



مقدمه

در این سری تمرین مراحل تبدیل راه حل یک مشکل به یک نرم افزار را بررسی خواهیم کرد، نهایتاً امتیازی که شما از تمرین کسب میکنید وابسته به پیاده سازی قسمتی از راه حل میباشد.

در یک سرویس تاکسی تلفنی بزرگ و قدیمی که بیش از ۱۰۰۰ خودرو در سطح کشور دارد اخیراً شکایات مسافران از نحوه رانندگی راننده ها و همچنین مسیرهای ترسناک! که برای رسیدن به مقصد انتخاب میکنند زیاد شده و مدیران مجموعه ابتدا با اضافه کردن بند هایی به قرارداد راننده ها این حق را گرفته اند که به ازای خطاهای راننده ها درصدی از سود سفر را از راننده کسر کنند و به حساب مشتری برگردانند با این فرایند سرویس تاکسی موفق شده رضایت دوباره مشتری ها را کسب کند اما تعدادی از راننده ها به اعتبار سنجی شکایت مشتری معترض اند. مدیران مجموعه که سالهاست با روش کسب و کار سنتی موفق شده اند تجارت خودشان رو اداره کنند حالا کلافه و گنگ از تیم IT که مسئول پشتیبانی و توسعه نرم افزار های اداری و عملیاتی و حساب داری مجموعه اند، درخواست چاره اندیشی میکنند. تیم قبلاً نرم افزاری تحویل حسابداری مجموعه داده که با لحاظ تعداد شکایات مشتریان حقوق راننده را محاسبه میکند پس تا حدی با فضای مسئله آشناست و از طرفی تیم پیش از این هم چندین بار با مسئله تبدیل منطق کسب و کار پیچیده – Complex business logic – به نرم افزار روبرو بوده پس اعضا دست به کار میشوند

پیش از شروع مفید است مطالعه

● [Domain-Driven Design](#)

● [DDD Crash Course](#)

● [Problem/Solution Space](#)

● [Product Manager](#) سمت شغلی مدیر محصول

● [Even-bus Pattern](#)

● [Event-Driven Architecture](#)

● [Hexagonal architecture](#)

● [CQRS Pattern](#)

شرح اپلیکیشن

نهایتاً تیم به این نتیجه رسیده اند که برای حل مشکل اعتبار سنجی راه حلی به جز نصب ردیاب روی خودرو ها و مانیتورینگ آنها ندارد. به واسطه اطلاعات ردیاب تیم میتواند خطاهای رانندگی را بررسی و ثبت کند و گزارش های لازم را با حساب داری مجموعه به اشتراک بگذارد، و همچنین در هر لحظه از مشخصات جغرافیایی راننده باخبر باشد. پیش از وارد شدن به فضای حل مسئله تیم زیرساخت مجموعه به شما این گارانتی را میدهد که پکت اطلاعات لحظه ای خودرو ها را با نام VehicleLog شامل :

VehicleUniqueCode

VehicleTime

VehicleLatitude

VehicleLongitude

VehicleSpeed

Temperature

VehicleStatus

را در قالب json و بر بستر redis و با استفاده از pub/sub برای شما ارسال کند. (اطلاعات اتصال به سرور برای شما ایمیل خواهد شد، همچنین برای دستیابی به فرمت دقیق پیام دریافتی و مثال های کار با pub/sub redis به [این لینک](#) مراجعه کنید).

Fraud Detection:

1. host:80/go/fraud/rules

کاربر با ورود به این صفحه با فرمی مواجه میشود که میتواند تنظیمات مربوط به خطاهای رانندگی را در آن مشاهده و ویرایش کند. فرم باید دو فیلد رنج دمای مجاز کابین و حداکثر سرعت مجاز را از کاربر دریافت کند و برای سرور ارسال کند، همچنین در صورتی که اطلاعات از پیش ثبت شده باشد امکان ویرایش آنها باید برای کاربر مهیا شده باشد. کد سرور شما باید برای هر لاگ دریافتی بررسی کند که در صورتی که سرعت خودرو بیشتر از سرعت مجاز بود یا دمای کابین خارج از محدود ثبت شده توسط کاربر بود برای خودرو خطا ثبت کند.

2. host:80/go/fraud/report

کاربر در این صفحه باید بتواند گزارش عملکرد راننده را دریافت کند. فرمی وجود دارد که کد خودرو و بازه زمانی دریافت میکند و اطلاعاتی شامل مسافت طی شده در بازه مورد نظر، مجموع زمان روشن بودن خودرو و تعداد خطاهای هر دسته بندی - دما و سرعت - به کاربر نمایش داده میشود. برای شهود بهتر میتوانید به فایل دیزان پیشنهادی برای فرانت مراجعه کنید.

Monitoring :

نرم افزار شما باید امکان نمایش لحظه ای مختصات و مشخصات خودرو را داشته باشد.

1. host:80/go/monitoring

کاربر با ورود به این صفحه باید با صفحه ی نقشه ای روبرو شود که مشخصات لحظه ای خودرو ها روی آن مشخص شده، همچنین با کلیک روی هر خودرو باید کادری حاوی مشخصات ماشین و سرعت لحظه ای آن و دمای داخل کابین به کاربر نمایش داده شود. این اطلاعات باید به صورت live و به محض دریافت VehicleLog از ردیاب بروز رسانی شوند. همچنین در صورتی که تخلف برای خودرو ثبت شود کاربر باید به صورت live مطلع شود. اتصال کلاینت شما به سرور باید توسط دو WS یا GRPC STREAM مجزا پیاده سازی شود.

کد سرور شما علاوه بر web-client استفاده کنندگان دیگری هم دارد، سیستم حساب داری مجموعه برای اتوماتیک محاسبه کردن حقوق افراد نیاز به ارتباط با سیستمی که تیم شما نوشته دارد و تیم تضمین داده که end-point های موجود در این [API Document](#) را پشتیبانی و پیاده سازی کند.

پیوست های پیشنهادی

- در [این لینک](#) دیزاین پیشنهادی در بستر figma برای شما تهیه شده که میتوانید front سیستم را طبق آن پیاده سازی کنید، این بستر به شما اجازه میدهد جزئیات دقیق پیاده سازی را در دسترس داشته باشید، همچنین میتوانید برای کامپوننت های جداگانه auto generated CSS دریافت کنید
- در [این لینک](#) معماری پیشنهادی برای back-end با رویکرد آموزشی مدل و ترسیم شده همچنین لینک حاوی interface های پیشنهادی برای پیاده سازی میباشد و در [این ویدیو](#) به توضیح بیشتر مدل پرداخته شده، همانطور که اشاره شد مدل سازی و ویدیوی ارایه آن با رویکرد آموزشی تدارک دیده شده اند که مطالعه و فهم آنها میتواند بسیار مفید باشد.
- فایل protobuf پیشنهادی را میتوانید از [این لینک](#) دریافت کنید

نحوه ی تحویل تمرین

سلامت باشید