|  |
| --- |
| Пермский филиал федерального государственного автономного образовательного учреждения высшего образования  «Национальный исследовательский университет  «Высшая школа экономики»  *Факультет экономики, менеджмента и бизнес-информатики* |
|  |
| Кочетов Денис Дмитриевич  **Проектирование информационной системы торговой компании на технологии блокчейн**  *Курсовая работа*  по направлению подготовки *38.03.05 Бизнес-информатика*  образовательная программа «Бизнес-информатика»   |  |  | | --- | --- | |  | Руководитель  Преподаватель кафедры информационных технологий в бизнесе  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  О.Б. Бартов | |

Пермь, 2020 год

Аннотация

Реализация информационной системы для торговой компании на технологии блокчейн.

Автор: Кочетов Денис Дмитриевич

Факультет экономики, менеджмента и бизнес-информатики. Третий курс, группа БИ-17-2, 2020 год. Кафедра информационных технологий в бизнесе.

В первой главе был проведен анализ предметной области, сформулирована постановка задачи проекта, описана предметная область. Были рассмотрены примеры платформ для разработки, выявлены их преимущества и недостатки.

Вторая глава – проектирование системы. Представлены и описаны происходящие бизнес-процессы, акторы, с помощью описания прецедентов и диаграмм UML будет представлен принцип работы системы. Результатом главы стали функциональные и не функциональные требования к системе.

В третьей главе, описывается интеграция системы, описаны структуры данных, инструменты для разработки и проведено тестирование приложения.

Оглавление

[1. Введение 5](#_Toc39520448)

[Глава 1. Анализ информационных систем на технологии блокчейн 7](#_Toc39520449)

[1.1. Описание технологии блокчейн 7](#_Toc39520450)

[1.1.1 Публичный блокчейн 8](#_Toc39520451)

[1.1.2 Приватный блокчейн 8](#_Toc39520452)

[1.1.2 Сравнение видов блокчейна 9](#_Toc39520453)

[1.2. Описание применения технологии блокчейн в бизнесе 9](#_Toc39520454)

[1.1.1. Документы, регулирующие предметную область 12](#_Toc39520455)

[1.2 Постановка задачи 12](#_Toc39520456)

[1.2.1 Назначение системы 12](#_Toc39520457)

[1.3 Анализ существующих систем 12](#_Toc39520458)

[1.3.1 Hyperledger fabric 12](#_Toc39520459)

[1.3.2 Exonum 14](#_Toc39520460)

[1.3.4 Сравнение функционала аналогов 15](#_Toc39520461)

[Глава 2. Проектирование информационной системы на технологии блокчейн 16](#_Toc39520462)

[2.1 Бизнес-процессы 16](#_Toc39520463)

[2.2 Акторы 18](#_Toc39520464)

[2.3 Требования к системе 18](#_Toc39520465)

[После постановки задачи у анализа аналогов, были предложены следующие требования к будущей системе. 18](#_Toc39520466)

[2.3.1 Нефункциональные требования 18](#_Toc39520467)

[2.3.2 Функциональные Требования 19](#_Toc39520468)

[2.3.3 Неформальные требования 19](#_Toc39520469)

[2.4 Описание прецедентов 20](#_Toc39520470)

[2.4.1 Диаграммы деятельности 23](#_Toc39520471)

[2.3. Выгрузка данных с фильтрацией для раздела «Отчеты» 26](#_Toc39520472)

[Глава 3. Интеграция системы в бизнес 28](#_Toc39520473)

[3.1 Структуры данных 28](#_Toc39520474)

[3.2 Инструменты 29](#_Toc39520475)

[3.3. Тестирование приложения 36](#_Toc39520476)

[Заключение 37](#_Toc39520477)

[Библиографический список 38](#_Toc39520478)

[Приложение A. Архитектура аналогов 40](#_Toc39520479)

[Приложение Б. Техническое задание 41](#_Toc39520480)

[Приложение В. Руководство пользователя 48](#_Toc39520481)

[Приложение Г. Пример интерфейса 51](#_Toc39520482)

1. Введение

На рынке существует огромное количество торговых компаний, конкуренция среди которых крайне высока. Возможный способ обойти конкурентов – внедрение новых технологий, дающих различные преимущества. Одной из таких технологий является блокчейн.

Технология блокчейн позволяет хранить данные в распределенном реестре с уверенностью, что никто не сможет изменить их без ведома остальных участников сети. Такая возможность делает систему гораздо прозрачней и надежней аналогов.

Бизнес модель торговой компании – это закупка товара и последующие за ней продажа и поставка. Процесс поставки товара зачастую осложняется в связи с невозможностью отследить возможные проблемы (товар может потеряться, сломаться, прийти не в том количестве или не того качество и т. п.) из-за недостаточной прозрачности. При этом любая поставка товара сопровождается большим количеством документов (счет-фактура, товарная накладная, доверенность на экспедитора), что замедляет и усложняет процесс для всех участников сделки. Эти проблемы и должна решить технология блокчейн.

*Объектом* исследования является возможное применение технологии блокчейн в информационной системе торговой компании.

*Предметом* исследования является внедрение технологии блокчейн для улучшения информационной системы торговой компании.

Целью данной работы является проектирование прототипа информационной системы для торговой компании на технологии блокчейн. Разработка прототипа подразумевает выполнение следующих задач:

Анализ предметной области. Изучение встроенных бизнес-процессов на основе существующих аналогов. Исследование возможных способов реализации системы.

Проектирование. Проектирование пользовательского интерфейса, создание диаграмм последовательностей, классов, состояний.

Интеграция. Интеграция системы в работу торговой компании.

В процессе выполнения работы будут применены следующие методы исследования:

1. Сбор, анализ, сравнение данных на этапе анализа.
2. Моделирование процессов на этапе проектирования.
3. Структурно-функциональный подход на этапе интеграции.

Результатом работы будет являться проект информационной системы для торговой компании на технологии блокчейн.

Глава 1. Анализ информационных систем на технологии блокчейн

В данной главе содержится информация о технологии блокчейн, ее видах, способах применения в бизнесе. Проведено сравнение различных блокчейн технологий, выделены основные преимущества и недостатки, а также информация о технологии блокчейн и ее применении.

* 1. Описание технологии блокчейн

Технология блокчейн – это распределенная база данных записей или общих государственных/частных реестров всех цифровых событий, которые были выполнены и распространены среди участвующих в блокчейне агентов. Блокчейн – это платформа проведения трансакций от равного к равному, для которой не требуется вовлечение третьей стороны. Для коммерческих целей различные юридические лица включены в процесс трансакций, они выступают в роли узлов. Процесс трансакций подтверждается с помощью криптографии. Записи о трансакциях хранятся и распространяются через распределенный реестр через всех участников сети. Блокчейн считается мета технологией, так как она является результатом интеграции нескольких других технологий, таких как база данных, программное приложение, сеть компьютеров, подключенных друг к другу, программная среда, на которой основана сеть, и инструменты для контроля над ней.[1] Технология блокчейн отличается от большинства существующих конструкций информационных систем включением четырех ключевых характеристики: децентрализация, безопасность, возможность аудита и интеллектуальное выполнение.

В блокчейне агент создает новую трансакцию для добавления в реестр. Эта новая трансакция транслируется на сеть для проверки и аудита. Как только большинство узлов в цепочке одобряют эту трансакцию в соответствии с предварительно определенными утвержденными правилами, эта новая транзакция добавляется в цепочку как новый блок. Запись об этой транзакции сохраняется в нескольких распределенных узлах для обеспечения безопасности. Между тем, смарт контракт, как важнейшая особенность технологии блокчейн, позволяет выполнение заслуживающих доверия трансакций без участия третьих лиц.

Основное различие между современным дизайном Интернета и технологиями блокчейна заключается в том, что Интернет был разработан для передачи информации, а не ценностей, и делиться копиями, а не оригинальной информацией. В блокчейне значение представлено в трансакциях, записанных в общем реестре и защищенных путем предоставления проверяемой записи трансакций с отметкой времени, которая предоставляет безопасную и проверяемую информацию. Эти трансакции появляются через процесс проверки, который согласуется с согласованными правилами сети. Как только новая запись проверена и добавлена ​​в блокчейн, несколько копий создаются децентрализованным способом для создания доверительной цепочки [2].

В различных блокчейнах бывают свои правила консенсуса, хэш-функции, количества участников и ролей участников. Чаще всего блокчейн делят на два типа – публичный и приватный.

* + 1. Публичный блокчейн

Блокчейн называется публичным, если все участники сети могут не только видеть и хранить все трансакции, но и участвовать в процессе консенсуса, то есть проверки блока трансакций. В публичном блокчейна отсутствует центральный регистр или доверенные третьи лица. Управлением публичных каналов занимается общество, поддерживающее принципы разработки с открытым исходным кодом. В такой системе, узлы сети, проверяющие трансакции, выбираются с помощью алгоритма, заложенным разработчиками, базированным на криптоэкономике, связанной с комбинацией материальной мотивации и криптографических механизмах верификации. Любой публичный блокчейн использует майнинг коинов или токенов [5].

Двумя наиболее популярными алгоритмами для определения консенсуса в публичных блокчейнах являются Proof of Work (подтверждение работой) и Proof of Stake (подтверждение долей).

1.1.2 Приватный блокчейн

Блокчейн называется приватным, если консенсус может быть достигнут только определенным ограниченным количеством участников. Доступ на запись трансакций выдается централизованно, чтение реестра может быть ограничено. Консенсус контролируется заранее выбранным набором узлов. Для консенсуса не требуется майнинг, алгоритм подтверждения работы, система вознаграждений.

Основные преимущества приватного блокчейна – низкая стоимость трансакций и большая пропускная способность (большее количество трансакций в секунду в сравнении с публичным блокчейном).

* + 1. Сравнение видов блокчейна

Для того, чтобы определить, какой блокчейн выбрать при разработке, надо решить, что важно для бизнеса. Для сравнительного анализа были выбраны критерии: скорость работы, безопасность, цена, прозрачность.

Таблица 1.1 — Сравнение видов блокчейна

|  | **Публичный блокчейн** | **Приватный блокчейн** |
| --- | --- | --- |
| **Скорость работы** | Низкая из-за алгоритма консенсуса | Быстрая |
| **Безопасность** | Риск атаки 51% | Из-за валидации проверенными узлами безопасность выше; существует опасность централизации |
| **Цена** | Большие энергозатраты для создания блоков | Для создания блоков майнинг не требуется, следовательно затрат меньше |
| **Прозрачность** | Вся информация доступна всем пользователям | Сохраняет коммерческую тайну |

* 1. Описание применения технологии блокчейн в бизнесе

Технология блокчейн может упростить сразу несколько задач, с которыми сталкиваются торговые компании.

1. **Обработка платежей**.

Расчеты в криптовалюте сокращают затраты для участников сделки, уменьшают конечную стоимость товаров и делают электронную торговлю проще и быстрее, убирая посредников в лице банков. Обработка платежей может быть интегрирована непосредственно в счета-фактуры, которые с помощью смарт-контрактов автоматически оплачиваются при поступлении груза.

Для предотвращения повторного использования одной и той же монеты (double-spending) все трансакции за всё время работы криптовалюты хранятся в цепочке блоков. В каждом блоке содержится заголовок, в котором находятся хэш предыдущего блока, и список трансакций. Из-за этой особенности сохранения информацию о трансакциях нельзя подделать, заменив в ней один из блоков, так как хэш блока всегда зависит от хэша предыдущего блока в цепочке. При изменении одного из блоков, придется изменять все идущие после него. Каждый участник сети хранит всю цепочку блоков, децентрализуя сеть. Данные о трансакциях не шифруются, и любой участник сети может вручную проследить их все.

1. **Документооборот**.

Распределительный реестр и неизменность записей гарантирует точность информации и защищают от несанкционированных изменений и человеческих ошибок. Кроме того, смарт-контракты совместно с технологиями интернета вещей могут автоматически заполнять документы, ставить штампы, заключать и исполнять контракты, а также проводить по ним финансовые расчеты.

Смарт-контракт – это набор правил, условий, который помогает определить сеть взаимодействия акторов между собой и с системой. Смарт-контракты влияют на обмен сетевыми данными между участниками и постоянно улучшают этот процесс. Например, сертифицирующие и стандартизирующие организации проводят цифровую проверку профилей участников сети и их продуктов. У участников и продуктов в сети есть цифровые профили, в которых есть их описание, локация, сертификаты и связи с другими участниками и продуктами. Каждый участник цепочки поставок может ввести необходимую информацию о продукте и его статусе в блокчейн сети.

Управление смарт-контрактами и правила процессов в цепочке поставок на основе блокчейн могут управлять сертификацией и утверждением участников, а также процессами, к которым им разрешен доступ, и которые необходимы для выполнения. Изменения данных об участнике сети могут происходить в зависимости от типа цепочки поставок, локации или иных условий, определенных смарт-контрактом. Участники сети не могут изменить правила без какой-либо формы консенсуса.

Еще один пример применения смарт-контрактов – это закупки. Интеллектуальный договор между двумя торговыми партнерами может на законных основаниях обновлять автоматизированную запись того, какие товары были куплены, проданы и доставлены в режиме реального времени конечными пользователями по всей сфере бизнеса [1].

1. **Контроль геолокации товара**.

Используя RFID-метки, можно в реальном времени отслеживать местонахождение товара. Это поможет выявить проблемы логистики и наладить более эффективные пути доставки.

1. **Контроль качества**.

RFID-метки могут отслеживать не только геолокацию, но и температуру, влажность, скорость и резкие движения (встряску), что важно при доставке товаров, требующих строгого соблюдения температурных режимов (медикаменты, биоматериалы, продукты питания).

RFID-метки, как известно, обеспечивают множество преимуществ для управления цепочками поставок, включая улучшенные активы управление, улучшенный инвентарный контроль, сохранение экономических показателей, повышение доступности продукта и выполнения поставки, снижение стоимости рабочей силы, сокращение времени работы, эффективность обработки материалов, повышение производительности процесса, улучшение обслуживания клиентов, включая возврат и управление отзывами, а также обслуживание и гарантия авторизация, повышенная видимость в пути, подтверждение соблюдение нормативных требований посредством записей о цепочке поставок [4].

У блокчейна есть потенциал обеспечить доверие в закупочном цикле. Прозрачность блокчейна в данной сфере наиболее привлекательна. В традиционной сфере поставок, которую гарантирует множество контрактов, есть период ожидания платежа между доставкой товара, составлением счет-фактуры и окончательным проведением платежа. Приложение, в котором есть возможность заключения смарт контрактов, поможет организациям снизить издержки, возникающие из-за периода ожидания платежа, благодаря интеграции доставки и оплаты в цифровые контракты, которые распространяются на предприятия и интегрируются с партнерами по логистике и банками.

Интеграция также позволит сократить требования к оборотному капиталу и упростить финансовые операции, ведущие к устойчивости цепочки поставок [3].

1.1.1. Документы, регулирующие предметную область

В конце 2017 года министерство промышленности и торговли Российской Федерации выпустило приказ о создании комитета по стандартизации “Программно-аппаратные средства технологий распределенного реестра и блокчейн”. [10]

ISO все еще в разработке единых стандартов для блокчейн реестров.

1.2 Постановка задачи

Задача работы — спроектировать информационную систему для отслеживания цепочки поставок с помощью технологии блокчейн.

1.2.1 Назначение системы

Система должна обеспечивать возможность занесения данных о поставках в блокчейн реестр, отслеживания товаров во время поставки, а также получение информации о происхождении товара и проделанного им пути. Система должна быть одновременно безопасной и прозрачной, а также удобной в пользовании как для поставщиков, так и для конечных потребителей.

1.3 Анализ существующих систем

Многие крупные компании занимаются внедрением технологии блокчейн. Аналоги будут сравниваться по сложности развертывания сети, стоимости интеграции и качестве реализации существующих систем.

1.3.1 Hyperledger fabric

Hyperledger Fabric – это платформа для создания распределенных реестров, проведения смарт-контрактов, использующая знакомые и проверенные технологии, с модульной архитектурой, допускающей интеграцию различных дополнительных функций.

В Hyperledger Fabric различают два вида участников сети (пиров). Первый вид – подтверждающие участники (валидаторы) – это узлы сети (ноды), отвечающие за процесс консенсуса, подтверждающие трансакции и поддерживающие работоспособность сети. Второй вид – участники, не подтверждающие трансакции – это узлы, который выступают в качестве прокси для соединения клиентов к валидаторам. Эти узлы не проводят трансакции, но могут их проверять.

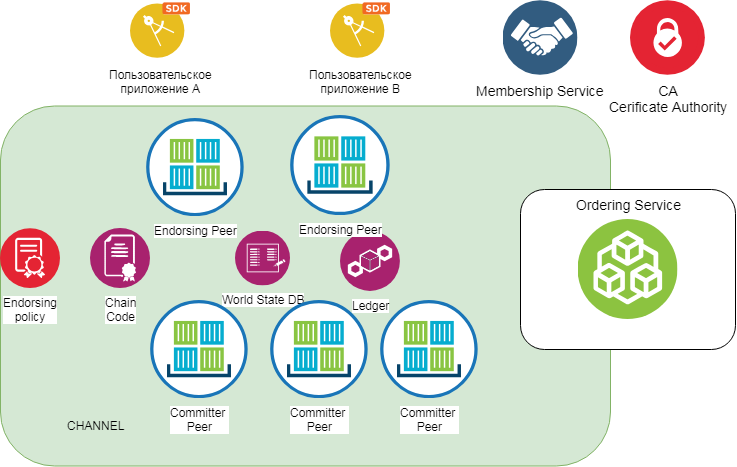


Рисунок 1 — Архитектура Hyperledger

Некоторые особенности Hyperledger Fabric:

* Приватный блокчейн с обязательной конечностью.
* Проводит произвольные смарт-контракты, реализованные на языке Go.
* Определяемые пользователями смарт-контракты упакованы в Docker контейнеры.
* Протокол консенсуса можно подключать, в данный момент применяется протокол Византийских генералов, также доступны протоколы SIEVE, NOOPS.
* Безопасность подтверждается центрами сертификации для TLS сертификатов, сертификатов регистрации и сертификатов трансакций.

Hyperledger Fabric – это проект с открытым исходным кодом, основанный на стандартах, поддерживающих произвольные (определяемые пользователями) смарт-контракты, поддерживающих возможность улучшения защиты и дополнительной идентификации участника сети. Проект в данный момент развивается и активно разрабатывается организацией Hyperledger Project, основанной при поддержке Linux Foundation, в которую входят такие компании, как IBM, Microsoft, Intel и другие.

1.3.2 Exonum

Exonum – это платформа с открытым исходным кодом для создания блокчейн приложений. Exonum может быть использован для создания распределенных реестров с криптографической защитой в любых сфера, в том числе финансовой, государственной и юридической. Платформа ориентирована на создания приватных блокчейнов.

Exonum использует язык программирования Rust для обеспечения максимальной безопасности; использует сервис-ориентированную архитектуру для обеспечения гибкости, расширяемости и модульности платформы; верификация клиентов основана на битовой схеме обязательства (дерево Меркла и префиксное дерево Меркла) для обеспечения прозрачности системы и безопасности клиентов.

Exonum предоставляет инструмент разработки Java Binding вместе с набором средств разработки. Это позволяет строить кастомные сервисы на языке Java – одном из самых популярных языков разработки поверх основанной на Rust платформе Exonum.

Смарт-контракты на Exonum способны обработать до 5000 трансакций в секунду с пингом в 0,5 секунд.

Программируются смарт-контракты на Rust и Java. Благодаря принципам языка Rust, смарт-контракты вписываются напрямую в ноды(узлы) сети. Таким образом гарантируется безопасное использование памяти и высокая скорость выполнения контрактов. Контракты на Java можно добавлять в произвольное время после запуска сети.

Exonum использует алгоритм шифрования SHA-256 для хеширование трансакций, для защиты цифровой подписи используется алгоритм Ed25519. Хэш вычисляется на полных узлах при помощи библиотеки libsodium, для легких клиентов используется sha.js. Шифрование сообщений между узлами реализовано на протоколе Noise [16].

Для обеспечения производительности в 5000 трансакций в секунду, необходим сервер с довольно скромными по меркам блокчейн сетей характеристиками.

* Пропускная способность сети: 1 Мб при задержке в 100 мс.
* Процессор — 2 ядра, 3 ГГц.
* Оперативная память — 1–2 Гб.
* Внешняя память — 30 Гб (120 оборотов в секунду).
* Количество нод (узлов) — от четырех до семи.

Можно использовать сервера Amazon EC2 t2. small.

1.3.4 Сравнение функционала аналогов

Ниже проанализированы аналоги существующих платформ для разработки распределенных реестров.

Таблица 1.2 — Сравнение аналогов блокчейна для бизнеса

| **Блокчейн платформа** | **Достоинства** | **Недостатки** |
| --- | --- | --- |
| Hyperledger Fabric | Подробная документация, поддержка крупными корпорациями, успешное применение в крупном бизнесе | Дорогостоящая интеграция, высокие системные требования |
| Exonum | Низкие системные требования; сервис анкоринга, предотвращает сговор владельцев узлов с целью изменения истории данных; русскоязычные разработчики | Примеры успешного применения пока только в государственной сфере |

Глава 2. Проектирование информационной системы на технологии блокчейн

В данной главе необходимо определить основные бизнес-процессы, выявить акторов и составить требования к системе.

2.1 Бизнес-процессы

При изучении деятельности торговой компании в соответствии с системой на технологии блокчейн были выделены следующие основные бизнес-процессы. Добавление новых пользователей, товаров в распределенный реестр; трансакция – изменение статуса товара – добавление в блок; возможность отслеживания пути товара, перемещения добавляются в реестр так же, как и цену, дату, локацию, качество и заключение смарт-контрактов.

Основные бизнес-процессы были декомпозированы, результаты представлены на диаграмме прецедентов (рис. 2.1).

На диаграмме представлены как традиционные процессы торговой компании, так и новые процессы, связанные с особенностями технологии блокчейн, которым предстоит внедрение в систему.



Рисунок 2.1 — Диаграмма прецедентов

2.2 Акторы

Для проектирования системы были выбраны следующие акторы: торговая компания, поставщики, клиенты-покупатели.

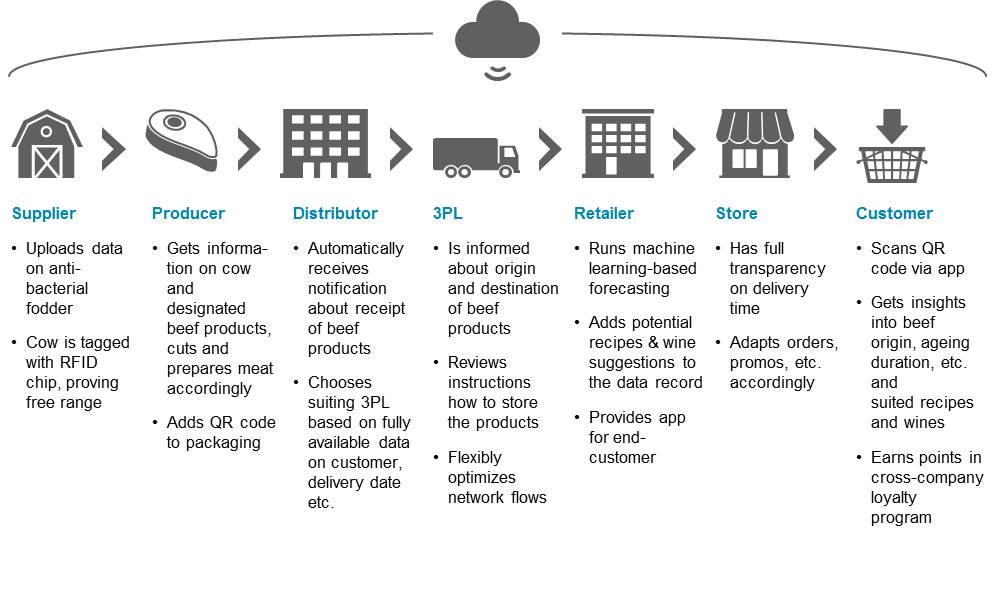


Рисунок 2.2 — Акторы

* 1. Требования к системе

После постановки задачи у анализа аналогов, были предложены следующие требования к будущей системе.

2.3.1 Нефункциональные требования

К нефункциональным требованиям относится:

1. Наличие удобного интерфейса.
2. Разграничение прав пользователей, сокрытие коммерческой тайны.
3. Обеспечение безопасности данных благодаря криптографическому алгоритму.
4. Язык программирования Rust, база данных RocksDB, Java SDK для создания смарт-контрактов.
   * 1. Функциональные Требования
5. Ввод, вывод, хранение, обновление данных, возможность создания смарт-контрактов.
6. Возможность вывода отчета о пути товара:

* Microsoft Excel.
* Microsoft Word.



Рисунок 2.3 — Диаграмма прецедентов

2.3.3 Неформальные требования

Добавление участника в сеть, добавление информации о товаре в реестр, составление смарт контракта, заключение смарт контракта, отслеживание информации о поставке.

2.4 Описание прецедентов

После составления общей диаграммы прецедентов некоторые из них были рассмотрены по отдельности, для них были построены диаграммы в нотации UML; также диаграммы были описаны с помощью таблиц, представленных ниже.

Таблица 2.1 — Описание прецедентов

|  | **Название прецедента** | **Описание прецедента** |
| --- | --- | --- |
| 1 | Добавить запись | Необходим для ввода новых данных сотрудниками |
| 2 | Редактировать запись | Необходим для изменения данных сотрудниками |
| 3 | Удалить запись | Необходим для удаления данных сотрудниками |
| 4 | Найти запись | Необходим для поиска данных сотрудниками и другими пользователями |
| 5 | Выгрузить данные в Excel | Необходим для экспорта данных в Excel |

Таблица 2.2 — Прецедент Добавить запись

|  |  |
| --- | --- |
| **Акторы** | **Сотрудник** |
| Предусловия | Открытие приложения, необходимость добавить запись |
| Основной поток | Сотрудник нажимает кнопку «Добавить запись» и вводит необходимые данные |
| Альтернативный поток | При вводе некорректных данных выводится сообщение об ошибке |
| Постусловия | В случае успешного ввода данных в базу добавляется новая запись, иначе выводится сообщение об ошибке |

Таблица 2.3 — Прецедент Редактировать запись

|  |  |
| --- | --- |
| **Акторы** | **Сотрудник** |
| Предусловия | Открытие приложения, необходимость изменить запись |
| Основной поток | Сотрудник нажимает кнопку «Изменить запись» и изменяет необходимые данные |
| Альтернативный поток | При вводе некорректных данных выводится сообщение об ошибке |
| Постусловия | В случае успешного ввода данных в базу запись изменяется, иначе выводится сообщение об ошибке |

Таблица 2.4 — Прецедент Найти запись

|  |  |
| --- | --- |
| **Акторы** | **Сотрудник** |
| Предусловия | Открытие приложения, необходимость найти запись |
| Основной поток | Сотрудник нажимает кнопку «Найти запись», вводит необходимые данные, и система выдает результат |
| Альтернативный поток | При вводе некорректных данных выведется сообщение об ошибке |
| Постусловия | Система выведет записи, удовлетворяющие запросу, в ином случае будет выведено сообщение об их отсутствии |

Таблица 2.5 — Прецедент Удалить запись

|  |  |
| --- | --- |
| **Акторы** | **Сотрудник** |
| Предусловия | Поиск нужной записи, необходимость ее удаления |
| Основной поток | Сотрудник ищет необходимую запись, нажимает кнопку «Удалить запись» |
| Альтернативный поток | Отсутствует |
| Постусловия | Запись удаляется из базы |

Таблица 2.5 — Прецедент Выгрузить данные в Excel»

|  |  |
| --- | --- |
| **Акторы** | **Сотрудник** |
| Предусловия | Выбор нужных данных |
| Основной поток | Сотрудник выбирает необходимый фильтр данных, нажимает кнопку «Экспорт в Excel» |
| Альтернативный поток | При отсутствии необходимых данных будет выведено соответствующее сообщение |
| Постусловия | Данные будут экспортированы в файл Excel |

2.4.1 Диаграммы деятельности



Рисунок 2.4 — Диаграмма прецедента «Добавить запись»



Рисунок 2.5 — Диаграмма прецедента «Найти запись»



Рисунок 2.6 — Диаграмма прецедента «Редактировать запись»



Рисунок 2.7 — Диаграмма прецедента «Удалить запись»



Рисунок 2.8 — Диаграмма прецедента «Экспорт данных в Excel»

Приложение было поделено не несколько основных подзадач, была спроектирована блок-схема с типичным сценарием поведения пользователя, она представлена на рисунке 2.1.

2.3. Выгрузка данных с фильтрацией для раздела «Отчеты»

Ниже (рис. 2.9) представлена блок-схема, показывающая алгоритм выгрузки данных в отчеты.



Рисунок 2.9 — Блок-схема, описывающая алгоритм выгрузки данных из БД



Рисунок 2.10 — Блок-схема, описывающая алгоритм выгрузки данных с фильтрами для раздела «Отчеты»

Глава 3. Интеграция системы в бизнес

Задачи главы: описать используемые инструменты и структуры данных, интерфейсы, инструкции.

3.1 Структуры данных

Для каждой из созданных таблицы в базе данных был создан класс с соответствующими свойствами.

Таблица 3.1 — Типы данных

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Имя атрибута** | **Идентификатор атрибута** | **Тип данных** | **Обязательность заполнения (да/нет/условие)** | **Описание** |
| Код участника сети | PEER\_ID | Целое число | Да | Номер участника сети |
| Описание участника сети | PEER\_INFO | Строка | Да |  |
| Код товара | PRODUCT\_ID | Целое число | Да | Номер товара |
| Наименование товара | PRODUCT\_NAME | Строка | Да | Название товара |
| Дата изготовления товара | PRODUCTION\_NAME | Дата | Да | Дата изготовления |
| Срок хранения товара | PRODUCT\_SHELF\_LIFE | Строка | Нет | Срок хранения |
| Координаты | LAST\_SCAN\_COORD | Строка | Да | GPS координаты последнего сканирования |
| Изготовитель товара | PRODUCT\_MANUFACTURER | Строка | Да | Наименование физического или юридического лица, производящего товар |
| Заказчик | CUSTOMER\_NAME | Строка | Да | Наименование физического или юридического лица, купившего товар |

* 1. Инструменты

Для разработки будут применяться следующие инструменты:

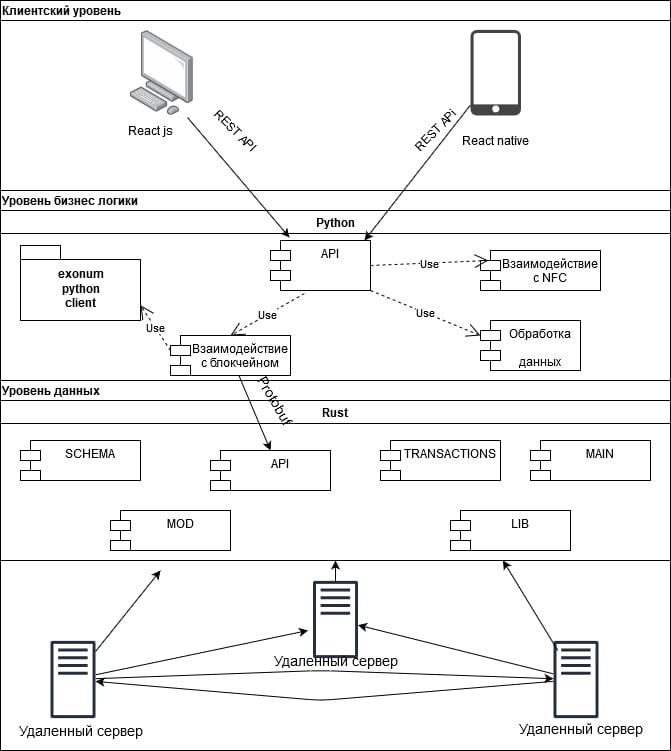


Рисунок 3.1 — Диаграмма развертывания

1) Схема базы данных:  Начало формы

Конец формы

Начало формы

Конец формы

|  |  |
| --- | --- |
|  | use exonum::merkledb::{ |
|  | access::{Access, FromAccess, RawAccessMut}, |
|  | Group, ObjectHash, ProofListIndex, RawProofMapIndex, |
|  | }; |
|  | use crate::product::Product; |
|  | use crate::manufacturer::Manufacturer; |
|  | use crate::owner::Owner; |
|  | use exonum::crypto::{Hash, PublicKey}; |
|  |  |
|  | /// Products table name |
|  | pub const Products\_TABLE: &str = "counterfeitBC.product"; |
|  |  |
|  | #[derive(Debug)] |
|  | pub struct Schema<T> { |
|  | view: T, |
|  | } |
|  |  |
|  | impl<T> AsMut<T> for Schema<T> { |
|  | fn as\_mut(&mut self) -> &mut T { |
|  | &mut self.view |
|  | } |
|  | } |
|  |  |
|  | impl<T> Schema<T> |
|  | where |
|  | T: IndexAccess, |
|  | { |
|  | /// Creates a new schema from the database view. |
|  | pub fn new(view: T) -> Self { |
|  | Schema { view } |
|  | } |
|  | /// Returns `ProofMapIndex` with pipe types. |
|  | pub fn products(&self) -> ProofMapIndex<T, PublicKey, Product> { |
|  | ProofMapIndex::new(Products\_TABLE, self.view.clone()) |
|  | } |
|  |  |
|  | /// Returns pipe type for the given public key. |
|  | pub fn product(&self, pub\_key: &PublicKey) -> Option<Product> { |
|  | self.products().get(pub\_key) |
|  | } |
|  | /// Returns the state hash of service. |
|  | pub fn state\_hash(&self) -> Vec<Hash> { |
|  | vec![self.queues().object\_hash()] |
|  | } |
|  | ///method for adding attributes to queu |
|  | pub fn create\_product ( |
|  | &mut self, |
|  | product\_public\_key: &PublicKey, |
|  | manufacturer\_public\_key: PublicKey, |
|  | owner\_public\_key: PublicKey, |
|  | product\_name: str, |
|  | last\_scan\_coord: str, |
|  | product\_info: str, |
|  | product\_nfc: str, |
|  | production\_date: str, |
|  | ) { |
|  | let product = { |
|  | Product:: new( |
|  | product\_public\_key, |
|  | manufacturer\_public\_key, |
|  | owner\_public\_key, |
|  | product\_name, |
|  | last\_scan\_coord, |
|  | product\_info, |
|  | product\_nfc, |
|  | production\_date, |
|  | ) |
|  | }; |
|  | self.products().put(key, profile); |
|  | } |

2) Описание передачи данных в прото файле

|  |
| --- |
| syntax = "proto3"; |
|  | package counterfeit; |
|  | // library. |
|  | // System user |
|  | import "exonum/crypto/types.proto"; |
|  |  |
|  | message Manufacturer { |
|  | exonum.crypto.PublicKey manufacturer\_public\_key = 1; |
|  | string manufacturer\_name = 2; |
|  | string manufacturer\_info = 3; |
|  | } |
|  | message Owner { |
|  | exonum.crypto.PublicKey owner\_public\_key = 1; |
|  | string owner\_name = 2; |
|  | string owner\_info = 3; |
|  | } |
|  | message Product { |
|  | exonum.crypto.PublicKey product\_public\_key = 1; |
|  | exonum.crypto.PublicKey manufacturer\_public\_key = 2; |
|  | /// Public key of the owner |
|  | exonum.crypto.PublicKey owner\_public\_key = 3; |
|  | /// Name of the product. |
|  | string product\_name = 4; |
|  | /// Coordinates of the last scan |
|  | string last\_scan\_coord = 5; |
|  | /// Information about the product |
|  | string product\_info = 6; |
|  | /// Id of the nf chip |
|  | uint64 product\_nfc = 7; |
|  | /// Date of production |
|  | string production\_date = 8; |
|  | } |

1. Сериализация данных

use exonum\_build::{ProtoSources, ProtobufGenerator};

fn main() {

ProtobufGenerator::with\_mod\_name("protobuf\_mod.rs")

.with\_input\_dir("src/proto")

// The exact list of included files may differ depending on

// what Protobuf messages defined in Exonum you need to use.

.with\_includes(&[

"src/proto".into(),

ProtoSources::Exonum,

ProtoSources::Crypto,

])

.generate();

}

1. Добавление нового продукта

use exonum::{crypto::PublicKey};

use exonum\_proto::ProtobufConvert;

use super::proto;

/// Product information stored in the database.

#[derive(Clone, Debug, ProtobufConvert)]

#[protobuf\_convert(source = "proto::Product", serde\_pb\_convert)]

pub struct Product {

/// Public key of the product

pub product\_public\_key: PublicKey,

/// `PublicKey` of the manufacturer.

pub manufacturer\_public\_key : PublicKey,

/// Public key of the owner

pub owner\_public\_key: PublicKey,

/// Name of the product.

pub product\_name: String,

/// Coordinates of the last scan

pub last\_scan\_coord: String,

/// Information about the product

pub product\_info: String,

/// Id of the nf chip

pub product\_nfc: u64,

/// Date of production

pub production\_date: String,

}

impl Product {

/// Create new product

pub fn new(

&product\_public\_key: &PublicKey,

manufacturer\_public\_key : PublicKey,

owner\_public\_key: PublicKey,

product\_name: &String,

last\_scan\_coord: &String,

product\_info: &String,

product\_nfc: u64,

production\_date: &String,

) -> Self {

Self {

product\_public\_key,

manufacturer\_public\_key ,

owner\_public\_key,

product\_name: product\_name.to\_owned(),

last\_scan\_coord: last\_scan\_coord.to\_owned(),

product\_info: product\_info.to\_owned(),

product\_nfc,

production\_date: production\_date.to\_owned(),

}

}

}

1. Регистрация пользователя

const mapState = (state: IState) => ({

isLoading: state.user.isLoadingUser,

error: state.user.error

})

const mapDispatch = {

loginUser,

}

const connector = connect(mapState, mapDispatch)

type PropsFromRedux = ConnectedProps<typeof connector>

type ILoginPageProps = PropsFromRedux & {}

function LoginPage(props: ILoginPageProps) {

const [email, setEmail] = useState<string>('')

const [password, setPassword] = useState<string>('')

const onSubmit = () => {

props.loginUser({

email,

password

})

}

return (

<EmptyLayout>

<div className={formStyles['form-container']}>

<div className={formStyles['form-container\_\_header']}>

<h1>Войти</h1>

</div>

<Grid container spacing={4} alignItems="flex-end">

<Grid item>

<Face />

</Grid>

<Grid item md={true} sm={true} xs={true}>

<TextField id="username" label="Email" type="email" fullWidth autoFocus required />

</Grid>

</Grid>

<Grid container spacing={4} alignItems="flex-end">

<Grid item>

<Fingerprint />

</Grid>

<Grid item md={true} sm={true} xs={true}>

<TextField id="username" label="Пароль" type="password" fullWidth required />

</Grid>

</Grid>

<Grid container justify="flex-end" style={{ marginTop: '24px' }}>

<Button color="primary">

<Link style={{color: 'inherit', textDecoration: 'none'}} to="/signup">Регистрация</Link>

</Button>

<Button onClick={() => onSubmit()} style={{ marginLeft: '24px' }} variant="contained" color="primary">

Войти

</Button>

</Grid>

</div>

</EmptyLayout>

);

}

export default connector(LoginPage)

3.3. Тестирование приложения

При разработке приложения учитывались возможные проблемы последовательного исполнения функций и наложены ограничения на их исполнение.

Таблица 3.1 — Тестирование входа пользователя

|  |  |  |  |
| --- | --- | --- | --- |
| № | Исходные данные | Ожидаемый результат | Полученный результат |
| 1 | Ввод логина в неверном формате login | Вывод сообщения об ошибке | Сообщение об ошибке: "Неверный формат логина" |
| 2 | Ввод неверного логина: login@mail.ru | Вывод сообщения об ошибке | Сообщение об ошибке: "Введите логин и пароль" |
| 3 | Ввод пароля без логина | Вывод сообщения об ошибке | Сообщение об ошибке: "Введите логин и пароль" |
| 4 | Ввод верного логина и неверного пароля: логин: user1@mail.ru, пароль: User | Вывод сообщения об ошибке | Сообщение об ошибке: "Ошибка неверно введен логин или пароль" |
| 5 | Верный ввод данных: логин: user1@mail.ru, пароль: User\_1 | Переход к начальной странице | Переход к начальной странице |

Для всех данных было проведено тестирование добавления и редактирования. Ошибок выявлено не было.

Таблица 3.2 — Тестирование добавления и редактирования данных

|  |  |  |  |
| --- | --- | --- | --- |
| № | Исходные данные | Ожидаемый результат | Полученный результат |
| 1 | Ввод не всех данных | Данные не добавляются, пользователь остается в том же окне | Данные не добавляются, пользователь остается в том же окне |
| 2 | Верный ввод всех данных | Добавление элемента в базу данных | Добавление элемента в базу данных |

Было проведено тестирование удаления данных из таблиц. Все удаления происходят корректно. Ошибок не выявлено.

Заключение

В результате работы были решены поставленные задачи и достигнута главная цель – проектирование информационной системы для торговой компании на технологии блокчейн.

В ходе анализа были построены модели бизнес-процессов в нотации UML, рассмотрены аналоги платформ для разработки блокчейн реестров.

Были разработаны требования к информационной системе, нормализованы исходные данные, спроектированы запросы к базе данных.

Были спроектированы интерфейсы приложения, основные функции были кратко описаны и протестированы.

Созданы формы Windows Forms для взаимодействия пользователя с приложением.

Составлено техническое задание, а также руководство пользователя.

Была спроектирована информационная система для торговой компании на технологии блокчейн.

Библиографический список

1. Saberi, S., Kouhizadeh, M., Sarkis, J., & Shen, L. (2018). Blockchain technology and its relationships to sustainable supply chain management. International Journal of Production Research, 1–19. doi:10.1080/00207543.2018.1533261
2. Kamble, S., Gunasekaran, A., & Arha, H. (2018). Understanding the Blockchain technology adoption in supply chains-Indian context. International Journal of Production Research, 1–25. doi:10.1080/00207543.2018.1518610
3. Kamble, S., Gunasekaran, A., & Arha, H. (2018). Understanding the Blockchain technology adoption in supply chains-Indian context. International Journal of Production Research, 1–25. doi:10.1080/00207543.2018.1518610
4. Kapoor, G., Zhou, W., & Piramuthu, S. (2009). Challenges associated with RFID tag implementations in supply chains. European Journal of Information Systems, 18(6), 526–533. doi:10.1057/ejis.2009.41
5. Dominique Guegan. Public Blockchain versus Private blockhain. 2017. ffhalshs-01524440f
6. Saveen A. Abeyratne1 , Radmehr P. Monfared2 Blockchain Readu Manufacturing Supply Chain Using Distributed Ledger - International Journal of Research in Engineering and Technology eISSN: 2319-1163 | pISSN: 2321-7308
7. Kari Korpela, Jukka Hallikas, Tomi Dahlberg Digital Supply Chain Transformation toward Blockchain Integration - Conference: Hawaii International Conference on System Sciences (HICSS), At Big Island, Hawaii, Volume: 50
8. [Sara Saberi](https://www.tandfonline.com/author/Saberi%2C+Sara), [Mahtab Kouhizadeh](https://www.tandfonline.com/author/Kouhizadeh%2C+Mahtab), [Joseph Sarkis](https://www.tandfonline.com/author/Sarkis%2C+Joseph) Blockchain technology and its relationships to sustainable supply chain management [International Journal of Production Research](https://www.tandfonline.com/toc/tprs20/current)Volume 57, 2019 - [Issue 7](https://www.tandfonline.com/toc/tprs20/57/7)
9. Руководство по программированию на C# [Электронный ресурс]. - Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/csharp/programming-guide/
10. Терехов А. Н. Технология программирования: учебное пособие. / А. Н. Терехов: – 2-е изд. – М.: БИНОМ. Лаборатория знаний, 2012. —149 c.
11. Троелсен, Эндрю. Язык программирования C# 5.0 и платформа .NET 4.5, 6-е изд.: Пер. с англ.— М.: ООО “И. Д. Вильямс”, 2013. — 1312 с.
12. Шилдт, Герберт. C# 4.0: полное руководство.: Пер. с англ. — М.: ООО "И.Д. Вильямс", 2011.—1056 с.
13. Линн, Бейли. Изучаем SQL. — СПб.: Питер, 2012. — 592 с.
14. Советов Б.Я. Базы данных: теория и практика: учебное пособие /Б.Я. Советов, В.В. Цехановский, В.Д. Чертовский: – 2-е изд. – М.: Высш. шк.,2007. —463 с.
15. МИНИСТЕРСТВО ПРОМЫШЛЕННОСТИ И ТОРГОВЛИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ АГЕНТСТВО ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ ПРИКАЗ от 15 декабря 2017 г. N 2831 О СОЗДАНИИ ТЕХНИЧЕСКОГО КОМИТЕТА ПО СТАНДАРТИЗАЦИИ "ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ТЕХНОЛОГИЙ РАСПРЕДЕЛЕННОГО РЕЕСТРА И БЛОКЧЕЙН"
16. Exonum [Электронный ресурс]. – Режим доступа: <https://exonum.com/doc>
17. Rust [Электронный ресурс]. – Режим доступа: <https://doc.rust-lang.org/book/>
18. Java Development Kit [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/en/java/javase/13/docs/api/index.html>

**Приложение A. Архитектура аналогов**

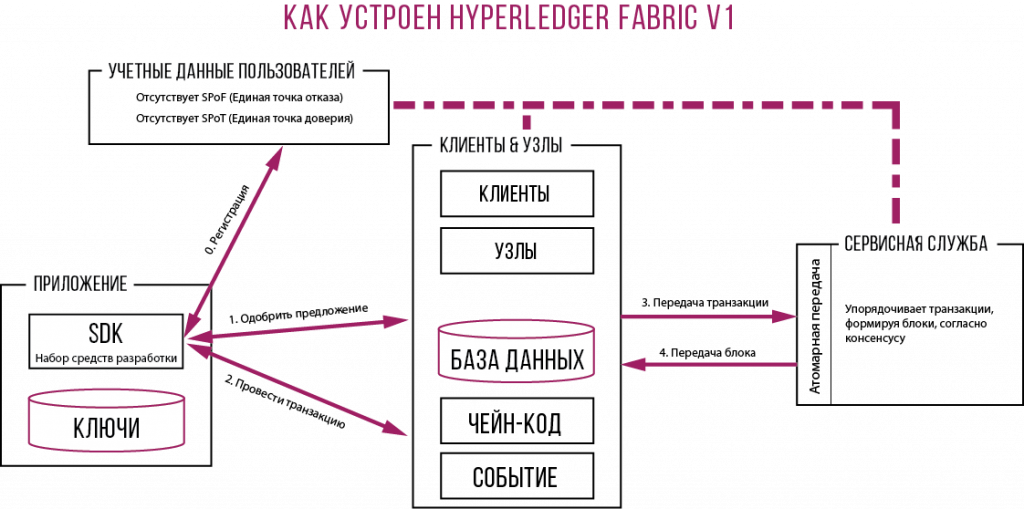


Рис. А.1. —  Архитектура Hyperledger Fabric

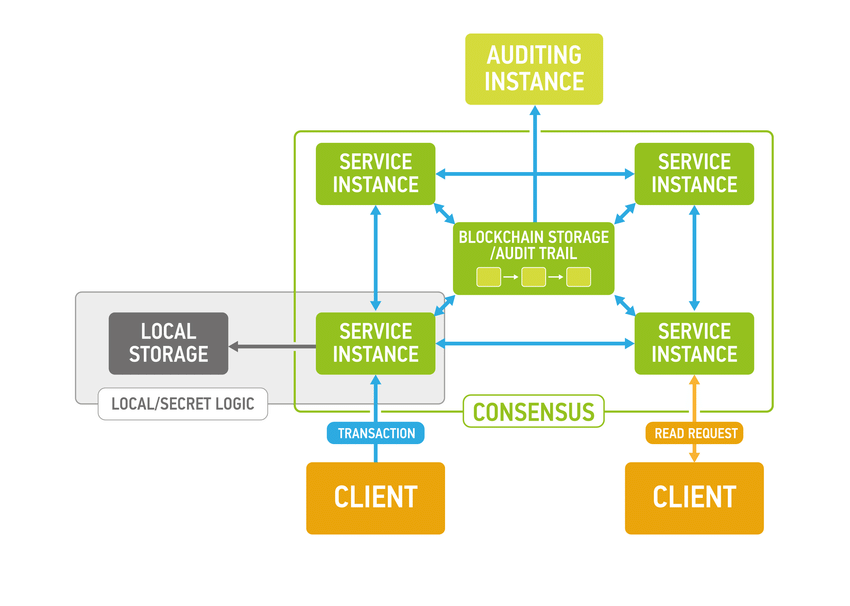


Рис. А.2 — Архитектура Exonum

Приложение Б. Техническое задание

Пермский филиал федерального государственного автономного

образовательного учреждения высшего образования

«Национальный исследовательский университет

«Высшая школа экономики»

|  |  |
| --- | --- |
| УТВЕРЖДАЮ |  |
| Руководитель: преподаватель кафедры информационных технологий в бизнесе |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /О.М. Бартов/  «\_\_\_»\_\_\_\_\_\_\_\_ 2020 г. |  |

Техническое задание информационной системы для торговой компании на технологии блокчейн

НИУ ВШЭ-БИ-17-2

Пермь, 2020

Аннотация

Настоящий документ является техническим заданием к информационной системе для торговой компании на технологии блокчейн.

В данном документе приводится следующая информация:

1. Введение
2. Назначение разработки
3. Требования к системе
4. Стадии и этапы разработки

Г.1. Введение

Г.1.1. Наименование системы

Наименование программы: "Информационная для торговой компании на технологии блокчейн".

Г.1.2. Назначение и область применения

Программа предназначена для создания распределенного реестра с информацией о товарах и участников сети.

Г.2. Назначение разработки

В разделе представлена информация о функциональном и эксплуатационном назначении программы.

Г.2.1. Функциональное назначение

Функциональным назначением системы является предоставление пользователям сети возможности внесения новых записей, обновление данных и поиск в реестре, а также заключение смарт-контрактов.

Г.2.2. Эксплуатационное назначение

Информационная система должна эксплуатироваться в профильных подразделениях на объекте Заказчика.

Конечными пользователями программы должны являться сотрудники профильных подразделений объектов Заказчика.

Г.3. Требования к системе

В данном разделе представлены различного рода требования к программе (функциональные, к надежности, к эксплуатации и т. д.).

Г.3.1. Требования к функциональным характеристикам

К функциональным требованиям системы относятся возможность выполнять следующие действия с хранящимися данными о товарах:

1. 1) Ввод
2. 2) Вывод
3. 3) Редактирование
4. 4) Удаление

Помимо этого, необходимо реализовать запросы и вывод их результатов, такие как поиск по:

1. 1) Региону
2. 2) Учебному заведению/месту работу
3. 3) Году рождения (при поиске несовершеннолетних и т. п.)

Г.3.2. Требования к надежности

Г.3.2.1. Требования к обеспечению надежного функционирования программы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением Заказчиком совокупности организационно-технических мероприятий, перечень которых приведен ниже:   
а) организацией бесперебойного питания технических средств;   
б) использованием лицензионного программного обеспечения;   
в) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г.   
Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;   
г) регулярным выполнением требований ГОСТ 51188-98. Защита информации. При вводе пользователем некорректных данных ему будут не доступны любые действия над базой данных, а также будет выведено сообщение с ошибкой.

Г.3.2.2. Время восстановления после отказа

Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбоем (не крахом) операционной системы,   
не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.   
Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Г.3.3. Условия эксплуатации

Г.3.3.1. Климатические условия эксплуатации

Климатические условия эксплуатации, при которых должны обеспечиваться заданные характеристики, должны удовлетворять требованиям,   
предъявляемым к техническим средствам в части условий их эксплуатации.

Г.3.3.2. Требования к квалификации и численности персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 2 штатных единиц — системный администратор и конечный пользователь программы — оператор.   
Системный администратор должен иметь высшее профильное образование и сертификаты компании-производителя операционной системы. В перечень задач, выполняемых системным администратором, должны входить:   
а) задача поддержания работоспособности технических средств;   
б) задачи установки (инсталляции) и поддержания работоспособности системных программных средств — операционной системы;   
в) задача установки (инсталляции) системы.   
г) задача создания резервных копий базы данных.

Г.3.4. Требования к составу и параметрам технических средств

В состав технических средств должен входить IВМ-совместимый персональный компьютер (ПЭВМ), включающий в себя:   
3.4.1. операционную систему Windows 2003 или более современную версию;   
3.4.2. Microsoft SQL Server 2000

Г.3.5. Требования к информационной и программной совместимости

Г.3.5.1. Требования к информационным структурам и методам решения

База данных работает под управлением Microsoft SQL Server. Используется многопоточный доступ к базе данных. Необходимо обеспечить одновременную работу системы с той же базой данной модулей экспорта внешних данных.

Г.3.5.2. Требования к исходным кодам и языкам программирования

Дополнительные требования не предъявляются.

Г.3.5.3. Требования к программным средствам, используемым программой

Системные программные средства, используемые программой, должны быть представлены лицензионной локализованной версией операционной системы Windows 2003 и Microsoft SQL Server 2000.

Г.3.5.4. Требования к защите информации и программ

В информационной системе ведется работа с персональными данными клиентов, поэтому при работе с ней необходимо соблюдать следующий законодательный акт:

3.5.4.1 Постановление Правительства Российской Федерации от 1 ноября 2012 г. №1119 г. "Об утверждении требований к защите персональных данных при их обработке в информационных системах персональных данных"

Г.3.5. Специальные требования

Специальные требования к данной программе не предъявляются.

Г.4. Стадии и этапы разработки

Г.4.1. Стадии разработки

Разработка информационной системы торговой компании на технологии блокчейн, должна быть проведена в несколько стадий:

1. Анализ предметной области, включающий в себя описание предметной области, описание автоматизируемых процессов, описание данных.
2. Анализ разрабатываемой базы данных: нормализация данных, описание таблиц и форматов данных.
3. Проектирование и разработка базы данных.
4. Проектирование приложения.
5. Проектирование интерфейса пользователя.
6. Разработка приложения учета земельных участков.
7. Тестирование и отладка.

Г.4.2. Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. Проектирование базы данных.
2. Разработка приложения баз данных.
3. Проектирование пользовательского интерфейса.
4. Разработка запросов и отчетов.
5. Разработка программной документации.
6. Испытания программы.

На стадии внедрения должен быть выполнен этап разработки, подготовка и передача программы.

Г.4.3. Содержание работ по этапам

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:   
1. постановка задачи;   
2. определение и уточнение требований к техническим средствам;   
3. определение требований к программе;   
4. определение стадий, этапов и сроков разработки программы и документации на неё;   
5. согласование и утверждение технического задания.   
На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.   
На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с требованиями к составу документации.   
На этапе подготовки и передачи программы должна быть выполнена работа по подготовке и передаче программы и программной документации в эксплуатацию на объектах Заказчика.

Приложение В. Руководство пользователя

Пермский филиал федерального государственного автономного

образовательного учреждения высшего образования

«Национальный исследовательский университет

«Высшая школа экономики»

|  |  |
| --- | --- |
| УТВЕРЖДАЮ |  |
| Руководитель: преподаватель кафедры информационных технологий в бизнесе |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ / О.Б. Бартов /  «\_\_\_»\_\_\_\_\_\_\_\_ 2020 г. |  |

Руководство пользователя информационной системы торговой компании на технологии блокчейн.

Пермь, 2020

Аннотация

Настоящий документ является руководством пользователя по информационной системы.

В данном руководстве приводится следующая информация:

1. Введение.
2. Назначение и условия применения.
3. Подготовка к работе.
4. Описание операций.
5. Аварийные ситуации.
6. Рекомендации по освоению.

Е.1. Введение

Е.1.1. Область применения программы

Программу могут использовать большинство пользователей, имеющих примитивные навыки работы с компьютером.

Е.1.2. Возможности программы

Программа создана с целью автоматизировать и сделать прозрачней процесс отслеживания товара.

Е.1.3 Требуемый уровень знаний

Пользователь информационной системы должен иметь примитивный опыт работы с компьютером и обладать следующими знаниями:

1. Скачивание программы.
2. Установка программы.

Е.2. Назначение и условия применения

Работа с программой возможна в любое время.

Е.3. Подготовка к работе

Е.3.1. Состав и содержание дистрибутивного носителя данных

Для работы с программой требуется следующее программное обеспечение:

1. Браузер (для скачивания установочного файла или исходного кода).
2. Среда программирования Visual Studio (если пользователь имеет исходный код).
3. Офисное приложение Microsoft Excel (для импорта отчетов).

Е.3.2. Порядок загрузки данных и программ

Для работы с программой необходимо выполнить следующие действия:

1. Требуется скачать установочный файл или исходный код программы, отправленный разработчиком на электронную почту.
2. Открыть файл или исходный код.
3. Запустить программу двойным щелчком на ярлык либо нажатием кнопки «Пуск» в среде программирования Visual Studio.

Е.4. Описание операций

Е.4.1. Выполняемые функции и задачи

Таблица. Е.4.1. Выполняемые функции и задачи

Е.4.2. Описание операций технологического процесса обработки данных, необходимых для выполнения задач

Задача: «Выбор режима «Гость»

Условия, при соблюдении которых возможно выполнение операции:

1. Приложение запущено.

Подготовительные действия:

1. Пользователь выбрал режим «Гость».

Основные действия в требуемой последовательности:

Ввести требуемые данные в поля ввода, нажать кнопку найти (см. рис. Е.4.1).

Задача: ««Добавить запись»

Подготовительные действия:

Основные действия в требуемой последовательности:

Ввести требуемые данные в поля ввода, нажать кнопку «Добавить» (см. рис. Е.4.2).

Задача: «Создание отчетов».

Условия, при соблюдении которых возможно выполнение операции:

Пользователь выбрал режим «Сотрудник», «Создание отчета».

Основные действия в требуемой последовательности:

Выбрать необходимый раздел для создания отчета, нажать кнопку «Добавить» (см. рис. Е.4.3).

Е.5. Аварийные ситуации

Таблица Е.5.1. Аварийные ситуации

|  |  |  |  |
| --- | --- | --- | --- |
| Класс ошибки | Ошибка | Описание ошибки | Требуемые действия пользователя при возникновении ошибки |
| Ошибки с информацией. | Данные отсутствуют. | Выбранных данных в системе не существует. | Добавить данные. |
| Выводится удаляемая информация. | Удаленная информация должна отсутствовать в системе. | Обратиться в службу поддержки. |
| Не выводится добавленная информация. | Добавленная информация должна выводиться. |

Е.6. Рекомендации по освоению.

Рекомендаций к освоению программа не требует.

Приложение Г. Пример интерфейса

Изображение выглядит как снимок экрана, еда, фрукт

Автоматически созданное описание

Рисунок Г1 — Интерфейс система