



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

پروژه مهندسی نرم افزار ۲

طرح جامع پروژه کارگزاری بورس

استاد درس:

دکتر گوهری

طراحان پروژه:

سپهر میرنصراللهی

علیرضا نعمتی

مهندی روحانی

حسین زارعی نژاد

پاییز ۱۴۰۴

بخش اول: معرفی کسب و کار و اهداف کلان

در این پروژه، یک سامانه کارگزاری بورس آنلاین طراحی و پیاده‌سازی خواهد شد که به کاربران اجازه می‌دهد تا به صورت آنلاین اقدام به خرید و فروش سهام کنند. هدف اصلی این سامانه ایجاد یک پلتفرم شفاف و قابل اعتماد برای معاملات آنلاین سهام است. این سیستم از فناوری‌های روز برای ایجاد امنیت و شفافیت بیشتر در عملیات خرید و فروش سهام استفاده خواهد کرد.

اهداف کلان پروژه:

1. ایجاد بستر شفاف و کاربرپسند برای خرید و فروش سهام:

- فراهم آوردن پلتفرم آنلاین برای انجام معاملات بورسی به صورت سریع و امن.
- طراحی رابط کاربری ساده و کاربرپسند که برای تمامی کاربران با هر سطح دانش از بازار بورس مناسب باشد.

2. افزایش اعتماد به سیستم با استفاده از تکنولوژی‌های پیشرفته:

- استفاده از بلاکچین برای ثبت تراکنش‌ها به منظور اطمینان از شفافیت و عدم تغییر اطلاعات.
- استفاده از برنامه‌نویسی سوکت^۱ برای ارسال اعلان‌های لحظه‌ای وضعیت سفارش‌ها و معاملات.

3. کاهش هزینه‌ها و حذف واسطه‌ها:

- ایجاد سیستم‌های میکروسرویس برای تفکیک هر بخش از سیستم و افزایش مقیاس‌پذیری.
- فراهم آوردن امکان خرید و فروش مستقیم بدون نیاز به واسطه‌های اضافی.

4. پشتیبانی از تراکنش‌های واقعی و شفاف:

- ثبت تمامی اطلاعات خرید و فروش در بلاکچین خصوصی برای حفظ اصالت و جلوگیری از دستکاری داده‌ها.
- ارائه گزارشات شفاف از تاریخچه خرید و فروش و وضعیت حساب‌های کاربران.

اهداف سازمان:

- ایجاد بستری امن و شفاف برای خرید و فروش سهام که به اعتماد کاربران نسبت به سیستم کمک کند.
- حمایت از کاربران با فراهم آوردن ابزارهای دقیق و شفاف برای انجام معاملات و گزارشات دقیق از تمامی فعالیت‌ها.
- افزایش دسترسی به اطلاعات بازار بورس به طور آنی و آنلاین، به طوری که کاربران بتوانند لحظه به لحظه تغییرات بازار را پیگیری کنند.
- پشتیبانی از مقیاس‌پذیری و انعطاف‌پذیری سیستم برای پشتیبانی از تراکنش‌ها در مقیاس‌های بزرگ‌تر در آینده.

¹ Socket Programming

بخش دوم: ویژگی‌ها و مستندات سامانه

ویژگی‌های اصلی سامانه:

1. مشاهده لیست سهام:

- کاربران می‌توانند لیست سهام موجود در بازار بورس را مشاهده کنند، با جزئیات شامل قیمت‌ها و اطلاعات شبیه‌سازی شده یا واقعی، برای کمک به تصمیم‌گیری‌های سرمایه‌گذاری.

2. ثبت سفارش خرید و فروش سهام:

- کاربران قادر خواهند بود سفارش‌های خرید یا فروش سهام خود را ثبت کنند و این سفارش‌ها به طور خودکار توسط موتور تطبیق سفارش آسیستم، تطبیق داده خواهند شد.

3. اطلاع‌رسانی لحظه‌ای وضعیت سفارش‌ها از طریق سوکت:

- سیستم از برنامه‌نویسی سوکت برای ارسال اعلان‌های لحظه‌ای وضعیت سفارش‌ها و معاملات به کاربران استفاده خواهد کرد.

4. ثبت تراکنش‌های نهایی در بلاکچین خصوصی:

- تمامی تراکنش‌های خرید و فروش در سیستم به صورت امن در بلاکچین خصوصی ذخیره می‌شوند تا از امنیت و عدم دستکاری داده‌ها اطمینان حاصل شود.
- معرفی راهکارهای موجود برای پیاده سازی بلاکچین خصوصی.

- معرفی قراردادهای هوشمند استاندارد بر بلاکچین های پایه EVM(Ethereum Virtual Machine) مناسب برای این کار مانند ERC20، ERC1155

5. نمایش سبد سهام و وضعیت معاملات:

- کاربران می‌توانند سبد سهام خود را مشاهده کنند، که شامل تمامی سهام‌هایی است که در حساب کاربری خود دارند و وضعیت هر یک از این سهام‌ها را پیگیری کنند.

6. ورود با استفاده از بلاکچین(امتیازی)

- کاربران می‌توانند به جای استفاده از راهکارهای عمومی ورود در اینترنت نسل 2.0، از راهکارهای جدید مانند EIP-4361 Sign-In with Ethereum چیزی که میتواند در راستای غیر مرکز شدن سرویسمان کمک کند.

² Matching Engine

مستندات مورد نیاز:

1. سند تحلیل ریسک^۳:

- این سند باید ریسک‌های مرتبط با بازار بورس، نوسانات قیمت، و اختلالات در انجام معاملات را شناسایی کرده و استراتژی‌هایی برای مدیریت و کاهش این ریسک‌ها ارائه دهد.

2. سند چشم‌انداز^۴:

- این سند اهداف بلندمدت سامانه، استراتژی‌های رشد، و نیازهای کاربران را مشخص کرده و به تیم کمک می‌کند که نقشه‌راهی برای توسعه سیستم ایجاد کنند.

³ risk analysis

⁴ Vision Document

بخش سوم: معماری، مدل‌سازی و نمودارهای سیستم

در این بخش، مدل‌سازی سامانه با استفاده از نمودارهای استاندارد UML انجام می‌شود. نمودارها باید با ابزارهای تخصصی رسم شده و به صورت PDF ارائه شوند.

نمودارهای مورد نیاز:

1. Use Case Diagram (نمودار موارد کاربرد):

- جستجوی سهام: این قابلیت به کاربران اجازه می‌دهد تا سهام‌های مختلف را براساس معیارهایی مانند نوع، قیمت، تاریخ و زمان، و حجم معاملات جستجو کنند.
- ثبت سفارش خرید و فروش: کاربران می‌توانند سفارش‌های خرید یا فروش سهام خود را ثبت کرده و سیستم آن‌ها را پردازش می‌کند.
- نمایش سبد سهام: مشاهده جزئیات سبد سهام کاربران و وضعیت فعلی معاملات.

2. Class Diagram (نمودار کلاس‌ها):

- نمایش کلاس‌های اصلی و مربوط به سیستم:
 - Stock (سهام): اطلاعات مربوط به سهام مانند قیمت، تاریخ، و تعداد موجود.
 - User (کاربر): اطلاعات کاربران سیستم (مشتریان).
 - Order (سفارش): جزئیات سفارش‌های ثبت شده (خرید و فروش).
 - Transaction (تراکنش): اطلاعات مربوط به تراکنش‌های خرید و فروش سهام.
 - Portfolio (سبد سهام): اطلاعات موجودی و سبد سهام هر کاربر.
 - Blockchain Ledger (دفتر کل بلاکچین): برای ذخیره تراکنش‌های خرید و فروش در بلاکچین.
 - Notification (اعلان‌ها): ارسال اعلان‌های لحظه‌ای وضعیت سفارش‌ها به کاربران.

3. Sequence Diagram (نمودار توالی):

- فرآیند ثبت سفارش خرید و فروش: نمایش مراحل مختلف از زمان ثبت سفارش توسط کاربر تا تطبیق آن و ثبت در بلاکچین.
- فرآیند ارسال اعلان وضعیت سفارش: نمایش ارسال اعلان به مشتریان از طریق سوکت پس از تغییر وضعیت سفارش.

Activity Diagram (نمودار فعالیت) .4

- نمایش جریان کار سیستم در فرآیند ثبت سفارش، تایید و تطبیق سفارش‌ها، پردازش و ثبت آن‌ها در بلاکچین، و اطلاع‌رسانی به کاربر.
- فرآیند ارسال اعلان وضعیت سفارش: از زمان ثبت سفارش تا ارسال به کاربر.

Component Diagram (نمودار کامپوننت) .5

- نمایش تفکیک سامانه به کامپوننت‌های مختلف:
 - User Management Service (مدیریت کاربران): شامل ثبت‌نام و ورود کاربران.
 - Order Management Service (مدیریت سفارش‌ها): برای ثبت، تطبیق و پردازش سفارش‌های خرید و فروش.
 - Blockchain Service (خدمات بلاکچین): برای ثبت تراکنش‌ها و طریقه اتصال به قرارداد هوشمند.
 - Notification Service (خدمات اطلاع‌رسانی): برای ارسال اعلان‌های لحظه‌ای به کاربران.
 - API Gateway (درگاه API): برای ارتباط با سرویس‌ها و ارائه خدمات به کاربران.

Deployment Diagram (نمودار استقرار) .6

- نمایش نحوه استقرار سیستم در محیط‌های مختلف، از جمله استفاده از کلاسترها سرور و سیستم‌های توزیع شده برای مقیاس‌پذیری و دسترس‌پذیری.
- استفاده از کلاسترها Kubernetes برای مدیریت میکروسرویس‌ها و Prometheus و Grafana برای مانیتورینگ سیستم.

نمایش نحوه استقرار قرارداد هوشمند بر روی بلاکچین و انجام تراکنش بر روی آن.

Infrastructure as Code (امتیازی) .7

- ارائه نمودار Terraform یا معادل آن برای تعریف و استقرار زیرساخت‌های سامانه.
- بهطور خودکار زیرساخت‌ها را ایجاد می‌کند، بدون نیاز به دخالت دستی، و اطمینان از قابل اطمینان بودن و مقیاس‌پذیری سیستم.

بخش چهارم: فرآیند توسعه چابک (اسکرام)

برای توسعه سامانه از اسکرام استفاده می شود تا تعامل، انعطاف پذیری و پاسخگویی به تغییرات تضمین گردد. ابتدا در سامانه Jira یک پروژه برای محصول جدید بسازید و نام محصول و Backlog مربوط به آن را تعریف کنید. در این مرحله، تمام نیازمندی ها و ویژگی ها باید به Backlog اضافه شوند.

گام های کلیدی در اسکرام برای پروژه کارگزاری بورس آنلاین:

1. Product Backlog (لیست نیازمندی ها):

- شامل تمامی نیازمندی های سامانه به تفکیک ویژگی های اصلی همچون ثبت سفارش خرید و فروش، تطبیق سفارش ها، گزارش های لحظه ای وضعیت، و ثبت تراکنش ها.
- این نیازمندی ها باید اولویت بندی شوند تا تیم توسعه بر اساس اولویت ها اقدام کند.

2. Story و Epic :

- Epic ها بزرگ و کلی مانند "مدیریت سفارش های خرید و فروش" و "ثبت تراکنش ها" به Story های کوچکتر تقسیم می شوند (برای مثال: "ثبت سفارش خرید"، "تطبیق سفارش خرید و فروش").
- تقسیم Story ها به فضول مختلف یا Quarter برای پیگیری راحت تر در هر اسپرینت.

3. Sprint Planning (برنامه ریزی اسپرینت):

- انتخاب Story هایی از Backlog که برای رسیدن به اهداف Sprint لازم هستند.
- تقسیم وظایف تیم و اولویت بندی آنها به طوری که در هر اسپرینت تکمیل شوند.

4. Sprint Execution (اجرای اسپرینت):

- تیم توسعه شروع به کار روی Story های تعیین شده در برنامه اسپرینت می کند.
- در این مرحله، کدها باید بر اساس Backlog و Epic ها و پیاده سازی شوند و تست های اولیه انجام شوند.

5. Sprint Review (جلسه بررسی اسپرینت):

- بررسی کار انجام شده در اسپرینت و ارائه گزارش به تیم و Stakeholders.
- دریافت بازخورد از تیم های دیگر برای بهبود ویژگی ها و عملکرد سیستم.

6. Sprint Retrospective (بازنگری اسپرینت):

- ارزیابی عملکرد تیم و شناسایی نقاط قوت و ضعف فرآیند.
- پیشنهاد راهکار هایی برای بهبود عملکرد تیم در اسپرینت های آینده.

مستندات و گزارشات اسکرام:

برنامه‌ای برای تست‌های سیستم که نحوه آزمایش عملکرد سفارش‌ها، تطبیق سفارش‌ها، و ثبت تراکنش‌ها را شرح می‌دهد.

تعريف تست‌های مختلف برای هر بخش از سیستم.

برای نشان دادن میزان پیشرفت کار در هر اسپرینت و اینکه تیم چقدر کار را در اسپرینت‌های مختلف انجام داده است.

امتیازی‌ها:

1. گزارش‌های اسپرینت:

برای هر اسپرینت باید گزارش‌هایی از جلسات Planning، Review و Retrospective آماده کنید.

استفاده از Refinement برای بهبود و اولویت‌بندی نیازمندی‌ها در طول اسپرینت‌های آینده.

2. مستندات در Jira:

برای مدیریت هر پروژه، باید از ابزار Jira استفاده کنید.

Backlog باید بین تیم‌ها به‌طور دقیق تقسیم و مدیریت شود.

3. پشتیبانی از مستندات توسعه:

GitHub یا GitLab برای مدیریت کد استفاده شود.

انجام Merge Requests برای کدهای موجود و استفاده از Task‌ها در هر Epic.

نکات تکمیلی:

برای پیگیری میزان پیشرفت اسپرینت‌ها استفاده شود.

تیم باید با Backlog، Epic و Jira های خود را به درستی مدیریت کرده و به دقت تقسیم‌بندی کند.

برای استفاده از GitHub یا GitLab، کد باید در Merge Requests قرار گیرد و تیم‌ها به درستی بین ابزارها هماهنگ شوند.

بخش پنجم: میکروسرویس‌ها، دیپلوی و مدیریت حوادث

با افزایش ترافیک پرور، معماری میکروسرویسی برای افزایش مقیاس‌پذیری و انعطاف‌پذیری توصیه می‌شود. تفکیک سرویس‌ها شامل سرویس‌های مختلف محصول، سفارش‌ها، تراکنش‌ها، گزارش‌ها و اطلاع‌رسانی به کاربران می‌شود.

تفکیک سرویس‌ها:

- سرویس‌های محصول: شبیه‌سازی تغییرات قیمت، ارائه لیست سهام به صورت آنی و دریافت اطلاعات مربوط به هر سهم.
- سرویس‌های سفارش: ثبت سفارشات خرید و فروش و تطبیق آن‌ها با سیستم‌های تطبیق سفارش (Matching Engine).
- سرویس‌های تراکنش: ثبت و ذخیره تراکنش‌ها در بلاکچین خصوصی.
- سرویس‌های سفارش: ثبت سفارشات خرید و فروش و تطبیق آن‌ها با سیستم‌های تطبیق سفارش (Matching Engine) به صورت غیر متمرکز بر روی بلاکچین و توضیح پیاده سازی‌های فعلی آن. (امتیازی)
- سرویس‌های اطلاع‌رسانی(Notification Service): ارسال اعلان‌های لحظه‌ای وضعیت سفارش‌ها به کاربران از طریق Message Broker.
- برای ارتباط غیرهمزمان بین میکروسرویس‌ها، کاهش کول‌پلیک و افزایش انعطاف در برابر افزایش حجم ترافیک استفاده می‌شود.

(امتیازی) نمودار دیپلوی:

استقرار بر روی Kubernetes برای مدیریت ساده‌تر کانتینرها، افزایش انعطاف در مقیاس‌پذیری افقی، و ارتقاء مداوم سرویس‌ها بدون downtime و مانیتورینگ با Grafana و Prometheus برای پایش مستمر معیارهای کارایی، تاخیر و خطا. همچنین برای راحتی تنها نیاز است تا نمودارهای مربوطه رسم کنید و نیازی به کدنویسی نمی‌باشد.

مدیریت حوادث(Incident Management):

در این بخش، مدیریت حوادث و فرآیند پاسخ‌دهی به مشکلات و خطاهای در سامانه بورس آنلاین توضیح داده می‌شود. همچنین روش مدیریت حوادث بر روی قرارداد هوشمند و بلاکچین نیز توضیح داده شود.

On-Call Engineer باید در صورت بروز مشکلات احتمالی، اقدامات لازم را انجام دهد.

1. تخصیص مسئولین:

- تخصیص مسئول On-Call Engineer برای پاسخگویی سریع به مشکلات سیستم و Alert‌ها بر اساس معیارهایی همچون حجم تراکنش‌ها یا تأخیر در پاسخ‌ها.

- تخصیص کلید owner بر روی بلاکچین و مشخص کردن سطح دسترسی آن. توضیح آنکه چه راهکار های دیگه ای برای غیرمت مرکز کردن این دسترسی owner وجود دارد.

:Incident Postmortem .2

- مستندسازی و تحلیل حوادث به طور دقیق بعد از وقوع، برای شناسایی دلایل مشکلات و ارزیابی اثرات آنها.
- این مستندات به تیم کمک می کند تا در آینده جلوی بروز مشکلات مشابه را بگیرد.

:Incident Postmortem سندهای مربوط به

1. اختلال در سرویس های سفارش: زمانی که سرویس های سفارش دچار مشکل می شوند یا تأخیر در پردازش رخ می دهد.
2. افت عملکرد سرویس پیش بینی گر: زمانی که سرویس پیش بینی گر نتواسته باشد به درستی پیش بینی ها را انجام دهد.
3. مشکل در ارتباط با سرویس ها: زمانی که ارتباط با یکی از میکروسرویس ها مانند Order Matching Engine دچار مشکل شود.

نکات مربوط به انجام پروژه:

1. طول پروژه:

در طول انجام پروژه، لازم است که خود را در نقش‌های مختلفی مانند توسعه‌دهنده، تحلیلگر سیستم، آزمایشگر و غیره قرار دهید. از دید هر نقش، تمامی جنبه‌های پروژه را بررسی کنید.

2. توانایی تطبیق پروژه با نیازهای عملیاتی:

به هنگام انجام پروژه، باید اطمینان حاصل کنید که تمامی جنبه‌ها و جزئیات پروژه به‌طور عملی و منطقی اجرا می‌شوند و پیاده‌سازی پروژه به‌درستی قابل انجام باشد.

3. اطمینان از قابل اجرا بودن پروژه:

تمامی بخش‌های پروژه باید به‌گونه‌ای طراحی شوند که بدون مشکل و به راحتی در سیستم‌های مختلف پیاده‌سازی شوند.

4. توجه به مسائل استاندارد:

لازم است که تمامی بخش‌های پروژه بر اساس استانداردهای علمی و صنعتی طراحی و پیاده‌سازی شوند و تمامی مستندات پروژه باید دقیقاً به این استانداردها پایبند باشند.

5. مستندات و گزارشات:

تمامی مستندات و گزارشات مربوط به هر بخش از پروژه باید بدقت و به صورت حرفه‌ای تهیه و نگهداری شوند.

6. پشتیبانی از تعامل با سیستم‌های دیگر:

پروژه باید به‌گونه‌ای طراحی شود که تعامل و ارتباط با سیستم‌های دیگر و سایر ابزارهای توسعه‌دهنده نیز میسر باشد.

7. حضور تمامی اعضای تیم در جلسات ارائه پروژه:

تمامی اعضای تیم باید در جلسات ارائه پروژه حاضر باشند تا در پاسخ‌دهی به سوالات و بحث‌های مربوط به پروژه، نقش فعالی ایفا کنند.