

Software Engineering

Investment league game

[Final Report]

바람만스쳐도에러 Team.

장준호

박웅빈

오상택

김도영

Project Repository

2015.12.08.

To whom may it concern

This program is a investment league game to compete with others. We will provide a virtual cash. An user invested stocks and earn some money. If the user would earn money most, the user would win the game league.

We brought a stock information data from 'YAHOO finance'. The data are saved in the database of our server. User's Android interface operates our server to calculate and show the data.

What you have to check this document seriously is that we progressed this project by a specific plan. At first, We tried to write down a user story(ST) that the user would do like this when they use this program. And we came up with a specific use case(UC) derived from the user story(ST). Then the interface, domain model and databases are derived from the user story(UC).

Contents

Index

- 1. Team Profile**
- 2. Project Proposal**
- 3. Product Ownership**
- 4. Project Management & Plan of Work**
- 5. Customer Statement of Requirements**
- 6. System Requirements**
- 7. Functional Requirments Specification**
- 8. Interface**
- 9. Domain Model**

1. Team Profile

프로젝트 팀의 리드는 오상택 팀원이 맡았으며, 이외 3명의 팀원들이 이에 따른다. 팀원 전부는 같은 학과의 선후배 관계이며 실력에 따라 각자 할 수 있는 분야를 할당받아서 해결 하였다. 각각 프로젝트에 할애할 수 있는 시간을 기재 했으며, 그 시간은 팀원의 개인 상황에 따라 달라질 수 있습니다. 대부분의 코딩은 학교에서 이뤄 졌으며, 회의 역시 교수님의 지원에 따라 강의실 회의실 학교 서버 등을 제공 받았다.

팀의 커뮤니케이션은 기본적으로 카카오톡(메신저)로 이루어지며, 매주 월요일이나 화요일에 토론과 회의를 거쳐 프로젝트의 진행을 검토하고 결정합니다. 필요에 따라 같이 모여 평일과 주말에 직접적인 코딩을 하였습니다. 프로그래밍 상으로는 서버는 Github, 클라이언트는 Bit bucket을 통한 소스 관리가 이루어지며, 필요시 Team Viewer를 통한 원격연결 또는 화상 회의도 할 것입니다.

1.01 오상택

C, C++, Java, Python, C# 등의 언어를 사용하며, 리눅스 프로그래밍, Data Base, 안드로이드 프로그래밍, 서버 프로그래밍, 웹 프로그래밍을 해보았다. 현 프로젝트와 비슷한 안드로이드 및 웹서버 프로그래밍을 이미 해보았으며, 이외에 4개 정도의 프로젝트를 수행한 적이 있습니다. 시간을 가장 많이 할애할 수 있으며, 프로젝트 전체적인 기획과 기술적인 리드를 이끈다.

1.02 김도영

C, C++, Java, Python, C# 등의 언어를 사용하며, 리눅스 프로그래밍, Data Base, 안드로이드 프로그래밍, 서버 프로그래밍을 해보았다. 프로젝트와 비슷한 안드로이드 및 웹서버 프로그래밍을 이미 해보았으며, 1개의 프로젝트를 수행한 적이 있습니다. 매주 리눅스 교습을 수행함으로써 리눅스에 정통하지만, 팀프로젝트는 처음이며, 프로젝트에 하루 3시간 정도의 시간을 할애 할 수 있습니다. 서버와 클라이언트 두 부분을 지원 담당합니다.

1.03 박웅빈

C, C++, Java, Python, C# 등의 언어를 사용하며, 리눅스 프로그래밍, Data Base, 안드로이드 프로그래밍, 서버 프로그래밍을 해보았다. 프로젝트와 비슷한 안드로이드 및 웹서버 프로그래밍을 이미 해보았으며, 1개의 프로젝트를 수행한 적이 있습니다. 팀프로젝트는 처음입니다. 현재 수행하고 있는 프로젝트가 하나 있지만, 하루에 7시간 이상을 프로젝트에 할애할 수 있어 서버 부문에 많은 지원을 할 예정입니다.

1.04 장준호

C, C++, Java의 언어를 다루고 안드로이드 프로그래밍을 해보았다. 안드로이드 기반 1개의 프로젝트를 수행한 적이 있다, 팀프로젝트는 처음이지만, 주식 경험자임으로 프로젝트의 진행 방향에 도움이 많이 될 것입니다. 안드로이드 프로그래밍 경험으로 UI와 안드로이드 클라이언트를 전반적으로 담당합니다.

2. Project Proposal

2014년도에 외국의 Stock Market Investment Fantasy League 프로젝트를 수행한 팀1의 The Paramount Investments League와 같이 '주식 시장 게임'을 만들 것입니다. 하지만, The Paramount Investments League에서 구현된 '소규모 리그를 만들어서 리그 내에 서만의 경쟁'이 이루어졌던 게임구조와는 달리, 리그를 확장시켜 '모든 사용자가 리그에 참여하되, 리그는 시즌별로 이루어지며, 시즌 이후에는 모두 초기화된다.' 라는 설정을 모토로 잡았다.

기타 구체적인 부분들이 The Paramount Investments League와는 많이 달라진다. 웹사이트를 통한 서비스인 반면에 안드로이드 앱 대 웹으로 클라이언트 대 서버로 개발되며, 이외에 OpenID / OAuth2를 사용하지 않고 안드로이드 어플리케이션에서 요청하면 데이터베이스에서 연산을 하여 로그인을 제동해줍니다. 금융 API 경우는 야후 금융을 그대로 사용할 것입니다. 또한, 교육과 연습용도의 The Paramount Investments League였지만, 팀은 좀 더 현실성에 가깝게 하기 위하여 판을 크게 만들고 시즌별로 운영하는 등으로 시즌 게임용도에 맞춰 개발이 이루어질 것입니다. 편의성을 위해 안드로이드 어플리케이션에서 프로그램을 접할 수 있도록 만들었습니다.

2.01 등록

아이콘을 눌러서 실행을 하면, 로그인 창이 나오면서 시작합니다. 로그인을 위한 클라이언트 등록은 UI에서 쉽게 회원가입을 할 수 있습니다. 가입된 정보는 서버에 있는 데이터베이스에 바로 저장되어 사용이 가능합니다. 이를 등록하여 클라이언트에서 바로 접속할 수 있습니다. 한번 등록되면 이후 계속 등록 과정 없이 로그인을 할 수 있습니다..

2.02 리그 계정

등록 후 리그에 참가하기 위해서는 사용자가 계정을 만들어 줘야합니다. 계정은 팀의 DB에 저장되며, 해당 계정은 해당 시즌까지만 유지가 됩니다.

2.03 포트폴리오 관리 및 주식 거래

만들어진 계정으로 시즌내의 모든 거래가 실시간으로 이루어진다. 시즌 내의 모든 거래는 로그로 기록되어지며, 사용자는 자신의 포트폴리오에 대한 관리를 할 수 있습니다.

2.04 리그

여러 개의 소규모 리그가 생성과 소멸을 반복하는 구조보다는 1년에 2번 정도의 시즌별로 거대한 리그 하나만 생성되어 모든 사용자들이 참여하게 되는 구조입니다. 시즌 동안 매번 랭킹은 갱신되며, 리그 참여타이밍에 관한 판단은 전적으로 유저가 하게 됩니다.

2.05 예약 매매(큐잉시스템)

안드로이드 앱 으로서 큐잉 시스템입니다. 단말기가 인터넷만 연결되어 있다면 클라이언트가 매시간 신경을 쓰고 있지 않더라도 매매가 가능하게 될 것입니다.

2.06 주식 검색

사실적인 게임 진행을 위하여 하루 또는 초 단위로 외부 금융 API를 통하여 종목 리스트들과 정보들을 받아오고 화면 위쪽에 뿌려주게 됩니다.

2.07 랭킹

시즌마다 순위가 매겨지며, 매번 순위는 갱신된다. 순위 판별 기준은 수익률에 따릅니다. 시즌이 끝나면 순위는 다시 초기화 상태로 돌아갑니다.

3. Product Ownership

부문에 따라 긴밀한 협의와 토론으로 최대한 간략한 구조를 지향합니다. 크게 클라이언트 부문과 서버 부문으로 나눌 수 있습니다. 클라이언트 부문은 박웅빈, 장준호가 담당하였고, 서버 부문은 오상택, 김도영이 담당하게 되었습니다. 하지만, 각자의 내용을 학부때 습득 하고 있습니다. 때문에 아이디어를 공유하고 서로의 제안을 받아드리는데 문제는 전혀 존재 하지 않았습니다.

3.01 등록 (클라이언트)

3.02 리그 계정 (클라이언트, 서버)

3.03 포트폴리오 관리 및 주식 거래 (클라이언트, 서버)

3.04 리그 (서버)

3.05 큐잉 시스템(클라이언트)

3.06 검색(클라이언트)

3.07 랭킹 (클라이언트, 서버)

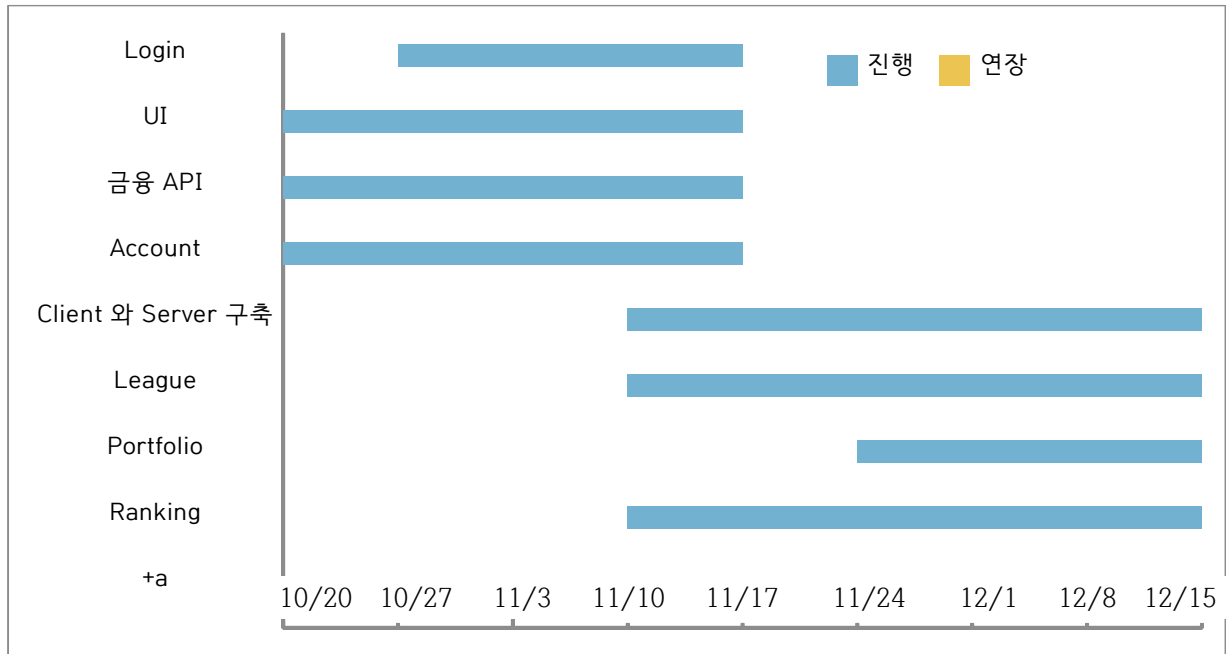
3.09 광고 연계(교수님 바램)

3.10 현재 담당하고 있는 팀원 분포

Category	W	오상택	김도영	박웅빈	장준호
Project Management	10	O			
Customer Requirements	15			O	O
System Requirements	15	O	O	O	
Functional Requirements	15	O	O	O	
User Interface	15			O	
SEP_SMIFL Analysis	25	O	O		
Plan of Work	5	O			O
Total	100	Server		Client	

4. Project Management & Plan of Work

4.01 일정



4.02 야후 금융

사전 조사에서 NH투자증권 제공의 TX OpenAPI로 사용하려 했으나, 토론을 통하여 기존 프로젝트에서 사용된 야후 API 사용을 결정했습니다.. 사용하기 위한 분석이 이루어지고 있습니다.

4.03 서버

서버 컴퓨터의 구축을 했으며, 원격 접속등을 확인했습니다.

4.04 코드 관리

클라이언트는 BitBucket을 통한 Mercurial로 관리됩니다.

서버는 GitHub를 통한 Git으로 관리됩니다.

5. Customer Statement of Requirements

5.1 문제

팀은 2014년도에 외국의 Stock Market Investment Fantasy League 프로젝트를 수행한 팀1의 The Paramount Investments League를 리버스 엔지니어링으로 분해를 해보고 더 나은 결과물을 만드는 것을 목표로 프로젝트를 수행합니다.

이 프로젝트는 주식시장의 높은 진입 장벽의 대안으로서 체험 기회를 제공하는 주식 투자 리그 모바일 어플리케이션입니다. 누구나 쉽게 어플리케이션을 통하여 시스템에 등록하고, 리그에 참여하여 주식 모의 투자를 시작할 수 있습니다. 주식 투자로서 사용자들은 가상의 현금과 포트폴리오를 제공 받고 시즌제 리그에 참여합니다. 주문은 실제 시장에서 실시간으로 변동하는 가격과 그래프를 이용합니다. 기존 목표보다 더 사실성을 높이기 위해서 리스의 규모를 크게 만들었습니다. 시즌제 리그에서 참여한 모든 사용자들은 가능한 큰 수익률에 따라 순위가 매겨지고, 기간이 끝날 때마다 순위와 현금 및 포트폴리오들은 초기화되어 새로운 리그를 시작하게 됩니다. 리그는 미국이나 한국 주식만이 아니라 글로벌 리그로서 치열한 경쟁이 예상됩니다. 통화의 기준은 US달러로 하였으며, 각 나라의 환율API를 이용하여 랭크를 만들 수 있도록 하였습니다. 시장을 더 쉽게 이해시키기 위해 리그에서 일어나는 현재 거래의 대화 형 차트를 제공하여 기업의 공급을 통해, 사용자가 주식 매매에 대한 판단을 할 수 있습니다. 또한, 사용자는 앱을 통하여 큐잉 시스템 예약 매매를 통하여 사용자의 집중도부담을 줄이는데 주력했습니다. 해당 시즌에서 리그의 우승은 리그 참여의 물리적 시간에 전혀 영향을 받지 않습니다. 절대적 자산으로만 링크를 판단하게 됩니다.

5.2 용어 설명

소프트웨어의 설계의 목적을 충분히 이해를 위해 필요해서 일부 항목의 작은 개요를 제공합니다. 제공된 세부적인 사항들은 소프트웨어 설계 내에서 목표와 프로세스의 완전한 이해를 얻는 것이 중요하기에 기술하였습니다. 또한, 용어는 또한 소프트웨어의 이해와 작동 방식에 매우 중요한 기능뿐만 아니라 중요한 금융 용어도 포함합니다.

거래 시세 : 지속적으로 시장에서 가장 최근의 거래의 스크롤을 업데이트하는 것. 사용자는 포트폴리오에 있을 수도 있고 없을 수도 있는 글로벌 증시에서 시장 동향을 관찰 할 수 있습니다.

리더보드 : 플레이어의 순이익률에 의해 결정됩니다. 글로벌 리그를 기반으로 순위를 결정하는 시스템.

배당금 : 주주들에 대한 지불은 회사에 의해 만들어진, 일반적인 이익의 분포. 이것은 보통 주주 가치의 고정된 비율로 분배됩니다.

파생 상품 - 옵션 :사용자에게 또는 특정 날짜 이전에 특정 가격으로 자산을 구매 또는 판매 할 수 있는 옵션을 제공합니다. 구매자와 판매자가 모두 옵션을 가지고가는 경우에 지정된 날짜에 거래를 이행 할 의무가 있습니다.

파생 상품 - 선물 :구매자는 현재의 가격으로 자산을 구입 할 수 있도록 허용되고 선물 가격에 지불합니다. 선물은 일반적인 거래로 교환된다. 구매자와 판매자 모두 선물을 가지고 있는 경우에 지정된 날짜에 거래를 이행 할 의무가 있습니다.

파생 상품 - 선도 :구매자는 현재의 가격으로 자산을 구입 할 수 있도록 허용되고 선물 가격에 지불합니다. 선도는 반드시 시장 지분 기반으로 하지 않고 구매자와 판매자 사이에 개인 계약입니다. 구매자와 판매자 모두 선물을 가지고 있는 경우에 지정된 날짜에 거래를 이행 할 의무가 있습니다.

리그 : 승자를 결정하기 위하여 여러 투자자들이 경쟁하는 시장 시뮬레이션입니다 시스템이 정해진 시즌별로 이루어지며, 이것으로 순위가 매겨지고 끝날 때 승자가 정해집니다. 리그의 참여는 시작과 시즌 마감 전에 언제든지 참여 할 수 있습니다.

포트폴리오 : 주어진 리그에서 특정 투자자와 관련된 자산의 세부 계정입니다. 포트폴리오는 각 사용자마다 고유하며, 손실, 성능, 평균, 실적뿐만 아니라 주어진 포트폴리오 내 주식의 자세한 자산 실적 등과 같은 특정 세부 사항을 포함합니다.

주문 : 투자자는 자산의 구매 또는 판매를 위해 하는 주문

주식 : 회사의 지분을 나타내는 자산의 유형

Ask 가격(파는 가격) : 주식을 판매하고자 할 때 상인의 가격

Bid 가격(사는 가격) : 상인이 주식에 대해 지불할 가격

매매가격차이 : 입찰 요청과 가격의 차이를 보여줍니다. 두 가격이 소폭 다르지만, 이 둘 중 어느 것이 더 비싼지 보는 것이며, 주식 거래 고유의 마찰을 의미합니다.

시세 기호 : 이 용어는 특정 주식 시장에서 특정 주식의 상장 주식을 식별하는 데에 유일하게 사용됩니다.

기호 리스트 : 시장/여러 시장의 시세 기호 목록입니다.

시장 주문 : 사용자가 특정 콘텐츠를 구입하기로 결정했고 그 주식에 대한 입찰을 배치하는 것입니다.

판매 : 사용자는 특정 주식을 판매하기로 결정하고 가격을 책정하는 것입니다.

검색 : 특정한 주식을 찾고 싶을 때 검색기능을 사용할 수 있도록 제공하였습니다.

초단타매매 : 일반적으로 주식의 값이 감소 할 것으로 예상하는 투자자에 의해 사용됩니다. 투자자들은 특정 주식의 주식을 빌려 시장 가격에 판매하기도 하며, 주식을 수익의 낮은 차이로 투자자가 다시 살 수 있습니다. 투자자는 주식은 그 값이 증가하는 것에 대해 여전히 책임 있습니다.

제한 : 투자는 주어진 가격 범위에서 이루어집니다. 투자자가 구매 제한에서 지불할 수 있는 최대 금액을 배치할 수 있고, 판매 제한의 투자자는 주식 판매의 최소값을 배치할 수 있습니다. 제한 주문들은 지금까지 과정들을 보장 할 수 없고, 단지 특정 한계에 도달 할 때 할 수 있습니다.

정지 : 특정 정지 아래로 떨어지거나 특정 가격 이상으로 상승 할 경우 활성화되는 기능입니다. 투자자에 대한 이익과 손실을 최소화하는데 사용됩니다.

주식(share) : 구입하거나 다른 상인에서 판매 할 수 있는 특정 회사의 작은 비율입니다. 참고 : 채권 (Bond)

6. System Requirements

6.01 유저 스토리

아래에 기록된 '유저 케이스'는 각 요구 사항의 상대적으로 중요하게 측정된 가중치뿐만 아니라, 몇 가지 특정 인스턴스를 보이고 프로그램 및 기능에 대한 요구 사항을 가집니다. 특히 이러한 기능은 반드시 특정 가중치와 기능의 우선 순위를 간략하게 보여주기 위함입니다. 하지만, 최종 유저 케이스 사양과 프로그램 작동 흐름의 단순한 목록이기도 합니다. 이 문서의 추가 섹션에서 참조하여 검토하는 상황들을 관찰하는 것이 중요합니다.

Index	유저 케이스	W
ST-1	사용자로서, 어플리케이션 로그인 창에서 회원가입 버튼을 클릭하고 회원가입을 할 수 있습니다.	10
ST-2	사용자로서, 계정을 가지고 있다면 아이디와 패스워드를 입력하고 로그인 할 수 있습니다.	10
ST-3	사용자로서, 실시간에 가까운 주식 데이터에 기초하여 주식 시장 환경에서 다른 사용자와 경쟁 할 수 있도록 기간제 리그에 참여 할 수 있습니다.	10
ST-4	사용자로서, 주식 기호에 의해 회사를 검색 할 수 있습니다.	10
ST-5	사용자로서, 그들에 투자 할 것인지 여부를 결정할 수 있도록 최근 주가 동향에 따른 그래프를 확인 할 수 있습니다.	6
ST-6	사용자로서, 선물 투자를 연구 할 수 있도록 자신의 현재의 금융 정보인 보유현금, 보유자산, 보유주식을 조회 할 수 있습니다.	5
ST-7	사용자로서, 내 포트폴리오를 구축 할 수 있도록 내가 주식을 매매 할 수 있습니다.	6
ST-8	사용자로서, 내 포트폴리오에서 실시간으로 변하는 시세에 따른 보유자산을 표시하므로 나의 재정을 추적 할 수 있습니다.	4
ST-9	사용자로서, 리그 기간 중에 승리를 향한 자신의 진행 상황을 볼 수 있습니다.	10

위의 요구 사항들은 소프트웨어의 핵심 기능을 제공하는 것으로 반영할 예정이며, 요구 사항의 일반적인 목록입니다. (더 높은 가중치 항목이 더 높은 우선순위의 역할로 먼저 시행될 것입니다.). 소프트웨어의 궁극적인 목표는 실제 주식 시장의 시뮬레이션 하는 것입니다. (ST-6 참조). 우리는 전년도에 비해 리그의 개념을 글로벌 영역으로 확장시키거나 변화시킬 계획입니다. 새로 추가 된 기능과 관련하여 실질적으로 더 높은 우선 순위로 메겨, 주식 실행 및 매매뿐만 아니라 관리자 권한에 해당하는 항목에 집중할 것입니다. 소프트웨어의 핵심 기능이 작동에 절대적으로 중요하기 때문입니다. 이 외에는 보충하는 것으로 확장 하듯이 단계적인 개발을 할 계획입니다.

6.02 핵심 요구 사항

이러한 요구 사항들은 소프트웨어의 실행 가능성과 진행에 절대적으로 중요합니다.

“사용자가 회원가입으로 계정을 만든 후 로그인을 할 수 있습니다.” : 이 작업을 구현하기 위하여 내부의 인증 시스템을 사용합니다. (ST-1, ST-2)

“사용자가 기간 제 리그에 참여 할 수 있습니다.” : 이 작업은 DB와 연결되어 계정들의 명단을 관리합니다. 계정이 없다면 리그 관전과 참여가 불가능 합니다. (ST-3)

“사용자가 주식 기호에 의해 회사를 검색 할 수 있습니다.” : 이 작업은 주식 API를 PHP로 가져와 현재 주식의 시가를 검색 할 수 있습니다. (ST-4)

“사용자는 투자할 주식의 여부를 결정할 수 있도록 최근 주가 동향에 따른 그래프를 확인 할 수 있습니다.” (ST-5)

“리그 관리자는 리그 내에서 투자자의 진행 상황을 모니터링 할 수 있는 특정 구성 설정에 액세스 할 수 있습니다.”: 이 기능은 게임 내 리그의 형성에 핵심입니다.(ST-8)

“이 프로젝트는 개인의 시장이 될 수 없습니다.” 완전 경쟁과 공평 시장 가격의 사상을 유지하기 위해, 거래는 여기에 서만 이루어지고 모든 데이터는 야후 금융에서 데이터를 가지고 옵니다. 이 소프트웨어는, 사람들이 실제 주식을 거래 할 수 있는 방법이 없습니다. 시장을 배우기 위한 자원이자, 거래 도구에 불과합니다. :(ST- 3)

5.03 사용 경우

“사용자는 자신의 거래뿐만 아니라 최근 시장 거래 동향을 볼 다른 사용자의 진행 상황에 대한 업데이트가 유지됩니다.”: (ST-9, ST-10)

“사용자로서, 주식의 실적이나 거래의주기적인 알림 서비스로 수신하도록 선택할 수 있다” : (ST-12)

5.04 비 기능적 요구 사항

“기능”: 인증의 목적은 사용자의 계정을 리그참여에 있어서 관리가 필요하기 때문이다. 키 인증 기능, 저장 암호를 암호화 할 수 있는 기능이 포함되어 있으며, 암호를 잊어버린 사용자를 위해 복구 옵션을 제공, 세션의 유효성을 검사하기 위해 쿠키를 저장합니다.

“사용성”: 이 응용 프로그램의 설계에서 중요한 점은 사용자에게 사용 및 어필의 용이성입니다.

“신뢰성”: 인터넷 또는 서버 장애시에 사용자에게 어떤 혼동이 발생하지 않도록, 모든 거래는 최종 확인으로 끝나야하고, 계정에 대한 변경 사항이 확인이 끝날 때까지 수행되지 않습니다. 사용자의 포트폴리오는 항상 일관성 있는 상태에 있을 것이고 사용자가 다시 로그인 할 때 복원됩니다. 응용 프로그램을 떠나고 나중에 돌아오는 사용자는 여전히 로그인 되었어야 하며, 서버 장애는 사용자 데이터의 백업을 유지하여 처리해야 합니다. 특정 콘텐츠 소스를 사용할 수 없는 곳은 적절한 치료 상황을 처리하기 위해 취해 져야합니다. (야후 금융).

“사양”: 우수한 성능을 갖기 위해, 웹 서버와 앱 클라이언트는 최소 하드웨어 요구 사항을 유지하여 가능한 한 경량화 해야 합니다. 효율적으로 하기 위해서는, 사용자에게 의해 개시된 임의의 태스크는 적시에 완료되어야 합니다. 웹 서버는 특히 다수의 사용자가 로그인할 때 동시 요청을 처리 할 수 있어야 합니다. 사용되는 모든 프레임 워크는 경량화 해야 하지만 고려 사항은 조기에 최적화되지 않도록 주의해야 합니다. 따라서 많은 토론을 통해서 코드를 최적화시켜야 합니다.

“지원 가능성”: 모든 서버 구성 요소를 업데이트하기 위하여 확장이 가능해야하고, 관리자만 설치 할 수 있는 모듈의 개선된 버전을 포함합니다. 스케일 아웃 목적으로, 이것은 로드 균형을 달성하기 위해 서버의 수를 추가로 포함하는 것이 용이하여야 합니다. 통제된 환경에서 새로운 기능을 테스트와 확장 기능의 유일한 목적을 위해 시스템 자체도 백업이 되어야 합니다.

6.06 개발 환경

서버 : CentOS Linux

서버언어 : PHP

클라이언트 : 안드로이드 스튜디오(1.4.1)

안드로이드 4.0.3 Icecream sandwich

하드웨어 : 각자의 노트북과 학교 실습실 컴퓨터

IP : 학교에서 제공하는 공인 아이피 (223.195.109.208)

7. Functional Requirments Specification

7.01 배우와 목표

투자자	우리의 서버에 계정을 가지고 있으며 자신의 계정에 로그인 한 사용자입니다. 이들은 우리의 최상위 이해 관계자로 간주합니다. (시장에서 최신 업데이트를 연구, 포트폴리오 보기, 어떤 종류의 명령을 실행, 리그 참여)
손님	로그인 또는 단순한 방문자하지 않은 하나의 웹 사이트에 방문자입니다. 방문자가 차 상위 이해 관계자로 간주합니다. (등록 및 네이버 계정을 통하여 계정을 생성,)
관리자(리그)	관리자는 전반적인 리그를 관리하는 담당자입니다.
DB 시스템	모든 사용자의 계좌에 대한 정보를 보유하고, 계정 생성 등의 정보를 삽입합니다. (사용자 / 이벤트에 대한 뷰에 데이터를 전송, 이벤트에 대한 새로운 데이터를 저장)
금융 API	날짜 가격까지 우리의 DB에 주식 종목 리스트들을 제공합니다. 실제 정보를 가져 오고 그에 따라 우리의 DB를 업데이트합니다.
앱 클라이언트	사용자와 시스템 사이의 중개 프로그램입니다. 사용자 데이터 볼 수 있으며, 데이터를 검색합니다.
야후! 금융	현재의 금융 정보들을 알 수 있게 해줍니다. (주식에 대한 데이터를 검색)
큐잉 시스템	스케줄링 주문 서브 시스템은 사용자 상호 작용을 차단하지 않도록, 주문을 실행 또는 비동기 취소를 할 수 있습니다. (일정 이벤트 및 시스템에 대한 우편물)

7.02 사용 사례 및 상황들

사용자는 계정을 생성하고 로그인한 즉시 사이트의 기능에 즉시 액세스 할 수 있습니다. 즉, 그들은 투자자가 수행 할 수 있는 모든 기능을 수행 할 수 있습니다.

상황별 캐주얼 사용자 설명서

UC-1 “로그인 시스템을 사용하여 계정을 생성 / 등록“ : 서버에 계정을 만들 수 있습니다. 플레이어가 쉽게 자신의 계정으로 로그인 할 수 있어야합니다. 그 후 계정을 등록키면 리그에 참여할 수 있습니다. [ST-1, ST-2 참고]
추가 : 사용자들이 이전에서 로그인 한 앱 에서 자신의 계정에 로그인할 때 다시 인증 할 필요가 없습니다.

UC-2 “시즌별 리그에 참여하기”: 등록 된 사용자는 시즌별 리그에 참여 할 수 있습니다. 이 유스 케이스는 공표된 기간 동안의 한정된 자원으로 플레이어들에게 제공해야합니다. 플레이어는 리그에서 글로벌 주식 시장에서 순이익률로 경쟁하게 됩니다. [ST-3 참고]

UC-3 “시장 데이터 보기”: 주어진 시간 동안 등록 된 사용자는 시장 데이터를 볼 수 있습니다. 사용자는 주식, 기업, 거래에 데이터를 볼 수 있어야합니다. 시장이 닫힌 후, 사용자가 데이터를 보는 경우, 리턴 된 데이터가 마지막 업데이트 된 정보이며, 그렇지 않으면, 반환 된 데이터는 실시간입니다. [ST-4, ST-5, ST-10 참고]

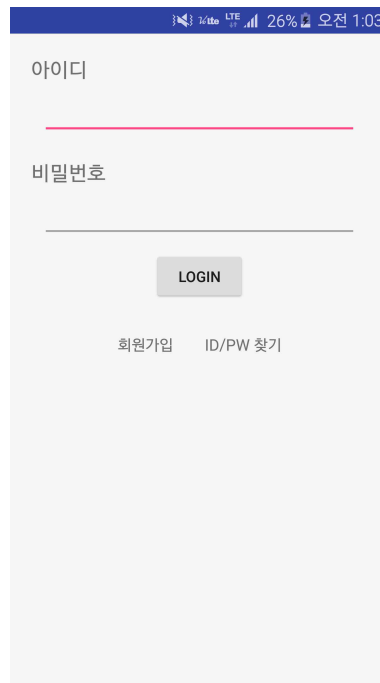
UC-4 “포트폴리오 관리”: 관리 포트폴리오는 자신의 포트폴리오를 관리 할 수 있는 사용자 등록을 할 수 있습니다. 이 사용 사례는 플레이어들이 참여하고 있는 다른 리그에서 자신의 위치를 볼 수 있도록 해야 합니다. [ST-7 참고]

UC-5 “주문”: 주문은 등록 된 사용자가 주식을 판매 / 매수 주문을 할 수 있고, 이동 또는 만들 수 있습니다. 주문은 정확한 시세로 시장의 작동 시간 내에 배치되어야 합니다. 주문 배치 한 후 포트폴리오에 대한 변경 사항은 즉시 반영됩니다. [ST-6 참고]

UC-6 “관리 작업을 수행”: 리그 관리자가 운영을 지속하기 위해 여러 조치를 취할 수 있도록 합니다. 이 유스 케이스는 리그 중 선수를 제외하는 등 모든 것을 할 수 있습니다. 단지 리그에 대한 관리 작업을 수행하기 위해 하나의 관리자 개체가 있을 수 있습니다. [ST-13, ST-14 참고]

8. Interface

8.01 User Interface

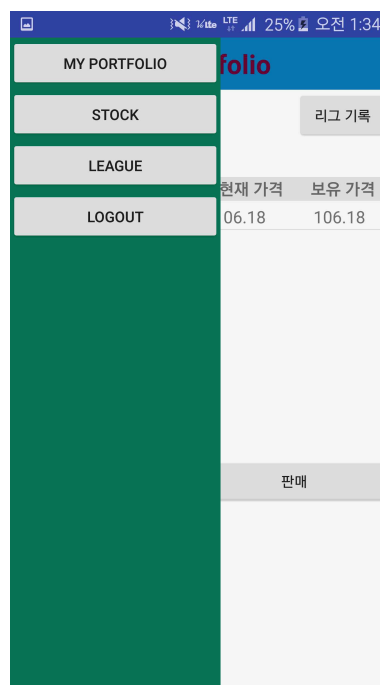


Login View

아이디와 비밀번호를 입력하는 Editbox가 있다.

처음 게임을 접하는 사용자들을 위해 회원가입 버튼이 있다.

아이디와 패스워드를 잊어버린 사용자들을 위해 ID/PW 찾기 버튼이 있다.



Menu 선택 View

각각의 View로 이동할 수 있다.

로그아웃하고 다른 계정으로 로그인 할 수 있다.



MyPortfolio View

현재 보유 현금과 보유 주식 가치를 환산한 보유 자산을 text로 제공한다.

현재 보유 주식은 주식이름, 보유량, 현재가격, 보유가격 컬럼의 ListView를 제공한다.

현재 보유한 주식을 선택하면 Editbox에 수량을 입력하여 매매를 할 수 있다.

구입 버튼을 누르면 현재 선택한 주식을 Editbox에 입력한 양 만큼 구매할 수 있다.

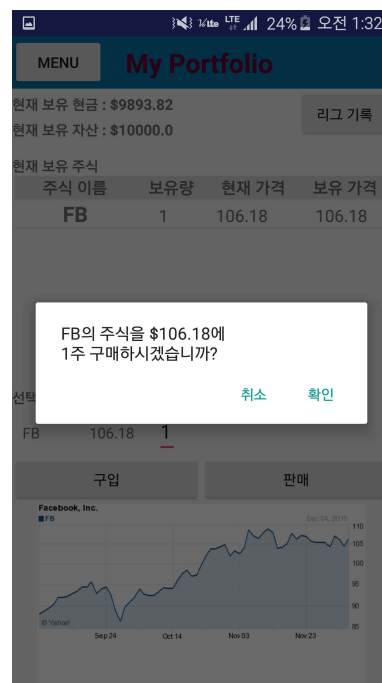
판매 버튼을 누르면 현재 선택한 주식을 Editbox에 입력한 양 만큼 판매할 수 있다.

리그 기록을 볼 수 있는 리그 기록 버튼이 있다.

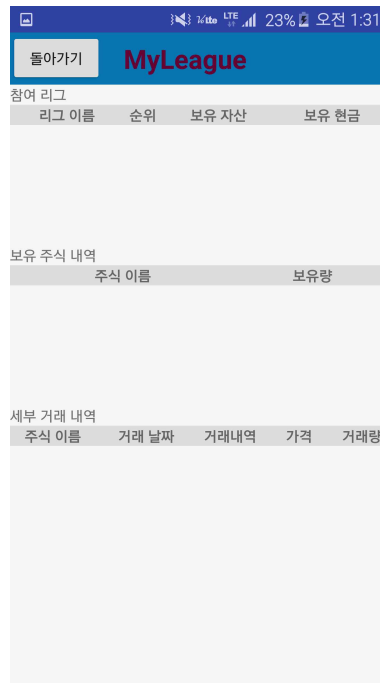
리그 기록에서 참여했던 리그에서 리그이름, 순위, 보유자산, 보유현금 컬럼으로 ListView를 제공한다.

리그 기록에서 보유 주식 내역에서 주식이름, 보유량 컬럼으로 ListView를 제공한다.

리그 기록에서 세부 거래 내역에서 주식이름, 거래 날짜, 거래내역, 가격, 거래량으로 ListView를 제공한다.

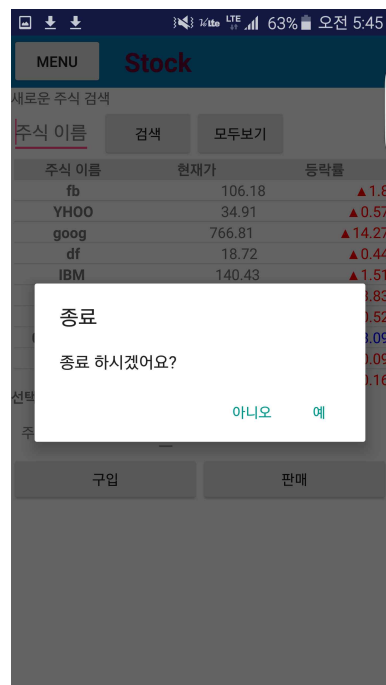


현재 보유 주식 ListView에서 선택한 주식의 그래프 이미지가 표시 된다.
구입이나 판매 버튼을 눌렀을 때 다시 한번 확인하기 위해 확인/취소 다이얼로그를 제공한다.



MyLeague View

MyPortfolio에서 리그 기록 버튼을 누르면 현재까지 참여했던 리그에 대한 기록을 볼 수 있다.
 참여했던 리그 ListView에서 레코드를 누르면 해당 리그에 보유하고 있던 주식 내역을 볼 수 있다.
 보유 주식 내역 ListView에서 레코드를 누르면 해당 주식 내역에 세부 거래 내역을 볼 수 있다.



Stock View

각종 주식에 대한 정보가 ListView에 나타난다.
 Editbox에 찾고자 하는 주식의 종목코드를 입력하고 검색 기능을 이용할 수 있다.
 뒤로가기 버튼을 이용하여 종료하면 종료 확인 메시지 다이얼로그가 뜬다.



League View

현재 진행되는 리그에 대한 정보를 제공한다.

리그는 보통 일주일 단위로 진행된다.

리그에 참가버튼을 이용해 리그에 참여할 수 있다.

8.02 서버 Interface

d UI_Server

- 주식을 구매하는 php

Purchase.php (**Id** : 유저 id,
 P_name : 주식 name
 P_amount : 주식의양)

- 주식을 판매하는 php

Sell.php (**P_id** : 유저 id
 f_name : 주식 이름
 s_count : 주식의양)

-리그 참가 하는 php

JoinLeague.php (**name** : 유저 id)

d Server_UI

-사용자 보유현금을 가져오는 php

GetCash.php(**Name** : 사용자 id)
 <id> : 사용자id
 <cash> : 보유 현금

*리그

-다른 php파일에서 개최리그id 가져오는 php

GetRecentLeagueId.php.php

-현재 개최 리그를 xml로 출력

GetRecentLeagueId.php
 <League_id> : 현재 개최 리그

-현재 개최되고 있는 리그의 정보 출력 php

GetLeagueStat.php
 <league_name> : 리그 id
 <numpeople> : 리그 참여 사용자
 <league_start_date> : 리그 개최 날짜
 <league_end_date> : 리그 마감 날짜

-리그 참가 여부 php

IsInLeague.php(**id** : 사용자id)

*주식

-모든 주식을 보여주는 php

GetStock.php
 <name>,<open>,<high>,<low>,<close>,<volume>,
 <Adj_close>

-주식검색한 주식만 보여주는 php

GetNewFinance.php(**Name** : 주식이름)
 <name>,<open>,<high>,<low>,<close>,<volume>,<Adj_close>

-자신의 보유 주식 php

GetMyPortfolio.php(**Name** : 사용자 id)
 <Stockname> : 주식이름
 <amount> : 주식량
 <open> : 시세
 <league_id> : 리그 id

-자신의 보유 주식 량 php

GetMyAmount.php(**Name** : 사용자 id
 Stock : 주식이름)
 <name> : 사용자
 <amount> : 주식량

-현재 주식 시세 php

GetOpen.php(**Name** : 주식이름)
 <name> : 주식이름
 <open> : 시세

*로그

-사용자가 참가한 리그 php

GetEachLeague.php
 <rank> : 순위
 <league_name> : 리그 이름
 <totoal_money> : 보유 자산
 <cash> : 보유 현금

-id와 리그에 해당하는 로그출력 php

GetEachLog.php(**id** : 사용자id)
 <id> : 사용자 id
 <Stockname> : 주식 이름
 <amount> : 주식량
 <League_id> : 리그 id
 <date> : 주식 매매 날짜

*기타

-사용자의 보유자산 php

GetTotalMoney.php(**Name** : 사용자 id)
 <id> : 사용자 id
 <total> : 보유자산

-사용자의 rank php

GetRank.php(**Name**:사용자id)
 <id> : 사용자 id
 <total_money> : 보유자산
 <Rank> : 순위

d Carry_On

-주식 api 가져오는 php
GetFinance.php

-주식검색버튼 기능수행 php
StockSearch.php([Name](#) : [검색할 주식이름](#))

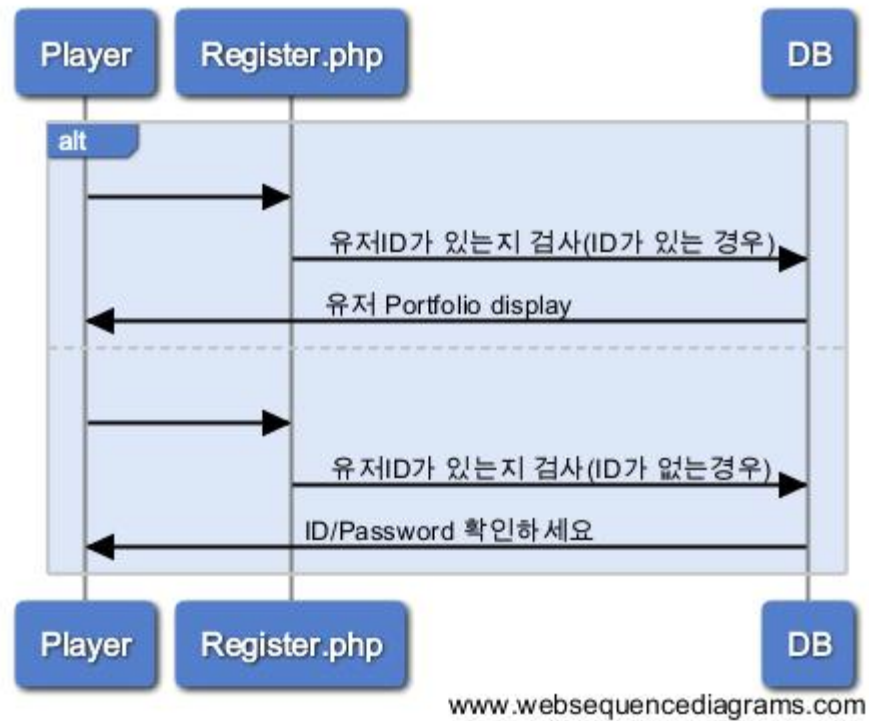
<name>,<open>,<high>,<low>,<close>,<volume>,<Adj_close>

-개최되고 있는 리그 참가자들의 보유자산을 계산 php
total_money.php

-리그 자동 생성 php
CreateLeague.php

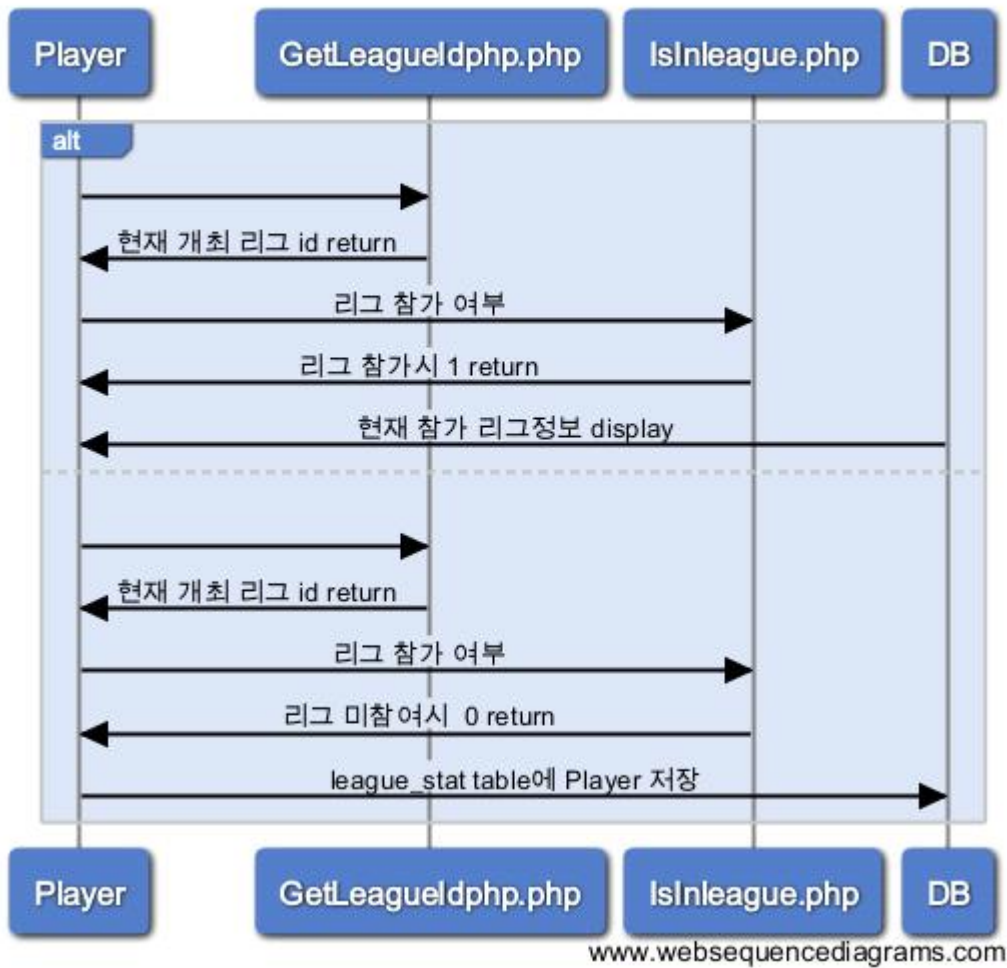
8.03 서버 Interface 도식화

UC-1 로그인



로그인시 DB에 동일 ID가 존재하는지 확인 후, 동일 ID가 있는 경우 Player의 Portfolio화면을 보여준다.
만약 DB에 동일 ID 없거나 password가 틀린 경우 ID/Password를 확인하라는 경고 창을 띄워준다.

UC-2 리그 참가

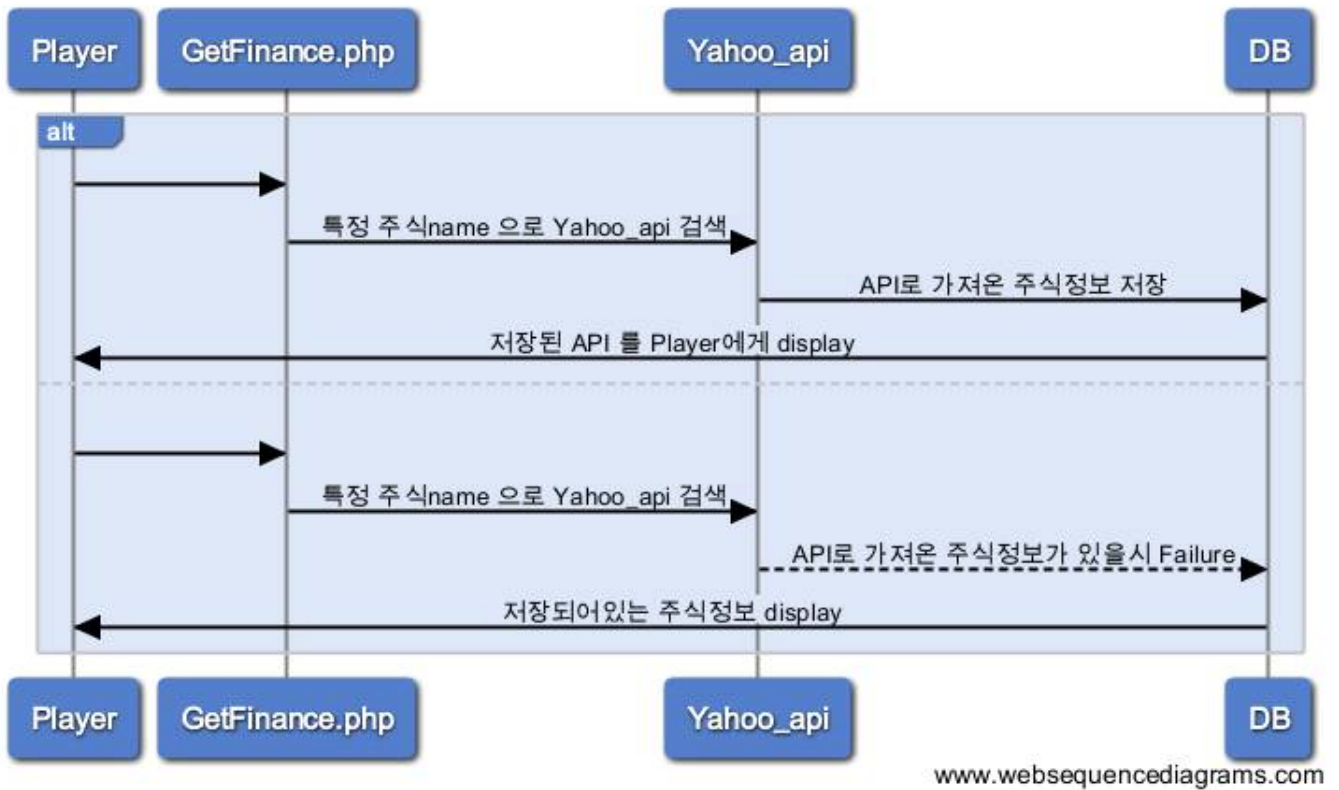


player가 리그를 참가했을 경우 UI에서는 참가버튼이 안보이게 된다. 반대의 경우로 만약 아직 player가 리그를 참가하지 않은 경우, UI에서 참가버튼이 보이게 된다.

사용자가 리그 참가 버튼을 누른 경우. GetLeagueId.php에서 현재 개최되고 있는 리그 id값을 return을 해준다. 그런 다음 IsInleague.php를 통해 player가 리그를 참가 여부를 확인하게 된다.

참가를 아직 하지 않았을 경우 league DB에 Player를 update해준다.

UC-3 시장데이터 보기



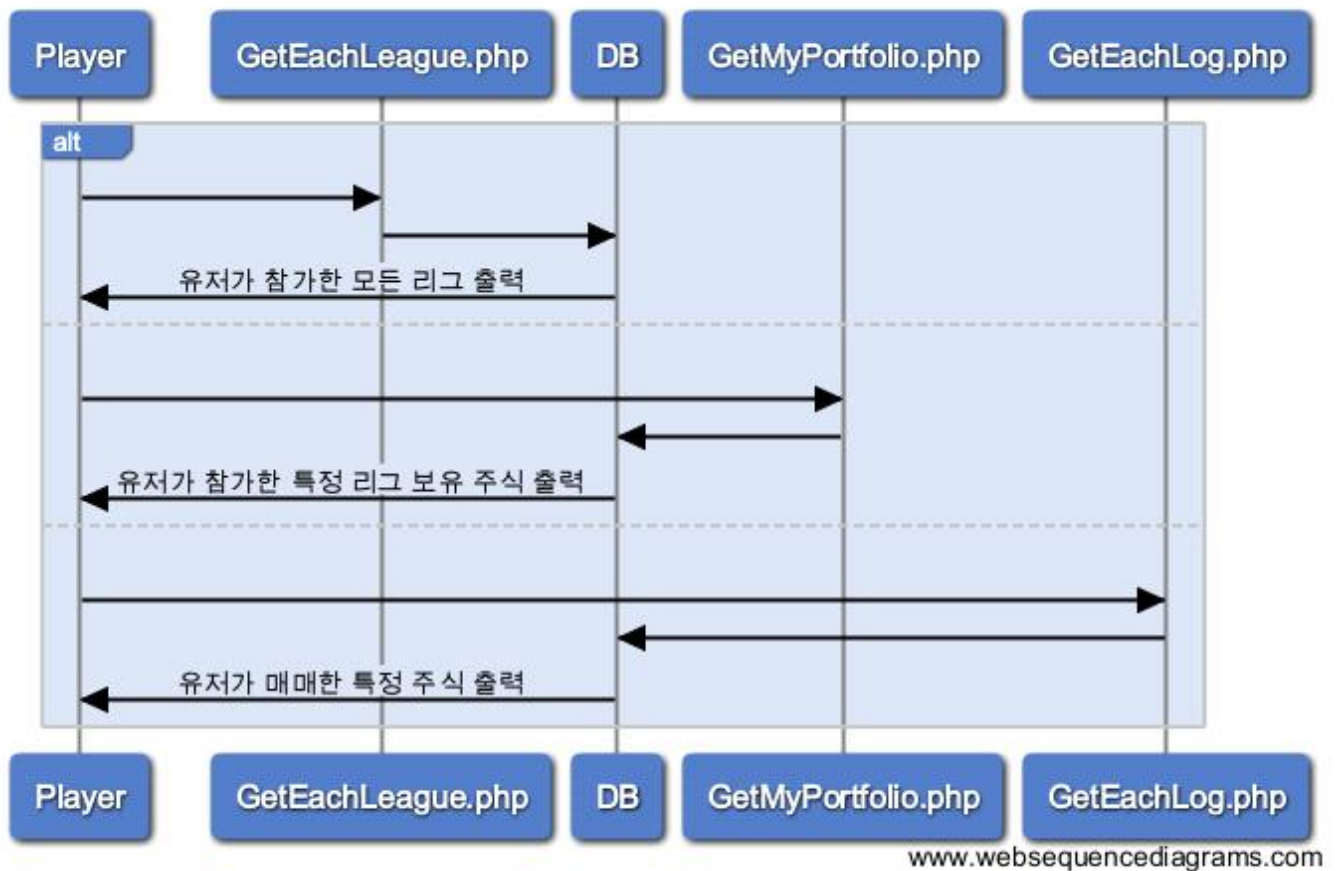
Player가 특정 주식이름으로 검색을 한 경우.

Yahoo api에서 특정 주식의 정보를 가져온다. 그런 다음 finace DB에 저장을 한다.

만약 특정주식의 정보가 DB에 있는 경우, DB에 저장하지 않고 DB에 있는 주식정보를 Player에게 바로 보여준다.

반대로 DB에 주식정보가 없는 경우, finace DB에 저장 후 Player에게 주식정보를 보여준다.

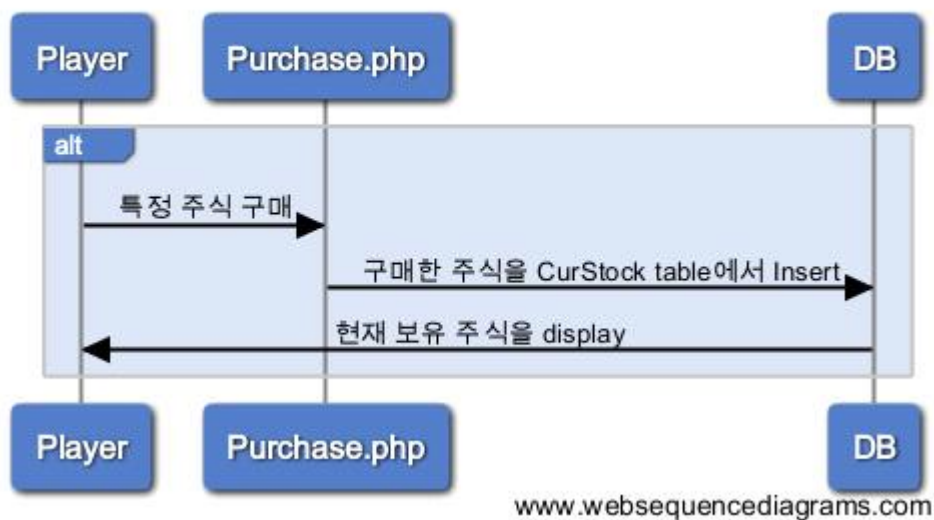
UC-4 포트폴리오 관리



리그기록을 볼 수 있다.

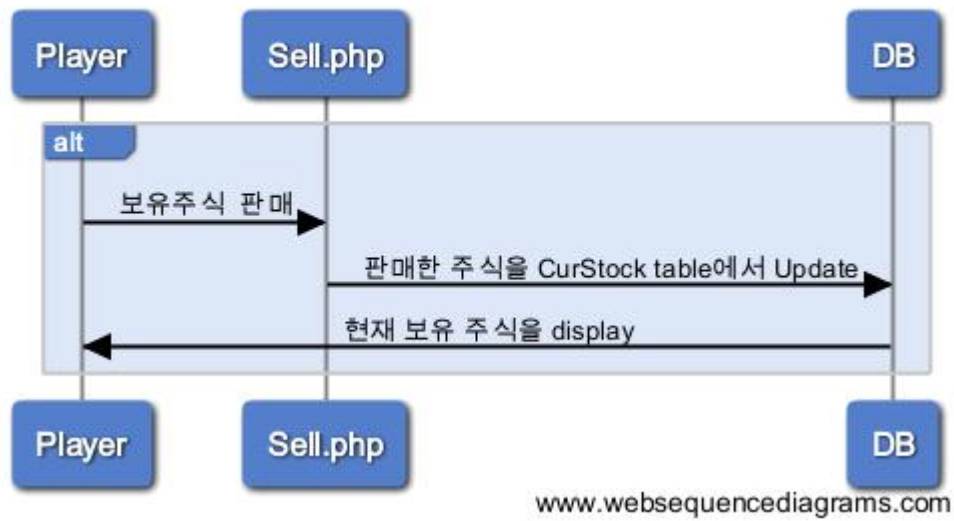
Player의 Portfolio화면에서 리그기록을 클릭시, GetEachLeague.php를 통해 지금까지 참가했던 리그 목록이 나온다. 참가했던 리그를 클릭 시, 그 때 당시 보유했던 주식들을 GetMyPortfolio.php를 통해 보여준다. 그때 당시 보유했던 특정 주식을 클릭 시 GetEachLog.php를 통해 그 주식에 대한 매매 Log기록이 보여 진다.

UC-5 주문 - 구매



구매의 경우, player가 특정주식에 대해 구매버튼을 누른경우 Purchase.php를 통해 주식을 구매하게된다.

UC-5 주문 - 판매



판매의 경우, player가 보유주식에 대해 판매버튼을 누른경우 Sell.php를 통해 보유주식을 판매하게 된다.
만약 Player가 리그에 참가하지 않았을시 구매 또는 판매가 불가능하다.

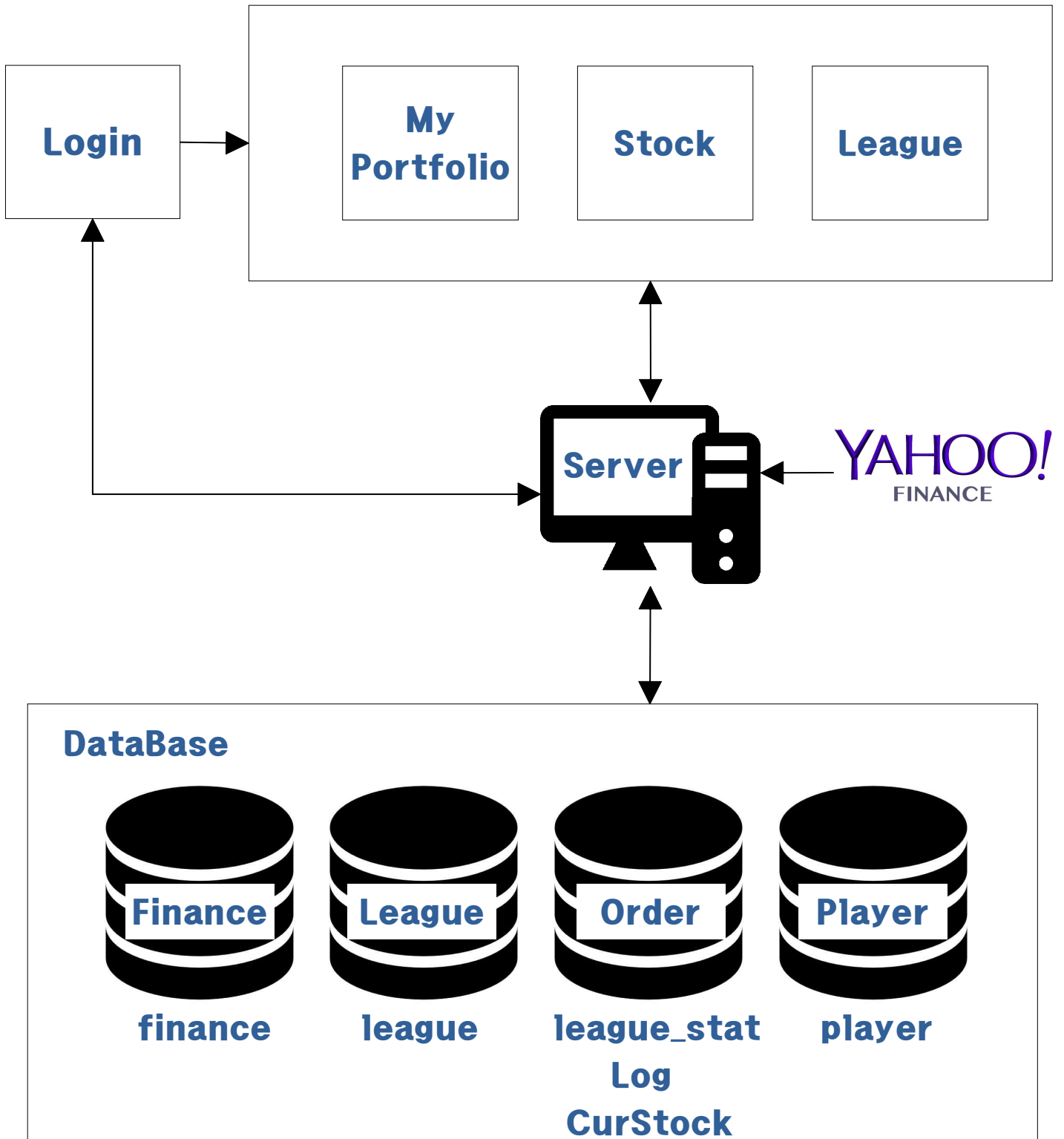
9. Domain Model

Domain Model 컨셉은 6장 Use Case에 들어있는 책무들로부터 파생된다.

Responsibility	Concept
R1 : 로그인 시스템을 사용하여 계정을 생성 / 등록	Login
R2 : 시즌별 리그에 참여하기	League
R3 : 시장데이터 보기	Finance
R4 : 포트폴리오 관리	My portfolio
R5 : 주문	Stock
R6 : 관리 작업을 수행	Server

9.01 Domain Diagram

도시 한 도면 (우리의 도메인 모델)은 우리의 시스템은 다른 서브 시스템과 상호 작용으로 분해되는 방법을 보여준다.
각 상자는 우리의 시스템의 엔티티를 나타내고,
각 화살표는 시스템 내의 각각의 엔티티가 다른 항목과 통신하는 기능을 나타낸다.



Login

로그인 엔티티는 말 그대로 아이디와 패스워드를 입력하고 계정을 가지고 게임을 시작하는 첫 단계이다. 계정을 생성하기 위해 회원가입을 할 수 있다. 아이디와 패스워드를 잊었을 경우를 대비하여 ID/PW찾기 기능도 제공한다.

My Portfolio

My Portfolio를 사용함의 목적은 사용자 본인의 현재 게임 정보를 얻는 것에 있다. 먼저 사용자가 가지고 있는 현금과 주식 가치를 환산하여 측정한 자신의 자산을 볼 수 있다. 현재 보유하고 있는 주식을 볼 수 있다. 현재 가지고 있는 주식을 매매할 수 있는 기능을 제공한다. 현재 보유하고 있는 주식들의 그래프를 제공한다. 자신이 현재까지 게임을 했던 정보들을 모두 접근할 수 있기 위해 리그 기록도 제공한다. 리그 기록은 참여 리그 목록, 해당 리그의 보유했던 주식 내역, 해당 보유 주식의 세부 거래 내역을 볼 수 있다.

Stock

Stock을 사용함의 목적은 주식들을 검색함에 있다. 주식이름을 입력하고 검색하고 여러 주식들을 한 눈에 비교한다. 구매하고자 하거나 판매하고자 하는 주식의 양을 입력하고 구입 또는 판매할 수 있다. 주식을 클릭하면 해당 주식의 그래프를 확인할 수 있다. 찾고자 하는 주식 목록에 없다면 검색기능을 이용하여 해당 주식을 목록에 추가 시킬 수 있다.

League

게임에 참가할 수 있다. 현재 어떤 리그가 개최 돼 있는지 확인할 수 있다. 현재 리그에 참석한 유저들의 목록을 제공한다. 현재 진행 중인 리그의 랭킹을 볼 수 있다.

Server

서버에서는 각 엔티티에서 데이터베이스로 접근 할 때의 연산과 내부적으로 랭킹과 보유자산 연산 그리고 야후 API에서 주식 데이터를 가지고 오는 연산을 수행한다.

Finance Database

야후 API에서 xml로 제공하는 주식 데이터를 실시간으로 저장한다. 현재가, 시가, 종가, 데이터를 가지고 와서 finance table에 저장한다.

League

서버에서 실제 게임 league에 해당하는 데이터를 저장한다. 현재일을 기준으로 하여 보통은 일주일 단위로 리그가 개최된다. 개최될 때마다의 각각의 리그 id가 주어진다. 당일의 날짜와 시작날짜 끝나는 날짜가 저장된다. 각각의 날짜 데이터를 가지고 새로운 리그를 개최하고 폐막한다.

Order

주식의 구매와 판매에 관한 모든 데이터와 보유자산, 랭킹을 저장한다. league_stat table에서 랭킹과 자산현황이 저장된다. 현재 보유하고 있는 주식과 양을 CurStock table에 저장한다. Log table에서 구매하고 판매했던 모든 기록이 남는다.

Player

회원가입을 하고 등록한 계정의 신상 정보들이 player table에 모두 저장된다.

9.02 데이터베이스 테이블 Diagram

