Elijah Senior
Tradie
SDD

Software Design Description (SDD)

6.1 Introduction

This document presents the architecture and detailed design for the Insider Trading Alert mobile application. The project aims to provide real-time alerts based on insider trade filings, allowing users to make informed stock market decisions.

6.1.1 System Objectives

The objective of this application is to:

- Aggregate insider trading data from OpenInsider via web scraping.
- Store and process data using a FastAPI backend.
- Provide a mobile application interface using React Native.
- Offer AI-generated summaries of insider profiles and stock predictions.
- Authenticate users with Clerk and store user data in Supabase.
- Deliver notifications for significant insider trades.

6.1.2 Hardware, Software, and Human Interfaces

- Hardware Interfaces: Mobile devices (Android/iOS), cloud-hosted API services.
- Software Interfaces:
    - Backend: FastAPI (Python), Pandas (for data processing), Supabase (database), Fly.io (potential hosting option).
    - Frontend: React Native, Tailwind, for mobile app UI.
    - APIs: Gemini, Yahoo Stocks API for stock information.
- Human Interfaces: Mobile app UI with real-time stock charts, AI summaries, and notifications.

6.2 Architectural Design

6.2.1 Major Software Components

- Web Scraper: Collects insider trading data from OpenInsider.
- Data Processor: Uses Pandas to structure the data.
- FastAPI Backend: Serves data to the mobile application.
- Database (Supabase): Stores user information and preferences.
- Mobile App (React Native): Displays data, AI insights, and stock charts.
- AI Module: Summarizes insider activities and predicts stock trends.

6.2.2 Major Software Interactions

- Web scraper fetches data periodically and updates the FastAPI backend.
- Mobile app requests data from FastAPI.
- AI module processes insider trade data and generates summaries.
- Firebase sends notifications to users based on trade significance.

6.2.3 Architectural Design Diagrams

(To be included: UML diagrams for system architecture, data flow, and component interactions.)

6.3 CSC and CSU Descriptions

6.3.1 Class Descriptions

6.3.1.1 Web Scraper

- Fields: source_url, dataframe
- Methods: fetch_data(), parse_data()

6.3.1.2 DataProcessor

- Fields: dataframe
- Methods: clean_data(), transform_data()

6.3.1.3 FastAPIBackend

- Fields: endpoints, database
- Methods: get_insider_trades(), store_user_preferences()

6.3.1.4 MobileApp

- Fields: user_interface, notifications
- Methods: display_trades(), send_alerts()

6.3.2 Detailed Interface Descriptions

- FastAPI Endpoints: /trades, /user, /alerts
- Database Tables: Users, InsiderTrades, Notifications

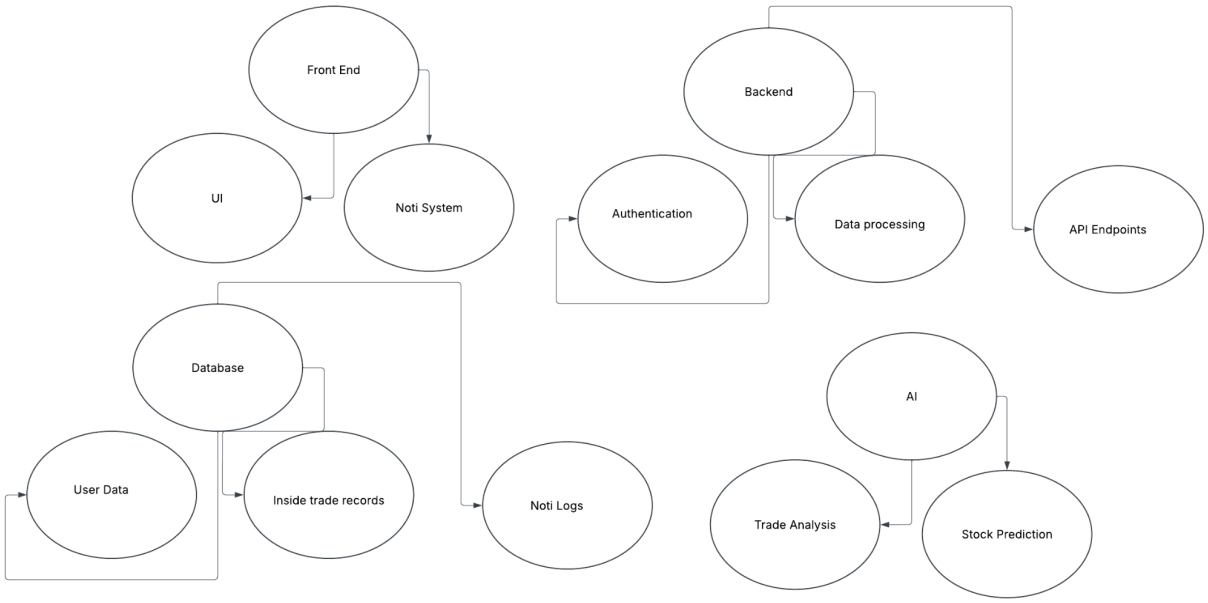
6.3.3 Detailed Data Structure Descriptions

Data attributes include:

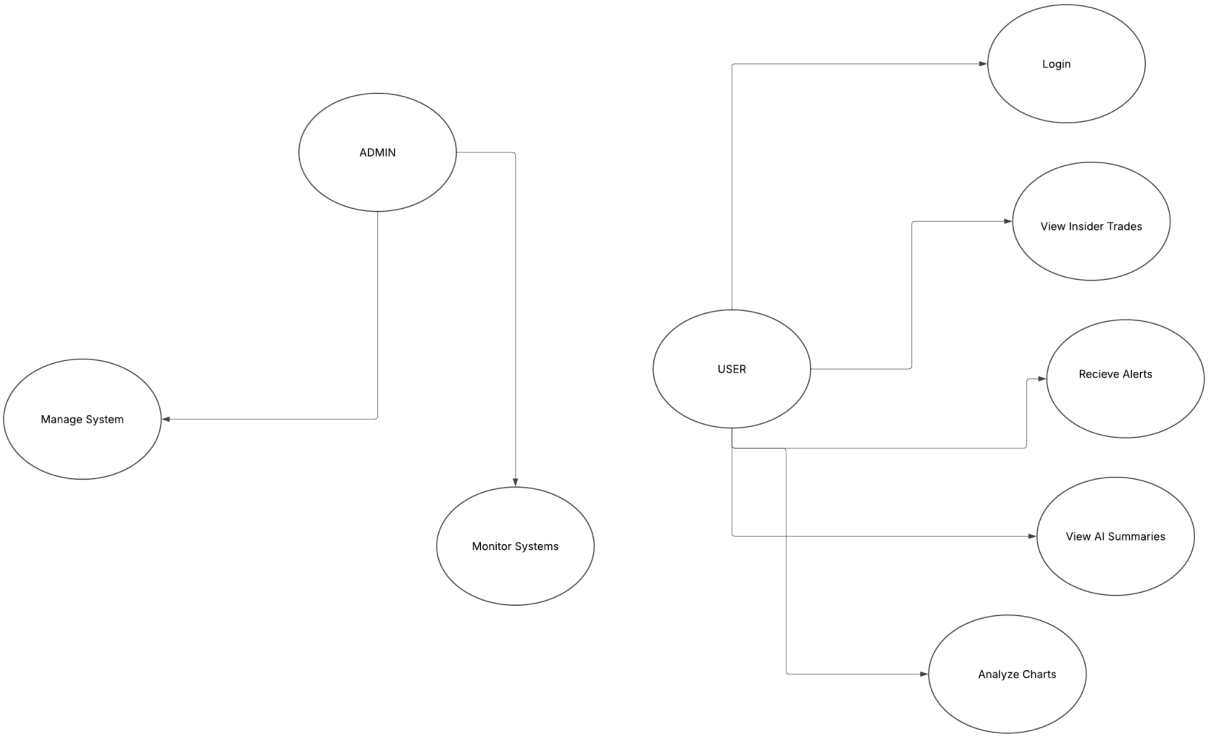
- filingDate, tradeDate, ticker, companyName, insiderName, tradeType, price, quantity, moneyValueIncrease

6.4 Database Design and Description

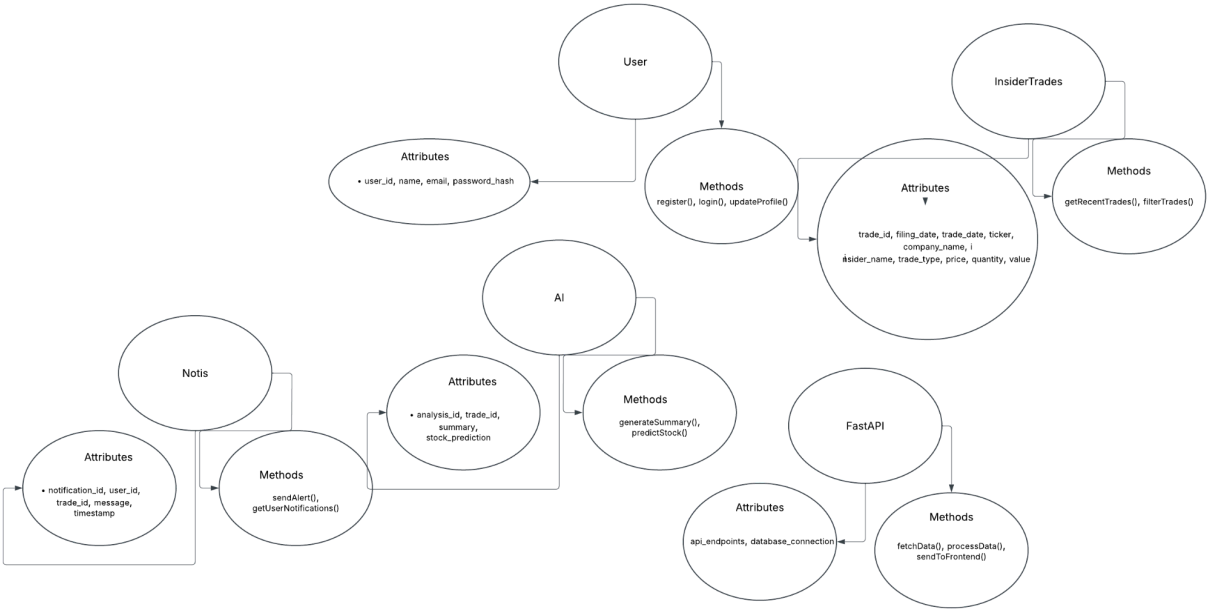
6.4.1 Use Case Diagram



Package Diagram



Class Diagram



6.4.2 Database Access

- Supabase API used for secure data access.

6.4.3 Database Security

- JWT authentication via Clerk.
- Role-based access control for data integrity.
