## General

The project was developed using the Truffle Develop console. After completing the features and passing all the tests, the contract was deployed to the Ethereum Rinkeby network using Infura.

The project implemented the optional feature that allows a Farmer to upload a picture to IPFS and the picture’s hash is saved with the transaction. The picture is uploaded to Infura’s IPFS endpoint.

Contract Address: 0x3F5c68d9BAb59BE4012bEee4AeA13D7577F6AA3d

Transaction: 0xaa455f71be8dcc9f75f7a39714f544ce5bc3f9bb4ef1fe498710f5cb5a2dfd9d

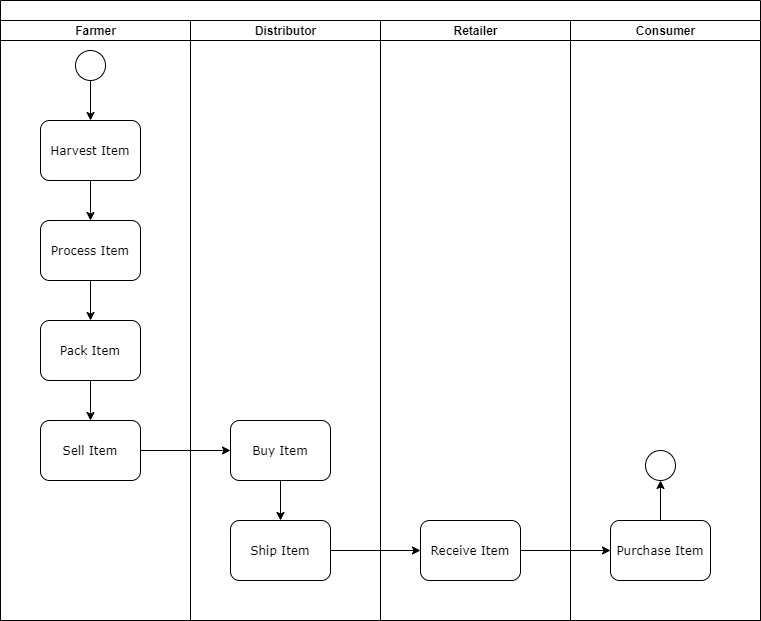
Versions:

Node v12.18.2

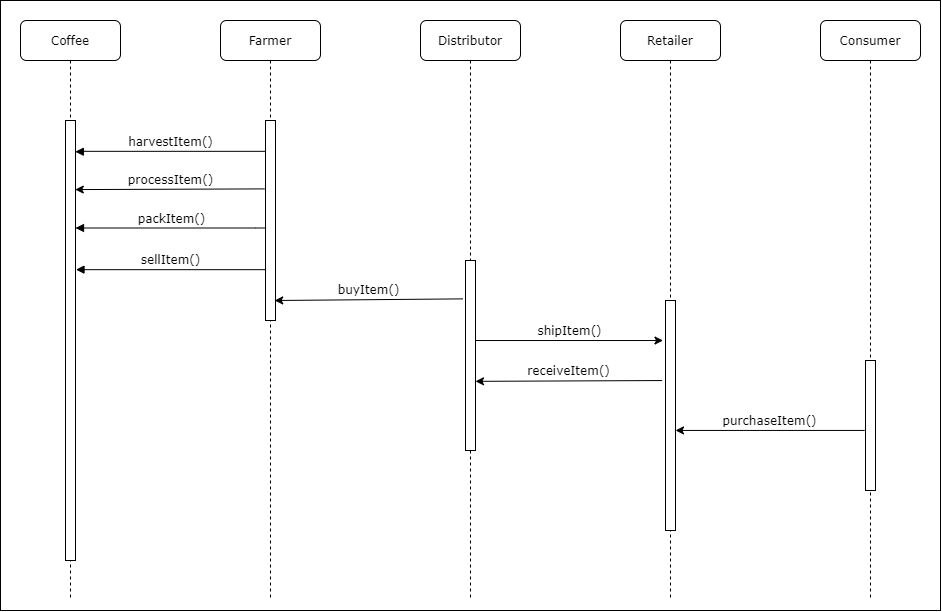
Truffle v5.1.57

Web3

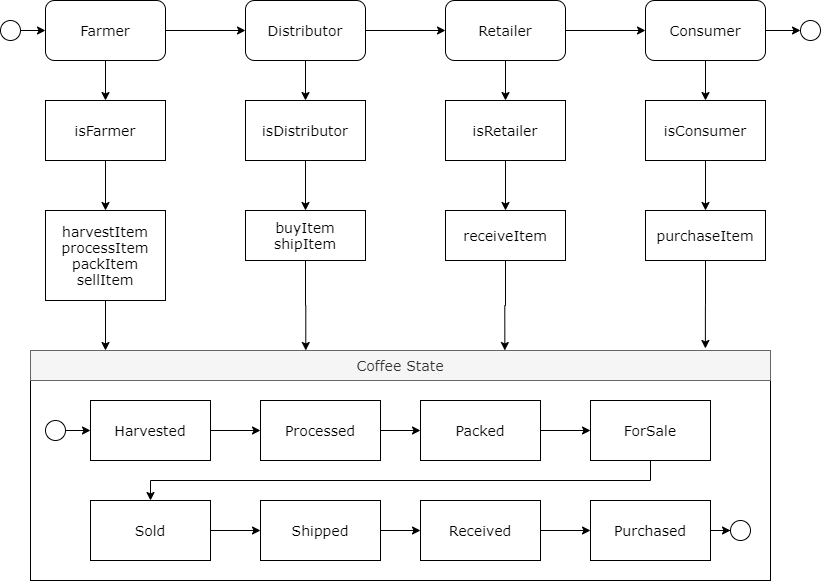
## Diagram 1: Activity Diagram



## Diagram 2: Sequence Diagram



## Diagram 3: State Diagram



## Diagram 4: Data Model

## Libraries

|  |  |
| --- | --- |
| Library | Reason |
| lite-server | This was used for a developing the frontend. It creates a lightweight node server that refreshes the web app whenever the html or javascript changed |
| truffle | Development framework to develop, test and deploy Ethereum dApp |
| truffle-hdwallet-provider | Used with truffle. It is used to sign transactions for the connected wallet. |
| jquery | A DOM manipulation library that made updating HTML easier. |
| web3 | Provides an API that allows program to interact with Ethereum nodes using HTTP. Need this to connect frontend application to Ethereum network. |
| infura | Provides a way to deploy Ethereum app to network and provides an api to upload images to IPFS |

## IPFS

IPFS was used in this project. Whenever the user adds a file to the Product Picture input, the file is uploaded to Infura’s IPFS endpoint. If the upload is successful, the request returns the file hash. This hash is saved with the transaction. The has can later be used to load the related file.