ChatGPT - Project feedback generator - Project Feedback Summary

Project Workflow.pdf
PDF

Please give feedback on my project attached.

Here is detailed feedback on your project based on the attached workflow document.

Strengths:

1. **Clarity and structure**: The workflows are well-organized and consistently follow a pseudocode-like structure, aligning with Domain-Driven Design (DDD) principles. The use of ubiquitous language is clear, and the overall format is easy to follow. It is clear which events trigger certain workflows and the expected outcomes, which makes the document navigable.
2. **Comprehensive workflow coverage**: You have a broad range of workflows that address key functional requirements from the project specification, including trading strategy initialization, arbitrage opportunity identification, and order fulfillment. Important concepts like stop conditions and input/output flows are generally covered.
3. **Correct workflow steps**: Several workflows, such as "Request cross-trading crypto info" and "Start Trading," align well with the requirements of identifying arbitrage opportunities and managing the lifecycle of trades. The use of external systems (e.g., Bitfinex, Kraken) is explicitly mentioned and clearly integrated into workflows.
4. **Domain Errors**: Your explicit definitions for domain errors like `DBConnectionError` and `InvalidTradingParamsError` show good attention to error handling, a critical part of robust software design.
5. **Event definitions**: The majority of the events are well-structured, with clear and distinct inputs and outputs, such as `Cross-tradedCryptoCurrenciesInfoStored` and `OrderQuantityAdjusted`.

Weaknesses:

1. **Missing End-to-End Traceability**: While the workflows generally cover individual steps, the overall flow from identifying an arbitrage opportunity through to finalizing trades

lacks cohesive traceability. Specifically, workflows like “Start Trading” do not clearly feed into follow-up steps, making it harder to trace the process fully through to completion. This could be enhanced by a clearer connection between workflows, such as how order status updates trigger the completion of the order.

2. **Real-time data handling**: The handling of real-time data, especially with regard to comparison of market data across multiple exchanges, is under-specified. It is unclear how the system will continuously store the latest prices to facilitate real-time arbitrage detection. This is critical for identifying arbitrage opportunities in workflows like “Start Trading.”
3. **Handling of Stop Conditions**: While some stop conditions are addressed, they are inconsistently applied. In several workflows like "Start Trading," "Continue partially fulfilled order," and "Notify user when one side of an order filled," the conditions under which the trading activity fully stops are not clearly outlined. This can lead to confusion regarding when and how the system should cease its operations.
4. **Unclear P&L and Annual Return Calculation**: The workflows related to the Profit and Loss (P&L) and Annual Return calculations, such as "Calculate Annual Return" and "Calculate Historical P&L," could be clearer. Specifically, the `formula` for calculating the annual return is provided, but the workflow lacks sufficient detail about how the data (e.g., cumulative P&L, number of years) is collected and maintained over time.
5. **Workflow gaps**:
    - The workflow for **retrieving order status** is missing, which is essential for maintaining an overview of ongoing trades.
    - There is no specific step explaining how to **retrieve and store historical arbitrage opportunities**, which is required by the project specification.
    - **User notification workflow** (such as via email when certain events like one-side order filling happen) could benefit from more explicit steps, particularly regarding how the email notification integrates with external systems.

Weaknesses Restated in Workflows:

- **End-to-End Traceability**: The workflow flow lacks an end-to-end structure. For example, in "Start Trading," after orders are placed, there is no explicit link to follow-up actions like updating the order status or notifying the user when the orders are partially completed.

To improve, consider linking workflows where outputs from one serve as inputs to another.

- **Real-Time Data Handling**: In workflows like "Request cross-trading crypto info" and "Start Trading," there is no mechanism described for maintaining the most recent real-time market data for comparison across exchanges. A solution could include a step for updating and persisting the latest market prices at regular intervals.
- **Stop Conditions**: Workflows like "Continue partially fulfilled order" should detail when and how trading stops after orders are completed. Right now, it's unclear how the system fully halts when certain conditions are met (like a P&L threshold being reached). Stop conditions need to be more explicit in every relevant workflow.
- **Profit and Loss (P&L)**: In "Calculate P&L," the steps are too abstract. The workflow should indicate how P&L is recalculated continuously and how this ties into other system components like data storage and user notifications. Consider adding a workflow that governs P&L recalculation and user updates on the status.

Summary of Improvement Areas:

1. **Improve traceability** across workflows to ensure end-to-end system behavior can be followed.
2. **Real-time data comparison** and its use in detecting arbitrage opportunities need more detail.
3. **Clarify stop conditions** across workflows, making the system's stopping behavior explicit.
4. **Elaborate P&L workflows**, ensuring data collection and storage is adequately described.
5. Ensure all **required workflows**, like retrieving order status and user notifications, are complete and integrated.

Your project demonstrates a solid understanding of the core concepts, but focusing on these areas will greatly improve the coherence and completeness of your system.