Decentralized Stock Exchange Software Requirements Specification

### Business Background and problem statement.

To buy or sell assets like stocks, debt, and commodities, you need a way to keep track of who owns what. Financial markets today accomplish this through a complex chain of brokers, exchanges, central security depositories, clearinghouses, and custodian banks. These different parties have been built around an outdated system of paper ownership that is not only slow, but can be inaccurate and prone to deception.

In order to manage this complexity, Buyers and sellers outsource the shares to custodian banks for safekeeping. Because buyers and sellers don’t always rely on the same custodian banks, the custodians themselves need to rely on a trusted third party to hold onto all the paper certificates.

### Technical Specification.

The project team will design and implement a simple order book that will match buyers and sellers and execute the order on the blockchain via a series of smart contracts. The developers will tokenize a list of stocks using a Non Fungible Token, register buyers and sellers, and match orders between buyers and sellers. The developers should implement at least two order types:

1. Market Order

2. Limit Order.

The application should use the blockchain to register and validate all trades between the buyer and the seller.

### Functional Requirements.

|  |  |
| --- | --- |
| **Functional ID** | **Functional Requirement** |
| MFR-1 | Implement an order book that uses the blockchain and smart contracts to register and validate orders between buyers and sellers of equities. |
| MFR-1.1 | Create three roles, Buyer, Seller and Administrator. Only administrators can register new buyers and sellers. Note that entities can be both buyers and sellers simultaneously. |
| MFR-1.2 | Administrators must ensure that buyers have enough ether in their account to cover the purchase order. |
| MFR-1.3 | For limit orders, the project should implement at least the Buy Limit order type. |
| MFR-1.4 | Equities should be tokenized on the blockchain using the ERC-20 or ERC-223 protocol. |
| MFR-2 | Create a simple front end using JavaScript and the web3.js library to send order information and receive confirmations of placed orders. |