



دانشکده مهندسی کامپیوتر

توصیفی از دامنه پروژه

طراحی پایگاه های داده

استاد درس: جناب دکتر رحمانی

نیم سال دوم

سال تحصیلی ۱۴۰۰-۱۴۰۱

## فاز ۱

### توصیفی از دامنه پروژه

پارسا عیسی زاده - امیرعلی پاکدامن دنیوی - نیکی نزاکتی - ۱۴۰۰/۱۲/۵

#### ۱.۱ شرح پروژه

این پروژه یک سامانه ی هوشمند مبارزه با ویروس کرونا است. افراد برای استفاده از این سامانه با نصب این نرم افزار و ثبت اطلاعات خود، به عنوان کاربر میتوانند از ویژگی های این سامانه استفاده کنند. کاربر میتواند محلهای رفت و آمد و کاربرهایی را که با آنها رفت و آمد داشتند را ثبت کند. این قابلیت، این امکان را به کاربر میدهد که اگر کاربر دیگری که به محلهای مشابه کاربر رفته باشد به کرونا مبتلا شود، از طریق این نرم افزار به این کاربر اطلاع داده شود. در این سامانه وضعیت کرونایی مکان های مختلف روی نقشه و کاربرها به چهار دسته تقسیم میشوند.

- سفید
- زرد
- قرمز
- سیاه

هر یک از دسته ها برای کاربر نشان دهنده احتمال مبتلا شدن آن ها و برای مکان ها احتمال آلوده بودن

آن است. هر کاربر در حالت عادی در وضعیت سفید قرار دارد و اگر به مکانهای با خطر متوسط به بالا رفته باشد، به مدت دو هفته در وضعیت زرد قرار میگیرد و اگر به جایی برود که در کمتر از دو هفته قبل از آن کاربری با وضعیت قرمز (مبتلا به کرونا) به آنجا رفته باشد، وضعیت او قرمز میشود و باید هر چه زودتر تست کرونا بدهد. اگر نتیجه تست مثبت شد، وضعیت او قرمز میشود و باید به همه کاربرهایی که با او در تماس بوده اند پیام اخطار صادر شود و همه آن ها در وضعیت قرمز قرار بگیرند. همچنین تمامی مکانهایی که کاربر در دو هفته اخیر به آنجا رفته است، در وضعیت قرمز قرار میگیرند و مکان هایی که پیش از دو هفته به آنجا رفته بود در وضعیت زرد قرار میگیرند. اگر نتیجه تست منفی شد فرد در وضعیت سفید قرار میگیرد. با استفاده از ویژگی های این سامانه کاربر میتواند از مکان ها و افراد احتمال به آلودگی دوری کند و ریسک مبتل شدن خود را کاهش دهد.

## ۲.۱ موجودیت ها

۱. کاربر ها
۲. مکان ها
۳. موجودیت سابقه ای که هر کاربر در چه مکان هایی بوده
۴. اخطار ها
۵. محدودیت ها

### ۱.۲.۱ موجودیت کاربر

در جدول این موجودیت اطلاعات کاربر ها ذخیره می شود و وضعیت هر کاربر باید معلوم باشد. چون وضعیت کرونایی بعد از دو هفته به رنگ سفید میرود در نتیجه نیاز است تا زمانی که برای بهبود وضعیت کرونایی پیش بینی می شود را ذخیره کنیم. در صورتی که کاربر به مکان های الوده رفته باشد نیاز است تا تست بدهد در نتیجه لازم بودن تست هم باید در جدول ذخیره شود. همچنین به نتیجه آخرین تست هم نیاز است تا برحسب مثبت یا منفی بودنش رنگ وضعیت کاربر را تعیین کنیم.

- نام و نام خانوادگی
- نتیجه آخرین تست
- زمان به روز رسانی وضعیت

- نیاز به تست
- وضعیت کرونایی

### ۲.۲.۱ موجودیت مکان

در این موجودیت باید معلوم باشد که وضعیت کرونایی هر مکان به چه صورت است . در نتیجه هر سطر باید شامل عنوان های زیر باشد :

- اسم مکان
- latitude
- longitude
- وضعیت کرونایی
- باز یا بسته بودن

### ۳.۲.۱ موجودیت سابقه رفت و آمد

در جدول این موجودیت رکورد هایی ثبت می شود که نشان می دهد در هر مکان چه کسانی بوده اند و کدام یک از کاربران یکدیگر را دیده اند . در این جدول تنها لازم است طرفین ملاقات یا اسم مکان و کسی که در آنجا بوده را بنویسیم به همراه زمانش که اگر زمان مورد نظر سپری شد وضعیت کرونایی مکان و کاربر ها تغییر کند . به کمک این جدول مشخص می شود که چه مکان هایی مشکوک به کرونا هستند و اینکه چه مکان هایی واقعا آلوده هستند .

ستون ها در این جدول :

- نام کاربر که در ملاقاتی بوده یا به جایی رفته
- نام مکان یا نام کاربر
- زمان

### ۴.۲.۱ موجودیت اخطار

کسانی که برایشان باید پیام اخطار ارسال شود از طریق جدول سوابق رفت و آمد محاسبه می شوند . اما بحث اینجاست که اگر هنگام ساخته شدن پیام اخطار کاربر درون برنامه نباشد امکان دارد پیام اخطار برایش ارسال شود . / برای حل این مشکل دو راه داریم .

۱. به هنگام ورود کاربر به برنامه به جدول های موجود کوئری بزند و ببیند که آیا پیام اخطار لازم است داشته باشد یا نه

۲. پیام های اخطار یکبار محاسبه شوند و سپس در جدولی ذخیره شوند تا به هنگام ورود کاربر به اپلیکیشن برایش ارسال شود .

برای کم بودن بار سرور گزینه دوم راه بهینه تری است . در نتیجه به موجودیت پیام های اخطار نیاز داریم . در جدول این موجودیت نیاز است تا زمان ساخته شدن پیام هم ذخیره شود تا اگر مثلاً مدت زیادی از زمان خطر گذشته بود کاربر آگاه باشد . اگر پیامی خوانده شد دوباره نباید برای کاربر ارسال شود . در نتیجه به attribute دیگری نیاز داریم تا در صورتی که پیام خوانده شده بود مقدارش تغییر کند . / در نتیجه ستون های جدول به شکل زیر خواهد بود :

- نام کاربری که باید پیام را دریافت کند
- زمان ساخته شدن پیام
- وضعیت پیام ( خوانده شده و خوانده نشده )

### ۵.۲.۱ موجودیت محدودیت

با توجه به محدودیت هایی که بر اثر موج ها ایجاد می شود ، امکان دارد بعضی مکان ها بسته باشند و امکان رفت و آمد نداشته باشند . ( موجودیت مکان attribute ای به اسم باز یا بسته بودن داشت که از طریق همین جدول محاسبه می شود . )

در هر محدودیت اسمی مکان هایی که بسته می شوند به جدول این موجودیت اضافه می شود و ثبت می شود که تا کی بسته خواهند بود .

در نتیجه ویژگی های این موجودیت به شکل زیر خواهد بود :

- نام مکان
- زمان اتمام محدودیت

### ۳.۱ روابط بین موجودیت ها

۱. کاربر، حضور دارد در، صفر یا تعداد بیشتری سوابق (حضور در مکان یا ملاقات با فرد)

۲. کاربر، اخطارهایی دارد، به تعداد صفر یا بیشتر

۳. سوابق، شامل میشود، یک کاربر (کاربر دوم) یا یک مکان را

۴. محدودیت، شامل میشود، یک مکان را

## ۴.۱ کوئری های پرکاربرد

۱. ایجاد کاربر و ثبت اطلاعات کاربری هنگام نصب نرم افزار

۲. اضافه کردن مکان ها به دیتابیس

۳. گرفتن مکان ها برای نمایش در نقشه و فیلتر بر اساس وضعیت

۴. ثبت حضور یک کاربر در یک مکان

۵. ثبت ملاقات یک کاربر با فرد دیگر

۶. بررسی ملاقات ها و مکان های دو هفته اخیر کاربر و بروزرسانی وضعیت کاربر

۷. بروزرسانی وضعیت افراد و مکان های دو هفته اخیر در صورت تغییر وضعیت یک کاربر

۸. بروزرسانی وضعیت کاربر در صورت مشخص شدن نتیجه تست

۹. بررسی کسانی که وضعیتشان قرمز شده است و یافتن کاربرهایی که با آنها در تماس بوده اند

۱۰. ارسال پیام به افراد در تماس با افراد با وضعیت قرمز

## ۵.۱ چگونگی و مراحل طراحی و پیاده سازی سیستم

### ۱.۵.۱ طراحی منطقی پایگاه داده ها

در این مرحله دانشجو باید تمامی موجودیتهای ممکن و رابطه های مهم بین آنان را در محیط عملیاتی و بر اساس نیازهای اطلاعاتی شناسایی نماید. در انجام اینکار میتوان از نمودارهای ER و یا UML استفاده کرد. برای

مثال

- رابطه بین جدول سوابق و جدول کاربر ها
- رابطه بین جدول سوابق و جدول مکان ها
- رابطه بین جدول پیام ها و جدول کاربر ها

### ۲.۵.۱ سنتز پایگاه داده ها

در این مرحله با استفاده از تکنیکهای فراگرفته در درس پایگاه داده ها دانشجو باید اقدام به نرمال سازی رابطه های منتجه نماید.

### ۳.۵.۱ ایجاد پراسیجرهای ذخیره شده

قرار است تمامی ( و یا اکثریت) عملیات ذخیره و بازیابی اطلاعات از طریق پراسیجرهای ذخیره شده در پایگاه داده ها انجام گیرد. به برخی از این پراسیجرها در بخش کوئری های پرکاربرد اشاره شد .