

**VII Открытый региональный чемпионат Ростовской области**

*24-28 февраля 2022 г*

**Конкурсное задание**

**по 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
2. Общее время на выполнение задания:	2
3. Задание для конкурса	2
4. Модули задания и необходимое время	2
5. Критерии оценки.	23
6. Приложения к заданию.	

**ERROR! BOOKMARK NOT DEFINED.**

## 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.

## 2. Общее время на выполнение задания: 18 ч.

## 3. Задание для конкурса

Целью конкурсного задания является создание и презентация программного решения с использованием технологии блокчейн.

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.

## 4. Модули задания и необходимое время

Таблица 1.

	Наименование модуля	Соревновательный день (С1, С2, С3)	Время на задание
<b>A</b>	Разработка на платформе Ethereum	С1	7
<b>B</b>	Разработка на платформе HyperLedger	С2	7
<b>C</b>	Аудит смарт-контракта	С3	2
<b>D</b>	Презентация проекта	С3	2

## Модуль А Разработка на платформе Ethereum

### Реализация смарт-контракта

В рамках Модуля А необходимо:

1. Выполнить инициализацию блокчейн-системы Ethereum 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 Модуля А 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 Петров Петр Петрович (логин petr, на счету 80 eth).

Остальные пользователи системы должны иметь возможность регистрироваться в системе самостоятельно.

**ВАЖНО!!!** Реализованное решение должно иметь исполняемый файл (.bat или .sh) 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.

**ВАЖНО!!!** Необходимо привести достаточные пояснения к работе всех функций смарт-контракта, описание входных параметров и возвращаемых значений.

**ВАЖНО!!!** При выполнении Модуля А у участников ограничен доступ к сети Интернет

**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 на этапе демонстрации 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.	2. Зайти в аккаунт Семенов Семен Семенович в роли продавец, показать личный кабинет.
3. Переключиться из роли Иванова Ивана Иванович администратор к роли покупатель, показать личный кабинет (доступный функционал)	4. Переключиться из роли Семенов Семен Семенович продавец к роли покупатель, показать личный кабинет (доступный функционал)
5. Переключиться из роли Иванова Ивана Иванович покупатель к роли продавец, показать личный кабинет (доступный функционал)	6. Переключиться из роли Семенов Семен Семенович покупатель к роли продавец, показать личный кабинет (доступный функционал)
7. 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.	8. Зайти в аккаунт Магазина №10. Показать счет. Отправить в банк запрос на получение денег (1000 eth).
9. Зайти в аккаунт Банк. Показать счет. Подтвердить получение Магазином №10 займа в 1000 eth. Показать счет Банка.	10. Показать счет в Магазине №10.
11. Зарегистрировать нового покупателя Сергеева Сергея Сергеевича (логин serg). Подать от него запрос стать продавцом магазина №10	12. Зайти в аккаунт Администратора. Подтвердить запрос Сергеева Сергея Сергеевича.
13. Показать статус Сергеева Сергея Сергеевича. Переключиться к роли продавец.	14. Удалить из списка магазинов Магазин №10. Показать список магазинов.
15. 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	16. 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».

для Магазина №1 «Рыба не свежая!». Оставить оценку магазину 1 из 10.	
17. Зайти в кабинет Петрова Петра Петровича. Подтвердить запись «Все супер!» и опровергнуть запись «Рыба не свежая!».	18. Зайти в аккаунт Семенова Семена Семеновича. Подать запрос на понижение роли до уровня «покупатель».
19. Зайти в аккаунт Администратора. Подтвердить запрос Семенова Семена Семеновича.	20. Показать личный кабинет Семенова Семена Семеновича, попробовать переключиться к роли продавец.
22. Зайти в аккаунт Администратора. Подтвердить запрос Владимиров Владимир Владимировича.	21. Зарегистрировать нового покупателя Владимиров Владимир Владимирович (логин vlad). Подать от него запрос стать продавцом магазина №4.
23. Зайти в аккаунт Владимиров Владимира Владимировича. Показать личный кабинет. Переключиться к роли продавец. Переключиться к роли покупатель.	24. Зайти в аккаунт Петрова Петра Петровича. Подать запрос на повышение до роли администратора.
25. Зайти в аккаунт Администратора. Подтвердить запрос Петрова Петра Петровича.	26. Зайти в аккаунт Петрова Петра Петровича. Переключиться к роли администратор. Переключиться к роли покупатель.
27. Зайти в аккаунт Поставщика. Показать баланс.	28. 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 – Day1». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

### 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 Day1». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

## Модуль В:

### Реализация смарт-контракта

В рамках Модуля В необходимо:

1. Выполнить инициализацию тестовой сети платформы HyperLedger Fabric.
2. Разработать смарт-контракт на одном из поддерживаемых платформой языков.
3. Создать в сети организации с названиями Shops, Bank, Users
4. Создать в сети канал WSR2021 для размещения в нем chaincode разрабатываемого смарт-контракта.
5. Реализовать функцию регистрации новых пользователей системы
6. Реализовать функцию двухфакторной авторизации пользователей системы
7. Реализовать функцию смены ролей пользователей
8. Реализовать функцию по работе с «Книгой жалоб и предложений магазина»
9. 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 Модуля В 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-сети, который относится к организации Bank. При старте системы на счету данного аккаунта должно быть 10000 монет CoinWSR. Банк дает заем новому магазину в виде стартового капитала в 1000 монет CoinWSR.

**Магазин** в рамках задания – это аккаунт в блокчейн-сети, который принадлежит организации Shops, аккаунт создается в системе при ее запуске либо вводится в систему в ходе ее использования путем регистрации новых аккаунтов. Всего в систему при старте должно быть заведено 9 магазинов согласно табл. 4.

**Поставщик** в рамках задания – это аккаунт в блокчейн сети, который создается при запуске сети и принадлежит организации Shops.

Таблица 4 – Список магазинов

№ магазина	город	Баланс, CoinWSR
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). Принадлежит организации Users.

### *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. Принадлежит организации Users.

## ***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. Принадлежит организации Users.

## ***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. Пользователь регистрируется в организации Users.

## ***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 - Принадлежат организации Shops.
- ✓ Банк (баланс 10000 монет CoinWSR) Принадлежат организации Bank.
- ✓ Поставщик Золотая рыбка (логин goldfish, на счету 100 монет CoinWSR).  
Принадлежат организации Shops.
- ✓ 1 Администратор системы Иванов Иван Иванович (логин ivan, на счету 50 монет CoinWSR). Принадлежат организации Users.
- ✓ Продавец Семенов Семен Семенович (логин semen, магазин №1, на счету 70 монет CoinWSR). Принадлежат организации Users.
- ✓ Покупатель Петров Петр Петрович (логин petr, на счету 80 монет CoinWSR). Принадлежат организации Users.

Остальные пользователи системы должны иметь возможность регистрироваться в системе самостоятельно. При регистрации они закрепляются за организацией Users.

**ВАЖНО!!!** Реализованное решение должно иметь исполняемый файл (.bat или .sh) 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). **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 Иванова Ивана Ивановича в роли администратор, показать личный кабинет.	2. Зайти в аккаунт Семенов Семен Семенович в роли продавец, показать личный кабинет.
3. Переключиться из роли Иванова Ивана Иванович администратор к роли покупатель, показать личный кабинет (доступный функционал)	4. Переключиться из роли Семенов Семен Семенович продавец к роли покупатель, показать личный кабинет (доступный функционал)
5. Переключиться из роли Иванова Ивана Иванович покупатель к роли продавец, показать личный кабинет (доступный функционал)	6. Переключиться из роли Семенов Семен Семенович покупатель к роли продавец, показать личный кабинет (доступный функционал)
7. 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 монет CoinWSR). Показать список магазинов, доступных для управления администратору.	8. Зайти в аккаунт Магазина №10. Показать счет. Отправить в банк запрос на получение денег (1000 монет).
9. Зайти в аккаунт Банк. Показать счет. Подтвердить получение Магазином №10 займа в 1000 монет. Показать счет Банка.	10. Показать счет в Магазине №10.
11. Зарегистрировать нового покупателя Сергеева Сергея Сергеевича (логин serg). Подать от него запрос стать продавцом магазина №10	12. Зайти в аккаунт Администратора. Подтвердить запрос Сергеева Сергея Сергеевича.
13. Показать статус Сергеева Сергея Сергеевича. Переключиться к роли продавец.	14. Удалить из списка магазинов Магазин №10. Показать список магазинов.
15. 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	16. 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».

для Магазина №1 «Рыба не свежая!». Оставить оценку магазину 1 из 10.	
17. Зайти в кабинет Петрова Петра Петровича. Подтвердить запись «Все супер!» и опровергнуть запись «Рыба не свежая!».	18. Зайти в аккаунт Семенова Семена Семеновича. Подать запрос на понижение роли до уровня «покупатель».
19. Зайти в аккаунт Администратора. Подтвердить запрос Семенова Семена Семеновича.	20. Показать личный кабинет Семенова Семена Семеновича, попробовать переключиться к роли продавец.
22. Зайти в аккаунт Администратора. Подтвердить запрос Владимиров Владимир Владимировича.	21. Зарегистрировать нового покупателя Владимиров Владимир Владимирович (логин vlad). Подать от него запрос стать продавцом магазина №4.
23. Зайти в аккаунт Владимиров Владимира Владимировича. Показать личный кабинет. Переключиться к роли продавец. Переключиться к роли покупатель.	24. Зайти в аккаунт Петрова Петра Петровича. Подать запрос на повышение до роли администратора.
25. Зайти в аккаунт Администратора. Подтвердить запрос Петрова Петра Петровича.	26. Зайти в аккаунт Петрова Петра Петровича. Переключиться к роли администратор. Переключиться к роли покупатель.
27. Зайти в аккаунт Поставщика. Показать баланс.	28. 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. 6 соответствия адресов и пользователей системы

Таблица 6 – Соответствие аккаунтов и адресов в сети Ethereum

Пользователь	Адрес в сети HypedLedger	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 Day2». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

### 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 Day 2». Все файлы, созданные в результате выполнения решения, должны быть помещены в эту папку.**

## Модуль С: 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] == 0x000000000000000000000000000000000000, "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.transferring(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;

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).

## 5. Критерии оценки.

Таблица 2

Критерий		Баллы		
		Судейские аспекты	Объективная оценка	Общая оценка
<b>A</b>	Разработка на платформе Ethereum	9.5	29	38.5
<b>B</b>	Разработка на платформе HyperLedger	9.5	29	38.5
<b>C</b>	Аудит смарт-контракта	0	11	11
<b>D</b>	Презентация проекта	2	10	12
<b>Итого</b>		14,8	85,2	100