

캡스톤디자인 중간보고서

제 목	국문	AI기반 가상화폐 자동매매 웹사이트		
	영문	Coin AI Trader(C.A.T)		
진 행 상 황	중요 마일스톤	<p>기능 요구사항에 대한 중요 마일스톤</p> <ul style="list-style-type: none">• 코인 차트 제공 : 바이낸스 과거 데이터를 통해 코인차트 코드 작성• 가상화폐 백테스팅 : 바이낸스 과거 데이터를 기반으로 백테스팅 기능• 자동매매 : 지표, 머신러닝 기능을 통한 자동매매 기능• 위의 기능을 사용하기 위한 데이터 관리 : 데이터베이스 최적화 <p>성능 요구사항에 대한 중요 마일스톤</p> <ul style="list-style-type: none">• 많은 과거 데이터로 인한 성능저하 발생 : 캐싱, 최적화를 통해 개선• 수익성 보장 : 수익성을 보장하는 전략 개발		
	진행상황	<ul style="list-style-type: none">• 바이낸스 API를 이용해 2017~2024 데이터 수집완료• 백테스팅 결과를 통해 수익을 내는 전략 개발완료.• UI 프로토타입 작성완료• AI를 이용한 추가 전략 개발 예정		
산출물	요구사항 정의서(별첨 1), 중간보고서(별첨 2)			
팀 구성원	학년	학 번	이 름	연락처(전화번호/이메일)
	4학년	20191748	이지현	010-3741-5330
	4학년	20191738	오진표	010-3281-3936
	4학년	20191795	한준서	010-2455-1959
<p>컴퓨터공학과와 프로젝트 관리규정에 따라 다음과 같이 요구사항 정의서와 중간보고서를 제출합니다</p> <p>2024 년 05월 02일</p> <p>책임자 : 이지현(인) 지도교수 : 황경호(인)</p>				

[별첨1]

프로젝트명 : AI기반 가상호폐 자동매매 웹사이트 개발

소프트웨어 요구사항 정의서

Version 1.0

개발 팀원 명(팀리더):이지현
오진표
한준서

대표 연락처:010-3741-5330

e-mail: 20191748@edu.hanbat.ac.kr

목차

1. 개요
2. 시스템 장비 구성요구사항
3. 기능 요구사항
4. 성능 요구사항
5. 인터페이스 요구사항
6. 데이터 요구사항
7. 테스트 요구사항
8. 보안 요구사항
9. 품질 요구사항
10. 제약 사항
11. 프로젝트 관리 요구사항

요구사항 정의서에 사용되는 양식 설명

요구사항 고유번호(ID): 제안요청서에 정의된 요구사항에 대해 계약, 사업수행, 사업완료 및 검수까지 변경, 삭제, 수정 여부에 대한 추적관리를 위해 고유의 번호를 부여하도록 한다.

요구사항 구분 및 ID부여 규칙

요구사항 구분		ID 부여 규칙
시스템 장비 구성 요구사항	Equipment Composition Requirement	ECR-000
기능 요구사항	System Function Requirement	SFR-000
성능 요구사항	Performance Requirement	PER-000
인터페이스 요구사항	System Interface Requirement	SIR-000
데이터 요구사항	Data Requirement	DAR-000
테스트 요구사항	Test Requirement	TER-000
보안 요구사항	Security Requirement	SER-000
품질요구사항	Quality Requirement	QUR-000
제약사항	Constraint Requirement	COR-000
프로젝트 관리 요구사항	Project Mgmt. Requirement	PMR-000
프로젝트 지원 요구사항	Project Support Requirement	PSR-000

요구사항 세부내용 작성표 양식 및 항목설명

요구사항 고유번호		(설명) 요구사항 추적관리를 위해 독립적인 고유번호(ID) 부여
요구사항 명칭		(설명) 요구사항 명칭을 작성함
요구사항 분류		(설명) 요구사항 분류기준에 따른 분류를 기입
요구사항 상세 설명	정의	(설명) 요구사항 정의
	세부 내용	(설명) 요구사항 구체적인 세부 내용을 설명
산출정보		(설명) 해당기능을 통해 산출되는 결과물 혹은 정보를 표기
관련 요구사항		(설명) 정의된 요구사항과 관련된 요구사항에 대해 기술
요구사항 출처		(설명) 기능 도출내용에 대한 출처(source) 표기

1. 시스템 개요

1) 목표

- 안정적인 수익성을 보장하는 자동매매 웹사이트 구현

다양한 전략과 과거 데이터를 통한 자동매매 알고리즘을 통하여 수익성을 보장하는 자동매매 서비스를 제공한다.

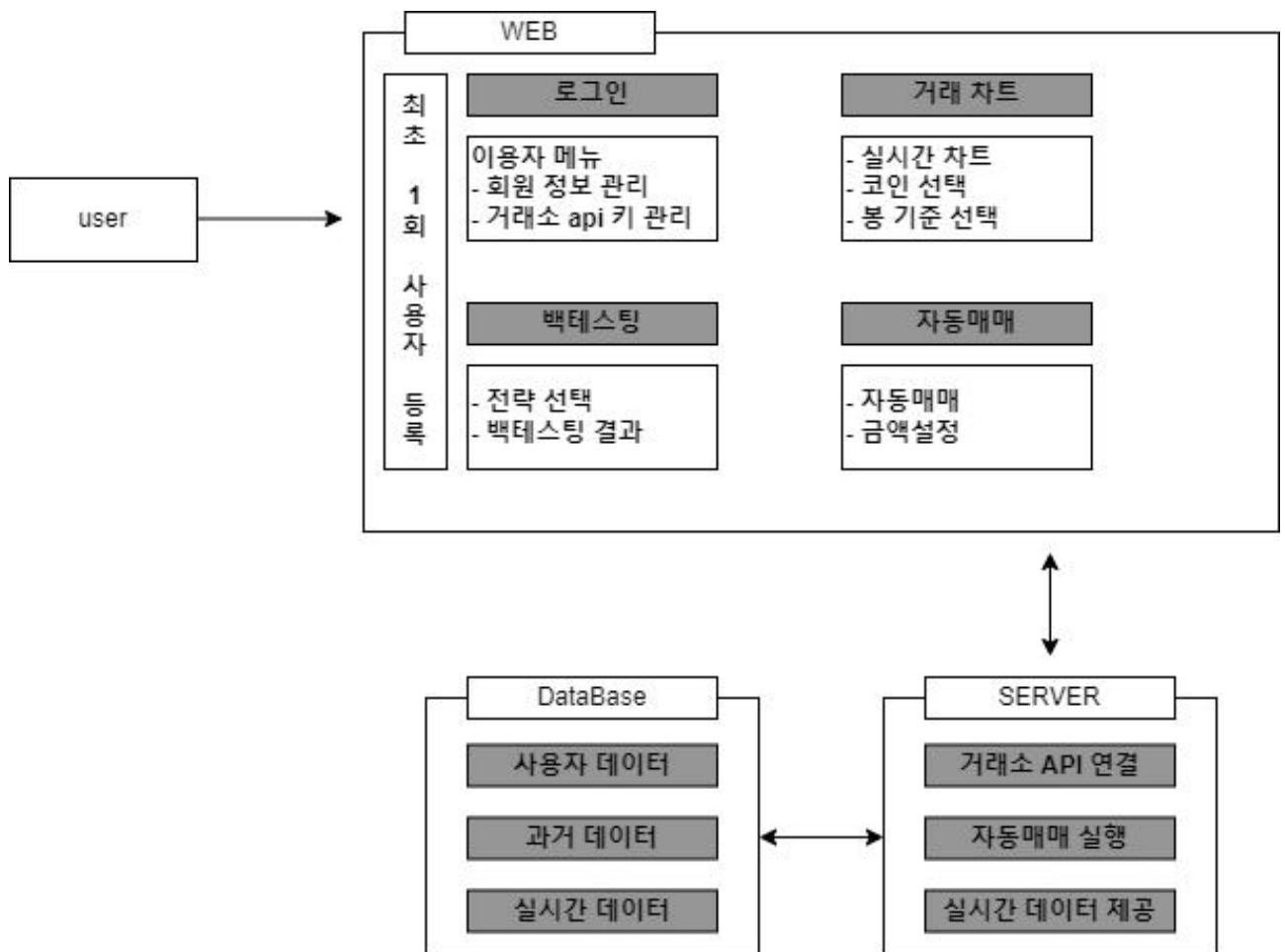
- 실시간 가상화폐의 거래차트 제공

거래소 API를 통한 실시간 거래상황을 전달받아 차트화 하여 사용자에게 제공한다.

- 백테스팅 서비스 제공

과거 데이터를 통한 가상 매매 서비스를 제공한다.

2) 시스템 개요



2. 시스템 장비 구성요구사항

요구사항 고유번호		ECR-001		
요구사항 명칭		웹 서버		
요구사항 분류		소프트 웨어	응락수준	필수
요구사항 상세설명	정의	- 웹 서버		
	세부 내용	<div>- 장비 품목 : AWS EC2</div> <div>- 장비 수량 : 1식</div> <div>- 장비 기능 : 시스템 구축</div> <div>- 장비 성능 및 특징 : 클라이언트로부터의 HTTP 요청을 받아들이고, 정적 및 동적 웹 페이지를 제공하는 소프트웨어 서비스</div>		

3. 기능 요구사항

요구사항 고유번호		SFR-001		
요구사항 명칭		코인 차트 제공		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	- 바이낸스 과거 데이터를 통해 코인차트 코드를 작성해야 함		
	세부 내용	<ul style="list-style-type: none"> • 사용자가 선택한 가상화폐의 과거 가격 데이터를 조회하여 시간에 따른 가격 변화를 그래픽으로 보여주는 차트 기능 • 차트는 다양한 시간 단위로 데이터를 표시할 수 있어야 하며, 사용자가 원하는 시간 범위를 선택할 수 있도록 하는 기능 		

요구사항 고유번호		SFR-002		
요구사항 명칭		백테스팅		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	- 과거 데이터를 이용하여 가상화폐 거래 전략의 성능을 테스트하는 기능 제공		
	세부 내용	<ul style="list-style-type: none"> • 사용자가 입력한 거래 전략에 대해 과거 데이터를 기반으로 시뮬레이션을 수행하여 거래 결과를 분석하는 백테스팅 기능 • 백테스팅 결과는 수익률, 손익률, 거래 횟수 등의 지표로 제공되어야 하며, 그래픽 차트를 통해 시각화된 결과를 사용자에게 제공 		

요구사항 고유번호		SFR-003		
요구사항 명칭		자동매매		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	- 지표와 머신러닝 알고리즘을 활용하여 자동으로 가상화폐를 매매하는 기능 제공		
	세부 내용	<ul style="list-style-type: none"> 매매 전략에 따라 자동으로 거래를 수행하는 자동매매 기능 시스템은 사용자의 매매 전략에 따라 실시간으로 시장 데이터를 분석하고, 최적의 매매 시점을 판단하여 거래를 실행 자동매매 기능은 사용자가 설정한 조건에 따라 자동으로 가상화폐를 매매하고, 매매 이력을 기록하고 관리 		

요구사항 고유번호		SFR-004		
요구사항 명칭		데이터 관리		
요구사항 분류		기능	응락수준	필수
요구사항 상세설명	정의	- 시스템에서 사용되는 데이터를 관리하고 최적화하기 위한 데이터베이스 관리 기능		
	세부 내용	<ul style="list-style-type: none"> 사용자가 조회하는 데이터를 데이터베이스에 효율적으로 저장하고 관리하기 위해 데이터베이스 구조를 최적화 데이터베이스 관리 기능은 데이터의 삽입, 갱신, 삭제 등의 작업을 지원하고, 데이터의 일관성과 무결성을 유지 		

4. 성능 요구사항

요구사항 고유번호		PER-001		
요구사항 명칭		정확하고 빠른 API전송		
요구사항 분류		성능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 매수,매도 시 빠른 주문		
	세부 내용	- 전략에 따라 빠르게 주문을 해야하므로 사용자 웹사이트가 request를 보낼 시 1s 이내로 response를 해야 함.		

요구사항 고유번호		PER-002		
요구사항 명칭		성능 저하에 대처		
요구사항 분류		성능 요구사항	응락수준	필수
요구사항 상세설명	정의	- 차트 로딩		
	세부 내용	- 많은 데이터를 차트화 해야 하므로 발생하는 느린 로딩을 캐싱, 데이터 최적화를 통해 개선 - 차트화 하는 중 시간이 걸릴 경우 차트 부분 UI에 Loading 화면 출력		

5. 인터페이스 요구사항

요구사항 고유번호		SIR001		
요구사항 명칭		웹 사용법		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세설명	정의	-웹 사용법 제공		
	세부내용	- 사용자가 별도의 교육을 받지 않더라도 웹 도움말을 이용하여 사용자 기능을 이용할 수 있어야 함		
주석				
요구사항 출처				

요구사항 고유번호		SIR002		
요구사항 명칭		다양한 이용 보장		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세설명	정의	- 웹, 모바일 서비스 제공		
	세부내용	- 반응형 웹 사이트 개발을 통해 웹 뿐만 아니라 모바일로도 서비스를 제공 받을 수 있어야 함.		
주석				
요구사항 출처				

요구사항 고유번호		SIR003		
요구사항 명칭		대시보드 제공		
요구사항 분류		인터페이스	응락수준	필수
요구사항 상세설명	정의	- 자동매매, 백테스팅 대시보드		
	세부내용	- 자동매매, 백테스팅의 결과를 효율적인 대시보드를 통해 제공되어야 함.		
주석				
요구사항 출처				

6. 데이터 요구사항

요구사항 고유번호	DAR-001		
요구사항 명칭	검증되고 사용율이 높은 DBMS 사용		
요구사항 분류	데이터 요구사항	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 검증된 DBMS를 사용해야 함 - PostgreSQL 채택 		

요구사항 고유번호	DAR-002		
요구사항 명칭	초기 데이터 구축		
요구사항 분류	데이터	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 바이낸스 API를 통해 과거 2017~2024 데이터 확보 		

요구사항 고유번호	DAR-003		
요구사항 명칭	데이터 처리 및 정제		
요구사항 분류	데이터 요구사항	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 바이낸스 API 사용을 위한 형식을 맞춰 데이터를 정제하여 처리해야함 		

7. 테스트 요구사항

요구사항 고유번호	TER-001		
요구사항 명칭	API 테스트		
요구사항 분류	단위 테스트	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 요청에 따른 매수,매도 수행을 하는지 테스트 해야함 		

요구사항 고유번호	TER-002		
요구사항 명칭	전략 테스트		
요구사항 분류	단위 테스트	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 수익성이 보장되는 전략이 맞는지 테스트 해야함 		

요구사항 고유번호	TER-003		
요구사항 명칭	주문 테스트		
요구사항 분류	단위 테스트	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 전략에 따른 매수,매도 주문을 실행 하는지 확인 해야함 		

요구사항 고유번호	TER-004		
요구사항 명칭	웹사이트 테스트		
요구사항 분류	단위 테스트	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 웹사이트의 요청에 따른 API를 실행 하는지 테스트 해야함 		

요구사항 고유번호	TER-005		
요구사항 명칭	통합 테스트		
요구사항 분류	통합 테스트	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 웹사이트에서 실시간 데이터 차트를 제공하는 것을 테스트 - 웹사이트에서 요청을 기반으로 작동하는지 테스트 - API 요청이 잘 전달되는지 테스트 - 모든 테스트에서의 오류가 발생하는지를 테스트 		

8. 보안 요구사항

요구사항 고유번호	SER-001		
요구사항 명칭	접근 권한 관리		
요구사항 분류	보안 요구사항	응락수준	선택
요구사항 상세설명	<ul style="list-style-type: none"> - 사용자별로 데이터베이스 접근 권한을 부여하고 관리해야 함. - 관리자, 운영자, 일반 사용자 등의 권한을 구분하여 접근 제어를 실시해야 함. 		

요구사항 고유번호	SER-002		
요구사항 명칭	사용자 정보 보안		
요구사항 분류	보안 요구사항	응락수준	선택
요구사항 상세설명	<ul style="list-style-type: none"> - 사용자의 정보(API key, 비밀번호) 등은 암호화 되어 관리 해야함. 		

요구사항 고유번호	SER-003		
요구사항 명칭	보안 인증		
요구사항 분류	보안 요구사항	응락수준	선택
요구사항 상세설명	<ul style="list-style-type: none"> - 사용자 인증을 위해 안전한 인증 수단을 활용해야 함. (예: 사용자 이름 및 비밀번호, 이중 인증) - 강력한 비밀번호 정책을 적용하여 데이터베이스 접근을 보호해야 함. 		

9. 품질 요구사항

요구사항 고유번호		QUR-001		
요구사항 명칭		안정성 보장		
요구사항 분류		품질 요구사항	응락수준	필수
요구사항 상세설명	정의	시스템 안정성 보장		
	세부 내용	<ul style="list-style-type: none"> - 시스템의 안정성을 높이기 위해 소프트웨어 버그 및 오류를 최소화해야 함. - 안정적인 운영을 위해 예상되는 장애에 대비한 복구 기능을 제공해야 함. - 시스템 장애 발생 시에도 중요한 데이터의 손실을 방지하기 위한 자동 백업 및 복구 시스템을 구축해야 함. 		
요구사항 고유번호		QUR-002		
요구사항 명칭		성능 최적화		
요구사항 분류		품질 요구사항	응락수준	필수
요구사항 상세설명	정의	시스템 성능 최적화		
	세부 내용	<ul style="list-style-type: none"> - 스템이 대량의 데이터를 신속하게 처리할 수 있도록 성능을 최적화해야 함. - 응답 속도를 개선하기 위해 사용자 인터페이스의 레이아웃 및 디자인을 최적화해야 함. - 사용자가 요청한 기능에 대한 응답 시간을 최소화하여 작업 효율성을 높여야 함. 		
요구사항 고유번호		QUR-003		
요구사항 명칭		보안 강화		
요구사항 분류		품질 요구사항	응락수준	필수
요구사항 상세설명	정의	보안 강화		
	세부 내용	<ul style="list-style-type: none"> - 사용자의 자산과 직결되는 정보를 관리해야 하므로 데이터베이스 암호화 기능을 필히 적용해야 함. - 시스템에 대한 무단 접근을 방지하기 위해 사용자 인증 및 접근 제어 시스템을 강화해야 함. - 보안 취약점을 식별하고 해결하기 위한 정기적인 보안 검토 및 강화 작업을 수행해야 함. 		

10. 제약 사항

요구사항 고유번호	COR-001		
요구사항 명칭	시스템 개발		
요구사항 분류	제약사항	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 프로그래밍 언어 : Python, JavaScript - 사용 라이브러리 : React , Spring - 개발도구 : intellij , VScode 		

11. 프로젝트 관리 요구사항

요구사항 고유번호	PMR-001		
요구사항 명칭	산출물 관리		
요구사항 분류	프로젝트 관리 요구사항	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - Github를 통해 소스코드를 관리 함. 		

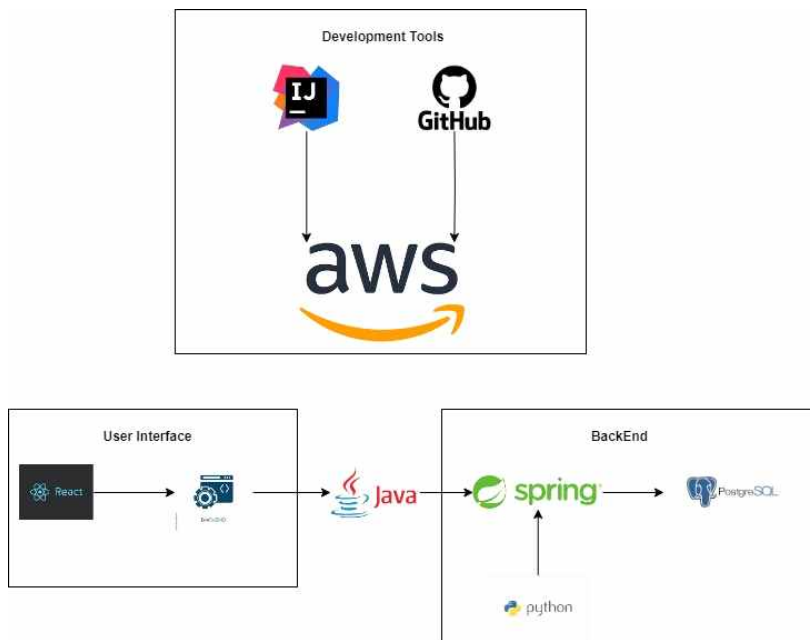
요구사항 고유번호	PMR-002		
요구사항 명칭	프로젝트 수행 조직		
요구사항 분류	프로젝트 관리 요구사항	응락수준	필수
요구사항 상세설명	<ul style="list-style-type: none"> - 이지현 : UI개발 - 오진표, 한준서 : 데이터 베이스, 서버 , 바이낸스 API 통합 - 공통 : 데이터 분석, 매매 알고리즘 분석 및 개발 		

[별첨2]

중간보고서

1. 요구사항 정의서에 명시된 기능에 대하여 현재까지 분석, 설계, 구현(소스코드 작성) 및 테스트한 내용을 기술하시오.

- 시스템 구성도



1. 프론트엔드

- React.js
- 사용자 인터페이스를 구축하기 위해 React.js를 사용함

2. 백엔드

- Java, Spring Framework
- 백엔드 서버를 구축하기 위해 Java와 Spring Framework를 사용함

3. 데이터 베이스

- PostgreSQL

5. 형상관리

- Git, GitHub
- Git을 통해 변경 사항을 추적하고, GitHub를 통해 협업하여 소스 코드를 관리함

6. 통합 개발환경

- IntelliJ, VScode

7. 언어

- python
- 거래소 API를 사용하고 자동매매 알고리즘을 실행이 효율적인 언어인 Python을 사용.

- 진행 상황

1) 과거 데이터 수집 및 API 동작여부 확인

과거 데이터 수집

binance_data_2017

2024-02-02 오전 2:42

Microsoft Excel ...

18,210KB

binance_data_2018

2024-02-02 오전 2:49

Microsoft Excel ...

59,235KB

binance_data_2019

2024-02-02 오전 2:52

Microsoft Excel ...

59,851KB

binance_data_2020

2024-02-02 오전 2:55

Microsoft Excel ...

61,030KB

binance_data_2021

2024-02-02 오전 3:01

Microsoft Excel ...

62,347KB

binance_data_2022

2024-02-02 오전 3:05

Microsoft Excel ...

61,424KB

binance_data_2023

2024-02-02 오전 3:05

Microsoft Excel ...

59,177KB

timestamp	open	high	low	close	volume	close_time	asset_vol	base_vol	quote_vol	ignore
2017-08-17 04:00:00	4261.4800	4261.4800	4261.4800	4261.4800	1.7751830	1.5E+12	7564.9068	3	0.0751830	320.390850
2017-08-17 04:01:00	4261.4800	4261.4800	4261.4800	4261.4800	0.0000000	1.5E+12	0.0000000	0	0.0000000	0.0000000
2017-08-17 04:02:00	4280.5600	4280.5600	4280.5600	4280.5600	0.2610740	1.5E+12	1117.5429	2	0.2610740	1117.54290
2017-08-17 04:03:00	4261.4800	4261.4800	4261.4800	4261.4800	0.0120080	1.5E+12	51.171851	3	0.0120080	51.1718510
2017-08-17 04:04:00	4261.4800	4261.4800	4261.4800	4261.4800	0.1407960	1.5E+12	599.99933	1	0.1407960	599.999330
2017-08-17 04:05:00	4261.4800	4261.4800	4261.4800	4261.4800	0.0000000	1.5E+12	0.0000000	0	0.0000000	0.0000000
2017-08-17 04:06:00	4261.4800	4261.4800	4261.4800	4261.4800	0.0000000	1.5E+12	0.0000000	0	0.0000000	0.0000000
2017-08-17 04:07:00	4261.4800	4261.4800	4261.4800	4261.4800	0.0000000	1.5E+12	0.0000000	0	0.0000000	0.0000000

바이낸스 API 확인														
<pre> from binance.client import Client from binance.exceptions import BinanceAPIException import os api_key = os.getenv('Binan_API_KEY') api_secret = os.getenv('Binan_SECRET_KEY') client = Client(api_key, api_secret) try: # 계좌 정보 받아오기 account_info = client.get_account() # 잔고 정보 출력 for balance in account_info['balances']: if float(balance['free']) > 0: print(f"{balance['asset']}: {balance['free']}") except BinanceAPIException as e: print(f"An error occurred: {e.status_code} - {e.message}") </pre>	<table> <tr> <th>Coin</th><th>Amount</th><th>Available</th></tr> <tr> <td> BTC Bitcoin</td><td>0.0001998 \$8.58</td><td>0.0001998</td></tr> <tr> <td> USDT TetherUS</td><td>0.7204861 \$0.7204861</td><td>0.7204861</td></tr> <tr> <td> XRP Ripple</td><td>0.435967 \$0.21920421</td><td>0.435967</td></tr> </table>	Coin	Amount	Available	BTC Bitcoin	0.0001998 \$8.58	0.0001998	USDT TetherUS	0.7204861 \$0.7204861	0.7204861	XRP Ripple	0.435967 \$0.21920421	0.435967	<p>-BINANCE 계좌-</p> <div> <div>BTC: 0.00019980</div> <div>USDT: 0.72048610</div> <div>XRP: 0.43596700</div> </div> <p>-출력화면-</p>
Coin	Amount	Available												
BTC Bitcoin	0.0001998 \$8.58	0.0001998												
USDT TetherUS	0.7204861 \$0.7204861	0.7204861												
XRP Ripple	0.435967 \$0.21920421	0.435967												

2) 전략 개발

지표 데이터 소스 코드

```
def get_macd(fast, slow, sig, data):
    df = data
    macd_fast = df['close'].ewm(span=fast, adjust=False).mean() # 단기
    macd_slow = df['close'].ewm(span=slow, adjust=False).mean() # 장기
    macd = macd_slow - macd_fast # MACD
    df['macd'] = macd # MACD column 추가
    signal = df['macd'].ewm(span=sig, adjust=False).mean() # 시그널
    df['signal'] = signal # 시그널 column 추가
    histogram = macd - signal # 히스토그램
    df['histogram'] = histogram # 히스토그램 column 추가

    df['maPosition'] = ''
    prev_value = df['histogram'].iloc[0]
    for i in range(1, len(df)):
        current_value = df['histogram'].iloc[i]
        if prev_value < 0 and current_value >= 0:
            if i >= 2:
                df.at[i-2, 'maPosition'] = 'long'
                df.at[i-1, 'maPosition'] = 'long'
            if i+2 < len(df):
                df.at[i, 'maPosition'] = 'long'
                df.at[i+1, 'maPosition'] = 'long'
        elif prev_value >= 0 and current_value < 0:
            if i >= 2:
                df.at[i-2, 'maPosition'] = 'short'
                df.at[i-1, 'maPosition'] = 'short'
            if i+2 < len(df):
                df.at[i, 'maPosition'] = 'short'
                df.at[i+1, 'maPosition'] = 'short'
        prev_value = current_value

    return df
```

```
# RSI 구현하기
def get_rsi_ma(df, period, ma_period):
    delta = df['close'].diff(1)
    gain = (delta.where(delta > 0, 0)).rolling(window=period).mean()
    loss = (-delta.where(delta < 0, 0)).rolling(window=period).mean()
    rs = gain / loss
    rsi = 100 - (100 / (1 + rs))
    df['rsi'] = rsi # RSI column 추가

    rsi_ma = rsi.rolling(window=ma_period).mean()
    df['rsi_ma'] = rsi_ma # RSI_MA column 추가
    rsi_maHistogram = rsi - rsi_ma
    df['rsi_maHistogram'] = rsi_maHistogram

    df['rsiPosition'] = ''
    prev_value = df['rsi_maHistogram'].iloc[0]
    for i in range(1, len(df)):
        current_value = df['rsi_maHistogram'].iloc[i]
        if prev_value < 0 and current_value >= 0:
            if i >= 2:
                df.at[i-2, 'rsiPosition'] = 'long'
                df.at[i-1, 'rsiPosition'] = 'long'
            if i+2 < len(df):
                df.at[i, 'rsiPosition'] = 'long'
                df.at[i+1, 'rsiPosition'] = 'long'
        elif prev_value >= 0 and current_value < 0:
            if i >= 2:
                df.at[i-2, 'rsiPosition'] = 'short'
                df.at[i-1, 'rsiPosition'] = 'short'
            if i+2 < len(df):
                df.at[i, 'rsiPosition'] = 'short'
                df.at[i+1, 'rsiPosition'] = 'short'
        prev_value = current_value

    return df
```

3) MACD,RSI 지표를 이용하여 매수, 최적의 익절, 손절 비율 찾은 후 주문하는 백테스팅

소스코드

```
File Edit Selection View Go Run Terminal Help
Welcome device_baktest2.py X
C:\Users\lee\Desktop> device_baktest2.py ...

362
363 if __name__ == "__main__":
364     # start = '2020-01-01'
365     start = sys.argv[1]
366     # end = '2020-12-31'
367     end = sys.argv[2]
368     fast = 12
369     slow = 26
370     sig = 9
371     per = 14
372     m_per = 50
373     b = 1000000
374     f = 0.02
375     r = 0.13
376     d = 10
377     l = 15
378     df = get_data(start, end)
379     df = get_macd(fast, slow, sig, df)
380     df = get_rsi_ma(df, per, m_per)
381     result = devide_baktest(b, f, r, df, d, l)
382     result = result.reset_index()
383     result.to_csv('20_baktest_lev3.csv')
384     print("시작일 (0), 종료일 (1), 분할일 (2)일, 레버리지 (3)백" .format(start, end, d, l))
385     print("자본금 : " + str(b))
386     last_index = result.shape[0] - 1
387     print("최종 원수금 : " + str(result['Deposit'][last_index]))
388     profit = result['Deposit'][last_index] - b
389     profit_ratio = profit / b * 100
390     print("수익(%) : " + str(profit) + "(" + str(profit_ratio) + "%)")
391     print("거래 횟수 : " + str(result.shape[0]))
392     winrate = check_winrate(result) * 100
393     print("승률 : " + str(winrate) + "%")
```

3-3) 결과

소스코드	
자본금 20%진입, 레버리지 10배	
<p>시작일 2018-01-01, 완료일 2018-12-31, 분할일 10일, 레버리지 10배</p> <p>자본금 : 1000000</p> <p>최종 환수금 : 2683584.505283579</p> <p>수익(%) : 1683584.5052835792(168.3584505283579%)</p> <p>거래 횟수 : 644</p> <p>승률 : 66.14906832298136%</p>	<p>시작일 2021-01-01, 완료일 2021-12-31, 분할일 10일, 레버리지 10배</p> <p>자본금 : 1000000</p> <p>최종 환수금 : 4370421.499244755</p> <p>수익(%) : 3370421.499244755(337.0421499244755%)</p> <p>거래 횟수 : 661</p> <p>승률 : 71.55824508320727%</p>
<p>시작일 2019-01-01, 완료일 2019-12-31, 분할일 10일, 레버리지 10배</p> <p>자본금 : 1000000</p> <p>최종 환수금 : 1805527.9103128219</p> <p>수익(%) : 805527.9103128219(80.55279103128218%)</p> <p>거래 횟수 : 666</p> <p>승률 : 69.36936936936937%</p>	<p>시작일 2022-01-01, 완료일 2022-12-31, 분할일 10일, 레버리지 10배</p> <p>자본금 : 1000000</p> <p>최종 환수금 : 2733528.7667497364</p> <p>수익(%) : 1733528.7667497364(173.35287667497363%)</p> <p>거래 횟수 : 605</p> <p>승률 : 70.08264462809917%</p>
<p>시작일 2020-01-01, 완료일 2020-12-31, 분할일 10일, 레버리지 10배</p> <p>자본금 : 1000000</p> <p>최종 환수금 : 2862018.998230573</p> <p>수익(%) : 1862018.9982305728(186.2018998230573%)</p> <p>거래 횟수 : 649</p> <p>승률 : 72.41910631741139%</p>	<p>시작일 2023-01-01, 완료일 2023-12-31, 분할일 10일, 레버리지 10배</p> <p>자본금 : 1000000</p> <p>최종 환수금 : 1259062.8190371464</p> <p>수익(%) : 259062.8190371464(25.90628190371464%)</p> <p>거래 횟수 : 561</p> <p>승률 : 65.24064171122996%</p>
백테스팅 결과 매년 최소 25% 최대 337% 수익	

- 프로젝트 수행을 위해 적용된 추진전략, 수행 방법의 결과를 작성하고, 만일 적용과정에서 문제점이 도출되었다면 그 문제를 분석하고 해결방안을 기술하시오.

- 전략 문제

여러 지표중 MACD,RSI 지표를 통합하여 사용하는 것이 수익이 최대로 나왔으나 매년 수익이 보장되지 않고 마이너스 수익도 나는 상황 발생

- MACD,RSI 기준으로 롱,숏 포지션 볼린저 밴드로 포지션 청산 (레버리지 2배)
- 수익 25.2 % (2021년 기준)

```
PS C:\Users\lee\Desktop\캡스톤> & C:/Users/lee/AppData/Local/Microsoft/WindowsApps/PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0/python.exe c:/Users/lee/Desktop/캡스톤/backtest/test0410.py
최종결과
속성 : fast=12, slow=26, sig=9, period=14, ma_period=50, base=1000000, fee=0.02, ratio=0.1, upSell=0.32, downSell=1.03, leverage=2
자본금 : 1000000
최종 환수금 : 1252276.7540475794
수익(%) : 252276.75404757936(25.22767540475794%)
거래 횟수 : 6653
승률 : 65.21869833157974%
PS C:\Users\lee\Desktop\캡스톤> ]
```

- 수익 -7 % (2022년 기준)

```
PS C:\Users\lee\Desktop\캡스톤> & C:/Users/lee/AppData/Local/Microsoft/WindowsApps/PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0/python.exe c:/Users/lee/Desktop/캡스톤/backtest/test0410.py
최종결과
속성 : fast=12, slow=26, sig=9, period=14, ma_period=50, base=1000000, fee=0.02, ratio=0.1, upSell=0.32, downSell=1.03, leverage=2
자본금 : 1000000
최종 환수금 : 925386.6672579156
수익(%) : -74613.3327420844(-7.46133327420844%)
거래 횟수 : 4667
승률 : 59.888579387186624%
```

- 수익 -1.8 % (2023년 기준)

```
PS C:\Users\lee\Desktop\캡스톤> & C:/Users/lee/AppData/Local/Microsoft/WindowsApps/PythonSoftwareFoundation.Python.3.12_qbz5n2kfra8p0/python.exe c:/Users/lee/Desktop/캡스톤/backtest/test0410.py
최종결과
속성 : fast=12, slow=26, sig=9, period=14, ma_period=50, base=1000000, fee=0.02, ratio=0.1, upSell=0.32, downSell=1.03, leverage=2
자본금 : 1000000
최종 환수금 : 981905.8927503992
수익(%) : -18994.107249600813(-1.8994107249600813%)
거래 횟수 : 2944
승률 : 54.85733695652174%
PS C:\Users\lee\Desktop\캡스톤> ]
```

1분 데이터로 지표를 그리다 보니 지표의 안정성이 부족 하였음. 15분 데이터를 통한 주문하는 방식으로 전환 하지만 추세 때문인지 수익률이 여전히 저조. 이를 해결하고자 10일 데이터를 통해 최적의 익절,손절비율 찾아 10일 적용 , 적용중 다시 최적의 비율찾기 반복 하여 최고 수익률을 내는 전략을 발견하여 해결완료.

캡스톤 디자인 | 중간보고서 채점표

평가도구	평 가 항 목	평 가 점 수				
		1	2	3	4	5
중간 보고서 및 실행 결과	1. 요구사항 정의서(기능, 성능, 인터페이스 등)가 구체적으로 작성되었는가?					
	2. 요구분석, 설계 산출물(모델, 프로토타입 등)의 내용이 충실한가?					
	3. 설계 및 구현 문제를 위해 적용한 이론, 문제해결 방법이 제시되었으며 그 적용이 적합한가?					
	4. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)가 버그없이 실행되었는가?					
	5. 구현된 소프트웨어(또는 이와 동등한 하드웨어 시스템)의 성능 요구사항은 충족되었는가?					
도구활용	6. 설계 및 구현을 위해 도구가 적절히 활용되었는가?					
	7. 도구의 활용수준(능숙도)은 프로젝트 수행에 적합한가?					
팀원의 업무 및 역할	8. 팀원의 업무분담에 따른 역할 및 협력이 충실히 이루어졌는가? (평가자에 의한 질의)					
	9. 프로젝트 중간 진척상황에 대해 팀원이 충분히 인지하고 있는가?(평가자에 의한 질의)					
합계						
*검토 의견(최종완료 때까지 보완해야할 점에 대해 작성 요망)						
심사위원(소속):		(이름)			(인)	