



## دانشگاه صنعتی امیر کبیر ( پلی تکنیک تهران )

دانشکده مهندسی کامپیوتر

مهندسی نرم افزار ۱

“فاز ۳ پروژه”

موضوع پروژه : Facilities & Transportation Service :

استاد درس : دکتر کلباسی

نام گروه: Axiom

## فهرست مطالب

3	..... Facilities & Transportation
3	..... تحلیل زمینه و ضرورت وجود سرویس (Context & Rationale)
4	..... تبیین دامنه مسئله و حدود عملکرد سرویس (Domain Characterization )
6	..... تدوین سناریوی عملیاتی (Operational Scenario)
6	..... عنوان سناریو
6	..... شرایط استفاده (Usage Context)
6	..... هدف استفاده‌کننده (User Goal)
6	..... بازیگران و سرویس‌های درگیر
7	..... جریان اصلی تعامل با سرویس (Main Flow)
8	..... پیامد و ارزش افزوده (Outcome & Added Value)
9	..... مطالعه موردی و الگوهای برای سرویس ما
9	..... مقدمه
9	..... بررسی سیستم‌های واقعی و ارتباط با نیازمندی‌های ما
10	..... چالش‌ها و محدودیت‌هایی که باید در طراحی مد نظر بگیریم
11	..... بیانش‌ها و توصیه‌های طراحی برای سرویس
11	..... نتیجه‌گیری نهایی
12	..... داستان‌های کاربر (User Stories)
12	..... نمای کلی داستان‌ها
13	..... جزئیات داستان‌های کاربر
22	..... استخراج نیازمندی‌ها
22	..... نیازمندی‌های عملکردی (Functional Requirements)
29	..... نیازمندی‌های غیر عملکردی (Non-Functional Requirements)
30	..... اعتبار سنجی نیازمندی‌ها
44	..... گزارش کوتاهی از جلسات برگزار شده با تدریسیار
44	..... گزارش جلسه اول اسکرام
45	..... گزارش جلسه دوم اسکرام

# سناپریوی پروژه برای سرویس Facilities & Transportation

## تحلیل زمینه و ضرورت وجود سرویس (Context & Rationale)

- جایگاه در سامانه کلان: سرویس Facilities & Transportation لایه‌ی زیرساختی و عملیاتی سامانه‌ی «وبسایت تعاملی معرفی کشور» را تشکیل می‌دهد. در این سامانه، سرویس‌هایی مانند Wiki محتوای متنی و فرهنگی را فراهم می‌کنند و Map بستر فضایی و نمایش نقشه را در اختیار می‌گذارند؛ اما این سرویس است که با ارائه‌ی اطلاعات لجستیکی (امکانات پیرامونی و مسیرهای دسترسی)، «ایده‌ی سفر» را به یک برنامه‌ی قابل اجرا تبدیل می‌کند. بدون این سرویس، سامانه بیشتر شبیه یک اطلس دیجیتال باقی می‌ماند تا یک ابزار واقعی برای سفر.
- مسئله‌ای که حل می‌کند: مسئله‌ی اصلی، گسست میان اطلاعات مکانی/فرهنگی و نیازهای واقعی سفر است. کاربر فقط نمی‌خواهد بداند «یک جاذبه کجاست»، بلکه سوالات او معمولاً این‌هاست:
  - در نزدیکی این جاذبه چه هتل‌ها و رستوران‌هایی وجود دارد؟
  - از موقعیت فعلی ام چقدر زمان می‌برد به آنجا برسم و بهترین مسیر کدام است؟
  - اگر مشکلی پیش بیاید، نزدیک‌ترین مرکز درمانی کجاست و چگونه به آن برسم؟در وضعیت موجود، کاربر برای پاسخ به این سوالات مجبور است بین چند اپلیکیشن نقشه، سایت رزرو هتل، اپهای نقد و بررسی و... جابه‌جا شود. سرویس Facilities & Transportation با یکبار چه‌سازی داده‌های POI(Ponit Of Interest)، مسیر و ترافیک، این گسست را برطرف می‌کند و پاسخ این سوالات را در یک جریان منسجم در اختیار کاربر قرار می‌دهد.

- ارزش افزوده و ضرورت وجود سرویس: این سرویس با تمرکز بر امکانات و حمل و نقل، سه نوع ارزش افزوده‌ی کلیدی برای سامانه ایجاد می‌کند:

- a. افزایش کارایی برنامه‌ریزی سفر: کاربر می‌تواند در همان محیطی که محتوای جاذبه را می‌خواند، هتل مناسب پیدا کند، رستوران‌های نزدیک را ببیند و زمان و مسیر جابه‌جایی بین نقاط را برآورد کند؛ بدون این‌که به سرویس‌های متعدد بیرونی وابسته شود.
- ii. افزایش ایمنی و کاهش اضطراب لجستیکی: نمایش نزدیک‌ترین مراکز درمانی و سایر خدمات حیاتی در کنار مسیریابی (به‌ویژه در سناریوهای اضطراری) به کاربر کمک می‌کند با اطمینان و آتش بیشتری سفر کند و در شرایط بحرانی، سریع‌تر تصمیم بگیرد.
- iii. بهبود تجربه کلی سامانه و نقش آن در معماری کلان: با توجه به وابستگی شدید به داده‌های بلاذرنگ (ETA، ترافیک) و API‌های خارجی نقشه، این سرویس به هسته‌ی عملیاتی معماری میکروسرویسی تبدیل می‌شود؛ سرویسی که خروجی‌های مستقیماً در Plan Trip و Event Map استفاده شده و کیفیت آن، تجربه‌ی نهایی کاربر از سامانه را تعیین می‌کند.

## تبیین دامنه مسئله و حدود عملکرد سرویس (Domain Characterization)

ویژگی	شرح
کاربران درگیر	<ul style="list-style-type: none"> <li>- گردشگران داخلی و خارجی که برای برنامه‌ریزی و اجرای سفر از سامانه استفاده می‌کنند (پیدا کردن هتل، رستوران، مسیر بین جاذبه‌ها و ...).</li> <li>- رانندگان در حال مسیریابی (کاربرانی که در حین حرکت می‌خواهند ترافیک لحظه‌ای، مسیرهای جایگزین و ETA را ببینند).</li> <li>- کاربران در شرایط اضطراری که نیاز فوری به یافتن نزدیکترین مرکز درمانی یا خدمات حیاتی دارند (بیمارستان، درمانگاه، داروخانه شبانه‌روزی).</li> </ul>
نوع داده‌ها	<ul style="list-style-type: none"> <li>- سرویس Trip به عنوان کاربر سیستمی که برای ساخت برنامه‌ی سفر، به فهرست امکانات اطراف هر جاذبه و اطلاعات مسیرها نیاز دارد.</li> <li>- سرویس Map به عنوان مصرف‌کننده‌ی داده‌های مکانی و مسیر برای نمایش روی نقشه.</li> <li>- مالک محصول / مدیر سامانه که کیفیت داده‌ها، پایداری سرویس و پاسخ‌گویی به نیازهای کسب‌وکاری را پایش می‌کند.</li> </ul>
حدودیت‌های اصلی	<ul style="list-style-type: none"> <li>- داده‌های نقاط مورد علاقه (POI): شامل انواع امکانات مانند هتل، رستوران، کافه، موزه، مرکز درمانی، ایستگاه‌های حمل و نقل عمومی و سایر خدمات مرتبط با سفر. برای هر POI حداقل این فیلدها نگهداری می‌شود: نوع، نام، مختصات جغرافیایی، آدرس متنی، و در صورت وجود شماره تماس، ساعات کاری، رتبه کاربران و لینک رزرو.</li> <li>- داده‌های خطی و مسیریابی: شامل مسیرهای جاده‌ای (اتوبان، خیابان)، مسیرهای حمل و نقل عمومی (ایستگاه‌ها و خطوط) و نتایج مسیریابی (فاصله، زمان تخمینی رسیدن، مسیرهای جایگزین).</li> <li>- داده‌های ترافیک و وضعیت لحظه‌ای: مانند سرعت متوسط در بخش‌های مختلف مسیر، حوادث و مسدودی‌ها که از API‌های نقشه دریافت می‌شود.</li> <li>- داده‌های زمینه‌ای کاربر: موقعیت فعلی کاربر، فیلترهای انتخاب شده (نوع امکانات، محدوده قیمت، حداقل رتبه، نوع حمل و نقل)، که صرفاً برای محاسبه نتایج استفاده و نگهداری طولانی مدت نمی‌شود.</li> <li>- داده‌های ورودی از سرویس‌های دیگر: امتیازها و نظرات کاربران و رسانه‌ها</li> </ul>
محدودیت‌های اصلی	<ul style="list-style-type: none"> <li>- وابستگی به API‌های نقشه (مثل Neshan): دقت موقعیت، پوشش جغرافیایی، اطلاعات ترافیک و نتایج مسیریابی به کیفیت سرویس خارجی وابسته است؛ محدودیت نرخ درخواست، قطعی سرویس یا تغییر نسخه API می‌تواند روی عملکرد اثر بگذارد.</li> <li>- نیاز به پردازش سریع محاسبات مسیر (ETA): برای حفظ تجربه کاربری، مسیریابی و بهروزرسانی ترافیک باید در زمان قابل قبول انجام شود، در حالی که منابع پروژه‌ی درسی محدود است و نمی‌توان از زیرساخت‌های بسیار سنگین توزیع شده استفاده کرد.</li> </ul>

<ul style="list-style-type: none"> <li>- محدودیت دقت داده‌های مکانی و ترافیکی: خطای GPS. پوشش ناقص داده‌های POI در برخی شهرها و نابرابری کیفیت داده‌ها باعث می‌شود نتایج همیشه ۱۰۰٪ کامل و دقیق نباشند.</li> <li>- قیود حريم خصوصی و امنیت: سرویس باید حداقل داده‌ی لازم از موقعیت کاربر را دریافت کند، ارتباط با API‌های خارجی از طریق کانال امن انجام شود و کلیدهای دسترسی به صورت امن نگهداری شوند؛</li> <li>- محدودیت کیفیت شبکه‌ی کاربر: اینترنت ضعیف یا ناپایدار می‌تواند باعث تأخیر یا قطع نتایج مسیریابی شود؛ سرویس باید در این شرایط حداقل عملکرد (مثلاً نمایش مسیر بدون ترافیک یا آخرین اطلاعات کششده) را داشته باشد.</li> </ul>	
<ul style="list-style-type: none"> <li>- بازیابی POI در شعاع مشخص: دریافت موقعیت، نوع امکانات و شعاع جستجو از Plan Trip یا Frontend و برگرداندن فهرست امکانات نزدیک برای نمایش روی نقشه.</li> <li>- فیلترینگ پیشرفته امکانات: اعمال فیلترهای جزئی مانند حداقل رتبه، محدوده‌ی قیمت، نوع غذا/خدمت بر روی لیست POI‌های بازیابی شده.</li> <li>- مسیریابی و تخمین ETA: دریافت مبدأ و مقصد، فراخوانی API‌های نقشه، پردازش پاسخ و بازگرداندن مسیر بهینه، فاصله و زمان تخمینی رسیدن.</li> <li>- پیشنهاد مسیرهای جایگزین (خودرو / حمل و نقل عمومی): ترکیب داده‌های مسیر خودرو و مسیرهای حمل و نقل عمومی و ساخت جدول مقایسه‌ای برای نمایش به کاربر.</li> <li>- دریافت و ترکیب داده‌های رتبه‌بندی و نظرات: استفاده از امتیازها و نظرات در فیلترینگ و رتبه‌بندی POI‌ها.</li> <li>- اضافه کردن مکان‌های پیشنهادی کاربران: دریافت موقعیت و مشخصات مکان مورد نظر کاربر و قرار دادن آن در پایگاه داده.</li> </ul>	<b>فرآیندهای مرتبه</b>
<ul style="list-style-type: none"> <li>- فهرست امکانات نزدیک: مجموعه‌ای از POI‌ها شامل شناسه، نام، نوع، مختصات، آدرس، رتبه و در صورت وجود بخشی از اطلاعات تماس؛ این خروجی مستقیماً در Plan Trip و Map استفاده می‌شود.</li> <li>- نتایج فیلترینگ: لیست POI‌های محدودشده بر اساس فیلترهای کاربر (مثلاً هتل‌های یک محدوده‌ی قیمت با رتبه‌ی بالاتر از ۴).</li> <li>- اطلاعات مسیریابی: ساختاری شامل هندسه‌ی مسیر، فاصله، ETA، و در صورت وجود چند گزینه، مجموعه‌ای از مسیرهای جایگزین (برای خودرو و حمل و نقل عمومی).</li> <li>- جزئیات کامل یک امکانات: آدرس دقیق، شماره تماس، وبسایت، رتبه‌بندی کلی و در صورت وجود لینک هدایت به پلتفرم رزرو خارجی.</li> <li>- پیام‌ها و وضعیت‌های خطاب: در صورت قطعی API‌های مسیریابی یا نبودن داده‌ی کافی، پیام‌های خطای قابل فهم و در صورت پیاده‌سازی، مسیر یا پیشنهاد Fallback برای جلوگیری از قطع کامل تجربه‌ی کاربر.</li> </ul>	<b>خروجی‌ها</b>

# تدوین سناریوی عملیاتی (Operational Scenario)

## عنوان سناریو

سفر نوروزی خانواده احمدی به اصفهان با استفاده از سرویس Facilities & Transportation

### شرایط استفاده (Usage Context)

خانواده احمدی (شامل آقای احمدی، خانم احمدی و دخترشان نیلوفر) قصد دارند در تعطیلات نوروز به استان اصفهان سفر کنند. آن‌ها قبل‌از طریق سرویس Wiki با جاذبه‌های شهر (مانند کاخ چهلستون و میدان نقش‌جهان) آشنا شده‌اند و اکنون در فاز برنامه‌ریزی عملی سفر هستند. آقای احمدی به عنوان استفاده‌کننده اصلی سامانه، وارد وبسایت معرفی کشور می‌شود و در سرویس Trip Plan در حال نهایی‌کردن برنامه‌ی روز اول سفر است؛ نمای نقشه شهر اصفهان از طریق Map Service در اختیار او قرار گرفته است.

در این مرحله، سرویس Facilities & Transportation مسئول ارائه اطلاعات امکانات نزدیک (POI)، مسیرهای دسترسی، ETA و داده‌های ترافیکی برای نقاط انتخاب‌شده توسط آقای احمدی است.

### هدف استفاده‌کننده (User Goal)

- پیدا کردن اقامتگاه مناسب در نزدیکی جاذبه‌های اصلی سفر؛
- تعیین بهترین مسیر رانندگی بین هتل و جاذبه‌های انتخاب‌شده، با اطلاع از زمان تقریبی رسیدن (ETA)؛
- آگاهی از امکانات خدماتی ضروری (مثل مراکز درمانی) در طول مسیر و اطراف مقصد، برای اطمینان از امنیت و کاهش استرس در سفر.
- برنامه‌ریزی عملی سفر در یک محیط یکپارچه.

### بازیگران و سرویس‌های درگیر

- آقای احمدی: کاربر اصلی سامانه که برنامه‌ی سفر را طراحی و از مسیریابی استفاده می‌کند.
- خانواده احمدی: استفاده‌کنندگان نهایی نتایج برنامه‌ریزی و مسیریابی در سفر.
- Trip Plan Service: برای طراحی برنامه‌ی روزانه‌ی سفر.
- Map Service: برای نمایش نقشه، مسیر و POI‌ها.
- Facilities & Transportation Service: سرویس مورد مطالعه؛ مسئول امکانات و مسیریابی.
- API: سرویس نقشه‌ی خارجی (مثل Neshan)؛ برای محاسبه‌ی مسیر، ETA و داده‌های ترافیک.

## جریان اصلی تعامل با سرویس (Main Flow)

### a. انتخاب جاذبه‌ها در Trip Plan

آقای احمدی در سرویس Trip Plan شهر اصفهان را انتخاب می‌کند و چند جاذبه‌ی کلیدی (مثل کاخ چهلستون و میدان نقش‌جهان) را به برنامه‌ی روز اول اضافه می‌کند. Trip Plan برای هر جاذبه، مختصات مکانی را از Map Service دریافت می‌کند و برای تکمیل برنامه به سرویس Facilities & Transportation متکی است که این داده‌ها از طریق Facilities می‌باشند.

### b. جستجوی هتل‌های اطراف جاذبه‌ی اصلی

آقای احمدی در نمای نقشه‌ی اصفهان، روی جاذبه‌ی «چهلستون» زوم می‌کند و فیلتر «هتل و اقامتگاه» را فعال می‌کند.

- درخواست شامل موقعیت جاذبه و نوع POI (هتل) به سرویس Facilities & Transportation ارسال می‌شود.

- سرویس Facilities & Transportation، فهرست هتل‌های موجود در شعاع مشخص را از پایگاهداده‌ی

بازیابی کرده و نتیجه را به صورت ساخت‌یافته به Map Service می‌فرستد.

○ Map Service این هتل‌ها را به صورت آیکون روی نقشه نمایش می‌دهد تا خانواده احمدی بتوانند گزینه‌ها را ببینند.

### c. انتخاب اقامتگاه مناسب

آقای احمدی با مشورت خانم احمدی و نیلوفر روی چند هتل کلیک می‌کند، جزئیات آن‌ها (آدرس، موقعیت روی نقشه، امتیاز کاربران، و در صورت وجود لینک رزرو) را بررسی می‌کنند و یکی از آن‌ها را به عنوان هتل (هتل عباسی) به پیشنهاد نیلوفر، محل اقامت انتخاب می‌کند.

### d. درخواست مسیریابی از هتل تا جاذبه

پس از انتخاب هتل، آقای احمدی می‌خواهد بداند صبح روز اول چگونه از هتل تا کاخ چهلستون برود. او در Trip Plan گزینه‌ی

«نمایش مسیر» را برای روز اول انتخاب می‌کند؛ مبدأ = هتل، مقصد = چهلستون.

- درخواست مسیریابی را (شامل مختصات مبدأ و مقصد و نوع حمل و نقل = خودروی شخصی) به سرویس

Facilities & Transportation ارسال می‌کند.

- سرویس Facilities & Transportation، این درخواست را به API سرویس نقشه‌ی خارجی ارسال کرده، مسیر

بهینه و ETA را دریافت می‌کند، پاسخ خام را به یک ساختار داخلی تبدیل کرده و همراه با اطلاعاتی مانند طول مسیر و

زمان تخمینی، به Map Service برمی‌گرداند.

e. نمایش مسیر و زمان سفر روی نقشه:

Map Service مسیر دریافتی را روی نقشه‌ی اصفهان ترسیم می‌کند؛ آقای احمدی می‌تواند ببیند:

- طول مسیر (مثلاً ۶ کیلومتر)،
- زمان تقریبی رسیدن (مثلاً ۱۵ دقیقه)،
- و در صورت وجود، مسیرهای جایگزین (مسیر سریع‌تر، مسیر کوتاه‌تر).

آقای احمدی با توجه به این اطلاعات، زمان خروج از هتل را در برنامه‌ی سفر تنظیم می‌کند و خانواده احمدی تصویر روشنی از زمان‌بندی روز اول پیدا می‌کنند.

f. مواجهه با وضعیت اضطراری / یافتن مرکز درمانی نزدیک:

در میانه‌ی سفر، نیلوفر کمی مجروح می‌شود و خانواده نگران وضعیت او می‌شوند. آقای احمدی تصمیم می‌گیرد سریع‌ترین راه برای رسیدن به مرکز درمانی را پیدا کند.

- او در نمای نقشه، فیلتر «مراکز درمانی» را فعال می‌کند.
- سرویس Facilities & Transportation نزدیک‌ترین بیمارستان‌ها و درمانگاه‌ها را بر اساس موقعیت فعلی خودرو بازیابی کرده، لیست POI‌ها را همراه با فاصله و ETA تقریبی به Map Service برمی‌گرداند.
- این مراکز را روی نقشه و در قالب یک لیست قابل انتخاب نمایش می‌دهد؛ خانواده احمدی یکی از بیمارستان‌ها را انتخاب می‌کنند و فرآیند مسیریابی بین موقعیت فعلی و مرکز درمانی طی می‌شود تا آن‌ها بتوانند در کوتاه‌ترین زمان به آن‌جا برسند.
- 

## پیامد و ارزش افزوده (Outcome & Added Value)

در پایان این سناریو:

- خانواده احمدی بدون خروج از سامانه، از مرحله‌ی تصمیم‌گیری (انتخاب هتل و جاذبه‌ها) تا مرحله‌ی اجرا (مسیریابی، مدیریت ترافیک و رسیدگی به وضعیت اضطراری) تمام نیازهای لجستیکی سفر خود را برطرف کرده‌اند.
- سرویس Facilities & Transportation نشان می‌دهد چگونه با ترکیب داده‌های POI، مسیریابی و ترافیک، شکاف بین «دانش مکانی» و «اجرای واقعی سفر» را پر می‌کند و تجربه‌ای ایمن، قابل‌پیش‌بینی و کم‌استرس برای یک خانواده‌ی واقعی در سفر فراهم می‌سازد.

# مطالعه موردنی و الگوهای برای سرویس ما

## مقدمه

مطالعه موردنی ابزار تحلیلی مهمی در مهندسی نرم افزار است که نشان می دهد یک سامانه واقعی چگونه نیاز کاربران را برآورده کرده، چه عماری و چه محدودیت هایی داشته، و از تجربیات آن می توان در طراحی سیستم های مشابه بهره گرفت. در این پژوهه، با بررسی سامانه هایی مثل Neshan Maps Platform (و به طور مکمل نمونه هایی مثل Waze و Google Maps)، سعی می کنیم ساختار، امکانات، محدودیت ها و درس های کلیدی را استخراج کنیم تا در طراحی سرویس خودمان استفاده شوند.

## بررسی سیستم های واقعی و ارتباط با نیازمندی های ما

### :Neshan Maps Platform

- Neshan یک پلتفرم ایرانی است که API عمومی برای مسیریابی، جستجوی مکان/POI، نقشه، تبدیل مختصات و سایر سرویس های مکان محور ارائه می دهد.
- بر اساس مستندات وب سرویس Neshan، قابلیت هایی مانند Routing API برای خودرو و موتور سیکلت، محاسبه مسیر همراه با در نظر گرفتن ترافیک، و Reverse Geocoding وجود دارد.
- اپلیکیشن Neshan ویژگی هایی مانند مسیریابی با ترافیک زنده، هشدار به مواردی مثل دوربین سرعت یا موانع، و امکان جستجوی اماکن (هتل، رستوران، خدمات و ... ) دارد.

نتیجه: Neshan تقریباً تمام نیازمندی های پایه ما (مسیریابی، POI، نقشه، جستجو) را پشتیبانی می کند و گزینه مناسب و اقتصادی برای زیرساخت است.

### :Google Maps و Waze

- Waze بر اساس داده های crowd-sourced است: کاربران خود گزارش ترافیک، تصادف، موانع، پلیس/دوربین سرعت و وضعیت جاده را ارسال می کنند؛ این باعث به روز رسانی زنده مسیرها و هشدار های پویا می شود.
- Google Maps نیز ترکیبی از داده تاریخی، داده رسمی، و داده زنده دارد؛ برای مسیر های شهری، حمل و نقل عمومی، پیاده روی، و برنامه ریزی سفر مناسب است.
- به علت محدودیت این دو برنامه در ایران، این برنامه ها در مقایسه با نشان اطلاعات کمتری نظیر وضعیت دوربین های کنترل سرعت و دست انداز ها در اختیار ما قرار میدهند و امکانات مسیریابی برای استفاده از وسایل نقلیه همگانی مانند اتوبوس ها و مترو و همچنین میزان شلوغی، ترافیک معابر و مسیریابی پویا بر اساس ترافیک را پشتیبانی نمی کنند.
- تعداد تصاویر مکان های مشابه در Google Maps نسبت به نشان بیشتر بوده و آرشیو کامل تری دارد.
- Google Maps بر خلاف نشان از قابلیت دوربین 360 درجه پشتیبانی می کند.
- تعداد نظرات در Google Maps در مورد مکان های دینی مختلف نسبت به نشان بیشتر است.

## چالش‌ها و محدودیت‌هایی که باید در طراحی مد نظر بگیریم

توضیح و اثر پر سیستم ما	چالش / محدودیت
اگر API سیستمی مثل Neshan قطع شود یا محدودیت نرخ داشته باشد، عملکرد سرویس کاهش می‌یابد — نیاز به fallback داریم.	«وابستگی به سرویس خارجی / API»
ممکن است POI، ترافیک یا داده‌های بعنوان برای همه مناطق موجود نباشد. باید مکانیسم گزارش کاربر یا ذخیره داخلی داشته باشیم.	«کامل نبودن پوشش داده در مناطق کم‌کاربر یا روستایی»
مانند سیستم Waze: اگر گزارش‌ها تأیید نشوند یا اسپم شوند، اطلاعات غلط وارد می‌شود پس نیاز به اعتبارسنجی/تأیید گزارش‌ها.	«داده crowd-sourced قابل اعتماد نیست»
هرچه امکانات بیشتر باشد، UI ممکن است گیج‌کننده شود. طراحی UX باید ساده و قابل کنترل باشد.	«تضاد بین امکانات زیاد و پیچیدگی UI»
استفاده از API تجاری یا تحت مجوز. باید هزینه، محدودیت‌ها، کلید API، نرخ درخواست و شرایط استفاده را بررسی کرد.	«حقوق، مجوز و مصرف API»

## بینش‌ها و توصیه‌های طراحی برای سرویس

- (1) پایه زیرساخت با Neshan: به عنوان اولین و عملی‌ترین انتخاب برای نقشه، مسیریابی و داده POI.
- (2) لایه داده داخلی برای POI خاص و مراکز امدادی: برای پوشش نیازهای خاص (هتل، رستوران، مراکز اضطراری، امکانات بین‌راهی) و تضمین کیفیت داده.
- (3) ترکیب چند منبع داده: Neshan + داده داخلی + امکان گزارش کاربر منجر به پوشش جامع‌تر و قابل اعتمادتر.
- (4) طراحی UI انعطاف‌پذیر با لایه‌های قابل روشن/خاموش شدن: مسیر، POI، امکانات مسیر، هشدار، انتخاب نوع حملونقل، نمایش لیست/نقشه.
- (5) مکانیزم fallback و مدیریت خطأ: وقتی داده بیرونی نیست، سیستم با POI/داده داخلی یا پیام مناسب کاربر را راهنمایی کند.
- (6) مکانیزم اعتبارسنجی گزارش‌ها (برای داده crowd-sourced): اگر گزارش کاربر فعل بشود، باید تأیید یا بازبینی داشته باشد تا از گزارش‌های جعلی جلوگیری شود.
- (7) پشتیبانی از چند حالت حملونقل: خودرو، موتورسیکلت، پیاده، و در صورت امکان حملونقل عمومی / ترکیبی.
- (8) ساده‌سازی UI و تجربه کاربر (UX): عدم فشار اطلاعات زیاد به کاربر، امکان سفارشی‌سازی نمایش.

## نتیجه‌گیری نهایی

با استفاده از تجربه واقعی سیستم‌هایی مانند Neshan، Google Maps و Waze و با در نظر گرفتن محدودیت‌ها و واقعیت‌های محیط ایران امکان طراحی یک سرویس قوی، انعطاف‌پذیر و کاربرمحل برای مسیر، امکانات و لجستیک سفر وجود دارد. اگر طراحی بر پایه زیرساخت مناسب + داده دقیق + معماری ماژولار + UI/UX خوب انجام شود، سرویس شما می‌تواند هم کاربردی، هم قابل اعتماد و هم قابل توسعه باشد.

# داستان‌های کاربر (User Stories)

نمای کلی داستان‌ها

هدف (Goal)	بازیگر (Actor)	عنوان	شناسه
یافتن سریع خدمات موردنیاز (مانند رستوران، پمپ بنزین، موزه) در شعاع مشخص اطراف یک نقطه	گردشگری که روی یک جاذبه روی نقشه کلیک کرده است	بازیابی امکانات نزدیک بر اساس نوع	US-5.1
آگاهی از بهترین مسیر و زمان تقریبی لازم برای جابه‌جایی بین دو نقطه	کاربری که در حال برنامه‌ریزی سفر با خودرو است	محاسبه زمان و مسیر سفر (ETA)	US-5.2
یافتن نزدیک‌ترین و در دسترس‌ترین مرکز درمانی نسبت به موقعیت فعلی	کاربری که در شرایط اضطراری یا نگرانی از سلامت است	جستجوی مراکز درمانی در حالت اضطراری	US-5.3
محدود کردن نتایج جستجو برای یافتن امکانات منطبق با بودجه و کیفیت مدنظر (مثل هتل ۵ ستاره ارزان)	کاربری با محدودیت بودجه یا ترجیحات کیفی خاص	فیلتر امکانات بر اساس پارامترهای جزئی	US-5.4
دسترسی به اطلاعات کلیدی (آدرس، تلفن، رتبه) و لینک هدایت به سرویس رزرو خارجی	گردشگری که قصد رزرو اقامتگاه یا رستوران دارد	مشاهده جزئیات امکانات و لینک رزرو	US-5.5
مقایسه روش‌های مختلف جابه‌جایی (خودرو شخصی، اتوبوس، مترو، قطار) برای انتخاب گزینه‌ی بهینه	کاربری که به دنبال گزینه‌های حمل و نقل متعدد است	پیشنهاد مسیرهای جایگزین (جاده‌ای و عمومی)	US-5.6
انتخاب جاذبه‌های گردشگری مناسب با توجه به نیاز و اولویت‌های شخصی و زمان محدود در سفر	گردشگری که در حال برنامه‌ریزی سفر است	انتخاب جاذبه‌های گردشگری بر اساس ترجیحات و اولویت‌های شخصی	US-5.7
یافتن سریع‌ترین مسیر به نزدیک‌ترین مرکز درمانی با کمترین زمان واکنش، از طریق فعال‌سازی دکمه حالت اضطراری	کاربر در شرایط اضطراری	فعال‌سازی حالت اضطراری و یافتن نزدیک‌ترین مرکز درمانی	US-5.8
مشاهده امکانات مرتبط مانند پمپ بنزین، رستوران و خدمات بین‌راهی در طول مسیر برای برنامه‌ریزی بهتر توقف‌ها	گردشگری که مسیر سفر خود را روی نقشه انتخاب کرده است	نمایش امکانات روی مسیر انتخاب‌شده	US-5.9

## جزئیات داستان‌های کاربر

### US-5.1 – بازیابی امکانات نزدیک بر اساس نوع:

- Actor (بازیگر): گردشگری که در نزدیکی یک جاذبه قرار دارد و نقشه شهر را در سامانه باز کرده است.
  - Goal (هدف): مشاهده امکانات اطراف مقصد یا مسیر و در صورت نبودن یک مکان خاص، امکان پیشنهاد/افزودن مکان جدید
- سناریوی تعامل:

1. کاربر وارد بخش جستجوی امکانات (POI) می‌شود.
2. سیستم فهرست POI‌های موجود را بر اساس موقعیت یا فیلترهای کاربر نمایش می‌دهد.
3. کاربر POI‌ها را مشاهده و یکی را انتخاب می‌کند.
4. اگر مکان مورد نظر کاربر در فهرست نبود، گزینه «افزودن مکان جدید» نمایش داده می‌شود.
5. کاربر اطلاعات پایه مکان (نام، نوع، موقعیت، توضیح) را وارد می‌کند.
6. سیستم درخواست را ثبت می‌کند و در وضعیت «در انتظار تأیید» قرار می‌دهد.

### معیارهای پذیرش (Acceptance Criteria)

- شاعع جستجو باید قابل تنظیم باشد (مثلاً حداقل سه مقدار پیش‌فرض: ۲، ۵ و ۱۰ کیلومتر).
- برای هر POI، حداقل نام، نوع، آدرس، فاصله تقریبی و در صورت وجود رتبه‌ی اولیه نمایش داده شود.
- نتایج باید حداقل ظرف X ثانیه (مثلاً ۳ ثانیه) برای ۹۵٪ درخواست‌ها بارگذاری شوند.
- در صورت نبودن هیچ امکاناتی در شاعع تعریف شده، پیام مناسب (بدون خطای فنی) به کاربر نمایش داده شود.
- گر مکان وجود نداشت، سیستم باید گزینه «پیشنهاد مکان جدید» ارائه دهد.

### وابستگی‌ها (Dependencies)

- Service Map برای موقعیت‌یابی و نمایش نقشه.
- پایگاه داده‌ی POI و در صورت وجود، Service 2 (Rating & Comment & Media) برای رتبه‌بندی.

## US-5.2 - محاسبه زمان و مسیر سفر (ETA):

- Actor: کاربری که در حال برنامه‌ریزی سفر با خودروی شخصی است (مثلاً آقای احمدی در سناریوی خانواده احمدی).
- Goal: محاسبه مسیر بهینه و ETA بین دو نقطه (هتل و جاذبه، دو جاذبه، هتل و مرکز درمانی) برای برنامه‌ریزی زمانی دقیق.

سناریوی تعامل:

1. کاربر در Map Plan Trip یا مبدأ و مقصد را روی نقشه انتخاب می‌کند (یا از لیست هتل/جادبه انتخاب می‌نماید).
2. نوع حملونقل «خودروی شخصی» را انتخاب می‌کند و روی «محاسبه مسیر» کلیک می‌کند.
3. سرویس Facilities & Transportation درخواست مسیریابی (شامل مختصات مبدأ، مقصد و نوع حملونقل) را به API سرویس نقشه خارجی ارسال می‌کند.
4. مسیر بهینه و ETA از سرویس نقشه دریافت و به ساختار داخلی (Geometry، فاصله، زمان تخمینی) تبدیل می‌شود.
5. مسیر و ETA به Service Map ارسال و روی نقشه نمایش داده می‌شود.
6. کاربر، مسیر و زمان را می‌بیند و در صورت نیاز، زمان‌بندی سفر خود را تنظیم می‌کند.

معیارهای پذیرش:

- برای هر درخواست معابر، حداقل یک مسیر همراه با فاصله و ETA باید برگردانده شود.
- میانگین زمان پاسخ برای محاسبه مسیر باید کمتر از ۱.۵ ثانیه برای حداقل ۹۵٪ درخواست‌ها در بار عادی باشد.
- در صورت نامعتبر بودن مختصات یا در دسترس نبودن API مسیریابی، پیام خطای قابل فهم نمایش داده شود.
- داده‌های مسیر باید قابل استفاده در Map (Geometry) باشند و روی نقشه بدون مشکل نمایش داده شوند.

وابستگی‌ها:

- برای نمایش مسیر Service Map.
- برای تعیین موقعیت فعلی در صورت استفاده از «موقعیت من». Service Location
- API های مسیریابی خارجی (مانند Neshan).

### US-5.3 - جستجوی مراکز درمانی در حالت اضطراری:

- Actor: کاربری که در شرایط اضطراری یا نگرانی درباره سلامت یکی از همراهان است.
- Goal: پیدا کردن نزدیکترین و در دسترس‌ترین مراکز درمانی به موقعیت فعلی و شروع سریع مسیریابی به آن‌ها.

#### سناریوی تعامل:

1. کاربر در حین مسیریابی یا مشاهده نقشه، گزینه‌ی «مراکز درمانی نزدیک» را انتخاب می‌کند.
2. سرویس Facilities & Transportation بر اساس موقعیت فعلی کاربر (از Service Location) نزدیک‌ترین بیمارستان‌ها، درمانگاه‌ها و داروخانه‌های شبانه‌روزی را جستجو می‌کند.
3. نتایج همراه با فاصله، ETA تقریبی و نوع مرکز درمانی به Service Map برگردانده و روی نقشه و به صورت لیست نمایش داده می‌شود.
4. کاربر یکی از مراکز درمانی را انتخاب کرده و گزینه‌ی «شروع مسیریابی» را می‌زند.
5. سرویس، مسیر بهینه تا مرکز درمانی را محاسبه و روی نقشه نمایش می‌دهد.

#### معیارهای پذیرش:

- حداقل N مرکز درمانی (مثلًاً تا سقف ۱۰ مورد) به ترتیب کمترین ETA نمایش داده شوند.
- امکان شروع مسیریابی از همان صفحه لیست فراهم باشد.
- این درخواست در اولویت پردازش قرار گیرد و زمان پاسخ آن از آستانه‌ی مشخص (مثلًاً کمتر از ۲ ثانیه در ۹۵٪ موارد) فراتر نزود.
- در صورت نبودن مرکز درمانی در شعاع تعریف شده، پیام مناسب نمایش و پیشنهاد افزایش شعاع داده شود.

#### وابستگی‌ها:

- Service Location برای تعیین موقعیت فعلی کاربر.
- API نقشه‌ی خارجی برای مسیریابی و تخمین ETA.
- POI برای مراکز درمانی.

#### US-5.4 - فیلتر امکانات بر اساس پارامترهای جزئی

- Actor: کاربری با محدودیت بودجه یا ترجیحات خاص
- Goal: محدود کردن نتایج جستجو برای یافتن امکانات منطبق با ترجیحات خاص (مثل هتل با رتبه بالا و قیمت مناسب).

سناریوی تعامل:

1. کاربر فیلترهای مورد نظر خود (مثل رتبه، نوع غذا، محدوده قیمت و ...) را انتخاب می‌کند.
2. سیستم درخواست جستجو را دریافت کرده و آن را پردازش می‌کند.
3. فهرست امکانات مناسب بر اساس فیلترها نمایش داده می‌شود.

معیارهای پذیرش:

- نتایج باید بر اساس فیلترهای مشخص شده نمایش داده شوند.
- حداقل باید نام، نوع و رتبه امکانات در فهرست نمایش داده شود.
- نتایج باید به‌طور سریع و در زمان کم بارگذاری شوند، حتی در صورت درخواست‌های همزمان بالا.

وابستگی‌ها:

- سیستم فیلترینگ که اطلاعات فیلتر شده را از پایگاه داده POI بازیابی می‌کند.
- داده‌های POI (هتل‌ها، رستوران‌ها و دیگر امکانات).
- سیستم نمایش نقشه و اطلاعات به کاربر.
- Rating & Comment میکرو سرویس

## US-5.5 - مشاهده جزئیات امکانات و لینک رزرو:

○ Actor: گردشگری که قصد رزرو دارد

○ Goal: دسترسی به اطلاعات کلیدی امکانات (هتل‌ها، رستوران‌ها، مراکز درمانی، ...) و رزرو آنها

سناریوی تعامل:

1. کاربر از لیست امکانات (در نقشه یا صفحه جستجو) یکی از امکانات مورد نظر خود را انتخاب می‌کند.
2. سیستم جزئیات مربوط به آن امکان (آدرس، شماره تماس، رتبه، نوع امکانات، خدمات ارائه شده) را نمایش می‌دهد.
3. سیستم لینک رزرو به سایتها رزرو خارجی (مانند Airbnb یا Booking.com) را نمایش می‌دهد.
4. کاربر از طریق لینک‌های موجود، به سایت رزرو خارجی منتقل می‌شود و فرآیند رزرو را تکمیل می‌کند.

معیارهای پذیرش:

- برای هر امکان باید حداقل اطلاعات زیر نمایش داده شود: نام، آدرس، رتبه، نوع امکانات، لینک رزرو.
- لینک‌های رزرو باید به طور مستقیم کاربر را به سایتها معتبر برای رزرو منتقل کنند.
- زمان بارگذاری اطلاعات هر امکان باید حداقل ۳ ثانیه باشد.
- در صورت عدم وجود لینک رزرو، باید پیام "رزرو در دسترس نیست" نمایش داده شود.

وابستگی‌ها:

- پایگاه داده POI برای نمایش اطلاعات امکانات.
- سیستم رزرو آنلاین و لینک‌های رزرو خارجی.
- سیستم نمایش جزئیات به صورت واضح و ساختارمند.

## US-5.6 - پیشنهاد مسیرهای جایگزین (جادهای و عمومی):

- Actor: کاربری که به دنبال گزینه‌های حمل و نقل مختلف است
- Goal: مقایسه مسیرهای مختلف (خودروی شخصی، اتوبوس، مترو، قطار) برای انتخاب مسیر بهینه با توجه به زمان و ترافیک.

سناریوی تعامل:

1. کاربر مبدأ و مقصد خود را وارد می‌کند و درخواست مسیریابی می‌دهد.
2. سیستم مسیرهای مختلف را محاسبه کرده و به کاربر نمایش می‌دهد (برای هر نوع حمل و نقل: خودرو، اتوبوس، مترو و قطار).
3. سیستم اطلاعات زمان تقریبی رسیدن (ETA) و فاصله هر مسیر را محاسبه کرده و به کاربر نمایش می‌دهد.
4. در صورت وجود ترافیک یا تأخیر در مسیر اصلی، سیستم به طور خودکار گزینه‌های جایگزین را به کاربر پیشنهاد می‌دهد.

معیارهای پذیرش:

- برای هر درخواست مسیریابی، سیستم باید حداقل یک مسیر جایگزین با ETA، فاصله و اطلاعات ترافیکی ارائه دهد.
- مسیرها باید به طور واضح در فهرستی قابل مقایسه نمایش داده شوند.
- زمان محاسبه مسیر نباید بیشتر از ۳ ثانیه برای هر درخواست باشد.

وابستگی‌ها:

- سرویس مسیریابی برای خودرو، اتوبوس، مترو و قطار.
- داده‌های ترافیکی لحظه‌ای و زمان تخمینی رسیدن (ETA).
- سیستم نمایش نتایج به صورت مرتب و واضح.

## US-5.7 - انتخاب جاذبه‌های گردشگری بر اساس ترجیحات و اولویت‌های شخصی:

- Actor: گردشگری که در حال برنامه‌ریزی سفر است
- Goal: انتخاب جاذبه‌های گردشگری مناسب با توجه به نیاز و اولویت‌های شخصی و زمان محدود در سفر

سناریوی تعامل:

1. کاربر به سیستم وارد شده و مقصد خود را انتخاب می‌کند.
2. سیستم فهرستی از جاذبه‌های گردشگری مرتبط با مقصد ارائه می‌دهد (مانند موزه‌ها، آثار تاریخی، پارک‌ها، و غیره).
3. کاربر می‌تواند از فیلترهای مختلف استفاده کند تا جاذبه‌ها را بر اساس ترجیحات شخصی خود (مانند نوع جاذبه، مدت زمان بازدید، امتیاز کاربران، فاصله از هتل و غیره) انتخاب کند.
4. پس از انتخاب جاذبه‌ها، سیستم برنامه‌ای روزانه برای بازدید از آن‌ها با در نظر گرفتن زمان‌های مناسب و فاصله‌ها ارائه می‌دهد.
5. سیستم به کاربر اطلاع می‌دهد که برای هر جاذبه چه مدت زمان لازم است و پیشنهادات مختلفی برای ترتیب بازدید ارائه می‌دهد تا کاربر بتواند بهترین تجربه را داشته باشد.

معیارهای پذیرش:

- فهرست جاذبه‌ها باید متناسب با مقصد انتخاب شده و نیازهای شخصی کاربر باشد.
- سیستم باید فیلترهایی برای انتخاب جاذبه‌ها بر اساس اولویت‌های کاربر (نوع، مدت زمان بازدید، و ...) فراهم کند.

وابستگی‌ها:

- سیستم پایگاه داده جاذبه‌های گردشگری.
- سیستم فیلترینگ برای جاذبه‌ها.
- سیستم مدیریت زمان برای برنامه‌ریزی سفر و پیشنهادات زمانی.

## US-5.8 - فعالسازی حالت اضطراری و یافتن نزدیک ترین مرکز درمانی:

- Actor: کاربر در شرایط اضطراری یا نگرانی جدی درباره سلامت همراهان است
- Goal: دسترسی سریع به شماره‌ها، مراکز و مسیرهای امدادی (پلیس، اورژانس، آتش‌نشانی) از طریق فعالسازی حالت اضطراری

سناریوی تعامل:

1. کاربر با مشاهده یک وضعیت اضطراری (تصادف، خطر آتش‌سوزی، درگیری، تهدید امنیتی و ...) دکمه «حالت اضطراری» را فعال می‌کند.
2. سیستم فوراً سه دسته مرجع امدادی را نمایش می‌دهد:
  - a. پلیس
  - b. اورژانس پزشکی
  - c. آتش‌نشانی
3. سیستم نزدیک‌ترین مراکز امدادی مرتبط را بر اساس موقعیت کاربر بازیابی می‌کند (مانند ایستگاه پلیس، ایستگاه آتش‌نشانی، مرکز خدمات اورژانس).
4. سیستم امکان تماس فوری با هر یک از نهادهای امدادی را از طریق دکمه Call Now فراهم می‌کند.
5. در صورتی که کاربر نیاز به مراجعه حضوری داشته باشد، سیستم مسیر سریع ترین دسترسی را ارائه می‌دهد.
6. سیستم در تمام مدت، رابط کاربری را در حالت هشدار (رنگ قرمز، آیکون SOS) نگه می‌دارد تا کاربر بتواند سریع تصمیم بگیرد.

معیارهای پذیرش:

- پس از فعالسازی حالت اضطراری، گزینه‌های «پلیس»، «اورژانس» و «آتش‌نشانی» باید به صورت برجسته نمایش داده شوند.
- شماره تماس هر مرکز باید به صورت یک دکمه تماس فوری (One-tap Call) قابل استفاده باشد.
- نزدیک‌ترین مرکز هر نهاد باید بر اساس فاصله یا ETA نمایش داده شود.
- در صورت نبود داده کامل، سیستم باید حداقل شماره تماس اضطراری عمومی را نمایش دهد.
- رابط کاربری باید به صورت بصری حالت اضطراری را نشان دهد (رنگ‌ها و هشدارهای واضح).

وابستگی‌ها:

- سیستم نقشه (Map Service)
- پایگاه داده مراکز امدادی (Police / EMS / Fire Stations)
- API مسیریابی
- سیستم تماس (لینک تلفنی)

## US-5.9 - نمایش امکانات روی مسیر انتخاب شده:

- Actor: گردشگری که مسیر سفر خود را روی نقشه انتخاب کرده است
- Goal: مشاهده امکانات مرتبط (پمپ بنزین، رستوران، خدمات بین راهی و ...) که در طول مسیر قرار دارند تا بتواند توقفهای بهینه تری برنامه ریزی کند

سناریوی تعامل:

- کاربر مسیر سفر خود را پس از انتخاب مقصد مشخص می کند.
- گزینه «نمایش امکانات روی مسیر» را فعال می کند.
- سیستم فهرستی از POI های مرتبط را که در شعاع قابل قبول از مسیر قرار دارند ارائه می دهد (مانند پمپ بنزین، رستوران، استراحتگاه و ...).
- سیستم این POI ها را روی نقشه و در امتداد مسیر نمایش می دهد.
- کاربر می تواند POI موردنظر را انتخاب کرده و جزئیات آن را مشاهده کند.
- در صورت نیاز، کاربر POI انتخابی را به عنوان توقف میانی در برنامه سفر خود قرار می دهد.

معیارهای پذیرش:

- فقط POI هایی نمایش داده شوند که واقعاً در محدوده مسیر کاربر قرار دارند.
- فاصله POI تا مسیر باید به صورت واضح نمایش داده شود.
- نمایش POI ها نباید مسیر اصلی را مخدوش یا شلوغ کند.
- در صورت نبود POI مناسب در مسیر، پیام مناسب به کاربر نمایش داده شود.

وابستگی ها:

- سیستم نقشه (Map Service)
- پایگاه داده POI برای نمایش اطلاعات امکانات

# استخراج نیاز مندی ها

## نیازمندی های عملکردی (Functional Requirements)

1) FR-5.1 – بازیابی امکانات نزدیک بر اساس نوع:

- شرح: سرویس باید قادر باشد برای هر جاذبه، لیستی از امکانات نزدیک (مانند رستوران، هتل، مراکز درمانی و ...) را بر اساس نوع جاذبه و شعاع مشخص بازیابی کند.
- محرک: درخواست کاربر برای مشاهده امکانات نزدیک به یک جاذبه خاص.
- پیششرطها: کاربر باید موقعیت جغرافیایی خود یا موقعیت جاذبه را مشخص کند.
- پسشرطها: لیست امکانات باید بر اساس فیلترهای مشخص شده نمایش داده شود.
- ورودیها: مختصات جاذبه، نوع جاذبه، شعاع جستجو.
- خروجیها: فهرست POI (امکانات) شامل نام، نوع، آدرس و رتبه.
- وابستگیها: داده های POI، سیستم فیلترینگ برای جستجو.
- داستان های کاربر مربوطه: US-5.1.

2) FR-5.2 – یافتن مسیر

- شرح: سیستم باید مسیر را برای سفر بین دو نقطه (مبدأ و مقصد) مشخص کند.
- محرک: درخواست کاربر برای مسیریابی از مبدأ به مقصد.
- پیششرطها: کاربر باید موقعیت مبدأ و مقصد خود را وارد کرده باشد.
- پسشرطها: سیستم باید مسیر دقیق را به کاربر نمایش دهد.
- ورودیها: مختصات مبدأ و مقصد.
- خروجیها: مسیر بهینه با توجه به ترافیک و شرایط مسیر.
- وابستگیها: سرویس مسیریابی، داده های ترافیکی لحظه ای (اگر لازم باشد).
- داستان های کاربر مربوطه: US-5.2.

### FR-5.3 - محاسبه زمان (ETA) (3)

- **شرح:** سیستم باید زمان تخمینی رسیدن (ETA) برای مسیر انتخابی را نمایش دهد.
- **محرك:** درخواست کاربر برای دریافت زمان تخمینی رسیدن (ETA) به مقصد.
- **پیش شرط ها:** مسیر باید توسط سیستم مشخص شده باشد.
- **پس شرط ها:** سیستم باید زمان تخمینی رسیدن (ETA) را نمایش دهد.
- **ورودی ها:** مسیر محاسبه شده و داده های ترافیکی برای ETA.
- **خروجی ها:** زمان تخمینی رسیدن (ETA) بر اساس وضعیت ترافیک.
- **وابستگی ها:** API های ترافیک مانند Google Maps Neshan
- **دادستان های کاربر مربوطه:** US-5.2.

### FR-5.4 - نمایش اطلاعات امکانات (4)

- **شرح:** سیستم باید برای هر امکانات انتخاب شده، اطلاعات کامل شامل نام، آدرس، شماره تماس، رتبه و لینک رزرو را نمایش دهد.
- **محرك:** انتخاب کاربر برای مشاهده جزئیات یک امکان خاص.
- **پیش شرط ها:** کاربر باید یکی از امکانات را انتخاب کرده باشد.
- **پس شرط ها:** اطلاعات امکان انتخابی باید به طور دقیق و کامل نمایش داده شود.
- **ورودی ها:** شناسه یا نام امکان انتخاب شده.
- **خروجی ها:** جزئیات امکان شامل نام، آدرس، رتبه.
- **وابستگی ها:** سیستم نمایش جزئیات و پایگاه داده POI.
- **دادستان های کاربر مربوطه:** US-5.5.

FR-5.5 - رزرو کردن امکانات:

- **شرح:** سیستم باید برای هر امکانات انتخاب شده، لینک رزرو را نمایش دهد تا کاربر بتواند از طریق آن امکان را رزرو کند.
- **محرك:** انتخاب کاربر برای رزرو یک امکان خاص.
- **پیش شرط ها:** کاربر باید یکی از امکانات را انتخاب کرده باشد.
- **پس شرط ها:** لینک رزرو باید به طور دقیق و صحیح برای امکان انتخاب شده نمایش داده شود.
- **ورو دی ها:** شناسه یا نام امکان انتخاب شده.
- **خروجی ها:** لینک رزرو برای امکان انتخاب شده.
- **وابستگی ها:** سیستم رزرو آنلاین و پایگاه داده POI.
- **داستان های کاربر مربوطه:** US-5.5.

FR-5.6 - انتخاب جاذبه های گردشگری بر اساس ترجیحات و اولویت های شخصی:

- **شرح:** سرویس باید به کاربر این امکان را بدهد که جاذبه های گردشگری را بر اساس فیلتر های مختلف (نوع جاذبه، مدت زمان بازدید، رتبه، فاصله از محل اقامت و ...) انتخاب کند.
- **محرك:** درخواست کاربر برای جستجوی جاذبه های گردشگری.
- **پیش شرط ها:** کاربر باید مقصد سفر خود را مشخص کرده باشد.
- **پس شرط ها:** جاذبه ها باید به طور دقیق بر اساس فیلتر های کاربر نمایش داده شوند.
- **ورو دی ها:** فیلتر های انتخاب شده توسط کاربر (نوع جاذبه، رتبه، مدت زمان و ...).
- **خروجی ها:** فهرست جاذبه های گردشگری مناسب با فیلتر ها.
- **وابستگی ها:** داده های جاذبه ها، سیستم فیلتر کردن.
- **داستان های کاربر مربوطه:** US-5.7.

(7) FR-5.7 - حالت اضطراری و دسترسی سریع به پلیس، اورژانس و آتشنشانی:

- **شرح:** سرویس باید به کاربر این امکان را بدهد که با فعالسازی «حالت اضطراری»، فوراً به فهرست مراکز امدادی (پلیس، اورژانس، آتشنشانی) و گزینه تماس اضطراری دسترسی پیدا کند. همچنین سیستم باید نزدیک‌ترین مرکز امدادی را بر اساس موقعیت و ETA نمایش دهد.
- **محرك:** فعالسازی حالت اضطراری توسط کاربر در زمان وقوع حادثه یا وضعیت بحرانی
- **پیش شرط‌ها:**
  - کاربر باید اجازه دسترسی به موقعیت مکانی را داده باشد.
  - سیستم باید به داده‌های مراکز امدادی دسترسی داشته باشد.
- **پس شرط‌ها:**
  - POI‌های مرتبط در امتداد مسیر نمایش داده شوند.
  - فاصله POI از مسیر به صورت واضح ارائه شود.
- **ورودی‌ها:**
  - مسیر انتخاب شده توسط کاربر
  - فعالسازی On-route POI Toggle
- **خروجی‌ها:**
  - فهرست POI‌های نزدیک مسیر
  - نمایش POI روی نقشه به صورت لایه مکمل
- **وابستگی‌ها:**
  - API
  - Map Service
  - دیتا بیس POI‌ها
- **دانستان‌های کاربر مربوطه:** US-5.8

(8) FR-5.8 - انتخاب نوع حمل و نقل (Transport Mode Selector):

- **شرح:** سرویس باید امکان انتخاب حالت‌های مختلف حمل و نقل (خودرو، پیاده‌روی، حمل و نقل عمومی) را ارائه دهد و مسیر و ETA را بر اساس mode انتخابی محاسبه کند.
- **محرك:** انتخاب نوع حمل و نقل توسط کاربر هنگام مسیریابی
- **پیش شرط‌ها:**
  - کاربر مبدأ و مقصد را مشخص کرده باشد.
  - سیستم به API مسیریابی مناسب با mode دسترسی داشته باشد.

• پس شرط‌ها:

- مسیر جدید مطابق mode انتخابی نمایش داده شود.
- ETA متناسب با حالت حمل و نقل محاسبه شود.

• ورودی‌ها: نوع حمل و نقل انتخابی کاربر

• خروجی‌ها:

- مسیر به روز شده
- mode ETA متناسب با

• وابستگی‌ها:

- Map Service
- API نشان

• داستان‌های کاربر مربوطه: US-5.

(9) FR-5.9 - ثبت مکان جدید توسط مدیر

• **شرح:** سیستم باید امکان اضافه کردن مکان جدید (POI) توسط مدیر را فراهم کند. مکان جدید باید شامل اطلاعات پایه باشد (نام مکان، موقعیت جغرافیایی، نوع POI، توضیحات، شماره تماس، ساعت کاری و ...). پس از تأیید، مکان جدید به سیستم اضافه می‌شود.

• **محرك:** درخواست مدیر برای اضافه کردن یک مکان جدید به سیستم.

• پیش شرط‌ها:

- مدیر باید دسترسی لازم به سیستم داشته باشد و اطلاعات مکان جدید باید کامل باشد.
- سیستم به دیتابیس POI متصل باشد.

• پس شرط‌ها:

- مکان جدید به سیستم اضافه و قابل مشاهده توسط کاربران می‌شود.
- در صورتی که مکان تایید نشود، پیام مناسب به مدیر ارسال می‌شود.

• ورودی‌ها:

- نام مکان
- موقعیت جغرافیایی
- نوع POI (рестoran، هتل، مرکز امدادی، خدمات...)
- توضیحات اختیاری، عکس، ساعت کاری
- اطلاعات تماس (اختیاری)

- خروجی ها:

- نمایش مکان جدید برای مدیر جهت بررسی
- ثبت نهایی POI پس از تأیید
- در صورت رد شدن، پیام مناسب به مدیر ارسال می شود

- وابستگی ها:

- Map Service ○
- POI ○ دیتابیس
- سیستم مدیریت ○

- داستان های کاربر مربوطه: US-5.1

- FR-5.10 - ثبت مکان جدید توسط کاربر

**شرح:** سیستم باید امکان اضافه کردن مکان جدید (POI) توسط کاربر را فراهم کند. مکان جدید باید شامل اطلاعات پایه باشد (نام مکان، موقعیت جغرافیایی، نوع POI، توضیحات، شماره تماس، ساعت کاری و ...). پس از ارسال، مکان جدید برای بررسی توسط مدیر به سیستم ارسال می شود.

**محرك:** درخواست کاربر برای اضافه کردن یک مکان جدید به سیستم.

- پیش شرط ها:

- کاربر باید دسترسی لازم برای پیشنهاد مکان جدید را داشته باشد و اطلاعات مکان باید کامل باشد.
- سیستم به دیتابیس POI متصل باشد.

- پس شرط ها:

- مکان جدید در دیتابیس ذخیره می شود.
- اگر کاربر عادی آن را ثبت کرده باشد، در وضعیت «در انتظار تأیید» قرار می گیرد.
- در صورت تأیید مدیر، POI جدید برای همه کاربران نمایش داده می شود.

- ورودی ها:

- نام مکان
- موقعیت جغرافیایی
- نوع POI (رستوران، هتل، مرکز امدادی، خدمات...)
- توضیحات اختیاری، عکس، ساعت کاری
- اطلاعات تماس (اختیاری)

- خروجی ها:

- نمایش مکان جدید برای مدیر جهت بررسی
- ثبت نهایی POI پس از تأیید

- در صورت رد شدن، پیام مناسب به کاربر ارسال می‌شود
- وابستگی ها:

- Map Service ○
- POI ○
- سیستم مدیریت کاربران
- داستان های کاربر مربوطه: US-5.1

## نیازمندی‌های غیرعملکردی (Non-Functional Requirements)

: NFR-5.1 – کارایی (Performance)

- شرح: زمان پاسخ‌دهی برای درخواست‌های مسیریابی باید کمتر از ۲ ثانیه باشد.
- معیار ارزیابی: زمان پاسخ باید کمتر از ۲ ثانیه باشد حتی زمانی که سیستم تحت بار سنگین است.
- سناریو یا انگیزه: در صورت بیش از ۲ ثانیه زمان پاسخ‌دهی، تجربه کاربری کاهش می‌یابد و کاربر احساس نارضایتی خواهد کرد، که در نهایت منجر به کاهش استفاده از سیستم می‌شود.
- داستان‌های کاربر مربوطه: US-5.2, US-5.3

: NFR-5.2 – مقیاس‌پذیری (Scalability)

- شرح: سیستم باید قادر باشد درخواست‌های همزمان بالایی را در هنگام اوج ترافیک (مثلاً در ایام تعطیلات) پردازش کند.
- معیار ارزیابی: سیستم باید قادر به پردازش حداقل ۵۰۰۰ درخواست در هر ثانیه باشد.
- سناریو یا انگیزه: در موقع پیک، مانند تعطیلات یا رویدادهای خاص، سیستم باید به‌طور موثر با حجم بالای درخواست‌ها مقابله کند تا تجربه کاربری بهمود یابد.
- داستان‌های کاربر مربوطه: US-5.2, US-5.6

: NFR\_5.3 – امنیت (Security)

- شرح: تمامی داده‌ها باید به‌صورت رمزنگاری شده و امن ذخیره شوند.
- معیار ارزیابی: استفاده از پروتکل‌های رمزنگاری استاندارد (مثلاً AES-256) برای ذخیره و انتقال داده‌ها.
- سناریو یا انگیزه: از آنجا که کاربران اطلاعات حساس مانند موقعیت مکانی خود را وارد می‌کنند، امنیت داده‌ها برای حفظ اعتماد کاربران بسیار مهم است.
- داستان‌های کاربر مربوطه: US-5.5

: NFR-5.4 – قابلیت اطمینان (Reliability)

- شرح: سیستم باید قادر باشد به‌طور مداوم به کار خود ادامه دهد و در صورت بروز خطا یا مشکل، به سرعت بازیابی شود.
- معیار ارزیابی: سیستم باید حداقل ۹۹.۹٪ در دسترس باشد و زمان تعمیر پس از هر خطا باید بیشتر از ۳۰ دقیقه باشد.
- سناریو یا انگیزه: سیستم باید بتواند در مواجهه با خطاها و مشکلات، عملکرد خود را بدون وقفه حفظ کند. از آنجا که بسیاری از کاربران به سیستم برای مسیریابی و جستجو برای جاذبه‌ها وابسته هستند، هرگونه خرابی یا وقفه در سیستم می‌تواند به تجربه منفی کاربران منجر شود.
- داستان‌های کاربر مربوطه: US-5.6, US-5.3, US-5.2

# اعتبار سنجی نیازمندی ها

❖ FR-5.1 بازیابی و نمایش امکانات نزدیک به جاذبه‌ها

(چه کسی): Who

- کاربران نهایی این نرمافزار مسافران، گردشگران یا کاربران اپلیکیشن هستند که نیاز به پیدا کردن خدمات نزدیک به جاذبه‌ها دارند.

(چه چیزی): What

- سیستم باید قابلیت بازیابی و نمایش امکانات (POI) مانند هتل‌ها، رستوران‌ها، مراکز درمانی و غیره را در نزدیکی جاذبه‌ها فراهم کند.

(چرا): Why

- کاربران نیاز دارند که خدمات و امکانات مرتبط با جاذبه‌ها را به راحتی پیدا کنند تا تجربه بهتری از سفر خود داشته باشند.

(چه زمانی): When

- در ابتدای توسعه، پس از تکمیل بخش‌های اصلی سیستم جستجو و نمایش POI‌ها، این ویژگی باید پیاده‌سازی شود.

(کجا): Where

- این ویژگی باید در بخش جستجوی نزدیک‌ترین امکانات به جاذبه‌ها پیاده‌سازی شود.

(چگونه): How

- از سیستم فیلترینگ برای جستجوی امکانات استفاده می‌شود و باید به پایگاه داده POI متصل شود.

(چقدر): How Much

مدت زمان انجام پروژه: این بخش نیاز به 2 روز زمان دارد تا تمام نیازمندی‌ها برای جستجوی امکانات و اتصال به پایگاه داده POI پیاده‌سازی شوند.

منابع مورد نیاز:

- پایگاه داده POI برای ذخیره‌سازی و دسترسی به اطلاعات امکانات.

- سرور یا سیستم ابری برای پردازش داده‌ها و اتصال به API‌های مرتبط.

هزینه‌ها: هزینه‌های سرور و پایگاه داده برای ذخیره‌سازی و پردازش داده‌ها (که می‌تواند از منابع موجود دانشگاهی یا سرورهای رایگان تأمین شود).

**Who (چه کسی):**

- کاربران نهایی این سیستم شامل رانندگان و مسافرانی هستند که به دنبال مسیریابی و پیدا کردن بهترین مسیر برای سفر به مقصد خود هستند.

**What (چه چیزی):**

- سیستم باید مسیر بهینه را برای سفر بین دو نقطه (مبداً و مقصد) محاسبه کند.

**Why (چرا):**

- کاربران نیاز دارند که مسیر دقیق و بهینه برای رسیدن به مقصد خود را پیدا کنند تا از زمان و هزینه صرفه جویی کنند و سفر بهتری داشته باشند.

**When (چه زمانی):**

- این ویژگی باید بعد از پیادهسازی سیستم جستجو و نمایش امکانات پیادهسازی شود.
- در ابتدای توسعه، پس از پیادهسازی سیستم‌های پایه مسیریابی، این ویژگی باید فعال شود.

**Where (کجا):**

- این ویژگی باید در بخش مسیریابی و نمایش مسیرها پیادهسازی شود.

**How (چگونه):**

- از الگوریتم‌های مسیریابی مانند الگوریتم دایکسترا یا A\* و منابع ترافیکی لحظه‌ای برای محاسبه مسیر بهینه استفاده می‌شود.

**How Much (چقدر):**

- زمان انجام پروژه: 5 روز
- منابع مورد نیاز:

- API‌های ترافیک مانند Neshan Google Maps یا
- پایگاه داده برای ذخیره اطلاعات مسیرها و داده‌های ترافیکی.
- هزینه‌ها: هزینه‌های مربوط به API‌های ترافیک و نیروی انسانی.

Who (چه کسی):

- کاربران نهایی این سیستم شامل رانندگان و مسافرانی هستند که نیاز به زمان تخمینی رسیدن (ETA) به مقصد خود دارند.

What (چه چیزی):

- سیستم باید زمان تخمینی رسیدن (ETA) برای مسیر بین دو نقطه (مبدا و مقصد) محاسبه کند.

Why (چرا):

- کاربران نیاز دارند که زمان دقیق رسیدن به مقصد خود را بدانند تا بتوانند سفر خود را به طور بهتر مدیریت کنند و برنامه ریزی کنند.

When (چه زمانی):

- بعد از پیاده سازی سیستم مسیریابی اولیه و اتصال به داده های ترافیکی.

Where (کجا):

- این ویژگی باید در بخش مسیریابی و نمایش مسیرها پیاده سازی شود.

How (چگونه):

- از API های ترافیکی مانند Neshan API یا Google Maps Traffic API برای محاسبه زمان تخمینی رسیدن استفاده می شود.

- سیستم باید الگوریتم های مسیریابی و داده های ترافیکی را برای محاسبه ETA به طور دقیق ترکیب کند.

How Much (چقدر):

- زمان انجام پروژه: 4 روز

- منابع مورد نیاز:

○ API های تراکم ترافیک مانند Neshan یا Google Maps

- پایگاه داده برای ذخیره اطلاعات مسیرها و داده های ترافیکی.

- هزینه ها: هزینه های مربوط به API های ترافیک و نیروی انسانی.

#### ❖ FR-5.4: مشاهده جزئیات امکانات

(چه کسی): Who

- کاربران این ویژگی می‌توانند مسافران و گردشگران باشند که قصد دارند امکانات نزدیک به مقصد خود را مشاهده کنند.

(چه چیزی): What

- سرویس باید جزئیات امکانات مختلف (هتل‌ها، رستوران‌ها، مراکز درمانی) را نمایش دهد.

(چرا): Why

- کاربران باید به راحتی بتوانند امکانات موجود را مشاهده کرده و برای سفر خود برنامه ریزی کنند.

(چه زمانی): When

- بعد از تکمیل پیاده‌سازی نمایش امکانات و اتصال به پایگاه داده POI‌ها.

(کجا): Where

- این ویژگی باید در بخش جستجوی نزدیک‌ترین امکانات به جاذبه‌ها پیاده‌سازی شود، جایی که کاربر اطلاعات جاذبه‌ها را مشاهده می‌کند و به صورت مستقیم می‌تواند امکانات اطراف آن را بررسی کند.

(چگونه): How

- سیستم باید از فیلترهای جستجو برای انتخاب امکانات نزدیک به جاذبه‌ها استفاده کند و باید به پایگاه داده POI متصل شود تا داده‌های دقیقی در مورد امکانات مختلف (رستوران‌ها، هتل‌ها، مراکز درمانی) ارائه دهد.

(چقدر): How Much

- مدت زمان انجام پروژه: با توجه به ساده بودن این بخش، حدود 2 روز زمان برای پیاده‌سازی این بخش کافی است.
- منابع مورد نیاز:

- پایگاه داده POI برای ذخیره‌سازی داده‌ها.
- سرورهای موجود یا API‌های رایگان برای اتصال به امکانات و نمایش اطلاعات.
- هزینه‌ها: هزینه‌ها به طور عمده شامل هزینه نیروی انسانی و سرورهای رایگان خواهد بود.

## ❖ FR-5.5: رزرو کردن امکانات

(چه کسی): Who

- کاربران این نرمافزار گردشگران هستند که قصد دارند امکانات (هتل‌ها، رستوران‌ها، مراکز درمانی و غیره) را برای سفر خود رزرو کنند.

(چه چیزی): What

- سیستم باید برای هر امکانات انتخاب شده، لینک رزرو را نمایش دهد تا کاربر بتواند از طریق آن امکان را رزرو کند.

(چرا): Why

- کاربران باید قادر باشند تا با استفاده از لینک‌های رزرو، امکانات مورد نظر خود را به راحتی رزرو کنند و سفر خود را تکمیل کنند. این ویژگی به کاربر کمک می‌کند تا فرآیند رزرو را به صورت یکپارچه و بی‌دردسر از داخل نرمافزار انجام دهد.

(چه زمانی): When

- بعد از تکمیل پیاده‌سازی نمایش امکانات و اتصال به پایگاه داده POI‌ها.

(کجا): Where

سیستم باید به پایگاه داده‌های آنلاین رزرو (مانند Airbnb یا Booking.com) متصل شود و لینک‌های رزرو را برای امکانات نمایش دهد. این ارتباط باید امن و بدون مشکل انجام شود تا کاربران بتوانند بدون نگرانی از مشکلات امنیتی، امکانات را رزرو کنند.

(چگونه): How

- چون نمی‌توان از API‌های رزرو آنلاین استفاده کرد، باید یک فرم رزرو داخلی ایجاد شود که کاربران بتوانند از طریق آن رزرو خود را انجام دهند.
- فرم رزرو شامل فیلد‌هایی برای نام، تاریخ رزرو، نوع اتاق یا خدمات و اطلاعات تماس کاربر خواهد بود.
- کاربران پس از پر کردن فرم، اطلاعات رزرو را به پایگاه داده ارسال خواهند کرد.

(چقدر): How Much

- مدت زمان انجام پروژه: چون نیازی به ارتباط با API‌های خارجی نداریم، برای این بخش 3 روز زمان کافی است.
- منابع مورد نیاز:
  - 2 نفر از تیم برای طراحی فرم رزرو و پیاده‌سازی ارتباطات با پایگاه داده.
  - پایگاه داده برای ذخیره‌سازی رزروها و اطلاعات کاربران.
  - ایمیل یا سیستم پیام‌رسان برای ارسال تایید رزرو به کاربران.
- هزینه‌ها: هزینه‌ها شامل نیروی انسانی برای پیاده‌سازی فرم رزرو و همچنین هزینه‌های زیرساختی مانند سرور و پایگاه داده خواهد بود.

## ❖ FR-5.6: انتخاب جاذبه‌های گردشگری بر اساس ترجیحات و اولویت‌های شخصی

(چه کسی): Who

- کاربران نهایی این سیستم مسافران و گردشگران هستند که به دنبال جاذبه‌های گردشگری مناسب با سلیقه خود هستند.

(چه چیزی): What

- سیستم باید به کاربران این امکان را بدهد که جاذبه‌ها را بر اساس فیلترهای مختلف مانند نوع جاذبه، مدت زمان بازدید، رتبه‌بندی و غیره انتخاب کنند.

(چرا): Why

- کاربران نیاز دارند که جاذبه‌ها را بر اساس ترجیحات و محدودیت‌های خود انتخاب کنند تا تجربه سفر بهتری داشته باشند.

(چه زمانی): When

- بعد از تکمیل فیلتر کردن جاذبه‌ها و آماده‌سازی سیستم جستجو.

(کجا): Where

- این ویژگی باید در بخش جستجو و نمایش جاذبه‌ها پیاده‌سازی شود.

(چگونه): How

- باید یک سیستم فیلترینگ برای جستجوی جاذبه‌ها بر اساس ویژگی‌های مختلف (نوع، مدت زمان بازدید و غیره) ایجاد شود.

(چقدر): How Much

- زمان انجام پروژه: 3 روز
- پایگاه داده POI برای ذخیره‌سازی و فیلتر کردن جاذبه‌ها.
- هزینه‌ها: هزینه‌های نیروی انسانی و هزینه‌های پایگاه داده.

## ❖ FR-5.7: حالت اضطراری و دسترسی سریع به پلیس، اورژانس و آتشنشانی:

(چه کسی): Who

کاربرانی که در شرایط اضطراری، حادثه، یا وضعیت تهدیدکننده قرار دارند و نیاز فوری به تعامل با پلیس، اورژانس یا آتشنشانی دارند.

(چه چیزی): What

سیستم باید پس از فعالسازی حالت اضطراری، مراکز امدادی نزدیک (پلیس، اورژانس، آتشنشانی) را نمایش دهد، و امکان تماس فوری (One-tap Call) و مشاهده مسیر دسترسی را فراهم کند.

(چرا): Why

کاربر باید بتواند در شرایط بحرانی سریع‌ترین دسترسی ممکن به مراکز امدادی را داشته باشد تا از موقع آسیب‌های شدید جلوگیری شود و همچنین وجود حالت اضطراری باعث افزایش اعتماد و اینمنی در استفاده از سیستم می‌شود.

(چه زمانی): When

این ویژگی باید از فاز اول پیاده‌سازی UI در دسترس باشد، زیرا ویژگی‌های اضطراری به عنوان الزامات اینمنی، نباید به فازهای بعد موکول شوند.

(کجا): Where

این قابلیت باید در بخش Navigation و امکانات مرتبط با مسیرپیما (Route Planning) پیاده‌سازی شود. دکمه Emergency Mode باید در صفحات اصلی مسیریابی قابل مشاهده باشد.

(چگونه): How

سیستم از داده‌های مراکز امدادی (Police/EMS/Fire Stations) استفاده کرده و نزدیک‌ترین آن‌ها را بر اساس فاصله و ETA بازیابی می‌کند. با API مسیریابی، مسیر کوتاه‌ترین زمان تا مرکز انتخابی را محاسبه می‌کند. یک دکمه تماس فوری برای هر مرکز ارائه می‌دهد.

(چقدر): How Much

- زمان انجام پروژه: 3 روز
- منابع مورد نیاز:
  - 3 نفر از تیم برای پیاده‌سازی ویژگی‌های اضطراری و اتصال به پایگاه داده POI.
  - API‌های مسیریابی و اطلاعات تماس با اورژانس.
  - هزینه‌ها: هزینه‌های نیروی انسانی و هزینه‌های API‌های خارجی.

## ❖ FR-5.8: انتخاب نوع حمل و نقل (Transport Mode Selector)

:Who (چه کسی):

تمام کاربران سیستم شامل گردشگران، رانندگان، و کاربران پیاده که می‌خواهند مسیر مناسب خود را انتخاب کنند.

:What (چه چیزی):

سیستم باید این امکان را فراهم کند که کاربر از میان حالات حمل و نقل مختلف (خودرو، پیاده، حمل و نقل عمومی) یک Mode انتخاب کند و مسیر و ETA بر اساس آن به روزرسانی شود.

:Why (چرا):

مسیر مناسب برای خودرو با مسیر مناسب برای پیاده روی یا حمل و نقل عمومی تفاوت اساسی دارد، پس کاربر باید کنترلی بر نوع مسیر داشته باشد تا بهترین تصمیم را بگیرد.

:When (چه زمانی):

این قابلیت در فاز توسعه مسیریابی سیستم باید پیاده‌سازی شود، یعنی زمانی که ETA و کشف مسیر فعال باشد.

:Where (کجا):

در بخش مسیریابی (Routing) و در UI مربوط به انتخاب مقصد و مشاهده مسیر، دکمه Transport Mode باید کنار گزینه‌های مسیریابی قرار گیرد.

:How (چگونه):

سیستم با تغییر mode، یک درخواست جدید به API مسیریابی ارسال می‌کند. الگوریتم مسیریابی مسیر جدید و ETA مخصوص mode انتخابی را بازمی‌گرداند. UI مسیر و زمان را به روزرسانی می‌کند.

:How Much (چقدر):

- زمان انجام پروژه: 3 روز
- منابع مورد نیاز:
  - 2 نفر از تیم برای پیاده‌سازی سیستم انتخاب نوع حمل و نقل و اتصال به سرویس‌های مسیریابی.
  - API‌های مسیریابی برای محاسبه مسیر.
- هزینه‌ها: هزینه‌های نیروی انسانی و هزینه‌های API‌های حمل و نقل.

## ❖ FR-5.9: اضافه کردن مکان جدید توسط مدیر

(چه کسی): Who

- مدیر سیستم که دسترسی کامل به سیستم دارد و می‌تواند مکان‌های جدید را به پایگاه داده اضافه کند.

(چه چیزی): What

- مدیر باید قادر باشد مکان‌های جدید (POI) را اضافه کند، که شامل اطلاعاتی مانند نام مکان، موقعیت جغرافیایی، نوع POI، توضیحات، شماره تماس و ساعت کاری باشد.

(چرا): Why

- این نیازمندی برای گسترش بانک اطلاعاتی امکانات موجود و افزودن مکان‌های جدید است که برای کاربران قابل دسترس باشد. این عمل به سیستم کمک می‌کند تا اطلاعات جدید و بهروز در اختیار کاربران قرار گیرد.

(چه زمانی): When

- در ابتدای پروژه، بعد از پیاده‌سازی سیستم مدیریت مکان‌ها.
- پس از تکمیل بخش‌های اصلی سیستم مدیریت، این ویژگی باید پیاده‌سازی شود.

(کجا): Where

- این ویژگی باید در پنل مدیریت پیاده‌سازی شود که مدیر سیستم بتواند از طریق آن مکان‌های جدید را اضافه کند.

(چگونه): How

- مدیر باید از یک فرم مخصوص استفاده کند که شامل فیلد‌هایی برای نام مکان، مختصات جغرافیایی، نوع، توضیحات، شماره تماس و ساعت کاری باشد.
- پس از پر کردن فرم، اطلاعات به پایگاه داده ذخیره خواهد شد و سپس تایید می‌شود.
- مکان‌هایی که تایید نشده‌اند، در وضعیت "در انتظار تایید" قرار می‌گیرند و برای نمایش به کاربران قابل دسترس نخواهند بود.

(چقدر): How Much

- مدت زمان انجام پروژه: این بخش ساده است و حدوداً 2 روز زمان برای پیاده‌سازی فرم و ذخیره‌سازی اطلاعات مکان‌های جدید در پایگاه داده کافی است.
- منابع مورد نیاز:
  - 2 نفر از تیم توسعه برای طراحی فرم و پیاده‌سازی رابط کاربری برای مدیر.
  - پایگاه داده برای ذخیره‌سازی اطلاعات مکان‌های جدید.
  - سرور برای پردازش داده‌ها و ذخیره‌سازی اطلاعات.

هزینه‌ها: هزینه‌های نیروی انسانی و استفاده از سرور و پایگاه داده، که عمدهاً به منابع دانشگاهی یا رایگان محدود خواهد بود.

## ❖ FR-5.10: اضافه کردن مکان جدید توسط کاربر

(چه کسی): Who

- کاربران که به سیستم دسترسی دارند و می‌خواهند مکان‌های جدید را به سیستم پیشنهاد دهند (کاربران عادی).

(چه چیزی): What

- سیستم باید به کاربران این امکان را بدهد که مکان‌های جدید (POI) را پیشنهاد کنند. این مکان‌ها باید شامل اطلاعاتی مانند نام مکان، موقعیت جغرافیایی، نوع POI، توضیحات، شماره تماس و ساعت کاری باشد. سپس این اطلاعات برای تایید به مدیر ارسال خواهد شد.

(چرا): Why

- این ویژگی به کاربران این امکان را می‌دهد که مکان‌های جدیدی که در پایگاه داده موجود نیستند، به سیستم پیشنهاد دهند. این کار باعث می‌شود که پایگاه داده به‌طور مستمر به‌روز و کامل باقی بماند.

(چه زمانی): When

- بعد از پیاده‌سازی سیستم مدیریت مکان‌ها توسط مدیر.

(کجا): Where

- این ویژگی باید در پنل کاربری پیاده‌سازی شود که کاربر بتواند از آن برای ارسال مکان‌های پیشنهادی استفاده کند.

(چگونه): How

- کاربران باید از طریق یک فرم آنلاین مکان‌های جدید را پیشنهاد دهند. این فرم شامل فیلد‌هایی برای نام مکان، مختصات جغرافیایی، نوع، توضیحات، شماره تماس و ساعت کاری خواهد بود. پس از ارسال فرم، مکان پیشنهادی در وضعیت "در انتظار تایید" قرار می‌گیرد و برای بررسی توسط مدیر ارسال می‌شود. مدیر بعد از بررسی و تایید مکان‌ها، آنها را به سیستم اضافه خواهد کرد.

(چقدر): How Much

- مدت زمان انجام پروژه: این بخش نسبتاً ساده است و برای پیاده‌سازی فرم پیشنهاد مکان و ارسال آن برای تایید، حدوداً 2 روز زمان برای تکمیل آن کافی است.
- منابع مورد نیاز:
  - پایگاه داده برای ذخیره‌سازی اطلاعات مکان‌های پیشنهادی.
  - سرور برای پردازش داده‌ها و ارسال اطلاعات به مدیر.

**Who (چه کسی):**

- تمامی کاربران نهایی سیستم، شامل مسافران و رانندگان، از این ویژگی بهره‌مند خواهند شد.

**What (چه چیزی):**

- زمان پاسخ‌دهی برای درخواست‌های مسیریابی و جستجو باید کمتر از ۲ ثانیه باشد.

**Why (چرا):**

- برای اطمینان از تجربه کاربری سریع و بدون وقفه، که موجب رضایت کاربران و جلوگیری از ترک سیستم می‌شود.

**When (چه زمانی):**

- این نیازمندی باید از همان ابتدا در فاز توسعه رعایت شود.

**Where (کجا):**

- این ویژگی باید در تمامی بخش‌های مسیریابی، جستجوی POI و نمایش نتایج اعمال شود.

**How (چگونه):**

- از الگوریتم‌های بهینه‌شده و تکنیک‌های کشینگ برای کاهش زمان پاسخ‌دهی استفاده می‌شود.

**How Much (چقدر):**

- زمان انجام پروژه: در طول پروژه

- منابع مورد نیاز: تمام اعضا در هنگام توسعه

- سرور برای پردازش درخواست‌ها و نمایش سریع نتایج.

- هزینه‌ها: هزینه‌های نیروی انسانی و سرور.

(چه کسی): Who

- این نیازمندی به طور کلی برای تمامی کاربران سیستم است که در موقع اوج ترافیک باید سیستم عملکرد مناسبی داشته باشد.

(چه چیزی): What

سیستم باید بتواند حداقل ۵۰۰۰ درخواست همزمان را پردازش کند.

(چرا): Why

- در موقع پیک مانند تعطیلات یا رویدادهای خاص، سیستم باید قادر باشد حجم بالای درخواست‌ها را به طور مؤثر پردازش کند.

(چه زمانی): When

- این نیازمندی باید در طول فاز توسعه پیاده‌سازی شود و از همان ابتدا تست‌های بارگذاری و مقياس‌پذیری انجام شود.

(کجا): Where

- در تمامی سیستم‌های پردازش داده‌ها و سرورهای مسیریابی و جستجو این ویژگی باید پیاده‌سازی شود.

(چگونه): How

- از سرورهای توزیع‌شده و سیستم‌های مقياس‌پذیر مانند Kubernetes برای مقياس‌پذیری استفاده می‌شود.

(چقدر): How Much

- این نیازمندی باید در تمام مراحل توسعه بررسی و توسعه داده شود.
- Kubernetes برای مدیریت مقياس‌پذیری.
- هزینه‌ها: هزینه‌های نیروی انسانی و استفاده از منابع ابری برای مقياس‌پذیری.

(چه کسی): Who

- کاربران نهایی این نرمافزار مسافران، گردشگران، و مدیران سیستم هستند که به امنیت اطلاعات خود نیاز دارند.

(چه چیزی): What

- تمام داده‌ها باید به صورت رمزگاری شده و امن ذخیره شوند و از پروتکل‌های استاندارد برای انتقال داده‌ها استفاده شود.

(چرا): Why

- امنیت داده‌ها برای حفظ حریم خصوصی کاربران بسیار مهم است. در غیر این صورت، داده‌ها ممکن است در معرض سرقت یا دسترسی غیرمجاز قرار گیرند.

(چه زمانی): When

- این نیازمندی باید در مراحل اولیه پیاده‌سازی سیستم، بهویژه قبل از ذخیره و انتقال داده‌ها، اعمال شود.

(کجا): Where

- در بخش‌های ذخیره‌سازی و انتقال داده‌های حساس کاربران و اطلاعات ترافیکی یا مسیریابی.

(چگونه): How

- باید از پروتکل‌های استاندارد رمزگاری مانند AES-256 برای ذخیره و انتقال داده‌ها استفاده شود.

(چقدر): How Much

- حداقل 2 نفر از تیم برای پیاده‌سازی پروتکل‌های امنیتی و رمزگاری.
- پایگاه داده برای ذخیره‌سازی داده‌ها و انتقال ایمن اطلاعات.
- سرور برای انتقال امن داده‌ها.
- هزینه‌ها: هزینه‌های نیروی انسانی و استفاده از الگوریتم‌های رمزگاری استاندارد (مانند AES-256).

: Who (چه کسی)

- تمامی کاربران نرمافزار، از جمله مسافران و رانندگان که به عملکرد پایدار و بدون وقفه نیاز دارند.

: What (چه چیزی)

- سیستم باید قادر باشد به طور مداوم به کار خود ادامه دهد و در صورت بروز خطا، به سرعت بازیابی شود.

: Why (چرا)

- برای اطمینان از عملکرد بدون وقفه و ارائه تجربه کاربری پایدار، سیستم باید در صورت بروز خطا یا مشکل به سرعت به حالت عادی برگردد.

: When (چه زمانی)

- این نیازمندی باید در تمامی فازهای توسعه سیستم پیاده‌سازی شود و همچنین باید سیستم بازیابی به طور مداوم تست و بهینه‌سازی گردد.

: Where (کجا)

- در تمامی سیستم‌های پردازشی و ذخیره‌سازی داده‌ها و همچنین در بخش مسیریابی باید قابلیت بازیابی سریع از خرابی‌ها پیاده‌سازی شود.

: How (چگونه)

- با استفاده از سیستم‌های پایدار و مقاوم، مانند ایجاد پشتیبان‌گیری از داده‌ها، سیستم‌های توزیع شده و تست‌های استرس برای بررسی عملکرد در شرایط بحرانی.

: How Much (چقدر)

- باید در هنگام پیاده‌سازی تمام بخش‌ها به این نیازمندی توجه کرد که تمام بخش‌ها مطمئن باشند.
- حداقل دو نفر از اعضا باید به صورت پیوسته این نیازمندی را تست و بررسی کنند.
- پایگاه داده برای ذخیره‌سازی داده‌ها.
- سیستم‌های توزیع شده برای استفاده از قابلیت‌های مقیاس‌پذیری.

# گزارش کوتاهی از جلسات برگزار شده با تدریسیار

## گزارش جلسه اول اسکرام

تاریخ جلسه: 10 / 09 / 04

حاضرین: معین عنایتی (scrum master) ، بهرام امامی، ایلیا اسدی، دلارام روحانی و امیرعباس انتظاری

دستور جلسه: آشنایی با پروژه و شروع برنامه‌ریزی اولیه

خلاصه فعالیت‌های جلسه:

### 1. تبیین پروژه و تقسیم کار:

اهداف کلی پروژه توضیح داده شد و وظایف اصلی تیم مشخص گردید. هر عضو تیم با نقش و مسئولیت خود آشنا شد تا برنامه‌ریزی و هماهنگی بین اعضا بهبود یابد.

### 2. آشنایی با وظیفه اصلی سرویس:

اعضای تیم با سرویس اصلی که باید پیاده‌سازی شود آشنا شدند و سوالاتی در مورد وظایف و عملکردهای اصلی سرویس مطرح شد تا درک بهتری از نیازمندی‌ها و محدوده‌های پروژه حاصل شود.

### 3. آشنایی با API‌های ارائه شده توسط نشان:

API‌های ارائه شده توسط پلتفرم نشان بررسی شد و اعضای تیم با روش استفاده و محدودیت‌های آن آشنا شدند تا بتوانند در توسعه سرویس از آن‌ها بهره‌برداری کنند.

### 4. تصمیم برای استخراج نیازمندی‌ها:

مقرر شد پس از جلسه، اعضای تیم نیازمندی‌های پروژه را شناسایی و استخراج کنند تا پایه‌ای دقیق برای توسعه داستان‌های کاربر و برنامه‌ریزی اسپرینت فراهم گردد.

تصمیمات و اقدامات پس از جلسه:

- اعضای تیم مستندات API و اطلاعات پروژه را بررسی خواهند کرد.
- نیازمندی‌ها استخراج و برای جلسه بعد آماده خواهند شد.

تقسیم‌بندی وظایف انجام شده به عنوان مرجع کار هر عضو مورد استفاده قرار خواهد گرفت.

## گزارش جلسه دوم اسکرام

تاریخ جلسه: 15 / 09 / 04

حاضرین: معین عنایتی (scrum master) ، بهرام امامی، ایلیا اسدی، دلارام روحانی و امیرعباس انتظاری

مدت زمان جلسه: 30 دقیقه

دستور جلسه: بررسی و بهبود داستان‌های کاربر و نیازمندی‌های پروژه

خلاصه فعالیت‌های جلسه:

### 1. اضافه کردن نیازمندی‌های غیر عملیاتی:

در طول جلسه، نیازمندی‌های غیر عملیاتی (Non-Functional Requirements) شناسایی و به داستان‌های کاربر پیشنهاد شد تا جنبه‌های کیفیتی و عملکردی پروژه مانند امنیت، کارایی و مقیاس‌پذیری پوشش داده شوند.

### 2. کامل‌تر کردن مطالعه موردی:

مطالعه موردی مرتبط با پروژه بررسی و نقاط ضعف آن شناسایی شد تا اطلاعات دقیق‌تر و جامع‌تری برای تیم فراهم شود و درک بهتری از سیستم‌های مشابه حاصل شود.

### 3. شکستن نیازمندی‌ها به موارد کوچک‌تر:

نیازمندی‌های کلی پروژه تحلیل شدند و تصمیم گرفته شد که برای پیاده‌سازی بهتر، این نیازمندی‌ها به زیرمجموعه‌های کوچک‌تر تقسیم شوند.

### 4. اضافه کردن نیازمندی‌های نان‌فانکشنال:

نیازمندی‌های نان‌فانکشنال (غیرکارکردی) شناسایی شدند و مقرر شد پس از جلسه به داستان‌های کاربر اضافه شوند تا ویژگی‌هایی مانند قابلیت استفاده، قابلیت نگهداری و قابلیت اطمینان پروژه لحاظ شوند.

تصمیمات و اقدامات پس از جلسه:

- اعضای تیم نیازمندی‌ها را بازبینی و تکمیل خواهند کرد.
- داستان‌های کاربر با نیازمندی‌های غیر عملیاتی و نان‌فانکشنال به روزرسانی خواهند شد