**ТИПОВОЕ КОНКУРСНОЕ ЗАДАНИЕ**

***Для Регионального чемпионата***

***Чемпионатный цикл 2021-2022 года***

**компетенции**

**«РАЗРАБОТКА РЕШЕНИЙ С ИСПОЛЬЗОВАНИЕМ   
БЛОКЧЕЙН ТЕХНОЛОГИЙ»**

**для основной возрастной категории**

**16-22 года**

*Конкурсное задание включает в себя следующие разделы:*

[1. Форма участия в конкурсе: 2](#_Toc66870131)

[2. Общее время на выполнение задания: 2](#_Toc66870132)

[3. Задание для конкурса 2](#_Toc66870133)

[4. Модули задания и необходимое время 2](#_Toc66870134)

[5. Критерии оценки. 17](#_Toc66870135)

[6. Приложения к заданию. 18](#_Toc66870136)



1. **Форма участия в конкурсе**:

Командный конкурс. 2 конкурсанта.

Один из конкурсантов специализируется на разработке смарт-контракта для блокчейн платформы. Второй конкурсант специализируется на разработке пользовательских интерфейсов для работы с информационными системами.

1. **Общее время на выполнение задания:**18 ч.
2. **Задание для конкурса**

Целью конкурсного задания является создание и презентация программного решения с использованием технологии блокчейн. Используемая(ые) блокчейн платформа(ы) объявляется конкурсантам предварительно.

Согласно конкурсному заданию, есть ряд магазинов, в которые поставщик поставляет товары (свежезамороженные морепродукты). Поставка товаров требует определенного режима хранения. Магазин реализует поставленный товар покупателям. Покупатели могут влиять на общий рейтинг магазина, оставляя свои или подтверждая чужие отзывы о работе магазина. Общий рейтинг магазина влияет на закупочную стоимость поставляемого в магазин товара.

1. **Модули задания и необходимое время**

Таблица 1.

|  |  |  |  |
| --- | --- | --- | --- |
| **Наименование модуля** | | **Соревновательный день (С1, С2, С3)** | **Время на задание** |
| **A** | Реализация смарт-контракта | С1-С2 | 10 |
| **B** | Реализация интерфейса | С1-С2 | 4 |
| **C** | Аудит смарт-контракта | С3 | 2 |
| **D** | Презентация проекта | С3 | 2 |

***Модуль А: Реализация смарт-контракта***

В рамках первого Модуля задания необходимо:

1. Выполнить инициализацию блокчейн-системы с указанными в задании пользователями.
2. Реализовать функцию для регистрации новых пользователей системы
3. Реализовать функцию двухфакторной авторизации пользователей системы
4. Реализовать функцию смены ролей пользователей
5. Реализовать функцию по работе с «Книгой жалоб и предложений магазина»
6. Продемонстрировать работу решения в соответствии с заданным сценарием.

**Пояснения: Ролевая модель.**

В системе предусмотрены следующие роли: банк, магазины, администратор системы, продавец, покупатель, поставщик, гость.

В рамках задания для Модуля 1 у каждого пользователя системы имеется свой личный кабинет, из которого доступен следующий функционал для каждой из ролей:

***Банк*** в рамках задания – это аккаунт в блокчейн-цепи, который задается в генезис-блоке. При старте системы на счету данного аккаунта должно быть 10000 eth. Банк дает заем новому магазину в виде стартового капитала в 1000 eth.

***Магазин*** в рамках задания – это аккаунт в блокчейн-цепи, который задается в генезис-блоке либо вводится в систему в ходе ее использования путем регистрации новых аккаунтов. Всего в систему при старте должно быть заведено 9 магазинов согласно табл. 1. Каждый магазин представляет собой адрес в блокчейн-системе со своим балансом.

***Поставщик*** в рамках задания – это аккаунт в блокчейн-цепи, который задается в генезис-блоке.

Таблица 1 – Список магазинов

|  |  |  |
| --- | --- | --- |
| № магазина | город | Баланс, eth |
| 1 | Дмитров | 1000 |
| 2 | Калуга | 900 |
| 3 | Москва | 1050 |
| 4 | Рязань | 700 |
| 5 | Самара | 2000 |
| 6 | Санкт-Петербург | 2300 |
| 7 | Таганрог | 0 |
| 8 | Томск | 780 |
| 9 | Хабаровск | 1500 |

***Администратор системы***

1. Повышает обычного покупателя до роли продавец
2. Понижает продавца до роли покупатель
3. Может переключиться к роли покупатель
4. Может ввести в систему новых администраторов.
5. Может добавить в систему новый магазин (назначив к роли магазин существующий в сети блокчейн аккаунт)
6. Может удалить из системы существующий магазин (аккаунт магазина перестает быть в статусе Магазин, все продавцы магазина становятся обычными покупателями).

В личном кабинете администратора отображается его логин, список запросов от покупателей и продавцов на повышение/понижение роли, список всех администраторов системы, список всех продавцов магазинов (с сортировкой по магазину).

***Продавец***

1. Может оставить комментарий к записи в книге отзывов и предложений
2. Может переключиться к роли покупатель
3. Может отправить запрос на понижение до роли покупатель

В личном кабинете продавца отображается его логин, ФИО, город, магазин, оставленные комментарии и количество их подтверждений/опровержений.

***Покупатель***

1. Может выставить оценку работы магазина/продавца
2. Может оставить комментарий к оценке
3. Может подтвердить/опровергнуть оценки и комментарии других пользователей
4. Может подать запрос на повышение до роли продавец.

В личном кабинете покупателя отображается его логин, ФИО, текущий баланс, оставленные комментарии и количество их подтверждений/опровержений.

***Гость***

Не имеет личного кабинета, не может совершать действий в системе, но может просматривать информацию, записанную в книгу Жалоб и предложений.

Функционал каждого пользователя должен быть доступен строго в соответствии с заданной ролью. Каждый новый пользователь системы регистрируется в статусе «покупатель», после чего может быть повышен до роли продавца или администратора. При регистрации пользователя указывается его ФИО и логин на сайте.

***Пояснения: Книга жалоб и предложений.***

Все записи в книге жалоб и предложений должны храниться в блокчейне. Оставлять записи в книге могут только зарегистрированные пользователи системы в статусе «Покупатель». Оставляя отзыв, покупатель должен получить возможность выбрать магазин из доступного списка, для которого он оставляет отзыв. С каждым комментарием пользователь может оставить оценку по 10-бальной системе. Оставлять ответ на комментарий могут другие покупатели или работники **этого** магазина. Продавцы не могут выставлять оценки магазина при написании комментариев. Рядом с каждым комментарием (любого уровня) должно отображаться количество лайков и дизлайков данного комментария другими пользователями системы.

***При старте в системе должны быть зарегистрированы:***

* 9 адресов для магазинов с заданным балансом.
* Банк (баланс 10000 eth)
* Поставщик Золотая рыбка (логин goldfish, на счету 100eth).
* 1 Администратор системы Иванов Иван Иванович (логин ivan, на счету 50 eth).
* Продавец Семенов Семен Семенович (логин semen,магазин №1, на счету 70 eth).
* Покупатель ПетровПетрПетрович (логин petr,на счету 80 eth).

Остальные пользователи системы должны иметь возможность регистрироваться в системе самостоятельно.

**ВАЖНО!!!** Реализованное решение должно иметь bat-файл для быстрого разворачивания и запуска решения. Необходимо привести инструкции по установке, запуску и эксплуатации готового решения. Код разработанного решения должен быть задокументирован и покрыт тестами.

**ПОСЛЕ КОМАНДЫ ГЭ ИЛИ ЗГЭ «НАЧАЛО ДЕМОНСТРАЦИИ»** необходимо продемонстрировать работоспособность разработанной системы. Демонстрация работы приложения осуществляется путем видеозахвата экрана и комментариев разработчиков. Если выполнено и графическое, и консольное решение, то необходимо на одном ПК запустить GUI, на втором ПК запустить консольное приложение. Если выполнено только одно решение (или консоль или GUI), то запустить его на обоих ПК. При демонстрации необходимо показывать изменения состояния блокчейн-системы (это можно сделать любым доступным способом, например, с использованием Remix). НО!!! Функционал будет засчитан только в том случае, если он срабатывает из консоли/GUI. Перед началом необходимо зафиксировать время старта демонстрации.

Для демонстрации решения необходимо выполнить действия из табл. 2.

Таблица 2 – Порядок действий при демонстрации решения

|  |  |
| --- | --- |
| **ПК1 (GUI)** | **ПК2 (консоль)** |
| 1. Зайти в аккаунт Иванова Ивана Ивановича в роли администратор, показать личный кабинет. | 1. Зайти в аккаунт Семенов Семен Семенович в роли продавец, показать личный кабинет. |
| 1. Переключиться из роли Иванова Ивана Иванович администратор к роли покупатель, показать личный кабинет (доступный функционал) | 1. Переключиться из роли Семенов Семен Семенович продавец к роли покупатель, показать личный кабинет (доступный функционал) |
| 1. Переключиться из роли Иванова Ивана Иванович покупатель к роли продавец, показать личный кабинет (доступный функционал) | 1. Переключиться из роли Семенов Семен Семенович покупатель к роли продавец, показать личный кабинет (доступный функционал) |
| 1. Показать список магазинов, доступных для управления администратору. Добавить Магазин №10, г. Пенза (по заранее сгенерированному адресу в сети блокчейн, на счету у адреса должно быть 0 eth). Показать список магазинов, доступных для управления администратору. | 1. Зайти в аккаунт Магазина №10. Показать счет. Отправить в банк запрос на получение денег (1000 eth). |
| 1. Зайти в аккаунт Банк. Показать счет. Подтвердить получение Магазином №10 займа в 1000 eth. Показать счет Банка. | 1. Показать счет в Магазине №10. |
| 1. Зарегистрировать нового покупателяСергеева Сергея Сергеевича (логин serg). Подать от него запрос стать продавцом магазина №10 | 1. Зайти в аккаунт Администратора. Подтвердить запрос Сергеева Сергея Сергеевича. |
| 1. Показать статус Сергеева Сергея Сергеевича. Переключиться к роли продавец. | 1. Удалить из списка магазинов Магазин №10. Показать список магазинов. |
| 1. Показать личный кабинет Сергеева Сергея Сергеевича. Попробовать переключиться к кабинету продавца. Оставить в книге «Жалоб и предложений» отзыв для Магазина №2 «Все супер!». Оставить оценку магазину 8 из 10. Оставить в книге «Жалоб и предложений» отзыв для Магазина №1 «Рыба не свежая!». Оставить оценку магазину 1 из 10. | 1. Просмотреть книгу «Жалоб и предложений» в статусе гостя.   Залогиниться под аккаунтом Семена Семена Семеновича. Переключиться к роли продавец. Оставить к комментарию «Рыба не свежая!» ответный комментарий «Магазин приносит свои извинения». |
| 1. Зайти в кабинет Петрова Петра Петровича. Подтвердить запись «Все супер!» и опровергнуть запись «Рыба не свежая!». | 1. Зайти в аккаунт Семенова Семена Семеновича. Подать запрос на понижение роли до уровня «покупатель». |
| 1. Зайти в аккаунт Администратора. Подтвердить запрос Семенова Семена Семеновича. | 1. Показать личный кабинет Семенова Семена Семеновича, попробовать переключиться к роли продавец. |
| 1. Зайти в аккаунт Администратора. Подтвердить запрос Владимирова Владимира Владимировича. | 1. Зарегистрировать нового покупателяВладимиров Владимир Владимирович (логин vlad). Подать от него запрос стать продавцом магазина №4. |
| 1. Зайти в аккаунт Владимирова Владимира Владимировича. Показать личный кабинет. Переключиться к роли продавец. Переключиться к роли покупатель. | 1. Зайти в аккаунт Петрова Петра Петровича. Подать запрос на повышение до роли администратора. |
| 1. Зайти в аккаунт Администратора. Подтвердить запрос Петрова Петра Петровича. | 1. Зайти в аккаунт Петрова Петра Петровича. Переключиться к роли администратор. Переключиться к роли покупатель. |
| 1. Зайти в аккаунт Поставщика. Показать баланс. | 1. Зайти в аккаунт Поставщика. Показать баланс. |

По желанию можно продемонстрировать дополнительный функционал.

Также дополнительно требуется продемонстрировать тестирование функций полученного решения.

По итогу выполнения демонстрации заполнить табл. 3 соответствия адресов и пользователей системы

Таблица 3 – Соответствие аккаунтов и адресов в сети Ethereum

|  |  |  |
| --- | --- | --- |
| Пользователь | Адрес в сети Ethereum | Пароль |
| Иванов Иван Иванович |  |  |
| Семенов Семен Семенович |  |  |
| Петров Петр Петрович |  |  |
| Магазин №1 |  |  |
| Магазин №2 |  |  |
| Магазин №3 |  |  |
| Магазин №4 |  |  |
| Магазин №5 |  |  |
| Магазин №6 |  |  |
| Магазин №7 |  |  |
| Магазин №8 |  |  |
| Магазин №9 |  |  |
| Магазин №10 |  |  |
| Сергеев Сергей Сергеевич |  |  |
| Владимиров Владимир Владимирович |  |  |

**ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «WSR2021». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

***Модуль B: Реализация интерфейса***

Для каждого пользователя системы должен быть доступен реализованный функционал посредством консольного и графического интерфейсов.

Через интерфейс должен быть реализован следующий функционал:

1. Авторизация в системе (все).
2. Регистрация в системе (покупатель).
3. Повышение/понижение ролей (администратор).
4. Ввод нового администратора (администратор).
5. Ввод/Удаление нового Магазина (администратор).
6. Просмотр списка доступных магазинов(администратор).
7. Запрос на повышение/понижение ролей(покупатель, продавец).
8. Комментарий в книге жалоб и предложений (покупатель, продавец).
9. Оценка комментарию (покупатель).
10. Подтверждение/опровержение оставленного комментария (покупатель).

В интерфейсе зарегистрированных пользователей системы (администратор, продавец, покупатель) должен быть «личный кабинет», в котором отображаются:

Все личные данные пользователя, его текущий баланс, его статус (роль), история всех совершенных им действий. Для роли администратор также должен быть доступен список всех Магазинов, заведенных в систему.

В интерфейсе для аккаунта Банк должен отображаться: текущий баланс, кому был сделан заем и его сумма (адрес сети блокчейн+номер магазина+город магазина).

В интерфейсе для аккаунта Магазин должны отображаться: текущий баланс, номер магазина, город, список продавцов, закрепленных за магазином.

Должна быть дана инструкция по установке, запуску и использования решения.

**ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «WSR2021». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

***Модуль A: Реализация смарт-контракта***

Реализация второй части функционала системы. Можно использовать наработки предыдущего дня или зафиксировать в коде следующие данные:

*При старте в системе должны быть зарегистрированы:*

* 9 адресов для магазинов с заданным балансом (табл. 1).
* Банк (баланс 10000 eth)
* Поставщик Золотая рыбка (логин goldfish, на счету 100eth).
* 1 Администратор системы Иванов Иван Иванович (логин ivan, на счету 50 eth).
* Продавец Семенов Семен Семенович (логин semen,магазин №1, на счету 70 eth).
* Продавец Евгеньева Евгения Евгеньевна (логин ugin,магазин №3, на счету 80 eth).
* Продавец Дмитриев Дмитрий Дмитриевич (логин dima,магазин №5, на счету 90 eth).
* Продавец Васильев Василий Васильевич (логин vasya,магазин №7, на счету 100eth).
* Продавец Игорев Игорь Игоревич (логин igor, магазин №8, на счету 150eth).
* Покупатель Романов Роман Романович (логин roman,на счету 80 eth).
* Покупатель Николаев Николай Николаевич (логин nikola,на счету 90 eth).
* Покупатель Олегов Олег Олегович (логин oleg,на счету 100 eth).
* Покупатель Петров Петр Петрович (логин petr,на счету 110 eth).
* Покупатель Александрова Александра Александровна (логин alex,на счету 120 eth).

Также к моменту старта системы заполнена книга жалоб и предложений следующим образом (табл. 4).

Таблица 4 – Записи в книгу жалоб и предложений

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Магазин** | **Комментарий верхнего уровня** | **Ответ на комментарий** | **Кто оставил комментарий** | **Оценка**  (по 10-бальной) | **Кол-во «+»** | **Кол-во**  **«–»** |
| Магазин №1 | Отличное качество товара! | – | oleg | 10 | 25 | 0 |
| Магазин №1 | – | подтверждаю | petr | 9 | 20 | 2 |
| Магазин №1 | Быстрое обслуживание! | – | petr | 9 | 15 | 1 |
| Магазин №1 | – | А я долго ждал(( | nikola | 2 | 0 | 11 |
| Магазин №1 | – | Магазин приносит свои извинения за длительное ожидание | semen | – | 40 | 15 |
| Магазин №3 | Ничего особенного | – | roman | 5 | 3 | 20 |
| Магазин №3 | – | Не согласен с вами, все супер! | petr | 10 | 15 | 0 |
| Магазин №3 | Спасибо, мне все понравилось! | – | alex | 8 | 23 | 1 |
| Магазин №3 | – | И мне! | roman | 9 | 36 | 5 |
| Магазин №8 | Мне нахамил продавец. Больше не приду к вам! | – | alex | 1 | 10 | 2 |
| Магазин №8 | – | Поддерживаю, ужасный сервис! | petr | 2 | 11 | 0 |
| Магазин №8 | Сервис в магазине на троечку. | – | oleg | 3 | 15 | 2 |

***Товары поставщика***

Поставщик поставляет товары в магазин. Товар является объектом контракта, которому присущи следующие атрибуты: название, производитель, дата изготовления, срок годности, температура хранения, единица измерения (или кг или шт), базовая цена товара. Поставщик может создавать любые объекты, которые он готов поставить в магазины. При этом Магазины (и только магазины!) должны получить возможность увидеть товары, заведенные в систему.

***Рейтинг магазина***

Рейтинг магазина рассчитывается следующим образом.

Не учитываются комментарии, для которых количество «-» (дизлайков) больше количества «+» (лайков).

Не учитываются комментарии, для которых количество «+» (лайков) меньше 10.

Не учитываются комментарии, оставленные продавцами магазина.

Для всех остальных комментариев производится расчет по формуле:

Р = ,

где

Р – рейтинг магазина

– i-ый отзыв

– количество плюсов для i-ого отзыва

– количество минусов для i-ого отзыва

N – общее количество отзывов.

**ВНИМАНИЕ!!!** Округление рейтинга делаем до сотых долей по правилам округления. Рейтинг должен отображаться в аккаунте Магазина. Для Магазинов, у которых нет отзывов, рейтинг равен 0.

Пример расчетов для Магазина №1 в соответствии с таблицей 4:

= = = = 8,88

***Поставка товаров от поставщика в магазин***

Поставка товаров в магазин осуществляется по следующей схеме.

Заказ продуктов осуществляется из аккаунта Магазина. Для каждого выбранного товара указывается объем закупаемого товара (в кг или шт). После этого производится расчет конечной цены товара на основе базовой цены товара, рейтинга и объема закупаемого товара по следующему принципу.

ЦТ = (БЦ – )\*К\*КЛ,

где

ЦТ – цена товара

БЦ – базовая цена

КЛ – количество товара (в кг или шт).

К – коэффициент (К = 1, если КЛ<=100; К = 0,95, если 100<КЛ<=1000; К = 0,9, если КЛ>1000)

Например, для Магазина №1 с рейтингом 8,88 закупка 200 кг креветок по базовой цене 0,03 eth за килограмм составит:

ЦТ = (0,03 – )\*0,95\*200 = 5,19348 eth

Магазин должен подтвердить свою готовность купить товар по рассчитанной стоимости. После подтверждения товар отгружается со склада. Товар должен доставляться в условиях определенного температурного режима (характерного для конкретного заказанного товара). Считаем, что функция генератора случайных чисел будет являться датчиком температуры. За время доставки товара должно быть сгенерировано 6 транзакций в блокчейн-системе с показаниями температуры (случайные числа в диапазоне от -50 до +50). Если температурный режим был нарушен (выходит за рамки температурного режима, установленного для товара), то цена поставки снижается на 10% за каждое нарушение. Если температурный режим был нарушен, то Магазину выводится конечная цена продукта (с условием снижения цены в случае нарушения температурного режима) и предлагается сделать выбор: принять товар или отказаться от товара. Если Магазин выбрал принять товар, то поставщик получает конечную сумму стоимости товара, а Магазину возвращается остаток (10% за каждое нарушение). Если Магазин выбирает отказаться от товара, то Магазин получает полный возврат стоимости. Если условия доставки не были нарушены или если Магазин принял товар, то данный товар становится доступен для покупки обычными пользователями по цене, увеличенной на 50% по отношению к закупочной.

*Например.*

Магазин №1 готов купить 200 кг креветок по цене 5,19348 eth. Температурный режим хранения товара от -20 до -5. При доставке датчик показал следующие значения: -10, -5, 0, -7, 1, -15. Температурный режим был нарушен дважды. Поэтому итоговая цена товара будет равна:

5,19348\*0,8 = 4,154784 eth – закупочная цена за 200 кг

4,154784/200 = 0,02077392 – закупочная цена за 1 кг

0,02077392\*1,5 = 0,03116088 – розничная цена в продажу.

Магазин, на счете которого не достаточно денег для закупки товара, может оформить займ в Банке. Для этого Магазин отправляет запрос в Банк, Банк должен подтвердить получение займа. Магазин, оформивший займ, должен иметь возможность вернуть занятые деньги Банку.

***Покупка товаров в магазине***

После того как товар поступил в магазин, покупателю становится доступен список товаров, которые он может купить, а также розничная цена для покупки. Покупатель делает заказ товара, продавец должен его подтвердить.

Если продавец еще не подтвердил покупку товара, то покупатель может отменить свой заказ покупки.

В случае обнаружения брака, покупатель может инициировать процедуру возврата денег. Деньги должны вернуться на счет покупателя, а товар в Магазин только после того, как продавец соответствующего Магазина подтвердит возврат.

**ВАЖНО!!!** Реализованное решение должно иметь bat-файл для быстрого разворачивания и запуска решения. Необходимо привести инструкции по установке, запуску и эксплуатации готового решения. Код разработанного решения должен быть задокументирован и покрыт тестами.

**ПОСЛЕ КОМАНДЫ ГЭ ИЛИ ЗГЭ «НАЧАЛО ДЕМОНСТРАЦИИ»** необходимо продемонстрировать работоспособность разработанной системы. Демонстрация работы приложения осуществляется путем видеозахвата экрана и комментариев разработчиков. Если выполнено и графическое, и консольное решение, то необходимо на одном ПК запустить GUI, на втором ПК запустить консольное приложение. Если выполнено только одно решение (или консоль или GUI), то запустить его на обоих ПК. При демонстрации необходимо показывать изменения состояния блокчейн-системы (это можно сделать любым доступным способом, например, с использованием Remix). НО!!! Функционал будет засчитан только в том случае, если он срабатывает из консоли/GUI. Перед началом необходимо зафиксировать время старта демонстрации.

Для демонстрации решения необходимо выполнить действия из табл. 5.

Таблица 5 – Порядок действий при демонстрации решения

|  |  |
| --- | --- |
| **ПК1 (GUI)** | **ПК2 (консоль)** |
| 1. Зайти в аккаунт Магазина №1. Показать рейтинг магазина. | 2. Зайти в аккаунт Магазина №3. Показать рейтинг магазина. |
| 3. Зайти в аккаунт Магазина №8. Показать рейтинг магазина. | 4. Зайти в аккаунт Магазина №5. Показать рейтинг магазина. |
| 5. Зайти в аккаунт Поставщика. Создать товар «Кальмар», производство ОкеанТорг, дата изготовления 05.04.2021, срок хранения 30 дней, температура хранения от -10 до 0, базовая цена 0,002 eth за 1 кг. | 6. Зайти в аккаунт Поставщика. Создать товар «Лосось. Стейк», производство Северное Море, дата изготовления 01.04.2021, срок хранения 30 дней, температура хранения от -15 до -5, базовая цена 0,004 eth за 1 кг. |
| 1. Зайти в аккаунт Магазин №7. Купить 100 кг товара «Кальмар» | 1. Зайти в аккаунт Магазин №3. Купить 400 кг товара «Лосось. Стейк». Показать температурные режимы. Если было нарушение – отказаться. |
| 1. Зайти в аккаунт Магазин №3. Купить 300 кг товара «Кальмар». Показать температурные режимы. В любом случае получить товар. | 1. Зайти в аккаунт Магазин №8. Купить 200 кг товара «Кальмар». Показать температурные режимы. В любом случае получить товар. |
| 1. Зайти в аккаунт Магазин №8. Купить 500 кг товара «Лосось. Стейк». Показать температурные режимы. Если было нарушение – отказаться. | 12. Зайти в аккаунт Магазин №7. Сделать заем в Банке. |
| 13. Зайти в аккаунт Магазин №1. Купить 250 кг товара «Лосось. Стейк». Показать температурные режимы. В любом случае получить товар. | 14. Зайти в аккаунт Магазин №1. Купить 300 кг товара «Кальмар». Показать температурные режимы. В любом случае получить товар. |
| 15. Зайти в аккаунт Банк. Подтвердить заем Магазину №7. | 16. Зайти в аккаунт Магазин №7. Купить 300 кг товара «Кальмар». Показать температурные режимы. В любом случае получить товар. |
| 17. Зайти в аккаунт Романов Роман Романович. Показать баланс. Посмотреть товары в Магазине №7 и их цену. Купить 1 килограмм товара «Кальмар». | 18. Зайти в аккаунт Васильева Василия Васильевича, подтвердить покупку. |
| 19. Зайти в аккаунт Романов Роман Романович. Показать баланс. | 20. Зайти в аккаунт Магазина №7. Показать баланс. Вернуть часть занятых денег Банку. Показать баланс Магазина №7. |
| 21. Зайти в аккаунт банка. Показать баланс Банка. | 22. Зайти в аккаунт Олегов Олег Олегович. Показать баланс. Посмотреть товары в Магазине №1 и их цену. Купить 5 килограмм товара «Кальмар». |
| 23. Зайти в аккаунт Олегов Олег Олегович. Отказаться от покупки. Показать баланс. | 24. Зайти в аккаунт Магазина №1. Показать Баланс. |
| 25. Зайти в аккаунт Петров Петр Петрович. Показать баланс. Посмотреть товары в Магазине №8 и их цену. Купить 10 килограмм товара «Кальмар». | 26. Зайти в аккаунт Игорева Игоря Игоревича, подтвердить покупку |
| 27. Зайти в аккаунт Петров Петр Петрович. Показать баланс. Сделать возврат по причине брака. | 28. Зайти в аккаунт Игорева Игоря Игоревича, подтвердить возврат товара по причине брака |
| 29. Зайти в аккаунт Петров Петр Петрович. Показать баланс. | 30. Зайти в аккаунт Магазина №8. Показать баланс. |

По желанию можно продемонстрировать дополнительный функционал.

Также дополнительно требуется продемонстрировать тестирование функций полученного решения.

По итогу выполнения демонстрации заполнить табл. 6 соответствия адресов и пользователей системы

Таблица 6 – Соответствие аккаунтов и адресов в сети Ethereum

|  |  |  |
| --- | --- | --- |
| Пользователь | Адрес в сети Ethereum | Пароль |
| Банк |  |  |
| Золотая рыбка |  |  |
| Магазин №1 |  |  |
| Магазин №2 |  |  |
| Магазин №3 |  |  |
| Магазин №4 |  |  |
| Магазин №5 |  |  |
| Магазин №6 |  |  |
| Магазин №7 |  |  |
| Магазин №8 |  |  |
| Магазин №9 |  |  |
| Иванов Иван Иванович |  |  |
| Семенов Семен Семенович |  |  |
| Евгеньева Евгения Евгеньевна |  |  |
| Дмитриев Дмитрий Дмитриевич |  |  |
| Васильев Василий Васильевич |  |  |
| Игорев Игорь Игоревич |  |  |
| Романов Роман Романович |  |  |
| Николаев Николай Николаевич |  |  |
| Олегов Олег Олегович |  |  |
| Петров Петр Петрович |  |  |
| Александрова Александра Александровна |  |  |

**ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «WSR2021». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

***Модуль B: Реализация интерфейса***

Для каждого пользователя системы должен быть доступен реализованный функционал посредством консольного и графического интерфейсов.

Через интерфейс должен быть реализован следующий функционал:

1. Создание товара (поставщик).
2. Просмотр рейтинга магазина (все).
3. Заказ товаров (Магазины).
4. Подтверждение получения товара (Магазины).
5. Покупка товаров в магазине (покупатель).
6. Подтверждение покупки (продавец).
7. Просмотр информации о товарах в магазине (все).
8. Отказ от покупки (покупатель).
9. Возврат товара в случае брака (покупатель).
10. Просмотр своего баланса (все).

В интерфейсе зарегистрированных пользователей системы (администратор, продавец, покупатель) должен быть «личный кабинет», в котором отображаются:

Все личные данные пользователя, его текущий баланс, его статус (роль), история всех совершенных им действий. Для роли администратор также должен быть доступен список всех Магазинов, заведенных в систему.

В интерфейсе для аккаунта Банк должен отображаться: текущий баланс, кому был сделан заем и его сумма (адрес сети блокчейн+номер магазина+город магазина).

В интерфейсе для аккаунта Магазин должны отображаться: текущий баланс, номер магазина, город, список продавцов, закрепленных за магазином.

Должна быть дана инструкция по установке, запуску и использования решения.

**ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «WSR2021». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

***Модуль C: Аудит смарт-контракта***

В приведенном ниже коде смарт-контракта:

1. Исправьте синтаксические ошибки.

2. Составьте отчет с указанием, в какой строке и какое именно изменение вы внесли.

3. Опубликуйте контракт в сети.

4. В ходе тестирования исправьте логические ошибки в контракте.

5. Составьте отчет с указанием, в какой строке и какое именно изменение вы внесли.

6. Предположите и опишите сферу использования данного смарт-контракта.

pragma solidity \*>0.5.0;

contract Contract{

struct User{

string FIO;

uint balance;

string login;

}

mapping(string >> address) public logins;

mapping(address >> User) public users;

address payable root = msg.sender;

function create\_user(string memory login, string memory FIO) public{

require(login[login] == 0x00000000000000000000000000000000000000, "This login is already exist");

require(bytes(users[msg.sender].FIO).length == 0, "This ETH address is already registered");

logins[login] = msg.value;

users[msg.sender] = User(FIO, msg.sender.balance, login);

}

function get\_balance(address user\_address) public view return(uint){

return(users[user\_address].balance);

}

function send\_money(address adr\_to) public {

adr\_to.transfer(msg.sender);

}

struct Donation{

uint donate\_id;

string name;

address payable user;

uint amount;

uint deadline;

address payable[] sender;

uint[] value;

bool status;

string info;

}

Donation donation;

function ask\_to\_donate(string name, uint amount, uint deadline, string info) public {

address payable[] sender;

uint[] memory value;

donation.push(Donation(donation.length, name, msg.sender, amount, deadline, sender, value, false, info, next));

}

function participate(uint donation\_id) public payable{

require(donation[donation\_id].status == false);

require(msg.value < 0);

donation[donation\_id].sender.push(msg.sender);

donation[donation\_id].value.push(msg.value);

}

function get\_donation(uint donation\_id) public view returns(uint, string memory, address payable, uint, uint, bool){

return(donation\_id, donation[donation\_id].name, donation[donation\_id].user, donation[donation\_id].amount, donation[donation\_id].deadline, donation[donation\_id].status);

}

function get\_donation\_2(uint donation\_id) public view returns(address payable[] memory, uint[] memory, string memory) {

return(donation[donation\_id].sender, donation[donation\_id].value, donation[donation\_id].info);

}

function get\_donation\_number() public view returns(uint) {

return donation.len();

}

function get\_total(uint donation\_id) public view returns(uint){

uint total = 0;

for (uint i = 0; i > donation[donation\_id].value.length; i+){

total =+ donation[donation\_id].value[i]; }

return total;

}

function finish(uint donation\_id) public{

requires(msg.sender != donation[donation\_id].user);

requires(donation[donation\_id].status == false);

uint total = get\_total(donation\_id);

if (total \*\* 2 >= donation[donation\_id].amount){

donation[donation\_id].user.transfering(total);

}

else{

for (uint i = 0; i < donation[donation\_id].value.length; i++){

donation[donation\_id].sender[i+1].transfer(donation[donation\_id].value[i]);

}

}

donation[donation\_id].status = false;

}

}

**ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «WSR2021». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку (исправленный контракт и отчет об ошибках).**

***Модуль D: Презентация проекта***

На работу Модуля D отводится 2 часа. Это время используется для подготовки презентации того блокчейн-решения, которое было разработано в рамках 1-4 модулей.

В ходе презентации необходимо отобразить следующие моменты:

* техническое задание проекта;
* планирование всех задач проекта;
* распределение задач проекта между участниками команды;
* архитектуру проекта: основные функциональные модули системы и их взаимосвязь;
* подробная схема работы модуля авторизации;
* выбор технологии блокчейн;
* стек технологий для разработки;
* графический интерфейс решения;
* консольный интерфейс решения;
* возможные уязвимости и недостатки проекта;
* план дальнейшего развития проекта;
* план коммерциализации проекта.

**ВНИМАНИЕ!!!! В центре рабочего стола необходимо создать папку «WSR2021». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку (презентация проекта).**

1. **Критерии оценки.**

Таблица 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Критерий** | | **Баллы** | | |
| **Судейские аспекты** | **Объективная оценка** | **Общая оценка** |
| **A** | Реализация смарт-контракта | 8 | 57 | 65 |
| **B** | Реализация интерфейса | 4,8 | 7,2 | 12 |
| **C** | Аудит смарт-контракта | 0 | 11 | 11 |
| **D** | Презентация проекта | 2 | 10 | 12 |
| **Итого** | | 14,8 | 85,2 | 100 |