

Школа: Инженерная школа информационных технологий и робототехники  
 Направление подготовки: 09.03.04 «Программная инженерия»  
 Отделение школы (НОЦ): Отделение информационных технологий

### БАКАЛАВРСКАЯ РАБОТА

Тема работы
<b>Программное обеспечение для мониторинга торговой активности на криптовалютных биржах</b>

УДК 004.415.2:004.738.5:339:336.743

Студент

Группа	ФИО	Подпись	Дата
8К81	Авербах Андрей Викторович		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Аксёнов Сергей Владимирович	к.т.н., доцент		

### КОНСУЛЬТАНТЫ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН ШБИП	Гасанов Магеррам Али оглы	д.э.н., профессор		

По разделу «Социальная ответственность»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Старший преподаватель ООД ШБИП	Мезенцева Ирина Леонидовна	-		

### ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н., доцент		

## ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ ООП

Код компетенции	Наименование компетенции
УК(У)-1	Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач
УК(У)-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм, имеющихся ресурсов и ограничений
УК(У)-3	Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде
УК(У)-4	Способен осуществлять деловую коммуникацию в устной и письменной формах на государственном языке Российской Федерации и иностранном(-ых) языке(-ах)
УК(У)-5	Способен воспринимать межкультурное разнообразие общества в социально-историческом, этическом и философском контекстах
УК(У)-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
УК(У)-7	Способен поддерживать должный уровень физической подготовленности для обеспечения полноценной социальной и профессиональной деятельности
УК(У)-8	Способен создавать и поддерживать безопасные условия жизнедеятельности, в том числе при возникновении чрезвычайных ситуаций
УК(У)-9	Способен проявлять предприимчивость в практической деятельности, в т.ч. в рамках разработки коммерчески перспективного продукта на основе научно-технической идеи
ОПК(У)-1	Способен применять естественнонаучные и общинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности
ОПК(У)-2	Способен использовать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности
ОПК(У)-3	Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
ОПК(У)-4	Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью
ОПК(У)-5	Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем
ОПК(У)-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ОПК(У)-7	Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой
ОПК(У)-8	Способен осуществлять поиск, хранение, обработку и анализ информации из различных источников и баз данных, представлять ее

	в требуемом формате с использованием информационных, компьютерных и сетевых технологий
<b>ПК(У)-1</b>	Владение навыками разработки требований и проектирования программного обеспечения
<b>ПК(У)-2</b>	Владение навыками разработки документов и стратегии тестирования программного обеспечения
<b>ПК(У)-3</b>	Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения
<b>ПК(У)-4</b>	Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных
<b>ПК(У)-5</b>	Владение концепциями и атрибутами качества программного обеспечения (надежности, безопасности, удобства использования), в том числе роли людей, процессов, методов, инструментов и технологий обеспечения качества

Школа: Инженерная школа информационных технологий и робототехники  
 Направление подготовки: 09.03.04 «Программная инженерия»  
 Отделение школы (НОЦ): Отделение информационных технологий

УТВЕРЖДАЮ:  
 Руководитель ООП  
 \_\_\_\_\_ Чердынцев Е.С.  
 (Подпись) (Дата) (Ф.И.О.)

**ЗАДАНИЕ  
на выполнение выпускной квалификационной работы**

В форме:

<b>Бакалаврской работы</b>
<small>(бакалаврской работы, дипломного проекта/работы, магистерской диссертации)</small>

Студенту:

Группа	ФИО
8K81	Авербаху Андрею Викторовичу

Тема работы:

Программное обеспечение для мониторинга торговой активности на криптовалютных биржах	
Утверждена приказом директора (дата, номер)	№40-51/с от 09.02.2022

Срок сдачи студентом выполненной работы:	06.06.2022 г.
--	---------------

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ:**

<p><b>Исходные данные к работе</b>  <small>(наименование объекта исследования или проектирования; производительность или нагрузка; режим работы (непрерывный, периодический, циклический и т. д.); вид сырья или материал изделия; требования к продукту, изделию или процессу; особые требования к особенностям функционирования (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.).</small></p>	<p>Работа направлена на создание программного обеспечения для мониторинга торговой активности на криптовалютных биржах.</p>
<p><b>Перечень подлежащих исследованию, проектированию и разработке вопросов</b>  <small>(аналитический обзор по литературным источникам с целью выяснения достижений мировой науки техники в рассматриваемой области; постановка задачи исследования, проектирования, конструирования; содержание процедуры исследования, проектирования, конструирования; обсуждение результатов выполненной работы; наименование дополнительных разделов, подлежащих разработке; заключение по работе).</small></p>	<ol style="list-style-type: none"> <li>1. Исследование предметной области</li> <li>2. Проектирование программного обеспечения</li> <li>3. Разработка программного обеспечения</li> <li>4. Тестирование разработанного ПО</li> <li>5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение</li> <li>6. Социальная ответственность</li> </ol>
<p><b>Перечень графического материала</b>  <small>(с точным указанием обязательных чертежей)</small></p>	<ol style="list-style-type: none"> <li>1. Диаграмма Fishbone</li> <li>2. Диаграмма DFD</li> </ol>

	3. Рисунки, демонстрирующие разработанное ПО 4. Диаграмма Ганта
--	--

**Консультанты по разделам выпускной квалификационной работы**

*(с указанием разделов)*

Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна

<b>Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику</b>	09.02.2022 г.
---	---------------

**Задание выдал руководитель:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Аксёнов Сергей Владимирович	К.Т.Н., доцент		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8K81	Авербах Андрей Викторович		

Школа: Инженерная школа информационных технологий и робототехники  
 Направление подготовки: 09.03.04 «Программная инженерия»  
 Отделение школы (НОЦ): Отделение информационных технологий  
 Период выполнения: весенний семестр 2022 учебного года

Форма представления работы:

Бакалаврская работа
---------------------

### КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Срок сдачи студентом выполненной работы:	06.06.2022
--	------------

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
02.03.2022	Глава 1. Исследование предметной области	15
31.03.2022	Глава 2. Проектирование программного обеспечения	25
04.05.2022	Глава 3. Разработка программного обеспечения	22
16.05.2022	Глава 4. Тестирование и апробация разработанного ПО	18
19.05.2022	Глава 5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	10
31.05.2022	Глава 6. Социальная ответственность	10

Составил преподаватель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Аксёнов Сергей Владимирович	к.т.н., доцент		

**СОГЛАСОВАНО:**

Руководитель ООП	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР	Чердынцев Евгений Сергеевич	к.т.н., доцент		

**ЗАДАНИЕ ДЛЯ РАЗДЕЛА  
«ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И  
РЕСУРСОСБЕРЕЖЕНИЕ»**

Студенту:

<b>Группа</b>	<b>ФИО</b>
8K81	Авербах Андрей Викторович

<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение школы (НОЦ)</b>	<b>ОИТ</b>
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

**Исходные данные к разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»:**

1. <i>Стоимость ресурсов научного исследования (НИ): материально-технических, энергетических, финансовых, информационных и человеческих</i>	<i>Оклад программиста – 40000 руб.</i>
2. <i>Нормы и нормативы расходования ресурсов</i>	<i>Премияльный коэффициент программиста 20%; Доплаты и надбавки программиста 30%; Коэффициент дополнительной заработной платы 12%; Накладные расходы 16%; Районный коэффициент 1,3.</i>
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	<i>Коэффициент отчислений на уплату во внебюджетные фонды 30,2 %</i>

**Перечень вопросов, подлежащих исследованию, проектированию и разработке:**

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	<i>Определение потенциального потребителя результатов исследования, SWOT-анализ разработанной стратегии.</i>
2. <i>Планирование и формирование бюджета научных исследований</i>	<i>Определение структуры работы. Расчет трудоемкости выполнения работ. Подсчет бюджета исследования.</i>
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	<i>Расчитать показатели финансовой эффективности, ресурсоэффективности и эффективности исполнения.</i>

**Перечень графического материала (с точным указанием обязательных чертежей):**

1. <i>Оценка конкурентоспособности технических решений</i>
2. <i>Матрица SWOT</i>
3. <i>График проведения и бюджет НИ</i>
4. <i>Оценка ресурсной, финансовой и экономической эффективности НИ</i>

<b>Дата выдачи задания для раздела по линейному графику</b>	
---	--

**Задание выдал консультант:**

<b>Должность</b>	<b>ФИО</b>	<b>Ученая степень, звание</b>	<b>Подпись</b>	<b>Дата</b>
Профессор ОСГН	Гасанов Магеррам Али оглы	Д.э.н., профессор		

**Задание принял к исполнению студент:**

<b>Группа</b>	<b>ФИО</b>	<b>Подпись</b>	<b>Дата</b>
8K81	Авербах Андрей Викторович		

## ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

<b>Группа</b> 8K81	<b>ФИО</b> Авербах Андрей Викторович		
<b>Школа</b>	<b>ИШИТР</b>	<b>Отделение (НОЦ)</b>	<b>ОИТ</b>
<b>Уровень образования</b>	Бакалавриат	<b>Направление/специальность</b>	09.03.04 Программная инженерия

Тема ВКР:

Программное обеспечение для мониторинга торговой активности на криптовалютных биржах	
<b>Исходные данные к разделу «Социальная ответственность»:</b>	
<p><b>Введение</b></p> <ul style="list-style-type: none"> <li>– Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения.</li> <li>– Описание рабочей зоны (рабочего места) при разработке проектного решения</li> </ul>	<p>Объект исследования: программное обеспечение для мониторинга торговой активности на криптовалютных биржах.          Область применения: автоматизация бизнес-процессов торговли на криптовалютных биржах.          Рабочая зона: офис.          Размеры помещения: 18 м<sup>2</sup>.          Количество и наименование оборудования рабочей зоны: ноутбук 1 шт.          Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне: поиск информации, её анализ; проектирование и разработка программного обеспечения.</p>
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
<p><b>1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения:</b></p> <ul style="list-style-type: none"> <li>– специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства;</li> <li>– организационные мероприятия при компоновке рабочей зоны.</li> </ul>	<p>«Трудовой кодекс Российской Федерации» от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022)          ГОСТ 21889-76 Система «человек-машина». Кресло человека-оператора. Общие эргономические требования.          ГОСТ Р 50923-96 Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения</p>
<p><b>2. Производственная безопасность при разработке проектного решения:</b></p> <ul style="list-style-type: none"> <li>– Анализ выявленных вредных и опасных производственных факторов</li> </ul>	<p>Вредные факторы:</p> <ul style="list-style-type: none"> <li>• Статические физические перегрузки, связанные с рабочей позой.</li> <li>• Умственное перенапряжение, в том числе вызванное информационной нагрузкой.</li> <li>• Перенапряжение анализаторов, в том числе вызванное информационной нагрузкой.</li> <li>• Монотонность труда, вызывающая монотонию.</li> <li>• Отсутствие или недостаток необходимого естественного освещения.</li> <li>• Отсутствие или недостаток необходимого искусственного освещения.</li> </ul> <p>Опасные факторы:</p> <ul style="list-style-type: none"> <li>• Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий.</li> </ul> <p>Требуемые средства индивидуальной и коллективной защиты от выявленных факторов:</p> <ul style="list-style-type: none"> <li>• Очки защитные со спектральными фильтрами ЛС и НСФ.</li> </ul>

	<ul style="list-style-type: none"> <li>• Устройства защитного отключения (УЗО).</li> <li>• Заземление, зануление корпусов электрооборудования.</li> </ul>
<b>3. Экологическая безопасность при разработке проектного решения</b>	Воздействие на литосферу – вышедшие из строя ПЭВМ и оргтехника. Воздействие на гидросферу – не выявлено. Воздействие на атмосферу – не выявлено.
<b>4. Безопасность в чрезвычайных ситуациях при разработке проектного решения</b>	Возможные ЧС: наводнение, землетрясение, пожар. Наиболее типичная ЧС: пожар.
<b>Дата выдачи задания для раздела по линейному графику</b>	

**Задание выдал консультант:**

Должность	ФИО	Ученая степень, звание	Подпись	Дата
старший преподаватель	Мезенцева Ирина Леонидовна	-		

**Задание принял к исполнению студент:**

Группа	ФИО	Подпись	Дата
8К81	Авербах Андрей Викторович		

## Реферат

Выпускная квалификационная работа содержит 101 страницу, 41 рисунок, 13 таблиц, 0 приложений.

Ключевые слова: криптовалюты, трейдинг, биржа, программное обеспечение, cryptocurrency, trading, exchange.

Цель работы – создание программного обеспечения, позволяющего своевременно доводить до трейдера важную и актуальную рыночную информацию в компактном графическом и аудиальном представлении.

В первой главе представлена краткая теория предметной области и определены потребности пользователей.

Во второй главе определяется основная функциональность будущего программного обеспечения и описывается процесс проектирования архитектуры системы.

Третья глава содержит информацию о реализации программного обеспечения, в том числе обоснование выбора библиотек и средств разработки.

В четвертой главе описывается закрытое тестирование разработанного программного обеспечения.

Пятая глава представляет собой выполненное задание по разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение».

В шестой главе содержится выполненное задание по разделу «Социальная ответственность».

Наиболее значимым результатом работы является использование разработанного программного обеспечения конечными пользователями.

Выпускная квалификационная работа выполнена в текстовом редакторе Microsoft Office Word 2016.

## Оглавление

Термины и сокращения .....	14
Введение.....	16
1. Исследование предметной области.....	18
1.1. Краткая теория предметной области .....	18
1.2. Определение потребностей пользователей.....	22
2. Проектирование программного обеспечения .....	27
2.1. Определение основного функционала приложения .....	27
2.2. Проектирование высокоуровневой архитектуры приложения.....	27
2.3. Проектирование серверной части .....	30
3. Разработка программного обеспечения.....	41
3.1. Разработка серверной части .....	41
3.2. Разработка клиентской части .....	55
3.3. Обзор основных возможностей приложения.....	55
3.3.1. Крупные плотности.....	57
3.3.2. Графики .....	58
3.3.3. Уведомления.....	58
3.3.4. Автоматический поиск горизонтальных уровней.....	59
3.3.5. Скринер .....	59
3.3.6. Группы линковки и перетаскивание (drag and drop) инструментов .	62
3.3.7. Взаимодействие с торговыми терминалами.....	64
4. Тестирование и апробация разработанного ПО .....	65
5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение ...	69
Введение.....	69
5.1. Оценка коммерческого потенциала и перспективности разработки .....	69

5.1.1.	Потенциальные потребители разрабатываемого решения .....	69
5.1.2.	Анализ конкурентных технических решений .....	70
5.1.3.	SWOT-анализ .....	72
5.2.	Планирование работ по научно-техническому исследованию .....	73
5.2.1.	Структура работ в рамках научного исследования .....	73
5.2.2.	Определение трудоемкости выполнения работ .....	74
5.2.3.	Разработка графика проведения научного исследования .....	74
5.3.	Бюджет научно-технического исследования .....	77
5.3.1.	Расчет материальных затрат НТИ .....	77
5.3.2.	Расчет затрат на специальное оборудование для научных работ .....	77
5.3.3.	Основная заработная плата исполнителей .....	78
5.3.4.	Дополнительная заработная плата .....	79
5.3.5.	Отчисления во внебюджетные фонды .....	80
5.3.6.	Накладные расходы .....	80
5.3.7.	Формирование бюджета затрат научно-исследовательского проекта .....	81
5.4.	Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования .....	81
	Вывод по главе .....	84
6.	Социальная ответственность .....	85
	Введение .....	85
6.1.	Правовые и организационные вопросы обеспечения безопасности .....	85
6.1.1.	Правовые нормы трудового законодательства .....	85
6.1.2.	Эргономические требования к правильному расположению и компоновке рабочей зоны .....	86

6.2. Производственная безопасность .....	87
6.2.1. Статические физические перегрузки .....	89
6.2.2. Умственное перенапряжение и перенапряжение анализаторов, монотонность труда .....	89
6.2.3. Отсутствие или недостаток необходимого естественного освещения .....	91
6.2.4. Отсутствие или недостаток необходимого искусственного освещения .....	92
6.2.5. Опасность поражения электрическим током .....	93
6.3. Экологическая безопасность .....	94
6.4. Безопасность в чрезвычайных ситуациях .....	94
Выводы по главе.....	96
Заключение .....	97
Список использованных источников и литературы .....	98

## **Термины и сокращения**

API (Application Programming Interface) – описание способов, которыми одна компьютерная программа может взаимодействовать с другой программой.

HTTP (HyperText Transfer Protocol) – протокол прикладного уровня передачи данных.

HTTPS – расширение протокола HTTP для поддержки шифрования в целях повышения безопасности.

JSON (JavaScript Object Notation) – текстовый формат обмена данными, основанный на JavaScript.

SIMD — принцип компьютерных вычислений, позволяющий обеспечить параллелизм на уровне данных.

SSL — криптографический протокол, который подразумевает более безопасную связь. Он использует асимметричную криптографию для аутентификации ключей обмена, симметричное шифрование для сохранения конфиденциальности, коды аутентификации сообщений для целостности сообщений.

WebSocket — протокол связи поверх TCP-соединения, предназначенный для обмена сообщениями между браузером и веб-сервером в режиме реального времени.

Инструмент – криптовалютная пара либо фьючерс на криптовалютную пару.

Криптовалюта — разновидность цифровой валюты, учёт внутренних расчётных единиц которой обеспечивает децентрализованная платёжная система, работающая в полностью автоматическом режиме.

Парсинг – это принятое в информатике определение синтаксического анализа.

Сопрограмма (англ. coroutine) — программный модуль, особым образом организованный для обеспечения взаимодействия с другими модулями по принципу кооперативной многозадачности: модуль приостанавливается в определённой точке, сохраняя полное состояние (включая стек вызовов и счётчик команд), и передаёт управление другому, тот, в свою очередь, выполняет задачу и передаёт управление обратно, сохраняя свои стек и счётчик.

Таймфрейм — интервал времени, используемый для группировки котировок при построении элементов ценового графика (бара, японской свечи, точки линейного графика).

Треjder — это участник биржевой торговли, осуществляющий сделки с целью получения краткосрочной прибыли.

Фреймворк — программное обеспечение, облегчающее разработку и объединение разных модулей программного проекта.

## **Введение**

В настоящее время криптовалюты доказали свою значимость и полезность, а интерес к ним только продолжает расти. Изначально созданные для быстрых переводов денежных средств с низкими комиссиями без необходимости в центральном регулирующем органе, криптовалюты сейчас приобрели множество различных функций, среди которых можно выделить инвестиционную и спекулятивную.

Появлению и укреплению этих ролей способствовало появление централизованных и децентрализованных бирж, где люди и компании могут покупать и продавать криптовалюту, а также производить операции с производными инструментами таким же образом, как и на традиционных фондовых биржах.

В то время, как одни предпочитают купить криптовалюту и ждать значительного повышения ее цены, то есть занимать позицию инвестора, другие используют гораздо более высокую волатильность криптовалют по сравнению с акциями, облигациями и т.д., чтобы совершать быстрые сделки с целью максимизировать эффективность использования своего торгового капитала (однако следует отметить, что степень риска здесь во много раз больше).

В число последних входят так называемые дневные трейдеры (дейтрейдеры), которые совершают сделки внутри одного торгового дня (торговой сессии), и скальперы, которые совершают еще больше сделок, а время нахождения в отдельной сделке может составлять от нескольких секунд до нескольких часов.

Так как рынок криптовалют является достаточно молодым по сравнению с традиционными рынками (фондовым, товарным), то на нем присутствует большое количество различных неэффективностей, которые, будучи правильно распознанными и использованными, могут увеличить результативность и прибыльность работы скальпера.

Криптовалютная биржа Binance[1] является одной из крупнейших криптовалютных бирж в мире по величине суточного оборота торгов: 36 миллиардов долларов на базовых активах, то есть криптовалютах непосредственно, и 50 миллиардов долларов на производных инструментах (фьючерсы и опционы) по данным сайта coinmarketcap.com[2] на момент 13 мая 2022 года. На бирже существует более 120 криптовалютных пар к USDT, более 100 фьючерсных пар к USDT. USDT или Tether[3] представляет собой криптовалюту, одна единица которой соответствует одному доллару США, так называемый стэйблкоин. Пары к USDT имеют наибольший объем торгов среди прочих.

Существенной проблемой для скальпера становится одновременное оперативное отслеживание изменений и ситуаций, возникающих на всех доступных инструментах (криптовалютных парах или фьючерсах), так как человеку физически не хватает внимания, чтоб отслеживать такое количество событий.

Таким образом, возникает нужда в программном обеспечении, **целью** которого является своевременно доводить до трейдера важную и актуальную рыночную информацию в компактном графическом и аудиальном представлении. Это позволит не следить одновременно за всем разнообразием доступных инструментов, а сосредоточить своё внимание на тех из них, где в данный момент существуют интересующие пользователя условия.

Для создания такого ПО необходимо выполнить следующие **задачи**:

1. Определить потребности пользователей и основной функционал.
2. Спроектировать программное обеспечение и выбрать средства его реализации.
3. Реализовать программное обеспечение.
4. Предоставить программное обеспечение ограниченному кругу пользователей для тестирования с последующим получением от них обратной связи.

## **1. Исследование предметной области**

### **1.1. Краткая теория предметной области**

Торги проводятся на специальной электронной площадке, именуемой биржей.

Участники торгов могут подавать свои поручения на покупку, либо продажу определённого актива по определённой цене и в определённом количестве — заявки. Наиболее распространёнными видами заявок являются лимитная и рыночная заявки.

Торговое поручение — выражение в электронной форме волеизъявления участника торгов, представляющее собой твердое намерение участника торгов совершить одну или несколько сделок купли-продажи определённого актива и содержащее все необходимые для его исполнения сведения. Участники подают свои поручения, используя специализированное программное обеспечение, которое в свою очередь взаимодействует с API, предоставляемым биржей.

Лимитная заявка на покупку (продажу) — заявка участника торгов, содержащая поручение купить (продать) определённое число единиц актива по цене не выше (не ниже) заданной. Если лимитная заявка не может быть исполнена сразу полностью, то неисполненный объём выставляется в очередь заявок.

Очередь заявок (очередь, книга заявок, книга ордеров, жарг. «стакан») — упорядоченная последовательность зарегистрированных неисполненных заявок и/или частично исполненных заявок. При постановке заявки в очередь её место в очереди определяется ценой, указанной в заявке (первыми в очереди заявок на покупку находятся заявки с большими ценами; первыми в очереди заявок на продажу находятся заявки с меньшими ценами). Среди заявок с равными ценами первыми в очереди находятся заявки, зарегистрированные раньше.

На рисунке 1 представлена визуализация очереди заявок на Московской Бирже, однако аналогичное представление используется и на биржах

криптовалют. В крайнем левом и правом столбце указан совокупный объем заявок на покупку (продажу) по данной цене.

Покупка	Цена	Продаж
	99,89	444
	99,88	171
	99,87	421
	99,86	175
	99,85	298
	99,84	418
	99,83	263
	99,82	1 063
	99,81	106
	99,80	1 097
	99,79	356
212	99,77	
157	99,76	
140	99,75	
241	99,74	
348	99,73	
283	99,72	
39	99,71	
700	99,70	
236	99,69	
671	99,68	
1 219	99,67	
380	99,66	

Рисунок 1 – Визуализация очереди заявок

Рыночная заявка на покупку (продажу) — поручение участника торгов купить (продать) определённое число единиц актива без учёта цены.

Пример представлен на рисунке 2: пусть в очередь заявок поступает рыночная заявка на покупку 2000 единиц актива.

Сначала будут куплены 356 единиц по цене 99,79. Это будет отражено как сделка (покупка) объемом 356 по цене 99,79. От изначальной заявки осталось 1644 единицы.

Далее будут куплены 1097 единиц по цене 99,80. Это будет отражено как сделка (покупка) объемом 1097 по цене 99,80. От изначальной заявки осталось 547 единиц.

Следующими будут куплены 106 единиц по цене 99,81. Это будет отражено как сделка (покупка) объемом 106 по цене 99,81. От изначальной заявки осталось 441 единица.

Наконец, будет куплена 441 единица по цене 99,82. Это будет отражено как сделка (покупка) объемом 441 по цене 99,82. Заявка полностью исполнена.

Теперь первой заявкой в очереди на продажу является заявка по цене 99,82 объемом 622.

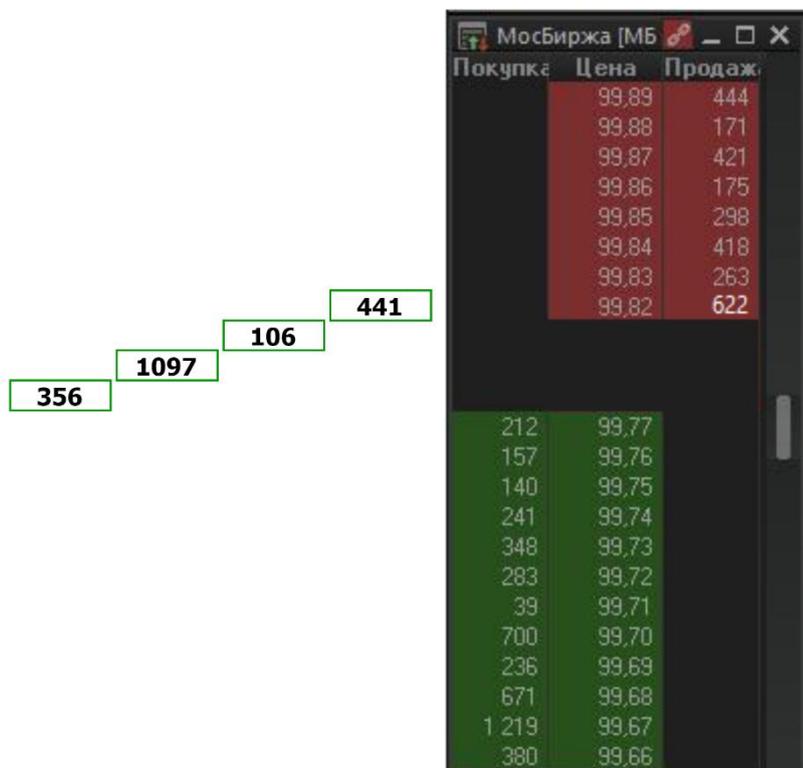


Рисунок 2 – Последовательность сделок и визуализация очереди заявок после исполнения рыночной заявки на покупку объемом 2000 единиц

Таким образом, результатом исполнения одной заявки могут стать одна или несколько сделок. Каждая сделка имеет единственную цену.

Цена актива — цена последней сделки, которая была совершена по данному активу.

Можно построить график зависимости цены актива от времени. Наиболее популярным видом такого графика является график типа «японские свечи»[4], представленный на рисунке 3.

Свеча, как показано на рисунке 4, состоит из тела и тени. Верхняя граница тени отображает максимальную цену актива за данный период времени, нижняя — минимальную. Нижняя граница тела показывает цену открытия свечи, то есть

цену актива на момент начала периода свечи, верхняя граница тела — цену закрытия свечи, то есть цену актива на момент окончания периода свечи. Если цена закрытия свечи выше цены открытия, то цвет тела светлый (на примере выше бирюзовый), иначе — тёмный (на примере выше красный).



Рисунок 3 – Пример пятнадцатиминутного свечного графика цены в зависимости от времени[5]

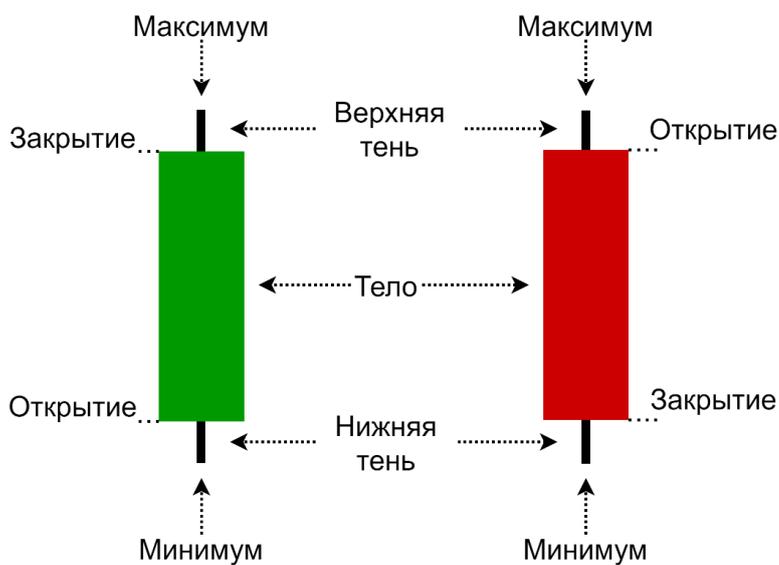


Рисунок 4 – Строение свечи

В нижней части рисунка 3 присутствует ещё один график — график объёмов, каждый столбик которого соответствует сумме объёмов всех сделок, совершенных за данный период, независимо от их направления (на покупку или на продажу).

Следует обратить внимание на то, что период должен начинаться в момент времени «кратный» этому периоду. Например, 15-минутная свеча должна начинаться в 10:00, 12:45 или 16:30, но не может начинаться в 10:03, 09:54 или 15:12.

Период длится ровно до начала следующего периода. То есть, если сделка была совершена в 10:14:59, то она будет учтена в свече, соответствующей времени 10:00; если же сделка была совершена в 10:15:00 или 10:15:01, то она будет учтена в свече, соответствующей времени 10:15.

Если за данный период не было совершено сделок, то цена открытия, цена закрытия, цена максимума, цена минимума свечи полагается равной цене закрытия предыдущей свечи.

## **1.2. Определение потребностей пользователей**

Скальперы торгуют, используя специальное программное обеспечение, именуемое скальперский привод, привод или просто терминал. Оно помогает видеть текущий баланс спроса и предложения, структуру спроса и предложения, быстро (одним кликом мыши) выставлять и отменять заявки, совершать сделки. Интерфейс популярного среди русскоязычных трейдеров терминала CScalp[6] представлен на рисунке 5.

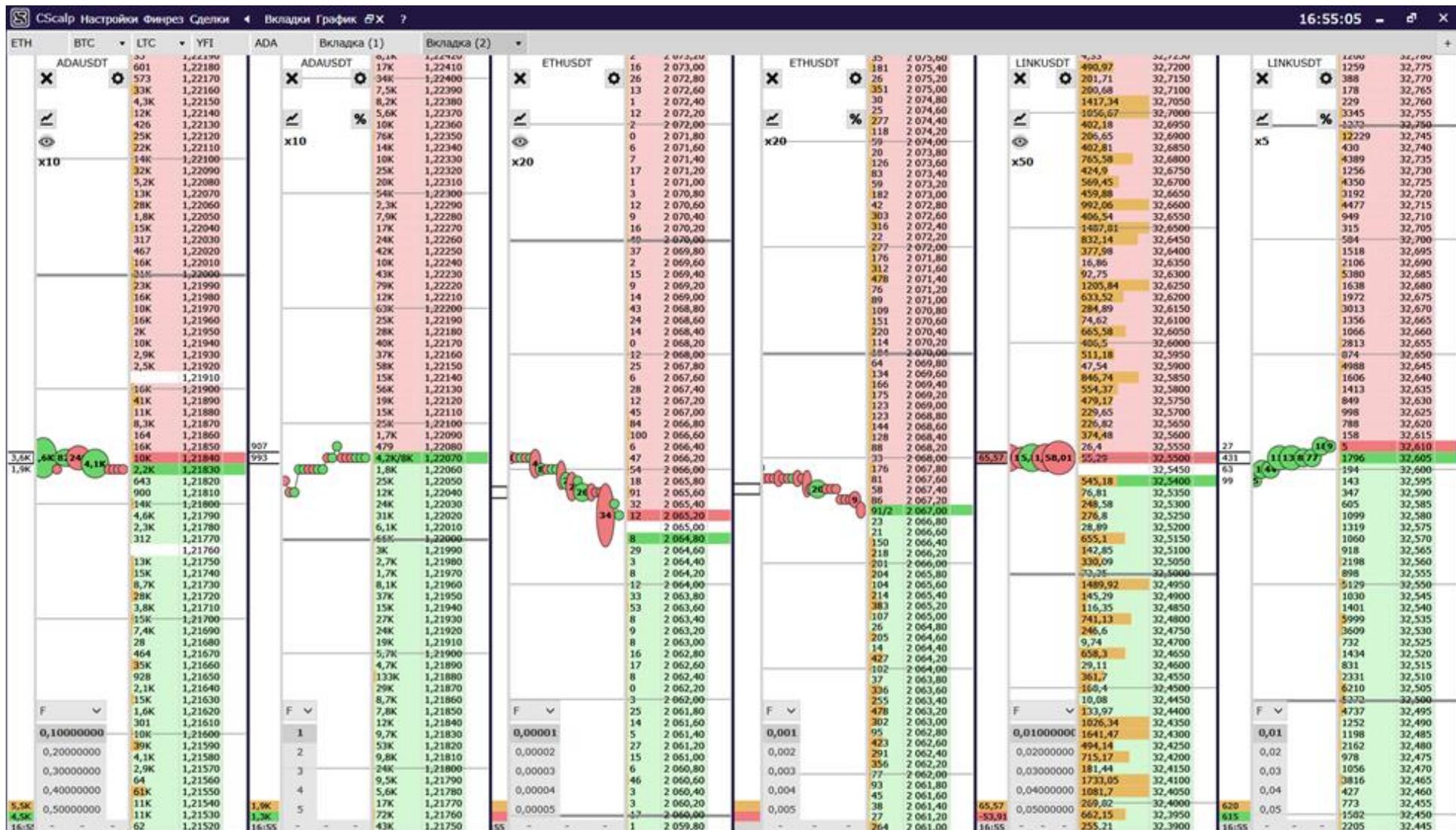


Рисунок 5 – Интерфейс терминала CScalp

Правая половина каждой из шести колонок отображает текущее содержимое очереди заявок («стакана») по конкретному выбранному инструменту. Скальперов интересует, когда в очереди заявок появляется необычно крупная по размеру заявка, что создает дисбаланс спроса и предложения, которым можно воспользоваться и спрогнозировать краткосрочное движение цены актива, однако, чтоб обнаружить такую ситуацию, нужно непрерывно наблюдать за всеми инструментами, общее количество которых можно оценить как превышающее 200. Для отображения сразу 200 стаканов не хватает, во-первых, мониторов (даже с учетом того, что нормальным для внутридневных трейдеров и скальперов считается иметь три-четыре монитора), во-вторых, вычислительной мощности компьютера для непрерывного обновления информации на экране, в-третьих, внимания пользователя.

Еще одной потенциально интересной ситуацией является такая, когда один из участников начинает через равные промежутки времени равным объемом продавать или покупать актив, меняя таким образом баланс спроса и предложения и двигая тем самым цену актива. Скальперы называют таких участников роботами. Но для того, чтоб заметить такого участника, нужно внимательно наблюдать за так называемой лентой (центральная часть каждой из шести колонок на рисунке 5), где овалами отображаются исполненные рыночные заявки. Здесь снова необходимо отметить невозможность наблюдения сразу за всеми инструментами.

Интерес для скальперов также представляют ситуации, когда цена актива растёт или падает на несколько процентов за короткий промежуток времени (1–5 минут), или когда происходит рост суммарного объёма рыночных сделок, проходящих по какому либо инструменту за фиксированный период времени (как правило, пять минут), в 4-5 раз по сравнению с его средним объемом за предыдущие периоды времени. Считается, что оба этих фактора свидетельствуют о наличии интереса участников к инструменту.

Исходя из сказанного выше, можно сформулировать проблему: трейдер упускает потенциально прибыльные ситуации из-за нехватки внимания и времени при визуальном анализе рыночных данных по многим инструментам сразу: крупные плотности в стаканах, роботы в ленте, резкий рост/падение цены, подход цены к уровню и т.д.

Для выявления причин проблемы удобно использовать диаграмму Ishikawa (Fishbone), представленную на рисунке 6, которая позволяет выявить причинно-следственные связи.

Описанные выше причины, такие, как нехватка внимания и концентрации пользователя и нехватка аппаратных средств его компьютера, могут быть дополнены следующими:

- Существующее ПО разработано преимущественно для торговли и имеет недостаточный аналитический функционал.
- То ПО, которое нацелено на анализ, производит такой анализ с использованием только истории цен активов, без возможности анализировать стакан и ленту.

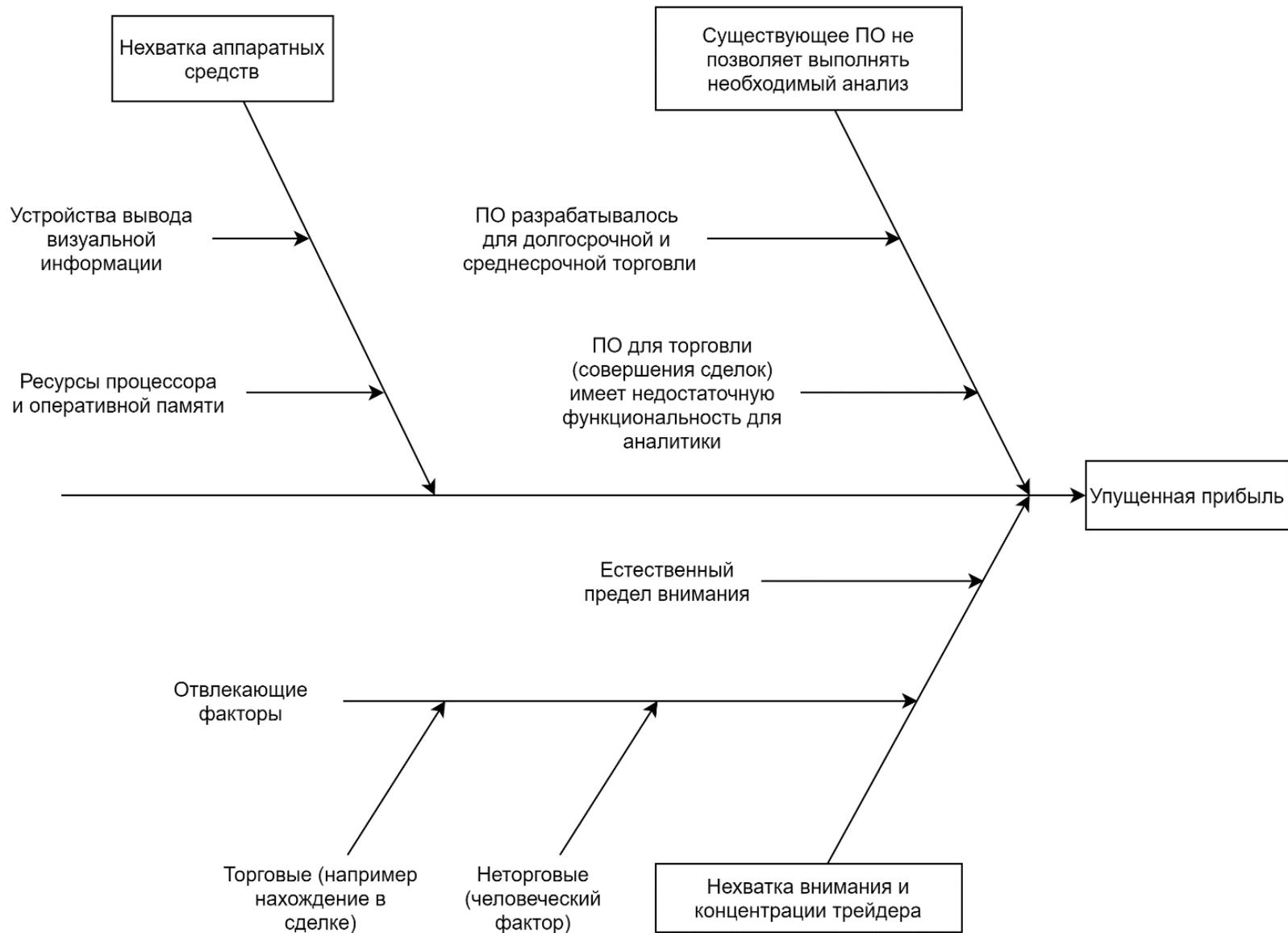


Рисунок 6 – Диаграмма Fishbone для выявления причин проблемы

## **2. Проектирование программного обеспечения**

### **2.1. Определение основного функционала приложения**

Для решения проблемы, описанной в предыдущей главе, разрабатываемое приложение должно производить отбор наиболее важной и актуальной информации из всего входящего потока и своевременно доводить её до пользователя в компактном виде. А именно оно должно:

- отображать заданное количество наиболее крупных плотностей на покупку и на продажу из очереди заявок в заданном диапазоне от спреда (полусуммы лучшей цены предложения и лучшей цены спроса) по всем указанным пользователем инструментам;
- оповещать пользователя голосовым сообщением и звуковым сигналом о появлении крупной плотности в заданном диапазоне от спреда;
- оповещать пользователя звуковым сигналом о появлении рыночной сделки заданного объема;
- оповещать пользователя звуковым сигналом о появлении робота с заданными параметрами по заданным инструментам;
- оповещать пользователя голосовым сообщением и звуковым сигналом об изменении цены инструмента за заданный промежуток времени на заданное число процентов;
- оповещать пользователя голосовым сообщением и звуковым сигналом о росте объема сделок на инструменте в заданное число раз по сравнению со средним объемом сделок за несколько предыдущих периодов.

### **2.2. Проектирование высокоуровневой архитектуры приложения**

В первом приближении задачу можно представить, как задачу реакции на входящие события и отображения на основе этого некоторой информации для пользователя (а также воспроизведения звуковых и голосовых сигналов), то есть как задачу событийно-ориентированного программирования.

При более детальном рассмотрении можно заметить, что для каждой из нескольких сотен криптовалютных пар, за которыми должно вестись наблюдение, существует одинаковый сценарий действий, суть которого заключается в выполнении следующего цикла:

1. Ждать входящего сообщения с биржи, содержащего обновленную информацию о каком-либо параметре инструмента (криптовалютной пары).
2. На основе полученной информации произвести какие-либо расчеты, преобразования.
3. Вывести преобразованную информацию куда-либо в виде (в том числе аудиальном), пригодном для восприятия пользователем. Перейти к пункту 1.

Как следует из описания, обработка входящей информации для каждого инструмента не зависит от обработки информации по другим инструментам, являясь отдельным логическим «потокom выполнения». Наиболее простым решением выглядит для обработки данных по каждому инструменту создать отдельный поток и в нём производить все необходимые операции. Однако следует учесть количество одновременно обрабатываемых инструментов, особенно с перспективой обрабатывать данные сразу от нескольких бирж, а также то, что по одному инструменту необходимо обрабатывать несколько разных потоков информации (сделки, обновления свечных данных, обновления стакана). Отсюда следует, что требуемое количество одновременно активных потоков составит от нескольких тысяч до нескольких десятков тысяч штук. А так как это потоки операционной системы, то значительное количество времени будет уходить на операции переключения контекста между потоками, что негативно отразится на общей производительности системы вплоть до того, что всё процессорное время будет уходить на переключение потоков.

К счастью, характер разрабатываемой системы таков, что каждый логический поток большую часть времени проводит в ожидании входящих по

сети данных, то есть ожидает завершения операции ввода-вывода. Время же, потребное на обработку входящих данных, во много раз меньше среднего времени ожидания новых данных. Таким образом, может быть применена асинхронная модель обработки, что позволит избавиться от множества потоков операционной системы, заменив их функциями обратного вызова, которые будут вызываться операционной системой по факту возникновения какого-либо события, например окончания чтения из сокета. Обработка в этом случае может производиться всего лишь одним потоком, снижая непроизводительные затраты на переключение контекста почти до нуля, а также значительно упрощая синхронизацию. Минус использования одного потока состоит в отсутствии масштабируемости при увеличении нагрузки: большинство современных процессоров обладают более, чем одним, физическим ядром – типичное количество 4 или 8 ядер. И на протяжении последних лет это число также растёт.

Для максимальной эффективности использования ресурсов процессора следует использовать число потоков операционной системы, равное числу ядер процессора, а в каждом из таких потоков обработку производить асинхронно. В идеальном случае это позволяет также избавиться от слишком большого числа переключений контекста. Минусом является возросшая сложность синхронизации по сравнению с однопоточным вариантом.

Прототип приложения, разработанный на языке программирования Python с использованием библиотеки `asyncio`[7] для реализации асинхронной обработки данных, показал, что при использовании лишь одного ядра процессора (Python исполняет программу в одном потоке; можно было бы использовать модуль `multiprocessing`[8], который позволяет организовать выполнение программы с помощью нескольких процессов, задействовав таким образом все ядра, но взаимодействие между процессами было гораздо медленнее, чем между потоками в языках, где такая возможность поддерживается; это нивелировало бы выигрыш от использования нескольких процессов) с ростом числа обрабатываемых инструментов быстро наступает

нехватка вычислительной мощности. При этом большая часть вычислений затрачивается не на непосредственно полезные расчеты, а на расшифровку SSL трафика и парсинг данных в формате JSON. Была попытка использовать библиотеку для парсинга JSON, написанную на C, но выигрыш в скорости от её использования терялся из-за необходимости конвертации структур данных на границе языков C-Python.

С учетом сказанного выше, было принято решение разделить приложение на две части: условно серверную, которая будет отвечать за приём данных с бирж и их эффективную обработку, и клиентскую, которая будет запускаться в браузере и полностью отвечать за взаимодействие с пользователем и вывод обработанной информации, полученной от серверной части. Клиентская часть, выполненная в браузере, позволит быстрее и проще реализовать пользовательский интерфейс. Серверная часть условна, так как будет запускаться на том же устройстве, что и клиентская, в целях, во-первых, сокращения сетевых задержек, во-вторых, упрощения взаимодействия между двумя частями (не нужно реализовывать обработку сетевых ошибок).

Для реализации серверной части решено применить C++ и библиотеку Boost.Asio[9], которая реализует многопоточную асинхронную обработку данных, в том числе сетевой ввод-вывод – практически идеальное решение поставленной задачи.

### **2.3. Проектирование серверной части**

Для представления архитектуры серверной части была выбрана диаграмма DFD (data flow diagram, диаграмма потоков данных). Этот вид диаграмм используется для описания бизнес-процессов, графического представления системы. DFD имеет дело только с бизнес-процессами и потоками данных. В диаграммах используется 4 символа:

- внешняя сущность – для описания интерфейсов;

- накопитель данных – всё, что может сохранять в себе данные (очереди, базы данных и т.п.);
- процесс;
- поток данных.

На рисунке 7 представлена контекстная диаграмма, построенная для серверной части разрабатываемой системы. Она позволяет оценить границы системы, её интерфейсы (с чем и каким образом она взаимодействует).

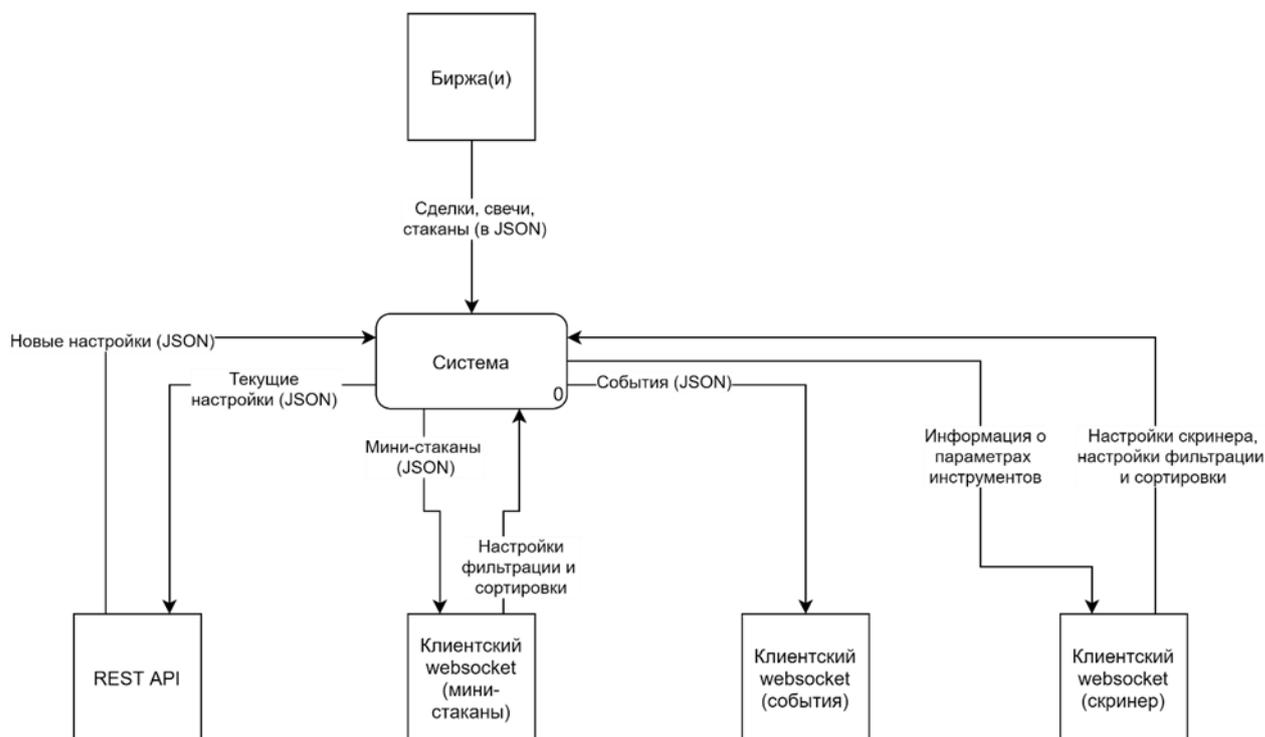


Рисунок 7 – Контекстная диаграмма

На рисунке 8 приведена диаграмма функциональной декомпозиции, представляющая систему в виде дерева процессов, функций и подфункций. Дальнейшая декомпозиция отражена на рисунках 9-19.

На рисунке 9 показано, как клиентская часть может получать текущие значения настроек и вносить в настройки изменения.

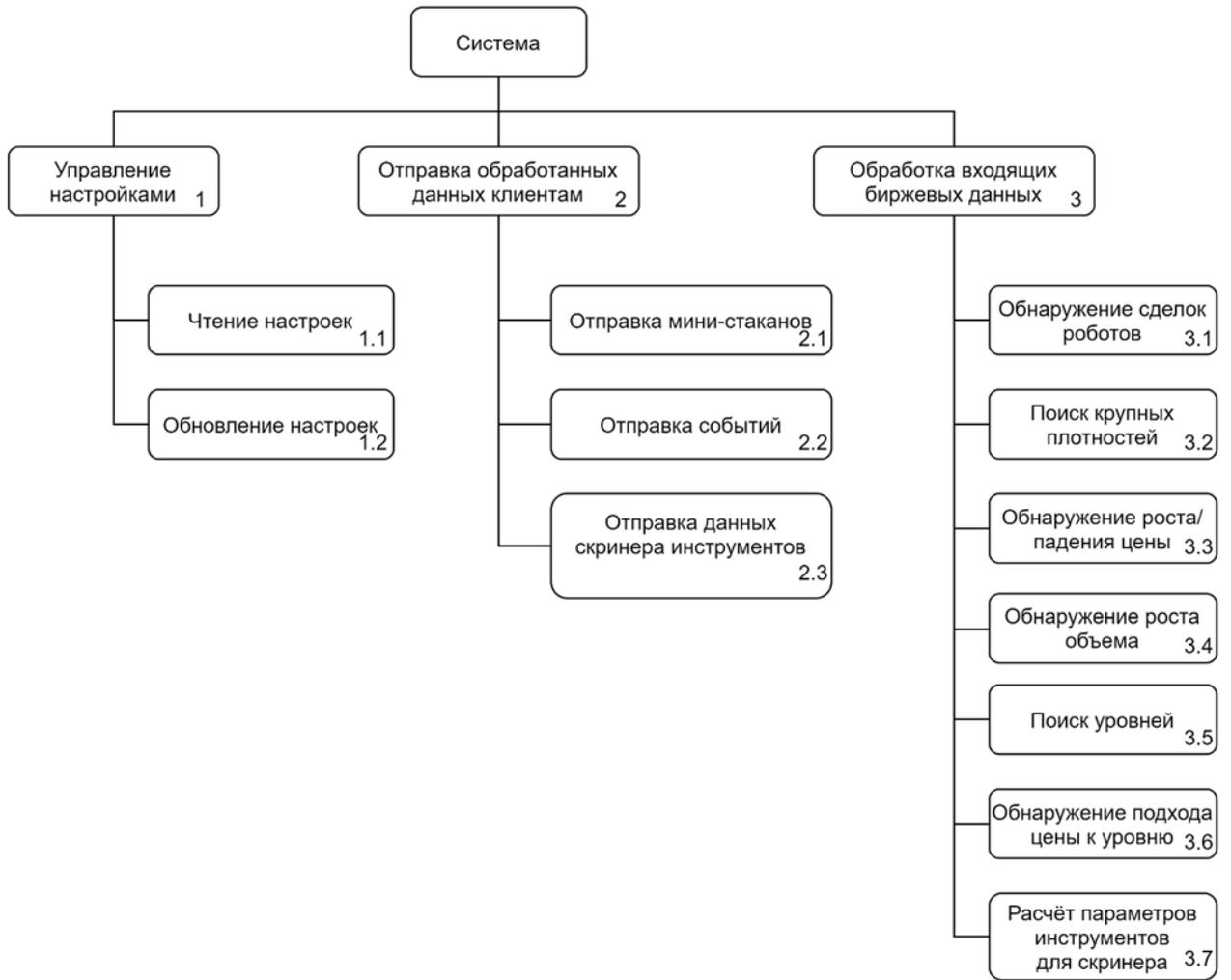


Рисунок 8 – Диаграмма функциональной декомпозиции

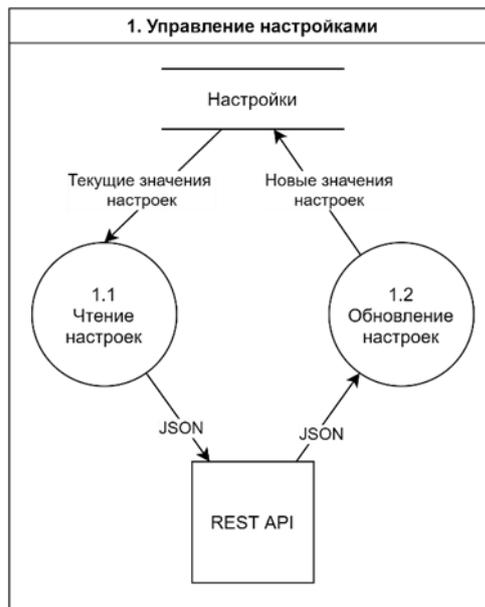


Рисунок 9 – Декомпозиция «Управление настройками»

Как следует из рисунка 10, на клиентскую часть (браузер) отправляются не все мини-стаканы, а только та их часть, которая соответствует параметрам фильтрации, заданным пользователем. Также сортировка отфильтрованных мини-стаканов происходит на стороне сервера. Это позволяет снять значительную нагрузку с браузера.

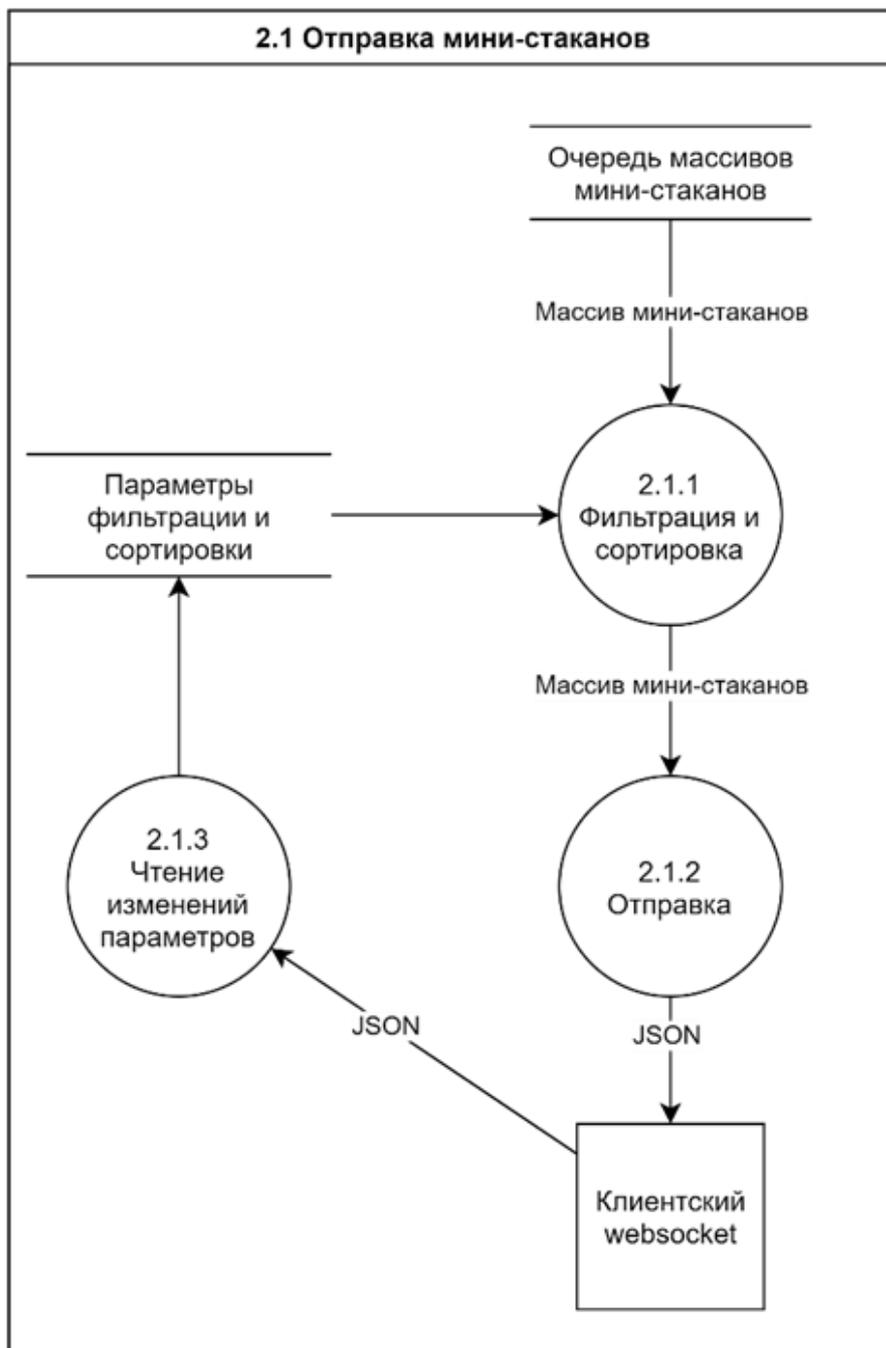


Рисунок 10 – Декомпозиция «Отправка мини-стаканов»

Схожим образом, как показано на рисунке 11, происходит фильтрация и сортировка данных для скринера инструментов (о его функциональности

рассказано в разделе «Обзор основных возможностей приложения»). Это предоставляет те же преимущества в виде сниженной нагрузки на клиентскую часть.

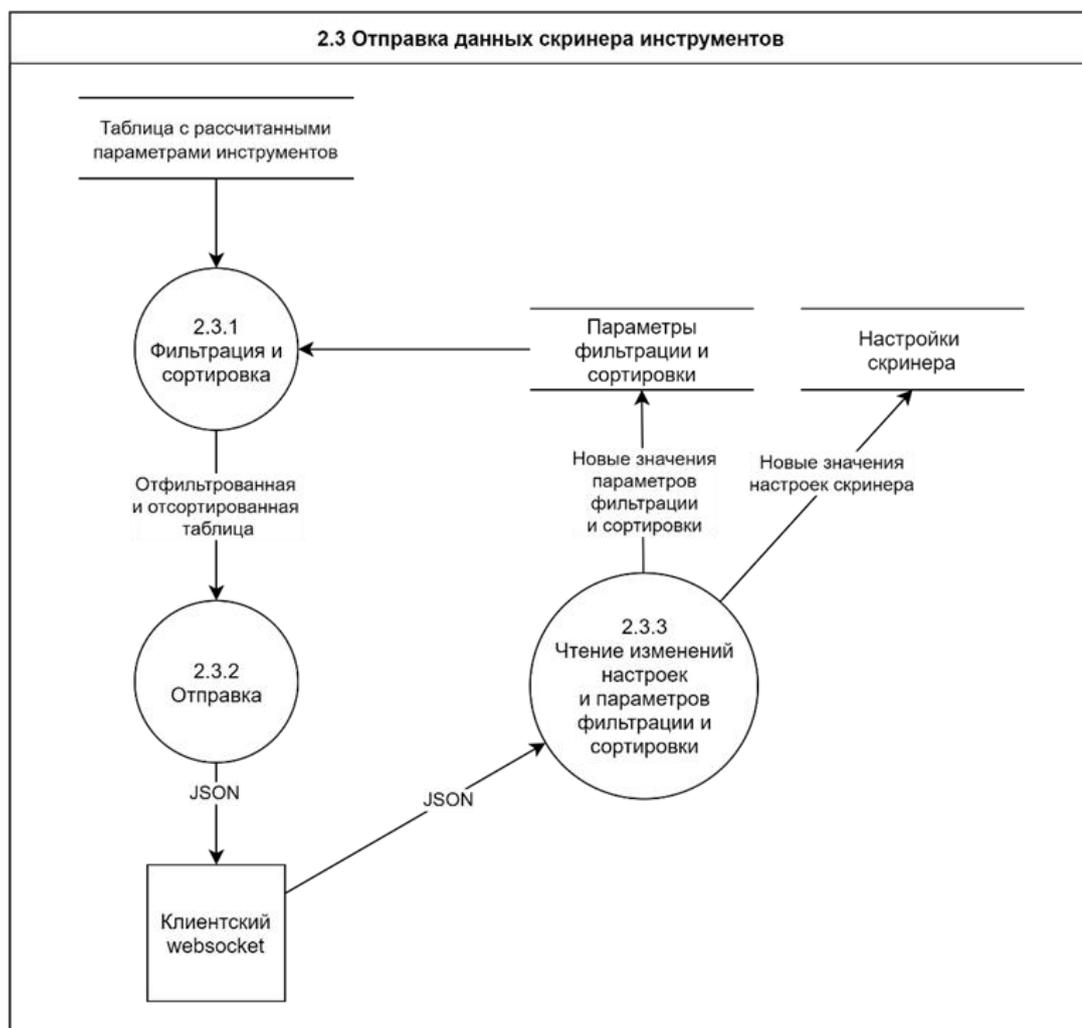


Рисунок 11 – Декомпозиция «Отправка данных скринера инструментов»

Отправка событий на клиентскую часть, отображенная на рисунке 12, является тривиальным чтением из очереди и записью прочитанного в websocket.

Процесс обнаружения сделок роботов, представленный на рисунке 13, использует известные параметры торговых роботов, такие как объем одной сделки и время между сделками, заданные в настройках, чтобы во входящем с бирже потоке сделок отыскивать сделки, совершаемые такими роботами, и при обнаружении таковых генерировать соответствующие события.

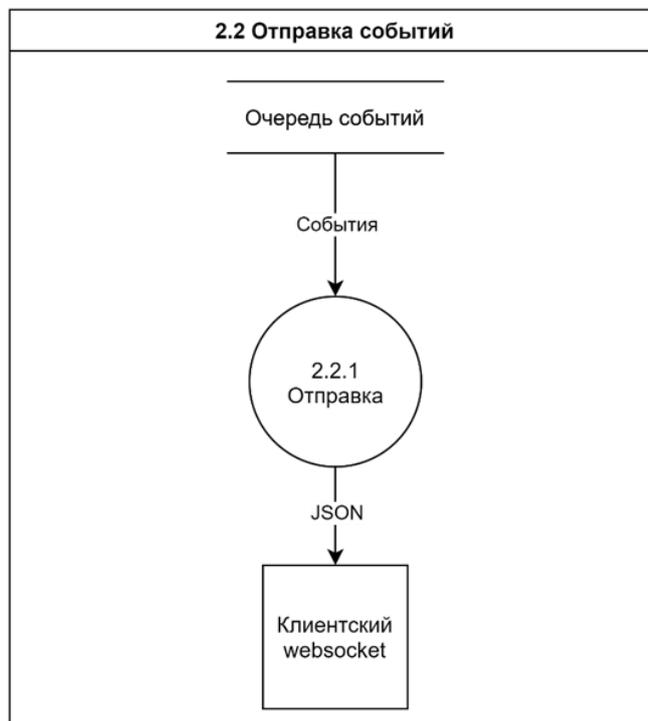


Рисунок 12 – Декомпозиция «Отправка событий»

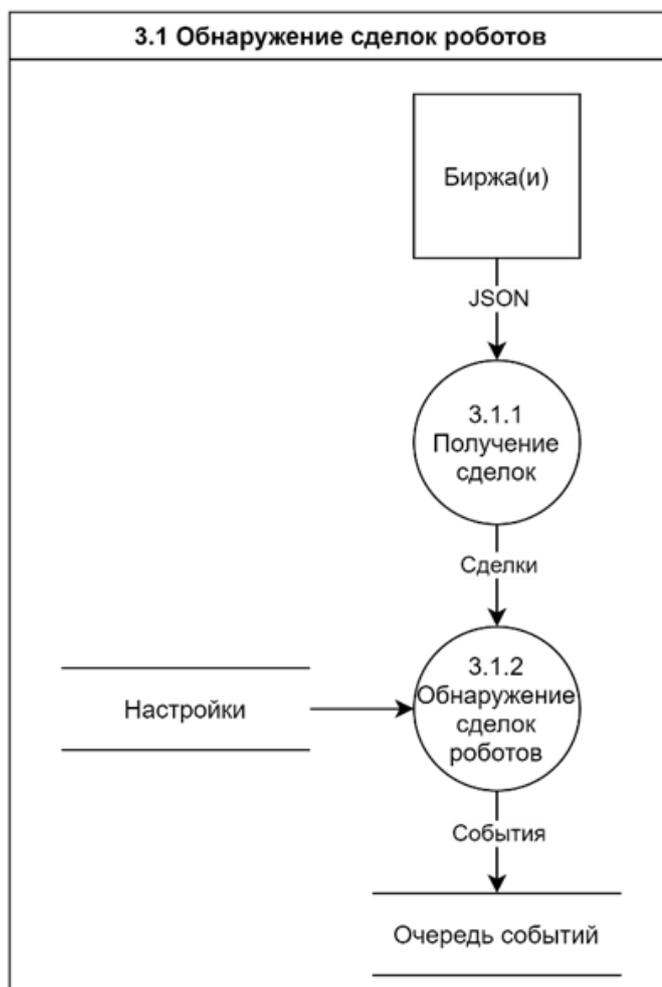


Рисунок 13 – Декомпозиция «Обнаружение сделок роботов»

Процессы обнаружения роста/падения цены и обнаружения роста объема, представленные на рисунках 14 и 15, соответственно, во многом схожи – оба используют в качестве данных для расчета только свечи, получаемые с биржи.

Отличие обнаружения роста/падения цены состоит в том, что размер (в процентах относительно цены открытия текущей свечи) текущей формирующейся свечи сравнивается с заданным в настройках фиксированным числом, в то время как для обнаружения роста объема используется сравнение объема формирующейся свечи со средним объемом нескольких (количество задается в настройках) свечей перед ней (то есть текущая свеча в расчет среднего объема не включается).

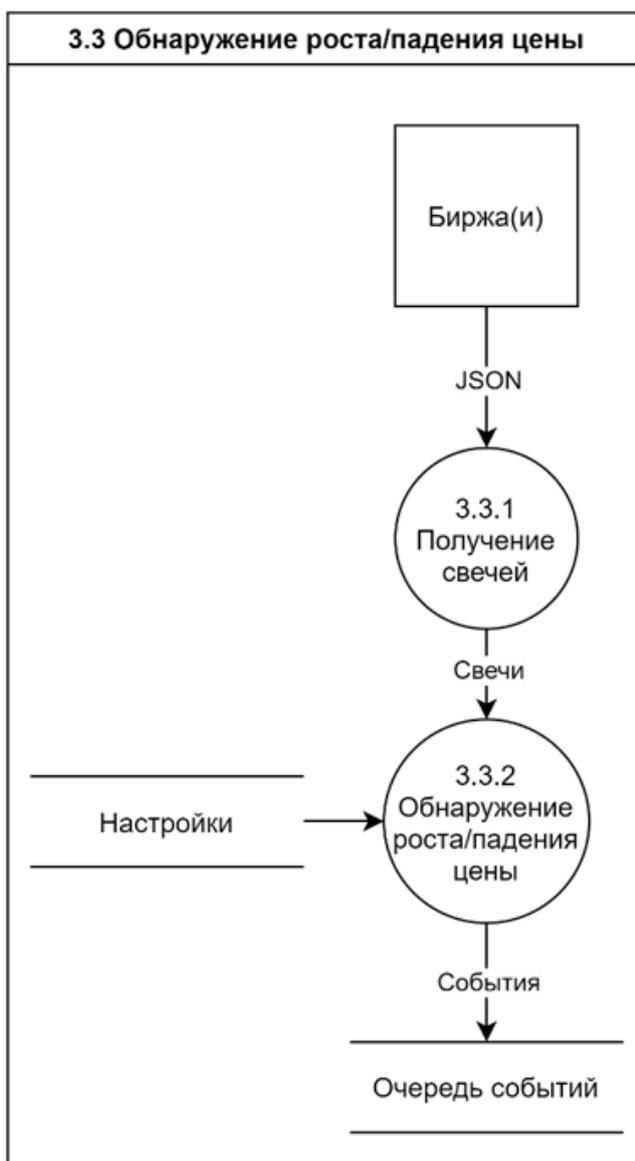


Рисунок 14 – Декомпозиция «Обнаружение роста/падения цены»

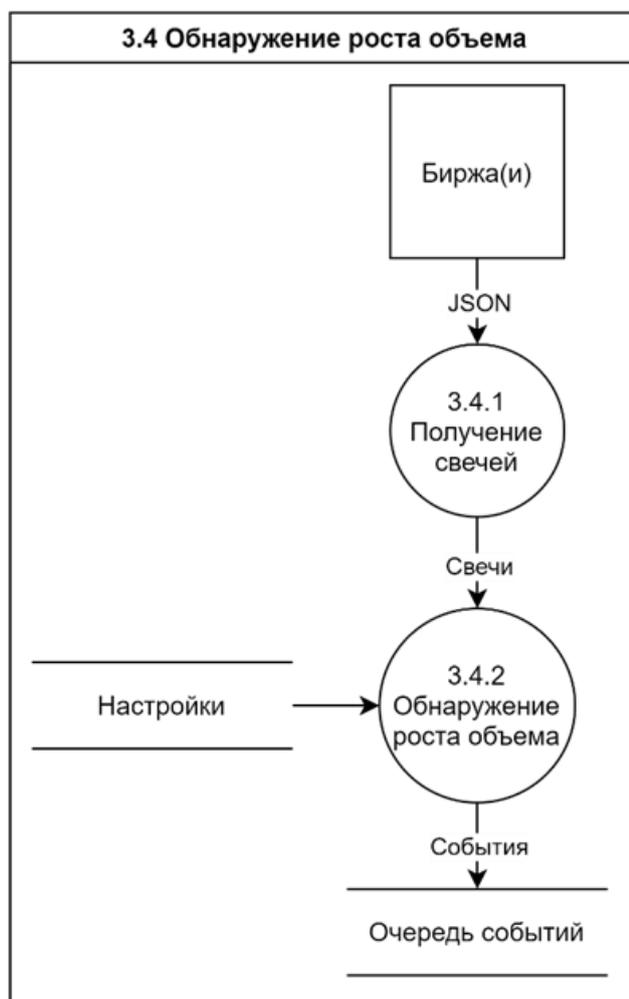


Рисунок 15 – Декомпозиция «Обнаружение роста объема»

В процессе поиска крупных плотностей, как показано на рисунке 16, используется несколько видов биржевых данных: стаканы, свечи, сделки. Стаканы используются непосредственно для поиска плотностей. Сделки – для определения, были ли совершены с найденными плотностями сделки в прошлом. Свечи – для расчета степени крупности плотностей в одном из способов расчета.

Поиск крупных плотностей происходит по каждому инструменту в отдельности, а процесс «Агрегация мини-стаканов» используется, чтобы свести мини-стаканы, содержащие найденные плотности, в единый массив.

Поиск уровней приведен на рисунке 17. Он выполняется с некоторой периодичностью, которая задаётся пользователем в настройках. Рекомендуемая периодичность от 20 до 60 минут, так как минимальный таймфрейм поиска уровней – час, поэтому чаще искать смысла нет.

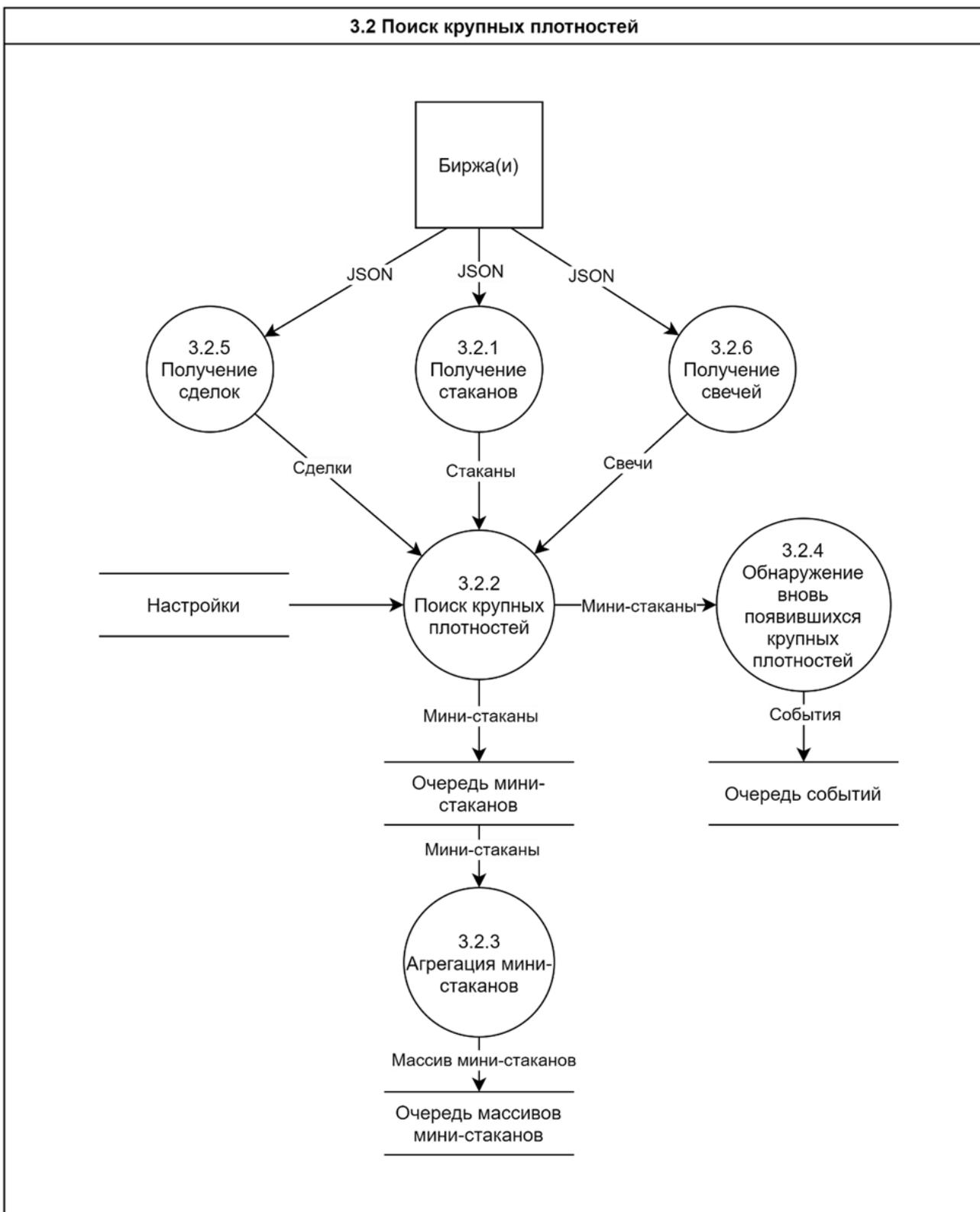


Рисунок 16 – Декомпозиция «Поиск крупных плотностей»

Обнаружение факта подхода цены к уровню реализуется в процессе, показанном на рисунке 18.

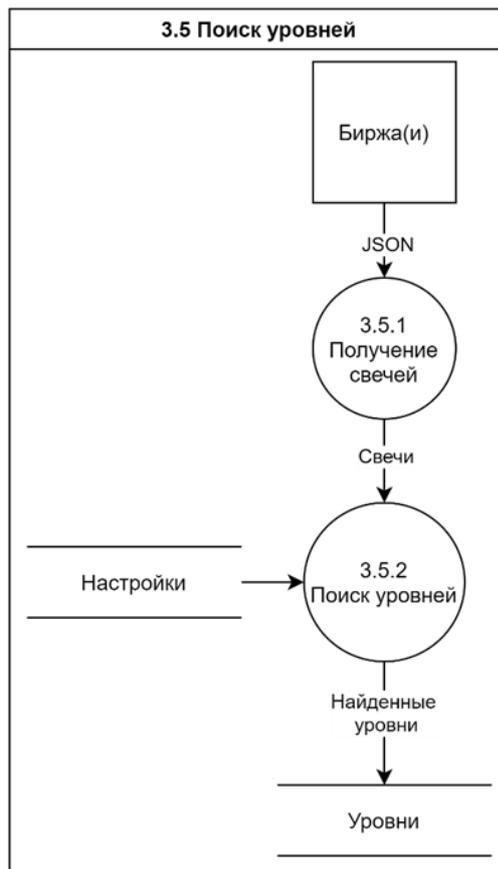


Рисунок 17 – Декомпозиция «Поиск уровней»

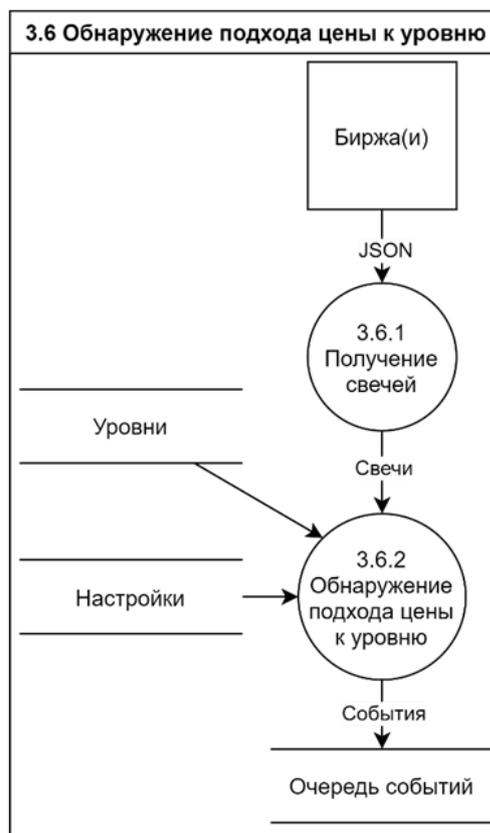


Рисунок 18 – Декомпозиция «Обнаружение подхода цены к уровню»

«Скринер», как показано на рисунке 19, не использует общие настройки. Вместо этого используются локальные «Настройки скринера». Связано это с тем, что «скринеров» может быть несколько, каждый со своими настройками. Однако уровни, найденные в процессе, описанном выше, используются для всех «скринеров» общие.

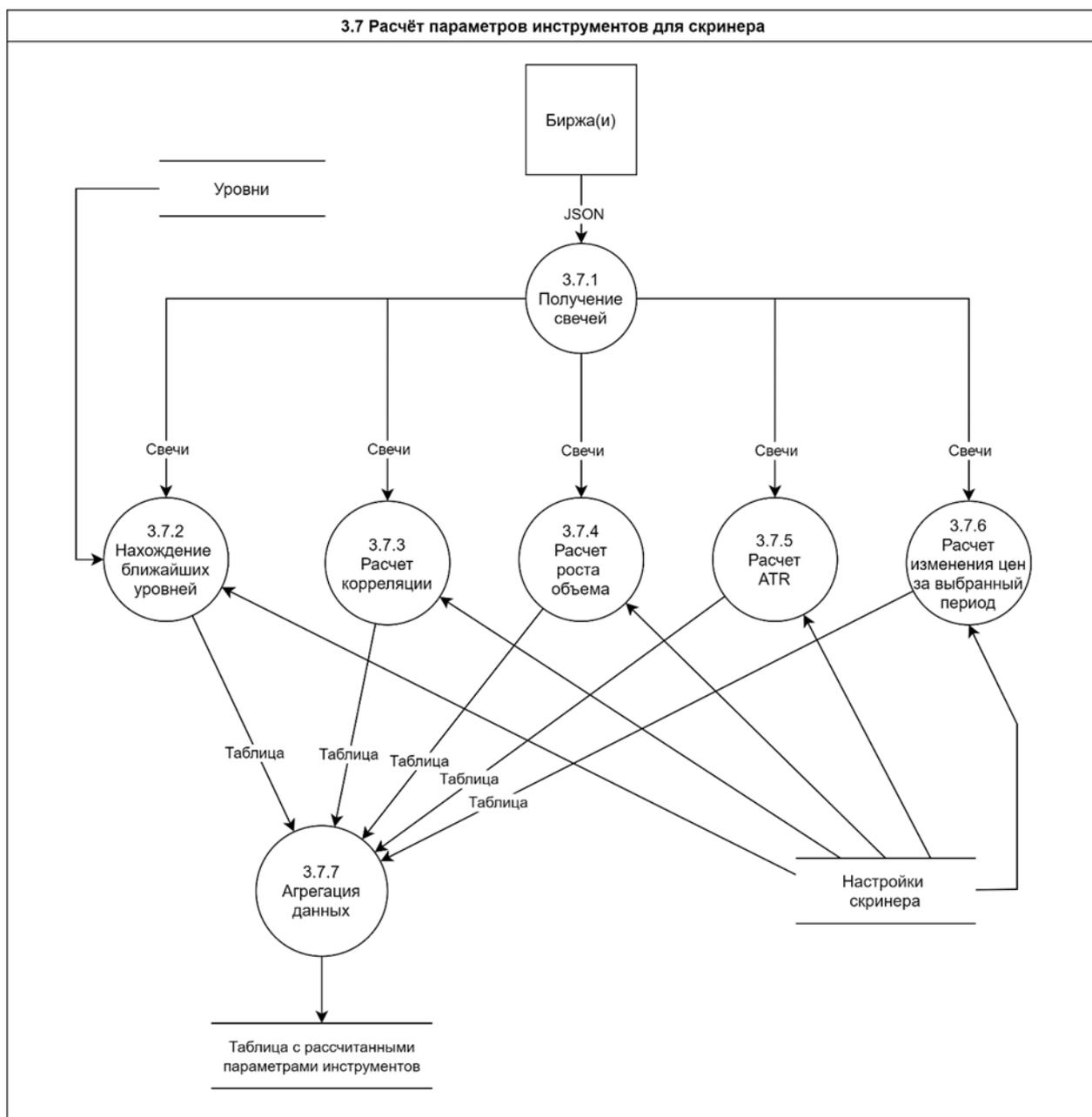


Рисунок 19 – Декомпозиция «Расчет параметров инструментов для скринера»

Полученная диаграмма позволяет перейти к проектированию на более низком уровне и в дальнейшем к разработке.

### 3. Разработка программного обеспечения

#### 3.1. Разработка серверной части

Асинхронное программирование на C++ часто подразумевало использование функций обратного вызова («коллбэков»), однако такой подход отличался значительной сложностью в разработке и поддержке, так как программисту нужно было самостоятельно разбивать код в точках вызова асинхронных функций на отдельные процедуры, а также следить за сохранением состояния до и после асинхронного вызова. Проблема получила название «ада коллбэков» (англ. «callback hell»[10]) и существует не только в языке C++.

Стандарт C++20 предлагает путь решения этой проблемы, схожий с тем, который предлагается, например, языком Python или C# - использование сопрограмм[11] (или же корутин) вместо коллбэков. Такой подход позволяет сохранить удобочитаемость кода, воспользовавшись при этом всеми преимуществами асинхронного программирования.

Основой разрабатываемой системы станет, как упоминалось в предыдущей главе, библиотека Boost.Asio[9]. Её преимуществами являются:

- поддержка асинхронного сетевого ввода-вывода;
- поддержка таймеров;
- поддержка многопоточной обработки;
- долгое время существования;
- лицензия Boost Software License;
- использование в большом числе проектов.

Для повышения уровня абстракции и обработки таких протоколов, как HTTP/HTTPS, Websocket будет использоваться библиотека Boost.Beast[12], которая функционирует поверх Boost.Asio. Её особенности:

- симметричность алгоритмов относительно роли: клиент или сервер;
- простота использования для пользователей, знакомых с Boost.Asio;

- гибкость: программист принимает важные решения, такие как использование буферов или многопоточности;
- производительность: возможность создания приложений, обрабатывающих тысячи соединений;
- хорошая база для построения поверх неё будущих абстракций.

Для поддержки защищенных протоколов, которые необходимы для работы с API бирж (HTTPS, WebSocket Secure), будет использоваться библиотека криптографии OpenSSL[13] 3.0.1. Версия 3.0 распространяется по более простой лицензии Apache 2.0 в сравнении с версией 1.1.1, бывшей долгое время основной. Boost.Beast и Boost.Asio по умолчанию имеют интеграцию с OpenSSL. Некоторым недостатком является нестандартный процесс сборки данной библиотеки из исходного кода.

Информация с бирж приходит в формате JSON, для его быстрого превращения в программные структуры данных необходим быстрый парсер JSON. На роль такой библиотеки отлично подходит simdjson[14], которая является самым быстрым парсером JSON за счёт использования векторных инструкций (SIMD)[15].

Особенностью биржи Vinance является то, что дробные числа отправляются в виде строк, чтобы избежать проблем, связанных с потерей точности при использовании чисел с плавающей запятой, что является важным, когда передаются денежные значения. Для быстрого преобразования строкового представления этих чисел в числовое применяется библиотека fast\_float[16].

Для представления «стаканов» внутри программы требуется словарь, ключом которого являлась бы цена, а значением – количество актива по данной цене. unordered\_map из стандартной библиотеки языка C++ является достаточно медленным решением. Кроме того, на хранение каждой записи в таком словаре приходится отдельное выделение памяти, что потенциально может привести к снижению производительности при последовательном переборе значений словаря из-за промахов кэша. Перебор значений словаря нужен, в частности, при

поиске наибольших заявок. Библиотека `robin_hood`[17] предлагает свой вариант `unordered_map` и `unordered_set`, позволяя, кроме того, выбрать будут ли данные храниться в одном непрерывном массиве (значительно снижает количество промахов кэша, но увеличивает максимальное время вставки элемента) или в виде отдельных узлов (как в классической реализации `unordered_map`). `robin_hood` является одной из самых быстрых библиотек, согласно тестам производительности[18], и потребляет значительно меньше памяти, чем `std::unordered_map`, что становится весомым фактором при наличии нескольких сотен таких словарей, каждый из которых может содержать по несколько тысяч записей.

Нельзя обойти стороной вопрос удобного логирования, для которого будет применяться библиотека `spdlog`[19], которая обеспечивает необходимую производительность, поддерживает богатое форматирование, вывод в файл, вывод по сети, асинхронное логирование.

Вопрос хранения данных в разрабатываемой системе состоит из двух частей: хранения биржевых данных и хранения настроек.

Характерной чертой биржевых данных является необходимость быстрого доступа к ним (имеется ввиду как низкая задержка, так и высокая пропускная способность) и их относительно небольшой размер (нет нужды хранить минутные графики за последние несколько лет, достаточно нескольких часов). Еще одним условием является факт исполнения разрабатываемого приложения непосредственно на клиентском устройстве, что исключает использование «полновесных» СУБД, таких как MySQL или PostgreSQL. В качестве решения рассматривалось использование SQLite, но даже в режиме размещения базы данных в памяти скорость доступа к данным оказывалась недостаточной. В связи с этим решено было использовать простое хранение биржевых данных в оперативной памяти без использования СУБД, что позволило достичь необходимой скорости доступа.

Доступ к настройкам необходим как серверной, так и клиентской (браузер) части. Скорость доступа всё ещё играет значительную роль. Остаётся ограничение, связанное с исполнением приложения непосредственно на пользовательском устройстве. Кроме того, настройки предполагают иерархическую структуру. Все обозначенные ограничения приводят к решению хранить настройки в виде JSON-структуры в оперативной памяти. JSON подразумевают иерархическую структуру данных, является нативным форматом для браузера, легко сериализуется и десериализуется, что позволяет сохранять настройки в виде простого файла на диске.

Для унифицированной обработки данных с разных бирж разработана концепция потока данных. Поток данных – это способ приёма изменяющихся во времени данных с помощью последовательного чтения обновлений этих данных. Обновлениями для стакана будут изменяющиеся во времени состояния стакана. Для сделок – новые произошедшие сделки. Для свечей – информация об изменении текущей (формирующейся свечи). В разрабатываемой системе один поток данных (к примеру, обновления стакана по инструменту BTCUSDT на бирже Vinance) может разделяться между несколькими потребителями. Такой подход позволяет не создавать несколько физических подключений для получения одних и тех же данных, что в свою очередь позволяет использовать наименьшее количество трафика и снизить требования к пропускной способности сети. Внутри системы, когда данные уже получены, они могут дублироваться между разными потребителями (могут дублироваться как сами данные для небольших сущностей вроде обновлений свечей и сделок, так и только указатель на данные, если они достаточно объемные, например для обновлений стаканов), чтобы каждый потребитель мог обрабатывать входящие обновления в индивидуальном темпе, и чтоб при этом потребителям не нужно было синхронизировать между собой глобальное состояние.

На рисунке 20 представлены интерфейсы, с помощью которых осуществляется доступ к потокам данных. Суффикс «View» в названии каждого

интерфейса означает, что это не сам поток данных непосредственно, а лишь разделяемый вид на него. В дальнейшем для краткости «вид на поток данных» будет именоваться как «поток данных».

`ITradeStreamView` – поток сделок. `ICandleStreamView` – поток обновлений свечных данных, но позволяет также получать информацию о свечах в прошлом (то есть о прошлых ценах инструмента). `IOrderBookStreamView` – поток «снимков» состояния стакана в определенные моменты времени.

Интерфейс `IQuittable` позволяет уведомить поток данных посредством вызова метода `SendQuit` о том, что не нужно больше ожидать обновлений и следует начать завершать внутренние процессы (в случае, если таковые имеются для данной реализации потока данных). Задача `WaitForQuit` используется для ожидания полного завершения внутренних процессов, если таковые имеются.

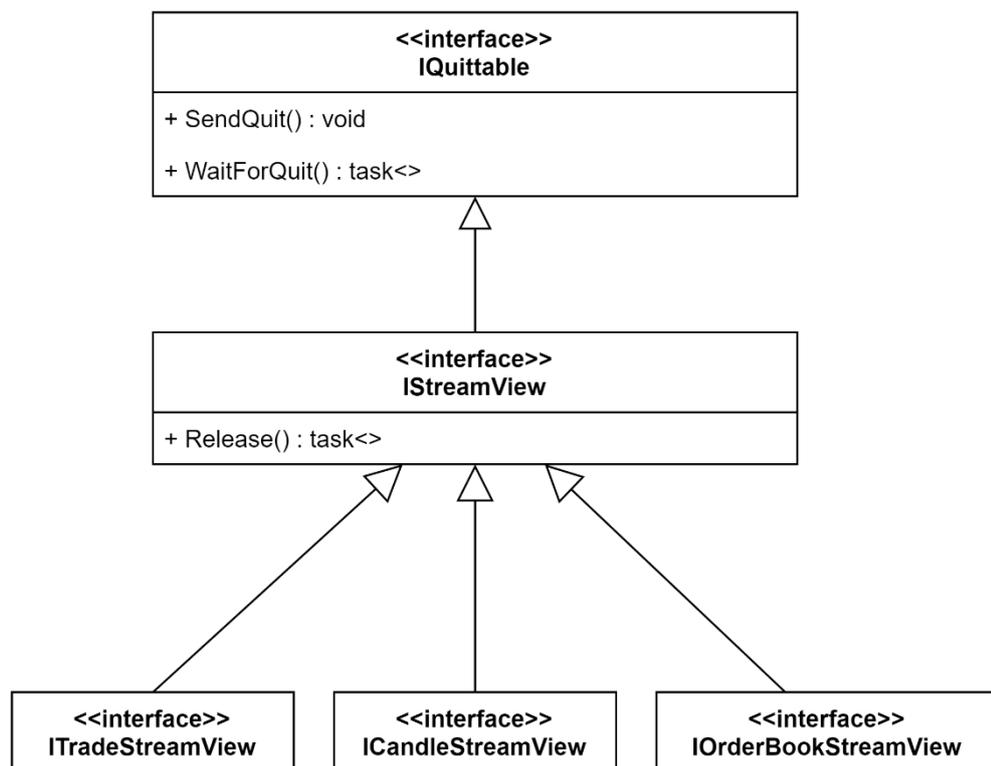


Рисунок 20 – Иерархия интерфейсов потоков данных

Интерфейс `IStreamView` предоставляет единственный метод-задачу `Release`, который вызывается при завершении работы с потоком данных.

Конкретная реализация переопределяет этот метод, чтобы произвести необходимые операции по прекращению получения данных с биржи (это отличается от «внутренних процессов», упомянутых выше). Если поток данных используется несколькими потребителями, то прекращение получения данных с биржи произойдет только, когда последний из потребителей вызовет Release.

Как показано на рисунке 21, интерфейс ITradeStreamView предоставляет метод GetNext, который ждет появления очередной сделки на бирже и возвращает информацию о ней. Ожидание может быть прервано вызовом метода SendQuit, в этом случае метод GetNext завершается досрочно, бросая исключение operation\_cancelled.

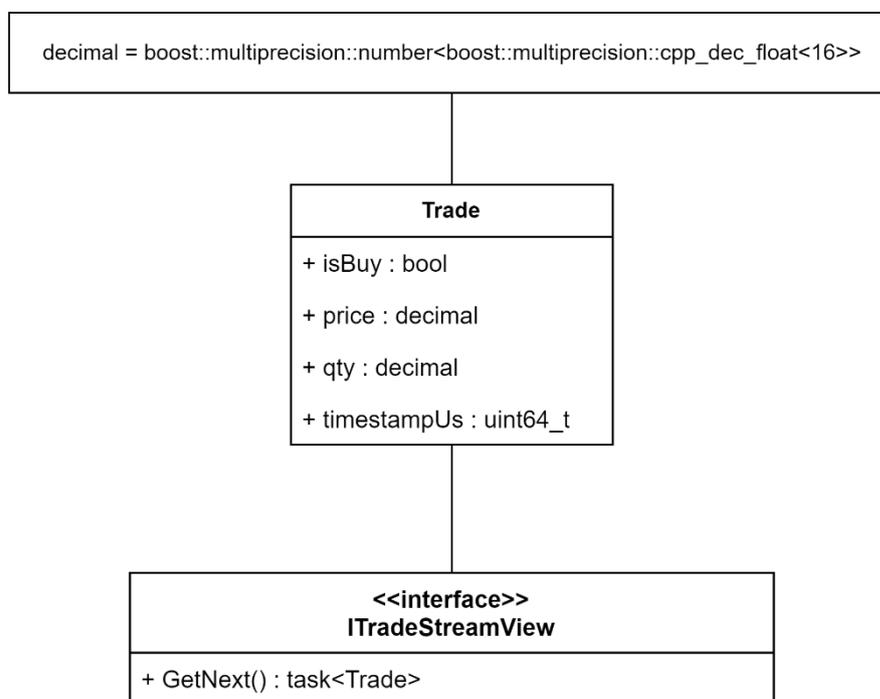


Рисунок 21 – Детализация ITradeStreamView

Информация о сделке представлена структурой Trade, содержащей следующие поля:

- isBuy – является ли сделка покупкой;
- price – цена (в единицах котируемого актива), по которой была совершена сделка;

- qty – размер сделки (в единицах базового актива);
- timestampUs – биржевое время совершения сделки Unix timestamp с точностью до микросекунд.

Чтоб избежать ошибок округления, возникающих при арифметических операциях над числами с плавающей запятой, для полей price и qty используется тип с фиксированной запятой с точностью 16 знаков после запятой из библиотеки Boost.Multiprecision.

На рисунке 22 представлена детализация интерфейса IOrderBookStreamView. Метод GetNext ожидает следующего обновления стакана и возвращает ссылку на текущее его состояние. После успешного вызова GetNext можно также узнать, какое обновление привело к данному состоянию, вызвав метод GetLastUpdate, который вернет информацию об обновлении стакана.

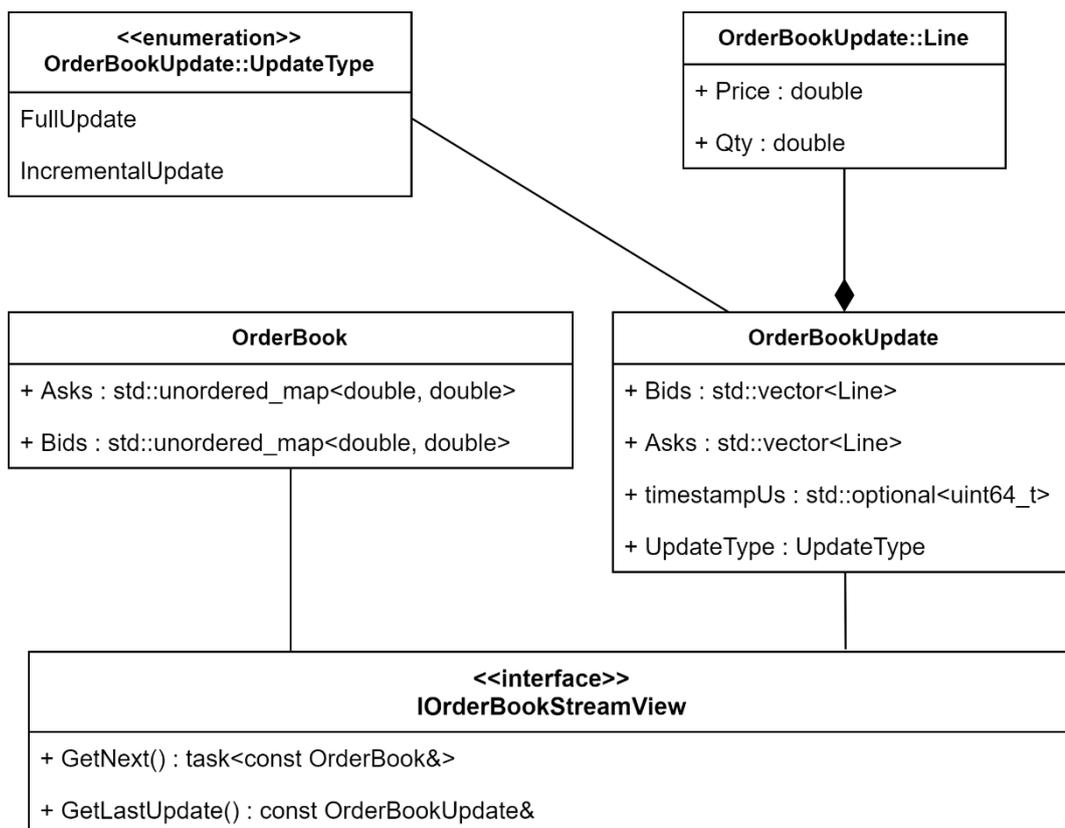


Рисунок 22 – Детализация IOrderBookStreamView

Обновление стакана (OrderBookUpdate) содержит следующие поля:

- Bids – изменившиеся заявки на покупку;
- Asks – изменившиеся заявки на продажу;
- timestampUs – опциональное (некоторые биржи не предоставляют такую информацию) биржевое время обновления стакана, Unix timestamp с точностью до микросекунд;
- UpdateType – тип обновления: полное или инкрементальное (в последнем случае обновление содержит только те строки стакана, где поменялась сумма заявок).

Информация о полном стакане представляет собой информацию о сумме заявок на покупку или на продажу по каждой из цен, хранимую в `std::unordered_map`, где ключом выступает цена, а значением – сумма заявок по этой цене.

На рисунке 23 представлена детализация интерфейса `ICandleStreamView`, отвечающего за работу со свечными данными. Метод `GetNext` ожидает очередного обновления текущей свечи и возвращает её. Свеча представлена при помощи структуры `Candle`, содержащей следующие поля:

- Open – цена открытия;
- High – наибольшая цена;
- Low – наименьшая цена;
- Close – цена закрытия;
- Volume – объем совершенных сделок (в единицах базового актива);
- StartTimeUs – биржевое время открытия свечи, Unix timestamp с точностью до микросекунд

После успешного вызова `GetNext` можно получить свечи в прошлом при помощи методов `GetCandles`. Метод, принимающий один аргумент, возвращает количество последних свечей, указанное в этом аргументе. Метод, принимающий два аргумента, возвращает свечи, находящиеся в указанном промежутке времени. Последовательность свечей представляется с помощью

структуры CandleLine, содержащей поля Period и Data. Period указывает на интервал времени свечей, а Data содержит непосредственно свечи. Для хранения свечей используется flat\_map из библиотеки Boost.Container, это обеспечивает доступ к свечам как по индексу, подобно массиву, так и по ключу, которым выступает время открытия свечи. Во flat\_map элементы отсортированы по возрастанию значения ключа. Кроме того, данные хранятся непрерывно в едином участке памяти, что улучшает их локальность, повышая таким образом производительность при последовательном чтении за счет снижения количества промахов кэша.

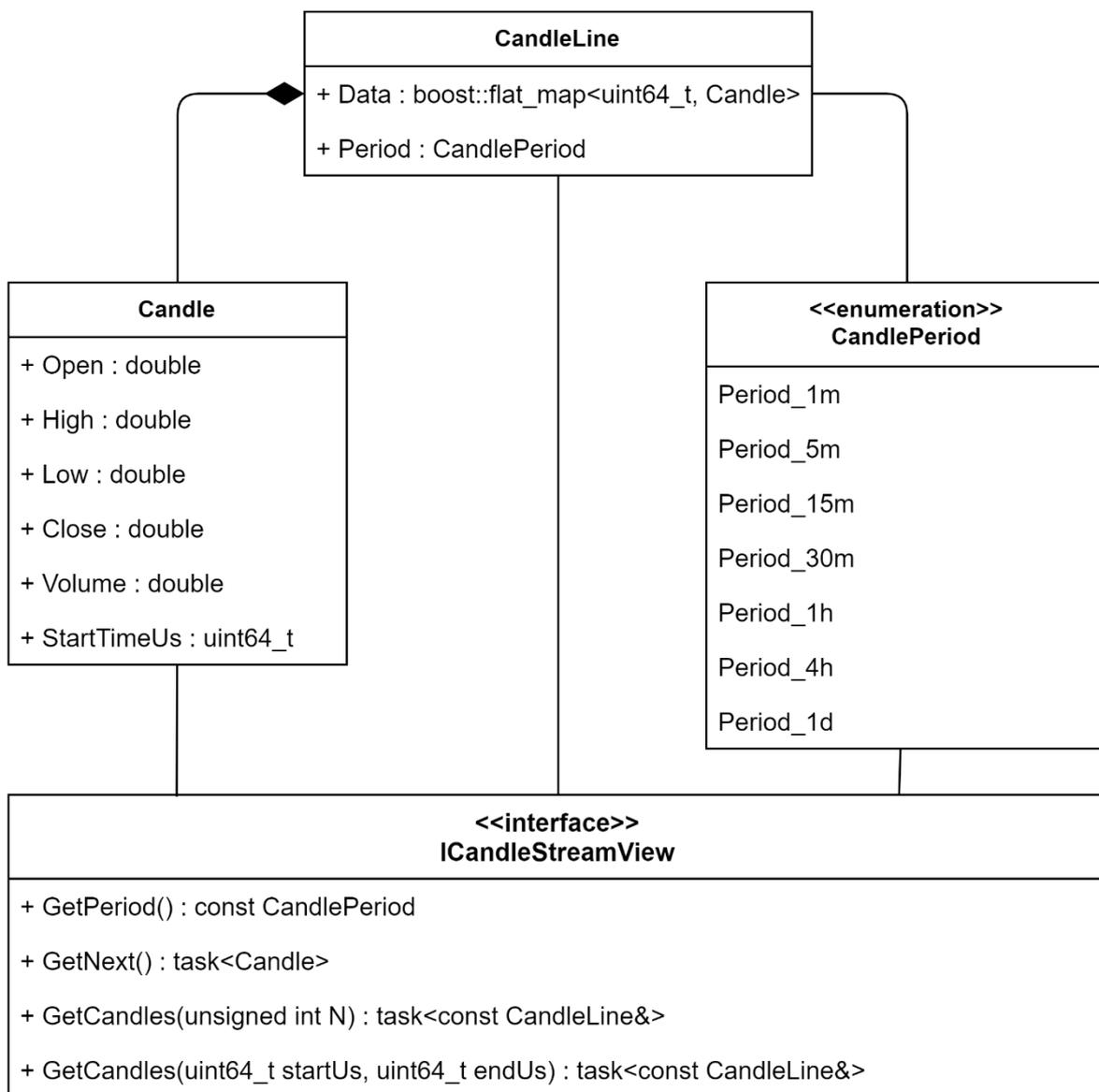


Рисунок 23 – Детализация ICandleStreamView

Для создания объектов потоков данных предусмотрены интерфейсы с суффиксом «Provider», представленные на рисунке 24. Каждый из них имеет единственный метод, возвращающий поток данных для инструмента, указанного в аргументе вызываемого метода. `ICandleStreamProvider` в соответствующем методе имеет два аргумента, второй отвечает за желаемый таймфрейм свечного потока данных. Реализация этих интерфейсов для конкретной биржи при вызове указанных методов производит подключение к бирже и запрос необходимых данных прежде, чем предоставить поток данных.

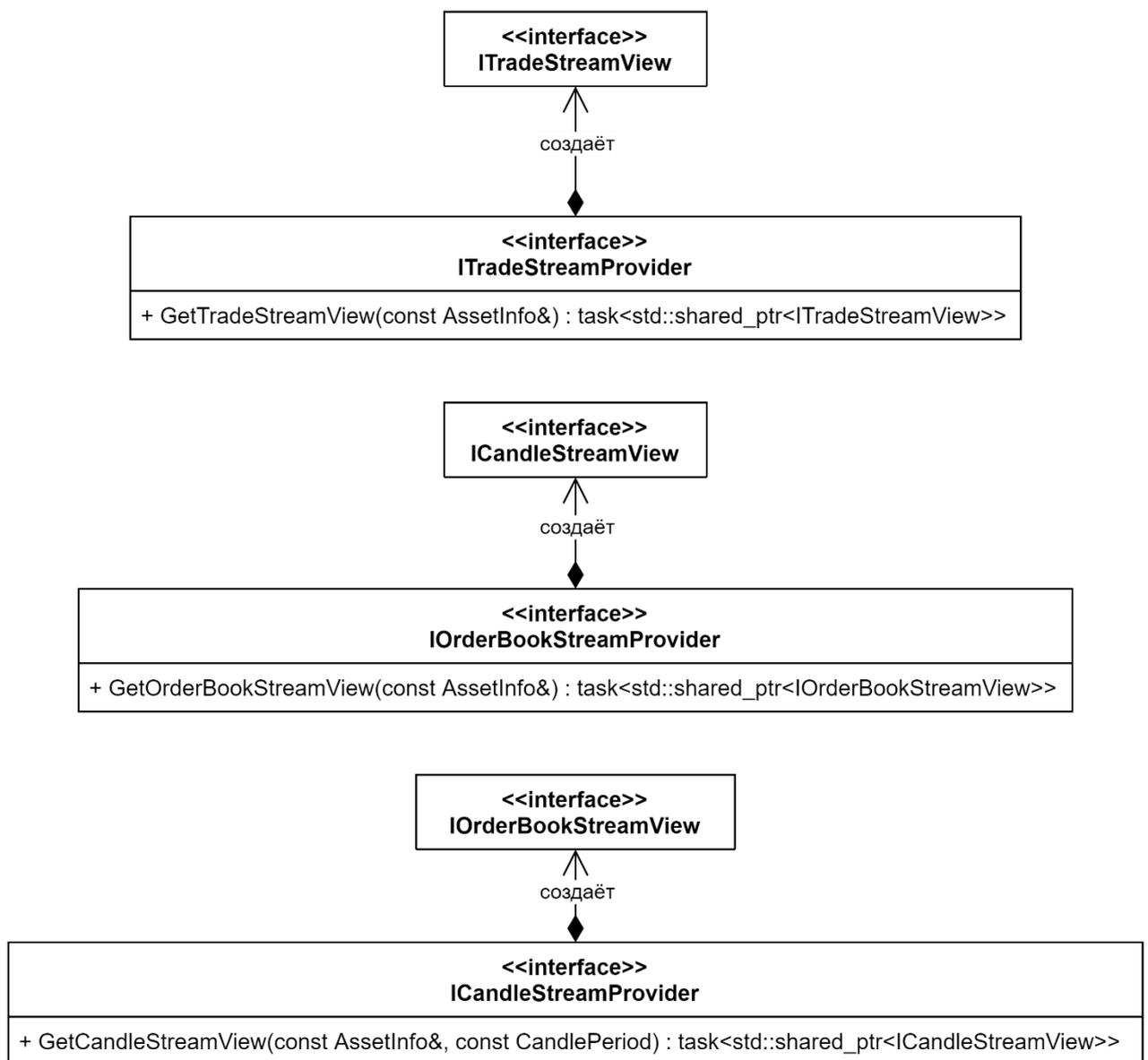


Рисунок 24 – Порождение потоков данных

Для представления информации об инструменте используется структура AssetInfo, представленная на рисунке 6. Поле AssetType обозначает тип инструмента и может принимать два значения: Spot (базовый актив) или PerpetualFuture (бессрочный фьючерс). Поля BaseAsset и QuoteAsset содержат информацию о названии базового и котируемого активов соответственно. Поле PriceDecimal показывает, сколько десятичных знаков после запятой необходимо для представления цены (в единицах котируемого актива). Поле SizeDecimal показывает, сколько десятичных знаков после запятой необходимо для представления размера заявки или сделки (в единицах базового актива).

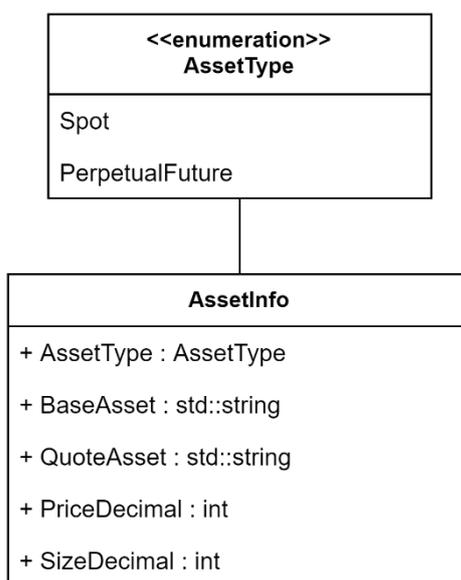


Рисунок 25 – Информация об инструменте

В процессе разработки было замечено, что иногда работа приложения завершалась критической ошибкой. Данная ошибка была найдена и устранена, но оставалась вероятность, что всё еще есть (или могут появиться в будущем) другие критические ошибки. Наиболее неприятным становится то, что они могут проявляться только при очень специфических условиях. Для того, чтоб пользователь мог предоставить информацию в случае появления таких ошибок, а не только сам факт их появления, был разработан механизм сбора частичного дампа памяти и сохранения его в файл, который затем может быть проанализирован разработчиком с целью установить причину ошибки.

Содержание заголовочного файла CrashHandler.h приведено в листинге 1. В нём объявляется единственная функция SetCrashHandler, которая должна быть вызвана в начале программы для установки обработчика непоименных исключений.

Листинг 1 – Заголовочный файл CrashHandler.h

```
#pragma once

#include <string>

namespace CrashReporter {

    void SetCrashHandler(const std::wstring& dumpFileName =
L"crash.dmp");

}
```

Содержание файла реализации CrashHandler.cpp приведено в листинге 2.

Листинг 2 – Файл реализации CrashHandler.cpp

```
#include <windows.h>
#include <minidumpapiset.h>

#include "CrashReporter.h"

namespace CrashReporter {
    // Вспомогательная функция для отображения пользователю сообщения
    // о факте возникновения критической ошибки
    void ShowCrashMessageBox() {
        MessageBox(NULL, (L"Application was crashed. Dump has been
saved."), (L"Crash handler"), MB_ICONERROR | MB_OK);
    }
    // Объявление типа функции для записи информации об ошибке и дампа
    памяти в файл
    typedef BOOL(WINAPI* MINIDUMPWRITEDUMP)(HANDLE hProcess, DWORD dwPid,
HANDLE hFile,
        MINIDUMP_TYPE DumpType,
        CONST PMINIDUMP_EXCEPTION_INFORMATION ExceptionParam,
        CONST PMINIDUMP_USER_STREAM_INFORMATION UserStreamParam,
        CONST PMINIDUMP_CALLBACK_INFORMATION CallbackParam
    );

    const wchar_t* DBGHELP = L"DbgHelp.dll";
}
```

## Листинг 2 (продолжение)

```
bool MakeDump(const std::wstring& dumpFile, _EXCEPTION_POINTERS*
ExceptionInfo)
{
    bool success = false;

    _MINIDUMP_EXCEPTION_INFORMATION mei;
    mei.ClientPointers = FALSE;
    mei.ExceptionPointers = ExceptionInfo;
    mei.ThreadId = ::GetCurrentThreadId();

    // Загрузить динамическую библиотеку dbghelp.dll
    HINSTANCE dbgHelpDLL = ::LoadLibrary(DBGHELP);
    if (dbgHelpDLL != NULL)
    {
        // Получить адрес функции из загруженной динамической
библиотеки
        MINIDUMPWRITEDUMP pDump =
MINIDUMPWRITEDUMP(::GetProcAddress(dbgHelpDLL, "MiniDumpWriteDump"));

        if (pDump)
        {
            // Создать файл, куда будет записан дамп
            HANDLE fileHandle = ::CreateFileW(dumpFile.c_str(),
GENERIC_WRITE, FILE_SHARE_WRITE, nullptr, CREATE_ALWAYS,
FILE_ATTRIBUTE_NORMAL, nullptr);

            if (fileHandle != INVALID_HANDLE_VALUE)
            {
                // Собрать информацию об ошибке и дамп памяти
текущего процесса
                // и записать в подготовленный ранее файл
                BOOL bOK = pDump(GetCurrentProcess(),
GetCurrentProcessId(),
fileHandle,
MiniDumpNormal,
&mei,
nullptr,
nullptr);

                if (bOK)
                {
                    success = true;
                }

                ::CloseHandle(fileHandle);
            }
        }

        ::FreeLibrary(dbgHelpDLL);
    }
}
```

## Листинг 2 (продолжение)

```
        return success;
    }

    static std::wstring DumpFileName = L"crash.dmp";

    LONG WINAPI TopLevelFilter(_EXCEPTION_POINTERS* ExceptionInfo) {
        MakeDump(DumpFileName, ExceptionInfo);
        ShowCrashMessageBox();
        return EXCEPTION_EXECUTE_HANDLER;
    }

    void SetCrashHandler(const std::wstring& dumpFileName) {
        DumpFileName = dumpFileName;
        // Установить функцию-обработчик, которая будет перехватывать
        // необработанные исключения и обрабатывать их
        ::SetUnhandledExceptionFilter(TopLevelFilter);
    }
}
```

### **3.2. Разработка клиентской части**

Клиентская часть разрабатываемой системы является приложением в смысле интерактивности предоставляемого контента, а не сайтом, где больший акцент был бы сделан на предоставлении некоторой статичной информации. Поэтому предпочтительной является разработка клиентской части как одностраничного приложения (SPA), а не классического многостраничного (MPA).

Для разработки клиентской части будет использоваться реактивный JavaScript фреймворк VueJS[20], который прост в освоении и использовании.

Для быстрого прототипирования и создания пользовательского интерфейса планируется применять фреймворк Quasar[21], который работает поверх VueJS. Он содержит большое количество готовых элементов пользовательского интерфейса.

### **3.3. Обзор основных возможностей приложения**

Работа с приложением происходит в одном или нескольких окнах браузера. Внутри каждого окна браузера можно расположить одно или несколько функциональных окон, образующих в совокупности рабочее пространство, как показано на рисунке 26, предоставляя таким образом возможность гибко конфигурировать рабочее место под различные задачи.

Информация доводится до пользователя в том числе аудиально с помощью звуковых уведомлений, представляющих собой звуки, воспроизводимые в ответ на возникновение того или иного события, а также с помощью голосовых уведомлений, использующих технологию синтеза речи, что даёт возможность оповестить пользователя о существенных деталях наступившего события. Синтез речи на русском языке наилучшим образом работает в браузере Google Chrome, поэтому пользователям рекомендуется использовать именно его.

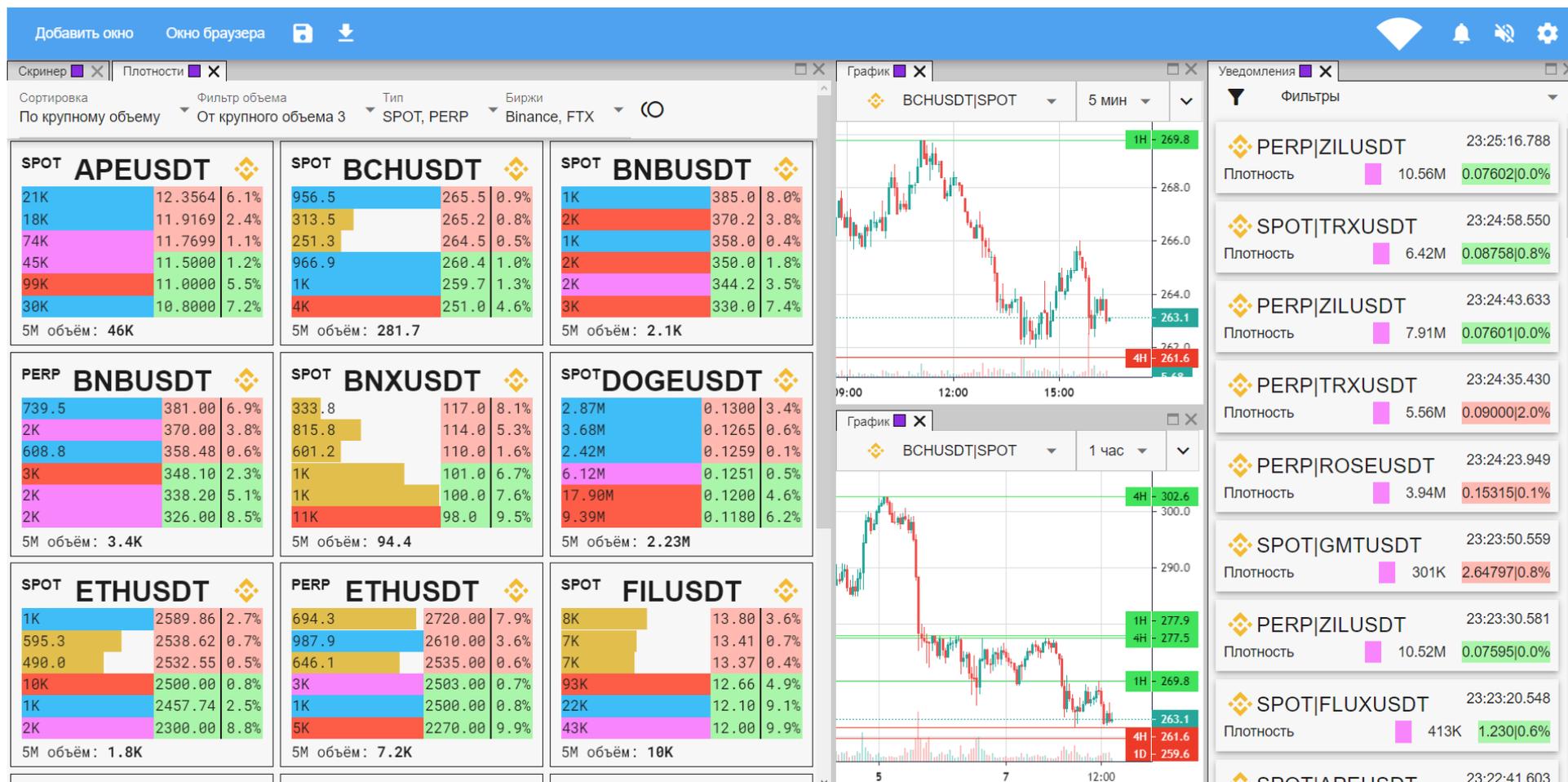


Рисунок 26 – Пример рабочего пространства внутри окна браузера

### 3.3.1. Крупные плотности

Приложение позволяет отображать наличие, размер, время обнаружения, факт касания сделками плотностей («трогали ли плотность») в виде мини-стаканов, расположенных сеткой, как показано на рисунке 27. К мини-стаканам применима фильтрация и сортировка, а также поддерживается отображение объема плотностей в долларах даже для пар вида ADABTC или LTCETH, то есть для тех, где котируемая валюта не является долларом или его эквивалентом.

Факт появления плотности ближе заданного расстояния (в процентах) от спреда отражается с помощью уведомлений (как звуковых/голосовых, так и приходящих в окно «Уведомления»).

Кроме того, крупные плотности отображаются на графике соответствующего инструмента.

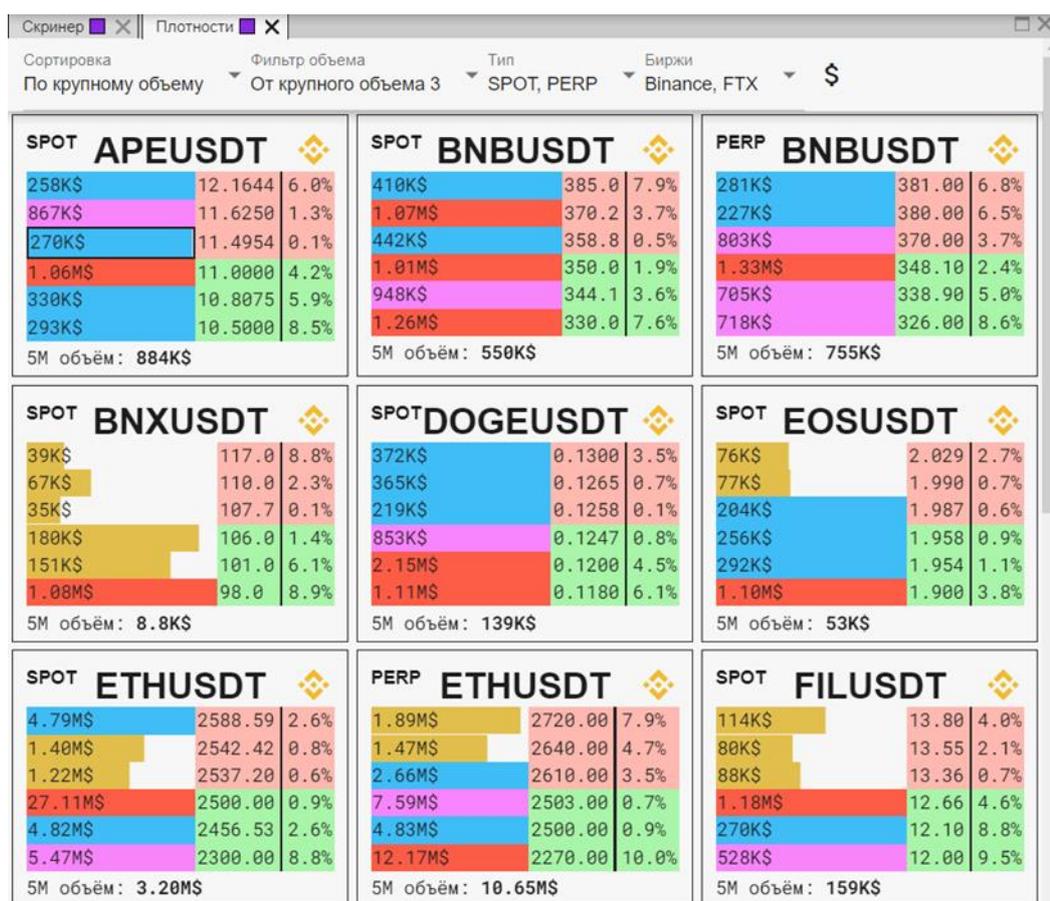


Рисунок 27 – Отображение крупных плотностей

### 3.3.2. Графики

Графики позволяют просматривать график цены и объема выбранного инструмента на одном из таймфреймов. Внешний вид графика приведен на рисунке 28. На графике могут отображаться горизонтальные уровни и обнаруженные крупные плотности. Наиболее полно возможности графиков раскрываются при применении их в группах линковки.



Рисунок 28 – Внешний вид графика

### 3.3.3. Уведомления

Окно «Уведомления» (оно же «окно уведомлений», «лента уведомлений»), показанное на рисунке 29, позволяет просматривать уведомления о произошедших событиях с указанием инструмента, где произошло событие, локального времени события и дополнительной информации, специфической для каждого вида событий. Присутствует возможность фильтрации событий по нескольким критериям.

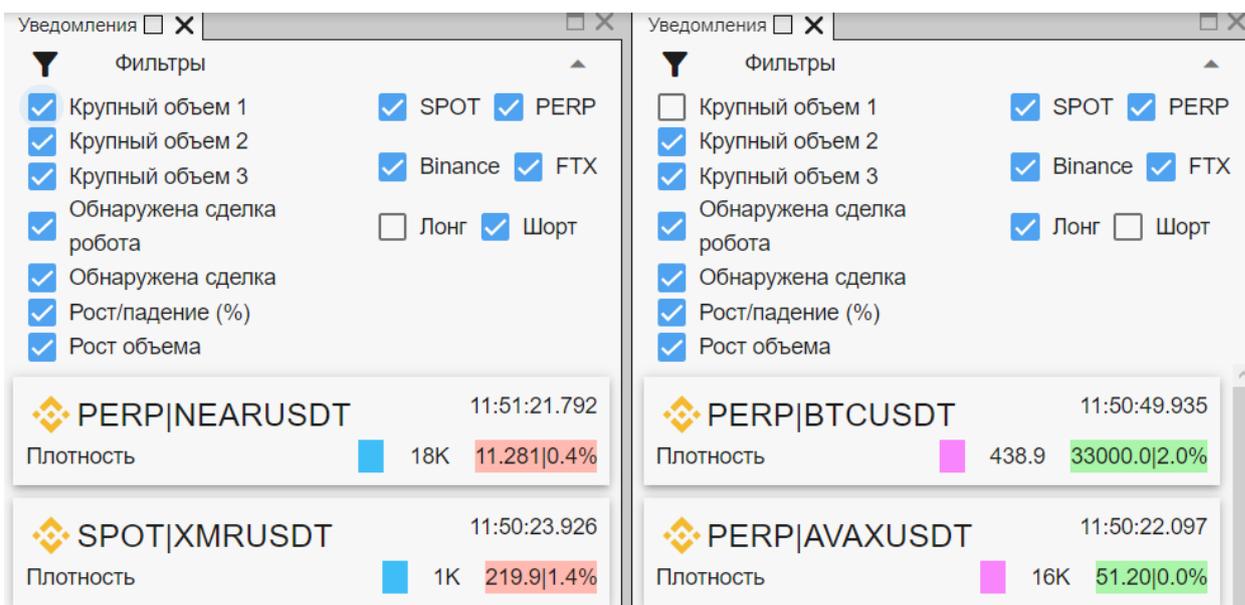


Рисунок 29 – Окна уведомлений с открытыми параметрами фильтрации

### 3.3.4. Автоматический поиск горизонтальных уровней

Приложение имеет возможность автоматического поиска горизонтальных уровней на следующих таймфреймах: час, четыре часа, день. Найденные уровни отображаются на графиках, а также рассчитывается расстояние до ближайшего из них по каждому инструменту (в окне «Скринер»).

### 3.3.5. Скринер

На рисунке 30 представлено окно «Скринер», которое позволяет отображать несколько показателей, рассчитываемых на основе свечных данных, для каждой монеты в реальном времени; производить сортировку по любому из этих параметров; фильтровать (в том числе, используя несколько условий) монеты по одному или нескольким параметрам (настройки фильтрации приведены на рисунке 31).

Наименование	Тип	↓  %	Corr	Объем		x	ATR		Уровни %
				Avg\$	Now\$		%	x	
ANCUSDT	PERP	0.4	0.9	373k	192k	0.5	0.8	0.55	39.2   1H L
NBTUSDT	SPOT	-5.2	0.62	40	728	18.1	0.5	-10.54	6.1   4H S
AXSUSDT	PERP	1.6	0.61	2.6M	3.4M	1.3	0.9	1.67	4.9   1H L
AXSUSDT	SPOT	1.6	0.6	331k	343k	1	0.9	1.75	5.6   1H L
DOTDOWNUSDT	SPOT	-0.9	-0.98	21k	14k	0.7	1.5	-0.6	11.4   4H L
BNBUPUSDT	SPOT	0.8	0.98	7.4k	3.8k	0.5	1.2	0.68	22   1H L
DNTUSDT	SPOT	0.8	0.97	8.9k	11k	1.3	0.6	1.51	8   1H L
GMTUSDT	PERP	1	0.35	12.6M	10.5M	0.8	1.2	0.82	4.3   1H L
ALPINEUSDT	SPOT	0.8	0.94	31k	64k	2.1	0.7	1.08	20.1   4H L
INJUSDT	SPOT	0.8	0.95	8.1k	16k	2	0.5	1.67	3.6   1D S
GMTUSDT	SPOT	1	0.36	2.6M	1.7M	0.6	1.2	0.87	4.3   1H L
XRPDOWNUSDT	SPOT	-0.9	-0.95	18k	24k	1.4	1	-0.91	4   1H S
ADADOWNUSDT	SPOT	-0.6	-0.96	12k	8.1k	0.7	1	-0.56	4.2   1H S
MOBUSDT	SPOT	0.7	0.34	12k	8.6k	0.7	0.6	1.21	1.8   1H S
BLZUSDT	SPOT	0.3	0.97	7.8k	4.8k	0.6	0.5	0.48	1.4   4H S
REIUSDT	SPOT	0.7	0.74	17k	10k	0.6	1	0.68	29.2   1H L
UNFIUSDT	SPOT	0.4	0.97	28k	9.8k	0.4	0.8	0.44	3.2   4H S
LUNAUSDT	SPOT	0.8	0.89	3.8M	2.7M	0.7	1.2	0.63	8   1H S
TRBUSDT	SPOT	0.7	0.94	3.6k	824	0.2	0.6	1.04	2.7   1H L
ERNUSDT	SPOT	0.6	0.97	5.7k	2.5k	0.4	0.6	1.03	13.1   4H L
REQUUSD	SPOT	0.6	0.79	39k	83k	2.1	1.2	0.54	6   4H S

Рисунок 30 – Внешний вид окна «Скринер»

Возможна раскраска фона и текста каждой ячейки в зависимости от её значения. Для каждого столбца имеется возможность задать такую раскраску индивидуально. Раскраска задается с помощью редактора линейного градиента, интерфейс которого представлен на рисунке 32.

Общие Модули Фильтры

Максимум строк в таблице

Условие между фильтрами ИЛИ

---

Изменение цены %
Вне диапазона

Объем: x
Больше

Рисунок 31 – Фильтрация по нескольким условиям

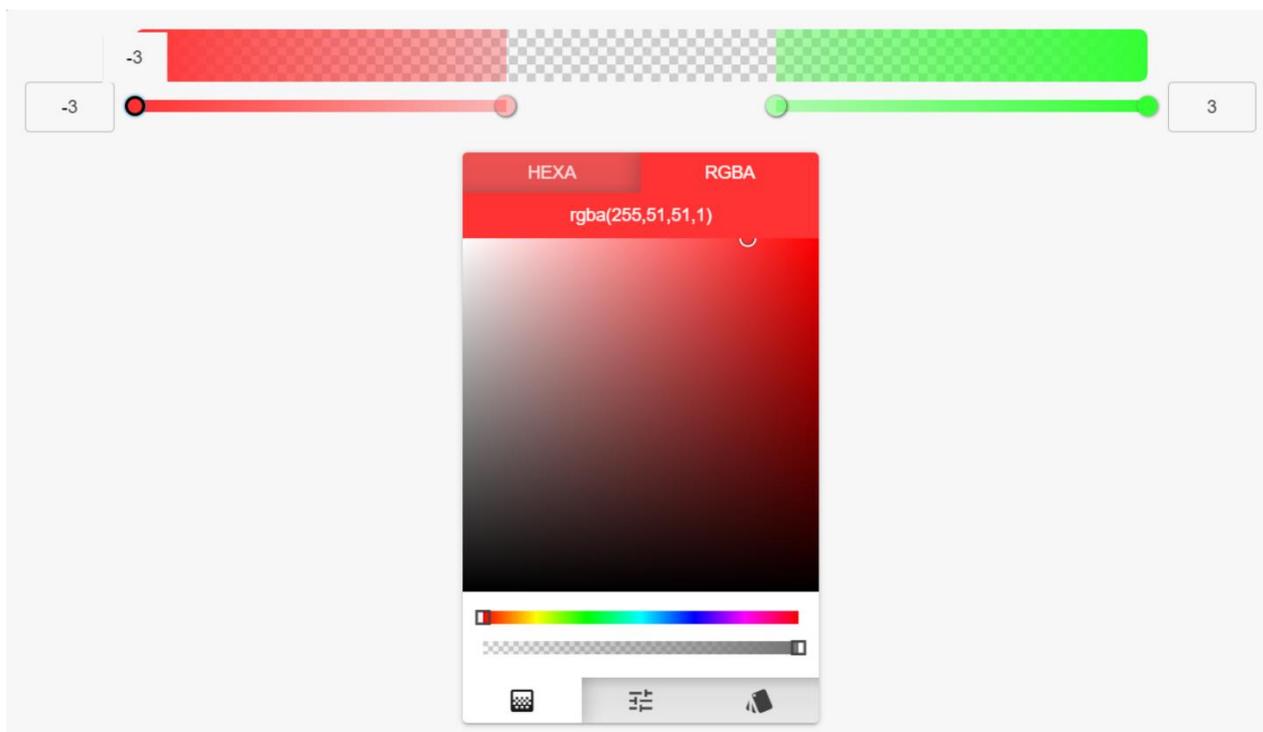


Рисунок 32 – Редактирование градиента для задания раскраски фона ячеек столбца в зависимости от содержащегося в них значения

Существует возможность отображать инструменты не только в виде таблицы, но и в виде сетки графиков, что показано на рисунке 33, на которые также распространяется фильтрация и сортировка.

Как в режиме таблицы, так и в режиме графиков поддерживается добавление инструментов в избранные, которые отображаются первыми перед всеми иными инструментами в таблице (перед всеми иными графиками).

Следует отметить, что существует возможность иметь одновременно несколько окон «Скринер», каждое со своими индивидуальными настройками, в том числе с настройками фильтрации и сортировки. Такой подход позволяет рассматривать рынок с разных уникальных позиций в одно и то же время.

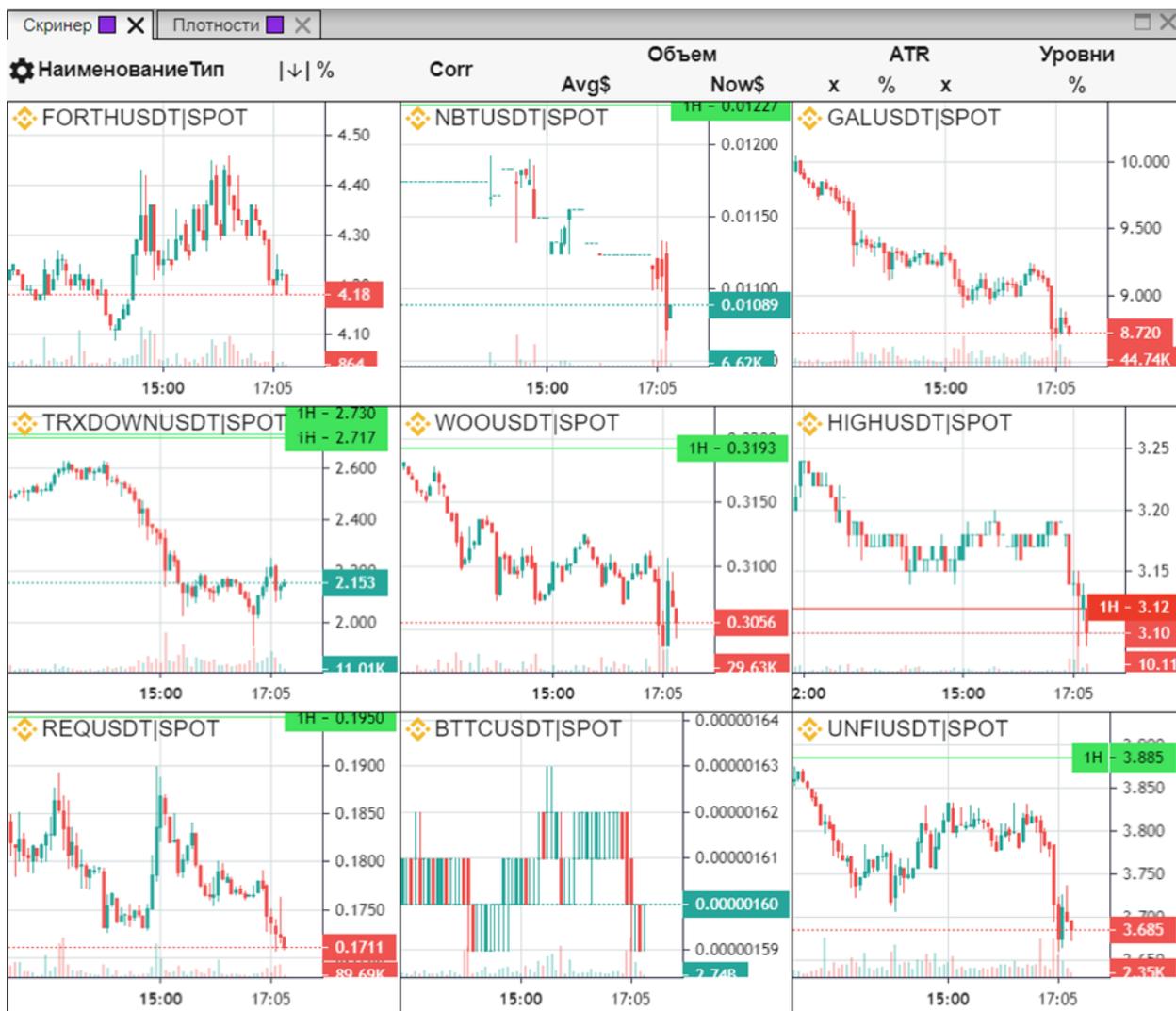


Рисунок 33 – Окно «Скринер» в режиме отображения графиков

### 3.3.6. Группы линковки и перетаскивание (drag and drop) инструментов

Для существенного ускорения процесса просмотра графиков предусмотрены группы линковки. Если несколько графиков принадлежат к одной группе линковки, то при выборе инструмента на одном из них этот же инструмент будет автоматически выбран на всех других графиках в этой группе линковки, что, например, позволяет одним кликом открывать четыре различных таймфрейма по одному инструменту. Или же одним кликом открывать два таймфрейма для спота и два для соответствующего ему фьючерса, как показано на рисунке 34.

В группу линковки может быть включено любое окно, а не только «График». При клике по названию инструмента (для окон «Плотности», «Уведомления»), по строке в таблице (для окна «Скринер» в режиме таблицы) или по заголовку графика (для окна «Скринер» в режиме графиков) этот инструмент становится текущим для группы линковки, и может быть быстро открыт на графиках, имеющих ту же группу линковки.

Группы линковки работают в том числе между разными окнами браузера.

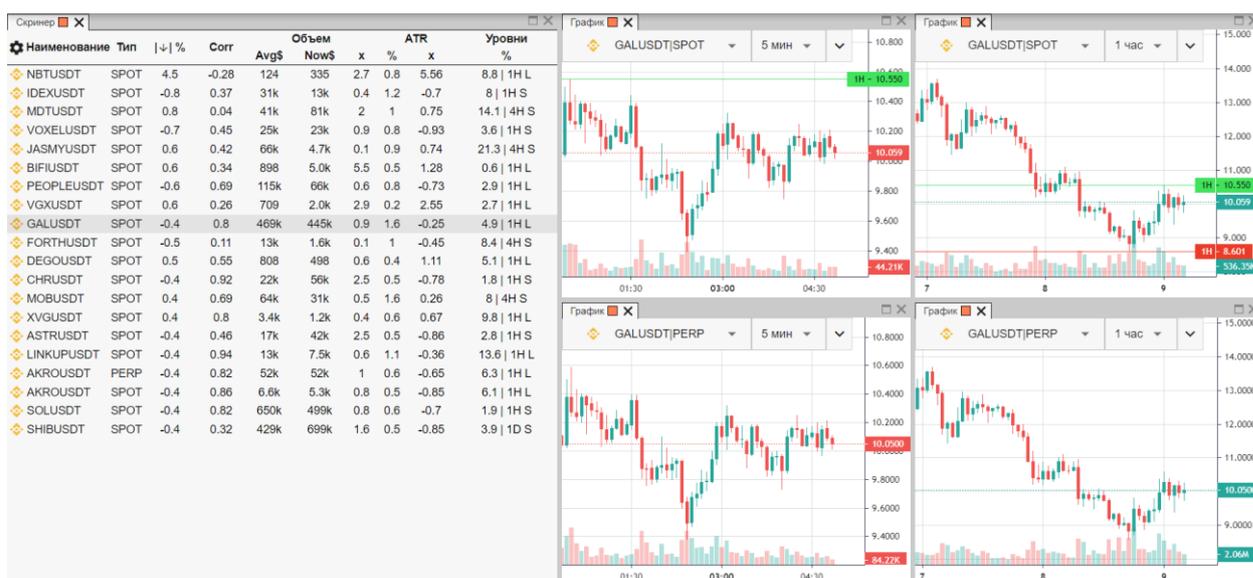


Рисунок 34 – Окно «Скринер» и 4 графика имеют одну группу линковки

Группа линковки отображается в виде маленького квадрата, расположенного справа от названия каждого окна. При клике на этот квадрат появляется меню, представленное на рисунке 35. В появившемся меню можно выбрать, к какой группе линковки будет принадлежать это окно, или указать, что окно не включено ни в одну из групп линковки.

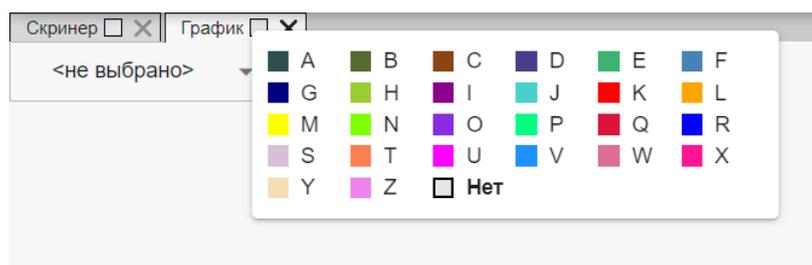


Рисунок 35 – Меню выбора группы линковки

Присутствует возможность перетащить название инструмента (для окон «Плотности», «Уведомления»), строку из таблицы (для окна «Скринер» в режиме таблицы) или заголовок графика (для окна «Скринер» в режиме графиков) в окно «График», что продемонстрировано на рисунке 36. Таким образом можно задать текущий инструмент на графике. Если график входит в группу линковки, то инструмент станет текущим для группы линковки.

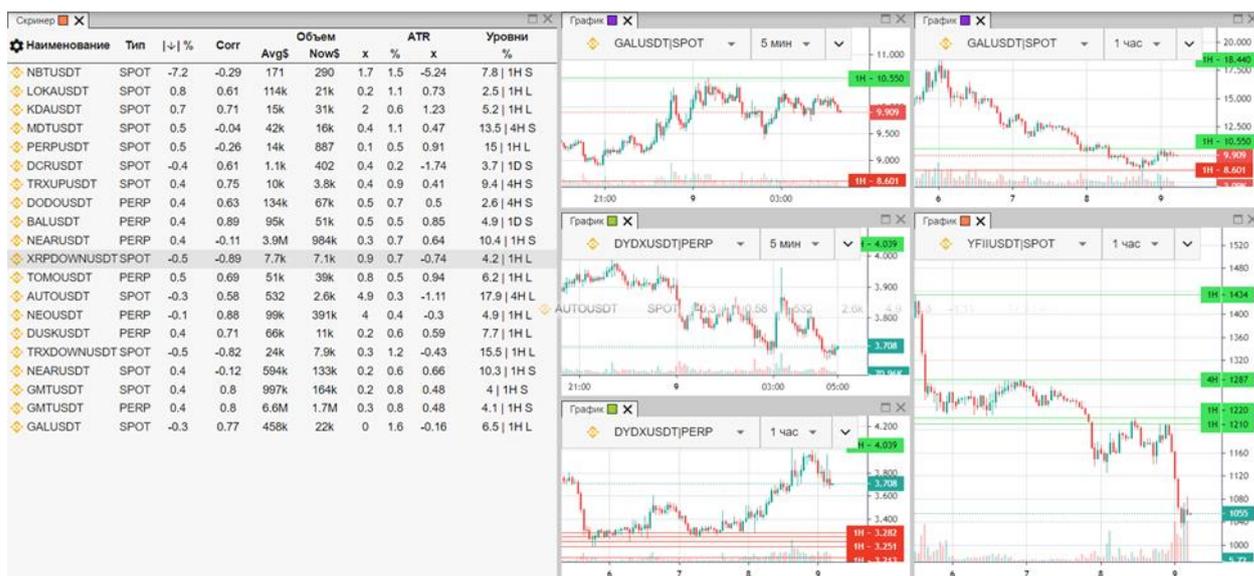


Рисунок 36 – Перетаскивание строки из таблицы окна «Скринер» на график

Если перетаскивание оканчивается в текстовом поле (например, в поисковой строке браузера), то туда будет вставлено название инструмента.

### 3.3.7. Взаимодействие с торговыми терминалами

По двойному клику на название инструмента (для окон «Плотности», «Уведомления»), строку из таблицы (для окна «Скринер» в режиме таблицы) или заголовок графика (для окна «Скринер» в режиме графиков) происходит открытие стакана выбранного инструмента (или пары стаканов спот – фьючерс) в торговом терминале.

На текущий момент поддерживается только терминал CScalp. Данная возможность не требует ввода координат кнопок и каких бы то ни было иных координат, равно как и расположения окна CScalp в определенном месте.

#### 4. Тестирование и апробация разработанного ПО

Закрытое тестирование разработанного ПО проводилось внутри сообщества трейдеров «League of Traders». Тестирование было начато 9 мая 2022 года.

Участники сообщества восприняли появление такого инструмента очень позитивно, так как с ним их деятельность становится значительно проще и удобнее.

На рисунке 37 сплошной линией отображено количество пользователей, принимавших участие в тестировании. Пунктирной линией отображено количество пользователей, нормализованное на 24 часа. Вместе эти показатели позволяют оценить не только количество пользователей за сутки, но и то, насколько длительно каждый использует приложение – то есть у какого эквивалентного числа пользователей приложение было запущено в течение 24 часов.



Рисунок 37 – Количество пользователей во время тестирования

В ходе тестирования было выяснено, что наиболее типичным набором инструментов для получения по ним биржевых данных являются все USDT-фьючерсы и все USDT-споты (то есть базовые активы) на бирже Binance. Начальная загрузка данных с биржи для такого набора занимает 3-5 минут. Это время определяется не скоростью передачи данных, а лимитом запросов к бирже в минуту.

Нагрузка серверной части (а именно там происходит прием и обработка биржевых данных, это наиболее ресурсозатратные операции) на процессор составила: 10-25% для i3-6006U, 3-6% для Ryzen 5 2600, 1-3% для Ryzen 5 5600X.

Потребляемая пропускная способность сетевого подключения составляла от 1,5 Мбит/сек на спокойном рынке до 12-20 Мбит/сек в периоды высокой волатильности.

В ходе разработки и тестирования было выяснено, что биржа FTX не предоставляет данные обо всех происходящих сделках, а лишь о части. Этот факт, а также то, что FTX не предоставляет обновления свечных данных через websocket, делает все расчеты на основе свечей и сделок для этой биржи ненадежными.

В ходе тестирования некоторые пользователи сталкивались с критическими ошибками, которые приводили к аварийному завершению работы серверной части. Позже они присылали файлы дампа, созданные приложением с помощью предусмотренного механизма. Данные файлы были проанализированы, один из этапов анализа продемонстрирован на рисунке 38. Это позволило обнаружить ошибки, которые не были выявлены на этапе разработки в силу специфических условий их появления, и приступить к поиску причин этих ошибок и их устранению.

Существует канал в Discord[22], где участники сообщества задают свои вопросы, предлагают идеи по улучшению и дальнейшему развитию приложения, примеры приведены на рисунках 39 и 40.

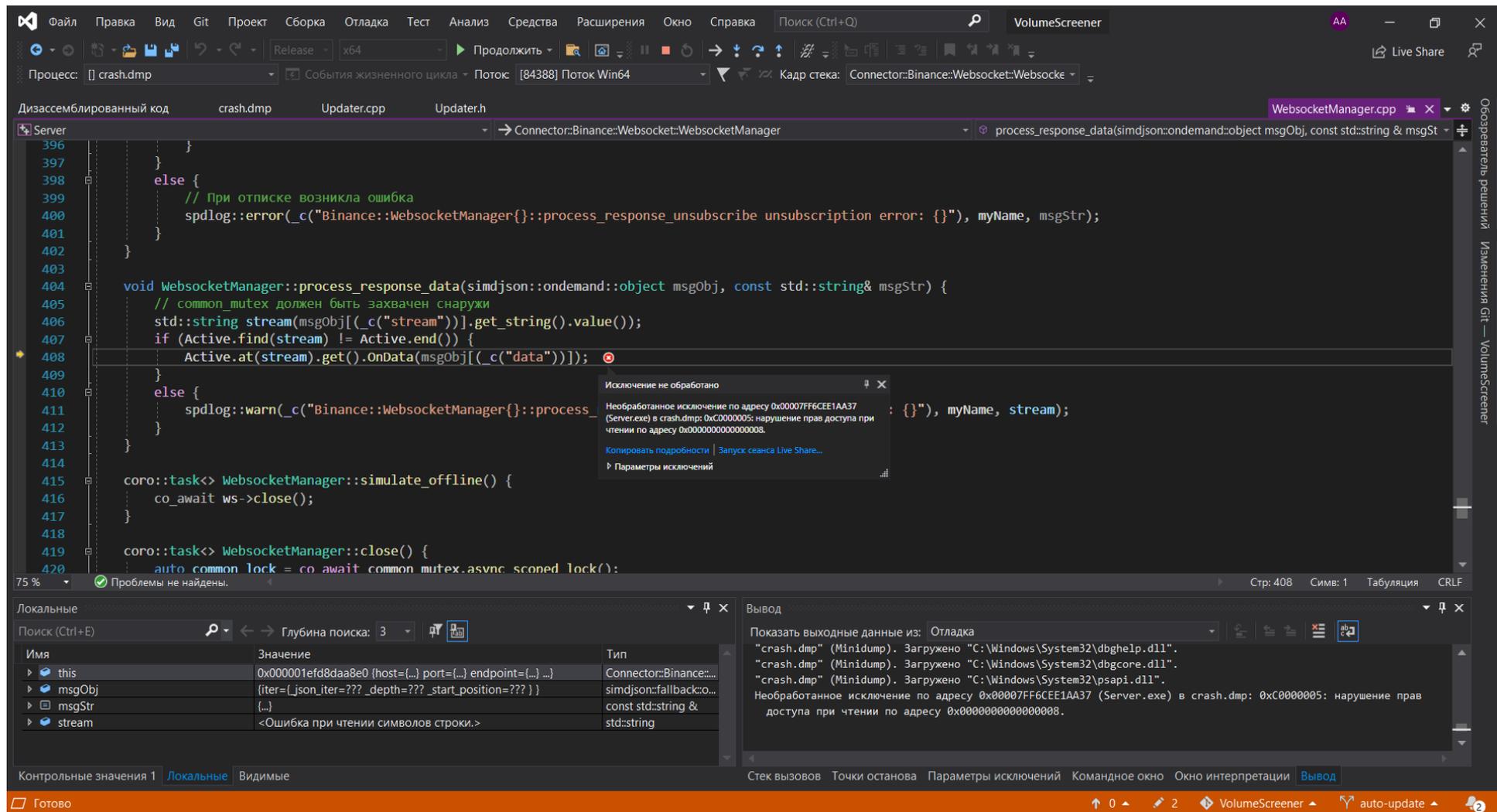


Рисунок 38 – Поиск причины критической ошибки

 **Виталий Ермак** 14.05.2022  
Ну и раз уж идёт обсуждение, добавлю своё мнение насчёт того, как могла бы выглядеть "серебряная пуля" для плотностей. Сначала нужно распределение инструментов по объёму в долларах за 5 минут. К примеру, следующие категории (которые можно настраивать самостоятельно): 0-200к, 200-600к, 600к-1,5М, 1,5М-4М, 4М-10М, 10М+. Дальше для каждой категории создаётся своё правило на основании относительного (пятиминутного) объёма. К примеру, для категории 0-200к базовая плотность равна одному пятиминутному объёму. А для категории 4М-10М базовая плотность равна 0,2 пятиминутного объёма. Так как, чем выше абсолютные объёмы на инструменте, тем меньшая по отношению к пятиминутке объёму плотность начинает оказывать существенное влияние на движение цены. И это правило, как мне кажется, универсально для всех инструментов. Надеюсь, у меня получилось более менее понятно выразить свою идею ). (изменено)

### Рисунок 39 – Предложение по улучшению алгоритма определения степени крупности плотностей

 @Виталий Ермак Согласен, что существует необходимость в поиске более универсального подхода к оценке плотностей. Но вариант ...

 **Александр Орлов** Сегодня, в 14:17  
Виталий, спасибо за мнение. Согласен, индикатор DF не лишён недостатков в плане реакции на волатильность - я тоже об этом писал, но мне казалось, что его достоинством по сравнению с другими методами является то, что он опирается на реальные плотности, которые проходили в кластерах, а не на усреднения общих объёмов. Что касается твоего предложения то его плюс в том, что его, как мне кажется, намного проще реализовать в скринере, чем поиск по кластерам, но используешь ли ты его на практике или это только теоретическое предположение?

 @Александр Орлов Виталий, спасибо за мнение. Согласен, индикатор DF не лишён недостатков в плане реакции на волатильность - ...

 **Виталий Ермак** Сегодня, в 14:37  
Да, именно таким подходом я пользуюсь при оценке плотностей после нахождения их в скринерах или просто в стакане. Я сначала оцениваю плотность в долларах. Меньше 100к я отбрасываю (если это не сделал скриннер), так как её либо легко могут принтануть, либо там такие низкие объёмы и волатильность, что смысла заходить мало. Дальше я провожу сравнение с пятиминутным объёмом. А после смотрю, какие долларовые объёмы проходят на инструменте и какая тут волатильность. И исходя из этого я принимаю решение. Такой подход меня, можно сказать, не подводит: реакция рынка на плотность практически всегда соответствует моим ожиданиям. (изменено)

### Рисунок 40 – Дальнейшее обсуждение алгоритма определения степени крупности плотностей

## **5. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение**

### **Введение**

Данная выпускная квалификационная работа заключается в разработке программного обеспечения для мониторинга торговой активности на криптовалютных биржах. Пользователями разрабатываемого программного обеспечения являются трейдеры, торгующие на криптовалютных биржах (прежде всего на Binance и FTX) и использующие такие стили торговли, как скальпинг и интрадей-трейдинг.

Целью разрабатываемого программного обеспечения является своевременно доводить до трейдера важную и актуальную рыночную информацию в компактном графическом и аудиальном представлении

Разрабатываемое программное обеспечение нацелено на извлечение прибыли посредством продажи его пользователям. В данной главе проводится его сравнение с конкурентными решениями, оценка коммерческого потенциала, определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности.

### **5.1. Оценка коммерческого потенциала и перспективности разработки**

#### **5.1.1. Потенциальные потребители разрабатываемого решения**

Как было упомянуто выше, потенциальными потребителями разрабатываемого программного обеспечения являются трейдеры, торгующие на криптовалютных биржах (прежде всего на Binance и FTX) и использующие такие стили торговли, как скальпинг и интрадей-трейдинг.

Для таких трейдеров остро стоит проблема отслеживания наличия интересующих их ситуаций на как можно большем числе криптовалютных пар сразу. Поэтому типичное рабочее место скальпера оборудовано сразу несколькими мониторами, на которых отображаются графики и стаканы

инструментов. Однако в связи с тем, что количество доступных инструментов на бирже Binance измеряется несколькими сотнями, невозможно следить за ними всеми одновременно, даже при наличии нескольких десятков мониторов (вопрос о необходимой мощности компьютера для одновременного отображения такого количества информации опускается). Внимание человека просто не может быть направлено на такое количество раздражителей одновременно.

Целью разрабатываемого решения является своевременно доводить до трейдера важную и актуальную рыночную информацию в компактном графическом и аудиальном представлении. Это позволит не следить одновременно за всем разнообразием доступных криптовалютных инструментов, а сосредоточить внимание на тех из них, где в данный момент существуют интересующие пользователя условия. Таким образом создаются условия для повышения результативности деятельности трейдера.

Трейдеров, торгующих более позиционно (среднесрочные и долгосрочные сделки), разрабатываемый продукт вряд ли заинтересует.

На данный момент разрабатываемое программное обеспечение поддерживает только русский язык, соответственно, нацелено на русскоязычных трейдеров, однако планируется добавление английского языка с перспективой выхода на международный рынок. Восточная Азия также является привлекательным рынком с точки зрения числа потенциальных пользователей, но известные проблемы, связанные с трудностью охраны авторских прав в Китае, ставят под вопрос целесообразность вложения усилий для выхода на этот рынок.

### **5.1.2. Анализ конкурентных технических решений**

В настоящий момент на рынке существуют несколько продуктов со схожим назначением и функциональностью. В сообществе трейдеров за такими продуктами закрепилось название «скринер».

Для оценки конкурентоспособности разрабатываемого программного обеспечения была составлена карта сравнения конкурентных технических

решений, представленная в таблице ниже. Индексом «ф» обозначено разрабатываемое решение, индексом «к1» - скринер «Скаल्पЛайв» (<https://scalp.live>), индексом «к2» - скринер «WinScreener» (<https://winscreener.live>). Баллы находятся в диапазоне от 0 до 10 включительно, где 0 является наихудшим значением по критерию, а 10 – наилучшим.

Таблица 1 – Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентоспособность		
		Б <sub>ф</sub>	Б <sub>к1</sub>	Б <sub>к2</sub>	К <sub>ф</sub>	К <sub>к1</sub>	К <sub>к2</sub>
1	2	3	4	5	6	7	8
<b>Технические критерии оценки ресурсоэффективности</b>							
1. Гибкость настройки	0,20	10	6	5	2	1,2	1
2. Понятность и удобство пользовательского интерфейса	0,15	6	7	9	0,9	1,05	1,35
3. Актуальность предоставляемых данных	0,10	10	8	6	1	0,8	0,6
4. Нагрузка на компьютер пользователя	0,14	6	4	9	0,84	0,56	1,26
5. Охват по количеству доступных бирж и криптовалютных пар	0,08	4	1	1	0,32	0,08	0,08
6. Разнообразие аналитических инструментов	0,12	10	4	7	1,2	0,48	0,84
<b>Экономические критерии оценки эффективности</b>							
7. Цена	0,21	5	7	9	1,05	1,47	1,89
<b>Итого</b>	<b>1</b>	<b>51</b>	<b>37</b>	<b>46</b>	<b>7,31</b>	<b>5,64</b>	<b>7,02</b>

Из данных оценочной карты следует, что разрабатываемое решение является наиболее конкурентоспособным среди представленных. Его наиболее сильными сторонами являются: высокая гибкость настроек, актуальность

предоставляемых данных, разнообразие доступных аналитических инструментов.

### 5.1.3. SWOT-анализ

SWOT-анализ позволяет дать качественную оценку текущей ситуации, а также показывает, насколько достижима реализация имеющихся возможностей при наличии внешних угроз.

Процесс анализа можно разделить на два этапа. Начальный представляет из себя выявление сильных и слабых сторон решения, возможностей, а также угроз, завершающий – определение того, как между собой соотносятся все факты, выявленные на предыдущем этапе.

Результаты выполненной в ходе данного этапа работы приведены в таблице ниже.

Таблица 2 – Матрица SWOT-анализа

	<p><b>Сильные стороны:</b></p> <p>С1. Возможность гибкой настройки.</p> <p>С2. Разнообразие доступных аналитических инструментов.</p>	<p><b>Слабые стороны:</b></p> <p>Сл1. Сложность интерфейса и настройки.</p> <p>Сл2. Невысокая скорость развития (единственный разработчик).</p>
<p><b>Возможности:</b></p> <p>В1. Выход на международный рынок.</p> <p>В2. Сотрудничество с известными трейдерами.</p>	<p>Опытным трейдерам (в том числе за рубежом), с потребностями в индивидуальных настройках, данный продукт понравится, и, скорее всего, они будут</p>	<p>Новичков может оттолкнуть сложность и настройки пользовательского интерфейса.</p>

	рекомендовать его своим друзьям и знакомым.	
<b>Угрозы:</b> У1. Полный запрет на операции с криптовалютами в РФ. У2. Более низкие цены (либо вовсе бесплатные решения) у конкурентов.	Предлагаемый функционал поможет противостоять даже бесплатным решениям, но будет бесполезен против юридических ограничений.	Наибольшей угрозой является риск проиграть конкурентную борьбу из-за недостаточно быстрого развития продукта.

По результатам проведенного SWOT-анализа можно сделать вывод, что несмотря на имеющиеся угрозы и наличие слабых сторон, сильные стороны разрабатываемого решения и возможности по его развитию делают целесообразными разработку решения и выведение его на рынок.

## **5.2. Планирование работ по научно-техническому исследованию**

### **5.2.1. Структура работ в рамках научного исследования**

Перечень этапов работы и распределение исполнителей представлен в таблице ниже.

Таблица 3 – Перечень этапов, работ и распределение исполнителей

Основные этапы	№ раб	Содержание работ	Исполнители
Исследование предметной области	1	Опрос экспертов предметной области	Авербах А.В.
	2	Определение потребностей пользователей	Авербах А.В.
Проектирование	3	Разработка требований	Авербах А.В.
	4	Проектирование ПО	Авербах А.В.

Разработка	5	Разработка серверной части	Авербах А.В.
	6	Разработка клиентской части	Авербах А.В.
Создание сайта	7	Проектирование сайта	Авербах А.В.
	8	Разработка сайта	Авербах А.В.
Проведение тестирования	9	Проведение закрытого тестирования разработанного ПО	Авербах А.В.
Оформление отчета	10	Написание пояснительной записки к ВКР	Авербах А.В.

### 5.2.2. Определение трудоемкости выполнения работ

Трудоемкость выполнения научного исследования оценивается экспертным путем в человеко-днях и носит вероятностный характер, который зависит от множества трудно учитываемых факторов. Для определения ожидаемого (среднего) значения трудоемкости  $t_{ожі}$  используется следующая формула:

$$t_{ожі} = \frac{3t_{\min i} + 2t_{\max i}}{5}, \quad (1)$$

где  $t_{ожі}$  – ожидаемая трудоемкость выполнения  $i$ -ой работы чел.-дн.;

$t_{\min i}$  – минимально возможная трудоемкость выполнения заданной  $i$ -ой работы, чел.-дн.;

$t_{\max i}$  – максимально возможная трудоемкость выполнения заданной  $i$ -ой работы, чел.-дн.;

### 5.2.3. Разработка графика проведения научного исследования

Согласно производственному календарю (для 6-дневной рабочей недели) в 2022 году 365 календарных дней, 299 рабочих дней, 66 выходных/праздничных дней. Таким образом, коэффициент календарности на 2022 год равен:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})} = \frac{365}{365 - 66} = 1.22$$

После расчета коэффициента календарности можно составить таблицу временных показателей (Таблица 4) проведения научного исследования и диаграмму Ганта.

Таблица 4 – Временные показатели проведения научного исследования

Наименование работы	Исполнители работы	Трудоемкость работ, чел-дни			Длительность работ, дни	
		$t_{\min}$	$t_{\max}$	$t_{\text{ож}}$	$T_p$	$T_k$
Опрос экспертов предметной области	Авербах А.В.	8	12	9,6	10	12
Определение потребностей пользователей	Авербах А.В.	3	4	3,4	3	4
Разработка требований	Авербах А.В.	3	5	3,8	4	5
Проектирование ПО	Авербах А.В.	15	22	17,8	18	22
Разработка серверной части	Авербах А.В.	50	75	60	60	73
Разработка клиентской части	Авербах А.В.	20	25	22	22	27
Проектирование сайта	Авербах А.В.	3	4	3,4	3	4
Разработка сайта	Авербах А.В.	5	7	5,8	6	7
Проведение закрытого тестирования разработанного ПО	Авербах А.В.	14	21	16,8	17	20
Написание пояснительной записки к ВКР	Авербах А.В.	14	21	16,8	17	20

Для визуализации продолжительности этапов разработки была построена диаграмма Ганта, представленная на рисунке 41.

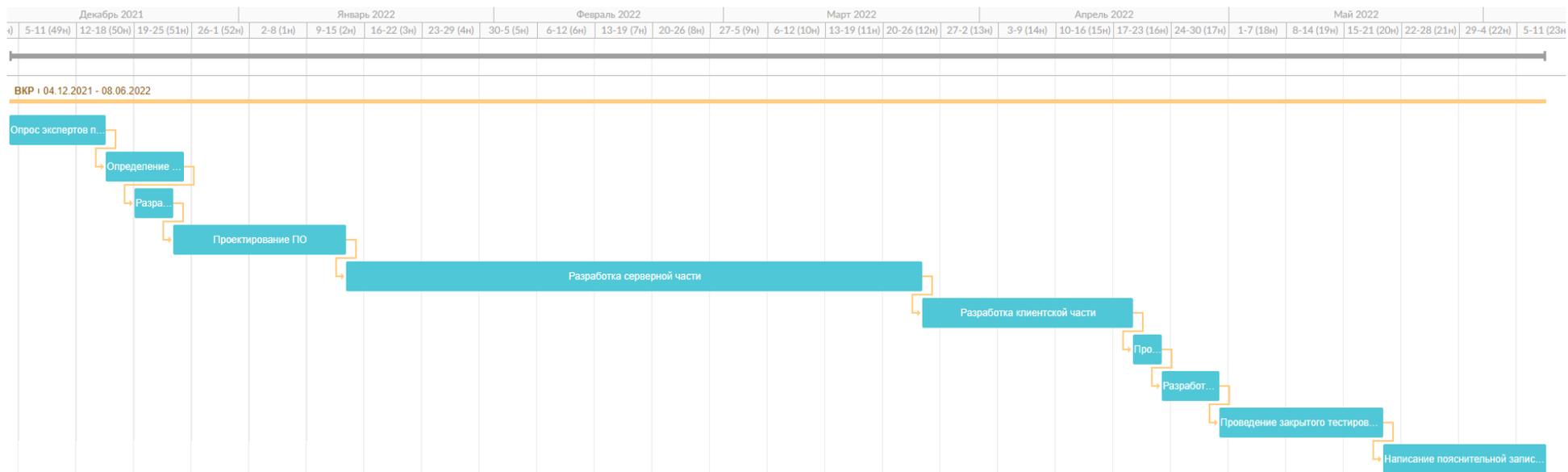


Рисунок 41 – Диаграмма Ганта

Ориентировочные даты выполнения работы: с 4 декабря 2021 года по 8 июня 2022 года.

### 5.3. Бюджет научно-технического исследования

Для обеспечения полного и достоверного отражения всех видов расходов, связанных с выполнением научно-технического исследования, необходимо провести бюджетное планирование проекта. Уделение данному вопросу должного внимания позволит облегчить планирование и координацию деятельности, а также сделать прозрачными все мероприятия и расходуемые ими ресурсы, что существенно повышает эффективность работ.

#### 5.3.1. Расчет материальных затрат НИИ

Так как затраты на электроэнергию, услуги связи, канцелярские принадлежности и т.д. учитываются как накладные расходы и не относятся в категорию материальных затрат, то сумма материальных затрат равняется нулю.

#### 5.3.2. Расчет затрат на специальное оборудование для научных работ

При выполнении работы использовался ноутбук стоимостью 41 тыс. руб. и компьютерная мышь стоимостью 1100 руб. Затраты на специальное оборудование отражены в таблице ниже.

Таблица 5 – Расчет бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество	Цена за ед., руб.
Ноутбук	шт.	1	41000
Компьютерная мышь	шт.	1	1100
Итого			42100

Согласно постановлению Правительства РФ от 01.01.2002 N 1 (ред. от 27.12.2019) «О Классификации основных средств, включаемых в амортизационные группы» срок полезного использования офисных машин (код

330.28.23.23) составляет от 2 до 3 лет. Для вычисления амортизации данный срок можно принять за 3 года.

Норма амортизации вычисляется как:

$$A_H = \frac{100\%}{3} = 33,33\%$$

Годовые амортизационные отчисления составляют:

$$A_G = S \cdot \frac{A_H}{100\%} = 42100 \cdot \frac{33,33\%}{100\%} = 14032 \text{ руб.}$$

Ежемесячные амортизационные отчисления составляют:

$$A_M = \frac{A_G}{12} = \frac{14032}{12} = 1169,3 \text{ руб.}$$

Итого за весь период выполнения исследовательской работы, с учётом того, что его продолжительность составляет 6,5 месяцев, сумма амортизации равна:

$$A = A_M \cdot 6,5 = 1169,3 \cdot 6,5 = 7600,45 \text{ руб.}$$

### 5.3.3. Основная заработная плата исполнителей

Расчёт баланса рабочего времени приведён в таблице 6.

Таблица 6 – Баланс рабочего времени (для 6-тидневной рабочей недели)

Показатели рабочего времени	Дни
Календарное число дней	365
Количество нерабочих дней	
- выходные дни	66
- праздничные дни	
Потери рабочего времени	
- отпуск	72
- невыходы по болезни	0
Действительный годовой фонд рабочего времени	227

Среднедневная заработная плата рассчитывается по формуле:

$$Z_{\text{дн}} = \frac{Z_{\text{м}} \cdot M}{F_{\text{д}}}$$

где  $Z_{\text{м}}$  – месячный должностной оклад работника, руб.;

$M$  – количество месяцев работы без отпуска в течение года:

при отпуске в 24 раб. дня  $M = 11,2$  месяца, 5–дневная неделя;

при отпуске в 48 раб. дней  $M = 10,4$  месяца, 6–дневная неделя;

$F_{\text{д}}$  – действительный годовой фонд рабочего времени научно–технического персонала, раб. дн.

$$Z_{\text{дн}} = \frac{40000 \cdot 10,4}{227} = 1832,6 \text{ руб.}$$

В таблице ниже приведен расчет основной заработной платы.

Таблица 7 – Расчет основной заработной платы

Исполнители	$Z_{\text{дн}}$ , руб.	$k_{\text{пр}}$	$k_{\text{д}}$	$k_{\text{р}}$	$T_{\text{р}}$ , раб. дн.	$Z_{\text{осн}}$ , руб.
Авербах А.В.	1832,6	0,2	0,3	1,3	160	571771,2
Итого						571771,2

Общий размер основной заработной платы составил 571771,2 руб.

#### 5.3.4. Дополнительная заработная плата

Дополнительная зарплата назначается за совмещение работы с учёбой, при предоставлении ежегодного оплачиваемого отпуска и пр. Дополнительная заработная плата рассчитывается умножением на надбавочный коэффициент ( $k_{\text{доп}}$ ). Величина надбавочного коэффициента принята за 0,12.

Расчет дополнительной заработной платы:

$$Z_{\text{доп}} = k_{\text{доп}} \cdot Z_{\text{осн}} = 0,12 \cdot 571771,2 = 68612,54 \text{ руб.}$$

Сумма основной и дополнительной заработной платы равняется 640383,74 руб.

### 5.3.5. Отчисления во внебюджетные фонды

В данной статье расходов отражаются обязательные отчисления по установленным законодательством Российской Федерации нормам органам государственного социального страхования (ФСС), пенсионного фонда (ПФ) и медицинского страхования (ФФОМС) от затрат на оплату труда работников.

Величина отчислений во внебюджетные фонды определяется исходя из формулы:

$$З_{\text{внеб}} = k_{\text{внеб}} \cdot (З_{\text{осн}} + З_{\text{доп}})$$

где  $k_{\text{внеб}}$  – коэффициент отчислений на уплату во внебюджетные фонды (пенсионный фонд, фонд обязательного медицинского страхования и пр.).

В соответствии с Федеральным законом от 24.07.2009 №212-ФЗ установлен размер страховых взносов равный 30,2%.

Величина отчислений во внебюджетные фонды составляет:

$$З_{\text{внеб}} = 0,302 \cdot (571771,2 + 68612,54) = 193395,9 \text{ руб.}$$

### 5.3.6. Накладные расходы

Накладные расходы учитывают прочие затраты организации, не попавшие в предыдущие статьи расходов. Их величина определяется по формуле:

$$З_{\text{накл}} = \left( \sum \text{статей} \right) \cdot k_{\text{нр}}$$

где  $k_{\text{нр}}$  – коэффициент, учитывающий накладные расходы (в данной работе принимается равным 0,16).

Величина накладных расходов составляет:

$$\begin{aligned} З_{\text{накл}} &= (0 + 42100 + 7600,45 + 571771,2 + 68612,54 + 193395,9) \cdot 0,16 \\ &= 141356,81 \text{ руб.} \end{aligned}$$

### **5.3.7. Формирование бюджета затрат научно-исследовательского проекта**

Рассчитанная величина затрат научно–исследовательской работы является основой для формирования бюджета затрат проекта. Определение бюджета затрат на научно–исследовательский проект приведено в таблице 8.

Таблица 8 – Расчет бюджета затрат НИИ

Наименование статьи	Сумма, руб.	Примечание
Материальные затраты НИИ	0	Пункт 1.4.1
Затраты на специальное оборудование для научных работ и его амортизацию	49700,45	Пункт 1.4.2
Затраты по основной заработной плате исполнителей	571771,2	Пункт 1.4.3
Затраты по дополнительной заработной плате исполнителей	68612,54	Пункт 1.4.4
Отчисления во внебюджетные фонды	193395,9	Пункт 1.4.5
Накладные расходы	141356,81	Пункт 1.4.6
Бюджет затрат НИИ	1024836,9	

### **5.4. Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования**

Для определения эффективности работы необходимо рассчитать интегральный показатель эффективности научного исследования, нахождение которого связано с определением финансовой эффективности и ресурсоэффективности. При расчёте показателей эффективности оценивались 2 варианта исполнения системы:

1. Серверная часть реализуется на языке программирования C++.
2. Серверная часть реализуется на языке программирования Python.

Стоимость разработки в этих двух вариантах является практически одинаковой, поэтому отдельный расчет для второго варианта не производился. Как следствие, интегральный финансовый показатель для обоих вариантов примем равным единице.

Существенным различием (и выгодным отличием первого варианта) является возможность масштабирования и развития первого варианта в будущем, в то время как второй вариант из-за технических особенностей вынужден будет столкнуться с ограниченной масштабируемостью, что повлечет более высокие затраты на дальнейшее развитие.

Расчет интегральных показателей ресурсоэффективности обоих вариантов исполнения приведен в таблице Таблица 9.

Таблица 9 - Сравнительная оценка характеристик вариантов исполнения проекта

Критерии / Объект исследования	Весовой коэффициент параметра	Исп.1	Исп.2
1. Гибкость настройки	0,20	5	4
2. Понятность и удобство пользовательского интерфейса	0,15	5	5
3. Актуальность предоставляемых данных	0,10	5	4
4. Нагрузка на компьютер пользователя	0,15	4	3
5. Лёгкость масштабирования и дальнейшего развития	0,4	5	1
Итого	1	4,85	2,8

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных. Расчет приведен в таблице ниже:

Таблица 10 – Сравнительная эффективность разработки

№	Показатели	Исп.1	Исп.2
1	Интегральный финансовый показатель разработки	1	1
2	Интегральный показатель ресурсоэффективности разработки	4,85	2,8
3	Интегральный показатель эффективности	4,85	2,8
4	Сравнительная эффективность вариантов исполнения	1	0,58

Сравнив значения интегральных показателей эффективности, можно сделать вывод, что реализация программного обеспечения в первом исполнении является более эффективным вариантом решения задачи, поставленной в данной работе с позиции финансовой и ресурсной эффективности.

### **Вывод по главе**

В главе исследовательской работы, посвященной финансовому менеджменту, ресурсоэффективности и ресурсосбережению, была дана оценка коммерческого потенциала разработки, спланирован график работ, сформирован бюджет затрат и определена эффективность исследования. Также проведен SWOT-анализ, на основе возможностей, угроз, сильных и слабых сторон проекта сделаны выводы.

Кроме того, выполнено планирование научно-исследовательских работ по проекту. При планировании графика работ был составлен список задач. График работ визуализирован в виде диаграммы Ганта. Общая длительность проведения работ по проекту ориентировочно составляет 194 календарных дня. Сформированный бюджет затрат научного исследования равен 1024836,9 руб.

Сравнение интегральных показателей эффективности вариантов исполнения показывает, что наиболее выгодным с позиции финансовой и ресурсной эффективности является первый вариант исполнения (реализация серверной части на языке C++), который и был реализован.

## **6. Социальная ответственность**

### **Введение**

В рамках выпускной квалификационной работы было разработано программное обеспечение для мониторинга торговой активности на криптовалютных биржах. Пользователями разработанного программного обеспечения являются трейдеры, торгующие на криптовалютных биржах (прежде всего на Binance и FTX) и использующие такие стили торговли, как скальпинг и интрадей-трейдинг.

Целью разработанного программного обеспечения является своевременно доводить до трейдера важную и актуальную рыночную информацию в компактном графическом и аудиальном представлении. Это позволяет не следить одновременно за всем разнообразием доступных криптовалютных инструментов, а сосредоточить своё внимание на тех из них, где в данный момент существуют интересующие пользователя условия.

Рабочая зона является офисным помещением площадью 18 м<sup>2</sup>. В рабочей зоне при помощи ноутбука осуществляются следующие процессы: поиск информации, её анализ; проектирование и разработка программного обеспечения.

В данной главе приведен анализ некоторых аспектов, оказывающих влияние на безопасность трудовой деятельности.

### **6.1. Правовые и организационные вопросы обеспечения безопасности**

#### **6.1.1. Правовые нормы трудового законодательства**

Трудовые отношения между работодателем и работником регулируются с помощью законодательного акта «Трудовой кодекс Российской Федерации» от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022)[23]. В нём регламентируются права и обязанности работодателя и работника, вопросы организации труда, режим труда и отдыха, оплата и нормирование труда, компенсации работникам, защита

персональных данных работника, урегулирование трудовых споров. Ниже приведены несколько наиболее важных положений:

- Нормальная продолжительность рабочего времени не должна превышать 40 часов в неделю.
- В течение рабочего дня (смены) работнику должен быть предоставлен перерыв для отдыха и питания продолжительностью не более двух часов и не менее 30 минут, который в рабочее время не включается. Правилами внутреннего трудового распорядка или трудовым договором может быть предусмотрено, что указанный перерыв может не предоставляться работнику, если установленная для него продолжительность ежедневной работы (смены) не превышает четырех часов (в ред. Федерального закона от 18.06.2017 N 125-ФЗ).
- Всем работникам предоставляются выходные дни (еженедельный непрерывный отдых).

#### **6.1.2. Эргономические требования к правильному расположению и компоновке рабочей зоны**

Основными элементами рабочего места программиста являются: рабочий стол, рабочий стул (кресло), дисплей, клавиатура, мышь; вспомогательными – пюпитр, подставка для ног.

Согласно ГОСТ Р 50923-96 «Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения»[24], рабочее место с дисплеем должно обеспечивать оператору возможность удобного выполнения работ в положении сидя и не создавать перегрузки костно-мышечной системы.

Конструкция рабочего стола должна обеспечивать возможность размещения на рабочей поверхности необходимого комплекта оборудования и документов с учетом характера выполняемой работы. Регулируемая высота рабочей поверхности стола должна изменяться в пределах от 680 до 800 мм.

Размеры рабочей поверхности стола должны быть: глубина - не менее 600 мм, ширина - не менее 1200 мм.

Дисплей на рабочем месте оператора должен располагаться так, чтобы изображение в любой его части было различимо без необходимости поднять или опустить голову. Дисплей на рабочем месте должен быть установлен ниже уровня глаз оператора. Угол наблюдения экрана оператором относительно горизонтальной линии взгляда не должен превышать 60°.

Клавиатура должна иметь возможность свободного перемещения. Клавиатуру следует располагать на поверхности стола на расстоянии от 100 до 300 мм от переднего края, обращенного к оператору, или на специальной регулируемой по высоте рабочей поверхности, отделенной от основной столешницы.

В соответствии с ГОСТ 21889-76 «Система "Человек-машина". Кресло человека-оператора»[25] кресло оператора может быть с профилированными и непрофилированными элементами. Поверхность сиденья может быть плоской с наклоном 0-5°, или профилированной с углами наклона сиденья. Опорная плоскость сиденья может быть плоской или профилированной с радиусом кривизны поясничной опоры, равным 460 мм, радиусом изгиба для грудного отдела позвоночника, равным 620 мм и другими точками изгиба.

При размещении рабочих мест необходимо исключить возможность прямой засветки экрана источником естественного освещения.

При выполнении выпускной квалификационной работы правовых и организационных нарушений по указанным требованиям не было выявлено, рабочее место было оборудовано согласно всем нормам и правилам.

## **6.2. Производственная безопасность**

В данном подразделе приведен анализ вредных и опасных факторов, которые воздействуют на разработчиков программного обеспечения, выполняющих работы на своих рабочих местах.

Все выявленные факторы приведены в таблице 11.

Таблица 11 – Возможные опасные и вредные производственные факторы на рабочем месте «офис»

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
1. Статические физические перегрузки, связанные с рабочей позой.	ГОСТ 12.0.003-2015 «Система стандартов безопасности труда. Опасные и вредные факторы»[26]
2. Умственное перенапряжение, в том числе вызванное информационной нагрузкой.	МР 2.2.9.2311-07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности» [27]
3. Перенапряжение анализаторов, в том числе вызванное информационной нагрузкой.	МР 2.2.9.2311-07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности» [27]
4. Монотонность труда, вызывающая монотонию.	МР 2.2.9.2311-07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности» [27]
5. Отсутствие или недостаток необходимого естественного освещения.	СП 52.13330.2016 «Естественное и искусственное освещение» [28]
6. Отсутствие или недостаток необходимого искусственного освещения.	СП 52.13330.2016 «Естественное и искусственное освещение» [28]
7. Производственные факторы, связанные с электрическим током, вызываемым разницей электрических	ГОСТ Р 12.1.019-2017 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты» [29]

потенциалов, под действие которого попадает работающий.	
---	--

Из данной таблицы следует, что на разработчиков программного обеспечения в ходе их деятельности воздействуют только физические и психологические факторы, а химические и биологические факторы отсутствуют.

### **6.2.1. Статические физические перегрузки**

Работа программиста является малоподвижной деятельностью, а значит может вызывать гиподинамию. Как известно, ограничение физической активности приводит к нарушениям в работе опорно-двигательного аппарата, сердечно-сосудистой системы, желудочно-кишечного тракта. Все эти нарушения оказывают значительное влияние на состояние организма работника и негативно сказываются как на его повседневной жизни, так и на качестве его работы.

Для профилактики гиподинамии следует предпринимать следующие меры:

- увеличение физической активности;
- регулярные перерывы в работе для небольшой разминки.

### **6.2.2. Умственное перенапряжение и перенапряжение анализаторов, монотонность труда**

Монотонный труд является частью практически любого процесса разработки, так как каждый специалист выполняет ряд каких-либо действий, связанных с его задачами. Кроме того, программисты по роду своей деятельности проводят много времени, решая сложные и нестандартные задачи и проблемы, из-за чего может возникать умственное перенапряжение (переутомление). Этот негативный фактор играет важную роль в эмоциональном состоянии. При умственном перенапряжении у сотрудника могут наблюдаться следующие негативные последствия:

- нарушение сна или сонливость;
- отсутствие или повышение аппетита;

- приступы тошноты, обмороки;
- головная боль;
- раздражительность, нервозность;
- усталость, апатия;
- медлительность;
- снижение внимательности и работоспособности.

Для снижения уровня умственного перенапряжения разработчика программного обеспечения следует принимать следующие меры:

- во время рабочего дня делать регулярные перерывы;
- вне рабочего времени проводить время на свежем воздухе;
- нормализовать режим сна;
- регулярно и сбалансировано питаться;
- выделять время в течении дня на разминку или на полноценные тренировки;
- стараться организовывать свое рабочее время таким образом, чтобы не заниматься одной задачей на протяжении длительного времени;
- крупные задачи делить на небольшие подзадачи.

Согласно МР 2.2.9.2311-07 [27], меры профилактики стрессовых состояний предусматривают внедрение рациональных режимов труда и отдыха, комплекса оздоровительно-профилактических мероприятий для предупреждения воздействия стресс-факторов на организм работающих.

При постоянном взаимодействии с компьютером (набор текстов, ввод данных и т.п.), при исключении возможности периодического переключения на другие виды трудовой деятельности, не связанные с ПЭВМ, рекомендуется организация перерывов по 10 - 15 мин. через каждые 45 - 60 мин. работы.

Для повышения работоспособности, снижения зрительного утомления рекомендуется использовать очки защитные со спектральными фильтрами ЛС и НСФ, разрешенные Минздравом России для работы с ПЭВМ.

### **6.2.3. Отсутствие или недостаток необходимого естественного освещения**

Нехватка естественного освещения или его полное отсутствие в течение рабочего дня может повлечь болезни психологического характера, невротические расстройства, снизить работоспособность и концентрацию сотрудника.

Источником естественного освещения является солнечный свет, основными причинами его недостатка являются географическое положение здания, а также планировка его внутренних помещений, расположение рабочих мест. Окна в помещениях, где эксплуатируется вычислительная техника, преимущественно должны быть ориентированы на север и северо-восток. При этом столы следует располагать так, чтобы свет от окон падал на рабочую поверхность стола слева.

Разработка программного обеспечения относится к работам высокой точности (наименьший или эквивалентный объект различения 0,30 – 0,50 мм), разряд Б, подразряд 1, относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%. Требования к естественному освещению рабочего помещения согласно СП 52.13330.2016 «Естественное и искусственное освещение»[28] представлены в таблице 12.

Таблица 12 – Требования к естественному освещению рабочего помещения

Коэффициент естественной освещенности, %, при	
верхнем или комбинированном освещении	боковом освещении
3,0	1,0

При недостаточной освещенности помещения может помочь расширение оконных проемов и установка качественных источников искусственного освещения.

#### **6.2.4. Отсутствие или недостаток необходимого искусственного освещения**

Недостаточная освещенность рабочей зоны является вредным производственным фактором, приводящим к быстрому утомлению и снижению работоспособности человека на предприятии. При недостаточной освещенности помещения человек быстрее устает, снижается внимание и концентрация. Продолжительная работа в условиях низкой освещенности приводит к ухудшению зрения.

Нормы естественного, искусственного и совместного освещения регламентируются СП 52.13330.2016 «Естественное и искусственное освещение»[28]. Разработка программного обеспечения относится к работам высокой точности (наименьший или эквивалентный объект различения 0,30 – 0,50 мм), разряд Б, подразряд 1, относительная продолжительность зрительной работы при направлении зрения на рабочую поверхность не менее 70%.

В таблице 13 представлены требования к искусственному освещению рабочего помещения.

Таблица 13 – Требования к искусственному освещению рабочего помещения

Искусственное освещение			
Освещенность на рабочей поверхности от системы общего освещения, лк	Цилиндрическая освещенность, лк	Объединенный показатель UGR, не более	Коэффициент пульсации освещенности, Кп, %, не более
400	100	19	15

Искусственное освещение в помещениях для эксплуатации ПК должно осуществляться системой общего равномерного освещения. В случаях преимущественной работы с документами, следует применять системы

комбинированного освещения (к общему освещению дополнительно устанавливаются светильники местного освещения).

Яркий свет в зоне периферийного зрения заметно увеличивает напряжение глаз. Для снижения влияния вредного фактора недостаточной освещенности необходимо, чтобы уровень освещения рабочего пространства приблизительно совпадал с яркостью дисплея. Проблему недостаточной освещенности помещения можно решить при помощи установки дополнительных осветительных приборов.

#### **6.2.5. Опасность поражения электрическим током**

Работа программиста происходит в непосредственной близости от электрических сетей и приборов, поэтому работник должен с осторожностью обращаться с электропроводкой и компьютером, а также помнить об опасности поражения электрическим током.

Покрытие полов следует делать из однослойного линолеума, что снизит величины зарядов статического электричества. Несмотря на то, что эти величины безопасны для здоровья человека, вычислительная техника подвергается опасности при воздействии зарядов такого рода.

Поражение электрическим током является опасным производственным фактором. Вероятность поражения повышается при:

- повышенной влажности воздуха в помещении (более 75%);
- высокой температуре воздуха и поверхностей (более 35 °С);
- наличии токопроводящей пыли;
- неверной проектировке рабочего места;
- отсутствии защитных конструкций для проводов;
- наличии посторонних предметов на электроприборах.

Мерами защиты от воздействия электрического тока при неисправности изоляции являются защитное заземление, зануление[29] и использование устройств защитного отключения (УЗО).

### **6.3. Экологическая безопасность**

В ходе выполнения выпускной квалификационной работы использовался ноутбук.

Утилизация компьютерной и организационной техники ограничена законодательно, так как в производстве такой техники используются материалы, способные нанести вред окружающей среде (класс опасности IV). Утилизация компьютерного оборудования происходит через обязательное извлечение компонентов, их сортировку и последующую отправку для повторного использования. Такая утилизация обязательно производится на оборудованных полигонах с привлечением квалифицированного персонала[30].

### **6.4. Безопасность в чрезвычайных ситуациях**

Возможные ЧС:

- Техногенные (взрывы, пожары, обрушение помещений);
- Природные (наводнения, ураганы, бури, природные пожары);
- Биологические (эпидемии, пандемии);
- Антропогенные (война, терроризм).

При работе с вычислительной техникой наиболее типичной чрезвычайной ситуацией является пожар, так как в современных ЭВМ очень высокая плотность размещения элементов электронных схем. К причинам электрического характера можно отнести короткое замыкание, искрение, статическое электричество.

Для предотвращения возникновения пожара необходимо:

- Регулярно проводить инструктажи по пожарной безопасности.
- Разместить в помещении план эвакуации и плакаты с краткой информацией о действиях, предпринимаемых при возникновении пожара.

- Соблюдать правила и нормы при монтаже электрических приборов и электрической проводки.
- Оборудовать помещение пожарной сигнализацией, а также средствами тушения пожара.

В случае возникновения пожара каждый сотрудник должен:

- незамедлительно сообщить об этом в пожарную охрану;
- принять меры по эвакуации людей, материальных ценностей согласно плану эвакуации;
- отключить электроэнергию, приступить к тушению пожара первичными средствами пожаротушения.

Возможный пожар на рабочем месте относится к классам А1, А2, Е[31].  
Первичные средства пожаротушения являются: огнетушители порошковые переносные с порошками типа АВСЕ, огнетушители углекислотные.

## **Выводы по главе**

В результате работы по главе «Социальная ответственность» были выявлены основные нормативные акты для обеспечения безопасности жизнедеятельности на рабочем месте, рассмотрены наиболее значимые опасные и вредные факторы, возникающие при проектировании и разработке программного обеспечения для мониторинга торговой активности на криптовалютных биржах, описано влияние процесса разработки программного обеспечения на окружающую среду и меры, необходимые для уменьшения влияния вредных и опасных факторов на организм человека и для сокращения негативного влияния процесса разработки программного обеспечения на окружающую среду.

Согласно пункту 1.1.13 ПУЭ-7 рабочая зона является помещением без повышенной опасности. Согласно «Правилам по охране труда при эксплуатации электроустановок»[32] персонал должен иметь первую группу по электробезопасности.

Работа в офисе относится к категории тяжести труда Ia[33] – работы выполняются при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки.

Рабочая зона относится к категории В по взрывопожарной и пожарной опасности.

В заключение можно отметить, что нарушений по организации рабочего процесса при выполнении ВКР выявлено не было, а все необходимые требования и нормы безопасности были соблюдены.

## Заключение

В результате выполнения выпускной квалификационной работы было разработано программное обеспечение для мониторинга торговой активности на криптовалютных биржах, целью которого является своевременно доводить до трейдера важную и актуальную рыночную информацию в компактном графическом и аудиальном представлении.

Основные результаты проведенной работы:

- определены потребности пользователей и основной функционал программного обеспечения;
- спроектировано программное обеспечение и выбраны средства его реализации;
- программное обеспечение реализовано;
- программное обеспечение предоставлено пользователям для тестирования и получения обратной связи.

Программное обеспечение реализовано по клиент-серверной модели, однако и клиент, и сервер запускаются на устройстве пользователя. Сервер реализован на языке C++ с использованием библиотеки Boost.Asio. Клиент реализован с помощью фреймворков Vue.js и Quasar.

В главе, посвященной финансовому менеджменту, было произведено сравнение разработанного решения с конкурентами, составлен бюджет проекта и определены сроки реализации.

В главе «Социальная ответственность» проанализированы правовые и организационные вопросы обеспечения безопасности при разработке проектного решения, дана оценка производственной и экологической безопасности, а также безопасности в чрезвычайных ситуациях при разработке проектного решения.

Наиболее значимым результатом является факт использования разработанного программного обеспечения реальными конечными пользователями.

## Список использованных источников и литературы

1. Binance [Электронный ресурс] // – Режим доступа: <https://www.binance.com> (дата обращения: 29.05.2022)
2. CoinMarketCap [Электронный ресурс] // – Режим доступа: <https://coinmarketcap.com/ru/rankings/exchanges/> (дата обращения: 29.05.2022)
3. Tether [Электронный ресурс] // – Режим доступа: <https://tether.to> (дата обращения: 29.05.2022)
4. Мерфи, Джон Дж. Технический анализ финансовых рынков: полный справочник по методам и практике трейдинга / Джон Дж. Мерфи ; Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2016. – 496 с.
5. TradingView [Электронный ресурс] // – Режим доступа: <https://ru.tradingview.com> (дата обращения: 29.05.2022)
6. CScalp [Электронный ресурс] // – Режим доступа: <https://fsr-develop.ru> (дата обращения: 29.05.2022)
7. asyncio — Asynchronous I/O [Электронный ресурс] // – Режим доступа: <https://docs.python.org/3/library/asyncio.html> (дата обращения: 29.05.2022)
8. multiprocessing — Process-based parallelism [Электронный ресурс] // – Режим доступа: <https://docs.python.org/3/library/multiprocessing.html> (дата обращения: 29.05.2022)
9. Boost.Asio [Электронный ресурс] // – Режим доступа: [https://www.boost.org/doc/libs/1\\_79\\_0/doc/html/boost\\_asio.html](https://www.boost.org/doc/libs/1_79_0/doc/html/boost_asio.html) (дата обращения: 29.05.2022)
10. Callback Hell [Электронный ресурс] // – Режим доступа: <http://callbackhell.com> (дата обращения: 29.05.2022)
11. Coroutines (C++20) [Электронный ресурс] // – Режим доступа: <https://en.cppreference.com/w/cpp/language/coroutines> (дата обращения: 29.05.2022)

- 12.Boost.Beast [Электронный ресурс] // – Режим доступа: [https://www.boost.org/doc/libs/1\\_79\\_0/libs/beast/doc/html/index.html](https://www.boost.org/doc/libs/1_79_0/libs/beast/doc/html/index.html) (дата обращения: 29.05.2022)
- 13.OpenSSL Cryptography and SSL/TLS Toolkit [Электронный ресурс] // – Режим доступа: <https://www.openssl.org> (дата обращения: 29.05.2022)
- 14.The simdjson library [Электронный ресурс] // – Режим доступа: <https://simdjson.org> (дата обращения: 29.05.2022)
- 15.Simdjson: Performance results [Электронный ресурс] // – Режим доступа: <https://github.com/simdjson/simdjson#performance-results> (дата обращения: 29.05.2022)
- 16.fast\_float number parsing library: 4x faster than strtod [Электронный ресурс] // – Режим доступа: [https://github.com/fastfloat/fast\\_float](https://github.com/fastfloat/fast_float) (дата обращения: 29.05.2022)
- 17.robin\_hood unordered map & set [Электронный ресурс] // – Режим доступа: <https://github.com/martinus/robin-hood-hashing> (дата обращения: 29.05.2022)
- 18.Hashmaps Benchmarks - Overview [Электронный ресурс] // – Режим доступа: <https://martin.ankerl.com/2019/04/01/hashmap-benchmarks-01-overview/> (дата обращения: 29.05.2022)
- 19.spdlog [Электронный ресурс] // – Режим доступа: <https://github.com/gabime/spdlog> (дата обращения: 29.05.2022)
- 20.The Progressive JavaScript Framework [Электронный ресурс] // – Режим доступа: <https://v3.vuejs.org> (дата обращения: 29.05.2022)
- 21.Quasar Beyond the framework [Электронный ресурс] // – Режим доступа: <https://quasar.dev> (дата обращения: 29.05.2022)
- 22.Discord [Электронный ресурс] // – Режим доступа: <https://discord.com> (дата обращения: 29.05.2022)
- 23.Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022) [Электронный ресурс] // – Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_34683/](http://www.consultant.ru/document/cons_doc_LAW_34683/) (дата обращения: 19.05.2022).

- 24.ГОСТ Р 50923-96 «Дисплеи. Рабочее место оператора. Общие эргономические требования и требования к производственной среде. Методы измерения» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/1200025975> (дата обращения: 19.05.2022).
- 25.ГОСТ 21889-76 «Система "Человек-машина". Кресло человека-оператора» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/1200012832> (дата обращения: 19.05.2022).
- 26.ГОСТ 12.0.003-2015 «Система стандартов безопасности труда. Опасные и вредные факторы» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/1200136071> (дата обращения: 19.05.2022).
- 27.МР 2.2.9.2311-07 «Профилактика стрессового состояния работников при различных видах профессиональной деятельности» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/1200072234> (дата обращения: 19.05.2022).
- 28.СП 52.13330.2016 «Естественное и искусственное освещение» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/456054197> (дата обращения: 19.05.2022).
- 29.ГОСТ Р 12.1.019-2017 «Система стандартов безопасности труда (ССБТ). Электробезопасность. Общие требования и номенклатура видов защиты» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/1200161238> (дата обращения: 19.05.2022).
- 30.ГОСТ Р 53692-2009 «Ресурсосбережение. Обращение с отходами. Этапы технологического цикла отходов» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/1200081740> (дата обращения: 29.05.2022).
- 31.Федеральный закон от 22.07.2008 N 123-ФЗ (ред. от 30.04.2021) «Технический регламент о требованиях пожарной безопасности» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/902111644> (дата обращения: 29.05.2022).

32. Приказ Минтруда России от 15 декабря 2020 года N 903н «Об утверждении Правил по охране труда при эксплуатации электроустановок» [Электронный ресурс] // – Режим доступа: <https://docs.cntd.ru/document/573264184> (дата обращения: 29.05.2022).
33. Приказ Минтруда России от 24.01.2014 N 33н (ред. от 27.04.2020) «Об утверждении Методики проведения специальной оценки условий труда, Классификатора вредных и (или) опасных производственных факторов, формы отчета о проведении специальной оценки условий труда и инструкции по ее заполнению» [Электронный ресурс] // – Режим доступа: [http://www.consultant.ru/document/cons\\_doc\\_LAW\\_158398/](http://www.consultant.ru/document/cons_doc_LAW_158398/) (дата обращения: 29.05.2022).