



Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the core implementation details for **FinSight - Real-Time Financial Portfolio Tracker**. FinSight is a secure, fullstack platform for tracking investments, analyzing portfolio performance, and receiving real-time market updates. The system leverages FastAPI, PostgreSQL, React, and integrates with third-party APIs for market data.

1. System Components

Component	Technology	Key Responsibilities
Frontend	React, TS, Tailwind	UI, data visualization, user interaction
Backend API	FastAPI (Python)	Business logic, API endpoints, auth, data processing
Database	PostgreSQL	Persistent storage, transactions, concurrency
Auth Service	JWT	Secure authentication, role-based access
Market Data Adapter	Python	Fetch/normalize third-party market data

2. Class/Interface Overview

Name	Type	Key Methods/Attributes	Relationships
User	Model	id, email, hashed_password, role	Owns Portfolios, Transactions
Portfolio	Model	id, user_id, name, created_at	Has many Holdings
Holding	Model	id, portfolio_id, asset_symbol, quantity	Belongs to Portfolio
Transaction	Model	id, portfolio_id, type, amount, timestamp	Belongs to Portfolio
MarketDataService	Service	fetch_prices(), cache_prices()	Used by Backend API
AuthService	Service	login(), verify_jwt(), get_user_role()	Used by Backend API

Key API Endpoints (FastAPI):

```
POST  /auth/login
GET   /portfolios/
POST  /portfolios/
GET   /portfolios/{id}/holdings
POST  /transactions/
GET   /market-data/{symbol}
```



3. Data Structure Overview

Table/Model	Fields
User	id (PK), email (unique), hashed_password, role (enum: user, admin)
Portfolio	id (PK), user_id (FK), name, created_at
Holding	id (PK), portfolio_id (FK), asset_symbol, quantity
Transaction	id (PK), portfolio_id (FK), type (buy/sell), asset_symbol, amount, timestamp
MarketData	symbol (PK), price, last_updated

4. Algorithms / Logic

Portfolio Value Calculation (Pseudocode):

```
def calculate_portfolio_value(portfolio_id):
    holdings = get_holdings(portfolio_id)
    total = 0
    for h in holdings:
        price = MarketDataService.fetch_price(h.asset_symbol)
        total += h.quantity * price
    return total
```

Role-Based Access (Summary):

- Endpoints check JWT for user role.
- Admins can view all portfolios; users can only access their own.

Concurrency Control (Summary):

- Database transactions with row-level locking for updates to holdings/transactions.

5. Error Handling

Scenario	Handling Approach
Invalid credentials	Return 401 Unauthorized
JWT expired/invalid	Return 401 Unauthorized, prompt re-login
Insufficient permissions	Return 403 Forbidden
Market data API failure	Return cached data or 503 Service Unavailable
Database constraint violation	Return 400 Bad Request with error details
Concurrent transaction conflict	Retry or return 409 Conflict