Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых»

«ВлГУ»

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Студент Горева Анна Дмитриевна

Институт ИТР

Направление 09.03.02 «Информационные системы и технологии»

**Тема выпускной квалификационной работы**

Информационная система «Децентрализованный

реестр документов об образовании»

Руководитель ВКР Вершинин В.В.

(подпись) (ФИО)

Студент Горева А.Д.

(подпись) (ФИО)

**Допустить выпускную квалификационную работу к защите**

**в государственной экзаменационной комиссии**

Заведующий кафедрой Жигалов И.Е.

(подпись) (ФИО)

« » 20 г.

Лист задания

АННОТАЦИЯ

Тема: ИС Децентрализованный реестр документов об образовании.

Ключевые слова: Blockchain, Ethereum, Solidity, реестр, документ об образовании.

В данной работе представлены этапы проектирования ИС Децентрализованный реестр документов об образовании, проведены расчеты надежности и достоверности системы, описаны особенности реализации и тестирования проектов на Ethereum, решены некоторые задачи информационного менеджмента в отношении ИС.

ВКР представлена на 30 страницах, рисунков – 8, таблиц – , использованных источников – 10.

ANNOTATION

Theme: Information system “Decentralized register of documents on education”

Keywords: Blockchain, Ethereum, Solidity, register, documents on education.

In this paper, the stages of designing the information system "Decentralized register of education documents" are presented, calculations of reliability and reliability of the system, features implementation and testing of projects on Ethereum are described, some issues of information management are resolved.

The final qualifying work is presented on 30 pages, drawings - 8, tables - 3, used sources - 20.

# СОДЕРЖАНИЕ

[ОПРЕДЕЛЕНИЯ, СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ 4](#_Toc515447629)

[ВВЕДЕНИЕ 5](#_Toc515447630)

[1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 7](#_Toc515447631)

[1.1 Обзор объектов предметной области 7](#_Toc515447632)

[1.2 Обзор аналогов 8](#_Toc515447633)

[2 ПРОЕКТИРОВАНИЕ 11](#_Toc515447634)

[2.1 Составление технического задания на проектирование 11](#_Toc515447635)

[2.2 Проектирование серверной части 15](#_Toc515447636)

[2.3 Проектирование клиентской части 17](#_Toc515447637)

[2.4 Проектирование пользовательского интерфейса 19](#_Toc515447638)

[2.5 Разработка проектного решения 21](#_Toc515447639)

[2.6 Разработка мероприятий по повышению надежности и достоверности выдаваемой информации 31](#_Toc515447640)

[3 РЕАЛИЗАЦИЯ 34](#_Toc515447641)

[3.1 Реализация серверной части 34](#_Toc515447642)

[3.2 Тестирование контракта 36](#_Toc515447643)

[3.3 Реализация клиентской части 37](#_Toc515447644)

[3.4 Тестирование клиентской части 39](#_Toc515447645)

[4 ИНФОРМАЦИОННЫЙ МЕНЕДЖМЕНТ 41](#_Toc515447646)

[4.1 Расчет полной стоимости владения ИС ДРДО 41](#_Toc515447647)

[4.2 Инструкция по работе с ИС ДРДО 46](#_Toc515447648)

[4.3 SWOT-анализ ИС ДРДО 50](#_Toc515447649)

[ЗАКЛЮЧЕНИЕ 55](#_Toc515447650)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 56](#_Toc515447651)

# ОПРЕДЕЛЕНИЯ, СОКРАЩЕНИЯ И ОБОЗНАЧЕНИЯ

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

АС ФРДО – информационная система «Федеральный реестр документов об образовании» - действующая система, разработанная по заказу министерства образования Российской Федерации [1].

БД – база данных.

Blockchain – распределенное хранилище данных, построенное по определенным правилам объединения блоков в цепочки [2].

Смарт-контракт – обработчик транзакций, предназначенный для определения правил изменения состояния блокчейна [3].

Ethereum – платформа для создания децентрализованных онлайн-сервисов на базе блокчейна [4].

Solidity – язык виртуальной машины Ethereum, предназначенный для написания умных контрактов [5].

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

# 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 [6].

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’е [7].

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.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Обзор объектов предметной области

В предметной области используются следующие объекты.

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.

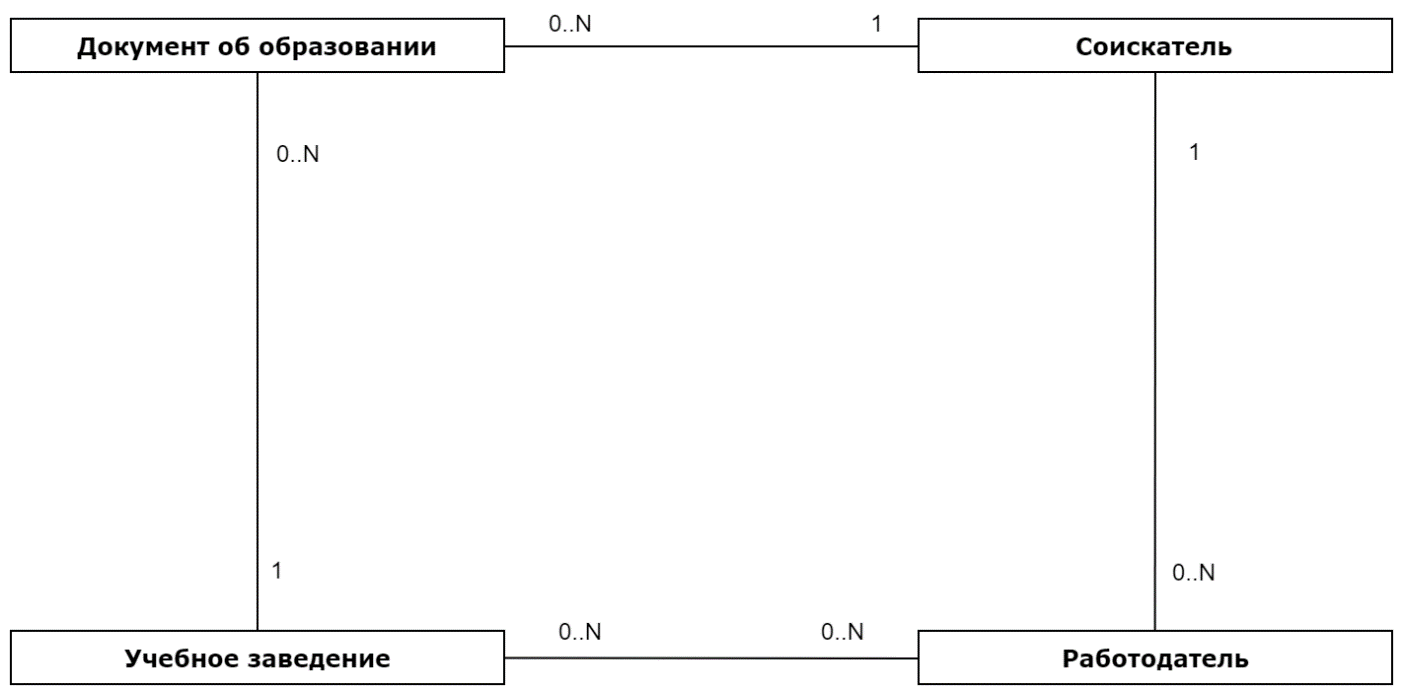
Документ об образовании служит исключительно подтверждением, что его обладатель закончил обучение в выпускающем учреждении и сопутствующие выпуску документа данные. Документ об образовании не содержит сведений о контактах, адресах вузов. Классический пример документа об образовании – диплом о высшем образовании, который получают выпускники вузов [8].

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 рисунке 1.1 приведено взаимоотношение основных объектов предметной области.

Рисунок 1.1 – Концептуальная модель данных ИС ДРДО

## 1.2 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 (ФАС ФРДО), обеспечивающая сбор сведений о выданных документах с образовательных учреждений, накопление этих сведений в единой базе данных» [9].

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.

# 2 ПРОЕКТИРОВАНИЕ

## 2.1 Составление технического задания на проектирование

2.1.1 Общие сведения

2.1.1.1 Полное наименование системы и ее условное обозначение

Информационная система «децентрализованный реестр документов об образовании»

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

Инициативный проект

Компания разработчика: ИП Штых А.Д.

2.1.1.3 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.

2.1.1.4 Плановые сроки начала и окончания работы по созданию системы

Начало: февраль 2017

Конец: апрель 2018

2.1.2 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.

2.1.3 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 рисунке 2.1. Их описание приведено в таблице 2.1.

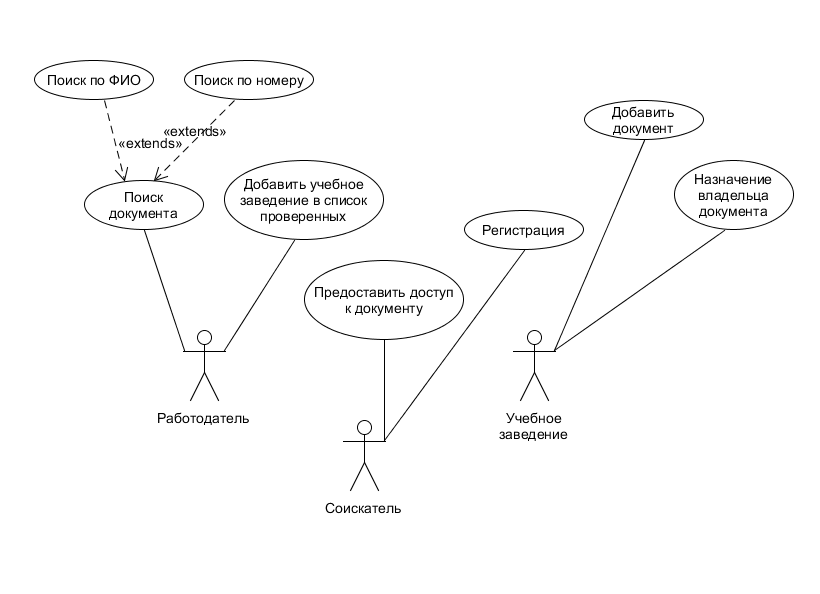


Рисунок 2.1 - Диаграмма прецедентов ИС Децентрализованный реестр документов об образовании

Таблица 2.1 – 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 [10];
* библиотека Web3j для Android – позволяет работать с JSON RPC API через обертки, без непосредственного ручного составления запросов [11];
* Remix – онлайн компилятор Solidity, который преобразует контракт на языке Solidity в метаданные и байт-код виртуальной машины Ethereum [12];
* web3j wrapper – инструмент для работы со смарт контрактами на языке Java [13];
* Android Studio 3.0;
* NinjaMock – инструмент проектирования пользовательского интерфейса [14].

## 2.2 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 [15].

В таблице 2.2 представлены некоторые методы серверной части и их параметры.

В таблице 2.2 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 [16].

Таблица 2.2 - 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 |

На рисунке 2.2 представлена диаграмма, отражающая действие смарт контрактов, написанных на языке Solidity, на виртуальную машину Ethereum.

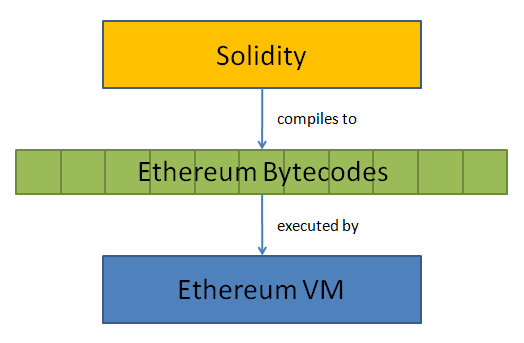


Рисунок 2.2 - Взаимодействие смарт контрактов и Ethereum VM [17]

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

## 2.3 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 рисунке 2.3 представлена диаграмма развертывания системы.

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).

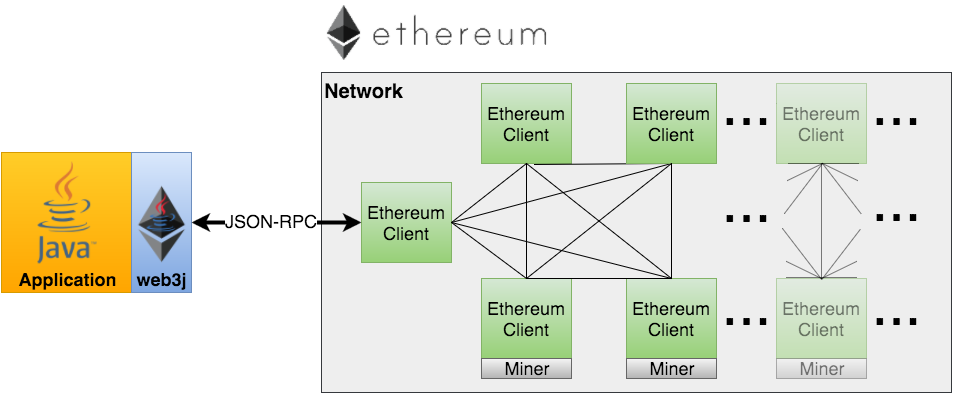


Рисунок 2.3 - Архитектура децентрализованного приложения [18]

Помимо библиотеки для Java web3j поставляет 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».

## 2.4 Проектирование пользовательского интерфейса

Для работы пользователю необходимы будут следующие формы:

* авторизация;
* главная форма;
* добавление документа.

В связи с этим, были разработаны макеты, представленные на рисунках 2.4 – 2.6.

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

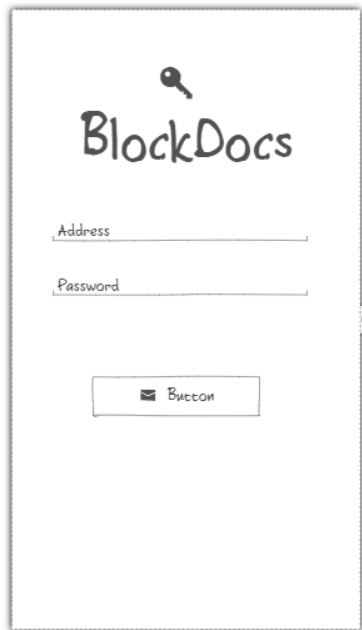


Рисунок 2.4 - Макет формы авторизации

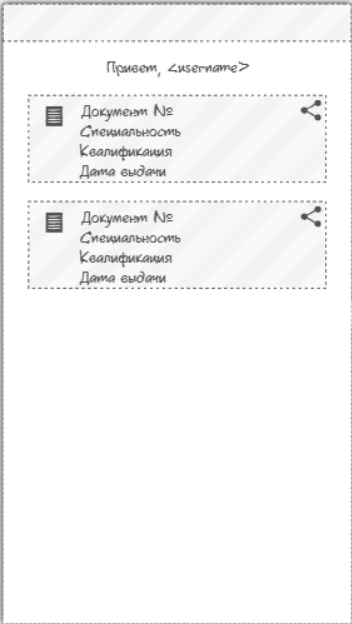


Рисунок 2.5 - Макет главной формы

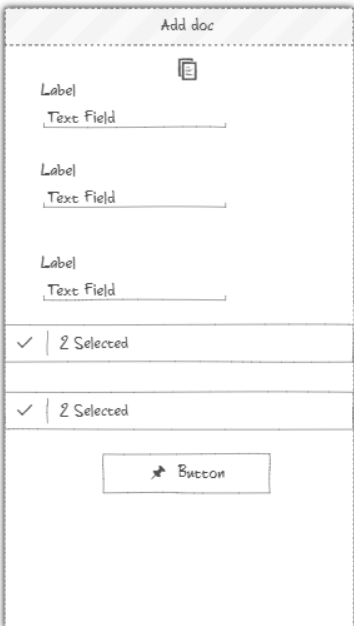


Рисунок 2.6 - Макет формы добавления документа

## 2.5 Разработка проектного решения

Рассмотрим три варианта архитектуры ИС для работы с документами об образовании:

* классическая архитектура с использованием реляционных баз данных;
* блокчейн, развернутый в сети интернет и арендуемый эксплуататором системы;
* блокчейн, развернутый в приватной локальной сети министерства образования (или в любой другой).

Первый вариант, скорее всего, применяется в АС ФРДО. Он прост в разработке, существует множество удобных решений для такой архитектуры, мировой опыт в создании и сопровождении этой архитектуры велик. Однако, остро стоит проблема доверия как источниками документов (ее успешно решают с помощью использования ЭЦП), так и самому реестру. Все процессы, происходящие в системе с такой архитектурой, скрыты от обычного пользователя.

Также приходится решать проблемы надежности и доступности системы с помощью избыточного резервирования хранилищ данных. На рисунке 2.7 представлена диаграмма развертывания для первого варианта архитектуры.

**

Рисунок 2.7 – Диаграмма развертывания ИС с использованием реляционных БД

Второй и третий варианты, ориентированные на использование технологии блокчейн, пользуются всеми преимуществами классического подхода к хранению данных, а также добавляют еще несколько. К ним можно отнести:

* децентрализованность;
* естественное дублирование данных;
* прозрачность всех транзакций в цепи блоков.

В отличие от классической архитектуры, блокчейн не имеет выделенного центра, а данные хранятся одновременно в нескольких копиях в сети. Это защищает данные от взлома и фальсификации, так как злоумышленнику необходимо изменить больше половины копий данных в сети, чтобы изменения были приняты всеми участниками контракта [19].

Под естественным дублированием данных я подразумеваю то, что технология для своей работы априори требует дублирования данных, частично или полностью. Это сказывается как на надежности, так и на доступности системы, и на достоверности данных.

Надежности системы, построенной на основе технологии блокчейн, способствует тот фактор, что отказ одного из узлов сети не приведет к критическим последствиям, как, например, отказ сервера для классической клиент-серверной архитектуры. Чтобы система отказала полностью, все ее узлы должны быть непригодны одновременно. По этой же причине класс доступности у систем, построенных по такому принципу выше, чем у систем с классической архитектурой.

И, наконец, достоверность данных обеспечивается за счет прозрачности всех изменений в системе и контроля доступа к ресурсам [20].

Вернемся к вариантам архитектуры, приведенным выше. Второй вариант хорош тем, что нет необходимости задумываться о ИТ-инфраструктуре предприятия, эту роль на себя берет сторонний сервис. И, хотя полностью отказаться от системного администрирования нельзя, можно значительно снизить нагрузку на системного администратора. Минусами второго варианта являются существенные затраты на услуги внешних сервисов и недостаточная управляемость ресурсами. Для этого варианта всегда есть риск отказа, который по срочности для арендатора будет критичнее, чем для поставщика услуг.

Третий вариант – наиболее гибкий в управлении. По сравнению со вторым вариантом архитектуры он позволяет быстро и относительно безболезненно масштабировать сеть, конфигурировать систему и включать дополнительные методы защиты данных, например, резервирование. Его и примем за оптимальный и будем рассматривать в дальнейшем.

2.5.1 Организационная структура

На рисунке 2.7 представлена организационная структура подразделений, работающих с ИС ДРДО.

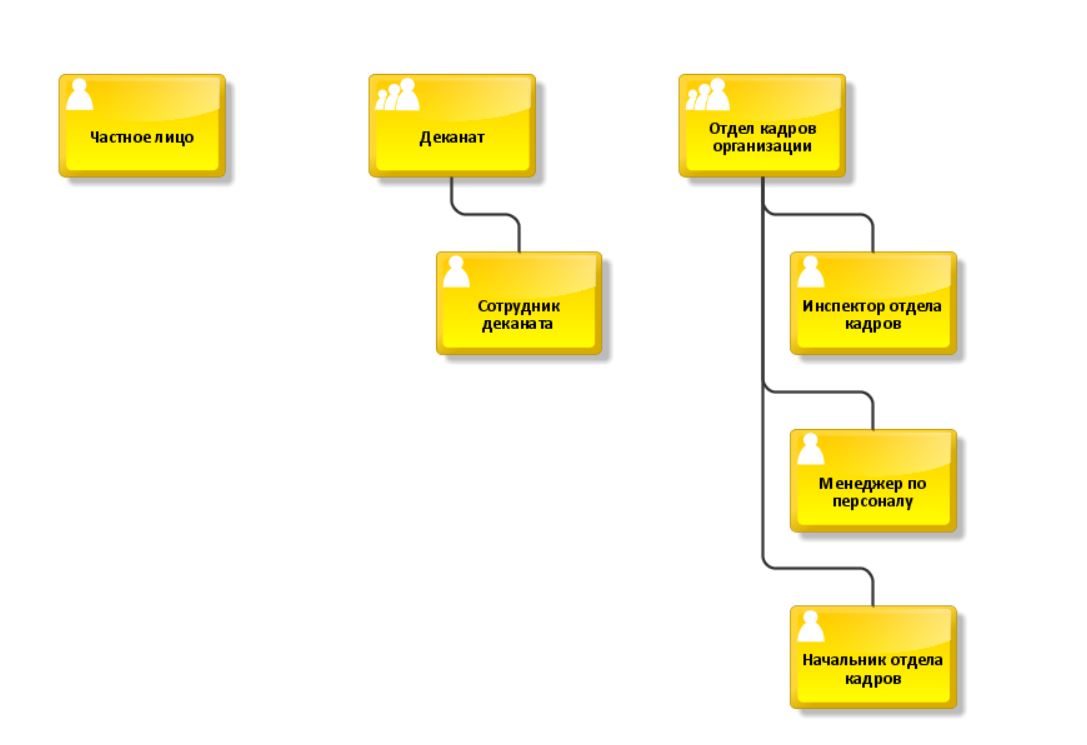


Рисунок 2.7 - Организационная структура

2.5.2 Информационное обеспечение

На рисунке 2.8 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 сведенья 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.

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 сведенья о документе об образовании, представленные в формате JSON. Подразумевается, что речь о нормальных формах, применимых исключительно к реляционным БД, здесь вести нельзя.

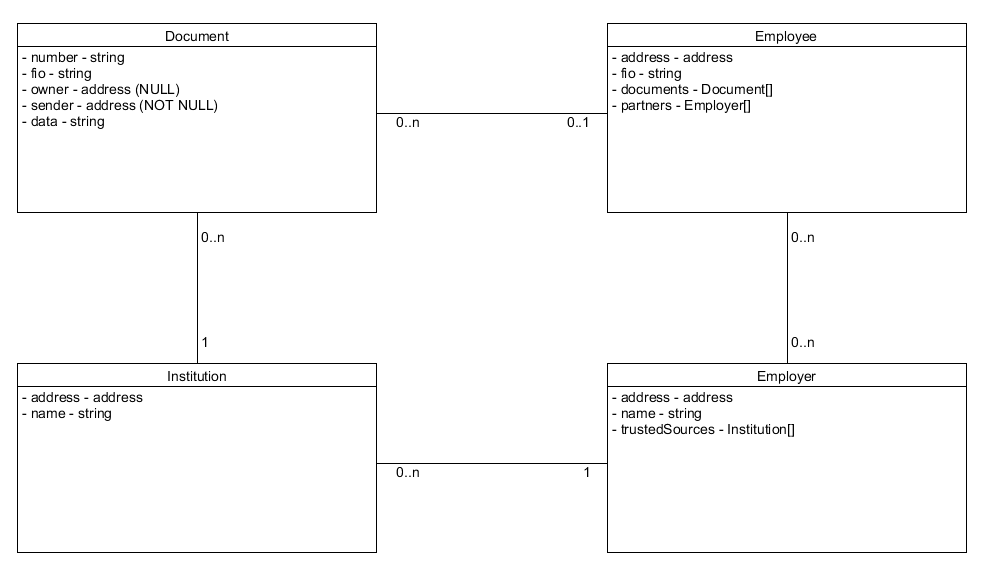


Рисунок 2.8 - Схема информационного обеспечения ИС ДРДО

2.5.3 Математическое обеспечение

Алгоритм работы приложения представлен на рисунке 2.9.

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, сведенья о владельце, учетные данные владельца (для зарегистрированных).

Выходные данные: сведенья об успешности завершения процедуры.

Схема, моделирующая данные системы, представлена на рисунке 2.10.



Рисунок 2.9 - Алгоритм учета документа в ИС ДРДО



Рисунок 2.10. - Схема данных ИС ДРДО

Схема взаимодействия программных модулей представлена на рисунке 2.11.



Рисунок 2.11 - Схема взаимодействия программных и технических модулей

2.5.4 Техническое обеспечение

Структурная схема ИС представлена на рисунке 2.12.

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



Рисунок 2.12 – Структурная схема ИС

## 2.6 Разработка мероприятий по повышению надежности и достоверности выдаваемой информации

2.6.1 Повышение надежности



Рисунок 2.12 - Схема расчета надежности ИС ДРДО

При решении поставленной задачи предполагается известным достигнутый уровень вероятности безотказной работы группы технических средств Pi(t)i = 1,2,...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 РЗ(t) зависит только от правильной работы технических средств РЗ(t) = P(t);
* все задачи, решаемые системой, имеют одинаковую заданную вероятность правильного решения.

Тогда можем определить вероятность правильного решения задачи:

, i=1,2,…N (2.1)

Сенсорный экран – λ = 1,05\*10-4 1/час (первичные датчики)

Мобильное устройство - λ = 2,00\*10-4 1/час (персональная ЭВМ)

Сетевая карта – λ = 1,16\*10-4 1/час (распределительные устройства)

Канал связи – λ = 0,10\*10-4 1/час (линии связи)

Вычислительная нода – λ = 1,00\*10-4 1/час (процессор)

Время непрерывной работы – 8 часов.

Вероятность безотказной работы незарезервированной подсистемы равна: 0,986.

2.6.2 Повышение достоверности выдаваемой информации

Достоверность выдаваемой информации находится по формуле:

, (2.2)

где – достоверность выдаваемой информации;

Р – 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: Pзаданн(t)= 0,986

вероятность отказа подсистемы .

При известной вероятности исправной работы системы по формуле найдем:

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

(2.3)

Рассчитаем коэффициент достоверности контрольной информации по формуле 2.4.

Из этого следует, что:

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

# 3 РЕАЛИЗАЦИЯ

## 3.1 Реализация серверной части

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

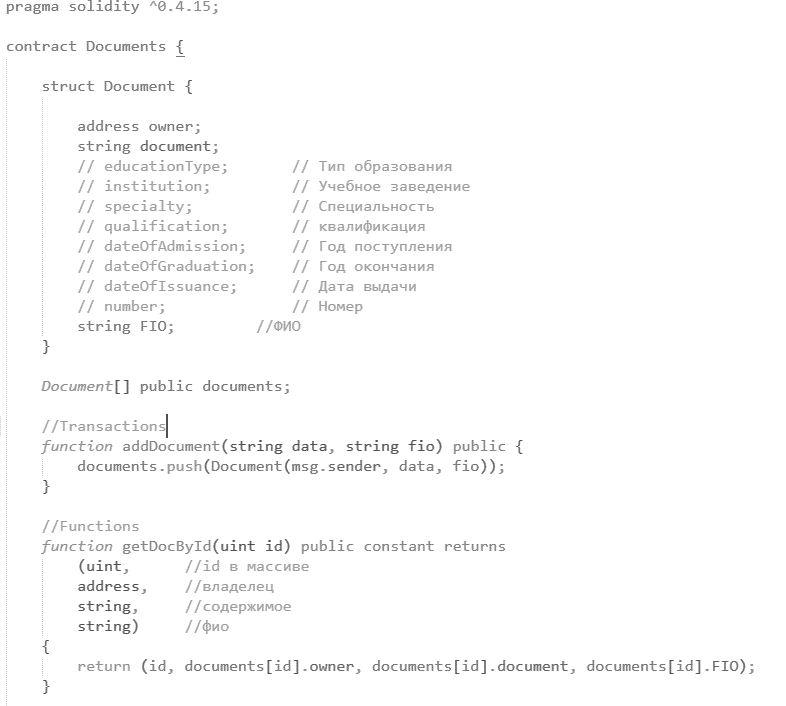
Это свойство используется для тестирования контрактов. 

Рисунок 3.1 - Структура контракта

## 3.2 Тестирование контракта

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

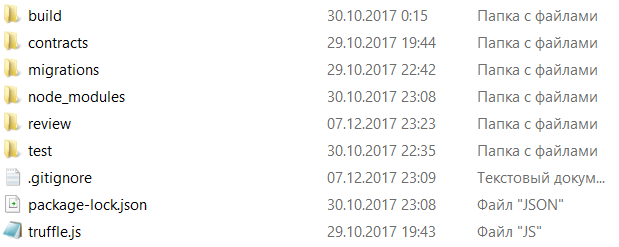


Рисунок 3.2 - Структура проекта Truffle

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. Резултаты прохождения тестов представлены на рисунке 3.3.

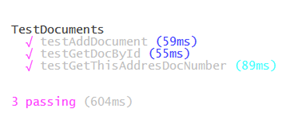


Рисунок 3.3 - Тестирование контрактов

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

## 3.3 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 рисунке 3.4. представлена структура проекта Android-приложения.

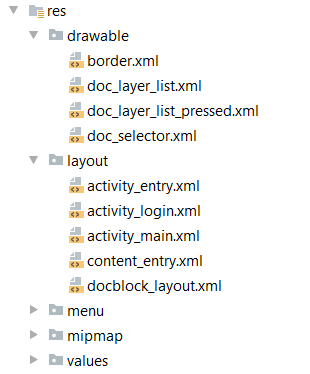
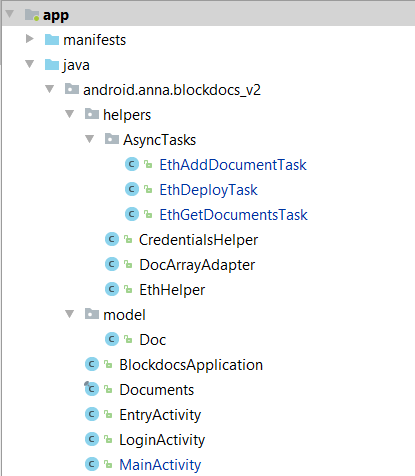


Рисунок 3.4 - Структура проекта Android-приложения

Рассмотрим отдельные элементы 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 однотипных 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 методы, представленные на рисунке 3.5.

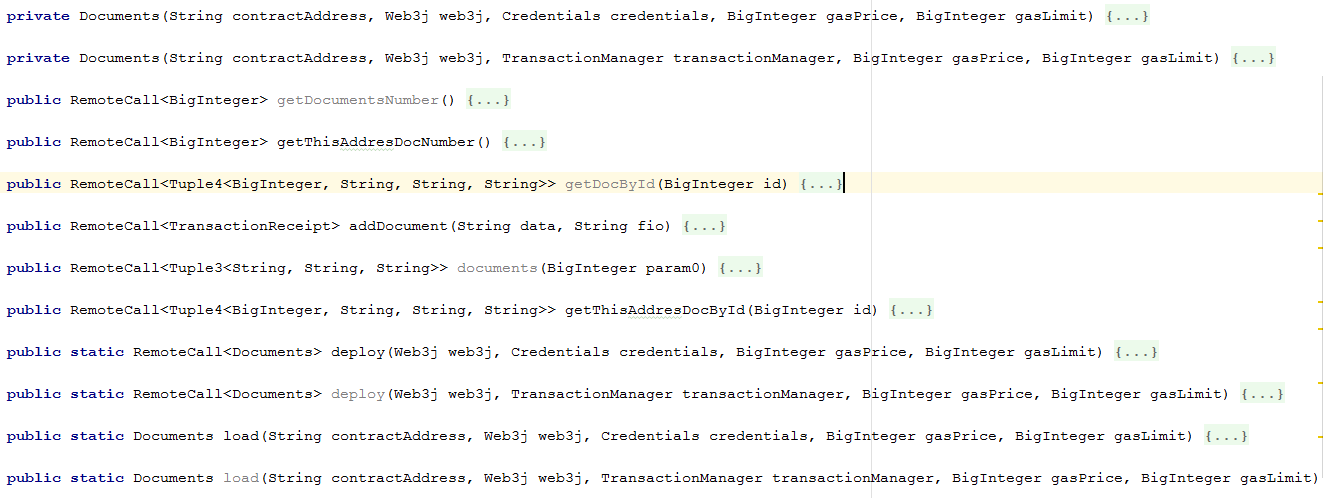


Рисунок 3.5 - Методы класса Documents

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 поставляет 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);
* выгрузки контракта по его адресу – таким образом можно получить экземпляр контракта для дальнейшей работы с ним.

## 3.4 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, выгрузить контракт, создавать транзакции, вызывать функции контракта и т.п. На рисунке 3.6. представлен пользовательский интерфейс TestRPC.

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

Дальнейшее тестирование состоит в проверке на соответствие продукта функциональным требованиям.



Рисунок 3.6 - Интерфейс эмулятора TestRPC

# 4 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 [21].

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 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.

## 4.1 Расчет полной стоимости владения ИС ДРДО

Полная стоимость владения ИС – это экономический показатель, с помощью которого можно объединить все затраты на владение ИС, в том числе и на создание (покупку), и на ежегодные затраты, связанные с эксплуатацией системы.

При определении ТСО можно руководствоваться следующим набором статей затрат:

* заработная плата персонала сферы информатизации;
* амортизация основных средств;
* материалы, детали, узлы и комплектующие;
* технологическая энергия;
* накладные расходы [22].

Стоимость ИС ДРДО складывается из следующих составляющих:

1. затраты на заработную плату участникам процесса разработки системы;
2. расходов на амортизацию оборудования
3. затраты на электроэнергию

Стоимость разработки системы автоматизации рассчитывается по следующей формуле:

, (4.1)

где Сис – стоимость разработки системы;

З – затраты по заработной плате специалистам, задействованным в разработке систем;

А – амортизация оборудования и нематериальных активов, используемых в процессе разработки системы;

Э – затраты на электроэнергию.

Для расчета затрат на выплату заработной платы специалистам, задействованным в разработке системы, целесообразно составить квалификационный план проекта разработки системы. План представлен в таблице 4.1.

Общая длительность выполнения проектных работ составляет:

Тис = 55 рабочих дней.

Затраты по заработной плате рассчитываются следующим образом:

З = Ззп + СВ, (4.2)

где Ззп – заработная плата задействованных специалистов;

СВ – страховые взносы.

Ззп =∑(Оi /Д\*ti), i=1..n, (4.3)

где n – количество задействованных специалистов;

Оi – оклад i-го специалиста (рубли);

Д – количество рабочих дней в месяце (дни);

ti – время участия специалиста в проекте (дни), определяется в соответствии с разработанным сетевым планом проектных работ.

Таблица 4.1 - Квалификационный план проекта разработки системы автоматизации отношений с рекламодателями

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Должность | Оклад, руб/мес. | З/п, руб/день | Ответственность в процессе разработки | Длительность работ, % от ТИС |
| Аналитик (1) | 120 000 | 5 500 | Разрабатывает проектные решения и документацию | 20% |
| Разработчик (3) | 50 000 | 2 400 | Реализует проектные решения | 80% |
| Специалист по обеспечению качества (1) | 50 000 | 2 400 | Обеспечивает требуемое качество системы | 50% |
| Системный администратор (1) | 50 000 | 2 400 | Поддерживает информационную инфраструктуру проекта | 100% |

Отчисления в Фонд оплаты труда составляют 30,2%:

ЕСН= Ззп\*0,302 (4.4)

Учитывая сроки разработки системы и квалификационный план выполнения проектных работ, затраты на заработную плату задействованных специалистов составят

Ззп = 11 дн. \* 5 500 руб. + 3 \* 44 дн. \* 2 400 руб. + 23 дн. \* 2 400 руб. + 55 дн. \* 2 400 руб. = 564 500 руб.

С полученной суммы в фонд оплаты труда будут произведены отчисления в размере:

ЕСН = 2 591 000 руб. \* 0,302 = 170 479 руб.

В итоге затраты по заработной плате составят:

З = 734 979 руб.

Основным расходным материалом, задействованным при эксплуатации ИС, является электроэнергия, необходимая для работы сервера. Для нормальной работы системы необходимо хотя бы 3 сервера. Доступность серверов должна быть в режиме 24:7:365. Номинальная мощность сервера составляет 350 Вт/ч.

Э – стоимость электроэнергии.

Э = Р\*Ц\*Т, (4.5)

Р – мощность сервера;

Ц – цена потребляемого ресурса;

Т – период, за который рассчитывается расход в часах.

Э = Р\*Ц\*Т = 3 \* 0,350 \* 4,8 \* 365 \* 24 = 44 150 руб.

Амортизация, входящая в формулу стоимости ИС – это амортизации оборудования, используемого при эксплуатации системы.

А = А1 , (4.6)

где А – общая амортизация,

А1 – амортизация оборудования.

В таблице 4.2 приведены расчеты норм амортизации оборудования, а в таблицу 4.3 сведены затраты на амортизацию оборудования и нематериальных активов, используемых в процессе разработки системы.

Исходя из полученных расчетных данных стоимость разработки системы составляет:

С = 3 363 873,24 руб. + 44 150 руб. + 12 400 руб. ≈ 792 000 (руб).

Так как на разработку системы был выделен бюджет в 1 миллион рублей, планируемая стоимость укладывается в эти границы.

Таблица 4.2 - Расчеты норм амортизации оборудования и программного обеспечения

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименование | Стоимость, руб | Срок эксплуатации, лет | Норма амортизации, руб/мес | Норма амортизации, руб/день |
| Сервер | 40 000,00 | 5 | 40 000/5/12=670,00 | 670/20 = 34 |
| Всего |  |  | 670,00 | 34,00 |

Таблица 4.3 - Расчет амортизации оборудования и нематериальных активов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Амортизационные активы | Кол-во, шт. | Норма амортизации, руб/день | Длительность, дней | Стоимость, руб |
| Сервер | 3 | 34 | 365 | 12 400 |
| Всего (А) |  |  |  | 12 400 |

## 4.2 Инструкция по работе с ИС ДРДО

Для эффективной работы с системой ДРДО была составлена инструкция, призванная помочь пользователям освоить основной функционал ИС в целом и мобильного приложения в частности. Инструкция приведена ниже.

4.2.1 Первый запуск.

После установки приложения откройте его, нажав на ярлык на рабочем столе. Для работы необходимо подключение к интернету по WiFi или через мобильные сети. Откроется форма входа и регистрации, представленная на рисунке 4.1.

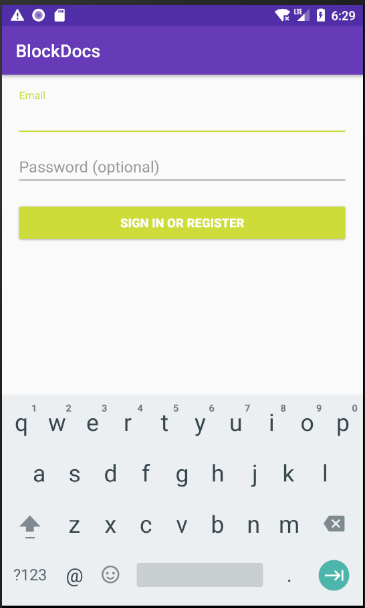


Рисунок 4.1 – Форма входа мобильного приложения ИС ДРДО

Для доступа к системе необходимо зарегистрироваться, используя уникальные логин и пароль. Следуйте подсказкам на форме для регистрации.

4.2.2 Вход в приложение.

Для входа используйте логин и пароль, который вы вводили при регистрации. Нажмите кнопку «Enter or register» после заполнения полей входа. Отобразится список документов, представленный на рисунке 4.2.

4.2.3 Добавление документа.

Для добавления документа нажмите кнопку «Add document» на главной форме приложения. Откроется форма с полями ввода данных документа, представленная на рисунке 4.3.

Заполните все поля корректными данными и нажмите кнопку внизу экрана. Если вы все сделали правильно, документ появится на главной форме приложения.

4.2.4. Назначение владельца документа

Для назначения владельца документа перейдите к документу, нажав на него на форме поиска или на главной форме. Откроется форма деталей документа. В поле «Владелец» нажмите на иконку редактирования. Введите публичный адрес пользователя-владельца, представленный в формате «0xca30e63200a0fe3182dc61fc5605efc41456f32».

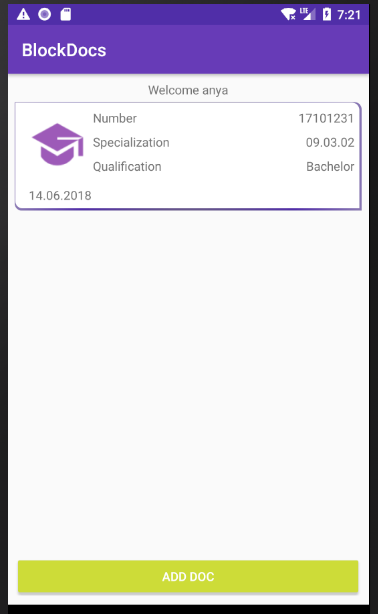


Рисунок 4.2 – Список документов

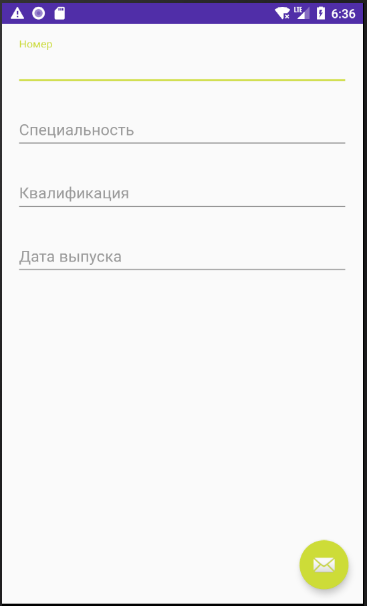


Рисунок 4.3 – Форма добавления документа

Во избежание ошибок при вводе рекомендуем воспользоваться функцией сканирования QR-кода пользователя. Для этого нажмите на иконку камеры в поле «Владелец». Откроется окно камеры. Если потребуется, разрешите приложению доступ к камере.

Наведите камеру на QR-код пользователя, чтобы считать его. Если вы все сделали правильно, камера закроется, а в поле «Владелец» появится ФИО пользователя, указанного в качестве владельца.

4.2.5 Поиск документа

Перейдите к форме поиска документов, нажав на иконку лупы в правом верхнем углу главной формы. Введите параметры, по которым необходимо осуществить поиск. Нажмите кнопку «Найти» и ожидайте результата. Если документы не найдены, попробуйте изменить параметры фильтра.

4.2.6 Предоставление доступа к документу

Если вы являетесь владельцем документа, вы можете менять права доступа к этому документу. Выберете документ на главной форме. На форме деталей нажмите кнопку «Могут просматривать». На форме списка пользователей, имеющих право доступа, нажмите на кнопку с плюсом. Сканируйте QR-код или вводите вручную адреса пользователей, как это описано в пункте 4.2.4.

4.2.7 Добавление публикаторов в список проверенных

Перейдите к форме доверенных источников, нажав кнопку «Источники» на главной форме. Добавляйте источники аналогично тому, как это описано в пункте 4.2.6.

## 4.3 SWOT-анализ ИС ДРДО

SWOT-анализ – это метод стратегического планирования, который заключается в выявлении сильных и слабый 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» [23].

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. Исполнитель проектных работ
2. Руководитель проекта
3. Ведущий специалист по смарт-контрактам
4. Консультант по смарт-контрактам в Ethereum
5. Ведущий архитектор систем в Ethereum

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.

Экспертами была сформирована матрица SWOT-анализа, представленная таблицей 4.4.

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

* Pj – 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%).
* Kj – 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%).
* Ai – 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 [24].

Таблица 4.4 - Матрица SWOT-анализа

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Интенсивность (Аi) | Возможности (О) | | | | Угрозы (Т) | | | |
|  |  | О1 | О2 | О3 | O4 | Т1 | Т2 | Т3 | T4 |
| Вероятность появления (Pj) | | 0,5 | 0,9 | 0,7 | 0,2 | 0,1 | 0,8 | 0,1 | 0,2 |
| Коэффициент  влияния (Кj) | | 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 аij 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 (Аj), 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:

(4.1)

Затем производится суммирование полученных оценок по строкам и столбцам матрицы, а также разработка выводов и рекомендаций. Преобразованная матрица представлена в таблице 4.5 [25].

Таблица 4.5 – Результирующая матрица SWOT-анализа ИС ДРДО

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Интенсивность (Аi) | Возможности (О) | | | | Угрозы (Т) | | | |
|  |  | О1 | О2 | О3 | O4 | Т1 | Т2 | Т3 | T4 |
| Вероятность появления (Pj) | | 0,5 | 0,9 | 0,7 | 0,2 | 0,1 | 0,8 | 0,1 | 0,2 |
| Коэффициент влияния (Кj) | | 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 | -11,25 | -17,5 | -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).

1. Самая слабая сторона - W1 - недостаточная известность, она создает значительные препятствия при достижении почти всех перечисленных возможностей.
2. 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.

# ЗАКЛЮЧЕНИЕ

В данной работе была рассмотрено проектирование и создание информационной системы для работы с документами об образовании – ИС ДРДО. Данная система предназначена для выпускников учебных заведений и работодателей, которые хотят удостовериться в подлинности документов соискателей.

Система спроектирована с использованием технологии Blockchain и максимально использует основные преимущества этой технологии, а именно:

* прозрачность реализации;
* простота объединения хранилищ данных;
* масштабируемость;
* надежность;
* доверие со стороны других государств.

Также было проведено сравнение проектируемой системы с существующей и использующей классическое хранилище данных – базы данных. В результате были выделены отличительные черты ИС ДРДО, позволяющие ей не только параллельно работать с существующей системой, но и, возможно, полностью заменить ее в будущем.

Были поставлены и решены такие задачи информационного менеджмента как управление капиталовложениями, управление персоналом и развитие информационной системы. В результате

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Федеральный реестр сведений документов об образовании и (или) о квалификации, документах об обучении [Электронный ресурс]. URL: <http://ic-dv.ru/uslugi/uslugi/fis_frdo/>.
2. Криптовалюта термины и определения [Электронный ресурс]. URL: <https://zona-1.ru/kriptovalyuta/kriptovalyuta-terminy-i-opredeleniya.html>.
3. Андреас Антонопулос: Mastering Bitcoin, 2nd Edition, 2017
4. Равал С. Децентрализованные приложения. Технология Blockchain в действии, 2016
5. Solidity — Solidity 0.4.20 documentation [Электронный ресурс] URL: <https://solidity.readthedocs.io/en/develop/>
6. Портал государственных услуг Российской Федерации [Электронный ресурс]. URL: <https://www.gosuslugi.ru/>
7. Обзор применения технологии блокчейн в государственном управлении [Электронный ресурс]. URL: <https://medium.com/@slavasolodkiy/обзор-применения-технологии-блокчейн-в-государственном-управлении-ac53602cec7f>
8. Документы об образовании и (или) квалификации. Документы об обучении [Электронный ресурс]. URL: <https://mti.edu.ru/entrance/diploma>
9. 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/>
10. Fast Ethereum RPC client for testing and development [Электронный ресурс]. URL: <https://github.com/trufflesuite/ganache-cli>
11. Lightweight Java and Android library for integration with Ethereum clients [Электронный ресурс]. URL: <https://github.com/web3j/web3j>
12. Remix – Solidity IDE [Электронный ресурс]. URL: <https://remix.ethereum.org/>
13. Smart Contracts — web3j 3.4.0 documentation [Электронный ресурс]. URL: <https://web3j.readthedocs.io/en/latest/smart_contracts.html#smart-contract-wrappers>
14. NinjaMock online wireframe and mockup tool [Электронный ресурс]. URL: <https://ninjamock.com/>
15. Ethereum Pet Shop -- Your First Dapp | Truffle Suite [Электронный ресурс]. URL: <http://truffleframework.com/tutorials/pet-shop>
16. Types — Solidity 0.4.24 documentation [Электронный ресурс]. URL: <http://solidity.readthedocs.io/en/v0.4.24/types.html>
17. Ethereum Virtual Machine (EVM) [Электронный ресурс]. URL: <https://opentutorials.org/course/2869/18360>
18. Integrating the Ethereum Blockchain into Java Apps [Электронный ресурс]. URL: <https://blogs.oracle.com/java/integrating-the-ethereum-blockchain-into-java-apps>
19. Чем технология блокчейн отличается от базы данных? [Электронный ресурс]. URL: <https://blockchainwiki.ru/chem-tehnologiya-blokchejn-otlichaetsya-ot-bazy-dannyh/>
20. Что такое Blockchain (блокчейн)? Технология, платформа, транзакции [Электронный ресурс]. URL: <https://mining-cryptocurrency.ru/blockchain/#i-4>
21. Информационный менеджмент - Информационные технологии в менеджменте [Электронный ресурс]. URL: <https://studme.org/62395/menedzhment/informatsionnyy_menedzhment>
22. Костров А.В. Основы информационного менеджмента. Москва, Финансы и статистика, 2001 – 336 с.
23. SWOT-анализ — Википедия [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/SWOT-анализ>
24. Гвозденко А.Н. Использование методики многофакторного SWOT-анализа для разработки стратегических направлений деятельности предприятий, 2006. – № 4.
25. Голубков Е.П. Основы маркетинга : учебник. – 3-е изд. – М. : Финпресс, 2008.