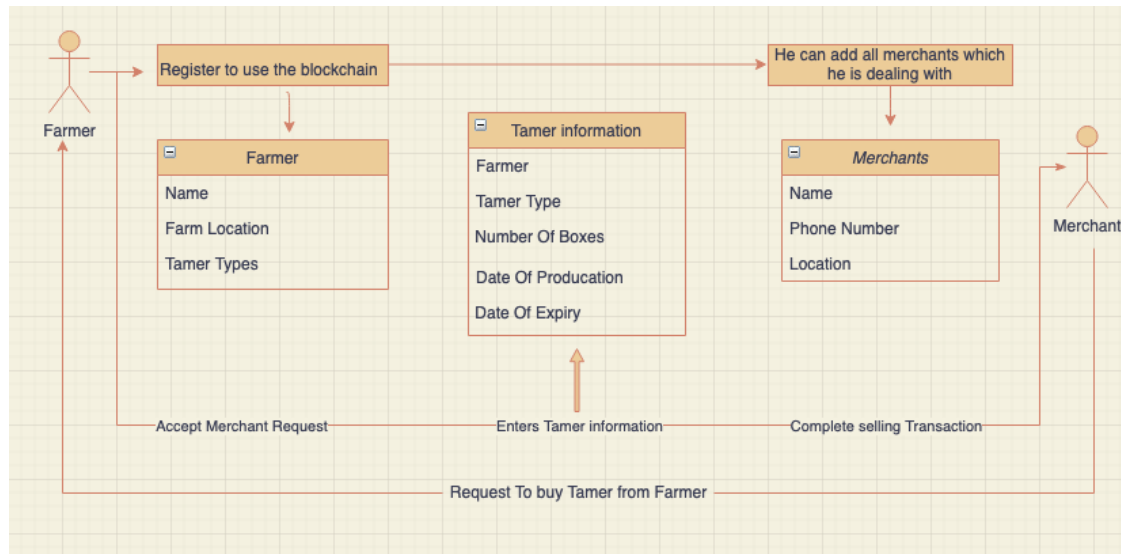
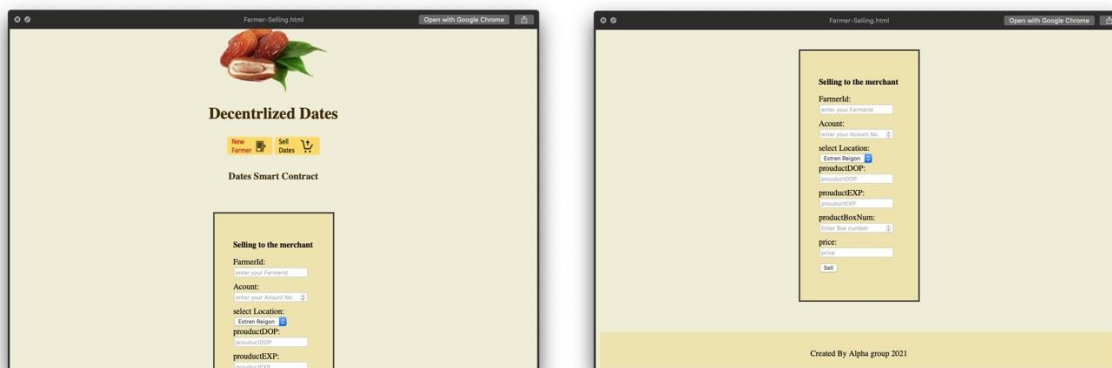


## Decentralized Dates (Tamer)



In this project we are creating a decentralized system for Farmers to enter the Dates (Tamer) information into a Blockchain Individually, give only them the authorization to enter and edit the Tamer metadata, sell the Dates to the merchant through the system and finally develop the system in the future to consumers so they can check all Tamer information using a par code we are going to implement in the Tamer packaging before we send it to the merchant for selling.

## Frontend Of The Smart Contract



The farmer can use the website to complete the selling process between him and the merchant by adding all the dates information before completing the transaction.

## Prerequisites To Use the System

Both Merchant and Farmer needs the listed below:

- Have an internet connection.
- Matamask installed in Firefox, Chrome, Brave or edge browser.
- Creating an account in Matamask to generate new Ethereum wallet address.

Once they have everything settled, they can use the frontend website we created to ease the processing and their transaction will be complete through the Matamask wallet.

## How To Built It?

- We wrote the smart contract using solidity language in <https://remix.ethereum.org/>.
- To create the frontend HTML pages for the smart contract we used Visual Studio Code.
- Use Ganache as Hotspot to create the test connection with Matamask by using privet key.
- Finally use Truffle to test your smart contract.

## Requirements To The Above

- Visual Studio Code installed on your device.
- To install Ganache go to <https://nodejs.org/en/> and after that you will need to install Git from <https://gitforwindows.org> to get Ganache working on your System.
- For testing your smart contract using Truffle you have to install NodeJS v8.9.4 or later first, then you open your terminal and type

```
npm install -g truffle
```

## Running The Test

- 1- In Visual studio install the expiation for solidity.
- 2- Go the Terminal and write `Truffle init`
- 3- you will have files to your left for your contract, you need to create new file and name it the exact name of your contract and save it as .sol file.
- 4- In the migration file there is a file called initial\_migration “ copy the content and create new file and name it example.js in Java Script”
- 5- In the file change every migration word to the name of your contract
- 6- Go the Terminal and write `Truffle develop`
- 7- Then `migrate`

## **Authors**

Ammar Al-Raddadi, Abdulrhman Al-Amri, Ghadeer Almousa, Munirah Al-Saeed and Abeer Bin Merdah.

## **Acknowledgments**

Anwar Alruwaili

Arry Yu

Adam Norris