

اینجا قصد داریم مسئله یه شرکت پستی رو حل کنیم. این شرکت کارش اینه که بعضی سفارشها رو باید از خونه مردم تحویل بگیره و بیاره توی شرکت اش، بعضی سفارشها رو هم باید از شرکت برداره و تحویل خونه مردم بده. تعدادی هم پیک موتوری داره که این کار رو براش انجام دهند. برای اینکه سرعت کار هم بره بالا، پیک هاش رو مامور کرده که توی سطح شهر بچرخند. منتها الان به این مشکل برخورد کرده که نمیدونه هر ماموریت رو به کدوم پیک بسپره، و حساب ماموریت ها از دستش در میره.

برای این کار نیاز به یه سیستم جمع و جور داره که توی این کار به داد خودش و پیک هاش برسه.

سیستمی که میتونه کمکش کنه میتونه این شکلی باشه که:

1. یه کاربر مدیر بشه براش تعریف کرد که قراره پشت سیستم بشینه و با سیستم کار کنه. کارهایی هم که قراره بکنه اینه که بتونه:

a. راننده تعریف کنه

b. ماموریت تعریف کنه

c. لیست همه ماموریت ها و راننده ها رو ببینه.

d. ماموریت رو به راننده تخصیص بده.

2. بشه تعدادی کاربر راننده تعریف کرد (که قراره به هر کدومشون یه ماموریت تخصیص داده بشه)

a. هر راننده میتونه در حال انجام ماموریت باشه و یا آزاد باشه.

b. هر راننده در هر لحظه فقط میتونه یه ماموریت در حال انجام داشته باشه.

c. راننده ها باید بتونن ماموریت هایی که بهشون تخصیص پیدا کرده (گذشته و حال) رو ببینن.

3. امکان تعریف ماموریت داشته باشه، هر ماموریت شامل جابجا کردن یه سری سفارش از شرکت تا خونه یه مشتری هست یا برعکس. ماموریتها رو به دو صورت میشه تعریف کرد:

a. وقتی ماموریت تعریف شد، خودکار تخصیص پیدا کنه به یک راننده (انتخاب معیار با شما).

b. بشه صرفا ماموریت رو تعریف کرد و به صورت دستی به یکی از راننده های آزاد تخصیص داد.

\* برای اینکه درگیر مسئله پیک هایی که تو سطح شهر دارن می چرخد نباشین، فرض کنید که به ازای هر راننده، مختصات جغرافیایی فعلی راننده (longitude, latitude) در بستر mqtt روی یک تاپیک با عنوان زیر در حال انتشاره:

/test/driver/{driver\_id}

برای دریافت لوکیشن لحظه ای هر راننده، کافیه به تاپیک مورد نظر وصل بشید و دیتا رو دریافت کنید.

انتظاری که ما از شما داریم، اینه که سیستمی با مشخصات فوق پیاده سازی کنید:

Entity - ها و روابط بین شون رو تشخیص بدید و دیتابیس رو به نحو مناسبی طراحی کنید.

- فرضیاتی که در صورت مسئله مطرح شده رو لحاظ کنید.

- و نهایتا API های مورد نیاز برای کار با چنین سیستمی رو توسعه بدید.

+ پروژه رو داکرایز کنید که نهایتا با اجرای یک فایل compose-docker قابل اجرا باشد.