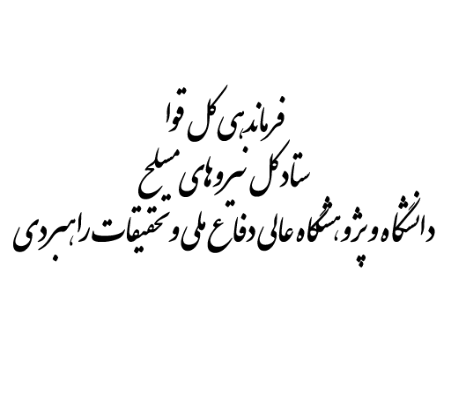
****

**مرکز نخبگان واستعدادهای برتر نیروهای مسلح**

**« گزارش پروژه تحقیقاتی نخبگان »**

|  |  |
| --- | --- |
| **عنوان طرح پژوهشي** | **سامانه‌ی آگاهی وضعیتی تاکتیکی** |

**گزارش دوم**

|  |
| --- |
| **کارگروه تخصصی: فنی و مهندسی** |
| **نام و امضای نماینده سازمان کاربر طرح: مهندس اشکان مقدسی** |
| **نام و امضای مجري طرح: محمدرضا منتظری شاتوری** |
| **نام و امضای ناظر طرح: دکتر بهمن رحمتی** |
| **طبقه‎بندي طرح:** |
| **شماره و تاریخ نامه مصوبه طرح:** |

### روکش گزارش

#### عنوان طرح که به تایید مرکز نخبگان رسیده است :

سامانه‌ی آگاهی وضعیتی تاکتیکی

#### هدف طرح که به تایید مرکز نخبگان رسیده است :

در زمینه‌ی عملیات‌های نظامی آگاهی وضعیتی تاکتیکی یک قابلیت حیاتی برای نیروهای مسلح است. چنین قابلیتی، فرماندهان و نیروهای تحت امر آنان را قادر می سازد تا وضعیت فعلی میدان نبرد، از جمله موقعیت و فعالیت های نیروهای دوست و دشمن، و همچنین وضعیت زمین و شرایط آب و هوایی را بهتر درک کنند. این اطلاعات برای تصمیم گیری آگاهانه در مورد نحوه انجام عملیات و دستیابی به اهداف ماموریت ضروری است.

سامانه‌ی پیشنهادی در این طرح، سامانه‌ای خواهد بود که در آن کاربران در سطوح مختلف، بتوانند آگاهی وضعیتی مورد نیاز برای انجام ماموریتشان را کسب کنند. هر یک از کاربران این سامانه، توسط یک دستگاه الکترونیکی مانند رایانه، گوشی همراه، تبلت و ...) به سامانه متصل شده و سامانه پیشنهادی با استفاده از ظرفیت‌های خاص محیطی مثلا ساختار شبکه‌ی در نظر گرفته شده)، اطلاعات محیطی مورد نیاز را در سطوح مختلف دریافت نموده، روی آن‌ها فرآیندهای لازم برای پردازش آن‌ها را انجام داده و به کاربران متصل به سامانه (با توجه به سطوح دسترسی و میزان نیاز به این اطلاعات) ارسال خواهد کرد تا کاربران به صورت بلادرنگ آگاهی وضعیتی مناسبی را نسبت به محیط خود کسب کنند.

#### شرح خدمات و مراحل انجام و گام های تحقیق و جدول زمان بندی که به تایید مرکز نخبگان رسیده است:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **رديف** | **مراحل و گام‎هاي اجراي پروژه** | **زمان اجرا (ماه)** | | | | | | | | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** |
| 1 | **جمع‌آوری کامل نیازمندی‌ها و طراحی سیستم** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **2** | **پیاده‌سازی رابط کاربری بر مبنای نقشه و نشان دادن نقاط مهم** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **3** | **پیاده‌سازی ارتباط بین کاربران و ذخیره‌سازی پیام‌ها** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **4** | **تست، عیب‌یابی و رفع مشکلات** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **5** | **تهیه مستندات فنی پروژه** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

#### مختصری از گزارش قبلی، اصلاحات درخواست شده مرکز نخبگان و اصلاحات انجام شده

در گزارش پیشین، اقدامات انجام شده در راستای اجرای گام‌های ۱، ۲ و بخشی از گام ۳ پروژه، بیان شد که در راستای عدم نیاز به تکرار موضوعات در گزارش فعلی، شرح آن‌ها در بخش ۴ مستند فعلی آورده شده است. خلاصه‌ی موضوعات ارائه شده در گزارش پیشین شرح زیر است:

* بررسی، اجرا و مقایسه‌های پیاده‌سازی‌های مختلف TAK
* اجرا کردن پروژه از طریق Docker
* پیاده‌سازی ارسال پیام به صورت Multicast
* پیاده‌سازی ارسال پیام به صورت Broadcast
* ایجاد رابط کاربری با زبان فارسی
* پیاده‌سازی سرور نقشه محلی
* تنظیم و اجرا کردن gpsd از طریق Docker، برای دریافت موقعیت کاربر

#### خلاصه نتایج کسب شده در فاز جاری

* پیاده‌سازی سازوکار دریافت اطلاعات از سنسورهای GPS، AIS و Radar
* پردازش پیام‌های استاندارد پروتکل NMEA
* پیاده‌سازی سازوکار ارسال اطلاعات سنسورها به دیگر کاربران
* افزودن قابلیت مشاهده، تعریف و حذف سنسورها در رابط کاربری
* بازتعریف نحوه‌ی ارسال و دریافت اطلاعات بین کاربران به صورت فیدهای ارتباطی
* افزودن قابلیت امکان مشاهده، تعریف و حذف فیدهای ارتباطی در رابط کاربری
* پیاده‌سازی دو نوع فید ارتباطی برای برقراری ارتباط از طریق UDP و RabbitMQ
* رفع مشکلات رابط کاربری و ایجاد تغییرات لازم برای کاربرپسند کردن آن

فهرست مطالب

[۱) عنوان طرح: سامانه آگاهی وضعیتی تاکتیکی 5](#_Toc176695255)

[۲) مقدمه 5](#_Toc176695256)

[۳) بيان مسأله و هدف از انجام طرح 6](#_Toc176695257)

[۳.۱) شرح کلی طرح 6](#_Toc176695258)

[۳.۲) کاربران و ساختار شبکه‌ی ارتباطی سامانه 7](#_Toc176695259)

[۳.۳) ساختار پیام‌های اطلاعاتی 9](#_Toc176695260)

[۳.۴) نقشه 10](#_Toc176695261)

[۳.۴) کیت حمله تاکتیکی (TAK) 11](#_Toc176695262)

[۳.۵) بخش‌های مختلف نرم‌افزار 11](#_Toc176695263)

[۳.۶) حدود طرح 13](#_Toc176695264)

[۴) نتایج کسب شده تا گزارش اول پروژه 13](#_Toc176695265)

[۴.۱) بررسی پیاده‌سازی‌های مختلف TAK 13](#_Toc176695266)

[۴.۲) معماری نسخه‌ی کاربر عادی (سرباز) 14](#_Toc176695267)

[۴.۳) برقراری ارتباط در شبکه‌ی محلی بدون نیاز به سرور 15](#_Toc176695268)

[۴.۴) دریافت موقعیت کاربر 16](#_Toc176695269)

[۴.۵) سرویس نقشه 17](#_Toc176695270)

[۵) نتایج کسب شده تا گزارش دوم پروژه (گزارش فعلی) 19](#_Toc176695271)

[۵.۱) سنسور چیست؟ 19](#_Toc176695272)

[۵.۲) سنسور جی‌پی‌اس 19](#_Toc176695273)

[۵.۳) سنسور AIS 20](#_Toc176695274)

[۵.۴) سنسور رادار 21](#_Toc176695275)

[۵.۵) سنسورها در رابط کاربری سامانه 24](#_Toc176695276)

[۵.۶) فیدهای ارتباطی 25](#_Toc176695277)

[۵.۷) انواع فید‌های ارتباطی 26](#_Toc176695278)

[۵.۸) تنظیم فیدهای ارتباطی 27](#_Toc176695279)

[۵.۳) تغییرات رابط کاربری 28](#_Toc176695280)

[۶) نتیجه گیری 29](#_Toc176695281)

### ۱) عنوان طرح: سامانه آگاهی وضعیتی تاکتیکی

### ۲) مقدمه:

به طور کلی آگاهی وضعیتی توانایی شناسایی، پردازش و درک عناصر مهم اطلاعاتی در مورد آنچه در اطراف اتفاق می افتد، برای تصمیم گیری آگاهانه است. این توانایی شامل آگاهی از محیط اطراف، درک زمینه‌های کلی‌تر موقعیت، و توانایی پیش‌بینی تغییرات و تحولات بالقوه است. آگاهی وضعیتی در زمینه‌های مختلف از جمله عملیات نظامی، واکنش به موقعیت‌های اضطراری (مثل آتش‌نشانی و پلیس)، هوانوردی و بسیاری موارد دیگر ضروری است. با حفظ سطح بالایی از آگاهی وضعیتی، افراد و سازمان ها می توانند تصمیمات بهتری بگیرند، به رویدادهای غیرمنتظره به طور موثر پاسخ دهند و مخاطرات را کاهش دهند.

در زمینه‌ی عملیات‌های نظامی آگاهی وضعیتی تاکتیکی یک قابلیت حیاتی برای نیروهای مسلح است. چنین قابلیتی، فرماندهان و نیروهای تحت امر آنان را قادر می سازد تا وضعیت فعلی میدان نبرد، از جمله موقعیت و فعالیت های نیروهای دوست و دشمن، و همچنین وضعیت زمین و شرایط آب و هوایی را بهتر درک کنند. این اطلاعات برای تصمیم گیری آگاهانه در مورد نحوه انجام عملیات و دستیابی به اهداف ماموریت ضروری است.

جنگ‌های مدرن مستلزم استقرار تعداد زیادی حسگر در تجهیزات و صحنه نبرد هستند. داده‌های صحنه‌ی نبرد توسط تجهیزاتی نظیر رادار، سونار، ویدئو و سنسورهای مادون‌قرمز توسط سیستم‌های هوابرد، ماهوارههای نظارتی، وسایل بدون سرنشین هوایی (UAV) و زمینی (UGV)، کشتیها و سربازان درمیدان نبرد جمعآوری میشوند. این مقادیر عظیم از داده به سطوح بالاتر در قالب توزیعی، تحویل داده می‌شود. تجمیع و پردازش این حجم از داده دارای چالشهایی از قبیل توان پردازشی، مقیاس‌پذیری و پاسخ بی‌درنگ و به موقع برای استفاده از آن‌ها در صحنه نبرد است.

در سال‌های اخیر، علاقه فزاینده‌ای به توسعه سامانه آگاهی وضعیتی تاکتیکی در نیروهای مسلح کشورها وجود داشته است که می‌توانند دید جامع و به‌روز از میدان نبرد را در اختیار نیروها قرار دهند. این سامانه‌ها معمولا داده‌ها را از منابع مختلف، از جمله حسگرها، گزارش‌های اطلاعاتی و مشاهدات انسانی دریافت و یکپارچه می‌کنند. سپس از الگوریتم‌های پیشرفته برای پردازش و تجزیه و تحلیل این داده‌ها استفاده می‌کنند تا آن‌ها را به گونه‌ای به نیروها ارائه کنند که درک و استفاده از آن آسان باشد. از جمله این سامانه‌ها می‌توان سامانه‌ی FBCB2 ساخت ارتش آمریکا را نام برد که با استفاده از شبکه‌ی رادیویی EPLRS و شبکه‌ی ماهواره‌ای BFT اطلاعات مکانی مربوط به نیروهای این کشور را مخابره کرده و در قالب نقشه اختیار فرماندهان قرار می‌دهد.

سامانه‌های آگاهی وضعیتی تاکتیکی این پتانسیل را دارند که شیوه اجرای عملیات‌های نظامی را متحول کنند. با ارائه درک بهتر به نیروهای مسلح از میدان نبرد، این سامانه‌ها می توانند آنها را قادر به تصمیم گیری سریعتر و آگاهانه تر و در نهایت موفقیت بیشتر در نبرد کنند.

به طور کلی قابلیت‌های چنین سامانه‌هایی به سه بخش کلی تقسیم می‌شود:

1. دریافت و یکپارچه‌سازی داده‌ها: این سامانه‌ها باید بتوانند داده‌ها را از منابع مختلف از جمله حسگرها، گزارش‌های اطلاعاتی و مشاهدات انسانی استخراج و جمع‌آوری کنند. پیاده‌سازی چنین قابلیتی با چالش‌های زیادی همراه است، زیرا این داده ها ممکن است در قالب‌های مختلفی بوده و از نظر فرم و محتوا با یکدیگر سازگار نباشند.
2. پردازش داده ها: این سامانه‌ها باید بتوانند حجم زیادی از داده ها را به صورت بلادرنگ پردازش و تجزیه و تحلیل کنند. بنابرین نیاز به توان محاسباتی بالا و اطمینان از اینکه داده ها به طور دقیق و کارآمد پردازش می شوند، در چنین سامانه‌هایی وجود دارد.
3. ارائه داده ها: این سامانه‌ها باید داده ها را به صورتی به نیروها ارائه دهند که درک و استفاده از آن آسان باشد. با توجه به احتمال پیچیدگی داده‌ها و ازدیاد اطلاعات در صحنه‌ی نبرد، مهم است که نحوه‌ی ارائه داده‌ها با توجه به نیاز هر کاربر سامانه، به ساده‌ترین شکل و با اولویت داده‌های مورد نیاز وی انجام پذیرد.

### ۳) بيان مسأله و هدف از انجام طرح:

#### ۳.۱) شرح کلی طرح

سامانه‌ی پیشنهادی در این طرح، سامانه‌ای خواهد بود که در آن کاربران در سطوح مختلف، بتوانند آگاهی وضعیتی مورد نیاز برای انجام ماموریتشان را کسب کنند. هر یک از کاربران این سامانه، توسط یک دستگاه الکترونیکی (مانند رایانه، گوشی همراه، تبلت و ...) به سامانه متصل شده و سامانه پیشنهادی با استفاده از ظرفیت‌های خاص محیطی (مثلا ساختار شبکه‌ی در نظر گرفته شده)، اطلاعات محیطی مورد نیاز را در سطوح مختلف دریافت نموده، روی آن‌ها فرآیندهای لازم برای پردازش آن‌ها را انجام داده و به کاربران متصل به سامانه (با توجه به سطوح دسترسی و میزان نیاز به این اطلاعات) ارسال خواهد کرد تا کاربران به صورت بلادرنگ آگاهی وضعیتی مناسبی را نسبت به محیط خود کسب کنند.

نیازمندی‌های در نظر گرفته شده برای این طرح، به شرح زیر است:

1. قابلیت نشان دادن محل قرارگیری نیروهای خودی بر روی نقشه
2. قابلیت نشانه‌گذاری روی نقشه، برای مشخص کردن نیروی‌های دشمن، نقاط‌های مهم و ...
3. قابلیت ارسال و دریافت پیام‌های متنی (چت)
4. قابلیت انتقال اطلاعات بر روی ساختار شبکه‌ی غیرمتمرکز از طریق ارتباطات رادیویی

#### ۳.۲) کاربران و ساختار شبکه‌ی ارتباطی سامانه

برای توصیف دقیق طرح، ابتدا لازم است انواع کاربران این سامانه و ساختار ارتباط آنان با یکدیگر از طریق سامانه‌ی آگاهی وضعیتی شرح داده شود. با توجه به نیازمندی‌ها و ملاحظات خاص ارتباطی در صحنه‌های نظامی، در ساختار شبکه‌‌ی این سامانه، ارتباطات به صورت غیرمتمرکز و از طریق ارتباطات رادیویی در نظر گرفته شده است. بر این اساس، گروه‌های مختلف کاربران، در چند زیرشبکه قرار می‌گیرند که هر کدام این زیرشبکه‌ها از طریق یکی از گره‌های متصل (که آن‌ها را سرگروه می‌نامیم)، می‌توانند با سطح بالاتر خود که خارج از آن زیرشبکه قرار دارد به طرق دیگر ارتباط برقرار کنند.



شکل 1- ساختار شبکه‌ی سامانه آگاهی وضعیتی

انواع مختلف کاربرانی که از این سامانه استفاده می‌کنند به شرح زیر است:

1. سرباز: اولین نوع کاربر سامانه، سربازانی هستند که در صحنه‌ی عملیاتی با دستگاه خود به یک زیرشبکه از سامانه متصل شده‌اند. کاربرد سامانه برای این افراد، متصل بودن به فرمانده‌ی خود و دیگر سربازان زیرشبکه‌ی خود است تا اولا اطلاعات مربوط به محل حضور آن‌ها جمع‌آوری شده و به باقی کاربران ارسال شود، دوما در صورت نیاز از طریق سامانه بتوانند محل فرمانده و دیگر سربازان زیرشبکه را مشاهده کرده و با آن‌ها از طریق چت ارتباط برقرار کنند. برای اتصال به سامانه، این افراد نیاز به یک دستگاه الکترونیکی (مثلا تلفن همراه یا تبلت) برای مشاهده نقشه‌ی محیط، محل حضور خود و دیگر سربازان، نقاط دیگری مانند محل حضور نیروهای دشمن و سایر اطلاعات دارند. این دستگاه از طریق رادیوی همراه آنان، با سایر افراد زیرشبکه ارتباط برقرار خواهد کرد.
2. سرگروه: در هر زیرشبکه، یک کاربر سرگروه قرار دارد که وظیفه‌ی فرماندهی و هماهنگی بین سربازان گروه خود را برعهده دارد. این کاربر علاوه بر اتصال به سربازان زیرشبکه‌ی خود، از طرق دیگر نیز به زیرشبکه‌های دیگر و شبکه‌ی فرماندهی متصل است. با توجه به شرایط، این ارتباط می‌تواند از طریق رادیوهایی با برد بیشتر یا ارتباطات ماهواره‌ای باشد. نقش سرگروه‌ها در سامانه، این است که بین زیرشبکه‌ها و با مرکز فرماندهی ارتباط برقرار کنند، تا بتوانند اطلاعات زیرشبکه‌ی خود را در اختیار کاربران خارج زیرشبکه قرار داده و هم‌چنین اطلاعات را از کاربران خارج زیرشبکه، دریافت کرده و در صورت لزوم به سربازان متصل به زیرشبکه بفرستند.

با توجه به نقش سرگروه، این کاربر نیاز است که علاوه بر دستگاهی برای مشاهده‌ی نقشه و اطلاعات روی آن، به سخت‌افزارهای دیگری برای ایجاد ارتباط با دیگر زیرشبکه‌ها و مرکز فرماندهی نیز دسترسی داشته باشد. هم‌چنین نرم‌افزار نصب شده برای این کاربر، علاوه بر جمع‌آوری اطلاعات و ارسال آن‌ها در زیرشبکه، باید امکان مسیریابی پیام‌های اطلاعاتی بر روی بیش از یک شبکه ارتباطی را نیز فراهم کند.

1. مرکز فرماندهی: خارج از ساختار زیرشبکه‌ها، یک کاربر مرکز فرماندهی در نظر گرفته شده‌است که بر کل شبکه نظارت دارد. این کاربر اطلاعات را از تمامی زیرشبکه‌ها (از طریق سرگروه‌ها) دریافت کرده و امکان مانیتورینگ کامل و ارسال پیام‌های اطلاعاتی (چت و گزارش‌های مکانی) به هر یک از کاربران سامانه را داراست. نرم‌افزار نصب شده برای این کاربر، علاوه بر امکانات نرم‌افزار سرگروه، امکان آرشیو کردن اطلاعات به صورت بلندمدت را نیز فراهم می‌کند.

در ساختار شرح داده شده، کاربران متصل به یک زیرشبکه، می‌توانند به صورت مستقیم با یک دیگر ارتباط داشته باشند. ولی ارتباط بین زیرشبکه‌ها باید لزوما از طریق از سرگروه‌ها در ساختار شبکه مسیریابی شده و از این مسیر انتقال داده اتفاق بیفتد. این مسئله، چالش‌های بیشتری برای انتقال داده‌های وضعیتی مورد نیاز برای بهبود آگاهی وضعیتی ایجاد خواهد کرد که در طراحی این سامانه در نظر گرفته شده‌اند.

در راستای ایجاد آگاهی **وضعیتی** مناسب، کاربران نیاز به کسب اطلاعات بلادرنگ از محیط خود دارند. به عنوان مثال، یکی از اطلاعات بسیار مهم برای تصمیم‌گیری افراد در موقعیت‌های نظامی، محل قرارگیری نیروهای خودی در صحنه‌ی درگیری‌ است.

اولین گام برای رفع این نیازمندی، دریافت و استخراج این اطلاعات از محیط است. در مورد موقعیت نیروهای خودی، برنامه‌ی نصب‌شده بر دستگاه هر کاربر، از طریق حسگرهای موقعیت‌یاب (مثلا GPS) این اطلاعات را در بازه‌های زمانی مشخص (مثلا هر یک دقیقه) استخراج می‌کنند.

در گام دوم، لازم است اطلاعات محیطی استخراج شده، از طریق شبکه به سامانه آگاهی وضعیتی ارسال شود. با توجه به ساختار غیرمتمرکز شبکه، این اطلاعات در کل زیرشبکه منتشر می‌شود، تا تمامی کاربران در زیرشبکه به آن دسترسی پیدا کنند.

در گام آخر کاربران دیگر سامانه، این اطلاعات را دریافت خواهند کرد. در صورت نیاز بخشی یا تمام این اطلاعات توسط سرگروه به لایه‌های بالاتر شبکه ارسال خواهد شد تا عملیات آرشیو شدن و توزیع به سایر زیرشبکه‌ها اتفاق بیفتد.

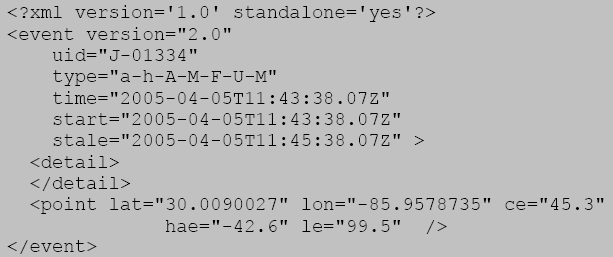
#### ۳.۳) ساختار پیام‌های اطلاعاتی

برای ارسال اطلاعات در بین کاربران مختلف سامانه، نرم‌افزار سامانه این اطلاعات را در قالب پیام‌های اطلاعاتی درآورده و ارسال و دریافت می‌کند. استاندارد پیشنهادی در این طرح برای این پیام‌ها، پروتکل Cursor on Target است که برای مصارف نظامی به خصوص سامانه‌های آگاهی وضعیتی طراحی شده است. این پروتکل در شرکت MITRE و به سفارش نیروی هوایی ارتش آمریکا طراحی شد تا بتواند به عنوان زبانی مشترک برای اتصال انواع سیستم‌های نظامی به هم، به کار گرفته شود. این پروتکل بر پایه‌ی XML ساخته شده است، ولی برای کم کردن اندازه‌ی هر پیام، می‌توان آن را به صورت protobuf نیز ارسال و دریافت کرد.

به طور کلی، هر پیام CoT شامل چند داده‌ی کلی است:

1. شناسه‌ی اتفاق مورد توصیف (فیلد uid)
2. نوع اتفاق (فیلد type)
3. اطلاعات زمانی مربوط به اتفاق (فیلدهای time، start و stale)
4. مکان (تگ point)
5. اطلاعات بیشتر (تگ detail)

یک نمونه‌ی ابتدایی از یک پیام CoT در شکل ۲ آورده شده است. برای درک این پیام، لازم است هر یک از بخش‌های مختلف فیلد type آن را که با خط فاصله مشخص شده است، معنی کنیم.

بخش اول آن (a)، نشان‌دهنده‌ی این است که این پیام مربوط به یک موجودیت واقعی است. مثال‌های دیگری مقادیر دیگر بخش اول، b برای ارسال داده (مثلا برای ارسال چت) و ‌t برای ارسال یک دستور یا درخواست است.

شکل 2 - مثالی از یک پیام CoT

بخش دوم فیلد type در مثال زیر (h)، به معنی این است که شی مورد توصیف، دشمن (hostile) است.

بخش‌های بعدی این فیلد نیز، نشانگر نوع دقیق این نیرو، بر اساس استاندارد MIL-STD-2525 می‌باشد. بنابراین مثال زیر (AMFUM) در مورد یک هواپیمای ثابت‌بال نظامی چندمنظوره‌ی متوسط (Medium Utility Fixed-wing Aircraft) است.

جدا از داده‌های ابتدایی‌ای که در پیام‌های CoT ارسال می‌شود، تگ detail این امکان را به ما می‌دهد تا هر نوع داده‌ی دیگری که مورد نیاز است را داخل پیام خود جا داده و ارسال کنیم.

استفاده از پروتکل CoT به سامانه این قابلیت را می‌دهد که بتواند به راحتی با تمامی سیستم‌های دیگر نظامی و غیرنظامی که از این پروتکل استفاده می‌کنند ارتباط برقرار کند و بتواند اطلاعات دیگر سیستم‌ها را در اختیار کاربران سامانه قرار بدهد. به عنوان مثال، با استفاده از نرم‌افزار تبدیل پیام‌های سامانه‌ی کنترل ترافیک هوایی ADS-B به CoT، می‌توان اطلاعات پروازها را به سامانه‌ی آگاهی وضعیتی فرستاد و در اختیار کاربران قرار داد.

#### ۳.۴) نقشه

قابلیت مهم دیگر پیشنهادی سامانه مورد نظر، قابلیت نمایش اطلاعات در قالب نقشه جغرافیایی در سطوح مختلف کاربران موجود در شبکه است. برای این منظور کاربران لازم است که بتوانند از طریق دستگاه‌های خود به نقشه‌ی محیط خودشان دسترسی داشته تا بتوانند محل حضور دیگر کاربران را روی نقشه مشاهده کنند. همچنین افراد خواهند توانست که اطلاعات دیگر، از جمله محل قرارگیری نقاط مهم را روی نقشه مشخص کرده و برای افراد دیگر به طور خودکار ارسال کنند. این قابلیت‌ها به کاربران امکان این را خواهد داد که بتوانند اطلاعات وضعیتی مهم، به طور خاص اطلاعات جفرافیای محیط عملیاتی خود را با یک‌دیگر به اشتراک بگذارند.

#### ۳.۴) کیت حمله تاکتیکی (TAK):

کیت حمله تاکتیکی یک سامانه آگاهی وضعیتی تاکتیکی ساده‌تر و منبع باز است که توسط انواع سازمان های نظامی و مجری قانون استفاده می شود.

برخی از ویژگی های کلیدی TAK عبارتند از:

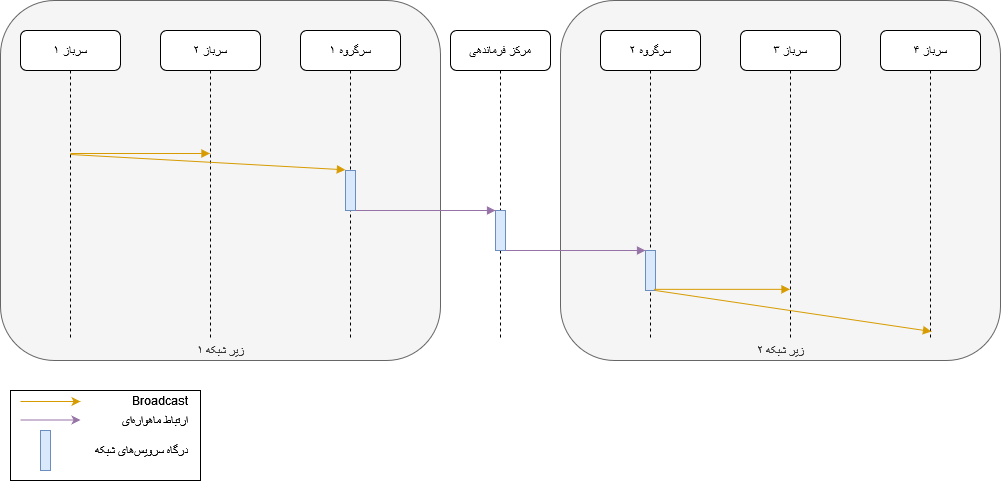
1. تصویر عملیاتی مشترک: TAK تصویر عملیاتی مشترک از میدان نبرد را در اختیار کاربران قرار می دهد. این تصویر شامل اطلاعاتی از موقعیت و فعالیت نیروهای دوست و دشمن و همچنین زمین و شرایط آب و هوایی است.
2. به روز رسانی داده ها: TAK داده های به روزرسانی‌شده از منابع مختلف دریافت می کند. سپس این داده ها پردازش شده و به گونه ای به کاربران ارائه می شود که درک و استفاده از آن آسان باشد.
3. ابزارهای همکاری: TAK شامل انواع ابزارهای همکاری است که به کاربران اجازه می دهد اطلاعات را به اشتراک بگذارند و فعالیت های خود را هماهنگ کنند. این ابزارها شامل چت، صوتی و ویدئو کنفرانس است.

از آن‌جایی که TAK دارای پروتکلی بر پایه‌ی CoT است، قابلیت‌های CoT را ارتقا می‌دهد. معماری TAK بر پایه‌ی کلاینت/سرور طراحی شده است، با این وجود در آن امکان ایجاد ارتباط بین کلاینت‌ها در یک زیرشبکه، بدون دخالت سرور نیز وجود دارد. با توجه به متن باز بودن پروژه‌های کلاینت و سرور TAK، و پوشش دادن بخش خوبی از نیازمندی‌های آگاهی وضعیتی، این سیستم کاندیدای خوبی برای تغییر و پیاده‌سازی نیازمندی‌های سامانه‌ی آگاهی وضعیتی ماست.

#### ۳.۵) بخش‌های مختلف نرم‌افزار

بخش‌های مختلف نرم‌افزار تولیدی برای هر یک از کاربران سامانه، به شرح زیر خواهد بود:

1. **رابط کاربری:** برای نمایش اطلاعات آگاهی وضعیتی، به طور خاص مکان کاربران دیگر سامانه، بخش رابط کاربری بر پایه‌ی نقشه توسعه داده خواهد شد. موقعیت نقاط مهم، نیروهای خودی و دشمن به صورت سمبول‌های استاندارد MIL-STD-2525 روی نقشه نشان داده می‌شوند. این رابط علاوه بر این قابلیت اضافه کردن نقاط بر روی نقشه به منظور به اشتراک‌گذاری مشاهدات کاربران از عوامل محیطی و نیروهای شناسایی شده‌ی دشمن را نیز خواهد داشت. بخش دیگر این رابط مربوط به قابلیت چت، برای ارسال پیام‌های متنی به دیگر کاربران خواهد بود. برای این‌که رابط کاربری مورد نظر قابل استفاده در محیط‌های مختلف باشد، این رابط به صورت یک web app توسعه داده می‌شود.
2. **استخراج داده‌های محیطی:** بسته به نوع حسگرهایی که کاربران سامانه به آن دسترسی دارند، نیاز است که نرم‌افزار پیاده‌سازی شده بتواند داده‌هایی را از حسگرها استخراج کند. داده‌ی مهمی که در حال حاضر برای کاربران عملیاتی سامانه در نظر گرفته شده‌است، داده‌ی مربوط به مکان کاربران است. این داده توسط حسگرهای موجود از جمله GPS در بازه‌های معین استخراج می‌شود تا برای دیگر کاربران از جمله سرگروه ارسال شود.
3. **انتشار پیام:** داده‌های محیطی استخراج شده از حسگرها، نقاط مشخص شده توسط کاربر روی نقشه و پیام‌های چت، در داخل نرم‌افزار سامانه، به پیام‌هایی با استاندارد CoT تبدیل می‌شوند که توسط این بخش نرم‌افزار به کاربران دیگر انتشار می‌یابند. دو وظیفه‌ی اصلی این بخش، یکی مشخص کردن گیرندگان هر پیام، بر اساس نوع پیام و دیگری ارسال پیام از طریق زیرساخت‌های ارتباطی در اختیار سامانه (مثلا رادیو) است.
4. **مدیریت داده‌ها:** داده‌های سامانه، از جمله پیام‌های ارسالی کاربران لازم است که در پایگاه داده‌ی نرم‌افزار سامانه آرشیو و مدیریت شوند. همچنین در هنگام نیاز به بازنگری این پیام‌ها، لازم است که چنین داده‌هایی در پایگاه داده جستجو شده و داده‌های مرتبط با آن پرس‌وجو بر اساس سطح دسترسی کاربر درخواست‌دهنده انتخاب و برای او ارسال شود.
5. **درگاه سرویس‌های شبکه:** با توجه به نیاز سرگروه‌ها به پهنای باند بیشتر برای ارسال اطلاعات به رده‌های بالاتر، ممکن است آن‌ها نیاز به بیش از یک روش برای ارسال اطلاعات داشته باشند. از طرف دیگر، با توجه به نقش خاص آن‌ها و احتمالا حضور آن‌ها در تجهیزاتی مانند نفربرها، امکان استفاده از تجهیزات ارتباطی و مخابراتی بیشتری برای آنان وجود دارد. بنابراین آن‌ها می‌توانند علاوه بر تجهیزات رادیویی در دسترس کاربران، بر اساس روش‌های دیگری (مانند ارتباط ماهواره‌ای یا بی‌سیم‌هایی با برد بیشتر) با کاربران هم‌رده یا بالاتر از خود ارتباط برقرار کنند. بر همین اساس، با توجه به حضور آنان در دو شبکه‌ی مختلف (یکی در سطح زیرشبکه و دیگری شبکه‌ی بالاتر) لازم است که بتوانند این دو شبکه را با یک‌دیگر تلفیق کنند. این اتفاق از طریق درگاه سرویس‌های شبکه اتفاق خواهد افتاد. این درگاه اطلاعات دریافت‌شده از طریق ماهواره یا بی‌سیم‌های دوربرد را با شبکه‌ی رادیویی سطح زیرشبکه به کاربران ارسال خواهد کرد.



شکل 3- نحوه‌ی ارسال و انتشار پیام در سامانه

لازم به ذکر است که تمامی این بخش‌ها، لازم نیست در اختیار تمامی کاربران قرار گیرد. به طور خاص، موارد مربوط به درگاه سرویس شبکه و مدیریت داده‌ها به طور خاص مربوط به سطوح سرگروه و بالاتر هستند، زیرا صرفا در این سطوح لازم است که ذخیره‌سازی و مسیریابی اتفاق بیفتد. برای بقیه‌ی کاربران، صرفا داشتن اطلاعات شخصی و پیام‌های دریافتی برای نمایش در رابط کاربری آنان کفایت خواهد کرد.

#### ۳.۶) حدود طرح:

بر اساس شرح داده شده در این طرح، سرویس‌های نرم‌افزاری لازم برای پیاده‌سازی این سامانه طراحی و پیاده‌سازی خواهند شد. پیش‌نیاز پیاده‌سازی برخی از این سرویس‌ها، امکان ایجاد برقراری ارتباطات رادیویی یا ماهواره‌ای از طریق تجهیزات سخت‌افزاری خاصی است که تهیه، تولید، راه‌اندازی و نگه‌داری آن‌ها خارج از حیطه‌ی این طرح است. در تهیه‌ی این طرح، ویژگی‌های خاص ارتباطات رادیویی (از جمله پهنای باند پایین و احتمال خطای بالا)، در نظر گرفته‌شده‌است و فرض بر این خواهد بود که سرویس‌های نرم‌افزاری سامانه در لایه‌ی Application مدل OSI قرار دارند و می‌توانند روی شبکه‌های رادیویی (و/یا ماهواره‌ای) پکت‌های TCP و UDP به یک‌دیگر ارسال کنند.

### ۴) نتایج کسب شده تا گزارش اول پروژه

همان‌طور که پیش از این گفته شد، سیستم TAK برای پیاده‌سازی نیازمندی‌های آگاهی وضعیتی سامانه‌ی ما بسیار مناسب است. با توجه به در دسترس بودن پروتکل‌های ارتباطی سیستم TAK (شامل CoT و استانداردهای ارتباطی کلاینت با سرور و کلاینت‌های دیگر)، پیاده‌سازی‌های متعدد رسمی و غیررسمی به صورت متن‌باز برای سیستم TAK انجام شده است.

#### ۴.۱) بررسی پیاده‌سازی‌های مختلف TAK

برای بررسی انواع مختلف پیاده‌سازی‌ها، بررسی نقاط قوت و ضعف، و امکان توسعه‌ی سامانه‌ی ما بر پایه‌ی پیاده‌سازی‌های متن‌باز انجام شده، چندین پروژه‌ی در دسترس بررسی شدند:

1. TAKServer: این پروژه، نسخه‌ی رسمی پیاده‌سازی سرور سیستم TAK است که با حذف برخی از قابلیت‌های نظامی آن، به صورت متن‌باز در اختیار همگان قرار داده شده است.

پیاده‌سازی این پروژه با استفاده از زبان جاوا انجام شده است. از لحاظ امکانات و قابلیت‌ها، این پروژه کامل‌ترین پیاده‌سازی را دارد. از طرف دیگر، همین مسئله، باعث پیچیدگی بسیار بالای این پروژه شده است، به طوری که صرفا اجرای این پروژه، نیاز به چندین ساعت ویرایش تنظیمات مختلف دارد.

1. FreeTAKServer: این پروژه، با استفاده از زبان پایتون نوشته شده است و هدف آن این است که تا جای ممکن قابلیت‌های نسخه‌ی رسمی را پیاده کند. این پروژه  مستندات مناسبی برای پروتکل و استاندارد‌های TAK فراهم کرده است که به ما برای پیاده‌سازی سامانه‌ی خود کمک زیادی می‌تواند بکند.

با این وجود، مثل TAKServer انی پروژه صرفا شامل پیاده‌سازی بخش سرور TAK است و نمی‌توان برای پیاده‌سازی کلاینت از آن کمک گرفت.

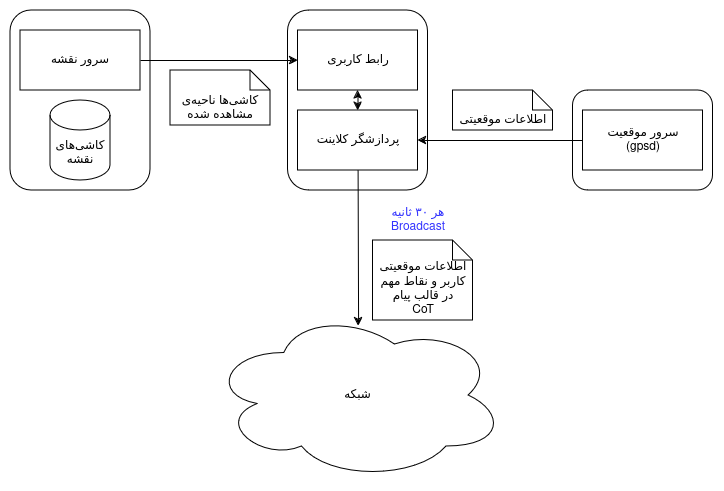
1. GoATAK: این پروژه با زبان go نوشته شده است و شامل هر دو بخش سرور و کلاینت است. بخش رابط کاربری کلاینت این پروژه با استفاده از فریمورک Vue انجام شده است.

از لحاظ پیاده‌سازی قابلیت‌ها و استاندارد‌های TAK، این پروژه قابل مقایسه با پروژه‌های قبلی نیست و امکانات پیاده‌سازی شده در آن، به صورت حداقلی انجام شده است. این مسئله، امکان استفاده از این پروژه، برای ارتقای آن و پیاده‌سازی نیازمندی‌های سامانه‌ی خود ما را فراهم می‌کند، زیرا پیچیدگی‌های غیرلازم و قابلیت‌هایی که در سامانه‌ی ما نیازی به آن‌ها وجود ندارد، در آن نیست.

#### ۴.۲) معماری نسخه‌ی کاربر عادی (سرباز)

در گام نخست برای پیاده‌سازی بخش رابط کاربری و عملکرد‌های اصلی سامانه، معماری نسخه‌ی مربوط به کاربر طراحی شد که شامل بخش‌های زیر است:

1. رابط کاربری، شامل  یک اپلیکیشن وب، به شکل زیر است که به کاربر این امکان را می‌دهد که با سامانه تعامل کند.
2. پردازشگر کلاینت: این بخش، برنامه‌ای است که از یک سو وظیفه‌ی برقراری ارتباط با کاربران دیگر از طریق شبکه را داشته و از طرف دیگر با اتصال به رابط کاربری، ارتباط دو طرفه‌ی لازم برای دریافت فرمان‌های کاربر و هم‌چنین ارائه‌ی اطلاعات دریافت‌شده از کاربران دیگر به رابط کاربری برای نمایش را فراهم می‌کند.
3. سرور موقعیت: این سرویس با دریافت موقعیت از دستگاه GPS و ارسال مداوم موقعیت کاربر به پردازشگر کلاینت نیازمندی ما برای پیدا کردن موقعیت کاربر را رفع می‌کند.
4. سرور نقشه: این نقشه، وظیفه‌ی نگه‌داری و ارائه‌ی نقشه برای نمایش در رابط کاربری را دارد.



شکل 4- بخش‌های مختلف کلاینت

برای پیاده‌سازی نسخه‌ی کاربر، بر پایه‌ی پروژه‌ی GoATAK جلو رفتیم. پس از تنظیم و اجرای موفق این پروژه به صورت تستی، تصمیم بر این شد که برای ایجاد قابلیت اجرا در محیط‌های مختلف، این پروژه با استفاده از Docker اجرا شود. به این دلیل برای برنامه‌ی کلاینت در پروژه‌ی GoATAK یک Dockerfile مناسب نوشته شد که شامل ۲ بخش build و execution است. در بخش build، کتابخانه‌هایی که پروژه به آن‌ها وابسته است، دانلود شده و فایل‌های پروژه کامپایل می‌شوند. در بخش execution فایل‌های کامپایل شده در docker image قرار می‌گیرند، تا سایز نهایی آن کاهش پیدا کند.

با توجه به اجزای متعددی که نیاز است به همراه GoATAK اجرا شوند تا نیازمندی‌های پروژه پوشش داده شود، از docker compose استفاده کردیم تا بتوان تمامی این بخش‌ها را به همراه هم، روی انواع محیط‌ها بدون مسئله اجرا کنیم.

#### ۴.۳) برقراری ارتباط در شبکه‌ی محلی بدون نیاز به سرور

یکی از نیازمندی‌های سامانه‌ی ما این است که بر اساس طراحی انجام شده و ارائه شده در طرح پروژه، کاربران یک زیرشبکه، بتوانند بدون وابستگی به یک سرور، با یک‌دیگر ارتباط برقرار کنند. از آن‌جایی که در GoATAK پیاده‌سازی بر اساس کلاینت/سرور انجام شده است، این امکان وجود نداشت.

در پروتکل TAK برای این امکان، در نظر گرفته شده است که کلاینت‌ها علاوه بر ارسال اطلاعات به سرور، در شبکه‌ی محلی خود نیز به صورت Multicast با یک‌دیگر ارتباط برقرار کنند. روش Multicast روشی برای ارسال اطلاعات به گروه خاصی از گیرندگان است، برخلاف روش Unicast که اطلاعات را فقط به یک گیرنده خاص ارسال می‌کند و روش Broadcast که اطلاعات را به همه گیرندگان موجود در شبکه ارسال می‌کند. Multicast در پروتکل IP نسخه ۴ با استفاده از آدرس‌های IP خاص به نام آدرس‌های Multicast انجام می‌شود. این آدرس‌ها با یک رنج مشخص از آدرس‌های کلاس D (بین ۲۲۴.۰.۰.۰ تا ۲۳۹.۲۵۵.۲۵۵.۲۵۵) تعریف شده‌اند. هنگامی که یک فرستنده قصد ارسال اطلاعات به گروهی از گیرندگان را دارد، به جای آدرس IP تک‌دستگاهی، آدرس IP Multicast گروه را در پکت‌ها قرار می‌دهد. روترهای شبکه این آدرس خاص را تشخیص داده و بسته را به تمام رابط‌های خود که عضو گروه Multicast هستند، تکرار می‌کنند. دستگاه‌های گیرنده نیز از پروتکلی مانند IGMP برای عضویت در گروه‌های Multicast استفاده می‌کنند. با دریافت بسته‌هایی که آدرس Multicast گروه مدنظر را دارند، دستگاه متوجه می‌شود که عضو آن گروه است و در غیر این صورت، بسته‌ها را نادیده می‌گیرد. این روش باعث می‌شود که بسته‌های اطلاعاتی تنها به دستگاه‌های عضو گروه ارسال شوند و از اتلاف پهنای باند و ترافیک اضافی روی کل شبکه جلوگیری به عمل آید.

بر همین اساس، پیاده‌سازی روش ارسال Multicast در GoATAK به زبان Go انجام شد. در این پیاده‌سازی، مستقل از ارتباط کلاینت با سرور، هر کلاینت، در فواصل زمانی مشخص، موقعیت خود، نقاط مهم و انواع پیام‌های CoT دیگر را به آدرس Multicast به آی‌پی ۲۳۹.۲.۳.۱ و پورت ۶۹۶۹ ارسال می‌شود که بر اساس استاندارد TAK انتخاب شده‌اند تا کلاینت‌های ما قابلیت کار کردن با کلاینت‌های رسمی را نیز داشته باشند. هم‌چنین کلاینت‌ها در همین گروه نیز عضو می‌شوند تا بتوانند پیام‌های ارسالی از دیگر کلاینت‌ها را دریافت کنند.

پس از پیاده‌سازی و تست موفق این قابلیت در سیستم‌عامل لینوکس، با این مسئله مواجه شدیم که در سیستم‌عامل ویندوز، با توجه به برخی از محدودیت‌های Docker، پیام‌های Multicast به درستی ارسال نمی‌شوند. پس از تلاش‌های ناموفق برای رفع این مسئله، تصمیم گرفتیم تا علاوه بر Multicast به روش Broadcast نیز اطلاعات را ارسال کنیم.

پیاده‌سازی قابلیت ارسال به صورت Broadcast نیز در کد انجام شد تا کلاینت علاوه بر ارسال پیام به صورت Multicast در شبکه‌ی محلی خود، اطلاعات را به صورت Broadcast نیز ارسال کنند.

#### ۴.۴) دریافت موقعیت کاربر

gpsd یک نرم‌افزار متن‌باز و رایگان است که روی سیستم‌عامل‌های مختلف اجرا می‌شود و به عنوان یک پل بین گیرنده‌های GPS و برنامه‌های کاربردی عمل می‌کند.

gpsd به طور مداوم با گیرنده GPS ارتباط برقرار کرده و داده‌های خام موقعیت‌یابی، از جمله طول و عرض جغرافیایی، زمان، و اطلاعات ماهواره را دریافت می‌کند. سپس داده‌های خام را پردازش و به فرمت‌های قابل استفاده برای برنامه‌های مختلف تبدیل می‌کند. برنامه‌های دیگر می‌توانند با gpsd ارتباط برقرار کرده و به داده‌های موقعیت‌یابی دسترسی پیدا کنند. این امر باعث می‌شود برنامه‌ها بدون نیاز به برقراری ارتباط مستقیم با گیرنده GPS، از موقعیت کاربر یا دستگاه مطلع شوند.

از آنجا که gpsd مسئولیت برقراری ارتباط و مدیریت داده‌های GPS را برعهده دارد، برنامه‌های دیگر نیازی به انجام این کار ندارند که منجر به صرفه‌جویی در منابع سیستم می‌شود. gpsd از گیرنده‌های GPS مختلف پشتیبانی می‌کند و با سیستم‌عامل‌های گوناگون سازگار است.

گیرنده GPS مورد نظر، معمولا سخت‌افزاری است که قابلیت مکان‌یابی را فراهم می‌کند. برای مثال در گوشی‌های هوشمند، چنین سخت‌افزاری وجود دارد و سیستم‌عامل اندروید، با استفاده از نرم‌افزار gpsd با آن ارتباط برقرار می‌کند. از طرف دیگر می‌توان برای سناریوهای تست یا در مواقعی که دسترسی به گیرنده‌ GPS سخت‌افزاری وجود ندارد و موقعیت بر اساس روش‌های دیگری محاسبه می‌شود (مثلا سامانه‌ی AIS در کشتیرانی) می‌توان نوعی گیرنده‌ی خاص برای gpsd تعریف کرد که این امکان را به ما می‌دهد که با ارسال موقعیت روی یک پورت شبکه به gpsd، اطلاعات موقعیتی را به برنامه‌های کاربردی بفرستیم.

در فاز یک پروژه، سرویس gpsd با این گیرنده شبکه تنظیم شده است که به ما امکان تست اتصال سامانه به GPS را می‌دهد. پردازشگر کلاینت، با اتصال به سرویس gpsd، از آن موقعیت فرضی کاربر را دریافت می‌کند. موقعیت دریافت شده، بر روی رابط کاربری نمایش داده شده و در فواصل زمانی معین به دیگر کاربران سامانه ارسال خواهد شد. برای تست این قابلیت، با استفاده از نرم‌افزار OpenCPN و افزونه‌ی ShipDriver توانستیم حرکت یک کشتی در خلیج فارس را شبیه‌سازی کرده و موقعیت این کشتی را از طریق پروتکل UDP به سرویس gpsd به عنوان موقعیت کاربر ارسال کنیم. بدین صورت، با تنظیمات صورت گرفته موفق شدیم در سامانه‌ی آگاهی وضعیتی خود، کاربر را با اتصال به gpsd روی نقشه‌ی سامانه مشاهده کنیم که با موقعیت فرضی کشتی جابجا می‌شود.

برای اجرای این سرویس در محیط‌های مختلف، با پیاده‌سازی Dockerfile مناسب برای این سرویس، امکان اجرای این سرویس از طریق Docker نیز فراهم شد.

#### ۴.۵) سرویس نقشه

همان‌طور که در پروپوزال این پروژه گفته شد، لازم است رابط کاربری این سامانه نقاط مهم (از جمله محل حضور نیروها) را بر روی بستر نقشه نمایش دهد. برای پیاده‌سازی این قابلیت، در رابط کاربری نیاز به استفاده از یک کتابخانه‌ی نقشه است که با استفاده از آن بتوان نقشه را نمایش داده و نقاط مختلف را با استفاده از SDK آن، به نقشه اضافه کرد.

پیاده‌سازی نقشه در سامانه با استفاده از کتابخانه‌ی Leaflet انجام شده است. Leaflet یک کتابخانه جاوااسکریپت متن باز و رایگان است که برای ساختن نقشه‌های تعاملی وب به کار می‌رود. این کتابخانه به دلیل سادگی در استفاده، وزن کم و انعطاف‌پذیری بالا، محبوبیت زیادی در بین توسعه‌دهندگان وب پیدا کرده است.

با استفاده از Leaflet می‌توان نقشه‌های پایه‌ای را از سرویس‌های مختلف (مثلا OpenStreetMap) فراخوانی کرد و سپس لایه‌های اطلاعاتی مورد نظر را بر روی آن قرار داد. این کتابخانه از انواع کنترل‌کننده‌ها مانند زوم، پیمایش، مکان‌یابی و ... پشتیبانی می‌کند که به کاربر امکان برقراری تعامل با نقشه را می‌دهد.

در سرویس‌های ارائه‌ی نقشه، یک نقشه به بخش‌های کوچک‌تر و مدیریت‌‌پذیرتری به نام کاشی (Tile) تقسیم می‌شود. این کاشی‌ها می‌توانند مربع یا مستطیل شکل باشند و در یک شبکه سازماندهی می‌شوند. هر کاشی بزرگ‌نمایی و مختصات جغرافیایی خاص خود را دارد و به طور مستقل قابل دانلود است. به عنوان مثال، در بزرگنمایی صفر (کمترین حالت ممکن بزرگنمایی)، کل نقشه‌ی جهان در یک کاشی قابل دانلود است. با یک‌بار زوم کردن روی نقشه، وارد بزرگنمایی یک می‌شویم، که در آن کل نقشه‌ی جهان، به ۴ کاشی تقسیم شده است، که با توجه به ناحیه‌ی قابل مشاهده‌ی کاربر تنها کاشی‌های قابل مشاهده لازم است دانلود شوند. به همین ترتیب، هر چه بزرگنمایی بیشتر شود، تعداد کاشی‌ها بیشتر می‌شود، ولی هر کدام از آن‌ها ناحیه‌ی کوچک‌تری را، با جزئیات بیشتر نمایش می‌دهد.

**مزایای کاشی‌کاری نقشه:**

* **بارگذاری سریع‌تر:** به جای کل نقشه، فقط کاشی‌های مربوط به ناحیه قابل مشاهده کاربر دانلود می‌شوند. این امر باعث می‌شود نقشه سریع‌تر بارگذاری شود و کاربر بتواند سریع‌تر با آن کار کند.
* **پیمایش روان:** با حرکت و زوم کردن روی نقشه، تنها کاشی‌های مورد نیاز دانلود می‌شوند. این باعث می‌شود پیمایش و زوم کردن روی نقشه روان و بدون وقفه انجام شود.
* **مصرف کمتر حافظه:** برنامه‌های کاربردی فقط کاشی‌های قابل مشاهده را در حافظه نگه می‌دارند، که منجر به استفاده کارآمدتر از حافظه دستگاه کاربر می‌شود.
* **کشینگ مؤثر:** مرورگرها می‌توانند کاشی‌های دانلود شده را کش کنند. بدین صورت، هنگام بازدید مجدد از همان ناحیه، نیازی به دانلود مجدد کاشی‌ها نخواهد بود.

کتابخانه‌ی Leaflet به صورت پیش‌فرض از سرویس آنلاین OpenStreetMap استفاده می‌کند، که نقشه‌ی جهان را به صورت رایگان ارائه می‌دهد. با اتصال به این سرویس، کتابخانه‌ی Leaflet، کاشی‌های نقشه ناحیه‌ی قابل مشاهده کاربر را دانلود کرده و نمایش می‌دهد. با این وجود، برای استفاده از سرویس آنلاین OpenStreetMap نیاز است دستگاه کاربر، به شبکه‌ی اینترنت متصل باشد، که قطعا به خصوص در محیط‌های عملیاتی، امکان‌پذیر نیست.

برای ارائه‌ی نقشه به کاربر، بدون اتصال به اینترنت، نیازمند این هستیم که کاشی‌های مورد نیاز کاربر را پیشاپیش دانلود کرده و با شبیه‌سازی سرویس آنلاین OpenStreetMap، در زمان لازم آن‌ها را به Leaflet ارائه دهیم.

برای انجام این کار، با استفاده از یک برنامه‌ی متن باز، به نام wms-tiles-downloader، کاشی‌های نقشه‌ی سرتاسر جهان را در بزرگنمایی‌های صفر تا ۵ دانلود کردیم. با توجه به این‌که با افزایش بزرگنمایی، تعداد کاشی‌ها (و متناسب با آن، حجم فایل‌ها) به صورت نمایی زیاد می‌شود،  امکان دانلود کل نقشه‌ی جهان در تمامی بزرگنمایی‌ها و قرار دادن آن در دستگاه کاربر وجود ندارد. با این وجود، برای امکان استفاده از نقشه در سامانه، کاشی‌های نقشه‌ی کشور ایران، تا بزرگنمایی ۱۱ دانلود شد تا کاربر بتواند روی نقشه، بر روی کشور ایران، میزان بیشتری زوم کند.

پس از این، نیازمند این بودیم که بتوانیم کاشی‌های دانلود شده را مشابه سرویس OpenStreetMap ارائه دهیم. برای این‌کار، یک وب‌سرور ساده با استفاده از امکانات زبان Python پیاده کردیم. از طرف دیگر، در کد‌های مربوط به بخش رابط کاربری، روی نقشه‌ی ارائه شده توسط Leaflet دو لایه نقشه اضافه کردیم، که یکی نقشه‌ی تمامی جهان را تا بزرگنمایی ۵، و دیگری نقشه‌ی ایران را تا بزرگنمایی ۱۱ نمایش مي‌دهد. با انداختن این دو لایه روی هم، کاربر کل نقشه‌ی جهان را می‌بیند، و اگر در حال دیدن محدوده‌ی کشور ایران باشد، می‌تواند با زوم کردن بیشتر، نقشه را با دقت بیشتری نیز ببیند.

فایل‌های دانلود شده به همراه وب‌سرور ارائه‌دهنده‌ی آن‌ها، با استفاده از docker در یک docker image قرار گرفتند تا بتوان آن‌ها را به همراه بخش‌های دیگر سامانه به سادگی روی دستگاه کاربر اجرا کرد.

### ۵) نتایج کسب شده تا گزارش دوم پروژه

در پیاده‌سازی‌های پیشین بخش نرم‌افزار کاربر با رابط کاربری مبتنی بر نقشه انجام شد که با استفاده از آن کاربران می‌توانستند با ارتباط با ابزار gpsd مکان فعلی خود را دریافت کرده و این اطلاعات به طور خودکار با دیگر کاربران به اشتراک گذاشته می‌شود. پس از این با صحبت با ناظر پروژه و نماینده‌ی سازمان، تصمیم بر این شد که تمرکز بر روی قابلیت‌هایی قرار بگیرد که می‌تواند به نیروهای استفاده کننده از این سامانه، امکان تنظیم ساده‌تر نحوه‌ی ارتباطات بین کاربران را بدهد و با توجه به نیازهای این نیروها، این امکان افزوده شود که بتوان اطلاعات مهم دیگری، مانند اطلاعات انواع سنسورهای مختلف را نیز بین کاربران ارسال و دریافت کرد.

#### ۵.۱) سنسور

با توجه به این که انواع مختلفی از نیروها، می‌توانند کاربران سامانه باشند و هر یک از آن‌ها ممکن است به دستگاه‌هایی دسترسی داشته باشند که داده‌های مختلفی تولید می‌کند و اهمیت انتقال این داده‌ها به دیگر کاربران، تصمیم بر این شد که موجودیتی به نام سنسور برای سامانه تعریف شود. سنسور در این سامانه به معنای منبع اطلاعاتی‌ای است که می‌توان در بازه‌های زمانی مشخص از آن نوعی داده را دریافت کرد. برای اثبات ایده، چندین سنسور برای سامانه پیاده‌سازی شده است که قابلیت این را دارند که با پروتکل ارتباطی خاص دستگاه موردنظر، داده‌ها را دریافت کرده و از طریق سامانه به دیگر کاربران ارسال کنند.

##### ۵.۱.۱) سنسور جی‌پی‌اس

در پیاده‌سازی‌های پیشین، با توجه به اهمیت مکان‌یابی نیروها، با استفاده از gpsd و برقراری ارتباط با آن، مکان جغرافیای کاربران دریافت و به دیگران ارسال می‌شد. با توجه به تعریف موجودیت سنسور در سامانه، در اولین گام اطلاعات مکانی دریافتی از gpsd به صورت یکی از انواع سنسورها در نظر گرفته شد. به این صورت، اطلاعات جی‌پی‌اس نیز مانند باقی سنسورها دریافت و پردازش می‌شود و از حالت خاص پیشین درآمده است.

سنسور تعریف شده برای جی‌پی‌اس، با اتصال به gpsd و ایجاد یک اشتراک، تغییرات مکانی کاربر را دریافت کرده و در پیام‌های ارسالی کاربر جایگذاری می‌کند. با توجه به این‌که پیام‌های ارسالی کاربر برای معرفی خود و ارسال اطلاعات خود به دیگر کاربران با پروتکل CoT انجام می‌شود و در پیام‌های CoT فیلدهای مربوط به موقعیت مکانی وجود دارد، نیازی به ارسال فیلدهای اضافه‌ای برای اطلاعات این سنسور وجود ندارد.

##### ۵.۱.۲) سنسور AIS

سامانه شناسایی خودکار (AIS) یک سیستم ارتباطی دریایی است که برای افزایش ایمنی و کارایی کشتی‌ها طراحی شده است و به کشتی‌ها و مقامات ساحلی این امکان را می‌دهد تا داده‌های بلادرنگ مانند موقعیت، سرعت و مسیر را از طریق فرکانس‌های رادیویی VHF به اشتراک بگذارند. این سیستم نقش مهمی در جلوگیری از برخورد، عملیات جستجو و نجات، و مدیریت ترافیک دریایی دارد. AIS از طریق فرستنده‌هایی که بر روی کشتی‌ها نصب شده‌اند کار می‌کند و به صورت خودکار اطلاعات کشتی را به سایر کشتی‌ها، ماهواره‌ها و ایستگاه‌های ساحلی ارسال می‌کند. انواع مختلفی از فرستنده‌های AIS وجود دارد که کلاس A برای کشتی‌های بزرگ‌تر مانند کشتی‌های باری و مسافری الزامی است و کلاس B برای کشتی‌های کوچک‌تر استفاده می‌شود.

دستگاه‌های AIS کلاس A انواع مختلفی از پیام‌ها را ارسال می‌کنند که حاوی اطلاعات ضروری برای ایمنی دریایی هستند. این پیام‌ها به طور کلی به سه نوع اصلی تقسیم می‌شوند: داده‌های دینامیک، استاتیک و اطلاعات مربوط به سفر. داده‌های دینامیک شامل موقعیت بلادرنگ کشتی، سرعت، مسیر و وضعیت ناوبری است که به صورت مکرر ارسال می‌شود. داده‌های استاتیک شامل اطلاعات هویتی کشتی مانند نام کشتی، شماره شناسه سرویس دریایی (MMSI) و ابعاد کشتی است که هر ۶ دقیقه پخش می‌شود. پیام‌های مربوط به سفر شامل اطلاعاتی مانند مقصد کشتی، زمان تخمینی ورود و نوع محموله هستند. ترکیب این نوع پیام‌ها به نظارت لحظه‌ای و مدیریت موثر ترافیک دریایی کمک می‌کند.

با توجه به این که اطلاعات AIS از طریق امواج رادیویی ارسال می‌شوند، برای گرفتن این اطلاعات در سامانه، نیازمند یک رابط نرم‌افزاری هستیم که بتواند اطلاعات را به فرمت متنی مناسب تبدیل کند. چون نرم‌افزار gpsd از AIS نیز پشتیبانی می‌کند و با ایجاد اشتراک در آن، پیام‌های مربوط به اطلاعات AIS دستگاه‌های متصل به gpsd نیز توسط سامانه دریافت می‌شود، برای پیاده‌سازی سنسور AIS نیز سامانه‌ی ما از gpsd استفاده می‌کند.

بر خلاف اطلاعات GPS در پروتکل CoT اطلاعات AIS فیلد خاصی ندارند. بر همین اساس نیاز بود که با توسعه‌ی پروتکل CoT و افزودن فیلدهایی به آن، امکان ارسال اطلاعات ‌AIS فراهم شود. با توجه به ؟؟؟؟؟؟پذیری پروتکل CoT به بخش Details پیام، فیلدی با نام SensorsData تعریف شد که لیستی از موجودیتی به نام SensorData است. موجودیت SensorData شامل دو فیلد name و data است که برای سنسور AIS اطلاعات با نام AIS فرستاده می‌شوند و فیلد data نیز شامل اطلاعات دریافت شده از gpsd است که اطلاعات AIS را در قالب JSON درآورده است. برای ارسال اطلاعات این سنسور، با توجه به این که اطلاعات مربوط به خود کاربر است (و نه شی یا شخص دیگری) در پیامی که کاربر به صورت عادی برای معرفی و ارسال اطلاعات خود به دیگران می‌فرستد، اطلاعات این سنسور نیز جای‌گذاری می‌شود.

قابل ذکر است که موجودیت SensorData و توسعه‌ی پروتکل CoT برای افزودن لیستی از این موجودیت به پیام‌ها، به ما این امکان را می‌دهد که مستقل از نوع سنسور، اطلاعات دریافت شده از انواع سنسورهای دیگر را نیز در پیام‌ها جا داده و ارسال کنیم و سنسور AIS صرفا برای اثبات ایده و نشان دادن این قابلیت پیاده‌سازی شده است.

##### ۵.۱.۳) سنسور رادار

سنسور رادار، وظیفه‌ی این را دارد که اطلاعات دریافت شده از رادار را پردازش کرده و اهداف مشخص شده در رادار را با قالبی مناسب به صورت پیام‌های CoT به دیگر کاربران بفرستد. برای دریافت اطلاعات از رادار، سامانه از طریق ارتباط UDP به داده‌های ارسالی رادار که با پروتکل NMEA ارسال می‌شوند گوش می‌دهد.

NMEAیک پروتکل استاندارد شده است که امکان ارتباط بین دستگاه‌های الکترونیکی دریایی مختلف مانند گیرنده‌های GPS، فرستنده‌هایAIS ، عمق‌سنج‌ها و رادارها را فراهم می‌کند. این پروتکل به این دستگاه‌ها اجازه می‌دهد که داده‌ها را در قالبی ساختاریافته به اشتراک بگذارند و سازگاری و همکاری بین تجهیزات مختلف را تضمین می‌کند. در ‌NMEA اطلاعات از طریق پیام‌هایی به نام Sentence یا جمله منتقل می‌شوند. این جملات رشته‌هایی متنی هستند که از فرمت خاصی پیروی می‌کنند. هر جمله، معمولا با یک علامت دلار ($) شروع شده و پس از آن یک دنباله از فیلدهای داده قرار می‌گیرد که با ویرگول از هم جدا شده‌ند. هر جمله نوع خاصی از اطلاعات را نشان می‌دهد، مثلا مختصات جغرافیایی، سرعت یا مسیر. ساختار این جملات به دقت تعریف شده است تا دستگاه‌ها بتوانند داده‌ها را به درستی تفسیر و پردازش کنند.

به طور خاص برای داده‌های ارسالی از رادار، عموما سه جمله از NMEA استفاده می‌شوند:

**۱. RSD (داده‌های سیستم رادار):**

جمله RSD اطلاعاتی درباره پارامترهای سیستم رادار ارائه می‌دهد. این جمله به سایر سیستم‌ها کمک می‌کند تا تنظیمات فعلی و داده‌های اهداف شناسایی شده توسط رادار را درک کنند.

فرمت نمونه:

13

1 2 3 4 5 6 7 8 9 10 11 12| 14

| | | | | | | | | | | | | |

$--RSD,x.x,x.x,x.x,x.x,x.x,x.x,x.x,x.x,x.x,x.x,x.x,a,a\*hh <CR><LF>

1) فاصله مبدأ 1 از کشتی

2) جهت مبدأ 1، درجه از 0°

3) نشانگر متغیر محدوده 1 (VRM1)، فاصله

4) خط جهت 1 (EBL1)، درجه از 0°

5) فاصله مبدأ 2

6) جهت مبدأ 2

7) VRM 2، فاصله

8) EBL 2، درجه

9) فاصله نشانگر از کشتی

10) جهت نشانگر به صورت ساعت‌گرد از 0 درجه

11) مقیاس فاصله در حال استفاده

12) واحدهای فاصله: K = کیلومتر، N = مایل دریایی، S = مایل زمینی

13) چرخش نمایشگر

14) چک‌سام

نکته ۱) چرخش نمایشگر:

C = مسیر به بالا، مسیر نسبت به زمین رو به بالا، درجه واقعی

H = سر به بالا، جهت کشتی (خط مرکز) 0° رو به بالا

N = شمال به بالا، شمال واقعی 0° رو به بالا

نکته ۲) مبدأ 1 و مبدأ 2 در محدوده و جهت مشخص‌شده از کشتی قرار دارند و دو مجموعه مستقل از نشانگرهای متغیر محدوده (VRM) و خطوط جهت الکترونیکی (EBL) را که از موقعیت کشتی جدا هستند، فراهم می‌کنند.

**۲. TLL (عرض و طول جغرافیایی هدف):**

جمله TLL موقعیت جغرافیایی (عرض و طول جغرافیایی) یک هدف راداری ردیابی شده را گزارش می‌دهد. این جمله برای ادغام داده‌های اهداف راداری با سیستم‌هایی مانند نقشه‌نگارهای الکترونیکی (Chart Plotter) استفاده می‌شود، که به نمایش موقعیت اهداف روی نقشه‌ها کمک می‌کند.

فرمت نمونه:

1 2 3 4 5 6 7 8 9 10

| | | | | | | | | |

$--TLL,xx,llll.ll,a,yyyyy.yy,a,c--c,hhmmss.ss,a,a\*hh<CR><LF>

1) شماره هدف 00 – 99

2) عرض جغرافیایی هدف

3) شمال/جنوب (N/S)

4) طول جغرافیایی هدف

5) شرق/غرب (E/W)

6) نام هدف

7) زمان به UTC

8) وضعیت هدف

9) هدف مرجع

10) چک‌سام

نکته ۱) وضعیت هدف:

L = گم‌شده، هدف ردیابی‌شده گم شده است

Q = پرس‌وجو، هدف در حال پردازش برای شناسایی است

T = ردیابی، هدف در حال ردیابی است

نکته ۲) هدف مرجع:

اگر هدف به عنوان مرجع برای تعیین موقعیت یا سرعت کشتی خود استفاده شود، به "R" تنظیم می‌شود، در غیر این صورت خالی است.

**۳. TTM (پیام هدف ردیابی شده):**

جمله TTM اطلاعات دقیقی درباره یک هدف راداری ردیابی شده ارائه می‌دهد. این اطلاعات شامل برد، جهت، سرعت، مسیر و فاصله هدف، همچنین شماره هدف و پرچم وضعیت (tracking status) است که نشان می‌دهد آیا هدف به طور فعال در حال ردیابی است یا نه. این جمله برای سیستم‌های راداری که بخشی از یک سیستم یکپارچه ردیابی یا جلوگیری از برخورد هستند بسیار مهم است.

فرمت نمونه:

11 13

1 2 3 4 5 6 7 8 9 10| 12| 14

| | | | | | | | | | | | | |

$--TTM,xx,x.x,x.x,a,x.x,x.x,a,x.x,x.x,a,c--c,a,a\*hh <CR><LF>

1) شماره هدف

2) فاصله هدف

3) جهت از کشتی خودی

4) واحدهای جهت

5) سرعت هدف

6) زاویه مسیر هدف

7) واحدهای زاویه‌ی مسیر

8) فاصله نزدیک‌ترین نقطه تقرب (Closest point of approach)

9) زمان تا نزدیک‌ترین نقطه تقرب، "-" به معنی افزایش

10) واحدهای سرعت/فاصله، K/N/S

11) نام هدف

12) وضعیت هدف

13) هدف مرجع

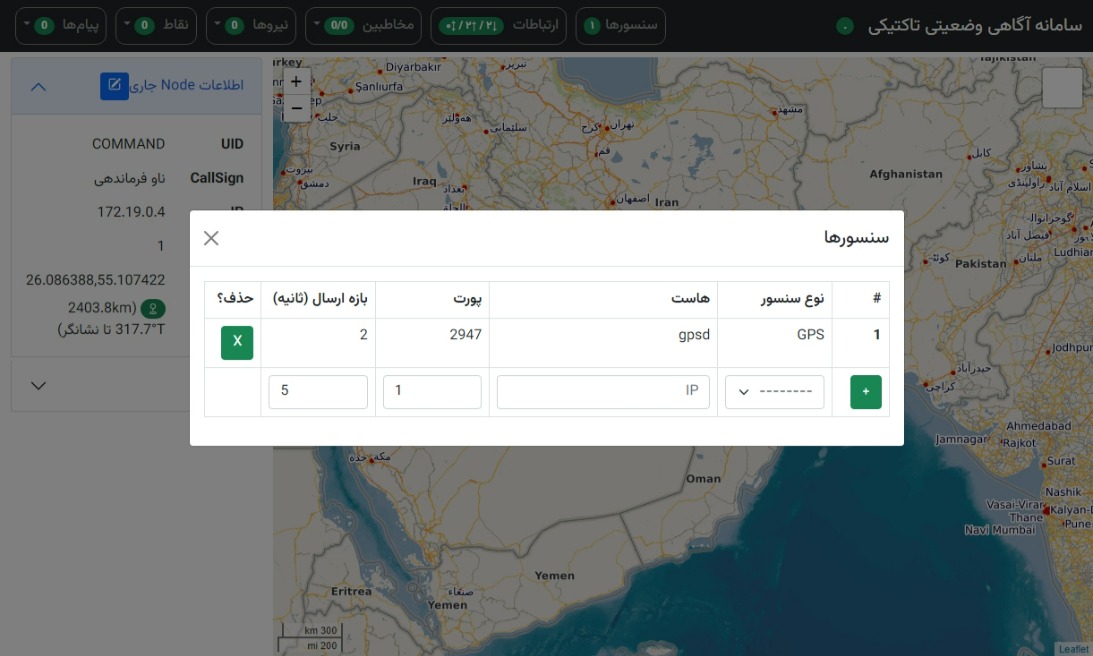
14) چک‌سام

اطلاعات دریافت شده از اهداف رادار، پس از پردازش، در قالب پیام‌های پروتکل CoT در می‌آیند. با توجه به این‌که برخلاف سنسورهای GPS و AIS اطلاعات دریافتی از سنسور مربوط به خود کاربر نمی‌شود و اشیای دیگری را توصیف می‌کند، لازم است که به ازای هر یک از اهداف شناسایی شده در رادار، یک پیام CoT ایجاد شود.

با ارسال این پیام‌ها به دیگر کاربران، آن‌ها می‌توانند اهداف را در نقشه‌ی سامانه‌ی خود ببینند و آگاهی وضعیتی بهتری برای اجرای ماموریت‌های خود داشته باشند.

##### ۵.۱.۴) سنسورها در رابط کاربری سامانه

برای ساده‌تر کردن ایجاد، مشاهده و حذف سنسورهای متصل به سامانه، در رابط کاربری بخشی در نظر گرفته و پیاده‌سازی شد که از طریق آن کاربران می‌توانند سنسورهای متصل به سامانه را مشاهده کنند و در صورت نیاز سنسور یا سنسورهایی به خود اضافه کنند.



شکل 5 - صفحه‌ی سنسورها در رابط کاربری

همان‌طور که در تصویر مشخص است، کاربر می‌تواند با کلیک روی دکمه‌ی سنسورها در رابط کاربری، صفحه‌ی سنسورها را باز کند. در این صفحه لیست سنسورهای متصل به سامانه آمده است.

با کلیک بر روی دکمه‌ی حذف هر یک از سنسورهای لیست، آن سنسور از سامانه حذف شده، و دیگر اطلاعات از آن سنسور جمع‌آوری نمی‌شود. همین‌طور اطلاعات آن سنسور در پیام‌های CoT سامانه نیز دیگر ارسال نخواهد شد.

از طریق همین صحفه، امکان ایجاد سنسور جدید نیز وجود دارد. برای تعریف سنسور جدید، لازم است کاربر نوع آن سنسور را مشخص کرده و اطلاعات مربوط به اتصال شبکه را نیز وارد کند. این اطلاعات عموما شامل IP سنسور و Portای است که اطلاعات از آن ارسال می‌شود. هم‌چنین برای قابل تنظیم بودن فرکانس ارسال اطلاعات هر سنسور، فیلدی اضافه شده است که کاربر با استفاده از آن می‌تواند تنظیم کند اطلاعات هر چند ثانیه ارسال شوند.

با افزودن سنسور، مشخصات سنسور جدید از رابط کاربری به سامانه منتقل شده و در سامانه، سنسوری با آن مشخصات تعریف می‌شود و به صورت خودکار شروع به کار می‌کند. با شروع به کار سنسور در سامانه، اطلاعات از IP و Port مشخص شده دریافت شده و در فواصل زمانی تنظیم شده برای دیگر کاربران ارسال خواهد شد.

#### ۵.۲) فیدهای ارتباطی[[1]](#footnote-1)

برای تنظیم‌پذیرتر کردن ارتباطات کاربران سامانه و ایجاد قابلیت ارتباط از طریق روش‌های مختلف ارتباطی، مفهوم فیدهای ارتباطی در سامانه تعریف شد. هر فید ارتباطی، مشخص کننده‌ی یک مسیر ارتباطی شبکه است که کاربر از طریق آن می‌تواند به ارسال و/یا دریافت اطلاعات به کاربران دیگر سامانه بپردازد. هم‌چنین با ایجاد قابلیت تعریف راحت این فید‌ها، کاربران مختلف می‌توانند ورودی‌ها و خروجی‌های مختلفی بر اساس نقش و عملکرد خود تعریف کنند. برای مثال کاربری که در نقش فرمانده است می‌تواند، فیدهای ارتباطی برای ایجاد ارتباط با فرماندگان دیگر تعریف کند.

##### ۵.۲.۱) انواع فید‌های ارتباطی

هر فید ارتباطی می‌تواند به سه حالت مختلف تعریف شود:

**۱. فید ورودی:** در این حالت، با توجه به مشخصات فید تعریف شده، کاربر پیام‌های CoT ارسالی از دیگر کاربران را از طریق روش ارتباطی تنظیم شده در فید، دریافت می‌کند.

**۲. فید خروجی:** در حالت خروجی، پیام‌های CoT از طریق آن فید دریافت نمی‌شوند و صرفا هر پیام ارسالی کاربر، با توجه به روش ارتباطی تنظیم شده و مشخصات فید روی شبکه ارسال می‌شود تا کاربران دیگری که فید ورودی با مشخصات مشابه را تعریف کرده‌اند، آن پیام‌ها را دریافت کنند.

**۳. فید دوطرفه:** در این حالت که ترکیبی از دو حالت قبلی‌است، هم پیام‌ها از فید دریافت می‌شوند، هم پیام‌های ارسالی کاربر به فید فرستاده می‌شود.

برای ایجاد امکان ارسال و دریافت پیام از روش‌های ارتباطی مختلف، هر فید می‌تواند روش ارتباطی خاص خود را داشته باشد. در حال حاضر دو روش ارتباطی برای فیدها در سامانه تعریف شده است.

**روش ارتباطی UDP:**

در این روش، در حالت خروجی، پیام‌های CoT به صورت خام (بدون پردازش اضافی) بر روی یک آدرس و پورت شبکه با پروتکل UDP نوشته می‌شوند. این روش، روش پیش‌فرض سامانه‌ی TAK است و بنابراین، تمامی نرم‌افزارهای مبتنی بر سامانه‌ی TAK با این روش سازگار هستند.

برای ایجاد چنین فیدی، لازم است که آدرس و پورتی که قرار است پیام‌ها به آن ارسال شوند، در سامانه تنظیم شوند. در حالت ورودی، سامانه روی پورت داده شده، به بسته‌های شبکه‌ی ارسالی گوش می‌کند و پیام‌های CoT ارسالی از سمت دیگر کاربران را دریافت می‌کند.

بدیهی است که برای ایجاد ارتباط بین دو کاربر، لازم است که کاربر فرستنده، یک فید خروجی با آدرس کاربر گیرنده داشته باشد و کاربر گیرنده نیز یک فید ورودی تعریف کرده باشد. هم‌چنین لازم است که پورت تنظیم شده در هر دو فید با یک‌دیگر برابر باشند.

با توجه به این‌که ممکن است لزوما کاربران آدرس IP یک‌دیگر را نداشته باشند و اهمیت بحث Broadcast پیام‌ها در زیرشبکه‌های کاربران، فیدهای خروجی کاربران می‌تواند روی آدرس Broadcast IP زیرشبکه تنظیم شوند، تا پیام‌های ارسالی آن‌ها به تمامی کاربران دیگر زیرشبکه ارسال شود.

در واقع، فید UDP حالت فراگیرتری از نحوه‌ی ارتباطی‌ای‌است که در پیش از این پیاده‌سازی شده بود و در گزارش پیشین توضیحات آن موجود بود. با تغییرات فعلی و تعریف مفهوم فید می‌توان کنترل بسیار بیشتری بر روی ارتباط کاربران با هم و نحوه‌ی ارسال پیام‌ها بین آن‌ها داشت.

**روش ارتباطی RabbitMQ:**

روش ارتباطی دیگری که در سامانه طراحی و پیاده‌سازی شده است، ایجاد ارتباط از طریق ابزار RabbitMQ است. RabbitMQ سرویسی است که به برنامه‌های مختلف این امکان را می‌دهد که از طریق ارسال پیام‌ها به یه سری صف‌ با یک‌دیگر ارتباط برقرار کنند. این سیستم اطمینان حاصل می‌کند که پیام‌ها حتی در صورتی که برخی از قسمت‌های سیستم مشغول یا کند باشند، تحویل داده شده و پردازش می‌شوند. این کار به برنامه‌ها کمک می‌کند که به صورت کارآمدتر عمل کنند و وظایف بیشتری را از طریق پردازش پیام‌ها در پس‌زمینه انجام دهند. RabbitMQ معمولاً برای مدیریت حجم کار و بهبود عملکرد در سیستم‌های توزیع‌شده استفاده می‌شود.

اهمیت این روش ارتباطی از این لحاظ است که با استفاده از آن می‌توان بخش ارتباطی سامانه را به نرم‌افزار دیگری سپرد، تا به هر نحوی که لازم است ارتباط را برقرار کند و پیام‌های CoT را بین کاربران جابجا کند. به عنوان مثال، در حالتی که کاربر سامانه مستقیما با دیگر کاربران امکان ارتباط شبکه را ندارد و لازم است که ارتباطات از طریق رادیو انجام شود، لازم است که یک نرم‌افزار دیگر به عنوان Data Link پیام‌ها را دریافت کرده، به صورت موج‌های رادیویی درآورده و از طریق رادیو به گیرندگان ارسال کند. برای یک‌پارچه کردن سامانه‌ی آگاهی وضعیتی با Data Link مورد نظر، پیام‌های ارسالی سامانه‌ی آگاهی وضعیتی در صفی که Data Link از آن می‌خواند، فرستاده می‌شوند تا Data Link به ترتیب آن‌ها را از طریق رادیو ارسال نماید. پیام‌های دریافتی رادیویی نیز، پس از آن‌که در Data Link دریافت شد، در یک صف دیگر قرار می‌گیرد. سامانه‌ی آگاهی وضعیتی با خواندن پیام‌ها آن صف، به پیام‌های ارسالی دیگر کاربران از طریق رادیو دست پیدا می‌کند.

##### ۵.۲.۲) تنظیم فیدهای ارتباطی

برای تنظیم ساده‌ی فیدهای ارتباطی در سامانه، دو روش مختلف در نظر گرفته شده است:

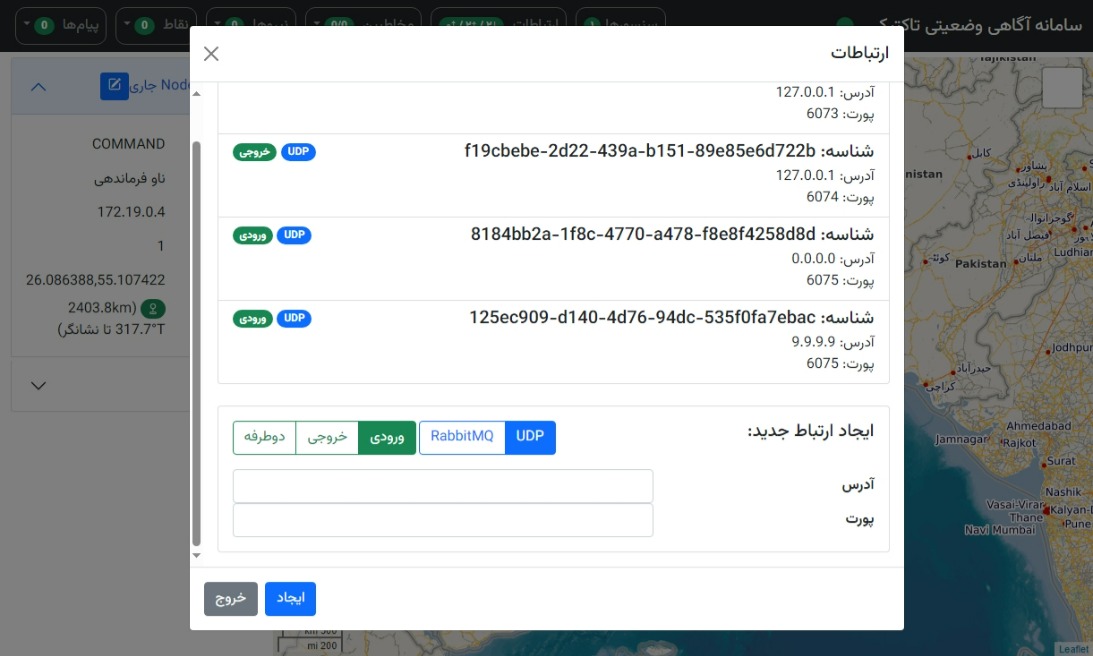
**۱. تنظیم فیدها از طریق فایل‌های Config**

در این روش، اطلاعات مربوط به فیدهای ارتباطی، از فایل‌های Config موجود در پروژه خوانده می‌شوند. در این فایل‌ها که با فرمت YAML نوشته شده‌اند، می‌توان تنظیمات مربوط به انواع مختلف فیدها را وارد کرد، تا در زمان اجرای پروژه، فیدها از پیش تعریف شده باشند و با اجرای سامانه شروع به ارسال و دریافت پیام کنند.

این حالت تنظیمات، برای زمانی مناسب است که قرار است سامانه در اختیار نیروها قرار بگیرد و لازم است که تنظیمات از پیش انجام شود تا نیازی نباشد که کاربران سامانه، یکی یکی تنظیمات فیدهای مختلف را وارد کنند. احتمال خطاهای انسانی در تنظیم ارتباطات در این حالت کمتر است.

**۲. تنظیم فیدها از طریق رابط کاربری:**

در این حالت، کاربران می‌توانند از طریق رابط کاربری سامانه و با کلیک بر روی دکمه‌ی ارتباطات، وارد صفحه‌ی فیدها شوند که در آن لیست فیدهای تنظیم شده قابل مشاهده است.



شکل 6 - صفحه‌ی ارتباطات در رابط کاربری

از طریق این صفحه که در شکل نشان داده شده است، مشابه صفحه‌ی سنسورها، کاربر می‌تواند اقدام به حذف یا افزودن یک فید جدید ارتباطی کند. برای افزودن فید جدید، لازم است که ورودی، خروجی یا دوطرفه بودن فید، و تنظیمات شبکه فید وارد شود که تنظیمات شبکه، بر اساس روش ارتباطی فید می‌تواند متفاوت باشد. در روش ارتباطی UDP برای تنظیمات شبکه، فیلدهای IP و Port لازم هستند که آدرس شبکه‌ای را مشخص می‌کنند که سامانه به آن پیام‌ها را ارسال می‌کند یا با گوش کردن به آن، پیام‌ها را دریافت می‌کند.

در روش ارتباطی RabbitMQ، کاربر باید آدرس RabbitMQ را مشخص کند و هم‌چنین نام صف‌های ورودی و خروجی در RabbitMQ را نیز وارد کند که پیام‌ها از طریق آن‌ها دریافت و ارسال می‌شوند.

#### ۵.۳) تغییرات رابط کاربری

در راستای کاربرپسند کردن رابط کاربری سامانه و ساده‌تر کردن کار کردن با آن، تغییرات عمده‌ای بر روی رابط کاربری ایجاد شد. در گام نخست، برای منظم‌تر کردن ساختار داخلی کدهای رابط کاربری، از Componentهای فریمورک Vue استفاده شد تا بخش‌های مختلف کد رابط کاربری در فایل‌های جدا از هم قرار بگیرند تا فرایند توسعه‌ی آن سریع‌تر و کم‌خطاتر شود.

در گامی دیگر، فونت فارسی مناسبی برای رابط کاربری انتخاب شده و در پروژه گنجانده شد. هم‌چنین تمامی کدهای رابط کاربری برای تطابق با زبان فارسی و راست‌چین بودن رابط کاربری، بررسی و ویرایش شد، تا برای کاربران تجربه‌ی کاربری بهتری خلق شود.

در پیاده‌سازی ویژگی‌های جدید سامانه، به این توجه شده است، که برای ایجاد و تغییر تنظیمات مختلف، کاربر بتواند بدون دانش فنی و صرفا از طریق رابط کاربری سامانه، تغییرات مورد نیاز خود انجام دهد. هم‌چنین ويژگی‌هایی که پیش از این پیاده شده بودند، مانند اطلاعات خود کاربر، در رابط کاربری سامانه، قابل تنظیم و تغییر شدند.

هم‌چنین برای مشخص‌تر شدن عناصر موجود بر روی نقشه (نیروهای خودی و نقاط گذاشته شده روی نقشه) علاوه بر Icon آن‌ها، بخشی از اطلاعات آن‌ها نیز بر روی نقشه نمایش داده می‌شود که به کاربر این امکان را می‌دهد که بدون کلیک کردن روی نقاط و نیروها، آن‌ها را بشناسد و با یک نگاه بتواند آگاهی وضعیتی بهتری نسبت به محیط عملیاتی خود داشته باشد.

### ۶) نتایج کسب شده تا گزارش سوم (گزارش کنونی)

در ادامه‌ی تلاش‌ها برای توسعه‌ی «سامانه‌ی آگاهی وضعیتی تاکتیکی» و با توجه به بازخوردهای دریافتی و نیازمندی‌های شناسایی‌شده در مراحل پیشین، پیشرفت‌های قابل توجهی در بخش‌های مختلف سامانه حاصل شده است. این پیشرفت‌ها عمدتاً بر افزایش قابلیت‌های عملیاتی، بهبود تجربه‌ی کاربری و ارتقای زیرساخت فنی سامانه متمرکز بوده‌اند. در این فصل، به تشریح دستاوردهای کلیدی پس از ارائه‌ی گزارش دوم پرداخته می‌شود.

#### ۶.۱) توسعه قابلیت‌های عملیاتی کلیدی

به منظور افزایش کارایی سامانه در سناریوهای عملیاتی واقعی، چندین قابلیت جدید و حیاتی به سامانه افزوده شده است:

##### ۶.۱.۱) پیاده‌سازی امکان مشخص کردن نواحی بر روی نقشه

با پیاده‌سازی این امکان، کاربران می‌توانند با انتخاب ابزار چندضلعی و کلیک کردن روی نقشه، یک ناحیه تعریف کنند. با مشخص شدن نواحی این ناحیه، یک موجودیت از جنس ناحیه ساخته می‌شود که علاوه بر نشان داده شدن بر روی نقشه‌ی خود کاربر، قابلیت ارسال آن به دیگر کاربران نیز وجود دارد. پس از تکمیل ناحیه توسط کاربر، با باز شدن یک Modal امکان تنظیم مشخصات این ناحیه، مانند اسم و رنگ آن نیز وجود دارد.

##### ۶.۱.۲) سیستم حصار جغرافیایی (Geofencing) و هشدار

یکی از مهم‌ترین افزوده‌ها، قابلیت تعریف حصارهای جغرافیایی (Geofence) بر روی نقشه است. کاربران اکنون می‌توانند پس از مشخص کردن محدوده‌های جغرافیایی به صورت چندضلعی، که پیش از توصیف شد، برای این نواحی، حصار جغرافیایی نیز تعریف کنند. این قابلیت با سیستم هشداردهی سامانه یکپارچه شده است. در صورتی که یک نیروی خودی یا شناسایی‌شده وارد یک حصار تعریف‌شده به عنوان منطقه‌ی خطرناک شود (بسته به تنظیمات حصار)، هشداری به تمامی کاربران ارسال می‌شود. این ویژگی به فرماندهان امکان می‌دهد تا نظارت بهتری بر تردد نیروها در مناطق حساس داشته باشند و از ورود یا خروج ناخواسته به محدوده‌های خاص مطلع شوند.

همچنین، کاربران می‌توانند حصارهایی را پیرامون موقعیت فعلی خود ایجاد کنند تا در صورت نزدیک شدن سایر نیروها (اعم از خودی یا دشمن شناسایی‌شده) به این محدوده، هشدار دریافت کنند.

##### ۶.۱.۳) سیستم اعلام و لغو وضعیت اضطراری

قابلیت اعلام وضعیت اضطراری به شکلی کارآمد پیاده‌سازی شده است تا کاربران بتوانند در صورت قرار گرفتن در معرض خطر با استفاده از یک دکمه‌ی مشخص در رابط کاربری، وضعیت اضطراری خود را به سایر کاربران یا مرکز فرماندهی اعلام کنند. انواع مختلف وضعیت اضطراری (مانند نیاز به پشتیبانی، وضعیت پزشکی و...) قابل انتخاب است و آیکون‌های متفاوتی برای هر نوع در نظر گرفته شده تا تفکیک بصری آن‌ها بر روی نقشه آسان‌تر باشد.

علاوه بر اعلام وضعیت اضطراری، مکانیزم لغو هشدار اضطراری نیز پیاده‌سازی شده است. کاربری که وضعیت اضطراری اعلام کرده یا مرکز فرماندهی، می‌تواند پس از رفع شرایط اضطراری، هشدار مربوطه را لغو کند تا از نمایش اطلاعات نادرست بر روی نقشه‌ی سایر کاربران جلوگیری شود. این فرآیند لغو نیز به صورت آنی به سایر کاربران اطلاع‌رسانی می‌شود.

##### ۱.۳. بهبود مدیریت سنسورها و فیدهای اطلاعاتی

با توجه به اهمیت دریافت اطلاعات از منابع مختلف، بخش مدیریت سنسورها و فیدهای اطلاعاتی توسعه یافته است. رابط کاربری مربوط به تنظیمات سنسورها بهبود یافته و امکان فعال یا غیرفعال کردن هر فید اطلاعاتی به صورت مجزا فراهم شده است. این امکان به کاربر اجازه می‌دهد تا بر اساس نیاز عملیاتی و شرایط شبکه، منابع اطلاعاتی مورد استفاده‌ی خود را مدیریت کند. همچنین، زیرساخت لازم برای اتصال به سنسورهای جدید و پردازش پروتکل‌های متنوع‌تر (مانند داده‌های GPS از طریق TCP) فراهم شده و تست‌های اولیه بر روی آن صورت گرفته است.

#### ۲. بهبودهای رابط کاربری (UI/UX)

بهبود تجربه‌ی کاربری و افزایش سرعت دسترسی به اطلاعات و ابزارها، یکی از محورهای اصلی توسعه در این مرحله بوده است. تغییرات زیر در این راستا انجام شده‌اند:

##### ۲.۱. بازطراحی نسبی رابط کاربری با الهام از سامانه‌های استاندارد

با بررسی سامانه‌های مشابه و استاندارد در حوزه‌ی آگاهی وضعیتی تاکتیکی (مانند WinTAK)، تلاش شد تا چیدمان عناصر و دسترسی به ابزارها در رابط کاربری سامانه بهینه‌تر شود. دکمه‌های مربوط به ابزارهای اصلی (مانند ایجاد نقطه، تعریف حصار، اعلام اضطراری) از منوهای جانبی به صورت دکمه‌های شناور بر روی نقشه منتقل شده‌اند تا دسترسی به آن‌ها سریع‌تر و آسان‌تر باشد.

##### ۲.۲. بهبود نمایش اطلاعات بر روی نقشه

اطلاعات نمایش داده شده برای هر عنصر بر روی نقشه (مانند نیروهای خودی، نقاط تاکتیکی، حصارها) غنی‌تر شده است. علاوه بر آیکون، نام یا شناسه (Callsign) و برخی اطلاعات کلیدی دیگر به صورت مستقیم بر روی نقشه نمایش داده می‌شود تا کاربر با یک نگاه، اطلاعات بیشتری کسب کند. همچنین، آیکون‌های مورد استفاده برای نمایش انواع مختلف نیروها و وضعیت‌های اضطراری بازبینی و بهبود یافته‌اند تا خوانایی و تفکیک‌پذیری بهتری داشته باشند.

##### ۲.۳. افزودن منوی عملیاتی برای عناصر نقشه

برای هر عنصر قابل تعامل بر روی نقشه (مانند آیکون نیروهای خودی)، یک منوی بازشونده (Context Menu) در نظر گرفته شده است. با کلیک راست یا لمس طولانی بر روی این عناصر، منویی شامل عملیات مرتبط با آن عنصر (مانند حذف، ویرایش، ارسال پیام مستقیم، مشاهده‌ی جزئیات بیشتر) ظاهر می‌شود. این ویژگی سرعت انجام عملیات متداول بر روی عناصر نقشه را به طور چشمگیری افزایش می‌دهد.

##### ۲.۴. بهبود مودال‌ها و فرم‌ها

پنجره‌های مودال (Modal) که برای نمایش هشدارها، فرم‌های ورود اطلاعات (مانند تنظیمات کاربر، اطلاعات سنسور، جزئیات حصار) استفاده می‌شوند، بازبینی شده‌اند. فرآیند پاک‌سازی و بستن این پنجره‌ها بهبود یافته و نمایش اطلاعات در آن‌ها ساختاریافته‌تر شده است. به عنوان مثال، مودال نمایش هشدارها اکنون لیستی از هشدارهای فعال را به صورت خوانا نمایش می‌دهد.

#### ۳. ارتقای زیرساخت و بهینه‌سازی

در کنار افزودن قابلیت‌های جدید، اقداماتی نیز در جهت پایدارسازی و بهینه‌سازی عملکرد سامانه انجام شده است:

##### ۳.۱. بهینه‌سازی فرآیندهای Build و Deployment

فایل‌ها و تنظیمات مربوط به ساخت (Build) و استقرار (Deploy) سامانه با استفاده از Docker به‌روزرسانی شده‌اند. Image های پایه تغییر یافته و پرچم‌های (Flags) مربوط به کامپایل بهینه‌سازی شده‌اند تا حجم نهایی نرم‌افزار کاهش یافته و عملکرد آن بهبود یابد. فایل‌های غیرضروری از فرآیند Build مستثنی شده‌اند.

##### ۳.۲. بهبود پردازش و همگام‌سازی داده‌ها

منطق پردازش داده‌های دریافتی از سایر کاربران و منابع اطلاعاتی در سمت کلاینت بازبینی شده است. نحوه‌ی شناسایی و اعمال تغییرات (افزودن، به‌روزرسانی، حذف عناصر) بر روی نقشه بهینه‌تر شده و موارد خاصی مانند لغو هشدارها به درستی مدیریت می‌شوند تا از نمایش اطلاعات متناقض جلوگیری شود.

##### ۳.۳. رفع خطاها و افزایش پایداری

بر اساس تست‌های انجام شده و گزارش‌های دریافتی، تعدادی از خطاها (Bugs) شناسایی و رفع شده‌اند. این موارد شامل خطاهای مربوط به فرم‌ها، حذف نشدن صحیح برخی عناصر، مشکلات مربوط به ارتباطات تحت شبکه (به خصوص UDP) و ناپایداری‌های جزئی در رابط کاربری بوده است.

### ۶) نتیجه گیری

کارهای انجام شده از گزارش قبلی پروژه، شامل موضوعات عمده‌ی ۱) نواحی و مسیرها، ۲) حصارهای جغرافیایی ۳) ارسال هشدارها و ۴) بهینه‌سازی رابط کاربری پروژه است. علاوه بر این، پروژه در محیط واقعی مورد تست و بررسی قرار گرفته است و مشکلات و باگ‌های متعددی در این تست‌ها شناسایی و حل شده، که این سامانه را برای استفاده در محیط عملیاتی آماده‌تر کرده است.

مطابق با برنامه‌ی ارائه شده در پروپوزال، گام ۱ پروژه (جمع‌آوری کامل نیازمندی‌ها و طراحی سیستم) و گام ۲ (پیاده‌سازی رابط کاربری بر مبنای نقشه و نشان دادن نقاط مهم) به صورت کامل انجام شده و گام ۳ (پیاده‌سازی ارتباط بین کاربران و ذخیره‌سازی پیام‌ها) نیز پیشرفت قابل توجهی داشته است. هم‌چنین با توجه به مشاوره‌های ناظر محترم و نماینده‌ی گرامی سازمان، گام ۴ (تست، عیب‌یابی و رفع مشکلات) نیز آغاز شده و در تست‌های انجام شده، علاوه بر عیب‌یابی و رفع مشکلات، بخش مهمی از نیازهای عملی سازمان برای استفاده از سامانه نیز مشخص و پیاده‌سازی آن‌ها به طور کامل انجام شده است. گام ۵ (تهیه مستندات فنی پروژه) نیز آغاز شده و پیشرفت زیادی داشته است. گزارش فعلی، بخشی از این مستندات را تشکیل داده و در کدهای پروژه نیز به مسئله‌ی مستندسازی توجه شده است.

1. Communication Feeds [↑](#footnote-ref-1)