<모의 주식>

<anotn> - 백종빈 , 조상훈, 김우진, 김선빈, 임수빈

Version <1.0>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **날짜** | **버전** | **설명** | **Author** |
| **<2024/03/26>** | **<1.0>** | **<문서 생성 및 프로젝트 기초 구상 >** | **백종빈** |
| **<2024/04/01>** | **<1.0>** | **< 구상정리>** | **백종빈** |
| **<2024/04/08>** | **<1.0>** | **<역할 구성 및 역할에 따른 조사>** | **백종빈** |
| **<20204/05/07>** | **<1.0>** | **<앱 기획물 제작> -카카오 오븐** | **백종빈** |
| **<2024/05/23>** | **<1.0>** | **<2.8 투자 학습 및 성과 관리>** | **김선빈** |
| **<2024/05/24>** | **<1.0>** | **<유스케이스 작성>** | **김선빈** |
| **<2024/05/25>** | **<1.0>** | **<지원자료 작성>** | **김선빈** |
| **<2024/05/28>** | **<1.0>** | **< ERD 작성>** | **조상훈** |
| **<2024/05/28>** | **<1.0>** | **<테이블 코드 작성>** | **조상훈** |
|  |  |  |  |

목차

1. 서론

1.1 시스템 개요

1.2 정의 및 약어

1.3 주식 거래

1.3.1 사용자 관련 혜택사항

2. 본론 (전체 설명)

2.1 주식 거래

2.2 포트폴리오 관리

2.3 실시간 시장 정보

2.4 교육 및 통계

2.5 보안 및 안전성

2.6 모바일 특화 기능

2.7 사용 사례 모델 조사 (벤치마킹)

2.8 투자 학습 및 성과 관리

3. 특정 요구사항

3.1 유스케이스

3.2 ERD

3.3 ERD 테이블 코드

4. 지원 자료

소프트웨어 요구사항

# 서론

주식 시장에 대한 이해도와 실전 투자 경험은 많은 사람들이 가지고 있지 않습니다. 이에 모의주식 모바일 어플리케이션은 사용자들이 실제 주식 시장의 환경에서 거래를 시뮬레이션하고 투자 스킬을 향상시킬 수 있는 플랫폼으로 개발되었습니다. 이 보고서는 어플리케이션의 요구사항 및 설계를 다루며, 특히 사용 사례 모델 조사 후 좋은 부분은 흡수하는 형태로 하겠습니다.

## 시스템 개요

이 어플리케이션은 주식 시장에서의 거래를 시뮬레이션하는 데 목적을 둔다.

사용자들은 가상의 자본으로 주식을 매수하거나 매도할 수 있다.

실시간 가격에 따른 모의 주식을 진행합니다.

## 정의 및 약어

다양한 주식과 관련된 용어 및 주식 거래를 돕는 HTS의 용어들을 알기 쉽게 정의 및 풀이한다.

## 사용자 관리

사용자는 개별 계정을 생성할 수 있어야 한다.

계정 생성 시에는 사용자의 기본 정보와 로그인 자격 증명이 필요하다.

비밀번호는 암호화되어 저장되어야 한다.

## 사용자 관련 기능

개인 프로필 꾸미기

모의 포인트 충전

모의 포인트 랭킹과 각종 미션에 따른 칭호.

# 본론 (전체 내용)

본격적으로 한국투자증권 오픈 API를 통하여 실시간 가격을 제공받으며 이것을 통해 모의주식의 데이터를 받아와 완성할 것 입니다.

역할

- 화면 UI (로그인, 개인 프로필 등) – 임수빈

- JSON 분석 및 적용, 문서 정리 – 김선빈

- 데이터베이스 레이아웃 설계 및 구축 – 조상훈

- 시나리오 (프론트, 백엔드) – 백종빈

- 클라이언트 (차트 데이터, 디자인 등) – 백종빈

- 백엔드 구축 – 김우진

## 주식 거래

사용자는 가상 자본을 사용하여 주식을 매수하거나 매도할 수 있다.

주식 거래 시에는 실제 시장 데이터를 사용하지 않고, 가상의 가격을 적용한다.

사용자는 종목 검색을 통해 원하는 주식을 찾을 수 있어야 한다.

거래 시에는 거래량, 가격, 주식 종목 등의 정보를 제공해야 한다.

## 포트폴리오 관리

사용자는 보유 중인 주식의 포트폴리오를 확인할 수 있어야 한다.

포트폴리오에는 보유 중인 주식의 종류, 수량, 평가 금액 등의 정보가 포함되어야 한다.

사용자는 포트폴리오를 기반으로 매수 또는 매도 결정을 내릴 수 있어야 한다.

## 실시간 시장 정보

어플리케이션은 실시간으로 주식 시장의 정보를 제공해야 한다.

주식 시세, 시장 지표 등의 정보를 제공한다.

실제 주식 시장 데이터를 기반으로 하되, 가상 거래이므로 실제 거래에 영향을 주지 않아야 한다.

## 교육 및 통계

어플리케이션은 사용자들에게 주식 시장에 대한 교육 자료를 제공해야 한다.

거래 기록, 수익률, 포트폴리오 분석 등의 통계 정보를 제공한다.

사용자들이 자신의 거래 활동을 분석하고 학습할 수 있도록 돕는다.

## 보안 및 안전성

사용자 데이터와 거래 정보는 안전하게 보호되어야 한다.

강력한 인증 및 권한 시스템을 구현하여 불법적인 접근을 방지해야 한다.

취약점 분석 및 보안 업데이트가 주기적으로 이루어져야 한다.

## 모바일 특화 기능

모바일 환경에 최적화된 사용자 경험을 제공해야 한다.

모바일 기기에서의 화면 크기, 터치 입력 등을 고려한 UI/UX가 요구된다.

푸시 알림 등의 모바일 특화 기능을 제공할 수 있다.

## 사용 사례 모델 조사 (벤치마킹)

사용 사례 모델 조사는 유사한 제품 또는 서비스를 제공하는 다른 회사들의 사례를 분석하여 우리의 어플리케이션 개발에 도움을 주는 중요한 단계입니다. 다음은 주식 시뮬레이션 어플리케이션 분야에서 Play스토어 기준 다운로드가 가장 많은 2가지 어플을 참고하여 제작할 예정입니다.

**2.8 투자 학습 및 성과 관리**

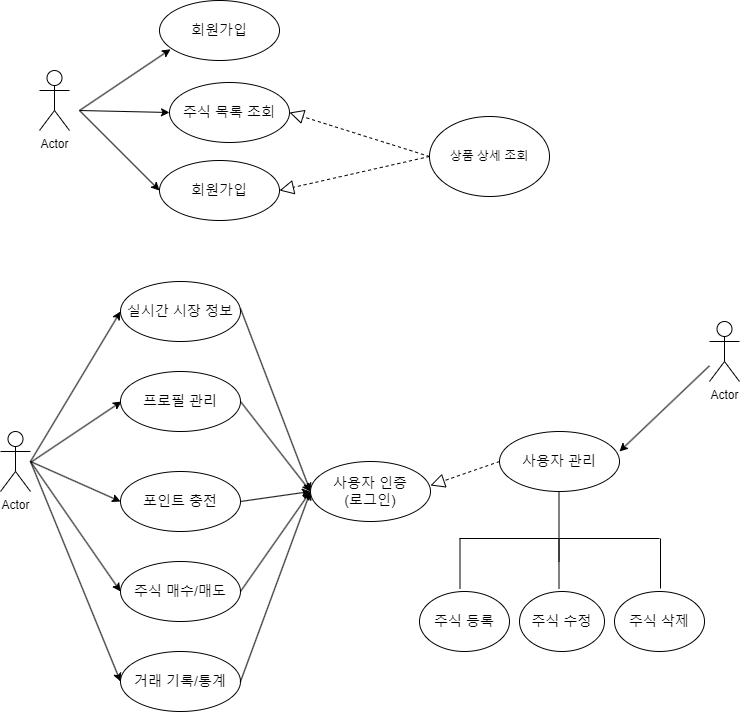
초보 투자자들을 위한 튜토리얼 및 가이드 기능을 제공한다.

사용자 개인의 모의투자 성과를 한눈에 확인할 수 있다.

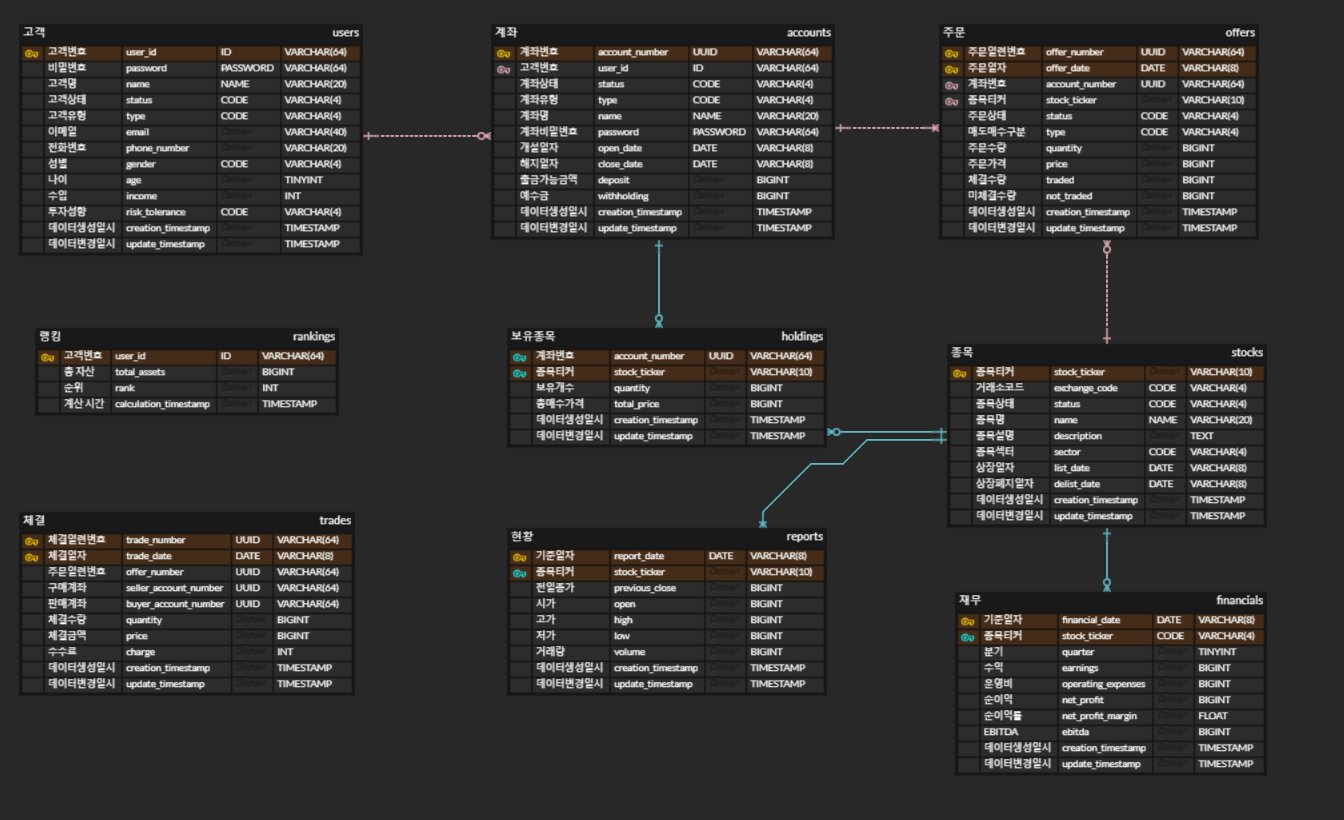
모의투자 랭킹 시스템을 통해 동기부여를 제공한다.

# 특정 요구사항

## 유스케이스



**3.2 ERD**

****

**3.3 ERD 테이블 코드**

**CREATE TABLE users (**

**user\_id VARCHAR(64) PRIMARY KEY,**

**password VARCHAR(64) NOT NULL,**

**name VARCHAR(20),**

**status VARCHAR(4),**

**type VARCHAR(4),**

**email VARCHAR(50),**

**phone\_number VARCHAR(20),**

**gender CHAR(1),**

**age TINYINT,**

**income INT,**

**risk\_tolerance VARCHAR(4),**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP**

**);**

**CREATE TABLE accounts (**

**account\_number UUID PRIMARY KEY,**

**user\_id VARCHAR(64),**

**status VARCHAR(4),**

**type VARCHAR(4),**

**name VARCHAR(20),**

**password VARCHAR(64),**

**open\_date DATE,**

**close\_date DATE,**

**deposit BIGINT,**

**withholding BIGINT,**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,**

**FOREIGN KEY (user\_id) REFERENCES users(user\_id)**

**);**

**CREATE TABLE stocks (**

**stock\_ticker VARCHAR(10) PRIMARY KEY,**

**exchange\_code VARCHAR(4),**

**status VARCHAR(4),**

**name VARCHAR(20),**

**description TEXT,**

**sector VARCHAR(20),**

**list\_date DATE,**

**delist\_date DATE,**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP**

**);**

**CREATE TABLE holdings (**

**account\_number UUID,**

**stock\_ticker VARCHAR(10),**

**quantity BIGINT,**

**total\_price BIGINT,**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,**

**PRIMARY KEY (account\_number, stock\_ticker),**

**FOREIGN KEY (account\_number) REFERENCES accounts(account\_number),**

**FOREIGN KEY (stock\_ticker) REFERENCES stocks(stock\_ticker)**

**);**

**CREATE TABLE offers (**

**offer\_number UUID PRIMARY KEY,**

**offer\_date DATE,**

**account\_number UUID,**

**stock\_ticker VARCHAR(10),**

**status VARCHAR(4),**

**type VARCHAR(4),**

**quantity BIGINT,**

**price BIGINT,**

**traded BIGINT,**

**not\_traded BIGINT,**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,**

**FOREIGN KEY (account\_number) REFERENCES accounts(account\_number),**

**FOREIGN KEY (stock\_ticker) REFERENCES stocks(stock\_ticker)**

**);**

**CREATE TABLE trades (**

**trade\_number UUID PRIMARY KEY,**

**trade\_date DATE,**

**offer\_number UUID,**

**seller\_account\_number UUID,**

**buyer\_account\_number UUID,**

**quantity BIGINT,**

**price BIGINT,**

**charge INT,**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,**

**FOREIGN KEY (offer\_number) REFERENCES offers(offer\_number),**

**FOREIGN KEY (seller\_account\_number) REFERENCES accounts(account\_number),**

**FOREIGN KEY (buyer\_account\_number) REFERENCES accounts(account\_number)**

**);**

**CREATE TABLE reports (**

**report\_date DATE,**

**stock\_ticker VARCHAR(10),**

**previous\_close BIGINT,**

**open BIGINT,**

**high BIGINT,**

**low BIGINT,**

**close BIGINT,**

**volume BIGINT,**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,**

**PRIMARY KEY (report\_date, stock\_ticker),**

**FOREIGN KEY (stock\_ticker) REFERENCES stocks(stock\_ticker)**

**);**

**CREATE TABLE financials (**

**financial\_date DATE,**

**stock\_ticker VARCHAR(10),**

**quarter VARCHAR(2),**

**earnings BIGINT,**

**revenue BIGINT,**

**operating\_expenses BIGINT,**

**net\_profit BIGINT,**

**net\_margin FLOAT,**

**ebitda BIGINT,**

**creation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**update\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,**

**PRIMARY KEY (financial\_date, stock\_ticker),**

**FOREIGN KEY (stock\_ticker) REFERENCES stocks(stock\_ticker)**

**);**

**CREATE TABLE rankings (**

**user\_id VARCHAR(64),**

**total\_assets BIGINT,**

**rank INT,**

**calculation\_timestamp TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,**

**PRIMARY KEY (user\_id),**

**FOREIGN KEY (user\_id) REFERENCES users(user\_id)**

**);**

# 지원자료



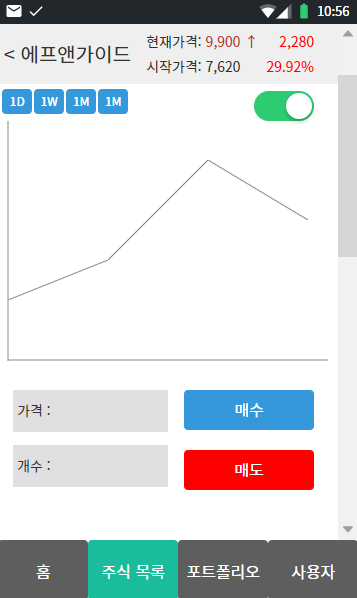
앱 첫 실행 시 나오는 초기 화면으로 주요 기능들을 설명하고 있습니다.



하단 탭 메뉴로 홈, 주식 목록, 포트폴리오, 사용자 등의 메뉴를 갖추고 있습니다.



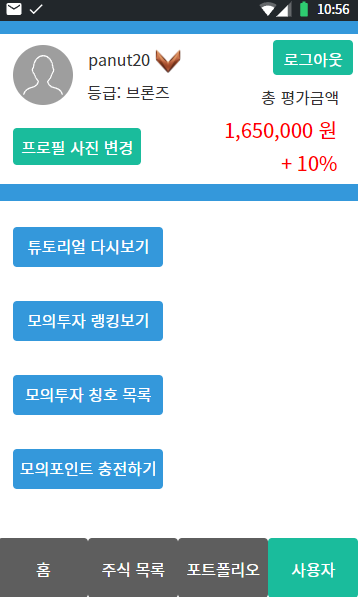
종목 코드나 이름으로 검색하여 매수/매도 주문을 할 수 있는 화면입니다.



주식 현재가, 차트, 거래량 등의 정보를 제공합니다.



현재 보유 중인 주식 포트폴리오와 수익률 등을 한눈에 확인할 수 있습니다.



사용자 페이지는 로그인한 개인 사용자의 정보와 설정을 확인하고 관리할 수 있는 화면입니다.

<이미지 구상 및 계획> <카카오 오븐>



https://ovenapp.io/view/89aM0JVBB4QeWuDIscTLfoMWBO0grktl/