**АНОТАЦІЯ**

кваліфікаційної роботи магістра на тему:

“Підсистема інтерактивного рекогностування і планування розгортання військ на основі доповненої реальності”

Кваліфікаційна робота магістра містить: 81 сторінки, 45 рисунки, 2 таблиці, 11 джерел, 1 додаток.

Робота присвячена розробці та аналізу підсистеми інтерактивного рекогностування та планування розгортання військ з використанням технологій доповненої реальності. Проведено детальний огляд сучасних методів рекогностування, їхніх переваг та недоліків, а також аналіз існуючих систем інтерактивного рекогностування. В роботі розглянуто ключові вимоги до системи та використані технології для її розробки. Практична цінність роботи полягає у потенціалі застосування розробленої підсистеми для підвищення ефективності та точності військових операцій. Розроблена підсистема може бути використана як у практичних, так і в теоретичних дослідженнях з даної тематики.

Головні цілі дослідження включають в себе розробку ефективного інтерфейсу для взаємодії користувачів з *AR*-середовищем, розробку алгоритмів автоматизованого розпізнавання об'єктів та ресурсів, а також інтеграцію з існуючими системами рекогностування та планування військових операцій.

**Ключові слова:** *доповнена реальність, військове планування, інтерактивне рекогностування, візуалізація даних.*

***ANNOTATION***

*of the master's qualification paper on theme:*

*“A subsystem for interactive reconnaissance and planning of troop deployment based on augmented reality”*

*The bachelor’s qualification paper comprises 81 pages, 45 pictures, 2 tables, 11 sources, 1 appendienc.*

*The work is devoted to the development and analysis of a subsystem for interactive reconnaissance and planning of troop deployment using augmented reality technologies. A detailed review of modern reconnaissance methods, their advantages and disadvantages, as well as an analysis of existing interactive reconnaissance systems is carried out. The key requirements for the system and the technologies used for its development are considered. The practical value of the work lies in the potential of using the developed subsystem to increase the efficiency and accuracy of military operations. The developed subsystem can be used in both practical and theoretical research on this topic.*

*The main objectives of the research include the development of an effective interface for user interaction with the AR environment, the development of algorithms for automated recognition of objects and resources, as well as integration with existing systems for reconnaissance and planning of military operations.*

***Keywords****: augmented reality, military planning, interactive reconnaissance, data visualization.*

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ 9](#_Toc157137154)

[РОЗДІЛ 1 АНАЛІЗ ПРОЦЕСУ РЕКОГНОСТУВАННЯ ВІЙСЬКАМИ МІСЦЕВОСТІ ДЛЯ ПОДАЛЬШОГО РОЗГОРТАННЯ ВІЙСЬК 12](#_Toc157137155)

[1.1. Огляд сучасних методів рекогностування 12](#_Toc157137156)

[1.2. Недоліки існуючих підходів 20](#_Toc157137157)

[1.2.1. Недоліки використання БПЛА 20](#_Toc157137158)

[1.2.2. Недоліки використання супутникового спостереження 21](#_Toc157137159)

[1.2.3. Недоліки використання традиційних методів рекогностування 22](#_Toc157137160)

[1.2.4. Недоліки використання традиційних методів рекогностування 22](#_Toc157137161)

[1.2. Визначення ключових вимог до системи 23](#_Toc157137162)

[Висновки до розділу 1 25](#_Toc157137163)

[РОЗДІЛ 2 РОЗРОБКА ПІДСИСТЕМИ ІНТЕРАКТИВНОГО РЕКОГНОСТУВАННЯ 27](#_Toc157137164)

[2.1. Проектування архітектури системи та опис її структури 27](#_Toc157137165)

[2.2. Розробка та опис ключових алгоритмів системи 44](#_Toc157137166)

[2.2.1. Розробка серверної частини 45](#_Toc157137167)

[2.2.2. Розробка частини клієнта 54](#_Toc157137168)

[2.3. Вибір апаратного забезпечення для використання системи 60](#_Toc157137169)

[Висновки до розділу 2 61](#_Toc157137170)

[РОЗДІЛ 3 ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗАЦІЇ РЕКОГНОСТУВАННЯ 64](#_Toc157137171)

[3.1. Алгоритм функціонування системи 64](#_Toc157137172)

[3.2 . Оцінка потреби у подальших дослідженнях 75](#_Toc157137173)

[Висновки до розділу 3 77](#_Toc157137174)

[ВИСНОВКИ 78](#_Toc157137175)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 81](#_Toc157137176)

# **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ**

*AR* – *augmented* *reality*

БПЛА – безпілотний літальний апарат

*UI* – *user* *interface*

*API* – *application* *programming* *interface* (прикладний програмний інтерфейс)

*JSON* – *JavaScript* *Object* *Notation*

*JWT* – *JSON* *Web* *Token*

*DOM* – *Document* *Object* *Model*

БД – база даних

СУБД – систе́ма управління ба́зами да́них

*JPA* – *Java* *Persistence* *API*

*JPQL* – (*Java Persistence Query Language*)

*XML* – *eXtensible* *Markup* *Language*

*SQL* – *Structured* *Query* *Language*

ІС – інформаційна система

ДР – доповнена реальність

ІТ – інформаційні технології

*ID* – *identifier* (ідентифікатор)

ПЗ – програмне забезпечення

**ВСТУП**

**Актуальність роботи.** На сьогоднішній день рекогностування і планування розгортання військ виявляється надзвичайно актуальною і важливою. Сучасні конфлікти та воєнні дії вимагають від військових стратегій та командування надзвичайно точних та оперативних рішень, щоб забезпечити ефективну реакцію на загрози. Доповнена реальність, завдяки своїм інтерактивним можливостям, може значно підвищити рівень інформаційної обробки та допомогти командам у прийнятті обґрунтованих рішень.

Додатково, розвиток *AR*-технологій відкриває нові можливості для тренування та підготовки військового персоналу. Спрощене візуалізоване моделювання реального місцевості, включаючи всякі складні умови і терейни, може допомогти покращити навички та підготовку військових до дій в реальних умовах.

Завершуючи, дана дипломна робота є важливим внеском у розвиток сучасних технологій для військового застосування та може значно підвищити ефективність військових операцій, зменшити ризик та сприяти збереженню життів військових.

**Мета дослідження:** розробка підсистеми інтерактивного рекогностування і планування розгортання військ на основі доповненої реальності:

* провести огляд сучасних досягнень та існуючих *AR*-систем для військового застосування та визначити їх переваги та недоліки.;
* розробити програмне та апаратне забезпечення для підсистеми інтерактивного рекогностування, що буде включати в себе відображення геопросторових даних та інформаційних шарів на основі доповненої реальності;
* провести серію тестів та експериментів для оцінки функціональності та ефективності розробленої *AR*-підсистеми в реальних та симульованих військових сценаріях;
* вивчити можливості і виконати інтеграцію розробленої *AR*-підсистеми з існуючими системами контролю та управління доступом, що використовуються в армії;
* оцінити отримані результати, визначити переваги та обмеження *AR*-підсистеми та сформулювати рекомендації щодо її подальшого використання та розвитку в контексті військового рекогностування та планування розгортання військ.

**Об’єкт дослідження:** рекогностування на місцевості та планування розгортання військ.

**Предмет дослідження:** технологічний компонент для застосування в контексті військового рекогностування та планування розгортання військ з використанням доповненої реальності.

Для реалізації поставленої мети сформульовані **наступні завдання**:

* огляд сучасних інформаційних технологій, необхідних для створення і використання доповненої реальності
* аналіз процесу рекогностування місцевості для вибору позиції для розгортання військ;
* запропонування узагальненої структурної схеми та алгоритму роботи модуля для автоматизації процесу проведення рекогностування для вибору позиції розгортання військ;
* опис особливостей програмної реалізації модуля для автоматизації процесу проведення рекогностування;
* інтегрувати геопросторові дані та *AR*-технології для точного відображення місцевості та об'єктів в реальному часі.

РОЗДІЛ 1 АНАЛІЗ ПРОЦЕСУ РЕКОГНОСТУВАННЯ ВІЙСЬКАМИ МІСЦЕВОСТІ ДЛЯ ПОДАЛЬШОГО РОЗГОРТАННЯ ВІЙСЬК

## Огляд сучасних методів рекогностування

Будь-яка місцевість може сприяти виконанню бойового завдання або ж значно утруднювати його. Для того, щоб використати сприятливі властивості місцевості, її потрібно найбільш повно вивчити, правильно оцінити та уміло використати в конкретних бойових умовах. Тому військові статути і настанови розглядають місцевість як один із найважливіших елементів бойової обстановки.

Одним із способів вивчення місцевості, що може надати найдокладніші та найдостовірніші дані, є розвідка місцевості - збирання та систематизація даних про місцевість, окремі її елементи та об'єкти.

Розвідку місцевості безперервно та цілеспрямовано організують штаби та командири, вона є складовою тактичної розвідки - важливої частини бойового забезпечення.

Основним завданням розвідки місцевості є з'ясування сучасного стану основних елементів місцевості: рельєфу, населених пунктів, водних перешкод, дорожньої мережі, рослинності та ґрунтів, якісних та кількісних характеристик окремих об'єктів: форм рельєфу, будівель та промислових об'єктів, водойм, гідротехнічних споруд, доріг, дорожніх споруд, ділянок рослинності, боліт тощо - з метою визна­чення тактичних властивостей місцевості та прийняття обґрунтованого рішення на виконання бойового завдання.

Командири підрозділів вивчають і оцінюють місцевість, як правило, за топографічними картами з порівняно великою докладністю. Але при цьому задовольнити вимоги військ лише топографічними картами дуже важко, оскільки на карті неможливо показати, наприклад, кліматичні умови і пов'язані з цим умови прохідності місцевості за різної пори року, повені та паводки на ріках, період їх замерзання, товщину льоду тощо. Крім того, карти з часом старіють і зміст карти може не повністю відповідати дійсному стану місцевості..

Рекогностування - це систематичний процес збору, обробки, аналізу та використання інформації про місцевість та ворога, його розташування, збройні сили, ресурси та інші стратегічно важливі аспекти. Воно може здійснюватися за допомогою різних методів: візуально, за допомогою електронних засобів, агентурної розвідки, космічних засобів тощо.

Рекогностування проводить особисто командир батальйону (роти) із залученням своїх заступників, командирів штатних, доданих і підтримуючих (взаємодіючих) підрозділів. Для проведення рекогностування можуть використовувати ксП (сП) підрозділів, що діють на даному напрямку попереду.[1]

Командир батальйону (роти) повинен знаходити можливість для проведення рекогностування, уточнення завдань підрозділам і організації взаємодії на місцевості, особливо на напрямку зосередження основних зусиль. Якщо умови обстановки дозволяють, рекогностування може проводитись до затвердження замислу старшим командиром.

рекогностування проводиться у світлий час доби з наступним уточненням необхідних питань вночі. Під час її проведення командир батальйону (роти), крім звичайних питань, уточнює: видимі вночі орієнтири, місця постановки світлових орієнтирів і напрямки світлових створів; цілі (об’єкти) противника, які підлягають освітленню (засліпленню).

При організації оборони в умовах відсутності зіткнення з противником, командир батальйону (роти) визначає замисел і завершує формулювання рішення по карті, доводить його до своїх заступників і командирів підрозділів; проводить рекогносцировку, в ході якої уточнює на місцевості своє рішення; віддає бойовий наказ, організовує взаємодію, управління і всебічне забезпечення бою. У призначений час виводить батальйон (роту) у район оборони (опорний пункт), організовує його зайняття, створює бойовий порядок, систему вогню та систему інженерних загороджень.

У сучасних військових операціях рекогностування має значний вплив на здобуття переваги над противником. Спостереження за військовими конфліктами показують, що перевага в інформаційній сфері, зокрема здобута за допомогою рекогностування інформація впливає на кінцевий результат зіткнення на різних рівнях військових операцій. Крім того, воєнне рекогностування має велике значення в процесі підготовки потенційних сценаріїв протидії новим, маловідомим воєнним загрозам. Цей аспект воєнної розвідки формує її результати в імпульс для проведення стратегічних розвідувальних дій. Не дивно, цей вид розвідки є невід'ємною складовою діяльності військових, оскільки надає інформацію, необхідну для визначення стратегічних напрямів держави.

Історичний розвиток методів рекогностування в Україні та світі характеризується еволюцією від традиційних, ручних методів до сучасних технологічних рішень. У минулому, рекогностування в основному здійснювалося за допомогою наземних військ, які здійснювали розвідку території, використовуючи переважно свої візуальні та слухові спостереження. З часом, з розвитком техніки та технологій, особливо у 20-му столітті, почали використовуватися більш складні засоби, такі як літаки та пізніше супутники для аерофотозйомки.

З появою комп'ютерних технологій і, особливо, інтернету, з'явилися нові можливості для аналізу та обробки даних. Сучасне рекогностування включає в себе використання різних видів датчиків, безпілотних літальних апаратів, супутникових систем спостереження та інших сучасних технологій.

Таблиця 1.1 Порівняльний аналіз традиційних та сучасних методів рекогностування.

|  |  |  |
| --- | --- | --- |
| Критерій | Традиційні методи | Сучасні методи |
| Технології | Наземні війська, візуальні та слухові спостереження | Безпілотні літальні апарати, супутникове спостереження, датчики |
| Обсяг даних | Обмежений обсяг даних, отриманих вручну | Великий обсяг даних, здатних до швидкої обробки |
| Точність | Залежить від індивідуального досвіду | Висока точність за рахунок технологій |
| Швидкість збору даних | Повільний, залежить від фізичних факторів | Швидкий, автоматизований процес |
| Безпека | Високий ризик для розвідників | Знижений ризик, можливість дистанційного керування |
| Вартість | Відносно низька | Висока через технологічні рішення |

Україна, як країна з активно розвиваючою сферою оборони та безпеки, також інтегрує сучасні методи рекогностування в свої військові структури. Керівні документи та стандарти, прийняті в Україні, відображають глобальні тенденції та напрямки розвитку у цій області. Вони включають детальні вказівки щодо використання сучасних технологій, їх інтеграції з традиційними методами, а також враховують аспекти кібербезпеки та захисту інформації.

Рекогностування проводиться також з метою вивчення районів (рубежів) можливого розташування та розгортання військ, маршрутів їх висунення, виправлення та доповнення топографічних карт.

Для цього створюються спеціальні рекогностувальні групи з офіцерів штабу і управлінь родів військ, спеціальних військ (служб). Для отримання даних у межах своєї території в ході рекогностування використовуються [вертольоти](https://uk.wikipedia.org/wiki/%D0%92%D0%B5%D1%80%D1%82%D0%BE%D0%BB%D1%96%D1%82), [літаки](https://uk.wikipedia.org/wiki/%D0%9B%D1%96%D1%82%D0%B0%D0%BA), швидкохідні машини управління, прилади [радіолокації](https://uk.wikipedia.org/wiki/%D0%A0%D0%B0%D0%B4%D0%B0%D1%80) та інше.[2].

Напрямок і детальність вивчення місцевості залежать від характеру бойового завдання. Наприклад, при підготовці до переправи через ріку вивчаються підходи до обраної ділянки ріки, характер її заплави і берегів; ширина, глибина і швидкість течії ріки, характер ґрунту на дні і берегів; місця для прихованого розташування військ та інші дані, а при підготовці до маршу командири всіх рівнів детально вивчають, головним чином, дорожню мережу і характеристики дорожніх споруд; умови руху поза дорогами, перешкоди на шляху руху військ, умови їх подолання тощо.

Таким чином, розвідка місцевості є важливою складовою військової розвідки і проводиться з метою визначення впливу тактичних властивостей місцевості на виконання бойового завдання, надійного орієнтування на ній та ефективного використання своїх бойових і транспортних засобів. Постійне та наполегливе ведення розвідки місцевості поряд з розвідкою противника за будь-яких обставин є одним із найважливіших обов’язків командирів підрозділів.

Розвідка безпосереднім оглядом дозволяє найкращим чином вивчити особливості місцевості, оцінити її прохідність, захисні і маскувальні властивості, визначити умови ведення вогню та орієнтування. Безпосередній огляд об’єктів місцевості проводиться в тому випадку, коли є можливість наблизитись до об’єктів, які розвідують, докладно обстежити їх і провести необхідні виміри та розрахунки.

Розвідка безпосереднім оглядом може проводитись розвідувальними дозорами (групами) на марші, у наступі, в обороні при відсутності зіткнення з противником та в інших випадках. Порядок руху, віддалення та способи дії дозору визначається поставленим йому завданням і залежить від обстановки, що склалася. Розвідка місцевості ведеться дозором під час руху, а також на коротких зупинках.

У сучасному військовому рекогностуванні ключову роль відіграють різноманітні технології, зокрема безпілотні літальні апарати (БПЛА), супутникове спостереження, а також традиційні наземні війська.

БПЛА відіграють важливу роль у сучасному процесі рекогностування, особливо у військовій сфері. Їх використання дозволяє збирати розвідувальні дані, здійснюючи спостереження з повітря, що є ключовим для точного аналізу місцевості та виявлення потенційних загроз.

Основною перевагою БПЛА є можливість виконання завдань у високоризикованих або недоступних для людини зонах. Це включає зони активних військових дій, райони зі складними природними умовами чи обмеженим доступом. БПЛА забезпечують високу точність збору даних та можливість оперативного реагування на зміни на місцевості.

Іншою важливою характеристикою БПЛА є їх варіативність. Існують різні типи БПЛА, від малогабаритних (рис. 1.1), які можуть використовувати солдати, до великих, здатних виконувати тривалі місії на великих висотах (рис 1.2).

Така варіативність дозволяє адаптувати використання БПЛА під конкретні завдання та умови.

Супутникове спостереження є одним з ключових методів у сучасному військовому рекогностуванні. Цей метод включає використання супутників для збору даних про земну поверхню, що дозволяє отримати цінну інформацію про місцевість, потенційні загрози та військові об'єкти.

Супутникове спостереження має кілька переваг:

* Здатність забезпечувати високу роздільну здатність зображень;
* Можливість спостереження за великими територіями без втручання в місцевість;
* Незалежність від географічних та політичних обмежень.

Також слід враховувати сучасні тенденції та інновації у галузі космічних технологій, які відкривають нові можливості для вдосконалення супутникового спостереження, зокрема, збільшення частоти обновлення даних та поліпшення якості зображень.

Розвідка безпосереднім оглядом є одним із класичних та найбільш традиційних методів рекогностування в військовій практиці. Цей метод передбачає фізичне присутність військового персоналу на місцевості, що досліджується, для збору розвідувальних даних. Основними перевагами цього підходу є можливість отримання детальної та точної інформації про місцевість, її особливості, а також потенційну активність ворога.

Використання цього методу вимагає від військових розвідників значних навичок та досвіду, оскільки вони повинні бути здатні не лише точно оцінювати обстановку, а й залишатися непоміченими ворожими силами. Це включає в себе вміння правильно користуватися природним і штучним прикриттям, розуміння питань маскування та тактики малого підрозділу.

Також є цікавий метод рекогностування під назвою кіберрекогностування. Кіберрекогностування — це процес збору розвідувальної інформації через кіберпростір. Це включає в себе моніторинг та аналіз цифрових даних, які можуть бути використані для отримання стратегічної інформації про противника або конкурента. Кіберрекогностування зосереджується на зборі даних з різних джерел, таких як соціальні медіа, веб-сайти, форуми, бази даних, мережевий трафік та інші цифрові платформи.

Цей процес включає в себе використання різних методів та інструментів, таких як веб-скрапінг, аналіз метаданих, інтерцептація комунікацій, виявлення цифрових слідів та інші кібернетичні техніки. Важливим аспектом є також захист власних даних та мережевої інфраструктури від рекогностування противниками. Кіберрекогностування має ключове значення в сучасному цифровому світі, де інформаційна перевага може мати значний вплив на військові, політичні та економічні рішення.

Кіберрекогностування являє собою важливий компонент сучасних військових та розвідувальних операцій, забезпечуючи збір та аналіз інформації у кіберпросторі. Воно включає в себе різноманітні техніки та методи для ідентифікації, моніторингу, аналізу та збору цифрових даних, які можуть бути використані для підтримки стратегічних військових цілей. За допомогою кіберрекогностування можливо визначити потенційні загрози, слабкі місця противника, а також отримати важливі дані, які можуть вплинути на хід бойових дій.

Одним з ключових елементів кіберрекогностування є кіберрозвідка, яка зосереджена на отриманні інформації про противників чи конкурентів, що включає в себе аналіз мережевого трафіку, збір даних з відкритих та закритих джерел, виявлення вразливостей у кіберінфраструктурі противника. Використання кіберрекогностування дозволяє ефективно здійснювати моніторинг, передбачення та запобігання кібератакам, а також готувати власні кібероперації.

Зростання значення кіберпростору в сучасному світі змушує військові та розвідувальні служби звертати увагу на розвиток кіберрекогностувальних здібностей. Розвідувальна інформація, отримана через кіберпростір, може включати в себе деталі про військові плани, стратегії, технології та інші важливі аспекти, що дозволяють краще зрозуміти потенціал та наміри противника.

Кіберрекогностування може відігравати значну роль у аналізі місцевості для розгортання військ, оскільки цей процес включає збір та аналіз великого обсягу даних, які мають пряме відношення до географічних та стратегічних аспектів. Завдяки кіберрекогностуванню можна здійснювати глибокий аналіз мережевих даних, які можуть містити важливу інформацію про топографію, кліматичні умови, наявність водних та інших природних ресурсів у потенційній зоні розгортання військ.

Крім того, кіберрекогностування дозволяє ефективно відслідковувати та аналізувати кіберактивність у зоні розгортання, включаючи комунікаційні мережі, розташування кіберінфраструктури та потенційні кіберзагрози. Це допомагає військовим планувальникам розробляти більш ефективні стратегії розгортання та забезпечувати кібербезпеку військових оперативних зон.

Наостанок, кіберрекогностування сприяє збору інтелектуальної інформації про ворожі військові розташування, логістику та планування. Ця інформація є критично важливою для прийняття рішень щодо вибору місця та способу розгортання власних військових сил, дозволяючи адаптуватися до потенційних загроз і використовувати вигідні тактичні переваги.

## Недоліки існуючих підходів

Важливо відзначити, що кожен метод має свої обмеження, які варто враховувати при їх виборі та застосуванні. Ці обмеження можуть бути пов'язані з різними факторами, такими як технологічні обмеження, економічна вартість, вразливість до зовнішніх умов, а також питання безпеки та конфіденційності. Крім того, ефективність методів рекогностування може варіюватися в залежності від специфіки операційного середовища і цілей місії. Таким чином, ретельний аналіз недоліків є ключовим для розуміння загальної картини та вибору найбільш оптимальних методів рекогностування.

* + 1. Недоліки використання БПЛА

БПЛА як метод рекогностування мають ряд недоліків, які важливо враховувати при їх використанні. Одним із основних недоліків є їх вразливість до протиповітряної оборони та електронних завад. БПЛА можуть легко стати ціллю для засобів протиповітряної оборони ППО або бути збитими за допомогою радіоелектронного придушення. Інший важливий аспект – це обмежена тривалість польоту та необхідність періодичного підзаряджання або заправки, що може обмежувати час спостереження та вимагає додаткової логістичної підтримки.

Крім того, БПЛА залежать від якості погодних умов і можуть бути неефективними у випадку сильних вітрів, дощу або інших метеорологічних перешкод. Існує також питання конфіденційності та безпеки даних, оскільки інформація, передана БПЛА, може бути перехоплена або зламана. Висока вартість БПЛА та їх обслуговування також є значним недоліком, особливо для обмежених бюджетів.

* + 1. Недоліки використання супутникового спостереження

Супутникове спостереження як метод рекогностування володіє рядом недоліків. Перш за все, супутники залежні від погодних умов та часу доби, що може обмежувати якість зображення або робити зйомку неможливою в певних умовах. Висока вартість запуску та утримання супутників також є значним фактором, особливо для країн із обмеженим бюджетом. Існує також питання затримки у передачі даних, що може бути критичним у ситуаціях, що швидко змінюються. Крім того, розвідувальні супутники можуть бути вразливими до протисупутникової зброї, що становить ризик для безпеки інформації. Заключно, орбітальні обмеження та фіксовані траєкторії супутників можуть обмежувати можливості спостереження за певними регіонами, що знижує гнучкість використання цього методу у різних тактичних ситуаціях.

* + 1. Недоліки використання традиційних методів рекогностування

Традиційні методи рекогностування наземними військами мають декілька значних недоліків, особливо у контексті сучасних воєнних конфліктів і технологій. По-перше, вони часто спираються на застарілі карти та інформацію, яка може не відображати поточні географічні чи інфраструктурні зміни. Це робить планування на основі таких карт ризикованим і менш ефективним, оскільки реальна місцевість може значно відрізнятися від картографічних даних.

Крім того, фізичне рекогностування місцевості вимагає значних часових та людських ресурсів. Воно може бути повільним і часто піддає персонал небезпеці, особливо у зонах активних бойових дій. Іншим важливим аспектом є зменшення елементу сюрпризу, оскільки фізичне рекогностування може бути легко виявлене противником.

Ще одним недоліком є обмеженість інформації, яку можна зібрати через наземне рекогностування. Такий підхід не завжди здатен забезпечити повне уявлення про масштаби або специфіку території, що особливо актуально при наявності складного рельєфу або прихованих об'єктів.

Наземне рекогностування також має обмеження у термінах оперативності та гнучкості. Військові ситуації можуть швидко змінюватися, а традиційні методи не завжди здатні швидко адаптуватися до нових умов чи надати актуальну інформацію в короткі терміни.

Враховуючи ці недоліки, необхідність впровадження сучасних технологій, таких як системи доповненої реальності, стає очевидною. Ці технології можуть допомогти вирішити багато з цих проблем, пропонуючи більш безпечні, оперативні та ефективні рішення для рекогностування місцевості.

* + 1. Недоліки використання традиційних методів рекогностування

Недоліки використання кіберрекогностування можуть бути значними та впливати на ефективність військових операцій. Перший з них - це висока залежність від технологічної інфраструктури. Кіберрекогностування потребує складних технологічних систем, які можуть бути вразливими до атак та збоїв. Наприклад, перебої у мережі або втрата даних можуть серйозно ускладнити збір розвідувальної інформації.

Друга проблема - це питання безпеки даних. Коли збирається та обробляється велика кількість інформації, існує ризик витоку даних, що може призвести до компрометації військових операцій. Захист інформації, яка збирається через кіберрекогностування, є критично важливим, але також складним завданням.

Третій недолік - це обмежена точність інформації. Хоча кіберрекогностування може забезпечити доступ до великих обсягів даних, ця інформація може бути неточною або застарілою. Це створює ризик прийняття рішень на основі недостовірної або неповної інформації.

Четвертий недолік - це високі вимоги до кваліфікації персоналу. Ефективне використання кіберрекогностування вимагає висококваліфікованих фахівців, які мають глибокі знання в області кібербезпеки, аналізу даних та інформаційних технологій. Залучення та підготовка таких фахівців може бути складним та дороговартісним процесом.

Нарешті, п'ятий недолік пов'язаний з етичними та правовими аспектами. Кіберрекогностування часто оперує в зоні, де законодавчі та етичні норми можуть бути нечіткими. Існує ризик порушення приватності та інших прав людини при зборі даних, особливо коли мова йде про моніторинг цивільних осіб та організацій. Це вимагає ретельного розгляду етичних і правових аспектів під час планування та виконання кіберрекогностувальних операцій.

## Визначення ключових вимог до системи

Визначаючи ключові вимоги до системи рекогностування, необхідно розглянути ряд фундаментальних аспектів, які впливають на її ефективність та надійність. Серед основних критеріїв слід виокремити точність та надійність даних. Система повинна забезпечувати високу точність зібраних даних, щоб мінімізувати ризик помилкових інтерпретацій, які можуть призвести до неправильних тактичних рішень. Це включає в себе як точність збору даних, так і їх подальшу обробку.

Швидкість обробки інформації в системах рекогностування відіграє вирішальну роль, особливо у воєнних умовах, де кожна секунда може мати стратегічне значення. У сучасному світі, де обстановка на полі бою може мінятися миттєво, здатність швидко збирати, аналізувати та передавати інформацію є ключовою для успішного ведення бойових дій та виживання військових підрозділів.

Важливим є також організаційний — забезпечення належної комунікації та координації між різними військовими підрозділами. Інтеграція інформації з різних джерел та її оперативне надання командирам на місцях дає можливість для швидкого прийняття рішень, що може включати пересування військ, зміну тактики чи виклик підкріплення. Таким чином, сучасна воєнна стратегія вимагає не лише швидкості збору даних, але й швидкості їх обробки та передачі, що забезпечує тактичну перевагу та підвищує шанси на успіх у воєнних конфліктах.

Важливим аспектом є також гнучкість системи. Вона повинна бути адаптивною до різноманітних умов та здатна працювати в широкому діапазоні сценаріїв – від пустельних до гірських ландшафтів. Гнучкість також стосується здатності системи інтегруватися з іншими технологічними рішеннями, включаючи супутникове спостереження, БПЛА та інші засоби рекогностування.

Безпека персоналу – ще один важливий аспект. У сучасних умовах ведення війни висока вартість людського життя робить важливим забезпечення безпеки персоналу, що займається рекогностуванням. Технології повинні зменшувати ризик для військових, дозволяючи виконувати місії дистанційно або автоматизовано.

Економічна ефективність також грає важливу роль. Системи рекогностування повинні бути не лише технологічно продуктивними, але й економічно вигідними. Це включає в себе вартість розробки, утримання та експлуатації системи. Важливо, щоб інвестиції у систему рекогностування пропорційно відповідали отриманим вигодам та перевагам.

Важливо також, щоб система була інтуїтивно зрозумілою та легкою в експлуатації для військового персоналу, забезпечуючи швидке введення в дію та ефективне використання у стресових або бойових умовах. Це включає в себе розробку зручного користувацького інтерфейсу та забезпечення сумісності з існуючим обладнанням та програмним забезпеченням.

Система повинна бути розроблена з урахуванням потенціалу для майбутніх модернізацій та оновлень. Технологічний прогрес вимагає постійного оновлення систем для підтримки їх актуальності та ефективності, що передбачає можливість легкої інтеграції нових технологій та покращень.

## **Висновки до розділу 1**

На основі аналізу процесу рекогностування військами місцевості для подальшого розгортання військ, можна зробити вагомі висновки. По-перше, сучасні методи рекогностування, такі як використання безпілотних літальних апаратів та супутникового спостереження, надають значні переваги порівняно з традиційними методами, зокрема у точності, швидкості обробки даних та безпеці. Водночас, ці методи мають певні обмеження, такі як вразливість до протиповітряної оборони, залежність від погодних умов та висока вартість.

Розвиток технологій у галузі рекогностування відкриває нові можливості для покращення військових стратегій та тактик, забезпечуючи командування важливою інформацією для прийняття рішень. Важливість точної та актуальної інформації не може бути недооцінена, оскільки вона безпосередньо впливає на успіх військових операцій.

В умовах сучасного конфлікту ключовим є розуміння необхідності інтеграції різних методів рекогностування та постійного оновлення технологій. Це дозволить створити більш гнучку та ефективну систему розвідки, здатну швидко адаптуватися до змін умов та вимог.

З огляду на це, критично важливим є врахування всіх аспектів при розробці та впровадженні систем рекогностування, включаючи технічні характеристики, безпеку даних, економічну ефективність та здатність до інтеграції з іншими системами. Висновки дослідження підкреслюють важливість постійного розвитку та оновлення систем рекогностування, адаптації до сучасних вимог та інтеграції з новітніми технологіями.[3]

Отже, було вирішено розробити систему рекогностування, яка використовує доповнену реальність на базі платформи *Flutter*. Такий підхід передбачає інтеграцію новітніх технологічних рішень з класичними методами рекогностування. Система на базі *Flutter* дозволить використовувати мобільні пристрої для візуалізації та аналізу геопросторових даних в реальному часі, значно підвищуючи оперативність і точність інформації, доступної військовому персоналу.

Завдяки використанню доповненої реальності, система забезпечить більш наочне представлення місцевості, що є критично важливим для планування та проведення військових операцій[4]. Це також дозволить командирам і військовим аналітикам краще оцінювати потенційні загрози та планувати стратегічні дії. Таким чином, інтеграція сучасних технологій доповненої реальності відкриває нові перспективи для розвитку систем рекогностування, роблячи їх більш ефективними у відповіді на сучасні воєнні виклики.

.

РОЗДІЛ 2 РОЗРОБКА ПІДСИСТЕМИ ІНТЕРАКТИВНОГО РЕКОГНОСТУВАННЯ

## Проектування архітектури системи та опис її структури

Архітектура будь-якої інформаційної системи відіграє вирішальну роль у забезпеченні її ефективності, гнучкості та масштабованості. В контексті розробки системи інтерактивного рекогностування, основними цілями архітектури є створення надійної, безпечної та легко адаптованої платформи, яка здатна ефективно обробляти великі обсяги даних, забезпечувати високу швидкість обробки запитів та володіти високим рівнем інтеграції з різними компонентами системи.

Основні завдання архітектури включають забезпечення взаємодії між клієнтськими та серверними частинами системи, інтеграцію з базами даних, а також розробку інтуїтивно зрозумілого та зручного інтерфейсу для кінцевих користувачів.[5] Важливо також забезпечити можливість легкого оновлення та масштабування системи, а також високий рівень безпеки обробки та зберігання даних.

Саме такі цілі та завдання вимагають ретельного планування та глибокого аналізу на етапі проектування архітектури, щоб система могла ефективно відповідати всім вимогам та очікуванням.

Правильне проектування архітектури системи має ключове значення, оскільки воно впливає на всі аспекти роботи системи, від її продуктивності до масштабованості. Детальне та продумане архітектурне проектування впливає на легкість внесення змін, обслуговування системи, а також на можливість інтеграції з іншими системами і технологіями. Це дозволяє системі бути стійкою до змін у технологіях та потребах бізнесу.

Таблиця 2.1 Цілі та завдання архітектури.

|  |  |
| --- | --- |
| Цілі архітектури | Завдання для досягнення цілі |
| Масштабованість | Розробка модульної структури, що дозволяє додавати або оновлювати компоненти без зайвих витрат часу та ресурсів. |
| Безпека | Впровадження сучасних механізмів шифрування та аутентифікації, розробка політик безпеки для захисту даних. |
| Інтеграція | Створення єдиного інтерфейсу для взаємодії з різними зовнішніми системами та сервісами. |
| Продуктивність | Оптимізація алгоритмів та використання ефективних технологій для прискорення обробки даних. |
| Гнучкість | Впровадження принципів гнучкого проектування для легкого адаптування системи до змінних вимог. |
| Надійність | Забезпечення стабільної роботи системи, мінімізація помилок та збоїв у роботі. |

Правильно спроектована архітектура при процесі проектуванні архітектури (рис. 2.1) також дозволяє уникнути багатьох проблем, які можуть виникати в майбутньому, включаючи проблеми з продуктивністю, масштабованістю та безпекою. Вона забезпечує основу для створення системи, що може ефективно адаптуватися до змінних вимог та забезпечити високий рівень обслуговування протягом усього часу її використання.

Таким чином, важливість правильного проектування архітектури не може бути недооцінена[6]. Вона лежить в основі створення ефективної, надійної та гнучкої системи, яка здатна відповідати поточним і майбутнім вимогам.

Стек, що складається з *Flutter (dart), Spring Boot (java), React (js), MySQL,* був обраний через їхню взаємодоповнюваність, ефективність та сучасність.

*Flutter* відіграє ключову роль у розробці мобільного додатку системи. Його переваги включають:

Крос-платформність: *Flutter* дозволяє розробляти додатки для різних платформ, включаючи *iOS* та *Android* , з одного кодового базису. Це значно спрощує та прискорює процес розробки.

Висока продуктивність: *Flutter* використовує компіляцію з мови *Dart* у рідний код, що забезпечує високу продуктивність додатків.

Гнучкий *UI: Flutter* надає широкі можливості для створення налаштованого та динамічного інтерфейсу користувача, що є критично важливим для додатків, які використовують доповнену реальність.

Велика спільнота та підтримка: Як продукт від *Google, Flutter* має велику та активну спільноту розробників, що забезпечує гарну підтримку та велику кількість ресурсів для навчання.

Інтеграція з іншими технологіями: *Flutter* легко інтегрується з різними бекенд-сервісами та базами даних, що робить його ідеальним для створення комплексних мобільних додатків.

Ці фактори роблять *Flutter* ідеальним вибором для створення інтерактивного мобільного додатку, який є частиною системи інтерактивного рекогностування.

Архітектура *Flutter* (рис.2.3) базується на реактивному програмуванні та використовує компонентно-орієнтований підхід, що дозволяє створювати гнучкий та динамічний інтерфейс користувача. Вона включає в себе ряд рівнів: від нижнього рівня, який забезпечує базові функції рендерингу та керування станом, до вищого, що включає віджети та фреймворки. Ці віджети є основними будівельними блоками для будь-якого *Flutter*-додатку, дозволяючи розробникам легко створювати складні *UI*-структури.[7] *Flutter* також ефективно використовує складові машинного коду для оптимізації продуктивності та швидкості.

У данній системі застосовується *Flutter* для розробки мобільного додатку, який відіграє ключову роль у забезпеченні інтерфейсу користувача та взаємодії з серверною частиною. *Flutter* дозволяє нам створити гнучкий та інтуїтивно зрозумілий інтерфейс, який ефективно інтегрується з різними функціональними елементами системи, включаючи розміщення та управління об'єктами доповненої реальності. Важливим аспектом є також розділення ролей користувачів, зокрема між командиром та підлеглими, забезпечуючи ефективне управління та координацію дій в реальному часі.

Для серверної частини системи інтерактивного рекогностування, вибір впав на *Spring Boot*, що є одним з найпопулярніших фреймворків для розробки на *Java*. *Spring Boot* вирізняється своєю здатністю спрощувати процес розробки, забезпечуючи швидке та легке створення стійких застосунків.

*Spring Boot* — це фреймворк, який призначений для спрощення процесу розробки та розгортання веб-додатків, написаних на мові програмування *Java*. Він забезпечує швидке створення автономних, готових до виробництва веб-додатків з мінімальними зусиллями. Основна особливість *Spring Boot* полягає в тому, що він автоматично налаштовує *Spring*-додатки на основі доданих залежностей, значно спрощуючи конфігурацію та запуск додатків. Це включає в себе вбудований сервер, автоконфігурацію компонентів і зручність управління залежностями.

Основні характеристики та переваги *Spring Boot*:

* автоконфігурація;
* вбудований сервер;
* легкість в розгортанні;
* широкі можливості тестування;
* мікросервісна архітектура;
* спільнота та підтримка.

Автоконфігурація в *Spring Boot* - це процес, за допомогою якого *Spring Boot* автоматично налаштовує додаток на основі доданих бібліотек (залежностей) у проекті. Це означає, що якщо додається залежність для веб-функціоналу, *Spring Boot* автоматично налаштує додаток для роботи як веб-додаток. Це включає в себе різні аспекти, такі як налаштування веб-сервера, баз даних, безпеки, кешування та інші звичайні конфігураційні завдання, які зазвичай вимагають ручного налаштування в традиційних *Spring*-додатках. Такий підхід спрощує розробку, зменшує кількість коду для налаштування та дозволяє розробникам зосередитися на бізнес-логіці додатку.

Вбудований сервер у *Spring Boot* значно спрощує процес розгортання та управління серверним середовищем. Зазвичай, для запуску *Java*-додатку потрібен зовнішній сервер застосунків, такий як *Apache Tomcat, IBM WebSphere* або *Oracle WebLogic*. Однак, *Spring Boot* включає вбудовані версії серверів, як-от *Tomcat, Jetty* або *Undertow*, що дозволяє запускати веб-додатки без необхідності додаткового серверного програмного забезпечення.

Мікросервісна архітектура (рис. 2.4), що підтримується *Spring Boot*, є ключовою для сучасних застосунків через її гнучкість та модульність. Ця архітектура дозволяє розробникам створювати окремі сервіси, кожен з яких виконує певну функцію і може бути розгорнутий незалежно від інших. Такий підхід сприяє кращій масштабованості та легкості управління, а також забезпечує можливість швидкої адаптації до змін у вимогах чи технологіях. Використання *Spring Boot* в контексті мікросервісної архітектури також спрощує розробку, оскільки розробники можуть зосередитись на конкретному сервісі, не турбуючись про весь застосунок в цілому. Це знижує складність системи та полегшує її тестування та розгортання.

У системі інтерактивного рекогностування, *Spring Boot* виконує ключову роль серверу. Він забезпечує високу продуктивність та легкість управління бізнес-логікою. *Spring Boot* спрощує конфігурацію та інтеграцію з базою даних (БД), що є критично важливим для забезпечення швидкого доступу до даних та їх надійного зберігання.

Одним з основних аспектів є використання *Spring* *Boot* для реалізації *API*, що дозволяє підключення клієнтської частини (розробленої на *Flutter*) та адміністративної панелі (створеної з використанням *React*). Це гарантує гнучкість системи та зручність управління даними.

Важливою частиною безпеки системи є використання бібліотеки "*jjwt*" для роботи з *JSON Web Tokens (JWT)* з її зручною та водночас безпечною структурою (рис. 2.5) . Це забезпечує безпечну аутентифікацію та авторизацію користувачів, що є критично важливим для військових додатків, де безпека даних має вирішальне значення.

У контексті збереження файлів, *Spring Boot* дозволяє легко інтегрувати рішення для ефективного управління файлами, що є важливим для збереження великих обсягів даних, включаючи картографічні матеріали та розвідувальні дані.

Переваги використання *Spring Boot* у цій системі включають високу швидкість розробки, простоту впровадження змін, вбудовані інструменти для забезпечення безпеки, а також легкість інтеграції з іншими компонентами системи, що робить його ідеальним рішенням для розробки військових додатків.

У системі керування користувачами та даними виконується за допомогою адміністративної панелі, розробленої на *React. React* — це декларативний, ефективний та гнучкий *JavaScript*-фреймворк для побудови користувацьких інтерфейсів. Його основна перевага полягає у забезпеченні швидкого та гладкого взаємодії користувача з веб-додатком завдяки використанню віртуального *DOM* (*Document Object Model*).

Основні характеристики та переваги *React* у контексті розробки системи включають:

* компонентний підхід;
* однонаправлене прив'язування даних;
* віртуальний *DOM*;
* широка екосистема та спільнота.

Таким чином, використання *React* для розробки адміністративної панелі дозволяє створити надійний, ефективний та інтуїтивно зрозумілий інтерфейс для керування користувачами та даними у системі.

Компонентний підхід у *React* полягає в тому, що інтерфейс додатка розбивається на окремі частини, які називаються компонентами. Кожен компонент відповідає за конкретну функціональність або відображення частини інтерфейсу. Це дозволяє розробникам розбивати складні завдання на більш прості, оскільки кожен компонент може бути розроблений і тестований окремо.

Кожен компонент може мати свій власний стан і власну логіку, і вони можуть бути використані в інших компонентах.[8] Це спрощує розробку, оскільки різні частини інтерфейсу можуть бути розроблені незалежно одна від одної і згодом об'єднані в єдиний додаток.

Компоненти також полегшують тестування, оскільки вони можуть бути перевірені окремо на коректну роботу, а потім об'єднані в цілісний додаток. Це дозволяє виявляти і виправляти помилки та дефекти швидше і ефективніше. Крім того, компоненти можуть бути перевикористані в інших проектах або частинах додатка, що також сприяє підвищенню продуктивності та підтримці коду.

Однонаправлене прив'язування даних - це підхід у програмуванні, який дозволяє краще управляти станом додатку та полегшує його розробку та розуміння. У цьому підході дані рухаються лише в одному напрямку - від джерела до споживача. Це означає, що зміни в даних автоматично відображаються в інших частинах додатку, які використовують ці дані, без необхідності вручну синхронізувати їх.

Однонаправлене прив'язування даних спрощує розуміння, як працює програма, і зменшує ризик виникнення непередбачених помилок. Воно також робить код більш масштабованим і обслуговуваним, оскільки зменшує взаємозалежність між різними частинами програми. Крім того, цей підхід полегшує тестування та відлагодження, оскільки зміни в даних можна легко відстежувати та контролювати.

Віртуальний *DOM*, який використовується у фреймворку *React*, відіграє важливу роль у підвищенні продуктивності додатку. Цей механізм дозволяє оптимізувати процес оновлення інтерфейсу, мінімізуючи кількість дорогих за часом операцій маніпуляції з реальним *DOM*. Віртуальний *DOM* працює як легка копія реального *DOM*, де зберігається стан інтерфейсу (рис. 2.6). Коли відбуваються зміни в інтерфейсі, спочатку вони застосовуються до віртуального *DOM*. Після цього *React* порівнює поточний стан віртуального *DOM* із його попереднім станом, визначаючи оптимальний спосіб внесення змін до реального *DOM*. Цей процес, відомий як "порівняння дерев" (*reconciliation*), дозволяє значно зменшити кількість маніпуляцій з реальним *DOM*, що є дуже витратними за ресурсами, тим самим підвищуючи швидкодію та відгук інтерфейсу користувача.

Адміністративна панель розроблена за допомогою *React* включає функції обробки користувачів, такі як додавання, видалення та управління їхніми правами. Особливу увагу потрібно приділити завантаженню та управлінню моделями бойових одиниць, що є критичним для забезпечення їх ефективного відображення в клієнтській частині системи, розробленої з використанням *Flutter*. Інтеграція між адміністративною панеллю та серверною частиною, заснованою на *Spring Boot*, забезпечує надійний обмін даними і високий рівень безпеки.[9] Серверна частина відіграє ключову роль у забезпеченні стабільної роботи системи та ефективної взаємодії з базою даних, реалізованою на *MySQL*, що дозволяє зберігати та обробляти великі обсяги даних, необхідних для функціонування системи.

Вибір бази даних для системи є критичним рішенням, оскільки він впливає на багато аспектів: продуктивність, надійність, масштабованість та безпеку системи. Розглянемо перелік деяких популярних баз даних:

* *Oracle Database*: Високопродуктивна система управління базами даних, ідеально підходить для великих корпорацій та складних додатків.
* *Microsoft SQL Server*: Популярний у корпоративному секторі, пропонує інтегровані рішення для аналізу та управління даними.
* *PostgreSQL*: Відкритий, безкоштовний, масштабований та має потужні функції для роботи з великими даними.
* *MongoDB*: Документоорієнтована *NoSQL* БД, яка є відмінним вибором для додатків, які потребують швидкого оброблення великих обсягів неструктурованих даних.

Було обрано *MySQL* через його гнучкість, відносну простоту, високу продуктивність та широке співтовариство підтримки. *MySQL* ефективно впорається з великою кількістю транзакцій, є надійним і легко інтегрується з іншими компонентами системи інтерактивного рекогностування.

*MySQL* — це відкритий і широко використовуваний реляційний система управління базами даних (СУБД), який є вибором для багатьох веб-додатків, в тому числі використовується такими великими та популярними платформами, як *WordPress, Drupal, Joomla* та іншими. Завдяки своїй надійності, гнучкості та простоті використання, *MySQL* зарекомендував себе як одна з провідних технологій у сфері баз даних. Він підтримує велику кількість операційних систем та має велику спільноту розробників, яка постійно працює над його вдосконаленням та підтримкою. *MySQL* є відмінним рішенням для реалізації як невеликих, так і великомасштабних проектів, завдяки своїй високій продуктивності, масштабованості та безпеки.

*MySQL* пропонує значні переваги для військових систем рекогностування, завдяки своїй надійності, безпеці та гнучкості. Як надійна та випробувана система управління базами даних, *MySQL* забезпечує високий рівень безпеки, який є критичним для військових додатків, де захист конфіденційної інформації має вищий пріоритет. *MySQL* підтримує різноманітні методи шифрування, що дозволяє забезпечити безпечне зберігання та передачу чутливих даних.

Завдяки своїй високій продуктивності та масштабованості, *MySQL* ефективно обробляє великі обсяги даних, що є типовими для військових систем рекогностування, де потрібно швидко обробляти інформацію з різних джерел. Крім того, *MySQL* має гнучку систему управління, що дозволяє легко інтегруватися з різними програмними середовищами, що спрощує розробку та підтримку військових додатків.

Надійна підтримка транзакцій та можливість відновлення після збоїв роблять *MySQL* ідеальним рішенням для військових застосувань, де критично важливо забезпечувати цілісність та доступність даних. Враховуючи ці переваги, *MySQL* є відмінним вибором для систем рекогностування, що вимагають високої надійності, безпеки та гнучкості.

Основа будь-якої інформаційної системи – це її база даних, архітектура якої повинна бути чітко продуманою для забезпечення ефективності та безпеки даних. Розглянемо структуру бази даних, представлену на скріншоті, для системи рекогностування (рис. 2.7).

У цій базі даних ми маємо кілька взаємопов’язаних таблиць, кожна з яких містить важливі дані для системи:

* таблиця *neovision\_client*: Зберігає інформацію про користувачів системи. Включає поля для ідентифікатора користувача, електронної пошти, логіну, пароля, ролі у системі та ідентифікатора командира, якщо це застосовно;
* таблиця *neovision\_session*: Використовується для збереження сесій, що є важливими для відстеження активних процесів рекогностування. Містить інформацію про центральні координати сесії, опис, зображення мапи, назву та час початку сесії;
* таблиця *neovision\_recon*: Служить для запису звітів рекогностування. Включає поля для ідентифікаторів сесії та користувача, а також час створення звіту;
* таблиця *neovision\_report\_unit*: Використовується для опису конкретних одиниць обладнання чи військових одиниць, виявлених під час рекогностування. Містить деталі, такі як опис, координати розташування та ідентифікатор пов'язаного звіту;
* таблиця *neovision\_unit*: Зберігає інформацію про різні типи одиниць, які можуть бути використані в системі. Це можуть бути моделі обладнання, військова техніка тощо;
* таблиця *neovision\_client\_session*: Забезпечує зв'язок між користувачами та сесіями, визначаючи, які користувачі є частиною яких сесій;
* таблиця *neovision\_user*: Є основою для управління користувачами системи, включаючи їх аватари, електронні адреси, логіни та інші персональні дані;
* таблиця *neovision\_report\_unit\_photos*: Забезпечує збереження фотографій, пов'язаних з конкретними звітами про одиниці, що дозволяє збагатити звіти візуальним контентом.

Кожна з цих таблиць має свої первинні ключі (наприклад, *id*) для унікальної ідентифікації кожного запису, що є стандартною практикою в базах даних для забезпечення цілісності даних. Відносини між таблицями визначені через зовнішні ключі, що дозволяє встановити зв'язки між різними сутностями — наприклад, зв'язок між користувачем і сесією рекогностування, або між звітом і одиницею обладнання.

Індекси використовуються для оптимізації запитів до бази даних, особливо для таблиць з великою кількістю записів. Вони дозволяють швидко виконувати пошук та фільтрацію даних за різними параметрами, такими як ідентифікатор сесії чи координати одиниць рекогностування.

З огляду на використання бази даних у військовій системі рекогностування, важливо підкреслити потребу в ретельному проектуванні з урахуванням вимог безпеки, надійності та швидкості доступу до даних. Такий підхід забезпечить, що система буде не тільки ефективною у своїй роботі, але й захищеною від несанкціонованого доступу та здатною витримувати високі навантаження під час військових операцій.

Обрана структура бази даних відображає оптимальний баланс між нормалізацією даних та їх доступністю, що є ключовим для ефективності обробки даних. Нормалізація даних дозволяє уникнути дублювання та забезпечити послідовність, що важливо для забезпечення цілісності військових даних та зниження ризику помилок. Водночас, ретельно спроектовані індекси та використання зовнішніх ключів підсилюють швидкість пошуку та запитів, забезпечуючи швидкий доступ до відповідної інформації.

Спрощення структури, де це можливо, без втрати необхідної інформації, сприяє підвищенню продуктивності та зниженню навантаження на систему. Це особливо важливо в умовах, де швидкість обробки даних може бути критичною, наприклад, під час військових операцій, коли рішення потрібно приймати швидко на основі останніх даних рекогностування.

Загалом, обрана структура бази даних відповідає високим вимогам до обробки, зберігання та захисту військової інформації, забезпечуючи надійну основу для розробки системи інтерактивного рекогностування.

Безпека даних у *MySQL* є критичною складовою для будь-якої системи, особливо у військовому контексті, де дані мають високу чутливість. Основними факторами, що забезпечують безпеку в *MySQL*, є реалізація міцних механізмів аутентифікації та контролю доступу, що дозволяють обмежити доступ до даних тільки для авторизованих користувачів. Шифрування даних, як на рівні зберігання, так і при передачі, відіграє ключову роль у захисті від несанкціонованого доступу або перехоплення. Окрім того, регулярне оновлення та патчування *MySQL* допомагають захистити базу даних від відомих вразливостей та забезпечити відповідність сучасним стандартам безпеки. Резервне копіювання та стратегії відновлення даних є важливими для запобігання втраті даних у разі технічних збоїв або кібератак. Впровадження цих заходів забезпечує надійний захист даних у системі на базі *MySQL*.

Використання *JPA/Hibernate* у системі дозволить ефективно керувати реляційною базою даних через *Java* об'єкти. *Hibernate* - це реалізація *JPA*, яка допомагає в автоматизації та управлінні взаємозв'язками між *Java* класами та таблицями бази даних, а також в запитах до бази даних з використанням *JPQL* (*Java Persistence Query Language*). Ви можете визначати сутності у вашій системі за допомогою анотацій або *XML* конфігурацій, керувати транзакціями, і виконувати запити до БД, не пишучи безпосередньо *SQL* код. Це значно спрощує розробку та підтримку додатку, особливо при роботі з великим обсягом даних та складними взаємозв'язками.

На основі вище зазначеної архітектури представлено структурну схему системи інтерактивного рекогностування (рис. 2.8), яка показує взаємодію компонентів системи та пояснює внутрішню складову.

Розглянувши структурну схему, можна визначити, що система інтерактивного рекогностування складається із двох основних блоків: клієнтської частини, представленої мобільним додатком, і серверної частини, що включає в себе декілька модулів та базу даних. Основний зв'язок між цими частинами забезпечується через *Public* *API*, що дозволяє обмін інформацією між сервером та клієнтом.

Компоненти серверної частини:

* модуль Адміністрування Системи Рекогностування: включає інструменти для управління користувачами та сесіями, а також для контролю дій користувачів і захисту інформації;
* модуль Взаємодії з Користувачем на Основі *React*: цей модуль представляє функціонал адміністративної панелі та забезпечує інтерфейс для взаємодії з користувачами системи;
* демон Керування Внутрішніми Модулями Системи: це ядро серверної частини, яке координує роботу всіх внутрішніх модулів, обробляючи запити та управляючи потоком даних.

БД відповідає за зберігання всієї інформації, яка використовується в системі, включаючи дані про користувачів, сесії, геопросторові дані тощо.

СКБД містить утиліти для управління даними, такі як драйвери *ODM*, які дозволяють серверу взаємодіяти з базою даних, запитувати, оновлювати та обробляти дані.

Клієнтська частина (мобільний додаток) розроблена з використанням *Flutter*, мобільний додаток включає інтерфейси для командира та підлеглих, дозволяючи їм взаємодіяти з системою рекогностування в реальному часі.

Центральна роль в системі відводиться серверу, який обробляє внутрішні та зовнішні запити, управляючи потокомданих, координації сесій рекогностування та управління діями користувачів. Ця багатомодульна архітектура дозволяє гнучко адаптуватися до змінних вимог військового рекогностування та оперативно реагувати на виклики, що виникають під час військових дій.

Кожен модуль в системі має своє чітко визначене призначення:

* модуль створення та редагування користувачів: забезпечує управління аккаунтами всередині системи;
* модуль створення та редагування бойових одиниць: дозволяє командирам планувати та моделювати розгортання підрозділів на території;
* модуль захисту інформації: включає інструменти для забезпечення безпеки даних, відповідно до військових стандартів.

Всі модулі тісно інтегровані через інтерфейси програмування додатків (*API*), що дозволяє забезпечити надійний обмін даними між різними частинами системи. Для внутрішнього спілкування модулів використовуються приватні *API*, що забезпечують безпеку та інкапсуляцію внутрішніх процесів.

База даних та СКБД виконують роль центрального сховища для всієї інформації, що обробляється системою. Вони повинні бути розроблені з урахуванням високих вимог до швидкодії, надійності та масштабованості, а також бути захищеними від несанкціонованого доступу.

Для розробки модуля взаємодії зі сховищем даних може бути використана *JPA*/*Hibernate*, що дозволить абстрагувати логіку взаємодії з базою даних та спростити розробку за рахунок використання об'єктно-орієнтованого підходу до роботи з даними.

## Розробка та опис ключових алгоритмів системи

При розгляді цієї теми потрібно зосередитись на розробці та детальному описі ключових алгоритмів системи інтерактивного рекогностування, що є фундаментальною частиною нашої дипломної роботи. Мета полягає в тому, щоб розробити ефективні та надійні алгоритми, які використовують можливості нашого технологічного стеку: *Flutter* для мобільного додатку, *Java* *Spring* для серверної частини, *React* для адміністративної панелі та *MySQL* для управління базою даних.

Алгоритми системи будуть розроблені з урахуванням вимог до швидкості обробки, точності даних, масштабованості та безпеки. Особлива увага буде приділена інтеграції між різними компонентами системи, щоб забезпечити плавну взаємодію та обмін даними. Ми також зосередимо увагу на створенні користувацьких інтерфейсів, які забезпечать зручне та інтуїтивно зрозуміле використання системи. Ця тема має важливе значення для розуміння технічної реалізації проекту та його практичної придатності в контексті інтерактивного рекогностування.

. У попередньому розділі було описано деталі розробки бази даних для системи, яка використовує *MySQL* як основу для зберігання та обробки даних. Для ініціалізації та управління базою даних використовувався інструмент *phpMyAdmin*, який є зручним веб-інтерфейсом для адміністрування *MySQL*.

*PhpMyAdmin* дозволяє легко створювати, змінювати, видаляти та управляти структурою та даними бази. Важливою перевагою *phpMyAdmin* є його інтуїтивний інтерфейс, який полегшує роботу з базою даних навіть для користувачів, які не мають глибоких знань в *SQL*.

У контексті проекту використовувалася конфігурація *Docker* *Compose* для створення та управління мультиконтейнерним середовищем, що включало сервіси для серверної частини (*Java* *Spring*), адміністративної панелі (*React*), бази даних (*MySQL*) та *phpMyAdmin*. *Docker* *Compose* дозволяє оркеструвати роботу декількох контейнерів, забезпечуючи їхню інтеграцію та спрощуючи процеси розгортання та тестування.

Особливу увагу слід звернути на налаштування сервісів у *Docker* *Compose* файлі, які визначають параметри підключення та конфігурації для кожного сервісу, забезпечуючи правильну взаємодію між ними. Це включає вказівку портів, змінних оточення та залежностей між сервісами.

Таким чином, використання *phpMyAdmin* у поєднанні з *Docker* *Compose* дозволяє ефективно управляти базою даних та інтегрувати її з іншими компонентами системи, забезпечуючи гнучкість та продуктивність у роботі з даними.

* + 1. Розробка серверної частини

Ініціалізація серверної частини проекту розпочинається з використання *Spring* *Initializr* (рис. 2.9), інструменту, який дозволяє швидко налаштувати та створити проект *Spring* *Boot*. Вибір конфігурації включає встановлення мови програмування *Java*, системи збірки *Maven* та останньої стабільної версії *Spring* *Boot*. Для проекту додаються необхідні залежності: *Spring* *Web* для створення веб-додатків, *Spring* *Data* *JPA* для роботи з базами даних, *Spring* *Security* для забезпечення безпеки, *Lombok* для автоматизації створення шаблонного коду, та *Spring* *Boot* *DevTools* для спрощення розробки.

Структура файлів проекту визначається з урахуванням модульності та чіткого розділення відповідальностей. У пакеті *controller* розміщуються класи контролерів, які обробляють *HTTP*-запити, зокрема *AuthController* для аутентифікації, *ClientController*, *ReconController*, *SessionController*, *UnitsController* для відповідних бізнес-операцій. У пакеті *model* знаходяться класи доменних моделей, такі як *Client*, *Recon*, *Session*, *Unit*, *User*. Пакет *repository* містить інтерфейси *Spring* *Data* *JPA* репозиторіїв для кожної з моделей, які забезпечують взаємодію з базою даних. У пакеті *service* реалізовані класи сервісів, які містять бізнес-логіку та викликаються контролерами. Завдяки цій структурі, проект легко масштабується та підтримується.

Структура файлів (рис. 2.10) у *Spring* *Boot* проекті відіграє ключову роль у забезпеченні чистоти коду та його легкості для розуміння. Вона організована таким чином, щоб сприяти розділенню та незалежності компонентів системи

Правильна організація проекту забезпечує легкість управління залежностями, модульність та легкість впровадження змін, що є особливо важливим у великих та складних системах.

У розробці серверної частини на базі *Spring* *Boot* створення моделей відіграє важливу роль, оскільки вони служать як представлення даних, які будуть оброблятися системою. Моделі визначають структуру, з якою розробники та база даних можуть взаємодіяти, забезпечуючи необхідну абстракцію для виконання бізнес-логіки. Кожна модель відповідає таблиці в базі даних і інкапсулює поля як властивості об'єкта, забезпечуючи строгу типізацію та зручність у валідації даних.

Необхідно створювати і конфігурувати моделі, що відображатимуть сутність у базі даних як наприклад *Client* (рис. 2.11). Клас *Client* слугує як модель у домені додатку і використовує анотації *JPA* для відображення властивостей класу на колонки таблиці в базі даних. Клас інтегрується з системою безпеки *Spring* через імплементацію інтерфейсу *UserDetails*, що дозволяє використовувати об'єкти *Client* для аутентифікації та авторизації.

Далі, анотації, такі як @*Entity*, @*Table*, @*Id*, та @*GeneratedValue*, визначають клас як сутність *JPA* та налаштовують деталі його відображення. Властивості сутності, такі як *login*, *password*, *email*, та *role*, відображаються на відповідні колонки таблиці.

Зв'язки між сутностями реалізовані за допомогою анотацій @*ManyToOne*, @*OneToMany*, та @*ManyToMany*, які встановлюють відносини між клієнтами, сесіями, та іншими сутностями в системі. Це включає в себе як прямі так і обернені відносини, дозволяючи навігувати між пов'язаними даними в обох напрямках.

Кожен екземпляр класу *Client* містить безпекову інформацію, яка використовується *Spring* *Security* для перевірки повноважень користувача. Методи, як *getAuthorities*(), *getPassword*(), та *getUsername*(), надають необхідні дані для системи безпеки.

Важливо також налаштувати відповідні репозиторії, сервіси та контролери для роботи з об'єктами *Client*. Репозиторії будуть використовувати *Spring* *Data* *JPA* для забезпечення доступу до бази даних, сервіси відповідатимуть за бізнес-логіку, а контролери за обробку *HTTP* запитів.

Репозиторії у *Spring* *Boot* дозволяють абстрагувати рівень доступу до даних, забезпечуючи канал комунікації між базою даних та бізнес-логікою. Наприклад, *ClientRepository* (рис. 2.12) є інтерфейсом, який розширює *JpaRepository*, надаючи цілий ряд стандартних методів для взаємодії з базою даних, а також дозволяє додавання власних методів, таких як *findByLogin* або *findByEmail*, для виконання специфічних запитів до бази даних.

Сервіси у *Spring* *Boot* є основним місцем, де виконується бізнес-логіка. Вони використовують методи репозиторіїв для доступу до даних і виконання на них операцій, обробляючи транзакції та логіку прикладного додатку. *SessionService* (рис. 2.13), наприклад, може управляти сесіями користувача, включаючи їх створення, видалення та отримання списку сесій за певними критеріями. Використання анотації @*Transactional* забезпечує, що операції, які змінюють стан бази даних, виконуються у межах транзакції, що дозволяє управління цілісністю даних.

Інтеграція репозиторіїв та сервісів створює міцну основу для розвитку серверної частини додатку, де репозиторії забезпечують прямий доступ до даних, а сервіси керують бізнес-логікою, створюючи абстракцію між джерелом даних та іншими шарами додатку. Цей підхід дозволяє підтримувати код чистим, модульним та легко тестованим.

Контролери в *Spring* *Boot* відповідають за обробку вхідних *HTTP* запитів, взаємодію з сервісами для виконання бізнес-логіки та повернення відповідей клієнтам. *SessionController* (рис. 2.14), наприклад, є компонентом, що відповідає за управління сесіями користувачів. Він використовує сервіс *SessionService* для отримання, створення та видалення сесій.

Основні ендпойнти, які можуть бути визначені в *SessionController*, включають методи для отримання списку сесій, створення нової сесії та її видалення. Для кожної з цих операцій використовуються відповідні *HTTP* методи: *GET* для отримання даних, *POST* для створення нових записів, та *DELETE* для їх видалення.

@*GetMapping*("/*list*") використовується для отримання списку сесій, пов'язаних з авторизованим користувачем. @*PostMapping*("/*create*") забезпечує створення нової сесії, де використовуються дані, передані через *SessionDTO* та файл мапи як частина запиту. @*DeleteMapping*("/*delete*") дозволяє видалення сесії за її ідентифікатором.

Використання анотації @*PreAuthorize* з параметром '*hasAuthority*' дозволяє обмежити доступ до певних дій тільки для користувачів з відповідними ролями, зокрема, для командирів.

Ця конфігурація демонструє, як у *Spring* *Boot* можна легко інтегрувати безпеку, валідацію та бізнес-логіку всередині контролера, що робить його важливою частиною серверного додатка.

Безпека сервера, що використовує бібліотеку *jjwt* для роботи з *JSON* *Web* *Tokens* (*JWT*), важлива для аутентифікації та авторизації користувачів. У методі *appLogin* (рис. 2.15) контролера авторизації реалізовано процес логіну в додаток, де спочатку здійснюється перевірка користувача на основі електронної пошти. Якщо користувача з такою електронною поштою не знайдено, кидається виключення *UsernameNotFoundException*.

Після успішної аутентифікації, що здійснюється за допомогою *authenticationManager*, генерується *JWT*, який потім повертається клієнту. *JWT* містить всю необхідну інформацію про користувача та використовується для подальших запитів до сервера.

У фільтрі безпеки (рис. 2.16), що налаштований в *Spring* *Security*, для кожного запиту спочатку парситься *JWT* з *HTTP* запиту. Якщо токен валідний, здійснюється витяг імені користувача з токена, завантажуються його деталі та створюється об'єкт *UsernamePasswordAuthenticationToken*, який встановлюється у контекст *SecurityContextHolder*. Це дозволяє системі ідентифікувати користувача у подальших запитах.

Такий підхід забезпечує, що кожен запит від аутентифікованого користувача містить валідний токен, який дозволяє серверу ідентифікувати та авторизувати запити, забезпечуючи безпеку взаємодії між клієнтом та сервером.

*Swagger* (рис. 2.17) є потужним інструментом для документації *API*, який дозволяє розробникам легко описувати, виробляти, споживати та візуалізувати *RESTful* веб-сервіси. Використання *Swagger* в проекті *Spring* *Boot* значно спрощує процес тестування та інтеграції *API*, надаючи інтерактивний веб-інтерфейс, де розробники та користувачі можуть переглядати всі доступні ендпойнти, їх параметри, відповіді та навіть відправляти запити безпосередньо через браузер.

Під кінець опису серверної частини можна підкреслити, що була створена міцна та безпечна платформа, яка відповідає всім сучасним вимогам до веб-додатків. Використання *Spring* *Boot* забезпечило ефективну інтеграцію з базою даних *MySQL*, автентифікацію через *Spring* *Security* та *jjwt* для управління *JWT* токенами, що дозволило нам створити надійне *API* для мобільного додатку. Архітектура сервера спроектована таким чином, що може бути легко масштабована і адаптована до майбутніх вимог, а чітко структуровані контролери та сервіси забезпечують високу підтримку коду. Впровадження комплексного тестування та документації є ключем до забезпечення стабільності та ефективності серверної частини.

* + 1. Розробка частини клієнта

Ініціалізація проекту в рамках розробки системи інтерактивного рекогностування здійснювалась у комплексному середовищі, використовуючи *Android* *Studio* та *Xcode* (рис. 2.18), що забезпечували ефективний розробницький процес. Використання *Flutter* як основного *SDK* вимагало точного визначення залежностей та пакетів у файлі *pubspec*.*yaml*, зокрема, таких як *dio* для мережевих запитів, *sqflite* для роботи з базою даних, *geolocator* для визначення геопозиції.

Процес розробки включав в себе як використання емуляторів в *Xcode* для тестування *iOS*-версій, так і застосування фізичних пристроїв для перевірки роботи додатку в реальних умовах. Такий підхід дозволив досягти високої точності та надійності у функціонуванні системи. Ретельно налаштоване середовище розробки, з огляду на різноманітність використовуваних технологій та інструментів, стало ключем до ефективного процесу створення комплексної системи інтерактивного рекогностування.

Структура файлів у *Flutter* проекті для системи інтерактивного рекогностування (рис. 2.19) організована за допомогою *GetX*, який є потужним інструментом для управління станом, роутингу та залежностями в додатках *Flutter*. Використання *GetX* сприяє чистоті коду та спрощує розробку складних додатків.

В директорії *lib*, яка є основною для коду *Dart*, файли поділені за категоріями для легкого доступу та масштабованості:

* *constants* містить всі глобальні константи, які використовуються у додатку;
* *controllers* використовуються для управління станом різних частин додатку, зберігаючи бізнес-логіку відокремлено від *UI*;
* *global*\_*widgets* містить віджети, які можуть бути використані на багатьох екранах у додатку;
* *pages* містить різні екрани або сторінки додатку, кожен з яких є набором віджетів, що формують *UI*.
* *service* визначає логіку взаємодії з зовнішніми сервісами та *API*.

*GetController* використовується для управління станом, забезпечуючи реактивність і відділення логіки від інтерфейсу. *GetX* також надає засоби для простого роутингу та залежностей між сторінками, що полегшує навігацію та управління даними в багатосторінкових додатках.

Спілкування між клієнтською частиною системи інтерактивного рекогностування та сервером реалізовано через протокол *HTTP*, використовуючи бібліотеку *Dio* в контексті *Flutter*. *Dio* - це потужний інструмент для *Flutter* та *Dart*, що дозволяє здійснювати *HTTP*-запити, забезпечуючи легкість в обробці запитів і відповідей, а також управління станом сесії.[10]

Клас *NeoApi* (рис. 2.20), що є розширенням *GetxController*, відповідає за всю логіку мережевого спілкування. Він ініціалізує базові налаштування *Dio* з визначенням базового *URL* та заголовків, необхідних для взаємодії з сервером. Контролер може обробляти аутентифікацію, управління *API*-ключами та виконання *HTTP*-запитів (*GET*, *POST*, *PUT*, *DELETE*), повертаючи відповідні дані або обробляючи помилки.

У разі неполадок або помилок під час запиту, система може відображати користувачеві повідомлення про помилку або перенаправляти його на сторінку помилки, залежно від статус-коду відповіді або наявності мережевого з'єднання. Це забезпечує більш гладкий користувацький досвід, дозволяючи вирішувати проблеми з'єднання та доступності сервера більш ефективно.

Код ретельно організовано, щоб забезпечити чистоту та читабельність, полегшуючи подальше утримання та розширення системи рекогностування.

Сторінка доповненої реальності (рис. 2.21) у додатку виступає як візуальний інтерфейс для взаємодії користувача з тривимірними моделями та реальним середовищем через камеру пристрою. Використовуючи *arkit*\_*plugin*, сторінка керує сесією доповненої реальності та обробляє різноманітні взаємодії, такі як тапи, щипки та переміщення по 3*D* об'єктах.

Основні аспекти роботи сторінки *AR*:

* ініціалізація *ARKitController*, який відповідає за управління сесією доповненої реальності та користувацькі взаємодії;
* методи завантаження та зберігання зображень та 3*D* моделей, які використовуються у *AR* сцені;
* елементи інтерфейсу, що забезпечують взаємодію користувача, наприклад, кнопки для зйомки зображень або повернення до попередніх екранів;
* умовне відображення залежно від стану додатку, наприклад, показ індикатора завантаження під час підготовки активів або відображення різних опцій залежно від ролі користувача (командир або підлеглий);
* обробка помилок для елегантного управління непередбачуваними ситуаціями під час *AR* враження, включаючи проблеми з'єднання або завантаження активів.

Структура та поведінка сторінки *AR* розроблені таким чином, щоб забезпечити неперервний і занурювальний досвід доповненої реальності, адаптований до потреб системи інтерактивного рекогностування.

Сторінка *AR* у системі інтерактивного рекогностування є відповідальною за ключові аспекти візуалізації та інтеракції з бойовими одиницями. Вона адаптована до ролей користувачів: командир має можливість обробляти квадрат сесії, розміщуючи на ньому тривимірні моделі бойових одиниць. Крім того, результати рекогностування підлеглих автоматично інтегруються на карту у вигляді аналогічних 3*D* моделей. Для підлеглих ця сторінка використовується для додавання бойових одиниць до доповіді, причому моделі представлені в більшому масштабі для кращої оцінки місцевості.

Використання бібліотеки *arkit*\_*plugin* було визнано оптимальним для реалізації функцій доповненої реальності, проте зіткнулися з відсутністю необхідного функціоналу для повного втілення можливостей інтерактивного рекогностування. Зокрема, стандартні методи слухачів не надавали координати площини *AR* мапи. Вирішенням стало використання методу *onArTap* для пошуку об'єктів у просторі. Подальші відносні відстані між мапою та об'єктом визначалися від користувача, дозволяючи прикріплювати їх один до одного. Для визначення геолокації 3*D* об'єктів застосовувалася аналогічна техніка: від початкової геолокації підлеглого додавалася певна відстань від об'єкта, що дозволяло точно розмістити його на 3*D* карті командира з урахуванням масштабу та розмірів карти.

Де *s* – коефіцієнт масштабування об'єкта в *AR*, а – відстань від камери до об'єкта.

## Вибір апаратного забезпечення для використання системи

Вибір апаратного забезпечення для системи інтерактивного рекогностування має вирішальне значення для забезпечення ефективності та надійності всієї системи.[11] Цей процес містить аналіз ряду технічних характеристик та потреб користувачів, щоб забезпечити відповідність між можливостями обладнання та вимогами, які ставить перед ними військове застосування. Враховуються такі аспекти, як обчислювальна потужність, графічна продуктивність, автономність батареї, здатність працювати в складних погодних умовах та сумісність із різними системами зв'язку та датчиками.

Сучасні мобільні пристрої, які використовуються для доповненої реальності, повинні мати високоякісні камери для точного відтворення реального середовища, високопродуктивні процесори для обробки *AR*-сцен у реальному часі, а також спеціалізоване програмне забезпечення, що може ефективно інтегруватися з обраною платформою розробки. Обладнання має бути достатньо легким та зручним для тривалого використання на полі бою, а також міцним, щоб витримувати екстремальні умови використання.

В процесі вибору апаратного забезпечення важливо також забезпечити його масштабованість та гнучкість для можливих майбутніх оновлень системи. Від цього залежить, наскільки швидко та ефективно система зможе адаптуватися до змінних вимог військових та технологічних інновацій. Кожен компонент системи, від сенсорів та процесорів до дисплеїв та засобів зв'язку, має бути обраний з урахуванням його ролі в загальній архітектурі та його здатності до інтеграції з іншими елементами системи.

Використання бібліотеки *arkit*\_*plugin* для розробки додатку з доповненою реальністю призводить до обмеження доступності додатку лише на пристроях *Apple* *iPhone*. Ця бібліотека спеціально розроблена для використання можливостей *ARKit*, які є ексклюзивними для *iOS*. Отже, вибір апаратної платформи важливо враховувати ще на початкових етапах планування та проектування системи.

У контексті вибору *iPhone* як основного пристрою для системи інтерактивного рекогностування, слід врахувати наступні мінімальні вимоги: високопродуктивний процесор для обробки *AR* сцен, камера з високою роздільною здатністю для точного відображення місцевості, достатній обсяг оперативної та вбудованої пам'яті для збереження великих об'ємів даних, а також потужна батарея для тривалої автономної роботи.

У підсумку мінімальні вимоги до пристрою включають крайні покоління моделі *iPhone*, які підтримують останню версію *iOS* та мають доступ до оновлень *ARKit*, забезпечуючи тим самим гладке та ефективне використання системи інтерактивного рекогностування.

## **Висновки до розділу 2**

У другому розділі було розглянуто процеси та методології, що лежать в основі розробки підсистеми інтерактивного рекогностування. Починаючи від проектування архітектури системи, було визначено як ключові компоненти технологічного стеку (*Flutter*, *Java* *Spring*, *React*, *MySQL*) взаємодіють між собою, створюючи синергію для ефективної роботи системи.

Розроблені алгоритми демонструють глибоке розуміння задач інтерактивного рекогностування та планування розгортання військ. Особлива увага була приділена забезпеченню безпеки, надійності та ефективності обробки даних, що є критично важливими для військових додатків.

Враховано важливість забезпечення гнучкості системи для її адаптації до різних військових сценаріїв та умов. Це включає можливість масштабування та інтеграції з іншими системами та технологіями.

У процесі розробки підсистеми інтерактивного рекогностування вирішено ряд ключових проблем, які значно покращують її функціональність. Однією з таких проблем було точне визначення координат об'єктів у реальному часі. Завдяки інтеграції сучасних технологій геолокації та доповненої реальності, досягнуто високої точності у визначенні географічного розташування, що критично важливо для військових операцій.

На основі аналізу існуючих систем та врахування специфічних військових потреб, розроблено структурну схему нашої системи, яка оптимально відповідає на виклики сучасного поля бою. Ця схема включає елементи для обробки великих даних, аналізу зображень та взаємодії з користувачем, забезпечуючи високий рівень адаптивності та швидкості реакції.

Завершальним етапом стала інтеграція розроблених алгоритмів з системою. Було забезпечено сумісність різних модулів та впровадили протоколи безпеки, що гарантують надійність та конфіденційність інформації. Завдяки цьому, система стала здатною ефективно обробляти складні задачі інтерактивного рекогностування, що є ключовим елементом для успішного планування та ведення військових операцій.

Загалом, розробка цієї підсистеми відіграє важливу роль у підвищенні ефективності та точності військових операцій, допомагаючи покращити прийняття рішень та забезпечуючи командам перевагу в сучасних військових умовах.