

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)
Кафедра Информационных систем и программной инженерии

Отчет по преддипломной практике

Выполнил:
Ст. гр. ИСТ-114
Штых А.Д
Принял:
Кириллова С.Ю.

Владимир 2017

Оглавление	
ВВЕДЕНИЕ	3
АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	5
1 Глоссарий	5
2 Обзор аналогов	6
ПРОЕКТИРОВАНИЕ.....	8
1 Составление технического задания на проектирование	8
2 Проектирование серверной части	12
3 Проектирование клиентской части	14
4 Проектирование пользовательского интерфейса	15
5 Разработка мероприятий по повышению надежности и достоверности выдаваемой информации.....	25
РЕАЛИЗАЦИЯ	28
1 Реализация серверной части	28
2 Тестирование контракта	29
3 Реализация клиентской части	30
4 Тестирование клиентской части	32
ИНФОРМАЦИОННЫЙ МЕНЕДЖМЕНТ	34
3 SWOT-анализ ИС ДРДО	34

ВВЕДЕНИЕ

В последнее время информационные технологии шагнули так далеко вперед, что с каждым годом обстановка в мире начинает соответствовать критериям информационного общества. Все большая часть населения задействована в производстве информационных услуг, а производство товаров становится автоматизированным. При таких условиях государство вынуждено подстраиваться под развивающегося гражданина, создавать сервисы и системы, при которых поиск информации, оплата каких-либо начислений и подтверждение данных при поступлении на работу осуществлялись бы дистанционно.

В пример можно привести портал Госуслуг, который позволяет пользователю не только просматривать персональные данные, но и дистанционно оплачивать штрафы и налоги, записываться на прием к врачу, оформлять различные виды документов.

Одновременно с этим перед гражданами возникает проблема доверия тем данным, которые предоставляют им другие граждане. У работодателей все чаще возникает вопрос о подлинности документов об образовании (дипломов и сертификатов) соискателей на вакантные должности. При нынешнем уровне информатизации государственных структур работодатель вынужден запрашивать данные в Федеральной службе по надзору в сфере образования и науки или напрямую обращаться с письменным запросом в ВУЗы или другие учебные заведения. Естественно, процесс получения этих данных растягивается не на дни, а на недели и месяцы. Особенно актуальна данная ситуация для медицинских учреждений, правоохранительных органов, учебных заведений и других госучреждений.

Одной из наиболее надежных и безопасных технологий хранения данных в больших количествах в настоящее время является технология Blockchain. Наиболее перспективные направления его использования лежат в не-финансовой сфере. Многие страны уже сейчас задействуют эту технологию в системах, предназначенных для здравоохранения, документооборота,

хранения и подтверждения всевозможных прав. Например, в Нидерландах, Эстонии, США внедряются единые национальные реестры медицинских карт, построенные на blockchain'e [1].

Ввиду этих мировых тенденций и сформулированной выше проблемы было принято решение о создании новой прозрачной системы, которая позволит:

- безопасно хранить данные о документах об образовании;
- быстро получать эти данные;
- гарантировать подлинность и неизменность объектов в системе.

Были поставлены следующие задачи:

- обзор аналогов и конкурентов
- проектирование и разработка децентрализованного хранилища данных
- разработка соглашений о добавлении и считывании данных из хранилища
- реализация удобного пользовательского интерфейса для доступа к данным о документах
- анализ системы с точки зрения информационного менеджмента

АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1 Глоссарий

ИС ДРДО – информационная система «Децентрализованный реестр документов об образовании» - рассматриваемая в данной работе информационная система.

Blockchain – распределенное хранилище данных, построенное по определенным правилам объединения блоков в цепочки.

Смарт-контракт – обработчик транзакций, предназначенный для определения правил изменения состояния блокчейна.

Ethereum – платформа для создания децентрализованных онлайн-сервисов на базе блокчейна.

Solidity – язык виртуальной машины Ethereum, предназначенный для написания умных контрактов.

ЭЦП – электронно-цифровая подпись – реквизит электронного документа, позволяющий отследить подлинность подписанного ею файла.

Документ об образовании – документ, подтверждающий прохождение соискателем определенных курсов, дипломы, сертификаты и т.д. Характеризуется следующими параметрами:

- серия и номер
- владелец
- отправитель (учебное заведение, выпустившее данный документ)
- дата выдачи
- квалификация
- специальность

Соискатель — обладатель документа об образовании, который хочет подтвердить подлинность своего диплома или т.п для получения работы. Может быть владельцем документов об образовании.

Работодатель – организация или физическое лицо, которое хочет удостовериться в подлинности документа соискателя.

Учебное заведение – заведение, обладающее правами выпускать документы об образовании. Может выступать в качестве отправителя документа.

На рис. 1 приведено взаимоотношение основных объектов предметной области.

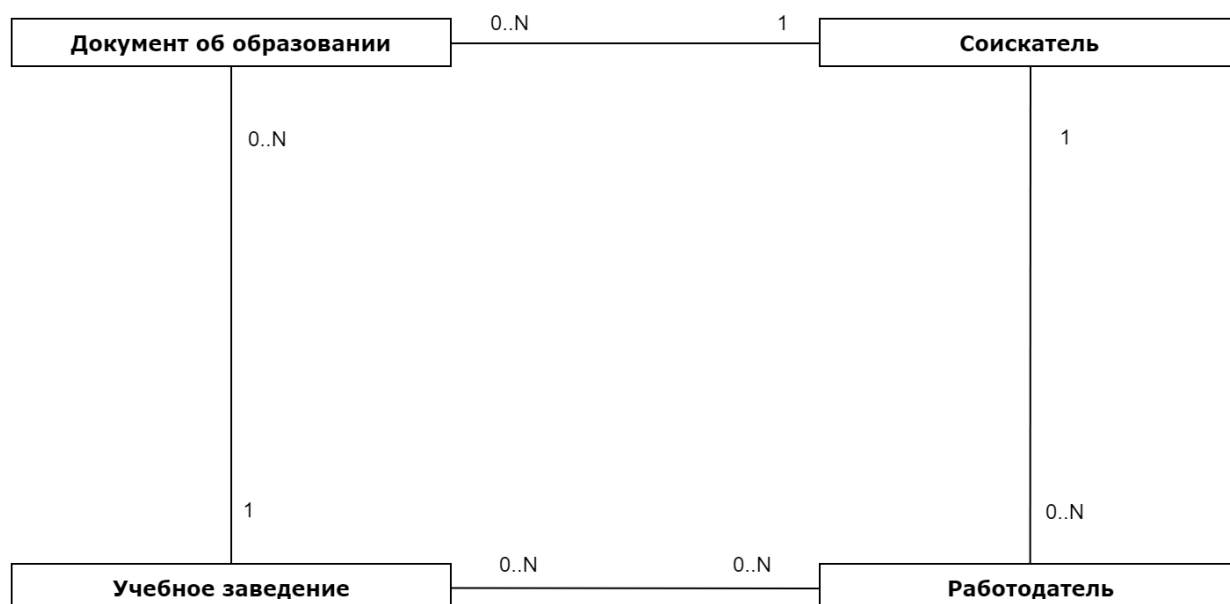


Рисунок 1 – Концептуальная модель данных ИС ДРДО

2 Обзор аналогов

На основании частей 9 и 10 статьи 98, пункта 2 части 15 статьи 107 Федерального закона от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации», и постановления Правительства Российской Федерации от 26 августа 2013 г. № 729. «О федеральной информационной системе «Федеральный реестр сведений о документах об образовании и (или) о квалификации, документах об обучении», Федеральная служба по надзору в сфере образования и науки осуществляет формирование и ведение Федерального реестра сведений о документах об образовании и (или) о квалификации, документах об обучении.

Целями создания Федерального реестра являются:

- Ликвидация оборота поддельных документов государственного образца об образовании

- Обеспечение ведомств и работодателей достоверной информацией о квалификации претендентов на трудоустройство
- Сокращение числа нарушений и коррупции в образовательных учреждениях
- Повышение качества образования за счет обеспечения общественности достоверной информацией о выпускниках

В рамках проектной деятельности Рособрнадзором создана Автоматизированная система формирования и ведения ФРДО (ФИС ФРДО), обеспечивающая сбор сведений о выданных документах с образовательных учреждений, накопление этих сведений в единой базе данных. [2]

Система доступна по следующему адресу:
<http://frdocheck.obrnadzor.gov.ru/>

Данная система отличается тем, что просто загрузить документ в реестр невозможно: необходимо пройти не только процедуру регистрации учебного заведения, но и получения ЭЦП. В свою очередь, ИС ДРДО позволяет упростить процесс регистрации за счет модернизированной системы прав и доверия.

Второй недостаток – полностью закрытый процесс добавления и поиска документов в федеральном реестре. Это способствует повышению уровня коррупции в данной области и увеличения числа нарушений при добавлении документов. В ИС ДРДО все транзакции записываются и доступны всем участникам блокчейна. В такой прозрачной системе легче предотвратить намеренные и случайные ошибки при добавлении документа.

И третье – следуя из того, что все транзакции (они же изменения) фиксируются в блокчейне, невозможно незаметно подменить объект. Это гарантирует подлинность внесенных данных о документе об образовании.

ПРОЕКТИРОВАНИЕ

1 Составление технического задания на проектирование

2.1 Общие сведения

2.1.1. Полное наименование системы и ее условное обозначение

Информационная система «децентрализованный реестр документов об образовании»

2.1.2. Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты:

Инициативный проект

Компания разработчика: ИП Штых А.Д.

2.1.3. Перечень документов, на основании которых создается система, кем и когда утверждены эти документы

Разработка Технического задания проводилась с использованием следующих стандартов:

ГОСТ 34.601-90 Автоматизированные системы. Стадии создания;

ГОСТ 34.602-89 Техническое задание на создание автоматизированной системы;

ГОСТ 34.201-89 Виды, комплектность и обозначение документов при создании автоматизированных систем (частичное использование);

ГОСТ 24.104-85 Автоматизированные системы управления. Общие требования;

ГОСТ Р 56214-2014/ISO/TS 8000-1:2011 Качество данных. Часть 1. Обзор.

2.1.4. Плановые сроки начала и окончания работы по созданию системы

Начало: апрель 2017

Конец: апрель 2018

2.2 Назначение и цели создания системы

Система предназначена для использования учебными заведениями в целях добавления данных документов об образовании в реестр, соискателями на вакантные должности и работодателями для подтверждения подлинности документов об образовании.

Система так же может использоваться другими органами, юридическими и физическими лицами для этих же целей.

Цель проектирования состоит в уменьшении времени, затрачиваемого на получение подтверждения подлинности документов об образовании, уменьшении затрат на процессы подтверждения подлинности.

2.4 Требования к системе

Продукт должен позволять совершать авторизацию пользователю по логину и паролю, давать доступ к приватному ключу и публичному адресу, получать документы для авторизованного пользователя и добавлять документы.

Use case'ы представлены на рис. 2.1.



Рисунок 2.1 - Диаграмма прецедентов ИС Децентрализованный реестр документов об образовании

Таблица 1 – Описание функций системы ДРДО

Название функции	Входные параметры	Выходные параметры	Описание
Регистрация	Данные о пользователе	Результат регистрации	Для соискателя и учебного заведения будет полезно предоставить дополнительные данные, кроме публичного адреса: ФИО, название ВУЗа и т.д. Или хотя бы определить себе одну (или несколько) из ролей.
Добавить документ	Данные о документе	Запись в реестре о документе	Пользователь добавляет данные о документе в реестр
Предоставить доступ к документу	Документ, пользователь	Результат работы функции	Для того, чтобы работодатель смог просматривать документ соискателя, тот может назначить работодателю право на просмотр выбранного документа
Назначить владельца документа	Документ, пользователь	Результат работы функции	При добавлении владелец документа может быть не зарегистрирован в системе. После регистрации он может

			запросить у отправителя назначить себя в качестве владельца документа.
Добавить УЗ в список проверенных	Пользователь	Результат работы функции	Чтобы пользоваться только проверенными источниками записей в реестре, работодатель должен составить список пользователей, документам которых он доверяет.
Поиск документа	Параметр(ы) документа	Запись в реестре	Работодатель может осуществить поиск по разным параметрам документа

При разработке должны быть использованы следующие технологии и инструменты:

- платформа Ethereum – виртуальная машина для разработки децентрализованных приложений на базе Blockchain.
- Solidity – язык виртуальной машины Ethereum, используется для написания smart contracts
- TestRPS – эмулятор JSON RPC API Ethereum
- библиотека Web3j для Android – позволяет работать с JSON RPC API через обертки, без непосредственного ручного составления запросов
- Remix – онлайн компилятор Solidity. Преобразует контракт на языке Solidity в метаданные и байт-код виртуальной машины Ethereum.
- web3j wrapper – инструмент для работы со смарт контрактами на языке Java

- Android Studio 3.0
- NinjaMock – инструмент проектирования пользовательского интерфейса

2 Проектирование серверной части

Серверная часть системы должна состоять из хранилища (блокчейн), смарт контрактов и внешнего API, позволяющего взаимодействовать с другими модулями системы.

Реализация технологии блокчейн предоставляется платформой Ethereum, как и JSON RPC API. Контракты должны быть разработаны самостоятельно. Контракты должны определять структуру хранимых в блокчейне данных и методы взаимодействия с ними извне.

В блокчейне должны храниться данные о пользователях системы и документах, которые были добавлены пользователями. При этом наружу должен предоставляться API для взаимодействия внешнего модуля и хранилища.

В таблице 2.1 представлены некоторые методы серверной части и их параметры.

Таблица 2.1 - API серверной части

Название	Входные параметры	Выходные параметры	Описание
addDocument	string data (JSON), string fio	-	Добавление документа с параметрами, перечисленными в data в формате JSON и с ФИО владельца в параметре fio
getDocById	uint id	uint id, address owner, string data, string fio	Возвращает параметры документа по его номеру
getDocumentsNumber	-	uint number	Возвращает количество документов в хранилище

getThisAddresDocNumber	-	uint number	Возвращает количество документов, успешно добавленных текущим пользователем
getThisAddresDocById	uint id	uint id, address owner, string data, string fio	Возвращает параметры документа текущего пользователя по номеру документа в списке

В табл. 2.1 входные и выходные параметры приведены в терминологии языка Solidity. Значения некоторых из них:

- uint – тип данных беззнакового целого числа ненормированной длины
- string – тип данных динамически расширяемой строки в кодировке UTF-8
- address – тип данных, определяющий ссылки в хранилище блокчейна, представлен 20-байтовым значением

На рис. 2.1 представлена диаграмма, отражающая действие смарт контрактов, написанных на языке Solidity, на виртуальную машину Ethereum.

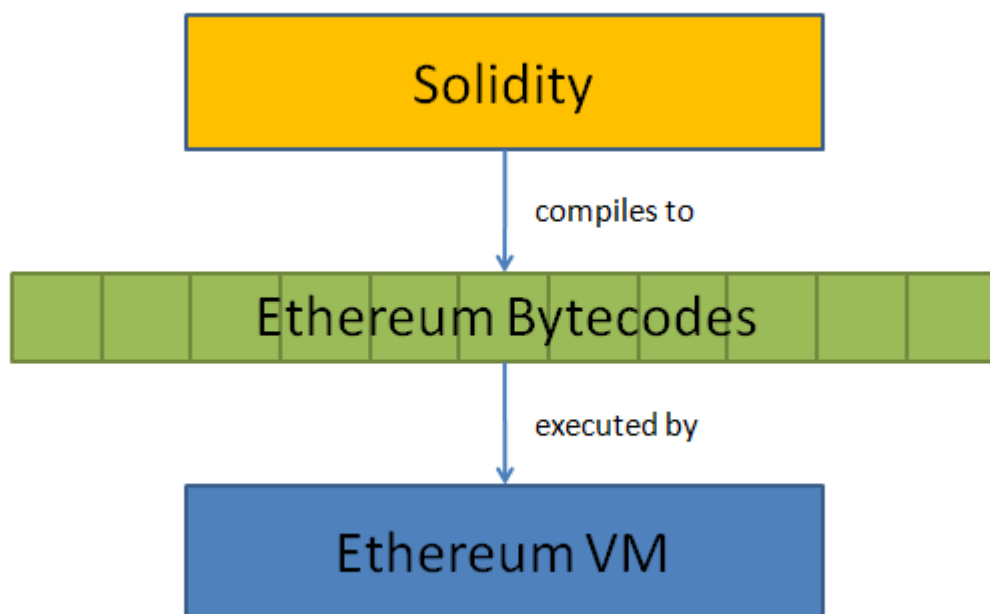


Рисунок 2.1. Взаимодействие смарт контрактов и Ethereum VM

Подготовка серверной части завершается развертыванием контракта(ов) в блокчейне и сохранением его адреса.

3 Проектирование клиентской части

JSON RPC API, предоставляемый виртуальной машиной Ethereum, требует достаточно много трудозатрат для использования, так как предполагает отправку и получение данных в виде 16-ричных строк. Это не позволяет разработчику вручную составлять запросы к Ethereum VM. К счастью, Ethereum – достаточно развитая платформа, поэтому есть набор инструментов, облегчающих доступ к данным в блокчейне из клиентских приложений.

На рис. 2.2 представлена диаграмма развертывания системы.

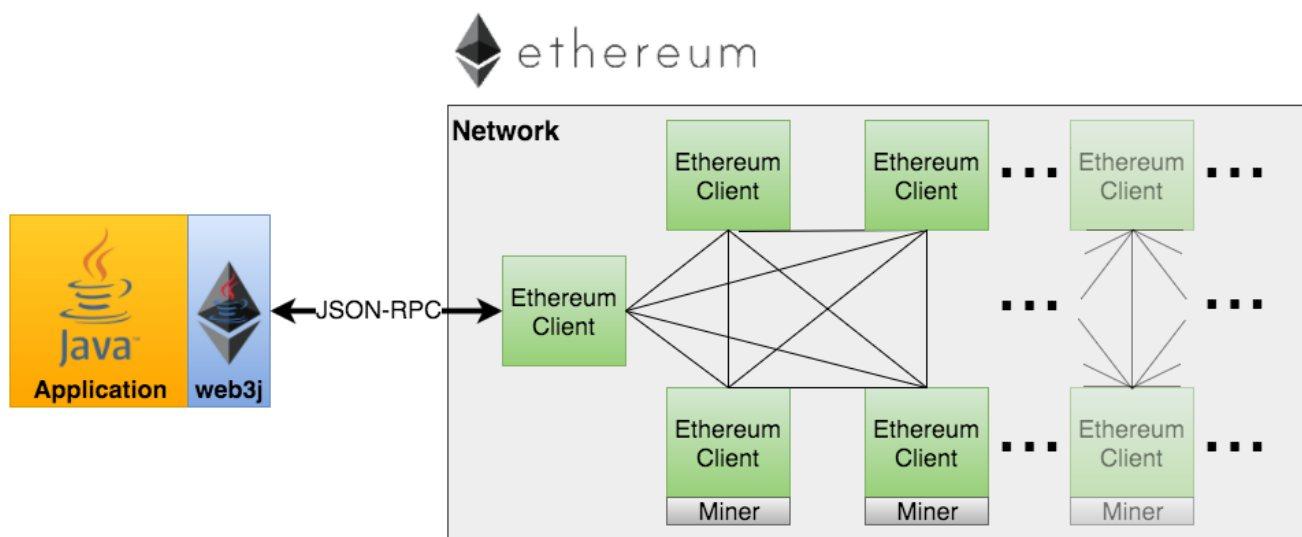


Рисунок 2.2. Архитектура децентрализованного приложения

Для данного продукта был выбран инструмент Web3j. Как указано на странице официальной странице, web3j – легкая Java и Android библиотека для интеграции с клиентом Ethereum (имеется в виду JSON RPC API).

Помимо библиотеки для Java web3j предоставляет командные инструменты для некоторых других действий, не предназначенных для исполнения во время выполнения. Например, wrapper – утилита для создания класса-обертки для контракта на языке Java. Этот класс необходимо включить в проект для взаимодействия с внешним API клиента Ethereum.

Так как `wrapper web3j` использует метаданные и байт-код контракта, а не команды на языке `Solidity`, контракт необходимо сначала скомпилировать. Для этой цели удобно пользоваться инструментом `Remix` – онлайн компилятором `Solidity`.

Получение и отправка данных в `Android` – это задачи, требующие длительного времени, поэтому их нельзя выполнять в главном потоке. Для использования потоков необходимо использовать специальные инструменты, например, `AsyncTask`. Для каждого запроса необходимо написать свой класс `AsyncTask`.

Помимо перечисленных выше компонентов, необходимо разработать классы-помощники работы с специфическими данными, например, класс для работы с профилем пользователя на устройстве. Более подробно об этом рассказано в разделе «Реализация».

4 Проектирование пользовательского интерфейса

Для работы пользователю необходимы будут следующие формы:

- авторизация
- главная форма
- добавление документа

В связи с этим, были разработаны следующие макеты:



A hand-drawn sketch of a login form. At the top center is a small key icon. Below it, the text "BlockDocs" is written in a large, stylized, handwritten font. Underneath the title are two horizontal input fields. The first field is labeled "Address" and the second is labeled "Password". Below these fields is a rectangular button. Inside the button, there is a small envelope icon followed by the word "Button". The entire form is enclosed in a simple rectangular border.

BlockDocs


Address


Password

Button

Рисунок 2.3. Макет формы авторизации

Привет, <username>

 Документ №
Специальность
Квалификация
Дата выдачи




 Документ №
Специальность
Квалификация
Дата выдачи



Рисунок 2.4. Макет главной формы

The wireframe depicts a mobile application interface for adding documents. At the top, a header bar with a diagonal hatching pattern contains the text "Add doc". Below this, a document icon (three overlapping sheets) is positioned above a "Label" and a "Text Field". This pattern is repeated twice more, creating three identical input sections. Below the input sections, there are two horizontal bars, each featuring a checkmark icon and the text "2 Selected". At the bottom of the form is a rectangular button with a star icon and the label "Button".

Рисунок 2.5. Макет формы добавления документа

Дизайн форм должен быть приближен к material design, что должно быть достигнуто использованием специфических компонентов.

6.1 Организационная структура

На рисунке 6.1 представлена организационная структура автоматизируемого объекта.

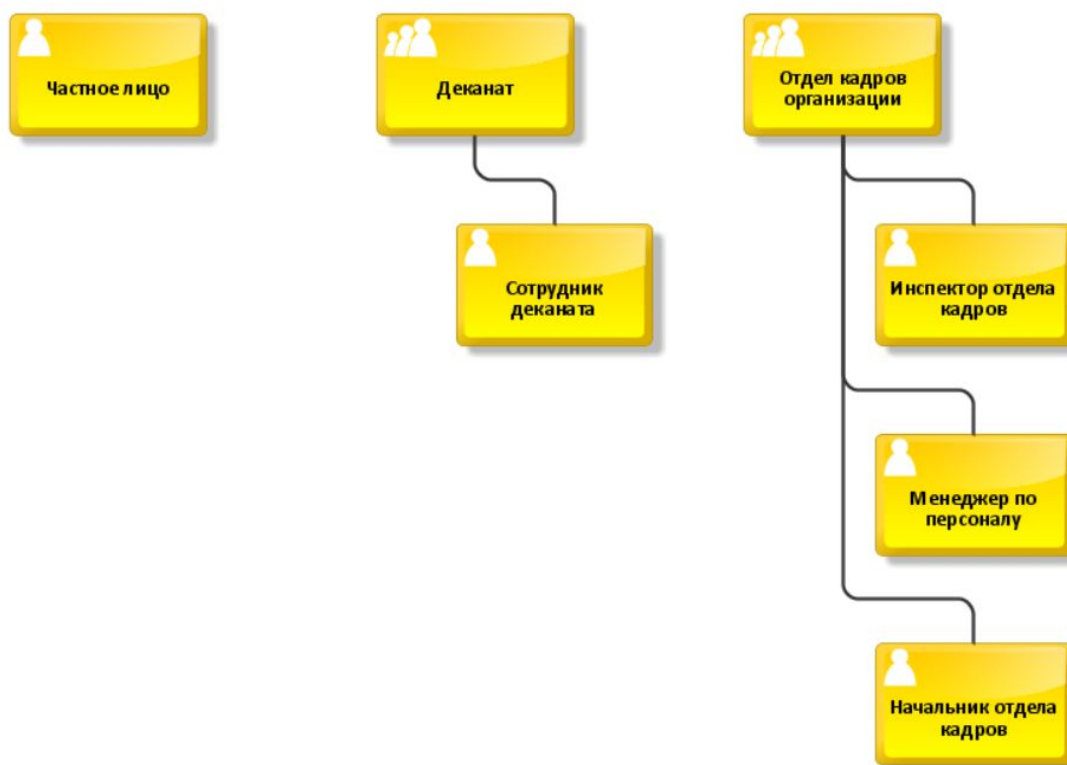


Рисунок 6.1 - Организационная структура

6.2 Информационное обеспечение

На рисунке 6.2 представлена схема информационного обеспечения.

Особенность данной информационной системы в том, что в качестве хранилища данных используется не база данных, как это принято в большинстве современных систем, а блокчейн. Блокчейн кроме всего прочего, что может содержаться в БД, сохраняет все сведения об изменении данных в хранилище. Данные в блокчейне хранятся с помощью смарт-контрактов – одновременно и вместилищ информации, и спецификаций по их обработке. Контракт создается один раз и работает весь жизненный цикл системы.

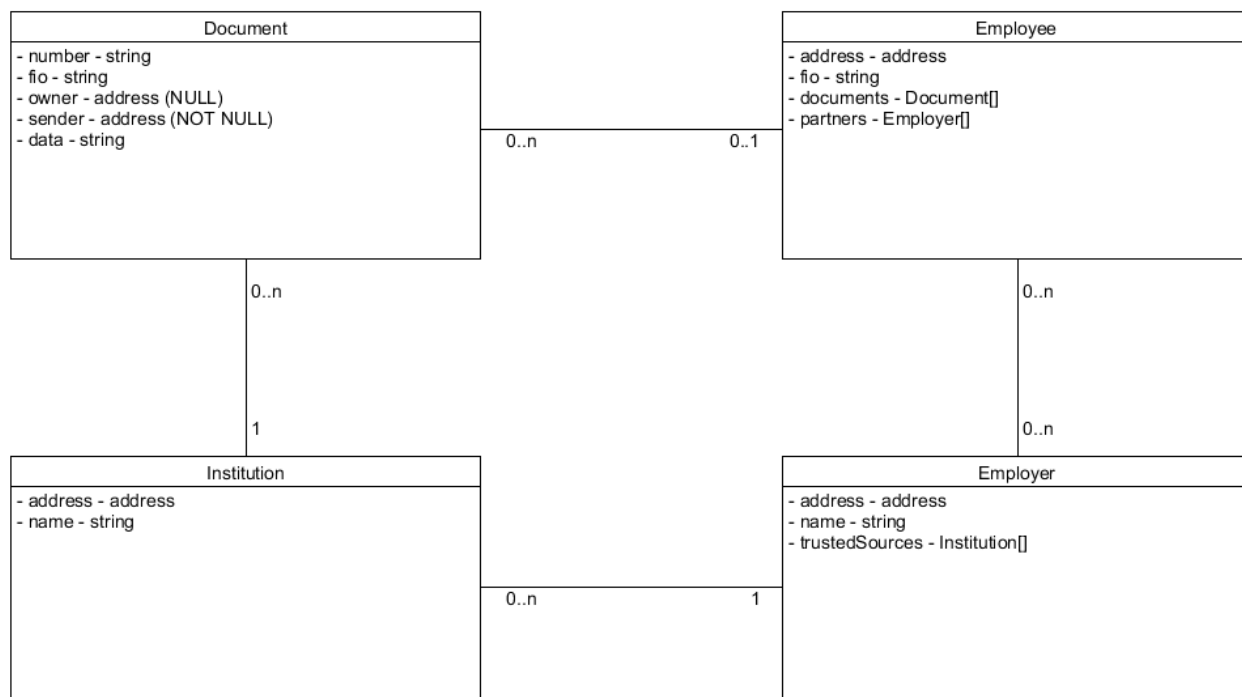


Рисунок 6.2 - Схема информационного обеспечения ИС Децентрализованный реестр документов об образовании

Блокчейн не содержит таблиц в чистом виде – в контракт можно загрузить любые данные, приведенные к строке или числу. Например, для хранения данных, по которым не предусмотрен поиск используется одно поле – data. Оно содержит сведения о документе об образовании, представленные в формате JSON. Подразумевается, что речь о нормальных формах, применимых исключительно к реляционным БД, здесь вести нельзя.

6.3 Математическое обеспечение

Алгоритм работы приложения представлен на рисунке 6.3.

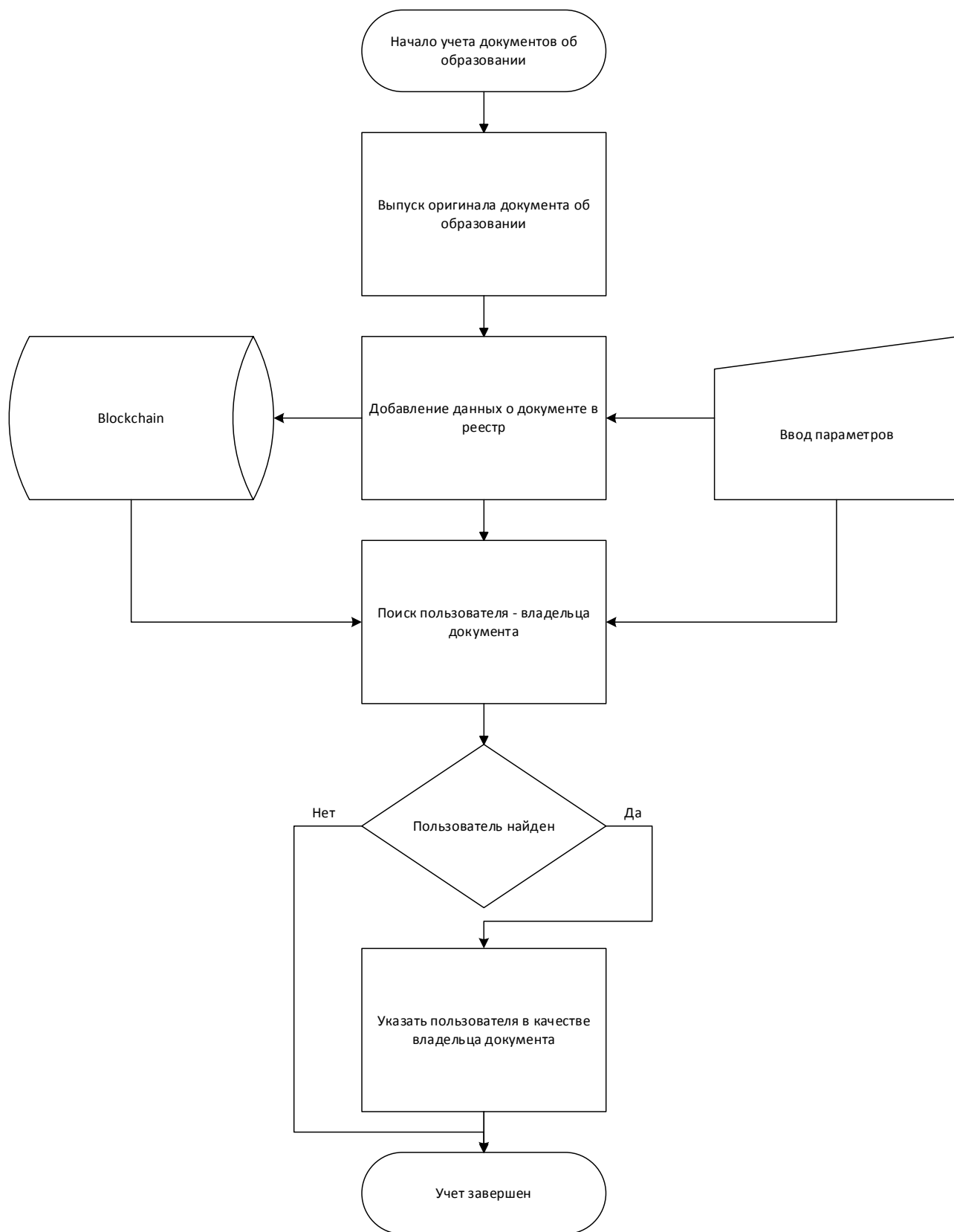


Рисунок 6.3 - Алгоритм учета документа

Наименование алгоритма: учет документа.

Назначение: предназначен для добавления документа в реестр, определения его владельца и указания ссылки на владельца в документе.

Входные данные: оригинал документа об образовании, сведения о владельце, учетные данные владельца (для зарегистрированных).

Выходные данные: сведения об успешности завершения процедуры.

Схема, моделирующая данные системы, представлена на рисунке 6.4.



Рисунок 6.4 - Схема данных

Схема взаимодействия программных модулей представлена на рисунке 6.5.

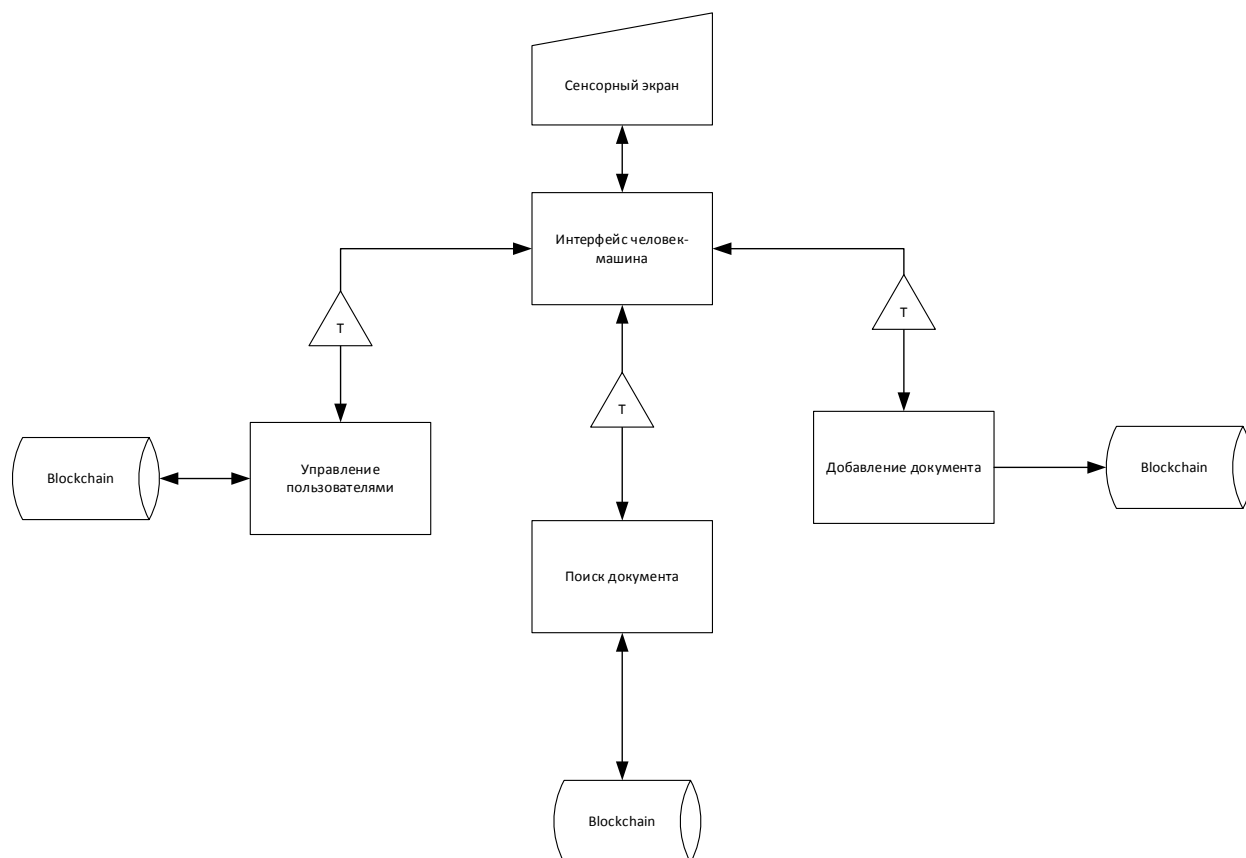


Рисунок 6.5 - Схема взаимодействия программных и технических модулей

6.4 Техническое обеспечение

Структурная схема ИС представлена на рисунке 6.5.

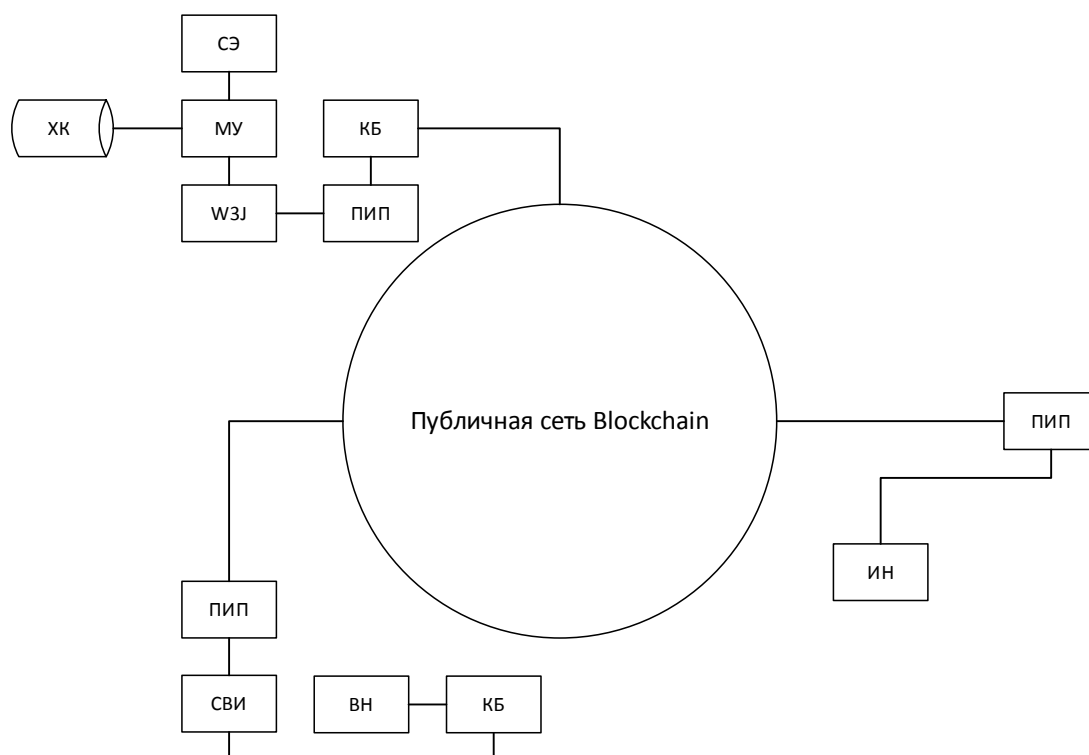


Рисунок 6.5 – Структурная схема ИС

СВИ – средство выхода в интернет

ВН – вычислительная нода

ИН – информационная нода

КБ – клиент блокчейна

ПИП – программный интерфейс приложения

W3J – библиотека web3j

МУ – мобильное устройство

СЭ – сенсорный экран

ХК – хранилище ключей

5 Разработка мероприятий по повышению надежности и достоверности выдаваемой информации

7.1 Повышение надежности

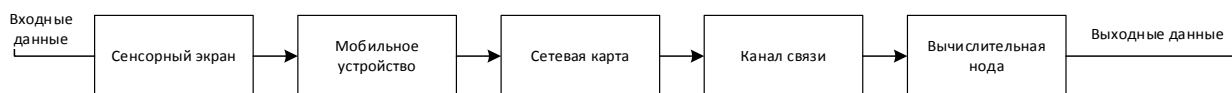


Рисунок 7.1 - Схема расчета надежности ИС

При решении поставленной задачи предполагается известным достигнутый уровень вероятности безотказной работы группы технических средств $P_i(t)$ $i = 1, 2, \dots, N$ и вводятся следующие допущения:

При решении любой задачи в ИС используются все технические средства системы;

Схема расчета надежности последовательная

Вероятность правильного решения задачи системой в заданном интервале времени $P_3(t)$ зависит только от правильной работы технических средств $P_3(t) = P(t)$;

Все задачи, решаемые системой, имеют одинаковую заданную вероятность правильного решения.

Тогда можем определить вероятность правильного решения задачи:

$$P_{\zeta}(t) = P(t) = \prod_i P_i(t) = \prod_i e^{-\lambda_i t}, i=1, 2, \dots, N \quad (7.1)$$

Сенсорный экран – $\lambda = 1,05 \cdot 10^{-4}$ 1/час (первичные датчики)

Мобильное устройство - $\lambda = 2,00 \cdot 10^{-4}$ 1/час (персональная ЭВМ)

Сетевая карта – $\lambda = 1,16 \cdot 10^{-4}$ 1/час (распределительные устройства)

Канал связи – $\lambda = 0,10 \cdot 10^{-4}$ 1/час (линии связи)

Вычислительная нода – $\lambda = 1,00 \cdot 10^{-4}$ 1/час (процессор)

Время непрерывной работы – 8 часов.

Вероятность безотказной работы незарезервированной подсистемы равна: 0,986.

7.2 Повышение достоверности выдаваемой информации

Достоверность выдаваемой информации находится по формуле:

$$J = \frac{P}{1 - Q_{kh}}, \quad (7.2)$$

где J – достоверность выдаваемой информации;

P – вероятность безотказной работы подсистемы;

Q – вероятность отказа подсистемы;

k – условная вероятность обнаружения ошибки;

h – коэффициент достоверности условной вероятности.

Исходные данные, рассчитанные ранее:

заданная достоверность: $J = 0,99$;

вероятность безотказной работы: $P_{\text{заданн}}(t) = 0,986$

вероятность отказа подсистемы $Q_{\text{заданн}}(t) = 0,004$.

При известной вероятности исправной работы системы по формуле найдем:

$$0,99 = \frac{0,986}{1 - hk \cdot 0,004}$$

$$1 - hk \cdot 0,004 = \frac{0,986}{0,99}$$

$$1 - hk \cdot 0,004 = 0,9963$$

$$hk = 0,931$$

При вероятности безотказной работы устройства контроля $P^* = 0,80$ вероятность отказа устройства контроля:

$$Q_0^* = \frac{3}{4}(1 - P^*) = 0,15$$

Рассчитаем коэффициент достоверности контрольной информации по формуле 6.

$$h = \frac{P^*}{1 - Q_0^*} \quad (7.3)$$

$$h = \frac{0,80}{1 - 0,15} = 0,95$$

Из этого следует, что:

$$0,95 * k = 0,931$$

$$k = \frac{0,931}{0,95} = 0,98$$

Для того чтобы подсистемы обеспечивала заданный уровень достоверности выдаваемой информации, подключать устройство контроля с $k = 0,98$.

РЕАЛИЗАЦИЯ

1 Реализация серверной части

В качестве серверной части выступает виртуальная машина Ethereum с развернутым смарт контрактом. Структура контракта на Solidity представлена на рис.3.1.

В первой строке указывается текущая версия Solidity. Самая новая версия на момент написания работы – 4.19, но это не самая стабильная версия. Далее идет название контракта – Documents. Внутри описания контракта содержится описание структуры данных, которая используется внутри контракта. Область видимости структуры – текущий контракт. Структура Document – эквивалент документа об образовании. Он характеризуется адресом владельца, данными в формате JSON и ФИО. Данные, по которым осуществляется поиск внутри

контракта, необходимо вынести из строки данных. Остальные данные можно хранить в одной строке в формате JSON. Из-за особенностей языка Solidity, который не позволяет полноценно работать со строками, мы не можем хранить все данные в строке data. С другой стороны, Solidity не поддерживает возврат функциями объектов (экземпляров структур), но позволяет возвращать множественные параметры. Для того, чтобы не раздувать выходные параметры, данные объединены в строку.

Для хранения документов используется динамический массив.

Все методы делятся на транзакции – изменяющие состояние контракта, - и функции – считывающие состояние. В нашем случае добавление документа будет транзакцией, а все остальные методы – функциями.

После составления контракта его необходимо развернуть в блокчейне. Пока контракт развернут, все его данные хранятся распределенными в сети. У

контракта тоже есть адрес, как и у пользователей, а значит, он может быть участником транзакций.

Это свойство используется для тестирования контрактов.

```
pragma solidity ^0.4.15;

contract Documents {

    struct Document {

        address owner;
        string document;
        // educationType;           // Тип образования
        // institution;             // Учебное заведение
        // specialty;               // Специальность
        // qualification;           // квалификация
        // dateOfAdmission;         // Год поступления
        // dateOfGraduation;        // Год окончания
        // dateOfIssuance;          // Дата выдачи
        // number;                  // Номер
        string FIO;                //ФИО
    }

    Document[] public documents;

    //Transactions
    function addDocument(string data, string fio) public {
        documents.push(Document(msg.sender, data, fio));
    }

    //Functions
    function getDocById(uint id) public constant returns
        (uint,          //id в массиве
        address,         //владелец
        string,          //содержимое
        string)          //фио
    {
        return (id, documents[id].owner, documents[id].document, documents[id].FIO);
    }
}
```

Рисунок 3.1. Структура контракта

2 Тестирование контракта

Для тестирования контракта используется среда разработки Truffle. На рис. 3.2 представлена структура проекта Truffle.










 build	30.10.2017 0:15	Папка с файлами
 contracts	29.10.2017 19:44	Папка с файлами
 migrations	29.10.2017 22:42	Папка с файлами
 node_modules	30.10.2017 23:08	Папка с файлами
 review	07.12.2017 23:23	Папка с файлами
 test	30.10.2017 22:35	Папка с файлами
 .gitignore	07.12.2017 23:09	Текстовый докум...
 package-lock.json	30.10.2017 23:08	Файл "JSON"
 truffle.js	29.10.2017 19:43	Файл "JS"

Рисунок 3.2. Структура проекта Truffle

Основная часть проекта – контракты – расположена в папке contracts. В папке build располагаются скомпилированные файлы контрактов. Папка test содержит тесты на языке Solidity и javascript.

Для тестирования контракта следует написать другой контракт, поместить его в папку test и запустить командой truffle test (рис.3.3).

```
TestDocuments
✓ testAddDocument (59ms)
✓ testGetDocById (55ms)
✓ testGetThisAddressDocNumber (89ms)

3 passing (604ms)
```

Рисунок 3.3. Тестирование контрактов

Так как контракт имеет собственный адрес, он может быть отправителем транзакций, а в нашем случае – владельцем документов.

3 Реализация клиентской части

В качестве клиента выступает приложение под платформу Android. Существует два типа клиентов для сети блокчейн – толстый клиент, который содержит копию хранилища, и тонкий клиент, который работает с хранилищем с помощью запроса.

Требование разработать приложение для мобильной платформы накладывает некоторые ограничения на продукт, а именно - оно должно занимать как можно меньше места на устройстве. В связи с этим сохранение полной копии хранилища на устройстве не представляется возможным.

На рис. 3.4. представлена структура проекта Android-приложения.

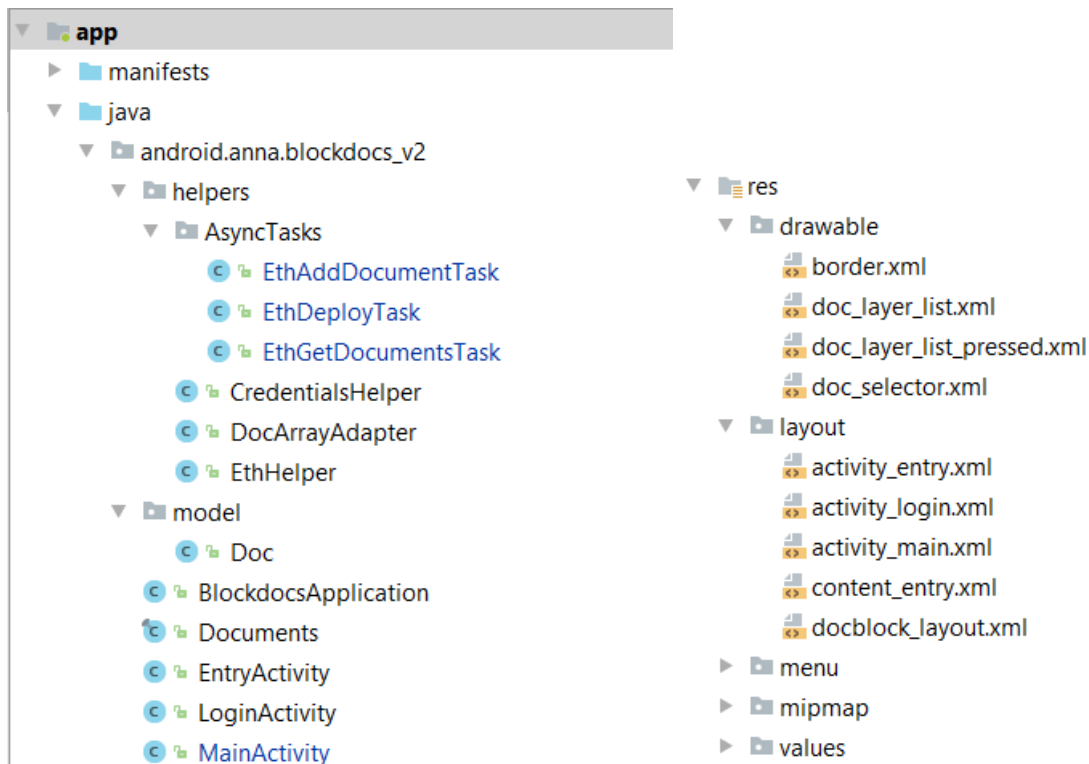


Рисунок 3.4. Структура проекта Android-приложения

- В качестве модели используется класс `Doc`, экземпляр которого содержит данные о конкретном документе.
- `DocArrayAdapter` – класс, который помогает правильно заполнить `ListView` на главной форме.
- `CredintailsHelper` – класс для безопасной работы с файлом кошелька на устройстве
- `EthHelper` – слой, реализующий доступ к хранилищу с помощью запросов
- `AsyncTasks` – набор однопоточных классов для выполнения различных запросов в отдельном потоке
- `Documents` – класс-обертка для работы со смарт-контрактом на языке Java
- `BlockDocsApplication` – кастомная реализация класса приложения для хранения необходимых констант
- набор классов `Activity` – классы для обработки событий на формах

Основной особенностью данного приложения является использование библиотеки web3j и класса-обертки смарт-контракта. Класс-обертка генерируется инструментами командной строки от web3j и содержит следующие методы (рис. 3.5).

```
private Documents(String contractAddress, Web3j web3j, Credentials credentials, BigInteger gasPrice, BigInteger gasLimit) {...}

private Documents(String contractAddress, Web3j web3j, TransactionManager transactionManager, BigInteger gasPrice, BigInteger gasLimit) {...}

public RemoteCall<BigInteger> getDocumentsNumber() {...}

public RemoteCall<BigInteger> getThisAddressDocNumber() {...}

public RemoteCall<Tuple4<BigInteger, String, String, String>> getDocById(BigInteger id) {...}

public RemoteCall<TransactionReceipt> addDocument(String data, String fio) {...}

public RemoteCall<Tuple3<String, String, String>> documents(BigInteger param0) {...}

public RemoteCall<Tuple4<BigInteger, String, String, String>> getThisAddressDocById(BigInteger id) {...}

public static RemoteCall<Documents> deploy(Web3j web3j, Credentials credentials, BigInteger gasPrice, BigInteger gasLimit) {...}

public static RemoteCall<Documents> deploy(Web3j web3j, TransactionManager transactionManager, BigInteger gasPrice, BigInteger gasLimit) {...}

public static Documents load(String contractAddress, Web3j web3j, Credentials credentials, BigInteger gasPrice, BigInteger gasLimit) {...}

public static Documents load(String contractAddress, Web3j web3j, TransactionManager transactionManager, BigInteger gasPrice, BigInteger gasLimit)
```

Рисунок 3.5 - Методы класса Documents

Методы, за исключением статических, совпадают с методами контракта и отличаются только типом выходных параметров. Проблема заключается в следующем: Solidity поддерживает возврат нескольких значений, а Java нет. Для решения этой проблемы web3j предоставляет набор классов под названием Tuple («кортеж»). Например, Tuple2<String, String> содержит в себе ровно два возвращаемых контрактом значения.

Статические методы нужны для:

- развертывания контракта в блокчейне с помощью мобильного клиента (эта функция не используется в приложении, так как контракт уже развернут раз и навсегда в сети блокчейн)
- загрузки контракта по его адресу – таким образом можно получить экземпляр контракта для дальнейшей работы с ним.

4 Тестирование клиентской части

Для тестирования клиентской части используется эмулятор сети блокчейн с готовыми аккаунтами – TestRPC. Он предоставляет JSON RPC API, с помощью которого можно развернуть, загрузить контракт, создавать

транзакции, вызывать функции контракта и т.п. На рис. 3.6. представлен пользовательский интерфейс TestRPC.

```
C:\WINDOWS\system32>testrpc
EthereumJS TestRPC v4.1.3 (ganache-core: 1.1.3)

Available Accounts
=====
(0) 0xf3f9818cf9f0669a95bf4546027bc5bfc7546630
(1) 0xdd7210bf7b9aac3c86012416b677e12d8b757e6b
(2) 0x62f0f4beb3567038e9e47b98bb0ee61c90d5ee3c
(3) 0x447391fb99b5494ba5505f498244692507db9365
(4) 0xec158614f10915c3d7f7f0d141d7ac5199b8fb51
(5) 0xe81ccba07977b3ad014d8c2d68a266bdb407e088
(6) 0x35ea43326f291ea2c11c7efc9c4e25e80d45c558
(7) 0x8e618aa006e968ef76421dbcea26d0928e8d6f2a
(8) 0xac27a862b83a76029a13977c6daa49c82b285a9c
(9) 0x59693efbbe4526d90a3d3c2f075bd389dc5dd497

Private Keys
=====
(0) 9ae40dd2fedc6a42c783835679970040739060b5b6b072e8f38dd798bbf483b7
(1) c3995da6960b0108fdc8d0b549ed2376e52f69457ce27458df34ed55dc690a12
(2) e9d2a3ae23b67b780c1ed036f6ef21cb74fdbdccc5465bbb9182ff8e702d2e047
(3) 81e96da80b7ac09434cc391230bdf9def2c466d4ad0a62d3a4b9bc89b2bd6bf3
(4) 3a2be8e31e7f518de1d072793a56200391a27fb492668c218702d5cf414659cd
(5) 88947e5d2c01d245133f7f665631e2c69831e6bdfd0c12fca6d9ab72a0fd79e1
(6) abfbeb6b37dbab9d9107e9da2d8ea6ec55f678f385164d6169b5f8749b5ca67
(7) 027e4103c5c5d7a2fa2f519e4510f714aa2d5aae2931ea12073f0057a07278a3
(8) eebb2229c4b13f157a61a32f347a108d73cafe99ab0246f9ef4ce1aa8ac1ee3f
(9) 4eeb0b7dde7cdeebbf954aa34ee1142cb0fc312a898930a25091e69bacb9dd89

HD Wallet
=====
Mnemonic:      clip educate doctor parrot ketchup desk hybrid castle volume glare glide wool
Base HD Path:  m/44'/60'/0'/0/{account_index}

Listening on localhost:8545
```

Рисунок 3.6. Интерфейс эмулятора TestRPC

Эмулятор запущен в локальной сети, поэтому адрес и используемые аккаунты прописаны в коде.

Дальнейшее тестирование состоит в проверке на соответствие продукта функциональным требованиям.

ИНФОРМАЦИОННЫЙ МЕНЕДЖМЕНТ

В сфере информационного менеджмента рассматриваются процессы управления проектом на всех этапах его жизненного цикла. При этом в информационный менеджмент в широком смысле занимается задачами, связанными не только с данными, но и со всеми другими ресурсами, которые косвенно или напрямую взаимодействуют с информацией.

В данной главе рассматриваются следующие задачи ИМ:

1. Управление капиталовложениями – эта задача решается путем расчета полной стоимости владения ИС ДРДО, включая затраты на создание и сопровождение.
2. Управление персоналом – составление должностной инструкции по работе с ИС ДРДО.
3. Развитие информационной системы и обеспечение ее обслуживания – проведение SWOT-анализа ИС ДРДО с привлечением экспертов в данной области.

2 Должностная инструкция для сотрудников, взаимодействующих с ИС ДРДО

Должностная инструкция является обязательным внутренним документом предприятия, призванным определить границы обязанностей сотрудников. Не существует какого-либо стандарта, по которому должна составляться эта инструкция и составляется руководителями предприятий, отделов, подразделений. Рассмотрим необходимые обязанности и квалификации для должностей «Менеджер по персоналу» (представитель роли «работодатель») и «Специалист по учебно-методической работе» в связи с их работой в системе ДРДО. Целиком должностные инструкции для этих должностей приведены в приложениях А и Б.

Должностные обязанности менеджера по персоналу:

1. При проведении собеседования запрашивать у соискателя данные о его образовании, в т.ч. сведения о выданных соискателю документах об образовании
2. Проверять предоставленные данные на подлинность с помощью ИС ДРДО путем поиска в реестре документов, полученных от соискателя.
3. При неуспешном поиске уточнить данные о документах у соискателя удобным обоим сторонам способом (по телефону, электронной почте или иным).
4. При повторном неуспешном поиске следовать индивидуальной инструкции предприятия.

Требования, предъявляемые к менеджеру по персоналу:

1. Уверенное владение ИС ДРДО во всех видах, в которых она поставляется на момент вступления менеджера по персоналу в должность – мобильной версией, версией для ПК, web-версией и т.д.
2. Ответственное отношение к работе с данными и клиентами.

Должностные обязанности специалиста по учебно-методической работе:

1. Оповещать обучающихся в учебном заведении о необходимости регистрации в ИС ДРДО.
2. Регистрировать обучающихся в случае, если они не могут самостоятельно зарегистрироваться в ИС ДРДО (например, в силу несовершеннолетия).
3. После выпуска документа об образовании вносить сведения об этом документе в ИС ДРДО.
4. Определять пользователя как владельца ИС ДРДО и указывать его в качестве владельца документа.

3 SWOT-анализ ИС ДРДО

SWOT-анализ – это метод стратегического планирования, который заключается в выявлении сильных и слабых сторон информационной

системы, а также в определении влияния, которые эти стороны оказывают на потенциальные возможности и угрозы для рассматриваемой системы. SWOT-анализ помогает решить одну из задач информационного менеджмента, а именно – «Развитие системы и обеспечение ее обслуживания».

Этап 1. Для выявления необходимых для анализа факторов был проведен подбор группы экспертов. При выборе экспертов была учтена психологическая совместимость (коэффициент совместимости по группе – $0,94 > 0,6$).

Участники группы экспертов:

1. Горева А.Д. – исполнитель проектных работ
2. Вершинин В.В. – руководитель проекта
3. Горев А.П. – ведущий разработчик BV LedgerLeopard, специалист по смарт-контрактам
4. Консультант по смарт-контрактам в Ethereum
5. Ведущий архитектор BV LedgerLeopard

Эксперты предложили набор сильных, слабых сторон, возможностей и угроз ИС ДРДО. При этом коэффициент конкордации составил более 0,75, что говорит о достаточной согласованности экспертов в этом вопросе.

Экспертами сформулированы следующие сильные и слабые стороны, а также возможности и угрозы:

Сильные стороны:

- S1 – безопасность хранения данных
- S2 – гарантированная подлинность документов
- S3 – высокая скорость доступа к данным
- S4 – предупреждение коррупции

Слабые стороны:

- W1 - недостаточная известность
- W2 - дефицит финансовых ресурсов
- W3 - низкая мотивация персонала
- W4 – проблемы с масштабируемостью

Возможности:

- O1 – упразднение архивов
- O2 – расширение типов документов, возможность добавления документов, не связанных с образованием
- O3 – выход на федеральный уровень
- O4 – выход на мировой уровень

Угрозы:

- T1 – проблема 51-го процента
- T2 – человеческий фактор
- T3 – компрометирование технологии в будущем (возможно)
- T4 – потеря текущей команды разработки

Этап 2. Формирование матрицы SWOT-анализа.

Матрица строится по следующему принципу - экспертная группа указывает следующие показатели:

- P_j – вероятность появления угрозы или возможности для рассматриваемой информационной системы. Может принимать значения от 0 до 1 и нечетко делится на Низкую (0 – 20%), Среднюю (20 – 50%), Высокую (50 – 85%) и Наиболее вероятную (85 – 100%).
- K_j – коэффициент влияния угрозы или возможности на дальнейшую деятельность предприятия. Так же оценивается значениями от 0 до 1 по шкале Нет влияния (0%), Низкое влияние (1 – 25%), Достаточное влияние (25 – 50%), Серьезное влияние (50 – 90%), Кардинальные изменения в процессах компании (90 – 100%).
- A_i – интенсивность сильных сторон оценивается от 1 до 5 баллов включительно. Интенсивность слабых сторон оценивается от -1 до -5 (-1 – наименьшая интенсивность, -5 – наибольшая) включительно. Интенсивность определяет, насколько значительное преимущество получает предприятие из-за своей

сильной стороны и насколько сильные потери получит из-за слабых сторон.

Таблица 1 - Матрица SWOT-анализа

		Интенсивность (Ai)	Возможности (O)				Угрозы (T)			
			O1	O2	O3	O4	T1	T2	T3	T4
Вероятность появления (Pj)			0,5	0,9	0,7	0,2	0,1	0,8	0,1	0,2
Коэффициент влияния (Kj)			0,9	0,5	1	1	1	0,4	1	0,5
S1	3		4	3	3	2	2	5	2	1
S2	5		5	4	5	4	2	5	1	1
S3	2		4	2	3	3	1	4	1	1
S4	5		2	4	5	4	3	5	1	2
W1	-5		5	5	5	5	4	1	1	4
W2	-3		2	3	2	2	3	4	1	5
W3	-2		2	4	2	2	1	5	1	4
W4	-4		4	5	5	5	4	1	1	2

В ячейках a_{ij} указывается способность сильных сторон содействовать реализации возможностей и противостоять угрозам и способность слабых сторон ослабить воздействие возможностей и усилить угрозы. Для упрощения процесса оценки рекомендуется использовать следующую шкалу:

Оценки в этих квадрантах должны выставляться без учета реальной интенсивности фактора для организации, т.к. это уже учтено в столбце интенсивность (A_j), то есть проводятся экспертные оценки влияния силы или слабости номинального выявленного фактора на отмеченные возможности или угрозы.

Этап 3 – Преобразование матрицы.

Преобразование исходной матрицы осуществляется на основании следующей формулы:

$$A_{ij} = A_i * K_j * P_j * a_{ij}$$

Затем производится суммирование полученных оценок по строкам и столбцам матрицы, а также разработка выводов и рекомендаций.

		Интенсивность (Ai)	Возможности (O)				Угрозы (T)			
			O1	O2	O3	O4	T1	T2	T3	T4
Вероятность появления (Pj)			0,5	0,9	0,7	0,2	0,1	0,8	0,1	0,2
Коэффициент влияния (Kj)			0,9	0,5	1	1	1	0,4	1	0,5
S1	3		5,4	4,05	6,3	1,2	0,6	4,8	0,6	0,3
S2	5		11,25	9	17,5	4	1	8	0,5	0,5
S3	2		3,6	1,8	4,2	1,2	0,2	2,56	0,2	0,2
S4	5		4,5	9	17,5	4	1,5	8	0,5	1
W1	-5		-11,25	-	-	-5	-2	-1,6	-0,5	-2
W2	-3		-2,7	-4,05	-4,2	1,2	-0,9	-3,84	-0,3	-1,5
W3	-2		-1,8	-3,6	-2,8	0,8	-0,2	-3,2	-0,2	-0,8
W4	-4		-7,2	-9	-14	-4	-1,6	-1,28	-0,4	-0,8

Вывод:

1. Наиболее сильные стороны: S2 – гарантированная подлинность документов и S4 – предупреждение коррупции.

2. Наиболее важными возможностями системы, воспользоваться которыми можно при помощи сильных сторон, являются:

- O3 – выход на федеральный уровень
- O1 – упразднение архивов

При этом они же являются наиболее уязвимыми из-за слабых сторон W1 (недостаточная известность) и W4 (проблемы с масштабируемостью).

3. Самая слабая сторона - W1 - недостаточная известность, она создает значительные препятствия при достижении почти всех перечисленных возможностей.

4. Выявлено, что угрозы маловероятны и успешно решаются сильными сторонами S2 (гарантированная подлинность документов) и S4 (предупреждение коррупции), к тому же, они не усугубляются слабыми сторонами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Слава Солодкий. Обзор применения технологии блокчейн в государственном управлении [Электронный ресурс]. URL: <https://medium.com/@slavasolodkiy/обзор-применения-технологии-блокчейн-в-государственном-управлении-ac53602сес7f> (дата обращения: 6.05.18)
2. Формирование и ведение федерального реестра сведений о документах об образовании и (или) о квалификации, документах об обучении [Электронный ресурс]. URL: http://obrnadzor.gov.ru/ru/activity/main_directions/reestr_of_education/ (дата обращения: 6.05.18)

ПРИЛОЖЕНИЕ А

ДОЛЖНОСТНАЯ ИНСТРУКЦИЯ ДЛЯ ДОЛЖНОСТИ «МЕНЕДЖЕР
ПО ПЕРСОНАЛУ»

1. Общие положения

1.1. Менеджер по персоналу назначается на должность и освобождается от нее приказом генерального директора компании.

1.2. Непосредственный начальник менеджера по персоналу - генеральный директор компании / директор по персоналу / начальник отдела кадров.

1.3. На время отсутствия менеджера по персоналу его заменяет лицо, занимающее аналогичную должность.

1.4. Требования, предъявляемые к менеджеру по персоналу:

- Высшее образование
- Опыт работы от года
- Уверенное владение ИС ДРДО во всех видах, в которых она поставляется на момент вступления менеджера по персоналу в должность – мобильной версией, версией для ПК, web-версией и т.д.

- Ответственное отношение к работе с данными и клиентами.

1.5. Менеджер по персоналу руководствуется в своей деятельности:

- законодательными актами РФ;
- нормативными документами компании;
- распоряжениями вышестоящих должностных лиц;
- инструкцией по работе с ИС ДРДО
- настоящей должностной инструкцией.

2. Должностные обязанности менеджера по персоналу

2.1. Обеспечивает компанию необходимым набором сотрудников с соответствующими должностям образованием, качествами и умениями.

2.2. Следит за укомплектованностью компании специалистами, при необходимости занимается рекрутингом новых сотрудников.

- 2.3. Проводит собеседования в различных формах для определения, соответствует ли соискатель требованиям должности.
- 2.4. При проведении собеседования запрашивать у соискателя данные о его образовании, в т.ч. сведения о выданных соискателю документах об образовании
- 2.5. Проверять предоставленные данные на подлинность с помощью ИС ДРДО путем поиска в реестре документов, полученных от соискателя.
- 2.6. При неуспешном поиске уточнить данные о документах у соискателя удобным обоим сторонам способом (по телефону, электронной почте или иным).
- 2.7. При повторном неуспешном поиске следовать индивидуальной инструкции предприятия.
- 2.8. Содействует адаптации сотрудника и обеспечивает корректное введение в должность.
- 2.9. Составляет рекомендации по повышению квалификации сотрудников и следит за их выполнением.
- 2.10. Занимается оформлением трудовых договоров и прочей кадровой документации.

3. Права менеджера по персоналу:

- 3.1. Получать информацию, в том числе и конфиденциальную, в объеме, необходимом для решения поставленных задач.
- 3.2. Запрашивать от руководителей структурных подразделений предприятия, специалистов и иных работников необходимую информацию (отчеты, объяснения, пр.).
- 3.3. Представлять руководству предложения по совершенствованию своей работы и работы компании.
- 3.4. Требовать от руководства создания нормальных условий для выполнения служебных обязанностей и сохранности всех документов, образующихся в результате деятельности компании.
- 3.5. Принимать решения в пределах своей компетенции.

4. Ответственность менеджера (специалиста) по персоналу

Менеджер по персоналу несет ответственность:

4.1. За невыполнение и/или несвоевременное, халатное выполнение своих должностных обязанностей.

4.2. За несоблюдение действующих инструкций, приказов и распоряжений по сохранению коммерческой тайны и конфиденциальной информации.

4.3. За нарушение правил внутреннего трудового распорядка, трудовой дисциплины, правил техники безопасности и противопожарной безопасности.