**Virtual Exchange**

Communication Protocols

**1 Overview**

The goal of this project is to build a virtual stock exchange (and facilitating components) that allows players to participate in this exchange of ownership of companies. This virtual exchange will attempt to simulate an actual trading environment and allow participants to practice making markets by submitting buy-sell orders based on the actions of other participants. There are three key components to the system: Players, Gateway (implementation of the exchange), and Matching Engines (implementation of Companies).

Players will communicate the Gateway by registering, submitting Buy/Sell orders, receiving confirmation messages, and receiving information updates on companies.

The Gateway will be the mediator between Players and Matching Engines. The Gateway will receive orders and determine if they can be fulfilled, and forwarding orders to the appropriate Matching Engine . The Gateway will also be asking for Book updates from the Matching Engines and sending those to each player, so players can know the current Bid/Ask prices.

The Matching Engines will communicate with the Gateway by registering as a company and by supplying order confirmations, current prices and top of book information.

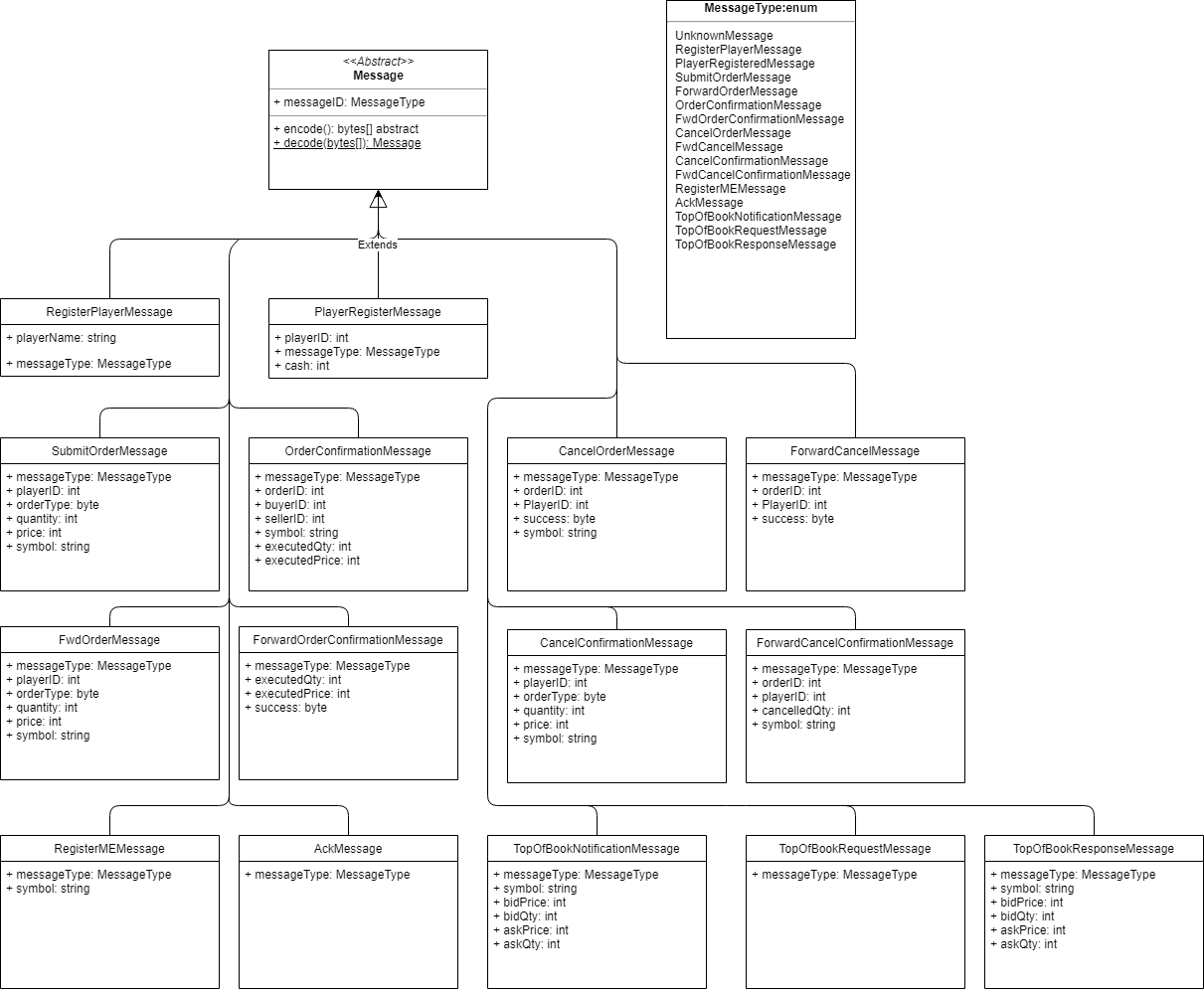
|  |  |  |  |
| --- | --- | --- | --- |
| **Purpose** | **Initiator** | **Other Processes** | **Pattern** |
| **Player Registration** | **Player** | **Gateway** | **Proxy** |
| **Order Submission** | **Player** | **Gateway, Matching Engine** | **Proxy** |
| **Order Cancellation** | **Player** | **Gateway, Matching Engine** | **Proxy** |
| **Matching Engine Registration** | **Matching Engine** | **Gateway** | **Request-Reply** |
| **Top of Book Notification** | **Gateway** | **Player** | **Publish-Subscribe** |
| **Top of Book Caching** | **Gateway** | **Matching Engine** | **Request-Reply** |

**2 Messages and Shared Objects**

**Shared Objects:**

* **Order Book** - A component of each matching engine which contains a list of buy and sell orders at given price levels
* **Matching Engine Registry** - Maintains a list of available matching engine IP addresses for each symbol (a company’s identifier)
* **Player Details** - Maintains a list of how much cash and stock a player holds as well as their last known IP address

**Messages:**

****

**Register Player:**

1. **MessageType=1 short**
2. **Player Name string**

**Player Registered:**

1. **MessageType=2 short**
2. **PlayerID short**
3. **InitialCash integer**

**Submit Order:**

1. **MessageType=3 short**
2. **PlayerID short**
3. **Type byte (1 for buy, 0 for sell)**
4. **Quantity short**
5. **Price int**
6. **Symbol string**

**Forward Order:**

1. **MessageType=4 short**
2. **PlayerID short**
3. **Type byte (1 for buy, 0 for sell)**
4. **Quantity short**
5. **Price int**
6. **Symbol string**

**Order Confirmation:**

1. **MessageType=5 short**
2. **BuyerPlayerID short**
3. **SellerPlayerID short**
4. **OrderID short**
5. **ExecutedQty short**
6. **RestingQty short**
7. **Price int**
8. **Symbol string**

**Forward Order Confirmation**

1. **MessageType=6 short**
2. **OrderID short**
3. **Type short (1 for buy, 0 for sell)**
4. **Symbol string**
5. **ExecutedQty short**
6. **RestingQty short**
7. **Price int**
8. **Status byte (0 means success)**

**Cancel Order:**

1. **MessageType=7 short**
2. **OrderID short**
3. **PlayerID short**
4. **Symbol string**

**Forward Cancel**

1. **MessageType=8 short**
2. **PlayerID short**
3. **OrderID short**
4. **Symbol string**

**Cancel Confirmation**

1. **MessageType=9 short**
2. **PlayerID short**
3. **OrderID short**
4. **CancelledQty short**
5. **Symbol string**

**Forward Cancel Confirmation:**

1. **MessageType=10 short**
2. **OrderID short**
3. **CancelledQty short**
4. **Symbol string**

**Register Matching Engine:**

1. **MessageType=11 short**
2. **Symbol string**

**Ack:**

1. **MessageType=12 short**

**Top of Book Notification:**

1. **MessageType=13 short**
2. **Symbol string**
3. **BidPrice int**
4. **BidQty short**
5. **AskPrice int**
6. **AskQty short**

**Top of Book Request:**

1. **MessageType=14 short**

**Top of Book Response:**

1. **MessageType=15 short**
2. **Symbol string**
3. **BidPrice int**
4. **BidQty short**
5. **AskPrice int**
6. **AskQty short**

**3 Communication Patterns**

* **Proxy:** A proxy process handles all communication between A and B, hiding the details of B and allowing for certain functionality, like access controls to be implemented independent of B.
* **Request-Reply:** Simple form of communication in which process A makes a request to process B, waits, and process B generates and sends a response to process A.
* **Publish-Subscribe:** Subscribers subscribe to receive messages from some process A. Process A can then notify all subscribers with messages. This pattern does not require that the source of the messages know all of the subscribers.

**4 Communication Protocols**

**Player Registration Sequence Diagram:**

****

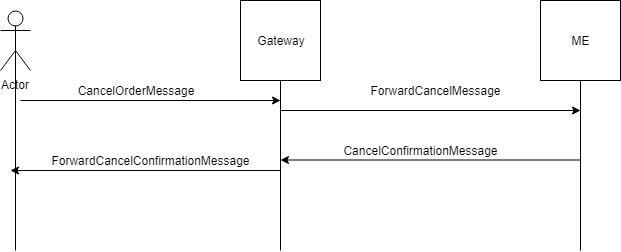
**End Result:** Player is registered with the Gateway, is assigned starting capital, and may begin trading.

**Order Submission Sequence Diagram:**

****

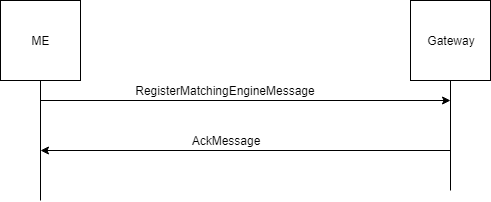
**End result:** Orders that player submitted are either bought, sold or the put onto the book. The player receives this information in a message detailing how many orders went through and how may orders are resting on the book. In the event that the player does not have enough money for the order that he placed, he will only receive an error message telling him that he has insufficient funds.

**Order Cancellation Sequence Diagram:**

****

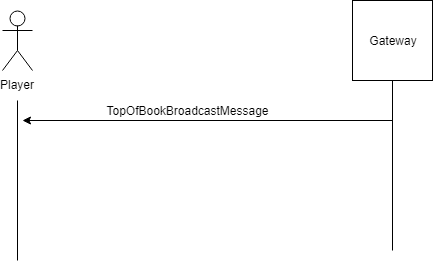
**End Result:** A Player’s specified bid or offer is cancelled.The Cancel Confirmation message informs the user how much of their order was cancelled. If the order has been partially filled, then whatever is still resting on the book is cancelled. If the order has been fully filled, then the cancelled quantity is 0.

**Matching Engine Registration Sequence diagram**

****

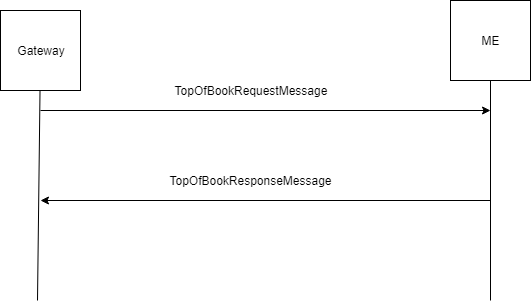
**End result:** A Matching engine is added to the game and the Gateway stores all of the information needed to run the matching engine. The Matching engine will be notified that it has been added and it will now be a part of the game.

**Top of Book Notification Sequence Diagram**

****

**End Result:** The player gets the information about the top of each matching engine book from the Gateway.

**Top of Book Caching Sequence Diagram**

****

**End result:** The gateway gets an updated value for the top of book from every matching engine that it is managing.