Technical Documentation

GoQuant — Position Management Backend

This document contains the complete technical specification for the GoQuant Position Management System, including architecture, formulas, smart contract design, backend service structure, and risk management.

---

1. System Architecture

1.1 Overview

The GoQuant Position Management System is built as a modular backend responsible for:

- Opening, updating, and closing positions

- Margin & leverage validation

- PnL and liquidation calculation

- Risk engine monitoring

- Persistent storage via PostgreSQL

- External price oracle integration

The system follows a clean separation of concerns:

API → Business Logic → Risk Engine → Database.

1.2 High-Level Components

API Layer (Axum)

- HTTP routing

- Validates user input

- Sends responses in JSON

Business Logic Layer

- Position math engine

- Computes PnL, liquidation, leverage

- Lifecycle transitions

Database Layer (SQLx + PostgreSQL)

- Stores all position fields

- Ensures ACID properties

Risk Engine

- Monitors equity

- Detects liquidation conditions

- Fetches oracle price

External Integrations

- Price feed / oracle

- Funding rate source

1.3 Component Interaction Diagram

```
flowchart LR

    User --> API\[API Layer (Axum)]

    API --> Logic\[Business Logic]

    Logic --> DB\[(PostgreSQL)]

    Logic --> Risk\[Risk Engine]

    Risk --> Oracle\[(Price Feed)]

    DB --> API

    Logic --> API




stateDiagram-v2

    \[\*] --> Created

    Created --> Open

    Open --> Updating

    Updating --> Open

    Open --> Closed

    Open --> Liquidated

    Closed --> \[\*]
```

```
    Liquidated --> \[\*]
```

Smart Contract Documentation

3.1 Account Structures

Margin Account

Field	Type	Description
owner	Pubkey	User wallet
collateral	u64	Deposited collateral
open_positions	u8	Number of active positions
last_funding_ts	u64	Timestamp

```
sequenceDiagram
    User ->> API: POST /positions
    API ->> Logic: Validate
    Logic ->> Risk: Check margin
    Risk ->> Logic: OK
    Logic ->> DB: Insert position
    DB ->> Logic: Success
    Logic ->> API: Response
    API ->> User: Position created
```

Mathematicaal Formulas

| Symbol | Meaning |

```
| ---------- | -------------------------- |
| E          | User equity / margin       |
| lev        | Leverage                   |
| Pe         | Entry price                |
| Mp         | Mark price                 |
| N          | Notional = E × lev         |
| Contracts  | Quantity = N / Pe          |
| imr        | Initial margin rate = 1/lev |
| mmr        | Maintenance margin rate    |
| PnL\_unreal | Unrealized PnL             |
| PnL\_real   | Realized PnL               |
| f          | Funding rate               |
```

3.3 Smart Contract State Machine

```
stateDiagram-v2
    \[\*] --> Created
    Created --> Open
    Open --> Updating
    Updating --> Open
    Open --> Closed
    Open --> Liquidated
    Closed --> \[\*]
```

Backend Service Documentation

4.1 File Structure

```
src/
```

```
 ├── api/
 ├── logic/
 ├── db/
 ├── risk/
 └── utils/
```

Module	Purpose
api	Routes, request validation
logic	PnL, liquidation, lifecycle
db	SQLx queries
risk	Margin \& liquidation checks
utils	Error handling

4.3 API Endpoints

POST /positions

Creates a position.

GET /positions

Lists positions.

GET /positions/{id}

Fetch a single position.

```
DELETE /positions/{id}
```

Close a position.

4.4 Database Schema (positions)

Column	Type
id	UUID
direction	TEXT
entry_price	DOUBLE
mark_price	DOUBLE
contracts	DOUBLE
notional	DOUBLE
leverage	INT
status	TEXT
created_at	TIMESTAMP
updated_at	TIMESTAMP

4.5 Deployment

Environment Variables

DATABASE_URL=

PORT=8080

ORACLE_URL=

RUST_LOG=info

```
cargo run --release

docker build -t goquant-backend .
docker run -p 8080:8080 goquant-backend
```

5\. Risk Management Guide

5.1 Monitoring Rules

Track notional, mark price

Compute equity

Compare with MMR

Use fresh oracle prices

5.4 Liquidation Flow

```
flowchart TD
    A\[Oracle Update] --> B\[Calculate Equity]
    B -->|Healthy| C\[OK]
    B -->|Below MMR| D\[Liquidate]
    D --> E\[Apply Penalty]
    E --> F\[Return Remaining Equity]
```