**XYZ Trading platform**

**Description**: - This is a low-level design documentation for the Trading platform at XYX Company. This is web-based application mainly used by the stock traders at the XYZ company. This application will have followed four core modules. In phase -2 we will be going to add some additional functionality.

1. **Stock Price: -** The user can add the top 10 stock quotes to their market watch list. The user should get real-time stock updates for the stocks on their watchlist. This application will pull the data from Bloomberg or Nasdaq in real time. Data may be for stock or funds. Stock price JSON will be in the following format.
2. **Computation: -** This module will be responsible for calculating the available balance and other validations. Available balance will be retrieved from the LifeCAD in real time.
3. **Reporting: -** User can generate the reports about their trade history. Monthly trade, Weekly trade, and yearly trade performance.
4. **Booking: -** This module allows users to buy and sell stocks. When a user purchases something, we should check the available balance in the LC in real time. This module will invoke the trading company API to place Buy or Sale order.

Sample Payload: -

{

**StockSymbol**: "MSFT",

**StockCompanyName**: "Microsoft Corporation",

**Open**: "271.02",

**High**: "272.5",

**Low**: "268.9",

**Close**: "270.02",

**Volume**: "28058958",

**DateTime**: "06-06-2022"

}

**System Design**

**Diagram

Description automatically generated**

Sequence Diagram

Diagram

Description automatically generated

This application contains three parts. External systems, Trading platform API’s Service, PWA front end application. Stock price service will establish an SSE with the external exchange. The external exchange will provide the real time quotes using SSE or websocket TCP/IP protocol.

User will add top 10 stocks or funds into the watchlist. This application will add user id and his top 10 stock list into the dictionary. Then it will establish the SSE with the exchange server.

There will be computation service which will get the available balance from Lifecad and store into the DB. This value will be real time.

User will decide to buy or sale signal based on the Mathematical calculations. The Buy or Sale order will go to the exchange server using the brokerage form API.

User will get the confirmation once the order is executed.

The transaction history will be store in the trading database. User can see the transaction history and can generate the report out of it.

**API Design**

Graphical user interface, application, email

Description automatically generated

**Database Design**

Graphical user interface, application

Description automatically generated