

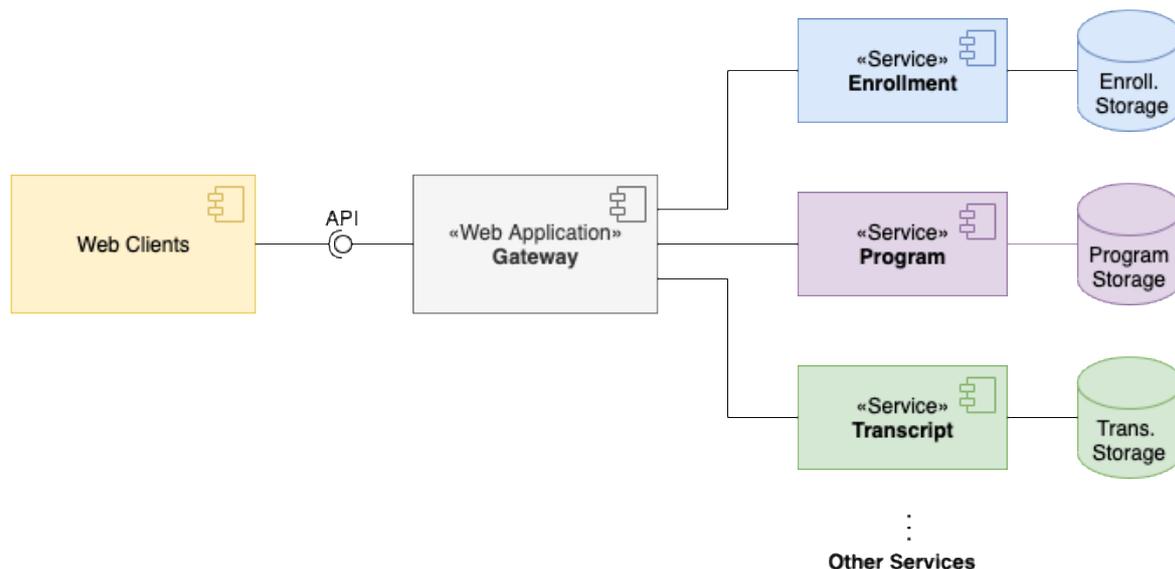


طراحی سرویس انتخاب واحد

در این تکلیف تصمیم‌هایی از یک معماری فرضی مبتنی بر سرویس برای کارکردهای انتخاب واحد از یک سامانه جامع آموزش را بررسی خواهیم کرد. برای انجام این تکلیف لازم است با الگوی event sourcing آشنا شوید که منابعی برای مطالعه در ادامه ذکر شده‌اند.

ساختار کلی معماری

در این معماری فرض شده است که مؤلفه‌های مختلف سامانه در قالب سرویس‌های مجزایی سازمان‌دهی شده‌اند. طبق معماری طراحی شده، کاربران سیستم با برنامه‌های کلاینت که روی مرورگر آنها اجرا می‌شود کار می‌کنند. این کلاینت‌ها درخواست‌های کاربران را برای یک مؤلفه درگاه (یک برنامه که روی وب‌سرور اجرا می‌شود) ارسال می‌کنند. این درگاه پس از اعمال کنترل‌های امنیتی و سایر عملیات عمومی (مثل لاگ کردن)، درخواست‌ها را به سرویس‌های مرتبط ارسال می‌نماید. سرویس‌ها برای پاسخ‌گویی به درخواست‌ها از منابع محاسباتی و داده‌های ذخیره‌شده خودشان استفاده می‌کنند و در صورت لزوم با بقیه سرویس‌ها ارتباط برقرار می‌کنند.



هر یک از سرویس‌ها بخشی از کارکردهای سیستم را برعهده دارد. به عنوان مثال سرویس انتخاب واحد (Enrollment) درس‌های ارائه‌شده در ترم و انتخاب واحد دانشجویان را مدیریت می‌کند، سرویس برنامه آموزشی (Program) برنامه‌های آموزشی (شامل رشته‌ها، چارت درس‌ها، پیش‌نیاز بین درس‌ها) را مدیریت می‌کند و سرویس سوابق تحصیلی (Transcript) سوابق تحصیلی دانشجویان شامل کارنامه آنها را مدیریت می‌کند. هر یک از این سرویس‌ها داده‌های مربوط به خود را در قالب یک پایگاه داده (با فناوری مناسب برای آن سرویس) ذخیره می‌کند. در این سامانه سرویس‌های دیگری نیز وجود دارند که در حد این تکلیف به آنها پرداخته نشده است.

کارکردهای انتخاب واحد

با توجه به تمرکز این تکلیف روی کارکرد انتخاب واحد، ابتدا کارکردهای اصلی سرویس انتخاب واحد در بازه زمانی انتخاب واحد را مرور می‌کنیم. این کارکردها عبارتند از:

- اخذ یک درس ارائه‌شده توسط دانشجو
- حذف یک درس انتخاب‌شده توسط دانشجو
- پیوستن یا ترک لیست انتظار یک درس ارائه‌شده توسط دانشجو
- تغییر ظرفیت درس‌های ارائه‌شده توسط کارشناس آموزش
- تغییر زمان کلاس یا امتحان درس‌های ارائه‌شده توسط کارشناس آموزش
- بستن یا بازکردن امکان انتخاب درس‌های ارائه‌شده توسط کارشناس آموزش
- ارائه اطلاعات انتخاب واحد (برنامه ترم، ثبت‌نام دانشجو، وضعیت درس‌ها، ...) برای نمایش در کلاینت

فرض می‌شود انتخاب واحد دانشجویان تابع مقررات زیر است:

- درسی که قبلاً گذرانده شده را نمی‌توان اخذ کرد
- درسی را بدون گذراندن پیش‌نیازهای آن نمی‌تواند اخذ کرد
- درس‌های ارائه شده که در زمان کلاس یا امتحان خود با یکدیگر تلافی دارند نمی‌توان هم‌زمان اخذ کرد
- با توجه به معدل کل دانشجو امکان اخذ تعداد واحد محدودیت سقف دارد (بدون محدودیت کف)

دقت نمایید درخواست اخذ درس‌های ارائه‌شده به صورت تک تک به سامانه داده می‌شود و در صورتی که هر یک از قواعد فوق نقض شوند درخواست رد می‌شود وگرنه درس مربوطه به فهرست درس‌های انتخابی دانشجو اضافه می‌شود. در این تکلیف، به درخواست‌های دانشجویان برای رفع مشکلات انتخاب واحد نمی‌پردازیم.

در زمان‌هایی خارج از بازه زمانی انتخاب واحد، این سرویس علاوه بر ارائه اطلاعات لازم به سایر سرویس‌ها و همچنین ارائه گزارش‌های لازم به کاربران، به درخواست‌های انتخاب واحد خارج از روال عادی (مثل دانشجویی که پس از اتمام بازه انتخاب واحد مجوز ادامه تحصیل را کسب کرده) پاسخ می‌دهد.

تصمیم‌های معماری سرویس انتخاب واحد

فرض کنید تا کنون تصمیم‌های زیر درباره سرویس انتخاب واحد اتخاذ شده است. دقت نمایید تصمیم‌های ذکرشده در اینجا صرفاً به منظور تمرین انتخاب شده‌اند و ممکن است در حالت کلی لزوماً تصمیم خوبی نباشند.

مدیریت داده‌های انتخاب واحد در حافظه

از آنجا که دانشگاه‌هایی که این سامانه در آنها استفاده می‌شود دارای تعداد زیادی دانشجو هستند، و به هدف سرعت بخشیدن به پردازش درخواست‌های انتخاب واحد، سرویس انتخاب واحد، تمام داده‌های مورد نیاز برای پردازش درخواست‌های انتخاب واحد را در حافظه اصلی نگهداری می‌کند. این داده‌ها علاوه بر برنامه درس‌های ارائه‌شده در ترم و انتخاب واحد دانشجویان، شامل اطلاعات لازم از برنامه‌های آموزشی و سوابق تحصیلی دانشجویان نیز می‌شود.

فرض می‌شود در شروع انتخاب واحد کلیه داده‌های مورد نیاز از سرویس‌های دیگر گرفته می‌شود و در قالب داده‌ساختارهایی مناسب برای کارکردهای انتخاب واحد در حافظه ذخیره می‌شوند. به این ترتیب، برای عملکرد عادی این سرویس در زمان انتخاب واحد، نیازی به تعامل با سرویس‌های دیگر ندارد. پس از اتمام بازه انتخاب واحد، نیازی به نگهداری اطلاعات انتخاب واحد در حافظه نیست و ارائه خدمات توسط این سرویس مانند سرویس‌های دیگر با دسترسی به پایگاه داده محلی خود قابل انجام است.

استفاده از الگوی Event Sourcing

سرویس انتخاب واحد برای ضبط تغییرات وضعیت این سرویس از این الگو استفاده می‌کند. به این ترتیب، در صورتی که به هر دلیلی این سرویس با خرابی مواجه شود برای بازیابی وضعیت آن از دنباله رخدادهای ضبط شده استفاده می‌کند. برای آشنایی با این الگو مطالعه [مقاله فاولر درباره الگوی مذکور](#) و دیدن [ویدیوی سخنرانی در مورد این الگو](#) (یا این [ویدیوی جامع‌تر در مورد معماری متنی بر رخداد](#)) توصیه می‌شود.

پرسش‌ها

با فرضیات مطرح شده در بالا، به پرسش‌های زیر در خصوص دغدغه‌های معماری سرویس انتخاب واحد پاسخ دهید.

پرسش ۱ - برای پیاده‌سازی کارکردهای مطرح شده سرویس انتخاب واحد، چه اطلاعاتی باید در حافظه اصلی نگهداری شوند؟ برای پاسخ به این پرسش داده‌ساختارهای مربوطه را طراحی و تایپ‌های مورد نیاز را در یکی از زبان‌های برنامه‌سازی جاوا یا سی‌پلاس‌پلاس پیاده‌سازی کنید. در اینجا نیازی به پیاده‌سازی عملیات روی این تایپ‌ها نیست و صرفاً روی ساختار داده‌ها تمرکز نمایید. استفاده از کتابخانه‌های استاندارد این زبان‌ها مجاز است.

پرسش ۲ - با توجه به این که کل داده‌های مورد نیاز انتخاب واحد در ابتدا از سرویس‌های جانبی گرفته می‌شود، در صورتی که در طول بازه انتخاب واحد تغییری در این داده‌ها رخ دهد (مثلاً تغییر یک نمره دانشجو یا حذف یک پیش‌نیاز یک درس) چگونه این تغییرات را به سرویس انتخاب واحد منتقل کنیم؟ در پاسخ این پرسش مشخصاً به سازوکارهای ارتباطی بین سرویس‌ها و همچنین کنترل روند اجرا در پردازش انواع رخدادهای مختلف (تغییر داده‌های پایه یا درخواست‌های انتخاب واحد) اشاره نمایید.

پرسش ۳ - در صورت بروز خرابی در سرویس انتخاب واحد، سازوکار ترمیم این سرویس (بازیابی وضعیت قبل از خرابی) را تشریح کنید. با توجه به استفاده از event sourcing چه رخدادهایی را برای بازیابی وضعیت سرویس تعریف می‌کنید؟ چگونه تضمین می‌کنیم که دقیقاً به وضعیت قبل از خرابی برمی‌گردیم؟

پرسش ۴ - فرض کنید بخواهیم برای پاسخ به زیاد شدن حجم درخواست‌ها، نمونه‌های متعددی از سرویس انتخاب واحد اجرا کنیم و بار درخواست‌ها را روی آنها پخش کنیم. این دو راه‌حل را با یکدیگر مقایسه کنید: (۱) وضعیت تمام نمونه‌ها یکسان است و هر یک از آنها تمام داده‌های انتخاب واحد را در خود ذخیره کرده است و (۲) درخواست‌ها بر اساس ویژگی خاصی (مثلاً شناسه یک نوع موجودیت) روی نمونه‌های این سرویس افراز می‌شوند. اگر بخواهیم مورد (۲) را انتخاب کنیم پیشنهاد می‌کنید وضعیت را بر چه اساسی روی این نمونه‌ها افراز کنیم؟ در هر یک از دو راه‌حل به هزینه‌های سازگار نگه‌داشتن داده‌های سرویس‌های مختلف اشاره کنید. آیا به نظر شما می‌توان حدی از ناسازگاری را تحمل کرد؟

نحوه تحویل

پاسخ‌های خود را در قالب یک فایل پی‌دی‌اف با نام TeamNN-A3.pdf در محل مشخص شده بارگذاری کنید. در ابتدای متن، نام اعضای گروه را ذکر نمایید. در پاسخ پرسش ۱ متن کدهای جاوا یا سی‌پلاس‌پلاس را با اندازه فونت و فاصله بین خطوط مناسب در همان فایل ذکر شود. در تهیه پاسخ پرسش‌ها هر جا که مفید است از نمودارهایی که فهم مطلب را ساده‌تر نماید استفاده کنید.

¹ instance