**Оптовый склад**

**Var 6.**

1. Для решения этой задачи будем использовать блокчейн-решение Telegram open network (Ton). Это децентрализованный блокчейн первого уровня со сверхбыстрыми транзакциями, с небольшими комиссиями, простотой в использовании и хорошо защищенный от мошенников.

TON обладает высокой скоростью обработки запроса, сопоставимой со скоростью работы платежных систем Visa и Mastercard. Для оптового склада это очень важно, ведь информация об покупателях, товарах на складе и сделки и продаже должны обновляться почти мгновенно. Не должно быть так, что покупатели берут товар, который уже отправили заказчику.

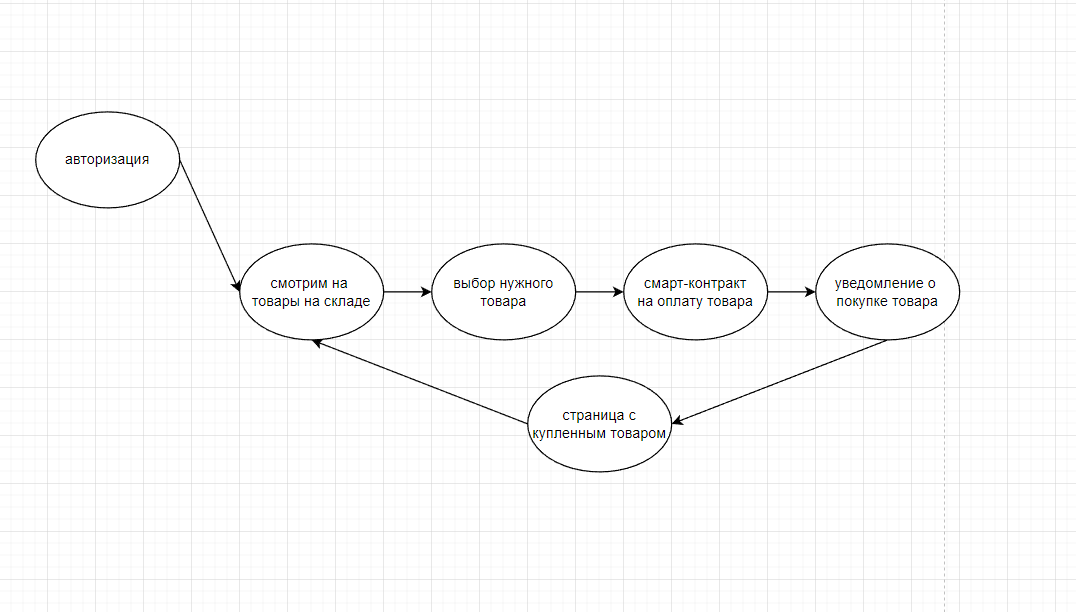
Большие покупки будут приводит к сложным транзакциям и сеть TON будет успешно справляться с ними. С поддержкой шардинга (поддерживает до 2^60 осколков на цепочке), сеть также будет загружаться меньше.

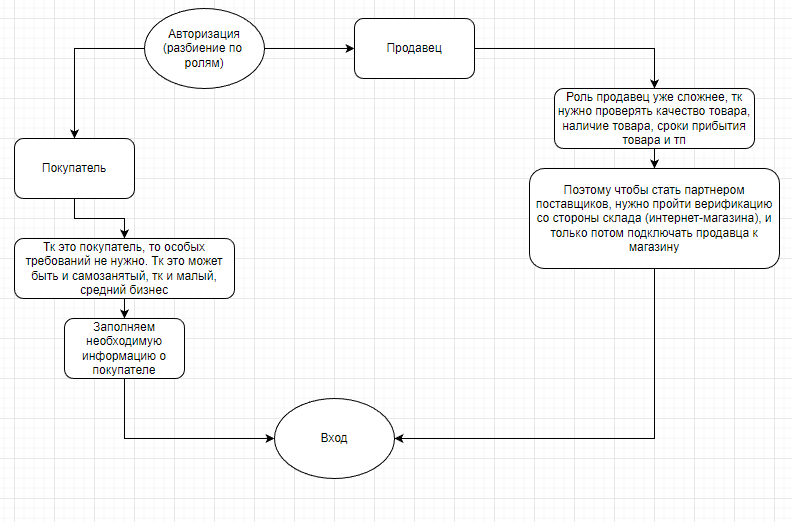
2. Первым делом написать смарт-контракт. Смарт-контракт в сети The Open Network пишется на высокоуровневом языке FunC. Программы FunC компилируются в Fift ассемблерный код, который генерирует соответствующий байт-код для виртуальной машины TON(TVM). Затем будет разработан API на Python с использованием фрейморка Flask.

Далее к API будет написан web-интерфейс, реализующий данный функционал.

Проверка системы на стабильность.

3.





4. Функции:

Структуры данных смарт контракта:

1 Поставщик и Покупатель будут иметь схожие поля. Пример:

struct User {

address adrProv;

uint id;

string nameFirmSell;

string addressFirmSell;

string phoneNumberSell;

}

1. Те адресс, куда будут поступать токены за покупку, продажу товаров;  
   уникальный номер;
2. Имя, адрес, телефонный номер фирмы.

2 Структура товара

struct Good{

uint id;

string nameFirm;

string nameGoods;

string units;

uint amount;

uint valueBuy;

uint valueSell;

}

Содержит:

1. Уникальный номер, по которому будет проводится закупка и продажа;  
   Название товара;
2. Наименование фирмы поставщика;
3. Поля для просмотра количества товара на складе, цены товара на продажу (покупателю не будет видна цена закупки) и валюта, в которой идет покупка.

3 Отчет о сделках

struct makeDeal{

string idBuyer;

string idSeller;

uint id;

uint amountSellGoods;

}

Содержит:

1. Id покупателя и фирмы продавца;
2. Id покупаемого товара
3. Сумма сделки

Функции смарт контракта.

* Добавление поставщика и покупателя
* Просмотр, оплата товаров (создание транзакции по обмену коинов)
* Функция «пнуть» поставщика, если товар нужен (те предупредить, что хотелось бы приобрести данный товар)
* Создание смарт-контракта на оплату товаров (Создание предусматривает:
  + Наличие товара
  + Наличие достаточного баланса
  + Возможность доставки товара

Только после этих требований создается смарт-контракт)

* Просмотр уже купленных товаров.
* Перевод оплаты с одно счета на другой

5. Дизайн – Figma. Это графический редактор, который подходит командам для совместной работы онлайн. Среди главных возможностей: создание прототипов, интерфейсов, внесение правок в режиме реального времени, обсуждение дизайна с командой в комментариях.

Frontend –html, CSS. Нестареющая классика. Специалистов в этой области довольно много, значит не будет проблемой в поиске специалиста.

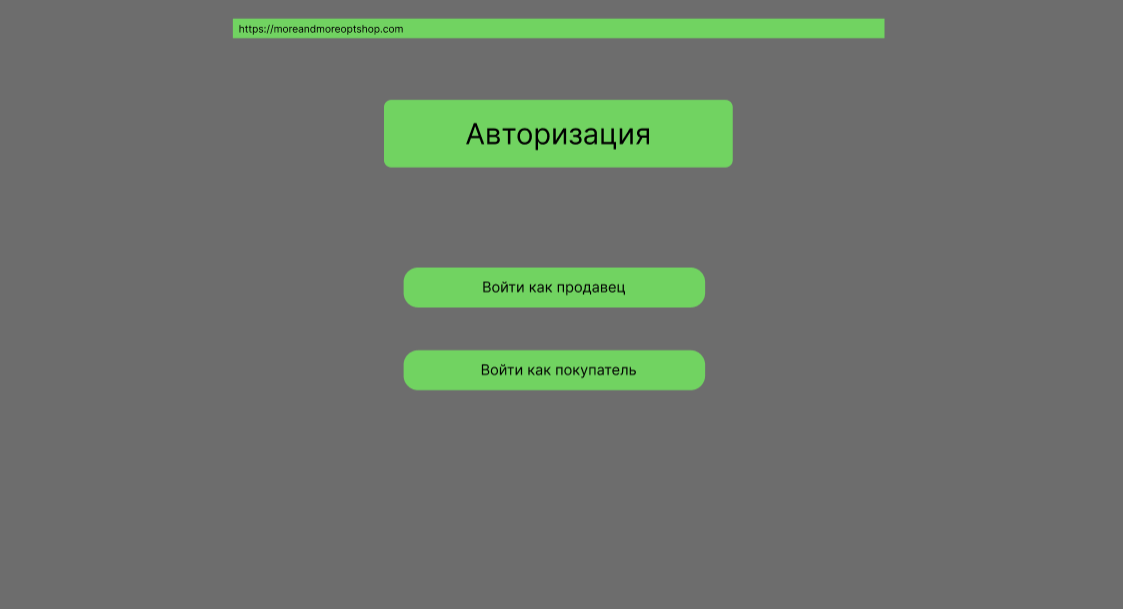
Backend – Node.js + TypeScript. Эта платформа позволяет писать серверный код для динамических веб-страниц и веб-приложений, а также для программ командной строки. С помощью Node.js реализуется парадигма «TypeScript для всего».

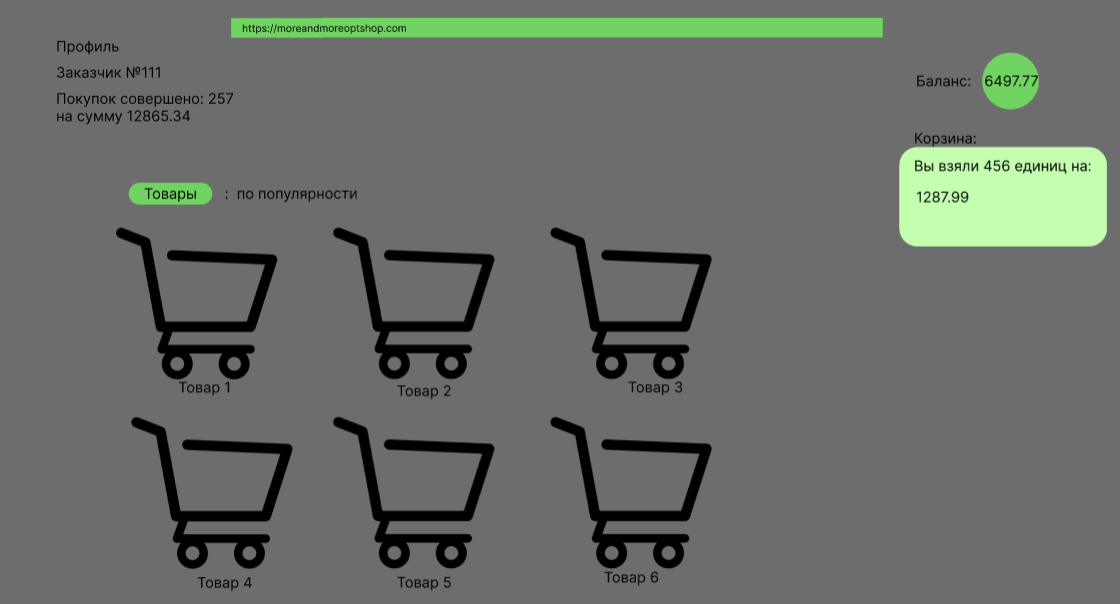
Смарт – контракт – FunC (специально разработанный язык для смарт-контрактов на блокчейне Telegram Open Network).

6. Команда: Frontentd-разработчик, дизайнер, блокчейн-разработчик, тестировщик, адвокат для компании.

* Дизайнер – 15ч (Для создания макетов сайта)
* Frontentd-разработчик – 30ч (Для создания интерфейса сайта и проверки работоспособности и поддержания сайта)
* Backend-разработчик – 40ч (Для создания функциональной части сайта и его поддержания)
* Блокчейн-разработчик – 30ч (Для написания смарт-контракта и проверки его работоспособности)
* Тестировщик – 10ч (Для проверки работоспособности всего кода).

7.





https://www.figma.com/file/HF09i5pnaJAH1IJvcrx7RC/Untitled?node-id=0%3A1&t=H429qsPxOAJDAJTd-1