Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ БАКАЛАВРА НА ТЕМУ:

Программная система выдачи электронных сертификатов

Студент	ИУ6-82Б		Е.В. Лебедев
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Руководитель			В.В. Гуренко
		(Подпись, дата)	(И.О. Фамилия)
Нормоконтролер			
		(Подпись, дата)	(И.О. Фамилия)

2022 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ___ИУ6___

_____ А.В. Пролетарский

« __ » _____ 2022 г.

З А Д А Н И Е
на выполнение выпускной квалификационной работы бакалавра

Студент группы __ИУ6-82Б_____

_____Лебедев Евгений Викторович_____
(Фамилия, имя, отчество)

Тема квалификационной работы ___Программная система выдачи электронных сертификатов_____
_____

Источник тематики (НИР кафедры, заказ организаций и т.п.)

_____НИР кафедры_____
_____

Тема квалификационной работы утверждена распоряжением по факультету ИУ № 03.02.01-04.03/20 от « _9_ » _декабря_ 2021 г.

Часть 1. Исследовательская

Провести анализ эффективности существующих методов выдачи электронных сертификатов образовательными учреждениями, предоставляющими услуги дополнительного образования. Определить критерии эффективности систем выдачи электронных сертификатов, провести сравнительный анализ существующих систем по заданным критериям, сформулировать требования к разрабатываемой программной системе.

Часть 2. Конструкторская

На основании требований к разрабатываемой системе провести анализ предметной области. Определить технологии, язык и среду разработки. Разработать структуру системы.

Спроектировать и реализовать компоненты системы с использованием выбранных средств разработки. Выполнить комплексное тестирование полученной системы.

Часть 3. Технологическая

Разработать технологию использования программной системы выдачи электронных сертификатов.

Оформление квалификационной работы:

Расчетно-пояснительная записка на 55–65 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

1. Диаграммы вариантов использования – лист А2.
2. Схема структурная информационной системы – лист А1.
3. Диаграммы классов предметной области и интерфейсной части программного, концептуального уровня и уровня реализации – лист А1.
4. Основные схемы алгоритмов программной системы – лист А2.
5. Графы состояний интерфейса – лист А2.
6. Графы диалогов – лист А2.
7. Формы интерфейса – лист А1.
8. Диаграмма компоновки программной системы – лист А2.
9. Таблицы тестов – лист А2.
10. Диаграммы бизнес-процесса использования программной системы – лист А2.

Дата выдачи задания « _01_ » _сентября_ 2021 г.

В соответствии с учебным планом выпускную квалификационную работу выполнить в полном объеме в срок до « 01 » июня 2022 г.

Руководитель квалификационной работы	_____	В.В. Гуренко
	(Подпись, дата)	(И.О. Фамилия)
Студент	_____	Е.В. Лебедев
	(Подпись, дата)	(И.О. Фамилия)

Примечание:1. Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

АННОТАЦИЯ

Данная работа посвящена исследованию и разработке программной системы выдачи электронных сертификатов.

В работе были определены критерии эффективности систем выдачи электронных сертификатов, проведен сравнительный анализ существующих систем по заданным критериям, а также сформулированы требования к разрабатываемой программной системе.

Была выполнена объектная декомпозиция предметной области, проработаны основные алгоритмы функций программы, выполнена программная реализация предметной области, спроектирован интерфейс пользователя, разработана диаграмма классов, осуществлена компоновка программы, проведено тестирование полученной программы, разработана технология использования программной системы.

Помимо этого, были разработаны диаграммы бизнес-процесса использования программной системы. Система предназначена для образовательных организаций, представляющих услуги дополнительного образования, а также для обучающихся в них лиц.

РЕФЕРАТ

Расчетно-пояснительная записка 84 страницы, 31 рисунок, 19 таблиц, 7 источников, 4 приложения.

ЭЛЕКТРОННЫЙ СЕРТИФИКАТ, ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ, БЛОКЧЕЙН, БЛОКЧЕЙН-СЕТЬ, СМАРТ-КОНТРАКТ, ТРАНЗАКЦИЯ, SOLIDITY, ТОКЕН, NFT, ERC-721, IPFS.

Цель работы — программная система, предназначенная для образовательных организаций, представляющих услуги дополнительного образования, осуществляющая выдачу электронных сертификатов обучающимся с использованием технологии блокчейн.

В работе выполнен анализ эффективности существующих методов выдачи электронных сертификатов образовательными учреждениями. Определены критерии эффективности систем выдачи электронных сертификатов. Сделан вывод о целесообразности создания собственной программной системы.

При выполнении данной работы был проведен технический обзор составных компонентов программной системы, выработаны и проанализированы требования к программному продукту, сделан вывод о целесообразности использования блокчейн-технологий для решения существующих проблем, связанных с выдачей электронных сертификатов образовательными организациями.

Разработан и реализован алгоритм создания электронного сертификата как уникального токена в публичной блокчейн-сети. Разработаны смарт-контракты для развертывания программной системы в публичной блокчейн-сети. Также была разработана программная подсистема администрирования, предназначенная для управления системой администрацией образовательной организации, и веб-сайт для обучающихся, предоставляющий функции получения электронного сертификата. Выполнено комплексное тестирование программной системы.

Помимо этого, была разработана технология использования программной системы и определены этапы введения системы в эксплуатацию.

Разработка программной системы велась на языках программирования Python, Solidity, TypeScript с использованием библиотек PyQT5, Brownie, Ethers, Web3, языка разметки HTML, языка таблиц стилей CSS, а также редактора исходного кода VS Code.

СОДЕРЖАНИЕ

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

Блокчейн – выстроенная по определенным правилам непрерывная последовательная цепочка блоков, представляющая собой распределенную базу данных, которая содержит информацию обо всех транзакциях, проведенных участниками системы.

Смарт-контракт – программа для ЭВМ, записанная в распределенном реестре и обеспечивающая автоматическое исполнение договорных обязательств или иных юридически значимых действий.

Токен – единица учета, предназначенная для представления цифрового баланса, представляющая запись в регистре, распределенную в блокчейн-цепочке.

Уникальный токен – вид криптографических токенов, каждый экземпляр которых не может быть обменян или замещен другим токеном.

ERC-721 – открытый стандарт, определяющий правила создания уникальных токенов на блокчейнах, совместимых с виртуальной машиной Ethereum.

NFT (Non-fungible token) – уникальный токен.

IPFS (Interplanetary File System) – децентрализованная распределенная файловая система с контентной адресацией.

EVM (Ethereum Virtual Machine) – виртуальная машина Ethereum.

ВВЕДЕНИЕ

В настоящее время сфера образования нередко сталкивается с проблемами, которые касаются выдачи учащимся документов об окончании обучения. Зачастую эти проблемы связаны с мошенничеством в академической среде, созданием фальсификаций и подделок документов об образовании учащимися. В связи с этим существует и такая проблема как сложность верификации документов об образовании при устройстве на работу. Также бумажный сертификат об обучении достаточно легко можно привести в негодный вид или потерять.

Для преодоления вышеописанных проблем необходимо разработать программную систему получения электронных сертификатов. Данная система должна быть отказоустойчивой и доступной для пользователей в любой момент времени.

Одной из главных особенностей программной системы выдачи электронных сертификатов является то, что ее функционирование осуществляется с использованием технологии блокчейн. Данная технология позволяет обеспечить пожизненную проверяемость цифрового сертификата, невозможность добавления сертификата в смарт-контракт блокчейн-сети лицом, не имеющим на это право, а также мгновенную идентификацию и верификацию электронного сертификата.

1 Анализ применимости технологии блокчейн к задаче формирования и выдачи электронных сертификатов

1.1 Особенности формирования и выдачи электронных сертификатов об обучении

В настоящее время существует большое количество образовательных организаций, предоставляющих услуги дополнительного образования. Открытые и закрытые курсы проводятся повсеместно как в очном, так и в дистанционном формате. Все большее число образовательных учреждений, предоставляющих услуги дополнительного образования, стремится к тому, чтобы проводимые занятия отвечали целям не только самообразования обучающихся: подавляющее большинство этих образовательных организаций нацелено на помощь обучающимся в продвижении по карьерной лестнице или помощь с получением бонуса для поступления в вуз, прохождения собеседования на работу и т.д. Многие компании прямо заявляют, что сертификаты различных онлайн-курсов являются для людей дополнительным преимуществом при трудоустройстве. В связи с этим, как правило, каждому учащемуся хочется получить документ, который выступит в роли подтверждения полученных знаний. Вследствие этого, в сфере образования, со стороны образовательных организаций принято выдавать сертификаты об успешном прохождении обучения. Это стимулирует обучающихся усерднее трудиться, чтобы получить заветный документ.

Процесс сертификации имеет множество преимуществ: он позволяет развиваться межвузовской системе признания сертификатов по образовательным курсам, а также постепенно такие сертификаты начинают признаваться и работодателями. Сертификат об обучении выдается на законных основаниях и констатирует факт участия человека в различных краткосрочных мероприятиях образовательного характера.

В настоящее время подавляющее число образовательных организаций выдает сертификаты об обучении в бумажном или электронном виде. Образовательная организация имеет право на самостоятельное утверждение

образца сертификата о прохождении курсов. Как правило, бумажный сертификат об обучении содержит номер, информацию о наименовании образовательной организации, выдавшей данный документ, названии пройденного курса, а также количестве затраченных часов и полученного количества баллов слушателем.

В свою очередь электронный вид сертификата подразумевает аналог бумажного вида – изображение сертификата, хранящееся на компьютере, в графическом формате. Выдача сертификата таким способом, к сожалению, сопряжена со множеством проблем, которые касаются подтверждения подлинности сертификата. Основная из них – мошенничество в академической среде, связанное с созданием фальсификаций и подделок сертификатов об обучении.

Верификация документов об обучении при устройстве на работу – сложный и длительный процесс, поэтому не все работодатели готовы потратить на это часть своего ценного времени. В связи с этим случаи создания поддельных документов об обучении возрастают с каждым годом. По данным Роскомнадзора, ежегодно блокируются тысячи ресурсов, предлагающих приобрести поддельные сертификаты о прохождении обучения. Особенно уязвимыми становятся электронные версии сертификатов. В условиях сложности процесса проверки, злоумышленники меняют лишь ФИО на изображении чужого сертификата об обучении, надеясь остаться незамеченными.

Еще одной проблемой современной системы выдачи электронных сертификатов можно назвать то, что не существует единого реестра, в котором хранятся все электронные сертификаты об обучении, выдаваемые организациями, предоставляющими услуги дополнительного образования. Как правило, сертификаты в данных организациях находятся в собственных базах данных, что означает их централизованное хранение. В таких условиях, надежность системы верификации выданного сертификата об обучении полностью зависит от надежности конфигурации базы данных

образовательной организации. В случае утери контроля над базой данных или потери информации с базы данных в связи с непредвиденными обстоятельствами или хакерской атакой, процесс подтверждения владения сертификатом об обучении становится невозможным.

Также существуют и проблемы, связанные с выдачей бумажных сертификатов, несмотря на то, что вероятность их подделок существенно ниже, хотя имеется. Бумажный сертификат можно привести в негодный вид, потерять или испортить. Как правило, время, затраченное на восстановление бумажного сертификата, существенно выше в связи с необходимостью проставления печатей и штампов на образце сертификата. Выдача электронных сертификатов экономит время сотрудников, повышает их производительность, снижает затраты на канцелярию, а также является намного более экологичной, чем выдача бумажных аналогов.

В таблице 1 представлена сравнительная характеристика бумажных и электронных сертификатов об обучении.

Таблица 1 – Сравнительная характеристика бумажных и электронных сертификатов

Критерии сравнения	Бумажные сертификаты	Электронные сертификаты
Вероятность фальсификаций	средняя	высокая
Скорость создания большого количества образовательной организацией	низкая	высокая
Централизованное место хранения данных о сертификатах	есть	есть
Время восстановления	высокое	низкое
Затраты предприятия	высокие	низкие
Экологичность	низкая	высокая

Таким образом, в сфере образования существует потребность избавиться от выдачи бумажных сертификатов, однако методы выдачи их электронных аналогов на данный момент не идеальны. Необходимо создать программную систему, способную решить данные проблемы, поддерживая работоспособность и доступность для пользователей.

1.2 Основные принципы применения технологии блокчейн

Винченцо Морабито – автор одной из самых известных книг о блокчейне «Business innovation through blockchain» – определяет технологию как распределенную децентрализованную защищенную шифром базу, публичный депозитарий информации, в котором каждая совершенная транзакция записывается и становится известна всем участникам сети [1]. Любая транзакция в реестре признается действительной, только если ее одобряет более чем половина участников сети. Это означает, что ни один участник системы или агент извне не могут провести валидную транзакцию без согласия других пользователей [5].

Таким образом блокчейн-сеть можно охарактеризовать, как распределенную базу данных, поддерживаемую большим количеством узлов. При этом состояние этой базы данных на каждом из узлов абсолютно идентичное. Под транзакцией в блокчейне понимается совершение любого действия по изменению общего публичного состояния сети [1]. Проведение транзакции – процесс при котором отсутствует доверие между ее участниками. В связи с этим существует необходимость присутствия третьей стороны, которая бы гарантировала правильность ее исполнения. Механизмы работы блокчейна позволяют участникам сети достигать договоренностей о проведении транзакции без посредников. Если более, чем половина блокчейн-сети посчитает транзакцию валидной, транзакция будет включена в блок. При этом, участникам сети невыгодно не подтверждать валидную транзакцию из-за экономической составляющей. Количество подтвержденных транзакций прямо соотносится с их заработком.

Блокчейн как информационный массив можно охарактеризовать следующим образом:

– функционирующая по принципу peer-to-peer децентрализованная распределенная система [1];
– для совершения или проверки транзакции используются криптографические алгоритмы и цифровая подпись;
– внесение изменений в сделанные записи невозможно из-за структуры блока.

Блокчейн-сеть представляет из себя цепочку блоков, реализованную на основе распределенной базы данных. Единственно возможный способ изменения реестра блокчейн-сети – проведение транзакции. Для добавления транзакции в блок необходимо согласие как минимум половины участников сети. Участники проверяют ее формат и подписи и, в случае валидации, происходит запись в блок. Блоки содержат в себе сведения о совершенных транзакциях за определенный период (время формирования блока), криптографический хэш предыдущего блока, хэш совершенных транзакций и другие дополнительные данные, зависящие от конкретной блокчейн-сети. На рисунке 1 представлена упрощенная цепочка блоков в блокчейн-сети.

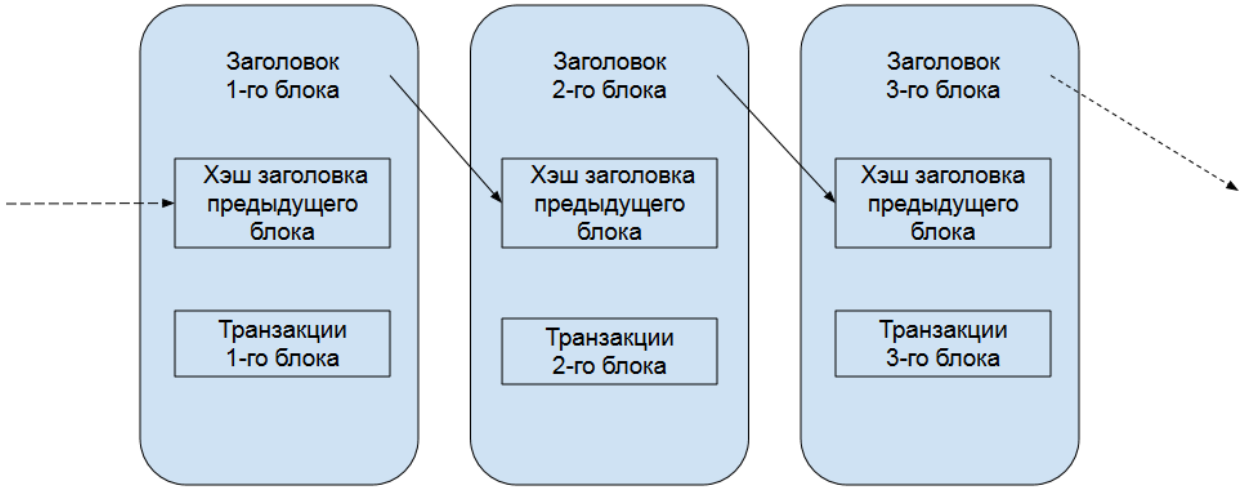

Рисунок 1 – Упрощенная цепочка блоков в блокчейн-сети

Структура блока не позволяет вносить изменения в уже существующие блоки, так как для этого необходимо изменить всю цепочку блокчейна. Такая

связность блоков обеспечивается применением хэш-функций и электронной подписи при совершении транзакций. Устойчивость и надежность блокчейна достигается за счет его децентрализации. Суть децентрализации заключается в том, что каждый участник сети имеет на своем жестком диске полную копию текущего реестра, что делает невозможным его компрометацию [2]. Таким образом, в случае выхода из строя части узлов, сеть продолжит корректно функционировать.

1.2.1 Организация транзакций в блокчейн-сети

Для проведения транзакций в блокчейн-сети необходимо владеть публичным и приватным ключом. Открытый ключ, как правило, является общедоступным и может использоваться для идентификации пользователя. В свою очередь, закрытый ключ должен храниться в секрете у каждого пользователя. Генерация ключей происходит одновременно, при этом ключи связаны между собой математической функцией так, что данные, зашифрованные одним ключом, могут быть расшифрованы другим ключом. Процедура шифрования в асимметричных криптографических алгоритмах обладает свойством необратимости по известному ключу шифрования – это условие является необходимым в системах асимметричной криптографии.

Первым шагом для проведения транзакции является ее создание: отправитель задает блокчейн-адрес получателя транзакции, предмет транзакции, а также подписывает ее с помощью криптографической цифровой подписи. Узлы сети оповещаются о транзакции и проверяют валидность транзакции путем дешифрования электронной подписи. Если транзакция проходит проверку, то она встает в режим ожидания на включение в блок. Один из узлов сети, называемый майнером, раз в несколько минут, упаковывает транзакции из листа ожидания в единый блок и отправляет его на проверку другим узлам сети, которые называются валидаторами. Валидаторы запускают повторяющийся процесс, который требует одобрения от других узлов-операторов для того, чтобы признать блок действительным. В случае одобрения всех транзакций блок присоединяется к цепочке.

Сложность процесса верификации блоков обеспечивает надежность блокчейн-сети. Блокчейн не зависит от централизованной компьютерной архитектуры, в связи с чем неправильное функционирование меньшинства узлов не вредит сети.

В настоящее время наиболее популярными и надежными блокчейн-сетями являются сети, совместимые с EVM (Ethereum Virtual Machine). В данных сетях, для подписания транзакций используется криптографический алгоритм ECDSA. В сети Ethereum, используемой в текущей работе, используется алгоритм асимметричного шифрования ECDSA-secp256k1. Стандарт secp256k1 определяет эллиптическую кривую для формирования открытого и закрытого ключа. Формула эллиптической кривой стандарта secp256k1 имеет вид:

$$y^2 = x^3 + 7 \quad (1)$$

Изображение эллиптической кривой представлено на рисунке 2.

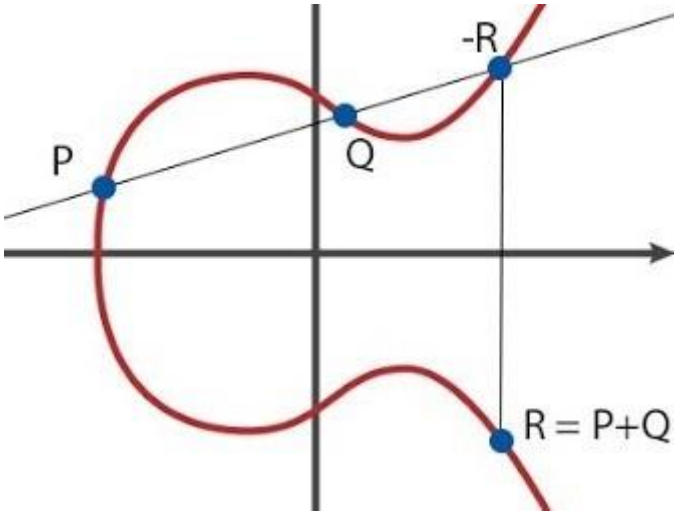

Рисунок 2 – Эллиптическая кривая стандарта secp256k1

Закрытый ключ эллиптической кривой стандарта secp256k1 блокчейн-сети Ethereum представляет собой 32-байтовое число с диапазоном значений от 1 до 2^{256} - 1. Число 0 является специальным закрытым ключом, которое не может быть сгенерировано алгоритмом создания закрытого ключа Ethereum.

Открытый ключ secp256k1 представляет собой 64-байтовое число, где первые 32 байта описывают координату X эллиптической кривой, а вторые 32

байта – координату Y, согласно формуле 1. Открытый ключ вычисляется из закрытого ключа secp256k1 [7].

Цифровые подписи Ethereum используют три параметра – $\{r, s, v\}$, которые могут быть объединены в 65-байтовую последовательность: 32 байта для r, 32 байта для s и 1 байт для v. Параметр v является значением из одного байта, которое описывает значения размера и знака определенной точки эллиптической кривой и может быть равным 27 или 28. Данный параметр используется для восстановления открытого ключа из подписи ECDSA.

Блокчейн адрес сети Ethereum соответствует последним 20 байтам хэша (keccak-256) открытого ключа secp256k1. В связи с детерминированностью хэш-функции keccak-256 для одной пары открытый-закрытый ключ всегда возвращается один блокчейн-адрес сети.

1.2.2 Особенности технологии смарт-контрактов

Блокчейн способен автоматизировать проведение транзакций с помощью фрагментов кода, которые именуются смарт-контрактами. Код смарт-контрактов используется для задания всех условий договора в форме “если-то”. Таким образом, код смарт-контрактов может выступать в качестве правил по изменению состояния блокчейн-сети. В случае исполнения условий, при запуске функции смарт-контракта, код самостоятельно способен исполнить транзакцию, тем самым изменив реестр распределенной базы данных.

Код смарт-контрактов является открытым и публикуется в блокчейн-сети при развертывании. Таким образом, каждый участник сети может удостовериться в прозрачности функционирования смарт-контракта. Помимо этого, смарт-контракт имеет собственное хранилище, которое может изменяться по вызову тех или иных функций.

Структура смарт-контракта подразумевает полное описание всех функций программной системы без возможности их корректировки в дальнейшем. Это означает, что при необходимости изменения функций смарт-контракта, необходимо развертывать новый смарт-контракт в блокчейн-сети.

Создать смарт-контракт может любой участник сети. Для этого достаточно отправить в блокчейн сообщение без адресата, назначив комиссию и указав код программы. Комиссия необходима, так как для загрузки смарт-контракта в блокчейн-сеть, майнеры используют свои ресурсы.

Принцип работы смарт-контракта представлен на рисунке 3.

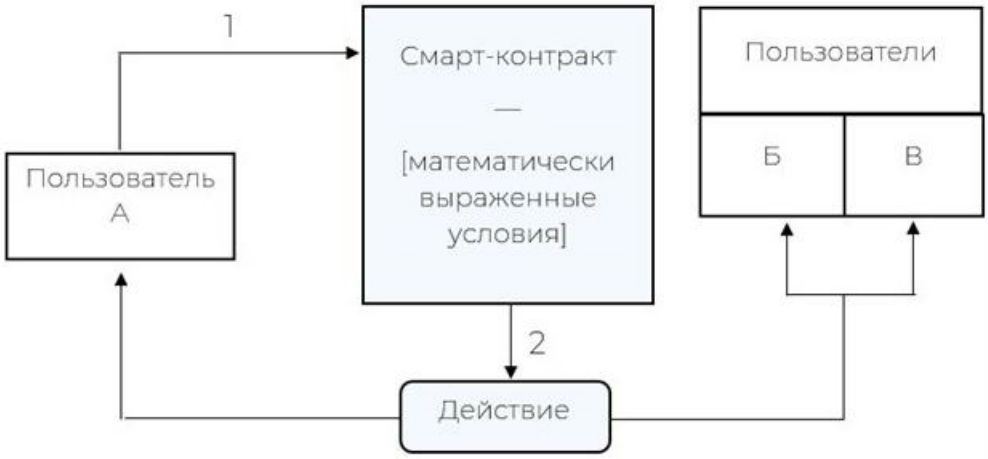

Рисунок 3 – Схема работы смарт-контракта

При инициировании действия со стороны пользователя А, смарт-контракт проверяет математически выраженные условия, хранящиеся в нем, и, в случае их корректности, инициирует действие, которое может быть проверено всеми участниками сети.

В рамках данной работы смарт-контракты могут использоваться как децентрализованное хранилище данных в блокчейн-сети о сертификатах обучающихся в рамках образовательной организации. Данные, которые при помощи смарт-контрактов, будут записаны в блокчейн-сеть невозможно изменить, что позволит обеспечить пожизненную проверяемость электронного сертификата. Помимо этого, в них невозможно “спрятать” никакую информацию, что обеспечит процессу выдачи электронного сертификата полную прозрачность.

1.3 Применение блокчейн-сети к задаче выдачи электронных сертификатов

Так как каждое совершение транзакции в блокчейн-сети обязательно сопровождается выплатой комиссии со стороны отправителя, в зависимости

от нагруженности операции, а объем памяти, предоставляемый смарт-контрактами не бесконечен, блокчейн-сеть нецелесообразно использовать как место фактического хранения электронных сертификатов. Для реализации выдачи электронных сертификатов с использованием технологии блокчейн необходимо использовать блокчейн-сеть лишь в качестве места регистрации электронных сертификатов. Таким образом, целесообразно хранить лишь хэш сертификата в блокчейн-сети, по которому возможно получить данный сертификат из информационного хранилища.

В свою очередь информационное хранилище должно обеспечивать уникальный хэш каждому электронному сертификату, отслеживать дублирование документов и пресекать возможность их изменения. При этом, информационное хранилище должно удовлетворять свойству децентрализации – информация, загружаемая в него, должна храниться и поддерживаться сразу на нескольких узлах.

Таким образом, предполагаемая схема процесса выдачи и верификации электронного сертификата с использованием технологии блокчейн представлена на рисунке 4.

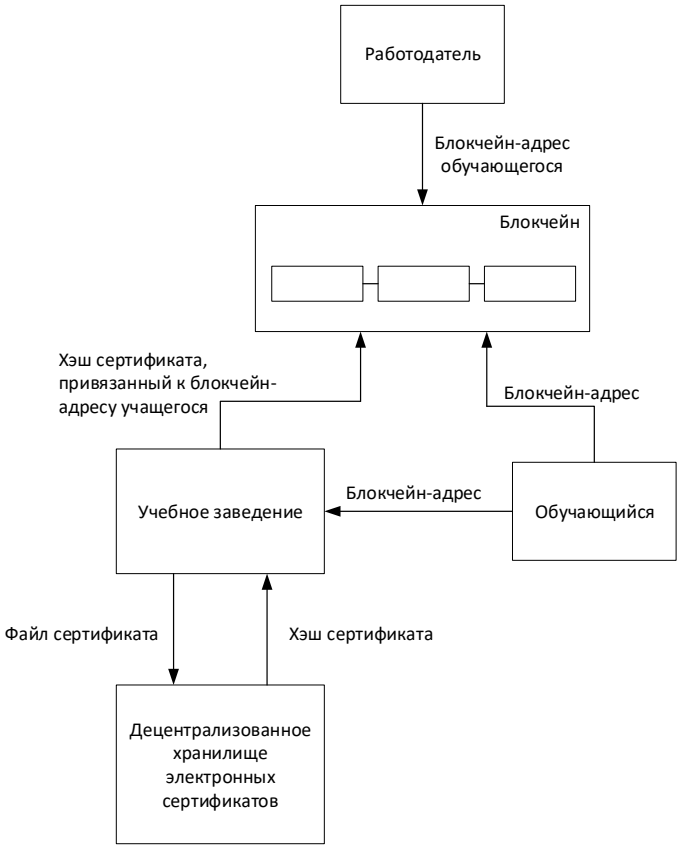

Рисунок 4 – Схема выдачи и верификации электронных сертификатов с использованием технологии блокчейн

В качестве сущности, обеспечивающей добавление хэша сертификата в блокчейн-сеть, может выступать смарт-контракт, развернутый в блокчейн-сети. Помимо этого, смарт-контракт может обеспечить дополнительные структуры для хранения информации об обучающихся в образовательной организации, количества курсов и прочей необходимой информации.

В качестве децентрализованного хранилища электронных сертификатов, удовлетворяющему требованиям, описанным выше, может выступать IPFS-сеть.

1.3.1 Особенности децентрализованной сети IPFS

IPFS - это одноранговая децентрализованная сеть, которая позволяет пользователям создавать резервные копии файлов и веб-сайтов, размещая их на многочисленных узлах [6]. Такой способ размещения файлов гарантирует устойчивость к централизованным точкам отказа – даже при отключении

большинства узлов, файлы, загружаемые в сеть, будут видны всем работающим узлам.

Уникальной особенностью сети IPFS является то, что самостоятельной единицей, передаваемой в сети, является блок. Как правило, блок содержит часть какого-либо файла и ссылки на другие блоки. Цепочка блоков представляет из себя направленный ациклический граф, из которого в дальнейшем собирается файл или целый каталог. Пример структуры блоков в IPFS представлен на рисунке 5.

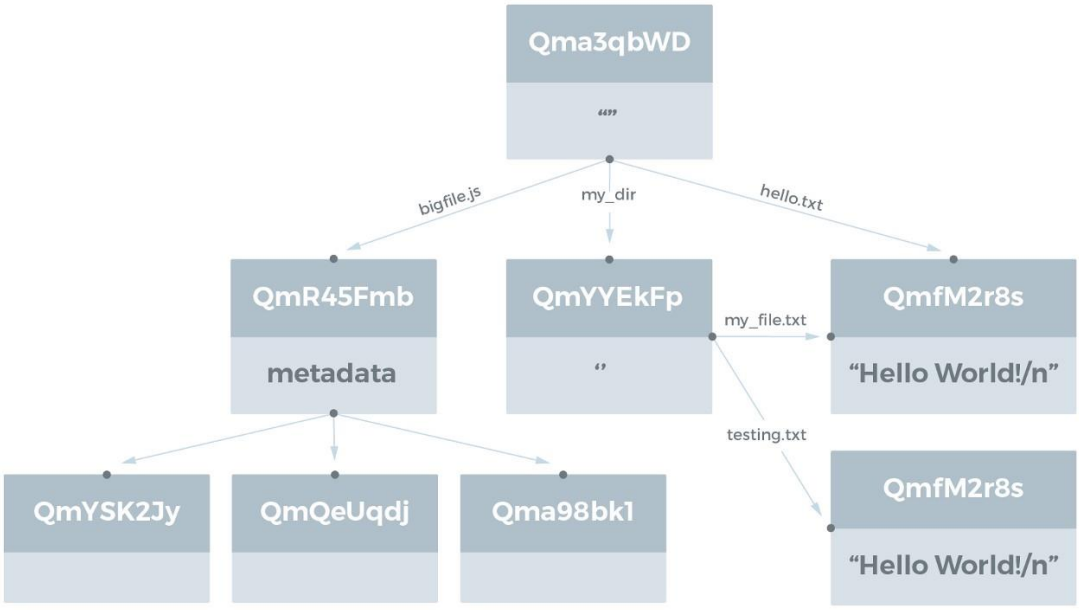

Рисунок 5 – Пример цепочки блоков в IPFS-сети

Идентификатором (адресом) файла в IPFS-сети служит его криптографический мультихэш SHA-128 от блока. Данный мультихэш однозначно определяет адрес файла в сети по его содержимому. Таким образом, при изменении злоумышленником даже одного бита изображения сертификата при загрузке изображения сертификата в IPFS-сеть, будет получен совершенно другой адрес файла.

Используя IPFS-сеть в качестве хранилища изображения сертификатов обучающихся, образовательная организация может однозначно идентифицировать каждый файл сертификата, получив его адрес в IPFS-сети, записать информацию об этом в блокчейн-сеть, сопоставив адрес сертификата

с блокчейн-адресом обучающегося, и, таким образом, обеспечить невозможность подделки электронного сертификата со стороны. Такой вид электронного сертификата в блокчейн-сети представляет из себя уникальный токен (NFT).

1.3.2 Электронный сертификат как уникальный токен

Уникальный токен (NFT) представляет собой уникальный криптографический сертификат цифрового объекта в блокчейн-сети [3]. Такой токен состоит из идентифицирующей информации, которая хранится в смарт-контракте. Существует три основных стандарта токенов в блокчейн-сети Ethereum: ERC-20, ERC-721 и ERC-1155. В таблице 2 представлена сравнительная характеристика стандартов токенов в блокчейн-сети.

Таблица 2 – Сравнение стандартов токенов в блокчейн-сети

Критерий	ERC-20	ERC-721	ERC-1155
Уникальность токенов	Все токены идентичны друг другу	Все токены являются исключительно уникальными	Поддерживаются как уникальные, так и взаимозаменяемые токены
Делимость токена	Делимый, по умолчанию 18 знаков после запятой	Неделимый	Неделимый
Экземпляр токена соотносится с метаданными	Нет	Да	Да
Индексация конкретного экземпляра токена	Нет	Да	Да

Исходя из сравнения стандартов токенов, представленного в таблице 2, можно сказать, что наилучшими выборами для создания электронного сертификата как уникального токена, являются стандарты токенов ERC-721 и ERC-1155. Однако, ERC-1155 имеет избыточный функционал, так как поддерживает создание взаимозаменяемых токенов. Используя протокол ERC-1155, владение двумя одинаковыми токенами в блокчейн-сети разными пользователями возможно, что противоречит решаемой задаче. В связи с этим наиболее разумным решением является использование стандарта создания токенов ERC-721.

ERC-721 - это открытый стандарт, который определяет, как создавать уникальные токены на блокчейнах, совместимых с EVM (виртуальная машина блокчейн-сети Ethereum); он также включает набор правил, упрощающих работу с NFT [3]. Смарт-контракты, реализующие стандарт ERC-721 содержат в себе поля имени и символа токена, поля общего количества токенов в цепочке блоков, а также функции передачи токенов и задания метаданных.

Для создания NFT-токена электронного сертификата необходимо загрузить изображение электронного сертификата в IPFS-сеть. Получив адрес файла сертификата в IPFS-сети, необходимо создать JSON-документ, содержащий поля наименования токена, его описания и адреса из IPFS. Адрес сформированного JSON-документа в IPFS-сети сопоставляется с блокчейн-адресом обучающегося в образовательной организации. Данная запись добавляется в смарт-контракт, что означает создание NFT-сертификата. Сертификат удовлетворяет свойству уникальности в блокчейн-сети, так как криптографический хэш изображения сертификата в IPFS уникален.

1.4 Сравнительный анализ способов выдачи электронных сертификатов

Существующие способы выдачи электронных сертификатов образовательными организациями, предоставляющими услуги дополнительного образования, заключаются в отправке обучающимся изображений сертификатов, хранящихся на компьютере, в графическом

формате, со всеми необходимыми заверяющими подписями и печатями, с указанием дополнительных сведений о пройденном курсе, добавляемых по усмотрению образовательной организации.

В таблице 2 представлена сравнительная характеристика безопасности существующих способов выдачи электронных сертификатов и метода производства электронных сертификатов с использованием технологии блокчейн.

Таблица 2 – Сравнение безопасности способов выдачи электронных сертификатов

Критерий	Существующие методы выдачи электронных сертификатов	Использование технологии блокчейн для выдачи электронных сертификатов
Наличие централизованной базы данных	да	нет
Возможность удаления сертификата пользователя из реестра	да	нет
Возможность добавления фальшивого сертификата об обучении в систему при утрате пароля (приватного ключа)	да	да
Гарантированная пожизненная проверяемость сертификата об обучении	нет	да

Продолжение таблицы 2

Расхожесть сертификатов при замене хотя бы одного пикселя в изображении	нет	да
Затраты по памяти, необходимые для хранения файлов сертификатов	да	нет

Таким образом, по итогам сравнения, способ производства электронных сертификатов с использованием технологии блокчейн является более предпочтительным с точки зрения безопасности по всем критериям. Он исключает наличие централизованной базы данных. При этом каждый произведенный сертификат в блокчейн-сети может быть доступен для любого обучающегося в любой момент и не может быть удален. Добавление несуществующего NFT-сертификата в смарт-контракт возможно лишь при утрате образовательной организацией своего приватного ключа от блокчейн-аккаунта, однако существующие методы выдачи электронных сертификатов также не могут обеспечить защиту при утрате пароля от базы данных.

Помимо этого, такой способ создания электронных сертификатов гарантирует их пожизненную проверяемость, так как данные, попавшие в блокчейн-сеть единожды, невозможно удалить из-за особенностей связи блоков, рассмотренных ранее. Также, изображения сертификатов, которые различаются даже на один пиксель, будут совершенно по-разному идентифицированы при использовании блокчейн-технологии из-за особенностей криптографического хэширования.

Использование блокчейна в связке с IPFS избавляет администрацию образовательной организации от лишних затрат на выделение памяти для хранения файлов сертификатов.

1.5 Выработка требований к программному продукту

Таким образом, технология блокчейн, благодаря своей децентрализованности, репликации базы данных между участниками, неизменяемости данных и сохранению всех транзакций в виде цепочки блоков, позволяет использовать данные положительные качества в программной системе выдачи электронных сертификатов в целях устранения недостатков, существующих на сегодняшний момент систем.

Программная система выдачи электронных сертификатов должна выполнять следующие функции:

– возможность развертывания администратором смарт-контрактов, реализующих логику программной системы в публичной блокчейн-сети;

– возможность редактирования администратором списка блокчейн-адресов обучающихся, добавления новых образовательных курсов, а также метаданных о выданном цифровом сертификате в блокчейн-сети;

– возможность загрузки администратором файла электронного сертификата в IPFS-сеть с целью получения его адреса в данной сети для дальнейшего создания NFT-сертификата по стандарту ERC-721 в публичной блокчейн-сети;

– возможность получения изображения сертификата, выданного образовательным учреждением, из блокчейн-сети, а также всех метаданных о нем со стороны обучающегося;

– обеспечение прозрачности программной системы;

– обеспечение отказоустойчивости программной системы;

– отсутствие возможности вносить изменения в память смарт-контрактов, хранящих информацию о выданных сертификатах, со стороны лиц, не имеющих на это право.

Таким образом, в данной программной системе только у администратора, которым является представитель образовательной организации, должна быть возможность изменения памяти смарт-контракта,

который реализует бизнес-логику программной системы в блокчейн-сети. Функции пользователя должны быть ограничены просмотром текущего состояния смарт-контракта, при этом со стороны администратора не должно быть возможности испортить данные о сертификате обучающегося или изменить их после выдачи электронного сертификата.

Помимо этого, программная система должна быть отказоустойчивой, то есть при выходе из строя устройства с которого осуществляется выдача электронных сертификатов, система должна продолжать работать с другого блокчейн-адреса представителя образовательной организации, т.к. база данных будет реплицироваться на устройства участников сети, в которой запущенно децентрализованное приложение.

Система также должна быть устойчивой ко взлому, то есть у злоумышленника не должно быть возможности выпускать сфальсифицированные сертификаты об обучении.

2 Разработка программной системы выдачи электронных сертификатов

2.1 Анализ технического задания, выбор технологии и средств разработки

Предметной областью реализуемого приложения являются образовательные документы, выдающиеся по окончании обучения, которые претерпевают изменения с целью перевода их в цифровой вид и добавления в блокчейн-сеть. К образовательным документам относятся файлы сертификатов об обучении на различных курсах конкретной образовательной организации.

Согласно требованиям технического задания, необходимо реализовать программную систему, которая будет осуществлять развертывание смарт-контрактов, содержащих всю информацию о выдаваемых обучающимся сертификатах, в публичной блокчейн-сети, из которой любой обучающийся сможет получить изображение полученного сертификата, а также все метаданные, связанные с ним. Помимо этого, необходимо предусмотреть возможность администрирования системы со стороны образовательной организации.

Переходя непосредственно к проектированию, на начальных этапах необходимо принять принципиальные решения, определяющие процесс проектирования, качество и трудоемкость разработки. Одним из главных решений является выбор подхода к разработке (структурного или объектного).

Был выбран объектный подход к разработке, так как вся предметная область представляет собой совокупность объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Разработка интерфейса программы также полностью относится к объектному подходу. Это обусловлено тем, что использование объектного подхода при разработке интерфейса программы позволяет создавать

расширяемые продукты, которые в будущем можно заставить работать с новыми компонентами без внесения в них каких-либо изменений.

Архитектура программного обеспечения может быть однопользовательской и многопользовательской. В данном случае, так как приложение рассчитано на обучающихся в конкретной образовательной организации, желающих получить сертификат цифрового образца, была выбрана многопользовательская модель архитектуры программы.

В силу выбора объектного подхода, наиболее подходящим вариантом похода к разработке станет нисходящий подход. В первую очередь будет спроектирован и реализован пользовательский интерфейс программного обеспечения, затем будут разработаны классы некоторых базовых объектов предметной области, а уже потом, используя эти объекты, будет спроектированы и реализованы остальные компоненты.

В качестве языка разработки смарт-контрактов, реализующих логику программной системы в блокчейн-сети был выбран Solidity, так как он является наиболее распространенным в сфере блокчейн-разработки, а также исполняется на виртуальной машине Ethereum (EVM), что наилучшим образом подходит для написания смарт-контрактов под блокчейн Ethereum.

В качестве языка разработки для программной системы администрирования образовательной организации был выбран Python, так как он является широко распространенным, а также имеет большое количество библиотек для связи подсистемы с блокчейн-сетью.

Для написания веб-сайта для обучающихся, который позволил бы им получить свой цифровой сертификат, основным языком разработки был выбран TypeScript, так как данный язык является обратно совместимым с JavaScript, который повсеместно используется при разработке веб-сайтов. Главными преимуществами TypeScript являются возможность явного статического назначения типов и поддержка использования полноценных классов. Статическое назначение типов упростит связь веб-сайта с блокчейн-

сетью. Знания языков HTML и CSS также были необходимы для реализации верстки веб-сайта.

Для разработки форм программной системы администрирования образовательной организации был выбран продукт Qt Designer, который базируется на библиотеке Qt. Для разработки интерфейса данной системы на языке Python была выбрана библиотека PyQT5, которая отлично взаимодействует с формами, полученными из QT Designer.

В качестве среды разработки был выбран редактор кода VS Code, который позволяет подключение расширений для выбранных языков программирования, что сделает более удобным процесс написания кода.

2.2 Разработка диаграммы вариантов использования

При анализе требований к функциональности, указанных в техническом задании, было выявлено: действующих лиц для данной программы двое – администратор (представитель образовательной организации), который имеет возможность выполнять все администрирующие функции программной системы; пользователь (обучающийся), которому доступны функции, непосредственно связанные с получением своего сертификата.

Варианты использования для администратора приведены в таблицах 3-10.

Таблица 3 – Описание для варианта использования “Получение справки”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку "Помощь”.	2. Открывается форма получения справки по работе программы, в которой подробно описаны правила пользования и информация о работе всех функций программы.

Таблица 4 – Описание для варианта использования “Описание образовательной организации”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку "Развертывание смарт-контрактов в блокчейн-сети”. 3. Пользователь ввел наименование и аббревиатуру образовательной организации.	2. Открывается форма развертывания смарт-контрактов в блокчейн сети с полями для ввода, в которых необходимо дать описание образовательной организации. 4. Поля ввода отображаются как корректные, так как не содержат более 40 и 10 символов соответственно, а также специальных символов.

Альтернатива:

4. Под полями ввода появляется сообщение “Проверьте правильность введенных вами данных”.

Таблица 5 – Описание для варианта использования “Развертывание контрактов в блокчейн-сети”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку "Развертывание смарт-контрактов в блокчейн-сети”.	2. Открывается форма развертывания смарт-контрактов в блокчейн сети с полями для ввода, в которых необходимо дать описание образовательной организации.

Продолжение таблицы 5

3. Пользователь ввел наименование и аббревиатуру образовательной организации и нажал на кнопку “Развернуть смарт-контракты”.	4. Появляется сообщение об успешном развертывании смарт-контрактов в блокчейн-сети. На экран отображаются адреса развернутых смарт-контрактов и количество потраченного газа в блокчейн-сети.

Альтернатива:

4. На экран выводится сообщение “Проверьте правильность введенных вами данных”.

4. На экран выводится сообщение о необходимости пополнения блокчейн-аккаунта нативной валютой блокчейн-сети для развертывания.

Таблица 6 – Описание для варианта использования “Редактирование списка учащихся”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку “Добавить/удалить обучающегося”.	2. Открывается форма редактирования списка обучающихся в образовательной организации.
3. Пользователь вводит наименование и дату начала курса, а также блокчейн-адрес обучающегося. После этого нажимает кнопку “Добавить” или “Удалить”.	4. Появляется сообщение о добавлении/удалении обучающегося, а также хэш успешной транзакции в блокчейн-сети.

Альтернатива:

4. На экран выводится сообщение об ошибке, связанной с неправильным вводом.

4. Если пользователь уже добавлен на данный курс, на экран выводится сообщение с предупреждением.

4. Если пользователя не существовало на данном курсе, на экран выводится сообщение с предупреждением.

4. На экран выводится сообщение о необходимости пополнения блокчейн-аккаунта нативной валютой блокчейн-сети для совершения транзакции.

Таблица 7 – Описание для варианта использования “Добавление метаданных о сертификате”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку “Создать описание сертификата”. 3. Пользователь вводит наименование курса, блокчейн-адрес обучающегося, дату начала курса, дату выдачи сертификата, полученное количество баллов, а также дополнительную информацию. После этого нажимает кнопку “Добавить описание в блокчейн-сеть”.	2. Открывается форма добавления описания сертификата. 4. Появляется сообщение о добавлении описания сертификата, а также хэш успешной транзакции в блокчейн-сети.

Альтернатива:

4. На экран выводится сообщение об ошибке, связанной с неправильным вводом.

4. Если пользователю уже выдан сертификат по данному курсу, на экран выводится сообщение с предупреждением.

4. Если пользователя не существовало на данном курсе, на экран выводится сообщение с предупреждением.

4. На экран выводится сообщение о необходимости пополнения блокчейн-аккаунта нативной валютой блокчейн-сети для совершения транзакции.

Таблица 8 – Описание для варианта использования “Загрузка сертификата в IPFS”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку “Добавить сертификат в IPFS”. 3. Пользователь нажал на кнопку “Выбрать изображение сертификата”. 5. Пользователь выбрал файл для загрузки и нажал “Открыть”. 7. Пользователь нажал на кнопку “Добавить выбранный сертификат в IPFS”.	2. Открывается форма добавления сертификата в IPFS. 4. Открывается окно выбора файла изображения сертификата с компьютера. 6. Отображается форма добавления сертификата в IPFS. 8. Появляется сообщение о добавлении сертификата в IPFS. В поле “Адрес добавленного сертификата в IPFS” появляется IPFS-хэш сертификата.

Альтернатива:

6. На экран выводится сообщение об ошибке, связанной с неразрешенным форматом выбранного файла.

8. Если такой сертификат уже добавлен в IPFS, на экран выводится предупреждение о коллизии IPFS-хэшей сертификатов.

Таблица 9 – Описание для варианта использования “Создание ERC-721 (NFT) токена сертификата”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку “Создать NFT-сертификат”. 3. Пользователь вводит наименование курса, блокчейн-адрес обучающегося и IPFS-хэш сертификата. После этого нажимает кнопку “Создать NFT-сертификат”.	2. Открывается форма создания ERC-721 токена сертификата. 4. Появляется сообщение о создании ERC-721 токена сертификата, а также список хэшей успешных транзакций в блокчейн-сети.

Альтернатива:

4. На экран выводится сообщение об ошибке, связанной с неправильным вводом.

4. На экран выводится сообщение об ошибке, если сертификат с данным IPFS-хэшем уже существует в памяти смарт-контракта в блокчейн-сети.

4. Если пользователю уже был выдан сертификат по данному курсу, выводится сообщение с предупреждением.

4. На экран выводится сообщение о необходимости пополнения блокчейн-аккаунта нативной валютой блокчейн-сети для совершения транзакции.

Таблица 10 – Описание для варианта использования “Добавление обучающего курса”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку “Создать новый курс”. 3. Пользователь вводит наименование курса, количество академических часов и дату начала курса. После этого нажимает кнопку “Создать образовательный курс”.	2. Открывается форма создания нового обучающего курса. 4. Появляется сообщение о создании образовательного курса, а также хэш успешной транзакции в блокчейн-сети.

Альтернатива:

4. На экран выводится сообщение об ошибке, связанной с неправильным вводом.

4. На экран выводится сообщение об ошибке, если курс с данным наименованием и датой начала уже был создан.

4. На экран выводится сообщение о необходимости пополнения блокчейн-аккаунта нативной валютой блокчейн-сети для совершения транзакции.

Варианты использования для обучающегося в образовательной организации описаны в таблицах 11-15.

Таблица 11 – Описание для варианта использования “Аутентификация в блокчейн сети”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь нажал на кнопку “Войти с Metamask”.	2. Открывается расширение Metamask в браузере с полем для ввода пароля от блокчейн-аккаунта.

Продолжение таблицы 11

3. Пользователь вводит пароль и нажимает кнопку “Разблокировать”.	4. Открывается главная страница веб-сайта.

Альтернатива:

4. На экран выводится сообщение об ошибке, связанной с неправильным вводом пароля.

Таблица 12 – Описание для варианта использования “Получение изображения сертификата”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь ввел наименование курса и нажал на кнопку “Загрузить сертификат”. 3. Пользователь нажимает кнопку “Открыть сертификат”.	2. Появляется форма успешного получения изображения сертификата с указанием его уникального номера. 4. Открывается вкладка с изображением полученного сертификата.

Альтернатива:

2. На экран выводится сообщение о том, что у данного пользователя нет сертификатов на текущем курсе.

Таблица 13 – Описание для варианта использования “Скачивание изображения сертификата на компьютер”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь ввел наименование курса и нажал на кнопку “Загрузить сертификат”.	2. Появляется форма успешного получения изображения сертификата с указанием его уникального номера.
3. Пользователь нажимает кнопку “Открыть сертификат”.	4. Открывается вкладка с изображением полученного сертификата.
5. Пользователь нажимает кнопку “Скачать”.	6. Сертификат скачивается на компьютер пользователя.

Альтернатива:

2. На экран выводится сообщение о том, что у данного пользователя нет сертификатов на текущем курсе.

Таблица 14 – Описание для варианта использования “Получение метаданных о NFT-токене сертификата”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь ввел наименование курса и нажал на кнопку “Загрузить информацию о NFT-токене”.	2. Появляется форма успешного получения данных о NFT-токене сертификата, содержащая адрес смарт-контракта в блокчейн сети, наименование блокчейн-сети и уникальный идентификатор ERC-721 токена.

Альтернатива:

2. На экран выводится сообщение о том, что у данного пользователя нет сертификатов на текущем курсе.

Таблица 15 – Описание для варианта использования “Получение метаданных о выданном сертификате об обучении”

Типичный ход событий	
Действие пользователя	Отклик системы
1. Пользователь ввел наименование курса и нажал на кнопку “Загрузить информацию о пройденном курсе”.	2. Появляется форма успешного получения данных о полученном сертификате на курсе. Отображается наименование курса, блокчейн-адрес обучающегося, дата начала курса, дата выдачи сертификата, полученное количество баллов, а также дополнительная информация, добавленная по усмотрению образовательной организации.

Альтернатива:

2. На экран выводится сообщение о том, что у данного пользователя нет сертификатов на текущем курсе.

Данные варианты использования, отраженные в таблицах 3-15, показаны на диаграммах (для администратора и обучающегося), представленных на рисунках 6, 7.

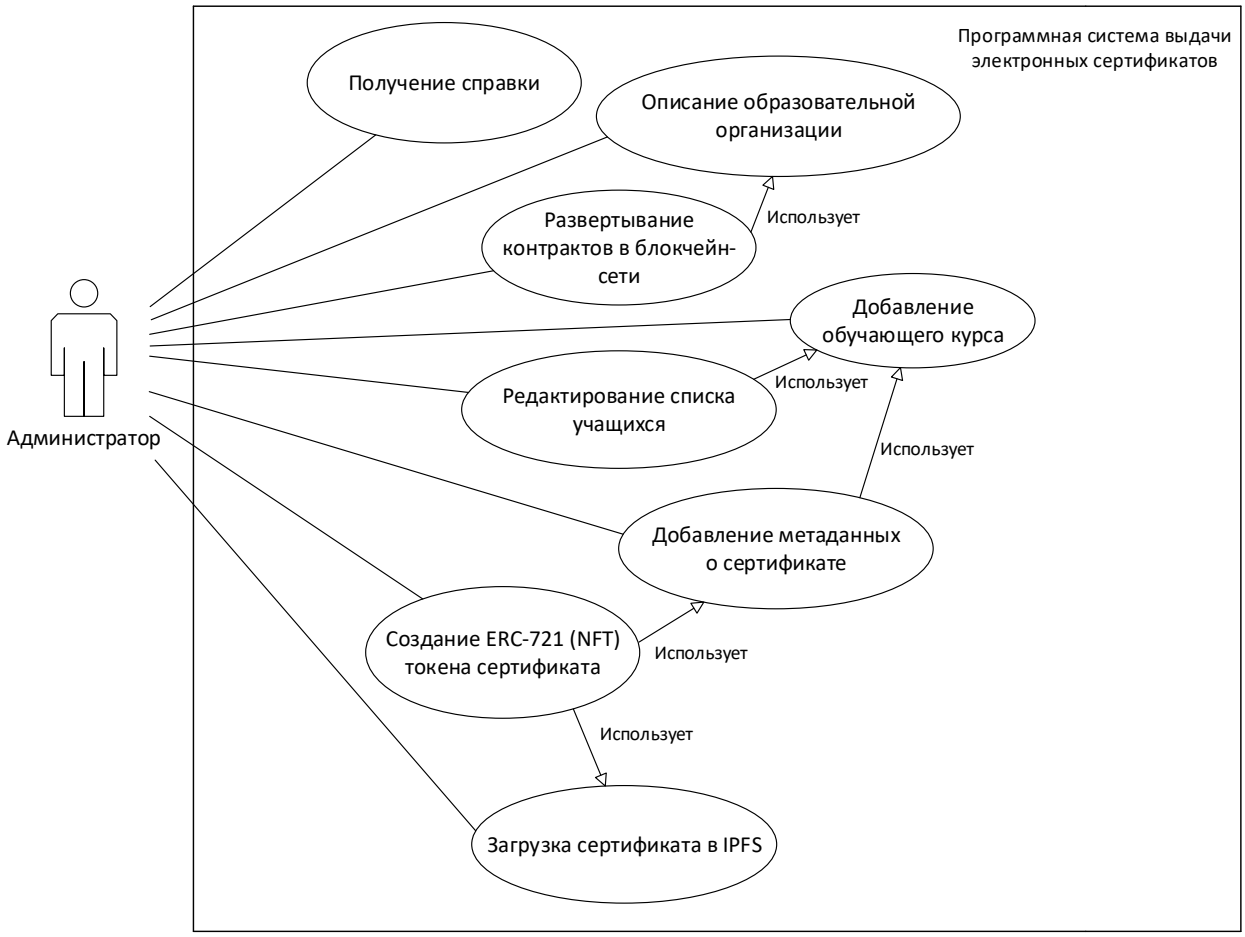

Рисунок 6 – Диаграмма вариантов использования для администратора

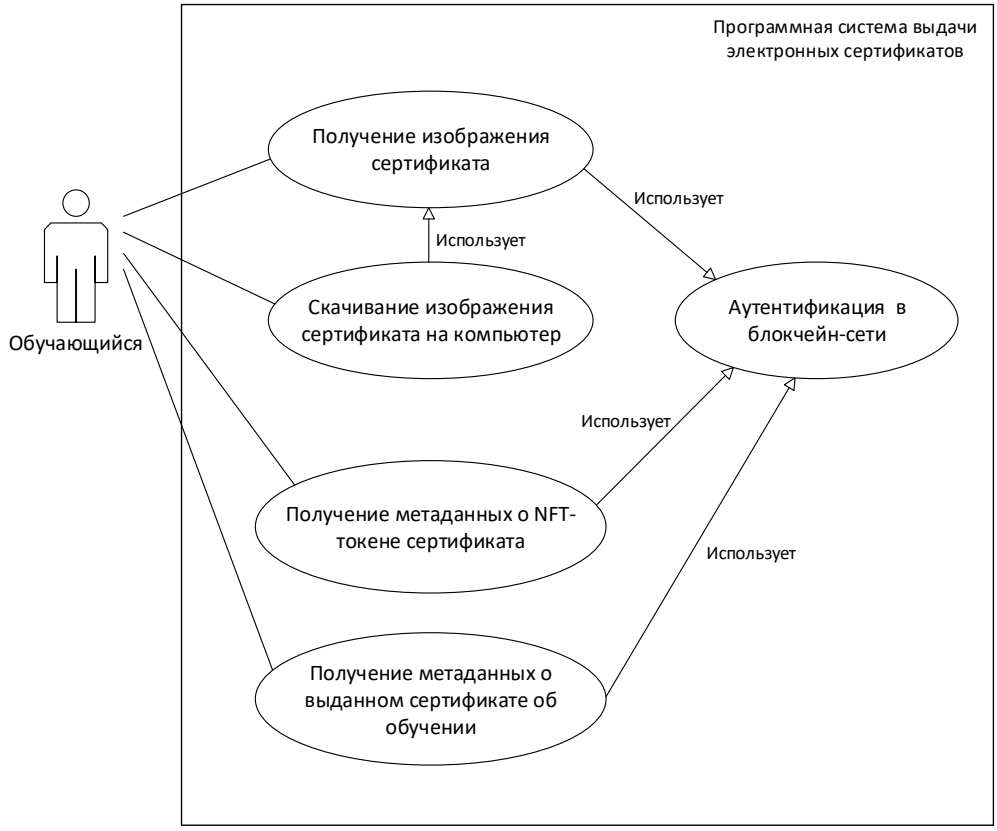

Рисунок 7 – Диаграмма вариантов использования для обучающегося

Данные диаграммы вариантов использования (см. рисунок 6, 7) представляют исходную концептуальную модель системы и отображают функциональные требования к программной системе.

2.3 Разработка структурной схемы программного продукта

Программная система выдачи электронных сертификатов подразделяется на программную подсистему администрирования, предназначенную для образовательной организации, а также на веб-сайт, который предназначен для обучающихся в данной образовательной организации.

Такое разделение наиболее целесообразно в связи с тем, что для программной подсистемы администрирования, которая представляет из себя десктопное приложение, важна надежность, простота в обслуживании и безопасность. Администрации образовательной организации намного легче установить данное приложение единожды и потом постоянно использовать его для создания цифровых сертификатов, нежели делать это через веб-сайт, который будет более уязвим с точки зрения безопасности, если добавить в него функции изменения состояния смарт-контрактов в блокчейн-сети. В свою очередь взлом программной подсистемы администрирования практически невозможен (только в случае утери пароля от блокчейн-адреса), если выполнять меры предосторожности при работе с системой.

Наличие веб-сайта для обучающихся в образовательной организации необходимо в связи с тем, что обучающиеся могут пользоваться программной системой однократно, например, для доказательства кому-либо владения сертификатом. В таком случае, им не придется скачивать приложение на свое устройство. Требования по безопасности функционирования веб-сайта также выполняются, так как взлом веб-сайта для обучающихся невозможен в связи с тем, что из-за разделения, веб-сайт не имеет функций для изменения состояния смарт-контрактов в блокчейн-сети. Таким образом, веб-сайт способен лишь отображать состояние блокчейн-сети и читать данные из нее, которые

добавляют представители образовательной организации через свою подсистему.

Программная подсистема администрирования со стороны образовательной организации включает в себя: модуль формирования ERC-721 токенов сертификатов обучающихся, модуль развертывания смарт-контрактов в блокчейн-сети, модуль администрирования, а также модуль хранения данных об обучающихся в блокчейн-сети.

Подсистема формирования ERC-721 токенов сертификатов обучающихся обеспечивает добавление в смарт-контракт, находящийся в блокчейн-сети, данных о формируемых NFT-токенах, а также привязку NFT-токена к адресу сертификата в IPFS. Подсистема развертывания смарт-контрактов в блокчейн-сети обеспечивает загрузку смарт-контракта NFT-токена и смарт-контракта функций образовательной организации в публичную блокчейн-сеть. Подсистема администрирования предоставляет администрации образовательной организации все необходимые функции (добавление метаданных о сертификатах, добавление новых курсов, загрузка файла сертификата в IPFS-сеть). Подсистема хранения данных об обучающихся в блокчейн-сети обеспечивает добавление в смарт-контракт информации о блокчейн-адресе обучающегося и его сертификате.

В свою очередь веб-сайт, предназначенный для обучающихся в образовательной организации, включает в себя подсистему получения электронного сертификата и данных о нем, а также подсистему связи с блокчейн-сетью. Подсистема получения электронного сертификата и метаданных о нем представляет функции получения изображения сертификата, метаданных, заданных администрацией образовательной организации, а также метаданных о NFT-токене сертификата Подсистема связи веб-сайта с блокчейн-сетью представляет механизмы аутентификации с помощью адреса в публичной блокчейн-сети.

После проведения декомпозиции предметной области была составлена структурная схема программного продукта, представленная на рисунке 8.

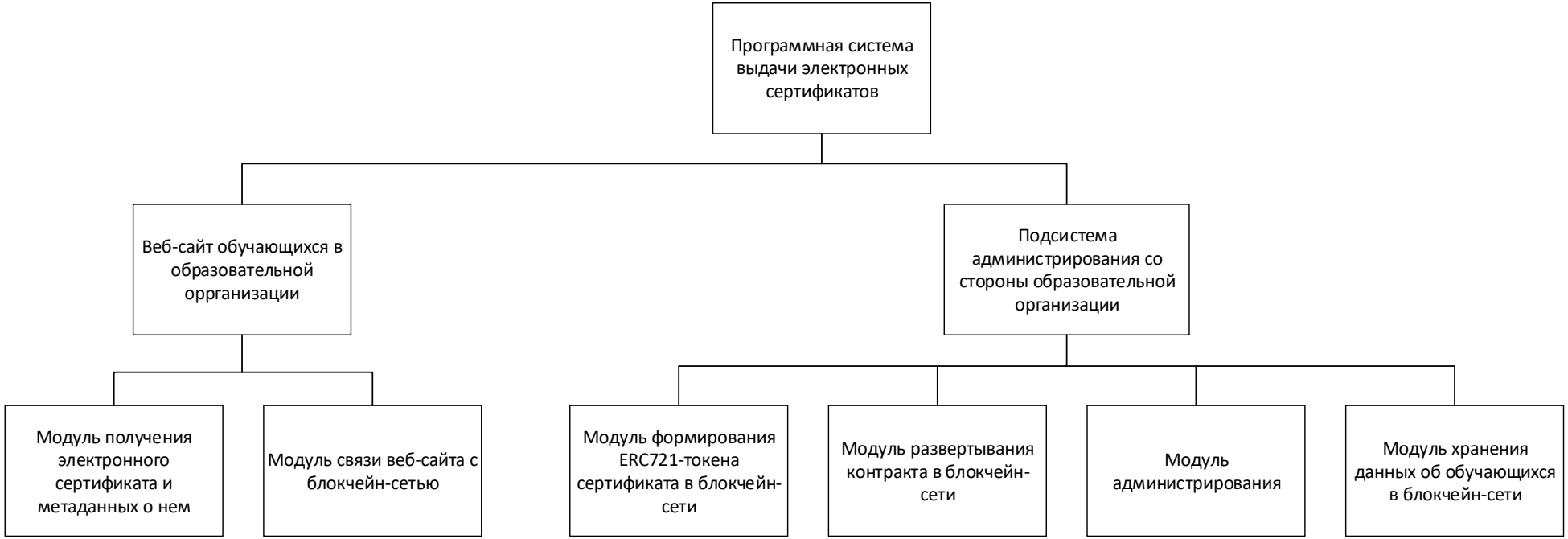

Рисунок 8 – Схема структурная программного продукта

На данной структурной схеме (см. рисунок 18) изображены все основные компоненты программной системы выдачи электронных сертификатов.

2.4 Разработка интерфейса пользователя

Учитывая функционал программы, особенно важно создать интуитивно понятный интерфейс, который позволит пользователю быстро выполнить требуемую задачу. Предполагается, что программа для администратора будет использоваться образовательной организацией на постоянной основе для выпуска новых цифровых сертификатов, поэтому создание благоприятного и дружелюбного интерфейса является одной из главных задач разработки. Для реализации интерфейса была выбрана библиотека PyQT5 языка Python, так как ее использование позволяет быстро создать визуальные представления форм в приложении QT Designer, после чего конвертировать код формы в Python-файл, который можно импортировать и использовать непосредственно в Python [4].

Для реализации интерфейса веб-сайта обучающихся в образовательной организации, помимо стандартных средств языков HTML и CSS, использовалась библиотека React, так как в ней удобно реализована система компонентов — повторяющихся частей кода, которые используются в разных условиях и обстоятельствах и меняются в зависимости от контекста.

При проектировании диалога был использован тип «управляемый пользователем», так как выполнение текущей функции напрямую зависит от выбора пользователя, т.е. не предусматривается строгого сценария выполнения программы.

В результате анализа возможных вариантов использования было принято решение использовать табличную форму диалога с пользователем, так как множество возможных ответов программной системы конечно.

2.4.1 Построение графа состояний интерфейса

Программная подсистема администрирования образовательной организации имеет 10 состояний интерфейса: главное меню, меню функций программы, окно справки по работе программы, окно редактирования блокчейн-адресов обучающихся, окно добавления наименования образовательной

организации, окно развертывания смарт-контракта в публичной блокчейн-сети, окно добавления метаданных о сертификате в блокчейн-сеть, окно добавления файла сертификата в IPFS-сеть, окно добавления нового обучающего курса и окно создания NFT-токена (ERC-721) сертификата в публичной блокчейн-сети.

Для изображения всевозможных изменений состояний интерфейса программной подсистемы администрирования со стороны образовательной организации был построен граф состояний интерфейса, представленный на рисунке 9, на котором показан отклик программы на нажатие соответствующих кнопок.

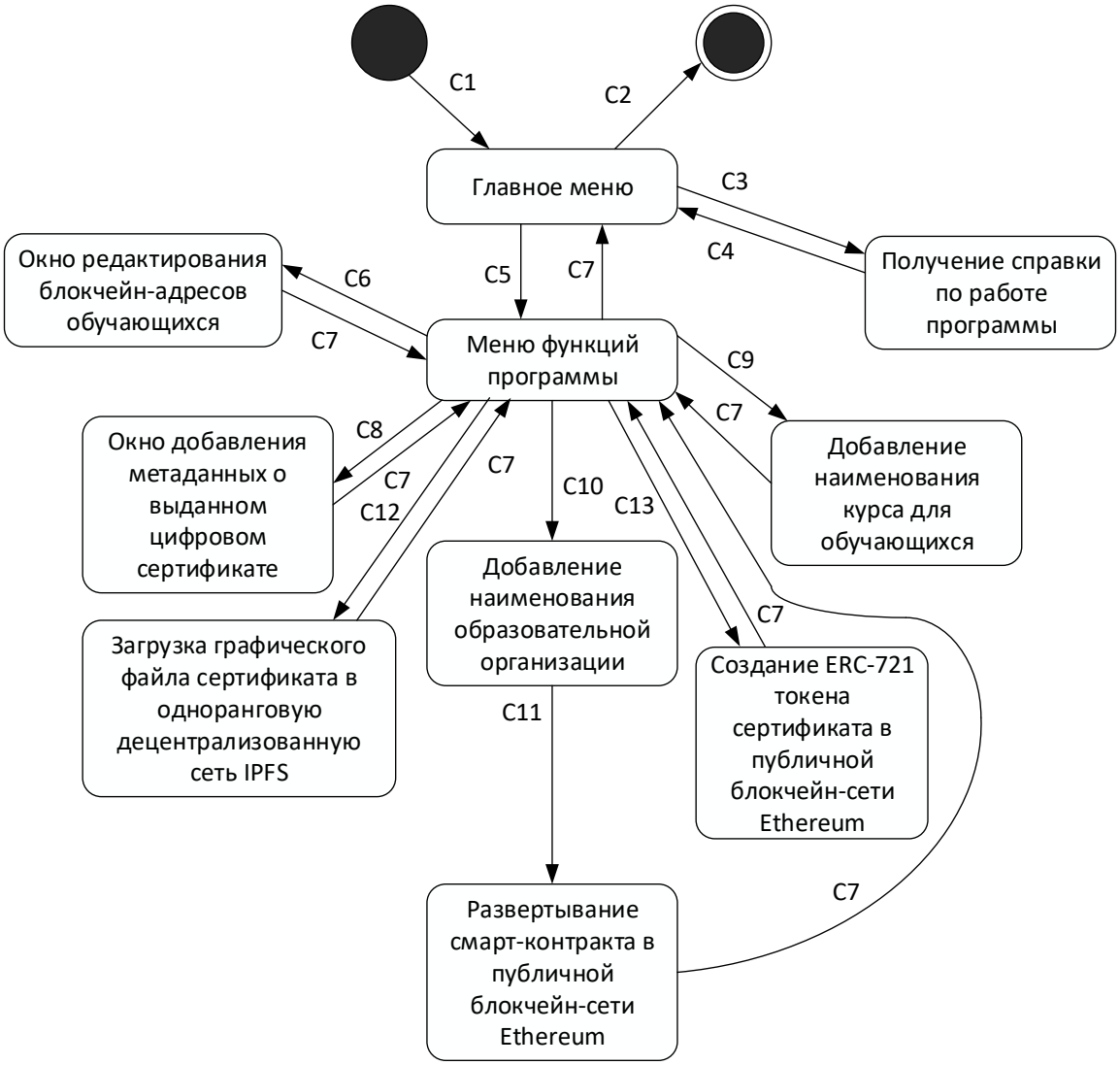

Рисунок 9 – Граф состояний интерфейса программной подсистемы администрирования со стороны образовательной организации

На графе отмечены следующие переходы между состояниями:

С1 – Запуск программы;

С2 – Нажатие кнопки закрытия окна главного меню;

С3 – Нажатие кнопки “Помощь”;

С4 – Возврат в главное меню при нажатии кнопки “Назад”;

С5 – Нажатие кнопки “Меню функций программы”;

С6 – Нажатие кнопки “Добавить/удалить обучающегося”;

С7 – Нажатие кнопки закрытия текущего окна;

С8 – Нажатие кнопки “Создать описание сертификата”;

С9 – Нажатие кнопки “Создать новый курс”;

С10 – Нажатие кнопки “Развертывание смарт-контрактов в блокчейн-сети”;

С11 – Нажатие кнопки “Развертывание смарт-контрактов в блокчейн-сети”;

С12 – Нажатие кнопки “Добавить сертификат в IPFS";

С13 – Нажатие кнопки “Создать NFT-сертификат".

В свою очередь веб-сайт обучающихся в образовательной организации имеет 6 состояний интерфейса: страница аутентификации с помощью блокчейн-адреса посредством расширения браузера Metamask, главная страница веб-сайта, страница получения изображения сертификата, страница загрузки информации о полученном сертификате, страница получения информации о NFT-токене полученного сертификата и страница скачивания изображения сертификата на компьютер.

Как и в случае программной подсистемы администрирования образовательной организации для изображения всевозможных изменений состояний интерфейса веб-сайта обучающихся был построен граф состояний интерфейса, представленный на рисунке 10, на котором показан отклик программы на нажатие соответствующих кнопок.

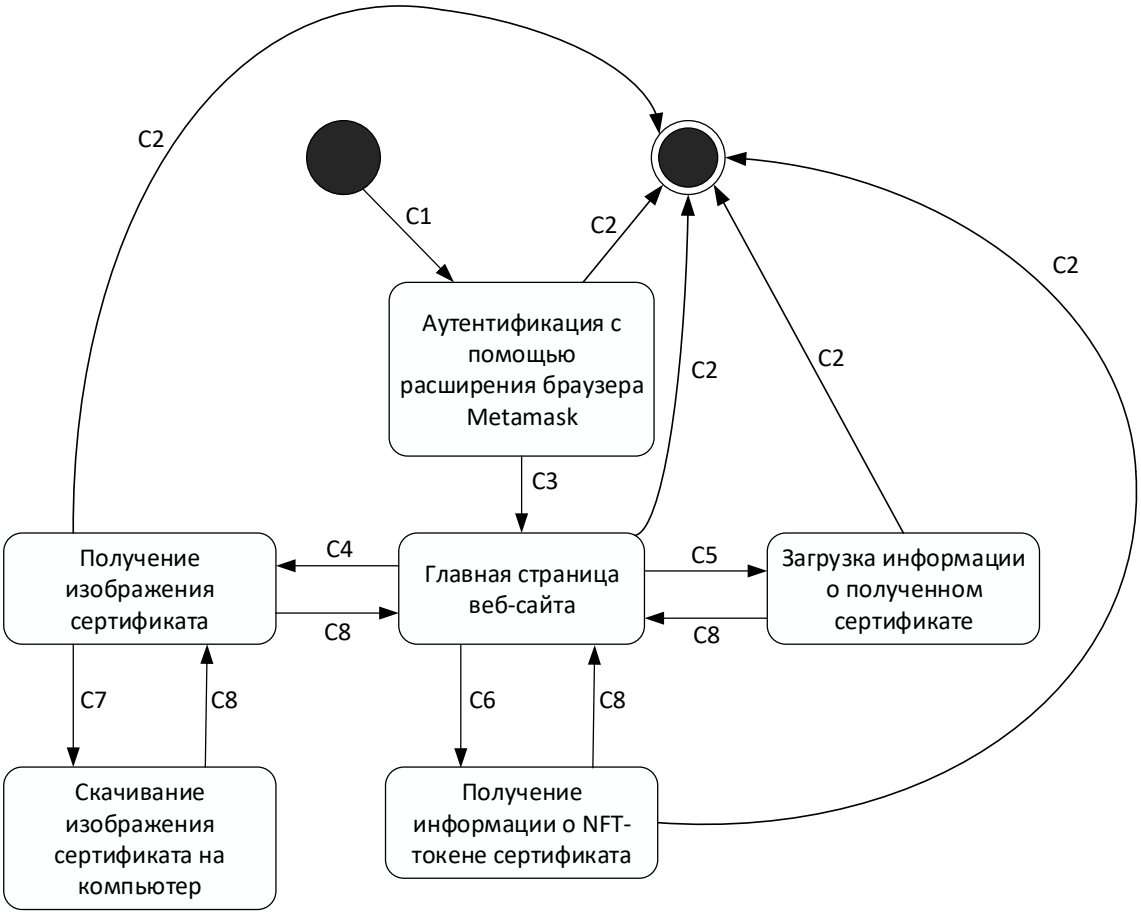

Рисунок 10 – Граф состояний интерфейса веб-сайта обучающихся в образовательной организации

На графе отмечены следующие переходы между состояниями:

С1 – Открытие страницы веб-сайта;

С2 – Закрытие вкладки веб-сайта;

С3 – Нажатие кнопки “Войти с Metamask”;

С4 – Нажатие кнопки “Загрузить сертификат”;

С5 – Нажатие кнопки “Загрузить информацию о полученном сертификате”;

С6 – Нажатие кнопки “Загрузить информацию о NFT-токене сертификата”;

С7 – Нажатие кнопки “Скачать сертификат”;

С8 – Нажатие кнопки закрытия текущей вкладки.

Разработанные графы полностью отображают всевозможные состояния интерфейса программной системы выдачи электронных сертификатов.

2.4.2 Проектирование диалогов

При проектировании диалогов был использован комбинированный тип с преобладанием типа «управляемого пользователем», так как выполнение текущей функции напрямую зависит от выбора пользователя, т.е. не предусматривается строгого сценария выполнения программы.

Интерфейс может быть классифицирован по формам диалога. Различают три формы диалога:

1) фразовая, которая предполагает «общение» с пользователем на естественном языке или его подмножестве;

2) директивная, которая предполагает использование со стороны пользователя команд (директив) специально разработанного формального языка;

3) табличная, которая предполагает выбор пользователем ответа из предложенных программой.

Исходя из функциональных особенностей программной системы, была выбрана табличная форма диалога, поскольку она не требует знания каких-либо команд пользователем, а также экономит ресурсы компьютера, необходимые для реализации поддержки взаимодействия системы с пользователем на ограниченно-естественном языке.

Граф абстрактного диалога формы “Меню функций программы” программной подсистемы администрирования образовательной организации представлен на рисунке 11.

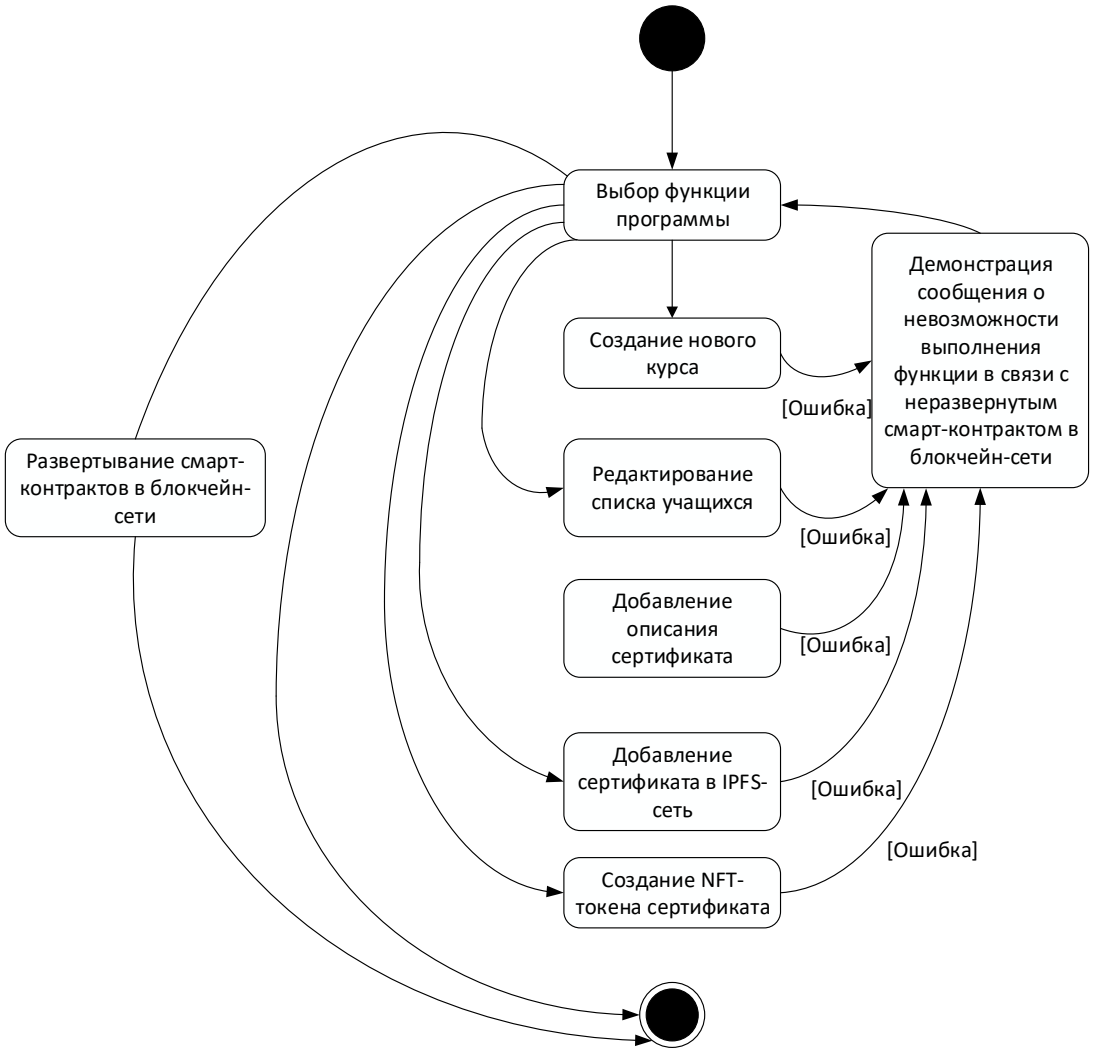

Рисунок 11 – Абстрактный диалог выбора функций программной подсистемы администрирования образовательной организации

Все формы программной подсистемы администрирования образовательной организации достаточно похожи друг на друга с точки зрения интерфейса форм. Страницы веб-сайта обучающихся в образовательной организации также практически идентичны, поэтому было принято решение построить граф абстрактного диалога лишь для одной из них. Граф абстрактного диалога страницы веб-сайта “Получение изображения сертификата” представлен на рисунке 12.

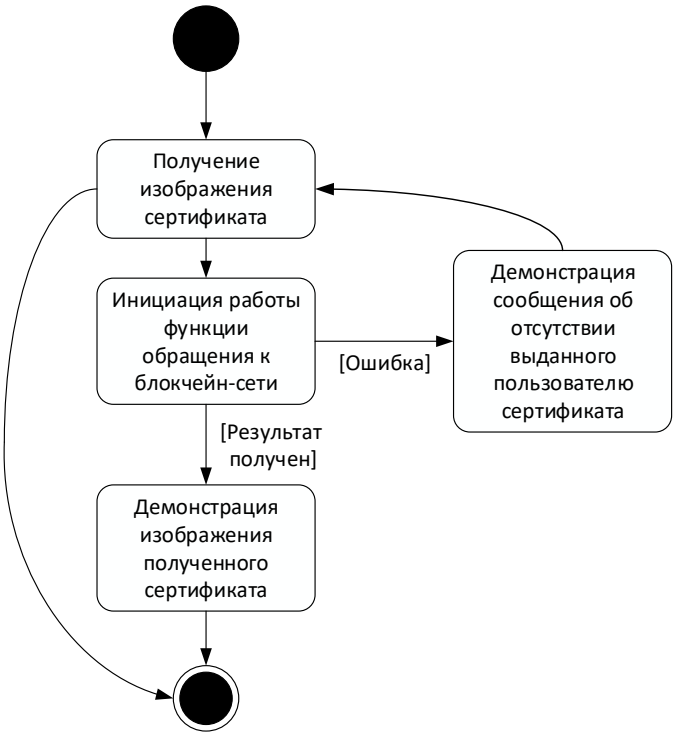

Рисунок 12 – Абстрактный диалог получения изображения сертификата

Разработанные абстрактные диалоги предназначены для того, чтобы экранные формы работали по определенному сценарию.

2.4.3 Разработка форм ввода-вывода данных

Исходя из решений, сформулированных выше, были сформированы экранные формы программной подсистемы администрирования образовательной организации. Вид данных экранных форм представлен на рисунках 13-18.

Экранная форма главного окна программы представлена на рисунке 13. На ней находится логотип программной системы выдачи электронных сертификатов, сведения о разработчике, а также кнопка перехода в меню функций программы, кнопка открытия справки по работе программы и кнопка выхода.

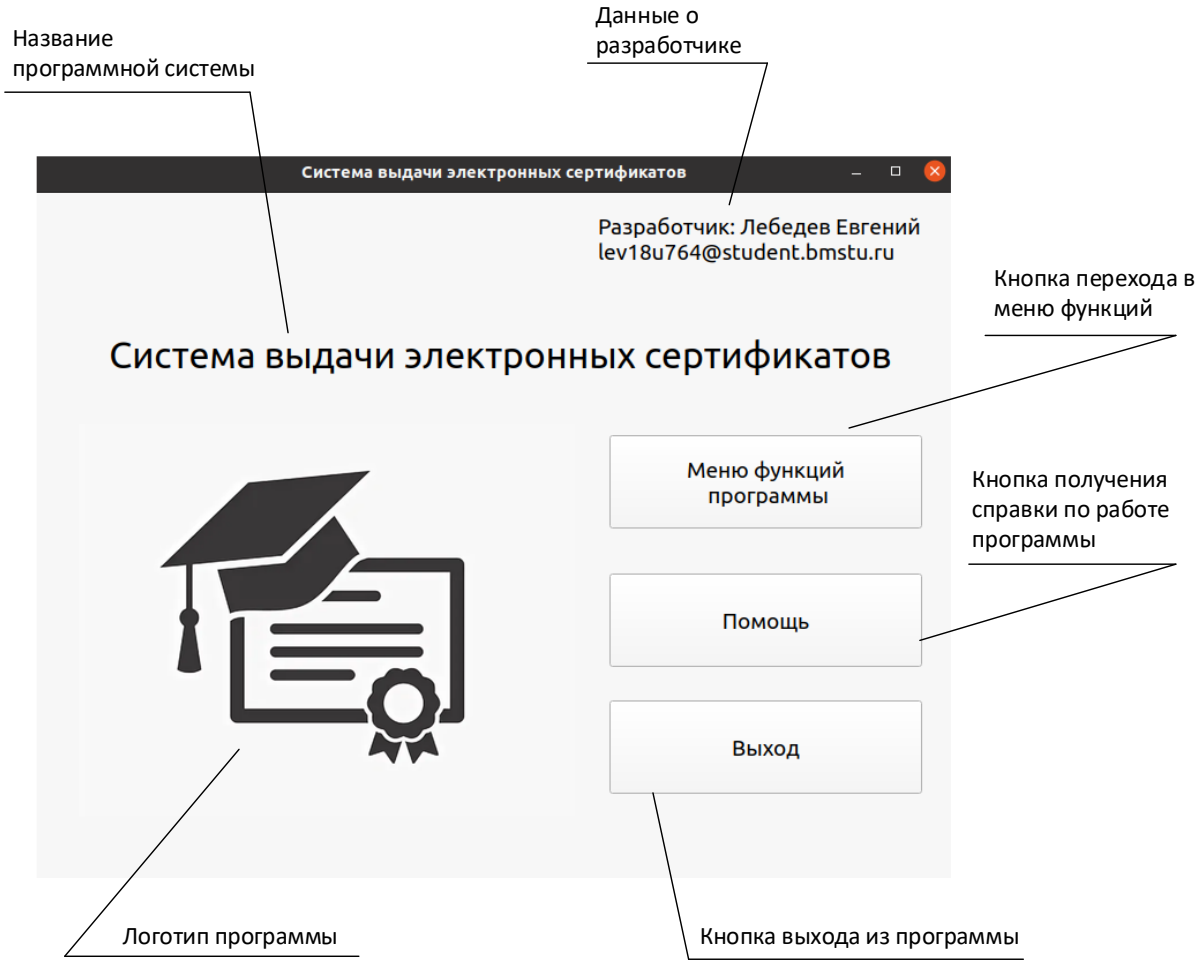

Рисунок 13 – Форма главного окна программы

На рисунке 14 представлена форма меню функций программы. На ней располагаются шесть кнопок: кнопка развертывания смарт-контрактов в блокчейн сети (смарт-контракт NFT-сертификата, а также смарт-контракт, хранящий функции образовательной организации), кнопка создания нового обучающего курса, кнопка редактирования списка учащихся, кнопка создания описания сертификата, кнопка добавления сертификата в одноранговую децентрализованную сеть IPFS, а также кнопка создания NFT-сертификата).

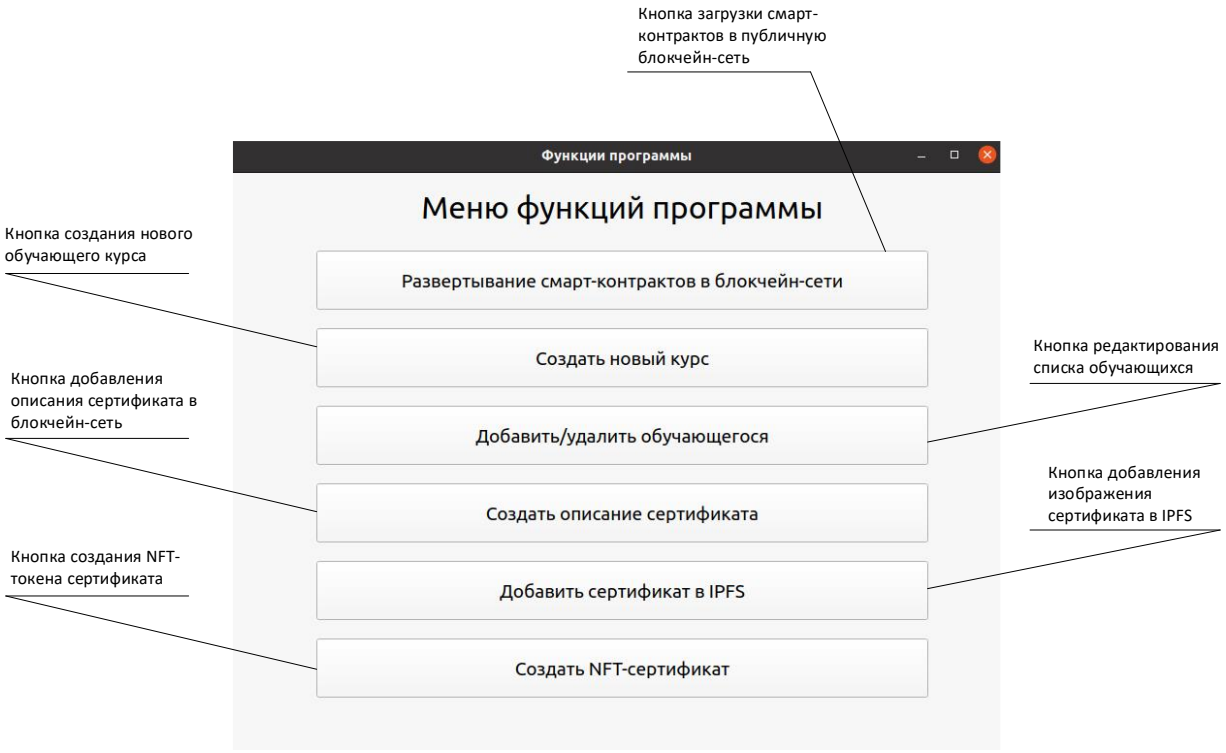

Рисунок 14 – Форма меню функций программы

На рисунке 15 представлена форма развертывания смарт-контрактов в блокчейн-сети. На ней располагаются поля для ввода наименования аббревиатуры образовательной организации и кнопка загрузки смарт-контрактов в публичную блокчейн-сеть. Помимо этого, на данной экранной форме располагается ее логотип (логотип блокчейн-сети Ethereum).

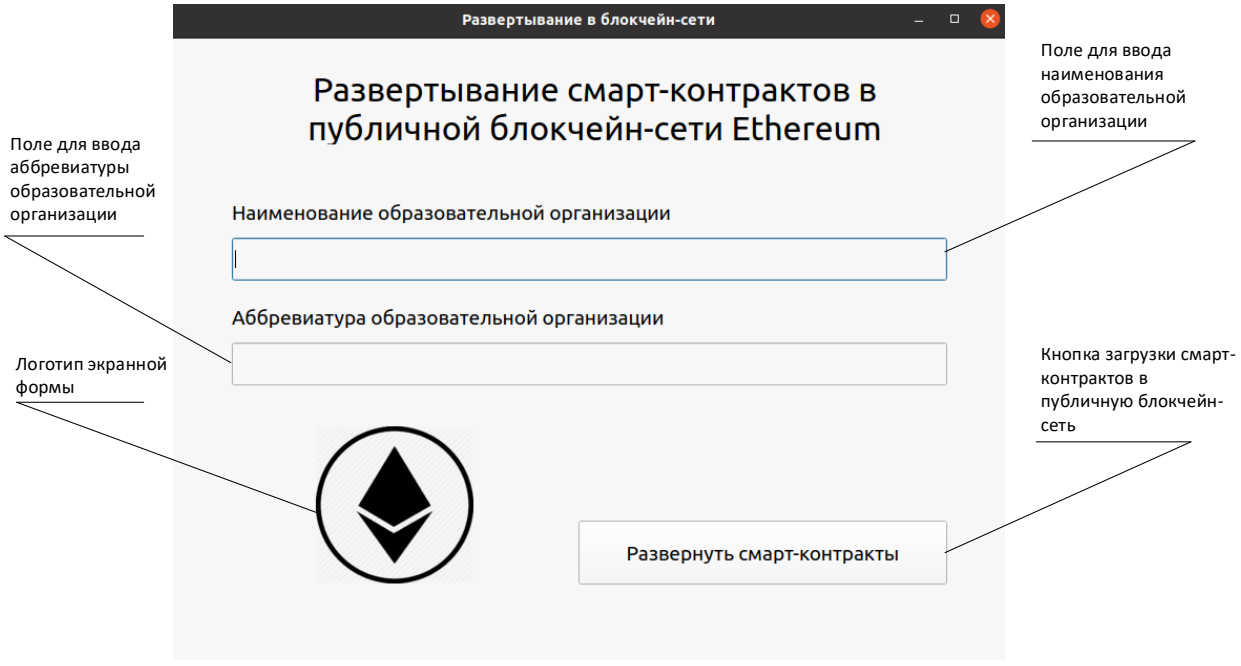

Рисунок 15 – Форма развертывания смарт-контрактов в блокчейн-сети

На рисунке 16 представлена форма добавления описания сертификата в блокчейн-сеть. На форме располагаются поля для ввода наименования образовательного курса, блокчейн-адреса учащегося, даты начала курса и выдачи сертификата, полученного количества баллов и дополнительной информации, а также кнопка добавления описания сертификата в блокчейн-сеть.

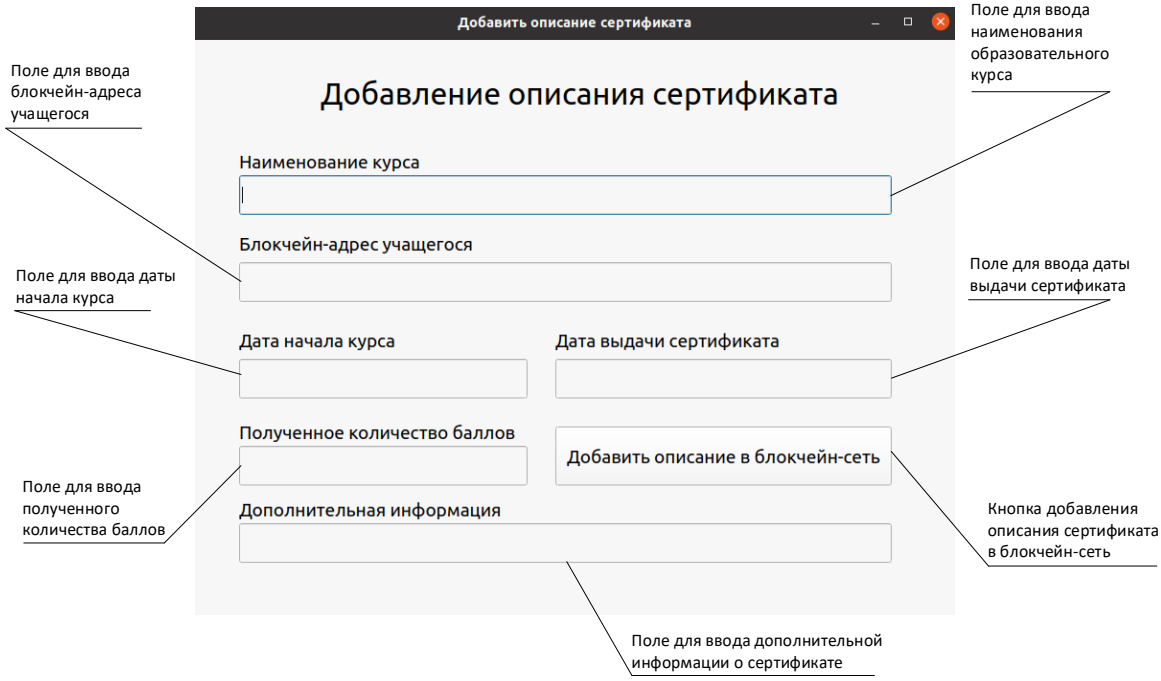

Рисунок 16 – Форма добавление описания сертификата

На рисунке 17 представлена форма добавления файла сертификата в IPFS-сеть. На ней расположены кнопка выбора изображения сертификата с компьютера и кнопка загрузки сертификата в IPFS-сеть. Помимо этого, на форме находится поле вывода адреса добавленного сертификата в IPFS-сети и ее логотип.

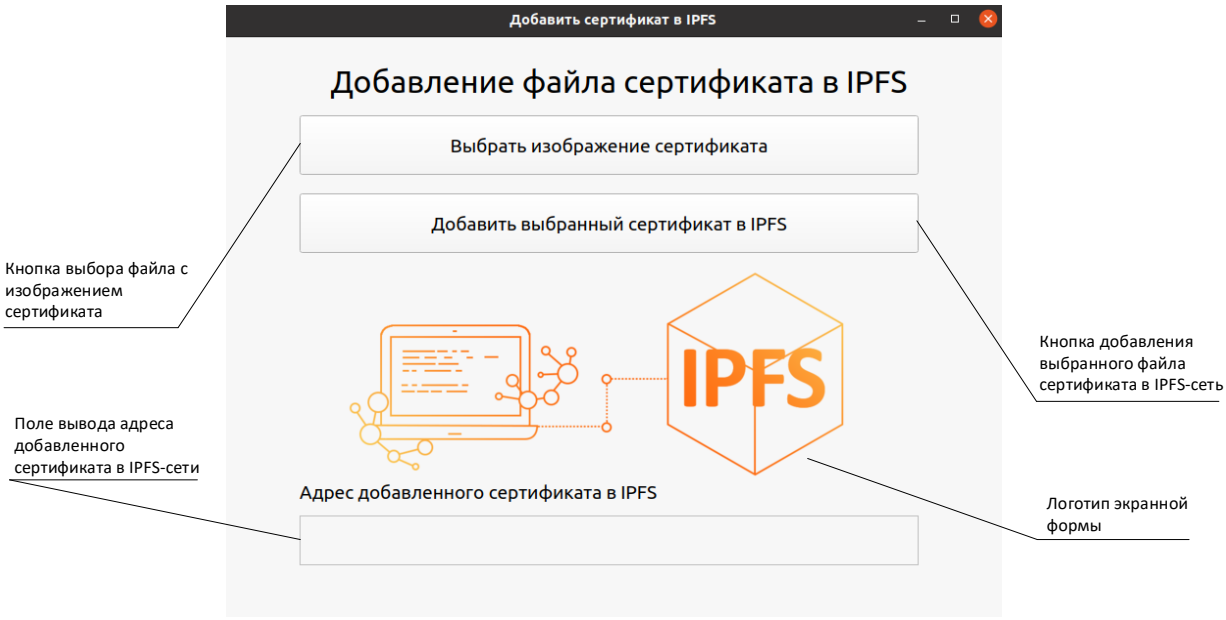

Рисунок 17 – Форма добавления сертификата в IPFS

На рисунке 18 представлена форма создания NFT-сертификата. На ней расположены поля ввода наименования курса, блокчейн-адреса учащегося и адреса сертификата в IPFS-сети, а также кнопка создания NFT-токена сертификата в блокчейн-сети с указанными в полях ввода параметрами и логотип экранной формы.

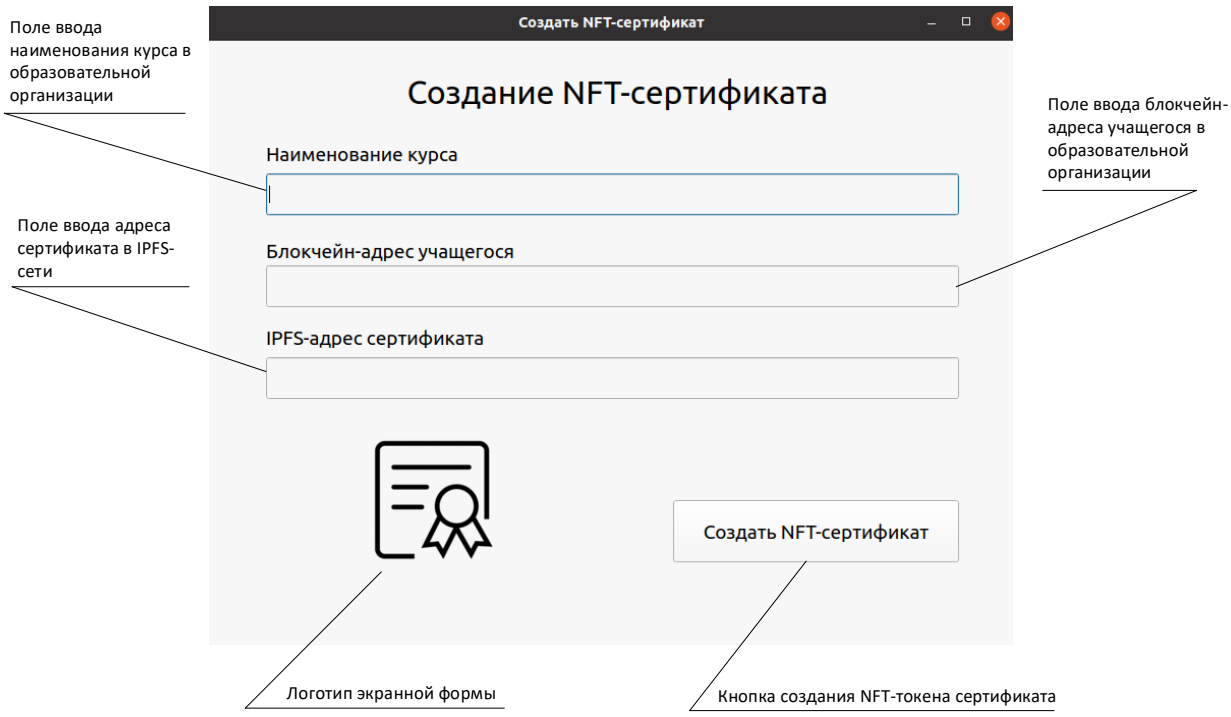

Рисунок 18 – Форма создания NFT-сертификата

Помимо этого, были сформированы две страницы веб-сайта обучающихся в образовательной организации: страница аутентификации с использованием блокчейн-адреса и главная страница веб-сайта.

На главной странице веб-сайта расположены 3 блока: блок получения изображения сертификата и скачивания его на компьютер, блок получения метаданных о NFT-токене сертификата, а также блок получения метаданных о выданном сертификате. На каждом из блоков находится поле ввода названия обучающего курса. Информация о блокчейн-адресе обучающегося хранится на веб-сайте и проверяется в процессе аутентификации через расширение браузера Metamask. Главная страница веб-сайта представлена на рисунке 19.

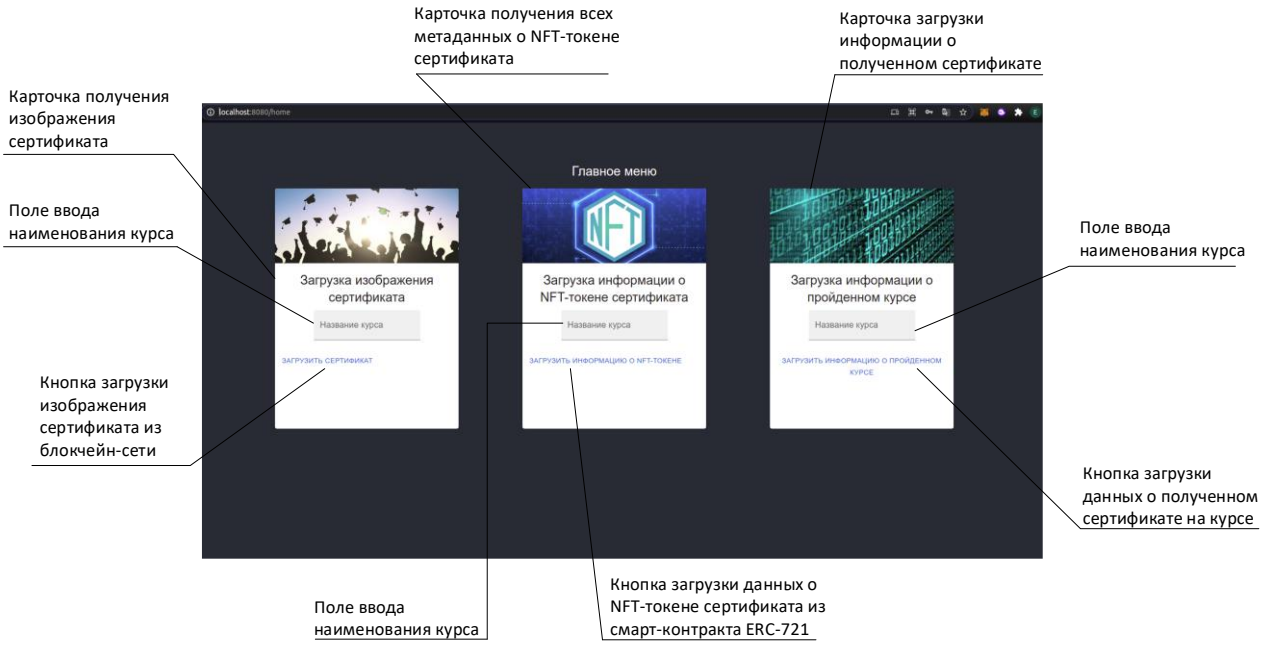

Рисунок 19 – Главная страница веб-сайта обучающихся в образовательной организации

Формы интерфейса были разработаны в соответствии с требованиями по функциональности, указанными в техническом задании.

2.4.4 Разработка диаграммы классов интерфейсной части

Для описания физической реализации существующих экранных форм была составлена диаграмма классов интерфейсной части. Все классы экранных форм наследуются от основного класса библиотеки PyQT5 – QWidget [4]. Каждый класс отвечает за конкретную экранную форму программы. Класс Mainmenu реализует главное окно программы, класс Help – окно справки, класс

Functions отвечает за реализацию меню основных функций программы. Класс DeployForm необходим для развертывания смарт-контрактов в публичной блокчейн-сети, класс AddCourse предоставляет механизмы добавления нового обучающего курса, класс Learners позволяет администраторам редактировать блокчейн-адреса обучающихся. Классы Description, IPFS и NFTForm предоставляют механизмы обработки цифровых сертификатов: добавление описания сертификата в блокчейн-сеть, добавление изображения сертификата в IPFS-сеть и создание NFT-токена сертификата соответственно.

Разработанная диаграмма классов интерфейсной части представлена на рисунке 20.

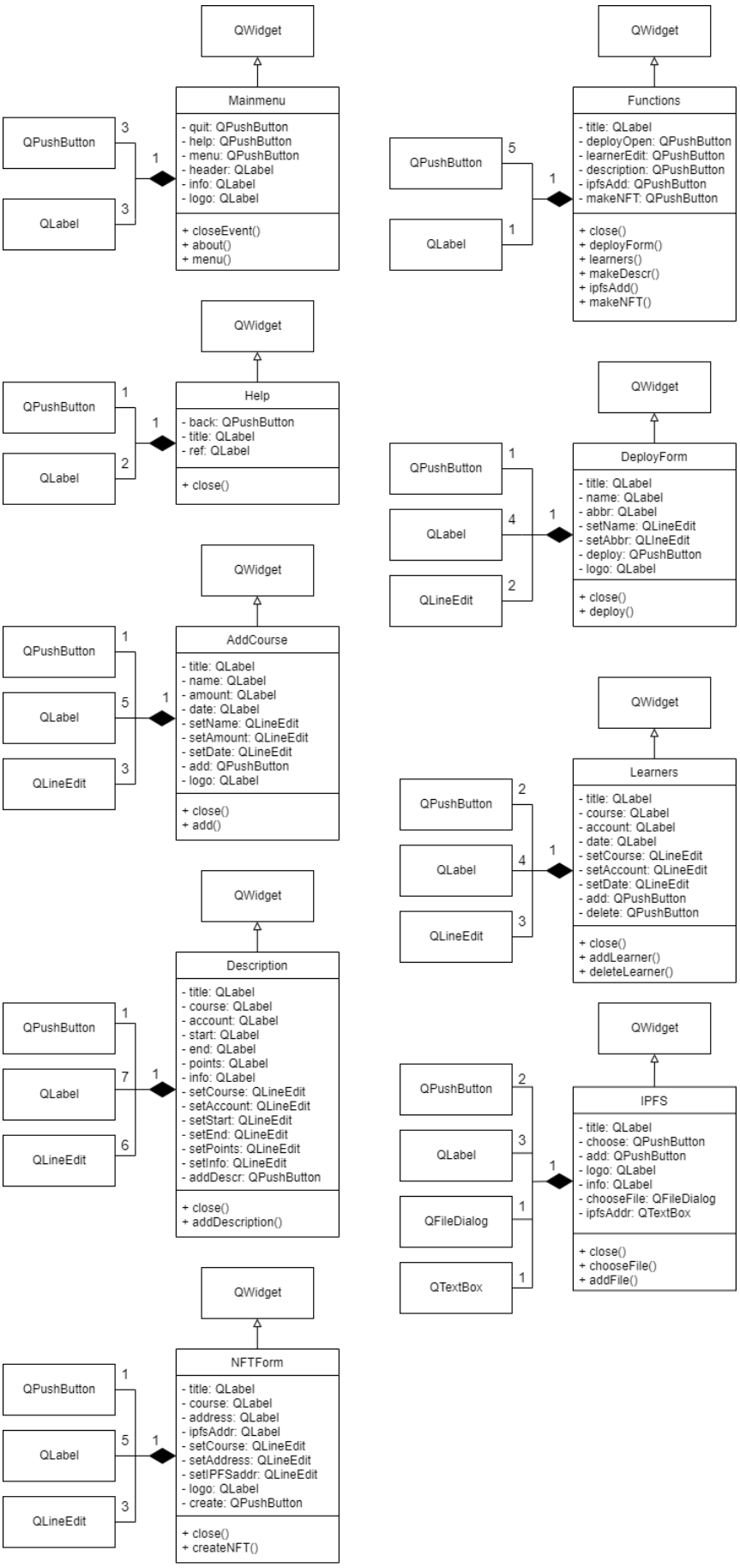

Рисунок 20 – Диаграмма классов интерфейсной части уровня реализации

Класс Mainmenu включает в себя три поля кнопок (QPushButton) и три поля заголовков (QLabel) библиотеки PyQT5. При этом реализованы методы класса, которые выполняются по нажатию на соответствующие кнопки: closeEvent() – метод закрытия окна программы с подтверждением выхода, about() – метод открытия окна со справкой и menu() – метод открытия окна выбора функций системы.

Класс Functions включает в себя пять кнопок и одно поле заголовка. Методы класса, которые выполняются по нажатию на соответствующие кнопки: close() – метод выхода из программы

2.5 Разработка концептуальной модели предметной области

При анализе предметной области и вариантов использования программной системы было выявлено:

Основные классы – это стандарт формирования NFT-токенов в блокчейн-сети ERC-721 и контроль доступа к методам (разграничение образовательной организации и обучающихся). Данные классы используются конкретным экземпляром произведенного NFT-сертификата. Помимо этого, существует класс функций образовательной организации, который производит NFT-сертификаты и использует управление доступом к своим методам. Класс обучающихся использует класс образовательной организации для получения электронных сертификатов.

Данная концептуальная модель отражена на диаграмме классов (см. рисунок 21).

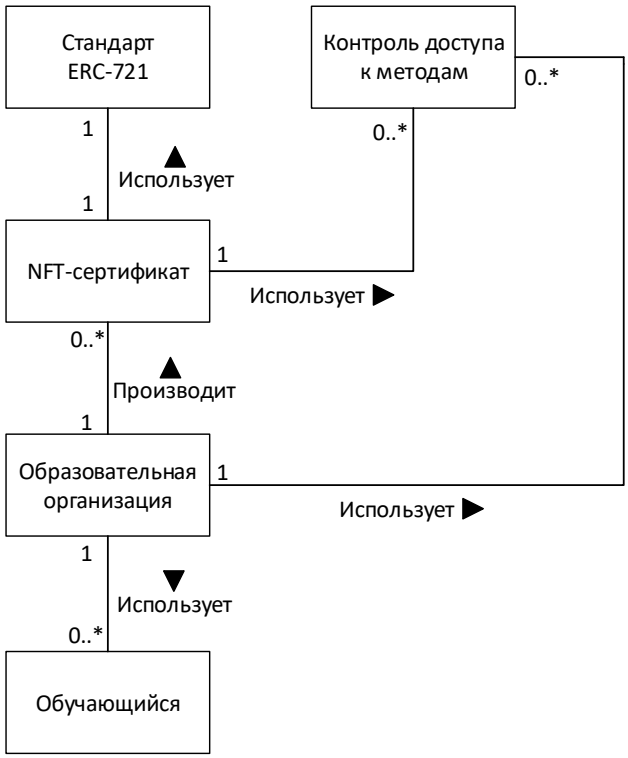

Рисунок 21 – Диаграмма классов концептуального уровня

Диаграмма классов концептуального уровня полно отражает основные понятия предметной области и связи между ними.

2.6 Разработка схем алгоритмов программного продукта

Программная система выдачи электронных сертификатов базируется на взаимодействии с блокчейн-сетью. В связи с тем, что в связи с большим количеством учащихся в образовательной организации, архитектура программной системы является многопользовательской, необходимо произвести развертывание смарт-контрактов, реализующих бизнес-логику программной системы и хранящих данные о сертификатах и учащихся, в публичной блокчейн-сети.

В связи с этим, были разработаны схемы алгоритмов: модуля развертывания смарт-контрактов в блокчейн-сети и создания электронного сертификата как уникального токена в блокчейн-сети. Схемы алгоритмов были созданы с целью уточнить варианты использования программного продукта для администратора (образовательной организации). Схема алгоритма развертывания смарт-контрактов в блокчейн-сети представлена на рисунке 22.

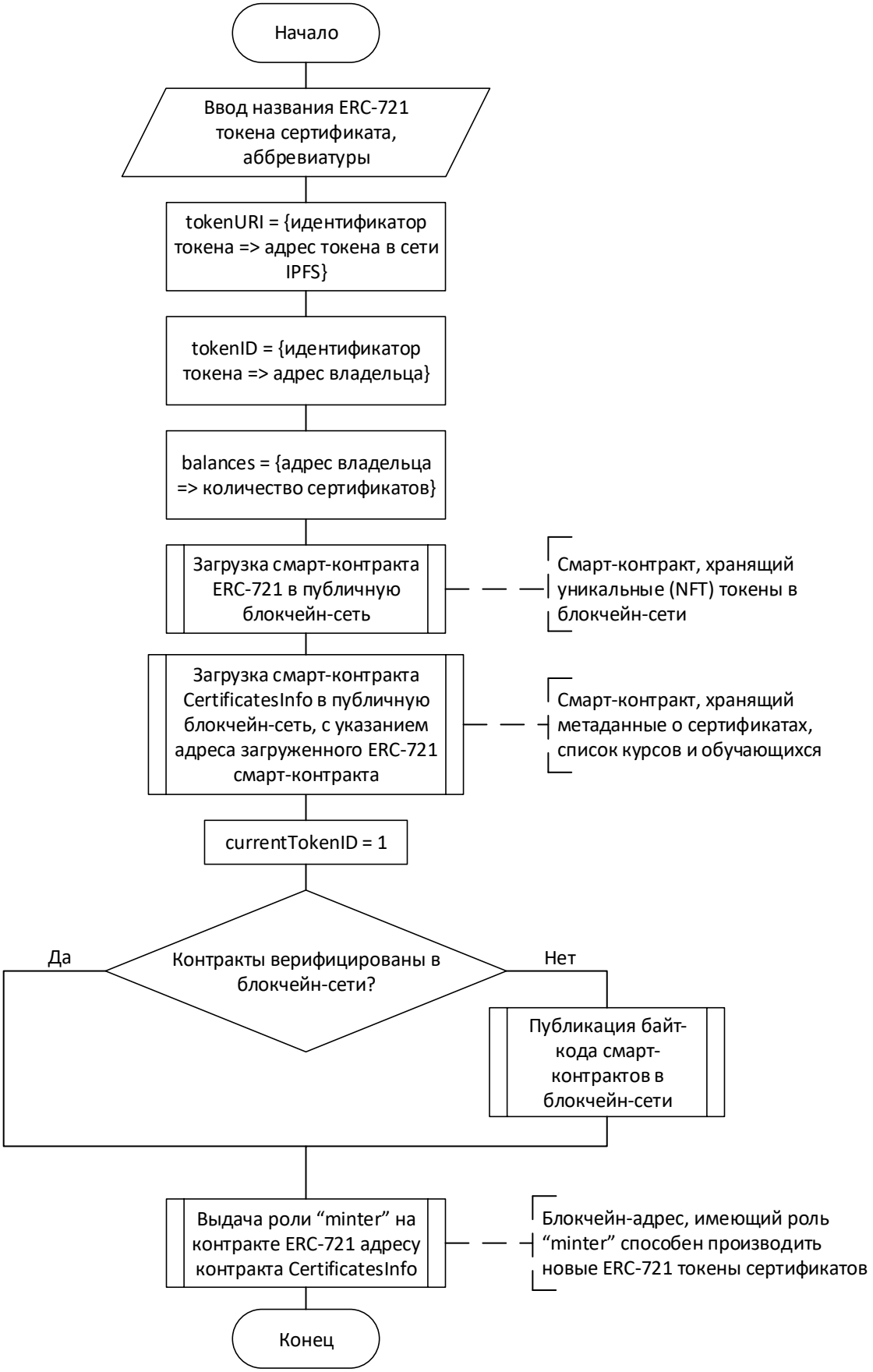

Рисунок 22 – Схема алгоритма модуля развертывания программной системы в блокчейн-сети

Названием ERC-721 токена сертификата является наименование образовательной организации, аббревиатурой токена – аббревиатура образовательной организации. После ввода названия инициализируются словари, которые необходимы по стандарту ERC-721 [3]. После инициализации словарей, происходит загрузка смарт-контрактов NFT и CertificatesInfo в публичную блокчейн-сеть, которая подразумевает плату комиссии со стороны образовательной организации за использование блокчейн-сети. Смарт-контракт NFT реализует хранение сертификатов в блокчейн-сети в виде уникальных токенов. Смарт-контракт CertificatesInfo предоставляет функции администрирования программной системы для образовательной организации, в том числе функцию создания цифрового сертификата. При загрузке смарт-контракта CertificatesInfo необходимо указать адрес уже развернутого в блокчейн-сети смарт-контракта NFT для разрешения обращения к нему.

После того, как смарт-контракты развернуты в блокчейн-сети, инициализируется значение счетчика текущего токена сертификата currentTokenId. После этого выполняется проверка верификации смарт-контрактов в блокчейн-сети. В случае отсутствия верификации, в блокчейн-сеть публикуется байт-код смарт-контрактов. Верификация необходима для отображения кода смарт-контрактов в блокчейн-сети. После верификации, адресу смарт-контракта CertificatesInfo выдается роль “minter”. Это означает, что только данный адрес может производить выпуск новых электронных сертификатов в блокчейн-сети.

Схема алгоритма создания электронного сертификата как уникального токена в блокчейн-сети представлена на рисунке 23.

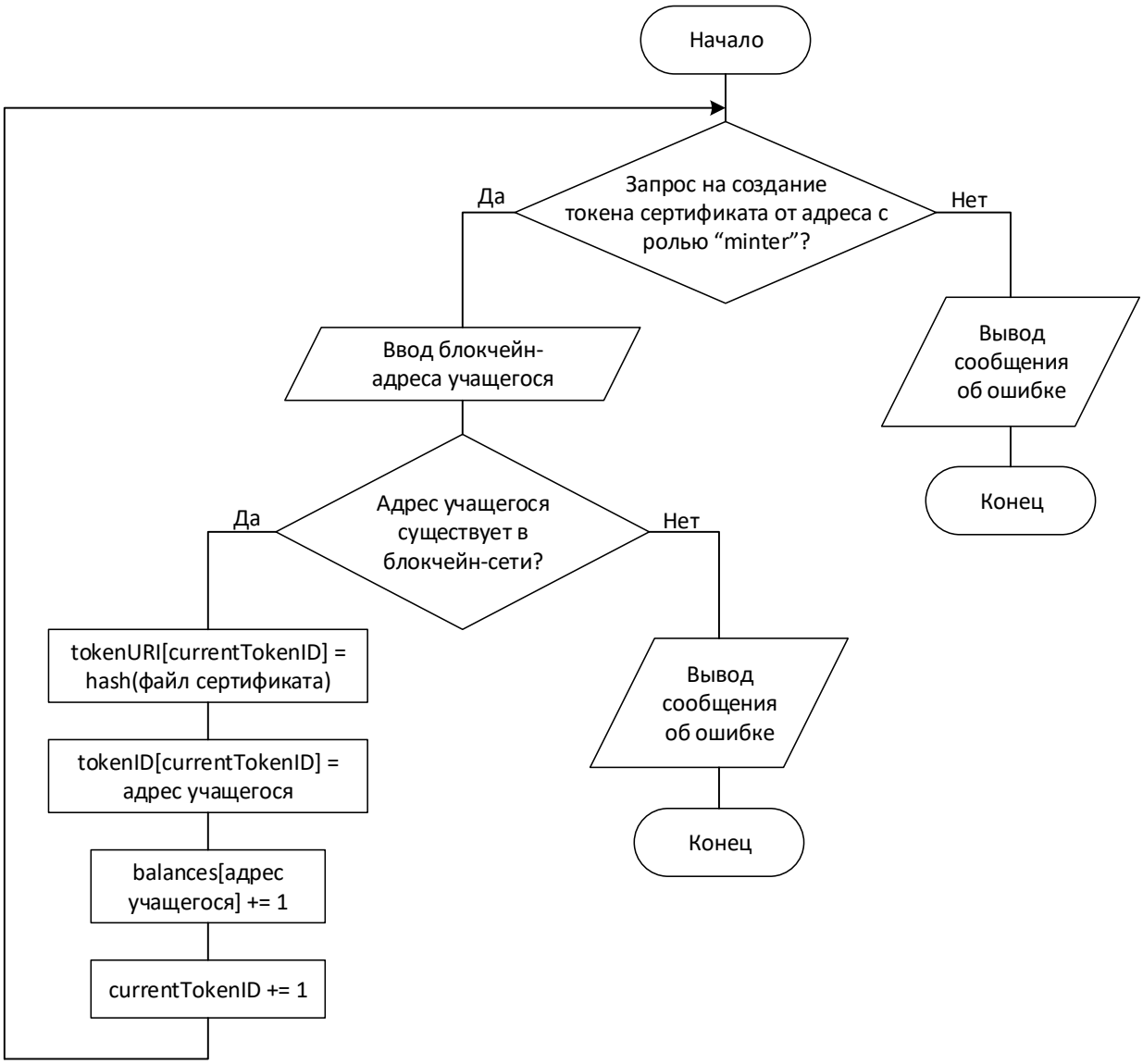

Рисунок 23 - Схема алгоритма создания электронного сертификата как уникального токена в блокчейн-сети

Таким образом, при поступлении запроса на создание токена сертификата, проверяется адрес, с которого был произведен запрос. Если адрес соответствует роли “minter”, администратор вводит блокчейн-адрес учащегося, которому собирается выпустить электронный сертификат. Если введенный адрес существует в блокчейн-сети, запоминаются данные связанные с выпуском токена в словари ERC-721, после чего инкрементируется значение счетчика произведенных токенов.

При запросе на выпуск токена не от роли “minter”, на экран выводится сообщение об ошибке, как и в случае отсутствия адреса учащегося в блокчейн-сети.

2.7 Разработка компонентов системы

Центральным звеном объектного подхода проектирования программного обеспечения является построение диаграмм классов, описывающих типы объектов (классы) и их статические отношения.

Используя построенную диаграмму классов концептуального уровня (см. рисунок 2), необходимо уточнить поля и методы конкретных классов. Результат данного уточнения – это реализация диаграммы классов предметной области уровня реализации, представленная на рисунке 23.

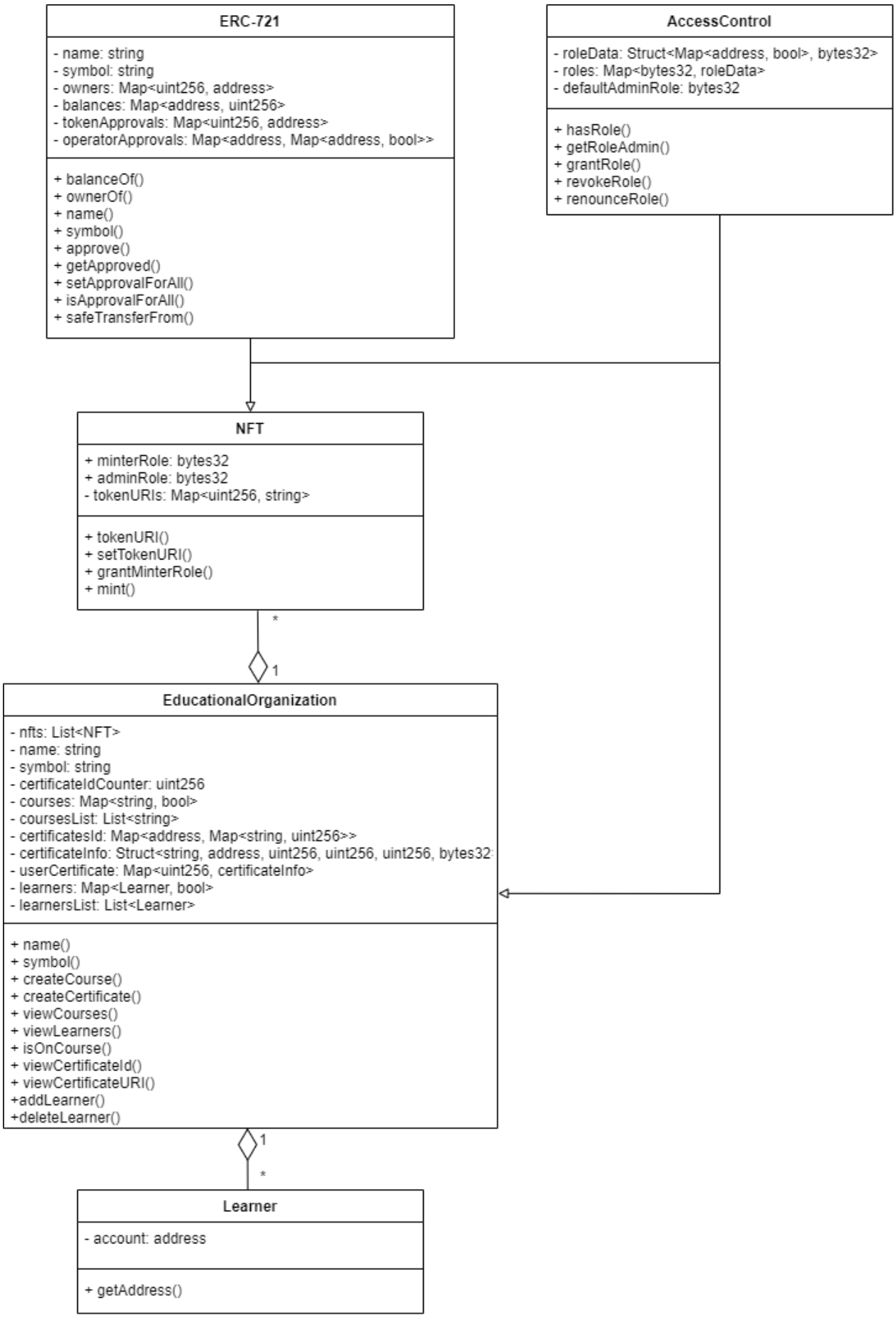

Рисунок 23 – Диаграмма классов предметной области уровня реализации

Класс ERC-721 представляет из себя стандарт формирования NFT-токенов в блокчейн-сети. В данном классе имеются поля названия NFT-токена и его аббревиатуры, поле владельцев токенов и их балансов, а также данные о подтверждении передачи токенов друг другу. Помимо этого, класс представляет методы для модификации данных полей. Использование всех полей обязательно, так как это является требованием по формированию NFT-токенов в блокчейн-сети.

Класс AccessControl предоставляет механизмы разграничения доступа к методам других классов. Данный класс имеет поля ролей, которые соотносятся с блокчейн-адресом аккаунта, а также поле роли администратора. Методы данного класса позволяют проверять наличие роли у пользователя, а также создавать новые роли. Данный класс необходим для разграничения возможности вызова методов смарт-контрактов образовательной организацией и обучающимися.

Класс NFT представляет расширение стандартного ERC-721 класса с добавлением поля tokenURIs, которое отвечает за адрес сертификата пользователя в IPFS-сети. Также класс имеет поля ролей: роли администратора (adminRole) и роли, которая может выпускать новые токены (minterRole).

Класс Learner представляет из себя класс обучающегося в образовательной организации. Данный класс содержит поле блокчейн-адреса обучающегося и метод его получения.

Класс EducationalOrganization представляет из себя класс образовательной организации. В данном классе имеются поля названия образовательной организации, ее аббревиатуры, список произведенных NFT-сертификатов, счетчик произведенных NFT-сертификатов, поля информации о курсах, сертификатах и обучающихся. Класс имеет методы получения содержимого полей, а также методы, изменяющие состояние блокчейн-сети, такие, как создание нового курса, сертификата или редактирование списка обучающихся в образовательной организации.

Разработанные классы представляют из себя смарт-контракты, написанные на языке Solidity. Данные смарт-контракты необходимо загрузить в блокчейн-сеть для начала их функционирования.

2.8 Компоновка программного продукта

При проектировании физической структуры программного продукта для каждой экранной формы был создан свой модуль. Файлы с расширением .sol представляют собой смарт-контракты, выкладываемые в публичную блокчейн-сеть. В отдельный модуль был вынесен файл развертывания программы в блокчейн-сети. Файлы с расширением .py представляют собой реализацию программной подсистемы администрирования для образовательной организации. Файлы с расширением .ts и .tsx представляют собой реализацию веб-сайта для обучающихся в образовательной организации. Помимо этого, было импортировано несколько изображений – логотип программы, использующийся на веб-сайте, а также изображения, использующиеся в оконных формах программной подсистемы администрирования. Диаграмма компоновки программы представлена на рисунке 24.

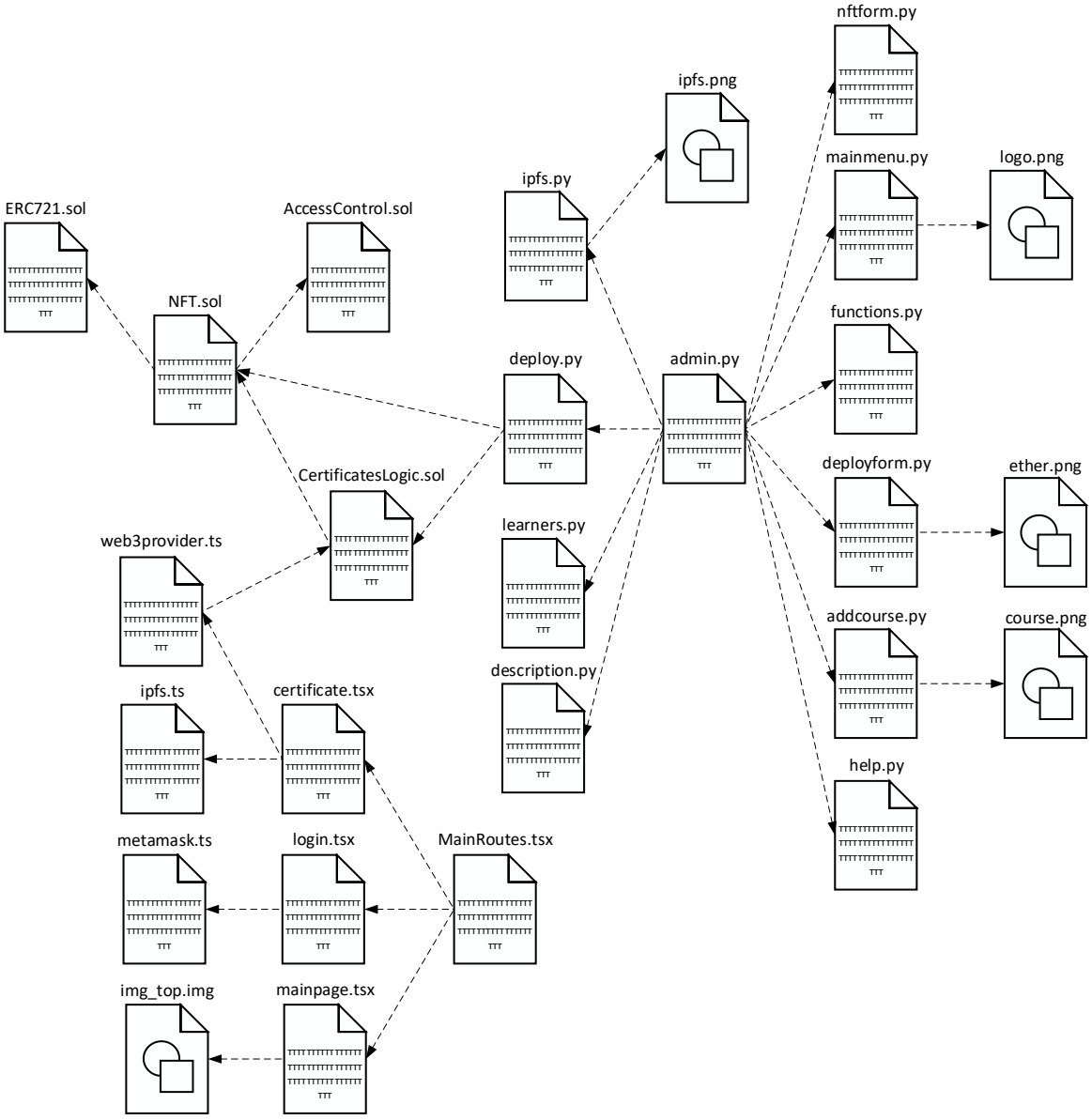

Рисунок 24 – Диаграмма компоновки программной системы

На диаграмме компоновки представлены следующие модули:

1) ERC721.sol – модуль стандарта создания NFT-токенов;

2) AccessControl.sol – модуль управления доступом к методам смарт-контрактов;

3) NFT.sol – модуль создания NFT-сертификата;

4) CertificatesLogic.sol – модуль функций программной системы в блокчейн-сети;

5) deploy.py – модуль развертывания программной системы в публично блокчейн-сети;

6) admin.py – основной модуль программы, содержащий интерфейсную часть администрирования для образовательной организации;

7) nftform.py – модуль экранной формы “Создание NFT-сертификата”;

8) mainmenu.py – модуль экранной формы “Главное окно”;

9) functions.py – модуль экранной формы “Функции программы”;

10) deployform.py – модуль экранной формы “Развертывание в блокчейн-сети”

11) addcourse.py – модуль экранной формы “Добавление нового курса”;

12) ipfs.py – модуль экранной формы “Добавление сертификата в ipfs”;

13) learners.py – модуль экранной формы “Редактирование списка учащихся”;

14) description.py – модуль экранной формы “Добавление описания сертификата”;

15) help.py – модуль экранной формы “Справка по работе программы”;

16) logo.png – логотип программы, используемый в главном меню

17) ether.png – изображение блокчейна “Ethereum”, используемое в форме развертывания смарт-контрактов;

18) course.png – изображение, используемое в форме добавления курса;

19) ipfs.png – изображение сети IPFS, используемое в форме добавления сертификата в IPFS;

20) web3provider.ts – модуль взаимодействия веб-сайта в блокчейн-сетью;

21) ipfs.ts – модуль обращения веб-сайта к IPFS-сети;

22) metamask.ts – модуль аутентификации пользователя с помощью блокчейн-аккаунта на веб-сайте;

23) certificate.tsx – модуль страницы веб-сайта получения данных о сертификате;

24) login.tsx – модуль страницы аутентификации на веб-сайт;

25) mainpage.tsx – модуль главной страницы веб-сайта;

26) MainRoutes.tsx – модуль маршрутизации страниц веб-сайта;

27) img_top.img – изображение, использующееся на главной странице веб-сайта.

Таким образом, диаграмма компоновки показывает все внутренние файлы проекта, необходимые для его правильного функционирования. Помимо этого, диаграмма компоновки полностью отображает то, как выглядит программная система выдачи электронных сертификатов на физическом уровне, то есть из каких частей она состоит и как эти части связаны между собой.

2.9 Тестирование системы

Тестирование представляет собой важный этап процесса разработки программного обеспечения. Оно преследует несколько целей: повысить вероятность правильной работы при любых обстоятельствах, повысить вероятность соответствия всем описанным требованиям, предоставить информацию о состоянии (готовности) программного продукта на конкретном этапе разработки.

Для тестирования разрабатываемого продукта проведем три вида тестирования: модульное тестирование (отдельных модулей подсистемы), функциональное тестирование (основных функций подсистемы) и оценочное тестирование.

2.9.1 Модульное тестирование

Модульное тестирование — тестирование программного продукта на уровне отдельных взятых модулей, методов. Целью данного вида тестирования является выявление локализованных ошибок в реализации алгоритмов и определение готовности к интеграции отдельных модулей в подсистему. Модульное тестирование считается низкоуровневым (близким к исходному коду модуля) и проходит по методу «белого ящика», т. е. для написания тестов используется знания о внутренней реализации (алгоритмах) модулей.

С помощью библиотеки Pytest было проведено модульное тестирование программы. В ходе него были протестированы все основные функции программы, а также реакция программы на входные данные. Результаты модульного тестирования функций программы приведены в таблице 16.

Таблица 16 – Результаты модульного тестирования

Тест	Ожидаемый результат	Полученный результат	Вывод
Инициализация наименования, аббревиатуры образовательной организации, адреса смарт-контракта ERC-721	[“Educational Organization 1”, “EO1”, “0xEAD7dcA2543126A5F8653Af2D532A162fCa05240”]	[“Educational Organization 1”, “EO1”, “0xEAD7dcA2543126A5F8653Af2D532A162fCa05240”]	Успешно
Создание образовательного курса, добавление количества академических часов и даты начала курса	“coursename_1” => 120 => “12.02.2022” => true	“coursename_1” => 120 => “12.02.2022” => true	Успешно
Добавление обучающегося на курс, инициализация его блокчейн-адреса	“coursename_1” => “12.02.2022” => “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f” => true	“coursename_1” => “12.02.2022” => “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f” => true	Успешно
Удаление обучающегося с курса	“coursename_1” => “12.02.2022” => “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f” => false	“coursename_1” => “12.02.2022” => “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f” => false	Успешно

Продолжение таблицы 16

Получение описания сертификата из блокчейн-сети	[“coursename_1”, “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f”, “12.02.2022”, “12.06.2022”, “66”, “additional info”]	[“coursename_1”, “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f”, “12.02.2022”, “12.06.2022”, “66”, “additional info”]	Успешно
Создание NFT-токена сертификата	_to = “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f” _balances[_to] => 1; _owners[1] => _to; 1 => IPFS-хэш сертификата	_to = “0x3Ba6810768c2F4FD3Be2c5508E214E68B514B35f” _balances[_to] =>1; _owners[1] => _to; 1 => “QmQEz56yDfZLZB1Zi7beJnC7jSie25djfgi8ViW3qM4MXw”	Успешно
Получение описания NFT-токена сертификата	https://ipfs.io/ipfs/QmQEz56yDfZLZB1Zi7beJnC7jSie25djfgi8ViW3qM4MXw	https://ipfs.io/ipfs/QmQEz56yDfZLZB1Zi7beJnC7jSie25djfgi8ViW3qM4MXw	Успешно
Получение изображения сертификата	https://ipfs.io/ipfs/QmXRd1gDhzpVXkxqRXwEaDH32JTsPzUfMh93n7sVFWY9Qn	https://ipfs.io/ipfs/QmXRd1gDhzpVXkxqRXwEaDH32JTsPzUfMh93n7sVFWY9Qn	Успешно

Таким образом, модульное тестирование показало внутреннюю корректность всех функций программной системы.

2.9.2 Функциональное тестирование

Для обеспечения качества программного продукта и своевременного выявления ошибок и несоответствий между реализованным и требуемым поведением функций системы необходимо своевременно проводить функциональное тестирование программного продукта. Целью данного мероприятия является обеспечение качества программного продукта и проверка соответствия реализованных функций требованиям к функциональной составляющей программного продукта.

Функциональное тестирование проводилось с помощью двух методов: метод эквивалентных разбиений, анализ причинно-следственных связей.

Метод эквивалентного разбиения предполагает разбиение всех возможных наборов входных данных программы на конечное число групп – классов эквивалентности.

Входные данные программы представляют собой графические файлы сертификатов. На одном из этапов в процессе создания NFT-токена сертификата, администратор загружает их в IPFS-сеть с целью получения их адреса в данной сети. При этом особенно важно, чтобы адреса сертификатов не совпадали, даже при незначительных графических изменениях исходного файла. Это обеспечило бы невозможность выдачи чужого сертификата за свой со стороны недобросовестных обучающихся в образовательной организации.

На основе этого можно выделить два класса эквивалентности для значений входных переменных: уникальные изображения сертификатов, дубликаты. Проведем тестирование путем загрузки сертификатов в IPFS-сеть и получением IPFS-адреса сертификата. Результаты отображены в таблице 17.

Таблица 17 – Результаты тестирования методом эквивалентного разбиения

Тест	Ожидаемый результат	Полученный результат	Вывод
Получение IPFS-адреса сертификатов путем загрузки двух одинаковых файлов в IPFS-сеть	Полученные IPFS-адреса будут одинаковыми	1: https://ipfs.io/ipfs/QmXRd1gDhzpVXkxqRXwEaDH32JTsPzUfMh93n7sVFWY9Qn, 2: https://ipfs.io/ipfs/QmXRd1gDhzpVXkxqRXwEaDH32JTsPzUfMh93n7sVFWY9Qn	Успешно
Получение IPFS-адреса сертификатов путем загрузки двух файлов, различающихся на один пиксель, в IPFS-сеть	Полученные IPFS-адреса будут разными	1: https://ipfs.io/ipfs/QmXRd1gDhzpVXkxqRXwEaDH32JTsPzUfMh93n7sVFWY9Qn, 2: https://ipfs.io/ipfs/QmXPf8VawymaUP8vB2CjtJAuttNTUmvcKaEfNSSJf1BsXV	Успешно

Помимо этого, программная система была протестирована методом анализа причинно-следственных связей для исходных данных об одном добавленном курсе, одном обучающемся и выданным ему образцом электронного сертификата с добавленным описанием. Результаты тестирования отображены в таблице 18.

Таблица 18 – Результаты тестирования методом анализа причинно-следственных связей

Тест	Ожидаемый результат	Полученный результат	Вывод
Аутентификация на веб-сайте с помощью блокчейн-адреса	Переход на главную страницу в случае корректности введенных данных	Отображение главной страницы сайта, ошибка в случае неправильно введенного пароля	Успешно
Получение изображения сертификата с курса	При правильно введенном курсе происходит переход на страницу с изображением сертификата	Отображение сертификата по IPFS-адресу, ошибка в случае неправильно введенного курса	Успешно
Получение метаданных о NFT-токене сертификата	При правильно введенном курсе отображается уникальный идентификатор NFT-токена, адрес смарт-контракта и блокчейн-сети	Отображение цифры 1 в графе идентификатор NFT-токена, адреса смарт-контракта, а также названия сети “Ethereum (Rinkeby)”, сообщение об ошибке в случае неправильно введенного курса	Успешно

Продолжение таблицы 18

Получение метаданных о выданном сертификате	При правильно введенном курсе отображается дата начала прохождения курса, дата получения сертификата, полученный балл	Отображение всех добавленных администратором данных о сертификате, сообщение об ошибке в случае неправильно введенного курса	Успешно

Таким образом, функциональное тестирование показало корректную работу программы на тестовых данных.

2.9.3 Оценочное тестирование

В связи с тем, что программная система рассчитана на большое количество пользователей, было проведено тестирование удобства ее эксплуатации. Данный вид тестирования представляет собой тестирование графического интерфейса системы на «дружелюбность», соответствие поведения интерфейса ожиданиям пользователя, очевидность расположения компонентов интерфейса.

Тестирование проводилось в двух фокус-группах, куда были приглашены респонденты по 5 человек. Респонденты в каждой из них относились к различным возрастным и социальным группам. По результатам тестирования были приняты меры по доработке интерфейса и улучшению сценариев взаимодействия с пользователем. Результаты тестирования представлены в таблице 19.

Таблица 19 – Результаты тестирования удобства эксплуатации

Выявленный недостаток	Меры по исправлению
Если расширение Metamask не установлено в браузере заранее, при нажатии на кнопку “Войти” ничего не происходит	Добавлена проверка на установку расширения в браузере. В случае отсутствия расширения, добавлен переход на страницу для скачивания.

Продолжение таблицы 19

При нажатии на кнопку “Открыть другой сертификат” поле для ввода курса не очищается	Добавлена функция очищения поля для ввода курса при нажатии на кнопку “Открыть другой сертификат”
Если дополнительная информация о выданном сертификате достаточно длинная, она выходит за рамки отображения на экране	Добавление переменной, хранящей максимальную длину одной строки дополнительной информации о выданном сертификате

Как видно из результатов тестирования удобства эксплуатации, жалобы пользователей связаны с неочевидностью поведения интерфейса и отсутствием отклика в некоторых сценариях. Все выявленные респондентами недостатки были устранены.

С целью проверки эффективности используемых механизмов выдачи электронных сертификатов, было проведено тестирование защиты программной системы, а именно защиты от несанкционированного доступа к функциям администрирования программной системы. Самыми потенциально уязвимыми местами программной системы являются функции, изменяющие состояние блокчейн-сети (добавление нового обучающего курса, добавление блокчейн-адреса обучающихся на курс, создание электронных сертификатов и загрузка метаданных о них).

Запуск данных функций возможен лишь с единственного блокчейн-адреса, который помечен как “владелец” смарт-контрактов. Таким образом, для получения доступа к данному адресу, необходимо подобрать закрытый ключ аккаунта владельца. Подбор закрытого ключа в наши дни невозможен, даже при наличии огромных вычислительных мощностей, из-за стойкости алгоритма ECDSA, который используется в блокчейн-сети Ethereum. Данный алгоритм основывается на проблеме дискретного логарифмирования в группе точек эллиптической кривой.

Таким образом, можно сказать, что добавление несуществующего NFT-сертификата в смарт-контракт, а также вызов любой функции, изменяющей состояние блокчейн-сети, лицами, не имеющими на это право, возможен лишь при утрате образовательной организацией своего приватного ключа от блокчейн-аккаунта.

3 Разработка технологии использования программной системы

3.1 Этапы введения в эксплуатацию

Для работы с программной системой необходимо наличие аккаунта в блокчейн-сети, как для администрирования со стороны образовательной организации, так и для обучающихся. Существует множество менеджеров блокчейн-аккаунтов, которые называются “кошельками”. Самым популярным и используемым является Metamask.

Metamask – это некастодиальный менеджер блокчейн-аккаунтов который позволяет совершать транзакции в блокчейн-сети. Кошелек Metamask существует как расширение для браузеров и приложение для смартфона. Он позволяет создавать приватные ключи, взаимодействовать с ними и управлять активами.

Первым этапом введения программной системы в эксплуатацию является создание своего аккаунта в блокчейн-сети со стороны образовательной организации. С данного аккаунта в дальнейшем будут вызываться функции администрирования программной системы. В свою очередь, каждый обучающийся также должен создать аккаунт в блокчейн-сети, публичный адрес которого будет выступать уникальным идентификатором обучающегося.

Для создания приватного ключа от блокчейн-аккаунта в расширении Metamask используется seed-фраза, состоящая из 12 слов. Особенно важно, обеспечить сохранность данной фразы в надежном месте на внешнем зашифрованном жестком диске или носителе, так как ее раскрытие может привести к утрате контроля над блокчейн-аккаунтом. Пример секретной фразы в расширении Metamask приведен на рисунке 25.

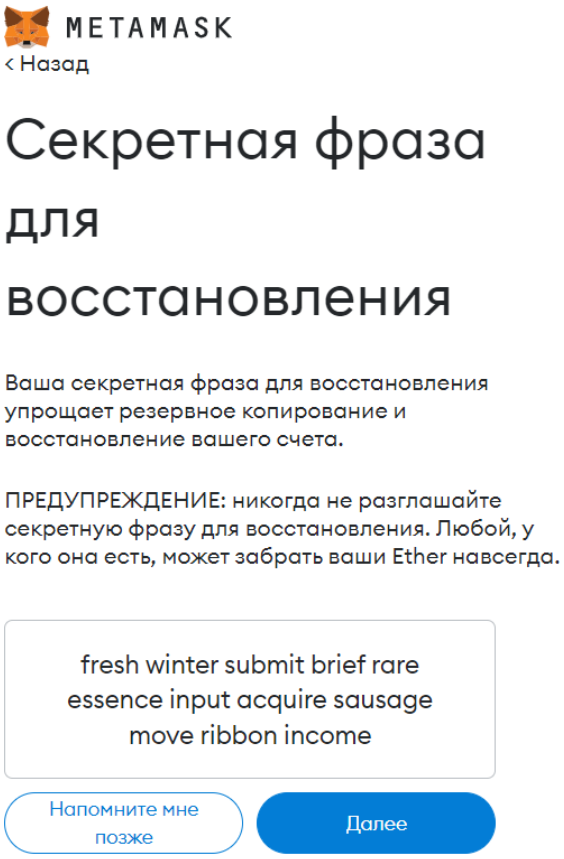

Рисунок 25 – Секретная фраза для восстановления блокчейн-аккаунта

Вторым этапом введения программной системы в эксплуатацию является пополнение аккаунта в блокчейн-сети нативными токенами. Так как функции, изменяющие состояние блокчейн-сети, являются платными, необходимо пополнить блокчейн-аккаунт. Нативными токенами являются управляющие токены сети. При использовании блокчейн-сети Ethereum нативным токеном является токен ETH, для сети Binance Smart Chain – токен BNB.

Третьим этапом введения программной системы в эксплуатацию является развертывание смарт-контрактов программной системы в блокчейн-сети администрацией образовательной организации. Самым удобным способом является использование интерфейса подсистемы администрирования программной системы.

В файле config.ts, в переменной “private_key” необходимо указать приватный ключ созданного ранее блокчейн-аккаунта образовательной организации. Данная операция нужна для подписания транзакции на создание смарт-контрактов в блокчейн сети. Предварительно необходимо пополнить

блокчейн-аккаунт нативной криптовалютой, так как загрузка смарт-контракта в блокчейн-сеть является платной в связи необходимостью изменения состояния блокчейн-сети.

При успешном развертывании смарт-контрактов в блокчейн-сети на экран будет выведено соответствующее сообщение с приложенным хэшем совершенных транзакций в блокчейн-сети. Пример успешной транзакции развертывания смарт-контракта в блокчейн-сети приведен на рисунке 26.

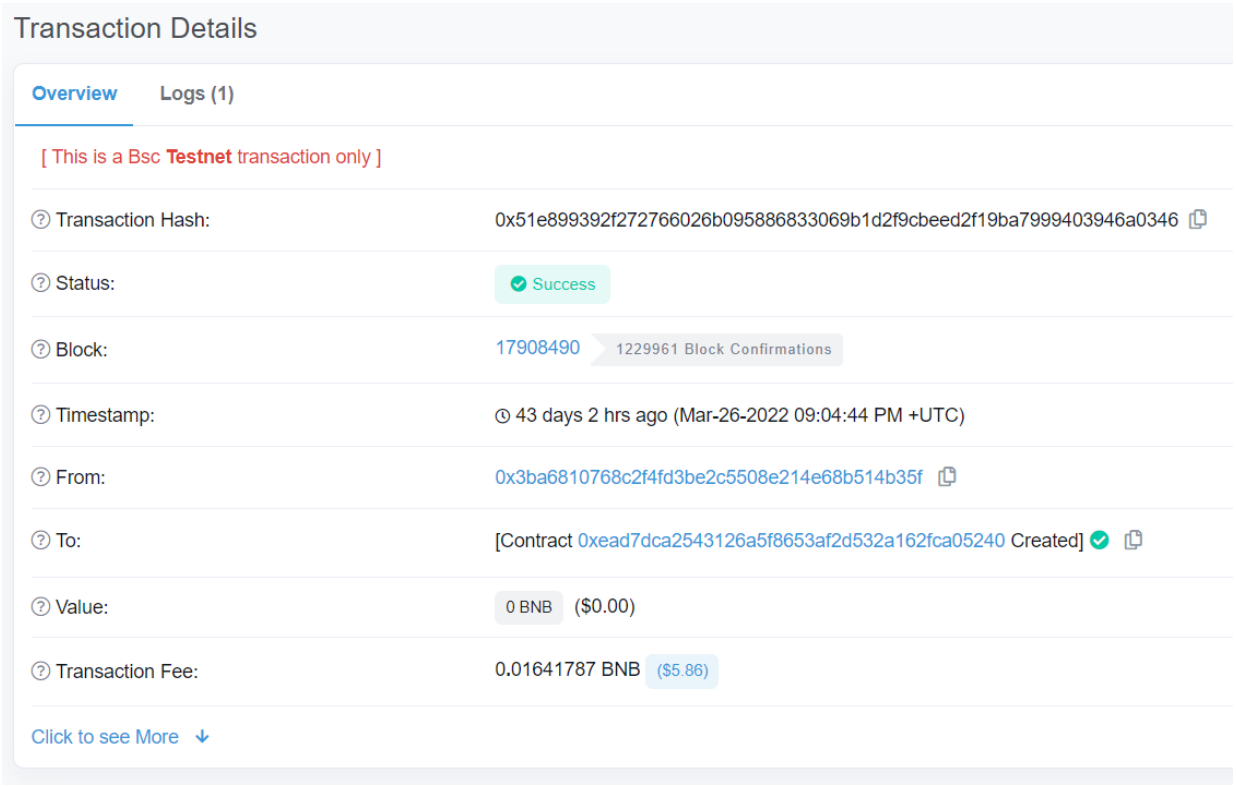

Рисунок 26 – Пример транзакции развертывания смарт-контракта в блокчейн-сети

Четвертым, пятым и шестым этапами введения программной системы в эксплуатацию являются добавление обучающих курсов, блокчейн-адресов обучающихся, а также загрузка сертификатов обучающихся в IPFS-сеть через интерфейс подсистемы администрирования образовательной организации соответственно. Все перечисленные действия также предполагают создание транзакций в блокчейн-сети. Пример транзакции создания нового обучающего курса “Math” приведен на рисунке 27.

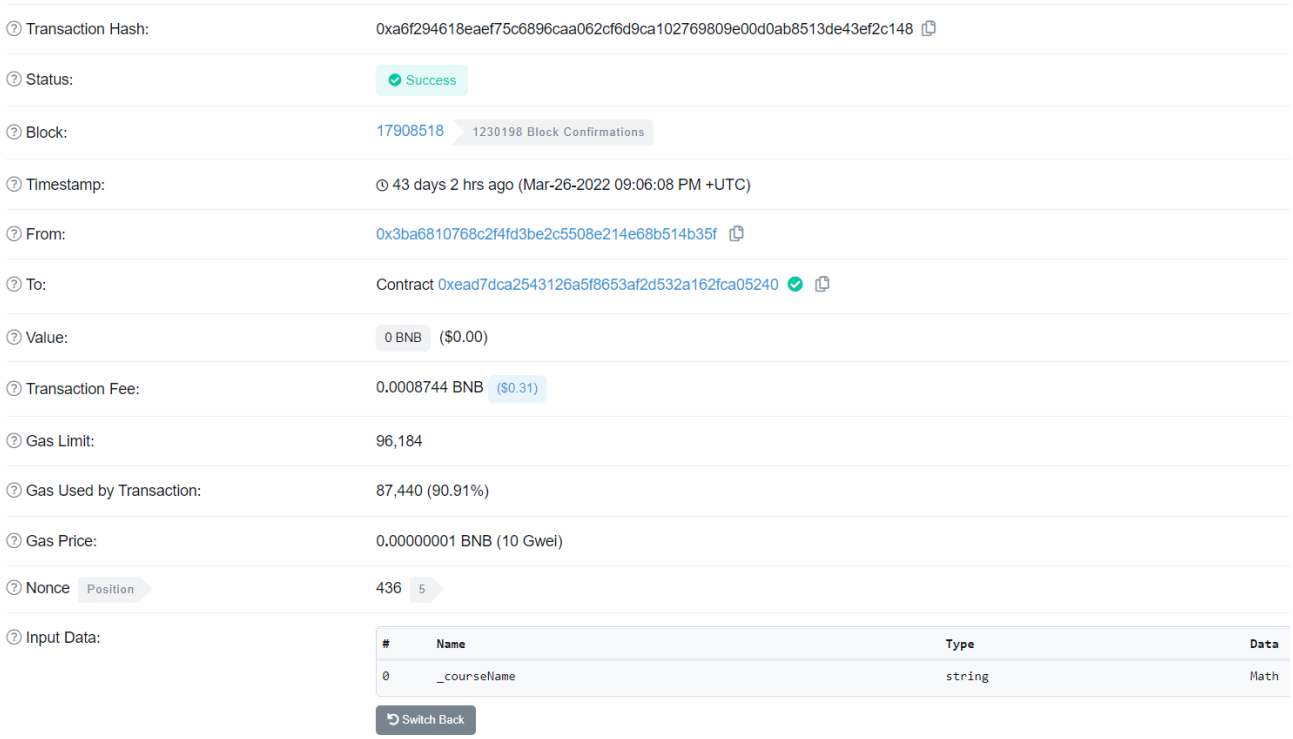

Рисунок 27 – Пример транзакции на создание нового обучающего курса “Math”

Седьмым этапом введения программной системы в эксплуатацию можно считать размещение веб-сайта программной системы в сети Интернет. Перед этим, необходимо заменить адреса развернутых смарт-контрактов в блокчейн-сети в конфигурационном JSON-файле. Данный файл содержит адреса используемых смарт-контрактов и их двоичный интерфейс. Наличие конфигурационного файла необходимо для реализации обращения веб-сайта к смарт-контрактам, развернутым в блокчейн сети.

Пример фрагмента конфигурационного файла веб-сайта с данными о смарт-контрактах в блокчейн-сети приведен на рисунке 28. Поле “abi” представляет из себя двоичный интерфейс приложения, который генерируется автоматически при развертывании смарт-контракта в блокчейн-сети. Данные этого поля определяют такие детали, как то, как вызываются функции смарт-контракта и в каком двоичном формате информация должна передаваться от одного компонента программы к другому.

```
{
  "address": "0xEAD7dcA2543126A5F8653Af2D532A162fCa05240",
  "abi": [
    {
      "inputs": [
        {
          "internalType": "string",
          "name": "_name",
          "type": "string"
        },
        {
          "internalType": "string",
          "name": "_symbol",
          "type": "string"
        },
        {
          "internalType": "address",
          "name": "_nftAddr",
          "type": "address"
        }
      ],
      "stateMutability": "nonpayable",
      "type": "constructor"
    },
```


Рисунок 28 – Пример фрагмента конфигурационного файла веб-сайта программной системы

Первые семь этапов отражали первичное внедрение программной системы. Восьмой этап, создание ERC-721 токенов путем вызова функции смарт-контракта NFT, отражает процесс использования программной системы. Процесс создания токенов сертификатов продолжается до момента вывода программной системы из эксплуатации. Сетевой граф этапов введения программной системы в эксплуатацию представлен на рисунке 29.

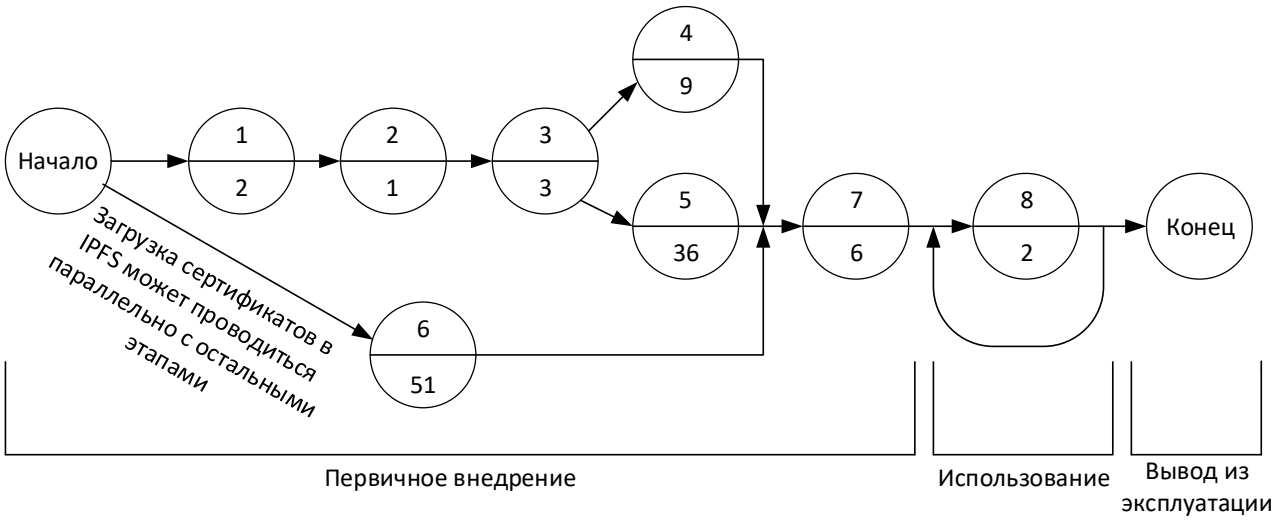

Рисунок 29 – Сетевой граф этапов введения программной системы в эксплуатацию

Вершины графа отражают этапы введения программной системы в эксплуатацию. Номер сверху в вершине графа отражает номер этапа, снизу – длительность этапа в условных единицах. После первичного введения система может быть использована на неограниченную длительность, вплоть до того, как не будет признана морально устаревшей и не будет выведена из эксплуатации.

3.2 Разработка схемы использования программной системы

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Основой построения IDEF0-модели является описание системы в целом и ее компонентов [7].

Если рассматривать всю систему в целом, для пользования ею необходимы следующие входные данные: наименование и аббревиатура образовательной организации, графические файлы изображений сертификатов, список обучающих курсов образовательной организации, список блокчейн-адресов обучающихся на курсах в образовательной организации и метаданные сертификатов, включающие информацию о датах начала курсов, датах выдачи сертификатов, полученных количествах баллов за прохождения курсов, а также дополнительную информацию, добавляемую по усмотрению образовательной организации. Управляющими воздействиями программной системы являются:

стандарт формирования токенов ERC-721 в публичной EVM-совместимой блокчейн-сети, стандарт хэширования SHA-128 и асимметричный криптографический алгоритм ECDSA-secp256k1, который применяется для подписания транзакций в блокчейн-сети. Механизмами, которые отражают, с помощью чего выполняется выдача электронных сертификатов, являются: IPFS-сеть и публичная EVM-совместимая блокчейн-сеть. Выходными данными системы являются сформированные NFT-токены сертификатов обучающихся в блокчейн-сети. Контекстная диаграмма системы представлена на рисунке 29.

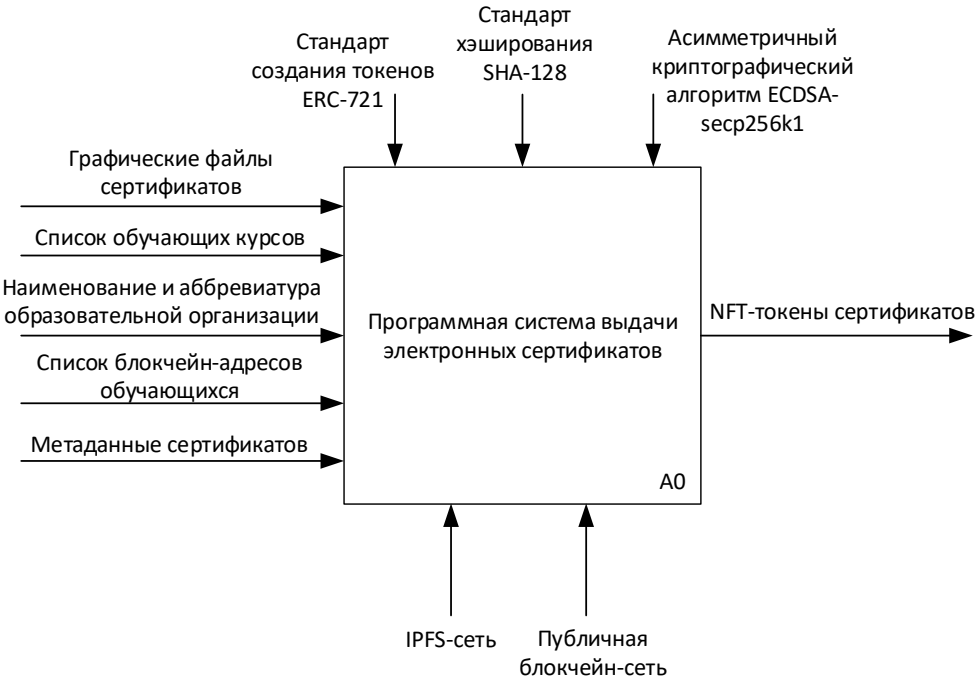

Рисунок 29 – Диаграмма бизнес-процесса использования системы первого уровня

Программная система, с точки зрения пользования ею, состоит из следующих компонентов: развертывание смарт-контрактов в блокчейн-сети, добавление сертификатов в сеть IPFS, формирование списка обучающих курсов в блокчейн-сети, формирования списка обучающихся на курсах образовательной организации в блокчейн-сети, добавление метаданных сертификатов в блокчейн-сеть и создание ERC-721 токенов сертификатов в блокчейн-сети.

После развертывания смарт-контрактов, система получает их адреса в блокчейн-сети, которые используются во всех остальных функциях. Адреса смарт-контрактов необходимы для обращения к конкретным экземплярам.

После развертывания, необходимо сформировать список обучающих курсов в блокчейн сети, добавить на данные курсы блокчейн-адреса обучающихся, а также метаданные, о выданных им сертификатах. Так как все эти функции изменяют состояние блокчейн-сети, лицо, вызывающее их, должно сформировать транзакции, предварительно подписав их своим приватным ключом ECDSA-secp256k1. Без подписи транзакция будет признана недействительной и отклонена сетью. Получив на смарт-контракте записи формата “Обучающий курс – блокчейн-адрес обучающегося – метаданные сертификата”, администратор должен загрузить файлы сертификатов в IPFS-сеть, и, сопоставить каждой записи адрес сертификата в IPFS-сети, представляющий из себя мультихэш SHA-128. Таким образом, будет создан уникальный токен ERC-721, который будет представлять электронный сертификат, выданный обучающемуся. IDEF-0 диаграмма бизнес-процесса использования программной системы представлена на рисунке 30.

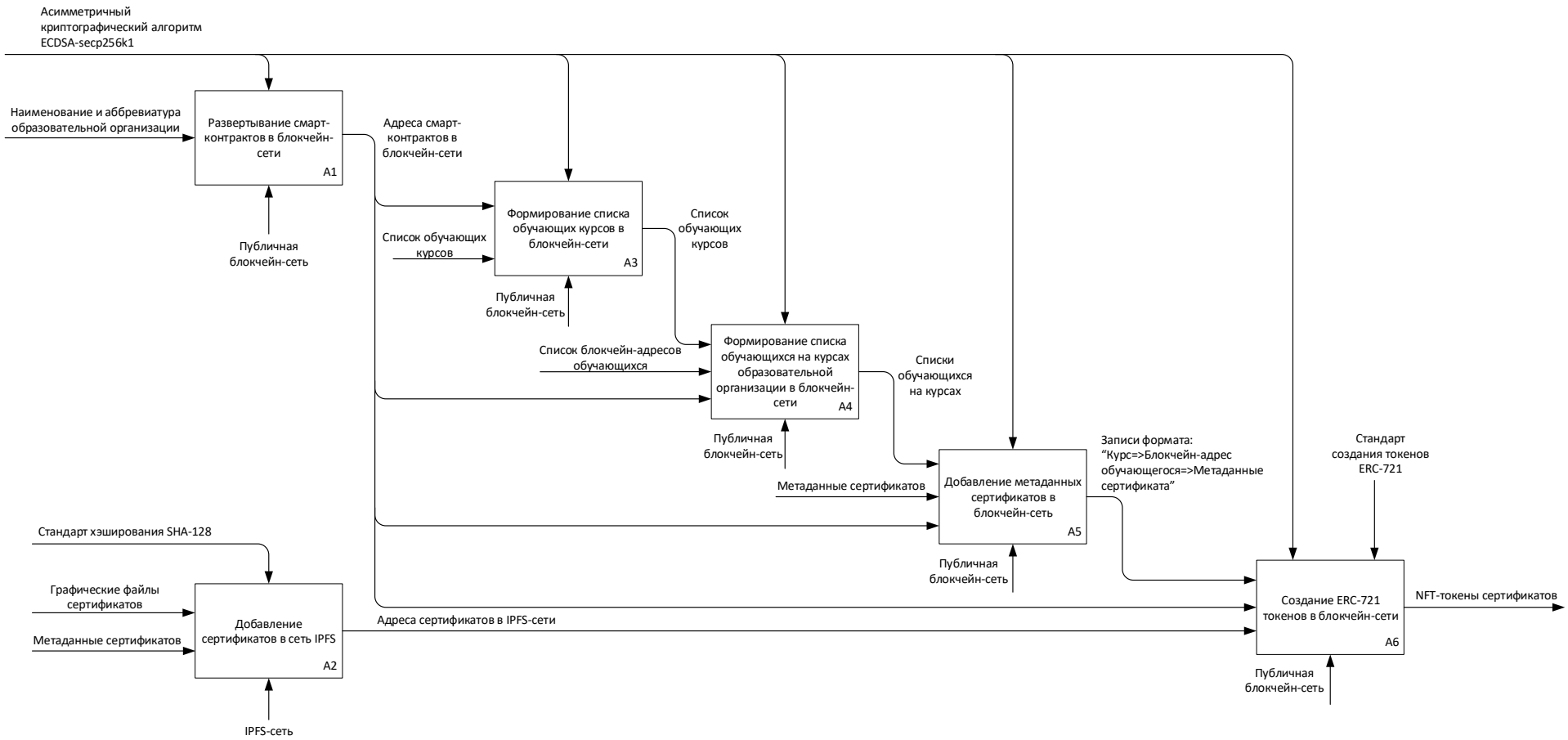

Рисунок 30 - IDEF-0 диаграмма бизнес-процесса использования программной системы второго уровня

Все компоненты, представленные на диаграмме необходимы для корректного взаимодействия с программной системой. После того, как NFT-токен сертификата сформирован, каждый обучающийся может удостовериться в этом, проверив хэш транзакции в сканере блоков публичной блокчейн-сети. Через веб-сайт обучающихся в образовательной организации, станут доступны функции просмотра изображения сертификата, скачивания сертификата на компьютер, получения информации о сертификате и сформированном NFT-токене.

Помимо этого, сертификат автоматически отобразится на вашем аккаунте в любом блокчейн-кошельке. Пример отображения сформированного ERC-721 сертификата в приложении “Metamask” с мобильного устройства представлен на рисунке 31.

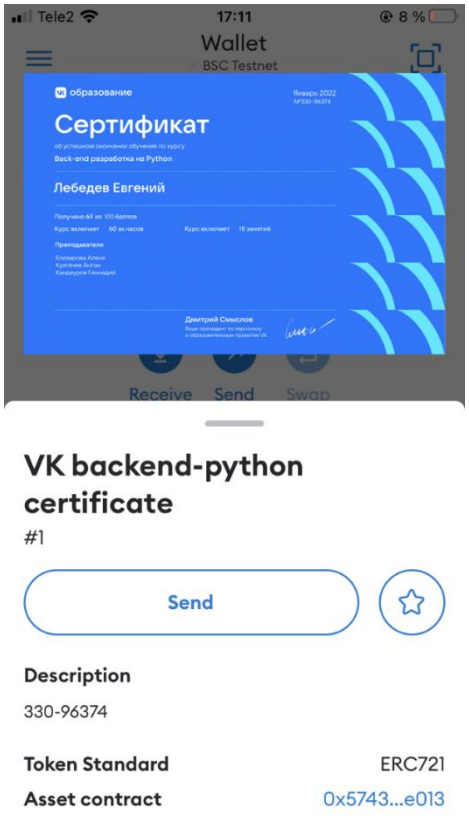

Рисунок 31 – Пример отображения сформированного ERC-721 сертификата в приложении “Metamask” с мобильного устройства

Описание токена полученного сертификата содержит в себе указание стандарта ERC-721, данные о сертификате из сети IPFS, а также адрес смарт-контракта в блокчейн-сети.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы бакалавра был разработана программная система выдачи электронных сертификатов. Программная система выдачи электронных сертификатов разрабатывалась с использованием блокчейн-технологий, что позволило обеспечить отказоустойчивость и надежность системы.

В ходе работы был проведен анализ предметной области, определены технологии, язык и среда разработки программной системы, разработана диаграмма вариантов использования программной системы, создана диаграмма классов системы концептуального уровня, построена схема алгоритма создания цифрового сертификата, разработаны оконные формы ввода-вывода информации, граф состояний интерфейса, структурная схема программного продукта, спроектирована структура и разработаны компоненты программного продукта. Помимо этого, было выполнено комплексное тестирование программной системы и разработана технология ее использования.

В процессе разработки были выполнены все задачи, поставленные в техническом задании, программная система выдачи электронных сертификатов способна работать в соответствии с предъявленными требованиями к ее функционалу.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Morabito V. Business Innovation Through Blockchain. Springer. 2017. 173 p.
2. Raval S. Decentralized Applications: Harnessing Bitcoin’s Blockchain Technology. Pearson. 2017. 47 p.
3. Non-fungible token (NFT): основы [Электронный ресурс]. – 2022. – URL: https://habr.com/ru/post/579908/ (дата обращения 15.02.2022).
4. Официальная документация библиотеки QT [Электронный ресурс]. – 2022. – URL: https://doc.qt.io/ (дата обращения 15.02.2022).
5. Andreas M. Antonopoulos Mastering Bitcoin: Unlocking Digital Cryptocurrencies. Integral. 2017. 84 p.
6. Nishara Nizamuddin, Haya R. Hasan, Khaled Salah. IPFS-Blockchain-based Authenticity of Online Publications. ICBC. 2018. 15 p.
7. IDEF-0. Знакомство с нотацией и пример использования [Электронный ресурс]. – 2022. – URL: https://itnan.ru/post.php?c=1&p=322832 (дата обращения 25.04.2022).

ПРИЛОЖЕНИЕ А

Техническое задание

Листов 10

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ

Заведующий кафедрой ___ИУ6___

_____ А.В. Пролетарский

« __ » _____ 2022 г.

ПРОГРАММНАЯ СИСТЕМА ВЫДАЧИ ЭЛЕКТРОННЫХ СЕРТИФИКАТОВ

Техническое задание

Листов 10

Студент ИУ6-82Б _____ Е.В. Лебедев
(Группа) (Подпись, дата) (И.О. Фамилия)

Руководитель _____ В.В. Гуренко
(Подпись, дата) (И.О. Фамилия)

2022 г.

1 ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку программной системы выдачи электронных сертификатов, используемой для создания цифровых аналогов существующих бумажных сертификатов в сфере образования по технологии блокчейн.

В настоящее время в сфере образования существуют проблемы, касающиеся выдачи обучающимся документов об окончании обучения. Основные из них: мошенничество в академической среде, связанное с созданием фальсификаций и подделок сертификатов об обучении, сложность верификации документов об образовании при устройстве на работу, возможность привести документ об образовании в негодный вид. Необходимо создать программную систему, способную решить данные проблемы, поддерживая работоспособность и доступность для пользователей.

2 ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

Программная система выдачи электронных сертификатов разрабатывается в соответствии с тематикой кафедры.

3 НАЗНАЧЕНИЕ РАЗРАБОТКИ

Основное назначение программной системы выдачи электронных сертификатов заключается в организации создания сертификатов об обучении образовательным учреждением в цифровом виде с помощью использования технологии блокчейн, гарантирующих пожизненную проверяемость документа об образовании. Система предназначена для двух категорий пользователей: администрации образовательного учреждения и обучающихся.

4 ИСХОДНЫЕ ДАННЫЕ, ЦЕЛИ И ЗАДАЧИ

4.1 Исходные данные

4.1.1 Исходными данными для разработки являются следующие материалы:

4.1.1.1 Перечень работ или письменных документов, содержащих исходные данные для разработки:

– описание функционального назначения программной системы выдачи электронных сертификатов;

– описание эксплуатационного назначения программной системы выдачи электронных сертификатов.

4.1.1.2 Прототипы программной системы:

- MIT Hyland Credentials (https://www.hylandcredentials.com/);
- Credentia (https://credentia.ru/);
- DNV (https://www.dnv.com/).

4.2 Цель работы

Целью работы является прототип программной системы выдачи электронных сертификатов.

4.3 Решаемые задачи

4.3.1 Анализ эффективности существующих методов выдачи электронных сертификатов образовательными учреждениями, предоставляющими услуги дополнительного образования.

4.3.2 Определение критериев эффективности систем выдачи электронных сертификатов.

4.3.3 Определение требований к разрабатываемой программной системе.

4.3.4 Определение технологий, языка и среды разработки разрабатываемой системы.

4.3.5 Разработка структуры системы.

4.3.6 Проектирование и реализация компонентов системы с использованием выбранных средств разработки.

4.3.7 Комплексное тестирование полученной системы.

4.3.8 Разработка технологии использования программной системы.

5 ТРЕБОВАНИЯ К ПРОГРАММЕ

5.1 Требования к функциональным характеристикам

5.1.1 Выполняемые функции

5.1.1.1 Для пользователя (обучающееся лицо):

- аутентификация на веб-сайте программной системы с помощью адреса в публичной блокчейн-сети;
- получение изображения сертификата, выданного по итогам обучения и зарегистрированного в публичной блокчейн-сети;
- скачивание файла с изображением сертификата на компьютер;
- получение метаданных о NFT-токене сертификата (адрес смарт-контракта ERC-721, наименование блокчейн-сети, уникальный идентификатор сертификата) из публичной блокчейн-сети;
- получение метаданных, связанных с сертификатом об обучении, из публичной блокчейн-сети (наименование пройденного курса, дата начала курса, дата выдачи сертификата, полученное количество баллов и прочая необходимая информация, добавляемая по усмотрению образовательной организации).

5.1.1.2 Для администратора (образовательная организация):

- развертывание смарт-контрактов, обеспечивающих бизнес-логику программной системы, в публичной блокчейн-сети;

- редактирование списка блокчейн-адресов обучающихся в образовательной организации;
- добавление нового обучающего курса в смарт-контракт;
- добавление изображения цифрового сертификата в распределенную файловую систему IPFS;
- сохранение всех необходимых метаданных о выданном цифровом сертификате в блокчейн-сети;
- создание из файла сертификата его уникального цифрового аналога, который представляет собой невзаимозаменяемый токен (NFT) по стандарту ERC-721 в публичной блокчейн-сети.

5.1.2 Исходные данные:

- список лиц, обучающихся в образовательной организации;
- список курсов, проводимых образовательной организацией;
- метаданные, связанные со всеми обучающимся в рамках курса (полученное количество баллов, дата выдачи сертификата и прочая дополнительная информация);
- сформированные файлы сертификатов, выдаваемых обучающимся, в любом текстовом или графическом расширении (pdf, docx, jpg, png и т.д.).

Максимально допустимое время ответа системы не более 40 с.

Максимальный объём используемой оперативной памяти не более 200 МБ.

Максимальный объём используемой внешней памяти не более 200 МБ.

5.2 Требования к надежности

5.2.1 Предусмотреть контроль вводимой информации.

5.2.2 Предусмотреть защиту от некорректных действий пользователя.

5.3 Условия эксплуатации

5.3.1 Условия эксплуатации в соответствие с СанПиН 2.2.2/2.4.1340-03.

5.3.2 Обслуживание

Специальное обслуживание не требуется.

5.3.3 Обслуживающий персонал

Обслуживающий персонал не требуется.

5.4 Требования к составу и параметрам технических средств

5.4.1 Программное обеспечение должно функционировать на IBM-совместимых персональных компьютерах.

5.4.2 Минимальная конфигурация технических средств:

5.4.2.1 Процессор..... Intel Core i3.

5.4.2.2 Объем ОЗУ ..... 200 Мб.

5.4.2.3 Объем внешней памяти..... 200 Мб.

5.5 Требования к информационной и программной совместимости

5.5.1 Программное обеспечение должно работать под управлением операционных систем семейств Unix-подобных.

5.6 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

5.7 Требования к транспортированию и хранению

Требования к транспортировке и хранению не предъявляются.

5.8 Специальные требования

Специальные требования не предъявляются.

6 ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

6.1 Разрабатываемые программные модули должны быть самодокументированы, т.е. тексты программ должны содержать все необходимые комментарии.

6.2 Разрабатываемое программное обеспечение должно включать справочную систему.

6.3 В состав сопровождающей документации должны входить:

6.3.1 Расчетно-пояснительная записка на 55-65 листах формата А4 (без приложений).

6.3.2 Техническое задание (Приложение А).

6.3.3 Руководство пользователя (Приложение Б).

6.3.4 Исходный текст программного модуля системы (Приложение В).

6.4 Графическая часть должна быть выполнена на 6 листах формата А1 и содержать следующие схемы, графы, диаграммы (копии формата А3/А4 включить в качестве приложений к расчетно-пояснительной записке):

6.4.1 Схема структурная информационной системы – лист А1.

6.4.2 Диаграммы вариантов использования – лист А2.

6.4.3 Диаграммы классов предметной области и интерфейсной части, концептуального уровня и уровня реализации – лист А1.

6.4.4 Основные схемы алгоритмов программной системы – лист А2.

6.4.5 Графы состояний интерфейса – лист А2.

6.4.6 Графы диалогов – лист А2.

6.4.7 Формы интерфейса – лист А1.

6.4.8 Диаграмма компоновки программной системы – лист А2.

6.4.9 Таблицы тестов – лист А2.

6.4.10 Диаграммы бизнес-процесса использования программной системы – лист А2.

7 ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

Выполнить технико-экономическое обоснование разработки.

8. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

№	Название этапа	Срок, даты, %	Отчетность
1	2	3	4
1.	Разработка технического задания	2.02.2022 - 28.02.2022 5 %	Утвержденное техническое задание и задание на выпускную квалификационную работу
2.	Анализ требований и уточнение спецификаций (эскизный проект)	28.02.2022 – 14.03.2022 15%	Спецификации программного обеспечения.
3.	Проектирование структуры программного обеспечения, проектирование компонентов (технический проект)	14.03.2022 – 31.03.2022 35%	Схема структурная системы и спецификации компонентов. Проектная документация: схемы, диаграммы и т.п.
4.	Реализация компонентов и автономное тестирование компонентов. Сборка и комплексное тестирование.	31.03.2022 – 20.04.2022 30%	Тексты программных компонентов.

№	Название этапа	Срок, даты, %	Отчетность
1	2	3	4
	Оценочное тестирование и (рабочий проект).		Тесты, результаты тестирования.
5.	Разработка документации.	20.04.2022 – 25.05.2022 8 %	Расчетно-пояснительная записка.
6.	Прохождение нормоконтроля, проверка на антиплагиат, получение рецензии, подготовка доклада и предзащита.	25.05.2022-6.06.2022 5 %	Иллюстративный материал, доклад, рецензия, справки о нормоконтроле и проценте плагиата.
7.	Защита выпускной квалификационной работы.	1.06.2022-04.07.2022 2 %	

9 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

9.1 Порядок контроля

Контроль выполнения осуществляется руководителем еженедельно.

9.2 Порядок защиты

Защита осуществляется перед государственной экзаменационной комиссией (ГЭК).

9.3 Срок защиты

Срок защиты определяется в соответствии с планом заседаний ГЭК.

10 ПРИМЕЧАНИЕ

В процессе выполнения работы возможно уточнение отдельных требований технического задания по взаимному согласованию руководителя и исполнителя.

ПРИЛОЖЕНИЕ Б

Руководство пользователя

Листов 10

1 Установка и запуск программы

Для установки программной подсистемы администрирования со стороны образовательной организации необходимо скачать содержимое репозитория по ссылке “https://github.com/catcatcat8/vkrb” с официального сайта GitHub. В случае, если у вас на компьютере не установлен язык Python или пакетный менеджер pip, необходимо выполнить команды “apt –y install python3.9” и “ apt -y install python3-pip” соответственно. После этого необходимо выполнить команду “pip install –r requirements.txt” из главного каталога скачанной папки для установки зависимостей Python.

После установки зависимостей нужно настроить программную подсистему администрирования перед запуском. Для этого необходимо зайти в файл secrets.ts и заменить приватный ключ от блокчейн-аккаунта, находящийся в поле “prod” на свой. После этого в файле hardhat.config.ts необходимо заменить поле “MY_WALLET” на публичный адрес вашего блокчейн-аккаунта. После совершения вышеперечисленных действий программная подсистема администрирования полностью готова к запуску. Для запуска программы необходимо выполнить команду “python3 ./admin.py” из главного каталога скачанной папки.

После этого можно перейти к настройке веб-сайта для обучающихся в образовательной организации. Необходимо выполнить команду “yarn install” из главного каталога для установки JavaScript и TypeScript зависимостей. Последним шагом является замена адресов смарт-контрактов в конфигурационном файле веб-сайта. Адреса смарт-контрактов в блокчейн-сети после развертывания будут автоматически сохранены в поля “address” файлов “NFT.json” и “CertificatesLogic.json” каталога “deployments”, а также выведены на экран в случае успеха.

После совершения всех вышеперечисленных действий программная подсистема администрирования для образовательной организации и веб-сайт обучающихся полностью готовы к работе.

2 Инструкция по работе с программой для администратора

После запуска программной подсистемы администрирования открывается ее главное меню, в котором можно выполнить три действия: открыть меню основных функций программы, получить справку по работе программы, выйти из программы. Начальный экран программы представлен на рисунке Б.1.

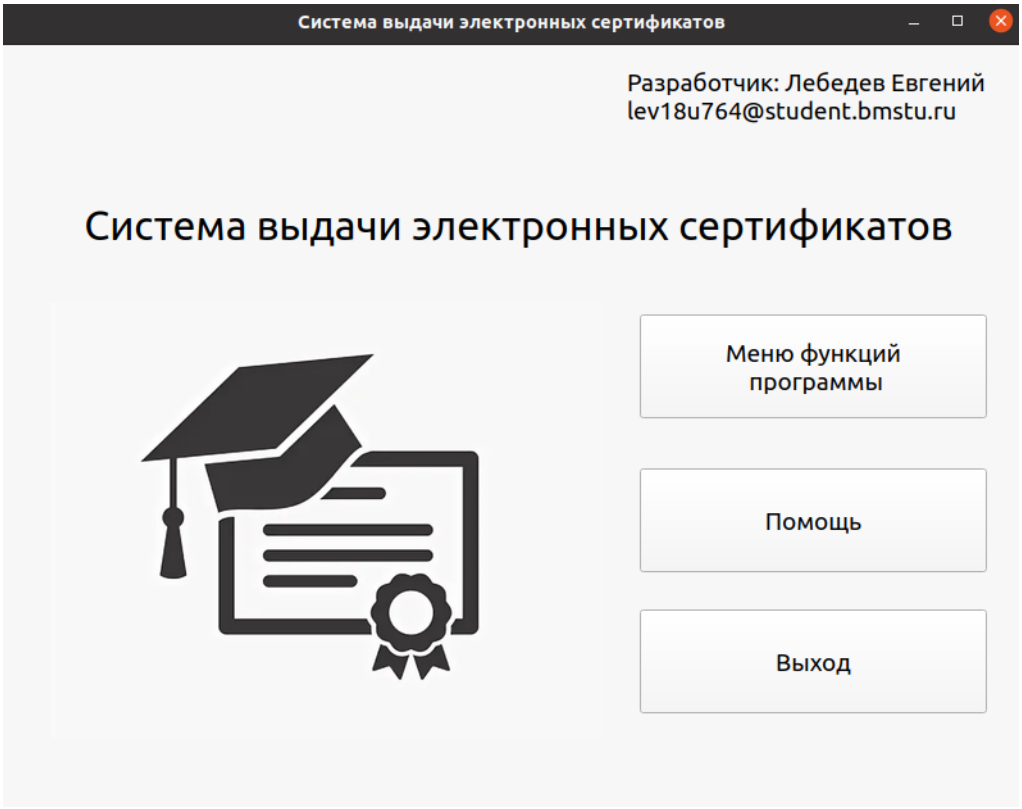

Рисунок Б.1 – Начальный экран программы

При нажатии на кнопку “Меню функций программы” произойдет переход в основное меню программы, в котором в первую очередь требуется выполнить развертывание смарт-контрактов в блокчейн-сети. Для этого необходимо нажать на кнопку “Развертывание смарт-контрактов в блокчейн-сети”, после этого ввести наименование образовательной организации и ее аббревиатуру в соответствующие поля и нажать на кнопку “Развернуть смарт-контракты”. Пример развертывания смарт-контрактов в блокчейн сети представлен на рисунке Б.2.

В случае успешного развертывания, на экран будут выведены адреса смарт-контрактов, загруженных в блокчейн-сеть, количество затраченного газа

и прочая дополнительная информация. Пример вывода информации об успешном развертывании смарт-контрактов в блокчейн-сети представлен на рисунке Б.3.

После того, как смарт-контракты загружены в блокчейн-сеть, необходимо создать обучающие курсы. Это можно сделать из меню функций программы, нажав на кнопку “Создать новый курс”. Произойдет переход на форму создания нового курса, в которой необходимо ввести его наименование, количество академических часов и дату его начала. После ввода этих данных, нужно нажать на кнопку “Создать образовательный курс”. Пример создания нового курса “Math” представлен на рисунке Б.4.

В случае успешного создания нового образовательного курса на экран будет выведена информация о сформированной транзакции в блокчейн-сети, с указанием вызванной функции смарт-контракта и затраченного на нее газа. Пример вывода информации об успешном создании нового образовательного курса представлен на рисунке Б.5.

После создания образовательных курсов, можно приступать к добавлению обучающихся на них. Для этого из главного меню необходимо нажать на кнопку “Добавить/удалить обучающегося”, после этого вписать наименование курса, на которого планируется добавить обучающегося, блокчейн-адрес обучающегося и дату начала учебного курса и нажать кнопку “Добавить обучающегося”. Дата начала курса необходима для того, чтобы обучающийся мог находиться в одинаковых по наименованию курсах в разные промежутки времени. В случае успешного добавления обучающегося на курс будет выведена информация о совершенной транзакции в блокчейн-сети, по структуре аналогичная примеру с добавлением обучающего курса. Пример добавления обучающегося на курс представлен на рисунке Б.6.

В данной оконной форме также можно удалить обучающего с курса. Необходимо вписать данные, как в случае добавления обучающего на курс, и нажать кнопку “Удалить обучающегося”. Удалить обучающего возможно только если он уже был добавлен на данный обучающий курс ранее.

Для добавления описания сертификата, выдаваемого обучающемуся, необходимо нажать на кнопку “Создать описание сертификата” в меню функций программы. Откроется оконная форма с полями для ввода данных, в которые необходимо вписать наименование обучающего курса, блокчейн-адрес обучающегося, дату начала курса, дату выдачи сертификата, полученное количество баллов обучающимся на курсе. По усмотрению образовательной организации, можно вписать дополнительную информацию в соответствующее поле оконной формы. После ввода необходимых данных, требуется нажать на кнопку “Добавить описание в блокчейн-сеть”. Пример добавления описания сертификата в блокчейн-сеть представлен на рисунке Б.7.

Для создания NFT-токена сертификата, необходимо добавить изображение сертификата в IPFS-сеть. Для этого необходимо нажать на кнопку “Добавить сертификат в IPFS” в меню функций программы. После открытия оконной формы необходимо нажать на кнопку “Выбрать изображение сертификата” и выбрать изображение сертификата с компьютера. Предпочитаемым форматом изображения является формат “jpg”, однако поддерживаются все форматы. После выбора изображения сертификата, необходимо нажать на кнопку “Добавить выбранный сертификат в IPFS”. После нажатия на кнопку в поле “Адрес добавленного сертификата” будет выведен его IPFS-хэш. После добавления изображения сертификата в IPFS-сеть необходимо добавить JSON-файл с метаданными сертификата в IPFS-сеть аналогичным образом. Необходимо наличие полей “Name”, “Description” и “Image” по стандарту формирования ERC-721 токена сертификата. Пример правильно сформированного JSON-файла сертификата для добавления в IPFS-сеть представлен на рисунке Б.8.

Последним шагом является создание NFT-токена сертификата в блокчейн-сети. Для этого необходимо нажать на кнопку “Создать NFT-сертификат” в меню функций программы, и, в открывшейся форме вписать наименование курса, блокчейн-адрес обучающегося и IPFS-адрес (хэш) сертификата в соответствующие поля, после чего нажать на кнопку “Создать NFT-сертификат”. Пример создания NFT-сертификата представлен на рисунке Б.9.

В случае успешного создания NFT-сертификата, более подробную информацию о совершенной транзакции в блокчейн-сети можно посмотреть на сайте “https://etherscan.io/” или “https://bscscan.com/” в зависимости от используемой блокчейн сети Ethereum или Binance Smart Chain. Пример информации о транзакции на создание NFT-сертификата в блокчейн-сети Binance Smart Chain представлен на рисунке Б.10.

В случае попытки добавления уже существующего курса, одного и того же обучающего на один и тот же курс, добавления обучающегося на несуществующий курс, удаления обучающего с несуществующего курса, создания второго экземпляра NFT-сертификата обучающему на один курс, а также при вводе недопустимых символов в поля ввода на экран будет выведено сообщение об ошибке совершения транзакции в блокчейн сети. Пример неудачной транзакции (добавление обучающего на несуществующий курс) представлен на рисунке Б.11.

Помимо этого, предусмотрена справка по работе программы. Для более детального ознакомления по работе с программой необходимо нажать на кнопку “Помощь” в главном меню.

3 Инструкция по работе с веб-сайтом для обучающегося

На странице аутентификации веб-сайта находится кнопка “Войти с Metamask”. Для попадания на главную страницу веб-сайта необходимо нажать

на данную кнопку, в появившемся окне расширения Metamask ввести пароль от блокчейн-кошелька. Необходимо скачать расширение Metamask из официального магазина расширений браузера, в случае его отсутствия. Пример аутентификации на веб-сайте представлен на рисунке Б.12.

На главной странице веб-сайта пользователю доступны три функции: получение изображения сертификата (с возможностью скачивания его на компьютер), загрузка информации о NFT-токене сертификата и загрузка информации о пройденном курсе. Для получения изображения сертификата пользователь должен ввести наименование курса в поле “Название курса” левой карточки и нажать на кнопку “Загрузить сертификат”. В случае отсутствия сертификата пользователю будет выведено соответствующее сообщение. В случае успеха, на экран будет выведено наименование сертификата и его описание из IPFS-сети. В этой же карточке появятся кнопки “Скачать сертификат” и “Загрузить другой сертификат”. Пример успешной загрузки сертификата представлен на рисунке Б.13.

При нажатии на кнопку “Скачать сертификат” в другой вкладке появится изображение, выданного обучающемуся сертификата, с возможностью скачивания. Пример полученного сертификата представлен на рисунке Б.14.

Для загрузки информации о NFT-токене сертификата необходимо ввести наименование обучающего курса в центральную карточку веб-сайта и нажать на кнопку “Загрузить информацию о NFT-токене”. В случае успеха, на экран будет выведено наименование блокчейн-сети, адрес смарт-контракта ERC-721, хранящего NFT-токены сертификатов, а также уникальный идентификатор токена. Пример получения информации о NFT-токене сертификата представлен на рисунке Б.15.

Для загрузки информации о пройденном курсе (с информацией о полученном сертификате), в правую карточку главной страницы необходимо добавить наименование обучающего курса и нажать на кнопку “Загрузить информацию о пройденном курсе”. В случае успеха, на экран будут выведены все данные, связанные с пройденным курсом и полученным, в рамках него, сертификатом. Пример загрузки информации о полученном сертификате представлен на рисунке Б.16.

Стоит отметить, что все функции, которые доступны пользователям на веб-сайте, не требуют внесения средств для совершения транзакций в блокчейн-сети, так как не являются функциями, изменяющими блокчейн-сеть. Таким образом, через веб-сайт невозможно изменить никакую информацию о выданных сертификатах, хранящуюся в блокчейн-сети.