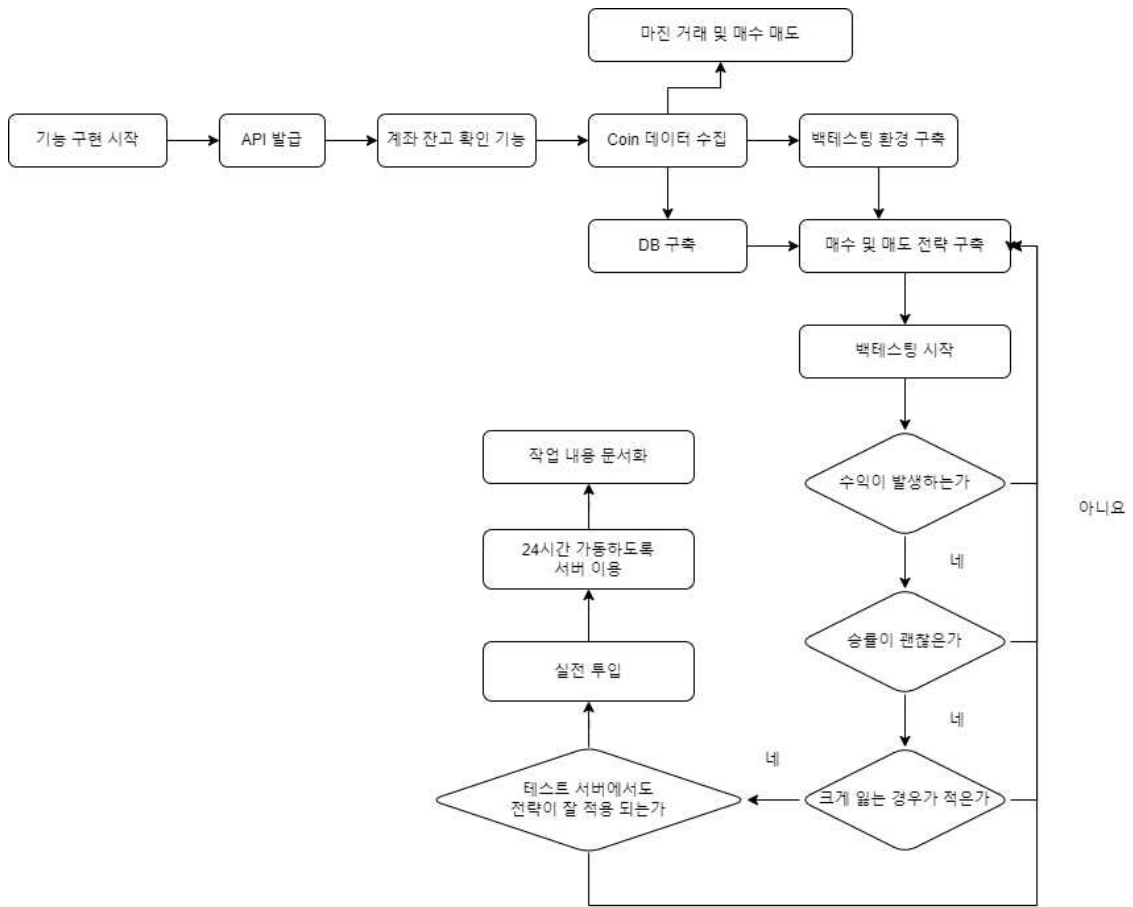


Quant Trading Service 계획서

1. 기능 구현

- 1) 계좌 잔고 확인
- 2) 백테스팅 환경 구축 (과거 기록을 가지고 시뮬레이션)
- 3) DB 구축
 - 3-1) BTC 코인 최근 3년 데이터 초 단위 거래량, 시세
(단위는 가용가능한 DB 크기에 따라서 변경)
 - 3-2) 매매액, 매수액, 거래 코인의 종류, 이익, 계좌 현황
- 4) 마진 거래
 - 4-1) 매수, 매도
 - 4-2) 레버리지 몇 배로 할 것인지 (3배, 10배)
 - 4-3) 수수료 계산
- 5) TestNet (테스트 서버 가상 통화로 거래) 사용법
- 6) 24시간 서버로 실행시키는 법
- 7) API키 발급 후 등록
- 8) 매수 및 매매 전략 구축 (ex 인공지능, 차트 분석 등등)
- 9) 깃허브 연동
- 10) 작업 내용 문서화

2. 플로우 차트



3. 상세 계획

1) 기능 구현 시작

- 구성원의 깃허브 계정을 모아서, 프로젝트용 Repository 생성 후 commit 규칙을 정한다.

ex) 1. main branch로부터 코드를 가져온 후, 한 branch를 만들어서 기능을 구현 후에 merge하고 그 branch는 삭제한다.

2. main 폴더를 토대로 하고 각 구성원의 폴더를 생성하고 그 폴더 내에서 작업한다.

... 등등

- 주석을 통일하기 위해 관련해서 규칙을 세운다.
- 전체적인 틀을 보면서 역할을 분담한다.
- 팀 규칙 정하기

ex) 1. 매주 일요일 저녁 8시에 코드 리뷰를 한다.

... 등등

2) API 발급

- Binance에서 API를 발급받아서 시스템 환경 변수에 API KEY, SECRET KEY를 등록하거나, 대표로 한 명이 발급을 받아둔다

3) 계좌 잔고 확인 기능

- Binance Api 문서를 확인해서 python으로 계좌 잔고 확인하는 함수를 만든다.

4) Coin 데이터 수집

- Binance에서 BTC 데이터를 읽어오는 함수를 작성한다.
- 데이터를 어떤 형식으로 저장할지 정한다.
- 불러들여온 데이터를 정제한다. BTC , USDT , KRW 등 단위를 서로 변경 할 수 있게끔 작용하는 함수를 작성한다.
- timestamp와 datetime 서로 변경하는 함수를 작성한다.

5) 마진 거래 및 매수 매도

- Binance에서 제공하는 거래 방식에 대해서 알아본다. oco, normal, margin 등
- Binance에서 매수, 매도, 주문 취소, 마진 거래 등의 함수를 작성한다.
- 수수료를 계산하는 함수를 작성한다.

6) DB 구축

- 4)에서 정한 데이터 방식에 따라서 저장할 데이터 형식을 정한다.
- 얼마나 촘촘하게, 어디까지의 데이터를 가져와서 저장할지 정한다.
- DB에 저장하기 위해 입력 query와 받는 데이터 함수의 형식을 정한다.
- AWS를 활용하여 DB를 구축한다.

7) 백테스팅 환경 구축

- class 형식으로 가상 인물을 생성한다. 자산을 보유하고, 매수와 매도 시에 보유금이 변경하고, 보유 화폐 또한 변경하게끔 작성한다.
- DB에서 불러들인 값으로, 매수와 매도 했을 때 변경되는 값들을 함수로 작성한다

8) 백테스팅

- 특정 조건에 도달했을 때, 매수와 매도하는 전략을 세우고 백테스팅을 한다.
- 각 전략을 평가하는 기준을 정해서, 가장 좋은 전략을 선정한다.
- 선정된 전략을 테스트넷에서 사용해본다.

9) 실전 투입

- 소액으로 돌려본다.

10) 서버를 활용해서 24시간 구동하기

- AWS의 서버를 활용해서 24시간 프로그래밍 되도록한다.

11) 작업 내용 문서화

- 그 동안 작업한 것들 문서로 작성하고, 코드를 깔끔하게 정리한다.