Coding Test

Goal: Internal wallet transactional system (with a front end)

Requirements:

- Based on relationships every entity e.g. User, Team, Stock or any other should have their own defined "wallet" to which we could transfer money or withdraw
- Every request for credit/debit (deposit or withdraw) should be based on records in database for given model
- Every instance of a single transaction should have proper validations against required fields and their source and targetwallet, e.g. from who we are taking money and transferring to whom? (Credits == source wallet == nil, Debits == targetwallet == nil)
- Each record should be created in database transactions to comply with ACID standards
- Balance for given entity (User, Team, Stock) should be calculated by summing records

Tasks:

1. Architect generic wallet solution (money manipulation) between entities (User, Stock, Team or any other)
2. Create model relationships and validations for achieving proper calculations of every wallet, transactions
3. Use STI (or any other design pattern) for proper money manipulation