**سامانه کنترل دستگاه های AVL**

**خلاصه مدیریتی**

در این مستند راهکار جامع مدیریت و کنترل دستگاههای AVL جهت ارائه به شرکت **داده نمایان ماندگار** تشریح شده است. معماری این راهکار در زیر آمده است:

Listener

Core

Fleet Management

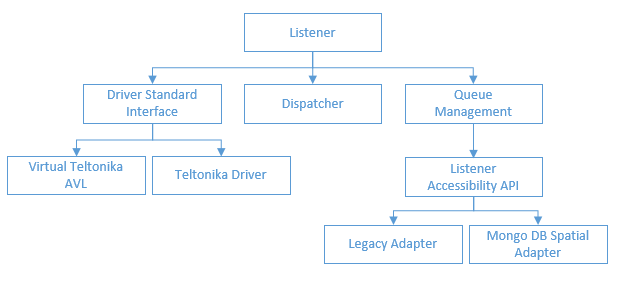
Web Portal

لایه Listener مسئولیت ارتباط با دستگاههای AVL و دریافت و ارسال داده را بر عهده دارد. لایه Core جهت مدیریت داده­های Spatial و لایه Web Portal برای نمایش داده­ها، گزارش­ها و سایر اطلاعات مورد نیاز کاربران طراحی شده­اند. همچنین لایه Fleet Management جهت ارائه سرویس­های مدیریت وسایل حمل و نقل با قابلیت شخصی سازی بالا جهت سازمان­ها و کاربران مختلف طراحی می­گردد. در ادامه قسمت­های هر لایه با جزئیات دقیق توضیح داده شده است.

لازم به ذکر است هر کدام از لایه ها در یک فاز جداگانه توسعه داده شده و در زمانبندی ارائه شده در قسمت زمانبندی قابل تحویل و استفاده می­باشد. این مورد سبب پیشبرد اهداف شرکت به صورت تدریجی و بدون نیاز به انتظار جهت اتمام کل پروژه می­باشد. همچنین هر لایه دارای استقلال کامل می­باشد که مدیریت، پشتیبانی و شخصی سازی لایه های دیگر را تسهیل می­نماید.

**اجزای راهکار**

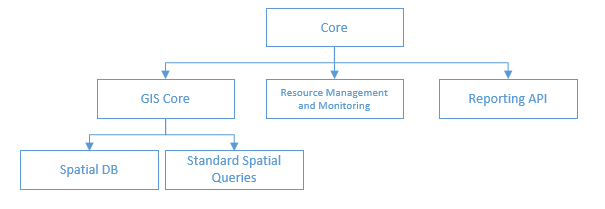
در این قسمت اجزای راهکار با ذکر جزئیات بیشتر ارائه گردیده است.

1. **طراحی و پیاده سازی Listener:**

این فاز شامل موارد زیر می­باشد:

* Driver Standard Interface: چارچوب ارتباط با دستگاه­های مختلف را مشخص می­کند و به عنوان یک لایه استاندارد برای پیاده­سازی درایورهای دستگاه­ها با پروتکل­های مختلف مورد استفاده قرار می گیرد. این قسمت سبب انعطاف پذیری راهکار جهت توسعه سیستم به منظور پشتیبانی از پروتکل­ها و دستگاه­های جدید می­گردد.
* Dispatcher: به عنوان سرور TCP بر پایه تکنولوژی Async I/O طراحی شده است که موجب فشار کمتر بر روی سرور و امکان پاسخگویی همزمان به Client های بیشتری نسبت به روش­های مرسوم می­شود.
* Queue Management: برای افزایش قابلیت اطمینان و Reliability سیستم در این بخش از تکنولوژی Message Queue استفاده می­شود که موجب کمتر شدن coupling بین سرویس­های مختلف و همچنین افزایش Scalability سیستم می­شود.
* Teltonika Driver: به عنوان درایور اصلی در سیستم Listener ارائه می­گردد.
* Virtual Teltonika AVL: دستگاه مجازی بر اساس پروتکل Teltonika FMXXX برای تست Listener تحت شرایط مختلف ( تعداد دستگاه های مختلف، مشکل در ارتباط، بار گذاری با فشار های مختلف بر روی Listener، آلارم و ...). وجود این بخش توسعه و تست سیستم را جهت کاربردهای گوناگون مانند بارگذاری بالا، همزمانی ارتباطات و ... سهولت بخشیده و تسریع می­نماید.
* Listener Accessibility API: لایه ارتباطی سبک بر روی سیستم Messaging در Listener برای ارتباط با سرویس ها و Component های دیگر سیستم. این قسمت سبب افزایش استقلال لایه Listener از سایر لایه­ها و سهولت در مدیریت سیستم می­گردد.
* Legacy Adapter: آداپتور ارتباطی برای Listener جدید که جهت درج داده­های جدید در پایگاه داده کنونی برای پشتیبانی از سیستم فعلی می­باشد. این موضوع سبب می­شود که سیستم فعلی تا زمان اتمام سیستم جدید قابل استفاده باشد.
* Mongo DB Spatial Adapter: آداپتور ارتباطی برای انتقال اطلاعات دریافتی از Listener در قالب اطلاعات استاندار GIS در دیتابیس Mongo در حالت Spatial. این آداپتور همزمان با Legacy Adaptor (و سایر آداپتورهایی که در آینده توسعه داده شوند) قابل استفاده می­باشد، بدین معنی که داده­ها همزمان در هر دو (یا چند) پایگاه داده درج خواهد شد.

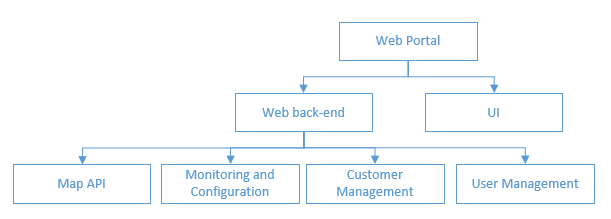
1. **طراحی و پیاده سازی Core:**



این فاز شامل موارد زیر میباشد:

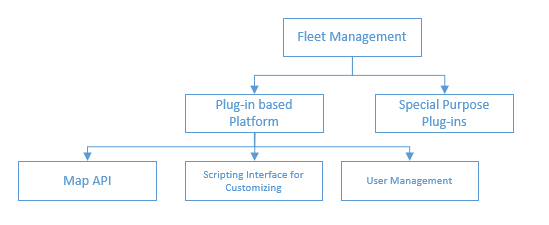
* GIS Core: لایه اصلی برای مدیریت داده­ها و ارتباط با پایگاه داده Spatial.
* Spatial DB: در این قسمت از پایگاه داده Mongo به عنوان پایگاه داده Spatial که مخصوص داده­های جغرافیایی می­باشد، استفاده می­شود. قسمت GIS Core سبب transparency در پایگاه داده مورد استفاده برای سایر لایه­ها و اجزا می­گردد که استقلال این قسمت را تضمین می­نماید.
* Standard Spatial Queries: در این قسمت Query های معمول برای بازیابی داده­ها ( سرعت متوسط، مسافت، جهت حرکت، مسیر و ...) پیاده سازی می­شوند و همینطور یک رابط استاندارد برای پیاده­سازی Query های جدید قرار داده می­شود. این امکان سبب افزایش قدرت بازیابی اطلاعات بر اساس نیازمندی مشتری، بدون نیاز به ایجاد تغییرات در سیستم می­گردد.
* Resource Management and Monitoring: مدیریت و مانیتورینگ دستگاه­های موجود در سیستم ، قابلیت اضافه، حذف و تغییر دستگاه در سیستم، در این قسمت پیاده­سازی خواهد شد.
* Reporting API: قابلیت گزارش­گیری بر اساس داده­های spatial موجود در پایگاه داده در این قسمت پیاده­سازی خواهد شد. این Component قابلیت ایجاد گزارش­های جدید بر اساس Query های تعریف شده (جدید و پایه) در سیستم را دارا می باشد.

1. **طراحی و پیاده سازی Web Portal:**



این فاز شامل موارد زیر می­باشد:

* Web back-end: سرور اصلی پورتال که بر اساس تکنولوژی Rest پیاده­سازی می­گردد.
* Map API: نمایش داده­های گزارش­گیری شده از پایگاه داده سیستم با استفاده از امکانات موجود یکی از نقشه های موجود (Google Maps, Bing Maps, Open Street, Nokia Here )
* Monitoring and Configuration: مانیتورینگ و تنظیم دستگاه­ها بر اساس دید مشتری و ادوات تجهیز شده
* Customer Management: مدیریت حساب مشتریان به صورت فردی، شرکتی و گروهی
* User Management: مدیریت کاربران، سطح دسترسی، وضعیت حساب و کنترل آن
* UI: رابط کاربری تحت وب به صورت Single page application برای سهولت استفاده و Responsive بودن آن.

1. **طراحی و پیاده سازی Fleet Management:**

این فاز شامل موارد زیر میباشد

* Plug-in based Platform: بستر نرم افزار بر اساس تکنولوژی­های Plug-in based برای سهولت در شخصی سازی سیستم.
* Special Purpose Plug-ins: برای سیستم­های خاص منظوره مانند اورژانس و گشت پلیس که این امکانات بدون نیاز به تغییر در کد بر روی بستر فعال می گردند.
* Map API: نمایش داده ها، وضعیت ها و گزارشات سیستم بر روی یکی از نقشه های موجود
* Scripting Interface for Customizing: قابلیت تغییر در بستر و شخصی سازی های دقیق تر با استفاده از زبان های scripting موجود.
* User Management: مدیریت کاربران، سطح دسترسی، وضعیت حساب و کنترل آن.

**زمان­بندی:**

زمان­بندی برای پیاده سازی نرم افزار. هزینه هر ساعت برنامه نویسی بر اساس تعاریف نظام صنفی رایانه ای کشور سال 1391 در نظر گرفته شده است.

|  |  |  |
| --- | --- | --- |
| **نام** | **زمان** | **مبلغ (ریال)** |
| طراحی و پیاده­سازی Listener | 5 هفته | 90،000،000 |
| طراحی و پیاده­سازی Core & DB | 8 هفته | 150،000،000 |
| طراحی و پیاده­سازی Web Portal | 6 هفته | 100،000،000 |
| طراحی و پیاده­سازی Fleet Management | 10 هفته | 160،000،000 |