طرح بازطراحی سیستم(Redesign Plan)

مقدمه

با توجه به تغییر نیازمندی مطرح‌شده در سیستم آموزش آنلاین و درخواست افزودن قابلیت نظارت بر آزمون آنلاین (Online Exam Proctoring)، لازم است معماری فعلی سیستم به گونه‌ای توسعه داده شود که امکان بررسی رفتار دانشجو در زمان برگزاری آزمون فراهم شود.

در نسخه اولیه سیستم، فرآیند آزمون شامل ایجاد آزمون، مدیریت اطلاعات آزمون، دریافت پاسخ‌ها و محاسبه نمره بود. با وجود کامل بودن این فرآیند، سیستم امکان تشخیص و ثبت رخدادهای غیرعادی مانند خروج از صفحه آزمون، تغییر مکرر مرورگر، قطع ارتباط غیرمجاز یا سایر رفتارهای مشکوک را نداشت.

هدف از بازطراحی، اضافه کردن قابلیت نظارت آزمون بدون ایجاد تغییرات گسترده در هسته اصلی سیستم است. به همین دلیل، قابلیت جدید به صورت یک ماژول مستقل طراحی می‌شود تا ضمن حفظ ساختار فعلی، امکان توسعه و نگهداری آسان‌تر سیستم در آینده فراهم شود.

در طراحی جدید تلاش می‌شود اصول مهم مهندسی نرم‌افزار مانند کاهش وابستگی بین اجزا، تفکیک مسئولیت‌ها و قابلیت توسعه‌پذیری رعایت شود.

بررسی معماری فعلی و نیاز به تغییر

معماری فعلی سیستم بر اساس لایه‌هایDomain ، Repository، Serviceو Controller طراحی شده است. این ساختار باعث شده است منطق اصلی سیستم از بخش ذخیره‌سازی و کنترل درخواست‌ها جدا باشد.
با اضافه شدن قابلیت نظارت آزمون، ایجاد تغییر مستقیم در Exam Service یا Controller باعث افزایش پیچیدگی سیستم و کاهش قابلیت نگهداری خواهد شد.
بنابراین به جای تغییر در بخش‌های اصلی، یک سرویس جدید به نام:
ProctoringService
به لایه Service اضافه می‌شود.

این سرویس به صورت مستقل وظیفه مدیریت فرآیند نظارت آزمون را بر عهده خواهد داشت و ارتباط آن با سایر بخش‌ها از طریق رویدادهای سیستمی انجام می‌شود.

طراحی ماژولProctoring Service

ماژول Proctoring Service مسئول مدیریت رفتارهای مرتبط با امنیت آزمون است.

وظایف اصلی این سرویس عبارت‌اند از:

- دریافت اطلاعات مربوط به فعالیت دانشجو هنگام آزمون
- تشخیص رخدادهای غیرعادی
- ثبت تخلفات شناسایی‌شده
- تعیین سطح اهمیت تخلف
- ارسال اطلاعات تخلف به سایر بخش‌های سیستم

برای مثال، زمانی که دانشجو در هنگام آزمون صفحه مرورگر را ترک می‌کند، این رخداد توسط سیستم ثبت شده و به عنوان یک رویداد تخلف به Proctoring Service ارسال می‌شود.

این سرویس پس از بررسی اطلاعات، در صورت نیاز تخلف را ثبت کرده و اطلاع‌رسانی لازم را انجام می‌دهد.

اضافه شدن موجودیتExamViolation

برای ذخیره اطلاعات مربوط به تخلفات آزمون، یک موجودیت جدید با نام:
ExamViolation
به بخش Domain اضافه می‌شود.

این موجودیت اطلاعات مربوط به هر تخلف را نگهداری می‌کند و شامل مواردی مانند:

- شناسه تخلف
- شناسه آزمون
- شناسه دانشجو
- نوع تخلف
- زمان رخداد
- میزان اهمیت تخلف

خواهد بود.

با اضافه شدن این موجودیت، سیستم قادر خواهد بود تاریخچه‌ای از تخلفات هر آزمون ایجاد کرده و در آینده از این اطلاعات برای تحلیل عملکرد دانشجویان یا تولید گزارش‌های مدیریتی استفاده کند.

توسعهRepository Layer

برای مدیریت اطلاعات تخلفات، یک Repository جدید با نام:
ViolationRepository
ایجاد خواهد شد.

هدف از ایجاد اینRepository ، جلوگیری از ارتباط مستقیم سرویس نظارت با پایگاه داده است.
در این طراحی، ProctoringServiceفقط با Repository ارتباط برقرار می‌کند و Repository مسئول ذخیره و بازیابی اطلاعات خواهد بود.
این روش باعث می‌شود در صورت تغییر پایگاه داده یا روش ذخیره اطلاعات، بخش‌های دیگر سیستم بدون تغییر باقی بمانند.
همچنین این ساختار با الگوی Repository Pattern که در طراحی اولیه سیستم استفاده شده است، هماهنگ خواهد بود.

استفاده ازEvent Driven Architecture

یکی از مهم‌ترین تغییرات در بازطراحی، استفاده از معماری مبتنی بر رویداد است.
در طراحی اولیه، ارتباط مستقیم بین سرویس‌ها باعث افزایش وابستگی می‌شود. برای جلوگیری از این مشکل، فرآیند ثبت تخلفات به صورت Event محور پیاده‌سازی خواهد شد.

برای مثال، هنگام تشخیص یک تخلف، رویدادی با عنوان:
ExamViolationDetected
ایجاد می‌شود.
این رویداد شامل اطلاعات مربوط به آزمون، دانشجو و نوع تخلف خواهد بود.

پس از ایجاد رویداد:

- Notification Serviceمی‌تواند پیام هشدار برای استاد ارسال کند.
- بخش گزارش‌گیری می‌تواند اطلاعات تخلف را ذخیره کند.
- سیستم مدیریت

مزیت این روش آن است که اضافه کردن قابلیت‌های جدید در آینده بدون تغییر در سرویس‌های اصلی امکان‌پذیر خواهد بود.

تغییرات درNotification Service

با اضافه شدن قابلیت نظارت آزمون، Notification Serviceنیز توسعه پیدا می‌کند.
هدف از این تغییر، اطلاع‌رسانی سریع در هنگام رخداد تخلفات مهم است.

برای مثال:

- ثبت چندین خروج از صفحه آزمون
- قطع ارتباط طولانی
- تشخیص رفتار غیرعادی

می‌تواند باعث ارسال هشدار برای استاد یا مدیر سیستم شود.
این ارتباط از طریق سیستم رویداد انجام شده و Notification Service وابستگی مستقیمی به Proctoring Service نخواهد داشت.

تغییرات درController Layer

در بازطراحی پیشنهادی، تغییرات Controller حداقلی خواهد بود.
Controllerهمچنان وظیفه دریافت درخواست‌ها و ارسال آن‌ها به سرویس مناسب را دارد و منطق مربوط به تشخیص تخلف در آن قرار نمی‌گیرد.
در صورت دریافت اطلاعات مربوط به رفتار دانشجو، Controllerاین اطلاعات را به ProctoringService ارسال می‌کند.

این روش باعث حفظ اصل:
Single Responsibility Principle
می‌شود؛ زیرا هر بخش تنها مسئولیت مشخص خود را انجام می‌دهد.

سازگاری طراحی جدید با اصول معماری

بازطراحی پیشنهادی با اصول معماری نرم‌افزار سازگار است.
با استفاده از این روش:

- مسئولیت‌ها بین بخش‌های مختلف تقسیم می‌شوند.
- وابستگی مستقیم بین سرویس‌ها کاهش پیدا می‌کند.
- امکان توسعه قابلیت‌های جدید افزایش می‌یابد.
- تغییرات آینده با هزینه کمتر انجام می‌شود.

به عنوان مثال، در آینده می‌توان قابلیت‌هایی مانند تحلیل تصویر، تشخیص چهره یا بررسی رفتار هوشمند دانشجو را بدون تغییر در بخش اصلی آزمون اضافه کرد.

جمع‌بندی

در این بازطراحی، قابلیت نظارت آزمون آنلاین به عنوان یک ماژول مستقل به سیستم اضافه می‌شود. استفاده از ProctoringService، ViolationRepositoryو معماری Event Driven باعث می‌شود سیستم علاوه بر پاسخگویی به نیاز جدید، همچنان ساختار قابل توسعه و نگهداری خود را حفظ کند.
این تغییر با کمترین میزان دستکاری در معماری فعلی انجام شده و امکان توسعه سیستم آموزش آنلاین به یک پلتفرم کامل‌تر و امن‌تر را فراهم می‌کند.