Testing).
3. Системное тестирование (System Testing).
4. Приемочное тестирование (Acceptance Testing).

Каждый уровень тестирования направлен на определенную часть программы и выполняет свои цели.

Компонентное/модульное тестирование

Этот вид тестирования выполняется на самой ранней стадии разработки программы – во время написания кода. Обычно его выполняет сам программист, который пишет код. Следовательно, ошибки, в большинстве случаев, исправляются сразу же и не попадают к специалистам по тестированию.

Как видно из названия, модульное тестирование направлено на тестирование отдельных модулей и компонентов программы, которые изолированы от других модулей и компонентов. Поэтому его стоит совмещать с другими видами тестирования, сам по себе он малоэффективен.

Для этого уровня тестирования характерно несколько целей:

1. Проверка компонента на соответствие требованиям,
2. Обнаружение ошибок в компоненте,

3. Предотвращение пропуска ошибок на более высокие уровни тестирования.

С помощью компонентного тестирования мы снижаем риски и укрепляем свою уверенность в качестве продукта. К сожалению, этот уровень тестирования требует большой ответственности и ресурсов со стороны разработки, и в большинстве случаев на него нет времени. Поэтому, такое тестирование редко используется в компаниях.

Интеграционное тестирование

Интеграционное тестирование необходимо для того, чтобы тестировать взаимосвязь между чем-либо.

В общем случае различают два вида интеграционного тестирования:

Компонентное интеграционное тестирование. Как видно из названия, оно необходимо для того, чтобы протестировать работу модулей в связке друг с другом.

Системное интеграционное тестирование. Если с предыдущим уровнем тестирования все понятно, то с системным интеграционным тестированием все несколько сложнее. Этот уровень необходим для тестирования систем друг с другом.

Давайте рассмотрим на примере. Предположим, мы разрабатываем игру для смартфона. Сама игра является системой, которую необходимо протестировать. Кроме этого, есть еще сервисы, которые взаимодействуют с игрой и такое взаимодействие тоже должно быть проверено. Таких сервисов достаточно много, хотя на первый взгляд их трудно заметить. Каждый из них является системой, которая интегрируется в нашу игру. Например, игра может поддерживать социальную сеть Facebook, чтобы можно было играть с друзьями. В этом случае, необходимо протестировать корректно ли работает вход в социальную игру через нашу игру, можем ли мы видеть список друзей и т.д.

Для этого уровня тестирования также характерно несколько целей:

1. Проверка интерфейсов на соответствие требованиям.

2. Обнаружение ошибок в интерфейсах.

3. Предотвращение пропуска ошибок на более высокие уровни тестирования.

С помощью интеграционного тестирования мы снижаем риски и укрепляем свою уверенность в качестве продукта.

Системное тестирование

Системное тестирование – это тестирование еще более высокого уровня. Напомню, что на компонентном тестировании мы тестируем отдельные модули, а на интеграционном – связь между компонентами. При системном тестировании наша задача уже состоит в том, чтобы убедиться в корректности работы в целом всей системы. Программа в этом случае должна быть максимально приближена к конечному результату. А наше внимание должно быть сосредоточено на общем поведении системы с точки зрения конечных пользователей.

Для этого уровня тестирования также характерно несколько целей:

1. Проверка системы на соответствие требованиям.

2. Обнаружение ошибок в системе.

3. Предотвращение пропуска ошибок на более высокие уровни тестирования.

С помощью системного тестирования мы снижаем риски и укрепляем свою уверенность в качестве продукта.

Приемочное тестирование

Приемочное тестирование – наиболее высокий уровень тестирования. Оно, также как и системное тестирование, необходимо для проверки работы программы в целом.

Тут также смещаются цели тестирования. Ошибок на этом этапе уже не должно быть. Скорее наоборот, программа должна быть максимально рабочей и пригодной для использования. Если на данном этапе обнаруживаются критичные дефекты, то есть большая вероятность того, программа была плохо протестирована на предыдущих уровнях.

Этот уровень тестирования используется для подтверждения готовности продукта и проводится преимущественно в самом конце цикла разработки программы.

4.2 Оценка скорости работы системы

Для оценки программного обеспечения была произведена оценка использования ресурсов и быстродействия персонального компьютера разработанной системы.

Для проведения тестирования были выбраны замеры скорости загрузки модуля работы с БД, время загрузки модуля формирования web-страниц, скорость загрузки модуля серверов для разных платформ, а также полное время работы алгоритма анализа данных. Кроме выше озвученных тестов, был проведен замер потребления оперативной памяти и загрузки процессора в заданных конфигурациях.

Таблица 10 - Производительность системы

№	Конфигурация	Загрузка модуля работы с БД, мс	Загрузка модуля формирования web-страницы, мс	Скорость работы алгоритма анализа
1	CPU 3.6 ГГц на 12 потоков, 16 ГБ RAM, Windows 10	4 578	21 367	3 мин 19 с
2	CPU 2.5 ГГц на 4 потока, 4 ГБ ОЗУ, Windows 10	10 256	34 546	5 мин 15 с
3	CPU 2.7 ГГц на 8 потоков, 8 ГБ ОЗУ, Windows 10	7 436	27 843	4 мин 58 с

В результате поведенного тестирования, можно сделать вывод о том, что наибольшее время в работе ИАС занимает модуль формирования web-

страниц, предназначенный для визуализации исторических данных и данных проведенного анализа. В соответствии с конфигурацией (таблица 10) был проведен замер потребления ОЗУ и нагрузки на процессор (таблица 11).

Таблица 11 - Нагрузка системы

№	Конфигурация	Потребление ОЗУ	Нагрузка на процессор
1	CPU 3.6 ГГц на 12 потоков, 16 ГБ RAM, Windows 10	693МБ	25%
2	CPU 2.5 ГГц на 4 потока, 4 ГБ ОЗУ, Windows 10	747МБ	79%
3	CPU 2.7 ГГц на 8 потоков, 8 ГБ ОЗУ, Windows 10	645МБ	56%

В соответствии с тестированием нагрузки, были сформулированы минимальные системные требования к серверной части программного обеспечения: процессор не менее 4 ядер/8 потоков и 2.70 ГГц, не менее 8 ГБ ОЗУ.

5 ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Для обеспечения информационной безопасности разрабатываемой системы необходимо достичь следующих целей безопасности:

- Разрабатываемая система должна обладать механизмами регистрации любых событий, относящихся к возможным нарушениям безопасности.
- Необходимо обеспечить управление параметрами системой антивирусной защиты.
- Управление системной антивирусной защиты должно быть доступно только уполномоченным группам лиц.
- Доступ к системе должен быть ограничен в соответствии с ролями пользователей в системе.
- В системе должны быть реализованы регулярные проверки с целью обнаружения файлов, зараженных компьютерными вирусами.
- В антивирусном программном обеспечении должна быть реализована обработка зараженных файлов.
- База данных антивирусного ПО должна регулярно обновляться и содержать актуальную информацию о списке существующих компьютерных вирусов.

Кроме того, организация информационной безопасности разрабатываемой системы подразумевает наличие у объекта следующих свойств:

- разрабатываемая ИАС должна иметь доступ ко всем объектам, которые необходимы для функционирования системы;
- должны быть обеспечены установка и управление разрабатываемой системой в соответствии с правилами эксплуатации;
- должна быть обеспечена физическая защита компонентов системы, на которых установлена ИАС, а также хранятся данные, необходимые для функционирования системы;

- взаимодействия между элементами системы должны быть синхронизированы по времени;
- для взаимодействий между элементами системы должны использоваться доверенный каналы связи, обеспечивающие конфиденциальность передаваемых данных;
- лица, обеспечивающие функционирование разрабатываемой ИАС, обязаны обеспечивать функционирование системы в соответствии с установленной документацией.

Для обеспечения вышеперечисленных свойств необходимо рассмотреть возможные уязвимости и угрозы, которым необходимо противостоять в рамках реализации информационной безопасности системы.

5.1 Анализ уязвимостей и угроз для разрабатываемой ИАС

Для разрабатываемой информационной системы характерны следующие группы угроз:

- угрозы, которым должна противостоять разрабатываемая система;
- угрозы, которым должна противостоять среда, в рамках которой функционирует ИАС.

При рассмотрении первой группы угроз, были выявлены следующие опасности для разрабатываемой системы.

Угроза получения несанкционированного доступа к данным. Источником данной угрозы является внешний нарушитель. Реализация данной угрозы возможна путем получения нарушителем доступа к файлам cookie. Уязвимостью для возможности реализации данной угрозы являются недостатки реализации компонента аутентификации пользователей. При этом нарушаются такие свойства безопасности системы как конфиденциальность и целостность. Последствиями реализации данной угрозы является утечка, изменение или удаление конфиденциальных данных пользователя.

Угроза перехвата передаваемых данных. Возможность реализации данной угрозы обусловлена передачей данных по незащищенному каналу

связи. Для реализации данной угрозы нарушитель может воспользоваться недостатками компонента шифрования разрабатываемой аналитической системы. Наличие данной уязвимости приводит к нарушению конфиденциальности и представляет угрозу для данных пользователя.

К рассмотренным угрозам также относится угроза подмены web-сайта. Данная угроза может быть реализована в случае, если используется протокол HTTP и отсутствует сертификат SSL. Реализация данной угрозы может привести к тому, что данные, передаваемые пользователем, будут переданы на сторонние ресурсы, тем самым будут нарушены свойства системы, такие как конфиденциальность и целостность.

Кроме того, были рассмотрены угрозы, которым должна противостоять среда, в которой функционирует разрабатываемая система. К таким угрозам можно отнести внедрение компьютерного вируса в устройство, на котором развернута ИАС и хранится база данных. Данная угроза реализуема в виду отсутствия или наличия неполного комплекса средств защиты информации в информационной системе. Ресурсами, подверженными угрозе, в данном случае являются конфиденциальная информация пользователей, хранящаяся в базе данных, файлы самой информационной системы и т.д. При реализации данной угрозы будут нарушены такие свойства системы как целостность, конфиденциальность и доступность. Последствиями угрозы являются утечка конфиденциальной информации и нарушение функционирования разрабатываемой ИАС.

Также существует угроза DDoS-атаки разрабатываемой системы. Данная угроза реализуется через уязвимости в настройках сервера. При реализации данной угрозы ресурс перегружается запросами и становится недоступным для использования, тем самым нарушается доступность системы. Последствиям реализации данной угрозы становится отказ пользователям в доступе к системе.

5.2 Реализация средств обеспечения безопасности в клиентских приложениях

Для предотвращения угрозы перехвата данных в разрабатываемой системе реализован модуль шифрования. Перед передачей информации данные шифруются, а после получения расшифровываются для последующей обработки (Рисунок 5.1). При этом данные по сети передаются в зашифрованном виде, тем самым, при перехвате данных, злоумышленник не сможет получить информацию.

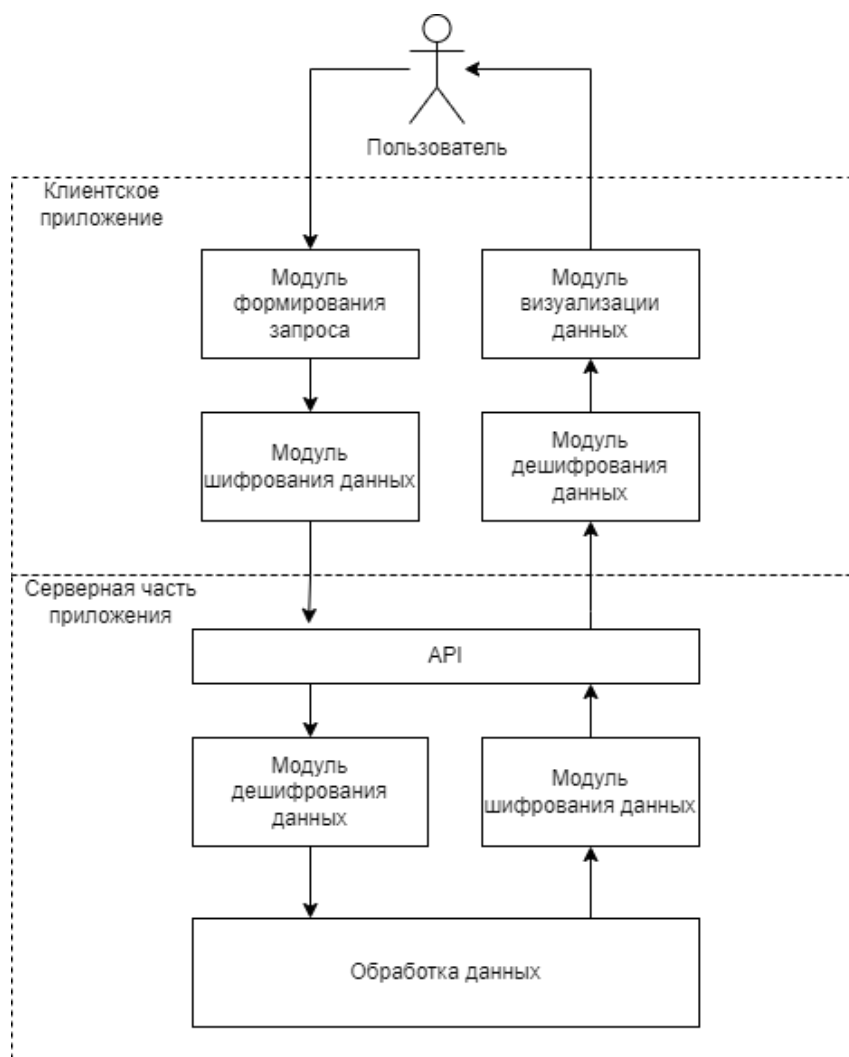


Рисунок 5.1 – Алгоритм передачи данных между клиентским приложением и сервером

Так как пользователь может взаимодействовать с системой посредством браузера, то и на этом уровне должна быть обеспечена защита передаваемых данных. Однако, встроить модули шифрования в программный код браузеров

не является возможным. Поэтому в разрабатываемой системе применяется протокол HTTPS. Протокол обеспечивает шифрование данных между пользователем и сервером. Кроме того, использование этого протокола позволяет реализовать защиту от угрозы подмены контента web-сайта. Получение SSL-сертификата реализовано с помощью сервиса Let's Encrypt.

В качестве дополнительного механизма защиты применяется технология HSTS. Суть технологии заключается в том, что браузеру передается защищенный заголовок HTTP Strict Transport Security, после чего браузер автоматически будет обращаться к ресурсу по протоколу HTTPS, а соединение с использованием протокола HTTP перестает обслуживаться.

Многие старые версии браузеров не могут обеспечить безопасность данных в достаточной мере. В виду этого, при разработке ИАС, было принято решение реализовать запрет на использование системы в устаревших версиях браузеров, которые не соответствуют современным нормам обеспечения информационной безопасности. Кроме того, на сервере используются последние версии программного обеспечения, что позволяет использовать актуальные средства защиты информации.

Также, для обеспечения информационной безопасности на программном уровне ИАС разрабатывалась в соответствии с правилами объектно-ориентированного программирования. В их число входит обеспечение такого свойства как инкапсуляция, благодаря чему в программном коде исключается передача скрытых данных в зону видимости пользователя. Это исключает возможное раскрытие конфиденциальной информации.

5.3 Реализация средств обеспечения безопасности в серверных приложениях

Для обеспечения информационной безопасности на серверной части разрабатываемого продукта реализованы такие компоненты, как модуль логирования, модуль шифрования и дешифрования сессии и модуль авторизации.

Модуль логирования необходим для сохранения информации о работе разрабатываемой ИАС. Наличие лог-файлов позволяет администратору или программисту быстрее определять неисправности и уязвимости системы для их предотвращения. Также анализ лог-файлов позволяет определить перечень действий, которые могли привести к некорректному поведению программного продукта, или вычислить и заблокировать доступ нарушителю, желающему навредить работе аналитической системы.

Реализованный модуль авторизации позволяет ограничить доступ пользователей к системе. Данный модуль решает проблему несанкционированного доступа к конфиденциальной информации. Доступ организуется посредством ввода логина и пароля. Логин и пароль хранятся в базе данных, при этом пароль подвергается хешированию. То есть пароли в базе данных не хранятся в открытом виде, что гарантирует обеспечение конфиденциальности даже в случае получения злоумышленником доступа к базе данных.

В случае получения нарушителем доступа к файлам cookie пользователя, нарушитель может получить доступ к личному кабинету пользователя. Для предотвращения данной угрозы в программный код разрабатываемого продукта внедрена дополнительная проверка данных браузера и IP-адреса пользователя.

В виде базового средства обеспечения безопасности на сервере в качестве операционной системы используется Linux Ubuntu 22.04 LTS. Выбор в пользу Linux был сделан, потому что большинство существующих вирусов ориентированы на Windows, также в Linux реализовано четкое разделение привилегий пользователей.

В качестве межсетевого экрана на сервере используется утилита iptables. Данная утилита с помощью установленных правил контролирует входящие и исходящие пакеты данных, а также в зависимости от правил при необходимости блокирует трафик.

Для подключения к серверу используется протокол SSH, это обеспечивает шифрование сеанса связи с сервером. Кроме того, для подключения к серверу необходимо пройти этап двухфакторной аутентификации. Данный механизм реализован с помощью модуля Google Authenticator PAM. Благодаря чему кроме имени пользователя и пароля, требуется дополнительно ввести генерируемый верификационный код. Данная реализация гарантирует сохранение конфиденциальности данных, даже в случае рассекречивания логина и пароля для подключения к серверу.

На сервере настроены ежедневные бэкапы данных, при этом данные сохраняются как в локальном, так и в облачном хранилище. Это необходимо, чтобы в случае ЧП можно было быстро восстановить утерянные данные. Бэкап серверных файлов осуществляется при помощи утилиты `gnome-disk-utility`, резервное копирование база данных осуществляется средствами СУБД. Важным в данном случае является хранение удаленных резервных копий в облачном хранилище, при это передача данных происходит с применением шифрования.

В качестве дополнительных средств обеспечения безопасности системы на сервере используется специальное программное обеспечение. В случае кражи носителя данных с сервера необходимо предусмотреть защиту от утечки данных, для шифрования файлов на сервере используется VeraCrypt 1.25.9. Плюсами данной программы является возможность шифрования диска сервера с использованием алгоритма шифрования «Кузнечик» и алгоритма вычисления хэш-функции «Стрибог».

Для управления базой данных на сервере установлена СУБД MySQL Server 8.0.29. Данная система поддерживает шифрование по алгоритму AES-256, благодаря чему существует возможность обеспечить защиту полей базы данных от несанкционированного доступа.

В качестве антивирусного ПО используется ESET NOD32 Antivirus Business Edition. Данное ПО входит в список лучших средств антивирусной

защиты для серверов. Выбранный антивирус входит в государственный реестр сертифицированных средств защиты информации ФСТЭК России.

Также важным аспектом обеспечения информационной безопасности серверной части разрабатываемой системы является использование последних версий программного обеспечения. Это необходимо, так как старые версии приложений могут иметь уязвимости, влияющие на безопасность работы ИАС.

Также стоит отметить, что угроза доступа к конфиденциальной информации пользователей системы может быть реализована путем внедрения компьютерного вируса на персональные устройства пользователя. Однако в данном случае ответственность за утечку данных несет сам пользователь.

5.4 Технические средства обеспечения безопасности

Технические средства необходимы для обеспечения таких свойств системы, как доступность, целостность и конфиденциальность.

Для защиты сервера от несанкционированного доступа, серверное оборудование располагается в специально оборудованном помещении, с ограниченным доступом. Для защиты от проникновения в данное помещение использована защитная дверь с запирающим механизмом, также помещение оснащено защитной сигнализацией.

При этом в помещении обеспечена постоянная поддержка оптимальной температуры воздуха, что реализуется установленным кондиционером. Кроме того, серверное помещение оборудовано системой автоматического пожаротушения.

Для обеспечения постоянной доступности ресурса, сервер обеспечен резервным источником питания, наилучшим решением является использование источников бесперебойного питания (ИБП). Также для обеспечения доступности ресурса через интернет сервер можно обеспечить

резервным каналом связи, и в случае неполадок в работе основного канала связи, провайдер сможет переключить сервер на резервный канал.

5.5 Описание организационных мер обеспечения безопасности

Для обеспечения безопасности системы разработана политика безопасности, которая включает в себя следующие правила:

- Должна быть разработана внутренняя документация, включающая правила работы с разрабатываемой системой.
- Должен быть составлен алгоритм действий, применяемых в случае выхода сервера из строя.
- Необходимо производить инструктаж персонала об ответственности за разглашение конфиденциальной информации.
- Должна быть реализована регистрация любых событий, относящихся к нарушениям безопасности.
- Настройка антивирусного программного обеспечения должна осуществляться только уполномоченными субъектами системы.
- Должна быть обеспечена защита от несанкционированного доступа к данной разрабатываемой ИАС.
- Должна быть обеспечена регулярная проверка файлов и областей памяти с целью обнаружения объектов, зараженных компьютерными вирусами.
- Должна быть обеспечена возможность изоляции и удаления файлов, зараженных компьютерными вирусам.
- В системе антивирусного программного обеспечения должно быть реализовано автоматическое обновление базы данных компьютерных вирусов.

ЗАКЛЮЧЕНИЕ

В результате выполнения данной работы была разработана информационно-аналитическая система для анализа фондового рынка. В работе была обоснована актуальность данной темы, исследована предметная область и проанализированы существующие решения.

На основе анализа были выставлены функциональные требования к программному продукту и составлено техническое задание. Также для реализации данного программного продукта были реализованы методы анализа изменения котировок акций на фондовых биржах.

Разработанная аналитическая система позволяет прогнозировать изменения на фондовых биржах, а также выявлять зависимость изменения показателей акций на фондовых биржах от событий, происходящих внутри компаний, отрасли и в мире в целом.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Васюткина, И. А. Разработка клиент-серверных приложений на языке C# : учебное пособие / И. А. Васюткина. — Новосибирск : Новосибирский государственный технический университет, 2016. — 112 с. — ISBN 978-5-7782-2932-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/91508.html> (дата обращения: 25.12.2021). — Режим доступа: для авторизир. пользователей
2. Васюткина И.А. Технология разработки объектно-ориентированных программ на JAVA : учебно-методическое пособие / Васюткина И.А.. — Новосибирск : Новосибирский государственный технический университет, 2012. — 152 с. — ISBN 978-5-7782-1973-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/45047.html> (дата обращения: 25.12.2021). — Режим доступа: для авторизир. пользователей
3. Пирская Л.В. Разработка мобильных приложений в среде Android Studio : учебное пособие / Пирская Л.В.. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2019. — 123 с. — ISBN 978-5-9275-3346-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/100196.html> (дата обращения: 25.12.2021). — Режим доступа: для авторизир. пользователей
4. Шацков В.В. Программирование приложений баз данных с использованием СУБД MS SQL Server : учебное пособие / Шацков В.В.. — Санкт-Петербург : Санкт-Петербургский государственный архитектурно-строительный университет, ЭБС АСВ, 2015. — 80 с. — ISBN 978-5-9227-0607-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/63638.html> (дата обращения: 25.12.2021). — Режим доступа: для авторизир. пользователей

5. Введение в СУБД MySQL : учебное пособие / . — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2021. — 228 с. — ISBN 978-5-4497-0912-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/102004.html> (дата обращения: 25.12.2021). — Режим доступа: для авторизир. пользователей

6. Тарасов С.В. СУБД для программиста. Базы данных изнутри / Тарасов С.В.. — Москва : СОЛОН-Пресс, 2018. — 320 с. — ISBN 978-2-7466-7383-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/90409.html> (дата обращения: 25.12.2021). — Режим доступа: для авторизир. пользователей

ПРИЛОЖЕНИЕ

Приложение 1. Листинг

```

public class EntityFrequency
{
    public Referent Entity { get; set; }
    public List<int> SentenseIds { get; set; }
    public int MentionsCount { get; set; }
}

public class SentimentPredictionModel
{
    public static string NEGATIVE = "negative";
    public static string POSITIVE = "positive";
    public static string NEUTRAL = "neutral";

    public static string DatasetPath = Path.Combine(Environment.CurrentDirectory,
"Models\\SentimentPrediction\\Model", "train.txt");
    public string Text { get; set; }
    public string[] Sentenses { get; set; }
    public List<EntityFrequency> EntitiesFrequency { get; set; }
    protected ITrainerBase trainer;
    public SentimentPredictionModel()
    {
        this.trainer = new OneVersusAllTrainer();
    }

    public void Train()
    {
        Console.WriteLine("*****");
        Console.WriteLine($"{trainer.Name}");
        Console.WriteLine("*****");

        trainer.Load();
    }
}

```

```

trainer.Fit(SentimentPredictionModel.DatasetPath);
var modelMetrics = trainer.Evaluate();

Console.WriteLine($"Macro                                     Accuracy:
{modelMetrics.MacroAccuracy:#.##}{Environment.NewLine}" +
    $"Micro                                     Accuracy:
{modelMetrics.MicroAccuracy:#.##}{Environment.NewLine}" +
    $"Log Loss: {modelMetrics.LogLoss:#.##}{Environment.NewLine}"
+
    $"Log                                     Loss                                     Reduction:
{modelMetrics.LogLossReduction:#.##}{Environment.NewLine}");
trainer.Save();
}

public List<EntitySentimentPrediction> Predict(string text)
{
    Console.WriteLine("*****");
    Console.WriteLine($" {trainer.Name}");
    Console.WriteLine("*****");

    List<SentimentData> sentimentData = new List<SentimentData>();
    string[] sentences = text.Split(" ", StringSplitOptions.RemoveEmptyEntries);
    List<EntityFrequency> entities = GetTextEntities(sentences);

    trainer.Load();
    var predictor = new Predictor();
    List<EntitySentimentPrediction> predictions = new
List<EntitySentimentPrediction>();

    foreach (EntityFrequency entity in entities)
    {
        SentimentData data = new SentimentData(GetTextWithReplaceEntity(entity,
sentences));

        var prediction = predictor.Predict(data);

```

```

        //if (prediction.PredictedLabel != SentimentPredictionModel.NEUTRAL)
        //{
            EntitySentimentPrediction entityPrediction = new
EntitySentimentPrediction(entity, prediction);
            predictions.Add(entityPrediction);
        //}
        Console.WriteLine("-----");
        Console.WriteLine($"Text: {data.SentimentText}");
        Console.WriteLine($"Prediction: {prediction.PredictedLabel:###}");
        Console.WriteLine("-----");
    }

    return predictions;
}

protected List<EntityFrequency> GetTextEntities(string[] sentences)
{
    List<EntityFrequency> entitiesFrequency = new List<EntityFrequency>();

    Pullenti.Sdk.InitializeAll();
    for (int i = 0; i < sentences.Length; i++)
    {
        AnalysisResult result = null;
        try
        {
            // создаём экземпляр процессора со стандартными анализаторами
            Processor processor = ProcessorService.CreateProcessor();
            // запускаем на тексте text
            result = processor.Process(new SourceOfAnalysis(sentences[i]));
        }
        catch (Exception e)
        {
            continue;
        }
        // получили выделенные сущности
    }
}

```

```

foreach (Referent entity in result.Entities)
{
    if (entity.InstanceOf.Name != "ORGANIZATION" &&
!(entity.InstanceOf.Name == "GEO" && ((GeoReferent)entity).IsState))
        continue;

    bool needToCreateNewEntityFrequency = true;
    foreach (EntityFrequency entityFrequency in entitiesFrequency)
    {
        if (entity.CanBeEquals(entityFrequency.Entity))
        {
            entityFrequency.MentionsCount++;
            if (!entityFrequency.SentenceIds.Contains(i))
            {
                entityFrequency.SentenceIds.Add(i);
            }
            if (i > 0 && !entityFrequency.SentenceIds.Contains(i - 1))
            {
                entityFrequency.SentenceIds.Add(i - 1);
            }
            if (i < sentences.Length - 1 && !entityFrequency.SentenceIds.Contains(i
+ 1))
            {
                entityFrequency.SentenceIds.Add(i + 1);
            }
            needToCreateNewEntityFrequency = false;
            continue;
        }
    }
    if (needToCreateNewEntityFrequency)
    {
        entitiesFrequency.Add(new EntityFrequency
        {
            Entity = entity,
            MentionsCount = 1,

```

```

        SentenseIds = new List<int>() { i }
    });
    if (i > 0 && !entitiesFrequency.Last().SentenseIds.Contains(i - 1))
    {
        entitiesFrequency.Last().SentenseIds.Add(i - 1);
    }
    if (i < sentences.Length - 1 &&
!entitiesFrequency.Last().SentenseIds.Contains(i + 1))
    {
        entitiesFrequency.Last().SentenseIds.Add(i + 1);
    }
}

    Console.WriteLine(entity.ToString());
}
}
Console.WriteLine(entitiesFrequency);
return entitiesFrequency;
}

protected string GetTextWithReplaceEntity(EntityFrequency entityFrequency,
string[] sentences, string replacer = "XxX")
{
    string text = String.Empty;
    foreach (int sentenseIndex in entityFrequency.SentenseIds)
    {
        text += sentences[sentenseIndex];
    }
    foreach (TextAnnotation annotation in entityFrequency.Entity.Occurrence)
    {
        text = text.Replace(annotation.ToString(), replacer);
    }
    return text;
}

```

```

public void TrainEvaluatePredict(ITrainerBase trainer, SentimentData newSample,
bool needToFit = false)
{
    Console.WriteLine("*****");
    Console.WriteLine($"{trainer.Name}");
    Console.WriteLine("*****");

    trainer.Load();

    if (needToFit)
    {
        trainer.Fit(SentimentPredictionModel.DatasetPath);
        var modelMetrics = trainer.Evaluate();

        Console.WriteLine($"Macro Accuracy:
{modelMetrics.MacroAccuracy:#.##}{Environment.NewLine}" +
            $"Micro Accuracy:
{modelMetrics.MicroAccuracy:#.##}{Environment.NewLine}" +
            $"Log Loss:
{modelMetrics.LogLoss:#.##}{Environment.NewLine}" +
            $"Log Loss Reduction:
{modelMetrics.LogLossReduction:#.##}{Environment.NewLine}");
        trainer.Save();
    }

    var predictor = new Predictor();
    var prediction = predictor.Predict(newSample);
    Console.WriteLine("-----");
    Console.WriteLine($"Prediction: {prediction.PredictedLabel:#.##}");
    Console.WriteLine("-----");
}
}

public class SentimentData
{

```

```
[LoadColumn(0)]
public string SentimentText;
```

```
[LoadColumn(1)]
public string Sentiment;
```

```
public SentimentData(string text)
{
    SentimentText = text;
}
}
```

```
public class SentimentPrediction
{
    [ColumnName("PredictedLabel")]
    public string PredictedLabel { get; set; }
}
```

```
public class EntitySentimentPrediction
{
    public SentimentPrediction Prediction;

    public EntityFrequency Entity;
```

```
    public EntitySentimentPrediction(EntityFrequency entity, SentimentPrediction
prediction)
    {
        Prediction = prediction;
        Entity = entity;
    }
}
```

```
public interface ITrainerBase
{
    string Name { get; }
```

```

void Fit(string trainingFileName);
MulticlassClassificationMetrics Evaluate();
void Save();
void Load();
}

```

```

/// <summary>

```

```

/// Base class for Trainers.

```

```

/// This class exposes methods for training, evaluating and saving ML Models.

```

```

/// Classes that inherit this class need to assign concrete model and name; and to
implement data pre-processing.

```

```

/// </summary>

```

```

public abstract class TrainerBase<TParameters> : ITrainerBase

```

```

    where TParameters : class

```

```

{

```

```

    public string Name { get; protected set; }

```

```

    protected static string ModelPath => Path.Combine(AppContext.BaseDirectory,
"classification.mdl");

```

```

    protected readonly MLContext MLContext;

```

```

    protected DataOperationsCatalog.TrainTestData _dataSplit;

```

```

    protected ITrainerEstimator<MulticlassPredictionTransformer<TParameters>,
TParameters> _model;

```

```

    protected ITransformer _trainedModel;

```

```

    protected TrainerBase()

```

```

    {

```

```

        MLContext = new MLContext(111);

```

```

    }

```

```

/// <summary>

```

```

/// Train model on defined data.

```

```

/// </summary>

```



```

/// <param name="trainingFileName"></param>
public void Fit(string trainingFileName)
{
    if (!File.Exists(trainingFileName))
    {
        throw new FileNotFoundException($"File {trainingFileName} doesn't exist.");
    }

    _dataSplit = LoadAndPrepareData(trainingFileName);
    var dataProcessPipeline = BuildDataProcessingPipeline();
    var trainingPipeline = dataProcessPipeline
        .Append(_model)

.Append(MIContext.Transforms.Conversion.MapKeyToValue("PredictedLabel"));

    _trainedModel = trainingPipeline.Fit(_dataSplit.TrainSet);
}

/// <summary>
/// Evaluate trained model.
/// </summary>
/// <returns>RegressionMetrics object which contain information about model
performance.</returns>
public MulticlassClassificationMetrics Evaluate()
{
    var testSetTransform = _trainedModel.Transform(_dataSplit.TestSet);

    return MIContext.MulticlassClassification.Evaluate(testSetTransform);
}

/// <summary>
/// Save Model in the file.
/// </summary>
public void Save()
{

```

```

        MIContext.Model.Save(_trainedModel, _dataSplit.TrainSet.Schema, ModelPath);
    }

    /// <summary>
    /// Load Model from the file.
    /// </summary>
    public void Load()
    {
        DataViewSchema schema;
        _trainedModel = MIContext.Model.Load(ModelPath, out schema);
    }

    /// <summary>
    /// Feature engeneering and data pre-processing.
    /// </summary>
    /// <returns>Data Processing Pipeline.</returns>
    private EstimatorChain<NormalizingTransformer> BuildDataProcessingPipeline()
    {
        var dataProcessPipeline =
MIContext.Transforms.Conversion.MapValueToKey(inputColumnName:
nameof(SentimentData.Sentiment), outputColumnName: "Label")
        .Append(MIContext.Transforms.Text.FeaturizeText("Features",
nameof(SentimentData.SentimentText)))
        .Append(MIContext.Transforms.NormalizeMinMax("Features", "Features"))
        .AppendCacheCheckpoint(MIContext);

        return dataProcessPipeline;
    }

    private DataOperationsCatalog.TrainTestData LoadAndPrepareData(string
trainingFileName)
    {
        var trainingDataView =
MIContext.Data.LoadFromTextFile<SentimentData>(trainingFileName, hasHeader: false,
separatorChar: '\t');

```

```

        return MIContext.Data.TrainTestSplit(trainingDataView, testFraction: 0.3);
    }
}

public class LbfgsMaximumEntropyTrainer :
TrainerBase<MaximumEntropyModelParameters>
{
    public LbfgsMaximumEntropyTrainer() : base()
    {
        Name = "LBFGS Maximum Entropy";
        _model = MIContext.MulticlassClassification.Trainers
            .LbfgsMaximumEntropy(labelColumnName: "Label", featureColumnName:
"Features");
    }
}

public class NaiveBayesTrainer :
TrainerBase<NaiveBayesMulticlassModelParameters>
{
    public NaiveBayesTrainer() : base()
    {
        Name = "Naive Bayes";
        _model = MIContext.MulticlassClassification.Trainers
            .NaiveBayes(labelColumnName: "Label", featureColumnName:
"Features");
    }
}

public class OneVersusAllTrainer : TrainerBase<OneVersusAllModelParameters>
{
    public OneVersusAllTrainer() : base()
    {
        Name = "One Versus All";
        _model = MIContext.MulticlassClassification.Trainers
            .OneVersusAll(binaryEstimator:
MIContext.BinaryClassification.Trainers.SgdCalibrated());
    }
}

```

```

    }
}

public class SdcaMaximumEntropyTrainer :
TrainerBase<MaximumEntropyModelParameters>
{
    public SdcaMaximumEntropyTrainer() : base()
    {
        Name = "Sdca Maximum Entropy";
        _model = MLContext.MulticlassClassification.Trainers
            .SdcaMaximumEntropy(labelColumnName: "Label", featureColumnName:
"Features");
    }
}

public class SdcaNonCalibratedTrainer :
TrainerBase<LinearMulticlassModelParameters>
{
    public SdcaNonCalibratedTrainer() : base()
    {
        Name = "Sdca NonCalibrated";
        _model = MLContext.MulticlassClassification.Trainers
            .SdcaNonCalibrated(labelColumnName: "Label", featureColumnName:
"Features");
    }
}

public class Predictor
{
    protected static string ModelPath => Path.Combine(AppContext.BaseDirectory,
"classification.mdl");
    private readonly MLContext _mlContext;

    private ITransformer _model;

    public Predictor()

```

```

    {
        _mlContext = new MLContext(111);
    }

    /// <summary>
    /// Runs prediction on new data.
    /// </summary>
    /// <param name="newSample">New data sample.</param>
    /// <returns>PalmerPenguinsData object, which contains predictions made by
model.</returns>

    public SentimentPrediction Predict(SentimentData newSample)
    {
        LoadModel();

        var predictionEngine =
_mlContext.Model.CreatePredictionEngine<SentimentData, SentimentPrediction>(_model);

        return predictionEngine.Predict(newSample);
    }

    private void LoadModel()
    {
        if (!File.Exists(ModelPath))
        {
            throw new FileNotFoundException($"File {ModelPath} doesn't exist.");
        }

        using (var stream = new FileStream(ModelPath, FileMode.Open, FileAccess.Read,
FileShare.Read))
        {
            _model = _mlContext.Model.Load(stream, out _);
        }

        if (_model == null)
        {

```

```

        throw new Exception($"Failed to load Model");
    }
}

```

```

public class WorldNewsController : Controller
{
    private readonly diplomContext _context;
    private readonly IConfiguration _configuration;

    public WorldNewsController(diplomContext context, IConfiguration configuration)
    {
        _context = context;
        _configuration = configuration;
    }

    // GET: WorldNews
    public async Task<IActionResult> Index()
    {
        return View(await _context.WorldNews.ToListAsync());
    }

    // GET: WorldNews/Details/5
    public async Task<IActionResult> Details(int? id)
    {
        if (id == null)
        {
            return NotFound();
        }

        var worldNews = await _context.WorldNews
            .FirstOrDefaultAsync(m => m.Id == id);
        if (worldNews == null)
        {
            return NotFound();
        }
    }
}

```

```

    }

    var newsQuotesImpacts = from impact in
worldNews.NewsQuotesImpacts.DistinctBy(_impact => _impact.CompanyId)
    orderby Math.Abs(impact.Influence) descending
    orderby impact.Influence descending
    select impact;

    ViewBag.ImpactCompanies = newsQuotesImpacts.Take(10).ToArray();

    return View(worldNews);
}

// GET: WorldNews/Create
public IActionResult Create()
{
    return View();
}

// POST: WorldNews/Create
// To protect from overposting attacks, enable the specific properties you want to bind
to.
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<IActionResult> Create([Bind("Id,DateTime")] WorldNews
worldNews)
{
    if (ModelState.IsValid)
    {
        _context.Add(worldNews);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }
    return View(worldNews);
}

```

```
// GET: WorldNews/Edit/5
```

```
public async Task<IActionResult> Edit(int? id)
```

```
{
```

```
    if (id == null)
```

```
    {
```

```
        return NotFound();
```

```
    }
```

```
    var worldNews = await _context.WorldNews.FindAsync(id);
```

```
    if (worldNews == null)
```

```
    {
```

```
        return NotFound();
```

```
    }
```

```
    return View(worldNews);
```

```
}
```

```
// POST: WorldNews/Edit/5
```

```
// To protect from overposting attacks, enable the specific properties you want to bind
```

to.

```
// For more details, see http://go.microsoft.com/fwlink/?LinkId=317598.
```

```
[HttpPost]
```

```
[ValidateAntiForgeryToken]
```

```
public async Task<IActionResult> Edit(int id, [Bind("Id,DateTime")] WorldNews
```

worldNews)

```
{
```

```
    if (id != worldNews.Id)
```

```
    {
```

```
        return NotFound();
```

```
    }
```

```
    if (ModelState.IsValid)
```

```
    {
```

```
        try
```

```
        {
```



```

        _context.Update(worldNews);
        await _context.SaveChangesAsync();
    }
    catch (DbUpdateConcurrencyException)
    {
        if (!WorldNewsExists(worldNews.Id))
        {
            return NotFound();
        }
        else
        {
            throw;
        }
    }
    return RedirectToAction(nameof(Index));
}
return View(worldNews);
}

// GET: WorldNews/Delete/5
public async Task<IActionResult> Delete(int? id)
{
    if (id == null)
    {
        return NotFound();
    }

    var worldNews = await _context.WorldNews
        .FirstOrDefaultAsync(m => m.Id == id);
    if (worldNews == null)
    {
        return NotFound();
    }

    return View(worldNews);
}

```

```

    }

    // POST: WorldNews/Delete/5
    [HttpPost, ActionName("Delete")]
    [ValidateAntiForgeryToken]
    public async Task<IActionResult> DeleteConfirmed(int id)
    {
        var worldNews = await _context.WorldNews.FindAsync(id);
        _context.WorldNews.Remove(worldNews);
        await _context.SaveChangesAsync();
        return RedirectToAction(nameof(Index));
    }

    public List<WorldNews> GetWorldNews(DateTime? date = null)
    {
        var query = _context.WorldNews;
        if (date != null)
            return query.Where(news => news.DateTime.Date ==
date.Value.Date).ToList();
        else
            return query.ToList();
    }

    private bool WorldNewsExists(int id)
    {
        return _context.WorldNews.Any(e => e.Id == id);
    }

    public async Task UpdateWorldNews()
    {
        IQueryable<WorldNews> worldNews = _context.WorldNews;
        WorldNews lastWorldNews = worldNews.Any() ? worldNews.OrderBy(news =>
news.DateTime).Last() : null;

        DateTime parsingDate = new DateTime();

```

```

if (lastWorldNews == null)
{
    int startYear = int.Parse(_configuration["ParsingPeriod:Start:year"]);
    int startMonth = int.Parse(_configuration["ParsingPeriod:Start:month"]);
    int startDay = int.Parse(_configuration["ParsingPeriod:Start:day"]);
    parsingDate = new DateTime(startYear, startMonth, startDay);
}
else
{
    parsingDate = lastWorldNews.DateTime;
}
parsingDate = parsingDate.AddDays(1);

while (parsingDate.Date <= DateTime.Now.Date)
{
    string formatDate = parsingDate.ToString(@"yyyy\MM\dd");
    string url = $"/news/{formatDate}";

    ParseWorldNews(url, parsingDate);
    parsingDate = parsingDate.AddDays(1);
    await _context.SaveChangesAsync();
}
}

private void ParseWorldNews(string url, DateTime parsingDate)
{
    string pattern = "item news.*?class=\"time\">(.*?)<.*?<a class=\"titles\" href=\"(.*?)\">.*?class=\"card-title\">(.*?)<";
    pattern = "_news.*?card-full-news__date\">(.*?)<.*?<a href=\"(.*?)\">.*?card-full-news__title\">(.*?)<";
    string formatDate = parsingDate.ToString("dd");
    string formatMonth = parsingDate.ToString("MM");
    string html = Helper.GetStringFromHtml(_configuration["WorldNewsDomain"] + url, Encoding.GetEncoding(65001));
    foreach (Match match in Regex.Matches(html, pattern))

```

```

    {
        try
        {
            bool isEconomyNews = true;
            string title = match.Groups[3].Value;
            string[] time = match.Groups[1].Value.Split(',')[0].Split(':');
            DateTime newsPublicationDate = parsingDate.AddHours(int.Parse(time[0]));
            newsPublicationDate
newsPublicationDate.AddMinutes(int.Parse(time[1]));
            string newsUrl = _configuration["WorldNewsDomain"] +
match.Groups[2].Value;
            Thread.Sleep(400);
            string? newsText = Helper.GetStringFromHtml(newsUrl,
Encoding.GetEncoding(65001));
            if (newsText == null)
                continue;
            string newsRubricPattern = "<a class=\"topic-header__item topic-
header__rubric\".*?>(.*?)<";
            foreach (Match rubric in Regex.Matches(newsText, newsRubricPattern))
            {
                if (rubric.Groups[1].Value != "Экономика")
                    isEconomyNews = false;
            }
            if (!isEconomyNews)
                continue;
            string newsTextPattern = "\"articleBody\": \"(.*?)\"";
            foreach (Match textMatch in Regex.Matches(newsText, newsTextPattern))
            {
                newsText = textMatch.Groups[1].Value;
            }

            if (_context.WorldNews.Where(news => news.Url == newsUrl).Count() ==
0)
            {

```

```

        WorldNews news = new WorldNews(newsPublicationDate, newsUrl,
newsText, title);

        _context.WorldNews.Add(news);
    }
}
catch (Exception e) { Console.WriteLine("Error: " + e.Message); }
}
}

public void GetWorldnewsImpact(int newsId, int? shareId = null)
{
    WorldNews worldNews = _context.WorldNews.Where(news => news.Id ==
newsId).ToList().First();
    if (worldNews == null)
        return;
    List<WorldNews> allNews = new List<WorldNews>() { worldNews
}; //_context.WorldNews.Where(news => news.Id > 29985).ToList();
    List<EntitySentimentPrediction> predictions = new
List<EntitySentimentPrediction>();
    foreach (var news in allNews)
    {
        if (news.Text.Contains("articleBody"))
        {
            string newsTextPattern = "\"articleBody\": \"(.*)\"";
            foreach (Match textMatch in Regex.Matches(news.Text, newsTextPattern))
            {
                news.Text = textMatch.Groups[1].Value;
                _context.WorldNews.Update(news);
            }
        }
    }

    SentimentPredictionModel model = new SentimentPredictionModel();
    predictions = model.Predict(news.Text);
    if (predictions.Count > 0)
    {

```

```

MorphAnalyzer morph;
List<MorphInfo> results = new List<MorphInfo>();
try
{
    morph = new MorphAnalyzer(withLemmatization: true);
    results = morph.Parse(news.Text.Split(' ').ToList());
}
catch (Exception)
{
    continue;
}
var keywords = new List<string>();
foreach (var morphInfo in results)
{
    if (keywords.Count >= 5)
        continue;
    var tag = morphInfo.BestTag;
    if (tag.HasLemma && tag.Power > 0.98)
    {
        Console.WriteLine($"{morphInfo.Text}:");
        Console.WriteLine($"    {tag} : {tag.Power}");
        keywords.Add(tag.Lemma);
    }
}
List<NewsQuotesImpact> impacts = new List<NewsQuotesImpact>();
foreach (var prediction in predictions)
{
    //bool isGeo = false;
    List<Share> shares = shareId != null ? _context.Shares.Where(share =>
share.Id == shareId).ToList() : _context.Shares.ToList();
    //List<Country> countries = new List<Country>();
    //var entity = prediction.Entity.Entity;
    List<Candle> candles = new List<Candle>();//
_context.Candles.Where(candle => candle.Time > news.DateTime.AddDays(-1).Date &&
candle.Time < news.DateTime.AddDays(5).Date).ToList();

```

```

foreach (var share in shares)
{
    if (share.Company == null || share.Company.BrandInfo == null)
        continue;

    var companyMorphInfo =
morph.Parse(share.Company.BrandInfo.Replace(" ", " ").Split(' ')).ToList();
    var companyKeywords = new List<string>();
    foreach (var morphInfo in companyMorphInfo)
    {
        if (companyKeywords.Count >= 5)
            continue;
        var tag = morphInfo.BestTag;
        if (tag.HasLemma && tag.Power > 0.5 && tag.Power < 0.7)
        {
            Console.WriteLine($"{morphInfo.Text}:");
            Console.WriteLine($"    {tag} : {tag.Power}");
            companyKeywords.Add(tag.Lemma);
        }
    }

    candles = share.Candles.Where(candle => candle.Time >
news.DateTime.AddDays(1).Date).ToList();
    if (candles.Count == 0)
    {
        SaveImpactForLastNews(share, news, companyKeywords, impacts);
    }
    else
    {
        candles = share.Candles.Where(candle => candle.Time >
news.DateTime.AddDays(-1).Date && candle.Time <
news.DateTime.AddDays(5).Date).ToList();
        if (candles.Count > 0)
        {
            Candle firstCandle = candles.First();

```

```

        Candle lastCandle = candles.Last();
        double? influence = ((firstCandle.Close - lastCandle.Close) * 100) /
(firstCandle.Close);

        if (influence != null && Math.Abs((double)influence) > 5)
        {
            impacts.Add(new NewsQuotesImpact(share.Company, news,
(double)influence));
        }
    }
}

if (impacts.Count > 0)
{
    foreach (var keyword in keywords)
    {
        Keyword dbKeyword = null;
        if (_context.Keywords.Where(k => k.Value == keyword).Count() > 0)
        {
            dbKeyword = _context.Keywords.Where(k => k.Value ==
keyword).First();
        }
        else
        {
            dbKeyword = new Keyword(keyword);
            _context.Keywords.Add(dbKeyword);
            _context.SaveChanges();
        }
        _context.WorldNewsKeywords.Add(new WorldNewsKeyword(news,
dbKeyword));

        _context.SaveChanges();
    }
    _context.NewsQuotesImpacts.AddRange(impacts);
    _context.SaveChanges();
}

```



```

    }
    }
}

private void SaveImpactForLastNews(Share share, WorldNews news, List<string>
companyKeywords, List<NewsQuotesImpact> impacts)
{
    double newsImpact = 0;
    foreach (var impact in share.Company.NewsQuotesImpacts)
    {
        double keywordsImpact = 0;
        foreach (var keyword in companyKeywords)
        {
            double oneKeywordImpact = 0;
            var keywordsCount = impact.WorldNews.Keywords.Where(kw =>
kw.Keyword.Value == keyword).ToList().Count();
            if (keywordsCount > 0)
            {
                oneKeywordImpact = impact.Influence * Math.Abs(impact.Influence /
keywordsCount);
            }
            keywordsImpact += oneKeywordImpact;
        }
        newsImpact += keywordsImpact / companyKeywords.Count();
    }
    if (newsImpact != 0)
    {
        impacts.Add(new NewsQuotesImpact(share.Company, news, newsImpact));
    }
}

public enum StockRecommendation
{
    Buy,

```

```

    Sell,
    Hold,
    Undefined
}

public static class FundamentalAnalysis
{
    private const double EnterpriseToEbitdaTurningPoint = 10;

    private const double DeptToEBITDAMaxTurningPoint = 2.5;
    private const double DeptToEBITDAMinTurningPoint = 2;

    private const double ForwardPETurningPoint = 15;

    private const double PriceToBookTurningPoint = 1;

    public static StockRecommendation EnterpriseValueToEBITDARatio(Share share)
    {
        if (share.Company == null || share.Company.EnterpriseToEbitda == null)
            return StockRecommendation.Undefined;

        if (share.Company.EnterpriseToEbitda <= EnterpriseToEbitdaTurningPoint)
            return StockRecommendation.Buy;
        else
            return StockRecommendation.Sell;
    }

    public static StockRecommendation PriceToBookRation(Share share)
    {
        if (share.Company == null || share.Company.PriceToBook == null)
            return StockRecommendation.Undefined;

        if (share.Company.PriceToBook <= PriceToBookTurningPoint)
            return StockRecommendation.Buy;
        else

```

```

        return StockRecommendation.Sell;
    }

    public static double DeptToEBITDARatioValue(Share share)
    {
        if (share.Company == null || share.Company.TotalDebt == null ||
share.Company.Ebitda == null)
            return double.NaN;

        return Math.Round((double)(double.Parse(share.Company.TotalDebt.ToString()) /
share.Company.Ebitda));
    }

    public static StockRecommendation DeptToEBITDARatio(Share share)
    {
        if (share.Company == null || share.Company.TotalDebt == null ||
share.Company.Ebitda == null)
            return StockRecommendation.Undefined;

        double deptToEBITDARatio = DeptToEBITDARatioValue(share);
        if (deptToEBITDARatio <= 0 || deptToEBITDARatio >=
DeptToEBITDAMaxTurningPoint)
            return StockRecommendation.Sell;
        if (deptToEBITDARatio < DeptToEBITDAMinTurningPoint)
            return StockRecommendation.Hold;
        else
            return StockRecommendation.Buy;
    }

    public static StockRecommendation ForwardPERatio(Share share)
    {
        if (share.Company == null || share.Company.ForwardPE == null)
            return StockRecommendation.Undefined;
    }

```

```

        if (share.Company.ForwardPE <= 0 || share.Company.ForwardPE >=
ForwardPETurningPoint)
            return StockRecommendation.Sell;
        else
            return StockRecommendation.Buy;
    }

    public static StockRecommendation GetSummaryRecommendation(Share share)
    {
        int result = 0;
        result += ParseRecommendationToInt(EnterpriseValueToEBITDARatio(share));
        result += ParseRecommendationToInt(DeptToEBITDARatio(share));
        result += ParseRecommendationToInt(ForwardPERatio(share));
        result += ParseRecommendationToInt(PriceToBookRation(share));

        if (result > 0)
            return StockRecommendation.Buy;
        if (result < 0)
            return StockRecommendation.Sell;
        else
            return StockRecommendation.Hold;
    }

    private static int ParseRecommendationToInt(StockRecommendation
recommendation)
    {
        switch (recommendation)
        {
            case StockRecommendation.Buy:
                return 1;
            case StockRecommendation.Sell:
                return -1;
            default:
                return 0;
        }
    }

```

```

    }

    public static bool IsHighPotentialShare(Share share)
    {
        if (share.Company == null || share.Candles.ToList().Count() < 0)
            return false;

        var lastCandle = share.Candles.Last();
        var preLastCandleTime = lastCandle.Time.AddYears(-1);
        Candle preLastCandle = null;

        foreach (var candle in share.Candles.Reverse())
        {
            if (candle.Time <= preLastCandleTime)
            {
                preLastCandle = candle;
                break;
            }
        }

        if(preLastCandle == null)
        {
            return false;
        }

        var delta = Math.Round((((double)preLastCandle.Close - (double)lastCandle.Close)
/ (double)lastCandle.Close * 100, 2);

        return share.Company.EnterpriseValue >= 100000000000 &&
share.Company.ForwardPE >= 15 && delta >= 10;
    }

    public static bool IsStableGrowthShare(Share share)
    {
        if (share.Company == null || share.Candles.ToList().Count() < 0)

```

```

        return false;

        var lastCandle = share.Candles.Last();
        var preLastCandle = share.Candles.First();

        var delta = Math.Round((((double)preLastCandle.Close - (double)lastCandle.Close)
/ (double)lastCandle.Close * 100, 2);
        return share.Company.EnterpriseValue >= 10000000000
            && (share.Company.ForwardPE >= 15 && share.Company.ForwardPE <= 35)
            && share.Company.DebtToEquity <= 300
            && share.Company.ProfitMargins >= 0.05
            && delta >= 20;
    }

    public static bool IsFastGrowingProfitShare(Share share)
    {
        if (share.Company == null || share.Candles.ToList().Count() < 0)
            return false;

        var lastCandle = share.Candles.Last();
        var preLastCandle = share.Candles.First();

        var delta = Math.Round((((double)preLastCandle.Close - (double)lastCandle.Close)
/ (double)lastCandle.Close * 100, 2);
        return share.Company.ForwardPE >= 15
            && share.Company.DebtToEquity <= 300
            && share.Company.TrailingEps >= 15
            && delta >= 30;
    }
}

public class ModelInput
{

```

```

    public float Close { get; set; }
    public DateTime Time { get; set; }
}

public class ModelOutput
{
    public float[] ForecastedClose { get; set; }
    public float[] LowerBoundClose { get; set; }
    public float[] UpperBoundClose { get; set; }
}

public class ForecastingModel
{
    const          string          connectionString          =
"server=localhost;database=test;user=root;password=1234";
    public enum ForecastHorizon
    {
        week,
        month,
        halfYear
    }
    private ForecastHorizon horizon;

    public ForecastingModel(ForecastHorizon horizon = ForecastHorizon.week)
    {
        this.horizon = horizon;
    }

    private string getFileName()
    {
        switch (this.horizon)
        {
            case ForecastHorizon.week:
                return "MLModel.zip";
            case ForecastHorizon.month:

```

```

        return "MLModel_month.zip";
    case ForecastHorizon.halfYear:
        return "MLModel_half_year.zip";
    default:
        return "MLModel.zip";
    }
}

```

```
private int getHorizonValue()
```

```

{
    switch (this.horizon)
    {
        case ForecastHorizon.week:
            return 7*24;
        case ForecastHorizon.month:
            return 24*30;
        case ForecastHorizon.halfYear:
            return 24*180;
        default:
            return 7*24;
    }
}

```

```
private string getModelPath()
```

```

{
    string rootDir =
Path.GetFullPath(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "../..../"));
    string fileName = this.GetFileName();
    return Path.Combine(rootDir, fileName);
}

```

```
private SsaForecastingEstimator GetForecastingPipeline(MLContext mlContext)
```

```

{
    SsaForecastingEstimator forecastingPipeline =
mlContext.Forecasting.ForecastBySsa(

```



```

        outputColumnName: "ForecastedClose",
        inputColumnName: "Close",
        windowSize: 365,
        seriesLength: 180 * 24,
        trainSize: 365 * 5,
        horizon: getHorizonValue(),
        confidenceLevel: 0.95f,
        confidenceLowerBoundColumn: "LowerBoundClose",
        confidenceUpperBoundColumn: "UpperBoundClose"
    );

    return forecastingPipeline;
}

private SsaForecastingTransformer getForecaster(long shareId, out MLContext
mlContext, out IDataView data, out DatabaseLoader loader)
{
    string modelPath = this.getModelPath();
    string query = $"SELECT close, time FROM candles where shareId =
{shareId}";

    SsaForecastingTransformer forecaster = null;

    DatabaseSource source = new DatabaseSource(MySqlConnectionFactory.Instance,
ForecastingModel.connectionString, query);
    mlContext = new MLContext();
    MLContext ctx = new MLContext();

    loader = mlContext.Data.CreateDatabaseLoader<ModelInput>();
    DatabaseLoader.Options options = new DatabaseLoader.Options();
    options.Columns = new[] { new DatabaseLoader.Column("Close", DbType.Single,
0), new DatabaseLoader.Column("Time", DbType.DateTime, 1) };
    loader = ctx.Data.CreateDatabaseLoader(options);

    data = loader.Load(source);

```

```

        //if (File.Exists(modelPath))
        //{
        //    using (var file = File.OpenRead(modelPath))
        //    {
        //        forecaster = (SsaForecastingTransformer)mlContext.Model.Load(file, out
DataViewSchema schema);
        //    }
        //}
        //else
        //{
        //    forecaster = this.GetForecastingPipeline(mlContext).Fit(data);
        //    mlContext.Model.Save(forecaster, loader, getFileName());
        //}

        return forecaster;
    }

    public void GetForecast(IConfiguration configuration, diplomContext context, int
shareId)
    {
        string rootDir =
Path.GetFullPath(Path.Combine(AppDomain.CurrentDomain.BaseDirectory, "../..../"));
        string modelPath = Path.Combine(rootDir, getFileName());
        string connectionString = ForecastingModel.connectionString;

        MLContext mlContext = new MLContext();

        IDataView data = null, evaluateDate = null;
        //SsaForecastingTransformer forecaster = null;
        ITransformer forecaster = null;
        TimeSeriesPredictionEngine<ModelInput, ModelOutput> forecastEngine = null;
        foreach (Share share in context.Shares.Where(share => share.Id ==
shareId).ToList())
        {

```

```

        DatabaseLoader loader =
mlContext.Data.CreateDatabaseLoader<ModelInput>();

        long candlesCount = context.Candles.Where(candle => candle.Share.Id ==
share.Id).Count();
        if (candlesCount < 30)
            continue;

        string query = @"select closeByDay.day as Time, closeByDay.close from (
            select
                DATE_FORMAT(Time, '%Y-%m-%d') as day,
                CAST(substring_index(group_concat(cast(close as CHAR) order by Time
desc), ',', 1 ) AS REAL) as close
            from
                candles
            where shareId = {share.Id}
            group by
                DATE_FORMAT(day, '%Y-%m-%d'), ShareId
            order by
                day
            ) as closeByDay;";
        query = $"SELECT close, time FROM candles where shareId = {share.Id}";
        DatabaseSource source = new
DatabaseSource(MySqlConnectionFactory.Instance, connectionString, query);
        MLContext ctx = new MLContext();
        DatabaseLoader.Options options = new DatabaseLoader.Options();
        options.Columns = new[] { new DatabaseLoader.Column("Close",
DbType.Single, 0), new DatabaseLoader.Column("Time", DbType.DateTime, 1) };
        loader = ctx.Data.CreateDatabaseLoader(options);

        data = loader.Load(source);

        //using (var file = File.OpenRead(modelPath))
        //{
        //    forecaster = mlContext.Model.Load(file, out DataViewSchema schema);

```

```

        //          //forecastEngine = forecaster.CreateTimeSeriesEngine<ModelInput,
ModelOutput>(mlContext);
        //}

        SsaForecastingEstimator forecastingPipeline =
mlContext.Forecasting.ForecastBySsa(
            outputColumnName: "ForecastedClose",
            inputColumnName: "Close",
            windowSize: 365,
            seriesLength: 180 * 24,
            trainSize: 365 * 5,
            horizon: getHorizonValue(),
            confidenceLevel: 0.95f,
            confidenceLowerBoundColumn: "LowerBoundClose",
            confidenceUpperBoundColumn: "UpperBoundClose");

        forecaster = forecastingPipeline.Fit(data);

        Evaluate(data, forecaster, mlContext);
        forecastEngine = forecaster.CreateTimeSeriesEngine<ModelInput,
ModelOutput>(mlContext);

        mlContext.Model.Save(forecaster, loader, this.GetFileName());
    }

    if (data != null && forecastEngine != null && forecaster != null)
    {
        forecastEngine.CheckPoint(mlContext, modelPath);
        Forecast(data, getHorizonValue(), forecastEngine, mlContext);
    }
}

public void Fit(long shareId)
{

```

```

        SsaForecastingTransformer forecaster = this.getForecaster(shareId, out
MLContext mlContext, out IDataView data, out DatabaseLoader loader);

```

```

        forecaster = this.GetForecastingPipeline(mlContext).Fit(data);
        Evaluate(data, forecaster, mlContext);
        mlContext.Model.Save(forecaster, loader, this.GetFileName());
    }

```

```

    public ModelOutput GetForecast(long shareId)
    {
        SsaForecastingTransformer forecaster = this.getForecaster(shareId, out
MLContext mlContext, out IDataView data, out DatabaseLoader loader);

```

```

        TimeSeriesPredictionEngine<ModelInput, ModelOutput> forecastEngine =
forecaster.CreateTimeSeriesEngine<ModelInput, ModelOutput>(mlContext);
        if (data != null && forecastEngine != null && forecaster != null)
        {
            forecastEngine.CheckPoint(mlContext, this.getModelPath());
            return Forecast(data, getHorizonValue(), forecastEngine, mlContext);
        }
        return null;
    }

```

```

    private void Evaluate(IDataView testData, ITransformer model, MLContext
mlContext)
    {
        IDataView predictions = model.Transform(testData);

        IEnumerable<float> actual =
            mlContext.Data.CreateEnumerable<ModelInput>(testData, true)
                .Select(observed => observed.Close);

        IEnumerable<float> forecast =
            mlContext.Data.CreateEnumerable<ModelOutput>(predictions, true)
                .Select(prediction => prediction.ForecastedClose[0]);

```

```

var metrics = actual.Zip(forecast, (actualValue, forecastValue) => actualValue -
forecastValue);

var MAE = metrics.Average(error => Math.Abs(error));
var RMSE = Math.Sqrt(metrics.Average(error => Math.Pow(error, 2)));
Console.WriteLine("Evaluation Metrics");
Console.WriteLine("-----");
Console.WriteLine($"Mean Absolute Error: {MAE:F3}");
Console.WriteLine($"Root Mean Squared Error: {RMSE:F3}\n");
}

private ModelOutput Forecast(IDataView testData, int horizon,
TimeSeriesPredictionEngine<ModelInput, ModelOutput> forecaster, MLContext mlContext)
{
    ModelOutput forecast = forecaster.Predict();

    IEnumerable<string> forecastOutput =
        mlContext.Data.CreateEnumerable<ModelInput>(testData, reuseRowObject:
false)
        .TakeLast(horizon)
        .Select((ModelInput input, int index) =>
        {
            float actualClose = input.Close;
            float lowerEstimate = Math.Max(0, forecast.LowerBoundClose[index]);
            float estimate = forecast.ForecastedClose[index];
            float upperEstimate = forecast.UpperBoundClose[index];
            return $"Date: {input.Time}\n" +
                $"Actual Rentals: {actualClose}\n" +
                $"Lower Estimate: {lowerEstimate}\n" +
                $"Forecast: {estimate}\n" +
                $"Upper Estimate: {upperEstimate}\n";
        });

    Console.WriteLine("Rental Forecast");
}

```

```
Console.WriteLine("-----");  
foreach (var prediction in forecastOutput)  
{  
    Console.WriteLine(prediction);  
}  
return forecast;  
}  
}
```