

МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)

Факультет _____ ИКСС
Кафедра _____ ПИВТ

Допустить к защите

Заведующий кафедрой

(подпись) _____ (Ф.И.О.)

« _____ » _____ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка программного обеспечения

блокчейн-сети IOTA

для обеспечения взаимодействия

устройств интернета вещей

(тема ВКР)

Направление/специальность подготовки

09.03.04 Программная инженерия

(код и наименование направления/специальности)

Направленность (профиль)

Разработка программного обеспечения

инфокоммуникационных сетей и систем

(наименование)

Квалификация _____ Бакалавр

(наименование квалификации в соответствии с ФГОС ВО)

Студент:

Багмут В.В ИКПИ-81

(Ф.И.О., № группы)

(подпись)

Научный руководитель:

д.т.н., доцент Киричек Р. В.

(учёная степень, учёное звание, Ф.И.О.)

(подпись)

Санкт-Петербург, 2022

работа написана мной самостоятельно

работа не содержит неправомерных заимствований

работа может быть размещена в электронно-библиотечной системе университета

(дата)

(подпись)

(Ф.И.О. студента)

Текст ВКР размещен в электронно-библиотечной системе университета.

Руководитель отдела комплектования библиотеки _____

(Ф.И.О.)

(дата)

(подпись)

Коэффициент оригинальности ВКР _____ %.

Проверил: _____

(должность, Ф.И.О.)

(дата)

(подпись)

**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ
КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

Факультет _____ ИКСС _____ Кафедра _____ ПИВТ _____
Направление (специальность) _____ 09.03.04 Программная инженерия _____

(код и наименование)

Утверждаю:

Зав. кафедрой _____

(Ф.И.О., подпись)

« _____ » _____ 20__ г.

ЗАДАНИЕ

на выполнение выпускной квалификационной работы (ВКР)

1. Студент _____ Багмут Владислав Владимирович _____ № группы ИКПИ-84
(фамилия, имя, отчество)

2. Руководитель _____ Киричек Руслан. Валентинович., д.т.н., доцент _____
(фамилия, имя, отчество, должность, уч. степень и звание)

3. Квалификация _____ Бакалавр _____
(наименование в соответствии с ФГОС ВО)

4. Тема ВКР _____ Разработка программного обеспечения
_____ блокчейн-сети IOTA
_____ для обеспечения взаимодействия
_____ устройств интернета вещей

утверждена приказом ректора университета от «14» _____ апреля _____ 2022 г.
№407/КС

5. Исходные данные (технические требования): _____ Visual Studio Code
(solidity,js,cmd,sh),IOTA(wasp node,Goshimer,), Docker desktop, Git bush.

6. Содержание работы (анализ состояния проблемы, проведение исследований, разработка, расчеты параметров, экономическое обоснование и др.):

Введение,

Обзор предметной области и анализ существующих решений,

Определение требований к разрабатываемому приложению,

Техническая реализация,

Демонстрация работы приложения

7. Вид отчетных материалов, представляемых в ГЭК (пояснительная записка, перечень графического материала, отчет о НИР, технический проект, образцы и др.) пояснительная записка, презентация, приложения, визуализация работы программы, адрес сайта для просмотра работоспособности скрипта ПО.

8. Консультанты по ВКР с указанием относящихся к ним разделов

Раздел	Консультант	Подпись, дата	
		Задание выдал	Задание принял
1. Обзор предметной области и анализ существующих решений	д.т.н., доцент Киричек Р. В.		
2. Определение требований к разрабатываемому приложению	д.т.н., доцент Киричек Р. В.		
3. Техническая реализация	д.т.н., доцент Киричек Р. В.		
4. Демонстрация работы приложения	д.т.н., доцент Киричек Р. В.		

Дата выдачи задания «_____» _____ 20__ г.

Дата представления ВКР к защите «_____» _____ 20__ г.

Руководитель ВКР _____
(подпись)

Студент _____
(подпись)

КАЛЕНДАРНЫЙ ПЛАН

№ п/п	Наименование этапов выпускной квалификационной работы (ВКР)	Срок выполнения этапов ВКР	Примечание
1.	Постановка цели выполнения ВКР и задач	23.04.2022	
2.	Работа с теоретическим материалом	23.04.2022 — 31.05.2022	
3.	Сбор информации, необходимой для написания работы		
4.	Систематизация и обработка материалов ВКР		
5.	Анализ полученных в работе результатов, обобщение		
6.	Подготовка отчетных материалов, представляемых в государственную экзаменационную комиссию, доклада к защите и презентации	01.06.2022 — 21.06.2022	
7.	Консультации с руководителем ВКР		
8.	Представление выполненной ВКР руководителю для подготовки отзыва.		
9.	Подготовка к защите ВКР, включая подготовку к процедуре защиты и процедуру защиты		

Студент

(подпись)

Руководитель ВКР

(подпись)

РЕФЕРАТ

В выпускной квалификационной работе рассматривается тема «Разработка программного обеспечения блокчейн-сети IOTA для обеспечения взаимодействия устройств интернета вещей».

Выпускная квалификационная работа содержит: 85 страниц.

В первом разделе приведен обзор предметной области и существующих решений.

Во втором разделе сформулированы требования к приложению.

В третьем разделе приведены детали технической реализации.

В четвертом разделе продемонстрирована работа приложения.

ПО, СМАРТ КОНТРАКТЫ, САЙТ ПО, ВИЗУАЛИЗАЦИЯ РАБОТЫ ПО ДЛЯ ОБЕСПЕЧЕНИЯ ВЗАИМОДЕЙСТВИЯ IoT УСТРОЙСТВ С ПОМОЩЬЮ БЛОКЧЕЙН СЕТИ IOTA.

Методы проведения работы

Теоретические: анализ и синтез, обобщение, классификация, прогнозирование, объяснение,

Практические (эмпирические): сравнение, моделирование, наблюдение

Частные: интеллектуальный, исследовательский, прогнозный анализ

Результаты работы и новизна

создано ПО для взаимодействия IoT устройств и сайт ПО, который позволяет выполнять смарт контракты на языке Solidity.

Сайт ПО, представляет собой сайт для бронирования жилья с функциями: бронь определенного жилья на определенный день недели, просмотр данных (пользователя, общих данных), очистка данных (пользователя, общих данных), просмотр баланса и вывод средств. Оплата осуществляется с помощью кошелька MetaMask, в любой удобной пользователю крипто валюте.

ПО для взаимодействия IoT устройств, представляет собой визуальную демонстрацию ПО, связывающего между собой электромобиль и зарядную станцию, связь происходит с помощью блокчейн сети IOTA и оплаты.

Концепция проекта

ПО для отельного бизнеса, основанного на анонимности постояльцев, анонимность происходит за счет развёртывание ПО в блокчейн сети, о пользователи из данных хранится только его адрес кошелька. В номер постояльцы смогут попадать с помощью второго ПО, который развёртывается в блокчейн-сети IOTA и позволит нам создать ПО для взаимодействия электронного ключ-замка с любым устройством, к которому подключается MetaMask. В итоге мы получим полностью жизнеспособный проект со своим сайтом, приемом и обработкой платежей и ключ замками, который сможем продавать отелям и компаниям.

Так же можно создать второй сайт для частных лиц, на котором пользователи смогут размещать объявления о сдаче в аренду своего жилья.

Области применения результатов

Области в сфере бронирования жилья и отельного бизнеса, возможно использование части ПО в области зарядных станций для электромобилей.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	15
I. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ И АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ	17
1.1 Обзор предметной области	17
1.1.1 Введение в предметную область IOTA (MIOTA)	18
1.1.2 Введение в предметную область IoT	25
1.1.3 Введение в предметную область Blockchain	28
1.1.4 Примеры областей применения IoT	31
1.1.5 Заключение	33
1.2 Анализ существующих решений	34
1.2.1 Платформы для бронирования отелей в Блокчейн-сети	34
Travala.com (TravelbyBit Booking.com)	34
1.2.2 Электронный замок 360lock	38
1.2.3 Зарядная станция для электромобиля	38
1.3 Выводы по разделу	39
2.1 Требования к функциям	40
2.1.1 Функциональная структура приложения	40
2.1.2 Функции сайта ПО	41
2.1.3 Функции кнопок сайта	41
2.1.4 Функциональные возможности кода	42
Основные функции блоков кода приведены в таблице 2.3.	42
2.2 Требования к эргономике и технической эстетике	43
2.2.1 Общие требования к внешнему виду	43
2.2.2 Требования по обработке исключительных ситуаций	43
2.2.3 Требования по вводу и контролю данных	43
2.2.4 Требования по организации экранных форм	43
2.2.5 Требования к техническому обеспечению	44
2.3 Выводы по разделу	44

III ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ	45
3.1 Описание необходимых библиотек и разработанных модулей	45
3.2 Результаты разработки	46
3.3 Выводы по разделу.....	51
IV ДЕМОНСТРАЦИЯ РАБОТЫ ПРИЛОЖЕНИЯ.....	52
4.1 Интеграционное тестирование	52
1.4 Выводы по разделу	59
ЗАКЛЮЧЕНИЕ	60
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	61
Приложения	62

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Blockchain- выстроенная по определённым правилам непрерывная последовательная цепочка блоков (связный список), содержащих информацию. Связь между блоками обеспечивается не только нумерацией, но и тем, что каждый блок содержит свою собственную хеш-сумму и хеш-сумму предыдущего блока. Изменение любой информации в блоке изменит его хеш-сумму. Чтобы соответствовать правилам построения цепочки, изменения хеш-суммы нужно будет записать в следующий блок, что вызовет изменения уже его собственной хеш-суммы. При этом предыдущие блоки не затрагиваются. Если изменяемый блок последний в цепочке, то внесение изменений может не потребовать существенных усилий. Но если после изменяемого блока уже сформировано продолжение, то изменение может оказаться крайне трудоёмким процессом. Дело в том, что обычно копии цепочек блоков хранятся на множестве разных компьютеров независимо друг от друга

Терминал- Терминал Windows — это современное ведущее приложение для предпочитаемых вами оболочек командной строки, таких как Командная строка, PowerShell и Bash (через подсистему Windows для Linux (WSL)).

Виртуальная машина- программная и/или аппаратная система, эмулирующая аппаратное обеспечение некоторой платформы (guest — гостевая платформа) и исполняющая программы для guest-платформы на host-платформе (host — хост-платформа, платформа-хозяин) или виртуализирующая некоторую платформу и создающая на ней среды, изолирующие друг от друга программы и даже операционные системы (см.: песочница); также спецификация некоторой вычислительной среды (например: «виртуальная машина языка программирования Си»)

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

ИОТ - концепция сети передачи данных между физическими объектами («вещами»), оснащёнными встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой[1]. Предполагается, что организация таких сетей способна перестроить экономические и общественные процессы, исключить из части действий и операций необходимость участия человека.

ИОТА- криптовалюта с открытым исходным кодом, DLT (распределённая бухгалтерская книга) на основе DAG (направленный ациклический граф), предназначена для защищённой и эффективной передачи данных и платежей без комиссий между устройствами в интернете вещей.

JS (JavaScript) - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript (стандарт ECMA-262).

Solidity - объектно-ориентированный, предметно-ориентированный язык программирования[2] самовыполняющихся контрактов для платформы Ethereum.

Docker — программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений. Позволяет «упаковать» приложение со всем его окружением и зависимостями в контейнер, который может быть развёрнут на любой Linux-системе с поддержкой контрольных групп в ядре, а также предоставляет набор команд для управления этими контейнерами.

MetaMask - это программный кошелек для криптовалюты, используемый для взаимодействия с блокчейном Ethereum. Он позволяет пользователям получить доступ к своему кошельку Ethereum через

расширение браузера или мобильное приложение, которое затем можно использовать для взаимодействия с децентрализованными приложениями.

High-mobility - Берлинский стартап High Mobility специализируется на разработке программного обеспечения для подключенных автомобилей.

Visual Studio Code (VS Code) — редактор исходного кода, разработанный Microsoft для Windows, Linux и macOS. Позиционируется как «лёгкий» редактор кода для кроссплатформенной разработки веб- и облачных приложений.

Bash — усовершенствованная и модернизированная вариация командной оболочки Bourne shell. Одна из наиболее популярных современных разновидностей командной оболочки UNIX.

Goshimmer — это программное обеспечение для работы нод без Координатора.

Wasp — это программное обеспечение узла, которое служит проверкой смарт-контракта в составе комитета, используя виртуальную машину. Несколько узлов Wasp соединяются и образуют комитет валидаторов. Когда они достигают консенсуса по изменению состояния виртуальной машины, они привязывают это изменение состояния к клубку IOTA, делая его неизменным.

ВВЕДЕНИЕ

Интернет вещей (англ. internet of things, IoT) — концепция сети передачи данных между физическими объектами («вещами»), оснащёнными встроенными средствами и технологиями для взаимодействия друг с другом или с внешней средой. Предполагается, что организация таких сетей способна перестроить экономические и общественные процессы, исключить из части действий и операций необходимость участия человека.

Интерес ученых и специалистов области IT к IoT технологиям растет с каждым годом. Все из-за того, что они позволяют устройствам связываться, анализировать и обмениваться данными о физическом окружающем мире с помощью облачных программных платформ и других сетей. Возрастает потребность в специалистах этой области, а вместе с тем и в упрощении процесса поиска наиболее жизнеспособных гипотез.

IoT используются для решения многих реальных задач: заторов на дорогах, взаимодействия с городскими службами, экономического развития, участия граждан в жизни города, а также общественной безопасности и охраны. Ядром в IoT технологиях выступают протоколы связи, которые связывают между собой устройства интернета вещей. Мир идёт к открытым стандартам данных, но унифицированного подхода пока нет. Есть ряд протоколов, но нет единой платформы для соединения между собой устройств от всех возможных производителей. Совместимость устройств — это основная задача, которую нужно решить, для повсеместного распространения решений IoT. Блокчейн-сеть IOTA была создана для решения проблемы: недостатка проработки стандартов для данных и унификации данных (единой платформы для соединения между собой устройств от всех возможных производителей). Так же таких проблем как: масштабирование, безопасность, стоимость и архитектура, которые присущи централизованным IoT сетям.

Целью данной работы является приложение для отдельного бизнеса, а так же отдельная часть ПО может использоваться для зарядных станций

электромобилей. Приложение предназначено для решения нескольких проблем безопасность, анонимность, стоимость, децентрализованность.

В качестве задач, которые решало бы данное приложение, были выбраны: жизнеспособный сайт для бронирования и оплаты брони жилого пространства (оплата брони должна происходить в любой удобной крипто валюте по выбору заказчика), так же создание ПО, для взаимодействия клиента с ключ замками с помощью идентификатора кошелька.

Для достижения целей работы необходимо выполнить следующие задачи:

1. Обзор предметной области и анализ существующих решений;
2. Написание формализованных требований;
3. Создание настольного приложения;
4. Визуализация тестирования моделей машинного обучения в пределах установленных задач.

I. ОБЗОР ПРЕДМЕТНОЙ ОБЛАСТИ И АНАЛИЗ СУЩЕСТВУЮЩИХ РЕШЕНИЙ

1.1 Обзор предметной области

ИОТА – революционная криптовалюта и платформа, созданная специально для концепции Интернета Вещей (Internet of Things). Она не похожа ни на один другой проект или коин, что делает ее уникальной и очень многообещающей. ИОТА способна стать тем транзакционным топливом, которое обеспечит реализацию умных предприятий с участием машин, объединенных в одну сеть.

ИОТА – это и криптовалюта, и система для осуществления мгновенных микроплатежей без какой-либо комиссии. Этим она серьезно отличается от других монет, так как здесь возможна транзакция размером в 1 coin или, грубо говоря, 0.001 цента в долларовом эквиваленте.

С помощью ИОТА можно обеспечить выполнение самых маленьких задач в объединённой сети: открутить кран в ванной, наклеить бирку, да хоть сдуть пылинку. И, благодаря таким особенностям, ИОТА может быть интегрирована в Интернет Вещей, выполняя транзакции между точками.

ИОТА и Интернет Вещей

Перспективность ИОТА измеряется перспективностью сферы, которую призвана обслуживать эта платформа. Еще в 2015 году индустрия Интернета Вещей «оценивалась» в 1,6 триллионов долларов. К 2025 году ожидается, что отрасль вырастет до 6,2 триллионов долларов, а количество активных устройств превысит отметку в 50 миллиардов.

Рост популярности Интернета вещей

Под Internet of Things подразумевают сеть, в которой физические предметы взаимодействуют друг с другом за счет встроенных технологий. Сейчас полноценной сети нет – есть, скорее, автономные очаги. Возьмите хотя бы смарт-квартиры, где свет, электричество и отопление регулируется автоматически. А ведь квартиры можно объединять в дома, дома в города, а города в целую сеть.

Интернет Вещей способен изменить экономические и социальные процессы, во многом рационализировав их. Впервые эту концепцию предложил исследователь MIT Кевин Эштон в своем докладе по модернизации логистической системы Procter & Gamble. За два десятилетия бэкграунд изменился – теперь количество устройств, подключенных к Интернет, измеряется миллиардами, появились такие вещи, как облачные вычисления, IPv6, возросла роль беспроводных сетей, появились многочисленные сенсорные датчики и девайсы.

ИОТА способна связать воедино практически все процессы в экосистеме IoT за счет настройки цепей транзакций и способности проводить микротранзакции в огромных количествах. Проблем с масштабируемостью у Tangle, в отличие от blockchain, нет. Причем платформа реализована так, что устройство для взаимодействия с другими узлами не должно иметь бесперебойного доступа к Интернету. Для некоторых «машин» достаточно будет подключаться раз в месяц или даже в год – все зависит от функционала. Это позволит экономить заряд батареи или даже электричество.

1.1.1 Введение в предметную область ИОТА (MIOTA)

ИОТА (MIOTA) — это проект, созданный немецкими разработчиками.

Представляет собой криптовалюту, разработанную на базе распределенного реестра, предназначенного непосредственно для использования в системе Интернета Вещей (Internet of Things (IoT)).

Основные задачи и цели этой криптовалюты заключаются в:

Обеспечении безопасных транзакций между «вещами» (т.е. гаджетами) в режиме реального времени. Архитектура платежей ИОТА была разработана на базе системы «machine-to-machine (M2M)» и направлена исключительно на проведение транзакций в Интернете Вещей. Уникальность криптовалюты и ее одно из самых главных отличий от других монет заключается в том, что

она позволяет совершать микро-транзакции (минимум \$0,001) без уплаты комиссии за обработку платежа.

Стать основой будущей промышленной революции, которая будет заключаться в массовом применении Интернета Вещей. Проект ИОТА публично поддержали известные криптобиржи, а также некоторые крупные корпорации.

Кратко объясним, что такое Интернет Вещей. Интернет Вещей (Internet of Things) — в будущем будет представлять собой сеть физических устройств (гаджетов), автомобилей, бытовой техники (например, холодильников, морозильников, систем отопления, смарт-устройств), которой можно будет управлять при помощи смартфонов, планшетов или ПК.

В долгосрочной перспективе ИОТА может воплотить в жизнь создание таких проектов как «умные» города, дома, системы транспорта.

История создания

История ИОТА началась в 2015 году с проведением ICO, которое продолжалось с 24 ноября по 20 декабря 2015 года. Тогда команде удалось привлечь 934 BTC (около \$420000 в долларовом эквиваленте). Несмотря на то, что сейчас это кажется не таким большим достижением, тогда это было отличным результатом, так как культура ICO не была так развита, как сейчас, и в новые криптопроекты инвестировали в большей степени криптоэнтузиасты.

Тема обсуждения проекта на [bitcointalk](#) была создана 21 октября 2015 года. Однако первоначальная идея возникла еще раньше — в 2014 году, когда Сергей Иванчегло разработал новый тип микропроцессора под названием JINN, работа которого основана на третичных вычислениях. По неподтвержденным данным позже JINN был интегрирован в ИОТА.

После окончания ICO ИОТА начался период бета-тестирования, который завершился лишь к концу 2016 года. Хотя в процессе бета-тестирования ИОТА и функционировала уже как платформа для проведения микро-платежей, саму криптовалюту можно было купить лишь «с рук».

Листинг на биржах и обменниках был отложен до того момента, когда завершится период бета-тестирования и будет представлен полноценный продукт.

Это произошло в 2017 году в июне этого года одна из самых популярных бирж Bitfinex добавила торговые пары BTC/IOTA, ETH/IOTA. Команда представила рабочую платежную систему, что поспособствовало листингу криптовалюты на большинстве популярных криптобирж.

Несмотря на популярность монеты и ее востребованность среди инвесторов, команда продолжает активно работать над улучшением технологии.

Технология

Сразу следует сказать, что технология, на которой основана работа IOTA, отличается от принципа работы любой другой криптовалюты, функционирующей на блокчейне. Проще говоря, IOTA использует не блокчейн, а особую технологию под названием Tangle.

Что такое Tangle?

Tangle — это технология распределенного реестра, основанная на направленном ациклическом графе (directed acyclic graph (DAG)), который предполагает развитие в одном направлении — от более старых транзакций к более новым.

Если говорить более простым языком, то криптовалюты (в том числе и IOTA), работа которых базируется на DAG, имеют следующие особенности:

- Нет процесса создания блоков;

- Не имеют блокчейна;

- Нет процесса майнинга (все монеты IOTA были эмитированы в genesis-блоке, общее количество составляет $2,779 \times 10^{15}$ IOTA).

На данный момент всего три криптовалюты используют DAG в качестве базы — IOTA, DagCoin, ByteBall.

IOTA имеет совершенно другую архитектуру по сравнению с обычными криптовалютами, работающими на блокчейне. В частности,

консенсус здесь достигается не посредством майнинга, а совершенно другим способом. В системе ИОТА каждый пользователь должен подтвердить две предыдущие транзакции, прежде чем он сможет отправить платеж. В обычных криптовалютах, основанных на блокчейне, данную функцию выполняют майнеры, работающие по алгоритму Proof-of-Work. Однако система работы ИОТА примечательна тем, что каждый пользователь вносит свой вклад в верификацию транзакций и систему безопасности. Транзакции проверяются несколько раз разными пользователями, что позволяет системе определять каждую из транзакций как действительную.

Процесс верификации транзакций происходит по следующей схеме:

Узел выбирает две транзакции для проверки (обычно это две последние транзакции);

После того как транзакции пройдут верификацию на отсутствие конфликтов, они подтверждаются узлом;

Узел должен решить криптографическую задачу, похожую на головоломки в блокчейне биткойна (нужно найти попсо для получения правильного хеша).

Это позволяет убрать комиссии, так как вместо комиссии пользователи «платят» мощностью своих устройств, подтверждая другие транзакции. Кроме того, система гарантирует наличие узлов, отвечающих за проверку транзакций, так как для отправки каждого платежа пользователь должен подтвердить две транзакции. Таким образом узлов всегда в два раза больше, чем транзакций.

Преимущества и недостатки

Без сомнения, ИОТА — это одна из самых перспективных криптовалют, так как она может решить одну из основных проблем цифровых валют — сложность применения для ежедневных платежей. Также у нее ряд других плюсов:

Большие перспективы рынка Интернета Вещей — IOTA была разработана непосредственно для использования в Internet of Things, который имеет огромный потенциал в будущем.

Уникальный проект — использование технологии Tangle отличает эту криптовалюту от всех других, что привлекает внимание не только инвесторов, но и крупных компаний.

Нет привязки к интернету — система устроена таким образом, что для проведения платежей нет необходимости соединения с интернетом. Для управления некоторыми сетями «умных» устройств будет достаточно подключаться к интернету раз в месяц или даже год.

Сильная и ответственная команда — в отличие от многих других проектов команда IOTA ведет активную постоянную работу над улучшением криптовалюты.

Если сравнить IOTA с другими криптовалютами, например, с Bitcoin, то она имеет как минимум пять весомых преимуществ:

Неограниченная масштабируемость сети — поскольку благодаря особенностям алгоритма Tangle за отправку каждой транзакции пользователь должен проверить две транзакции, что позволяет создать условия, при которых узлов, обеспечивающих верификацию, больше, чем транзакций. Это означает, что сеть IOTA масштабируется пропорционально количеству транзакций. Для проверки этих двух транзакций устройство должно выполнить ту же работу, что выполняют майнеры в условиях алгоритма «Proof-of-Work». Также в системе Tangle отсутствует такое понятие как сложность, поэтому проверку транзакций можно выполнять, даже используя обычный смартфон или ноутбук.

Быстрые транзакции — время подтверждения и доставки транзакций обратно пропорциональны количеству транзакций, т.е. чем больше транзакций будет отправлено, тем быстрее они будут подтверждены. Когда IOTA достигнет массового внедрения в экосистему Интернета Вещей, транзакции станут моментальными.

Отсутствие комиссий — поскольку технология Tangle не предполагает работу майнеров, нет необходимости в эмиссии новых монет и уплате комиссионных сборов. Поэтому транзакции IOTA являются абсолютно бесплатными.

Фиксированное количество монет и отсутствие эмиссии — все монеты IOTA изначально созданы в genesis-блоке — их количество никогда не увеличится и не уменьшится. Общее количество составляет $2,779 \times 10^{15}$ IOTA.

Устойчивость к квантовым атакам — создатели проекта утверждают, что IOTA устойчива к атаке 51%, которая может проводиться при помощи квантовых компьютеров, благодаря тому, что криптовалюта создана на системе сбалансированных тернарных вычислений вместо привычных бинарных, с которыми работают современные компьютеры.

Однако IOTA, как и любая другая криптовалюта сейчас не является идеальной и имеет ряд недостатков:

Наличие «координатора» — в систему IOTA встроен так называемый «координатор», предназначенный для борьбы с кибератаками, контроля подтверждения транзакций и обеспечения стабильной работы сети. Это противоречит всем законам децентрализованных криптовалют, так как код координатора является закрытым, и он централизован. Однако предполагается, что «координатор» будет удален из системы, как только сеть станет достаточно большой для обеспечения безопасности и надежности.

Высокая вероятность проведения атаки 51% одним из узлов — отсутствие майнеров и традиционного алгоритма Proof-of-Work кроме описанных выше плюсов также несет в себе и очень серьезный минус. Теоретически один из узлов может создать транзакцию, которая соберет в себе 51% всей мощности сети, что позволит ему провести атаку double-spending.

Сложности, связанные с использованием тернарных вычислений вместо бинарных — так как система IOTA работает на современном

оборудовании (имеющем в основе бинарные вычисления), а код криптовалюты написан на тернарных вычислениях, то все тернарные значения должны быть переведены в бинарный код, что требует дополнительных мощностей для хранения данных. Кроме того, применение тернарных вычислений стало причиной возникновения багов в Curl (хеш-функция IOTA).

Не разработана технология смарт-контрактов — в отличие от большинства криптовалют (особенно тех, которые созданы на блокчейне Ethereum) на данный момент IOTA не имеет разработанной технологии, которая бы позволяла заключать смарт-контракты. Однако разработчики уверяют, что до конца 2018 года эта возможность появится.

Несовершенство оригинального кошелька — для IOTA разработан оригинальный локальный кошелек, который можно скачать для ПК (Windows, Mac OS, Linux), однако для мобильных ОС (Android, iOS) на данный момент доступны лишь бета-версии. Основная масса пользователей хранит эту криптовалюту на популярных биржах, которые имеют встроенные кошельки

Единицы измерения

IOTA с греческого переводится как «йота» — буква греческого алфавита. В экосистеме криптовалюты IOTA (MIOTA) самая маленькая величина — это IOTA. Большие величины создаются путем добавления метрических приставок (Kilo, Mega, Giga и т.д.) к слову «IOTA». Курс этой криптовалюты, который отображается на биржах и обменниках, базируется на стоимости MegaIota или Miota (Mi), т.е. миллион монет IOTA. Ниже таблица со значениями IOTA в порядке увеличения:

Такая градация позволяет IOTA использоваться для проведения микроплатежей.

1.1.2 Введение в предметную область IoT

IoT — свойство интеллектуальных систем выполнять творческие функции, которые традиционно считаются прерогативой человека.

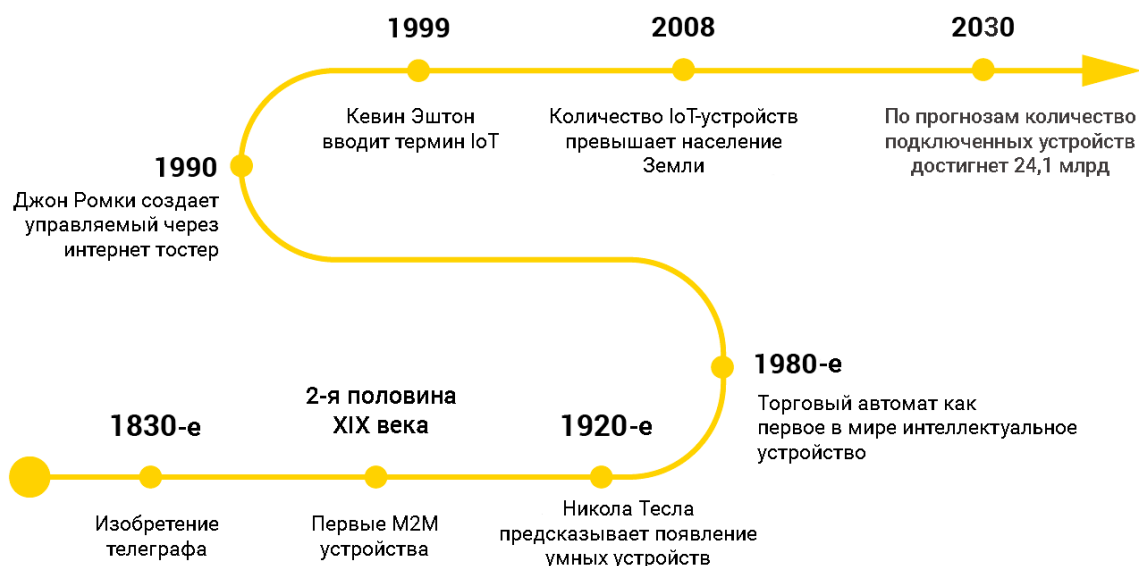
Интернет вещей (Internet of Things, IoT) — это множество физических объектов, подключенных к интернету и обменивающихся данными. Концепция IoT может существенно улучшить многие сферы нашей жизни и помочь нам в создании более удобного, умного и безопасного мира. Примеры Интернета вещей варьируются от носимых вещей, таких как умные часы, до умного дома, который умеет, например, контролировать и автоматически менять степень освещения и отопления. Также ярким примером служит так называемая концепция умного предприятия (Smart Factory), которое контролирует промышленное оборудование и ищет проблемные места, а затем перестраивается так, чтобы не допустить поломок. Интернет вещей занимает важное место в процессе цифровой трансформации в компаниях. Прогнозируется, что к 2030 году количество подключенных к сети устройств вырастет примерно до 24 млрд с годовой выручкой до 1,5 трлн долларов.

История происхождения

Термин «Интернет вещей» был впервые употреблен в 1999 году Кевином Эштоном, предпринимателем и соучредителем центра Auto-ID Labs (независимая сеть лабораторий и исследовательская группа в области сетевой радиочастотной идентификации и новых сенсорных технологий) при Массачусетском технологическом институте. Эштон состоял в команде, которая сумела изобрести способ подключения объектов к интернету с помощью технологии RFID. RFID-метка — это метка идентификации, позволяющая идентифицировать объекты посредством радиосигналов; на нее можно нанести определенную информацию, а позднее считать устройством.

В 2012 году произошли значительные изменения датчиков, что привело к ускорению рыночной готовности IoT, и для многих компаний это означало, что цифровая трансформация набирает обороты. Технологическое совершенствование сделало возможным появление МЭМС —

микроэлектромеханических систем (миниатюрное устройство, изготовленное методом микрообработки как из механических, так и из электрических компонентов). Благодаря этому датчики уменьшились настолько, что их стало возможно фиксировать, например, на одежде.



Из чего состоит IoT? Архитектура

Конечные устройства

Устройства — это объекты, которые фактически образуют «вещи» (Things) в Интернете вещей. Они играют роль интерфейса между реальным и цифровым мирами и принимают разные размеры, формы и уровни технологической сложности в зависимости от задачи, которую они выполняют в рамках конкретного развертывания IoT. Будь то микрофоны размером с булавочную головку или внушительного размера машины, практически любой материальный объект можно превратить в подключенное устройство путем добавления необходимых элементов (датчиков или приводов вместе с соответствующим программным обеспечением).

Программное обеспечение

Это то, благодаря чему подключенные устройства можно назвать «умными». Программное обеспечение отвечает за связь с облаком, сбор данных, интеграцию устройств и за анализ данных в реальном времени.

Также оно предоставляет возможности для визуализации данных и взаимодействия с системой IoT.

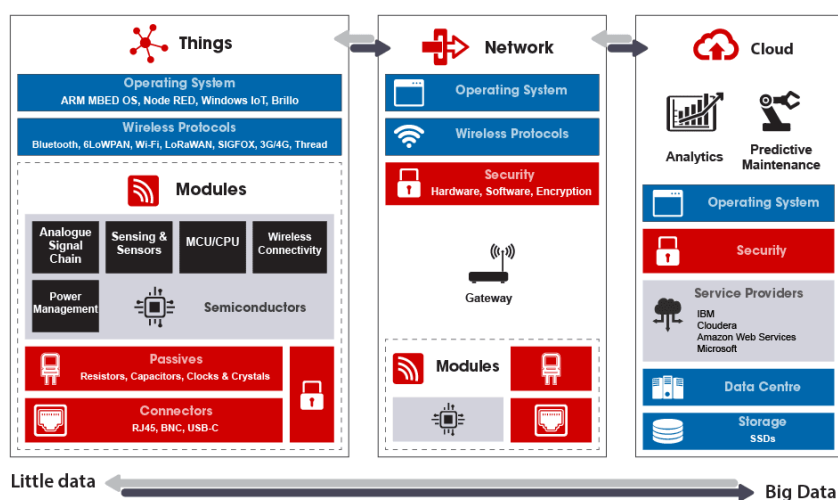
Коммуникации

Уровень коммуникации включает в себя как решения для физического подключения (сотовая и спутниковая связь, LAN), так и специальные протоколы, используемые в различных средах IoT (ZigBee, Thread, Z-Wave, MQTT, LwM2M). Выбор подходящего коммуникационного решения — одна из жизненно важных частей при построении каждой IoT-системы. Выбранная технология будет определять не только способы отправки и получения данных из облака, но способы связи со сторонними устройствами.

Платформа

Устройства способны «ощущать», что происходит вокруг и сообщать об этом пользователю через определенный канал связи. IoT-платформа — это место, где все эти данные собираются, анализируются и передаются пользователю в удобной форме. Платформы могут быть установлены локально или в облаке. Выбор платформы зависит от требований конкретного проекта IoT и многих факторов: архитектура и стек технологий, надежность, параметры настройки, используемые протоколы, аппаратная независимость, безопасность, эффективность, стоимость.

Ниже можно рассмотреть подробнее составляющие трех уровней IoT: конечных устройств (вещей), сети, облака.



Безопасность

Одновременно с тем фактом, что IoT-системы несут в себе значительную бизнес-ценность, интеллектуальные объекты также становятся уязвимы для киберпреступности, в результате которой может происходить утечка данных, в том числе и конфиденциальной информации. Несмотря на то, что поле работы с вопросом безопасности остается огромным, сейчас существуют решения, позволяющие осуществлять развертывание IoT более надежно. Например, для решения проблемы устаревания программного обеспечения устройств, есть возможности эффективных стратегий автоматического обновления.

Благодаря SOTA (Software Over the Air) «обновление по воздуху» и FOTA (Firmware Over the Air) — «прошивка по воздуху», программное обеспечение подключенных устройств и настройки можно обновлять с помощью беспроводной связи.

1.1.3 Введение в предметную область Blockchain

Блокчейн — это распределенная база данных, которая содержит информацию обо всех транзакциях, проведенных участниками системы. Информация хранится в виде цепочки блоков. В каждом из них записано определенное число транзакций.

Что такое распределенная база данных? Слово «распределенная» означает: нет никакой централизованной организации, которая бы проверяла этот процесс.

В дословном переводе блокчейн — это непрерывная цепочка блоков. В ней содержатся все записи о сделках — хоть с луковицами тюльпанов в ботаническом саду. В отличие от обычных баз данных, изменить или удалить эти записи нельзя, можно только добавить новые.

Разработчики отвечают за жизненный цикл приложений блокчейн: они создают архитектуру системы и смарт-контракты, оптимизируют протоколы,

и решают множество других задач. Таким специалистам необходимо разбираться в структурах данных и алгоритмах, криптографии и компьютерных сетях. Часто они работают с конкретным типом блокчейна, вроде Ethereum или Bitcoin.

Блокчейн-разработчики условно делятся на два типа:

- Разработчик программного обеспечения блокчейн;
- Базовый разработчик блокчейн.

Аналогия с блокчейном — ожерелье. Каждая бусина — это «блок» или запись действия. Это ожерелье — или «chain» (цепь) — не может быть уничтожено или разрушено. Таким образом, блокчейн — это нерушимая цифровая запись действий. Надежность этой системы позволила использовать ее для повышения эффективности денежных операций и обмена информацией среди частных лиц, корпораций и даже госсектора.

Внедрение блокчейна увеличивает скорость обмена, уменьшает временные затраты, улучшает качество, надежность и доступность услуг. При этом увеличивается прозрачность и надежность, снижаются риски.

Главная сфера применения блокчейна — криптоиндустрия. Но помимо этого проекты на блокчейне используются в банковском секторе, сфере финансовых услуг, платежных сервисах, госсекторе (госуслуги, реестры недвижимости, нотариат, электронное голосование и др.), транспорте и логистике, IoT, здравоохранении, управлении интеллектуальной собственностью, энергетике, и т. д.

В криптоиндустрии блокчейн стал технологической основой для выпуска криптовалют, которые являются осовремененной версией хайковских «частных денег». При этом крупнейшие из них, такие как биткоин и эфир, имеют глобальный характер обращения.

На основе этой технологии происходит токенизация. Выпуск токенов — это особая форма секьюритизации активов на базе массового и глобального спроса инвесторов. Снижение издержек при этом гораздо

значительнее, по сравнению с процедурами традиционных финансовых рынков.

Сферы применения блокчейна: банковское дело

Блокчейн позволяет сделать все процессы в банковской индустрии безопаснее, надежнее и прозрачнее. Денежные переводы, расчеты при сделках с ценными бумагами, аккредитивы, KYC-комплаенс, рутинная работа бэк-офисов банков — все эти операции теперь проходят с внедрением этой технологии.

Блокчейн может сократить затраты банков до 50%. Об этом заявляли еще в 2017 г. аналитики Morgan Stanley. По их мнению, блокчейн мог бы оптимизировать инфраструктуру, радикально сократить затраты и обеспечить необходимое повышение RoE (доходности собственного капитала) банков. Многим банкам, несмотря на присущую им консервативность, опасно было бы недооценивать потенциал этой технологии. Повсеместное ее внедрение может привести к ликвидации некоторых участников мировой финансовой системы.

Сферы применения блокчейна: кибербезопасность

На каждом узле блокчейн-системы хранятся копии всей базы данных в целом, и они сверяются между собой. Это придает системе жизнеспособность даже в случае успешных хакерских атак на ее одиночные узлы. Несмотря на то, что приложения на блокчейне предлагают анонимность, технология может использоваться, чтобы прикреплять реальные идентификационные данные к криптографическим в базе данных.

Известен стартап, создавший удобный способ безопасного внесения данных, их отслеживания и передачи имущественных прав через блокчейн-платформу. Это помогает риэлторским фирмам: управление записями становится удобнее, сокращается время поиска, растет конфиденциальность и прозрачность. Другой кейс — система репутации интернет-юзеров, начисляющая денежное вознаграждение в криптовалюте эфир, исходя из показателей репутации.

Сферы применения блокчейна: удостоверения личности

Такие ID представляют собой перевод персональных данных о человеке на блокчейн, создание его цифрового профиля. Есть большой перечень госуслуг, доступ к которым может быть осуществлен посредством блокчейна, и есть статистика передовых блокчейн-стран (Эстония, некоторые эмираты ОАЭ). Использование блокчейна и блокчейн-ID активным гражданином может стать обычным паттерном поведения уже при жизни современного поколения.

И в России, и на Западе государства в последние десятилетия активно ищут пути оптимизации своей административной деятельности. Хорошим решением для этого было бы перевести в блокчейн все рутинные и архаичные процессы. Проблематику приватности и защиты информационной безопасности при этом никто с повестки дня не снимал — напротив, в эру блокчейна она станет приоритетом.

Как работают платежные средства на блокчейне?

Можно выделить как минимум два типа платежных средств. Прежде всего, это традиционные негосударственные криптовалюты, такие как эфир и биткоин, которые имеют многомиллиардную (если измерять ее в «традиционных» фиатных долларах США) капитализацию.

Второй тип — криптовалюты центральных банков — это зачастую попытки фиатных государственных эмитентов влить «новое вино в старые меха»: представить новую форму зарабатывания сеньоража, т. е. эмиссию госвалюты, как нечто сверхмодное и прогрессивное. Только единицы проектов по выпуску КВЦБ на сегодня предусматривают подлинно децентрализованный характер их эмиссии.

1.1.4 Примеры областей применения IoT

IoT применим в разных отраслях для различных целей: отслеживания потребительского поведения в режиме реального времени, улучшения

качества работы машин и систем, нахождение инновационных методов работы в рамках цифровой трансформации и многое другое.

Розничная торговля

Среди примеров приложений IoT в сфере розничной торговли можно встретить множество случаев использования интеллектуальных устройств для повышения качества обслуживания в магазинах. В частности, различные приложения IoT здесь означают, что возможности использования смартфонов (на основе технологии Beacon — миниатюрных маячков) облегчают общение между розничными продавцами и покупателями, а наиболее востребованные товары и услуги появляются перед глазами клиентов в нужном месте. Кроме того, интеллектуальная розничная торговля открывает возможности для приложений IoT с точки зрения точной рекламы, улучшения цикла цепочки поставок и фактического анализа моделей спроса. Также приложения IoT уже включают приложения для платежей NFC и интеллектуальных покупок. И конечно, нельзя не упомянуть RFID-метки для маркировки товара, которые обеспечивают моментальный и точный сбор информации, что помогает непрерывно отслеживать перемещение товаров, упростить процесс инвентаризации и в целом сократить количество ошибок.

Производство

Благодаря IoT производство может получать общую картину о процессах производства и состоянии продукта на всех этапах — от поставки сырья до отгрузки готового продукта.

С помощью датчиков, установленных на заводском оборудовании и в складских помещениях, анализа больших данных и прогностического моделирования (predictive modeling) можно предотвратить множество ошибок, ведущих к простоям и убыткам, максимизировать производительность, уменьшить гарантийные расходы и в целом улучшить качество клиентского сервиса.

Здравоохранение

С помощью технологии IoMT (The Internet of Medical Things, Интернет медицинских вещей) в режиме реального времени происходит сбор потоков малых данных из медицинских сетевых и других носимых устройств, отслеживающих различные физиологические моменты, связанные со здоровьем пациентов — движения, динамика сна, сердечный ритм, аллергические реакции и прочее. Собранные данные помогают врачам в постановке точных диагнозов, построении плана лечения, повышают безопасность пациентов, упрощают уход за ними, дают возможность непрерывного мониторинга состояния тяжелобольных пациентов.

Применение Интернета вещей способствует созданию более персонализированного подхода к анализу состояния здоровья и более последовательных стратегий борьбы с болезнями.

Ключевые моменты в сфере здравоохранения, которые можно улучшить с помощью IoT.

Энергетика

Здесь с помощью IoT конструкция электрических сетей меняет правила потребления, автоматически собирая данные и обеспечивая мгновенный анализ циркуляции электроэнергии. В результате этого и клиенты, и поставщики лучше понимают, как оптимизировать использование ресурса.

1.1.5 Заключение

Революция в области Интернета вещей представляется важной для развития бизнеса, и это может относиться к любому типу предприятия. Будь то выращивание устриц или создание системы управления движением, самое ценное в технологической концепции IoT — это то, что он открыт к новым вызовам, и в нем достаточно возможностей для реализации практически любой.

1.2 Анализ существующих решений

1.2.1 Платформы для бронирования отелей в Блокчейн-сети

Travala.com (TravelbyBit Booking.com)

TravelbyBit - онлайн-платформа и агентство по бронированию отелей, созданное для поколения блокчейн на базе ведущей в мире криптовалютной биржи Binance. Команда состоит из путешественников и криптоэнтузиастов, обеспечивающих уникальный опыт путешествий. Платформа и ее агенты позволяют пользователям свободно путешествовать по миру, используя цифровые валюты. Забронируйте билеты на самолет и номер в отеле сегодня и узнайте, куда вас приведет Биткойн. Приложение Deep Network Designer позволяет проектировать, анализировать и обучать сети в графическом виде.

TravelbyBit - в данный момент является не рабочим и при попытке поиска отеля просто уходит в пустую страницу. При попытке авторизации переносит на Travala.com, на котором уже можно забронировать отель, а так же купить авиабилеты, так что рассматриваться будет Travala.com.

Основанная в 2017 году, Travala.com превратилась из небольшого стартапа в популярную платформу бронирования путешествий на основе блокчейна, которой тысячи клиентов по всему миру доверяют как предпочитаемому онлайн-туристическому агентству.

Travala.com предлагает путешественникам большое количество вариантов невероятных мест для проживания, включая все, от отелей, апартаментов, вилл и хостелов до роскошных 5-звездочных курортов. Платформа Travala.com в настоящее время предлагает более 2 000 000 объектов размещения, охватывающих 90 124 направления в 230 странах и территориях, по ценам до 40% ниже, чем у обычных платформ бронирования путешествий.

Ценностное предложение Travala.com поддерживается собственной криптовалютой AVA, которая может использоваться для платежей, получения и погашения вознаграждений за лояльность, возврата средств и в

качестве средства сбережения, а также в нескольких других случаях использования.

Пример бронирования в Travalala.com

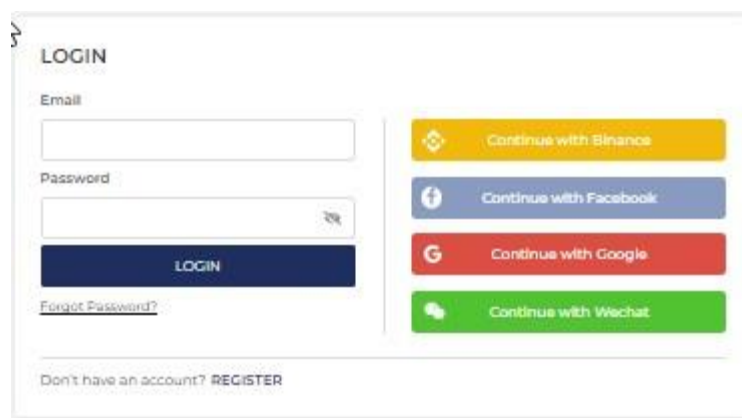


Рисунок I.1 — авторизация

На рис.1.1 представлен пример авторизации, список доступных способов, один из которых крупная биржа Binance с которой сотрудничает Travalala.com. Для оплаты в крипто валюте через эту площадку самым удобным способом будет оплата через Binance, из-за минимальной комиссии (оплата происходит в BUSD, крипто валюты прикрепленной к курсу доллара) и удобного интерфейса.

Оплата через биржу Binance происходит по qr-коду, который считывается через приложение Binance для IOS и Android пример оплаты представлен на рисунка 1.2 и 1.3

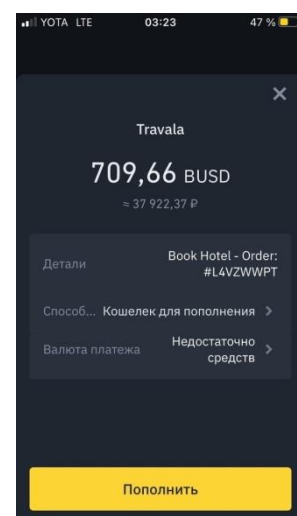
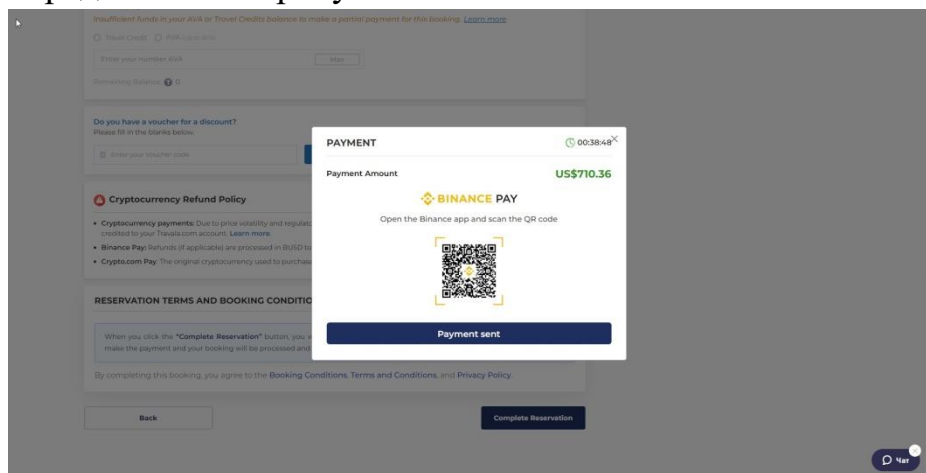


Рисунок 1.3,1.4 — qr-code для оплаты через биржу Binance - оплата через биржу Binance

Функционал сайта состоит из: авторизации (рис. 1.1), выбора отеля, авиабилетов или тура (рис. 1.4,1.5), просмотр информации о выбранном отеле или туре (рис. 1.5), бронирование отеля, покупка авиабилета или тура (рис. 1.6) оплата (рис 1.3,1.4), просмотр данных (рис. 1.7,1.8), техподдержка.

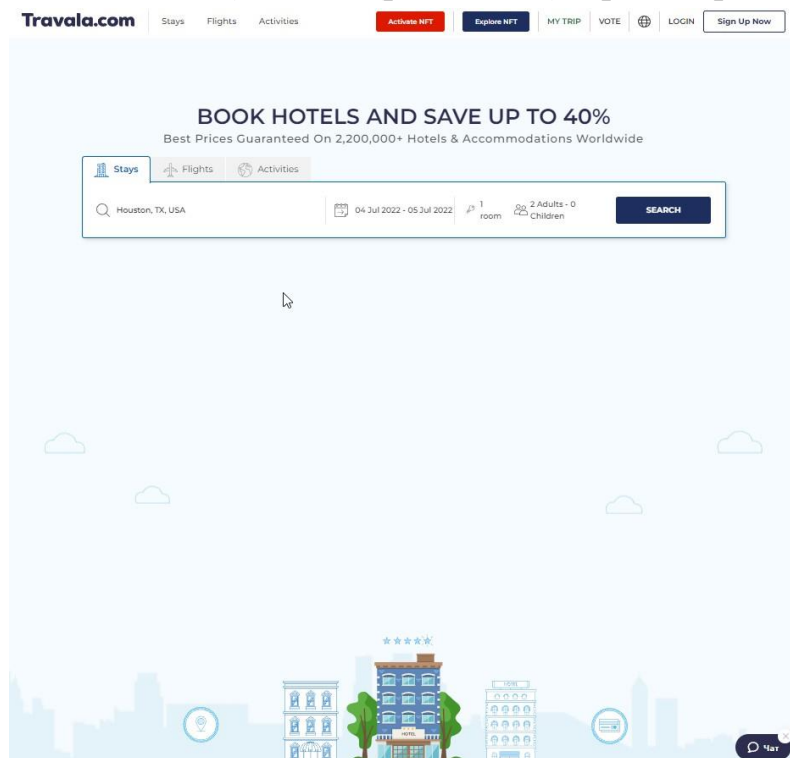


Рисунок I.4 - поисковая строка.

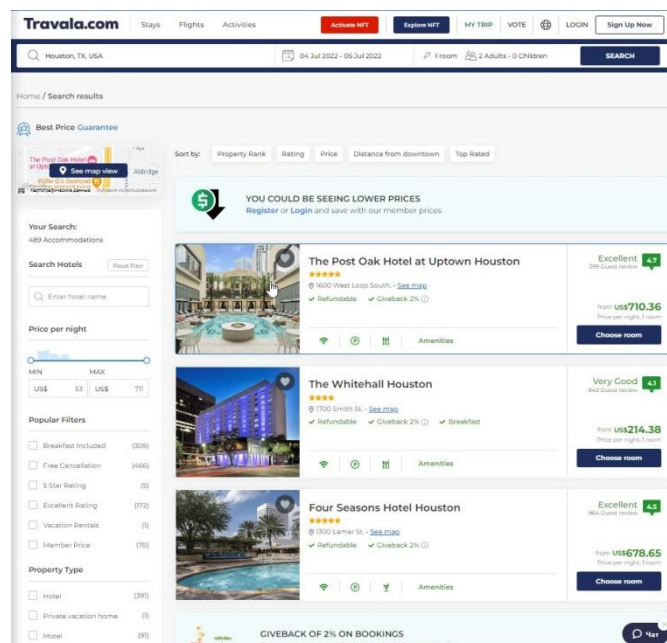


Рисунок I.5 – список отелей.

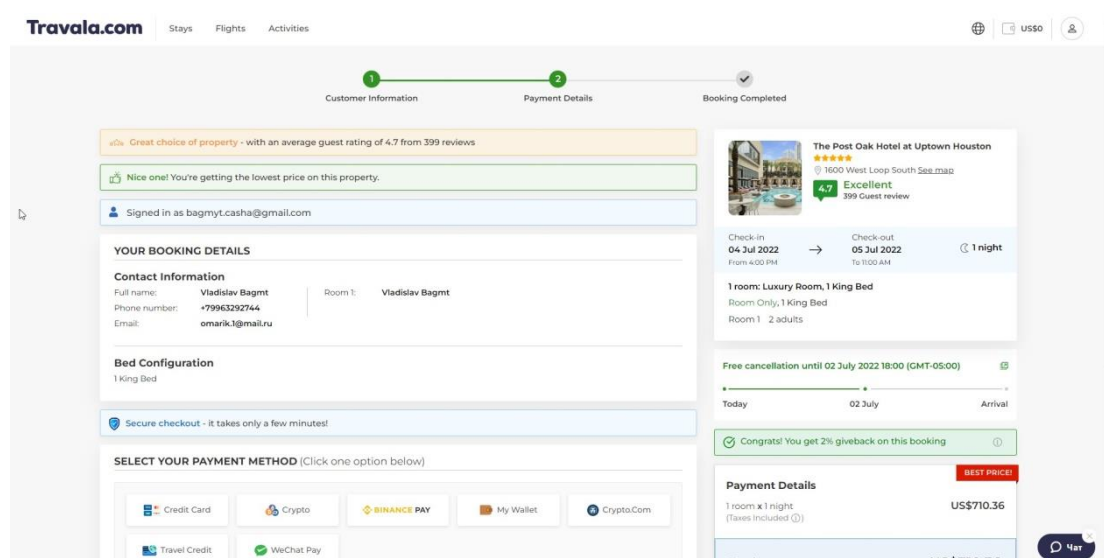


Рисунок I.6 – бронирование отеля.

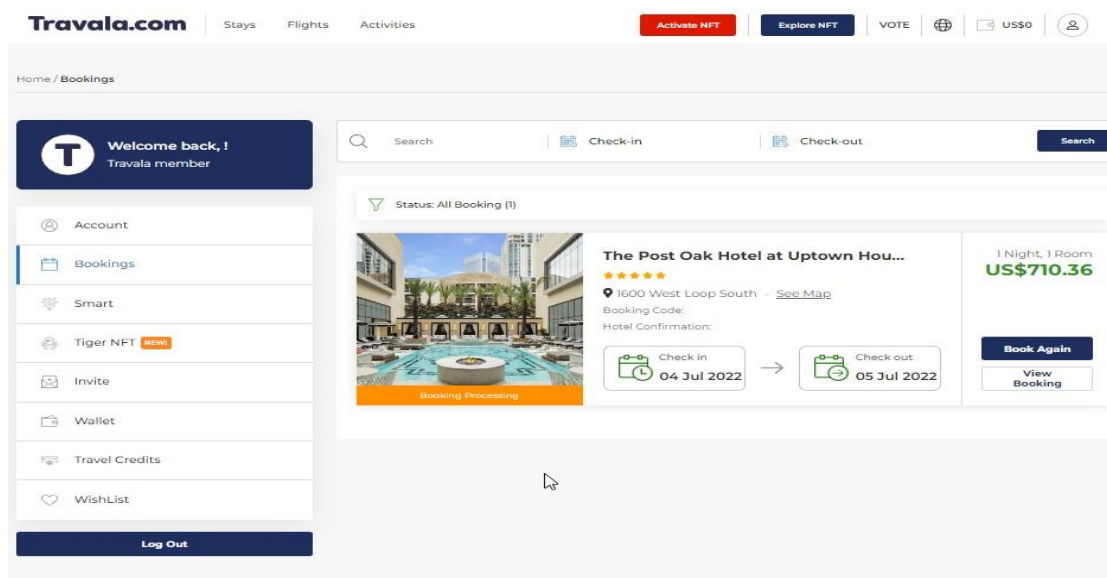


Рисунок I.7 – просмотр данных аккаунта.

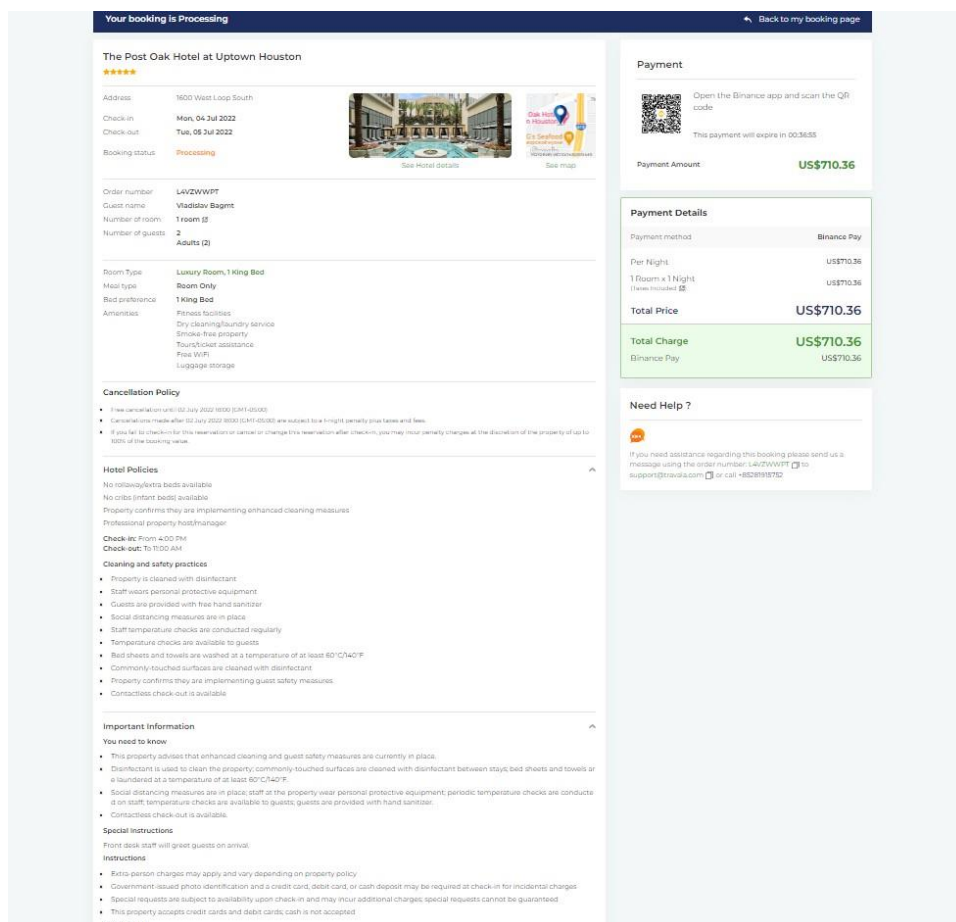


Рисунок I.7 – просмотр информации о брони.

1.2.2 Электронный замок 360lock

Основные характеристики:

Bluetooth с низким энергопотреблением 4.0

Модуль NFC (записывается до 20 операций)

приложение iOS и Android

сопутствующее приложение для Apple Watch и умных часов с WearOS (Android Wear)

совместное использование доступа и расписание (период, время, дни недели)

уведомлять о неожиданных открытиях

сертификация операции на базе Ethereum Blockchain

IP65

длительное управление батареей

дужка из закаленного железа и корпус из алюминиевого сплава

1.2.3 Зарядная станция для электромобиля

В данный момент аналоги в разработке. (статья Cointelegraph)

В течение следующих трех лет 50 000 станций зарядки электромобилей по всей Европе начнут принимать криптовалютные платежи. Две платежные компании объединились для развертывания криптоплатежей на 50 000 станций зарядки электромобилей (EV) в Европе.

Партнерство заключается между ирландской компанией HIPS Payment Group Ltd, специализирующейся на решениях для электронной коммерции и мобильных платежей, и шведской фирмой Vourity, которая специализируется на автоматических платежных средствах, таких как зарядные станции для электромобилей.

Интеграция криптоплатежей с зарядными станциями произойдет в течение следующих трех лет, начиная с ноября 2021 года.

Фирмы еще не раскрыли, какие криптовалюты будут поддерживаться, но Vourity дал довольно сильный намек на то, что Ethereum, вероятно, будет одним из первых после того, как он выпустил изображение платежного

терминала с логотипом ETH на нем. В следующем году Ethereum перейдет к гораздо более энергоэффективному механизму консенсуса Proof-of-Stake, который может смягчить негативную реакцию среди экологически сознательных водителей электромобилей.

Криптовалютные платежи будут интегрированы с зарядными станциями Vourity для электромобилей путем подключения к блокчейну через собственный токен протокола Hips Merchant Protocol Merchant Token.

Еще в мае платежная группа HIPS запустила торговый протокол HIPS, шлюз торгового протокола HIPS и управляющий торговый токен.

Протокол был построен на Ethereum и Solana в мае 2021 года, и в будущем планируется расширить поддержку Cardano.

1.3 Выводы по разделу

Исходя из количества найденных мною аналогов можно сказать, что сфера в которой ведется разработка проекта является новой, и практически пустой, есть лишь один зарубежный аналог, с которым сотрудничают остальные ПО для бронирования отелей, российских аналогов же, вообще нет.

II ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ К РАЗРАБАТЫВАЕМОМУ ПРИЛОЖЕНИЮ

2.1 Требования к функциям

2.1.1 Функциональная структура приложения

Приложение имеет функции для бронирования отеля на определенный день недели, и расчета оплаты в крипто валюте, в будущем с возможностью привязки к ПО ключ-замков, что позволит сохранять анонимность клиентов.

Сайт хранит только информацию о ID кошелька клиента и статистика посещения и ничего больше о данных пользователя.

Из функций для пользователя имеются: оплата брони отеля для себя, оплата брони отеля для другого человека, выбор дня недели и качества отеля, выбор способа оплаты и просмотр статистики брони.

Из функций для администрации имеется: тестирование брони отеля, просмотр статистики отдельного пользователя, просмотр статистики общей, чистка данных отдельного пользователя, чистка данных общих, проверка баланса кошелька компании и вывод средств с кошелька.

2.1.2 Функции сайта ПО

Основные функции сайта ПО приведены в таблице 2.1.

Таблица I.1 — Основные функции сайта ПО

Функция	Описание
Ввод адреса кошелька для бронирования	1. Пользователю предлагается ввести адрес кошелька на который будет оформляться бронь или оставить поле пустым для использования подключенного адреса. 2. Пользователь может бронировать отель, как на свой адрес кошелька, так и на сторонний при оплате со своего кошелька.
Адрес вашего кошелька	1. Поле, которое автоматически заполняется после подключения кошелька MetaMask, и считывается если остальные поля остаются пустыми. 2. Если кошелек не подключен, предлагается его подключение или же заполнение строки “Введите адрес кошелька для бронирования”.
Введите адрес кошелька для получения данных	1. Поле для поиска в базе данных ID кошелька пользователя 2. Если оставить поле пустым применяется ID пользователя прикрепленного в поле “Адрес вашего кошелька”
Название сайта	1. Заголовок с названием сайта
Информационное табло	1. Информационное табло внизу сайта
Подключение к кошельку MetaMask	1. Функция для подключения кошелька MetaMask 2. MetaMask запоминает сайт к которому подключается

2.1.3 Функции кнопок сайта

Основные функции проекта приведены в таблице 2.2.

Таблица I.2 — Основные функции кнопок сайта

Функция	Описание
Выбор дня недели для бронирования отеля	1. Пользователю предлагается удобная панель выбора дня недели для бронирования отеля, начиная с понедельника по воскресенье. 2. Данные о выбранном дне недели пользователем, хранятся в базе.
Выбор отеля	1. Пользователю предлагается удобная панель выбора отеля по качеству и количеству звезд, от одной до пяти звезд. 2. Данные о выбранном отеле пользователем, хранятся в базе данных.
Кнопка “Забронировать”	1. Кнопка является активной, активируется при нажатии 2. Кнопка учитывает данные выбранные пользователем и активирует скрипт который переводит в MetaMask для оплаты
Просмотр данных пользователя	1. Активная кнопка, активируется нажатием. 2. Предназначена для просмотра статистики посещения одного конкретного пользователя 3. Берет данные из базы данных сайта

Функция	Описание
Просмотр общих данных	<ol style="list-style-type: none"> 1. Активная кнопка, активируется нажатием. 2. Предназначена для просмотра общей статистики посещения 3. Берет данные из базы данных сайта 4. Функция администратора
Очистить данные для пользователя	<ol style="list-style-type: none"> 1. Активная кнопка, активируется нажатием. 2. Предназначена для очистки статистики посещения одного конкретного пользователя 3. Берет данные из базы данных сайта 4. Функция администратора
Очистить общие данные	<ol style="list-style-type: none"> 1. Активная кнопка, активируется нажатием. 2. Предназначена для очистки общей статистики посещения 3. Берет данные из базы данных сайта 4. Функция администратора
Узнать баланс	<ol style="list-style-type: none"> 1. Активная кнопка, активируется нажатием. 2. Предназначена для просмотра баланса отеля (смарт-контракта) 3. Функция администратора
Вывод средств	<ol style="list-style-type: none"> 1. Активная кнопка, активируется нажатием. 2. Предназначена для вывода средств отеля (смарт-контракта) 3. Функция администратора

2.1.4 Функциональные возможности кода

Основные функции блоков кода приведены в таблице 2.3.

Таблица I.3 — Основные функции блоков кода

Функция	Описание
setupchain	Скрипт в разрешении .sh запускающий блок - чейн – сеть IOTA
startnet	Скрипт устанавливающий Wasp библиотеки и запускающий Wasp-node
wasp	Wasp node – функция позволяющая писать контракт в IOTA на Solidity
Node_modules	Модули IOTA основанные на node.js
contract_new	Контракт написанный на solidity для развертывание блок - чейн сети на solidity с основными функциями сайта ПО
style	Файл с кодом верстки сайта ПО
bs-config	Файл со скриптом запуска локального порта 3000
package	Основной файл запуска сайта ПО
start	Файл запуска сайта ПО npm start

2.2 Требования к эргономике и технической эстетике

2.2.1 Общие требования к внешнему виду

В данный момент ПО является прототипом и единственные требования к внешнему виду является удобство пользования ПО.

Графические элементы без текстового сопровождения должны иметь всплывающие подсказки.

Пользователь должен получать информацию, как об успешном завершении операций, так и о возникновении сбоев в ходе их выполнения или невозможности выполнения.

При выполнении длительных операций, требующих значительного времени для выполнения, пользователь, по возможности, должен получать информацию о текущем ходе выполнения операции.

2.2.2 Требования по обработке исключительных ситуаций

Сообщения об ошибках или аварийных ситуациях должны быть на русском или английском языках, точно отражающими проблему.

Приложение должно обеспечивать корректную обработку аварийных ситуаций, вызванных неверными действиями пользователей, неверным форматом или недопустимыми значениями входных данных. После вывода пользователю соответствующего сообщения, система должна возвращаться в рабочее состояние, предшествовавшее неверной (недопустимой) команде или некорректному вводу данных.

2.2.3 Требования по вводу и контролю данных

Для данных, имеющих установленные форматы, должен осуществляться логический контроль ввода данных с выдачей сообщений об ошибках ввода.

2.2.4 Требования по организации экранных форм

Наименование полей в диалоговых окнах и формах системы должны приводиться с минимумом сокращений и аббревиатур. Допускается использование общепринятых сокращений при невозможности отображения полнотекстовых наименований.

2.2.5 Требования к техническому обеспечению

Рекомендуемые требования к прикладному программному обеспечению:

Рекомендуемые требования к системному программному обеспечению:

- Windows 8, 8.1, 10 64-разрядной версии / linux Ubuntu/Mac OS
- MetaMask

Рекомендуемые требования к аппаратному обеспечению:

- Подключение к интернету .
- Оперативная память: 4 Гбайт.
- Число ядер процессоров: 2.

2.3 Выводы по разделу

Во втором разделе определена функциональная структура приложения; подробно описаны требования к функциям, к эргономике и технической эстетике, а также требования к техническому обеспечению.

III ТЕХНИЧЕСКАЯ РЕАЛИЗАЦИЯ

3.1 Описание необходимых библиотек и разработанных модулей

Выбранные библиотеки для разработки приложения на IOTA:

- WASP node.
- Goshimmer
- Solidity
- Java-script

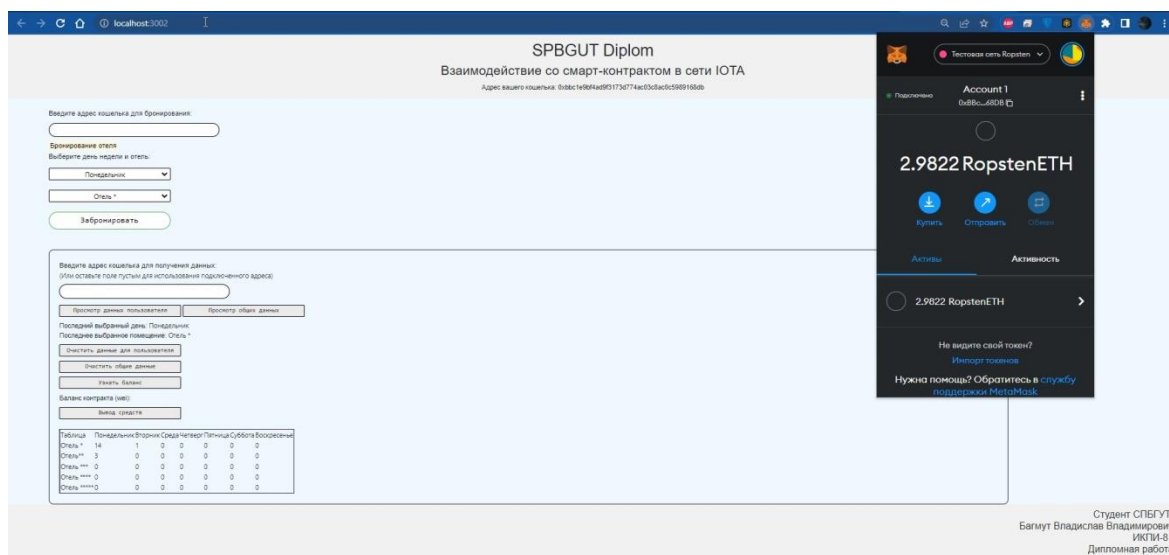


Рисунок 0.1 — Основные графические элементы приложения

3.2 Результаты разработки

В результате разработки в соответствии с требованиями было разработано приложение сайт ПО для отельного бизнеса с оплатой в криптовалюте со следующими характеристиками:

- Сайт состоит из:

Названия сайта:



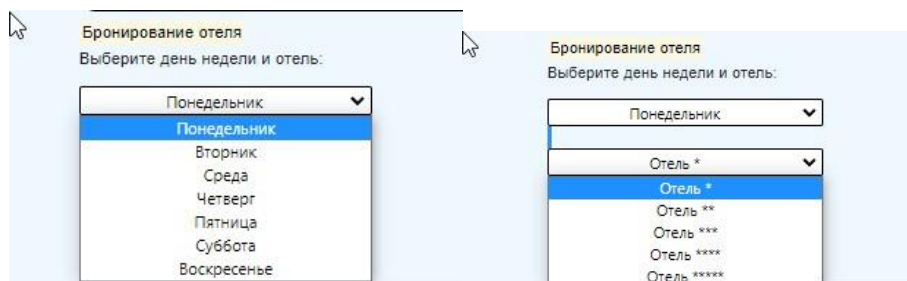
Строки адреса подключенного кошелька:



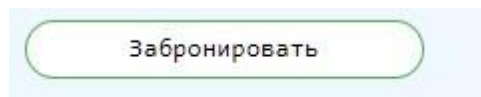
Строки для ввода кошелька:



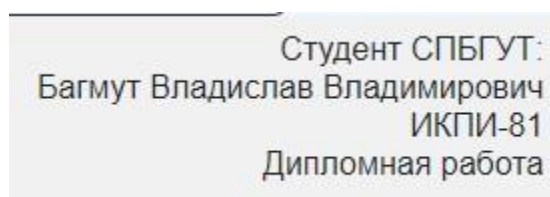
Двух всплывающих окон для выбора дня недели и отеля.



Активной кнопки для бронирования



Информационного табло



Таблицы администратора с активными кнопка

Введите адрес кошелька для получения данных:
(Или оставьте поле пустым для использования подключенного адреса)

Баланс контракта (wei):

Таблица	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Отель *	14	1	0	0	0	0	0
Отель **	3	0	0	0	0	0	0
Отель ***	0	0	0	0	0	0	0
Отель ****	0	0	0	0	0	0	0
Отель *****	0	0	0	0	0	0	0

В таблицу входят такие функции как:

Строки для ввода кошелька просматриваемого пользователя

Просмотр данных пользователя

Просмотр общих данных (всех пользователей)

Очистить данные пользователя

Очистить данные всех пользователей

Узнать баланс предприятия (смарт-контракта)

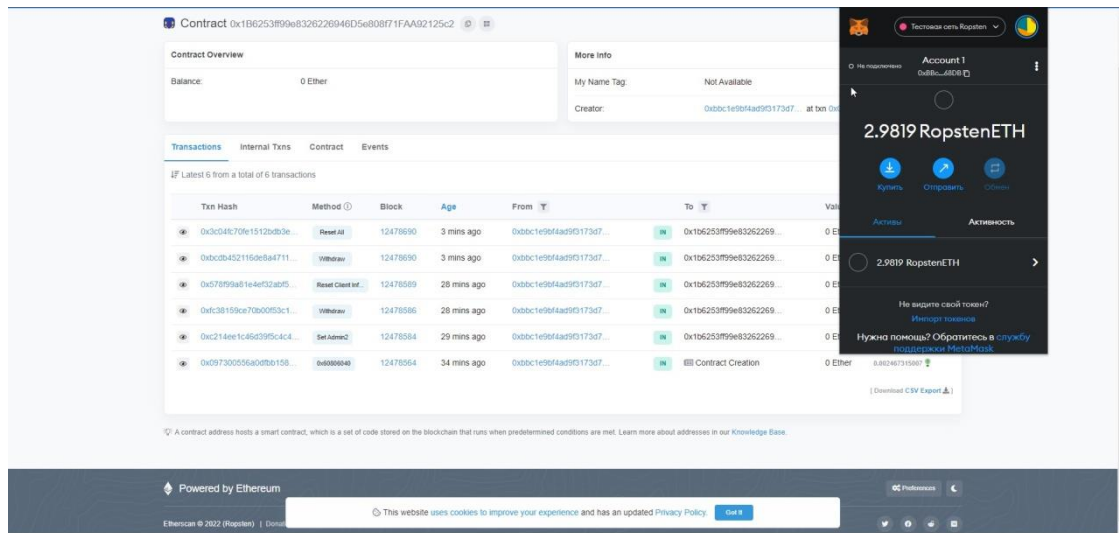
А так же визуальная таблица данных пользователей.

Из данных пользователей сайт хранит только хэш кошельков, по ним же и определяется, кто и на какое время и какой отель забронировал

[illegible]

48

менять кошельки на которые будут поступать средства.



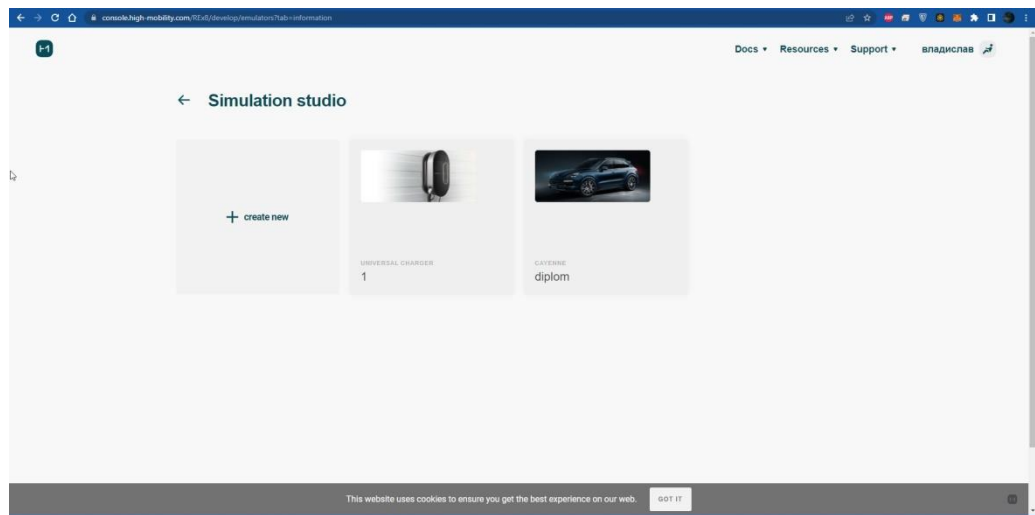
Вторая часть ПО для отелей в виде ключ карт еще в разработке, и нет возможности показать работу ПО, т.к для ключ-замков нет визуализации, но в качестве визуализации был выбран проект зарядной станции для электромобилей на базе блокчейна.

Функции такой зарядной станции состоят в том, что когда транзакция оплачивается и платеж приходит на смарт контракт активируется зарядка на то время которое было оплачено по смарт контракту.

Этот функционал схож с ключ-замками для отелей и может быть использован для визуализации ПО.

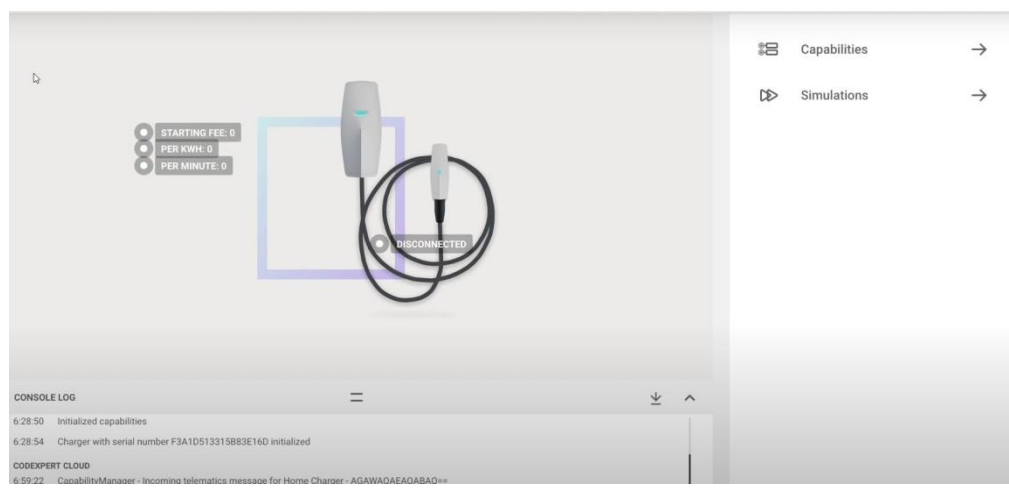
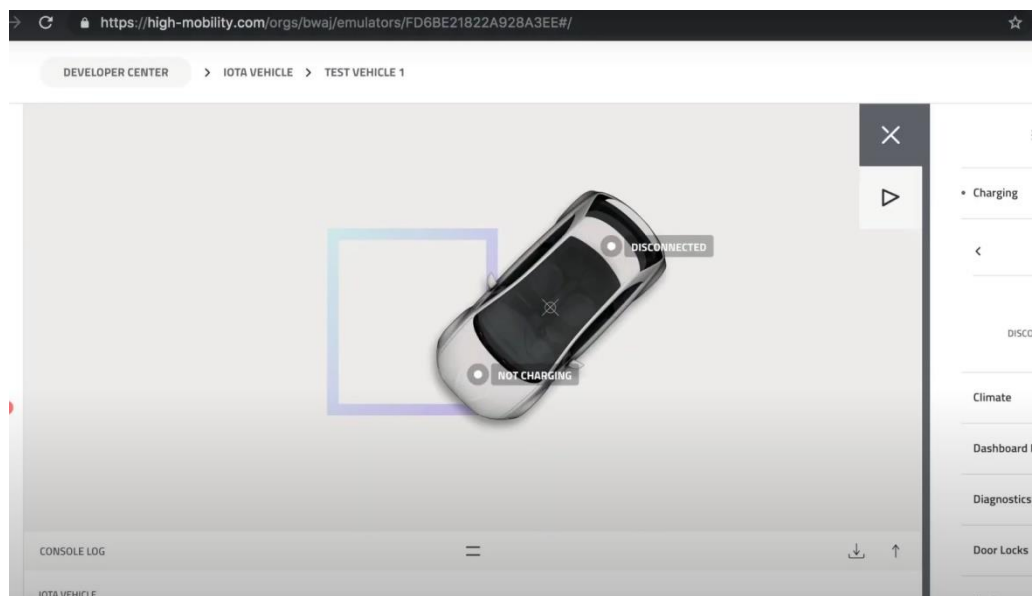
Идея состоит в том чтобы привязать ключ замок к оплате брони на сайте, после чего человек перейдя в отель сможет сразу пройти к своему номеру и открыть его с помощью хэша своего кошелька. И так как общая сумма возможных таких хэшей велика, для каждого замка будет свой идентификационный номер смарт контракта, в этом состоит задумка IOTA.

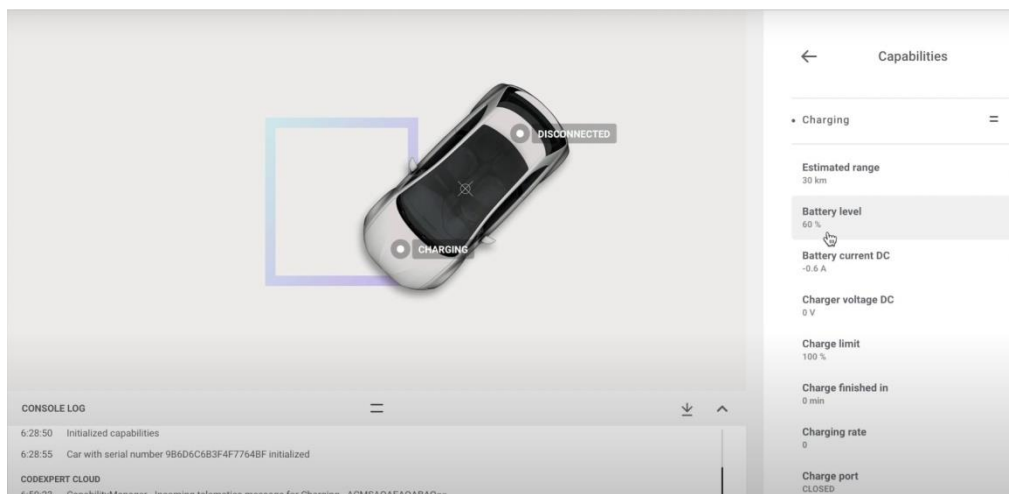
Для визуализации был взят API сайта <https://high-mobility.com>



В качестве среды разработки IOTA

Промежуточные результаты работы





Работа состояла в том, чтобы подключить зарядку к автомобилю, чтобы у автомобиля началась зарядка и показывался уровень заряда батареи, так же в процентах, сколько оплачено заряда

3.3 Выводы по разделу

В третьем разделе приведено описание необходимых библиотек и разработанных модулей, кратко представлены результаты разработки.

IV ДЕМОНСТРАЦИЯ РАБОТЫ ПРИЛОЖЕНИЯ

4.1 Интеграционное тестирование

Названия сайта:

SPBGUT Diplom
Взаимодействие со смарт-контрактом в сети IOTA

Строки адреса подключенного кошелька:

Введите адрес кошелька для бронирования:
0x6fe30E8147C97Ff4303F513F607403425C0C2e45

Строки для ввода кошелька:

Адрес вашего кошелька: 0xbbc1e9bf4ad9f3173d774ac03c8ac0c5989168db

Двух всплывающих окон для выбора дня недели и отеля.

Бронирование отеля
Выберите день недели и отель:

Понедельник
Понедельник
Вторник
Среда
Четверг
Пятница
Суббота
Воскресенье

Понедельник
Отель *
Отель **
Отель ***
Отель ****
Отель *****

Активной кнопки для бронирования

Забронировать

Информационного табло

Студент СПбГУТ:
Багмут Владислав Владимирович
ИКПИ-81
Дипломная работа

Таблицы администратора с активными кнопка

Введите адрес кошелька для получения данных:
(Или оставьте поле пустым для использования подключенного адреса)

Баланс контракта (wei):

Таблица	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Отель *	14	1	0	0	0	0	0
Отель **	3	0	0	0	0	0	0
Отель ***	0	0	0	0	0	0	0
Отель ****	0	0	0	0	0	0	0
Отель *****	0	0	0	0	0	0	0

В таблицу входят такие функции как:

Строки для ввода кошелька просматриваемого пользователя

Просмотр данных пользователя

Просмотр общих данных (всех пользователей)

Очистить данные пользователя

Очистить данные всех пользователей

Узнать баланс предприятия (смарт-контракта)

Вывести средства с баланса предприятия (смарт-контракта)

А так же визуальная таблица данных пользователей.

[illegible]

Transaction Details

Overview

State

1

[This is a Ropsten Testnet transaction only]

Transaction Hash:

0xe504ecd17ed4e3cd64302727fa1260a2a5555a706411c52169e3c00c22df59e

Status:

Success

Block:

12450067 134 Block Confirmations

Timestamp:

30 mins ago (Jun-28-2022 06:22:00 AM +UTC)

From:

0xb0bc1e9b44ad9f3173d774ac03cbac0c5989168db

To:

[Contract 0x4678cfa73d58a9209428716d24358c30d172401f Created]

Value:

0 Ether (\$0.00)

Transaction Fee:

0.001363445003873646 Ether (\$0.00)

Gas Price:

0.000000002500000007 Ether (2.5000000007 Gwei)

Click to see More

A transaction is a cryptographically signed instruction from an account that changes the state of the blockchain. Block explorers track the details of all transactions in the network. Learn more about transactions in our [Knowledge Base](#).

Etherscan
Explains Testnet Network

All Filters ▾ Search by Address / Transaction Hash / Block / Token / Etc.

[Home](#) | [Blockchain](#) → [Tokens](#) → [Misc](#) → [Pages](#)

Contract 0x467fbd47d3d5a920d42871602435c30d172401f

Contract Overview

Balance:	0.0006 Ether
----------	--------------

More Info [More ▾](#)

My Name Tag:	Not Available
Creator:	0xbcb1e95faad9f17367... at Sep 0xe0f6dc817ed4e3cd9...

Transactions Contract Events

⌵ Latest 7 from a total of 7 transactions 1

Txn Hash	Method	Block	Age	From	To	Value	Txn Fee
0x6f8fb087bf5a42693dec...	Add Reservation	12480140	14 mins ago	0xbcb1e95faad9f17367...	0x467fbd47d3d5a920d42...	0.0001 Ether	0.0001184035
0xdcfa2014c3c796f050c4...	Add Reservation	12480140	14 mins ago	0xbcb1e95faad9f17367...	0x467fbd47d3d5a920d42...	0.0001 Ether	0.0001184035
0x4e9834f032cecf8a60ca...	Add Reservation	12480138	14 mins ago	0xbcb1e95faad9f17367...	0x467fbd47d3d5a920d42...	0.0001 Ether	0.0001184035
0xdcc5ec3084ba4ea4039...	Add Reservation	12480132	16 mins ago	0xbcb1e95faad9f17367...	0x467fbd47d3d5a920d42...	0.0001 Ether	0.0001140350
0x5f5fab61ae5dc040ba...	Add Reservation	12480132	16 mins ago	0xbcb1e95faad9f17367...	0x467fbd47d3d5a920d42...	0.0001 Ether	0.0000228675
0xdba431b5000a45da0a...	Add Reservation	12480119	19 mins ago	0xbcb1e95faad9f17367...	0x467fbd47d3d5a920d42...	0.0001 Ether	0.0001141675
0xe0f6dc817ed4e3cd9...	GetReserveId	12480067	30 mins ago	0xbcb1e95faad9f17367...	EIP Contract Creation	0 Ether	0.00138240403

[\[Download CSV Export \]](#)

A contract address hosts a smart contract, which is a set of code stored on the blockchain that runs when predetermined conditions are met. Learn more about addresses in our Knowledge Base.

This website uses cookies to improve your experience and has an updated Privacy Policy [Go to](#)

Сверка баланса счета смарт – контракта с сайтом

SPBGUT Diplom
 Взаимодействие со смарт-контрактом в сети IOTA
Адрес вашего кошелька: Iotsf6t6h4d9D173d774e3D3a3d588198da

Введите адрес кошелька для бронирования:

Бронирование отелей

Выберите день недели и отель

Список

Отель ***

Забронировать

Введите адрес кошелька для получения данных:
(Или оставьте поле пустым для использования подключенного адреса)

Просмотр данных пользователя
Просмотр общих данных

Последний выбранный день: Понедельник
Последнее выбранное помещение: Отель *

Очистить данные для пользователя

Очистить общие данные

Узнать баланс

Баланс контракта (wei): 6000000000000000

Вывод средств

Таблица	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Отель *	0	0	0	0	0	0	0
Отель **	0	0	0	0	0	0	0
Отель ***	0	0	0	0	0	0	0
Отель ****	0	0	0	0	0	0	0
Отель *****	0	0	0	0	0	0	0

Студент СПбГУТ:
 Балугут Владислав Владимирович
 ИКСП-81
 Дипломная работа

Введите адрес кошелька для получения данных:

(Или оставьте поле пустым для использования подключенного адреса)

Просмотр данных пользователя

Просмотр общих данных

Последний выбранный день: Понедельник

Последнее выбранное помещение: Отель *

Очистить данные для пользователя

Очистить общие данные

Узнать баланс

Баланс контракта (wei): 6000000000000000

Вывод средств

Таблица	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Отель *	0	0	0	0	0	0	0
Отель **	0	0	0	0	0	0	0
Отель ***	0	0	0	0	0	0	0
Отель ****	0	0	0	0	0	0	0
Отель *****	0	0	0	0	0	0	0

Вывод ошибки “User is not admin”

Подтвердите действие на странице localhost:3002

[illegible]

OK

Назначение администратора и проверка кто в данный момент админ

ENVIRONMENT

Injected Web3

Ropsten (3) network

ACCOUNT

0xBBc...168DB (2.9773532)

GAS LIMIT

3000000

VALUE

10

Wei

CONTRACT

Rent - лв.sol

Deploy

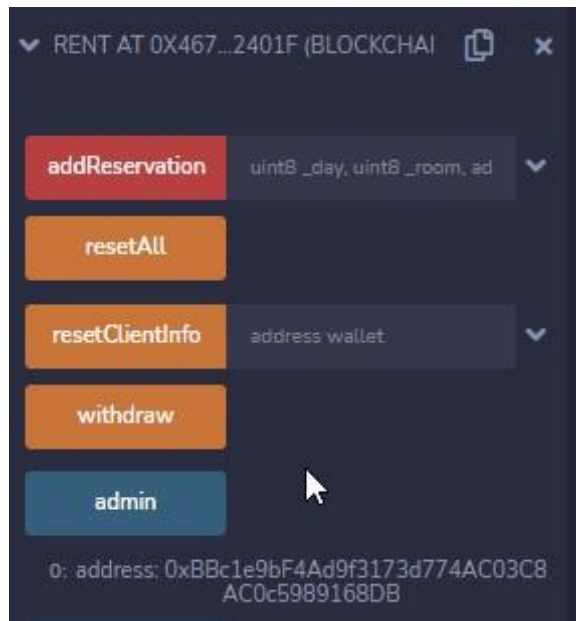
address_admin

☒ Publish to IPFS

OR

At Address

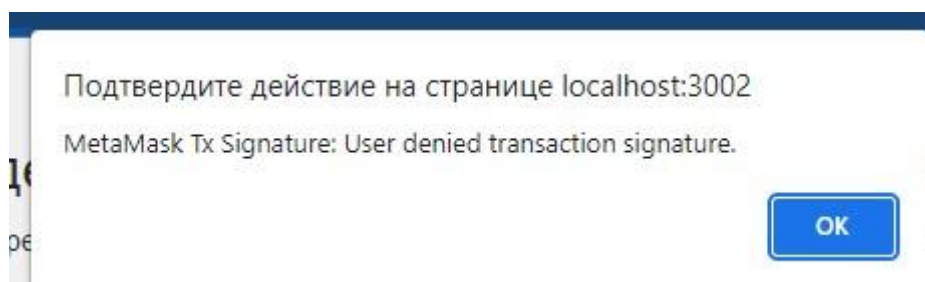
Load contract from Address



Оплата с введенными данными адреса



Ошибка об отмене оплаты



Оплата без введенных данных адреса, по прикрепленному адресу

SPBGUT Diplom

Взаимодействие со смарт-контрактом в сети IOTA

Адрес вашего кошелька: 0xabc1e9b4a99f173d774ac03d3ac0c5989168bd

Введите адрес кошелька для бронирования

Бронирование отеля

Выберите день недели и отель

Понедельник

Отель *

Забронировать

Введите адрес кошелька для получения данных:
(Или оставьте поле пустым для использования подключенного адреса)

Просмотр данных пользователя

Просмотр общих данных

Очистить данные для пользователя

Очистить общие данные

Узнать баланс

Баланс контракта (wei): 6000000000000000

Вывод средств

Account 1

0xABC...40H

Обнаружен новый адрес! Нажмите здесь, чтобы добавить этот адрес в свои подписки ethers.

https://kovanio.net:3002

0xABC...40H: ADD RESERVATION

0.0001

ПОДРОБНОСТИ ДАННЫЕ ШЕСТНАДЦАТИ

Примерные

Изменить

Примерные

0.000000007

0.0000007

Вариант ETH

Рекомендовано

создать

Минимум:

0.000000007

Вариант ETH

Максимум:

0.000000007

Вариант ETH

Студент СПбГУТ:
Багмут Владислав Владимирович
ИКПИ-81
Дипломная работа

Просмотр данных без введенного адреса

Введите адрес кошелька для получения данных:
(Или оставьте поле пустым для использования подключенного адреса)

Просмотр данных пользователя

Просмотр общих данных

Последний выбранный день: Среда
Последнее выбранное помещение: Отель ***

Очистить данные для пользователя

Очистить общие данные

Узнать баланс

Баланс контракта (wei): 6000000000000000

Вывод средств

Таблица

	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Отель *	2	0	0	0	0	0	0
Отель **	1	1	0	0	0	0	0
Отель ***	0	1	2	0	0	0	0
Отель ****	0	0	0	0	0	0	0
Отель *****	0	0	0	0	0	0	0

Студент СПбГУТ:
Багмут Владислав Владимирович
ИКПИ-81
Дипломная работа

Просмотр данных с введенным адресом

Введите адрес кошелька для получения данных:
(Или оставьте поле пустым для использования подключенного адреса)

Просмотр данных пользователя

Просмотр общих данных

Последний выбранный день: Среда
Последнее выбранное помещение: Отель ***

Очистить данные для пользователя

Очистить общие данные

Узнать баланс

Баланс контракта (wei): 6000000000000000

Вывод средств

Таблица

	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресенье
Отель *	2	0	0	0	0	0	0
Отель **	1	1	0	0	0	0	0
Отель ***	0	1	2	0	0	0	0
Отель ****	0	0	0	0	0	0	0
Отель *****	0	0	0	0	0	0	0

Google Chrome

Confirmed transaction

Transaction 58 confirmed! View on Etherscan

[illegible]

Вывод средств

[illegible]

1.4 Выводы по разделу

В четвертом разделе проведен подробный обзор приложения, продемонстрирована работа сайта.

ЗАКЛЮЧЕНИЕ

Целью работы было создание приложения Разработка программного обеспечения блокчейн - сети IOTA для обеспечения взаимодействия устройств интернета вещей, концептом работы было написание программного обеспечения для отдельного бизнеса, который основан на анонимности своих клиентов, был написан сайт ПО, продемонстрирована его работа, так же в дипломную работу были добавлены идеи по будущей реализации проекта, и дополнительная демонстрация возможностей IOTA

Я считаю что проект довольно много обещающий, так как разбирая аналоги я нашел всего один зарубежный аналог такого проекта, причем больших вложений этот проект не потребует,

Считаю работу отличной и многообещающей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Приложения

contract_new.sol

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.9;

contract Rent {
    address payable public admin;
    uint256[7][5] private allReserved;

    struct Client {
        uint256[7][5] reserved;
        uint8 day;
        uint8 room;
    }

    mapping(address => Client) private client;

    constructor(address payable _admin) {
        require(_admin != address(0), "Admin address can't be null");
        admin = _admin;
    }

    modifier onlyAdmin() {
        require(msg.sender == admin, "User is not an admin");
        _;
    }

    // 10**14 - цена для добавления бронирования
    function addReservation(uint8 _day, uint8 _room, address wallet)
    external payable {
        require(_day <= 6 && _room <= 4, "Unappropriate parametres");
        require(msg.value >= 10**14, "Unappropriate amount of wei");
        // 100000000000000 wei

        client[wallet].reserved[_day][_room] += 1;
        allReserved[_day][_room] += 1;
    }
}
```

```

        client[wallet].day=_day;
        client[wallet].room=_room;
    }

    function resetClientInfo(address wallet) external onlyAdmin {
        delete client[wallet];
    }

    function resetAll() external onlyAdmin {
        delete allReserved;
    }

    function getClientReservations(address wallet) external view
returns(uint8 _day, uint8 _room, uint256[7][5] memory _allroom) {
        if(msg.sender!=wallet){
            if(msg.sender!=admin)
                revert("User is not an admin");
        }
        _day = client[wallet].day;
        _room = client[wallet].room;
        _allroom = client[wallet].reserved;
    }

    function getAllTable() external view onlyAdmin returns(uint256[7][5]
memory){
        return allReserved;
    }

    function getBalance() external view onlyAdmin returns (uint) {
        return address(this).balance;
    }
    function withdraw() external onlyAdmin {
        admin.transfer(address(this).balance);
    }
}

```

style.css

```
body {  
    margin: 0;  
}  
  
.header {  
    background-color: #f1f1f1;  
    padding: 15px;  
    text-align: center;  
}  
  
.header h1 {  
    font-family: Arial, Helvetica, sans-serif;  
}  
  
.header h2 {  
    font-family: Arial, Helvetica, sans-serif;  
}  
  
.header span {  
    font-family: Arial, Helvetica, sans-serif;  
}  
  
.main {  
    background-color: aliceblue;  
}  
  
.useraddress {  
    font-size: 17px;  
    font-family: Arial, Helvetica, sans-serif;
```

```
}
```

```
.input-user-wallet {  
    font-size: 17px;  
    height: 35px;  
    width: 420px;  
    border-radius: 18px;  
    border: 1px solid;  
    padding: 10px;  
    margin-top: 10px;  
    margin-bottom: 10px;  
}
```

```
.info {  
    font-size: 17px;  
    font-family: Arial, Helvetica, sans-serif;  
}
```

```
.dayselector {  
    font-size: 16px;  
    border-radius: 3px;  
    width: 300px;  
    height: 30px;  
    text-align: center;  
    border: 1px solid black;  
    background-color: white;  
    cursor: pointer;  
}
```

```
.placeselector {
```

```
font-size: 16px;
border-radius: 3px;
width: 300px;
height: 30px;
text-align: center;
border: 1px solid black;
background-color: white;
cursor: pointer;
}
```

```
.inputread {
font-size: 17px;
height: 35px;
width: 420px;
border-radius: 18px;
border: 1px solid;
padding: 10px;
margin-bottom: 15px;
margin-top: 10px;
}
```

```
.debug-input-label {
font-size: 17px;
font-family: Arial, Helvetica, sans-serif;
}
```

```
.footer {
background-color: #f1f1f1;
padding: 10px;
text-align: center;
```



```
}
```

```
.footer h4 {  
    text-align: right;  
    font-family: Arial, Helvetica, sans-serif;  
}
```

```
.reservation-block {  
    padding-top: 25px;  
    margin-left: 100px;  
}
```

```
.admin-panel {  
    padding-top: 50px;  
    margin-left: 100px;  
}
```

```
.reserve-button {  
    font-family: Consolas;  
    font-size: 20px;  
    border-radius: 25px;  
    text-align: center;  
    border: 1px solid #4CAF50;  
    width: 300px;  
    height: 43px;  
    background-color: rgb(255, 255, 255);  
    cursor: pointer;  
}
```

```
.reserve-button:hover {
```

```
background: #4CAF50;
color: rgb(255, 255, 255);
}
```

```
.getting-data {
width: 85%;
height: 85%;
border: 1px solid;
padding: 25px;
border-radius: 12px;
}
```

```
.readUser {
font-family: Consolas;
font-size: 15px;
border-radius: 3px;
text-align: center;
border: 1px solid #000000;
width: 300px;
height: 30px;
background-color: rgb(235, 235, 235);
cursor: pointer;
margin-bottom: 10px;
}
```

```
.readUser:hover {
background-color: rgb(209, 209, 209);
}
```

```
.getalltable {
```

```
font-family: Consolas;
font-size: 15px;
border-radius: 3px;
text-align: center;
border: 1px solid #000000;
width: 300px;
height: 30px;
background-color: rgb(235, 235, 235);
cursor: pointer;
margin-bottom: 10px;
}
```

```
.getalltable:hover {
    background-color: rgb(209, 209, 209);
}
```

```
.clearusertable {
    font-family: Consolas;
    font-size: 15px;
    border-radius: 3px;
    text-align: center;
    border: 1px solid #000000;
    width: 300px;
    height: 30px;
    background-color: rgb(235, 235, 235);
    cursor: pointer;
    margin-top: 10px;
}
```

```
.clearusertable:hover {
```

```
background-color: rgb(209, 209, 209);  
}
```

```
.clearalltable {  
    font-family: Consolas;  
    font-size: 15px;  
    border-radius: 3px;  
    text-align: center;  
    border: 1px solid #000000;  
    width: 300px;  
    height: 30px;  
    background-color: rgb(235, 235, 235);  
    cursor: pointer;  
    margin-top: 10px;  
}
```

```
.clearalltable:hover {  
    background-color: rgb(209, 209, 209);  
}
```

```
.withdraw {  
    font-family: Consolas;  
    font-size: 15px;  
    border-radius: 3px;  
    text-align: center;  
    border: 1px solid #000000;  
    width: 300px;  
    height: 30px;  
    background-color: rgb(235, 235, 235);  
    cursor: pointer;
```

```
    margin-top: 10px;
}

.withdraw:hover {
    background-color: rgb(209, 209, 209);
}

.getbalance {
    font-family: Consolas;
    font-size: 15px;
    border-radius: 3px;
    text-align: center;
    border: 1px solid #000000;
    width: 300px;
    height: 30px;
    background-color: rgb(235, 235, 235);
    cursor: pointer;
    margin-top: 10px;
    margin-bottom: 10px;
}

.getbalance:hover {
    background-color: rgb(209, 209, 209);
}
```

web3_script.js

```
let addressContractMain = newFunction();
const newLocal = "0x467FBcfA73D5A920D428716D2435fC30D172401f";
var addressContractRopsten = newLocal;
var addressContractIOTA =
"0xAfC8Bc679f8e8c34643b8C9786dE3A8d001E7eaC";

var contractController;

var
day=["Понедельник","Вторник","Среда","Четверг","Пятница","Суббота","Вос
кресенье"];

var room = ["ОТЕЛЬ *","ОТЕЛЬ**","ОТЕЛЬ ***","ОТЕЛЬ ****","ОТЕЛЬ
*****"];

var contract_allroom;

var wallet, signer, current_network;

function newFunction() {
    return "0x467FBcfA73D5A920D428716D2435fC30D172401f";
}

function save() {
    var readWallet = $('#readWallet').val();
    if (readWallet == "") readWallet=wallet;
    if(!ethers.utils.isAddress(readWallet)) {
        return alert("invalid address");
    }
    var addDay = $('#addDay').val();
    var addRoom = $('#addRoom').val();
    console.log("addDay", addDay);
    console.log("addRoom", addRoom);
```

```

        contractController.addReservation(addDay, addRoom,wallet, { value:
ethers.utils.parseEther("0.0001") }).then((err, data) => {
            console.log("data", data);
        }).catch(function (error) {
            alert(error.message);
        });
    }
}

```

```

function readUser() {
    var readWallet = $('#readWallet').val();
    if (readWallet == "") readWallet=wallet;
    if(!ethers.utils.isAddress(readWallet)) {
        return alert("invalid address");
    }
    $('#showResult').hide();
    contractController.getClientReservations(readWallet).then((data) => {
        console.log("data", data);
        var contract_day = Number(data._day)
        var contract_room = Number(data._room)
        contract_allroom = data._allroom;
        console.log("data._day", contract_day);
        console.log("data._room", contract_room);
        console.log("data._allroom", contract_allroom);
        $('#showResult').show();
        $('#showDay').html(contract_day);
        $('#showRoom').html(contract_room);
        $('#targetDay').html(day[contract_day]);
        $('#targetRoom').html(room[contract_room]);
        generate_table();
    }).catch(function (error) {

```

```

        alert(error.message);
    });
}

function startApp() {
    $('#showResult').hide();
    console.log("wallet", wallet);
    $('#wallet').html(wallet);
}

function clearusertable(){
    var readWallet = $('#readWallet').val();
    if (readWallet == "") readWallet=wallet;
    if(!ethers.utils.isAddress(readWallet)) {
        return alert("invalid address");
    }
    contractController.resetClientInfo(readWallet).then((data) => {
        console.log("data", data);
    }).catch(function (error) {
        alert(error.message);
    });
}

function clearalltable(){
    contractController.resetAll().then((data) => {
        console.log("data", data);
    }).catch(function (error) {
        alert(error.message);
    });
}

```



```

function getalltable(){
    contractController.getAllTable().then((data) => {
        console.log("data", data);
        contract_allroom = data;
        console.log("room", contract_allroom);
        generate_table();
    }).catch(function (error) {
        alert(error.message);
    });
}

```

```

function withdraw(){
    contractController.withdraw().then((data) => {
        console.log("data",data);
    }).catch(function (error) {
        alert(error.message);
    });
}

```

```

function getbalance(){
    contractController.getBalance().then((data) => {
        console.log("data",data);
        balance=data.toNumber();
        $('#balance').html(balance);
        console.log("balance",balance);
    }).catch(function (error) {
        alert(error.message);
    });
}

```

```

window.addEventListener('load', async function () {
    window.ethereum.enable().then(provider = new
ethers.providers.Web3Provider(window.ethereum));
    signer = provider.getSigner();
    current_network = ethereum.networkVersion;
    console.log("current_network", current_network);

    const accounts = await ethereum.request({ method: 'eth_accounts' });
    wallet = accounts[0];

    initContracts();
})

function initContracts() {
    var addressContractController = {
        "1": addressContractMain,
        "3": addressContractRopsten,
        "1074": addressContractIOTA
    }
    $.ajax({
        url: 'abi.json',
        dataType: 'json',
        success: function (data) {
            var abiContract = data;
            var contractAddress = addressContractController[current_network];
            contractController = new ethers.Contract(contractAddress,
abiContract, signer);
            startApp();
        }
    }

```

```

    });
}

//////////TABLE//////////

function generate_table() {
    let
day=["Таблица","Понедельник","Вторник","Среда","Четверг","Пятница","Суб
бота","Воскресенье"];

    // get the reference for the body
    var body = document.getElementById("tablearr");
    // creates a <table> element and a <tbody> element
    var tbl = document.createElement("table");
    var tblBody = document.createElement("tbody");
    console.log(body.childNodes);
    if(body.childNodes()==true){
        body.removeChild( body.childNodes[0] );
    }

    // creating all cells
    // creates a table row
    var row = document.createElement("tr");

    for (var j = 0; j < 8; j++) {
        // Create a <td> element and a text node, make the text
        // node the contents of the <td>, and put the <td> at
        // the end of the table row
        var cell = document.createElement("td");
        var cellText = document.createTextNode(day[j]);
        cell.appendChild(cellText);
        row.appendChild(cell);
    }
}

```

```

    }

    // add the row to the end of the table body
    tblBody.appendChild(row);

    for (var i = 0; i < 5; i++) {
        // creates a table row
        var row = document.createElement("tr");
        var cell = document.createElement("td");
        var cellText = document.createTextNode(room[i]);
        cell.appendChild(cellText);
        row.appendChild(cell);
        for (var j = 0; j < 7; j++) {
            // Create a <td> element and a text node, make the text
            // node the contents of the <td>, and put the <td> at
            // the end of the table row
            var cell = document.createElement("td");
            var cellText = document.createTextNode(contract_allroom[i][j]);
            cell.appendChild(cellText);
            row.appendChild(cell);
        }

        // add the row to the end of the table body
        tblBody.appendChild(row);
    }

    // put the <tbody> in the <table>
    tbl.appendChild(tblBody);
    // appends <table> into <body>
    body.appendChild(tbl);

```

```
// sets the border attribute of tbl to 2;
tbl.setAttribute("border", "2");
}
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <link rel="stylesheet" href="css/style.css">
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
+0n0xVW2eSR5OomGNYDnhzAbDsOXxcvSN1TPprVMTNDbiYZCxYbOOl7+
AMvyTG2x" crossorigin="anonymous">
  <script src="https://kit.fontawesome.com/45a7f34b95.js"
crossorigin="anonymous"></script>
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.1/dist/js/bootstrap.bundle.min.js"
integrity="sha384-
gtEjrD/SeCtmISkJKNUaaKMoLD0//ElJ19smozuHV6z3Iehds+3Ulb9Bn9Plx0x4"
crossorigin="anonymous"></script>
  <link type="image/x-icon" rel="shortcut icon" href="css/favicon.ico">
  <title>Diplom</title>
</head>
<body>
```

```

<!-- Заголовок -->
<div class="header">
    <h1>SPBGUT Diplom</h1>
    <h2>Взаимодействие со смарт-контрактом в сети IOTA</h2>
    <span>Адрес вашего кошелька: </span> <span id="wallet"></span>
</div>

<!-- Основная часть -->
<div class="main">
    <div class="reservation-block">
        <label for="addWallet" class="useraddress">Введите адрес
кошелька для бронирования:</label><br>
        <input type="text" class="input-user-wallet" id="addWallet">

        <p class="info"><mark>Бронирование
отеля</mark><br>Выберите день недели и отель:</p>
        <form>
            <select id="addDay" class="dayselector">
                <option value="0">Понедельник</option>
                <option value="1">Вторник</option>
                <option value="2">Среда</option>
                <option value="3">Четверг</option>
                <option value="4">Пятница</option>
                <option value="5">Суббота</option>
                <option value="6">Воскресенье</option>
            </select>
        </form>
        <br>
        <form>
            <select id="addRoom" class="placeselector">

```

```

        <option value="0">Отель *</option>
        <option value="1">Отель **</option>
        <option value="2">Отель ***</option>
        <option value="3">Отель ****</option>
        <option value="4">Отель *****</option>
    </select>

</form>

<br>

    <button class="reserve-button"
onclick="save()">Забронировать</button>
</div>

<!-- Для таблиц -->
<div class="admin-panel">
    <div class="getting-data">
        <label for="readWallet" class="debug-input-label">Введите
адрес кошелька для получения данных:</label>

        <br>(Или оставьте поле пустым для использования
подключенного адреса)

        <br>
        <input type="text" class="inputread" id="readWallet">
        <br>
        <button class="readUser" onclick="readUser()">Просмотр
данных пользователя</button>

        <button class="getalltable" onclick="getalltable()">Просмотр
общих данных</button>

    <div id="showResult">
        <div>

```

```

        <span style="font-family: Arial, Helvetica, sans-serif; font-
size: 16px;">Последний выбранный день: </span> <span
id="targetDay"></span>
    </div>
    <div>
        <span style="font-family: Arial, Helvetica, sans-serif; font-
size: 16px;">Последнее выбранное помещение: </span> <span
id="targetRoom"></span>
    </div>
</div>
<div class="form-group">
    <button class="clearusertable"
onclick="clearusertable()">Очистить данные для пользователя</button>
    <br>
    <button class="clearalltable" onclick="clearalltable()">Очистить
общие данные</button>
    <br>
    <button class="getbalance" onclick="getbalance()">Узнать
баланс</button><br>
        <span style="font-family: Arial, Helvetica, sans-serif; font-size:
16px;">Баланс контракта (wei): </span> <span id="balance"></span>
    <br>
    <button class="withdraw" onclick="withdraw()">Вывод
средств</button>
</div>
<br>
<div id="tablearr"></div>
</div>
</div>
</div>

```



```

<!-- Footer -->
<div class="footer">
    <h4>Студент СПбГУТ:<br>Багмут Владислав
Владимирович<br>ИКПИ-81<br>Дипломная работа</h3>
</div>
<script src="https://cdn.ethers.io/lib/ethers-5.2.umd.min.js"
    type="application/javascript"></script>
<script src='./js/web3_script.js' type='text/javascript'></script>
</body>
</html>

```

bs-config.js

```

module.exports = {
  "port": 3000,
  "files" : "./src/**/*.{js, html, css}",
  "server": { "baseDir": "./src" }
}

```

package.json

```

{
  "name": "adminpage",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "start": "lite-server -c ./bs-config.js",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "vpomo",
  "license": "MIT",

```

```
"dependencies": {  
  "lite-server": "^2.6.1"  
}
```