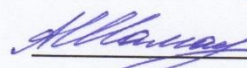


ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Н. П. ОГАРЁВА»

Факультет математики и информационных технологий
Кафедра систем автоматизированного проектирования

УТВЕРЖДАЮ

И. о. зав. кафедрой
канд. техн. наук

 А. В. Шамаев

«27» 06 2020 г.

БАКАЛАВРСКАЯ РАБОТА

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
АНАЛИЗА БИРЖЕВОГО РЫНКА**

Автор бакалаврской работы


12.06.2020  А. С. Вдовина

Обозначение бакалаврской работы БР-02069964-09.03.04-04-20

Направление 09.03.04 Программная инженерия

Руководитель работы

канд. техн. наук, доц.

22.06.2020  О. А. Гущина

Нормоконтролер

канд. техн. наук, доц.

15.06.2020  А. А. Буткина

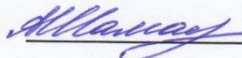
Саранск
2020

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМ. Н. П. ОГАРЁВА»

Факультет математики и информационных технологий
Кафедра систем автоматизированного проектирования

УТВЕРЖДАЮ

И. о. зав. кафедрой
канд. техн. наук

 А. В. Шамаев
«27» 06 2020 г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
(в форме бакалаврской работы)

- Студент Вдовина Алина Сергеевна 404 группа
- 1 Тема «Разработка программного обеспечения анализа биржевого рынка»
Утверждена приказом № 531-С от 30.01.2020 г.
- 2 Срок представления работы к защите 12.06.2020 г.
- 3 Исходные данные для научного исследования (проектирования) разработка веб-приложения для анализа котировок акций
- 4 Содержание выпускной квалификационной работы
- 4.1 Обзор подходов к решению задачи анализа биржевых котировок
- 4.2 Постановка задачи создания программного продукта анализа биржевых котировок акций
- 4.3 Разработка программного обеспечения анализа биржевых котировок акций
- 5 Приложения
- Приложение А (обязательное) Программный код серверной части ПО

Приложение Б (обязательное) Программный код клиентской части ПО

Приложение В (обязательное) Демонстрация работы ПО

Руководитель работы _____ *Гущина* *30.01.2020* *О А. Гущина*
подпись, дата инициалы, фамилия

Задание принял к исполнению _____ *31.01.2020* *В* *А. С. Вдовина*
подпись, дата инициалы, фамилия

РЕФЕРАТ

Выпускная квалификационная работа содержит 106 страниц, 13 рисунков, 10 таблиц, 1 формулу, 22 использованных источников, 3 приложения.

АНАЛИЗ БИРЖЕВОГО РЫНКА, ТЕХНИЧЕСКИЙ АНАЛИЗ, ЯПОНСКИЕ СВЕЧИ, ГИПОТЕЗЫ ФУНКЦИОНИРОВАНИЯ РЫНКА, ИНДИКАТОРЫ, ОСЦИЛЛЯТОРЫ.

Объект исследования – существующие методы анализа биржевых котировок.

Цель работы – разработка программного обеспечения анализа биржевого рынка.

Метод исследования – анализ и синтез.

Основные результаты – исследованы методы анализа биржевых котировок ценных бумаг фондовой биржи. Создано программное обеспечение анализа биржевых котировок акций российских компаний.

Область применения – биржевой фондовый рынок Российской Федерации.

Разработанное программное обеспечение позволит рядовому пользователю быстро получать информацию о котировках акций российских компаний, видеть график японских свечей по данным котировкам, анализировать индикаторы технического анализа на данном графике.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	7
1 Обзор подходов к решению задачи анализа биржевых котировок.....	9
1.1 Биржевой рынок. Фондовый рынок. Задачи фондовой биржи	9
1.2 Инструменты, торгуемые на бирже.....	11
1.3 Стратегии поведения на рынке: инвестирование и трейдинг	15
1.4 Фундаментальный анализ.....	18
1.5 Инструменты технического анализа	20
1.6 Свечной анализ. Принципы построения и применения японских свечей	25
1.7 Гипотезы функционирования финансовых рынков	34
2 Постановка задачи создания программного продукта анализа биржевых котировок акций	42
3 Разработка программного обеспечения анализа биржевых котировок акций	44
3.1 Разработка и анализ требований к программному обеспечению.....	44
3.1.1 Функциональные требования	45
3.1.2 Нефункциональные требования	55
3.2 Описание архитектуры программного обеспечения	56
3.3 Реализация программного обеспечения.....	60
3.3.1 Выбор средств разработки программного обеспечения	60
3.3.2 Реализация серверной части программного обеспечения	62
3.3.3 Реализация клиентской части программного обеспечения	63
ЗАКЛЮЧЕНИЕ	67
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	68
ПРИЛОЖЕНИЕ А (обязательное).....	70
Программный код серверной части ПО.....	70
ПРИЛОЖЕНИЕ Б (обязательное)	76

Программный код клиентской части ПО.....	76
ПРИЛОЖЕНИЕ В (обязательное)	103
Демонстрация работы ПО	103

ВВЕДЕНИЕ

В экономических условиях настоящего времени происходит совершенствование методов процесса принятия решений в отрасли биржевой торговли. Развитие инструментов анализа биржевых котировок служит основой для грамотного управления инвестициями и стратегического планирования деятельности во всевозможных отраслях.

Благодаря техническому прогрессу и развитию коммуникаций, торговля ценными бумагами перестала быть доступной только для организаций, предприятий и профессиональных брокеров. На сегодняшний день все больше людей зарабатывают на фондовой бирже, совершая покупку и продажу акций и других ценных бумаг. Трейдеры находятся в поиске надежного источника информации, способной помочь в достижении определенных финансовых целей максимально эффективно и с минимальными потерями.

Среди программных продуктов существуют системы, решающие задачи анализа временных рядов с извлечением лежащей в их основе структуры. Данные программные продукты являются узкоспециализированными, их отличают высокие аппаратные требования, существенные затраты на внедрение, сопровождение и интеграцию с функционирующей информационной структурой банков и инвестиционных компаний, недоступность для частных инвесторов.

Выводы о движении цен, основанные на интуиции и здравом смысле, в отношении поведения рынка часто являются ошибочными. Неумение анализировать биржевые котировки финансовых инструментов может привести к потере капитала как частных лиц, так и крупных предприятий. Из этого следует, что разработка программной системы, позволяющей решать комплекс задач, связанных с анализом биржевых котировок финансовых инструментов, является актуальной прикладной задачей. В этом заключается **актуальность** данной работы.

Целью данной работы является разработка программного обеспечения, позволяющего решать задачи, связанные с анализом биржевых котировок акций.

Для достижения поставленной цели были сформулированы следующие **задачи**:

- изучение основных принципов фундаментального анализа;
- изучение основных принципов технического анализа;
- выбор средств разработки для создания программного обеспечения анализа биржевого рынка;
- разработка функциональных и нефункциональных требований к программному обеспечению;
- разработка архитектуры программного обеспечения;
- реализация программного обеспечения анализа биржевого рынка.

Программное обеспечение анализа биржевых котировок представляет собой программный код, при каждом запуске формирующий веб-страницу, на которой выведены биржевые котировки акций в виде таблицы по заданным параметрам: тикеру, периоду, датам, индикаторам, и обеспечивающий построение графиков. В этом состоит **новизна** работы.

Новое решение позволит обеспечить быстрый и удобный режим доступа заинтересованных в биржевой торговле лиц к информации о биржевых котировках, ценах открытия и закрытия рынка, что позволит составить краткосрочный трендовый прогноз и предугадать направление предстоящих сделок. В этом состоит **практическая значимость** разработки программного обеспечения анализа биржевого рынка.

Основным средством разработки была выбрана интегрированная среда для языка программирования Python – PyCharm, предоставляющая средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживающая веб-разработку на Django – свободном фреймворке для веб-приложений.

1 Обзор подходов к решению задачи анализа биржевых котировок

1.1 Биржевой рынок. Фондовый рынок. Задачи фондовой биржи

Биржа – организационно оформленный регулярно функционирующий рынок, на котором совершается торговля ценными бумагами, оптовая торговля товарами или наем работников [1, стр.66].

Различают публичные и частные биржи. Доступ к публичной бирже предоставляется всем желающим участвовать в биржевой торговле, продавцы и покупатели работают без посредников. Деятельность публичной биржи контролируется государственными органами и регулируется соответствующими законами. Частные биржи организованы в форме акционерных обществ за счет частного капитала. Сотрудники компании управляют работой частных бирж, государство устанавливает правила торгов.

В зависимости от товара различают соответственно фондовые, товарные и валютные биржи. На бирже покупаются и продаются стандартные партии товаров определенного базисного сорта [1, стр.66]. Биржевой рынок имеет регламентированные правила для совершения сделок с различными финансовыми активами. Сделки совершаются брокерами по заявкам клиентов или от своего имени.

Ежедневно по всему миру на фондовых биржах совершаются сделки купли-продажи финансовых инструментов: акций, облигаций, фьючерсов, опционов. Рынок, на котором обращаются ранее выпущенные бумаги, обслуживает фондовая биржа – организованный и регулярно функционирующий рынок по купле-продаже ценных бумаг [1, стр.67]. Биржа организует торги, контролирует участников, регистрирует сделки. Порядок и законность работы бирж контролируется регуляторами – государственными или некоммерческими организациями, специальными комиссиями [3]. Регуляторы лицензируют и фиксируют возможные нарушения, лишают лицензии, выписывают штрафы и устанавливают запреты на занятие определенных должностей [3].

Биржи обеспечивают прозрачность финансового положения компаний для потенциальных инвесторов, допуская к торгам эмитенты, предоставившие

финансовую отчетность и обязующиеся соблюдать определенные правила. Биржи определяют и устанавливают правила для эмитентов самостоятельно [3].

К ведущим фондовым биржам США относят: NYSE (New York Stock Exchange), NASDAQ (National Association of Securities Dealers Automated Quotation), AMEX (American Stock Exchange), NYMEX (New York Mercantile Exchange).

Крупнейшей в мире торговой площадкой является Нью-Йоркская фондовая биржа (NYSE), основанная 17 мая 1792 г. На бирже определяется индекс Доу-Джонса, разработанный в XIX веке, представляющий собой среднюю арифметическую ежедневных котировок акций тридцати крупнейших корпораций США на момент закрытия [1, стр.181].

Автоматизированная система котировок Национальной ассоциации дилеров ценных бумаг (NASDAQ) – фондовая биржа, специализирующаяся на акциях высокотехнологичных компаний, созданная с целью автоматизации торговли ценными бумагами [2, стр.36].

Американская фондовая биржа (AMEX) специализируется на торговле биржевыми фондами, паи которых свободно обращаются на бирже [2, стр.36].

Нью-Йоркская товарная биржа (NYMEX/COMEX) специализируется на торговле фьючерсами на металлы, нефть и энергоносители.

Работу бирж в США регулирует комиссия по ценным бумагам и биржам (The United States Securities and Exchange Commission, SEC), созданная и функционирующая в соответствии с рядом законов, принятых в период с 1934 года по 2002 год.

Торги ценными бумагами в России происходят четырех фондовых биржах: Московской межбанковской валютной бирже (ММВБ), Российской торговой системе (РТС), Санкт-Петербургской валютной бирже (СПВБ), Фондовой бирже «Санкт-Петербург» (ФБ СПб) [2].

Работа бирж в России регулируется Центральным банком [3].

Фондовые биржи участвуют в ценообразовании, организуют торги, гарантируют исполнение сделок, формируют котировальные списки ценных

бумаг, публикуют информацию обо всех сделках. Торги на фондовой бирже происходят в определенное время и по установленным биржей правилам [3].

Физическое лицо не может совершать какие-либо сделки на бирже напрямую без посредника [2]. Посредником является брокер или брокерская компания, имеющая лицензию и торговый терминал, предоставляемый физическому лицу. Брокеры предлагают различные наборы услуг и суммы комиссионных, а также дополнительные возможности. В стандартный пакет брокерских услуг входят следующие действия: проведение текущих операций, перечисление прибыли и дивидендов, предоставление профессиональной помощи и консультаций, проведение курсов для начинающих, организация аналитических обзоров [2].

Биржевые торги проводятся в будние дни за исключением государственных праздников. Торговая сессия на фондовой бирже США длится с 4:00 до 20:00 по местному времени. Регулярный торговый день на фондовой бирже в России длится с 10:00 до 18:45 по московскому стандартному поясному времени [2].

1.2 Инструменты, торгующиеся на бирже

Одним из основных инструментов инвестирования на фондовом рынке являются акции. Данный вид ценных бумаг предоставляет возможность получения дополнительного дохода.

Согласно Федеральному закону «О рынке ценных бумаг», акция – эмиссионная ценная бумага, закрепляющая права ее владельца на получение части прибыли акционерного общества в виде дивидендов, на участие в управлении акционерным обществом и на часть имущества, остающегося после его ликвидации [4].

Среди типов акций, эмитируемых обществом, обыкновенные и привилегированные акции являются наиболее распространенными.

Обыкновенные акции предоставляют право голоса на общем собрании акционеров в соотношении «одна акция – один голос». Владельцы

обыкновенных акций могут участвовать в распределении прибыли акционерного общества после пополнения резервов и выплат дивидендов держателям привилегированных акций. По обыкновенным акциям выплата дивидендов не гарантируется [6].

Привилегированные акции, в свою очередь, не дают возможность участвовать в голосовании акционеров, однако их владельцы имеют право первоочередного получения дивидендов по фиксированной ставке вне зависимости от уровня прибыли акционерного общества в определенный период [7]. А также владельцы обладают преимущественным правом на возврат акционерного капитала в случае ликвидации предприятия [1, стр.18].

Ценная бумага не может являться акцией, если она не дает возможности участия в управлении и не предполагает наличия права голоса. Таким образом, именно право на участие в управлении организацией позволяет называть ценную бумагу акцией.

Выпуск привилегированных акций часто ограничен законом, количество акций не может превышать определенной доли уставного капитала. Увеличение количества выпущенных привилегированных акций обеспечит концентрацию управления капиталами многих участников рынка в немногочисленном слое, что противоречит идеям акционерного общества об объединении капиталов и коллективном управлении.

Акционер может не использовать личное право на участие в управлении организацией по некоторым субъективным причинам, но существуем возможность делегировать полномочия другому акционеру или доверенному лицу. Акционерное общество не имеет возможности нормально функционировать без привлечения участников со стороны своих акционеров для решения актуальных вопросов. На фондовой бирже юридические лица привлекают средства для развития и функционирования компании, частные лица инвестируют для получения дохода. Инвестирование в акции необходимо и инвестору, и компании.

Акции принято относить к группе бессрочных ценных бумаг, так как при эмиссии не устанавливается срок их существования. На практике сроки существования акции определяются акционерным обществом.

Существуют обязательные для любой акции реквизиты, основные среди которых: наименование ценной бумаги, наименование и юридический адрес акционерного общества, порядковый номер, вид акции, номинальная стоимость, размер уставного капитала акционерного общества, количество акций в данной эмиссии, имя владельца в случае именной акции, сведения о дивидендах (сроки и способы выплат), сведения о порядке регистрации для именных акций, подписи и печать эмитента [7]. Эмитентом является юридическое лицо или органы исполнительной власти либо местного самоуправления, несущие от своего имени обязательства перед владельцами ценных бумаг по осуществлению прав, закрепленных ими [1, стр.857].

Основными параметрами акции являются: ликвидность – способность быстро купить или продать по рыночной цене, цена и доходность. Доходность акции состоит из выплат дивидендов и роста стоимости курса [5]. Доходность на фондовом рынке может быть выше доходности по депозитным выплатам держателям ценных бумаг, но она не является гарантированной, так как инвестиции в ценные бумаги не застрахованы государством [3].

На практике номинальная стоимость акции, отражающая размер уставного капитала организации и используемая в бухгалтерском учете, не совпадает с ценой акции на фондовом рынке. Рыночная цена акции – стоимость, установленная котировкой на фондовой бирже, определяемая соотношением спроса и предложения. Котировка цен – определение стоимости ценных бумаг во время биржевых торгов [3].

Помимо акций на фондовой бирже торгуются облигации, фьючерсы, опционы.

Облигация – эмиссионная ценная бумага, закрепляющая право ее владельца на получение от эмитента в предусмотренный срок номинальной стоимости бумаги или иного имущественного эквивалента [4]. Облигация может

предусматривать получение владельцем установленных в ней процентов. Для данного вида ценных бумаг заранее предусматривается и указывается срок погашения, а также возможные условия и обстоятельства, при которых владелец не получает номинальную стоимость облигации. Облигация является инструментом кредитования – компании или государственные учреждения выпускают облигации с выплатой владельцам определенного процента в обмен на использование денег инвесторов [2].

Облигации являются более надежным торговым инструментом, так как в случае банкротства компании владельцы облигаций получают компенсацию раньше владельцев акций, однако акции представляют больший интерес для трейдинга ввиду более быстрого получения дохода.

Для торговли на фондовой бирже помимо акций и облигаций используются производные финансовые инструменты – деривативы. Деривативы – это высоколиквидные инструменты с переменной стоимостью, доходность которых основывается на базовых активах, в роли которых выступают товары и финансовые инструменты [2]. При торговле финансовыми производными, покупателям и продавцам на бирже не обязательно владеть базовым активом. Самыми распространенными типами производных финансовых инструментов являются опционы и фьючерсы. Трейдеры совершают сделки купли-продажи, предугадывая направление изменения ценовой тенденции исходных активов.

Фьючерс – контракт между двумя сторонами, одна из которых обязуется купить, а другая – продать актив по фиксированной цене в определенное время в будущем [2]. Фьючерсные контракты являются одним из самых волатильных инструментов, цены на контракты динамичны. Торговля фьючерсными контрактами базируется на ценах базовых активов. Трейдеры зарабатывают на ценовых изменениях базовых товаров, на которые заключаются контракты, являющиеся объектом торговли.

Опционом является финансовый инструмент, который дает покупателю право купить или продать определенный актив по фиксированной цене к

установленному сроку. Активом может быть товар, процентная ставка, индекс и т.п. По сравнению с фьючерсами опционы подвергают трейдера меньшему риску, поскольку не предусматривают обязательного выполнения условий контракта.

Существует ряд рисков, с которыми трейдер или инвестор может столкнуться при торговле на бирже. Трейдеры часто сталкиваются с тремя категориями рисков: рыночными, инвестиционными, торговыми [2]. Рыночный риск – это вероятность неблагоприятного изменения стоимости активов вследствие изменения рыночных факторов [15]. Среди рыночных рисков можно выделить: риск инфляции – рост затрат при росте инфляции, риск ликвидности – возможность вклада средств в неликвидные активы, риск курсовой разницы – вероятность возникновения несоответствий в переводе валют при торговле на зарубежной бирже. Среди инвестиционных рисков можно выделить: риск неиспользования благоприятных возможностей при выборе определенного финансового инструмента, риск концентрации средств на одном активе. Среди торговых рисков можно выделить: риск сокращения – несоответствия цен покупателей и продавцов, риск неэффективного исполнения приказа брокером, риск скачка цены (гэпа) [2].

1.3 Стратегии поведения на рынке: инвестирование и трейдинг

В купле-продаже акций требуется четкая формулировка целей, проработка различных вариантов их достижения, планирование стратегии и тактики, реализация поставленных задач, оценка результатов и исправление ошибок на основе приобретенного опыта [2].

Для выбора стратегии поведения на рынке – инвестирования или трейдинга необходимо иметь представление о целях торговли для инвестора и трейдера, о принципах их работы, знать возможности инструментария каждой стратегии.

Инвестирование – процесс поддержания (в случае пассивных инвестиций) или увеличения (в случае активных инвестиций) ценности организации

[1, стр.217]. Трейдинг можно охарактеризовать как анализ текущей ситуации на рынке и заключение торговых сделок. Инвесторы покупают ценные бумаги и хранят их длительное время с целью улучшения инвестиционного портфеля. Трейдеры покупают ценные бумаги с целью их продажи по более выгодной цене.

Инвестор занимается вложением капитала на долгое время, сделки совершаются довольно редко, после проведения детального фундаментального анализа, в процессе которого изучаются фундаментальные показатели компании, ценные бумаги которой наиболее привлекательны, и финансовые отчеты за большие периоды. Трейдер совершает множество сделок в течение одного торгового дня на фондовом рынке, зарабатывает на скачках рынка, на его волатильности – изменчивости, используя технический анализ.

Инвестор продает актив, когда компания не имеет потенциала роста, или при возможности извлечь максимальную прибыль. Трейдер фиксирует прибыль после каждой сделки, без интереса к долгосрочной перспективе компании.

Для совершения сделок купли-продажи финансовых активов трейдер должен обратиться к брокеру и оставить заявку (приказ, ордер) на покупку или продажу соответственно. Большинство заявок размещаются на бирже как внутридневные – реализуемые в течение торгового дня, но существует возможность размещать заявки на более долгий срок. К основным типам приказов относятся: рыночные, лимитные, «стоп», «стоп-лимит» [2].

Рыночный приказ дает брокеру запрос о покупке или продаже финансового актива по текущей рыночной цене без каких-либо дополнительных условий. На практике заявки не исполняются моментально, так как в ситуациях активной торговли они поступают с опозданием. Таким образом, заявка может исполниться по более высокой цене, что приведет к непредвиденным убыткам. К преимуществам рыночных приказов относится их гарантированное исполнение при первом удобном случае. К недостаткам рыночных приказов можно отнести невозможность прогнозирования всех рисков.

Лимитный приказ выставляется на определенную сумму покупки, данный вид удобен при торговле от определенного ценового уровня. Преимущество

лимитных заявок в возможности точно просчитать риски, так как точка входа определена заранее. Также, лимитная заявка помогает избежать проскальзывания – разницы цены отправленной заявки и цены исполнения заявки.

Стоп-заявка сообщает брокеру намерение трейдера купить или продать актив строго по указанной цене. При размещении стоп-заявки на покупку актива, указывается цена выше текущей рыночной, при которой трейдер отказывается от покупки. При размещении стоп-заявки на продажу актива, указывается цена ниже текущей рыночной, при которой трейдер фиксирует прибыль в случае падения цены. При активной торговле заявка может оказаться не исполненной, но установка предельных цен на покупку и продажу нивелирует риск купли-продажи по непривлекательной цене.

Заявка стоп-лимит является комбинацией лимитной заявки и стоп-заявки, при достижении указанной цены стоп-заявка становится лимитной. Существует риск неисполнения заявки стоп-лимит при значительных колебаниях цен и наличии быстро меняющихся тенденций.

Трейдер заинтересован в механике рынка, характеризующей закономерности изменения цены. Среди видов механики рынка различают: тренд, коррекцию, разворот, пробой уровня и отбой.

Трендом называется устойчивое однонаправленное движение рыночной цены, действующее в течение определенного времени. График цен может содержать различное количество тенденций, но трендом является преобладающая модель движения [12].

Различают бычий (повышательный) и медвежий (понижательный) тренды. Бычий рынок характеризуется большим наличием заявок на покупку активов, растущим курсом цен. Медвежий рынок характеризуется падением ценового курса и преобладанием на рынке продавцов. Быками называют торговцев ценными бумагами, пытающихся извлечь прибыль за счет роста стоимости активов. Медведями называют торговцев, пытающихся заработать на падении

рыночных цен. Если преобладающая модель движения не установлена, рынок не имеет тренда, находится в стагнации или консолидации – боковике [12].

Тренд характеризуется наличием хотя бы двух пиков и двух впадин, развивается относительно плавно на всем промежутке движения, под воздействием коррекционных движений, которые могут принимать форму откатов. Если волнообразная модель нарушается, то велика вероятность смены тренда [12]. Тренд начинается и заканчивается мощным импульсом. Ситуация начала движения цены после импульса называется пробоем уровня. Восходящий тренд сменяется боковым, затем переходит в нисходящий, и наоборот. В соответствии с теорией Чарльза Доу, тренды подразделяются на краткосрочные (несколько недель), среднесрочные (от месяца до полугода) и долгосрочные (более 6 месяцев) [12].

По причине больших скоплений заявок на покупку и продажу активов, на биржевом рынке формируются ценовые уровни, являющиеся линиями поддержки или сопротивления, демонстрирующие направление движения рынка.

Выявлять новые тренды в изменении цен акций и моменты завершения существующих трендов помогает технический анализ [2]. Также, технический анализ позволяет оценивать устойчивость тренда и обнаруживать вторичные точки входа в позиции и сигналы для краткосрочных операций [2].

1.4 Фундаментальный анализ

Термин «фундаментальный анализ» используется для обозначения методов прогнозирования цен фондовых инструментов, валюты и биржевых товаров. Главная задача фундаментального анализа – формировать и предугадывать новые тренды в динамике цен [1, стр.683]. Фундаментальный анализ является основой для стратегических инвесторов, которые осуществляют долгосрочное инвестирование, не заостряя внимание на краткосрочном колебании цен.

Анализ проводится на трех различных уровнях: на уровне экономики, на уровне отрасли, на уровне компании [9].

Фундаментальный анализ начинается с изучения и оценки условий на рынке и экономической ситуации в целом. При этом учитываются изменения ключевых экономических индикаторов: валового национального продукта (ВНП), валового внутреннего продукта (ВВП), уровня занятости, индекса потребительских цен (ИПЦ) и т.д. Изменение отдельных показателей влечет за собой разнонаправленное движение цен на различных рынках.

На отраслевом уровне инвесторы выявляют сегмент для вложений средств путем сравнения отраслей. Существует несколько факторов, определяющих эффективность отрасли: жизненный цикл отрасли и стадия ее развития, тип отраслевого рынка, порог входа на рынок, емкость рынка, потенциал его роста, степень насыщения, степень регулирования отрасли, цикличность отрасли, тенденции в поведении потребителя. Так же учитываются данные технического анализа, влияние политических решений, природные факторы и климатические показатели [9].

Ключевой частью фундаментального анализа является изучение финансово-хозяйственного положения компании эмитента. На данном уровне проводится анализ финансовой ситуации в компании, ее позиционирования на рынке, стратегии развития, изучение финансовых и бухгалтерских отчетов, исследование стратегии управления компанией и ведения бизнеса. Анализу подвергаются следующие финансовые показатели компании: выручка, EBITDA (Earnings Before Interests, Taxes, Depreciation and Amortization) – прибыль компании до вычета процентов, налога, физического или морального износа и амортизации [2], чистая прибыль, чистая стоимость компании, обязательства, денежный поток, величина выплачиваемых дивидендов и производственные показатели компании [8].

Для инвестора важно отслеживать объем денежной массы в обращении, темпы инфляции и реальный объем товаров и услуг.

Фундаментальный анализ – экспертный метод оценки инвестиционной стоимости ценных бумаг: акций, облигаций, производных ценных бумаг и т.д.

1.5 Инструменты технического анализа

Технический анализ – один из трех главных методов анализа рыночных цен на финансовых и товарных рынках наряду с фундаментальным анализом и интуитивным подходом к анализу [1, стр.683].

Технический анализ основывается на нескольких предположениях. Во-первых, цены учитывают все знания, желания и действия участников рынка, любой экономической и политической ценообразующий фактор, и отражают их в своей динамике. Поэтому трейдеру важно уметь абстрагироваться от новостного фона, фундаментальных показателей, и принимать во внимание прошлую динамику рыночных цен. Во-вторых, движение цен подчинено тенденциям, которые можно выявить и использовать в торговле. В-третьих, рынок цикличен – правила и закономерности, установленные в прошлом, актуальны для настоящего и будущего [11].

Технический анализ применяется на высоколиквидных свободных рынках – биржах, включает совокупность инструментов прогнозирования динамики цен на основе закономерностей изменения цен в прошлом.

Технический анализ имеет некоторые преимущества перед фундаментальным анализом. Он более легок и быстр в использовании, подходит для большего числа финансовых активов. В отличие от фундаментального анализа, технический анализ учитывает психологию трейдеров, которая влияет на движение цен наравне с финансовыми показателями, рыночными коэффициентами, экономическими прогнозами и т.д. Технический анализ помогает побороть эмоциональное напряжение и страх проигрыша, и при заключении сделок придерживаться дисциплины. Однако в анализе цен на долгосрочную перспективу технических анализ проигрывает фундаментальному.

Инструменты технического анализа помогают сформировать адекватный взгляд на ситуацию на рынке, принять объективное решение, определить риски и образовать рациональный подход к сделкам.

Главным предметом изучения в техническом анализе является цена. Люди часто принимают решения, опираясь на уровень цен в прошлом, так их решения влияют на ценообразование, и в то же время цены влияют на действия людей. Анализ поведения цены помогает определить соотношение спроса и предложения.

Одним из мощнейших инструментов технического анализа является метод «японские свечи», разработанный торговцами рисом в XVIII – XIX веках в Японии. Интенсивное развитие методов технического анализа приходится на XX век, начало положили публикации статей Чарльза Доу о рынках ценных бумаг в конце XIX века.

Методы, применяемые для анализа цен финансовых инструментов на фондовом рынке, совершенствуются с развитием компьютерных технологий, но некие базовые принципы остаются неизменными. Среди основных методов технического анализа выявляют: классический анализ, индикаторный анализ, волновой анализ, анализ объемов, анализ биржевого стакана и свечной анализ.

Классический анализ определяет тенденции изменения цены с помощью отражения линий и фигур на ценовом графике. На графиках отображают ценовые уровни последовательно изменяющихся минимумов и максимумов, соединенных линиями. Линии образуют восходящие и нисходящие тренды, периоды консолидации – боковики, и показывают направление изменения цены. Линии образуют фигуры технического анализа, интерпретация положения и характера образования которых предсказывает тенденцию изменения цен.

Индикаторный анализ определяет тенденции изменения цены с помощью определенных математических моделей – индикаторов и осцилляторов. Трендовые индикаторы включают модели, которые показывают направление сглаженной от хаотичных ценовых колебаний тенденции. Индикаторы подтверждают наличие или отсутствие выраженной тенденции,

идентифицируют ее образование и завершение действия. Индикаторы являются вторичными величинами, созданными на основе первичных – графиков цен. Индикаторы являются запаздывающими сигналами, поэтому применяются в комбинации с другими инструментами технического анализа.

Среди трендовых индикаторов различают: скользящие средние (Moving Average), индикатор среднего направленного движения (Average Directional Movement Index, ADMI/ADX), параболическую систему времени/цены (Parabolic Stop and Reverse, Parabolic SAR), полосы Боллинджера (Bollinger Bands), индикатор Ишимоку (Ichimoku), Аллигатор (Alligator) [14] и другие.

Самыми часто используемыми индикаторами являются скользящие средние и индекс среднего направления движения. Скользящие средние отражают среднее значение цен за торговый день или другой определенный период времени, обозначают тенденцию движения рынка, но не предсказывают. Индекс среднего направления движения (ADMI/ADX) определяют направленность и силу тренда. Вычисление данного индикатора производится на основе скользящей средней расширения диапазона цен за определенный период времени. ADMI эффективнее работает после периода консолидации.

Параболическая система времени/цены используется для определения ключевой точки, в которой тренд меняет направление. Наиболее эффективно индикатор работает на торговых инструментах с чередованием спокойных направленных движений с узкими консолидациями. Полосы Боллинджера показывают на графике направление и диапазон ценовых колебаний с учетом существующего тренда и характерной для данной фазы рынка волатильности. Индикатор используется для более точной идентификации фигур технического анализа. Индикатор Ишимоку используется при оценивании направления тренда и для выделения областей сопротивления и поддержки. Аллигатор представляет собой комбинацию трех сглаженных скользящих средних, смещенных на определенное количество периодов. Данный индикатор лучше работает в тренде и используется для определения направления, в котором требуется заключать сделки, а также в качестве уровней стоп-заявок.

Трендовые индикаторы используются в качестве фильтра.

К осцилляторам относятся математические методы определения перекупленных (чрезмерно завышенных), перепроданных (чрезвычайно заниженных) цен относительно предыдущих ценовых диапазонов. Среди осцилляторов выделяют: индикатор силы медведей/быков (Bears/Bulls power), тройную экспоненциально сглаженную скользящую среднюю (Triple Exponential Moving Average, TRIX), индикатор накопления/распределения Вильямса (Williams' Accumulation/Distribution, Williams' A/D) [14] и другие.

Индикатор силы медведей/быков помогает предугадать вероятную смену тенденций на рынке, демонстрируя расстановку сил «медведей», зарабатывающих на падении рынка, и «быков», зарабатывающих на росте стоимости финансовых активов. Тройная экспоненциально сглаженная скользящая средняя используется для фильтрации случайных ценовых колебаний, искажающих график. Индикатор наиболее эффективен на трендовых рынках. Индикатор накопления/распределения Вильямса представляет собой сумму восходящих и нисходящих движений цены. Индикатор подтверждает тенденцию, снижение и рост цен происходит совместно с индикатором.

Помимо данных осцилляторов различают осцилляторы объема, осцилляторы волатильности и осцилляторы импульса.

К осцилляторам объема относят: индекс накопления/распределения (Accumulation/Distribution, A/D), осциллятор Чайкина, индекс денежных потоков (Money Flow index, MFI), осциллятор объема (Volume Oscillator) [14] и другие.

Индекс накопления/распределения используется для выявления дисбаланса продавцов и покупателей. Осциллятор Чайкина демонстрирует ускорение изменения объема торгов. Индекс денежных потоков демонстрирует интенсивность вложений средств в ценную бумагу или интенсивность вывода средств из нее. Осциллятор объема отслеживает преобладающую тенденцию изменения объема торгов.

Среди осцилляторов волатильности выделяют: средний истинный диапазон (Average True Range, ATR), индекс товарного канала (Commodity Channel Index, CCI) [14] и другие.

Средний истинный диапазон используется для демонстрации волатильности. Индекс товарного канала используется для определения отклонения цены финансового инструмента от его средней цены за определенный временной период.

К осцилляторам импульса относят: индикатор ускорения/замедления (Acceleration/Deceleration, AC), индекс относительной силы (Relative Strength Index, RSI), индикатор схождения/расхождения скользящих средних (Moving Average Convergence/Divergence, MACD), индикатор скорости рынка (Momentum), темп изменения цены (Rate of Change, ROC), стохастический осциллятор [14] и другие.

Самыми часто используемыми осцилляторами являются индекс относительной силы (RSI), индикатор MACD и стохастический осциллятор.

RSI определяет силу тренда и вероятность его изменения, дает самый сильный сигнал технического анализа – дивергенцию (расхождение) и конвергенцию (схождение). MACD упрощает визуальное восприятие сигналов, снижает их запаздывание. Стохастический осциллятор демонстрирует движущую силу и отражает давление на рынок, оказываемое покупателями и продавцами. Индикатор ускорения/замедления используется для определения момента изменения тренда Momentum и ROC измеряют трендовую составляющую.

Осцилляторы применяются для более точного определения оптимальной цены.

Волновой анализ фондовых рынков назван в честь Ральфа Эллиотта, выделившего чередование рыночных колебаний, принимающих форму волн. Волны подразделяются на коррекционные и движущие. Эллиотт установил, что большинство ценовых тенденций включают модель, состоящую из пяти волн – трех движущих и двух коррекционных [12].

Анализ объемов изучает объемы сделок, совершенных с активом, которые побудили цену к изменениям. Данный метод технического анализа позволяет оценить интенсивность торговли в определенный временной период, а также показывает интерес к определенному финансовому активу.

Анализ биржевого стакана – таблицы лимитных заявок, отображающей заявки на продажу (аски) и на покупку акций (биды), подразумевает отслеживание заявок во время торгов, и принятие решений о сделке [12].

Свечной анализ основан на наблюдении за комбинациями японских свечей, выстраивающихся в определенные модели.

Методы и приемы технического анализа универсальны и успешно используются трейдерами по всему миру [12].

1.6 Свечной анализ. Принципы построения и применения японских свечей

Основой технического анализа является наблюдение за графиками цен – чартами и биржевыми стаканами.

Существует несколько разновидностей биржевых графиков: тиковый, линейный, график баров, свечной график, кластерный график (футпринт) и графики, не зависящие от времени.

Графики строятся в прямоугольной системе координат, по оси абсцисс откладывается время, по оси ординат – цена, объем, валютный курс и т.п.

Свечные графики эффективны в понимании конъюнктуры биржевого рынка, а также позволяют более точно прогнозировать движение цен. Свечные графики применимы к любым финансовым инструментам на фондовом, срочном, валютном и товарном рынках, используются инвесторами, спекулянтами, хеджерами. Анализ свечных графиков используется и как самостоятельный метод прогнозирования, и как дополнение к другим известным методам технического анализа.

К свечному анализу проявляется высокий интерес по многим причинам. Среди них основные:

- Японские свечи являются универсальным методом анализа, подходящим как для начинающих, так и для профессионалов.
- Свечной анализ совершенствовался на протяжении нескольких веков.
- Терминология свечного анализа необычна и довольно легка для запоминания.
- Возможность использовать свечной анализ наряду с другими средствами технического анализа максимизирует шансы успешного инвестирования, трейдинга.

Графики строятся по разным временным интервалам – таймфреймам. Различают внутрдневные, дневные, недельные, месячные таймфреймы. Принципы построения свечей и свечных моделей не зависят от таймфрейма. Существует множество разновидностей свечей и свечных моделей, множество способов их формирования и интерпретации. В свечном анализе нет четких и общепринятых определений и строгих правил, отчасти это объясняется особым менталитетом японцев, отчасти тем, что технический анализ не является наукой.

Графический анализ субъективен, несмотря на применение в анализе некоторых базовых принципов распознавания различных рыночных сигналов. Каждый определяет фигуры и интерпретирует графические сигналы по-разному, и окончательное решение остается за человеком.

Многие системы технического анализа опираются на цены закрытия, например, системы, построенные на базе скользящих средних. Таким образом, в конце биржевых сессий можно наблюдать торговую активность. Для формирования свечи в свечном анализе так же необходимо ожидать цену закрытия. Данный факт можно считать недостатком свечного анализа.

Недостаткам анализа японских свечей также приписывают невозможность определения цели движения цены без использования других инструментов технического анализа: уровней поддержки и сопротивления, уровней коррекции Фибоначчи и других. Свечи дают возможность получить множество сигналов,

но не позволяют составлять абсолютно точные прогнозы. Исходя из этого, свечной анализ более эффективен при его комбинации с другими техническими инструментами.

Для формирования свечей и построения свечного графика необходимо знать цены открытия (Open, O) и закрытия (Close, C), минимальную (Low, L) и максимальную (High, H) цены торговых сессий за определенный период [10, стр.25].

Широкая часть свечи – тело (real body), демонстрирует охват ценового диапазона между ценой открытия и ценой закрытия. Цены открытия (O) и закрытия (C) торгов влияют на цвет свечи. В классическом свечном анализе для окраса свечей используются черный и белый цвета. Если цена закрытия торгов (C) ниже цены открытия (O), то тело свечи будет заполненным – черным. Такая свеча называется медвежьей. Если цена закрытия торгов (C) выше цены открытия (O), то тело свечи будет пустым – белым. Такая свеча называется бычьей.

Линии над и под телом свечи – тени (shadows), демонстрируют ценовые экстремумы. Максимальной цене (H) соответствует высшая точка верхней тени (upper shadow), минимальной цене (L) соответствует низшая точка нижней тени (lower shadow). Когда у тела отсутствует верхняя тень – у свечи срезана вершина (shaven head), если отсутствует нижняя тень – у свечи срезано основание (shaven bottom). Считается, что важная информация о движении цены заключается в теле свечи, а тени рассматриваются как менее значимые дополнительные детали [10, стр.27].

На рисунке 1 представлена схема свечей.

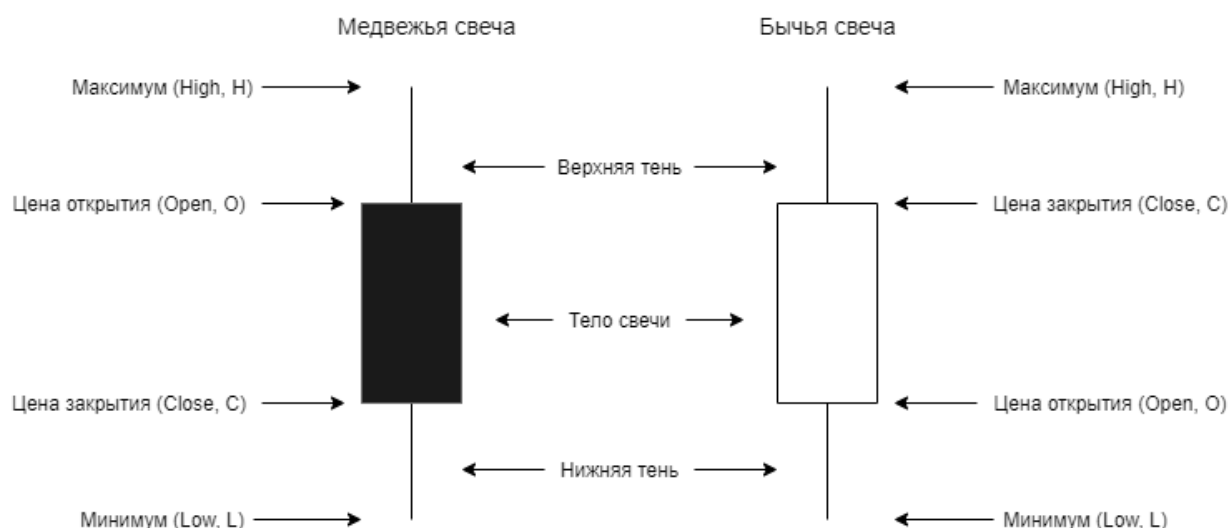


Рисунок 1 – Схема медвежьей и бычьей свечей в классическом свечном анализе

Большинство торговых платформ используют красный и зеленый цвета свечей вместо черного и белого цветов. Данное отображение представлено на рисунке 2.

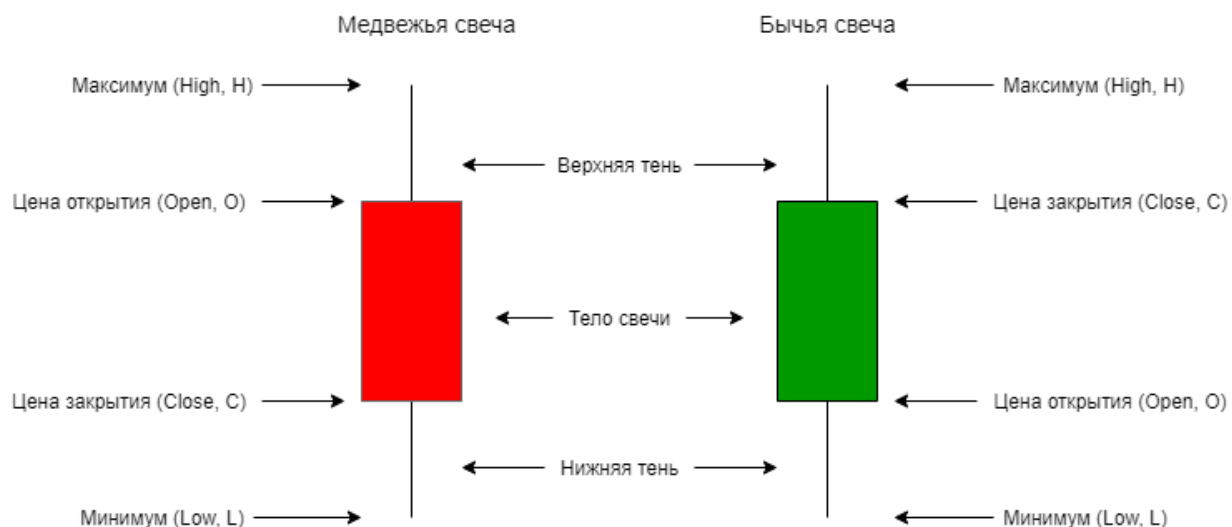


Рисунок 2 – Схема медвежьей и бычьей свечей современных торговых терминалов

Длинное тело свечи демонстрирует колебание цен в широком диапазоне. Для медвежьего периода характерно открытие торгов близко к максимальной цене (H), а закрытие – близко к минимальной цене (L). Для бычьего периода

характерно открытие вблизи минимальной цены (L), а закрытие – вблизи максимальной (H).

Разновидность свечей «марубозу» состоит из длинного тела, не имеет либо верхних теней, либо нижних теней, либо не имеет теней совсем. Черный марубозу демонстрирует господство продавцов (медведей) на рынке, цена открытия (O) совпадает с максимальной ценой (H), цена закрытия (C) совпадает с минимальной ценой (L). Белый марубозу демонстрирует преобладание на рынке покупателей (быков), цена открытия (O) равна минимальной цене (L), цена закрытия (C) – максимальной цене (H). У марубозу закрытия отсутствует тень со стороны цены закрытия, преобладающей над минимумом (бычья) или максимумом (медвежья) свечи. У марубозу открытия отсутствует тень со стороны открытия, что демонстрирует отсутствие движений против основного движения цены за время формирования свечи.

На рисунке 3 представлены схемы свечей марубозу.



Рисунок 3 – Схема свечей марубозу в классическом свечном анализе

Помимо свечей с длинным телом существуют звезды – свечи с маленьким телом, образующие ценовой разрыв с предыдущей свечой, и волчки (spinning tops) – свечи с небольшим размером тела, которое может быть и красным (черным), и зеленым (белым). Зонтики – свечи с небольшим телом и длинной нижней тенью. Перевернутые зонтики – свечи с небольшим телом и длинной

верхней тенью. Волчок, зонтик и перевернутый зонтик демонстрируют рынок в равновесии и могут предсказывать резкое движение цены.

Также, существуют свечи без тела – дожи (doji), при этом длина теней может быть любой, а тело заменяет горизонтальная черта. Дожи формируется, когда цены открытия и закрытия торговой сессии совпадают или очень близки. Различают несколько видов дожи: длинноногий дожи, стрекоза, надгробие, дожи четырех цен. Дожи предупреждают об изменении тенденции.

На рисунке 4 представлены схемы различных видов звезд, волчков, зонтиков и дожи.

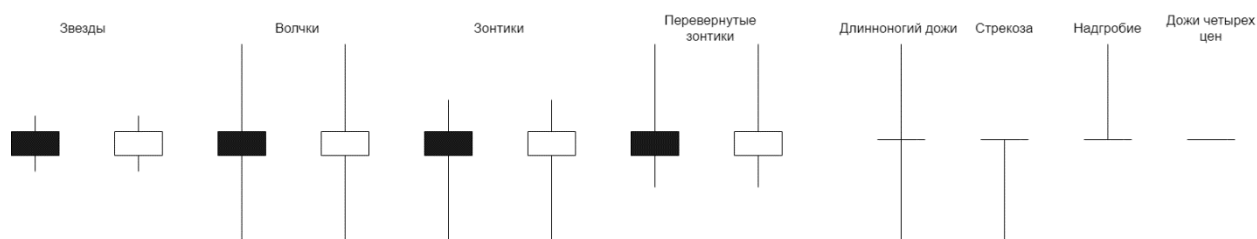


Рисунок 4 – Схема различных односвечных моделей в классическом свечном анализе

Особое значение в свечном анализе придается ценам открытия и закрытия, так как период открытия рынка дает направление дальнейшему движению цен, а период закрытия подтверждает или опровергает формирование той или иной модели на графике. Цена открытия формируется под воздействием вечерних новостей предыдущего дня и формирующихся на рынке тенденций.

В свечном анализе существуют термины «утренняя атака» и «ночная атака». Если в момент всплеска торговой активности в начале дня, более значимые участники рынка сдвигают цены в ту или иную сторону, выставляя большие заявки на продажу или покупку активов, то происходит утренняя атака. Если в момент закрытия торговой сессии либо чуть раньше участники торгов выставляют большие заявки на продажу или покупку активов, то происходит ночная атака.

Разнообразные графические модели свечного анализа носят яркие и запоминающиеся названия, позволяющие легко определять, кто лидирует на рынке в данный момент – продавцы (медведи) или покупатели (быки), какой является та или иная модель – медвежьей или бычьей.

Среди моделей свечного анализа выделяют модели разворота (reversal patterns) и модели продолжения тенденции (continuation patterns).

Модели разворота дают сигнал о возможном прекращении текущей тенденции, но не гарантируют изменение тренда. Обычно разворот тренда происходит поэтапно, по мере изменения настроений участников биржевых торгов. Умение распознавать модели разворота на графиках важно для понимания складывающейся на рынке ситуации, это помогает трейдеру в нужный момент скорректировать торговую стратегию.

К моделям разворота относятся: молот и повешенный (Hammer and Hanging Man), модель поглощения (Engulfing Pattern), завеса из темных облаков (Dark-Cloud Cover), просвет в облаках (Piercing Pattern), харами (Harami Pattern) и многие другие [10, стр.5].

«Молот» и «повешенный» – отдельно взятые свечи, предупреждающие о смене рыночной тенденции. Могут иметь как медвежий, так и бычий характер, в зависимости от тенденции и момента возникновения на графике. При нисходящем тренде свеча будет называться «молотом», при восходящем тренде – «повешенным». Свеча характеризуется наличием длинной нижней тени, телом, короче тени в два раза, короткой верхней тенью или ее отсутствием. Тело свечи должно находиться в верхней части ценового диапазона [10, стр.32].

Модель поглощения является одним из ключевых сигналов разворота, состоит из двух свечей, как правило, окрашенных в разные цвета. Для бычьей модели поглощения характерно появление бычьей свечи, перекрывающей предыдущую медвежью свечу, во время нисходящего тренда. Медвежья модель поглощения характеризуется появлением медвежьей свечи с более длинным телом после бычьей свечи в период восходящего тренда.

Завеса из темных облаков возникает по окончании восходящего тренда или у верхней границы коридора (боковика) и состоит из двух свечей – бычьей и медвежьей последовательно. Медвежья свеча должна перекрывать бычью. Модель имеет медвежий характер.

Аналогичная модель с бычьим характером – просвет в облаках. Данная модель сигнализирует о развороте в основании, появляется на нисходящем тренде, состоит из двух свечей – медвежьей и бычьей последовательно. Чем большая часть тела медвежьей свечи перекрыта бычьей свечой, тем выше вероятность возникновения разворота.

Существует группа моделей разворота, в которые входит звезда – свеча с небольшим телом, образующая гэп с предыдущей свечой. Появление звезды, особенно звезды дожи – свечи без тела, дает сигнал о возможном развороте тренда. Звезды входят в состав следующих моделей разворота: утренняя звезда (Morning Star), вечерняя звезда (Evening Star), звезды дожи (Doji Stars), падающая звезда (Shooting Star). Тело звезды может быть как бычьей, так и медвежьей.

Модель «утренняя звезда» дает сигнал о повышении цен, является сигналом разворота в основании. Состоит из медвежьей свечи с длинным телом, за которой с гэпом вниз возникает свеча с небольшим телом, следом за данной комбинацией образуется бычья свеча с телом, перекрывающим значительную часть тела медвежьей свечи [10, стр.57].

Модель «вечерняя звезда» служит сигналом разворота на вершине, формируется, как правило, после восходящего тренда. Первая свеча данной модели – бычья, имеет длинное тело, следом формируется свеча с коротким телом, далее – медвежья свеча с телом, перекрывающим значительную часть тела бычьей свечи.

Звезды дожи – модель, возникающая на восходящей тенденции, состоящая из свечи дожи с гэпом вверх по отношению к телу предыдущей свечи. Либо модель возникает на нисходящей тенденции и состоит из свечи дожи, формирующейся с гэпом вниз по отношению к телу предыдущей свечи. Сигнал об изменении тенденции должен подтвердиться появлением медвежьей свечи с

длинным телом после свечи дожи, которое перекроет значительную часть первой бычьей свечи на восходящей тенденции. На нисходящей тенденции третья свеча должна быть бычьей с длинным телом, и перекрыть значительную часть тела первой медвежьей свечи.

Модель «падающая звезда» предупреждает о вероятном достижении вершины, но является довольно слабым сигналом разворота. Модель состоит из двух свечей, вторая свеча имеет короткое тело и длинную верхнюю тень.

Модель «харамии» состоит из свечи с коротким телом – волчка, находящегося в границах более длинного тела предыдущей свечи. Харамии предупреждает о завершении тренда, возможной паузе рынка, но не является сильно значимым сигналом разворота.

К моделям продолжения тенденции относятся: гэпы тасуки вверх и вниз (Upward- and Downward-Gap Tasuki), гэпы на максимумах и минимумах (High-Price and Low-Price Gapping Plays), три ступени (Rising and Falling Three Methods) и многие другие [10, стр.6].

Гэпы тасуки вверх возникают при восходящем тренде. После гэпа формируется бычья свеча, следом за которой возникает медвежья с ценой открытия на границе тела бычьей свечи, и с ценой закрытия ниже границ ее тела. Тела свечей не должны сильно отличаться размерами. Гэп тасуки вверх служит сигналом к покупке финансовых активов. Гэп тасуки вниз формируется при нисходящем тренде и является обратной моделью гэпа тасуки вверх, и, соответственно дает противоположный сигнал.

Гэп на максимумах – модель, при которой сессия открывается гэпом вверх, формируется после тенденции роста в течение нескольких предыдущих торговых сессий и последующей консолидации.

Гэп на минимумах образуется во время консолидации в области низких цен после их резкого снижения.

У модели «три ступени» существует бычий и медвежий вариант, оба являются сигналами продолжения тенденции. В состав модели «бычья три ступени» входят: длинная бычья свеча, несколько свечей с коротким телом,

лежащих в границах диапазона первой свечи, бычья свеча с длинным телом, выше нижней границы тела предыдущей свечи и выше верхней границы тела первой свечи.

В свечном анализе нет четких правил, только рекомендации, эмпирически сформированные в результате наблюдения за поведением рынка. Данный вид технического анализа субъективен, сигналы рынка по-разному интерпретируются. Значимость сигналов японских свечей возрастает при их подтверждении с помощью других технических индикаторов.

1.7 Гипотезы функционирования финансовых рынков

Закономерности функционирования рынка акций и изменения цен на активы представляют широкий теоретический и практический интерес для инвесторов и ученых.

Существует несколько подходов к пониманию функционирования финансовых рынков. Среди них основные: гипотеза эффективного рынка, гипотеза фрактального рынка, гипотеза когерентного рынка.

Работа Луи Башелье «Теория спекуляции», опубликованная в 1900 году, является одной из первых работ по изучению финансовых рынков. В данной работе описаны методы, используемые для анализа азартных игр, в применении к котировкам акций. Л. Башелье предполагал, что прибыли – идентично распределенные независимые величины. Данное предположение – гипотеза о рынке, являющаяся основой линейной парадигмы функционирования рынка. На основе парадигмы построены научно признанные концепции функционирования рынка капитала и сформирована гипотеза эффективного рынка.

В 1960-х годах с помощью электронно-вычислительных машин было выявлено отсутствие каких-либо закономерностей в движении цен на акции.

В 1964 году было сформировано утверждение о случайном блуждании акций. Изменения котировок цен (доходности) независимы, являются случайными. Так как изменения цен случайны, ожидается нормальное

распределение, наличие устойчивого среднего значения (математического ожидания) и конечной дисперсии (разброса доходностей относительно математического ожидания) доходностей. Данное утверждение является следствием закона больших чисел: выборка независимых идентично распределенных случайных переменных будет нормально распределенной, если она достаточно велика [15, стр.6].

Американский экономист Юджин Фама формализовал гипотезу эффективного рынка (Efficient Market Hypothesis, EMH) в работах, опубликованных в период с 1965 по 1970 годы.

Гипотеза эффективного рынка выдвигает предположение о том, что цены на акции, существующие в данный момент, получены в результате анализа всей информации, имеющейся в настоящий момент. Нельзя найти недооцененные акции компании, цены справедливы и в равновесном состоянии. Соответственно, изменить состояние на рынке может единственный фактор – поступившая на рынок новая информация, которая становится доступной к использованию для всех участников рынка.

Гипотеза эффективного рынка критикует фундаментальный и технический подходы к анализу рынка.

Начиная с 1970-х годов EMH является фундаментальной гипотезой теории функционирования рынка капитала.

Классическая гипотеза эффективных рынков содержит допущения:

1. Рациональные инвесторы. Инвесторы имеют одинаковую информацию, одинаковые ожидания доходности и риска, принимают рациональные решения. Доходность (математическое ожидание), риск (стандартное отклонение) являются главными и устойчивыми во времени характеристиками любого актива.

2. Эффективный рынок. Вся информация уже учтена в ценах акций, рынки эффективны. Для очень коротких временных зависимостей изменения в ценах акций автокоррелированы, движения цен случайны. Фундаментальные аналитики общим соглашением определяют стоимость активов. При

поступлении на рынок новой информации цены моментально линейно изменяются. Доходность на рынке имеет нормальное распределение [15, стр.7].

3. Случайные блуждания. Исходя из двух предыдущих допущений, доходность является случайной величиной, следовательно, прибыль распределена по нормальному закону распределения.

Допущения гипотезы эффективных рынков открыли возможность развития статистического анализа фондовых рынков [15, стр.7]. На основе гипотезы было создано множество концепций, например, портфельная теория Г. Марковица и модель прогнозирования доходности У. Шарпа [15, стр.7].

На протяжении времени становления гипотезы эффективного рынка возникали аномалии, ставящие под сомнение выводы, которые были сделаны в рамках ЕМН, а также основанные на гипотезе линейные концепции анализа финансового рынка.

М. Осборн сделал вывод, что доходность акций на фондовом рынке описывается «приблизительно нормальным» распределением, так как график имел эксцесс и «тяжелые хвосты» – распределение с бесконечной дисперсией. Данная аномалия указывает на то, что доходность от торговли на фондовом рынке в определенные временные периоды имеет большее отклонение от средних значений, чем предсказывает закон больших чисел. Следовательно, инвестор получит слишком большую прибыль и слишком большой убыток при низкой вероятности их появления. В исследованиях Ю. Фамы и У. Шарпа так же подтверждается наличие у распределения доходностей акций «тяжелых хвостов». У. Шарп утверждал о чрезвычайно малой вероятности критических значений согласно нормальному закону распределения, в то время, как на практике критические значения нередки.

Помимо данной аномалии была выявлена серия других аномалий, противоречащая гипотезе эффективного рынка. Показатели социальных, экономических и природных систем, следовательно, и временные ряды цен акций не соответствуют нормальному закону распределения. Методы

статистического анализа рынка: корреляция, математическое ожидание, дисперсия, могут приводить к неточным результатам.

Фондовые рынки не всегда подчиняются гипотезе эффективного рынка, так как: инвесторы и трейдеры интерпретируют доступную им информацию по-разному, не принимают рациональных решений даже при известном соотношении риска и доходности, не принимают решений моментально при поступлении новой информации, зависят от предыдущей динамики цены активов.

Среди аномалий фондового рынка выделяют аномалии, не объяснимые с точки зрения гипотезы эффективного рынка. Календарные аномалии – эффекты, влияющие на динамику цены в зависимости от времени года, месяца или дня. К данным аномалиям относятся следующие эффекты: эффект января, эффект начала и конца года, эффект дня недели, эффект выходного дня и другие. Существуют определенные погодные аномалии, встречающимся на фондовом рынке, например, доходность в солнечные дни выше, чем в пасмурные. Выделяют так же эффект новолуния, эффект размера компании, эффект новостей, эффект высокой дивидендной доходности, эффекты технического анализа, эффект «память рынка» и другие.

Эффект памяти рынка отражает влияние прошлых данных рынка на текущий курс активов, повторение или имитация характера движения цены, наблюдаемого ранее. Понятие памяти рынка объединяет свойственные трейдерам и инвесторам предрассудки, что рынок накапливает информацию о движении цен и разворотов [16]. Так как тенденцию в основном формируют люди, то движение цены в настоящем времени выстраивается согласно действиям трейдеров, подчиненных влиянию прошлых движений цен. Трейдеры надеются на повторение прошлых ситуаций и действуют согласно сложившимся правилам совершения сделок. Рынок «помнит» определенные принципы механики рынка – структуры продолжения тренда или моделей разворота, так как повторяется существующее ограниченное количество сценариев борьбы продавцов и покупателей [16].

В рамках расчета эффекта «памяти рынка» используют автокорреляционную функцию (АКФ). Автокорреляционная функция определяет корреляционную связь между данными ценового ряда, позволяет выявить определенные свойства: тренд, цикличность, сезонность. АКФ состоит из корреляционной зависимости между значениями исходного ценового ряда и значениями ряда, сдвинутыми на определенное время запаздывания – лаг. Величина лага зависит от первоначального ценового ряда. Например, чтобы построить АКФ для цен закрытия на конец каждой недели за год, величина лага должна быть равной одной неделе.

Автокорреляционная функция отображает степень влияния прошлых рыночных цен акций на текущие цены. Чем глубже используются ценовые данные, тем менее заметно влияние, оказываемое на текущую цену акций. Соответственно, память рынка – это глубина влияния (лаг) и сила влияния (значение корреляции) предыдущей цены на текущую.

АКФ так же характеризует ценовую динамику акции. Для анализа применяются следующие правила:

1. Ценовой ряд являются трендовым, если первый член автокорреляционной функции максимален.
2. Ценовой ряд содержит циклические колебания, если самым большим оказывается коэффициент k -го порядка.
3. Ценовой ряд содержит сильную нелинейную тенденцию или это случайный ряд, если ни один из коэффициентов не является значимым.

Построение автокорреляции облегчает принятие торгового решения для инвестора, отбрасывая менее существенные данные.

Существование аномалий на финансовых рынках, демонстрирующих недостатки теории эффективного рынка, невозможность при любых обстоятельствах объяснить динамику цен с помощью линейных подходов к анализу, объясняет формирование альтернативных гипотез функционирования рынка.

Гипотеза фрактального рынка (Fractal Market Hypothesis, FMH) возникла под воздействием новых взглядов на функционирование многих природных и социальных процессов, как теория хаоса и фрактальная геометрия.

Бенуа Мандельброт обобщил работы ученых, исследующих многоструктурные геометрические объекты, разработав фрактальную геометрию. Под фракталом Б. Мандельброт понимал обширный класс объектов, выходящих за рамки евклидовой геометрии. Суть подхода к описанию многоструктурных объектов с помощью фрактальной геометрии можно сформулировать так: движение многих естественных процессов, которое с первого взгляда может казаться абсолютно случайным, не является случайным в глобальном масштабе [15]. Фрактальная структура формируется при наличии зависимости значения переменной от ее прошлых значений.

Б. Мандельброт обнаружил симметричность долгосрочных и краткосрочных колебаний цен на хлопок, исследуя статистику в компании IBM.

Гипотеза фрактального рынка объясняет эффективность методов технического анализа финансовых рынков, теории волн Р. Эллиота, уровней Фибоначчи. Определенные закономерности и некоторые повторяющиеся модели динамики цен не противоречат гипотезе фрактального рынка, основанной на предположениях:

1. Рынок стабилен при наличии инвесторов с различным горизонтом инвестирования. Если горизонт инвестирования однороден, появляются «тяжелые хвосты» в распределении доходностей.

2. Цены подчинены техническим факторам в краткосрочной перспективе, в долгосрочной перспективе – фундаментальным факторам.

3. Цена включает долгосрочную фундаментальную и краткосрочную техническую оценку. Трейдеры создают и анализируют краткосрочную тенденцию, отражающую настроение масс, в то время как долгосрочная тенденция отображает экономическую ситуацию компании.

Поведение финансовых рынков характеризуется наличием высокого пика и «тяжелых хвостов», что соответствует распределению Парето. Временной ряд,

описываемый распределением Парето, имеет выраженные тренды, циклы и развороты. Дисперсия распределения бесконечна или является неопределенной.

Фрактальный анализ отделяет нормальное распределение от распределения Парето. Временные ряды фрактальной структуры имеют особую структуру, характеризующуюся фрактальной размерностью. Размерность определяется различными методами, среди которых основанный на вычислении показателя Херста (H) с помощью R/S анализа – статистического метода анализа временных рядов. R/S анализ позволяет определить наличие неперiodических циклов. Многие процессы следуют тренду с шумом – смещенному случайному блужданию. Показатель Херста определяет случайность или не случайность исследуемого временного ряда, определяет наличие тренда, наличие «памяти рынка».

Показатель Херста принимает значения в диапазоне от 0 до 1 в целом. Если показатель находится в диапазоне от 0 до 0,5, то исследуемый ряд является антиперсистентным, что значит смену тенденции, разворот. Антиперсистентный ряд называют «розовым шумом». Если показатель Херста равен 0,5, то исследуемый временной ряд является стохастическим, его называют «белым шумом». Если показатель находится в диапазоне от 0,5 до 1, то исследуемый ряд является персистентным, что называется «черным шумом», что значит наличие тренда, направленности у ряда. Для финансовых рынков характерны персистентные ряды.

Инвесторы используют показатель Херста для получения информации о показателях финансового актива, что позволяет отбирать наиболее эффективные акции.

Показатель Херста рассчитывается по следующей формуле:

$$\sigma(T) = \sigma(\tau) * (T/\tau)^H, \quad (1)$$

где σ – стандартное отклонение доходности акций;

T – временной период;

τ – базовый временной период;

H – показатель Херста.

Показатель Херста применяем к любым временным рядам, включая ряды с неизвестными распределениями, что делает его универсальным инструментом анализа акций.

Гипотеза когерентного рынка (Coherent Market Hypothesis) утверждает, что цены рынка можно прогнозировать в определенный временной период. Данная гипотеза строится на создании нелинейных статистических моделей.

В 1990 году Т. Веге на основе теории социальной имитации создал первую нелинейную статистическую модель. Теория социальной имитации заключалась в утверждении, что поведение индивидов в социальных группах подобно движению молекул в металлическом бруске. Подверженный воздействию электромагнитного поля металлический брусок поляризован, при прекращении воздействия медленно возвращается к неполяризованному состоянию. При некоторых условиях индивиды ведут себя независимо, при других индивидуальность стирается, все действия подчинены инстинкту толпы.

Т. Веге предполагал, что между доходностью ценных бумаг и рыночной поляризацией существует некоторая связь [15, стр.26]. Рыночная доходность сопоставлялась с распределением вероятностей поляризации: она могла колебаться от нулевого уровня до высокой разницы в доходах участников рынка [15, стр.26].

Основой гипотезы служит идея: вероятностное распределение доходности на финансовых рынках основывается на фундаментальных окружающих условиях и массовых настроениях инвесторов [15, стр.26]. Состояние рынка не статично, так как существуют различные комбинации двух данных факторов.

Рынок находится в четырех различных фазах [15, стр.26]:

Фаза 1: Эффективный рынок. Доходность финансовых инструментов – случайный временной ряд, не поддающийся прогнозированию. Цены изменяются случайным образом. Инвесторы действуют независимо друг от друга. Распределение вероятностей прибыли соответствует нормальному распределению.

Фаза 2: Переходное состояние рынка к режиму толпы. Настроения индивидов и ожидания инвесторов смещаются в сторону «сознания толпы». Волатильность рынка увеличивается.

Фаза 3: Хаотичный рынок. Финансовые инструменты обладают «долгосрочной памятью». Фундаментальные факторы не оказывают влияния на доходность финансовых рынков. Настроение участников рынка провоцирует сложные движения цен. Существует вероятность скачкообразных изменений цен, смены тенденции.

Фаза 4: Когерентные рынки. Для данной фазы характерно влияние фундаментальных факторов совместно с «сознанием толпы», порождающим явно направленные движения цены.

Гипотеза когерентного рынка выявляет наличие нескольких фаз на финансовых рынках.

На практике необходимо знать, положительны, отрицательны или нейтральны факторы фундаментальных окружающих условий и массовых настроений инвесторов.

Недостатком нелинейных концепций функционирования финансовых рынков является невозможность построения четких рекомендаций и подходов к инвестированию. В отличии от гипотез фрактального и когерентного рынков, гипотеза эффективного рынка привела к созданию четких математических моделей прогнозирования риска и доходности.

2 Постановка задачи создания программного продукта анализа биржевых котировок акций

Целью данной работы является создание программного обеспечения анализа биржевых котировок акций с использованием возможностей интернета для передачи данных.

Задачами данного программного обеспечения являются:

- повышение точности, оперативности получения достоверных данных о котировках;

- снижение времени на изучение биржевых котировок акций;
- снижение времени на изучение финансовых графиков за желаемый период;
- повышение организационно-технического уровня проведения анализа.

Для достижения поставленной цели определены следующие задачи:

- разработка функциональных и нефункциональных требований к программному обеспечению;
- разработка архитектуры программного обеспечения;
- реализация программного обеспечения автоматизированной системы анализа биржевых котировок акций.

Программное обеспечение должно обеспечивать реализацию следующих функций:

- а) информационных функций:
 - 1) формирование на мониторе веб-страницы с возможностью выбора параметров отображение котировок акций;
 - 2) формирование на веб-странице информации в виде таблиц и графиков с учетом заданных параметров.
- б) управляющих функций:
 - 1) изменение параметров отображения данных котировок акций.

Основным средством разработки была выбрана интегрированная среда для языков программирования Python и JavaScript – PyCharm. Предоставляет средства для анализа кода, графический отладчик и поддерживает веб-разработку на Django [17].

Django – свободный фреймворк для веб-приложений на языке Python. Сайт на Django строится из одного или нескольких приложений [18]. В отличие от других фреймворков, обработчики URL в Django конфигурируются при помощи регулярных выражений [18]. Среди возможностей Django выделяют: встроенный интерфейс администратора с переводами на многие языки; расширяемая система

шаблонов с тегами и наследованием; подключаемая архитектура приложений, которые можно устанавливать на любые Django-сайты и многие другие [18].

Среди возможностей интегрированной среды разработки PyCharm можно выделить: статический анализ кода, подсветка синтаксиса и ошибок; отображение файловой структуры проекта, быстрый переход между файлами, классами, методами и использованиями методов; рефакторинг (переименование, извлечение метода, введение переменной, введение константы, подъём и спуск метода и т. д.); встроенный отладчик для Python; поддержка систем контроля версий с поддержкой списков изменений и слияния [17].

Для отображения создаваемой в результате работы программы веб-страницы необходим постоянный доступ в интернет.

3 Разработка программного обеспечения анализа биржевых котировок акций

3.1 Разработка и анализ требований к программному обеспечению

Для создания программного продукта необходимо точно определить цели его разработки, а также задачи, которые программный продукт будет решать. Важнейший этап в разработке программного обеспечения – разработка и анализ требований. Данный этап выявляет общую концепцию разработки программы, ее функциональные возможности и ограничения [19].

Программные требования можно определить, как свойства программного продукта, представленные в нем надлежащим образом для решения конкретных практических задач [19]. Процесс разработки требований включает в себя их извлечение и документирование. Анализ требований – это их рецензирование, процесс проверки требований на корректность, непротиворечивость, тестируемость, реализуемость. В процессе анализа требования представляются в понятной разработчикам форме. Представление требований в форме UML-моделей позволит наглядно их специфицировать, а также упростит анализ [19].

Для разработки программного продукта выделяют функциональные и нефункциональные требования.

Функциональные требования отражают весь набор функций разрабатываемой системы, которыми она должна быть наделена. Функциональные требования к системе позволяют выявить модель прецедентов. Прецедент описывает последовательность выполняемых системой действий с целью предоставления конкретных результатов пользователю. Составной частью модели прецедентов является диаграмма вариантов использования (use case diagram), которая включает представление всех возможных действующих лиц, ассоциированных с всевозможными вариантами использования. Диаграмма вариантов использования должна полностью описывать функциональность системы на некотором уровне абстракции [19].

Все параметры качества системы – атрибуты качества, описываются нефункциональными требованиями [19]. Атрибутами качества являются характеристики программного продукта, отражающие пользовательские характеристики: доступность, удобство установки, производительность, надежность, условия эксплуатации [19] и т.д. К наиболее важным атрибутам качества относятся: надежность, производительность, удобство использования программного продукта.

3.1.1 Функциональные требования

Далее будут рассмотрены основные варианты использования программной системы для анализа биржевых котировок акций.

Обзор прецедентов

Использование программной системы наглядно отражается с помощью диаграммы вариантов использования, представленной на рисунке 5.

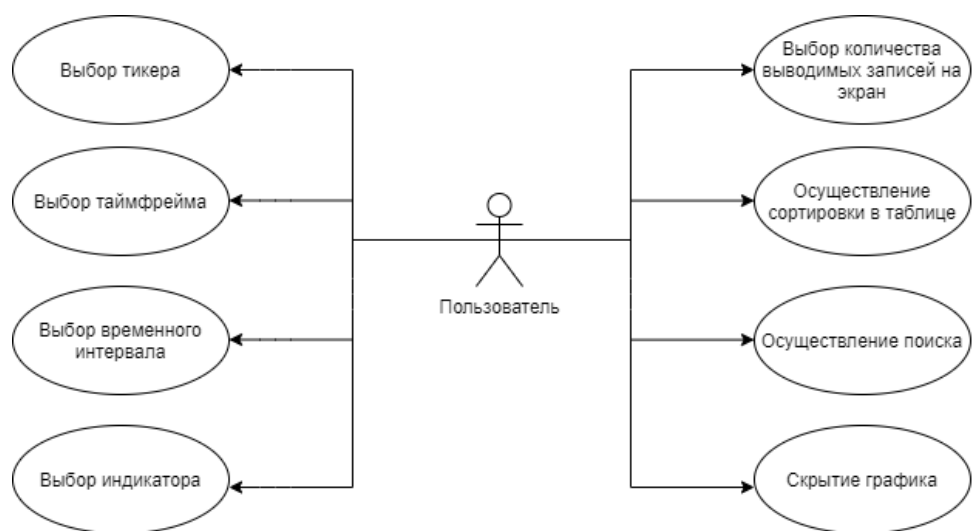


Рисунок 5 – Диаграмма вариантов использования системы

Акторы системы

Единственным актором, определенным в ходе моделирования, является пользователь. В роли пользователя может выступить человек, заинтересованный в данных о котировках акций.

Прецеденты

Основные прецеденты описаны в таблице 1.

Таблица 1 – Реестр вариантов использования

Код	Актор	Наименование прецедента	Формулировка
П1	Пользователь	Выбор тикера	Пользователь выбирает тикер компании из выпадающего списка
П2	Пользователь	Выбор таймфрейма	Пользователь выбирает период для вывода информации и построения графика из выпадающего списка
П3	Пользователь	Выбор временного интервала	Пользователь выбирает даты для вывода информации и построения графика из выпадающего списка
П4	Пользователь	Выбор индикатора	Пользователь выбирает индикатор из выпадающего списка
П5	Пользователь	Выбор количества выводимых записей на экран	Пользователь выбирает количество записей таблицы данных, отображаемых на странице, из выпадающего списка
П6	Пользователь	Осуществление сортировки в таблице	Пользователь осуществляет сортировку записей в таблице по возрастанию или убыванию в каждом столбце
П7	Пользователь	Осуществление поиска	Пользователь осуществляет поиск по данным таблицы
П8	Пользователь	Скрытие графика	Пользователь скрывает график нажатием кнопки

Описание вариантов использования

П1. Выбор тикера

Вариант использования П1 описан в таблице 2.

Таблица 2 – Вариант использования П1

Код	Актор	Наименование прецедента	Формулировка
П1	Пользователь	Выбор тикера	Пользователь выбирает тикер компании из выпадающего списка

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П2, П3, П4

Краткое описание.

Данный вариант использования позволяет пользователю выбрать главный параметр системы – тикер акций компаний Российской Федерации.

Основной поток событий:

Шаг 1. Система предоставляет пользователю форму с выпадающим списком тикеров акций компаний России.

Шаг 2. Пользователь выбирает тикер акций для дальнейшего анализа.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

П2. Выбор таймфрейма

Вариант использования П2 описан в таблице 3.

Таблица 3 – Вариант использования П2

Код	Актор	Наименование прецедента	Формулировка
П2	Пользователь	Выбор таймфрейма	Пользователь выбирает период для вывода информации и построения графика из выпадающего списка

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П1, П3, П4

Краткое описание.

Данный вариант использования позволяет пользователю выбирать один из основных параметров системы – таймфрейм для импорта данных и дальнейшего построения графика.

Основной поток событий:

Шаг 1. Система предоставляет пользователю форму с выпадающим списком таймфреймов.

Шаг 2. Пользователь выбирает таймфрейм для дальнейшего анализа.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

П3. Выбор временного интервала

Вариант использования П3 описан в таблице 4.

Таблица 4 – Вариант использования ПЗ

Код	Актор	Наименование прецедента	Формулировка
ПЗ	Пользователь	Выбор временного интервала	Пользователь выбирает даты для вывода информации и построения графика из выпадающего списка

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П1, П2, П4

Краткое описание.

Данный вариант использования позволяет пользователю выбрать один из основных параметров системы – даты для импорта данных и дальнейшего построения графика.

Основной поток событий:

Шаг 1. Система предоставляет пользователю форму с выпадающим календарем.

Шаг 2. Пользователь выбирает дату, либо временной промежуток для дальнейшего анализа.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

П4. Выбор индикатора

Вариант использования П4 описан в таблице 5.

Таблица 5 – Вариант использования П4

Код	Актор	Наименование прецедента	Формулировка
П4	Пользователь	Выбор индикатора	Пользователь выбирает индикатор из выпадающего списка

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П1, П2, П3

Краткое описание.

Данный вариант использования позволяет пользователю выбирать один из основных параметров системы – индикатор для построения графика и осуществления анализа.

Основной поток событий:

Шаг 1. Система предоставляет пользователю форму с выпадающим списком индикаторов.

Шаг 2. Пользователь выбирает индикатор для дальнейшего анализа.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

П5. Выбор количества выводимых записей на экран

Вариант использования П5 описан в таблице 6.

Таблица 6 – Вариант использования П5

Код	Актор	Наименование прецедента	Формулировка
П5	Пользователь	Выбор количества выводимых записей на экран	Пользователь выбирает количество записей таблицы данных, отображаемых на странице, из выпадающего списка

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П1, П2, П3, П6

Краткое описание.

Данный вариант использования позволяет пользователю выбирать количество выводимых на экран записей таблицы данных.

Основной поток событий:

Шаг 1. Система предоставляет пользователю форму с выпадающим списком количества выводимых на странице записей таблицы.

Шаг 2. Пользователь выбирает количество записей таблицы данных.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

П6. Осуществление сортировки в таблице

Вариант использования П6 описан в таблице 7.

Таблица 7 – Вариант использования П6

Код	Актор	Наименование прецедента	Формулировка
П6	Пользователь	Осуществление сортировки в таблице	Пользователь осуществляет сортировку записей в таблице по возрастанию или убыванию в каждом столбце

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П1, П2, П3, П5

Краткое описание.

Данный вариант использования позволяет пользователю сортировать записи таблицы данных.

Основной поток событий:

Шаг 1. Система предоставляет пользователю таблицу с котировками акций.

Шаг 2. Пользователь выбирает столбец для сортировки цен.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

П7. Осуществление поиска

Вариант использования П7 описан в таблице 8.

Таблица 8 – Вариант использования П7

Код	Актор	Наименование прецедента	Формулировка
П7	Пользователь	Осуществление поиска	Пользователь осуществляет поиск по данным таблицы

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П1, П2, П3, П5, П6

Краткое описание.

Данный вариант использования позволяет пользователю осуществлять поиск по записям таблицы данных.

Основной поток событий:

Шаг 1. Система предоставляет пользователю таблицу с котировками акций и форму для ввода данных и осуществления поиска.

Шаг 2. Пользователь вводит данные и производит поиск по записям таблицы.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

П8. Скрытие графика

Вариант использования П8 описан в таблице 9.

Таблица 9 – Вариант использования П8

Код	Актор	Наименование прецедента	Формулировка
П8	Пользователь	Скрытие графика	Пользователь скрывает график нажатием кнопки

Основное действующее лицо: Пользователь

Другие участники прецедента: отсутствуют

Связи с другими вариантами использования: П1, П2, П3, П4

Краткое описание.

Данный вариант использования позволяет пользователю скрывать график, построенный на основе импортированных данных.

Основной поток событий:

Шаг 1. Система предоставляет пользователю график.

Шаг 2. Пользователь нажимает кнопку и скрывает график.

Альтернативные потоки:

Ошибка вывода данных.

Шаг 1. Система обнаруживает, что произошла ошибка при импорте данных.

Шаг 2. Система выдает сообщение об отсутствии данных.

3.1.2 Нефункциональные требования

Нефункциональные требования не менее важны при разработке программного обеспечения. Они позволяют сделать работу с программным продуктом более комфортным для пользователя.

Требования к надежности

Одним из основных атрибутов качества программного обеспечения является его надежность – способность безотказно работать и сохранять характеристики при соблюдении определенных условий использования.

Среди требований к надежности системы выделяют:

1) Восстановление после сбоев.

При возникновении аппаратных сбоев (при отключении электропитания сети), система должна автоматически восстанавливаться после устранения неполадок и/или причин сбоев. Исключение составляют ситуации, связанные с техническим повреждением носителей информации с исполняемым программным кодом. А также, ситуации, связанные с доступом к информации на стороннем ресурсе.

2) Обработка ошибочных ситуаций.

В системе должны быть предусмотрены средства информирования пользователя об ошибочности его действий.

При возникновении исключений система должна проанализировать и попытаться идентифицировать ошибку. При успешной идентификации ошибки система должна вывести пользователю соответствующее сообщение.

Требования к производительности

Важнейшими аспектами разработки программного продукта являются продуктивность и производительность работы, то есть способность программы выполнять предусмотренные алгоритмом функции быстро и качественно.

Выделяют следующие требования к производительности системы:

1) Время отклика системы должно составлять не более 3 секунд, а время ответа стороннего ресурса на запрос соответствующих данных – не более 5 секунд.

2) Система должна быть способна выводить импортируемые данные за приемлемое время. Исключение составляет случай ошибки импорта данных.

Требования к эргономике и удобству эксплуатации

Пользовательский интерфейс обеспечивает взаимодействие пользователя и программного обеспечения, следовательно, от его характеристик зависит эффективность использования и удовлетворение пользователя от работы с системой.

Выделяются следующие требования к удобству эксплуатации и эргономике системы:

- 1) Интерфейс системы должен быть интуитивно понятен пользователю;
- 2) Сообщения об ошибках должны быть краткими, но информативными.
- 3) Внешнее поведение (отклик на действия пользователя) однотипных элементов интерфейса должны быть одинаково реализованы.

3.2 Описание архитектуры программного обеспечения

Архитектура программы определяет наиболее общее представление о структуре и логике функционирования, указывает основные составляющие и связи между ними, определяет последующую реализацию программного продукта [19]. Для описания архитектуры используется унифицированный язык моделирования (UML) в соответствии с рекомендациями методологии Rational Unified Process [20].

Выделяются следующие компоненты UML-спецификации системы:

- Концептуальная модель, включающая описание основных сценариев работы системы, вариантов использования, участников системы.

- Логическая модель, включающая описание классов и функциональных модулей системы.
- Модель размещения, включающая требования к аппаратному обеспечению пользователя, описание размещения компонентов системы.
- Модель реализации, включающая описание физических артефактов системы: файлы, исполняемые файлы, используемые библиотеки, а также связи между ними.

Концептуальная модель

В рамках концептуальной модели программного продукта выявляются участники системы – акторы, основные сценарии работы системы и варианты использования – прецеденты. Связи акторов и прецедентов представляются на диаграмме вариантов использования. Концептуальная модель программного обеспечения была построена на этапе выявления функциональных требований (см. 3.1.1 Функциональные требования).

Логическая модель

В рамках логической модели программного продукта описываются значимые функциональные модули и классы системы. Модули проектируемой системы представлены в диаграмме классов (см. рисунок 6).



Рисунок 6 – Диаграмма классов

Описание модулей системы приведено в таблице 10.

Таблица 10 – Описание модулей системы

Наименование	Описание
Пользовательский интерфейс	Содержит атрибуты взаимодействия пользователя с программой. Позволяет выбрать параметры данных, реализовать поиск и сортировку.
Импорт данных	Содержит атрибут, выполняющий импорт данных с сайта finam.ru
Транслятор данных	Содержит атрибут, формирующий импортированные данные в таблицу.
Графический модуль	Содержит атрибут, производящий построение графика и индикатора на основе импортированных данных.

Программное обеспечение создает веб-страницу. Котировки акций российских компаний импортируются с ресурса finam.ru. Данные находятся в свободном доступе, информация о ценах открытия, закрытия, максимума и минимума, об объемах продаж предоставляется пользователю с задержкой на 15 минут. Импортированные данные формируются в таблицу, затем производится построение графика и индикатора на графике.

Модель размещения

В рамках модели размещения программного продукта формируется диаграмма развертывания, содержащая описание физического пространства для размещения компонентов. Диаграмма развертывания представлена на рисунке 7.

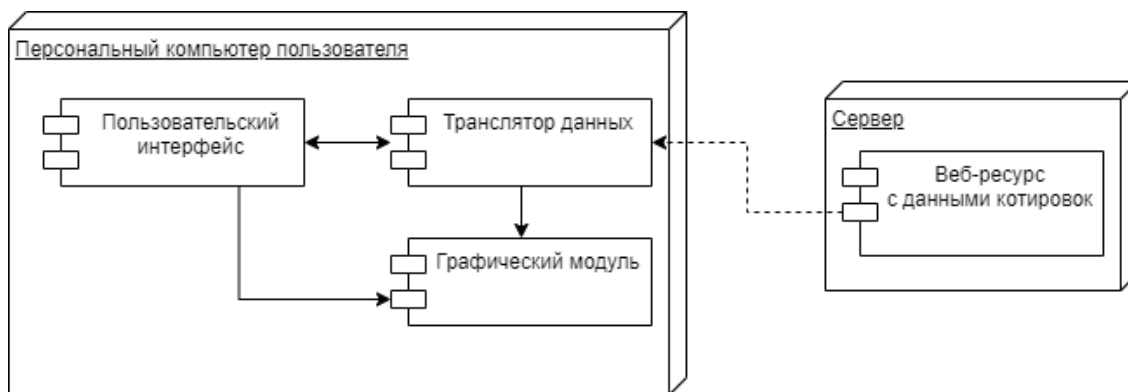


Рисунок 7 – Диаграмма развертывания

Модель реализации

Для непосредственной реализации системы необходимо выделить основные логические и функциональные структуры. Для описания элементов системы используется диаграмма компонентов. Диаграмма компонентов представлена на рисунке 8.

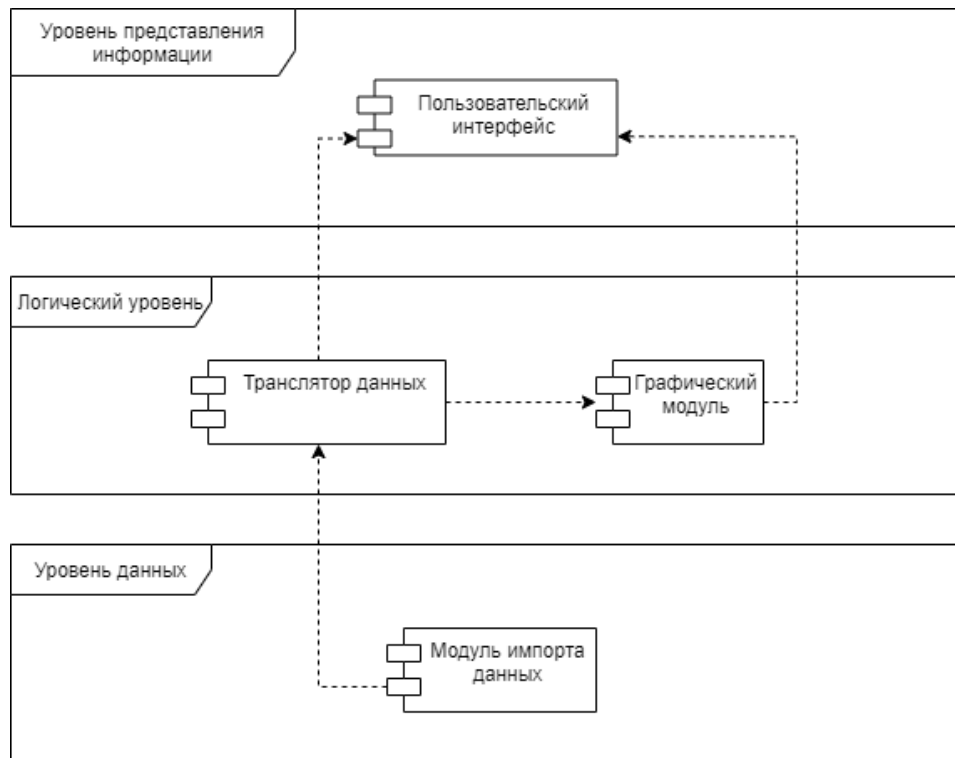


Рисунок 8 – Диаграмма компонентов

В рамках данной работы более подробно будет описана реализация программного обеспечения анализа биржевых котировок акций, модуля импорта данных и веб-приложения.

3.3 Реализация программного обеспечения

3.3.1 Выбор средств разработки программного обеспечения

Разработка программного обеспечения анализа биржевого рынка будет производиться с помощью интегрированной среды для языков программирования Python и JavaScript – PyCharm. Данная среда предоставляет средства для анализа кода, графический отладчик, поддерживает веб-разработку на Django [17].

Django – свободный фреймворк для веб-приложений на языке Python. Веб-сайт на Django строится из одного или нескольких приложений [18]. В отличие от других фреймворков обработчики URL в Django конфигурируются при помощи регулярных выражений [18]. Среди возможностей Django выделяют

[18]: встроенный интерфейс администратора с переводами на многие языки; расширяемая система шаблонов с тегами и наследованием; подключаемая архитектура приложений, которые можно устанавливать на любые Django-сайты и многие другие.

Среди возможностей интегрированной среды разработки PyCharm можно выделить [17]: статический анализ кода, подсветка синтаксиса и ошибок; отображение файловой структуры проекта, быстрый переход между файлами, классами, методами и использованиями методов; рефакторинг (переименование, извлечение метода, введение переменной, введение константы, подъём и спуск метода и т. д.); встроенный отладчик для Python; Поддержка систем контроля версий: общий пользовательский интерфейс для Mercurial, Git, Subversion, Perforce и CVS с поддержкой списков изменений и слияния.

Для отображения создаваемой в результате работы программы веб-страницы необходим постоянный доступ в интернет. Программное обеспечение создает веб-страницу, на которой отображается таблица с котировками акций определенных с определенным таймфреймом, за определенный период времени. По умолчанию страница формируется со следующими параметрами: тикер – SBER компании «Сбербанк», таймфрейм (период) – 1 час, временной период: 10 дней до даты запуска программы не включительно, в данный период попадают выходные дни, когда торговля не совершается, и соответственно, данные за эти дни не выводятся. По выводимым в таблицу данным (тикер компании, таймфрейм, дата, цены открытия и закрытия, максимальные и минимальные цены, объем) строится график японских свечей. На данном графике строятся индикаторы после выбора пользователем соответствующего индикатора из выпадающего списка. Пользователь имеет возможность менять данные непосредственно на веб-странице: выбирать тикер компании, период, даты, индикатор, количество выводимых записей в таблице, производить поиск по записям, сортировать данные в столбцах по возрастанию или убыванию, скрывать график. Данные о котировках компаний берутся с сайта finam.ru.

Веб-приложение состоит из серверной и клиентской частей.

3.3.2 Реализация серверной части программного обеспечения

Программное обеспечение – проект «stock_market» в среде разработки PyCharm, реализовано на языках программирования Python и JavaScript. Серверная часть программного обеспечения состоит из источников данных и кода подключения программы к веб-серверу по локальной сети.

В качестве источника данных используется электронное периодическое издание «ФИНАМ» [21]. Информационно-аналитическое агентство finam.ru является одним из наиболее востребованных финансово-информационных интернет-ресурсов российского сегмента [21]. «ФИНАМ» обеспечивает оперативную подачу биржевых сводок, публикующихся в свободном доступе с задержкой данных на 15 минут, и без задержки данных для зарегистрированных пользователей.

Для формирования URL-запроса (URL – Uniform Resource Locator, Единый указатель ресурса) необходимо подключить ряд библиотек:

```
import math
from datetime import datetime
from urllib.parse import urlencode
from urllib.request import urlopen
import pandas as pd
from django.http import JsonResponse
```

Модуль «math» содержит обширный функционал для работы с числовыми данными. Модуль «datetime» необходим для работы с датами и временем. Модуль «urllib.parse» служит для формирования гипертекстовых ссылок. Функция «urlencode» используется для кодирования строки запроса. Модуль «urllib.request» содержит функции и классы для открытия веб-ресурсов по URL-адресам. Функция «urlopen» возвращает URL-адрес полученного веб-ресурса, метainформацию веб-страницы, код состояния HTML-ответа. «Pandas» является высокоуровневой библиотекой для обработки и анализа данных на языке Python. Модуль «django.http» используется для создания представлений, принимающих веб-запросы и возвращающих веб-ответы. Класс «JsonResponse» возвращает

ответ в текстовом формате обмена данными JSON (JavaScript Object Notation). Для размещения данных библиотек и последующего кода используется файл `views.py`, расположенный в каталоге проекта «`stock_market`».

После подключения библиотек необходимо создать массивы данных для формирования URL-запросов на сайт `finam.ru`.

```
TICKERS = {'ABRD': 82460, 'AESL': 181867, 'AFKS': 19715, 'AFLT': 29,
'AGRO': 399716, 'AKRN': 17564, 'ALBK': 82616, 'ALNU': 81882}
```

Массив «`TICKERS`» содержит тикеры российских компаний, финансовые инструменты которых торгуются на российских фондовых биржах. Выше приведен небольшой фрагмент массива, полный вид которого представлен в Приложении А.

```
PERIODS = {'5min': 3, '10min': 4, '15min': 5, '30min': 6, 'hour': 7,
'daily': 8, 'week': 9, 'month': 10}
```

Массив «`PERIODS`» содержит таймфреймы для группировки котировок при формировании URL-запросов на веб-ресурс `finam.ru`.

Создается структура «`params`», содержащая все необходимые для формирования запроса данные о котировках: тикер акции, начальная и конечная даты (номер дня, месяца и год), таймфрейм, часовой пояс, форматы отображения информации, московское время и многие другие.

По выделенным структурам данных формируется ссылка на сайт `finam.ru`:
`url = f"{FINAM_URL}{ticker}_{start_date_rev}_{end_date_rev}.csv?{params}"`

Затем полученные данные формируются в таблицу, после чего используются для построения графика.

Полный код серверной части программного обеспечения представлен в Приложении А.

3.3.3 Реализация клиентской части программного обеспечения

Часть программного обеспечения, напрямую взаимодействующая с конечным пользователем, называется клиентской. Для создания клиентской

части веб-приложения используется множество разнообразных платформ на различных языках программирования.

Программное обеспечение «stock_market» использует мультипарадигменный язык программирования JavaScript.

Клиентская часть программного обеспечения «stock_market» содержит несколько файлов с расширением .js, основными являются: main.js, chartjs-financial.js, charts.js.

Файл main.js содержит функции формирования пользовательских форм на веб-странице. Формы состоят из выпадающих списков, элементы которых выбирает пользователь для формирования данных, выводящихся на веб-странице.

```
const DATERANGEPICKER_OPTIONS = {
  "maxDate": moment(),
  "locale": {
    "format": 'MMM D, YYYY',
    "separator": " - ",
  },
  ranges: {
    'Yesterday': [moment().subtract(1, 'days'),
moment().subtract(1, 'days')],
    'Last 7 Days': [moment().subtract(6, 'days'), moment()],
    'Last 30 Days': [moment().subtract(29, 'days'), moment()],
    'This Month': [moment().startOf('month'),
moment().endOf('month')],
    'Last Month': [moment().subtract(1, 'month').startOf('month'),
moment().subtract(1, 'month').endOf('month')],
    'This Year': [moment().startOf('year'),
moment().endOf('year')],
    'Last Year': [moment().subtract(1, 'year').startOf('year'),
moment().subtract(1, 'year').endOf('year')],
  }
};
```


Структура «DATERANGEPICKER_OPTIONS» содержит различные варианты для формирования пользователем временных промежутков. На веб-странице формируется календарь, на котором пользователь отмечает отдельные даты или временные промежутки. Помимо календаря используются структуры, формирующие следующие временные промежутки: вчерашний день, последняя неделя, последние 30 дней, текущий месяц, текущий год, предыдущий год. Дата формируется с учетом выходных и праздничных дней, информация о котировках в такие дни отсутствует, так как фондовые биржи проводят торги только в будние дни.

Далее создаются функции, формирующие формы с выпадающими списками данных:

```
function init_tickers()  
function init_periods()  
function load_quotes()
```

Функции фиксируют выбранные пользователем данные, связываются с серверной частью программного обеспечения для формирования нового запроса на веб-ресурс finam.ru.

Функция «init_tickers» фиксирует изменение тикера. Функция «init_periods» фиксирует изменение таймфрейма. Функция «load_quotes» обеспечивает связь с серверной частью программного обеспечения.

Файл `chartjs-financial.js` является лицензированной версией построения графиков японских свечей, расположенной на крупнейшем веб-сервисе для хостинга IT-проектов GitHub [22] в свободном доступе.

Файл `charts.js` содержит все необходимые данные для построения графиков японских свечей по импортируемой с веб-ресурса finam.ru информации. Файл содержит функции для построения графика и индикатора по необходимым данным:

```
function format_data()  
function get_dates()  
function draw_candlestick_chart()
```

```
function draw_oscillator_chart()
```

Функция «format_data» фиксирует дату, установленную пользователем в необходимом для построения графика формате. Функция «get_dates» возвращает даты в формате YYYYMMDD. Функция «draw_candlestick_chart» считывает данные о котировках финансовых инструментов и формирует график японских свечей. Функция «draw_oscillator_chart» формирует индикаторы на построенном ранее графике.

Полный код клиентской части программного обеспечения представлен в Приложении Б. Демонстрация работы программного обеспечения представлена в Приложении В.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы поставленные задачи были полностью решены.

В результате были достигнуты следующие цели:

- изучены основные принципы фундаментального анализа;
- изучены основные принципы технического анализа;
- разработаны функциональные и нефункциональные требования к программному обеспечению;
- разработана архитектура программного обеспечения;
- реализовано программное обеспечения анализа биржевого рынка.

В данной работе реализовано программное обеспечение анализа биржевого рынка, представляющее собой веб-приложение. Программное обеспечение заимствует данные котировок акций с сайта finam.ru, находящиеся в общем доступе.

Программное обеспечение создает веб-страницу, на которой отображается таблица с котировками акций определенных с определенным таймфреймом, за определенный период времени. По выводимым в таблицу данным (тикер компании, таймфрейм, дата, цены открытия и закрытия, максимальные и минимальные цены, объем) строится график японских свечей. На данном графике строятся индикаторы после выбора пользователем соответствующего индикатора из выпадающего списка. Пользователь имеет возможность менять данные непосредственно на веб-странице: выбирать тикер компании, период, даты, индикатор, количество выводимых записей в таблице, производить поиск по записям, сортировать данные в столбцах по возрастанию или убыванию, скрывать график.

Внедрение новых технических решений анализа биржевого рынка. Критерием качества созданного программного обеспечения является эффективная работа системы в автоматическом режиме с минимальным участием человека.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Румянцева Е. Е. Новая экономическая энциклопедия / Е.Е. Румянцева. – 4-е изд. – Москва: ИНФА-М, 2020 – XII, 882 с.
2. Петров К. Н. Как заработать на бирже для чайников / Петров К. В., Лукашевич Т. В. – М.: ООО «И.Д. Вильямс», 2017. – 256 с.
3. Как устроена фондовая биржа и как она работает [Электронный ресурс]. – Режим доступа: <https://journal.tinkoff.ru/guide/stock-exchange/>
4. О рынке ценных бумаг: Федеральный закон от 22.04.1996 № 39-ФЗ (ред. от 27.12.2019, с изм. от 07.04.2020) «О рынке ценных бумаг» (с изм. и доп., вступ. в силу с 28.03.2020)
5. Что такое акции? Все про акции [Электронный ресурс]. – Режим доступа: <https://place.moeх.com/useful/chto-takoe-akcii?list=vse-pro-aktsii>
6. Акции. Основные типы акций [Электронный ресурс]. – Режим доступа: <https://investobserver.info/akcii-osnovnye-tipy-akcij/>
7. Акция [Электронный ресурс]. – Режим доступа: <http://www.grandars.ru/student/finansy/aksiya.html>
8. Фундаментальный_анализ – Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Фундаментальный_анализ
9. 3 уровня фундаментального анализа [Электронный ресурс]. – Режим доступа: <https://mindspace.ru/abcinvest/3-urovnya-fundamentalnogo-analiza/>
10. Нисон С. Японские свечи: Графический анализ финансовых рынков / Стив Нисон ; Пер. с англ. – М.: Интеллектуальная Литература, 2020. – 290 с.
11. Технический_анализ – Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Технический_анализ
12. Введение в технический анализ на фондовом рынке: как читать графики + рабочие методы [Электронный ресурс]. – Режим доступа: <https://greedisgood.one/tehnicheckiy-analiz-fondovogo-rynka>
13. Методы технического анализа [Электронный ресурс]. – Режим доступа: <https://www.opentrainer.ru/articles/metody-tehnicheckogo-analiza/>

14. Все технические индикаторы торговой системы QUIK [Электронный ресурс]. – Режим доступа: <https://bcs-express.ru/novosti-i-analitika/vse-tekhnicheskie-indikatory-torgovoi-sistemy-quick>
15. Жданов И. Ю. Прогнозирование доходности и риска инвестиций на фондовом рынке: учебное пособие / Жданов И. Ю., Жданов В. Ю. – Москва: Проспект, 2020. – 128 с.
16. Память рынка и как ее использовать в торговле [Электронный ресурс]. – Режим доступа: <https://equity.today/pamyat-rynka.html>
17. PyCharm – Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/PyCharm>
18. Django – Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/Django>
19. Бубнов А. А., Бубнов С. А., Майков К. А. Разработка и анализ требований к программному обеспечению: учебник. Москва: Курс, 2018. – 176 с.
20. UML – краткое руководство [Электронный ресурс]. – Режим доступа: <https://coderlessons.com/tutorials/akademicheskii/uchit-uml/uml-kratkoe-rukovodstvo>
21. Финам [Электронный ресурс]. – Режим доступа: <https://www.finam.ru/>
22. GitHub – Википедия [Электронный ресурс]. – Режим доступа: <https://ru.wikipedia.org/wiki/GitHub>

ПРИЛОЖЕНИЕ А

(обязательное)

Программный код серверной части ПО

```
import math
from datetime import datetime
from urllib.parse import urlencode
from urllib.request import urlopen
import pandas as pd
from django.http import JsonResponse
from stockstats import StockDataFrame

TICKERS = {
    'ABRD': 82460, 'AESL': 181867, 'AFKS': 19715, 'AFLT': 29, 'AGRO':
399716, 'AKRN': 17564, 'ALBK': 82616,
    'ALNU': 81882, 'ALRS': 81820, 'AMEZ': 20702, 'APTK': 13855,
'AQUA': 35238, 'ARMD': 19676, 'ARSA': 19915,
    'ASSB': 16452, 'AVAN': 82843, 'AVAZ': 39, 'AVAZP': 40, 'BANE':
81757, 'BANEP': 81758, 'BGDE': 175840,
    'BISV': 35242, 'BISVP': 35243, 'BLNG': 21078, 'BRZL': 81901,
'BSPB': 20066, 'CBOM': 420694, 'CHEP': 20999,
    'CHGZ': 81933, 'CHKZ': 21000, 'CHMF': 16136, 'CHMK': 21001,
'CHZN': 19960, 'CLSB': 16712, 'CLSBP': 16713,
    'CNTL': 21002, 'CNTLP': 81575, 'DASB': 16825, 'DGBZ': 17919,
'DIOD': 35363, 'DIXY': 18564, 'DVEC': 19724,
    'DZRD': 74744, 'DZRDP': 74745, 'ELTZ': 81934, 'ENRU': 16440,
'EPLN': 451471, 'ERCO': 81935, 'FEES': 20509,
    'FESH': 20708, 'FORTP': 82164, 'GAZA': 81997, 'GAZAP': 81998,
'GAZC': 81398, 'GAZP': 16842, 'GAZS': 81399,
    'GAZT': 82115, 'GCHE': 20125, 'GMKN': 795, 'GRAZ': 16610, 'GRNT':
449114, 'GTLC': 152876, 'GTPR': 175842,
    'GTSS': 436120, 'HALS': 17698, 'HIMC': 81939, 'HIMCP': 81940,
'HYDR': 20266, 'IDJT': 388276, 'IDVP': 409486,
```

Продолжение ПРИЛОЖЕНИЯ А

'IGST': 81885, 'IGST03': 81886, 'IGSTP': 81887, 'IRA0': 20516,
'IRGZ': 9, 'IRKT': 15547, 'ISKJ': 17137,
'JNOS': 15722, 'JNOSP': 15723, 'KAZT': 81941, 'KAZTP': 81942,
'KBSB': 19916, 'KBTK': 35285, 'KCHE': 20030,
'KCHP': 20498, 'KGKC': 83261, 'KGKCP': 152350, 'KLSB': 16329,
'KMAZ': 15544, 'KMEZ': 22525, 'KMTZ': 81903,
'KOGK': 20710, 'KRKN': 81891, 'KRKNP': 81892, 'KRKO': 81905,
'KRKOP': 81906, 'KROT': 510, 'KROTP': 511,
'KRSB': 20912, 'KRSBP': 20913, 'KRSG': 15518, 'KSGR': 75094,
'KTSB': 16284, 'KTSBP': 16285, 'KUBE': 522,
'KUNF': 81943, 'KUZB': 83165, 'KZMS': 17359, 'KZOS': 81856,
'KZOSP': 81857, 'LIFE': 74584, 'LKOH': 8,
'LNTA': 385792, 'LNZL': 21004, 'LNZLP': 22094, 'LPSB': 16276,
'LSNG': 31, 'LSNGP': 542, 'LSRG': 19736,
'LVHK': 152517, 'MAGE': 74562, 'MAGEP': 74563, 'MAGN': 16782,
'MERF': 20947, 'MFGS': 30, 'MFGSP': 51,
'MFON': 152516, 'MGNT': 17086, 'MGNZ': 20892, 'MGTS': 12984,
'MGTSP': 12983, 'MGVM': 81829, 'MISB': 16330,
'MISBP': 16331, 'MNFD': 80390, 'MOBB': 82890, 'MOEX': 152798,
'MORI': 81944, 'MOTZ': 21116, 'MRKC': 20235,
'MRKK': 20412, 'MRKP': 20107, 'MRKS': 20346, 'MRKU': 20402,
'MRKV': 20286, 'MRKY': 20681, 'MRKZ': 20309,
'MRSB': 16359, 'MSNG': 6, 'MSRS': 16917, 'MSST': 152676, 'MSTT':
74549, 'MTLR': 21018, 'MTLRP': 80745,
'MTSS': 15523, 'MUGS': 81945, 'MUGSP': 81946, 'MVID': 19737,
'NAUK': 81992, 'NFAZ': 81287, 'NKHP': 450432,
'NKNC': 20100, 'NKNCP': 20101, 'NKSH': 81947, 'NLMK': 17046,
'NMTP': 19629, 'NNSB': 16615, 'NNSBP': 16616,
'NPOF': 81858, 'NSVZ': 81929, 'NVTK': 17370, 'ODVA': 20737,
'OFCB': 80728, 'OGKB': 18684, 'OMSH': 22891,
'OMZZP': 15844, 'OPIN': 20711, 'OSMP': 21006, 'OTCP': 407627,
'PAZA': 81896, 'PHOR': 81114, 'PHST': 19717,

Продолжение ПРИЛОЖЕНИЯ А

'PIKK': 18654, 'PLSM': 81241, 'PLZL': 17123, 'PMSB': 16908,
'PMSBP': 16909, 'POLY': 175924, 'PRFN': 83121,
'PRIM': 17850, 'PRIN': 22806, 'PRMB': 80818, 'PRTK': 35247,
'PSBR': 152320, 'QIWI': 181610, 'RASP': 17713,
'RBCM': 74779, 'RDRB': 181755, 'RGSS': 181934, 'RKKE': 20321,
'RLMN': 152677, 'RLMNP': 388313, 'RNAV': 66644,
'RODNP': 66693, 'ROLO': 181316, 'ROSB': 16866, 'ROSN': 17273,
'ROST': 20637, 'RSTI': 20971, 'RSTIP': 20972,
'RTGZ': 152397, 'RTKM': 7, 'RTKMP': 15, 'RTSB': 16783, 'RTSBP':
16784, 'RUAL': 414279, 'RUALR': 74718,
'RUGR': 66893, 'RUSI': 81786, 'RUSP': 20712, 'RZSB': 16455,
'SAGO': 445, 'SAGOP': 70, 'SARE': 11,
'SAREP': 24, 'SBER': 3, 'SBERP': 23, 'SELG': 81360, 'SELGP':
82610, 'SELL': 21166, 'SIBG': 436091, 'SIBN': 2,
'SKYC': 83122, 'SNGS': 4, 'SNGSP': 13, 'STSB': 20087, 'STSBP':
20088, 'SVAV': 16080, 'SYNG': 19651,
'SZPR': 22401, 'TAER': 80593, 'TANL': 81914, 'TANLP': 81915,
'TASB': 16265, 'TASBP': 16266, 'TATN': 825,
'TATNP': 826, 'TGKA': 18382, 'TGKB': 17597, 'TGKBP': 18189,
'TGKD': 18310, 'TGKDP': 18391, 'TGKN': 18176,
'TGKO': 81899, 'TNSE': 420644, 'TORS': 16797, 'TORSP': 16798,
'TRCN': 74561, 'TRMK': 18441, 'TRNFP': 1012,
'TTLK': 18371, 'TUCH': 74746, 'TUZA': 20716, 'UCSS': 175781,
'UKUZ': 20717, 'UNAC': 22843, 'UNKL': 82493,
'UPRO': 18584, 'URFD': 75124, 'URKA': 19623, 'URKZ': 82611,
'USBN': 81953, 'UTAR': 15522, 'UTII': 81040,
'UTSY': 419504, 'UWGN': 414560, 'VDSB': 16352, 'VGSB': 16456,
'VGSBP': 16457, 'VJGZ': 81954, 'VJGZP': 81955,
'VLHZ': 17257, 'VRAO': 20958, 'VRAOP': 20959, 'VRSB': 16546,
'VRSBP': 16547, 'VSMO': 15965, 'VSYD': 83251,
'VSYDP': 83252, 'VTBR': 19043, 'VTGK': 19632, 'VTRS': 82886,
'VZRZ': 17068, 'VZRZP': 17067, 'WTCM': 19095,

Продолжение ПРИЛОЖЕНИЯ А

```
'WTCMP': 19096, 'YAKG': 81917, 'YKEN': 81766, 'YKENP': 81769,
'YNDX': 388383, 'YRSB': 16342, 'YRSBP': 16343,
'ZHIV': 181674, 'ZILL': 81918, 'ZMZN': 556, 'ZMZN': 603, 'ZVEZ':
82001
}
PERIODS = {
    '5min': 3, '10min': 4,
    '15min': 5, '30min': 6,
    'hour': 7, 'daily': 8,
    'week': 9, 'month': 10,
}
def get_tickers(request):
    return JsonResponse(TICKERS)
def get_periods(request):
    return JsonResponse(PERIODS)
def get_quotes(request):
    ticker = request.GET.get('ticker', 'SBER')
    ticker_code = TICKERS[ticker]
    period = PERIODS[request.GET.get('period', 'hour')]
    FINAM_URL = "http://export.finam.ru/" # сервер
    market = 0
    start = request.GET.get('start_date')
    end = request.GET.get('end_date')
    start_date = datetime.strptime(start, "%Y-%m-%d").date()
    start_date_rev = datetime.strptime(start, "%Y-%m-%d").strftime('%Y%m%d')
    end_date = datetime.strptime(end, "%Y-%m-%d").date()
    end_date_rev = datetime.strptime(end, "%Y-%m-%d").strftime('%Y%m%d')
    params = urlencode([
        ('market', market),
        ('em', ticker_code),
```

Продолжение ПРИЛОЖЕНИЯ А

```
        ('code', ticker),
        ('apply', 0),
        ('df', start_date.day),
        ('mf', start_date.month - 1),
        ('yf', start_date.year),
        ('from', start_date),
        ('dt', end_date.day),
        ('mt', end_date.month - 1),
        ('yt', end_date.year),
        ('to', end_date),
        ('p', period),
        ('f', ticker + "_" + start_date_rev + "_" + end_date_rev),
        ('e', ".csv"),
        ('cn', ticker),
        ('dtf', 1),
        ('tmf', 1),
        ('MSOR', 0),
        ('mstime', "on"),
        ('mstimever', 1),
        ('sep', 1),
        ('sep2', 1),
        ('datf', 1),
        ('at', 1)])

    url =
    f"{FINAM_URL}{ticker}_{start_date_rev}_{end_date_rev}.csv?{params}"
    report = list()
    for row in urlopen(url).readlines():
        report.append(row.strip().decode("utf-8",
errors='ignore').split(','))
    columns = [h.replace('<', '').replace('>', '') for h in
report[0]]
    df = pd.DataFrame(report[1:], columns=columns)
```

Окончание ПРИЛОЖЕНИЯ А

```
df['OPEN'] = pd.to_numeric(df['OPEN'])
df['CLOSE'] = pd.to_numeric(df['CLOSE'])
df['HIGH'] = pd.to_numeric(df['HIGH'])
df['LOW'] = pd.to_numeric(df['LOW'])
df['VOL'] = pd.to_numeric(df['VOL'])
df.rename(columns={"VOL": "VOLUME"})
stock = StockDataFrame.retype(df)
cr = stock['cr']
cr_ma1 = stock['cr-ma1']
cr_ma2 = stock['cr-ma2']
cr_ma3 = stock['cr-ma3']
macd = stock['macd']
rci = stock['rsi_6']
kdjk = stock['kdjk']
return JsonResponse({
    'report': report,
    'cr': [0 if math.isnan(i) else i for i in
cr.values.tolist()],
    'cr_ma1': [0 if math.isnan(i) else i for i in
cr_ma1.values.tolist()],
    'cr_ma2': [0 if math.isnan(i) else i for i in
cr_ma2.values.tolist()],
    'cr_ma3': [0 if math.isnan(i) else i for i in
cr_ma3.values.tolist()],
    'macs': [0 if math.isnan(i) else i for i in
macd.values.tolist()],
    'rci': [0 if math.isnan(i) else i for i in
rci.values.tolist()],
    'kdjk': [0 if math.isnan(i) else i for i in
kdjk.values.tolist()],
})
```

ПРИЛОЖЕНИЕ Б

(обязательное)

Программный код клиентской части ПО

```
const DATERANGEPICKER_OPTIONS = {
  "maxDate": moment(),
  "locale": {
    "format": 'MMMM D, YYYY',
    "separator": " - ",
  },
  ranges: {
    'Yesterday': [moment().subtract(1, 'days'),
moment().subtract(1, 'days')],
    'Last 7 Days': [moment().subtract(6, 'days'), moment()],
    'Last 30 Days': [moment().subtract(29, 'days'), moment()],
    'This Month': [moment().startOf('month'),
moment().endOf('month')],
    'Last Month': [moment().subtract(1, 'month').startOf('month'),
moment().subtract(1, 'month').endOf('month')],
    'This Year': [moment().startOf('year'),
moment().endOf('year')],
    'Last Year': [moment().subtract(1, 'year').startOf('year'),
moment().subtract(1, 'year').endOf('year')],
  }
};

document.addEventListener("DOMContentLoaded", () => {
  google.charts.load('current', {'packages': ['corechart',
'bar']}]);
  let default_start_date = moment().subtract(10,
'days').format('MMMM D, YYYY');
```

Продолжение ПРИЛОЖЕНИЯ Б

```
    let default_end_date = moment().subtract(1, 'days').format('MMMM
D, YYYY');
    let interval_obj = $("#interval");
    interval_obj.val(`${default_start_date} - ${default_end_date}`);
    interval_obj.daterangepicker(DATERANGEPICKER_OPTIONS);
    interval_obj.onchange = () => { load_quotes() };

    $('#oscillators').dropdown({
      'onChange': function () { load_quotes(); }
    }).css("border-radius", "20px");

    document.getElementById('interval').onchange = () => {
      load_quotes();
    };

    init_periods();
    init_tickers();

  });

function init_tickers() {
  axios.get('/quotes/tickers/get').then((response) => {
    const select = document.getElementById('tickers');
    for (let ticker in response.data) {
      let option = document.createElement('option');
      option.value = ticker;
      option.innerHTML = ticker;
      option.selected = (ticker === 'SBER');
      select.appendChild(option);
    }
    $("#tickers").dropdown({
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        onChange: () => {
            load_quotes();
        }
    }).css("border-radius", "20px");
    load_quotes();
});
}

function init_periods() {
    axios.get('/quotes/periods/get').then((response) => {
        const select = document.getElementById('periods');
        for (let period in response.data) {
            let option = document.createElement('option');
            option.value = period;
            option.innerHTML = period;
            option.selected = (period === 'hour');
            select.appendChild(option);
        }
        $("#periods").dropdown({
            onChange: () => {
                load_quotes();
            }
        }).css("border-radius", "20px");
    });
}

function random_ID() {
    return '_' + Math.random().toString(36).substr(2, 9);
}

function load_quotes() {
```

Продолжение ПРИЛОЖЕНИЯ Б

```
let interval = document.getElementById("interval").value.split("
- ");
let start_date = moment(interval[0], 'MMMM D, YYYY');
let end_date = moment(interval[1], 'MMMM D, YYYY');
start_date = start_date.format('YYYY-MM-DD');
end_date = end_date.format('YYYY-MM-DD');

let ticker = $('#tickers').dropdown('get value');
let period = $('#periods').dropdown('get value');

let container = document.getElementById('table_container');
container.classList.add('loading');

axios.get(`/quotes/get?start_date=${start_date}&end_date=${end_date}&ticker=${ticker}&period=${period}`)
    .then((response) => {
        let table_id = random_ID();
        let chart_id = random_ID();
        let chart_container =
document.getElementById('chart_container');
        chart_container.classList.replace('basic',
'placeholder');
        chart_container.innerHTML = `
            <div class="ui icon header">
                <i class="chart icon"></i>
                No data are listed for this tickers.
            </div>
        `;
        container.innerHTML = `<table class="ui celled table"
id="${table_id}"></table>`;
        let report = response.data['report'];
```

Продолжение ПРИЛОЖЕНИЯ Б

```
if (!report.length) return;
let headers = report.shift();
let table = $('#${table_id}`).DataTable({
  responsive: true,
  bAutoWidth: false,
  data: report,
  columns: headers.map((item) => {
    return { 'title': item.replace('<',
''.replace('>', '')}
  })
});
let oscillator_type = $('#oscillators').dropdown('get
value');

chart_container.classList.replace('placeholder',
'basic');

chart_container.innerHTML = `
  <canvas style="width: 100%; height: 500px"
id="${chart_id}"></canvas>
`;
if (oscillator_type !== 'none') {
  chart_container.innerHTML += `
  <div id="oscillator_chart_container"
style="width: 100%; height: 500px; margin-top: -500px"></div>
`;
}
let data = format_data(report);
draw_candlestick_chart(chart_id, data);
if (oscillator_type !== 'none') {
  draw_oscillator_chart([
    get_dates(report),
    response.data[oscillator_type]
```


Продолжение ПРИЛОЖЕНИЯ Б

```
        ]);
    }
    }).finally(() => {
        container.classList.remove('loading');
    });
}
```

```
let CHART_OPTIONS = {
  chartArea: {
    width: '100%', height: '100%',
    left: 0, top: 0,
  },
  backgroundColor: { fill:'transparent' },
  vAxis: {
    textPosition: 'none',
    gridlines: { color: 'transparent' }
  },
  hAxis: {
    textPosition: 'none',
    gridlines: { color: 'transparent' }
  },
  crosshair: {
    trigger: 'both',
    orientation: 'both',
  },
  animation: {
    duration: 1000,
    easing: 'out',
    startup: true
  },
  legend: 'none',
};
```

Продолжение ПРИЛОЖЕНИЯ Б

```
Date.prototype.toYYYYMMDD = function () {
  let dd = this.getDate();
  let mm = this.getMonth() + 1;
  return [
    this.getFullYear(),
    (mm > 9 ? '' : '0') + mm,
    (dd > 9 ? '' : '0') + dd,
  ].join('-');
};
```

```
function format_data(data) {
  let formatted_data = [];
  for (let i in data) {
    let row = data[i];
    let year = row[2].slice(0, 4);
    let month = Number(row[2].slice(4, 6)) - 1;
    let day = row[2].slice(6, 8);
    let hour = row[3].slice(0, 2);
    let minute = row[3].slice(2, 4);
    let second = row[3].slice(4, 6);
    let date = new Date(year, month, day, hour, minute, second);
    formatted_data.push({
      t: luxon.DateTime.fromJSDate(date).valueOf(),
      o: row[4],
      h: row[5],
      l: row[6],
      c: row[7],
    });
  }
  return formatted_data;
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
function get_dates(data) {
    let dates = [];
    for (let i in data) {
        let row = data[i];
        let year = row[2].slice(0, 4);
        let month = Number(row[2].slice(4, 6)) - 1;
        let day = row[2].slice(6, 8);
        dates.push(new Date(year, month, day));
    }
    return dates;
}

function draw_candlestick_chart(container_id, data) {
    let ctx = document.getElementById(container_id).getContext('2d');
    let chart = new Chart(ctx, {
        type: 'candlestick',
        data: {
            datasets: [{
                label: 'Candlestick chart',
                data: data
            }]
        },
        options: {
            scales: {
                xAxes: [{
                    afterBuildTicks: function (scale, ticks) {
                        let majorUnit = scale._majorUnit;
                        let firstTick = ticks[0];
                        let i, ilen, val, tick, currMajor, lastMajor;
                        val = luxon.DateTime.fromMillis(ticks[0].value);
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        firstTick.major = (majorUnit === 'minute' &&
val.second === 0)
        || (majorUnit === 'hour' && val.minute
=== 0)
        || (majorUnit === 'day' && val.hour ===
9)
        || (majorUnit === 'month' && val.day <=
3 && val.weekday === 1)
        || (majorUnit === 'year' && val.month ===
0);
        lastMajor = val.get(majorUnit);

        for (i = 1, ilen = ticks.length; i < ilen;
i++) {
            tick = ticks[i];
            val =
luxon.DateTime.fromMillis(tick.value);
            currMajor = val.get(majorUnit);
            tick.major = currMajor !== lastMajor;
            lastMajor = currMajor;
        }
        return ticks;
    }
}
}
});
}

function draw_oscillator_chart(data) {
    let container =
document.getElementById("oscillator_chart_container");
```

Продолжение ПРИЛОЖЕНИЯ Б

```
    if (!container) return;

    let dataTable = new google.visualization.DataTable();
    dataTable.addColumn({type: 'date', id: 'Date', label: 'Date'});
    dataTable.addColumn({type: 'number', id: 'index', label:
'Index'});
    let rows = [];
    for (let i in data[0]) {
        rows.push([
            data[0][i],
            Number(data[1][i]),
        ]);
    }
    dataTable.addRow(rows);
    let chart = new google.visualization.LineChart(container);
    chart.draw(dataTable, CHART_OPTIONS);
}
/*!
 * @license
 * chartjs-chart-financial
 * http://chartjs.org/
 * Version: 0.1.0
 *
 * Copyright 2020 Chart.js Contributors
 * Released under the MIT license
 *
 * https://github.com/chartjs/chartjs-chart-
financial/blob/master/LICENSE.md
 */
(function (global, factory) {
    typeof exports === 'object' && typeof module !== 'undefined' ?
factory(require('chart.js')) :
```

Продолжение ПРИЛОЖЕНИЯ Б

```
typeof define === 'function' && define.amd ? define(['chart.js'],
factory) :
(global = global || self, factory(global.Chart));
}(this, (function (Chart) { 'use strict';
Chart = Chart && Object.prototype.hasOwnProperty.call(Chart,
'default') ? Chart['default'] : Chart;
```

```
const helpers = Chart.helpers;
const defaultConfig = {
  position: 'left',
  ticks: {
    callback: Chart.Ticks.formatters.linear
  }
};
```

```
const FinancialLinearScale =
Chart.scaleService.getScaleConstructor('linear').extend({
  _parseValue(value) {
    let start, end, min, max;
    if (typeof value.c !== 'undefined') {
      start = +this.getRightValue(value.l);
      end = +this.getRightValue(value.h);
      min = Math.min(start, end);
      max = Math.max(start, end);
    } else {
      value = +this.getRightValue(value.y);
      start = undefined;
      end = value;
      min = value;
      max = value;
    }
  }
});
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        return {
            min,
            max,
            start,
            end
        };
    },
    determineDataLimits() {
        const me = this;
        const chart = me.chart;
        const data = chart.data;
        const datasets = data.datasets;
        const isHorizontal = me.isHorizontal();

        function IDMatches(meta) {
            return isHorizontal ? meta.xAxisID === me.id :
meta.yAxisID === me.id;
        }
        // First Calculate the range
        me.min = null;
        me.max = null;
        helpers.each(datasets, (dataset, datasetIndex) => {
            const meta = chart.getDatasetMeta(datasetIndex);
            if (chart.isDatasetVisible(datasetIndex) &&
IDMatches(meta)) {
                helpers.each(dataset.data, (rawValue, index)
=> {
                    const value = me._parseValue(rawValue);
                    if (isNaN(value.min) || isNaN(value.max)
|| meta.data[index].hidden) {
                        return;
                    }
                })
            }
        })
    }
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        if (me.min === null || value.min < me.min)
    {
        me.min = value.min;
    }

        if (me.max === null || me.max < value.max)
    {
        me.max = value.max;
    }
    });
}
});

    // Add whitespace around bars. Axis shouldn't go exactly
from min to max
    const space = (me.max - me.min) * 0.05;
    me.min -= space;
    me.max += space;
    // Common base implementation to handle ticks.min,
ticks.max, ticks.beginAtZero
    this.handleTickRangeOptions();
}
});

    Chart.scaleService.registerScaleType('financialLinear',
FinancialLinearScale, defaultConfig);
    const helpers$1 = Chart.helpers;
    Chart.defaults.financial = {
        label: '',
        hover: {
            mode: 'label'
        },
    },
```


Продолжение ПРИЛОЖЕНИЯ Б

```
scales: {
  xAxes: [{
    type: 'time',
    distribution: 'series',
    offset: true,
    ticks: {
      major: {
        enabled: true,
        fontStyle: 'bold'
      },
      source: 'data',
      maxRotation: 0,
      autoSkip: true,
      autoSkipPadding: 75,
      sampleSize: 100
    }
  ]},
  yAxes: [{
    type: 'financialLinear'
  }]
},
tooltips: {
  intersect: false,
  mode: 'index',
  callbacks: {
    label(tooltipItem, data) {
      const dataset =
data.datasets[tooltipItem.datasetIndex];
      const point = dataset.data[tooltipItem.index];
      if (!helpers$1.isNullOrUndef(point.y)) {
        return
Chart.defaults.global.tooltips.callbacks.label(tooltipItem, data);

```

Продолжение ПРИЛОЖЕНИЯ Б

```
        }
        const o = point.o;
        const h = point.h;
        const l = point.l;
        const c = point.c;
        return 'O: ' + o + ' H: ' + h + ' L: ' + l +
' C: ' + c;
    }
}
};
/**
 * This class is based off controller.bar.js from the upstream
Chart.js library
 */
const FinancialController = Chart.controllers.bar.extend({

    dataElementType: Chart.elements.Financial,
    /**
     * @private
     */
    _updateElementGeometry(element, index, reset, options) {
        const me = this;
        const model = element._model;
        const vscale = me._getValueScale();
        const base = vscale.getBasePixel();
        const horizontal = vscale.isHorizontal();
        const ruler = me._ruler || me.getRuler();
        const vpixels = me.calculateBarValuePixels(me.index,
index, options);
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        const ipixels = me.calculateBarIndexPixels(me.index,
index, ruler, options);
        const chart = me.chart;
        const datasets = chart.data.datasets;
        const indexData = datasets[me.index].data[index];
        model.horizontal = horizontal;
        model.base = reset ? base : vpixels.base;
        model.x = horizontal ? reset ? base : vpixels.head :
ipixels.center;
        model.y = horizontal ? ipixels.center : reset ? base :
vpixels.head;
        model.height = horizontal ? ipixels.size : undefined;
        model.width = horizontal ? undefined : ipixels.size;
        model.candleOpen =
vscale.getPixelForValue(Number(indexData.o));
        model.candleHigh =
vscale.getPixelForValue(Number(indexData.h));
        model.candleLow =
vscale.getPixelForValue(Number(indexData.l));
        model.candleClose =
vscale.getPixelForValue(Number(indexData.c));
    },
    draw() {
        const ctx = this.chart.chart.ctx;
        const elements = this.getMeta().data;
        const dataset = this.getDataset();
        const ilen = elements.length;
        let i = 0;
        let d;
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        Chart.canvasHelpers.clipArea(ctx, this.chart.chartArea);
        for (; i < ilen; ++i) {
            d = dataset.data[i].o;
            if (d !== null && d !== undefined && !isNaN(d)) {
                elements[i].draw();
            }
        }
        Chart.canvasHelpers.unclipArea(ctx);
    }
});

const helpers$2 = Chart.helpers;
const globalOpts = Chart.defaults.global;

globalOpts.elements.financial = {
    color: {
        up: 'rgba(80, 160, 115, 1)',
        down: 'rgba(215, 85, 65, 1)',
        unchanged: 'rgba(90, 90, 90, 1)',
    }
};

function isVertical(bar) {
    return bar._view.width !== undefined;
}

/**
 * Helper function to get the bounds of the candle
 * @private
 */
```

Продолжение ПРИЛОЖЕНИЯ Б

```
* @param bar {Chart.Element.financial} the bar
* @return {Bounds} bounds of the bar
*/
function getBarBounds(candle) {
    const vm = candle._view;
    const halfWidth = vm.width / 2;
    const x1 = vm.x - halfWidth;
    const x2 = vm.x + halfWidth;
    const y1 = vm.candleHigh;
    const y2 = vm.candleLow;

    return {
        left: x1,
        top: y1,
        right: x2,
        bottom: y2
    };
}

const FinancialElement = Chart.Element.extend({

    height() {
        const vm = this._view;
        return vm.base - vm.y;
    },
    inRange(mouseX, mouseY) {
        let inRange = false;

        if (this._view) {
            const bounds = getBarBounds(this);
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        inRange = mouseX >= bounds.left && mouseX <=
bounds.right && mouseY >= bounds.top && mouseY <= bounds.bottom;
    }
    return inRange;
},
inLabelRange(mouseX, mouseY) {
    const me = this;
    if (!me._view) {
        return false;
    }

    let inRange = false;
    const bounds = getBarBounds(me);
    if (isVertical(me)) {
        inRange = mouseX >= bounds.left && mouseX <=
bounds.right;
    } else {
        inRange = mouseY >= bounds.top && mouseY <=
bounds.bottom;
    }
    return inRange;
},
inXRange(mouseX) {
    const bounds = getBarBounds(this);
    return mouseX >= bounds.left && mouseX <= bounds.right;
},
inYRange(mouseY) {
    const bounds = getBarBounds(this);
    return mouseY >= bounds.top && mouseY <= bounds.bottom;
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
    },
    getCenterPoint() {
      const vm = this._view;
      return {
        x: vm.x,
        y: (vm.candleHigh + vm.candleLow) / 2
      };
    },
    getArea() {
      const vm = this._view;
      return vm.width * Math.abs(vm.y - vm.base);
    },
    tooltipPosition() {
      const vm = this._view;
      return {
        x: vm.x,
        y: (vm.candleOpen + vm.candleClose) / 2
      };
    },
    hasValue() {
      const model = this._model;
      return helpers$2.isNumber(model.x) &&
        helpers$2.isNumber(model.candleOpen) &&
        helpers$2.isNumber(model.candleHigh) &&
        helpers$2.isNumber(model.candleLow) &&
        helpers$2.isNumber(model.candleClose);
    }
  });
```

```
const helpers$3 = Chart.helpers;
const globalOpts$1 = Chart.defaults.global;
```

Продолжение ПРИЛОЖЕНИЯ Б

```
globalOpts$1.elements.candlestick      =      helpers$3.merge({},
[globalOpts$1.elements.financial, {
    borderColor: globalOpts$1.elements.financial.color.unchanged,
    borderWidth: 1,
}]);

const CandlestickElement = FinancialElement.extend({
  draw() {
    const ctx = this._chart.ctx;
    const vm = this._view;
    const x = vm.x;
    const o = vm.candleOpen;
    const h = vm.candleHigh;
    const l = vm.candleLow;
    const c = vm.candleClose;
    let borderColors = vm.borderColor;
    if (typeof borderColors === 'string') {
      borderColors = {
        up: borderColors,
        down: borderColors,
        unchanged: borderColors
      };
    }
    let borderColor;
    if (c < o) {
      borderColor =
helpers$3.getValueOrDefault(borderColors ? borderColors.up : undefined,
globalOpts$1.elements.candlestick.borderColor);
```


Продолжение ПРИЛОЖЕНИЯ Б

```
        ctx.fillStyle = helpers$3.getValueOrDefault(vm.color
? vm.color.up : undefined, globalOpts$1.elements.candlestick.color.up);
        } else if (c > o) {
            borderColor =
helpers$3.getValueOrDefault(borderColors ? borderColors.down : undefined,
globalOpts$1.elements.candlestick.borderColor);
            ctx.fillStyle = helpers$3.getValueOrDefault(vm.color
? vm.color.down : undefined,
globalOpts$1.elements.candlestick.color.down);
        } else {
            borderColor =
helpers$3.getValueOrDefault(borderColors ? borderColors.unchanged :
undefined, globalOpts$1.elements.candlestick.borderColor);
            ctx.fillStyle = helpers$3.getValueOrDefault(vm.color
? vm.color.unchanged : undefined,
globalOpts$1.elements.candlestick.color.unchanged);
        }
        ctx.lineWidth =
helpers$3.getValueOrDefault(vm.borderWidth,
globalOpts$1.elements.candlestick.borderWidth);
        ctx.strokeStyle =
helpers$3.getValueOrDefault(borderColor,
globalOpts$1.elements.candlestick.borderColor);
        ctx.beginPath();
        ctx.moveTo(x, h);
        ctx.lineTo(x, Math.min(o, c));
        ctx.moveTo(x, l);
        ctx.lineTo(x, Math.max(o, c));
        ctx.stroke();
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        ctx.fillRect(x - vm.width / 2, c, vm.width, o - c);
        ctx.strokeRect(x - vm.width / 2, c, vm.width, o - c);
        ctx.closePath();
    }
});

Chart.defaults.candlestick = Chart.helpers.merge({},
Chart.defaults.financial);
Chart.defaults._set('global', {
    datasets: {
        candlestick: Chart.defaults.global.datasets.bar
    }
});
const CandlestickController = Chart.controllers.candlestick =
FinancialController.extend({
    dataElementType: CandlestickElement,
    updateElement(element, index, reset) {
        const me = this;
        const meta = me.getMeta();
        const dataset = me.getDataset();
        const options = me._resolveDataElementOptions(element,
index);

        element._xScale = me.getScaleForId(meta.xAxisID);
        element._yScale = me.getScaleForId(meta.yAxisID);
        element._datasetIndex = me.index;
        element._index = index;
        element._model = {
            datasetLabel: dataset.label || '',
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        color: dataset.color,
        borderColor: dataset.borderColor,
        borderWidth: dataset.borderWidth,
    };

    me._updateElementGeometry(element, index, reset,
options);

    element.pivot();
    },

});

const helpers$4 = Chart.helpers;
const globalOpts$2 = Chart.defaults.global;

globalOpts$2.elements.ohlc = helpers$4.merge({},
[globalOpts$2.elements.financial, {
    lineWidth: 2,
    armLength: null,
    armLengthRatio: 0.8,
}]]);

const OhlcElement = FinancialElement.extend({
    draw() {
        const ctx = this._chart.ctx;
        const vm = this._view;
        const x = vm.x;
        const o = vm.candleOpen;
        const h = vm.candleHigh;
        const l = vm.candleLow;
        const c = vm.candleClose;
```

Продолжение ПРИЛОЖЕНИЯ Б

```
const armLengthRatio =
helpers$4.getValueOrDefault(vm.armLengthRatio,
globalOpts$2.elements.ohl.c.armLengthRatio);
    let armLength = helpers$4.getValueOrDefault(vm.armLength,
globalOpts$2.elements.ohl.c.armLength);
    if (armLength === null) {
        armLength = vm.width * armLengthRatio * 0.5;
    }
    if (c < o) {
        ctx.strokeStyle =
helpers$4.getValueOrDefault(vm.color ? vm.color.up : undefined,
globalOpts$2.elements.ohl.c.color.up);
    } else if (c > o) {
        ctx.strokeStyle =
helpers$4.getValueOrDefault(vm.color ? vm.color.down : undefined,
globalOpts$2.elements.ohl.c.color.down);
    } else {
        ctx.strokeStyle =
helpers$4.getValueOrDefault(vm.color ? vm.color.unchanged : undefined,
globalOpts$2.elements.ohl.c.color.unchanged);
    }
}
```

Продолжение ПРИЛОЖЕНИЯ Б

```
        ctx.lineWidth = helpers$4.getValueOrDefault(vm.lineWidth,
globalOpts$2.elements.ohlc.lineWidth);
        ctx.beginPath();
        ctx.moveTo(x, h);
        ctx.lineTo(x, l);
        ctx.moveTo(x - armLength, o);
        ctx.lineTo(x, o);
        ctx.moveTo(x + armLength, c);
        ctx.lineTo(x, c);
        ctx.stroke();
    }
});
```

```
Chart.defaults.ohlc = Chart.helpers.merge({},
Chart.defaults.financial);
```

```
Chart.defaults._set('global', {
  datasets: {
    ohlc: {
      barPercentage: 1.0,
      categoryPercentage: 1.0
    }
  }
});
```

```
const OhlcController = Chart.controllers.ohlc =
FinancialController.extend({
  dataElementType: OhlcElement,
  updateElement(element, index, reset) {
    const me = this;
    const meta = me.getMeta();
    const dataset = me.getDataset();
```

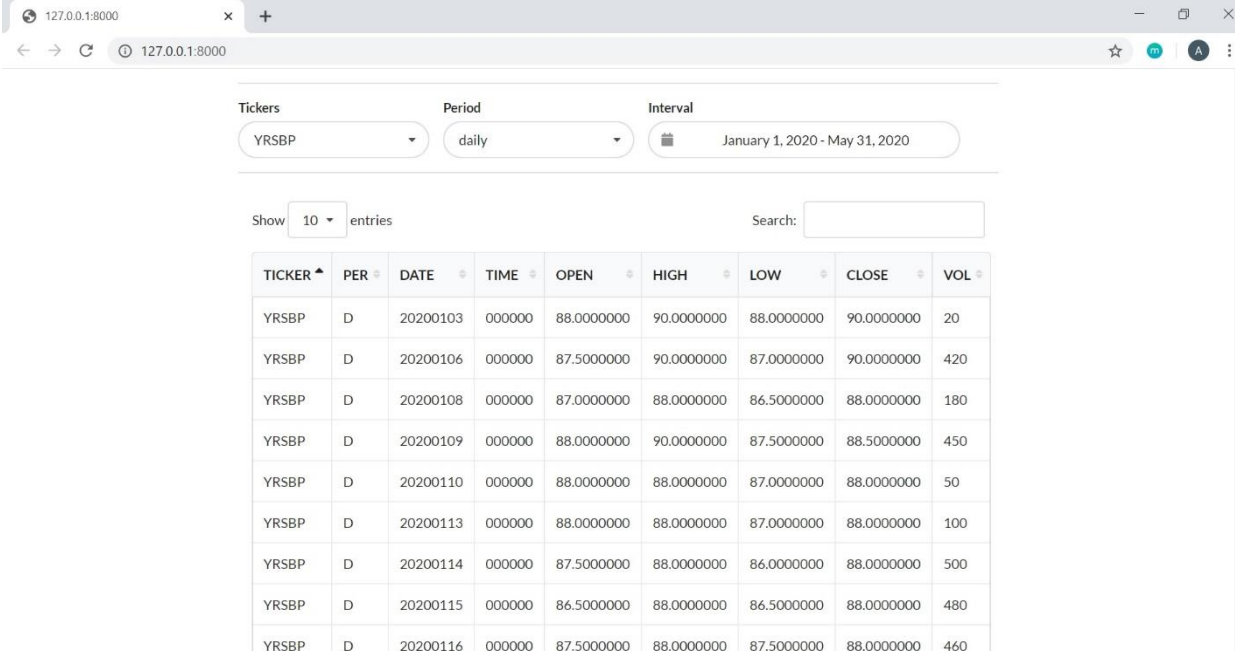
Окончание ПРИЛОЖЕНИЯ Б

```
const options = me._resolveDataElementOptions(element,
index);

element._xScale = me.getScaleForId(meta.xAxisID);
element._yScale = me.getScaleForId(meta.yAxisID);
element._datasetIndex = me.index;
element._index = index;
element._model = {
    datasetLabel: dataset.label || '',
    lineWidth: dataset.lineWidth,
    armLength: dataset.armLength,
    armLengthRatio: dataset.armLengthRatio,
    color: dataset.color,
};
me._updateElementGeometry(element, index, reset,
options);
element.pivot();
},
});
}}));
```

ПРИЛОЖЕНИЕ В (обязательное)

Демонстрация работы ПО



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000". The page content includes a search bar with "YRSBP" entered, a "Period" dropdown set to "daily", and an "Interval" dropdown set to "January 1, 2020 - May 31, 2020". Below the search bar, there is a "Show 10 entries" option and a "Search:" input field. The main content is a table with the following data:

TICKER	PER	DATE	TIME	OPEN	HIGH	LOW	CLOSE	VOL
YRSBP	D	20200103	000000	88.00000000	90.00000000	88.00000000	90.00000000	20
YRSBP	D	20200106	000000	87.50000000	90.00000000	87.00000000	90.00000000	420
YRSBP	D	20200108	000000	87.00000000	88.00000000	86.50000000	88.00000000	180
YRSBP	D	20200109	000000	88.00000000	90.00000000	87.50000000	88.50000000	450
YRSBP	D	20200110	000000	88.00000000	88.00000000	87.00000000	88.00000000	50
YRSBP	D	20200113	000000	88.00000000	88.00000000	87.00000000	88.00000000	100
YRSBP	D	20200114	000000	87.50000000	88.00000000	86.00000000	88.00000000	500
YRSBP	D	20200115	000000	86.50000000	88.00000000	86.50000000	88.00000000	480
YRSBP	D	20200116	000000	87.50000000	88.00000000	87.50000000	88.00000000	460

Рисунок 9 – Веб-страница с котировками акций компании ТНС ЭНЕРГО ЯРОСЛАВЛЬ ПАО с дневным таймфреймом с 1 января 2020 года по 31 мая 2020 года

Продолжение ПРИЛОЖЕНИЯ В

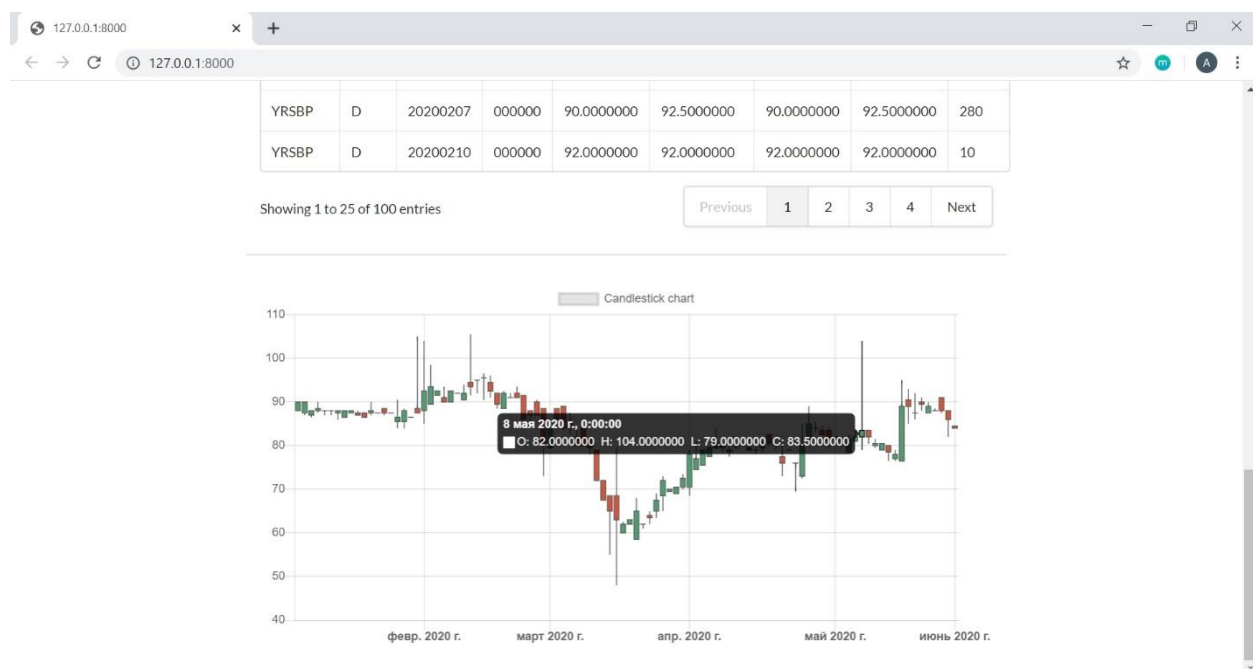


Рисунок 10 – Веб-страница с графиком японских свечей по котировкам акций компании ТНС ЭНЕРГО ЯРОСЛАВЛЬ ПАО с дневным таймфреймом с 1 января 2020 года по 31 мая 2020 года

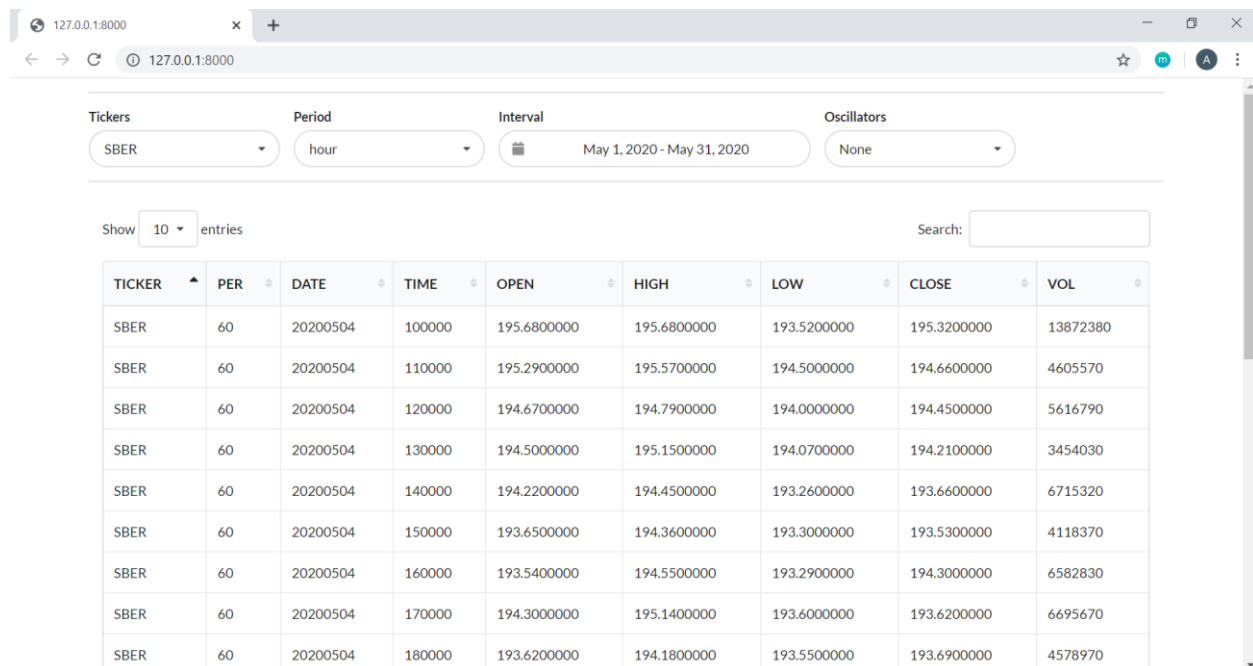


Рисунок 11 – Веб-страница с котировками акций компании СберБанк с часовым таймфреймом с 1 мая 2020 года по 31 мая 2020 года

Продолжение ПРИЛОЖЕНИЯ В

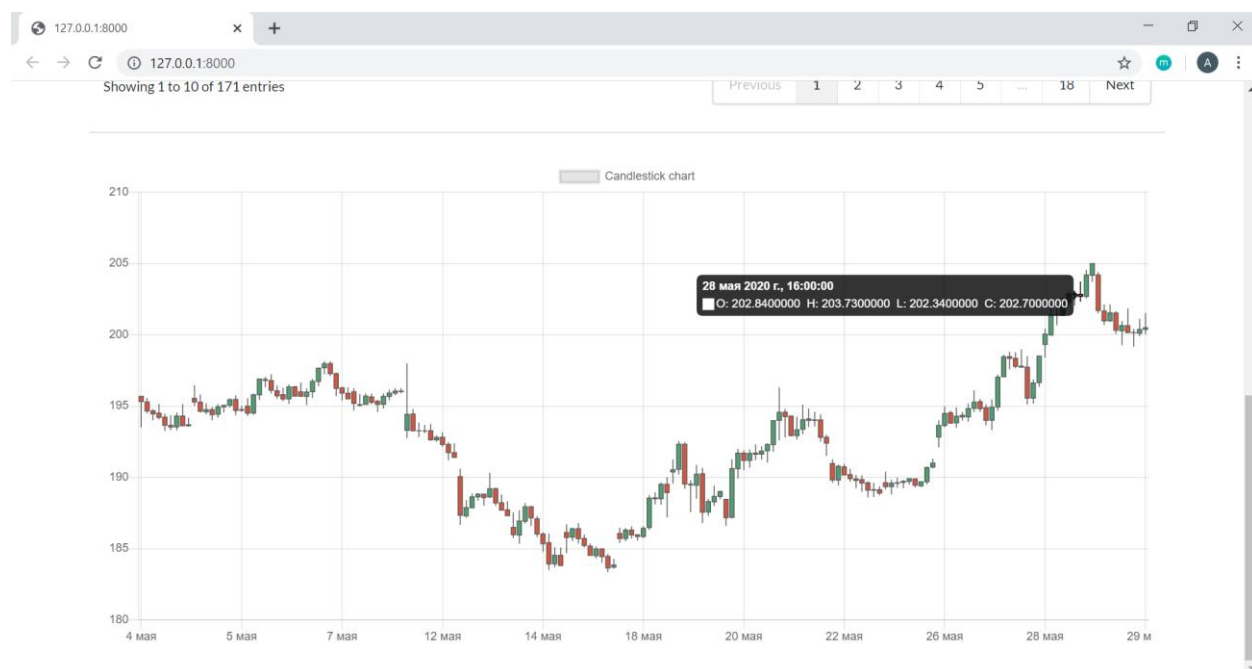


Рисунок 12 – Веб-страница с графиком японских свечей по котировкам акций компании СберБанк с часовым таймфреймом с 1 мая 2020 года по 31 мая 2020 года

Окончание ПРИЛОЖЕНИЯ В

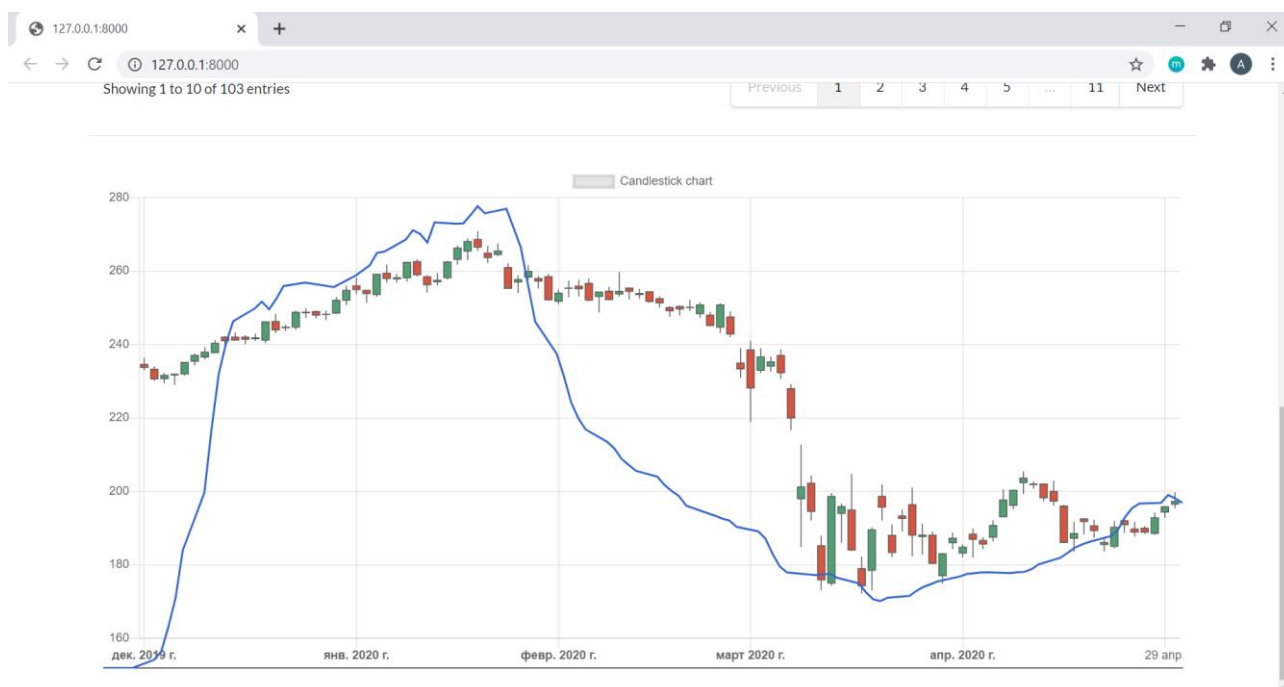


Рисунок 13 – Веб-страница с графиком японских свечей и индексом МА по котировкам акций компании СберБанк с дневным таймфреймом с 1 декабря 2019 года по 30 апреля 2020 года