

Школа: Инженерная школа информационных технологий и робототехники (ИШИТР)

Направление подготовки: 09.03.04 «Программная инженерия»

ООП/ОПОП: Разработка программно-информационных систем

Отделение школы: Отделение информационных технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

Тема работы
Разработка маркетплейса для торговли невзаимозаменяемыми токенами в сети Polygon

УДК 004.738.5:339:336.763-028.27

Обучающийся

Группа	ФИО	Подпись	Дата
8K92	Галась Данил		

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

КОНСУЛЬТАНТЫ ПО РАЗДЕЛАМ:

По разделу «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение»

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Профессор ОСГН	Гасанов Магеррам Али оглы	Д.Э.Н.		

По разделу «Социальная ответственность»

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

ДОПУСТИТЬ К ЗАЩИТЕ:

Руководитель ООП/ОПОП, должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ООП/ОПОП
«Разработка программно-информационных систем»

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
ОПК(У)-1	Способен применять естественнонаучные и общетехнические 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 использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов
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.
ПК(У)-1	Владение навыками разработки требований и проектирования программного обеспечения
ПК(У)-2	Владение навыками разработки документов и стратегии тестирования программного обеспечения
ПК(У)-3	Владение навыками моделирования, анализа и использования формальных методов конструирования программного обеспечения
ПК(У)-4	Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных
ПК(У)-5	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 (ИШИТР)

Направление подготовки: 09.03.04 «Программная инженерия»

Отделение школы: Отделение информационных технологий

УТВЕРЖДАЮ:
 Руководитель ООП/ОПОП
 _____ Чердынцев Е.С.
 (Подпись) (Дата) (ФИО)

ЗАДАНИЕ

на выполнение выпускной квалификационной работы

Обучающийся:

Группа	ФИО
8K92	Галась Данил

Тема работы:

<i>Разработка маркетплейса для торговли невзаимозаменяемыми токенами в сети Polygon</i>	
<i>Утверждена приказом директора (дата, номер)</i>	<i>12.04.2023, 102-29/с</i>

Срок сдачи обучающимся выполненной работы:

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Исходные данные к работе <i>(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 функционированию (эксплуатации) объекта или изделия в плане безопасности эксплуатации, влияния на окружающую среду, энергозатратам; экономический анализ и т. д.)</i>	Разработка маркетплейса невзаимозаменяемых токенов в сети Polygon – децентрализованное приложение, обеспечивающее функционал для торговли NFT-токенами в сети Polygon.
Перечень разделов пояснительной записки подлежащих исследованию, проектированию и разработке <i>(аналитический обзор литературных источников 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)</i>	1. Анализ предметной области 2. Обзор конкурентных решений 3. Проектирование архитектуры приложения 4. Программная реализация
Перечень графического материала <i>(с точным указанием обязательных чертежей)</i>	Диаграмма архитектуры UUPS Диаграмма вариантов использования Диаграммы классов Диаграмма IDEF0 и IDEF3

Консультанты по разделам выпускной квалификационной работы <i>(с указанием разделов)</i>	
Раздел	Консультант
Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	Гасанов Магеррам Али оглы
Социальная ответственность	Мезенцева Ирина Леонидовна

Дата выдачи задания на выполнение выпускной квалификационной работы по линейному графику	4.03.2023
---	-----------

Задание выдал руководитель:

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

Задание принял к исполнению обучающийся:

Группа	ФИО	Подпись	Дата
8K92	Галась Данил		05.03.2023



Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Школа: Инженерная школа информационных технологий и робототехники (ИШИТР)

Направление подготовки: 09.03.04 «Программная инженерия»

Уровень образования: бакалавр

Отделение школы: Отделение информационных технологий

Период выполнения: весенний семестр 2022/2023 учебного года

КАЛЕНДАРНЫЙ РЕЙТИНГ-ПЛАН выполнения выпускной квалификационной работы

Обучающийся:

Группа	ФИО
8K92	Галась Данил

Тема работы:

<i>Разработка маркетплейса для торговли невзаимозаменяемыми токенами в сети Polygon</i>

Срок сдачи обучающимся выполненной работы:	
--	--

Дата контроля	Название раздела (модуля) / вид работы (исследования)	Максимальный балл раздела (модуля)
15.03.2023	Анализ предметной области	25
25.04.2023	Проектирование программной системы	25
21.05.2023	Разработка программной системы	20
23.05.2023	Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	15
23.05.2023	Социальная ответственность	15

СОСТАВИЛ:

Руководитель ВКР

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

СОГЛАСОВАНО:

Руководитель ООП/ОПОП

Должность	ФИО	Ученая степень, звание	Подпись	Дата
Доцент ОИТ ИШИТР ТПУ	Чердынцев Евгений Сергеевич	К.Т.Н.		

Обучающийся

Группа	ФИО	Подпись	Дата
8K92	Галась Данил		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «ФИНАНСОВЫЙ МЕНЕДЖМЕНТ, РЕСУРСОЭФФЕКТИВНОСТЬ И РЕСУРСОСБЕРЕЖЕНИЕ»

Студенту:

Группа	ФИО
8K92	Галась Данил

Школа	Инженерная школа информационных технологий и робототехники	Отделение школы (НОЦ)	Отделение информационных технологий
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>	Стоимость материальных ресурсов и специального оборудования определены в соответствии с рыночными ценами г. Томска. Нормативные документы НИ ТПУ, ФЗ «О минимальном размере оплаты труда» для определения оплата труда исполнителей проекта.
2. <i>Нормы и нормативы расходования ресурсов</i>	Тариф на эл. энергию – 3,16 руб./квт·час Районный коэффициент 30%.
3. <i>Используемая система налогообложения, ставки налогов, отчислений, дисконтирования и кредитования</i>	Коэффициент отчислений на уплату во внебюджетные фонды 30 %

Перечень вопросов, подлежащих исследованию, проектированию и разработке:

1. <i>Оценка коммерческого потенциала, перспективности и альтернатив проведения НИ с позиции ресурсоэффективности и ресурсосбережения</i>	Оценить потенциальных потребителей исследования, проанализировать конкурентных решений, представить SWOT – анализ. Предложить возможные альтернативы проведения НИ.
2. <i>Планирование и формирование бюджета научных исследований</i>	Представить план этапов работ, определить трудоёмкость и построить календарный график, сформировать бюджет НИ.
3. <i>Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования</i>	Определить интегральные показатели финансовой эффективности, ресурсоэффективности разработки. Рассчитать сравнительную эффективность проекта.

Перечень графического материала (с точным указанием обязательных чертежей):

1. Оценочная карта для сравнения конкурентных технических решений. 2. Матрица SWOT-анализа 3. Морфологическая матрица 4. Перечень этапов, работ и распределение исполнителей 5. Временные показатели проведения НИ 6. Бюджет НИ 7. Оценка характеристик вариантов исполнения 8. Сравнительная эффективность разработки.

Дата выдачи задания для раздела по линейному графику

Задание выдал консультант:

Должность	ФИО	Ученая степень	Подпись	Дата
Профессор ОСГН	Гасанов Магеррам Али оглы	Д.э.н.		

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K92	Галась Данил		

ЗАДАНИЕ ДЛЯ РАЗДЕЛА «СОЦИАЛЬНАЯ ОТВЕТСТВЕННОСТЬ»

Студенту:

Группа		ФИО	
8К92		Галась Данил	
Школа	ИШИТР	Отделение (НОЦ)	ОИТ
Уровень образования	Бакалавриат	Направление/специальность	09.03.04 Программная инженерия

Тема ВКР:

Разработка маркетплейса для торговли невзаимозаменяемыми токенами в сети Polygon	
Исходные данные к разделу «Социальная ответственность»:	
Введение – Характеристика объекта исследования (вещество, материал, прибор, алгоритм, методика) и области его применения. – Описание рабочей зоны (рабочего места) при разработке проектного решения/при эксплуатации	<i>Объект исследования:</i> децентрализованное приложение <i>Область применения:</i> энтузиасты в сфере распределенных систем <i>Рабочая зона:</i> офис <i>Размеры помещения:</i> 6*4 м. <i>Количество и наименование оборудования рабочей зоны:</i> компьютер - 1 ед. <i>Рабочие процессы, связанные с объектом исследования, осуществляющиеся в рабочей зоне:</i> разработка приложения
Перечень вопросов, подлежащих исследованию, проектированию и разработке:	
1. Правовые и организационные вопросы обеспечения безопасности при разработке проектного решения: – специальные (характерные при эксплуатации объекта исследования, проектируемой рабочей зоны) правовые нормы трудового законодательства; – организационные мероприятия при компоновке рабочей зоны.	Трудовой кодекс РФ; СанПиН 1.2.3685-21. Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания ТК РФ Статья 91. Понятие рабочего времени. Нормальная продолжительность рабочего времени; ГОСТ 12.2.032-78. Система стандартов безопасности труда. Рабочее место при выполнении работ сидя.

2. Производственная безопасность при разработке проектного решения: – Анализ выявленных вредных и опасных производственных факторов	Вредные: 1. Отсутствие или недостаток необходимого искусственного освещения; 2. Повышенный уровень шума; 3. Нагрузка на зрительный аппарат; 4. Монотонный режим работы. Опасные: 1. Факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает рабочий. Требуемые средства коллективной защиты от выявленных факторов: системы естественного освещения, приборы искусственного освещения, изоляционные средства, предохранительные устройства.
3. Экологическая безопасность при эксплуатации проектного решения	Воздействие на селитебную зону: не выявлено Воздействие на литосферу: загрязнение почв веществами от компьютерной техники, батареек и других элементов питания Воздействие на гидросферу: из-за неверного способа утилизации рабочей техники Воздействие на атмосферу: не выявлено
4. Безопасность в чрезвычайных ситуациях при разработке проектного решения	Возможные ЧС: Землетрясение, пожар Наиболее типичная ЧС: пожар
Дата выдачи задания для раздела по линейному графику	

Задание выдал консультант:

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

Задание принял к исполнению студент:

Группа	ФИО	Подпись	Дата
8K92	Галась Данил		

РЕФЕРАТ

Выпускная квалификационная работа содержит 82 страницы, 22 рисунка, 26 таблиц, 24 источника.

Ключевые слова: децентрализованное приложение, NFT, токен, маркетплейс, Solidity, OpenZeppelin.

Объектом исследования является децентрализованный маркетплейс невзаимозаменяемых токенов.

Целью работы является разработка децентрализованного приложения, обеспечивающего функционал для торговли NFT-токенами.

В процессе разработки децентрализованного приложения был осуществлён анализ предметной области, обзор конкурентных технических решений, определена архитектура программной системы, спроектированы компоненты программной системы.

Оглавление

Введение	13
1. Исследование предметной области.....	15
1.1 Описание области блокчейна и цифровых активов.....	15
1.2 Анализ существующих блокчейнов	16
1.3 Анализ существующих аналогов маркетплейсов	18
1.4 Вывод по теоретической части.....	19
2.1 Требования к системе	20
2.3 Диаграмма вариантов использования	22
2.3 Диаграмма классов.....	22
2.4. Функциональные модели IDEF0 и IDEF3	25
2.5 Структура проекта	29
3. Программная реализация	30
3.1 Обоснование выбора средств разработки.....	30
3.2 Описание разработанных частей проекта.....	33
3.2.1 NFT-токен стандарта ERC721.....	35
3.2.2 NFT-токен стандарта ERC1155.....	35
3.2.3 Фабрика NFT-токенов стандарта ERC721 и ERC1155	36
3.2.4 Первичный маркетплейс NFT-токенов	36
3.2.5 Вторичный маркетплейс NFT-токенов	37
3.3 Тестирование смарт-контрактов с использованием фреймворка Hardhat.....	37
4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение	41
4.1 Введение	41
4.2 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения	41
4.3 Определение возможных альтернатив проведения научных исследований	47
4.4 Планирование научно-исследовательских работ.....	49
4.4.1 Структура работ в рамках научного исследования	49
4.4.2 Определение трудоёмкости выполнения работ	50
4.4.3 Разработка графика проведения научного исследования	54
4.4.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования	59
5. Социальная ответственность	64
5.1 Введение	64
5.2 Правовые и организационные вопросы обеспечения безопасности	64

5.3 Производственная безопасность	65
5.4 Анализ опасных и вредных производственных факторов	66
5.4.1 Отсутствие или недостаток необходимого искусственного освещения.....	66
5.4.2 Повышенная пульсация светового потока.....	67
5.4.3 Нервно-психические перегрузки, связанные с напряжённостью трудового процесса	68
5.4.4 Статические перегрузки, связанные с рабочей позой	68
5.4.5 Длительное сосредоточенное наблюдение	69
5.4.6 Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий.....	69
5.5 Экологическая безопасность.....	69
5.6 Безопасность в чрезвычайных ситуациях.....	70
5.6.1 Анализ возможных ЧС	70
5.6.2 Анализ наиболее вероятной ЧС, которая может возникнуть при разработке приложения.....	70
5.7 Вывод по разделу	71
Список используемых источников.....	75
Приложение А.....	78

Введение

Технология блокчейн набирает обороты, поскольку открывает множество новых возможностей в самых разных областях: финансы, экономика, операции с материальными и нематериальными активами, учет в государственных и частных организациях и так далее [1].

Ярким примером **актуальности** данной работы служит тот факт, что уже в 2023 году в России планируется подготовить юридическую базу для использования цифровых финансовых активов (ЦФА) в трансграничных платежах [2].

В связи с быстрорастущей популярностью данной технологии возникает потребность в осуществлении обмена ЦФА между субъектами. При условии использования смарт-контрактов для решения этой проблемы, мы получаем следующие преимущества.

1. **Безопасность обмена** - при исследовании программного кода смарт-контрактов, каждый участник сделки сможет изучить условия сделки и принять взвешенное решение.
2. **Отсутствие доверенного лица** - в сделке принимают участие лишь ее участники, без посторонних гарантов.
3. **Прозрачность** - все этапы совершения сделки будут доступны в открытом доступе, что повышает доверие к системе.
4. **Децентрализация** - отсутствие цензуры при совершении сделок.

Самым популярным цифровым финансовым активом является невзаимозаменяемый токен, или NFT-токен. Он позволяет закрепить право владения за конкретным человеком в распределенном реестре – сети блокчейн [3]. Более того, невзаимозаменяемость гарантирует, что право владения есть лишь у определенного человека и никого более. Под невзаимозаменяемостью подразумевается уникальность каждого отдельного NFT-токена.

Целью работы является разработка децентрализованного приложения, позволяющего осуществлять торговлю невзаимозаменяемыми токенами между участниками сети блокчейна.

Для достижения поставленной цели необходимо решить следующие задачи.

1. Исследовать существующие блокчейны для разрабатываемого приложения.
2. Исследовать сторонние площадки для обмена цифровыми активами.
3. Проанализировать применяемые технологии в ходе создания приложения.
4. Спроектировать децентрализованное приложение.
5. Протестировать приложение в тестовой сети.

Практическая значимость приложения — обеспечения децентрализованной торговли цифровыми финансовыми активами, свободной от цензуры и ограничений.

1. Исследование предметной области

1.1 Описание области блокчейна и цифровых активов

Для рассмотрения проблематики торговли токенами стоит рассмотреть понятие блокчейна и цифровых финансовых активов в целом.

Блокчейн — новейшая технология, интерес к которой вырос вместе с популярностью криптовалют. Сегодня это одна из самых обсуждаемых технологий не только в сфере финансов, но и хранении и обработке персональных данных, маркетинге и компьютерных играх. [1]

С точки зрения баз данных, **блокчейн** – распределенный реестр данных, ключевая особенность которого невозможность удалить или изменить данные в реестре. Обусловлено это архитектурой блокчейна и наличием консенсуса, который позволяет достичь согласия по части текущего состояния данных среди всех узлов. Блокчейн можно представить как последовательную цепочку блоков, каждый из которых содержит **заголовок** и **тело**. Заголовок содержит данные, отвечающие за постоянство и неизменяемость системы, а тело содержит список всех транзакций, исполненных и записанных в данный блок [10].

Все блоки связаны друг с другом посредством хеш-суммы. Каждый новый блок содержит хеш-сумму старого, что и обеспечивает неизменяемость сети. Если один из узлов системы изменит данные в старом блоке, изменится и его хеш-сумма, что приведет к делению сети - **форку** [11].

Для синхронизации записей между всеми участниками сети, блокчейн использует алгоритм достижения консенсуса. Наиболее распространенные и широко используемые механизмы консенсуса - Proof-of-Work (PoW) и Proof-of-Stake (PoS) [12].

Есть блокчейны двух видов: тьюринг-полные и тьюринг-неполные. Полнота по Тьюрингу, применительно к блокчейну, означает, что смарт-контракты этого блокчейна имеют возможность решить любую вычислимую задачу. Примером тьюринг-полного блокчейна является **Ethereum**, тьюринг-неполного - **Bitcoin**.

Смарт-контракт (англ. *smart contract* — умный контракт) — компьютерный алгоритм, предназначенный для формирования, управления и предоставления информации о владении чем-либо. В более узком смысле под смарт-контрактом понимается набор функций и данных (текущее состояние), находящихся по определенному адресу в блокчейне. Благодаря

неизменности блокчейна, развернутые внутри сети приложения являются устойчивыми к цензуре, распределенными и исключают возможность исправить право на владение несанкционированно [13].

Токен — цифровой актив, который используется в качестве представления определенного значения или права в блокчейн сети. Токены могут иметь различные формы и назначения, но обычно они представляют определенную сумму криптовалюты или доступ к определенному сервису или ресурсу. Например, токен может быть использован для получения доступа к децентрализованной платформе, голосования на блокчейн-голосованиях или покупки товаров и услуг в блокчейн магазине. Каждый токен является смарт-контрактом и имеет уникальный идентификатор, или адрес, который записывается в блокчейн и обеспечивает подлинность и неподменяемость токена. Это означает, что каждый токен может быть свободно передан между участниками блокчейн сети без необходимости доверять друг другу [14].

Взаимозаменяемость в контексте токенов в блокчейне означает, что каждый токен является одинаковым и может быть заменен на любой другой токен того же типа и стоимости без каких-либо различий или потерь ценности. Однако, не все токены в блокчейне являются взаимозаменяемыми.

NFT-токен (невзаимозаменяемый токен) - это цифровой актив, который не может быть заменен или подделан. Это означает, что каждый токен имеет уникальную ценность, которая может быть использована для представления прав на реальные активы, такие как золото, недвижимость или даже частичную собственность в компании, хотя на данный момент наиболее популярны NFT-токены как представление собственности на аудио или видео материалы [15].

1.2 Анализ существующих блокчейнов

На данный момент существует множество различных блокчейн сетей. Нам необходимы тьюринг-полные блокчейны. Полнота по Тьюрингу, применительно к блокчейну, означает, что смарт-контракты этого блокчейна имеют возможность решить любую вычислимую задачу.

Под данную категорию подходят все Ethereum совместимые блокчейны (от англ. EVM compatible), поскольку Ethereum — первый полный по тьюрингу

блокчейн, второй по капитализации среди всех блокчейнов на сегодняшний день [4].

Рассмотрим некоторые из наиболее популярных блокчейнов и их особенности:

1. **Ethereum** — управляемый сообществом блокчейн, запущенный в 2015 году. Основывается на инновациях биткойна, но при этом имеет ряд существенных отличий. В отличие от Bitcoin, является полным по тьюрингу блокчейном, работает согласно алгоритму консенсуса **proof of stake**, что считается более дружелюбным решением для экологии из-за низкого потребления энергии. На сегодняшний день, транзакции являются одними из самых дорогих среди всех блокчейнов, что делает его неподходящим для наших целей [5].
2. **Binance Smart Chain** — EVM совместимый блокчейн экосистемы BNB, работает согласно комбинации алгоритмов консенсуса **proof of authority** и **delegated proof of stake** [6]. Стоимость транзакций значительно ниже относительно блокчейна Ethereum, однако все еще недостаточно низкая.
3. **Polygon** — EVM совместимый сайдчейн блокчейна Ethereum, работающий согласно алгоритму консенсуса **proof of stake**. Сайдчейн (от англ. sidechain) — блокчейн, работающий на основе другого блокчейна или параллельно другому блокчейну [7]. Транзакции на блокчейне Polygon значительно ниже, чем на рассмотренных выше блокчейнах, что делает его лучшим вариантом для развертывания нашего приложения.

После рассмотрения ключевых различий вышеупомянутых блокчейнов несложно заметить, что блокчейн Polygon — сайдчейн блокчейна Ethereum — является самым подходящим блокчейном для развертывания децентрализованного приложения в тестовой сети. Обусловлено это низкими транзакциями относительно других блокчейнов и простотой во взаимодействии во время разработки, получении тестовой валюты в тестовой сети и хорошей, полной документацией.

1.3 Анализ существующих аналогов маркетплейсов

В последние годы NFT-токены получили широкое распространение в мире блокчейна и стали очень популярными в качестве цифровых активов. NFT-токены представляют уникальные цифровые активы, которые не могут быть заменены на другие токены и могут иметь высокую стоимость, особенно если они представляют искусство или коллекционные предметы.

Маркетплейсы NFT-токенов предоставляют платформы для покупки, продажи и обмена NFT-токенами между различными участниками. Эти маркетплейсы обычно имеют большой выбор NFT-токенов, которые охватывают множество тематик, от искусства до музыки, видеоигр и других цифровых предметов. Кроме того, маркетплейсы NFT-токенов обеспечивают прозрачность и безопасность для продавцов и покупателей. На маркетплейсах покупатели могут удостовериться в подлинности и уникальности NFT-токенов, а продавцы могут продавать свои цифровые активы безопасно и с минимальными рисками.

В целом, маркетплейсы NFT-токенов играют важную роль в развитии экономики NFT и создании новых возможностей для искусства, музыки, игр и других активов, в том числе реальных. Маркетплейсы будут продолжать развиваться и становиться все более важными для тех, кто хочет купить, продать или обменять NFT-токены.

На данный момент на рынке существует множество NFT-маркетплейсов, позволяющих осуществлять торговлю невзаимозаменяемыми токенами между участниками различных блокчейн сетей, однако все эти маркетплейсы являются централизованными по своей архитектуре. Большая часть функционала маркетплейса реализована на серверной части, что представляет собой большое пространство для манипуляций. Рассмотрим некоторые из наиболее популярных маркетплейсов и их особенности:

1. **Opensea** — первый в мире крупнейший архитектурно централизованный маркетплейс для NFT-токенов, существующий в блокчейн сетях Ethereum, Polygon, Solana и т.д [8].
2. **Rarible** — популярный NFT-маркетплейс, помимо основной деятельности предоставляет инструменты разработчикам для работы с NFT-токенами. Как и Opensea, является архитектурно централизованным [9].

Помимо того, рассмотренные выше маркетплейсы имеют ряд серьезных недостатков. Основные из них:

1. Высокие комиссии за транзакции – рассмотренные маркетплейсы взимают комиссию за совершение обмена между участниками, которая может достигать 10%.
2. Англоязычная целевая аудитория – данные маркетплейсы в основе своей ориентированы на англоязычную аудиторию, что осложняет взаимодействие с приложением для русскоговорящей аудитории.

На основе изложенного выше несложно заметить, что все рассмотренные маркетплейсы являются централизованными на архитектурном уровне, что позволяет манипулировать процессом торговли. Более того, перечисленные аналоги ориентированы на англоязычную аудиторию.

1.4 Вывод по теоретической части

В результате выполненного в ходе работы анализа предметной области можно заключить:

1. Наиболее подходящим для развертывания приложения блокчейном является блокчейн Polygon благодаря маленьким комиссиям и низкому порогу входа для разработчиков децентрализованных приложений на базе смарт-контрактов.
2. Имеющиеся на рынке маркетплейсы архитектурно централизованы.
3. Большинство аналогов ориентированы на англоязычную аудиторию, что осложняет процесс торговли.
4. Разработчики аналогов в большинстве своем имеют возможность манипулировать торговлей.

Более того, были рассмотрены основные понятия, необходимые для понимания предметной области, а именно: что такое блокчейн, наиболее популярные консенсусы и форк блокчейна, смарт-контракты в сети блокчейн, токены и понятием взаимозаменяемости применительно к ним а так же NFT-токены.

2. Проектирование системы

2.1 Требования к системе

Перед началом любого проекта необходимо определить функциональность и требования к системе. Это позволит более эффективно разрабатывать приложение и избежать непредвиденных задержек или проблем в будущем. В контексте заданной темы мы можем определить следующие функциональные требования к приложению:

1. Пользователь может создать новый невзаимозаменяемый токен двух наиболее распространенных стандартов ERC721 и ERC1155 со своими данными.
2. Система позволяет продать NFT-токен в трёх доступных режимах: фиксированная цена, аукцион и лучшее предложение.
3. В режиме аукциона пользователь может настроить параметры продажи: минимальный допустимый шаг цены ставок, начальную ставку и длительность аукциона. В случае отсутствия ставок аукцион считается неудавшимся и токен снимается с продажи.
4. В режиме лучшего предложения продавец не указывает цену, а ожидает предложения от заинтересованных пользователей. Данный режим продажи не имеет срока годности.
5. В режиме фиксированной цены пользователь указывает лишь цену продажи токена. Данный режим продажи также не имеет срока годности.
6. В случае необходимости пользователь может отменить продажу невзаимозаменяемого токена.
7. Пользователь может купить токен, выставленный на продажу.

2.2 Архитектура системы

При разработке маркетплейса невзаимозаменяемых токенов используется UUPS (Universal Upgradable Proxy Standart) [22] архитектура. Она позволяет обновлять логику приложения после развертывания в блокчейн сети и убирать возможность обновления в случае острой необходимости. Использование этого паттерна обосновано необходимостью возможности добавить новый функционал или исправить старый. Суть работы паттерна – система делится на прокси контракты и контракты с функционалом, прокси контракт хранит адрес контракта, содержащего логику системы. Пользователь отправляет запрос на прокси, который затем перенаправляет запрос на конечный контракт. Для обновления логики нужно развернуть новый контракт с логикой и обновить его адрес в прокси контракте. Поскольку функционал обновления находится в самом контракте с логикой, после очередного обновления его можно убрать, тогда система утратит возможность обновления логики.

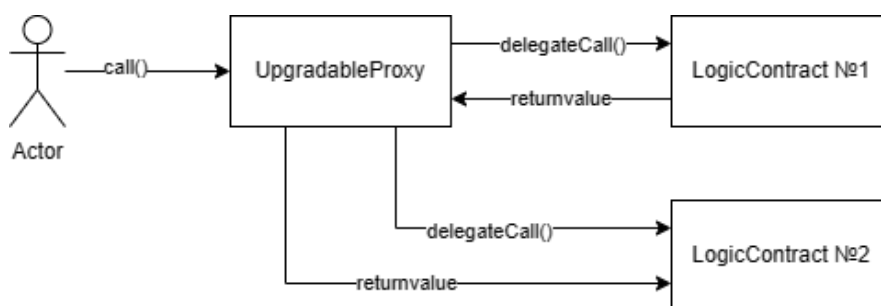


Рис.1 UUPS паттерн

2.3 Диаграмма вариантов использования

Диаграмма вариантов использования описывает возможные сценарии использования разрабатываемой системы для разных групп пользователей.

Данная диаграмма не описывает внутреннее устройство системы, а лишь упрощает взаимодействие пользователей с ней, демонстрируя наличие сценариев использования для каждой группы пользователей.

При этом пользователь или актер — это сущность, взаимодействующий с системой извне. Актером может быть как человек, так и некое устройство или другая программная система. Сценарий использования описывает имеющийся функционал системы.

Диаграмма вариантов использования представлена на рисунке 2.

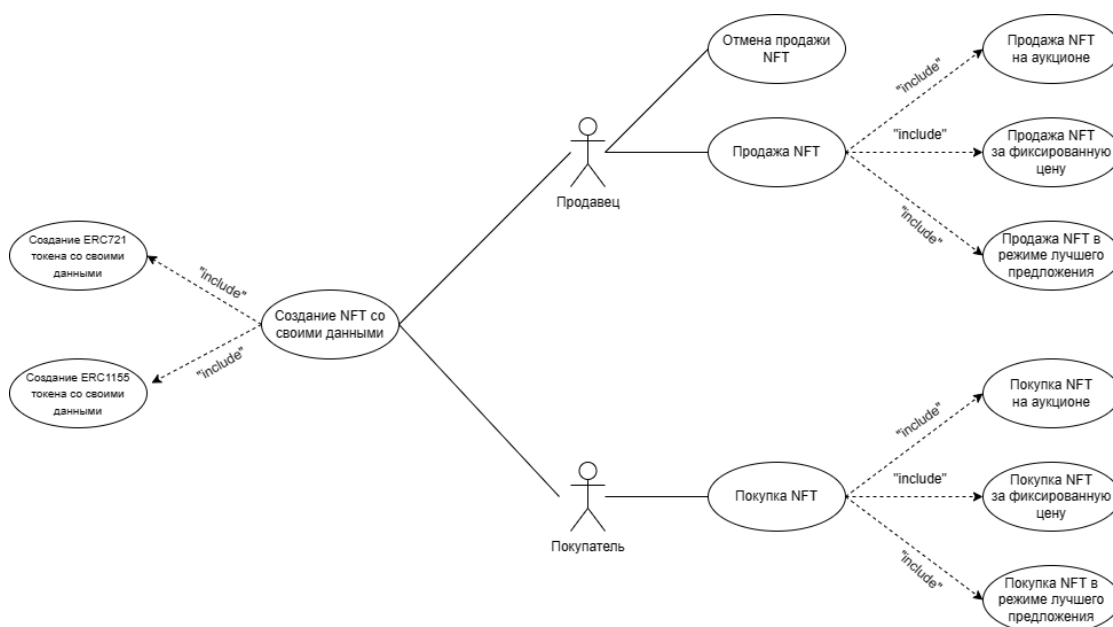


Рис. 2 Диаграмма вариантов использования

2.3 Диаграмма классов

Диаграмма классов используется для визуализации структуры и отношений классов в объектно-ориентированном программировании. Состоит она из классов, интерфейсов и связей между ними.

Классы представляют объекты, которые имеют общую структуру и поведение. Они содержат поля для хранения данных и методы для выполнения операций. Интерфейсы определяют сигнатуры методов без их реализации, поля описывают характеристики объектов этого класса, а методы – операции, которые могут быть выполнены объектами класса.

Диаграмма классов помогает спроектировать архитектуру системы, идентифицировать основные классы и их отношения. Помимо этого, она является удобным инструментом для коммуникации между членами команды разработчиков или с заинтересованными сторонами, предоставляя общее представление о классах и их связях.

Диаграммы классов представлены на рисунке 3, 4, 5.

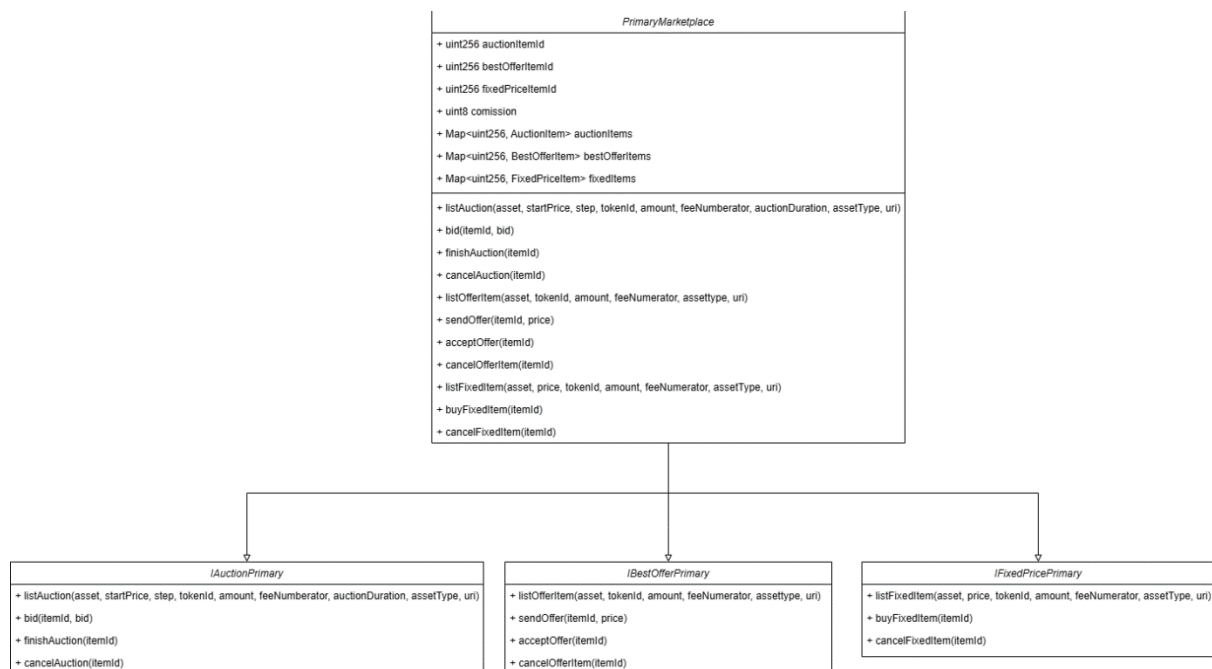


Рис.3 Диаграмма классов первичного маркетплейса

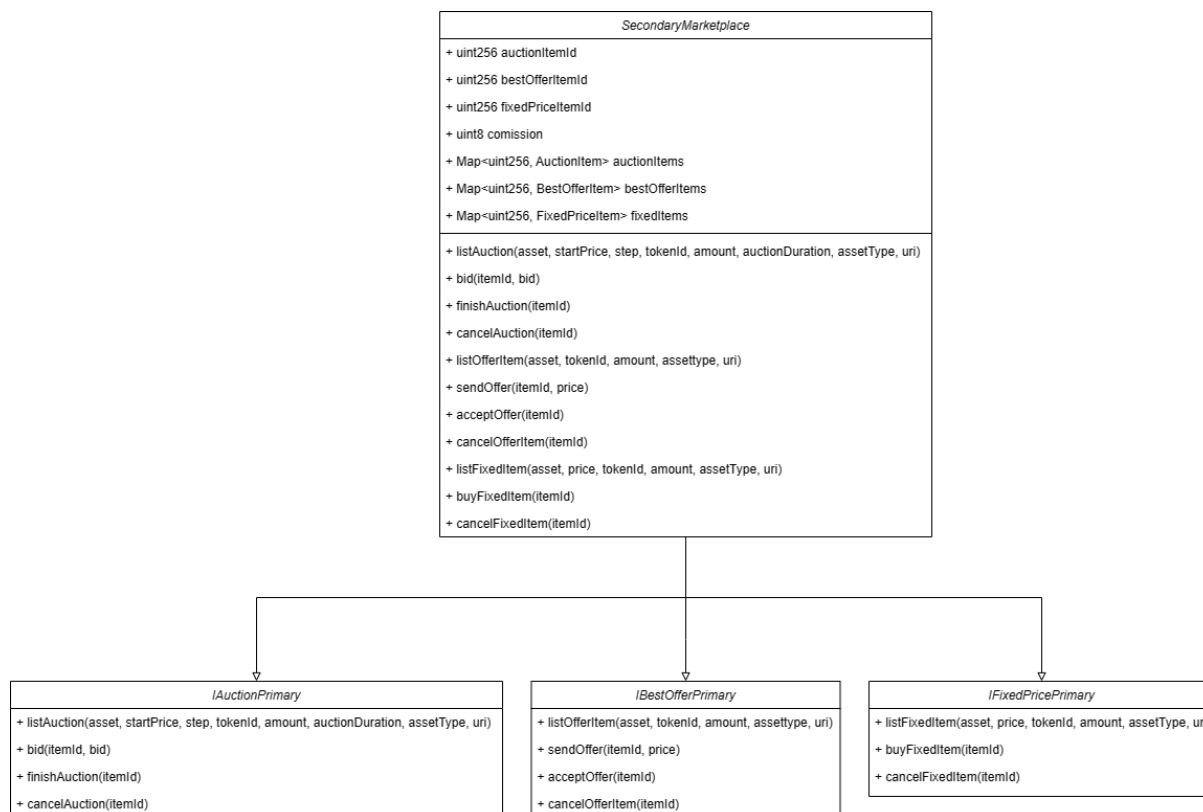


Рис. 4 Диаграмма классов вторичного маркетплейса

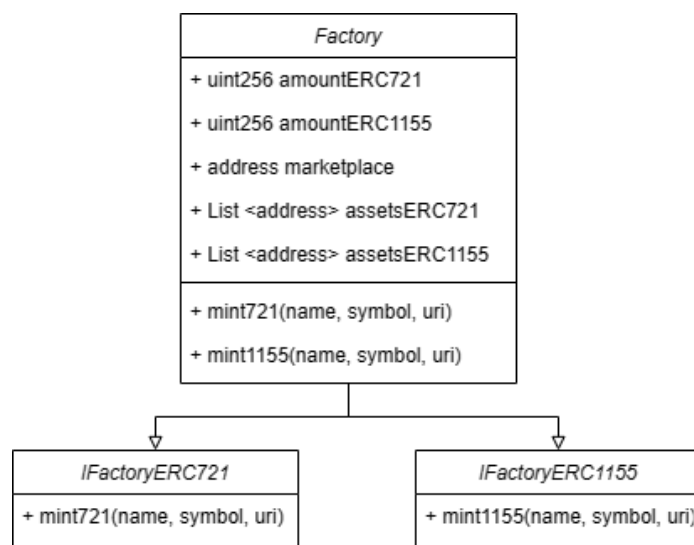


Рис. 5 Диаграмма классов фабрики

Сама система будет состоять из первичного и вторичного маркетплейса и фабрики невзаимозаменяемых токенов. На первичном маркетплейсе размещаются ордера на продажу только созданных на фабрике токенов. На вторичном рынке размещаются ордера на продажу

всех остальных токенов. Фабрика разворачивает пользовательские токены с заданными данными.

2.4. Функциональные модели IDEF0 и IDEF3

Функциональные модели IDEF0 и IDEF3 — модели структурного анализа и проектирования.

Модель IDEF0 используется для анализа и моделирования функций или деятельности системы. Она позволяет представить структуру функций, их связи и зависимости с помощью блок-схем, имеющих иерархическую структуру. Каждая функция представляется блоком, а связи между функциями обозначаются стрелками. Модель IDEF0 используется для оптимизации процессов, улучшения производительности и выявления проблем в системе.

Модель IDEF3 используется для описания и моделирования процессов и деятельности в системе. Она фокусируется на взаимодействии между объектами и информацией в процессе. Модель IDEF3 помогает понять, как объекты и информация перемещаются и обрабатываются в процессе выполнения действий.

Обе функциональные модели IDEF0 и IDEF3 позволяют разрабатывать более четкое представление о бизнес-процессах, функциях и их взаимодействии.

Диаграммы IDEF0 до внедрения системы представлены на рисунках 6, 7 и 8.

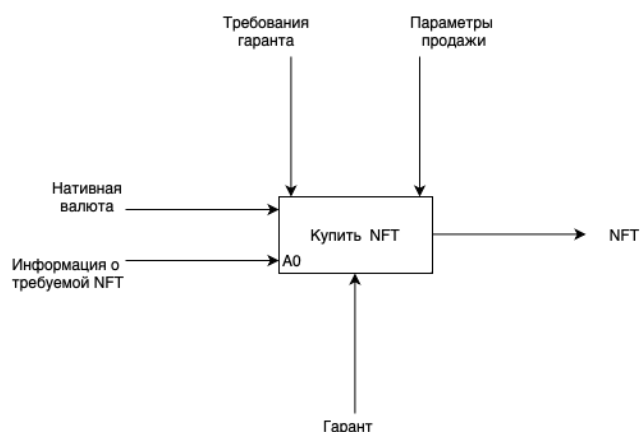


Рис. 6 Контекстная диаграмма IDEF0 до внедрения системы

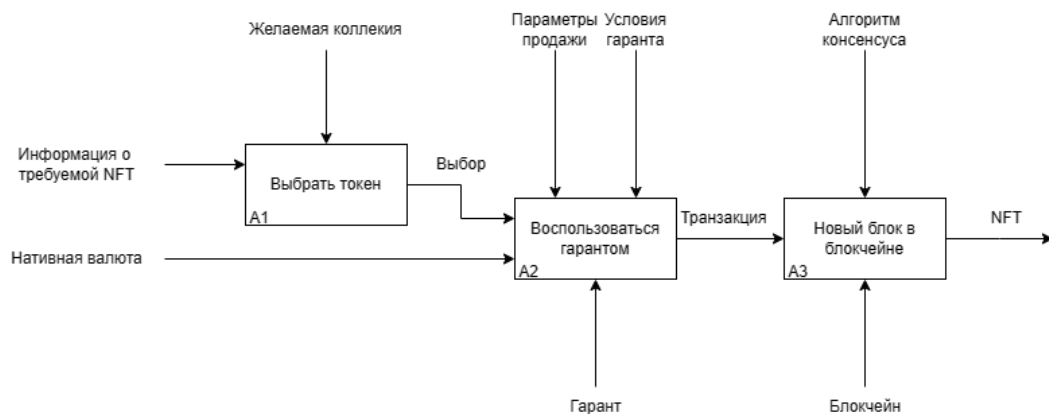


Рис 7. Декомпозиция контекстной диаграммы IDEF0 до внедрения системы

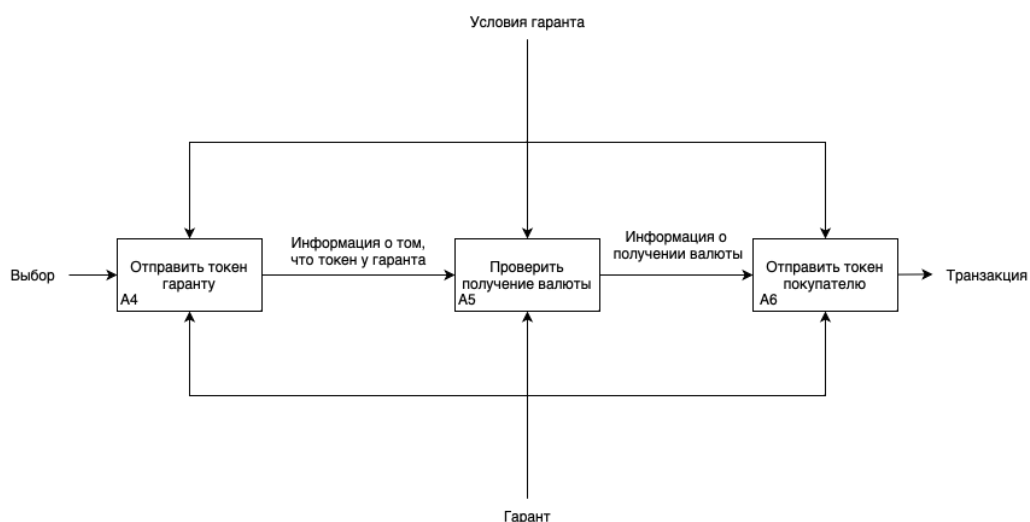


Рис 8. Декомпозиция блока A2

Диаграммы IDEF0 после внедрения системы представлены на рисунках 9, 10 и 11.

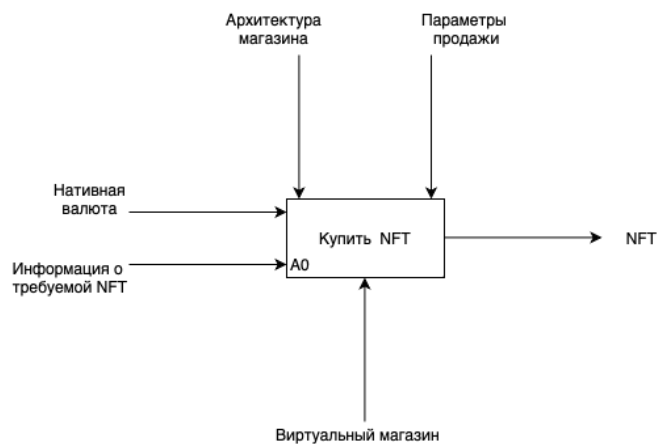


Рис. 9 Контекстная диаграмма IDEF0 после внедрения системы

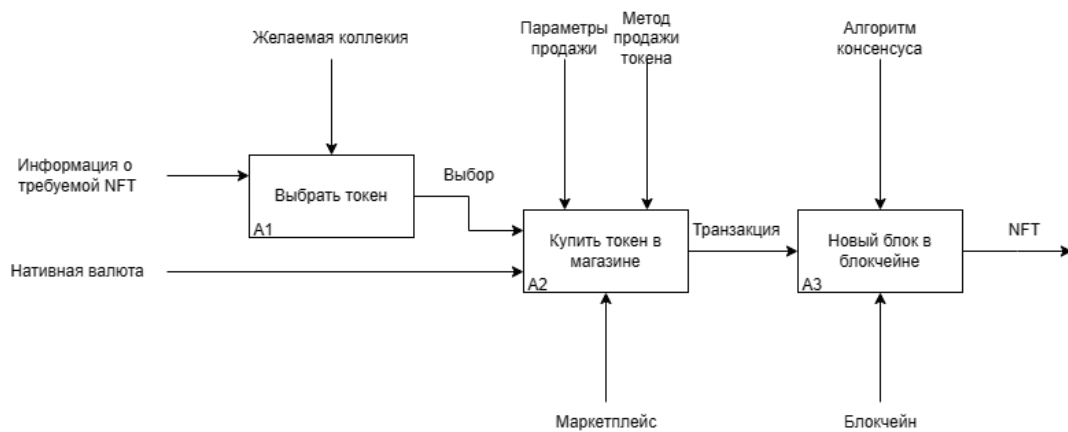


Рис. 10 Декомпозиция контекстной диаграммы IDEF0 после внедрения системы

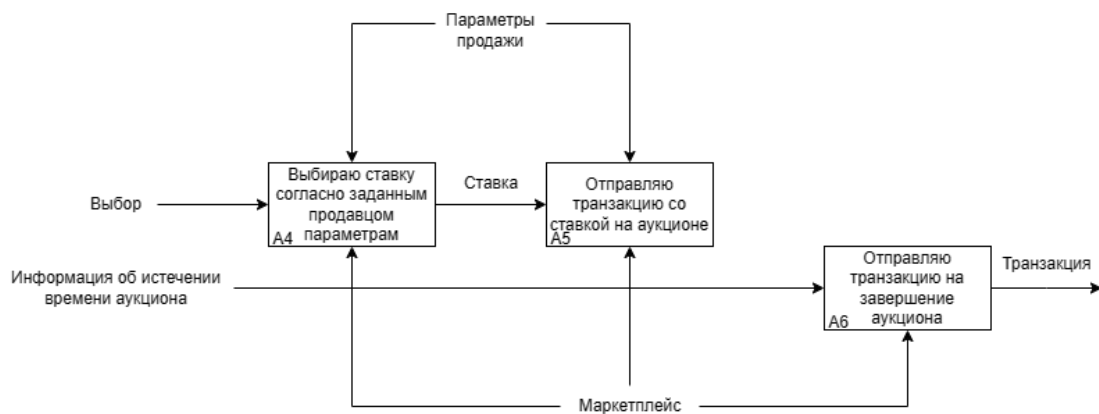


Рис. 11 Декомпозиция блока A2

Диаграммы IDEF3 до внедрения системы представлены на рисунках 12, 13 и 14.

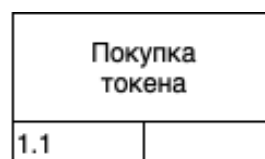


Рис. 12 Контекстная диаграмма IDEF3 до внедрения системы



Рис. 13 Первая декомпозиция IDEF3 до внедрения системы



Рис. 14 Вторая декомпозиция IDEF3 до внедрения системы

Диаграммы IDEF3 после внедрения системы представлены на рисунках 15, 16 и 17.



Рис. 15 Контекстная диаграмма IDEF3 после внедрения системы

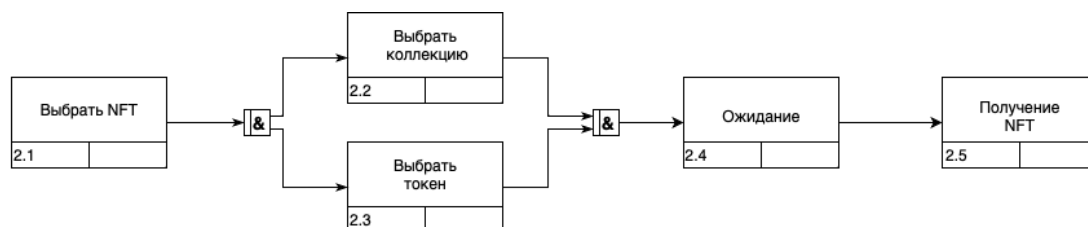


Рис. 16 Первая декомпозиция IDEF3 после внедрения системы



Рис. 17 Вторая декомпозиция IDEF3 после внедрения системы

2.5 Структура проекта

В таблице 1 представлена структура файлов проекта. Подобная структура аналогична и для других проектов, реализованных на Hardhat [16], поэтому сторонним разработчикам будет проще понять чужой код.

Таблица 1. Структура проекта

Файл / директория	Назначение
contracts/	Основной код приложения, включающий смарт-контракты первичного и вторичного маркетплейса и фабрики
docgen/	Документация, сгенерированная из NatSpec [17] комментариев в коде приложения
scripts/	Скрипты для развертывания контрактов и их обновления
tasks/	Скрипты для взаимодействия с развернутым приложением
tests/	Юнит-тесты для приложения, написанные на Typescript

3. Программная реализация

3.1 Обоснование выбора средств разработки

Разработка программной системы начинается с выбора средств разработки. Данный этап является критически важным, поскольку от этого зависит скорость и качество разрабатываемого продукта. Для написания смарт-контрактов и их последующего тестирования необходимы средства разработки, отвечающие требованиям качества и надежности. Для написания смарт-контрактов есть два наиболее популярных в сообществе языка программирования: Solidity и Vyper. В таблице 2 представлены сильные стороны этих языков.

Таблица 2. Сильные стороны языков программирования смарт-контрактов Solidity и Vyper.

Solidity	Vyper
Широко распространен и поддерживается большим сообществом.	Простой синтаксис и, как следствие, более доступный для новых разработчиков.
Богатый набор инструментов и библиотек для разработки и тестирования.	Высокий уровень безопасности по умолчанию.
Поддержка множества EVM-совместимых блокчейнов.	Простая формальная верификация и легкая интеграция с другими инструментами формальной верификации
Возможность использования существующего кода и разработанных библиотек, что ускоряет разработку и повышает удобство использования.	

Язык программирования Vyper является более доступным для новых разработчиков и более безопасным по умолчанию. Solidity в свою очередь имеет обширный инструментарий и библиотеку оптимизированных смарт-контрактов OpenZeppelin, что ускоряет разработку. Помимо этого, Solidity поддерживается множеством EVM-совместимых блокчейн-сетей, что позволит без лишних доработок развернуть приложение в разных блокчейн-сетях, чего не скажешь о языке Vyper.

Для полноты картины, стоит рассмотреть и слабые стороны этих языков. Они представлены в таблице 3.

Таблица 3. Слабые стороны языков программирования смарт-контрактов Solidity и Vyper.

Solidity	Vyper
Низкий уровень безопасности по умолчанию.	Ограниченный набор инструментов и библиотек.
Достаточно не простой синтаксис и неочевидные особенности языка.	Ограниченная поддержка платформ
Требует высокий уровень экспертизы для разработки надежных и безопасных смарт-контрактов.	Ограниченные возможности оптимизации и снижения количества газа за транзакции.

После рассмотрения слабых сторон этих языков программирования, выбор в пользу Solidity становится все более очевидным. Язык Vyper обладает ограниченным инструментарием и поддерживается небольшим количеством блокчейн-сетей, что усложняет перенос приложения в другие децентрализованные сети. Помимо этого, язык Vyper имеет ограниченные возможности оптимизации, что делает взаимодействие со смарт-контрактами, написанными на нем, не выгодным.

Как итог, язык Solidity является наиболее подходящим инструментом для проекта, поскольку он имеет более обширный инструментарий, библиотеку оптимизированных смарт-контрактов и обладает большей гибкостью касательно разработки.

Тестирование – неотъемлемый этап разработки качественного и надежного программного продукта, позволяющий обнаружить и исправить ошибки, уязвимости и недочеты в приложении. Для достижения наилучшего результата необходимо правильно выбрать инструменты тестирования. Наиболее популярные фреймворки для тестирования смарт-контрактов: Hardhat и Brownie. Начнем сравнение этих инструментов с рассмотрения их сильных сторон. В таблице 4 представлены сильные стороны фреймворков Hardhat и Brownie.

Таблица 4. Сильные стороны фреймворков Hardhat и Brownie.

Hardhat	Brownie
Богатый набор инструментов для отладки и тестирования	Простота и интуитивность.
Гибкая конфигурация	Интегрированное тестовое окружение
Активное сообщество разработчиков	Мощные инструменты разработки

Интеграции с другими инструментами	
------------------------------------	--

В целом, оба фреймворка предлагают мощные инструменты и функции для тестирования и отладки, такие как виртуальный блокчейн, плагины, пустые контракты для тестов и так далее. Отличие лишь в том, что Hardhat имеет более широкий спектр инструментов отладки и тестирования, при этом Brownie более легковесный и интуитивный.

Для полноты картины стоит рассмотреть и слабые стороны этих фреймворков. В таблице 5 представлены слабые стороны фреймворков Hardhat и Brownie.

Таблица 5. Слабые стороны фреймворков Hardhat и Brownie.

Hardhat	Brownie
Сложность конфигурации	Ограниченная поддержка платформ
Некоторая избыточность	Ограниченный инструментарий

Hardhat достаточно сложный в настройке фреймворк с настолько большим инструментарием, что для некоторых проектов это излишне. Brownie же наоборот более легковесный, но ограничен в поддержке платформ и наличии инструментария, что может замедлить разработку.

В нашем случае, выбор сделан в пользу Hardhat, поскольку он имеет больший инструментарий для оптимизации и отладки контрактов, что упрощает процесс тестирования. Более того, сообщество Hardhat более активно, по этой причине там больше интересных плагинов и примеров кода.

Таким образом, основные средства разработки децентрализованного приложения для торговли невзаимозаменяемыми токенами это язык программирования смарт-контрактов Solidity и фреймворк для отладки и тестирования Hardhat.

Поскольку в ходе тестирования использовался фреймворк Hardhat, соответственно тесты писались на языке Typescript. В ходе отладки было необходимо использовать библиотеку для взаимодействия с блокчейном. Самые популярные библиотеки: Web3.js и Ethers.js. Начнем сравнение этих инструментов с рассмотрения сильных сторон. На таблице 6 представлены сильные стороны библиотек Web3.js и Ethers.js.

Таблица 6. Сильные стороны библиотек Web3.js и Ethers.js.

Web3.js	Ethers.js
Популярность	Простота использования

Обширный функционал	Чистая и современная архитектура библиотеки
Стабильность и надежность	Широкий набор функциональности
	Наличие методов для работы с популярнейшими стандартами токенов, в том числе невзаимозаменяемыми.

Обе библиотеки предлагают мощный набор инструментов для взаимодействия с блокчейном. Ethers.js является более простым для освоения и молодым, а Web3.js – более популярным. Для полноты картины следует рассмотреть и слабые стороны данных библиотек. В таблице 7 представлены слабые стороны библиотек Ethers.js и Web3.js.

Таблица 7. Слабые стороны библиотек Ethers.js и Web3.js.

Web3.js	Ethers.js
Сложность использования	Большой объем библиотеки
Низкая универсальность	Низкая универсальность

Web3.js довольно сложный в использовании, при этом имеет низкую совместимость с блокчейн-сетями. Ethers.js при этом так же имеет низкую совместимость, при этом еще и большой объем функционала в целом.

Как итог, для разработки децентрализованного маркетплейса для торговли невзаимозаменяемыми токенами используется язык программирования смарт-контрактов Solidity, потому что он имеет обширный функционал и библиотеку оптимизированных смарт-контрактов OpenZeppelin. Для тестирования используется язык программирования Typescript и фреймворк Hardhat, поскольку данный фреймворк отличается высокой функциональностью для оптимизации и отладки. Для взаимодействия с децентрализованной сетью используется библиотека Ethers.js, так как он является простым в освоении и имеет широкий функционал.

3.2 Описание разработанных частей проекта

Проект децентрализованного маркетплейса для торговли невзаимозаменяемыми токенами – система из нескольких отдельных частей, которые в совокупности формируют ее. Каждый смарт-контракт развернут согласно стандарту UUPS ERC1822.

Маркетплейс поддерживает три режима продажи: фиксированная цена, аукцион и лучшее предложение. Фиксированная цена – продажа токена за заданную при создании ордера цену. Аукцион – продавец

выставляет начальную цену и минимальный шаг ставки, после чего потенциальные покупатели делают ставки. Забирает токен человек с самой большой ставкой. Лучшее предложение – частный случай аукциона, при котором не устанавливается стартовая цена и минимальный шаг цены. Заинтересованные лица отправляют предложения и токен отправляется человеку с лучшим предложением. Интересная деталь – пользователи видят только свое предложение. После первой продажи токен попадает на вторичный маркетплейс.

Изначально маркетплейс рассматривался как один контракт, который будет использоваться для торговли уже существующими невзаимозаменяемыми токенами и теми, которые будут созданы на фабрике после покупки. Однако в ходе разработки выяснилось, что маркетплейс следует разделить на две компоненты. Этому поспособствовало несколько причин:

1. Высокая цена транзакций – соединение всего функционала в один контракт повышает его размер, что как следствие делает транзакции дороже. Более того, чем больше данных хранит контракт, тем больше цена транзакции по получению этих данных. Разделение маркетплейса на два контракта помогло снизить цену транзакций.
2. Качество кода – разделение на два контракта сделало код читабельней для других разработчиков, что облегчит интеграцию со сторонними сервисами.

Контракт фабрики также претерпел ряд изменений. Изначально планировалось написать два контракта фабрики – для ERC721 токенов и для ERC1155 токенов. Позднее выяснилось, что разделение ни к чему, поскольку функционал развертывания довольно легковесный, а разделение лишь приведет к тому, что придется развертывать два контракта вместо одного, соответственно тратить больше денег.

Далее, для оптимизации развертывания токенов фабрикой было принято решение использовать стандарт Minimal Proxy ERC1167 [21]. Исходный вариант с полным развертыванием оказался слишком дорогостоящим, что смутит потенциальных пользователей.

Далее подробно рассмотрим каждую из компонент программной системы и ее функционал.

3.2.1 NFT-токен стандарта ERC721

Смарт-контракт стандарта ERC721 [19] представляет собой коллекцию невзаимозаменяемых токенов. На контракте хранятся балансы всех держателей токенов и информация, благодаря которой каждый токен уникален. Токены можно переводить по одному с одного адреса на другой и давать разрешение определенному адресу переводить с собственного баланса токены на любой другой адрес. Все реализация оптимизирована благодаря реализации библиотеки оптимизированных смарт-контрактов OpenZeppelin.

Помимо основного функционала, разрабатываемый контракт наследует реализацию стандарта NFT Royalty ERC2981 [18]. Данный стандарт позволяет указанному адресу получать заданный при развертывании процент с будущих продаж невзаимозаменяемого токена. Его оптимизированная реализация также присутствует в библиотеке OpenZeppelin.

3.2.2 NFT-токен стандарта ERC1155

Смарт-контракт стандарта ERC1155 [20] представляет собой коллекцию взаимозаменяемых и невзаимозаменяемых токенов. На контракте хранятся балансы всех коллекций токенов и информация, которая может представлять уникальность в случае, если такой токен в единственном экземпляре. Токены можно передавать как по одному, так и группами (батчами от англ. Batch) и давать разрешение определенному адресу распоряжаться всеми токенами или только токенами конкретной коллекции.

Более того, в данном стандарте есть защита от переводов токенов на адреса контрактов, которые не предназначены для отправки на них токенов стандарта ERC1155. При переводе токена проверяется адрес, на который его отправляют. Если это адрес контракта, и он не наследует интерфейс **onERC1155BatchReceived** или **onERC1155Received**, отправка отклоняется. Вся реализация наследуется смарт-контрактом из библиотеки оптимизированных смарт-контрактов OpenZeppelin. Помимо основного функционала, разрабатываемый контракт наследует реализацию стандарта NFT Royalty ERC2981 [18]. Данный стандарт позволяет указанному адресу получать заданный при развертывании процент с будущих продаж невзаимозаменяемого токена. Его оптимизированная реализация также присутствует в библиотеке OpenZeppelin.

3.2.3 Фабрика NFT-токенов стандарта ERC721 и ERC1155

Смарт-контракт фабрики невзаимозаменяемых токенов стандартов ERC721 и ERC1155 представляет собой контракт, который может развертывать контракты токенов с пользовательскими данными. Поскольку развертывание полного контракта является дорогостоящим процессом, используется стандарт Minimal Proxy ERC1167 [21]. Суть его в том, что полный контракт мы развертываем один раз, а далее развертываем минимальные жизнеспособные прокси, которые перенаправляют все запросы на эталонный контракт. Благодаря стандарту Minimal Proxy развертывание дешевле в несколько раз. Реализация данного стандарта наследуется смарт-контрактом фабрики из библиотеки оптимизированных смарт-контрактов OpenZeppelin. Фабрика также хранит количество развернутых прокси и их адреса.

3.2.4 Первичный маркетплейс NFT-токенов

Первичный маркетплейс невзаимозаменяемых токенов представляет собой смарт-контракт, используемый для торговли NFT-токенами, созданными с помощью фабрики. После развертывания, фабрика инициализирует только что созданную коллекцию и дает разрешение первичному маркетплейсу создавать токены этой коллекции (только по запросу пользователя). Далее, пользователь может сам создать токен или создать ордер на продажу на первичном маркетплейсе.

В случае создания ордера, токен фактически еще не создан. Известна лишь информация, которую он будет содержать, поскольку продавец вводит информацию о токене при продаже. Сделано это с целью сокращения стоимости транзакции создания ордера на продажу. В случае, если продавец по какой-либо причине передумал продавать свой NFT-токен, он может отменить ордер. После создания ордера, продавец ожидает, пока найдется покупатель. Во время создания ордера проверяется, разрешил ли продавец управлять первичному маркетплейсу токеном, который он продает. Если разрешения нет, создание ордера отменяется. Если во время ожидания продавец убрал разрешение на управление токеном для маркетплейса, сделка не состоится.

При создании ордера, продавец может выбрать три доступных режима продажи: аукцион, фиксированная цена и лучшее предложение.

3.2.5 Вторичный маркетплейс NFT-токенов

Вторичный маркетплейс невзаимозаменяемых токенов представляет собой смарт-контракт, используемый для торговли уже созданными NFT-токенами. Основное отличие от первичного маркетплейса в том, что размещенные на вторичном маркетплейсе токены уже существуют.

При создании ордера продавец разрешает вторичному маркетплейсу распоряжаться своими токенами и вводит данные о токене, который хочет продать. В случае, если продавец по какой-либо причине передумал продавать свой NFT-токен, он может отменить ордер. После создания ордера, продавец ожидает, пока найдется покупатель. Если в период ожидания продажи продавец убрал разрешение маркетплейсу на распоряжение токенами, сделка не состоится.

При создании ордера, как и в первичном маркетплейсе, продавец может выбрать три доступных режима продажи: аукцион, фиксированная цена и лучшее предложение.

3.3 Тестирование смарт-контрактов с использованием фреймворка Hardhat.

Все разработанные смарт-контракты должны быть протестированы перед развертыванием. Для тестирования и отладки был выбран фреймворк Hardhat, поскольку он имеет большой инструментарий для оптимизации и отладки контрактов, что упрощает процесс тестирования.

В ходе конфигурации были использованы различные плагины. Список плагинов приведен в таблице 8.

Таблица 8. Список плагинов Hardhat, используемых в ходе тестирования.

Название	Описание
hardhat-gas-reporter	Позволяет отслеживать количество газа, необходимое для вызова функций и развертывания смарт-контрактов.
solidity-coverage	Позволяет показать конкретные строки кода, которые были протестированы, или «покрыты» тестами.
hardhat-docgen	Генерирует документацию по NatSpec комментариям.

hardhat-contract-sizer	Отображает размер смарт-контракта
hardhat-abi-exporter	Генерирует ABI [23] для взаимодействия со смарт-контрактами извне блокчейна.

Плагин hardhat-gas-reporter отображает количество газа, необходимое для развертывания контрактов и вызова функций. Таким образом, можно отслеживать изменения в количестве газа и корректировать их при разработке. На рисунке 18 показана цена развертывания контрактов.

Deployments				% of limit	
AssetERC1155	-	-	3778980	12.6 %	-
AssetERC721	-	-	3871702	12.9 %	-
Factory	-	-	1752427	5.8 %	-
PrimaryMarketplace	-	-	3113195	10.4 %	-
SecondaryMarketplace	-	-	3381104	11.3 %	-

Рис. 18 Отчет плагина hardhat-gas-reporter

Плагин solidity-coverage отображает покрытие тестами. Данный плагин помогает проверить качество тестов и дополнить их в случае необходимости. Он генерирует отчет в формате HTML. Пример отчета показан на рисунке 19.

all files market/									
100% Statements 148/148 88.89% Branches 16/18 100% Functions 64/64 100% Lines 155/155									
File	Statements	Branches	Functions	Lines					
IAssetCommon.sol	100%	0/0	100%	0/0	100%	0/0	100%	0/0	
IOwner.sol	100%	0/0	100%	0/0	100%	0/0	100%	0/0	
PrimaryMarketplace.sol	100%	72/72	90%	9/10	100%	30/30	100%	76/76	
SecondaryMarketplace.sol	100%	76/76	87.5%	7/8	100%	34/34	100%	79/79	

Рис. 19 Отчет плагина solidity-coverage

Плагин hardhat-docgen генерирует документацию из комментариев типа NatSpec в формате HTML. Данная документация будет полезна сторонним разработчикам, настраивающим интеграцию с маркетплейсом. Пример документации показан на рисунке 20.



Рис. 20 Отчет плагина hardhat-docgen

После конфигурации фреймворка Hardhat я приступил к тестированию смарт-контрактов маркетплейса для торговли невзаимозаменяемыми токенами. В ходе тестирования использовалась библиотека Chai.js [24].

Смарт-контракты токенов были протестированы на добавленный функционал, поскольку основа была взята из библиотеки оптимизированных контрактов OpenZeppelin. В виртуальном блокчейне проверялась верность создания токенов коллекции.

Затем были протестирован смарт-контракт фабрики невзаимозаменяемых токенов, а конкретно следующее:

1. Развертывания минимальных жизнеспособных прокси-контрактов токенов.
2. Верность заполнения переданных данных.
3. Цена за транзакцию сопоставима с допустимой при развертывании токенов.

Именно на этом этапе было принято решение использовать стандарт Minimal Proху для оптимизации развертывания токенов.

Далее был протестирован основной функционал смарт-контрактов первичного и вторичного маркетплейса, а именно: функционал аукциона, настройка параметров аукциона, продажа за фиксированную цену и продажа в режиме лучшего предложения.

Все ошибки, найденные в ходе тестирования, были исправлены. Результат тестирования представлен на рисунке 21.

```
✓ Buying ERC721 item... (228ms)
✓ Buying ERC1155 item... (79ms)
Common funcs
✓ Setting comission (45ms)
✓ Setting validator (45ms)
✓ Checking if the marketplace is ERC721 receiver
✓ Checking if the marketplace is ERC1155 receiver
✓ Check if the marketplace implements ERC165
✓ Withdraw weth from the marketplace (108ms)
✓ Withdraw eth from the marketplace (86ms)
✓ Backdoor for erc20 (220ms)
✓ Backdoor for erc721 (158ms)
✓ Backdoor for erc1155 (125ms)

35 passing (9s)
```

Рис.21 Результат тестирования смарт-контрактов

4. Финансовый менеджмент, ресурсоэффективность и ресурсосбережение

4.1 Введение

Цель данной выпускной квалификационной работы заключается в разработке децентрализованного приложения, позволяющего осуществлять торговлю невзаимозаменяемыми токенами между участниками сети блокчейна.

Целью раздела «Финансовый менеджмент, ресурсоэффективность и ресурсосбережение» является оценка его эффективности. Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Оценить коммерческий потенциал и перспективность разработки НИ;
2. Осуществить планирование этапов выполнения исследования;
3. Рассчитать бюджет затрат на исследования;
4. Произвести сравнительную оценку эффективности научно-технического исследования.

4.2 Оценка коммерческого потенциала и перспективности проведения научных исследований с позиции ресурсоэффективности и ресурсосбережения

Потенциальные потребители результатов исследования. Разработка, которой посвящена данная работа, представляет собой децентрализованное приложение для торговли невзаимозаменяемыми токенами.

Исходя из особенностей приложения, можно судить о круге лиц, которые потенциально будут заинтересованы в разработке. Целевым рынком текущей разработки являются физические лица и юридические лица, так или иначе интересующиеся блокчейн технологией. В данной области есть проблема с торговлей токенами - существующие маркетплейсы централизованы, что представляет собой некоторые риски для пользователей,

такие как цензура, территориальные ограничения и так далее. Все компании и физические лица будут заинтересованы в текущем проекте, так как он позволяет без посредников и рисков торговать невзаимозаменяемыми токенами.

Сегментировать рынок услуг по данной тематике можно по степени использования подобных разработок и виду заинтересованных лиц.

Таблица 9 - Карта сегментирования рынка услуг по торговле токенами

		Вид заинтересованного лица							
		Блокчейн сети			Физические лица, покупающие токены часто			Физически е лица, покупающ ие токены не часто	
Степени использования	Высокая								
	Средняя								
	Низкая								



- Данная разработка

- OpenSea

- Rarible

На карте сегментирования рынка услуг по торговле невзаимозаменяемыми токенами указано текущее положение дел для продуктов-конкурентов OpenSea и Rarible. Ячейки, выделенные фиолетовым цветом, это области, в которые данный продукт будет интегрироваться.

Анализ конкурентных технических решений. Данное решение по торговле невзаимозаменяемыми токенами не является уникальным в своём роде, но отличается от конкурентов некоторыми компонентами. В качестве конкурентов было выбрано несколько похожих решений, которые были оценены путём экспертной оценки по следующим критериям:

- Функциональность
- Децентрализация
- Удобство использования
- Обновляемость
- Комиссия за продажу
- Комиссия блокчейн сети

В конечном итоге для анализа были выбраны следующие конкурентные технические решения:

- «OpenSea» K1
- «Rarible» K2

Самым оптимальным способом проведения такого рода анализа считается анализ с помощью оценочной карты.

Анализ конкурентных технических решений определяется по формуле:

$$K = \sum B_i * B_i, (1)$$

где K - конкурентоспособность научной разработки или конкурента,

B_i - вес показателя(в долях единицы),

B_i - балл i-го показателя.

Оценочная карта для сравнения конкурентных технических решений представлена в следующей таблице 10

Таблица 10 - Оценочная карта для сравнения конкурентных технических решений

Критерии оценки	Вес критерия	Баллы			Конкурентноспособность		
		$B_{ни}$	$B_{к1}$	$B_{к2}$	$K_{ни}$	$K_{к1}$	$K_{к2}$
1	2	3	4	5	6	7	8
Технические критерии оценки ресурсоэффективности							
1. Функцион альность	0,1	4	4	4	0,4	0,4	0,4

2. Децентрализация	0,4	4	2	2	1,6	0,8	0,8
3. Удобство использования	0,3	4	4	4	1,2	1,2	1,2
4. Обновляемость	0,1	3	4	4	0,3	0,4	0,4
5. Комиссия за продажу	0,05	4	3	3	0,2	0,15	0,15
1. Комиссия блокчейн сети	0,05	4	3	3	0,2	0,15	0,15
Итого	1	23	20	20	3,9	3,1	3,1

По данным оценочной карты можно увидеть, что разрабатываемое техническое решение является достаточно конкурентоспособным, его сильные стороны – это удобство использования, децентрализация и функциональность, а также низкая стоимость комиссии, но есть и слабые стороны – это обновляемость смарт-контрактов. Так как текущее решение только начинает свой путь, то сначала будет мало ресурсов. Поэтому необходимо принимать меры по продвижению проекта разными способами.

SWOT-анализ. На первом этапе SWOT анализа в таблице ниже были описаны сильные и слабые стороны проекта, выявлены возможности и угрозы реализации НИ.

Таблица 11. Матрица SWOT анализа

Сильные стороны	Возможности
С1. Удобство С2. Функциональность С3. Децентрализация	В1. Нетрудоемкая адаптация В2. Повышение качества приложения В3. Сотрудничество с блокчейн сетями
Слабые стороны	Угрозы
Сл1. Приложение доступно только на EVM-совместимых блокчейнах Сл2. Отсутствие технической поддержки после реализации	У1. Отсутствие спроса У2. Отсутствие обратной связи от пользователей У3. Нехватка финансирования

Второй этап состоит в выявлении соответствия сильных и слабых сторон научно-исследовательского проекта внешним условиям. В рамках данного этапа была построена интерактивная матрица проекта. Интерактивная матрица проекта представлена в табл. 12 и 13.

Таблица 12 - Интерактивная матрица сильных и слабых сторон и возможностей

	Сильные стороны				Слабые стороны	
Возможности проекта		C1	C2	C3	Сл1	Сл2
	B1	-	+	+	+	-
	B2	+	+	+	-	-
	B3	-	+	+	0	+

Таблица 13 - Интерактивная матрица сильных и слабых сторон и угроз

	Сильные стороны				Слабые стороны	
Угрозы проекта		C1	C2	C3	Сл1	Сл2
	У1	+	+	+	-	-
	У2	+	+	+	+	+
	У3	+	+	+	-	-

Анализ интерактивных таблиц представляется в форме записи сильно коррелирующих сильных сторон и возможностей или слабых сторон и возможностей:

- B2C2C3
- B3Сл2
- У1У2У3C1C2C3
- У2Сл2

Из всего вышеперечисленного можно сделать несколько выводов:

Приложение имеет большой потенциал при дальнейшем расширении вне РФ. Сотрудничество с блокчейн сетями позволит расширять базу пользователей.

Повышение качества приложения невозможно без технической поддержки после реализации.

Отсутствие спроса и обратной связи влияет на удобство и функциональность приложения для пользователей, без этих факторов развивать приложение будет невозможно.

Большой угрозой для приложения является отсутствие спроса из-за отсутствия технической поддержки после реализации, в приложении могут быть критические ошибки, мешающие его нормальной работе.

В рамках третьего этапа составлена итоговая матрица SWOT-анализа, представленная в таблице 14.

Таблица 14 - Итоговая матрица SWOT анализа

	Сильные стороны проекта: С1. Удобство С2. Функциональность С3. Децентрализация	Слабые стороны проекта: Сл1. Приложение доступно только на EVM-совместимых блокчейнах Сл2. Отсутствие технической поддержки после реализации
Возможности: В1. Нетрудоемкая адаптация В2. Повышение качества приложения В3. Сотрудничество с блокчейн сетями	Приложение имеет большой потенциал при дальнейшем расширении вне РФ. Сотрудничество с блокчейн сетями позволит расширять пользовательскую базу	Повышение качества приложения невозможно без технической поддержки после реализации.
Угрозы: У1. Отсутствие спроса У2. Отсутствие обратной связи от пользователей У3. Нехватка финансирования	Отсутствие спроса и обратной связи влияет на удобство и функциональность приложения для пользователей, без этих факторов развивать	Большой угрозой для приложения является отсутствие спроса из-за отсутствия технической поддержки после реализации, в приложении могут быть критические ошибки,

	приложение будет невозможно	мешающие его нормальной работе.
--	-----------------------------	---------------------------------

4.3 Определение возможных альтернатив проведения научных исследований

Разработать приложение, которое позволит торговать невзаимозаменяемыми токенами.

Приложение должно быть надежным и функциональным, обеспечивая возможность продажи в режиме аукциона, лучшего предложения и фиксированной цены.

Возможные характеристики приложения для торговли невзаимозаменяемыми токенами: добавление заявки на обмен, хранение информации об обменах (адрес коллекции, идентификатор токена, информация, представляющая уникальность и создатель токена).

Таблица 15 - Морфологическая матрица для приложения для торговли невзаимозаменяемыми токенами

	1	2	3
А. Добавление заявок на продажу	Ручной ввод информации о заявке(адрес коллекции, идентификатор токена и информация, представляющая собой уникальность, цена продажи) с помощью полей ввода данных при отправке транзакции через	Подключение кошелька через метамаск, выбор токена из личного кабинета и подпись транзакции с установленными параметрами через интерфейс	Добавление заявки в базу данных сервера веб-приложения без подписи транзакции и отправки ее в блокчейн

	обозреватель блокчейна		
Б. Отмена заявки на продажу	Ручная отмена уже добавленной заявки с помощью обозревателя блокчейна	Подключение кошелька метамаск, выбор заявки в личном кабинете и отмена заявки через кнопку в интерфейсе, подпись транзакции и ожидание	Удаление записи о заявке в базе данных
В. Принятие заявки на продажу	Ручное принятие выбранной заявки с помощью обозревателя блокчейна	Подключение кошелька, выбор заявки через поиск и подпись транзакции в интерфейсе	Добавление записи об исполнении заявки, подпись и отправка в блокчейн

Из данной морфологической матрицы проекта было выделено три варианта решения технической задачи:

1. A1B1V1
2. A2B2V1
3. A3B3V3

Эти варианты исполнения будут использованы в далее расчетах временных показателей научного исследования, в результате было определено, что вариант исполнения И1 является самым эффективным и был успешно реализован на практике. Этот вариант также обладает высокой гибкостью и позволяет легко сопровождать внедренную систему, а также вводить новый функционал.

4.4 Планирование научно-исследовательских работ

4.4.1 Структура работ в рамках научного исследования

Перечень этапов, работ и распределение исполнителей в рамках научного исследования представлены в таблице ниже:

Таблица 16 - Перечень работ и распределение исполнителей

Основные этапы	№ работ	Содержание работ	Должность исполнителя
Разработка технического задания	1	Составление и утверждение технического задания	Инженер, научный руководитель
Аналитический обзор	2	Анализ предметной области	Инженер
	3	Анализ конкурентных решений	Инженер
	4	Календарное планирование работ	Инженер, научный руководитель
Проектирование и разработка	5	Разработка приложения	Инженер
Обобщение и оценка результатов	6	Оценка эффективности полученных результатов	Научный руководитель

Оформление отчётов по НИР	7	Составление пояснительной записки	Инженер
------------------------------	---	---	---------

4.4.2 Определение трудоёмкости выполнения работ

Трудовые затраты в большинстве случаев образуют основную часть стоимости разработки, поэтому важным моментом является определение трудоёмкости работ каждого из участников научного исследования.

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

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

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

t_{min} - минимально возможная трудоёмкость выполнения заданной i -ой работы, чел.-дн.;

t_{max} - максимально возможная трудоёмкость выполнения заданной i -ой работы, чел.-дн.;

Исходя из ожидаемой трудоемкости работ, определяется продолжительность каждой работы в рабочих днях T_p , учитывающая параллельность выполнения работ по нескольким исполнителями.

$$T_{pi} = \frac{t_{ожі}}{q_i}, (3)$$

где T_{pi} - продолжительность одной работы, раб. дн.;

$t_{ожі}$ - - ожидаемая трудоёмкость выполнения i -ой работы чел.-дн.;

$Ч_i$ - численность исполнителей, выполняющих одновременно одну и ту же работу на данном этапе, чел.

Таблица 17 - Временные показатели проведения научного исследования

Название работы	Исполнители	Трудоёмкость работ, человеко-дни									Длительность работ					
		T _{min}			t _{max}			t _{ожі}			T _{рі} , рабочие дни			T _{кі} , календарные дни		
		И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3	И1	И2	И3
Составление и утверждение технического задания	И (НР)	3 (3)	3 (3)	3 (3)	4 (5)	4 (5)	4 (5)	3,4 (3,8)	3,4 (3,8)	3,4 (3,8)	1,7 (1,9)	1,7 (1,9)	1,7 (1,9)	2,5 (2,8)	2,5 (2,8)	2,5 (2,8)
Анализ предметной области	И	4	4	4	8	8	8	5,6	5,6	5,6	5,6	5,6	5,6	8,2	8,2	8,2
Анализ конкурентных решений	И	2	2	2	4	4	4	2,8	2,8	2,8	2,8	2,8	2,8	4,1	4,1	4,1

Календарное планирование работ	И (НР)	2 (2)	2 (2)	2 (2)	3 (3)	3 (3)	3 (3)	2,4 (2,4)	2,4 (2,4)	2,4 (2,4)	1,2 (1,2)	1,2 (1,2)	1,2 (1,2)	1,7 (1,7)	1,7 (1,7)	1,7 (1,7)
Разработка приложения	И	25	40	55	50	65	80	35	50	65	35	50	65	51,8	74	96,2
Оценка эффективности полученных результатов	НР	3	3	3	5	5	5	3,8	3,8	3,8	3,8	3,8	3,8	5,6	5,6	5,6
Составление пояснительной записки	И	10	10	10	20	20	20	14	14	14	14	14	14	20,7	20,7	20,7
Итого								63,2 (10)	78,2 (10)	93,2 (10)	60,3 (6,9)	75,3 (6,9)	90,3 (6,9)	89 (10,1)	111,2 (10,1)	133,4 (10,1)

4.4.3 Разработка графика проведения научного исследования

Наиболее удобным и наглядным представлением проведения научных работ является построение ленточного графика в форме диаграммы Ганта.

Диаграмма Ганта – горизонтальный ленточный график, на котором работы по теме представляются протяжёнными во времени отрезками, характеризующимися датами начала и окончания выполнения данных работ.

Для удобства построение графика, длительность каждого из этапов работ из рабочих дней следует перевести в календарные дни. Для этого необходимо воспользоваться следующей формулой:

$$T_{ki} = T_{pi} * k_{\text{кал}}, (4)$$

где T_{ki} - продолжительность выполнения i -ой работы в календарных днях;

T_{pi} - продолжительность одной работы, раб. дн.;

$k_{\text{кал}}$ - коэффициент календарности.

Коэффициент календарности определяется по следующей формуле:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})}, (5)$$

где $T_{\text{кал}}$ - количество календарных дней в году

$T_{\text{вых}}$ - количество выходных дней в году;

$T_{\text{пр}}$ - количество праздничных дней в году.

Расчёт коэффициента календарности:

$$k_{\text{кал}} = \frac{T_{\text{кал}}}{T_{\text{кал}} - (T_{\text{вых}} + T_{\text{пр}})} = \frac{365}{365 - 118} = 1,48$$

Составлен план научного исследования, в котором разработан календарный план выполнения работ. Для построения таблицы временных показателей проведения НИ был рассчитан коэффициент календарности. С помощью показателей в табл. 8 был разработан календарный план-график проведения НИ по теме. Для иллюстрации календарного плана была использована диаграмма Ганта, указывающая на целесообразность проведения данного исследования.

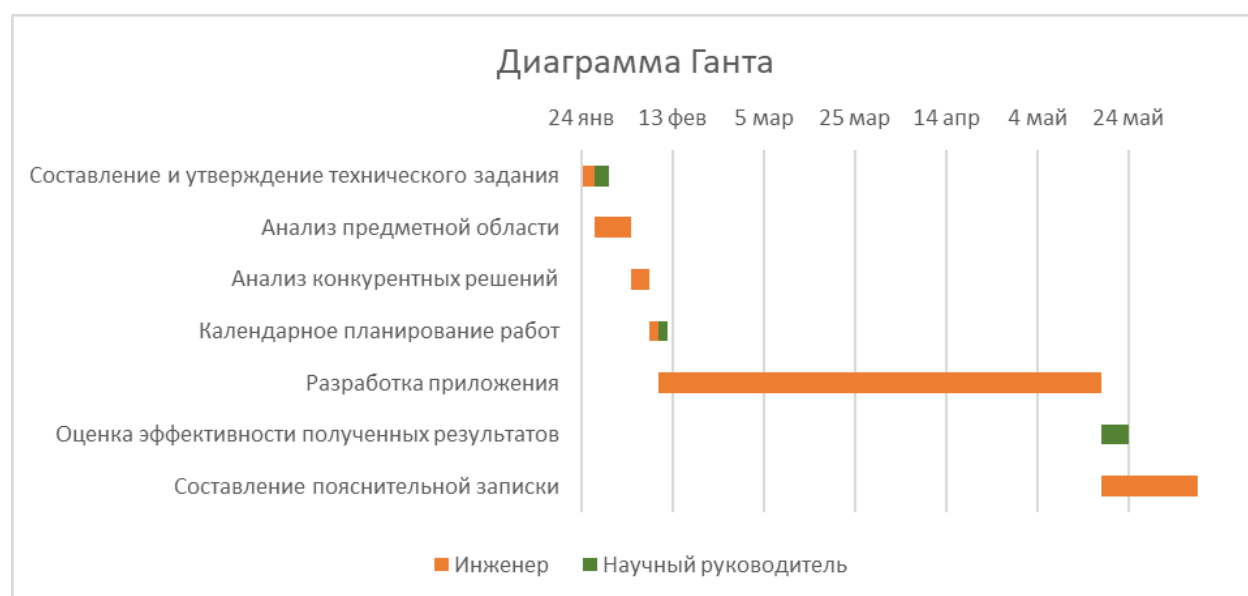


Рис. 22 Календарный план-график проведения научного исследования

1.3.4. Бюджет научно-технического исследования (НТИ)

Представим в таблице 18 расчет материальных затрат.

Таблица 18 - Материальные затраты

Наименование	Единица измерения	Количество	Цена за ед., руб.	Затраты, руб
электроэнергия	кВтч	250	3,5	875
интернет	Тариф * Месяц	2	242	485
Итого, руб				1360

Расчёт затрат по данной статье представлен в таблице 18. Все остальные ПО и специальные программы, которые были использованы, получено бесплатно.

Таблица 19 - Расчёт бюджета затрат на приобретение спецоборудования для научных работ

Наименование	Единица измерения	Количество	Цена за ед., тыс. руб.	Затраты, тыс. руб
Maibenben M543	Шт	1	30	34.5
Итого, тыс. руб				34.5

Расчёт основной заработной платы приводится в таблице 19.

Основная заработная плата руководителя рассчитывается по следующей формуле:

$$З_{осн} = З_{дн} * Т_p, (8)$$

где $З_{дн}$ - среднедневная заработная плата;

$Т_p$ - продолжительность работ, выполняемых научно-техническим работником, раб. дн.

Среднедневная заработная плата рассчитывается по формуле:

$$З_{дн} = \frac{З_m * M}{F_d}, (9)$$

где $З_m$ - месячный должностной оклад работника, руб.

M - количество месяцев работы без отпуска в течении года:

При отпуске в 24 раб. дней $M = 11,2$ месяца, 5-дневная неделя.

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

F_d - действительный годовой фонд рабочего времени научно-технического персонала, раб. дн.

Таблица 20 - Баланс рабочего времени

Показатели рабочего времени	Научный руководитель	Инженер
Календарное количество дней	365	365
Количество нерабочих дней:	66	118
Выходные дни	52	104
Праздничные дни	14	14
Потери рабочего времени:	48	24
Отпуск	48	24
Невыходы по болезни	-	-
Действительный годовой фонд рабочего времени	251	223

Месячный должностной оклад работника:

$$З_{\text{М}} = З_{\text{ТС}} * k_{\text{р}}, (10)$$

где $З_{\text{ТС}}$ - заработная плата по тарифной ставке, руб.

$k_{\text{р}}$ - районный коэффициент, равный 1,3 (для Томска).

Величина отчислений во внебюджетные фонды определяется исходя из следующей формулы:

$$З_{\text{внеб}} = k_{\text{внеб}} З_{\text{осн}}$$

где $k_{\text{внеб}}$ – коэффициент отчислений на уплату во внебюджетные фонды равный 30 %.

Расчёт основной заработной платы представлен в таблице 21.

Таблица 21 - Расчёт основной заработной платы

Исполнители	З _{ТС} , руб	k _р	З _м , руб	З _{дн} , руб	Т _р , раб.дн.			З _{осн} , руб			З _{внеб} , руб		
					И1	И2	И3	И1	И2	И3	И1	И2	И3
Научный руководитель	39300	1,3	51090	2019	6,9	6,9	6,9	13931,1	13931,1	13931,1	4179,33	4179,33	4179,33
Студент	16242	1,3	21115	1060	60,3	75,3	90,3	63918	79818	95718	19175,4	23945,4	28715,4
Итого								77849,1	93749,1	109649,1	23354,73	28124,73	32894,73

Определение бюджета затрат на научно–исследовательский проект приведено в таблице 22.

Таблица 22 - Расчёт бюджета затрат НТИ

Наименование статей	Сумма, руб		
	Исп. 1	Исп.2	Исп. 3
1. Материальные затраты НТИ	1360	1360	1360
2. Затраты на специальное оборудование для научных работ	34,5	34,5	34,5
3. Затраты по основной заработной плате	77849,1	93749,1	109649,1
4. Отчисления во внебюджетные фонды	23354,73	28124,73	32894,73
5. Бюджет затрат НТИ	102598,33	123268,33	143938,33

4.4.4 Определение ресурсной (ресурсосберегающей), финансовой, бюджетной, социальной и экономической эффективности исследования

Определение эффективности происходит на основе расчета интегрального показателя эффективности научного исследования. Его нахождение связано с определением двух средневзвешенных величин: финансовой эффективности и ресурсоэффективности.

Интегральный показатель финансовой эффективности научного исследования определяется как:

$$I_{\text{фин.р}}^{\text{исп.}i} = \frac{\Phi_{pi}}{\Phi_{\text{max}}}, (14)$$

$I_{\text{фин.р}}^{\text{исп.}i}$ - интегральный финансовый показатель разработки;

Φ_{pi} - стоимость i-го варианта исполнения;

Φ_{max} - максимальная стоимость исполнения научно-исследовательского проекта.

$$I_{\text{финр}}^{\text{исп. 1}} = \frac{\Phi_{p1}}{\Phi_{max}} = \frac{102598,33}{143938,33} = 0,71;$$

$$I_{\text{финр}}^{\text{исп. 2}} = \frac{\Phi_{p2}}{\Phi_{max}} = \frac{123268,33}{143938,33} = 0,86;$$

$$I_{\text{финр}}^{\text{исп. 3}} = \frac{\Phi_{p3}}{\Phi_{max}} = \frac{143938,33}{143938,33} = 1;$$

Интегральный показатель ресурсоэффективности вариантов исполнения объекта исследования можно определить следующим образом:

$$I_{pi} = \sum_{i=1}^n a_i * b_i, (15)$$

где I_{pi} - интегральный показатель ресурсоэффективности для i -го варианта исполнения разработки;

a_i - весовой коэффициент i -го варианта исполнения разработки;

b_i - бальная оценка i -го варианта исполнения разработки,

устанавливается экспертным путем по выбранной шкале оценивания;

n - число параметров сравнения.

Таблица 23 - Сравнительная оценка характеристик вариантов исполнения проекта

Критерии	Объект исследования			
	Весовой коэффициент	Исп.1	Исп.2	Исп.3
1. Функциональность	0,3	5	4	5
2. Пользовательский опыт	0,15	3	3	3
3. Надежность и стабильность	0,15	3	4	4

4. Совместимость и доступность	0,2	3	3	4
5.Эффективность и производительность	0,2	3	4	4
Итого	1	17	18	20

$$I_{p1} = 0,3 * 5 + 0,15 * 3 + 0,15 * 3 + 0,2 * 3 + 0,2 * 3 = 3,6$$

$$I_{p2} = 0,3 * 4 + 0,15 * 3 + 0,15 * 4 + 0,2 * 3 + 0,2 * 4 = 3,65$$

$$I_{p3} = 0,3 * 5 + 0,15 * 3 + 0,15 * 4 + 0,2 * 4 + 0,2 * 4 = 4,15$$

Интегральный показатель эффективности вариантов исполнения разработки (исп) определяется на основании интегрального показателя ресурсоэффективности и интегрального финансового показателя по формуле:

$$I_{исп\ i} = \frac{I_{pi}}{I_{фин.р}}, (17)$$

$$I_{исп\ 1} = \frac{3,6}{0,71} = 5,07;$$

$$I_{исп\ 2} = \frac{3,65}{0,86} = 4,24;$$

$$I_{исп\ 3} = \frac{4,15}{1} = 4,15;$$

Сравнение интегрального показателя эффективности вариантов исполнения разработки позволит определить сравнительную эффективность проекта и выбрать наиболее целесообразный вариант из предложенных.

Сравнительная эффективность проекта ($\mathcal{E}_{\text{ср}}$):

$$\mathcal{E}_{\text{ср}} = \frac{I_{\text{ИСП } 1}}{I_{\text{ИСП } 2}}, \quad (18)$$

Проведем сравнение относительно первого исполнения:

$$\mathcal{E}_{\text{ср}2} = \frac{5,07}{4,24} = 1,19;$$

$$\mathcal{E}_{\text{ср}3} = \frac{5,07}{4,15} = 1,22;$$

Таблица 24 - Сравнительная эффективность разработки

№	Показатели	Исп 1	Исп.2	Исп.3
1	Интегральный финансовый показатель разработки	0,71	0,86	1
2	Интегральный показатель ресурсоэффективности	3,6	3,65	4,15
3	Интегральный показатель эффективности	5,07	4,24	4,15
4	Сравнительная эффективность вариантов исполнения	1	1,19	1,22

Таким образом, сравнительный анализ эффективности различных вариантов исполнения приложения для торговли невзаимозаменяемыми токенами показал, что исполнение 2 превосходит текущий проект (исполнение 1) на 19%, а исполнение 3 превосходит текущий проект на 22%. Это объясняется тем, что исполнение 2 и 3 используют более расширенный набор функций и веб-сервер с базой данных. Однако,

исполнение 1, несмотря на более ограниченный набор функций, имеет намного более низкую стоимость реализации, так как требует меньшего числа аппаратных и программных систем.

Итак, приложение для торговли невзаимозаменяемыми токенами в рамках исполнения 1 обладает достаточно высокой ресурсоэффективностью по сравнению с двумя другими вариантами исполнения. Оно предоставляет основные функции для торговли, при этом имеет более доступную стоимость реализации.

5. Социальная ответственность

5.1 Введение

Область применения приложения - энтузиасты в сфере распределенных систем.

Приложение позволяет пользователям осуществлять торговлю невзаимозаменяемыми токенами в сети блокчейна Polygon.

Данное приложение будет применяться людьми, которые интересуются блокчейном и NFT-токенами. Они смогут выставлять токены на продажу и покупать уже выставленные токены.

Рабочей зоной является офис площадью 24 м². В офисе располагается компьютер для разработки приложения. Рабочим процессом в офисном помещении является разработка приложения.

5.2 Правовые и организационные вопросы обеспечения безопасности

Трудовой кодекс РФ регулирует отношения между организациями и их сотрудниками, в таких вопросах как: оплата и нормирование труда, наличие выходных, отпусков и так далее.

Согласно статье 91 Трудового Кодекса РФ «Понятие рабочего времени. Нормальная продолжительность рабочего времени» работодатель не должен заставлять своих подчинённых работать больше 40 часов в неделю.

В соответствии с законом об информации, информационных технологиях и о защите информации работодатель не должен собирать, хранить, использовать и распространять информацию о частной жизни своих работников без их письменного согласия.

В соответствии с законом о коммерческой тайне, работодатель должен обеспечить своих работников условиями для соблюдения

установленного режима коммерческой тайны. Результаты труда работников являются коммерческой тайной.

5.3 Производственная безопасность

Условия труда, в которых разрабатывается приложение, в том числе устройства, с помощью которых осуществляется разработка приложения, могут вызвать появление вредных и опасных факторов производства.

В соответствии с ГОСТ 12.0.003–2015 «Система стандартов безопасности труда (ССБТ). Опасные и вредные производственные факторы. Классификация» при разработке приложения могут возникнуть факторы, представленные в таблице ниже:

Таблица 25 - Возможные опасные и вредные производственные факторы в офисе

Факторы (ГОСТ 12.0.003-2015)	Нормативные документы
Отсутствие или недостаток необходимого искусственного освещения	СП 52.13330.2016 «Естественное и искусственное освещение». Актуализированная редакция СНиП 23-05-95
Повышенный уровень шума	СП 51.13330.2011 Защита от шума. Актуализированная редакция СНиП 23-03-2003
Нагрузка на зрительный аппарат	СП 52.13330.2016 «Естественное и искусственное освещение». Актуализированная редакция СНиП 23-05-95
Монотонность труда, вызывающая монотонию	Трудовой кодекс Российской Федерации от 30.12.2001 N 197-ФЗ (ред. от 25.02.2022) (с изм. и доп., вступ. в силу с 01.03.2022)
Факторы, связанные с электрическим током	ГОСТ 12.1.019-2017 «Электробезопасность. Общие требования и номенклатура видов защиты»

Действия пользователя осуществляются с помощью разработанного приложения. В силу того, что основная работа в приложении осуществляется с помощью экрана ноутбука, пользователь будет получать

негативные воздействия на зрительный аппарат, если будет пользоваться приложением более 4 часов свободного времени.

Соответственно, необходимо ограничить время работы в данном приложении в рамках установленных норм.

5.4 Анализ опасных и вредных производственных факторов

5.4.1 Отсутствие или недостаток необходимого искусственного освещения

Основной причиной возникновения данного вредного фактора является вредное воздействие освещения или недостаточное освещение рабочей зоны.

Освещение рабочего места, которое не удовлетворяет нормам может затруднять длительную работу так, как вызывает высокое утомление работников и способствует развитию нарушений зрения.

Нормы освещённости кабинетов, рабочих комнат, офисов и представительств указаны в СП 52. 13330.2016 «Естественное и искусственное освещение. Актуализированная редакция СНиП 23-05-95*».

Работа за персональным компьютером является работой очень высокой точности так, как наименьший размер объекта различения - это размер одного пикселя монитора, который равен 0,297 мм. Ниже приведены требования к освещению для данного типа работы.

Таблица 26 - Требования к освещению в офисе

Характеристика зрительной работы	Наименьший или эквивалентный размер объекта различения, мм	Искусственное освещение			
		Средняя освещённость на рабочей поверхности от системы общего освещения лк, не менее	Цилиндрическая освещённость, лк	Объединённый показатель UGR, не более	Коэффициент пульсации и освещённости К _п %, не более
Очень высокой точности	От 0,15 до 0,30	750	150	16	10

В офисах, где используются персональные компьютеры, освещение должно осуществляться системой общего равномерного освещения. Вдобавок, окна в офисе должны располагаться на севере или северо-востоке.

5.4.2 Повышенная пульсация светового потока

Причинами возникновения таких вредных факторов могут быть источники света, питаемые переменным или импульсным током.

При недолгом влиянии этого фактора чаще всего у работников появляется усталость органов зрения, может снизиться внимания, замедление активности мозга, также может быть тошнота и нарушение

пищеварения. При долгом влиянии последствия для организма могут быть более критическими так, например, бессонница, депрессия, патологии ЖКТ и органов зрения. В своде правил об естественном и искусственном освещении сказано, что коэффициент пульсации освещённости рабочей поверхности не должен превышать 20 процентов.

Использование источников света с коэффициентом пульсации меньше указанного ранее значения позволит избежать воздействия данного фактора на работников.

5.4.3 Нервно-психические перегрузки, связанные с напряжённостью трудового процесса

Данный вредный фактор отражает такое состояние организма человека, при котором происходит снижение эффективности труда на неопределённый промежуток времени, зачастую он проявляется неожиданно после совершения какой-либо работы.

Чтобы снизить влияние этого фактора на работников, работодателю следует предусмотреть перерывы во время рабочего дня для своих подчинённых. А работнику, в свою очередь, нужно правильно использовать время отдыха и работы, стараться ложиться спать в одно и то же время и заниматься спортом.

5.4.4 Статические перегрузки, связанные с рабочей позой

Данный фактор, чаще всего связан с рабочей позой, и зачастую может возникать из-за неправильного оснащения рабочего места, к примеру стул у работников на рабочем месте не даёт регулировать высоту или наклон спинки. Чтобы устранить влияние этого фактора, работодателю следует руководствоваться приказом Минтруда России об организации безопасного рабочего места. Рабочее место работников должно быть оборудовано креслом с возможностью регулировки его высоты и наклона спинки.

5.4.5 Длительное сосредоточенное наблюдение

Работа, в которой используются компьютеры, предполагает длительное наблюдение в монитор, это может вызвать постоянное напряжение зрительного анализатора. Чтобы снизить нагрузку, работодателю нужно предусмотреть дополнительные перерывы для своих работников.

5.4.6 Производственные факторы, связанные с электрическим током, вызываемым разницей электрических потенциалов, под действие которого попадает работающий

В процессе работы над приложением у работников могут появиться травмы от воздействий электрическим током. Это воздействие чаще всего может привести к серьёзным ожогам частей тела или даже к смерти, что является плохим результатом для работодателя. Поражение током, может произойти в результате, того что работник решил прикоснуться к частям техники или сделал это случайно, на которых появилось напряжение или остался заряд тока. Для переменного тока частотой 50 Гц допустимое значение напряжения прикосновения составляет 2 В, а силы тока – 0,3 мА, для тока частотой 400 Гц, соответственно – 2 В и 0,4 мА, для постоянного тока – 8 В и 1 мА.

Для того чтобы гарантировать защиту своим работникам, работодателю нужно использовать оградительные устройства, устройства автоматического контроля и сигнализации, изолирующие устройства и покрытия, устройства защитного заземления.

5.5 Экологическая безопасность

При эксплуатации проектного решения может произойти загрязнение литосферы нашей планеты.

Компьютеры которые используются для разработки приложения, имеют ограниченный срок службы и относятся к опасным отходам.

Согласно приказу Минприроды России «Об утверждении требований при обращении с группами однородных отходов I-V классов опасности» компаниям и индивидуальным предпринимателям было запрещено утилизировать компьютерную технику в мусорные баки. Поэтому для того, чтобы свести воздействие на литосферу к наиболее возможному минимуму, следует передавать компьютерную технику на утилизацию в специализированные центры.

В ходе деятельности также создавался бытовой мусор (канцелярские, пищевые отходы, искусственные источники освещения), который должен быть утилизирован в соответствии с определенным классом опасности или переработан, чтобы не оказывать негативное влияние на состояние литосферы и гидросферы.

5.6 Безопасность в чрезвычайных ситуациях

5.6.1 Анализ возможных ЧС

При эксплуатации проектного решения могут возникнуть следующие чрезвычайные ситуации: пожар, землетрясение.

5.6.2 Анализ наиболее вероятной ЧС, которая может возникнуть при разработке приложения

Ввиду того, что для разработки приложения необходимо использовать персональный компьютер, то наиболее возможной ЧС является пожар, случившийся из-за поломки оборудования или нарушения техники безопасности. В ГОСТ 12.1.004–91 «Пожарная безопасность. Общие требования», описаны нормы пожарной безопасности, которые необходимо соблюдать при работе за компьютером:

- Работы за персональным компьютером должны проводиться только при полностью исправном состоянии оборудования и электропроводки;

- В офисе должны быть установлены огнетушители;
- В офисе должны быть предусмотрены эвакуационные выходы и обеспечено беспрепятственное движение людей к ним.

В случае возникновения пожара работникам необходимо:

- Необходимо немедленно сообщить по телефону о пожаре в пожарную охрану;
- Оповестить о пожаре всех работников;
- Принять меры по эвакуации людей, тушению пожара, сохранению материальных ценностей.

В соответствии с Федеральным законом от 22.07.2008 N 123-ФЗ (ред. От 30.04.2021) «Технический регламент о требованиях пожарной безопасности», пожар, который возникнет будет относиться к классу А – пожары твёрдых горючих веществ и материалов. В офисе нужно предусмотреть следующие первичные средства пожаротушения: огнетушитель, пожарный кран (на этаже), покрывало для изоляции очага возгорания.

5.7 Вывод по разделу

В данном разделе были рассмотрены организационные и правовые нормы обеспечения безопасности при разработке приложения. Были проанализированы наиболее возможные опасные и вредные факторы производства, а также были приведены меры по снижению их воздействия.

Офис, в котором разрабатывается приложение соответствует категории по электробезопасности. Персонал в офисе относится к I группе по электробезопасности. Категория тяжести труда при разработке приложения – Ia. В помещении обращаются горючие и трудногорючие вещества и материалы, которые горят без образования взрывоопасных смесей и только при контакте с кислородом, водой или друг с другом, имеет

категорию В4 пожароопасности. Данный объект не оказывает какое-то значительное негативное воздействие на окружающую среду и не соответствует критериям нормативной документации, исходя из этого отнести его к какой-либо категории по этому фактору невозможно.

Заключение

В рамках ВКР был разработан децентрализованный маркетплейс, предоставляющий функционал для торговли NFT-токенами.

Создав децентрализованное приложение, была решена проблема отсутствия качественных децентрализованных маркетплейсов. Была создана экономическая площадка, в которой могут работать разработчики децентрализованных приложений, инвесторы, коллекционеры, художники и другие личности.

Для достижения поставленной цели были решены следующие задачи:

1. Исследованы существующие блокчейны для разрабатываемого приложения.
2. Исследованы сторонние площадки для обмена цифровыми активами.
3. Проанализированы применяемые технологии в ходе создания приложения.
4. Спроектировано децентрализованное приложение.
5. Написано децентрализованное приложение

В результате работы над программной системой были получены следующие навыки и компетенции:

1. Построение диаграмм при проектировании программной системы.
2. Разработка децентрализованных приложений для блокчейнов, совместимых с сетью Ethereum на языке Solidity.
3. Использование стандартов разработки токенов и не только, принятых сообществом разработчиков в сфере блокчейна.
4. Тестирование и развертывание децентрализованных приложений с помощью фреймворка Hardhat.

Разработка графического интерфейса не входила в список задач при работе над проектом из-за нехватки временного ресурса. Весь имевшийся временной ресурс был затрачен на изучение фреймворков для разработки и тестирования децентрализованных приложений, языка разработки смарт-контрактов Solidity, библиотек для взаимодействия с блокчейн-сетями и

различными стандартами разработки смарт-контрактов, принятых сообществом разработчиков.

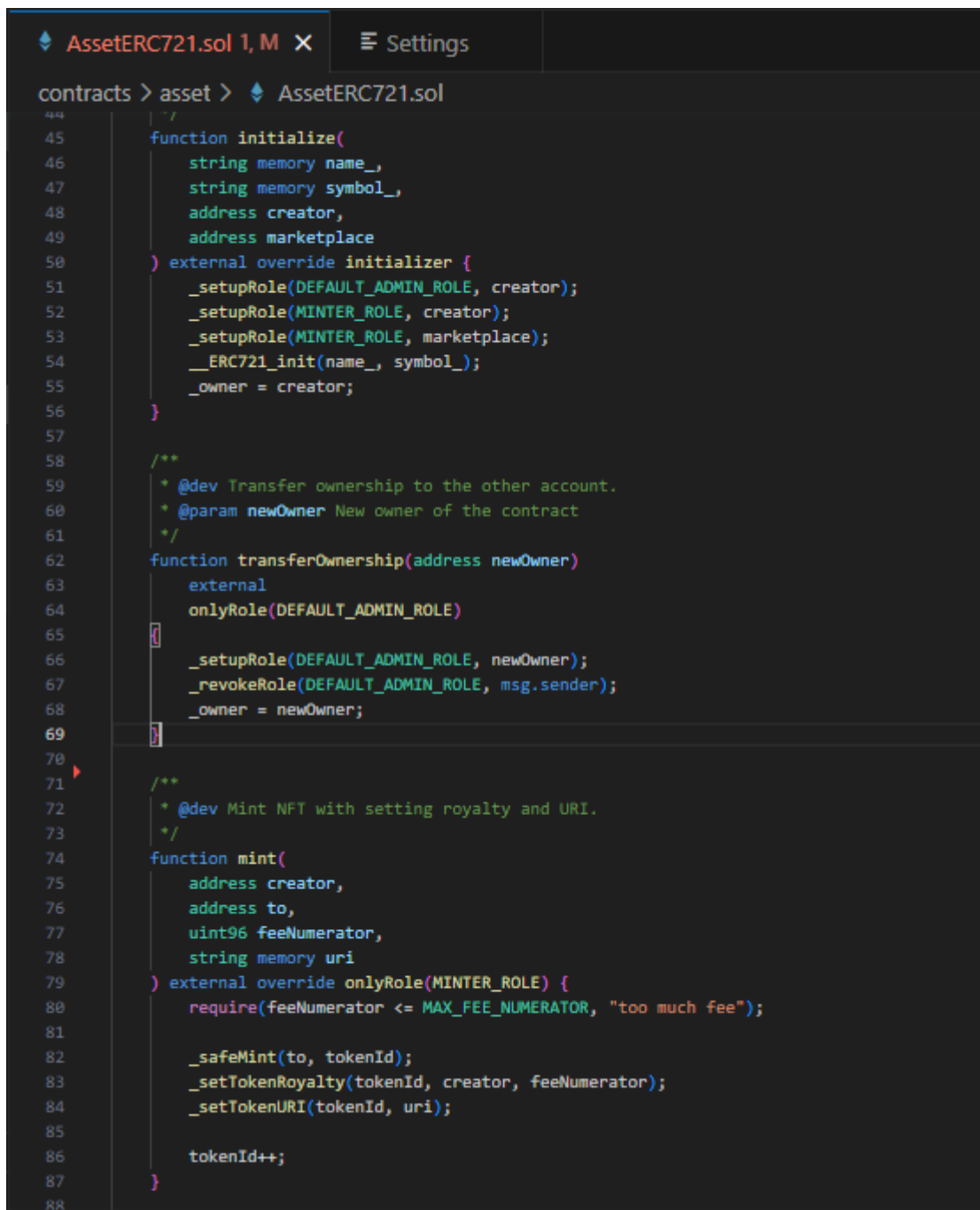
Список используемых источников

1. Технология блокчейн для бизнеса. Режим доступа – <https://vc.ru/crypto/150380-tehnologiya-blokcheyn-dlya-biznesa>
2. В России впервые допустили физлиц к сделкам с ЦФА.
Режим доступа – <https://www.vedomosti.ru/finance/articles/2022/11/29/952541-v-rossii-vpervie-dopustili-fizlits-k-sdelkam-s-tsfa>
3. Кому принадлежат авторские права на NFT: автору или покупателю?
Режим доступа – <https://www.binance.com/ru/blog/nft/%D0%BA%D0%BE%D0%BC%D1%83-%D0%BF%D1%80%D0%B8%D0%BD%D0%B0%D0%B4%D0%BB%D0%B5%D0%B6%D0%B0%D1%82-%D0%B0%D0%B2%D1%82%D0%BE%D1%80%D1%81%D0%BA%D0%B8%D0%B5-%D0%BF%D1%80%D0%B0%D0%B2%D0%B0-%D0%BD%D0%B0-nft-%D0%B0%D0%B2%D1%82%D0%BE%D1%80%D1%83-%D0%B8%D0%BB%D0%B8-%D0%BF%D0%BE%D0%BA%D1%83%D0%BF%D0%B0%D1%82%D0%B5%D0%BB%D1%8E-421499824684902084>
4. Цены, графики и рыночная капитализация криптовалют. Режим доступа – <https://coinmarketcap.com/ru/>
5. Ethereum. Режим доступа – <https://ethereum.org/en/>
6. Документация BNB Smart Chain. Режим доступа – <https://docs.bnbchain.org/docs/learn/intro>
7. Polygon. Режим доступа – <https://wiki.polygon.technology/docs/maintain/validator/architecture/>
8. OpenSea. Режим доступа – <https://opensea.io/about>
9. Rarible. Режим доступа – <https://rarible.com/explore/all/collections>
10. Все что вам нужно знать об архитектуре блокчейна.
Режим доступа – <https://icoda.io/ru/blog/all-you-need-to-know-about-blockchain-architecture/>
11. Криптовалюты: что такое fork? Режим доступа – <https://vc.ru/u/1134105-wallbtc/429356-kriptovalyuty-cto-takoe-fork>
12. Наиболее популярные алгоритмы консенсуса для блокчейн сетей.
Режим доступа – <https://vc.ru/u/478288-biki/122419-naibolee-populyarnye-algoritmy-konsensusa-dlya-blokcheyn-setey>

13. Концепции развития умных контрактов: Биткоин vs Ethereum.
Режим доступа – <https://forklog.com/exclusive/smart-contracts-concepts>
14. Что такое токены и какова их роль в система блокчейна?
Режим доступа – <https://vc.ru/crypto/398013-что-такое-tokeny-i-kakova-ih-rol-v-sisteme-blokcheyna>
15. Взаимозаменяемые и невзаимозаменяемые токены. В чем разница?
Режим доступа – <https://mpost.io/ru/fungible-vs-non-fungible-tokens-nft-what-is-the-difference/>
16. Hardhat фреймворк. Режим доступа – <https://hardhat.org/>
17. NatSpec Format.
Режим доступа – <https://docs.soliditylang.org/en/develop/natspec-format.html>
18. NFT Royalty Standart. Режим доступа – <https://eips.ethereum.org/EIPS/eip-2981>
19. Non-Fungible Token Standart.
Режим доступа – <https://eips.ethereum.org/EIPS/eip-721>
20. Multi Token Standart.
Режим доступа – <https://eips.ethereum.org/EIPS/eip-1155>
21. Minimal Proxy Standart.
Режим доступа – <https://eips.ethereum.org/EIPS/eip-1167>
22. UUPS. Режим доступа – <https://eips.ethereum.org/EIPS/eip-1822>
23. ABI.
Режим доступа – <https://docs.soliditylang.org/en/develop/abi-spec.html>
24. Chai.js. Режим доступа – <https://www.chaijs.com/>

Приложение А

На рисунках А.1 – А.5 представлены листинги программной системы.



```
contracts > asset > AssetERC721.sol
44  //
45  function initialize(
46      string memory name_,
47      string memory symbol_,
48      address creator,
49      address marketplace
50  ) external override initializer {
51      _setupRole(DEFAULT_ADMIN_ROLE, creator);
52      _setupRole(MINTER_ROLE, creator);
53      _setupRole(MINTER_ROLE, marketplace);
54      __ERC721_init(name_, symbol_);
55      _owner = creator;
56  }
57
58  /**
59   * @dev Transfer ownership to the other account.
60   * @param newOwner New owner of the contract
61   */
62  function transferOwnership(address newOwner)
63      external
64      onlyRole(DEFAULT_ADMIN_ROLE)
65  {
66      _setupRole(DEFAULT_ADMIN_ROLE, newOwner);
67      _revokeRole(DEFAULT_ADMIN_ROLE, msg.sender);
68      _owner = newOwner;
69  }
70
71  /**
72   * @dev Mint NFT with setting royalty and URI.
73   */
74  function mint(
75      address creator,
76      address to,
77      uint96 feeNumerator,
78      string memory uri
79  ) external override onlyRole(MINTER_ROLE) {
80      require(feeNumerator <= MAX_FEE_NUMERATOR, "too much fee");
81
82      _safeMint(to, tokenId);
83      _setTokenRoyalty(tokenId, creator, feeNumerator);
84      _setTokenURI(tokenId, uri);
85
86      tokenId++;
87  }
88
```

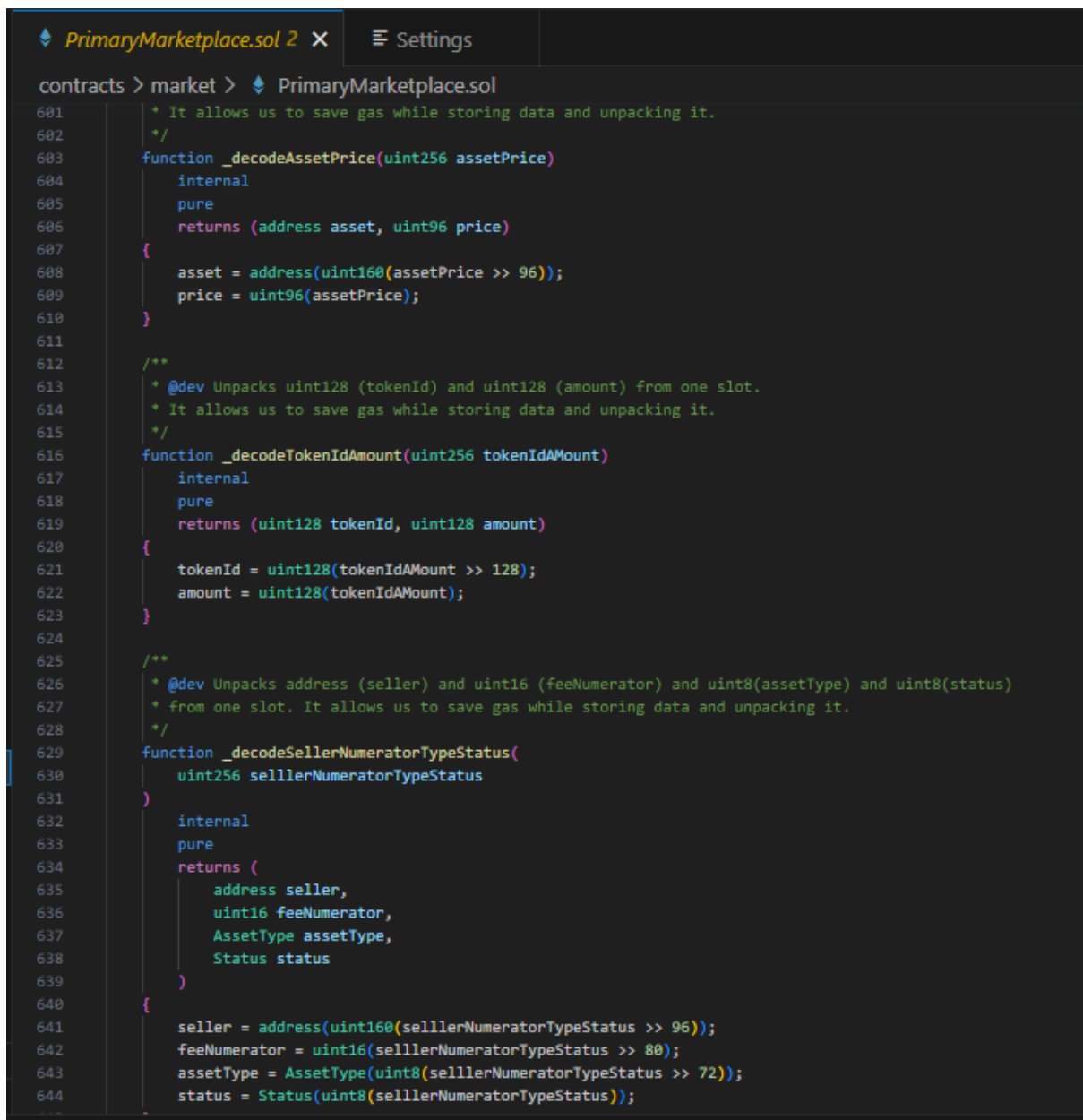
Рис. А.1 Листинг токена ERC721

```
contracts > asset > IAssetERC1155.sol
1  //SPDX-License-Identifier: Unlicense
2  pragma solidity ^0.8.0;
3
4  interface IAssetERC1155 {
5      /**
6       * @dev Initialize asset
7       * @param uri_ base URI
8       * @param creator address of the contract owner
9       * @param marketplace address of the marketplace for giving permission to mint
10     */
11     function initialize(
12         string memory uri_,
13         address creator,
14         address marketplace
15     ) external;
16
17     /**
18     * @dev Mint token.
19     * @param creator creator of the NFT
20     * @param to address you want to mint to
21     * @param id token ID you want to mint
22     * @param amount amount of the tokens
23     * @param data data
24     * @param feeNumerator numerator for setting fee. See eip2981
25     */
26     function mint(
27         address creator,
28         address to,
29         uint256 id,
30         uint256 amount,
31         bytes memory data,
32         uint96 feeNumerator
33     ) external;
34 }
```

Рис. А.2 Листинг интерфейса токена ERC1155

```
Factory.sol M X Settings
contracts > factory > Factory.sol
177
178
179 /**
180  * @dev See {IFactoryERC1155-mint721}
181  */
182 function mint721(
183     string memory name,
184     string memory symbol,
185     bytes memory signature
186 )
187     external
188     override
189 {
190     bytes32 data = keccak256(abi.encodePacked(name, symbol, msg.sender));
191     _validateSignature(data, signature);
192
193     address clone = ClonesUpgradeable.clone(_assetERC721);
194     AssetERC721(clone).initialize(name, symbol, msg.sender, _marketplace);
195     _assetsERC721.push(clone);
196
197     _amountERC721++;
198
199     emit MintedERC721(msg.sender, name, symbol, clone);
200 }
201
202 /**
203  * @dev See {IFactoryERC1155-mint1155}
204  */
205 function mint1155(string memory uri, bytes memory signature)
206     external
207     override
208 {
209     bytes32 data = keccak256(abi.encodePacked(uri, msg.sender));
210     _validateSignature(data, signature);
211
212     string memory URI = string(abi.encodePacked(PREFIX, uri));
213     address clone = ClonesUpgradeable.clone(_assetERC1155);
214     AssetERC1155(clone).initialize(URI, msg.sender, _marketplace);
215     _assetsERC1155.push(clone);
216
217     _amountERC1155++;
218
219     emit MintedERC1155(URI, msg.sender, clone);
220 }
```

Рис. А.3 Листинг фабрики токенов



```
contracts > market > PrimaryMarketplace.sol
601  * It allows us to save gas while storing data and unpacking it.
602  */
603  function _decodeAssetPrice(uint256 assetPrice)
604      internal
605      pure
606      returns (address asset, uint96 price)
607  {
608      asset = address(uint160(assetPrice >> 96));
609      price = uint96(assetPrice);
610  }
611
612  /**
613   * @dev Unpacks uint128 (tokenId) and uint128 (amount) from one slot.
614   * It allows us to save gas while storing data and unpacking it.
615   */
616  function _decodeTokenIdAmount(uint256 tokenIdAMount)
617      internal
618      pure
619      returns (uint128 tokenId, uint128 amount)
620  {
621      tokenId = uint128(tokenIdAMount >> 128);
622      amount = uint128(tokenIdAMount);
623  }
624
625  /**
626   * @dev Unpacks address (seller) and uint16 (feeNumerator) and uint8(assetType) and uint8(status)
627   * from one slot. It allows us to save gas while storing data and unpacking it.
628   */
629  function _decodeSellerNumeratorTypeStatus(
630      uint256 sellerNumeratorTypeStatus
631  )
632      internal
633      pure
634      returns (
635          address seller,
636          uint16 feeNumerator,
637          AssetType assetType,
638          Status status
639      )
640  {
641      seller = address(uint160(sellerNumeratorTypeStatus >> 96));
642      feeNumerator = uint16(sellerNumeratorTypeStatus >> 80);
643      assetType = AssetType(uint8(sellerNumeratorTypeStatus >> 72));
644      status = Status(uint8(sellerNumeratorTypeStatus));
```

Рис. А.3 Листинг кодирования данных в первичном маркетплейсе

```
PrimaryMarketplace.sol 2 X Settings
contracts > market > PrimaryMarketplace.sol
430
431 function _validatingParams(
432     address asset,
433     uint128 tokenId,
434     uint256 amount,
435     uint16 feeNumerator,
436     AssetType assetType,
437     string memory uri,
438     bytes memory validatorSignature,
439     bytes memory sellerSignature,
440     address seller,
441     uint256 bid,
442     address buyer,
443     uint256 nonce
444 ) internal {
445     bytes32 sellerData = keccak256(
446         abi.encodePacked(
447             asset,
448             tokenId,
449             amount,
450             feeNumerator,
451             assetType,
452             uri,
453             seller,
454             nonce
455         )
456     );
457     if(_usedSignatures[sellerData] == true) {
458         revert InvalidSellerData();
459     }
460     _usedSignatures[sellerData] = true;
461
462     bytes32 validatorData = keccak256(
463         abi.encodePacked(
464             sellerData,
465             bid,
466             buyer
467         )
468     );
469     _validateSignature(sellerData, sellerSignature, seller);
470     _validateSignature(validatorData, validatorSignature, _validator);
471 }
472
473 /**
```

Рис. А.4 Листинг валидирования параметров при создании ордера в первичном маркетплейсе.