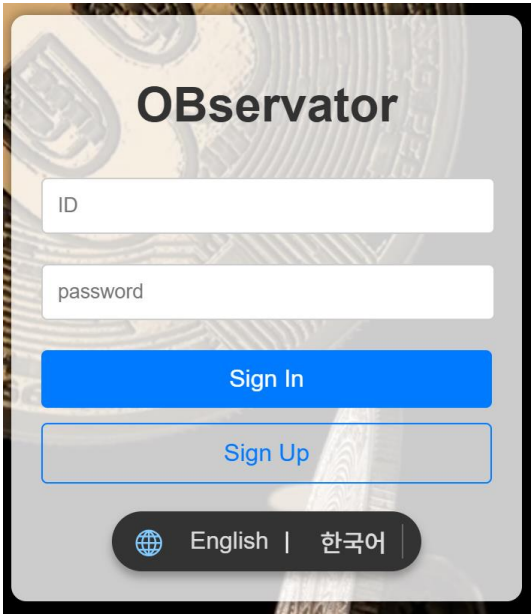




	<p>※ 포상실적은 해당사항이 있을시 필히 기재 요망.</p> <p>※ 포상실적을 허위로 기재시 신청인은 포상대상에서 제외됨</p> <p>※ 타기관에서 이미 수혜받은 정부포상 과제는 포상대상에서 제외됨</p>
--	----------------------------------------------------------------------------------------------------------------------

요약문			
과제의 목표	<p>부동산 가격 폭등으로 인해 월급만으로 내 집을 마련하기 어려운 상황에서, 사회적으로 투자에 대한 관심이 어느 때보다 높아지고 있다. 하지만 투자는 세계적으로 다양한 상황이 얹힌 복잡한 분야이기 때문에 분석이 쉽지 않으며, 미래를 예측하기 어렵고 이로 인해 안정적인 수익을 내는 것이 어렵다. 따라서, 전문적인 지식이 없더라도 안정적으로 수익을 올릴 수 있는 시스템 트레이딩 서비스를 개발하고자 한다. 본 프로젝트는 가상화폐 거래소의 API를 이용하여 호가창(Order Book) 데이터를 실시간으로 수집하고, 이를 기반으로 머신러닝(ML) 모델을 학습시켜 자동 매매 시스템을 구축하는 것을 목표로 한다. 실시간으로 호가창 데이터를 수집하고, 이를 데이터베이스에 저장한 후, 머신러닝 알고리즘을 사용하여 데이터를 분석한다. ML 모델을 사용하여 예측을 수행하며, 최종 매매 결정을 시행한다. 이 시스템은 가상화폐 거래소인 업비트 API와 연동되어 있으며, 실시간으로 분석된 데이터를 바탕으로 자동으로 매매가 실행된다.</p>		
과제 수행 내용	<p>크게 세가지의 구성을 가진 시스템을 구축했다.</p> <p>Client에서는 웹 클라이언트를 통해 계정을 생성하고 로그인할 수 있는 회원가입 및 로그인 기능, 현재 가격, 호가창, 체결 정보 등 실시간 데이터를 확인할 수 있는 실시간 가상화폐 정보 조회 기능, 자동 매매를 시작하거나 중지할 수 있는 가상화폐 자동거래 기능, 원하는 가격과 시간에 맞춰 매매를 예약할 수 있는 가상화폐 수동/예약거래 기능, 보유 자산, 실시간 수익률 등을 확인 할 수 있는 자산 정보 확인 기능을 구현했다.</p> <p>거래 서버는 AWS Cloud를 이용했으며 Upbit 거래소 API로부터 실시간 오더북과 체결 데이터를 수신하여 필요한 전처리를 수행하는 데이터 수집 및 전처리 기능, S3에 모델 학습 및 예측에 필요한 데이터(오더북, 체결 정보 등)를 저장하고 RDS에 사용자 정보, 예약 거래 정보, 설정된 임계값 등을 저장하는 데이터 저장 기능, 전처리된 데이터를 WSL의 머신러닝 서버로 실시간 전달하고 거래 명령과 예측값을 수신하여 Upbit 거래소로 전달하는 WSL 서버와 통신 기능을 구현했다.</p> <p>ML 서버는 WSL을 사용했으며 AWS Cloud와 통신하여 거래서버로부터 전처리된 데이터를 수신하여 사전 학습된 LSTM 모델을 통해 설정된 임계값에 따라 거래 명령과 예측값을 생성하여 거래서버로 전송하는 머신러닝 서버, 데이터 DB(S3)에 저장된 데이터를 주기적으로 불러와 학습하여 모델 성능을 유지하는 모델 학습 및 업데이트 기능을 구현했다.</p>		
결과 및 파급효과	<p>가상화폐 시장의 데이터를 분석하여 자동 매매 서비스를 제공, 초보자도 간단한 설정만으로 투자 관리가 가능하며, 머신러닝 기반의 실시간 의사결정으로 손실률을 줄이고 안정적인 수익을 기대할 수 있는 프로그램을 개발했다.. 차후 다양한 시장 및 거래소와도 통합이 가능하도록 설계되었다.</p>		
중요단어	가상화폐	자동매매	호가창
	머신러닝	거래소	투자
	인터랙티브	실시간	AWS

보고서					
작품명 (프로젝트명)	OBservator				
# Key Words	가상화폐	자동매매	호가창	머신러닝	실시간
1. 프로젝트 개요	<p><b>1) 과제의 개발동기 및 목적:</b> 부동산 가격 폭등으로 인해 월급만으로 내 집을 마련하기 어려운 상황에서, 사회적으로 투자에 대한 관심이 어느 때보다 높아지고 있다. 하지만 투자는 세계적으로 다양한 상황이 얹힌 복잡한 분야이기 때문에 분석이 쉽지 않으며, 미래를 예측하기 어렵다. 이로 인해 안정적인 수익을 내는 것이 쉽지 않다. 따라서, 전문적인 지식이 없더라도 안정적으로 수익을 올릴 수 있는 시스템 트레이딩 서비스를 개발하고자 한다.</p> <p><b>2) 개발목표 및 범위:</b> 본 프로젝트는 가상화폐 거래소의 API를 이용하여 호가창(Order Book) 데이터를 실시간으로 수집하고, 이를 기반으로 머신러닝(ML) 모델을 학습시켜 자동 매매 시스템을 구축하는 것을 목표로 한다. 실시간으로 호가창 데이터를 수집하고, 이를 데이터베이스에 저장한 후, 머신러닝 알고리즘을 사용하여 데이터를 분석한다. ML 모델을 사용하여 예측을 수행하며, 최종 매매 결정을 시행한다. 이 시스템은 가상화폐 거래소인 업비트 API와 연동되어 있으며, 실시간으로 분석된 데이터를 바탕으로 자동으로 매매가 실행된다.</p>				
2. 최종 결과물 소개	<div>  <div> <p>● 로그인(/login)</p> <ul style="list-style-type: none"> <li>○ ID, 비밀번호 입력 후 로그인</li> <li>○ 실패시 실패 문구 출력</li> <li>○ Sign In -&gt; 로그인 후 홈 페이지로 이동</li> <li>○ Sign Up -&gt; 회원가입 페이지로 이동</li> </ul> </div> </div> <p>그림 1 로그인 페이지</p>				

### 회원가입

중복확인

특수문자 없이 4자~20자로 입력해주세요.(중복확인 필수)

@ 도메인 선택

연락처 010 -  -

특수문자를 포함한 8~16자로 입력해주세요.

☐ (필수) 개인정보 수집 및 활용 동의 약관보기

☐ (선택) 이메일 알림 서비스 수신 동의

회원가입

이미 계정이 있으신가요? [로그인](#)

그림 2 회원가입 페이지

- 회원가입(/signup)
  - ID 중복확인
  - e-mail, 연락처, 비밀번호 입력
  - 개인정보 수집 및 활용 동의 체크박스
  - 약관보기 → 서비스 이용약관 팝업
  - 이미 계정이 있는 경우 로그인으로 이동

## OBservator

가상화폐 운용의 새로운 척도

프로필
거래 페이지

로그아웃

English | 한국어

© 2024 OBservator Project. All rights reserved.

그림 3 홈 페이지

- 홈(/home)
  - 언어 선택
    - 영어/한국어로 전체 번역
    - 페이지 이동
    - 프로필, 거래 페이지 버튼 클릭 시 각 페이지로 이동
  - 로그아웃
    - 로그아웃 → 로그인 페이지 이동
  - 내비게이션 바
    - 각 화면에서 다른 페이지로 이동하는 기능 구현

프로필 | English | 한국어 | 대시보드 | 프로필 | 홈 | 로그아웃

### 프로필 업데이트

업비트 키 입력 여부: Yes

사용자명  
ax4628

이메일  
ax4628@gmail.com

업비트 Access key: [발급받는 방법](#)  
Hdzjld8ufdYJdvwxOESFkww15xlgde1rWFE87Jx

업비트 Secret key:  
\*\*\*\*\*

[저장](#)

### 보유 자산 정보

[해당 보내기](#)

보유자산	보유수량	매수평균가
BTC	0.00000882	148369065.12319119
KRW	1576621.96095432	0

[홈으로 돌아가기](#)

## ● 프로필(/profile)

- 사용자 정보 확인 (ID, 이메일)
- 업비트 Access key 발급받는 방법 제공
  - 팝업에 따라 진행 후 복사
  - 저장 → DB에 반영
- 현재 보유자산 확인 (BTC, KRW)
- 비트코인 매수 평균가 확인
- 메일 보내기 → 등록된 이메일로 현재 자산 현황 전송
- 홈으로 돌아가기 → 홈 페이지로 이동

그림 4 프로필 페이지

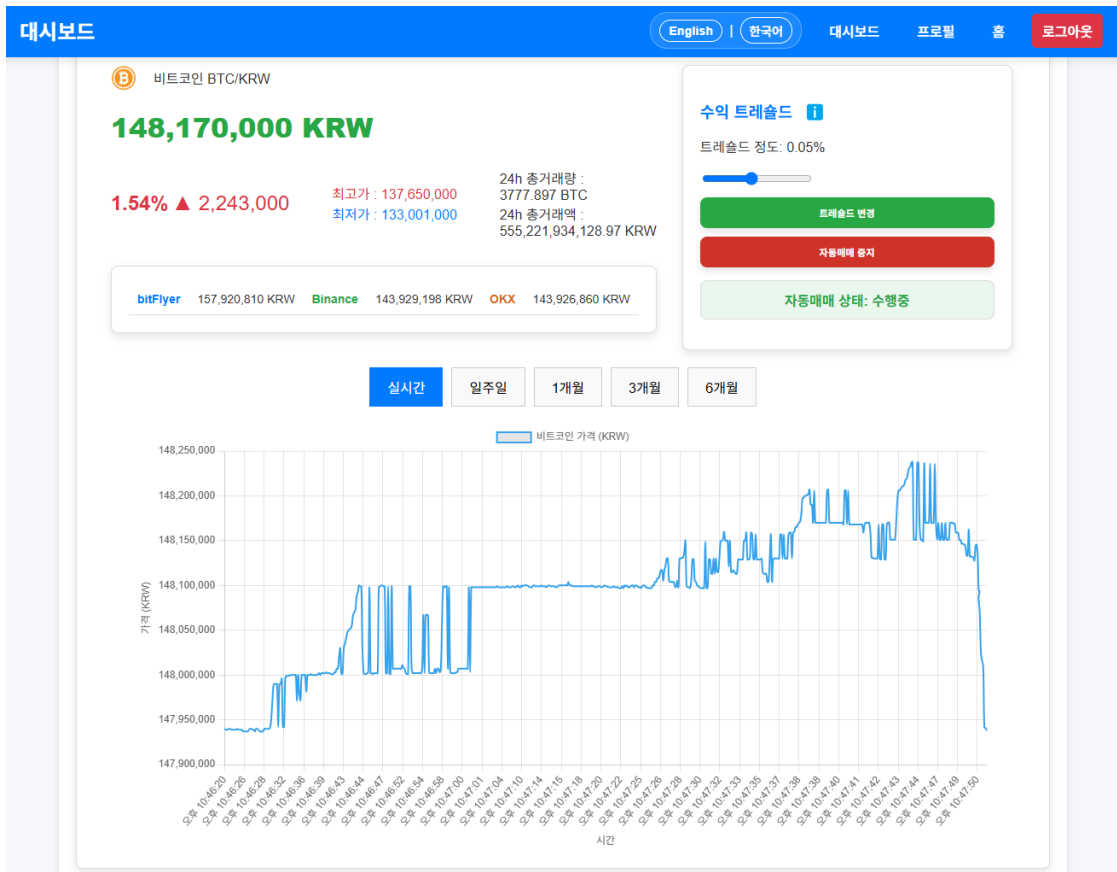


그림 5 트레이드 페이지(상단)

- 대시보드(/trade)
  - 실시간 가격 정보
    - 현재 가격, 변동률, 변동 금액, 최고/최저가 등 정보 표시
    - 비트플라이어, 바이낸스, OKX의 거래소 가격 표시
  - 수익 트레슬드
    - 슬라이더로 결정, 버튼으로 적용해 트레슬드 설정
    - 자동매매 시작/중지
  - 가격 변동 그래프
    - 실시간, 일주일 등 시간대별 비트코인 가격 표시

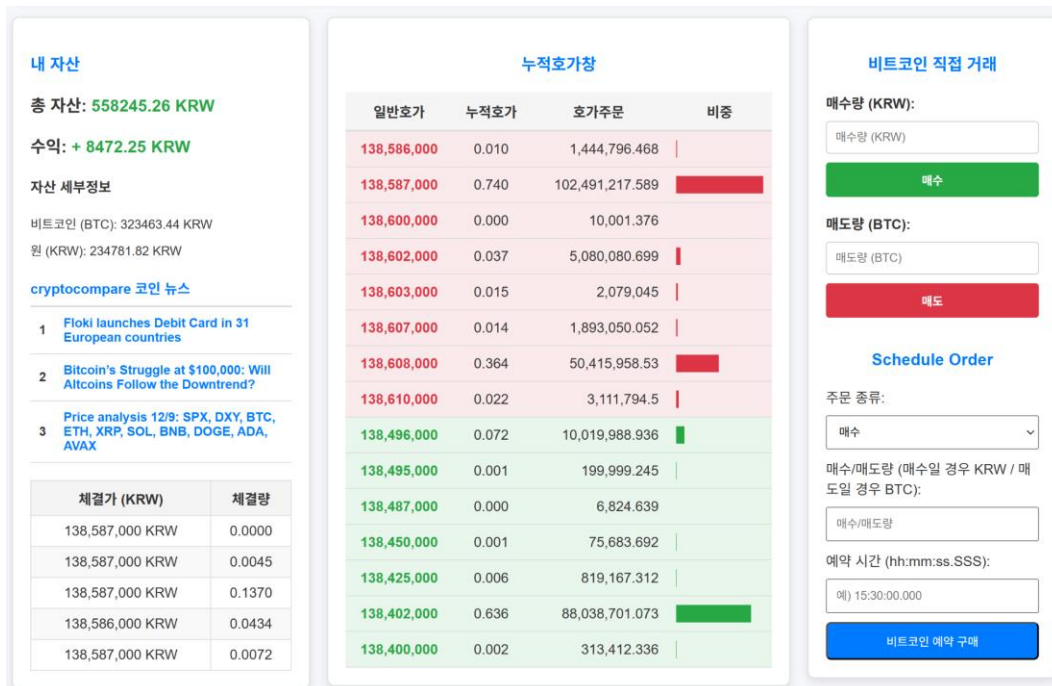


그림 6 트레이드 페이지(하단)

- 내 자산
  - 총 자산, 수익/손실 자산 세부정보
- 비트코인 관련 최신 뉴스 목록
- 체결 데이터
  - 최근 체결가/체결량을 표시
- 누적호가창
  - 현재 호가창 정보를 실시간 표시
- 비트코인 직접거래
  - 매수/매도 수동매매 수행
- 예약 주문
  - 예약된 시간에 매매 수행

<p>3. 프로젝트 추진 내용</p>	<p><b>3.1 추진 배경</b></p> <p><b>1) 개발 배경 및 필요성 :</b></p> <ul style="list-style-type: none"> <li>● 배경: 월급만으로 내 집을 마련하기 어려운 상황에서, 사회적으로 투자에 대한 관심이 어느 때보다 높아지고 있다. 하지만 투자는 세계적으로 다양한 상황이 얹힌 복잡한 분야이기 때문에 분석이 쉽지 않으며, 미래를 예측하기 어렵다. 이로 인해 안정적인 수익을 내는 것이 쉽지 않다. 따라서, 전문적인 지식이 없더라도 안정적으로 수익을 올릴 수 있는 시스템 트레이딩 서비스를 개발하고자 한다.</li> <li>● 필요성 : 바쁜 현대인들에게는 투자에 필요한 정보 수집과 분석에 시간을 할애하기 어려운 경우가 많다. 이에 따라 투자 과정을 자동화하고, 시간과 노력을 절감할 수 있는 시스템 트레이딩의 필요성이 증가하고 있다 이 과정에서 알고리즘과 데이터 기반의 분석을 통해 객관적이고 일관성 있는 결정을 내리는 시스템 트레이딩은 감정적인 요인을 배제하여 안정적인 수익률을 기대할 수 있다.</li> </ul> <p><b>2) 선행기술 및 사례 분석 :</b></p> <ul style="list-style-type: none"> <li>● CCXT:               <p>가상화폐 거래소 API를 통합한 오픈소스 라이브러리로, 여러 거래소에서 실시간 데이터 및 거래 기능을 제공.</p> <ul style="list-style-type: none"> <li>- CCXT는 거래소 에서 정보를 가져오거나 거래를 하는 용도의 라이브러리로 실제로 매매가를 결정하는 알고리즘은 직접 만들어야 함.</li> </ul> </li> <li>● Pyupbit:               <p>업비트(Upbit)에서 제공하는 API를 Python에서 쉽게 사용할 수 있도록 만든 오픈소스 라이브러리.</p> <ul style="list-style-type: none"> <li>- 업비트의 정보를 불러 오거나 거래를 쉽게 할 수 있는 라이브러리로 실제로 매매가를 결정하는 알고리즘은 직접 만들어야 함.</li> </ul> </li> <li>● gpt-bitcoin:               <p>ChatGPT를 활용해 코인 매매의 시기적 적절성을 확인 할 수 있는 프로그램.</p> <ul style="list-style-type: none"> <li>- ChatGPT를 활용하였기 때문에 매매 적절성에 대한 검증이 불가능하며 수정도 불가능. 또, 같은 상황에서 같은 행동을 반복했을 때 같은 결과 값을 낼 수 있을지에 대한 의구심.</li> </ul> </li> <li>● BAMOWL:               <p>가상화폐 가격 예측 AI 기술을 접목한 백테스팅 &amp; 자동매매 서비스.</p> <ul style="list-style-type: none"> <li>- 백테스팅 기반으로 예측하여 자동 매매 하는 서비스로 사용자가 직접 포트폴리오를 작성하여 실행할 수 있다는 특징. 이번 프로젝트에서는 호가창 데이터를 쓴다는 점에서 위 프로젝트와의 차별성을 두고 있음.</li> </ul> </li> <li>● 가상화폐 자동 트레이딩 장치 및 시스템:               <p>알고리즘을 이용하여 매수할 주식을 결정하고 딥러닝을 이용해 매도 가격을 조정하는 시스템</p> <ul style="list-style-type: none"> <li>- 이번 프로젝트에서는 호가창 데이터를 쓴다는 점에서 위 특허와의 차별성을 두고 있음.</li> </ul> </li> </ul>
----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### 3) 요구사항 분석 :

#### ● 요구사항 명세서

번호	대상	분류	기능명	기능 설명	화면	기능	비고
1	비회원	사용자	회원가입	비회원이 이메일, 비밀번호, 이름, 휴대폰 번호를 입력하여 회원 계정을 생성할 수 있음.	회원가입 페이지	사용자 정보를 입력하고 회원가입 버튼 클릭	회원 전환 후 회원 기능 이용 가능
2	회원	사용자	로그인	회원이 이메일과 비밀번호를 입력하여 시스템에 로그인할 수 있음.	로그인 페이지	로그인 버튼 클릭 후 인증 성공 시 메인 화면 이동	로그인 실패 시 오류 메시지 표시
3	회원	사용자	내 정보 조회	로그인한 회원이 자신의 계정 정보를 조회할 수 있음.	프로필 페이지	이름, 이메일, 가입 날짜, 거래 키 상태 표시	-
4	회원	사용자	가상화폐 정보 조회	회원이 주요 가상화폐의 현재 시세, 가격, 거래량 등을 조회할 수 있음.	대시보드 페이지	선택한 가상화폐의 상세 정보 표시	외부 API 연동
5	회원	사용자	거래 키 발급	회원이 가상화폐 거래소와 연동하여 거래 키를 발급받아 시스템에 저장.	프로필 페이지	거래 키 요청 버튼 클릭 후 결과 표시	거래 키 없으면 거래 수행 불가
6	회원	사용자	거래 키 설정	거래 키가 발급된 상태에서 수동 거래, 예약 거래, 자동 거래 수행 가능.	대시보드 페이지	거래 방식 선택 후 매수/매도 실행	-
7	회원	사용자	수동 거래	회원이 실시간으로 매수/매도를 입력하고 즉시 거래를 수행.	대시보드 페이지	매수/매도량 입력 후 실행	거래 키 필수
8	회원	사용자	예약 거래	회원이 특정 조건(시간, 가격)에 따라 예약 거래를 설정하고 수행.	대시보드 페이지	예약 조건 설정 후 시작	거래 키 필수
9	회원	사용자	자동 거래	ML 모델의 예측값을 기준으로 설정된 수익 프레임에 따라 거래를 자동으로 수행.	대시보드 페이지	자동 매매 시작/중지 버튼	거래 키 필수
10	관리자	관리자	ML 모델 학습	관리자가 시스템 내에서 ML 모델 학습을 수행하여 최신 데이터를 바탕으로 모델을 업데이트.	-	학습 시작 버튼 클릭 후 진행 상태 표시	학습 결과는 자동 거래에 반영
11	가상화폐 거래소	외부	거래 키 발급	외부 거래소가 회원별 고유 거래 키를 발급하고 시스템에 전달.	-	API 요청에 따라 거래 키 반환	거래 키는 암호화 저장

표 1 요구사항 명세서

#### 4) 설계의 현실적 제한요소 도출 :

- 대량의 데이터를 실시간으로 처리하는 ML 모델을 구현하기 위해서는 좋은 성능의 Hardware(GPU)를 필요로 함. 클라우드 GPU를 활용하는 방법이 있으나 높은 금액 산정으로 인해 클라우드 GPU 대신 고성능의 GPU를 가진 개인용 컴퓨터를 사용.
- ML 모델 구축시 대량의 정보로 인해 과적합 문제가 발생 가능. 따라서 모든 데이터를 다 이용하는 것이 아닌 최근 5일정도의 데이터와 과거 급등/급락 데이터만을 이용하여 너무 많은 데이터가 들어가는 것을 방지하고 outlier에 대응.
- 새로운 대량의 정보를 지속적으로 ML 재학습을 위해 사용시 프로그램의 실시간 처리의 지연이 발생 가능. 따라서 ML 재학습은 실시간으로 진행하는 것이 아닌 일정 시간마다 진행.
- 가상자산은 소득세법에 따라 세금을 납부해야 하나 서비스를 제공하는 입장에서는 직접적으로 이를 해 줄 수가 없어 사용자가 직접 이를 인지하고 납부해야 함.

### 3.2 프로젝트 구현과정

#### 1) 기능 정의 :

- 회원가입 및 로그인: 사용자들은 웹사이트에서 간편하게 회원가입과 로그인을 할 수 있으며, Upbit API 키를 등록해 시스템과 연동 가능.
- 실시간 대시보드: 웹사이트 대시보드를 통해 가상화폐의 실시간 정보(현재 가격, 호가창, 체결 정보 등)와 관련 뉴스를 한눈에 확인 가능.
- 임계값 설정: 대시보드에서 사용자 지정 임계값을 설정해 자동 매매 전략을 세부적으로 조정 가능.
- 자동매매 제어: 대시보드에서 자동매매 시작/중지 버튼을 통해 간단히 거래 실행 여부를 제어할 수 있음.
- 24시간 모델 업데이트: 시스템은 매일 24시간마다 수집된 새로운 데이터를 기반으로 기존 모델을 재학습하며, 가장 성능이 우수한 모델로 자동 업데이트하여 실시간 예측 정확도를 지속적으로 향상시킴.



## 2) 구체적인 설계안 :

### ● 프로젝트 설계 - 유스케이스 다이어그램

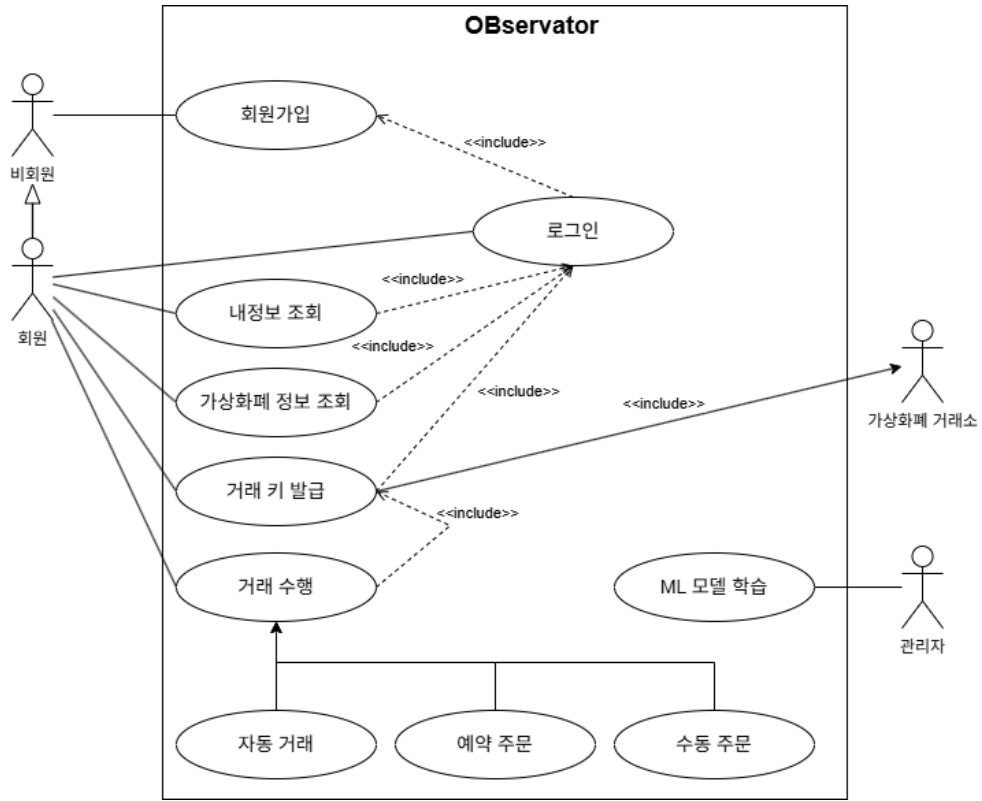


그림 7 유스케이스 다이어그램

### ● 프로젝트 설계 - API 명세서

index	대상	Method	URL	기능
1	사용자	GET	/login	로그인 페이지 불러오기
2	사용자	GET	/signup	회원가입 페이지 불러오기
3	사용자	POST	/signup	입력된 정보로 회원가입
4	사용자	POST	/logout	로그아웃
5	사용자	GET	/home	홈 화면 불러오기
6	사용자	GET	/profile	프로필 페이지 불러오기
7	사용자	POST	/profile	입력된 정보를 user directory에 저장
8	사용자	GET	/api/balances	지정한 user의 자산 정보를 불러오기
9	사용자	POST	/send-email	지정한 user의 자산 정보 이메일로 보내기
10	사용자	GET	/trade	대시보드 페이지 불러오기
11	사용자	POST	/buy	지정한 금액으로 비트코인 매도
12	사용자	POST	/sell	지정한 양으로 비트코인 매수
13	사용자	POST	/scheduleOrder	지정된 시간에 지정된 금액/양으로 비트코인 매도/매수
14	사용자	GET	/api/tradebalances	지정한 user의 자산 정보를 세부적으로 불러오기
15	사용자	PUT	/start	자동매매시작
16	사용자	PUT	/change	트래드슬드 변경
17	사용자	PUT	/stop	자동매매종료

표 2 API 명세서

● 프로젝트 설계 - 데이터 스키마

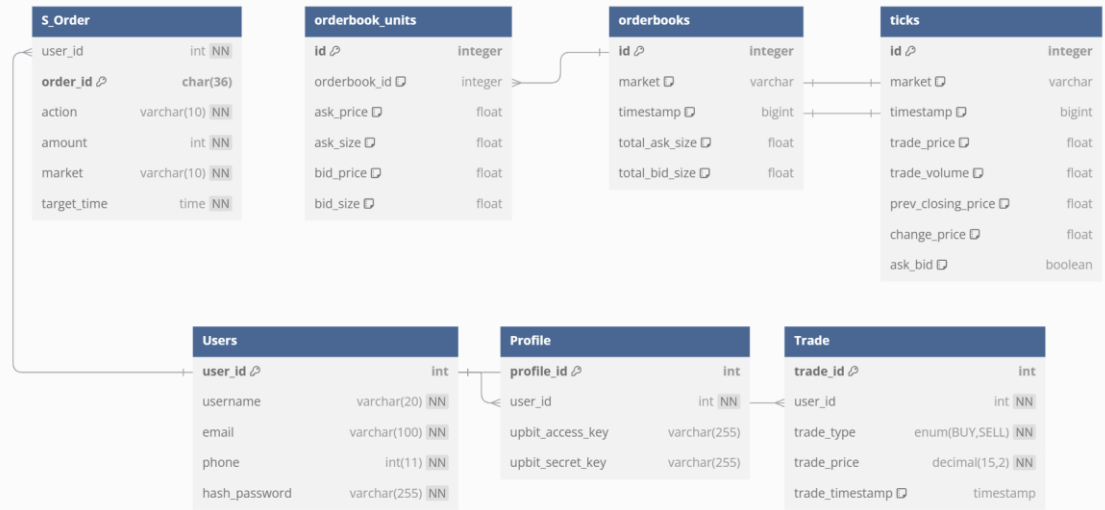


그림 8 데이터 스키마

● 프로젝트 설계 - 시스템 구성도

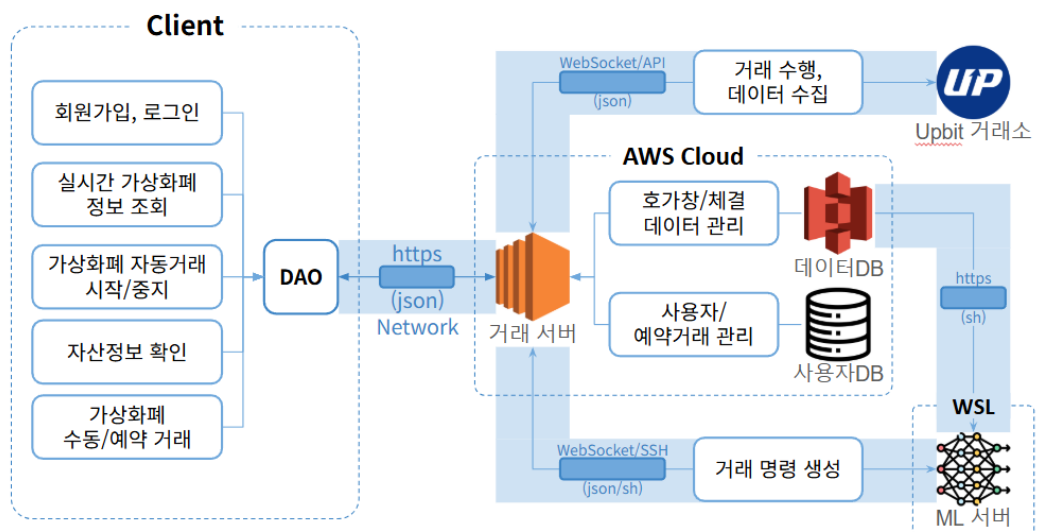


그림 9 시스템 구성도

● 프로젝트 설계 - 시퀀스 다이어그램

○ 시퀀스 다이어그램 1 - 회원가입, 거래키 발급/저장 시퀀스, ML 학습 시퀀스

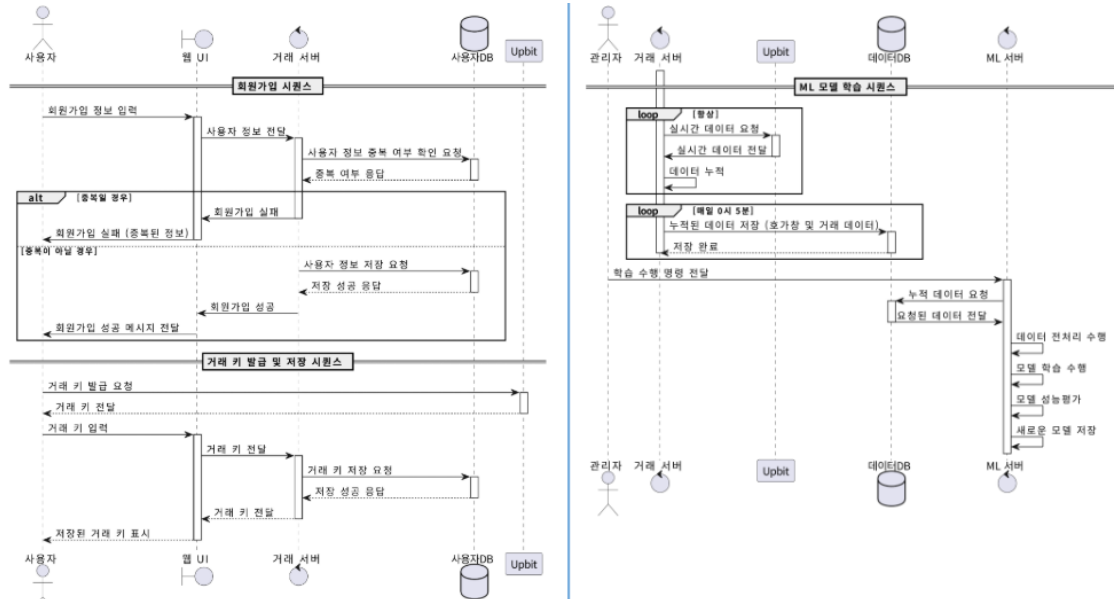


그림 10 시퀀스 다이어그램 1

○ 시퀀스 다이어그램 2 - 가상화폐 자동거래 시퀀스, 가상화폐 수동 및 예약 거래 시퀀스

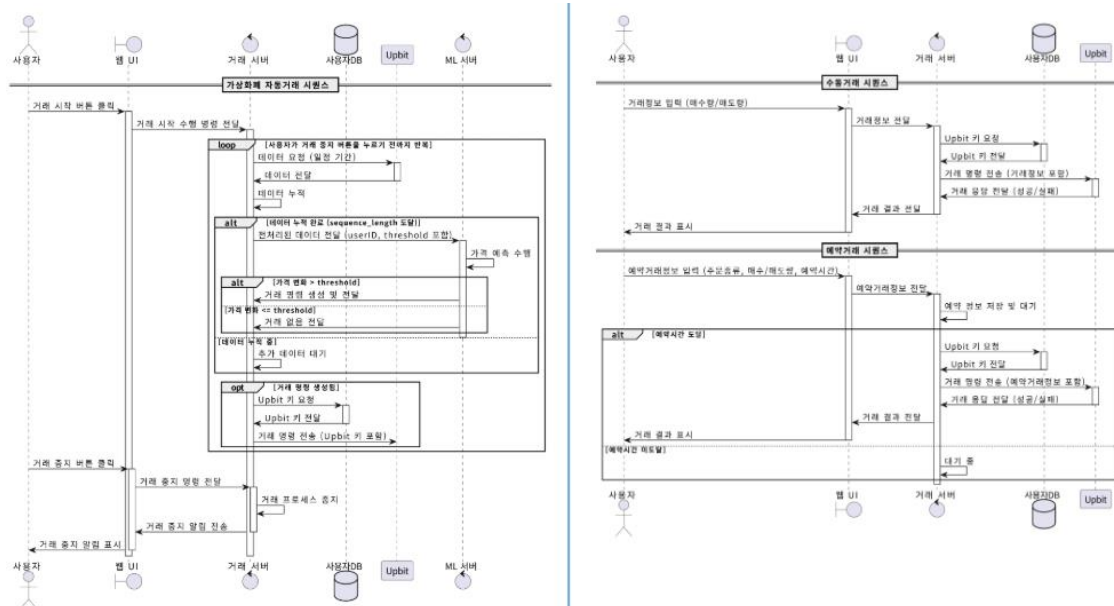


그림 11 시퀀스 다이어그램 2

### 3) 구체적인 구현방법 :

- Frontend: JavaScript, HTML, CSS를 사용하여 구현하였음. 기본 UI는 HTML과 CSS로 구성하고, 비트코인 가격 데이터를 시각적으로 보여주기 위해 Chart.js를 활용하여 라인 차트를 구현하였음. 차트는 실시간으로 업데이트되며, 5분간의 데이터만 유지되도록 설계해 성능과 가독성을 확보하였음. 실시간 데이터 처리는 초기에는 HTTP 요청 방식을 사용하였으나, API 호출 제한으로 Too Many Requests(429 오류)가 발생함. 이를 해결하기 위해 WebSocket 방식을 도입하여 데이터를 스트리밍 방식으로 실시간 수신하였고, 보다 안정적이고 효율적인 데이터 업데이트를 구현하였음.

#### 1. JavaScript, HTML, CSS 채택 이유

기술	채택 이유
JavaScript	<ul style="list-style-type: none"> <li>- 동적 웹 페이지 구현에 필수적임</li> <li>- 실시간 데이터 처리와 DOM 업데이트에 적합</li> <li>- Chart.js 등 라이브러리와의 호환성 우수</li> </ul>
HTML	<ul style="list-style-type: none"> <li>- 웹 페이지의 구조를 정의하는 기본 언어</li> <li>- 차트, 데이터, 텍스트 등을 표현하는 데 필수적임</li> </ul>
CSS	<ul style="list-style-type: none"> <li>- 웹 페이지의 스타일과 디자인을 손쉽게 설정</li> <li>- 반응형 디자인 구현으로 다양한 디바이스 지원 가능</li> </ul>

표 3 JavaScript, HTML, CSS 채택 이유

#### 2. WebSocket vs HTTP 요청 방식

항목	HTTP 요청 방식	WebSocket 방식
데이터 전송 방식	요청-응답 방식 : 클라이언트가 요청해야 서버가 응답	양방향 통신 : 서버와 클라이언트가 실시간으로 데이터 송수신 가능
실시간성	실시간 데이터 처리에 비효율적, 요청 간격이 길면 데이터 지연 발생	지속적인 연결을 통해 데이터가 즉시 전송되어 실시간 처리에 적합
네트워크 효율성	매 요청마다 연결 설정, 오버헤드 발생	초기 연결 후 데이터만 전송, 오버헤드 최소화
서버 부하	요청이 많을 경우 서버에 높은 부하 발생(Too Many Requests 문제 발생 가능)	지속적인 연결 유지로 API 호출 제한 문제 해결
사용 사례	데이터 요청 빈도가 낮은 경우 적합	실시간 데이터 스트리밍, 주식·암호화폐 가격 갱신 등 높은 빈도의 데이터 처리에 적합

표 4 WebSocket vs HTTP 요청 방식

- Backend: Spring Framework을 활용하여 백엔드의 기능적인 측면을 구현함. 서버의 경우, AWS에서 제공하는 클라우드 서비스를 적극 활용하였으며 이에 대한 구체적인 설명은 다음과 같음.

1. Upbit API를 사용하여 호가창 데이터의 경우 0.2초, 체결 데이터는 0.5초 단위로 불러와 하루 단위로 데이터를 파일 형태로 EC2 EBS 내에 저장. 각 데이터들을 수집 및 저장하는 코드는 백그라운드 프로세스에서 실행되도록 nohup 방식으로 수행됨.
2. 저장된 데이터 파일은 Linux의 crontab을 사용하여 매 0시 5분에 지정한 S3 버킷으로 자동 업로드되며, 업로드된 데이터는 ML 서버 측에서 매 0시 10분에 다운받아 ML 모델의 학습 데이터로써 사용할 수 있도록 버킷의 정책도 이와 동시에 업데이트.
3. 행여나 데이터 수집 및 업로드 과정에서 발생할 수 있는 오류를 디버깅하기 위해, 모든 코드들에 로그를 작성할 수 있도록 해줌.
4. 서버 최적화를 위해 저장된 호가창 및 체결 데이터 파일은 최근 3일까지만 서버 EBS 스토리지 내에 저장하며, 이보다 오래된 파일의 경우 이미 S3 버킷에 업로드되었기 때문에 자동 삭제함.
5. Nginx 엔진을 사용하여 기존에 할당받은 도메인(<https://observer.co.kr>)에 Springboot에서 실행된 프로젝트가 자동 연동되도록 포트 포워딩을 설정해줌.

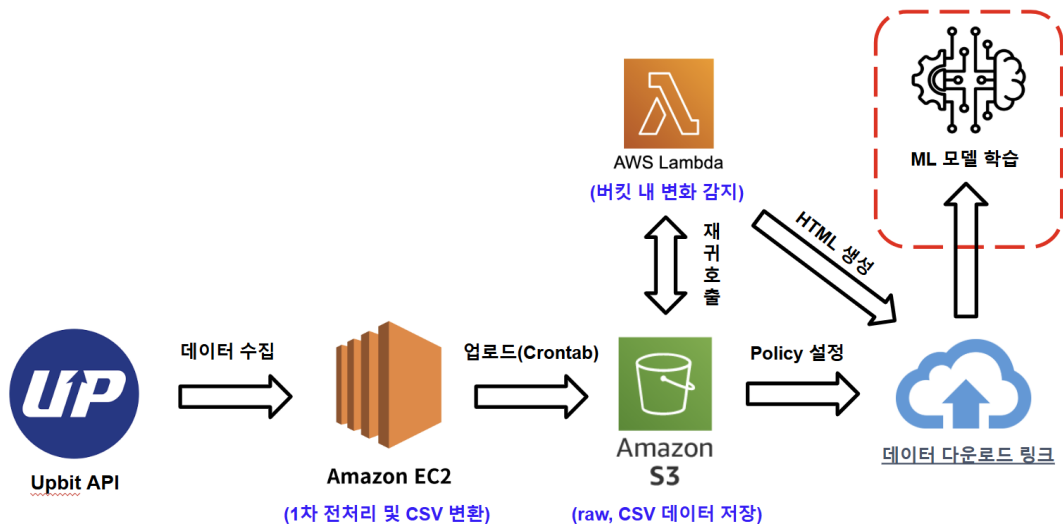


그림 12 AWS Cloud Server 구성도

- ML Model: 시계열 데이터 예측에 쓰이는 대표적인 5개의 모델 중 LSTM을 먼저 사용함. 다른 모델도 사용하며 비교해보려 했으나, GPU 성능의 한계 및 서비스 서버와의 연결에 오랜 시간 장애를 겪어 시간적 여유가 부족했음. 향후 더 개선할 예정임. 추가로, ML서버를 클라우드로 진행하려 했으나, 이용료가 너무 높은 관계로 좋은 성능의 GPU를 가진 팀원의 윈도우 데스크톱 위에 WSL2 환경으로 ML서버를 구축하여 학습/예측/통신을 수행하였음.

특징	LSTM	GRU	LightGBM	Random Forest	1D CNN
시계열 데이터 처리	우수	우수	보통	낮음	우수
비선형성 학습	우수	우수	우수	우수	우수
대용량 데이터 처리	보통	보통	우수	보통	우수
학습 속도	느림	보통	빠름	보통	빠름
실시간 예측 능력	보통	우수	우수	보통	우수
해석 용이성	낮음	낮음	낮음	우수	낮음

표 5 ML 알고리즘 선택 기준

- 액티비티 다이어그램

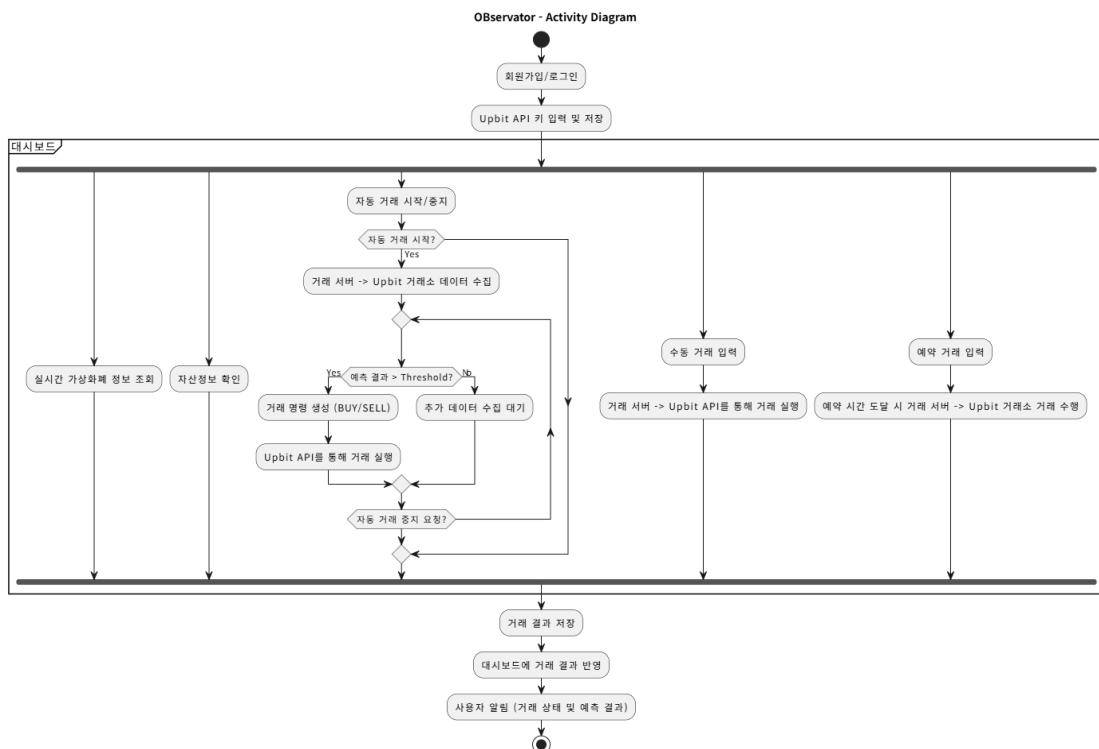
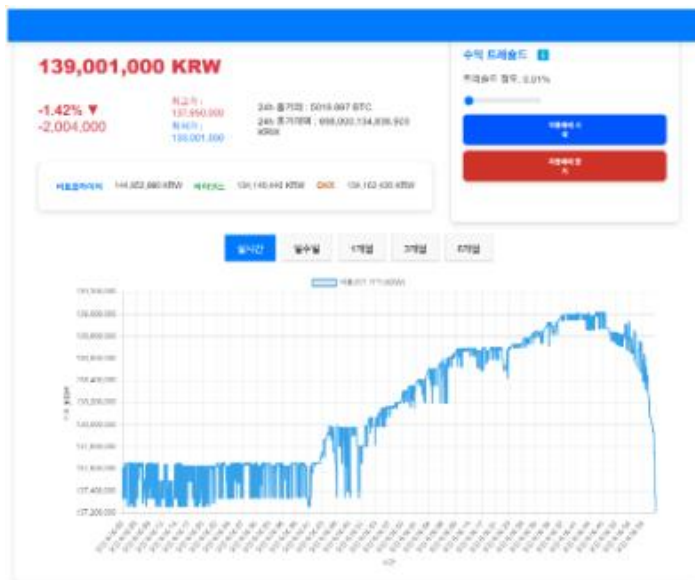


그림 13 액티비티 다이어그램

#### 4) 최종 설계 결과물의 구현 수단

- OS: Mac, Windows, Linux
- Code Editor: VS Code
- Collaboration Tool: Notion, Google Cloud, Github, Slack, Naver Mybox
- Frontend: JavaScript, HTML5, CSS3
- Backend: Java, Bash Script, Spring, Apache Maven, Apache Freemarker
- Server: Amazon EC2, nginx
- ML Model&Server: Python, Bash Script, Anaconda, PyTorch, NVIDIA CUDA, cuDNN
- DB: MySQL, Amazon S3, Amazon RDS
- DNS: Amazon Route 53, Gabia

### 3.3 결과 분석



- 서비스 실행 결과 1
- 거래 진행시간: 30분
- 수익 트레솔드: 0.05% (총 0.1%)
- 투자금액: 150만원
- 수익금액: 5308.55원
- 수익률: 0.3%
- 그래프가 안정적으로 올라가는 상태에서는 예측 정확도가 높게 나타나며 안정적으로 수익을 얻는 모습을 보여줌

내 자산		누적보통				수익 거래	
총 자산: 1557310.35 KRW		입금일자	누적일자	호가주문	비용	매수금 (KRW)	매도금 (KRW)
수익: + 5308.55 KRW		019, 110,000	0.263	36,590,875.964		매도금 (KRW)	
자산 세부정보		019, 110,000	0.081	88,490.38		매수	
매도금 (BTC): 0.77688.74 KRW		019, 110,000	0.022	5,041,541.950		매도금 (BTC)	
매수금 (BTC): 0.77688.74 KRW		019, 110,000	0.080	5,179.100		매도	
avgprice: 0.77688.74 KRW		019, 110,000	0.000	50,129.68			

그림 14 서비스 백테스

팅 결과 분석 1

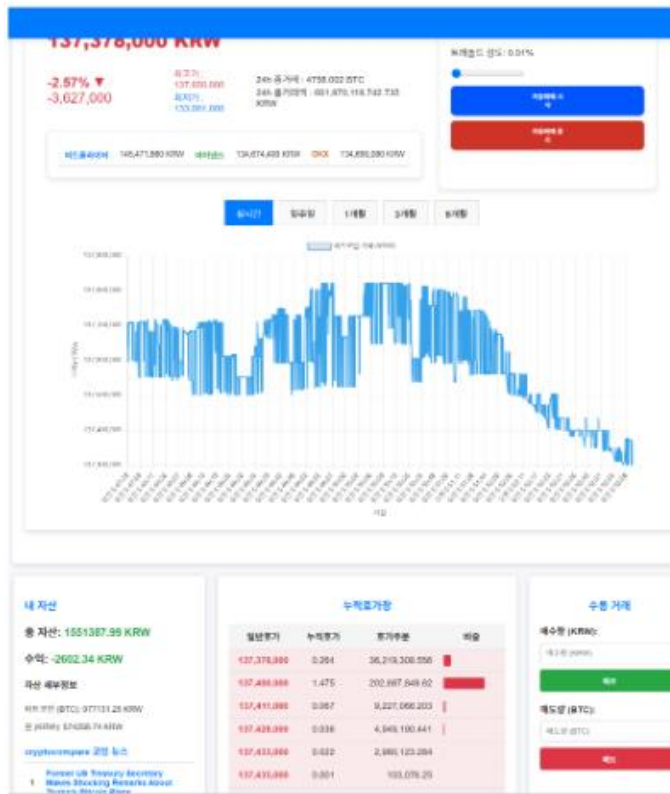


그림 15 서비스 백테스팅 결과 분석 2

- 서비스 실행 결과 2
- 거래 진행시간: 30분
- 수익 트레숄드: 0.05% (총 0.1%)
- 투자금액: 150만원
- 수익금액: -2602.34원
- 수익률: -0.11%
- 그래프가 불안정하여 요동치는 상태에서는 예측 정확도가 조금 낮게 나타나는 경향을 보임
- 이런 부분은 트레숄드를 사용자가 직접 조정하여 어느 정도 방지를 할 수 있다고 판단됨

#### 4. 기대 효과

##### 1. 경제적 측면

- 자동화된 매매로 인한 낮은 투자 손실률  
인공지능을 활용한 자동매매 시스템은 일반 투자자들도 전문적인 투자 전략을 활용할 수 있게 하며 감정에 의존한 투자에서 벗어나 손실률을 줄이고 안정적인 수익률을 기대할 수 있다.
- 투자 진입장벽 감소  
복잡한 가상화폐 시장의 데이터를 분석하고 자동 매매 서비스를 제공하여 초보 투자자도 쉽게 접근할 수 있도록 지원함. 사용자 친화적인 대시보드와 간단한 설정만으로 투자 관리를 시작할 수 있도록 함.

##### 2. 사회적 측면

- 기술 수용성 증대  
일반 대중이 인공지능 기술을 일상 생활에 활용하면서 기술에 대한 거부감이 줄어들고, 사회 전반의 기술 수용성이 높아짐. 이는 미래 기술 발전과 혁신에 대한 기반을 마련함.



## 5. 작업 분담 및 추진과정

## 1. 세부 작업 별 구성원의 역할

- 김민재: PM, ML 모델 개발
- 권일준: BE 개발(API 개발), 문서작업
- 이보성: FE 개발
- 이승호: BE 개발(서버, DB 구축), FE 보조
- 멘토 진대한: 코드 Review, ML 모델 Review

## 2. 간트 차트

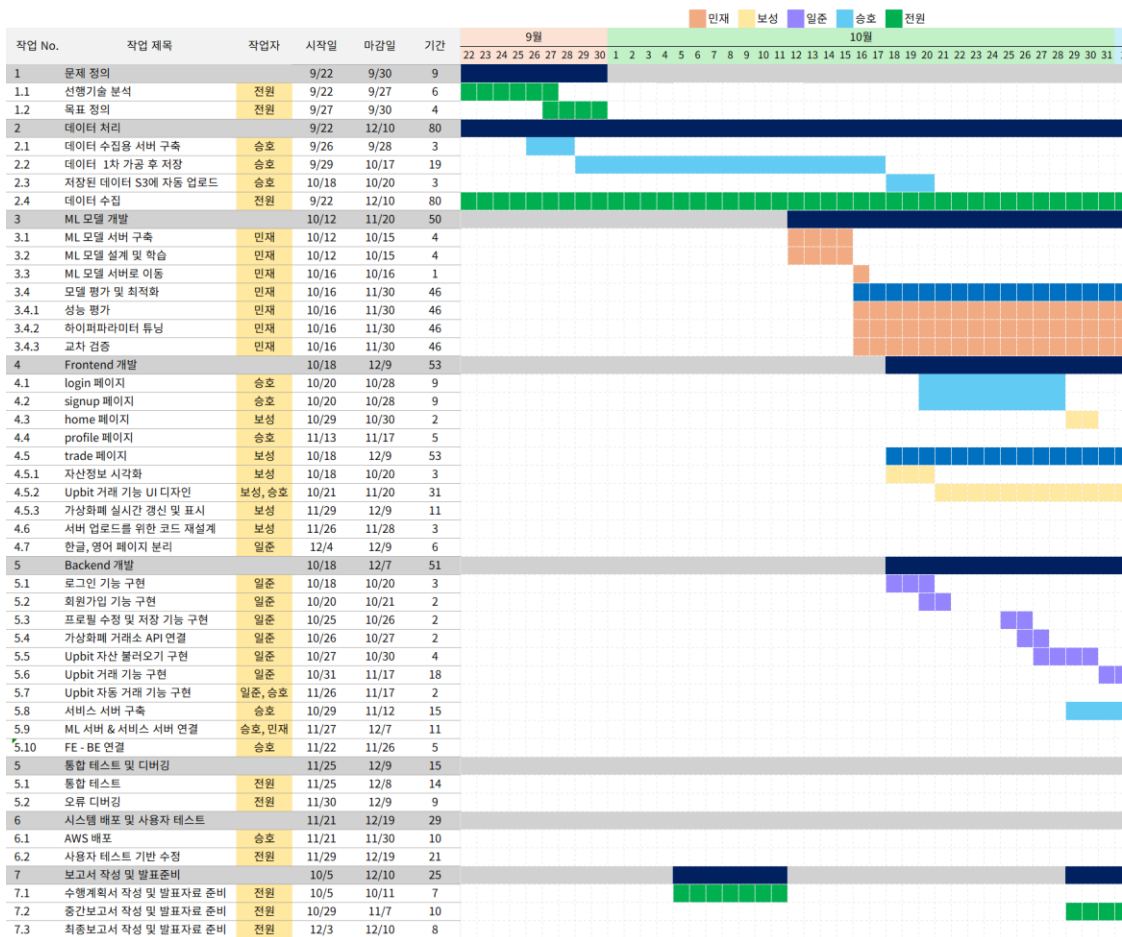

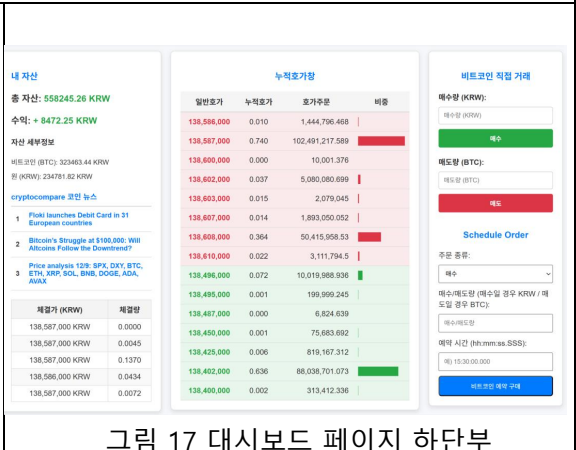



표 6 간트차트 1



표 7 간트차트 2

<p><b>6. 산학 협력</b></p>	<p>구성원의 역할</p> <ul style="list-style-type: none"> <li>● 김민재: PM, ML 모델 개발</li> <li>● 권일준: BE 개발(API 개발), 문서작업</li> <li>● 이보성: FE 개발</li> <li>● 이승호: BE 개발(서버, DB 구축), FE 보조</li> </ul> <p>1차 멘토링: 호가창 데이터 및 체결 데이터의 소개 및 활용 방법 설명, 호가창 데이터 수집 방법에 대한 정보, 보안 적용 방법</p> <p>2차 멘토링: 예측 정확도 향상을 위한 호가창 데이터 input에 대한 조언, 여러 매매 전략에 대한 소개</p> <p>3차 멘토링: 과적합을 방지하기 위한 호가창 데이터 input에 대한 조언</p>
<p><b>7. 참고 문헌</b></p>	<p>이상덕, 주식 자동 매매 시스템 및 방법, 10-2019-0104919, filed July 11, 2019, issued November 12, 2020</p>

<b>8. R&amp;D 성과</b>	특허명세서 제출 완료.
<b>9. 결과물 사진</b>	<div data-bbox="279 365 850 869">  <p>그림 16 대시보드 페이지 상단부</p> </div> <div data-bbox="850 365 1428 869">  <p>그림 17 대시보드 페이지 하단부</p> </div> <div data-bbox="279 869 850 965">                 대시보드 페이지의 상단부 사진             </div> <div data-bbox="850 869 1428 965">                 대시보드 페이지의 하단부 사진             </div>
<b>10. 첨부</b>	<p>1. 팀원 및 멘토가 함께 찍은 사진 1장 jpg</p> <div data-bbox="292 1016 1412 1852">  </div> <p>그림 18 회의 사진</p>

## 2. 작품설명용 사진 1~2장 jpg (ex. 4.개념설계.jpg)

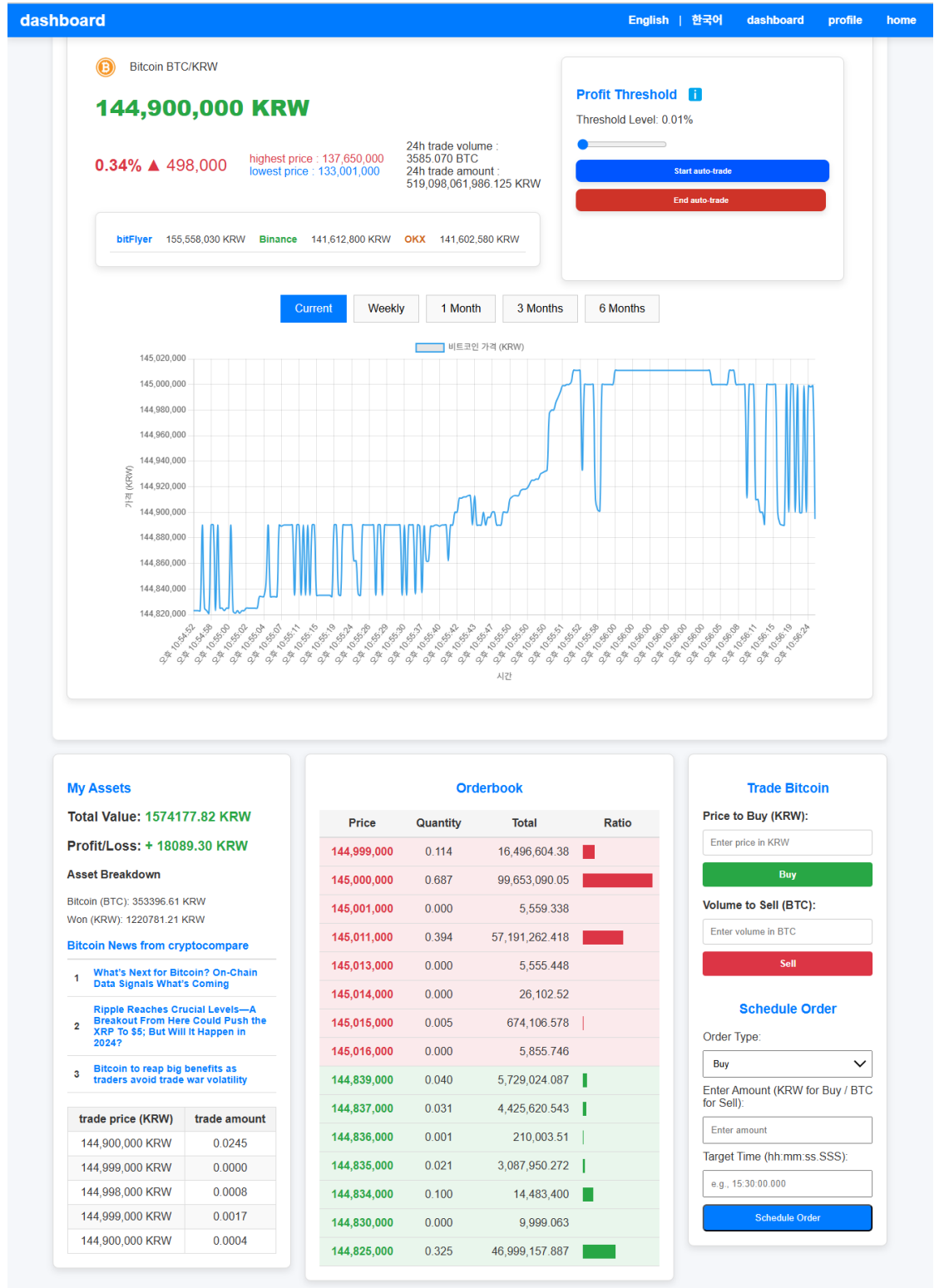


그림 19 대시보드 전체 페이지

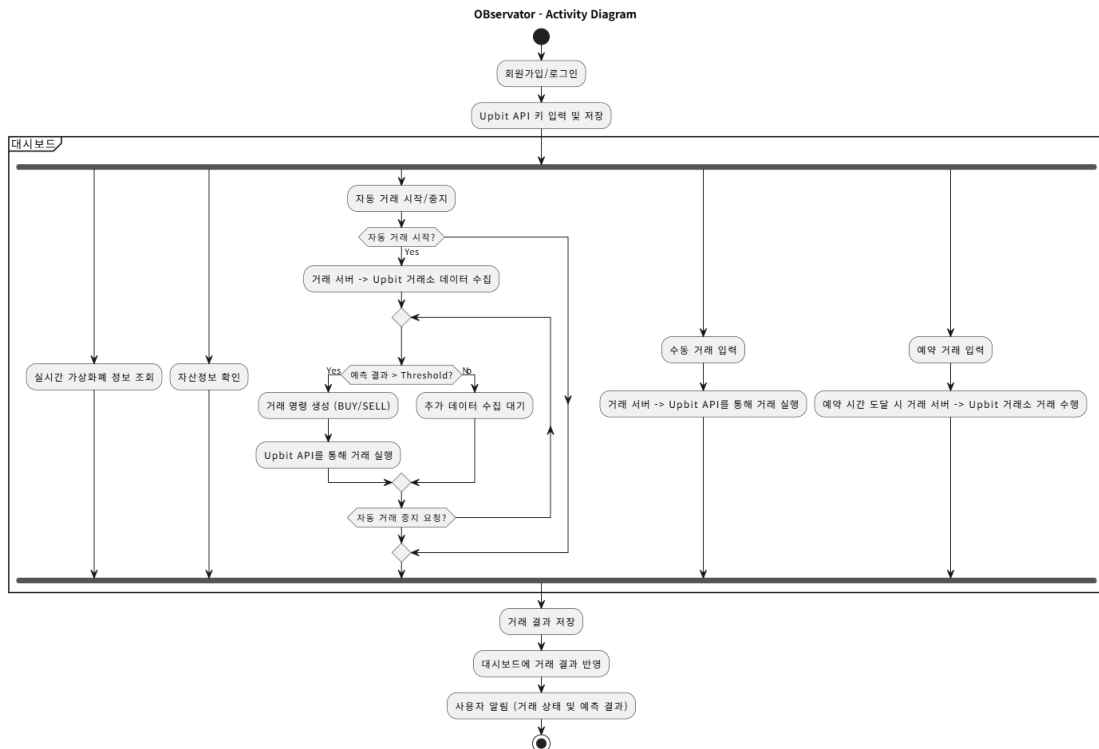


그림 20 액티비티 다이어그램

### 3. 시연동영상 링크 (실시간 시연을 통한 최종발표 백업용)

시연영상: <https://www.youtube.com/watch?v=VCR4LiKz62M>

거래 명령 생성 확인 영상: [https://www.youtube.com/watch?v=\\_7QG4PB51iY&t=0s](https://www.youtube.com/watch?v=_7QG4PB51iY&t=0s)

### 4. 코드 링크

Github: <https://github.com/CSID-DGU/2024-2-SCS4031-Happy-7>

Notion: <https://www.notion.so/0732b7d4e91448f48d157c8a2f3b6249>

### 5. 최종발표 질의사항 답변

Q1. 자동 매매 서비스는 소수의 사용자일 때는 효과적일 수 있지만 다수의 사용자가 동일한 전략을 사용할 경우 의미가 퇴색될 것 같은데 이 문제를 어떻게 해결할 것인가?

A1: 평소 비트코인의 거래량을 고려하면 다수의 사용자가 사용한다고 해도 비트코인의 시세를 바꿀 수 있을 정도의 변화가 생기지 않을 것이라고 생각한다.

Q2: 기존의 비트코인 자동매매 플랫폼과 차이점이 있는지? 확연하게 구별가능한 표가 있었으면 좋겠다.

A2: 중간 발표 때 선행 사례 비교를 위해 만든 표를 참고 바람.

요소	OBservator	CCXT	pyupbit	gpt-bitcoin	BAMOWL
자동화 수준	조건부 자동화 지원	매매 API 제공	매매 API 제공	AI 기반의 제한적인 자동화	자동화 매매 및 백테스팅 결합
사용자 설정 개입	사용자가 Threshold 직접 설정 가능	X	X	기본 설정 가능	다양한 검증 가능
실시간 데이터	실시간 호가창 데이터 수집	제공	제공	제공	수집 및 제공
데이터 분석	실시간 분석 가능	수동	수동	AI 기반 실시간 분석	딥러닝 기반 예측 및 백테스팅
거래소 지원 범위	다양한 가상화폐 거래소 지원	전세계 주요 거래소 지원	업비트 지원	제한적 거래소 지원	업비트 지원
안정성 및 성능	높은 처리량으로 인해 과부하 시 지연 가능성 존재	X	X	중간 수준	백테스팅 기반 성능 최적화

표 8 선행 사례 비교

Q3: Threshold가 정확히 어떤 역할을 하는지가 궁금하다. 변동성이랑 정확히 무슨 관련이 있고, 대시보드 페이지에 정보를 요소별로 어떤식으로 보여주게 하는지 원리가 궁금하다.

A3: 수익 트레슬드는 거래를 실행할 최소 예측 수익률이다. 예를 들어 0.01%로 threshold를 설정하면 예측 수익이 현재 가격 대비 0.01% + 거래 수수료(0.05%) 이상일 때 거래가 수행된다. 변동성이 작을 때 예측 수익이 threshold 기준보다 낮을 경우 거래가 이루어지지 않는다. 사용자는 이 기준치를 조절 할 수 있는것이다.

Q4: 왜 바이낸스 거래소 API 안쓰고 업비트 썼는지?

A4: 바이낸스는 2021년 이후 한국을 위한 서비스를 모두 삭제했다. 현재 원화를 이용할 수도 없기 때문에 한국인의 접근성이 좋은 거래소중 한국에서 가장 잘 알려진 업비트를 사용하기로 했다.

Q5: 거의 1분에 5번 이상 거래되는 등 짧은 시간동안 너무 많은 거래가 체결되어 예측의 의미가 있는건지, 예측이 체결수수료를 넘길 확률이 너무 낮은거 아닌가?

A5: 가상화폐는 일반 주식에 비해 변동성이 큰 편이다. 변동성이 threshold 기준보다 높을 경우 거래가 이루어지지 않는다. 사용자는 이 기준치를 조절 하여 거래 정도를 어느정도 제어가 가능하다. 그리고 기본적으로 (threshold + 거래 수수료(0.05%))의 수익이 예측될때만 거래 명령을 생성한다.

Q6: 백테스팅 결과는 없는지? 30분 이용으로 0.3퍼센트 정도 수익이라는 결과만 보여주셨는데, 성능 검증 시간과 횟수가 너무 적어서 모델 성능의 설득력이 부족한 것 같다.

A6: 거래를 지속적으로 수행할 시 서버 비용이 많이 발생해 백테스팅은 짧게 수행하였다.