МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

|  |
| --- |
| ДОПУСТИТЬ К ЗАЩИТЕ  Заведующий кафедрой, д.ф.-м.н., профессор  \_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

**Разработка системы электронного голосования на основе технологии блокчейн**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 09.03.04.2023.308-276.ВКР

|  |  |
| --- | --- |
|  | Научный руководитель,  доцент кафедры СП, к.ф.-м.н. \_\_\_\_\_\_\_\_\_\_ Г.И. Радченко  Автор работы, студент группы КЭ-403  \_\_\_\_\_\_\_\_\_\_ В.О. Богатырева  Ученый секретарь  (нормоконтролер)  \_\_\_\_\_\_\_\_\_\_\_\_\_ И.Д. Володченко  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2023 г. |

Челябинск, 2023 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

06.02.2023 г.

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**

студентке группы КЭ-403

Богатыревой Веронике Олеговне,

обучающейся по направлению

09.03.04 «Программная инженерия»

1. **Тема работы** (утверждена приказом ректора от \_\_.\_\_.2023 г. № \_\_\_)

Разработка системы электронного голосования на основе технологии блокчейн.

1. **Срок сдачи студентом законченной работы:** 05.06.2023 г.
2. **Исходные данные к работе2**
3. Фролов А.В. Создание смарт-контрактов Solidity для блокчейна Ethereum. Практическое руководство. – «ЛитРес: Самиздат», 2019.
4. Прасти Н. Блокчейн. Разработка приложений – «СПб.: БВХ-Петербург», 2018.
5. Trubochkina N., Poliakov S. The Concept Of Electronic Voting Based On Blockchain // INFORMACIONNYE TEHNOLOGII, 2019, 25(2), 75–85. [DOI: 10.17587/it.25.75-85](https://doi.org/10.17587/it.25.75-85).
6. Metamask.io. [Электронный ресурс] URL: https://metamask.io/ (дата обращения 11.02.2023 г.).
7. **Перечень подлежащих разработке вопросов**
8. Выполнить обзор литературы и существующих аналогов.
9. Спроектировать смарт-контракт для электронного голосования на основе технологии блокчейн.
10. Спроектировать веб-приложение для электронного голосования на основе технологии блокчейн.
11. Реализовать смарт-контракт и веб-приложение.
12. Провести тестирование работы приложения.
13. **Дата выдачи задания:** 06.02.2023 г.

**Научный руководитель,**

доцент кафедры СП, к.ф.-м.н. Г.И. Радченко

**Задание принял к исполнению** В.О. Богатырева

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 5](#_Toc127490738)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc127490739)

[1.1. Предметная область проекта 7](#_Toc127490740)

[1.2. Подходы к реализации методов голосования с использованием блокчейн 8](#_Toc127490741)

[1.3. Анализ аналогичных проектов 10](#_Toc127490742)

[1.4. Краткий обзор технологий для разработки веб-приложений 13](#_Toc127490743)

[2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 15](#_Toc127490744)

[2.1. Блокчейн 15](#_Toc127490745)

[2.2. Блокчейн Ethereum 15](#_Toc127490746)

[2.3. Смарт-контракт 16](#_Toc127490747)

[2.4. DApp 17](#_Toc127490748)

[3. ПРОЕКТИРОВАНИЕ 18](#_Toc127490749)

[3.1. Функциональные требования 18](#_Toc127490750)

[3.2. Нефункциональные требования 18](#_Toc127490751)

[3.3. Диаграмма вариантов использования 19](#_Toc127490752)

[3.4. Компоненты системы 20](#_Toc127490753)

[3.4.1. Компоненты смарт-контракта 21](#_Toc127490754)

[3.4.2. Компоненты веб-приложения 21](#_Toc127490755)

[3.5. Диаграмма деятельности 22](#_Toc127490756)

[3.4. Разработка макетов 23](#_Toc127490757)

[ЗАКЛЮЧЕНИЕ 26](#_Toc127490758)

[ЛИТЕРАТУРА 27](#_Toc127490759)

# ВВЕДЕНИЕ

**Актуальность**

В настоящее время выборы в каждой стране – это одна из самых важных и трудоемких задач, когда необходимо в жестко ограниченное время получить и обработать информацию от миллионов граждан. Существующие системы голосования часто дают сбои. Но блокчейн-голосование убирают из выборного процесса практически любой риск.

Многие страны мира, включая ЕС, Россию, Австралию, всерьез рассматривают переход на голосование на основе блокчейн. Не исключено, что в ближайшем будущем мы будем выбирать государственных лидеров, не опуская бюллетени в урны, а запуская смарт-контракт.

К тому же данный подход может помочь и с проблемой явки избирателей там, где инерция вызвана длинными очередями, заполнением множества бумаг и прочей волокитой.

**Постановка задачи**

Целью выпускной квалификационной работы является разработка системы электронного голосования на основе технологии блокчейн. Для достижения поставленной цели необходимо решить следующие задачи:

1. произвести обзор научной литературы и существующих аналогов;
2. спроектировать смарт-контракт и веб-приложение для электронного голосования;
3. реализовать смарт-контракт и веб-приложение;
4. провести тестирование работы приложения.

**Структура и содержание работы**

Работа состоит из введения, пяти глав, заключения и списка литературы. Объем работы составляет 40 страниц, объем списка литературы – 14 источников.

В первой главе описываются предметная область и аналогичные проекты по созданию электронного голосования.

Вторая глава содержит описание теоретической части по теме работы.

Третья глава посвящена определению функциональных и нефункциональных требований к системе и проектированию ее архитектуры.

Четвертая глава содержит в себе подробности и особенности реализации смарт-контракта для электронного голосования.

В пятой главе описывается процесс тестирования смарт-контракта.

В приложениях содержится спецификация диаграммы вариантов использования, код стандарта токенов ERC20 и протоколы функционального тестирования.

# 1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Предметная область проекта

Блокчейн – это особая структура данных, применяемая для создания децентрализованного регистра. Блокчейн состоит из блоков (block), особым образом соединенных в цепочку (chain). Блок содержит набор транзакций, хеш предыдущего блока, метку времени (время создания блока), сумму отчисления майнеру за блок и т. д. Поскольку каждый блок содержит хеш предыдущего блока, они связаны в цепочку. Каждый узел сети хранит полную копию блокчейна [1].

Проблемы традиционных избирательных систем рассматриваются в работах [2-4]. Авторы утверждают, что существующие методы не могут обеспечить достаточный уровень прозрачности и надежности голосования.

В последнее время электронные системы голосования стали использоваться во многих странах. Эстония первой в мире внедрила электронную систему голосования на национальных выборах. Вскоре после этого электронное голосование было принято Швейцарией для выборов в масштабах штата и Норвегией для выборов в совет. Но данные электронные системы голосования требуют серьезных доработок в области безопасности и анонимности [3].

В настоящее время блокчейн предлагает новые возможности для разработки приложения благодаря ключевым особенностям этой технологии, таким как прозрачность и защищенность процесса передачи данных.

Как описано в статье [5], преимуществами применения смарт-контрактов в сочетании с технологией блокчейн в процессе голосования являются:

1. прозрачность процесса голосования – любой человек получит возможность контролировать ход голосования;
2. анонимность голоса – любой из избирателей генерирует индивадуальный приватный и публичный ключ, который он имеет право не разглашать другим участникам голосования;
3. подлинность и надежность результатов – результаты голосования невозможно сфальсифицировать, так как любой участник голосования может проверить сколько токенов-голосов было выпущено в начале голосования и как они распределялись по кошелькам после;
4. экономическая целесообразность и скорость обработки данных.

1.2. Подходы к реализации методов голосования с использованием блокчейн

Блокчейн-голосования реализуются с помощью смарт-контрактов различными алгоритмами и методами. Рассмотрим делегированное голосование, голосование с использованием токенов и весовое голосование.

**Делегированное голосование**

В алгоритме делегированного голосования [6] лица, находящие в списке избирателей, могут либо голосовать сами, либо делегировать свой голос человеку, которому они доверяют.

Для этого подхода требуется указать адрес, на который будет начислен голос, а также проверить, голосовал ли избиратель и не совпадает ли его адрес с адресом делегата. После вызова метода делегирования, доверенное лицо имеет право голоса в блокчейн-голосовании.

Таким образом, в блокчейне содержатся данные о передаче права голоса другому лицу, а также транзакции о голосовании за выбранного кандидата.

**Голосование с использованием токенов**

Данный метод описывается в работах [5,7] на конкретном примере голосования. Для начала избиратель выбирает кандидата, на адрес которого будет переведен токен (голос). Эта транзакция пересылается в состоящую из компьютеров сеть равноправных узлов, называемых «нодами». Сеть нода подтверждает транзакцию, используя алгоритмы консенсуса. После подтверждения транзакция объединяется с другими подтвержденными транзакциями, формируя новый блок цифрового реестра. Затем данный блок добавляется в блокчейн с использованием хэша предыдущего блока.

Авторы утверждают, что транзакция, записанная в блокчейн, гарантирует ее достоверность и защищенность, а выбранный кандидат получает «голос», что автоматически отображается для всех наблюдателей (рисунок 1).



Рисунок 1 – Алгоритм работы голосования с использованием токенов

**Весовое голосование**

Метод голосования с весами предоставляет возможность назначать и учитывать веса участников пропорционально их доле в уставном капитале общества. Чтобы голосование было легитимным, кворум должен составлять 50% + 1 участник организации [8]. Данный подход может применяться, к примеру, при голосовании акционеров.

В работе [6] рассматривается случай, когда наибольший вес голоса имеет председатель голосования, т.е. создатель смарт-контракта. Остальные участники голосования не имеют права решающего голоса.

В работах [9, 10] применяется метод весового голосования. В сервисе существует возможность проведения данных голосований. Также авторы указывают на возможность указать требования по кворуму, чтобы оно стало легитимным.

1.3. Анализ аналогичных проектов

**Polys** [11]

Polys – система онлайн-голосований на блокчейне, созданная на базе «Лаборатории Касперского». За два года существования проекта на платформе проголосовали российские и зарубежные вузы, прошли [голосования](https://coinspot.io/technology/v-partii-yabloko-proshli-pervye-v-rossii-prai%CC%86meriz-s-golosovaniyami-na-blokchei%CC%86ne/) на выборах мэра Москвы, в Саратовский Молодежный Парламент.

Решение Polys представляет различные виды бюллетеней: с единичным и множественным выбором, с распределением кумулятивных голосов, бюллетени для референдума. В одних случаях не требуется анонимность, в других каждый голос участника голосования имеет определенный вес. Для этого авторы Polys обеспечили модульность системы [10].

Ключевые компоненты проекта:

1. блокчейн-платформа, которая предоставляет среду для выполнения смарт-контрактов, определяющих логику процесса голосования;
2. сервисный слой, отвечающий за аутентификацию пользователя, подпись зашифрованных сообщений и отправку уведомлений;
3. библиотека polys-protocol, которая обеспечивает слой абстракции над протоколом взаимодействия с блокчейном.

Блокчейн-платформа построена на базе фреймворка Exonum, написанного на языке программирования Rust.

Система работает следующим образом: зашифровываются сами голоса и шифруется личность избирателя. Каждый избиратель может проверить, что его голос принят и учтён в интерфейсе веб-приложения для голосования. Подсчет результатов происходит автоматически. Polys расшифровывает общий результат, а не каждый голос по отдельности. Это сделано для того, чтобы сохранить промежуточные результаты в тайне до конца голосования.

Рассмотрим панель создания голосования в системе. Пользователь может ввести название или основной вопрос голосования, задать ему фоновое изображение, ввести название организации. Существует возможность выбрать тип бюллетеня, создать список избирателей и период голосования. Затем необходимо добавить варианты ответов. Панель организатора при создании голосования изображена на рисунке 2.



Рисунок 2 – Панель организатора при создании голосования

**Московское голосование** [12]

Московское голосование **–**электронное голосование для жителей Москвы, которое проводится на портале мэра и правительства столицы. На основе этого сервиса были проведены выборы в Мосгордуму и выборы муниципальных депутатов.

Для проведения онлайн-голосования используется блокчейн Ethereum с алгоритмом консенсуса PoA (Proof of Authority). В данной системе любой желающий может отслеживать все происходящее в блокчейне, но создавать новые блоки могут только специальные узлы-валидаторы. За основу было взято программное обеспечение портала mos.ru. Анонимность избирателя гарантирует прокси-сервер, который генерирует уникальные ссылки на бюллетень, формируемые случайным образом [13].

Для участия в голосовании нужно было оставить заявку на включение в реестр избирателей на сайте mos.ru. Во время обработки заявки все необходимые данные, которые указал избиратель, проходят проверку. В день голосования избиратель переходит в раздел «Услуги» электронных выборов и получает доступ к форме голосования. Далее избиратель проставляет галочки напротив выбранных вариантов ответа и нажимает «Проголосовать». Анонимайзер генерирует ссылку и доступ к анонимному бюллетеню через личный кабинет: такая ссылка создается всего один раз. Блокчейн обеспечивает сохранение данных в ходе голосования и их интерпретацию по завершении голосования. Он размещается в отдельном защищенном сегменте сети внутри центра обработки данных. Система печати бюллетеней отвечает за вывод данных на ленту в ходе голосования [14].

**WE.Vote** [15]

WE.Vote – сервис блокчейн-голосований, разработанный российской компанией Waves Enterprise [16]. Сервис WE.Vote наиболее полно реализует концепцию безопасной системы онлайн-голосования. Сервис основан на блокчейн-сети Waves Enterprise Mainnet.

Со стороны WE.Vote выглядит как обычный сервис дистанционных голосований. Организатор голосования заходит в веб-интерфейс сервиса, создает новое голосование, настраивает бюллетени и добавляет электронные адреса участников. Далее бюллетени рассылаются участникам, система подсчитывает голоса и выдает готовый отчет с итогами голосования. Порядок не отличается от привычного для традиционных голосований.

На рисунке 3 изображена главная страница, на которой отображаются все голосования пользователя. В данной панели существует возможность найти конкретное голосование по его названию, сортировать все голосования по дате и его статусу.



Рисунок 3 – Главная страница We.Vote

1.4. Краткий обзор технологий для разработки веб-приложений

Рассмотрим основные фреймворки и библиотеки для разработки веб-приложений.

**Angular** [17]

Это JavaScript-фреймворк от Google [18], совместимый с большинством распространенных редакторов кода. Angular предназначен для создания динамических одностраничных веб-приложений (SPA – Single Page Applications). Фреймворк использует компонентный подход, а также преобразовывает документы на основе HTML [19] в динамический контент.

**Vue.js** [20]

Фреймворк с открытым исходным кодом для одностраничных приложений, который требует знания HTML и CSS [21]. Он использует модель разработки на основе компонентов и позволяет присоединять компоненты к проекту. Vue известен небольшим размером документов и синтаксисом на основе HTML. Так как фреймворк является самым «молодым», размер сообщества разработчиков небольшой.

**React.js** [22]

Библиотека для создания динамических пользовательских интерфейсов, которая была разработана компанией Facebook [23]. React основан на JavaScript [24] и [JSX](https://reactjs.org/docs/introducing-jsx.html) и позволяет создавать HTML-элементы для многократного использования. По сравнению со своими аналогами React достаточно быстро справляется с прогрузкой и ререндерингом страницы. Данная библиотека является наиболее популярной технологией для децентрализованных приложений.

**Выводы по первой главе**

В этой главе были рассмотрены предметная область проекта, подходы к реализации методов голосования, аналогичные проекты и технологии для разработки веб-приложений.

Таким образом, было принято решение реализовать систему электронного голосования на основе технологии блокчейн. В качестве инструментов разработки смарт-контракта был выбран блокчейн Ethereum и язык программирования Solidity. Для разработки веб-приложения была выбрана библиотека React.

# 2. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1. Блокчейн

Блокчейн [2] – это защищенный от несанкционированного доступа цифровой реестр общего пользования, который ведет учет транзакций в публичной или закрытой сети. Распределенный между всеми узлами сети реестр непрерывно записывает историю операций с активами между узлами сети в виде блоков информации.

Все утвержденные блоки транзакций соединяются в цепочку – с начального блока до последнего добавленного, отсюда и название технологии – блокчейн (англ. block chain – цепочка блоков).

Вместо того чтобы обращаться к третьим лицам, например, финансово-кредитным организациям, в качестве посредников при проведении транзакций, узлы блокчейн-сети используют специальный протокол консенсуса для согласования содержимого реестра, а также криптографические алгоритмы хеширования и электронно-цифровые подписи для обеспечения целостности транзакции и передачи ее параметров.

2.2. Блокчейн Ethereum

Блокчейн Ethereum [1] представляет собой платформу, на базе которой можно создавать распределенные приложения DApp – программы, работа которых поддерживается распределенной сетью узлов. В отличие от других платформ, Ethereum позволяет использовать так называемые умные контракты (смарт-контракты, smart contracts), написанные на языке программирования Solidity.

Эта платформа была создана в 2013 году Виталиком Бутериным, основателем журнала Bitcoin Magazine, и запущена в 2015 году.

2.3. Смарт-контракт

Смарт-контракт [2] представляет собой программный код, работающий в среде виртуальной машины Ethereum. Он запускается на всех узлах сети, и результаты его работы также реплицируются на все узлы.

С помощью смарт-контрактов очень удобно отслеживать выполнение транзакций. Если смарт-контракт получил тем или иным способом подтверждение выполнения условий сделки, то он может сам, автоматически, перевести средства поставщику. Если условия сделки были выполнены не полностью или не выполнены вовсе, смарт-контракт может вернуть средства покупателю или перевести сумму штрафа на счет пострадавшей стороны.

Смарт-контракт может хранить данные, например, значения баланса, флаги, строки и числа, идентификаторы документов, загруженных в Ethereum, таких как сканы прав владения чем-либо, выписки и т.п.

Одной особенностью смарт-контракта является то, что его нельзя оспорить. Если логика смарт-контракта сработает таким образом, что средства будут переведены, эти средства уже невозможно будет вернуть. Достоинством смарт-контрактов является то, что на их работу требуется очень мало средств.

Сравните это с оплатой юристов и различного рода пошлин в суде. Поэтому смарт-контракты можно использовать для поддержки очень недорогих сделок. Но вот разработка смарт-контракта может потребовать определенных затрат, в том числе и на юристов, с которыми имеет смысл согласовать логику работы смарт-контракта.

Кроме того, если с обычными контрактами при возникновении проблем вам могут помочь юристы и суды, то результаты выполнения смарт-контрактов будет очень сложно оспорить, если вообще возможно.

Перед публикацией смарт-контракта в сети Ethereum его необходимо компилировать в байт-код. Далее этот код сохраняется в сети с помощью транзакции.

2.4. DApp

DApp, или децентрализованное приложение – приложение, которое базируются на технологии блокчейн совместно с механизмом распределённого выполнения необходимых инструкций. В децентрализованных приложениях используются главные преимущества блокчейна: прозрачность, надежность и неизменность данных.

У децентрализованного приложения есть бэкенд-код, который работает в децентрализованной одноранговой сети. Децентрализованное приложение может иметь фронтенд-код и пользовательский интерфейс на любом языке (как и обычное приложение) для запросов к бэкенду. Более того, фронтенд может быть размещен в децентрализованном хранилище, таком как IPFS [3].

DApp могут стать важным компонентом будущего без цензуры, однако и они не лишены недостатков. Децентрализованные приложения находятся на ранних стадиях развития, и им еще предстоит решить проблемы масштабируемости, модификации кода и небольшой базы пользователей [4].

**Выводы по второй главе**

Во второй главе была рассмотрена основная теория по технологии блокчейн, блокчейн Ethereum, смарт-контрактам и децентрализованным приложениям.

# 3. ПРОЕКТИРОВАНИЕ

3.1. Функциональные требования

Можно выделить следующий набор функциональных требований к системе.

1. Система должна иметь возможность подключить кошелек гостем системы.
2. Система должна иметь возможность просмотреть голосование гостем и пользователем системы.
3. Система должна иметь возможность создать голосование пользователем системы.
4. Система должна иметь возможность проголосовать пользователем системы.

3.2. Нефункциональные требования

Можно выделить следующие нефункциональные требования к системе.

1. Система должна быть открытой (возможность проверки процесса голосования в режиме реального времени).
2. Смарт-контракты должны быть написаны на языке программирования Solidity.
3. Веб-приложение должно быть доступно в браузерах Google Chrome, FireFox, Яндекс Браузер последних версий.
4. Веб-приложение должно быть разработано с использованием таких инструментов, как: React, TypeScript, MUI, Redux Toolkit.

3.3. Диаграмма вариантов использования

Для проектирования системы был использован язык графического описания для объектного моделирования UML. Составлена диаграмма вариантов использования (рисунок 3). В системе определены два вида актера: гость и пользователь.

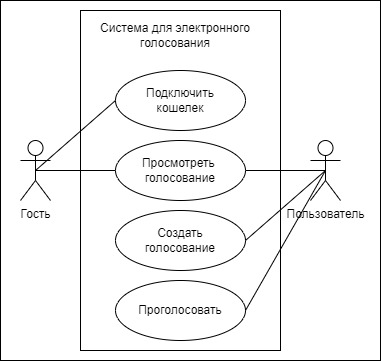


Рисунок 3 – Диаграмма вариантов использования

Гость – это неавторизированный посетитель веб-приложения, который может только просматривать голосования.

Пользователь – авторизованный посетитель веб-приложения, который может просматривать голосования, создать новое голосование и проголосовать, если он участвует в данном голосовании.

Данные актеры могут реализовать следующие варианты использования системы.

1. Гость может подключить кошелек.
2. Гость и пользователь может просмотреть голосование, в котором отображаются название, сроки голосования, варианты ответов и, если голосование завершено, результаты.
3. Пользователь может создать голосование: ввести название, сроки голосования, адреса кошельков и электронные почти избирателей, варианты ответов.
4. Пользователь может проголосовать.

3.4. Компоненты системы

Архитектура децентрализованных приложений изображена на рисунке 4. Данная архитектура состоит из следующих блоков.

1. Блокчейн Ethereum – глобально доступная детерминированная машина состояний, поддерживаемая одноранговой сетью узлов. Смарт-контракты хранятся и работают на блокчейне Ethereum и определяют логику изменений состояния, происходящих в блокчейне. Виртуальная машина Ethereum выполняет выполняет логику, определенную в смарт-контрактах, и обрабатывает изменения состояния.
2. Веб-приложение определяет логику пользовательского интерфейса и взаимодействует с логикой приложения, определенного в смарт-контрактах, с помощью Web3 Провайдера.
3. Web3 Провайдер – узел сети, к которому подключается пользователь для взаимодействия с блокчейном.

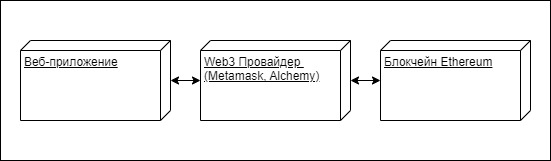


Рисунок 4 – Архитектура децентрализованных приложений

3.4.1. Компоненты смарт-контракта

Для наглядного представления архитектуры смарт-контрактов была построена диаграмма его компонентов (рисунок 5).

В управляющем смарт-контракте хранятся все созданные смарт-контракты для голосования. Программная логика каждого голосования находится в смарт-контракте для голосования.

В голосованиях используется токен, который отвечает за начисление токена избирателю на адрес кошелька и перевод этого токена на адрес кошелька кандидата. Стандарт ERC20 необходим для реализации токена для голосования, в нем содержатся методы по начислению и переводу токенов. Все данные о голосованиях и действия с ними записываются в блокчейн Ethereum.

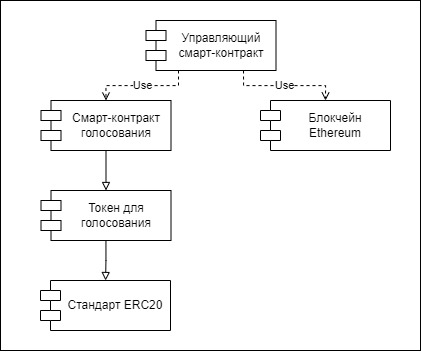


Рисунок 5 – Диаграмма компонентов смарт-контракта

3.4.2. Компоненты веб-приложения

Архитектура веб-приложения состоит из следующих компонентов.

Компонент отображения голосований необходимый для отображения всех голосований на главной странице веб-приложения.

Компонент подключения кошелька необходимый для авторизации пользователя в системе.

Компонент голосования, в котором отображается информация о голосовании и возможность проголосовать.

Компонент создания голосования необходимый для создания голосования.

Данная архитектура изображена на рисунке 6.

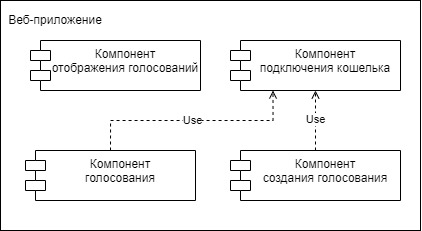


Рисунок 6 – Диаграмма компонентов веб-приложения

3.5. Диаграмма деятельности

В результате анализа требований реализована диаграмма деятельности, которая показывает, как происходит процесс взаимодействия актеров с системой. Рассмотрим прецедент «Создать голосование». Предусловием является то, что пользователь находится в разделе «Создать голосование».

В представленной диаграмме пользователь заполняет поля формы создания голосования: название, сроки голосования, варианты ответов, адреса кошельков и e-mail адреса участников голосования. Затем пользователь нажимает кнопку «Создать голосование». После этого веб-приложение формирует запрос для развертывания смарт-контракта голосования. Веб3 Провайдер выполняет этот запрос и развертывает смарт-контракт в блокчейне. После развертывания смарт-контракт производит начисление токенов-голосов на адреса кошельков избирателей. Веб-приложение отправляет уведомление избирателям о голосовании в виде электронных писем и переводит пользователя на страницу созданного голосования. Данная диаграмма деятельности представлена на рисунке 7.

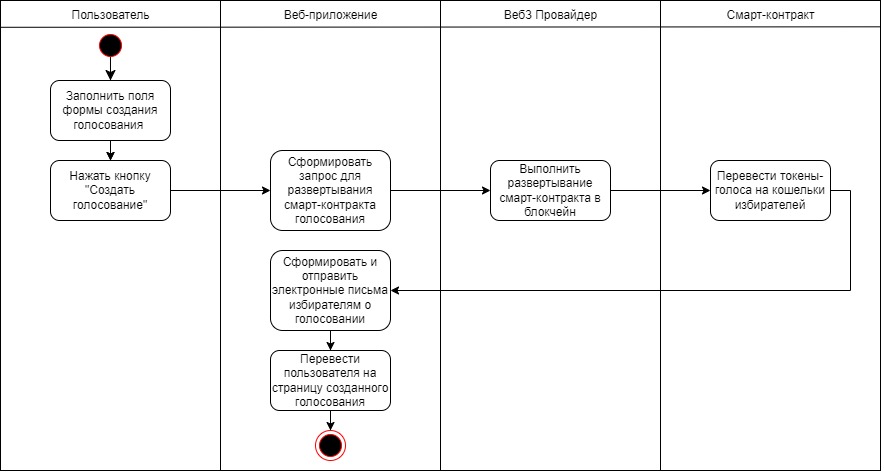


Рисунок 7 – Диаграмма деятельности

3.4. Разработка макетов

Макет – это эскиз, который используется для дальнейшего проектирования и разработки. В макете обычно отмечают положение всех основных элементов на странице. На главной странице приложения отображаются все голосования, а также есть кнопки для подключения кошелька, создания голосования, поиска и сортировки (рисунок 8).

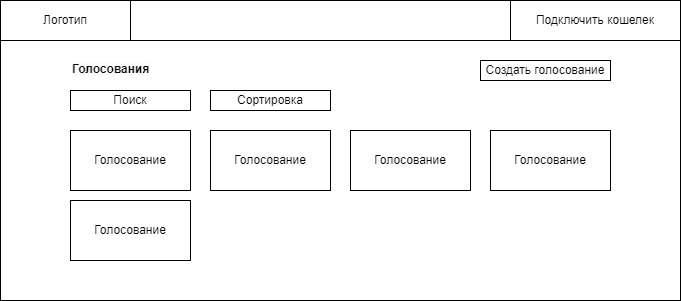


Рисунок 8 – Макет главной страницы

На рисунке 9 изображен макет страницы голосования. На странице голосования отображаются название, данные о голосовании (сроки, ссылка на смарт-контракт голосования в обозревателе блокчейна) и варианты ответов. Также есть кнопка «Проголосовать» для голосования и кнопка «Закрыть» для перехода на главную страницу сайта.



Рисунок 9 – Макет страницы голосования

На странице создания голосования изображены поля для ввода данных о голосовании и кнопка для создания голосования. Кнопка «Закрыть» предназначена для перехода на главную страницу сайта (рисунок 10).

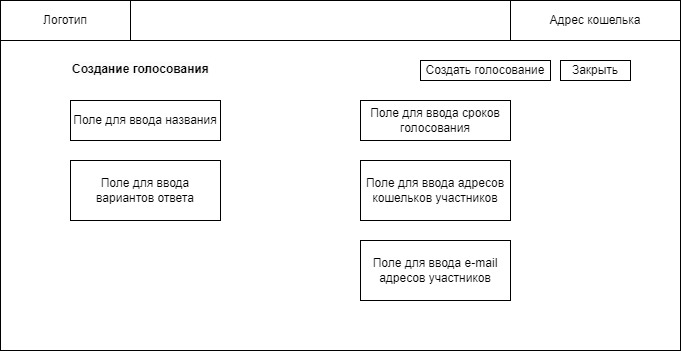


Рисунок 10 – Макет страницы создания голосования

В соответствии с данными макетами был разработан дизайн сайта [7]. Для создания макетов использовался бесплатный онлайн редактор Figma [8] и дизайн-система для создания интерфейсов Material Design [9].

На рисунке 11 представлен фирменный стиль веб-приложения, который включает в себя логотип, цветовую схему и шрифт. В качество основного шрифта на сайте был выбран шрифт Roboto. Цветовая схема сайта была сгенерирована дизайн-системой Material Design.

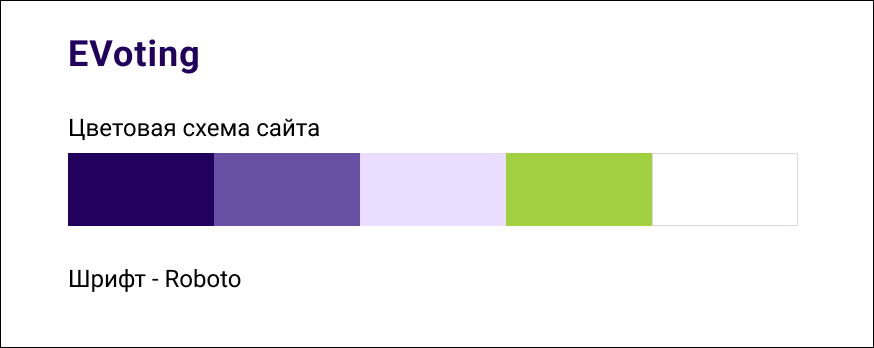


Рисунок 11 – Фирменный стиль веб-приложения

**Выводы по третьей главе**

В процессе анализа требований были определены функциональные и нефункциональные требования. Кроме этого были составлены диаграмма вариантов использования, диаграммы компонентов и диаграмма деятельности системы, макеты и дизайн веб-приложения.

# ЗАКЛЮЧЕНИЕ

В данной работе был спроектировано приложение для электронного голосования на основе технологии блокчейн. В ходе работы были решены следующие задачи.

1. Проведен обзор научной литературы и существующих аналогов.
2. Спроектирована архитектура смарт-контракта для электронного голосования.
3. Реализован смарт-контракт для электронного голосования.
4. Проведено тестирование смарт-контракта.

Планируется дальнейшее развитие проекта, в частности реализация децентрализованного приложения, основанного на смарт-контракте для электронного голосования.

В рамках работы были опубликованы следующие научные статьи.

1. Averin A., Degtyarev V., Bogatyreva V. «Review of E-Voting Systems Based on Blockchain Technology» // International Multi-Conference on Industrial Engineering and Modern technologies (FarEastCon2021), October, 2021.

# ЛИТЕРАТУРА

1. Прасти Н. Блокчейн. Разработка приложений – «СПб.: БВХ-Петербург», 2018.
2. Barnes A., Brake C., Perry T. Digital Voting with use of Blockchain Technology. [Электронный ресурс] URL: https://www.economist.com/sites/default/files/plymouth.pdf (дата обращения: 11.02.2023 г.).
3. Ben Ayed A. A conceptual Secure Blockchain – Based Electronic Voting System // International Journal of Network Security and Its Application (IJNSA), 2017. Vol.9, no. 3. DOI:[10.5121/ijnsa.2017.9301](http://dx.doi.org/10.5121/ijnsa.2017.9301).
4. Boucher P. What if blockchain technology revolutionized voting // European Union, 2016. [Электронный ресурс] URL: https://www.europarl.europa.eu/RegData/etudes/ATAG/2016/581918/EPRS\_ATA(2016)581918\_EN.pdf (дата обращения: 11.02.2023 г.).
5. Trubochkina N., Poliakov S. The Concept Of Electronic Voting Based On Blockchain // INFORMACIONNYE TEHNOLOGII, 25(2), 75–85. 2019. [DOI: 10.17587/it.25.75-85](https://doi.org/10.17587/it.25.75-85).
6. Solidity by Example: Voting. [Электронный ресурс] URL: <https://docs.soliditylang.org/en/v0.8.18/solidity-by-example.html> (дата обращения: 11.02.2023 г.).
7. Abuidris Y., Kumar R., Yang T., Onginjo J. Secure large-scale E-voting system based on blockchain contract using a hybrid consensus model combined with sharding // ETRI Journal, 2020. DOI: 10.4218/etrij.2019-0362.
8. 14-ФЗ «Об обществах с ограниченной ответственностью». [Электронный ресурс] URL: http://pravo.gov.ru/proxy/ips/?docbody=&nd=102051516&intelsearch=08.02.1998+14 (дата обращения: 11.02.2023 г.).
9. WE.Vote – общие собрания в ООО. [Электронный ресурс] URL: <https://we.vote/kogda-primenyaetsya-blockchain-golosovanie/golosovanie-v-ooo> (дата обращения: 11.02.2023 г.).
10. Polys Whitepaper. [Электронный ресурс] URL: https://polysdocs.website.yandexcloud.net/Whitepaper/7262\_WP\_Polys\_Ru\_WEB\_4.pdf (дата обращения: 11.02.2023 г.).
11. Polys. [Электронный ресурс] URL: https://org.polys.me/ (дата обращения: 11.02.2023 г.).
12. Официальный сайт Мэра Москвы. [Электронный ресурс] URL: https://www.mos.ru/ (дата обращения: 11.02.2023 г.).
13. Выборы-2022. [Электронный ресурс] URL: https://www.mos.ru/city/projects/vote2022/ (дата обращения: 11.02.2023 г.).
14. Кибервыборы v1.0: как создавалась система блокчейн-голосования в Москве. [Электронный ресурс] URL: <https://habr.com/ru/article/480152/> (дата обращения: 11.02.2023 г).
15. WE.Vote – Дистанционное электронное голосование на блокчейне. [Электронный ресурс] URL: https://we.vote/ (дата обращения: 11.02.2023 г.).
16. Waves Enterprise. [Электронный ресурс] URL: https://wavesenterprise.com/ (дата обращения: 11.02.2023 г.).
17. Angular. [Электронный ресурс] URL: https://angular.io/ (дата обращения: 11.02.2023 г.).
18. Google. [Электронный ресурс] URL: http://google.com/ (дата обращения: 11.02.2023 г.).
19. HTML. [Электронный ресурс] URL: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/HTML\_basics (дата обращения: 11.02.2023 г.).
20. Vue.js. [Электронный ресурс] URL: https://vuejs.org/ (дата обращения: 11.02.2023 г.).
21. CSS. [Электронный ресурс] URL: https://developer.mozilla.org/ru/docs/Learn/Getting\_started\_with\_the\_web/CSS\_basics (дата обращения: 11.02.2023 г.).
22. React.js. [Электронный ресурс] URL: https://reactjs.org/ (дата обращения: 11.02.2023 г.).
23. Facebook. [Электронный ресурс] URL: https://ru-ru.facebook.com/ (дата обращения: 11.02.2023 г.).
24. JavaScript [Электронный ресурс] URL: https://learn.javascript.ru/ (дата обращения: 11.02.2023 г.).
25. Главная | ethereum.org. [Электронный ресурс] URL: https://ethereum.org/ru/ (дата обращения: 11.02.2023 г.).
26. Создание смарт-контрактов Solidity для блокчейна Ethereum. Практическое руководство / А.В. Фролов – «ЛитРес: Самиздат», 2019.
27. Введение в децентрализованное приложение. [Электронный ресурс] URL: <https://ethereum.org/ru/developers/docs/dapps/#definition-of-a-dapp> (дата обращения: 11.02.2023 г.).
28. Что такое децентрализованные приложения. [Электронный ресурс] URL: https://academy.binance.com/ru/articles/what-are-decentralized-applications-dapps (дата обращения: 11.02.2023 г).
29. Дизайн веб-приложения. [Электронный ресурс] URL: https://www.figma.com/file/VojP3gQ6OhDoXFEXe2RKBe/EVoting?node-id=53097%3A27272&t=9FGhzLfubAYA3St5-1 (дата обращения: 11.02.2023 г).
30. Figma.com. [Электронный ресурс] URL: https://www.figma.com/ (дата обращения: 11.02.2023 г).
31. Material Design. [Электронный ресурс] URL: https://m3.material.io/ (дата обращения: 11.02.2023 г).