информационная система — библиотека

1. Выбор блокчейн-решения

Для реализации блокчейн-решения был выбран блокчейн Hyperledger Fabric.

Данная платформа поддерживает смарт-контракты — чейнкоды (chaincode), создаваемые на языках общего назначения, таких как Golang, JavaScript, Java, в отличие, от, например, Ethereum, в котором используется контрактно-ориентированный, ограниченный по функциональности язык Solidity.

Структура Hyperledger Fabric позволяет предприятиям настраивать сеть и использовать механизм консенсуса таким образом, который подходит им лучше, чем другие способы. Предприятие может выбрать модель, которая обеспечивает лучшую производительность его механизмов отказоустойчивости.

И так, Hyperledger Fabric будет использован в качестве блокчейн-решения так как:

1. Hyperledger Fabric — надежный реестр для хранения информации и работы с ней.
2. Это проверенный способ передачи данных.
3. Имеет высокую производительность и масштабируемость.
4. Он поддерживает многие языки программирования.
5. Он осуществляет безопасные сделки через smart-контакты.
6. Для нашей информационной системы не нужна мировая сеть, такая как Ethereum, поэтому мы используем локальную сеть.
7. Отсутствие комиссии.

Hyperledger Fabric позволяет осуществлять частные транзакции, что удобно в случае с содержанием данных в блокчейне о библиотеке, в которой есть читатель, берущий или дарящий книгу, и библиотекарь, отдающий эту книгу и ведущий учет. То есть глобализация в данном случае не важна.

Использование блокчейн-решения в такой информационной системе, как библиотека крайне важна. Потому что в библиотеке хранятся, как популярные и распространенные книги, так и рукописи, и манускрипты. Любой читатель может взять книгу и не вернуть, поэтому так важно, чтобы система вела учет приема/выдачи книг, хранила информацию обо всех читателях, книгах и последнем держателе конкретного экземпляра книги.

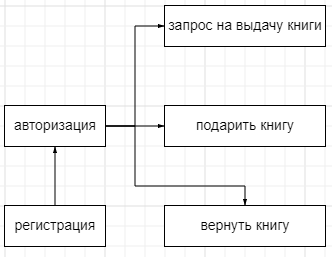
1. Техническое задание

В блокчейне содержатся сведения об абонентах библиотеки, имеющейся литературе и ее хранении. Необходимо вести учет выданной литературы. В каталоге на каждое название книги выписывается каталожная карточка со всеми реквизитами (обложка, уникальный идентификатор, автор, название книги, издатель, дата издания, сколько экземпляров есть в библиотеке). Для удобства создания БД составляются таблицы наличия книг (по пунктам каталожной карточки), причем обязательно указывается количество экземпляров. На каждый экземпляр книги выписывается формуляр с записями о выдаче читателю и возврате, и ФИО читателя. На каждого читателя заполняется абонементная карточка с перечнем полученных книг и ставится дата их выдачи и возврата.

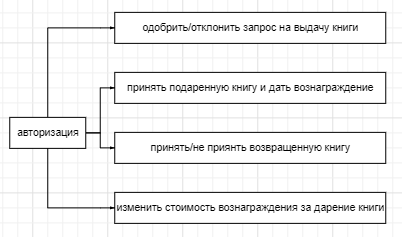
1. Основные функциональные модули системы

Модуль регистрации (рисунок 1) требуется для занесения данных о новом пользователе в систему. Только зарегистрированные пользователи имеют возможность воспользоваться работой сервиса. При регистрации требуется ввести свой логин (ФИО) и пароль.

Модуль авторизации читателя (рисунок 1) запрашивает логин пользователя и пароль. Если логин или пароль были введены неверно, или читатель не был зарегистрирован, то он не сможет зайти в личный кабинет. Модуль авторизации администратора (рисунок 2) запрашивает его пароль. Если пароль был введен неверно, то администратор не сможет зайти в свой личный кабинет.



1. Блок-схема читателя



1. Блок-схема администратора
2. Основные используемые механизмы блокчейн-системы

В основе блокчейн-составляющей решения (смарт-контракта) лежат функции: выдача/возврат/прием/дарение книги, регистрация пользователя, изменение стоимости вознаграждения за дарение книги.

Смарт-контракт – информация о пользователе, дате выдачи, возврата; об экземпляре книги, кто и когда его взял/вернул; о кол-ве конкретных книг, их авторе, годе издания, издательстве, их идентификаторе.

1. Стек технологий

Решение – веб-сайт.

HyperLedger, JavaScript, Golang – для написания смарт-контракта, основных функций и frontend-части сайта. Язык программирования Golang будет использован в данном решении, так как он наиболее удобен для написания смарт-контрактов в блокчейне Hyperledger fabric.

Figma – для создания макета будущего интерфейса блокченй-решения, так как является наиболее удобным, функциональным и популярным инструментом для создания и разработки макетов, дизайна будущих веб-приложений.

HTML, CSS – для верстки.

1. Задачи, которые необходимо решить для выполнения поставленного технического задания. Распределение задач между участниками

Для успешной реализации проекта требуется написать клиентскую часть приложения, серверную и «блокчейн-составляющую» (смарт-контракт).

Члены команды: Frontentd-разработчик, Backend-разработчик, дизайнер, блокчейн-разработчик, тестировщик.

Распределение по времени:

6.1) Frontentd-разработчик – 35 часов (Для верстки на HTML и CSS, отладки на JS и проверки работоспособности и удобности интерфейса).

6.2) Backend-разработчик – 60 часов (Для обеспечения работоспособности серверной части веб-приложения).

6.3) Блокчейн-разработчик – 40 часов для написания смарт контракта и проверки его работоспособности.

6.4) Дизайнер – 25 часов (Для создания макета интерфейса и разработки дизайна продукта).

6.5) Тестировщик – 10 часов (Проверка работоспособности всего приложения и удобности интерфейса).

7. Эскизы интерфейса

