



مهندسی نرم افزار ۱



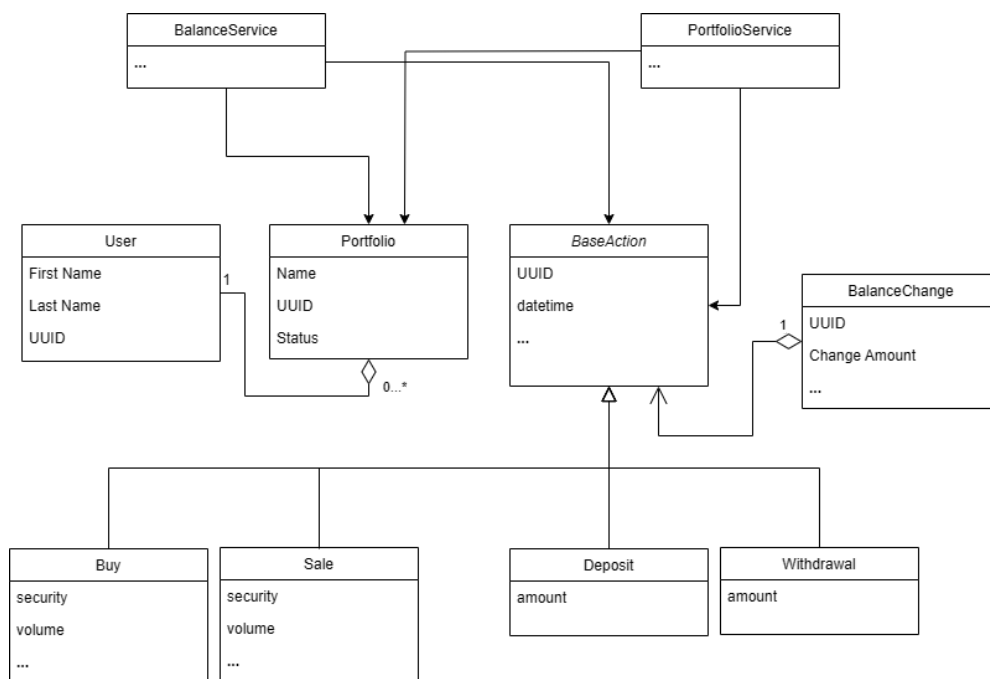
فاز چهارم

پیاده سازی مورد کاربرد

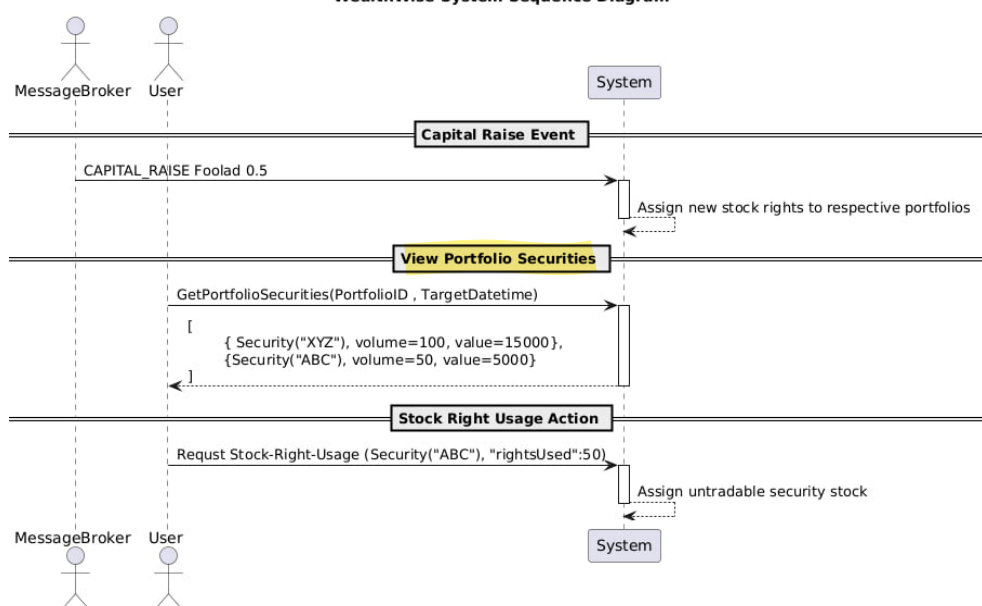
WealthWise

نیمسال دوم سال تحصیلی ۱۴۰۳-۰۴

تا این فاز از پروژه شما موفق به شناسایی موارد کاربرد، توصیف دقیق آن‌ها و طراحی مدل دامنه شده اید. در این فاز باید به کمک موارد تئوری قبلی به پیاده‌سازی سیستم در سطح کد بپردازید. برای سرعت بخشیدن به فرآیند شما، Operation های پایه و Entity های لازم شما از قبل پیاده‌سازی شده است. Backbone اولیه سیستم که روی صفحه گیت‌هاب [WealthWise](#) موجود است از روی دامین مدل و دیگرام توالی سیستم زیر پیاده‌سازی شده.



WealthWise System Sequence Diagram



توجه داشته باشید که ممکن است مواردی با مدل دامنه زیر تداخل داشته باشد که تغییرات کوچک حین پیاده‌سازی با توجه به فریم‌ورک استفاده شده (Spring Boot) اجتناب ناپذیر است.

در ادامه توضیحاتی بابت خواسته‌های این فاز از شما و پیاده‌سازی فعلی و مواردی که باید به آن اضافه کنید گفته خواهد شد.

پیشنهاد می‌شود پیش از مطالعه صورت پروژه ، فیلم مربوط به توضیح معماری سیستم فعلی را مشاهده کنید.

مروری بر روند افزایش سرمایه در سیستم ما

شرکت‌ها با منتشر^۱ کردن آگهی افزایش سرمایه^۲ در سایت سازمان بورس ، اقدام به اطلاع رسانی به سیستم می‌کنند . سیستم به صورت خودکار تمام پورتفولیو های موجود در سیستم را بررسی نموده و پورتفولیوهایی که از قبل سهامی از شرکت X(در اینجا شرکتی که افزایش سرمایه داشته) را شناسایی کرده و سپس اقدام به اختصاص یک میزان سهام حق تقدم با نماد "حق تقدم X" به پورتفولیو با توجه به میزان دارایی از آن سهام می‌کند.

فرض کنید که سه پورتفو در سیستم ما موجود باشد که دارایی های آنها به صورت زیر است :

سهام \ پورتفو	سهام شرکت A	سهام شرکت B
پورتفو ۱	۱۰۰ سهم	۱۰۰ سهم
پورتفو ۲	۲۰۰ سهم	۰ سهم
پورتفو ۳	۰ سهم	۵۰۰ سهم

^۱ Publish

^۲ Capital Raise

اگر شرکت A افزایش سرمایه دهد و فرضا به ازای هر سهام شرکت ۲ سهم حق تقدم به سهامداران اختصاص یابد، پس از افزایش سرمایه دارایی پورتنفو های سیستم به صورت زیر خواهد بود :

سهام \ پورتنفو	سهام شرکت A	سهام شرکت B	سهام حق تقدم شرکت A
پورتنفو ۱	۱۰۰ سهم	۱۰۰ سهم	۲۰۰ سهم
پورتنفو ۲	۲۰۰ سهم	۰ سهم	۴۰۰ سهم
پورتنفو ۳	۰ سهم	۵۰۰ سهم	۰ سهم

حالا پس از اینکه این سهام های حق تقدم به پورتنفو های متناظر اختصاص یافت، سبگردان (کاربر سیستم) موظف است که حق تقدم های تخصیص یافته را تعیین تکلیف کند.

همان طور که بخاطر دارید چند واکنش نسبت به حق تقدم ها می توان داشت:

۱ . خرید یا فروش سهام حق تقدم در بازار

۲ . استفاده از حق تقدم

۳ . عدم استفاده از حق تقدم

حالت ۱ ، در پروژه پیاده سازی شده است . مشابه دیگر سهام ها کاربر میتواند اکشن^۳ های خرید و یا فروش بر روی سهام حق تقدم تعریف کند. **حالت ۲ در این گام از پروژه انجام می شود.** که در این حالت ، کاربر با پرداخت پول مشخصی اقدام به تبدیل سهام حق تقدم به سهام اصلی خود می کند ، نکته ای که وجود دارد این است که با پرداخت این پول ، این سهم های حق تقدم به سهام اصلی تبدیل می شوند اما قابلیت معامله (Tradability) تا زمان مشخصی را ندارند . این زمان مشخص نیز در قالب یک آگهی دیگری به نام ثبت افزایش سرمایه است، که با ارائه این آگهی، سهام هایی از شرکت X که غیر قابل معامله^۴ هستند ، قابل معامله^۵ میشوند.

^۳ Action

^۴ Untradable

^۵ Tradable

محاسبه سهم‌های یک پورتفولیو در یک روز

برای آشنایی بیشتر با موجودیت^۶ های سیستم ابتدا یک سرویس بنویسید که ورودی های آن یک پورتفولیو و یک زمان^۷ باشد ، و خروجی آن دارایی‌های سهام کاربر باشد. دقت نمایید که خروجی بر اساس اسم سهام مرتب سازی شده باشد و خروجی به صورت یک لیستی از یک نمونه^۸ های یک کلاس باشد. که این کلاس در خود سهام (Security) ، تعداد سهام (Volume) ، ارزش روز (Value) باشد که ارزش روز به کمک قیمت روز سهام از طریق SecurityPriceRepository بدست می‌آید .

برای انجام این کار میتوانید از سرویسی که برای محاسبه موجودی پورتفولیو نوشته شده است ، ایده بگیرید . همچنین حتما نوشتن تست برای این سرویس را فراموش نکنید .

^۶ Entity

^۷ Datetime

^۸ instance

پیاده سازی فرآیند افزایش سرمایه

آگهی افزایش سرمایه

در سیستم از طریق Message Broker آرتمیس ، آگهی افزایش سرمایه منتشر می شود .
در این سیستم شما موظف به پیاده سازی یک Listener ساده هستید که بتواند پیام هایی که در این Message Broker منتشر⁹ می شود را مصرف¹⁰ کند. نحوه مصرف آن به این صورت خواهد بود که Listener شما باید شناسایی کند که این آگهی متعلق به کدام شرکت بوده و میزان اختصاص حق تقدم به ازای هر سهام شرکت چقدر است و به پورتفولیو های متناظر این سهام را اختصاص بدهد .
فرمت آگهی منتشره به شکل زیر است :

CAPITAL_RAISE <Security Symbol> <Stock Right Amount Per Share>

مثال :

CAPITAL_RAISE Foolad 0.5

مشخص می کند که هر پرتفولیویی که سهام Foolad را داشته باشد به ازای هر سهم آن 0.5 سهم حق تقدم دریافت کند. فرض کنید در صورتی که مقدار سهام عدد صحیح نبود به پایین گرد می شود .

برای این کار شما موظف به تعریف یک اکشن¹¹ جدید در سیستم هستید. این اکشن با نام CapitalRaise ، به خودی خود تغییری در موجودی¹² پورتفولیو ایجاد نمی کند ، صرفاً تعدادی سهم جدید از طریق SecurityChange به پورتفولیو مربوطه اضافه می کند که سهام آن در واقع سهام حق تقدم X است .
نکته: برای سادگی کار اگر نماد شرکتی X باشد ، نماد حق تقدم آن را HX در نظر می گیریم و فرض می شود که تمامی این سهام ها در سیستم در SecuirtyRepository موجود هستند.

⁹ Publish

¹⁰ Consume

¹¹ Action

¹² Balance

خرید و فروش سهام حق تقدم

در این قسمت نیازی نیست کار خاصی انجام دهید، فقط با نوشتن چند تست مطمئن شوید که کاربر بعد از تخصیص حق تقدم ها، قابلیت ثبت اکشن های خرید و فروش را دارد. نکته: فرض میشود که کاربر اکشن ناممکنی مانند فروش سهامی که در اختیار ندارد یا خرید با هزینه بیشتر از موجودی خود انجام نمی دهد. در صورتی که تمایل دارید می توانید این دسته کیس ها را با انتخاب خود مدیریت کنید.

استفاد از حق تقدم

در این قسمت شما موظف به نوشتن یک اکشن جدید در سیستم هستید. این اکشن به کاربر این قابلیت را میدهد که با پرداخت ۱۰۰ تومان به ازای هر سهم حق تقدم، این سهام را به سهام اصلی شرکت تبدیل کند. اما تا زمان آگهی ثبت افزایش سرمایه مربوطه، این سهم غیر قابل معامله باقی می ماند.

همچنین برای اینکه این تبدیل را انجام دهید، نیازی به پاک کردن سهام های حق تقدم نیست. با توجه به معماری پروژه، شما نیاز دارید که یک SecurityChange با میزان منفی تعداد حق تقدم برای پورتفولیو ثبت کرده و یک SecurityChange دیگر با میزان مثبت تعداد حق تقدم برای سهام اصلی ثبت نمایید. در این صورت اگر کاربر قصد داشته باشد که سهام های روز قبل از استفاده از حق تقدم خود را ببیند، کافی است سیستم برآیند تمام SecurityChange های قبل از استفاده از حق تقدم را محاسبه کند.

و اگر قصد داشته باشد که سهام های روز بعد از استفاده از حق تقدم خود را ببیند، سیستم کافی است برآیند تمام SecurityChange های بعد از استفاده از حق تقدم را محاسبه کند. با استفاده از این روش که ویژگی اصلی معماری **Event Sourcing**¹³ است، شما به راحتی می توانید بین روز های مختلف سوییچ نمایید.

نکته: نام این اکشن را **StockRightUsage** بگذارید. همچنین دقت داشته باشید که SecurityType مربوط به سهم های حق تقدم StockRight است.

¹³ <https://martinfowler.com/eaDev/EventSourcing.html>

موارد تحویلی

پس به طور خلاصه اگر بخواهیم کار هایی که موظف به انجام آن هستید را لیست کنیم به شرح زیر هستند :

- 1 - سرویسی برای محاسبه ی سهام های هر پورتفولیو بر اساس زمان
- تست های مربوط به سرویس بالا
- 2 - پیاده سازی مصرف کننده ی آگهی های منتشر شده در message broker
- اکشن مربوط به افزایش سرمایه
- تست های مربوط به اکشن افزایش سرمایه
- 3 - تست های خرید و فروش برای سهام های حق تقدم
- 4 - اکشن استفاده از حق تقدم
- تست های اکشن استفاده از حق تقدم

به طور کلی سیستم باید با استفاده از تعدادی API با دنیای بیرون در ارتباط باشد. برای مثال اگر کاربری بخواهد برای پورتفولیو خود یک اکشن Buy ثبت کند باید بتواند این کار را از طریق یک Create API (با استفاده از اندپوینت post از طریق Spring Boot) انجام دهد. اما با توجه به اینکه در این درس تمرکز ما بر معماری سیستم بوده؛ برای اینکه از درستی عملکرد سیستم مطمئن شویم به نوشتن تست های ساده ی اتوماتیک اکتفا می کنیم.

نکات تکمیلی

- کد های اولیه پروژه در صفحه گیت‌هاب [se1404/WealthWise](https://github.com/se1404/WealthWise) در دسترس است .
- در نوشتن برنامه ، به اصولی طراحی که در کلاس گفته شده است (مانند اصول SOLID) پایبند باشید .
- استفاده از Design Pattern ها حتما ، به شما توصیه میشود و بخشی از نمره شما را در برمیگیرد . پس سعی کنید با بررسی Design Pattern های مختلف سایت [Guru¹⁴](https://refactoring.guru/design-patterns/catalog) حتما دیزاین پترن های مناسب را در طراحی خود بگنجانید.
- هرگونه سؤالی را ترجیحا در گروه درس مطرح کنید تا مشکل شما به بحث گذاشته شود و دیگر دانشجویان نیز در جریان آن قرار بگیرند.
- دست شما در تغییر بخش های دیگر پروژه باز است . اگر فکر کردید در قسمتی از پروژه خطایی یا باگی وجود دارد میتوانید آن را اصلاح نمایید . فقط توجه داشته باشید که به معماری اصلی پروژه آسیبی وارد نکنید. همچنین اگر باگ اساسی در کد های اولیه پروژه مشاهده کردید می توانید با باز کردن یک Pull Request در صفحه گیت‌هاب درس و در صورت تایید باگ نمره امتیازی برای گروه خود دریافت کنید.
- اگر شک و شبهه‌ای دارید، نباید پیش‌فرضی در نظر بگیرید و باید از استاد و دستیاران آموزشی بپرسید. دستیار آموزشی مانند مالک محصول (*Product Owner*) است و در نهایت رضایت مالک محصول برای شما مهم خواهد بود. (Alireza Hosseini - Mahdi Haji Hosseiny)
- در قسمت تحویل صرفا کافی است لینک گیت‌هاب پروژه خود را قرار داده و آیدی گیت‌هاب [se1404](https://github.com/se1404/WealthWise) را به پروژه خود اضافه کنید. توجه داشته باشید که تا زمانی که پروژه از شما تحویل گرفته نشده وضعیت ریپازیتوری خود را به حالت Public تغییر ندهید!

موفق باشید

¹⁴ <https://refactoring.guru/design-patterns/catalog>