

Система підтримки прийняття рішень (СППР) для командира підрозділу

1. Про предметну область

- **Проект розроблено** для підвищення ефективності тактичного планування, швидкості реакції на зміни бойової обстановки та мінімізації людського фактора при прийнятті критичних рішень.
- **Система базується на** алгоритмах аналізу даних, архітектурі мікросервісів та забезпечує агрегацію даних з різних джерел (розвідка, БПЛА, суміжні підрозділи) в єдиний інформаційний простір.
- **Мета:** Моніторинг тактичної ситуації, оцінка наявних ресурсів, прогнозування дій противника та автоматичне формування оптимальних варіантів дій (рекомендацій) у реальному часі.
- **Технологічний стек:** Java (фреймворк Spring Boot для REST API та мікросервісів), React / TypeScript (для інтерфейсу командира), Deeplearning4j (для інтелектуального аналізу даних та прогнозування на Java), PostgreSQL з розширенням PostGIS (для роботи з просторовими даними на мапі), Apache Kafka (поточкова передача даних), WebSocket (миттєві сповіщення), Secure VPN (захист каналу).

2. Опис діаграм (UML)

I. Use Case Diagram (Діаграма варіантів використання) Відображає функціональні межі системи та рольову модель доступу.

- **Актори:**
 - *Командир:* Відповідає за затвердження планів операцій, запит аналітики та прийняття фінальних рішень.
 - *Офіцер-аналітик (Розвідка):* Вносить перевірені розвідувальні дані, коригує параметри системи.
 - *Автоматизований модуль аналітики:* Автоматичний актор, що постачає прогнози, оцінює ризики та генерує варіанти рішень.
- **Ключові особливості:**
 - Використано **Generalization** для спільної авторизації (Командир та Аналітик як користувачі системи).
 - Зв'язок **<<include>>** показує залежність прецеденту «Генерація плану дій» від обов'язкового «Аналізу тактичної ситуації».
 - Зв'язок **<<extend>>** виділяє «Запит на додаткову розвідку (БПЛА)» як розширення стандартного прецеденту «Оцінка сил противника» (активується лише за умови недостатності даних).

II. Sequence Diagram (Діаграма послідовності) Деталізує сценарій: «Реагування на раптову зміну обстановки (Виявлення наступу противника)».

- **Потік даних:** Демонструє шлях від отримання критичного сповіщення від систем спостереження до генерації варіантів оборони та виведення їх на екран командира.
- **Технічні деталі:**
 - Використання асинхронних запитів до Java-бекенду, щоб не блокувати клієнтський інтерфейс.
 - Паралельне виконання (**par**): запис інциденту в БД, оновлення тактичної мапи та миттєва розсилка попередження суміжним підрозділам.
 - Механізм **АСК (Acknowledge)**: гарантоване підтвердження отримання нових наказів підлеглими групами.

III. Activity Diagram (Діаграма діяльності) Описує алгоритм бізнес-процесу: «Планування тактичної операції».

- **Логіка:** Система автоматично оцінює життєздатність плану на основі наявних ресурсів (б/к, особовий склад, пальне) та даних про противника.
- **Критичні вузли (Decisions):**
 - *Достатність ресурсів:* Якщо ні — блокування затвердження плану та автоматичне формування запиту на логістичне забезпечення (Fail-safe).
 - *Рівень достовірності розвідданих:* Адаптивна поведінка (якщо достовірність висока -> перехід до генерації рішень; якщо низька -> автоматичний запит на відправку дрона на дорозвідку перед плануванням).

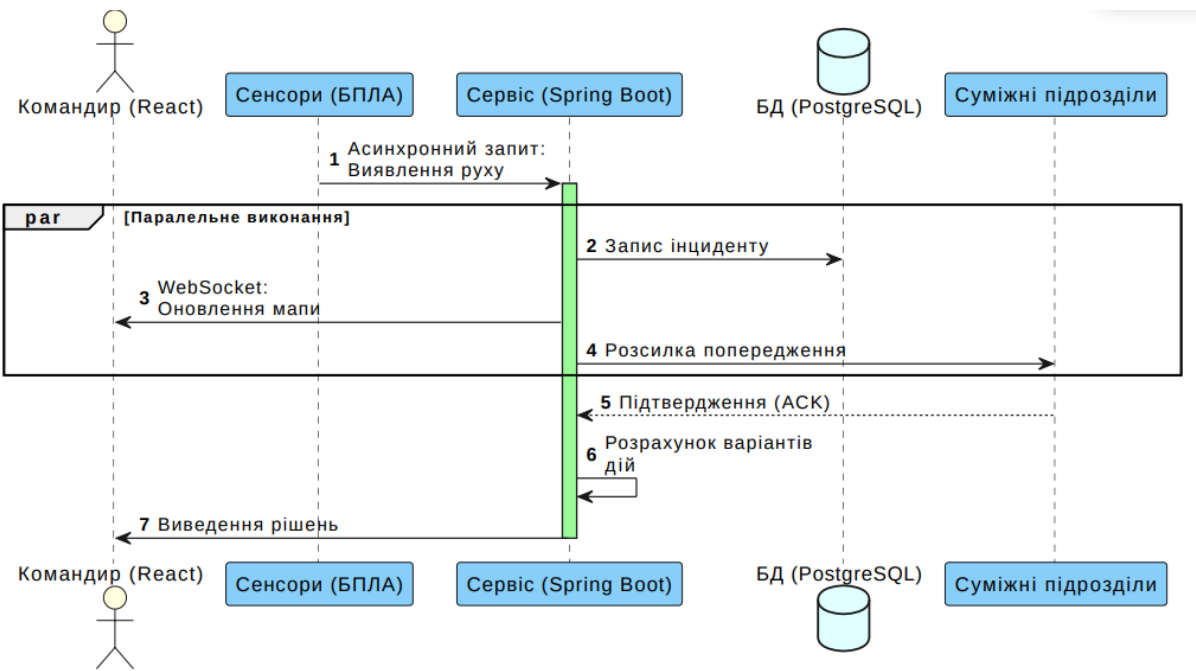
1. Use Case Diagram (Діаграма варіантів використання)

Ця діаграма відображає ролі та основні функції системи з урахуванням нового стеку.



2. Sequence Diagram (Діаграма послідовності)

Тут відображено взаємодію між клієнтом (React) та бекендом (Spring Boot) за допомогою Kafka, WebSocket та REST API.



3. Activity Diagram (Діаграма діяльності)

Алгоритм планування, який враховує логіку перевірки ресурсів та адаптивну поведінку.

