Requirment Analysis of

Postion Management Project

Version: 0.2

Date: 7 June, 2020

Author: Ted Wang

# Introduction

This Postion Management Project is an unsophisticated project for answering the interview test from CIeNet.

# Functionality

1. The program running will stand by and accept trade transactions.
2. The program tries to handle received transactions instantly. For those non-sequential transactions (which is not able to be processed instantly), store them into DB and process later.
3. All transactions are processed based on “TradeId”, and result into TradeStatus. Business rule on trade list here:
   1. A certain trade is initiated by a “INSERT” type of transaction.
   2. A trade quantity may be updated by a series of “UPDATE” type of transaction.
   3. A trade may be cancelled by a “CANCEL” type of transaction, quantity will be treated as 0.
   4. The detailed decryption of transaction is attached (supposed to attached the test paper here or a link is provided).
4. In any case when TradeStatus is updated, the EquityPosition will be accumulated again. This is to avoid aggregate function call in query EquityPosition.
5. The Program output is current equity position which should be identical with accumulation of all quantity of corresponding EquityPosition.

# Design

**Entity**:

Entity “Transaction” abstract the inbound transaction carrying all the transaction content.

Entity “TradeStatus” abstract the processed data model of this program. Each transaction will be processed and result in TradeStatus updating/cancelling. This entity is supposed to be stored in to database or other persistent approach.

Entity “EquityPosition” abstract the equity position.

**Service**:

EquityPositionService interface abstract the main function of this program, accept transaction, and then process and provide the result of equity position.

**REST Controller:**

PositionController expose REST api for accept transactions and query current equity positions.

**Exception:**

UnexpectedTransaction will be popped up while business error occurred.

1. INSERT type of trade with version != 1.
2. CANCEL type of trade has a version lower than any UPDATE trade.

# Comments

1. According to the requirements, there is no “END” flag of a trade other than “CANCEL”, that means the data size of table TradeStatus will be increased rapidly. Housekeeping work on this table should be considered.