НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ПОГОДЖЕНО **Декан факультету**	**ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ** **Завідувач кафедри інформаційних систем і технологій**
_____ **Ігор БОЛБОТ** (підпис)	_____ **Михайло ШВИДЕНКО** (підпис)
«___» _____ 2026 р.	«___» _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка підсистеми безпеки інтегрованої мобільної системи підтримки студентів НУБІП України**»**

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Гарант освітньої програми д.е.н., професор	_____ (підпис)	**Роман РУДЕНСЬКИЙ**
Керівник бакалаврської кваліфікаційної роботи д.ф. з к.н.	_____ (підпис)	**Ярослав ПОНЗЕЛЬ**
Консультант к.п.н., доцент	_____ (підпис)	**Тетяна ВОЛОШИНА**
Виконав	_____ (підпис)	**Владислав ГЛУЩЕНКО**

Київ-2026

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних систем і технологій

к.е.н., доцент _____ Михайло ШВИДЕНКО
(підпис)

«___» _____ 2026 р.

ЗАВДАННЯ ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ ЗДОБУВАЧУ

Глущенку Владиславу Руслановичу

Спеціальність 122 – «Комп'ютерні науки»

ОП «Комп'ютерні науки»

Тема бакалаврської кваліфікаційної роботи «Розробка підсистеми безпеки інтегрованої мобільної системи підтримки студентів НУБІП України» затверджена наказом від «06» січня 2026 р. № 46 «С»

Термін подання завершеної роботи на кафедру «25» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту):

Дані про укриття НУБіП; інтеграція з API сервісів оповіщення про повітряну тривогу; використання Google Maps API для візуалізації маршрутів та реалізація алгоритмів пошуку найкоротшого шляху до сховища.

Перелік питань, що підлягають дослідженню:

аналіз загроз та вимог до систем оповіщення студентів в умовах воєнного стану; проєктування інформаційного забезпечення системи; розробка програмного забезпечення для візуалізації мапи небезпек та автоматичного прокладання маршрутів до укриттів; тестування ефективності системи та оцінка швидкості доставки критичних сповіщень.

Дата видачі завдання «06» січня 2026 р.

Керівник бакалаврської кваліфікаційної роботи _____ (підпис) **Ярослав ПОНЗЕЛЬ**

Консультант _____ (підпис) **Тетяна ВОЛОШИНА**

Завдання взяв до виконання _____ **Владислав ГЛУЩЕНКО**

(підпис)

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – інтерфейс прикладного програмування.

GPS – система глобального позиціонування (необхідна для роботи з мапами та локацією студента).

GIS – географічна інформаційна система.

UI – інтерфейс користувача.

UX – досвід користувача.

SQL – мова структурованих запитів (для роботи з базою даних Supabase/PostgreSQL).

DB – база даних.

ER-діаграма – діаграма «сутність-зв’язок».

UML – уніфікована мова моделювання програмних систем.

JSON – текстовий формат обміну даними (використовується в API та JWT).

JWT (JSON Web Token) – токен у форматі JSON для безпечної автентифікації та авторизації.

REST – архітектурний стиль взаємодії компонентів системи.

SDK – набір засобів розробки програмного забезпечення (актуально для React Native та Expo).

HTTPS – захищений протокол передачі гіпертексту.

PUSH – технологія надсилання коротких повідомлень (для сповіщень про тривоги).

НУБіП – Національний університет біоресурсів і природокористування України.

Row Level Security (RLS) – це механізм безпеки на рівні бази даних (зокрема у PostgreSQL), який дозволяє розмежувати доступ до даних не на рівні всієї таблиці, а на рівні кожного окремого рядка.

BaaS – Backend-as-a-Service (Бекенд як послуга) Хмарна модель для розробників програмного забезпечення, що дозволяє делегувати створення та підтримку

серверної частини (бази даних, автентифікацію, хмарні функції) стороннім провайдерам.

RBAC – Role-Based Access Control (керування доступом на основі ролей).

OAuth – відкритий протокол авторизації.

SSO – Single Sign-On (технологія єдиного входу).

PostGIS – розширення PostgreSQL для роботи з географічними об’єктами.

EAS – Expo Application Services.

OTA – Over-The-Air updates (механізм оновлення додатка без повторної публікації).

WebSocket – протокол повнодуплексного зв’язку між клієнтом і сервером.

Edge Function – серверна функція Supabase, що виконується на краю мережі.

Realtime – технологія Supabase для миттєвої синхронізації даних.

APK – Android Package Kit (інсталяційний пакет Android).

CRUD – Create, Read, Update, Delete (основні операції з даними).

FCM (Firebase Cloud Messaging**)** – безкоштовний хмарний сервіс від Google, який дозволяє розробникам надсилати push-сповіщення та повідомлення на пристрої з Android, iOS та в браузери.

ВСТУП

Актуальність теми. В умовах повномасштабного військового вторгнення та постійних загроз безпеці, питання оперативного інформування студентів закладу вищої освіти набуло першочергового значення [1]. Національний університет біоресурсів і природокористування України має розгалужену інфраструктуру. Його територія налічує значну кількість об’єктів цивільного захисту. Тому критично важливою є наявність цифрового інструменту, що забезпечує швидку навігацію до безпечних місць. Актуальність роботи зумовлена необхідністю цифровізації процесів оповіщення та інтеграції сервісів для прокладання маршрутів до укриттів. Це потребує об’єднання в межах єдиної системи для мінімізації загроз життю та здоров’ю студентів.

Мета розробки. Метою бакалаврської роботи є створення мобільної підсистеми безпеки, в межах інтегрованої системи підтримки студентів. Вона забезпечуватиме автоматичне отримання сигналів тривоги, відображення мапи повітряних небезпек, доступ до переліку укриттів університету та побудову найкоротших маршрутів до них з урахуванням поточного розташування користувача.

Методи та технології розробки. Для досягнення поставленої мети використовувалися методи об’єктно-орієнтованого аналізу, моделювання систем за допомогою мови UML та технології інтеграції інформаційних і картографічних сервісів.

- фреймворк React Native та платформу Expo для створення кросплатформного мобільного додатка;
- хмарну платформу Supabase (PostgreSQL) для управління базою даних, автентифікації та push-сповіщень;
- картографічні сервіси Google Maps API для візуалізації об’єктів та побудови маршрутів.
- API сервісу alerts.in.ua alerts.in.ua для отримання актуальних даних

про повітряні тривоги та відображення мапи повітряних небезпек.

Апробація результатів. Основні результати дослідження та технічні рішення були апробовані на VII Всеукраїнській науково-практичній конференції студентів та аспірантів «Теоретичні та прикладні аспекти розробки комп'ютерних систем 2026», де робота була представлена в межах секції 2 «Технології розробки програмного забезпечення та інформаційних управляючих систем».

Структура пояснювальної записки. Робота складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Загальний обсяг пояснювальної записки становить 68 сторінок. Робота містить 22 рисунка та 2 таблиці. Список використаних джерел налічує 12 найменувань.

Короткий опис розділів:

- У першому розділі проведено системний аналіз предметної області, досліджено проблеми безпеки студентів у воєнний час, проаналізовано існуючі аналоги та сформульовано вимоги до програмного продукту.
- У другому розділі виконано проєктування архітектури системи, розроблено логічну та фізичну моделі бази даних, а також змодельовано бізнес-процеси за допомогою UML-діаграм.
- У третьому розділі описано процес програмної реалізації підсистеми, включаючи налаштування серверної частини, розробку мобільного інтерфейсу та інтеграцію з картографічними сервісами.
- У четвертому розділі наведено методику та результати тестування функціональних модулів системи, оцінено швидкість доставки сповіщень та зручність користувацького інтерфейсу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Актуальність теми та проблеми оперативного інформування студентів в умовах воєнного стану

В умовах повномасштабної війни питання безпеки студентів у вишах набуло особливої гостроти. Територія НУБіП України велика, корпусів і гуртожитків багато, тому під час тривоги швидко зібрати людей і знайти найближче укриття – непросте завдання. Державні системи оповіщення, хоч і функціонують, але погано адаптовані до роботи всередині великого університетського кампусу. Вони не враховують точне розташування студента та не вказують найближче перевірене укриття. Часто важлива інформація про безпечні маршрути лежить у застарілому форматі: плакати на стінах або сторінки сайту, доступ до яких у хвилюванні та поспіху надто складний. Розрив між потоками даних та реальною необхідністю швидко діяти робить стандартні способи передачі повідомлень складними в момент постійного стресу.

Замість багатьох окремих сервісів був потрібен один додаток, де зібрано все для безпеки. Через мапу повітряних тривог кожен користувач зможе спостерігати за тим, що відбувається скрізь по країні, не тільки поруч. Це допомагає швидше приймати рішення, якщо знадобиться. Сповіщення доходять миттєво, бо надсилаються прямо на телефон. Такий механізм скорочує затримки між сигналом і реакцією. Повідомлення поєднуються з функцією навігації - це основна суть системи. Коли починається тривога, дуже важливо також мати актуальний список укриттів. Користувач може самостійно обирати маршрут до укриття, але автоматична побудова, яка враховує відстань і доступність є набагато зручнішим варіантом. Нема потреби шукати входи чи орієнтуватися самостійно. У додатку передбачено інтеграцію зазначених функцій.

Відсутність єдиного цифрового інструменту уповільнює реакцію на

надзвичайні ситуації та посилює тривогу серед студентів і викладачів, яким у стресових умовах важко швидко знайти оптимальний шлях до укриття. Замість того щоб покладатися на традиційні методи, багато молодих людей очікують моментального доступу до корисних функцій прямо на своєму телефоні – саме тому персоналізовані повідомлення й навігація стають логічним рішенням. Через такий підхід система безпеки набуває не просто технічного значення, а формування впевненості студента у діях закладу під час кризи. Наявність чітких маршрутів і зв'язку між екстреними службами та додатком робить взаємодію прозорою, хоча головне – скорочення часу реакції. Людські помилки в таких умовах мають менше шансів завдати шкоди, коли технологія береться за основну частину роботи.

1.2 Порівняльний аналіз програмних аналогів та існуючих рішень у сфері цивільної безпеки

Вивчення поточного ринку програм для телефонів показало: немає жодної системи, створеної спеціально для безпеки й орієнтації студентів у межах одного університету під час війни. Деякі додатки допомагають стежити за небезпеками або будувати маршрути, проте жоден із них не поєднує ці можливості з детальним списком укриттів саме на території університету. Наявні альтернативи пропонують часткові рішення, тим не менш їхній загальний недолік – слабка адаптація до складної структури великих навчальних територій. Кожна з них працює окремо, через що користувач сам повинен поєднувати дані з кількох джерел, що ускладнює реакцію на загрози.

Перш за все – найчастіше люди користуються додатком «Повітряна тривога». Швидкість оповіщення там значно вища, адже воно працює разом із офіційними службами цивільного захисту [2]. Однак можливості програми закінчуються на простому повідомленні: почалася чи минула небезпека. Немає взаємодії з картою, немає можливості прокласти маршрут до укриття. Тож користувач просто отримує сигнал, але не знає, куди рухатися далі. Цей

інструмент не допомагає студентам орієнтуватися під час загрози – він лише передає факт, не даючи жодних подальших кроків.

Серед корисних інструментів – платформа «Київ Цифровий» з мапою сховищ столиці. Незважаючи на зручний дизайн і наявність фільтрів, ці мапи охоплюють лише загальноміський масштаб. Усередині системи багато зайвої інформації, при цьому без урахування готовності укриттів чи особливостей входу до приміщень під час тривог.

Хоча популярні карти, наприклад Google Maps, добре справляються з побудовою маршруту, доступу до систем попередження та інформації про бомбосховища немає взагалі. Такі сервіси просто не пов'язані з локальними структурами цивільного захисту у закладах освіти. Наслідком стає поширення неточної географічної інформації серед студентів та персоналу.

Існуючі системи оповіщення не повністю враховують особливості інфраструктури університетського кампусу. Потрібна окрема система, з’єднана з картою загроз, регіональними даними, швидкими повідомленнями та «розумною» маршрутизацією.

1.3 Визначення цілей та завдань дослідження

Основна мета – створити мобільний сервіс, спрямований на допомогу академічній спільноті, де автоматика передаватиме сигнали про небезпеку й показуватиме шлях до найближчих сховищ. Досліджуються способи управління інформацією та підтримки студентів у складних умовах, особливо коли починається тривога. А саме аналізуються конкретні методи, правила роботи програм і технічні інструменти для створення відповідних функцій.

Щоб досягти поставлену мету спочатку було сформульовано технічні завдання, які охоплюють основні етапи створення програмного продукту. Аналіз потенційних загроз та формування технічних умов для систем швидкого сповіщення стали першим кроком. Вони допомагають виділити обов’язковий функціонал під час роботи в нестабільній ситуації. Далі йде побудова

інформаційної основи: проєктуватиметься логічна архітектура бази даних для збереження координат сховищ та налаштування інтеграції з зовнішніми API державних сервісів оповіщення. Завдяки якому також буде додано мапу загроз. Одним із ключових завдань стало створення алгоритму, який автоматично будує маршрут до найближчого доступного укриття за допомогою Google Maps API [8]. Дуже важливо щоб при його розробці було включено особливість того, чи доступне укриття в даний момент. Завершальний етап дослідження є проведення загального тестування ефективності роботи підсистеми, зокрема це оцінка того наскільки стабільно до користувача доходять push-сповіщення, чи коректно система сповіщає користувача за обраним регіоном, та швидкодія алгоритмів навігації. Реалізація цих завдань дозволить створити ефективну підсистему безпеки.

1.4 Формулювання вимог до підсистеми безпеки мобільної системи підтримки студентів

При розробці модуля безпеки кожна деталь має велике значення, оскільки навіть невелика затримка чи помилка може мати серйозні наслідки. Тут має бути визначено чіткий набір операцій, які програма має реалізовувати задля ефективності в питаннях захисту. Замість загального огляду на ситуацію, буде вестись постійна взаємодія з державними системами через API, щоб одразу надсилати попередження на телефон. Важливо щоб користувач сам обирав територію контролю, тож повідомлення будуть стосуватися лише тієї місцевості, де він знаходиться. Також система відображатиме актуальну мапу повітряних тривог по всій Україні, яка оновлюється в реальному часі через відповідне API. В іншій вкладці виділяється список офіційних укриттів при університеті – для кожної точки доступні точні дані (фото входу, місткість та доступність).

Навігаційний модуль може будувати маршрут двома способами: при загрозі у користувача є можливість використати опцію провести шлях до найближчого сховища, інакше можна обрати потрібне укриття з переліку й прокласти до нього

напрямок. Зворотний зв’язок створюється через оцінювання сховищ студентами – люди діляться враженнями, ставлять рейтинг. Ця практика тримає інформацію про стан приміщень свіжою, показує, чи є там все необхідне, і допомагає адміністрації покращити перебування на території бомбосховищ.

Якісні атрибути й технічні обмеження, формують нефункціональні вимоги. Найвищим пріоритетом є швидкодія – затримка реакції інтерфейсу разом із проведенням маршруту має бути найменшим за часом. Безперебійна робота при пікових навантаженнях також є невід’ємною частиною. Робота на декількох платформах одночасно, буде реалізовано завдяки React Native, цей фреймворк забезпечить найбільш стабільну роботу на різних мобільних ОС [3]. Майбутній інтерфейс обов’язково має бути простим та контрастним. В стресі людина швидше помічає яскраве, тому позначенням вкладок будуть реалізовані з яскравих а головне різних кольорів. Окремо буде опрацьовано не менш важливий пункт – безпека даних. Використання хмарних сервісів гарантує захищений доступ до профілів та цілісність бази даних. Також необхідно врахувати ту ситуацію коли мережі немає взагалі – тому частина функцій працюватиме й без інтернету, показуючи, план-схему об’єктів цивільного захисту НУБіП України. Ці вимоги разом створюють чітку рамку для подальших кроків у розробці системи, особливо коли почнеться побудова архітектуру та продумування структури даних.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Вибір архітектурного підходу та технологічного стеку

Створюючи архітектуру для нової мобільної системи – з акцентом на безпеці й швидкому оповіщенні – довелося уважно підбирати технічні інструменти. Основним орієнтиром при виборі стало те, як швидко можна реалізувати проект, при цьому не втративши його якість та працездатність. Заздалегідь продуманими та важливими речами були сумісність з різними платформами разом та гнучкість щодо долучення сторонніх сервісів. Одним із пріоритетів при виконанні бакалаврської роботи була орієнтація на практичну користь для всієї академічної спільноти НУБіП України. Саме тому основу складає клієнт-серверна модель, побудована на сучасних фреймворках із застосуванням хмарних рішень.

Клієнтською частиною застосунку є React Native разом із Expo. Завдяки цьому фреймворку інтерфейс працює майже як нативний, хоча код пишеться один для обох систем: Android і iOS. У контексті університету це особливо доречно – студенти мають телефони на різних платформах. Інтерактивні карти чи оновлювані списки сховищ стають можливими через гнучкість JavaScript та React. Швидкість реагування залишається високою навіть під час активної взаємодії. Expo – це середовище, де тестування стає швидким завдяки додатку Expo Go [4]: оновлені елементи відображаються одразу на телефоні, нема потреби чекати довгої збірки. Завдяки вбудованим модулям, наприклад, для карти або визначення місцезнаходження, можливості взаємодії з оповіщеннями розширюються природним чином, формуючи основу механізму захисту.

Платформа для роботи з даними та серверним функціоналом побудована на основі Supabase – технології типу BaaS [5], що стає зручною заміною класичних backend-рішень. Цей крок спрямований не стільки на тренд, скільки на практичну економію часу: вся інфраструктура запускається майже одразу, без

потреби налаштовувати сервери вручну. Замість довгих процедур входження – готовий блок автентифікації, який дозволяє студентам входити через університетську пошту чи свій Google-профіль буквально за пару секунд. Також всередині вже є система зберігання файлів, де можна розміщувати фото сховищ чи документи. Додаткові сервіси типу Edge Functions допомагають виносити важкі операції на сторону хмарі, коли потрібно щось більше, ніж просто показати дані.

Найважливішим елементом Supabase стає реляційна база даних на основі PostgreSQL [6] – технологія, відома своєю потужністю й надійністю серед СУБД глобального масштабу. Завдяки їй можна організовувати складну архітектуру даних, де кожен студент – жорстко пов’язаний із відповідним регіоном сповіщення, або маршрут 100% проведений до найближчого укриття за найкоротшим шляхом, саме тому, що довгота та широта чітко прописані по регламенту та відповідають певному укриттю. Такий підхід забезпечує точність інформації, не даючи помилкам виникнути навіть під час стресового навантаження. У період повітряної тривоги система продовжує швидко опрацьовувати запити, адже одночасний доступ сотень користувачів не має бути проблемою. Реалізація механізму Realtime всередині PostgreSQL стає ключовою перевагою для систем безпеки: як тільки статус одного пункту укриття змінюється, дані миттєво доходять до кожного студента через живу синхронізацію. Цей функціонал перетворює базу з просто сховища в активний канал комунікації.

Замість класичного стеку тут працює комбінація, що спрощує підключення до сторонніх API – наприклад, офіційних систем повітряної тривоги чи сервісу карт від Google. Один інтерфейс стає центром, куди потрапляють дані з декількох незалежних джерел, поєднуючись логічно й візуально. Технологія BaaS разом із кросплатформеною архітектурою скорочує необхідність догляду за серверами, тож ресурси перетворюються на швидкий розвиток функцій для захисту користувачів.

Завдяки сполученню React Native, Expo і Supabase на основі PostgreSQL

виходить система, що легко адаптується під зростаючі потреби. Працездатний макет реалізується швидко – при цьому фінальний продукт працює без збоїв, особливо коли потрібне точне інформування чи орієнтація в надзвичайних обставинах.

2.2 Логічна і фізична модель бази даних

Проєктування бази даних – важлива частина створення мобільної підсистеми безпеки, адже саме тут треба продумати основу продуктивності, стабільності й подальшого розвитку системи. Замість прямолінійного будування структури даних, без можливості в подальшому розширення та інтегрування в неї сторонніх сервісів, довелося провести великий об’єм роботи над її продуманістю, під цим було закалдено чітко впорядкований набір даних, спрямований на оперативне виявлення небезпек і реагування у критичних станах. Через необхідність працювати з потоками актуальної інформації, враховано переваги реляційної моделі, реалізованої через PostgreSQL. Цей варіант запущено завдяки хмарному сервісу Supabase, який забезпечує гнучке управління даними без зайвих технічних складнощів.

Під час проєктування логічної моделі було проведено детальний системний аналіз предметної області, що дозволило виділити ключові сутності та зв’язки між ними. Забезпечено відповідність третій нормальній формі, через що дані не повторюються зайво, а помилки під час додавання чи зміни інформації практично неможливі. Усередині структури реалізовано механізми входу для користувачів, налаштування параметрів безпеки під особу, фіксацію сигналів про тривогу в режимі реального часу, а ще – контроль просторової інформації про об’єкти цивільного захисту.

Показана на рис. 2.1 модель використовує розширені можливості СУБД PostgreSQL, зокрема механізм перелічуваних типів (Enums) для жорсткого контролю за даними [6]. Такий підхід забезпечує цілісність інформації прямо всередині сервера баз даних, що мінімізує помилки. Наприклад, тип alert_type

визначає категорії небезпек: повітряна тривога, артилерійський обстріл, хімічна чи радіаційна загроза. Кожен користувач ототожнюється роллю через user_role – студент або адміністратор. Стан укриття так само строго регульований: відкритий чи закритий, його доступність фіксується через shelter_status. Ці механізми працюють разом, формуючи чітку логіку валідації.

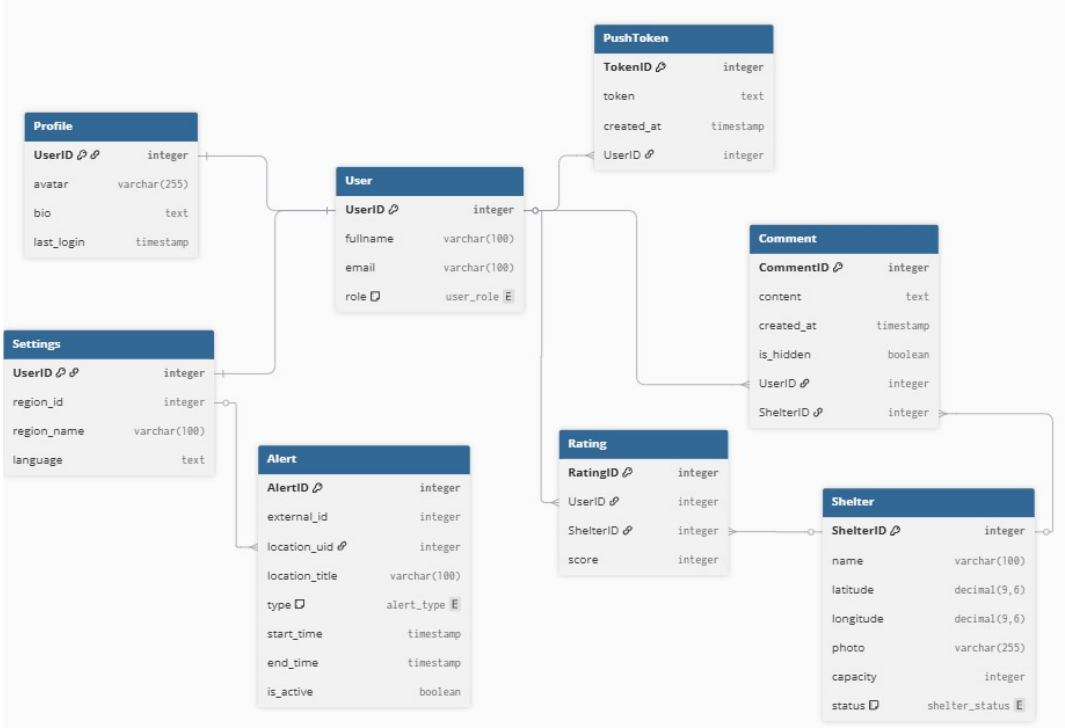

Рис. 2.1 – ER-діаграма підсистеми безпеки

Серце моделі – це система контролю над користувачами, навколо якої побудована структура таблиці User. Усередині неї реєструються основні відомості: повне ім’я та електронна адреса без повторень. Швидкість роботи й чітке поділення потоків даних досягаються шляхом перенесення особистих налаштувань і профільних деталей у самостійні сегменти – Profile і Settings. Кожен із них прикріплений до користувача через одиничний зв'язок (1:1), де ключем служить ID користувача. Особливо важливим виявився блок Settings у контексті захисту системи: там зберігається регіон за специфікацією API (region_id). Окрім цього, до сутності користувача прив'язано таблицю PushToken за допомогою зв'язку один до багатьох. Це дозволить зберігати токени з кількох різних пристроїв, одного і того ж користувача.

У таблиці Alert налаштовано відстеження небезпек у поточному режимі. Її

структура включає поле external_id – воно показує порядковий номер тривоги із офіційної платформами Alerts.in.ua. Зв’язок із Settings побудовано за кодом регіону, це робить можливим швидке виявлення потрібних користувачів для надсилання їм повідомлень.

У таблиці Shelter реалізовано географічні дані через поля latitude й longitude – формат DECIMAL(9,6) забезпечує потрібну детальність. Така точність важлива, щоб інтеграція з Google Maps API працювала без ускладнень. Маршрутизація до сховищ стає надійнішою завдяки цьому підходу. Крім локацій, у записах фіксують значення capacity – скільки людей може перебувати всередині. Стан кожного пункту також враховується: поле status показує, чи доступне сховище наразі.

Через таблиці Comment і Rating реалізовано соціальну взаємодію разом із механізмами зворотного зв’язку. Зв’язки типу один до багатьох з’єднують їх із користувачами й укриттями. У цих таблицях унікальна пара (UserID, ShelterID) запобігає повторним оцінкам одного й того самого укриття. Таке рішення допомагає підвищити чесність системи рейтингування. На ER-діаграмі видно, як всі елементи сполучаються між собою. Логічна структура підтримує надійний захист студентської спільноти.

2.3 Моделювання бізнес-процесів за допомогою UML-діаграм

На етапі проєктування підсистеми безпеки було критично важливо не лише визначити архітектурні рішення та технологічний стек, а й детально спрогнозувати поведінку системи в умовах реальних загроз. Згідно з завданням додаток мав допомогти викликати у користувача миттєву реакцію на ситуацію, тому що прийняття рішень в стресових ситуаціях, як показує загальний досвід – є важким. Для вирішення цього завдання було використано UML (Unified Modeling Language) – уніфіковану мову моделювання, яка дозволяє візуально відобразити логіку взаємодії компонентів, послідовність подій та розподіл ролей акторів.

У процесі розробки підсистеми UML-діаграми відіграли ключову роль у впорядкуванні сценаріїв, де задіяно взаємодію з державними API сповіщень (alerts.in.ua), картографічними сервісами Google Maps та хмарною інфраструктурою Supabase. Завдяки візуальному моделюванню вдалося сформулювати цілісну картину роботи системи ще до початку етапу активної розробки коду. Це дало змогу виявити потенційні «вузькі місця» у швидкості обробки push-сповіщень, спрогнозувати логіку автоматичного вибору найближчого укриття на основі GPS – координат та заздалегідь продумати механізми обробки виняткових ситуацій.

У наступних підпунктах наведено основні UML-діаграми, що були розроблені для проєктування підсистеми безпеки. Зокрема, було побудовано:

- Діаграму прецедентів, яка визначає функціональні можливості для студента (перегляд мапи тривог, вибір регіону, оцінювання укриттів) та адміністратора (модерація даних про безпечні місця);
- Діаграму діяльності, що візуалізує алгоритм дій від моменту отримання сигналу тривоги через API до побудови оптимального маршруту до обраного укриття;
- Діаграму послідовностей, яка демонструє часову взаємодію між мобільним клієнтом на React Native та серверними функціями Supabase під час синхронізації статусів тривоги;
- Діаграму розгортання, що описує фізичне розміщення компонентів системи на смартфонах користувачів та взаємодію з хмарною базою даних PostgreSQL.

Кожна з цих діаграм виконує свою специфічну функцію: одні фокусуються на поведінкових аспектах підсистеми, інші – на її структурі. У сукупності вони дозволили представити архітектуру розробки не як набір ізольованих функцій, а як злагоджений механізм, де кожен елемент (від вибору регіону до системи відгуків) має чітко визначене місце та роль у забезпеченні цивільного захисту студентів НУБіП України.

2.3.1 Діаграма прецедентів

На початковому етапі проєктування ключове значення має визначення функціональних меж системи та ролей, які взаємодіють із нею. Для візуалізації цих вимог було побудовано діаграму прецедентів (Use Case Diagram), яка дозволяє представити систему як сукупність дій (прецедентів) та дійових осіб (акторів). Дана діаграма є фундаментальним інструментом для опису того, що саме робить система, не заглиблюючись у технічні деталі реалізації як вона це робить.

Згідно з розробленою діаграмою (рис. 2.2), у системі виділено три основні актори:

1. Студент / Викладач НУБіП – основний споживач сервісів, чиєю головною метою є отримання оперативної інформації для збереження власного життя та здоров’я.
2. Диспетчер НУБіП – адміністративний актор, відповідальний за підтримання актуальності даних про безпечні локації та модерацію контенту.
3. API-мапа тривог – зовнішня інтелектуальна система, що виступає джерелом потокових даних про повітряні загрози в реальному часі.

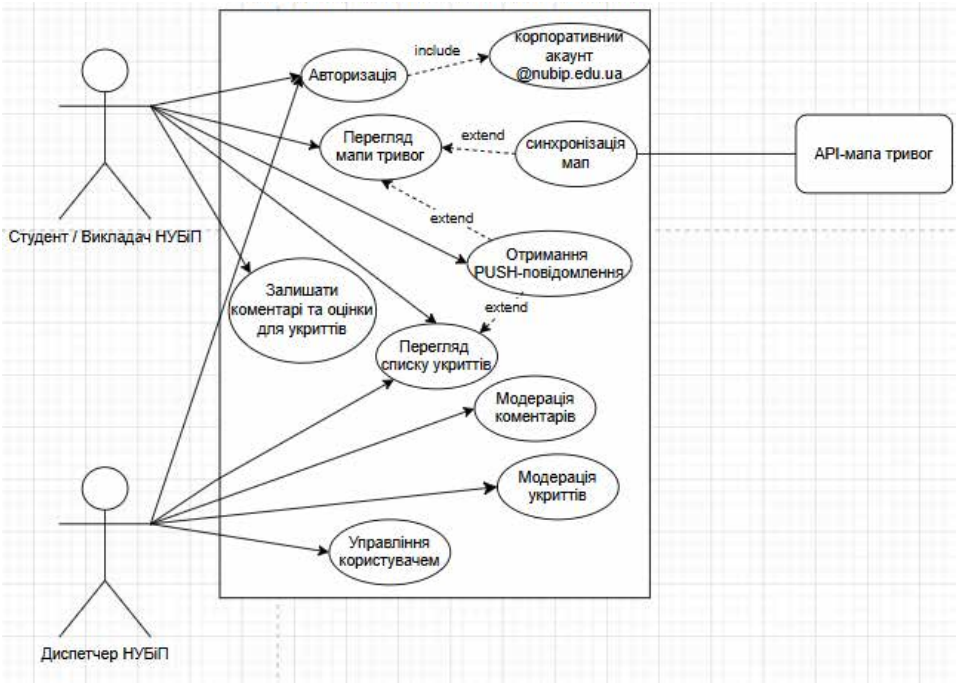

Рис. 2.2 – Діаграма прецедентів підсистеми безпеки

Центральним елементом функціональності для користувача є процес авторизації. Важливою особливістю розробки є використання відношення «include» (включає) для зв’язку з корпоративним акаунтом @nubip.edu.ua. Це гарантує, що доступ до внутрішніх сервісів університету (наприклад, до детальних планів корпусних укриттів) мають лише верифіковані члени університетської спільноти, що підвищує рівень кібербезпеки системи.

Прецедент «Перегляд мапи тривог» забезпечує візуалізацію поточної ситуації в Україні. Він має розширення («extend») – «синхронізація мап», що реалізовано через взаємодію з хмарним бекендом. Такий підхід дозволяє системі працювати динамічно: щойно статус загрози змінюється на сервері alerts.in.ua, дані оновлюються на сервері, а хмарний сервіс Supabase Realtime за допомогою протоколу WebSockets миттєво передає ці зміни в інтерфейс мобільного застосунку. Крім того, передбачено функцію «Отримання PUSH-повідомлення», яка також розширює базові сценарії перегляду, забезпечуючи проактивне інформування студента через сервіс Expo Push Notifications, навіть якщо застосунок не є активним у цей момент.

Для забезпечення навігації в межах кампусу реалізовано прецедент «Перегляд списку укриттів». Він доступний обом категоріям акторів, проте логіка використання відрізняється. Студенти використовують список для пошуку найближчої точки захисту, тоді як диспетчер має розширений функціонал – «Модерація укриттів». Це дозволяє адміністрації оперативно змінювати статус об’єкта (наприклад, тимчасове закриття на ремонт або зміна місткості).

Соціальний аспект системи реалізовано через можливість «Залишати коментарі та оцінки для укриттів». Цей прецедент створює механізм краудсорсингу: студенти можуть повідомляти про реальний стан сховища (наявність води, вільних місць, стан вентиляції). Для запобігання поширенню неправдивої інформації або нецензурного контенту, дії користувача підлягають модерації коментарів, яку виконує диспетчер.

Прецедент «Управління користувачем» з боку диспетчера в даній архітектурі зосереджений на адмініструванні облікових записів на рівні бази

даних. Це включає перевірку коректності реєстраційних даних, можливість коригування користувацьких налаштувань (наприклад, вибір регіону) у випадках технічних збоїв. Таким чином, діаграма прецедентів відображає збалансоване поєднання автоматизованих функцій моніторингу та інструментів ручного адміністрування, що разом створюють надійну систему підтримки безпеки університетської спільноти.

2.3.2 Діаграма діяльності

Для детального опису динамічної поведінки підсистеми безпеки та візуалізації алгоритмів взаємодії користувача з програмним продуктом було розроблено діаграми діяльності (Activity Diagrams). На відміну від статичних структур, цей вид моделювання дозволяє простежити послідовність виконання операцій, логічні розгалуження та реакцію системи на зовнішні події. У межах проєкту було виділено два ключові сценарії використання: плановий пошук інформації про укриття та оперативне реагування на надзвичайну ситуацію.

Сценарій вибору укриття та надання зворотного зв’язку. Перша діаграма діяльності (рис. 2.3) описує процес роботи користувача з інтерактивним списком об’єктів цивільного захисту.

Рис. 2.3 – Діаграма діяльності підсистеми безпеки (Процес пошуку укриття через список)

Цей сценарій передбачає підготовчий етап, коли студент має можливість ознайомитися з інфраструктурою безпеки кампусу НУБіП у спокійних умовах. Процес ініціюється дією «Відкрити список укриттів», що активує запит до бази даних для виведення доступних локацій. Після етапу «Переглянути інформацію про укриття» програма переходить до візуалізації детальних даних. Тут користувач отримує доступ до коментарів та фотоматеріалів.

Логічне розгалуження на діаграмі дозволяє обрати один із двох шляхів:

1. Навігаційний: дія «Прокласти маршрут до укриття» з наступним переходом до інтерфейсу карти.
2. Інформаційний: дія «Залишити коментар та оцінку». Важливою особливістю моделі є наявність циклічного зв’язку: після публікації відгуку потік повертається до екрана інформації, що дозволяє

користувачеві продовжити взаємодію з об’єктом або змінити свій вибір.

Друга діаграма (рис. 2.4) моделює критично важливий процес – напів-автоматизоване реагування на загрозу. Цей алгоритм є пріоритетним для підсистеми безпеки, оскільки він мінімізує час прийняття рішення в умовах стресу. Цикл діяльності починається з фонового процесу «Отримання сигналу тривоги (API)», який ініціює «Відправку PUSH-сповіщення». Ключовим моментом є перехід до дії «Користувач відкриває додаток через сповіщення», що дозволяє швидше перейти до вкладки сповіщення і провести найближчий маршрут.

Рис. 2.4 – Діаграма діяльності підсистеми безпеки (Процес оперативного реагування на тривогу)

Центральним елементом діаграми є вузол прийняття рішення щодо

доступу до геоданих пристрою. Модель передбачає два варіанти розвитку подій:

- У разі надання доступу (гілка «так»): система виконує «Визначення координат студента», після чого проводиться автоматичний «Пошук найближчого укриття в базі». Кінцевим етапом є «Побудова та відображення маршруту», що забезпечує пряму навігацію до безпечної зони без зайвих маніпуляцій з інтерфейсом.
- У разі відмови (гілка «ні»): система генерує повідомлення про помилку «Користувач не надав доступ до геоданих», що сигналізує про необхідність переходу до ручного вибору локації.

Представлені моделі діяльності дозволили оптимізувати архітектуру додатка, зробивши її адаптивною до потреб користувача. Використання паралельних сценаріїв гарантує, що підсистема безпеки виконує як інформаційну функцію (довідник укриттів), так і роль активного помічника в екстрених ситуаціях, забезпечуючи швидкий доступ до навігаційних сервісів університету.

2.3.3 Діаграма послідовностей

Для опису динамічної взаємодії між компонентами системи в часовому розрізі було розроблено діаграму послідовностей (Sequence Diagram). На відміну від діаграм діяльності, цей вид моделювання акцентує увагу на обміні повідомленнями між об'єктами та логічній черговості запитів. Діаграма послідовностей дозволяє детально простежити шлях проходження даних від інтерфейсу користувача до зовнішніх сервісів та бази даних.

Згідно з розробленою моделлю (рис. 2.5), у взаємодії беруть участь п’ять основних об'єктів: Студент, Мобільний застосунок (React Native клієнт), Провайдер Google OAuth2 (Supabase Auth), Провайдер даних про тривоги (Alarm Provider) та БД (хмарна інфраструктура Supabase).

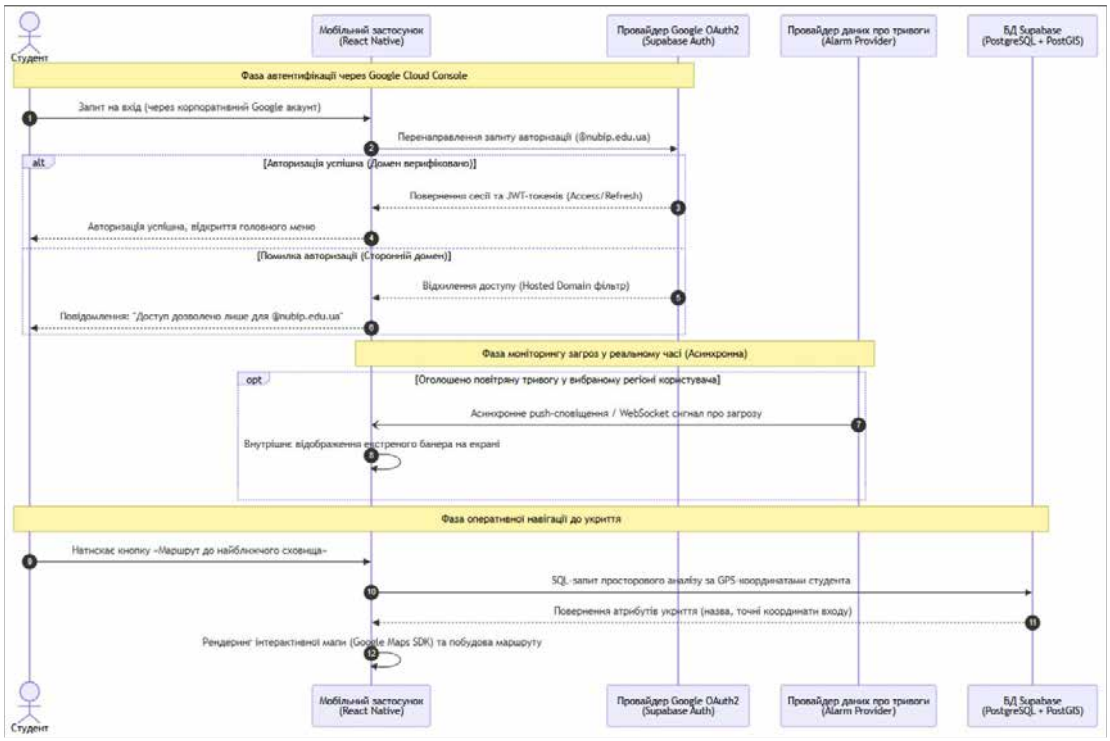

Рис. 2.5 – Діаграма послідовностей підсистеми безпеки

Процес взаємодії структуровано за наступними фазами:

Фаза автентифікації через Google OAuth: Студент ініціює запит на вхід через мобільний застосунок. Оскільки система використовує інтегрований механізм Google Auth, запит перенаправляється до зовнішнього провайдера [11]. Користувач авторизується за допомогою корпоративної пошти @nubip.edu.ua. Після успішної перевірки Google повертає токен підтвердження застосунку, який передає його до Supabase. Серверна частина Supabase верифікує токен, створює сесію та надає доступ до захищених даних (укриттів та налаштувань).

Фаза моніторингу статусу безпеки: Після отримання доступу застосунок звертається до API сервісу повітряних тривог для синхронізації актуальних даних у реальному часі. Отримавши сигнал про початок тривоги в конкретному регіоні, система обробляє ці дані та миттєво виводить екстрене сповіщення на екран смартфона користувача.

Фаза оперативного пошуку та навігації: Студент активує функцію побудови маршруту. Застосунок надсилає запит до бази даних (PostgreSQL через Supabase) із поточними геокоординатами. База даних виконує пошуковий алгоритм,

ідентифікує найближче укриття університету та повертає його атрибути (назву, координати входу та опис). Завершується послідовність відображенням інтерактивної карти з прокладеним маршрутом.

Використання такої послідовності дозволяє забезпечити високий рівень безпеки даних за рахунок делегування автентифікації надійному провайдеру (Google), водночас зберігаючи високу швидкість реагування підсистеми в моменти критичного навантаження.

2.3.4 Діаграма розгортання

Для візуалізації фізичної структури системи, конфігурації апаратних засобів та розподілу програмних модулів між вузлами мережі було розроблено діаграму розгортання, яку наведено на рис. 2.6. Дана модель дозволяє детально описати архітектуру підсистеми безпеки як сукупність взаємопов'язаних вузлів, що забезпечують стабільне функціонування мобільного застосунку в реальних умовах експлуатації. Архітектура підсистеми має розподілений характер і базується на інтеграції сучасних хмарних технологій із мобільними пристроями користувачів, що гарантує високу доступність сервісів навіть за умов значного навантаження.

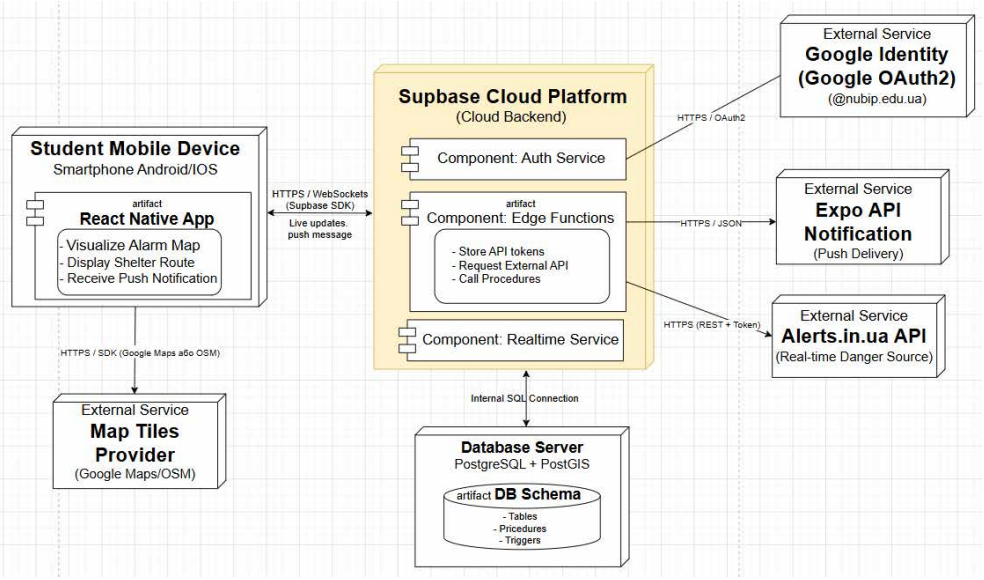

Рис. 2.6 – Діаграма розгортання інтегрованої системи з використанням хмарної інфраструктури Supabase та сервісу сповіщень Expo

Як видно з рис. 2.6, основним клієнтським вузлом у системі виступає «Student Mobile Device», що функціонує на смартфонах під управлінням операційних систем Android або iOS. Ключовим програмним артефактом на цьому рівні є застосунок, побудований на базі фреймворку React Native. Він виконує завдання візуалізації мапи тривог, відображення маршрутів до укриттів та обробку вхідних push-сповіщень. Для забезпечення картографічного функціоналу цей вузол взаємодіє із зовнішнім сервісом Map Tiles Provider. Проте, з метою оптимізації швидкодії та архітектурного розвантаження клієнта, взаємодію реалізовано без використання локальних нативних SDK-компонентів. Замість цього дані передаються через захищений протокол HTTPS.

Центральним вузлом обробки логіки та керування даними на діаграмі розгортання (рис. 2.6) виступає «Supabase Cloud Platform», що виконує роль хмарного бекенду. Вона включає компонент Auth Service для автентифікації користувачів через корпоративний домен за протоколом OAuth2, що дозволяє використовувати університетські акаунти для входу в систему. Важливу роль в архітектурі відіграють Edge Functions, які виступають посередником для взаємодії з зовнішніми API, та Realtime Service, що забезпечує миттєву синхронізацію статусів тривог через протокол WebSockets, забезпечуючи актуальність інформації в режимі реального часу.

Надійне збереження та просторовий аналіз даних забезпечується вузлом «Database Server» на базі PostgreSQL із розширенням PostGIS [7]. У поточній реалізації підсистеми просторові координати сховищ зберігаються у реляційних таблицях у форматі DECIMAL, а математичний розрахунок матриці відстаней за формулою гаверсинусів виконується безпосередньо на клієнтській частині для забезпечення автономності обчислень та працездатності алгоритму в умовах нестабільного зв'язку. Артефакт «DB Schema» об’єднує в собі реляційні таблиці, збережені процедури та тригери, які автоматизують внутрішню логіку обробки вхідних сигналів про загрози безпосередньо на рівні бази даних, що значно розвантажує клієнтську частину застосунку.

Додатково в загальній архітектурі, представленій на рис. 2.6, задіяні

зовнішні спеціалізовані сервіси, такі як Expo API Notification для гарантованої доставки критично важливих сповіщень та alert.in.ua API як первісне джерело даних про загрози. Така фізична конфігурація забезпечує високу масштабованість підсистеми та мінімізує затримки при передачі інформації. Обрана архітектура розгортання гарантує стабільну роботу системи при зростанні кількості одночасних підключень, що є пріоритетним завданням для сервісів підтримки безпеки університетської спільноти НУБіП України.

2.4 Ролі користувачів і політика доступу

Надійність та безпека функціонування підсистеми безпеки базується на принципі суворої автентифікації. На відміну від публічних інформаційних сервісів, дана система є закритою університетською платформою, доступ до якої можливий виключно після повної верифікації особи. Такий підхід дозволяє запобігти будь – якому несанкціонованому доступу до внутрішньої цифрової інфраструктури університету, гарантувати цілісність даних про укриття та забезпечити високий рівень модерації контенту. У системі реалізовано модель управління доступом на основі ролей (RBAC) у поєднанні з політиками безпеки на рівні рядків бази даних (Row Level Security, RLS), що дозволяє гнучко контролювати операції над кожним записом.

Центральним елементом політики безпеки є обов'язкова авторизація через корпоративний акаунт Google. Користувач не має технічної можливості отримати доступ до будь – якого функціоналу додатка – будь то перегляд мапи тривог чи пошук укриття – без успішного входу через доменну пошту @nubip.edu.ua. Це реалізовано на рівні програмного шлюзу, який блокує інтерфейс застосунку до моменту отримання валідного токена від Supabase Auth. Таким чином, система чітко розмежовує два типи суб'єктів: авторизований користувач (студент або співробітник) та адміністратор (диспетчер безпеки НУБіП).

Особлива увага при проєктуванні приділялася безпеці критично важливих даних, прикладом чого є конфігурація доступу до таблиці alert. На рис. 2.7

продемонстровано приклад реалізації RLS – політик для цієї таблиці. Оскільки дані про повітряні загрози повинні надходити в систему безперервно та бути доступними для миттєвого зчитування, політики SELECT налаштовані таким чином, щоб забезпечити безперебійну роботу Realtime-сервісів. Водночас права на операції INSERT, UPDATE та DELETE жорстко обмежені, що запобігає випадковому або навмисному спотворенню інформації про активні тривоги. Таке налаштування на рівні ядра бази даних гарантує, що навіть у разі спроби прямого втручання через API, структура даних залишиться цілісною, а користувачі бачитимуть лише верифіковані статуси безпеки.

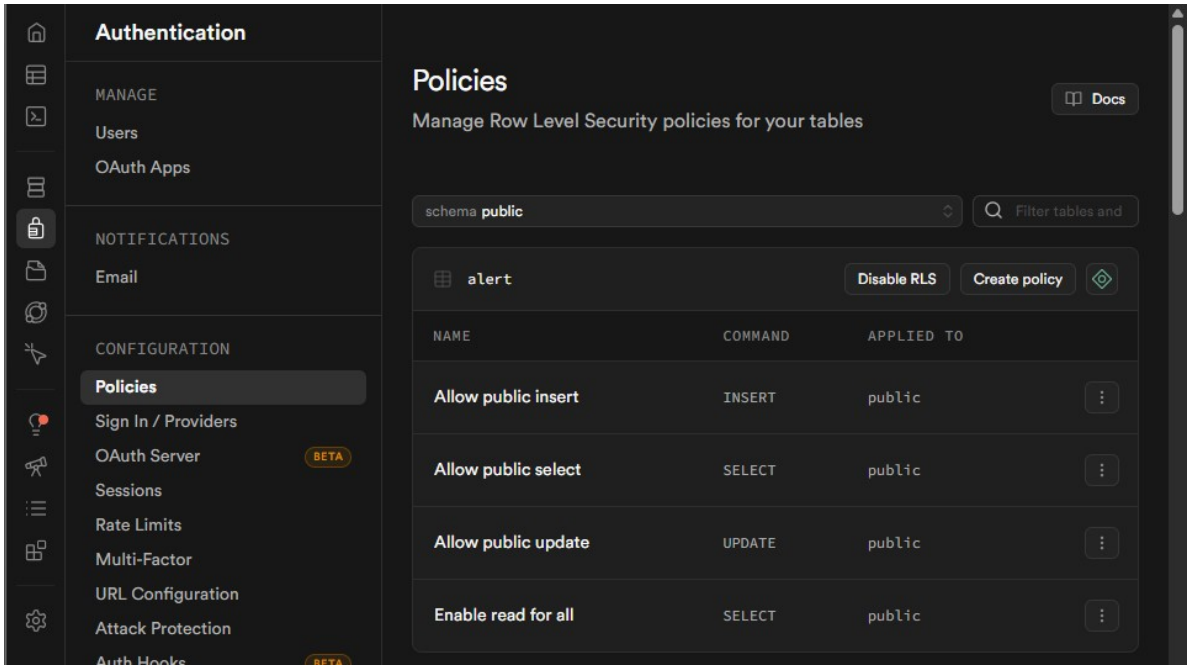

Рис. 2.7 – Приклад RLS доступу до таблиці alert

Авторизовані користувачі складають основну групу суб'єктів системи. Після успішного проходження процедури OAuth 2.0 вони отримують право на перегляд актуальної мапи тривог, доступ до повного інтерактивного списку укриттів університету та можливість побудови маршрутів. Згідно з налаштованими політиками RLS для таблиць comment та rating, студенти можуть залишати власні відгуки та оцінювати стан сховищ за допомогою команди INSERT. Безпека даних при цьому контролюється за принципом персональної відповідальності: користувач має право на видалення (DELETE) або оновлення

(UPDATE) виключно тих записів, які були створені ним особисто. Це автоматично перевіряється сервером через зіставлення ідентифікатора user_id поточної сесії та поля власника рядка в базі даних.

Адміністративний рівень доступу, закріплений за диспетчерами НУБіП, передбачає розширені повноваження щодо управління стратегічним контентом системи. Тільки користувачі з правами адміністратора мають технічний доступ до редагування та видалення інформації в таблиці shelter. Це гарантує, що дані про безпечні локації на території кампусу, їхню місткість та стан є офіційними та верифікованими адміністрацією. Крім того, адміністратори відповідають за загальну модерацію соціального контенту та управління глобальними технічними параметрами системи.

Технічна реалізація описаної моделі доступу на базі механізму Row Level Security у Supabase дозволяє перенести логіку безпеки з клієнтської частини безпосередньо на рівень PostgreSQL. Оскільки система вимагає обов’язкової авторизації, абсолютно всі запити до таблиць profile, comment та rating обробляються з обов'язковою перевіркою статусу authenticated. Навіть у разі спроби прямого звернення до API в обхід інтерфейсу, база даних автоматично відхилить запит, якщо токен користувача не відповідає встановленим вимогам або корпоративним обмеженням. Такий архітектурний підхід робить підсистему безпеки НУБіП України стійкою до зовнішніх втручань та забезпечує надійне й контрольоване середовище для цивільного захисту університетської спільноти.

2.5 Визначення ключових сценаріїв використання системи

Процес проектування інтелектуальних систем цивільного захисту вимагає глибокого аналізу поведінкових моделей користувачів у ситуаціях різного рівня критичності, особливо в умовах нестабільного зв'язку чи необхідності миттєвої синхронізації даних. Визначення ключових сценаріїв використання (Use Case Scenarios) дозволяє трансформувати абстрактні функціональні вимоги у конкретні послідовності дій, що забезпечують досягнення мети користувача

навіть за екстремальних обставин. У межах розробки підсистеми безпеки для студентів НУБіП України було виділено п'ять фундаментальних сценаріїв, які охоплюють повний життєвий цикл взаємодії користувача з додатком: від етапу авторизації до прийняття оперативних рішень у моменти реальної загрози та аналізу даних.

Перший сценарій – «Автентифікація та інтеграція в корпоративну екосистему». Оскільки система є закритим інструментом підтримки безпеки університетської спільноти, цей сценарій виступає обов'язковим «шлюзом». Процес передбачає використання Google OAuth 2.0 для верифікації приналежності користувача до доменної зони @nubip.edu.ua. Після підтвердження особи система автоматично ініціалізує сесію через Supabase Auth, завантажує персональний профіль та відкриває доступ до внутрішніх ресурсів. Цей сценарій гарантує, що конфіденційна інформація про внутрішні плани евакуації та статистику тривог кампусу залишається доступною лише для верифікованих членів спільноти.

Другий сценарій – «Автоматизований моніторинг загрози та оперативне сповіщення» – є найбільш технічно складним та критично важливим для життєдіяльності системи. На відміну від пасивних додатків, дана підсистема працює за принципом активного моніторингу в режимі реального часу. Сценарій базується на роботі зовнішнього сервісу автоматизації cron-job.org (приклад роботи наведено на рис. 2.8), який щохвилини надсилає запит до серверної Edge-функції на платформі Supabase. Ця функція, написана на середовищі Deno [9], виконує роль інтелектуального агента: вона звертається до API сервісу alerts.in.ua [12], отримує актуальний перелік тривог по всій країні та порівнює його з поточним станом бази даних.

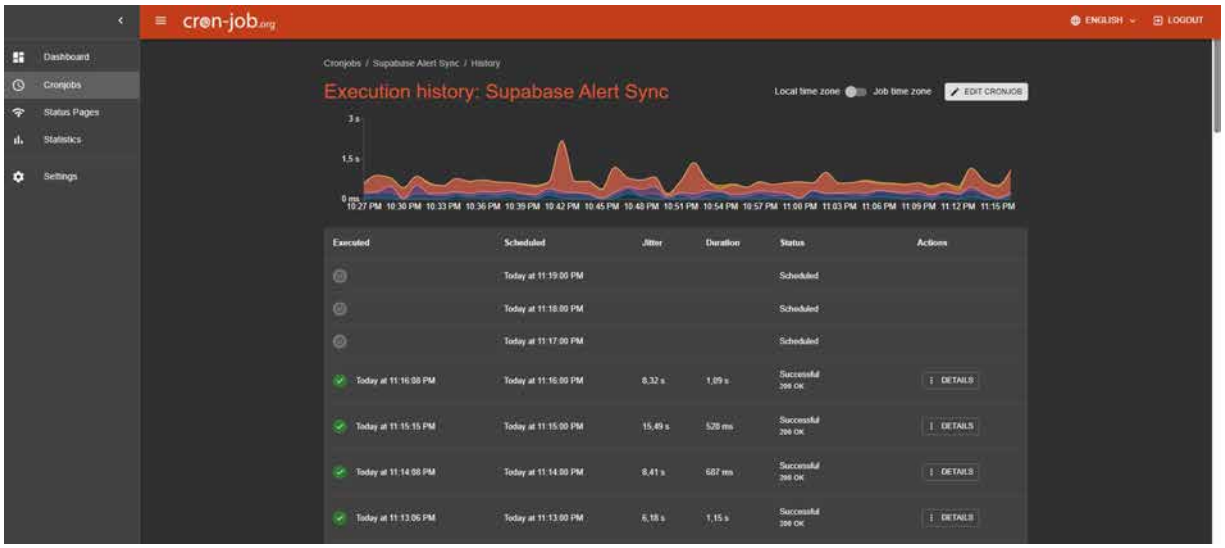

Рис. 2.8 – Робота сервісу cron-job.org

Логіка сценарію передбачає два вектори дій: ідентифікацію нових загроз та фіксацію «відбоїв». У разі виявлення нової тривоги у регіоні, що відповідає налаштуванням користувача, функція автоматично формує масив push-токенів через таблиці settings та push_tokens і надсилає запит до Expo Push API. Сценарій завершується на стороні користувача: при натисканні на отримане сповіщення застосунок автоматично перенаправляє студента на вкладку «Сповіщення», де акумулюється вся детальна інформація про характер загрози та необхідні кроки для безпеки. Аналогічно обробляється і сигнал про закінчення загрози, що дозволяє користувачеві миттєво дізнатися про стабілізацію ситуації в регіоні.

Третій сценарій – «Інтерактивна навігація та забезпечення відмовостійкості системи» – зосереджений на наданні інструментів порятунку в умовах деградації інфраструктури зв'язку. Основна гілка передбачає використання GPS-координат для розрахунку найкоротшого шляху до укриття. Проте, унікальність сценарію полягає у механізмах «глибокої відмовостійкості». При зникненні інтернет-з’єднання користувач зберігає можливість доступу до вкладки «Список укриттів», де завдяки локальному кешуванню на пристрої відображаються збережені дані про сховища. Більше того, у сценаріях повної відсутності GPS – сигналу або його навмисного глушіння, додаток надає доступ до статичних план – схем розташування укриттів у навчальних корпусах та

гуртожитках НУБіП України. Ці графічні схеми є частиною автономного сховища додатка і дозволяють користувачу зберігати можливість базового орієнтування навіть у разі втрати зовнішнього зв’язку.

Четвертий сценарій – «Соціальна модерація та контроль якості інфраструктури захисту» – впроваджує елементи громадського контролю. Під час або після завершення тривоги студент може оцінити стан сховища та залишити відгук. Цей сценарій забезпечує прямий зв'язок між користувачем та адміністрацією університету. Завдяки політиці Row Level Security (RLS), забезпечується авторство відгуків, що стимулює відповідальне ставлення до надання інформації про наявність комфортних умов, вільних місць чи працездатність інженерних мереж укриття. Дані, зібрані за цим сценарієм, стають фундаментом для покращення умов цивільного захисту на території кампусу.

П'ятий сценарій – «Адміністративний моніторинг та аналітична звітність щодо динаміки загроз» – є спеціалізованим функціоналом, доступним виключно для користувачів із правами адміністратора (диспетчерів безпеки). У цьому сценарії адміністратор взаємодіє з модулем поглибленої аналітики, який на основі даних, накопичених Edge-функціями, формує динамічні графіки та звіти. Система дозволяє відстежувати статистику тривог за останні 7 днів, надаючи точні цифри щодо інтенсивності загроз у конкретні дати, наприклад, 19 чи 20 число.

Візуалізація через лінійні графіки та секторні діаграми розподілу активності по регіонах дозволяє адміністрації університету оцінювати ризики, готувати офіційні звіти для керівництва та приймати управлінські рішення щодо режиму роботи навчальних корпусів. Завдяки обмеженню доступу на рівні RLS та інтерфейсу, цей сценарій забезпечує конфіденційність аналітичних даних, перетворюючи додаток на потужний інструмент стратегічного планування для служб цивільного захисту НУБіП.

Узагальнюючи, описані сценарії використання формують комплексну та відмовостійку функціональну модель підсистеми. Завдяки інтеграції

щохвилинної синхронізації через cron-сервіс, продуманій логіці push-повідомлень та наявності офлайн-схем евакуації для студентів, а також закритому модулю аналітики для адміністраторів, додаток стає надійним гарантом безпеки університетської спільноти. Така багаторівнева структура взаємодії забезпечує стабільну роботу системи за будь – яких технічних умов, поєднуючи оперативну допомогу з глибоким аналізом ситуації.

2.6 Інтеграція зовнішніх сервісів

Сучасні мобільні застосунки рідко функціонують як ізольовані системи; їхня ефективність та надійність здебільшого залежать від успішної інтеграції з перевіреними зовнішніми сервісами та хмарними інфраструктурами. У межах розробки підсистеми безпеки для НУБіП України було обрано сервіс-орієнтований підхід, що дозволив делегувати критично складні завдання – такі як автентифікація, картографія та доставка сповіщень – спеціалізованим платформам. Завдяки цьому вдалося забезпечити високу стабільність роботи застосунку, оскільки кожен компонент спирається на перевірені промислові стандарти.

Першим фундаментальним рівнем інтеграції є використання Google Auth через екосистему Supabase Auth. Вибір протоколу OAuth 2.0 як основного механізму входу дозволив реалізувати сувору політику доступу за корпоративною ознакою. Інтеграція дозволяє системі автоматично верифікувати приналежність користувача до доменної зони @nubip.edu.ua, що є критично важливим для закритої університетської платформи. Використання інфраструктури Google для автентифікації знімає з розробника відповідальність за зберігання паролів та забезпечує захист від атак перебором, оскільки всі процеси перевірки особи відбуваються на захищених серверах Google, а застосунок отримує лише зашифрований токен доступу.

Другим стратегічним компонентом є інтеграція з Google Maps API (або альтернативним Map Tiles Provider). Оскільки головною функцією підсистеми є

навігація до найближчого укриття, використання професійних картографічних сервісів є обов'язковим. Програмна інтеграція реалізована через компонент WebView та протоколи міжпрограмного зв’язку (Linking), що дозволяє динамічно завантажувати інтерактивні карти та візуалізувати складні навігаційні маршрути. Завдяки цій інтеграції додаток отримує доступ до актуальної бази доріг та пішохідних шляхів, що дозволяє будувати коректні маршрути з урахуванням реальної забудови кампусу НУБіП, забезпечуючи студента візуальним орієнтиром у момент небезпеки.

Третій рівень інтеграції стосується першоджерела даних про повітряні загрози – API сервісу alerts.in.ua. Цей сервіс надає структуровану інформацію у форматі JSON щодо статусу тривог у всіх регіонах України. Інтеграція реалізована на рівні бекенду через Edge Functions, що дозволяє системі щохвилини отримувати оновлення. Використання спеціалізованого API гарантує достовірність інформації, оскільки дані автоматично синхронізуються з офіційними державними джерелами. Програмна обробка цих даних дозволяє системі не просто констатувати факт тривоги, а й розрізняти її типи та визначати тривалість небезпеки, що є основою для подальшої аналітичної звітності.

Важливим ланцюгом у системі оперативного реагування є інтеграція з сервісом Expo Push Notifications. Традиційні методи передачі даних (наприклад, HTTP-запити від клієнта) є неефективними для екстрених сповіщень через високі затримки та енергоспоживання. Інтеграція з таким сервісом дозволяє системі використовувати архітектуру «push», коли сервер самостійно ініціює передачу даних на пристрій користувача. Використання Expo як проміжного рівня спрощує взаємодію з різними операційними системами (Android та iOS), гарантуючи доставку повідомлення навіть тоді, коли застосунок перебуває у фоновому режимі або телефон заблоковано.

Замикає архітектурне коло інтеграція з сервісом зовнішньої автоматизації cron-job.org. Оскільки хмарні Edge-функції за своєю природою є реактивними (вони запускаються лише за викликом), для створення системи постійного моніторингу знадобився зовнішній ініціатор. Сервіс cron-job.org налаштований

на виконання щохвилинних HTTP-запитів до розгорнутих функцій Supabase. Ця інтеграція гарантує, що перевірка нових тривог відбувається безперервно 24/7 без необхідності тримати серверний процес постійно активним, що значно оптимізує використання ресурсів та підвищує відмовостійкість архітектури.

Таким чином, синергія Google Auth, Google Maps, API мапи тривог, сервісів сповіщень та систем автоматизації перетворює мобільний застосунок на складну багатофункціональну платформу. Кожен інтегрований сервіс закриває певну критичну нішу: безпеку, навігацію, достовірність даних, швидкість зв’язку та стабільність моніторингу. Такий комплексний підхід до інтеграції зовнішніх сервісів дозволив створити систему, що відповідає сучасним вимогам до програмного забезпечення у сфері цивільного захисту та безпеки життєдіяльності.

3 РЕАЛІЗАЦІЯ ПРОГРАМНОЇ СИСТЕМИ

3.1 Серверна частина: структура, модулі, маршрути

Серверну структуру підсистеми побудовано на платформі Supabase. Таке рішення було прийнято оскільки ця платформа може забезпечити інтеграцію реляційної бази даних PostgreSQL, механізмів автентифікації та середовища виконання серверних функцій. Таку будову обрано, адже потрібно швидко працювати з великими обсягами даних без затримок. Основу логічного рівня становить реляційна схема даних, яка була оптимізована для зберігання геопросторової інформації про укриття та динамічних записів про повітряні загрози.

Центральним об'єктом системи є таблиця "User", яка синхронізована з сервісом автентифікації та має зв'язки «один-до-одного» з таблицями Profile та Settings. У таблиці Settings (структуру якої наведено на рис. 3.1) зберігається критично важливий параметр фільтрації контенту – region_id. Це унікальний ідентифікатор регіону (UID), отриманий з API, який дозволяє системі чітко асоціювати кожного студента з конкретною локацією (наприклад, м. Київ або Київська область). Саме цей ідентифікатор є ключовим фільтром у серверному коді: під час детекції нової тривоги система розсилає сповіщення не всім підряд, а лише тим користувачам, чий region_id збігається з ідентифікатором зони небезпеки.

userid uuid	sound_enabled bool	vibration_enabled bool	volume_level int4	geo_access b..	region_id int4	region_name varchar
2a06aa5a-df71-4dc6-9...	FALSE	FALSE	100	TRUE	1293	м. Харків
b3d37154-7043-46d6-...	TRUE	TRUE	80	FALSE	31	м. Київ
f9f16081-3d02-4430-8...	TRUE	TRUE	80	FALSE	31	м. Київ

Рис. 3.1 - Структура таблиці Settings із параметром локалізації користувача у базі даних Supabase

Взаємодія користувачів з об'єктами інфраструктури безпеки реалізована через таблиці Shelter, Comment та Rating. Таблиця Shelter містить географічні

координати об'єктів у форматі DECIMAL(9,6), що є стандартом для високоточної навігації. Соціальна складова підсистеми базується на системі зовнішніх ключів: таблиця Comment посилається на ідентифікатори користувача та укриття, що забезпечує цілісність ланцюжка відгуків. Для управління системою сповіщень використовується таблиця push_tokens, яка зв’язує унікальний токен пристрою Expo з конкретним UserID. Це дозволяє реалізувати надійну доставку повідомлень.

Логіка обробки зовнішніх подій зосереджена в Edge-функції sync-alerts, розгорнутій у середовищі Deno. Цей модуль виконує роль головного контролера серверної частини. Алгоритм роботи функції базується на циклічному зверненні до зовнішнього API alerts.in.ua. Функція виконує складну аналітичну роботу:

- Ідентифікація нових загроз: Порівняння JSON-відповіді API з існуючими записами в таблиці Alert для уникнення дублікатів.
- Детекція «відбоїв»: Перевірка активних тривог у базі даних. Якщо тривога з певним external_id зникла з API-відповіді, функція автоматично фіксує час закінчення (end_time) та змінює статус is_active на false.
- Таргетована розсилка: Функція виконує SQL-запит для пошуку всіх користувачів, чий region_id збігається з територією тривоги, та ініціює відправку push-повідомлення через Expo API.

Зберігання візуального контенту організовано за спеціальною схемою. Основна частина медіаданих, таких як фотографії укриттів, зберігається у хмарному сховищі Supabase Storage (див. рис. 3.2) в спеціалізованому контейнері shelters-image. Кожне фото укриття пов'язане з відповідним записом у таблиці Shelter через пряме посилання (URL), яке мобільний застосунок використовує для відображення. Крім того, щоб підвищити відмовостійкість, критично важливі план-схеми розташування сховищ у корпусах НУБіП України інтегровані безпосередньо в код застосунку як локальні ресурси (assets). Це забезпечує, що навіть якщо зникне інтернет-з'єднання або виникнуть проблеми з

хмарним сховищем, користувач все ще матиме доступ до графічних схем евакуації.

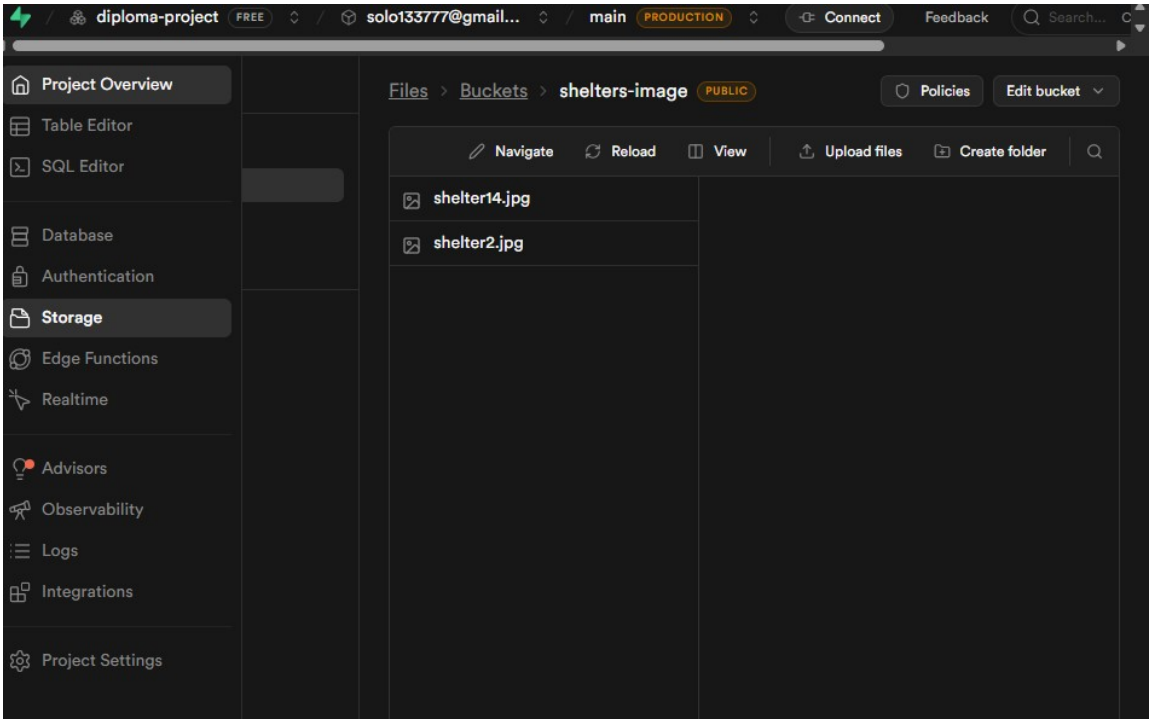

Рис. 3.2 – Інтерфейс керування об’єктним сховищем (Storage Bucket) для збереження медіаданих про укриття.

3.2 Клієнтська частина: компоненти, маршрути, форми

Клієнтська частина системи побудована на базі компонентно-орієнтованої архітектури React Native. Основний акцент зроблено на швидкодії та інтуїтивності інтерфейсу в умовах критичних ситуацій.

3.2.1 Автентифікація та вхід у систему

Задля безпеки даних та ідентифікації того чи відноситься користувач до закладу вищої освіти, в системі реалізовано механізм Single Sign-On (SSO) через Google OAuth.

Першим екраном системи є «Welcome screen» (рис. 3.3), який пропонує користувачеві авторизуватися за допомогою корпоративної пошти домену @nubip.edu.ua.

Рис. 3.3 – Стартовий екран авторизації мобільного застосунку підтримки студентів

Вибір облікового запису (рис. 3.4) інтегровано з системними сервісами Android/iOS, що виключає необхідність ручного введення паролів у додатку.

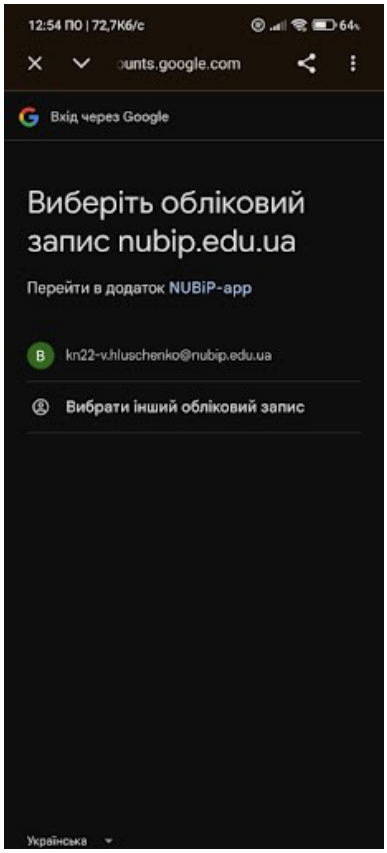

Рис. 3.4 – Реалізація процесу OAuth-автентифікації користувача через сервіс Google Identity

Після підтвердження облікового запису система автоматично створює або оновлює профіль користувача в базі даних Supabase, після чого відкриває доступ до основного функціоналу мобільного застосунку. Використання Google OAuth дало змогу мінімізувати кількість дій під час авторизації, уникнути ручного введення паролів та забезпечити безпечне зберігання даних сесії відповідно до сучасних вимог мобільної безпеки.

3.2.2 Головне меню та архітектура навігації

Після успішної автентифікації користувач переходить на головний екран мобільного застосунку (рис. 3.5). Інтерфейс системи поділено на два основні функціональні блоки: «Безпечний НУБіП» та «Навчання».

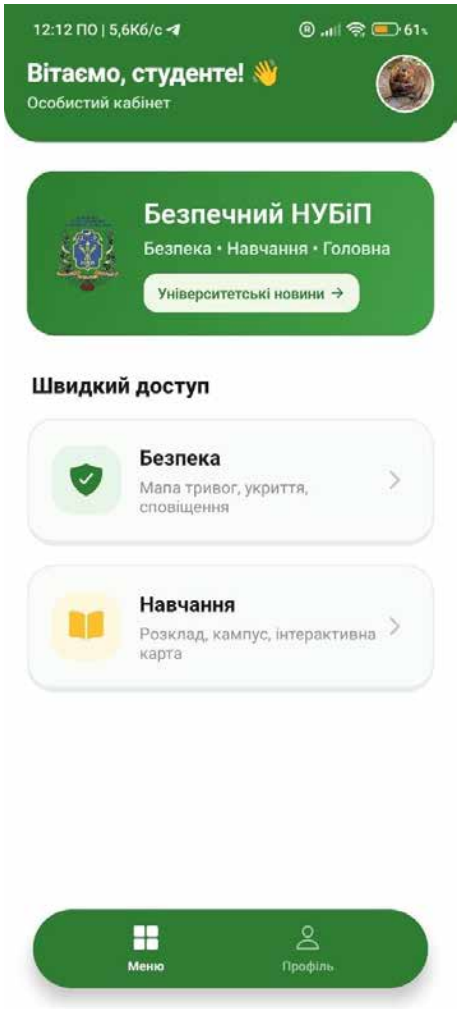

Рис. 3.5 – Інтерфейс головного меню мобільної системи підтримки студентів НУБіП

Такий підхід дозволяє розмежувати модулі оперативного реагування та функції, пов’язані з повсякденним освітнім процесом.

Для реалізації навігації використовується комбінація Stack Navigation, що забезпечує вкладені переходи між екранами, та Bottom Tab Bar для швидкого доступу до основних розділів застосунку.

3.2.3 Модуль «Безпечний НУБіП» та Realtime-дані

Розділ «Безпечний НУБіП» (рис. 3.6) і є підсистема безпеки. Він включає три основні модулі: «Мапа тривог», «Список укриттів» та «Сповіщення».

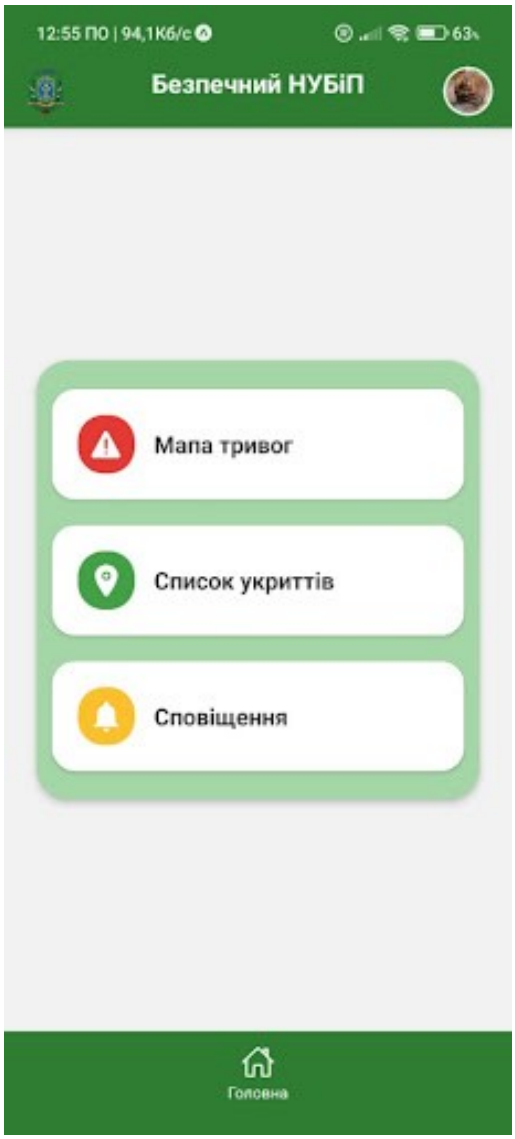

Рис. 3.6 – Панель інструментів для доступу до сервісів оперативного інформування та захисту.

Мапа тривог (рис. 3.7): Візуалізація поточної ситуації в Україні в режимі реального часу. Дані інтегруються через зовнішній API (alerts.in.ua), що забезпечує актуальність інформації. У мапи є можливість змінити тему (з світлої на темну), отримати додаткову інформацію по позначкам, а також велику можливість персоналізації перейшовши у вкладку налаштувань.

Рис. 3.7 – Інтерфейс модуля моніторингу статусу повітряних тривог у режимі реального часу.

Список укриттів (рис. 3.8): Перелік доступних захисних споруд університету. Для зручності користувачів реалізовано фільтри за статусом («Доступно», «Зачинено») та місткістю, а також пошук за назвою корпусу. На додачу ця вкладка має план-схему укриттів, яка працює в офлайн режимі оскільки додана локально до папки assets в проєкті під час розробки, і включалась в APK під час збірки.

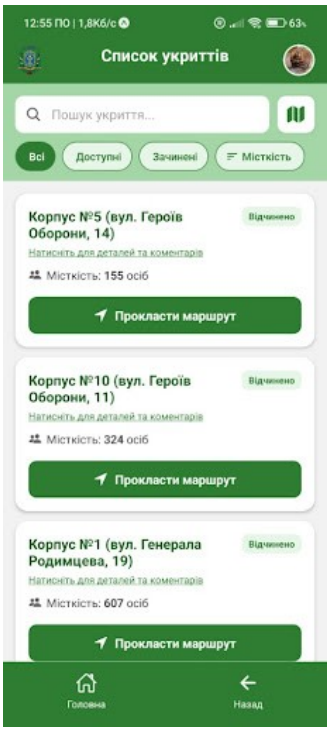

Рис. 3.8 – Перелік захисних споруд НУБіП України з інформацією про їх розташування та місткість.

Сповіщення (рис. 3.9): Стрічка останніх подій, де відображаються тривоги саме для обраного користувачем регіону.

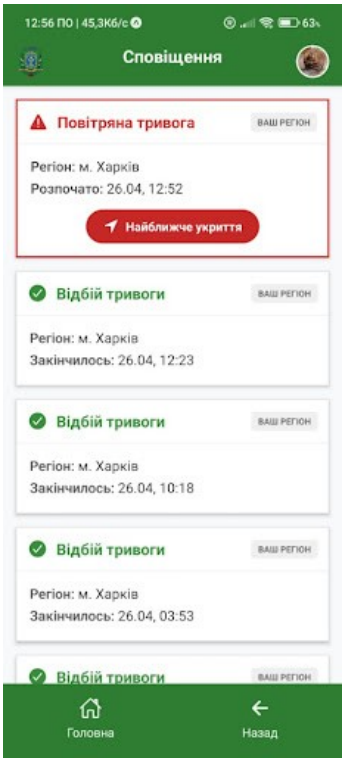

Рис. 3.9 – Інформаційна панель із хронологією сповіщень про небезпеку та актуальним станом тривоги.

Для візуалізації аналітичних даних у мобільному додатку було розроблено програмний модуль (рис. 3.10).

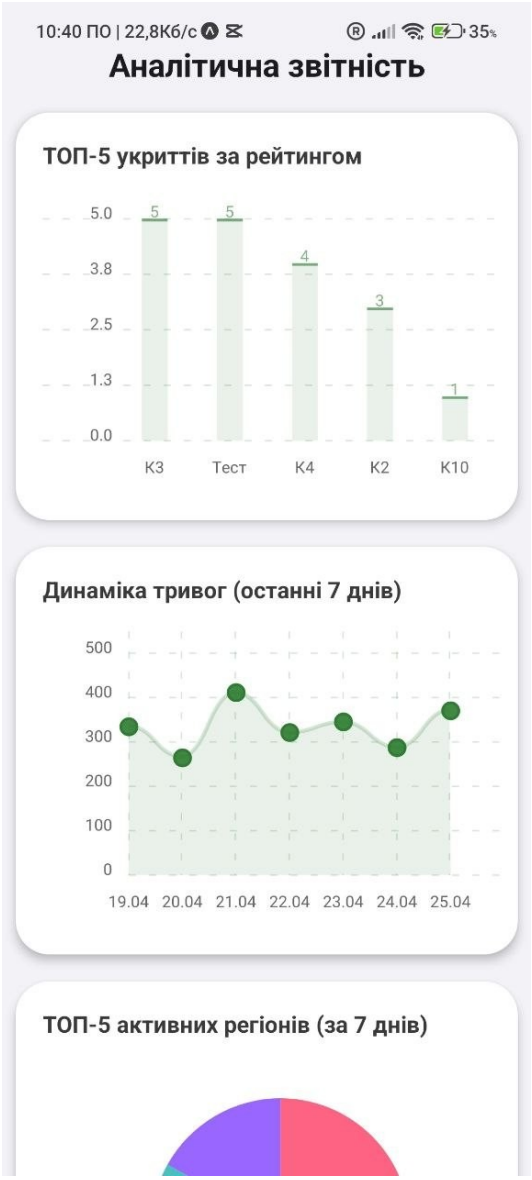

Рис. 3.10 – Візуалізація SQL View для агрегації статистики тривог

Модуль забезпечує асинхронне завантаження історичних даних безпосередньо з таблиць СУБД Supabase за допомогою паралельних запитів. Первинна обробка, фільтрація за останні 7 днів та агрегація статистичних показників (підрахунок кількості тривог за кожну дату, обчислення середнього рейтингу укриттів) виконуються на стороні клієнтської частини мобільної системи. Для побудови інтерактивних діаграм використано інструментарій бібліотеки react-native-chart-kit.

3.2.4 Форми вводу та налаштування користувача

Для збору та редагування даних використано механізм Controlled Components. Екран профілю (рис. 3.11): Відображає статичні дані, отримані під час ідентифікації.

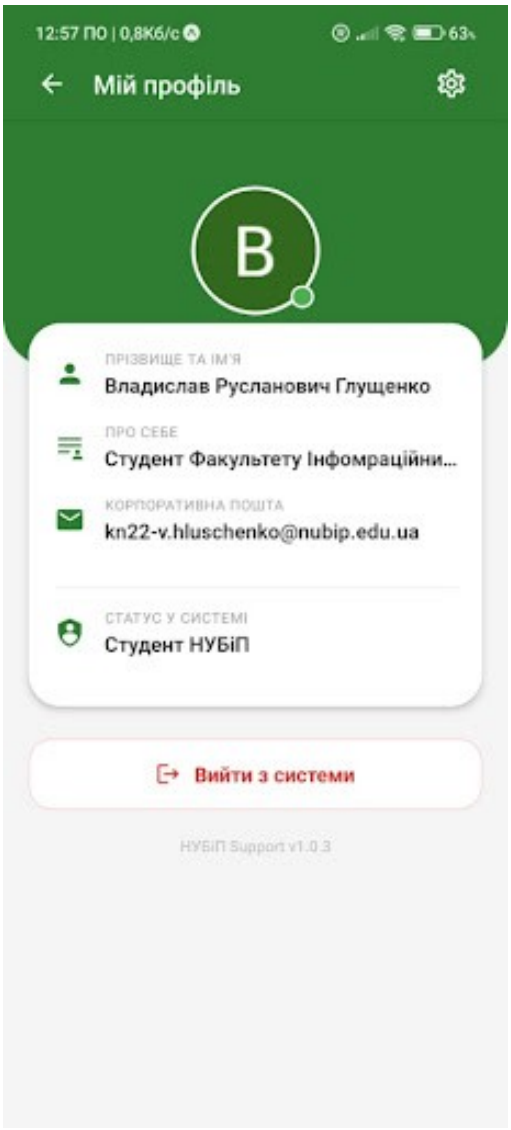

Рис. 3.11 – Інтерфейс особистого профілю користувача з відображенням ідентифікаційних даних та параметрів.

Екран налаштувань користувача (рис. 3.12): Налаштування регіону, мови та звукових сигналів реалізовані через компоненти Picker та Switch. Будь – яка зміна тригерить запит UPDATE до таблиці profile та settings у Supabase, забезпечуючи персоналізацію пуш – сповіщень.

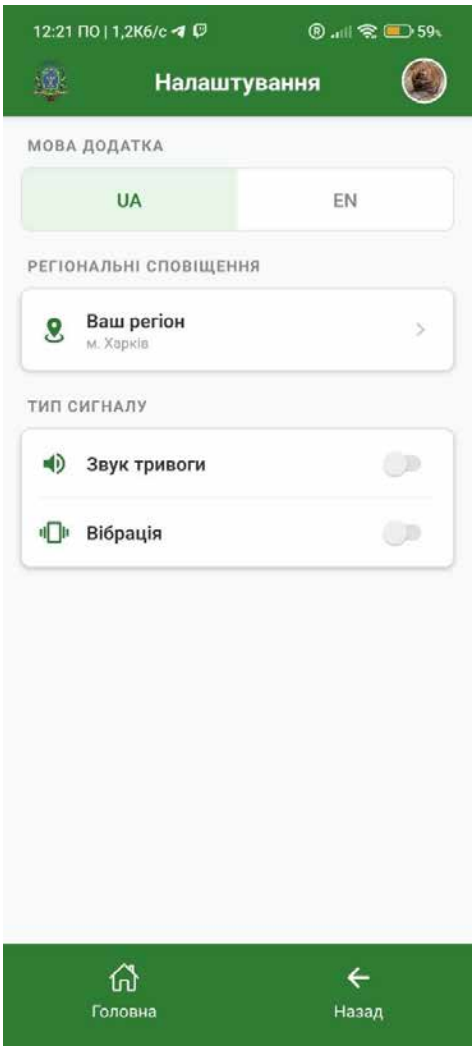

Рис. 3.12 – Інтерфейс модуля конфігурації параметрів підсистеми та керування сповіщеннями про загрози

3.2.5 Рольова модель адміністратора (RBAC)

Система підтримує розмежування прав доступу на рівні інтерфейсу. За допомогою перевірки метаданих користувача в JWT, додаток визначає наявність адміністративних прав. Для адміністратора рендеряться додаткові елементи управління (рис. 3.13):

- Кнопки редагування існуючих записів;
- Кнопка додавання нового об'єкта. Це дозволяє оперативно керувати інфраструктурою безпеки університету безпосередньо з мобільного пристрою.

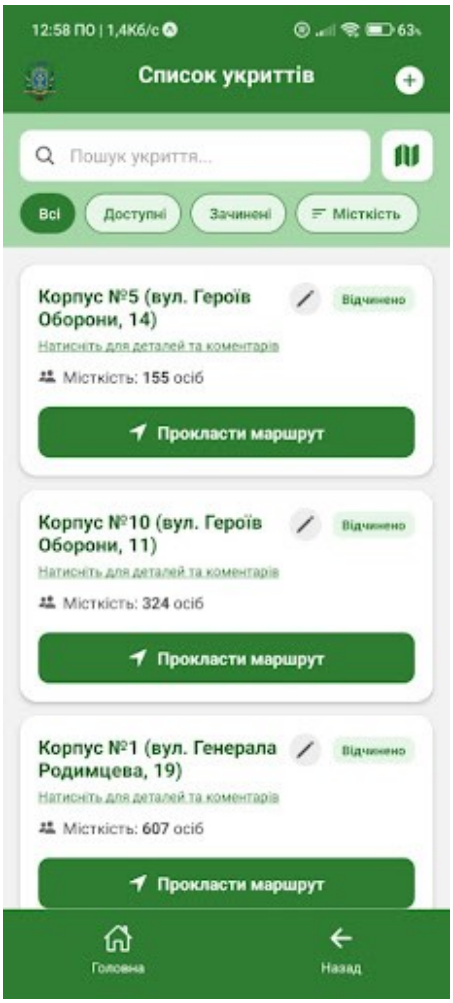

Рис. 3.13 – Інтерфейс адміністративного модуля для моніторингу та редагування бази даних захисних споруд

3.3 Реалізація автентифікації та авторизації

Реалізація механізмів автентифікації та авторизації в інтегрованій мобільній системі НУБіП є фундаментальним аспектом забезпечення інформаційної безпеки та розмежування функціональних можливостей різних категорій користувачів. Основою обраного технічного підходу є використання протоколу OAuth 2.0 у поєднанні з технологією OpenID Connect, що дозволяє безшовно інтегрувати додаток у наявну корпоративну екосистему університету через сервіси Google. Використання Single Sign-On через Google OAuth значно спрощує процес входу та знижує ризики, пов’язані зі зберіганням паролів у системі. Технічна реалізація цього процесу передбачає використання платформи

Supabase Auth як центрального ідентифікаційного вузла, який виступає надійним посередником між клієнтським додатком на базі React Native та провайдером облікових записів.

Ключовою особливістю архітектури безпеки є впровадження програмного фільтра на рівні доменного імені, який автоматично обмежує доступ до системи лише для власників корпоративних електронних адрес домену nubip.edu.ua, що гарантує цілісність університетської спільноти всередині цифрового середовища. Після проходження процедури верифікації на стороні провайдера, сервер автентифікації генерує набір токенів формату JSON Web Token (JWT) [10], що включає Access Token для короткострокового доступу до ресурсів API та Refresh Token для підтримки тривалої активності сесії без повторного введення даних. Усі отримані ідентифікаційні дані зберігаються у захищеному сховищі мобільного пристрою за допомогою нативних криптографічних модулів, що унеможливлює несанкціонований доступ до активної сесії з боку сторонніх програмних продуктів.

Надалі процес авторизації переходить на рівень управління повноваженнями на основі моделі Role-Based Access Control (RBAC). Кожен користувач отримує специфічну роль у структурі бази даних, яка визначає обсяг доступних йому функціональних модулів. Для забезпечення багаторівневого захисту авторизація реалізована як на рівні клієнтського інтерфейсу через механізми умовного рендерингу, так і на рівні серверної частини за допомогою політик Row Level Security (RLS) у системі керування базами даних PostgreSQL. Це означає, що розширені права адміністратора, які включають можливість редагування інформації про укриття або перегляд статистичних даних, захищені не лише видимістю елементів керування в додатку, а й строгими правилами доступу безпосередньо на рівні запитів до таблиць. Такий комплексний підхід до автентифікації та авторизації створює стійку до зовнішніх загроз систему, де процес ідентифікації виступає надійним вхідним фільтром, а гнучка модель повноважень гарантує безпечне виконання функцій згідно з посадовими обов'язками або статусом кожного учасника навчального процесу.

3.4 Побудова API-взаємодії між frontend і backend

Побудова високоефективної та відмовостійкої взаємодії між клієнтською частиною, розробленою на базі фреймворку React Native, та серверною інфраструктурою Supabase, є фундаментальним етапом проектування інтегрованої системи, оскільки саме архітектура обміну даними визначає надійність та швидкість інформування студентів у критичних ситуаціях. «Для взаємодії між клієнтською та серверною частинами обрано гібридний підхід: RESTful API для звичайних операцій та Realtime-модель (WebSockets) для миттєвої передачі критичних даних. Ключовим інструментом для реалізації цієї взаємодії виступає Supabase Client SDK, який інтегрується безпосередньо в логіку мобільного додатка та забезпечує високорівневу абстракцію над мережевими запитами. Це дозволяє розробнику оперувати об’єктами мови TypeScript замість маніпуляцій з сирими HTTP – запитами, що суттєво підвищує безпеку типів та мінімізує ймовірність виникнення помилок під час розробки та експлуатації системи.

Центральним елементом серверного API є прошарок PostgREST, який автоматично перетворює структуру бази даних PostgreSQL у доступні кінцеві точки RESTful API. Такий підхід забезпечує надзвичайно низьку латентність, оскільки запити від мобільного додатка обробляються безпосередньо базою даних без необхідності проходження через додаткові рівні серверної логіки, що є критичним фактором для систем оперативного реагування. Обмін даними здійснюється виключно у форматі JSON, що гарантує легкість серіалізації та десеріалізації об’єктів на стороні клієнта. Окремої уваги заслуговує реалізація механізму Change Data Capture через протокол WebSockets, який лежить в основі модуля реального часу. Замість традиційного методу періодичного опитування сервера (polling), мобільний клієнт встановлює стійке з’єднання, через яке сервер миттєво транслює оновлення конкретних таблиць бази даних. Це дозволяє користувачеві бачити актуальний статус доступності укриттів або нові тривожні повідомлення в ту саму секунду, коли адміністратор вносить зміни в систему, що

суттєво економить ресурси процесора та заряд акумулятора мобільного пристрою.

Для виконання складних бізнес-операцій, які виходять за межі простих маніпуляцій з таблицями, в архітектуру API інтегровано Supabase Edge Functions. Ці безсерверні функції, розроблені на платформі Deno, виступають у ролі ізольованих мікросервісів, що виконують код максимально близько до користувача (edge computing). Саме через цей механізм реалізовано інтеграцію із зовнішніми сервісами, такими як загальнодержавна система оповіщення про повітряні тривоги та сервіс Expo Go Notification для відправки пуш-сповіщень. Винесення цієї логіки на сторону бекенда дозволяє зберегти клієнтський додаток легким та забезпечити централізоване управління алгоритмами обробки сигналів. Безпека кожного етапу взаємодії гарантується використанням JSON Web Tokens, які автоматично додаються до кожного запиту та верифікуються сервером на відповідність встановленим політикам Row Level Security. Це створює надійний бар'єр, який запобігає несанкціонованому доступу до даних навіть у разі перехоплення мережевого трафіку.

Крім того, при проектуванні API-взаємодії було приділено увагу обробці нестабільних станів мережі, що часто трапляється під час переміщення користувачів до підземних укриттів. Реалізована система обробки помилок та автоматичних повторних спроб (retries) гарантує, що додаток відновить синхронізацію з базою даних одразу після відновлення зв’язку. Використання асинхронних функцій мови JavaScript дозволяє виконувати всі мережеві операції у фоновому режимі, не блокуючи основний потік інтерфейсу, що забезпечує плавний та професійний користувацький досвід. Таким чином, спроектована архітектура взаємодії frontend та backend частин являє собою складну інтегровану екосистему, яка поєднує в собі продуктивність REST – інтерфейсів, гнучкість безсерверних функцій та миттєву швидкість технологій реального часу, формуючи надійний технологічний фундамент для системи підтримки безпеки навчального процесу.

3.5 Процес автоматизованого збирання та дистрибуції мобільного застосунку

Завершальним етапом розробки інтегрованої системи підтримки безпеки НУБіП є організація процесу розгортання та підготовки продуктової версії мобільного застосунку для кінцевих користувачів. На відміну від традиційної контейнеризації серверних застосунків, життєвий цикл мобільної розробки вимагає специфічних інструментів для компіляції вихідного коду в нативні двійкові файли, що можуть бути запущені на операційній системі Android. Для вирішення цього завдання в проекті використано хмарну інфраструктуру Expo Application Services (EAS), яка дозволяє автоматизувати процес збирання, керувати сертифікатами безпеки та забезпечувати стабільну доставку оновлень. Використання такої хмарної моделі збирання дозволяє нівелювати проблеми, пов’язані з розбіжністю локальних оточень розробників, оскільки весь процес компіляції відбувається в ізольованих віртуальних середовищах із чітко визначеними версіями Android SDK, Java Development Kit та системних залежностей Gradle.

Центральним компонентом процесу розгортання є конфігураційний файл eas.json, у якому буде визначено профілі збирання для різних етапів життєвого циклу проекту, зокрема профілі для розробки, попереднього тестування та фінального релізу. Під час ініціалізації процесу збирання хмарна система виконує автоматичне встановлення всіх необхідних модулів, здійснює статичний аналіз коду на предмет помилок та проводить оптимізацію графічних ресурсів для зменшення розміру кінцевого пакета. Важливим аспектом безпеки є управління секретними змінними оточення, такими як ключі доступу до Supabase та ідентифікатори проектів Google OAuth. Ці дані не зберігаються у відкритому вигляді в репозиторії коду, а ін'єктуються в процес збирання безпосередньо через захищені змінні EAS Secrets, що гарантує цілісність системи навіть у разі відкритого вихідного коду. Результатом успішного виконання робочого циклу є формування файлу формату Android Package Kit (APK) для ручного встановлення

на пристрої або Android App Bundle (AAB) для публікації в магазині додатків Google Play.

Для забезпечення коректної роботи push-сповіщень у повноцінних автономних збірках (standalone APK) на цьому етапі реалізується інтегрована конфігурація Firebase. Оскільки сучасні версії Expo SDK вимагають обов'язкової прив'язки додатка до консолі розробника Google, у конвеєр хмарного збирання додано налаштування сертифікованого конфігураційного файлу google-services.json. Цей файл безпечно передається через середовище EAS Secrets, що дозволяє активувати нативний функціонал Firebase Cloud Messaging (FCM) на кінцевих пристроях користувачів без компрометації ключів доступу у відкритому вихідному коді. Процес генерації ідентифікаторів додатка та завантаження цифрових ключів у консолі керування сервісами безпеки представлено на рис. 3.14.

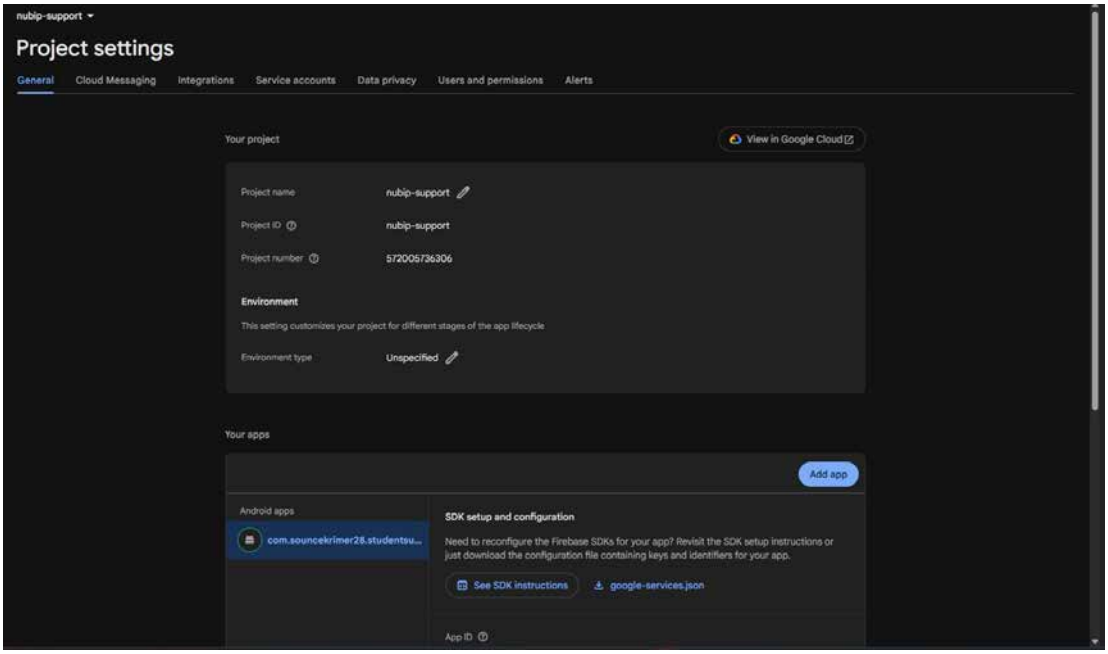

Рис. 3.14 – Інтерфейс консолі керування хмарними сервісами для реєстрації Android-клієнта та налаштування параметрів Firebase FCM

Особливе місце в стратегії дистрибуції займає технологія Over-the-Air (OTA) оновлень через модуль Expo Updates. Цей механізм дозволяє розробнику вносити критичні правки в логіку JavaScript-частини додатка без необхідності повного перезбирання нативного коду та повторного встановлення файлу

користувачем. У ситуаціях, пов’язаних із безпекою навчального закладу, можливість миттєвої доставки оновленого функціоналу або виправлення помилок інтерфейсу є критичною перевагою, оскільки вона гарантує, що всі студенти одночасно отримають актуальну версію системи. Також у рамках процесу розгортання буде реалізовано автоматичне підписання цифровими сертифікатами, що підтверджує автентичність додатка та забезпечує його коректну роботу в екосистемі Android. Таким чином, перехід від контейнеризації серверної частини до автоматизованого конвеєра збирання мобільного клієнта дозволить сформувати надійну технологічну базу для швидкої доставки програмного продукту, забезпечуючи високу якість коду та відповідність сучасним стандартам мобільної розробки, що є невід'ємною частиною успішного впровадження системи підтримки безпеки в університетську інфраструктуру.

4 ТЕСТУВАННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ

4.1 Методика тестування функціональних модулів

Після завершення розробки було проведено ручне тестування підсистеми безпеки, щоб перевірити роботу всіх реалізованих функцій. Тестування не обмежувалося лише перевіркою інтерфейсу. Основна увага приділялася реальним сценаріям використання системи під час повітряної тривоги.

Імітувалися дії студента від моменту авторизації через корпоративний обліковий запис @nubip.edu.ua до отримання екстреного сповіщення та пошуку найближчого укриття. Усі випробування проводилися у середовищі, максимально наближеному до реального використання: з використанням фізичних Android-пристроїв, живих сесій Google OAuth, реальних викликів до API Alerts.in.ua та бази даних Supabase, розгорнутої у хмарній інфраструктурі.

Під час тестування перевірялися такі функції:

- Сценарії авторизації та доступу: Перевірявся шлях користувача при вході через Google. Система мала не лише ідентифікувати особу, а й коректно обробити редирект (Deep Linking) назад у застосунок із поверненням JWT-токенів. Окремо перевірялася поведінка системи при спробі обрати або додати пошту з некоректним доменом.
- Обробка екстрених сповіщень та токенів: Критичним аспектом була перевірка модуля Expo Push Notifications. Тестувалися сценарії повторного встановлення додатка або зміни облікового запису на одному пристрої. Було перевірено логіку UPSERT для таблиці push_tokens у Supabase, щоб уникнути помилок дублювання первинних ключів (Primary Key) та гарантувати, що актуальний токен завжди прив’язаний до ідентифікатора користувача.
- Синхронізація часових поясів та даних: При отриманні даних про початок та закінчення тривоги перевірялася коректність конвертації часу. Оскільки

база даних зберігає часові мітки у форматі UTC, тестування підтвердило, що клієнтська частина додатка правильно трансформує їх у локальний час UTC+3 (Київ), забезпечуючи користувача точною інформацією про тривалість загрози.

- Стійкість інтерфейсу та крайові випадки: Перевірялися реакції додатка на втрату мережевого з'єднання під час завантаження карти укриттів або отримання списку повідомлень. Система мала коректно обробляти порожні відповіді від API та не допускати зависання інтерфейсу.
- Безпека та API-запити: На рівні взаємодії із Supabase перевірялася кожна відповідь сервера. Контролювалося, щоб доступ до таблиць із персональними даними або логами безпеки здійснювався виключно за наявності валідного accessToken.

4.2 Тест-кейси та результати перевірки

Для детальної перевірки працездатності підсистеми безпеки було розроблено набір тест-кейсів, що охоплюють як стандартні сценарії використання, так і обробку виняткових ситуацій. Особлива увага приділялася інтеграційним тестам, які перевіряють взаємодію мобільного клієнта з хмарною інфраструктурою Supabase та параметризацію зовнішніх сервісів.

4.2.1 Позитивне тестування (успішні сценарії)

Для перевірки працездатності розробленої системи було проведено функціональне тестування основних модулів застосунку. Під час тестування перевірялася коректність роботи користувацьких функцій, стабільність взаємодії із серверною частиною та правильність обробки даних. Особливу увагу було приділено роботі сповіщень, авторизації користувачів і навігації до укриттів.

Тестування виконувалося шляхом запуску основних сценаріїв використання системи та порівняння фактичних результатів з очікуваними. Було

перевірено роботу системи при коректному введенні даних і стабільному мережевому з’єднанні. У межах даного етапу були протестовані позитивні сценарії, які забезпечують основний функціонал застосунку.

Результати тестування основних функцій системи наведено в табл. 4.1.

Таблиця 4.1

Результати тестування позитивних сценаріїв

ID	Назва тесту	Очікуваний результат	Статус
TC-01	Авторизація через Google	Створення сесії та перехід на головний екран	Пройдено
TC-02	Отримання Push-сповіщення	Поява банера тривоги на екрані пристрою	Пройдено
TC-03	Навігація до укриття	Відображення маршруту на мапі від поточної локації	Пройдено
TC-04	Зміна налаштувань регіону	Оновлення параметра region_id у базі даних	Пройдено

4.2.2 Негативне тестування та перевірка обмежень доступу

Для забезпечення надійності було протестовано сценарії з некоректним використанням системи та перевірено роботу встановлених безпекових фільтрів. Результати тестування наведено в табл. 4.2.

Таблиця 4.2

Результати тестування негативних сценаріїв та обмежень

ID	Назва тесту	Реакція системи	Статус
TC-05	Доменне обмеження Google Auth	Автоматичне приховування акаунтів без домену @nubip.edu.ua	Пройдено
TC-06	Конфлікт унікальності токена	Успішне оновлення запису через логіку UPSERT без помилок	Пройдено
TC-07	Робота в умовах офлайну	Відображення кешованих даних та вбудованих схем укриттів	Пройдено

Аналіз результатів тестування.

Результати проведених випробувань підтверджують високу стабільність та захищеність підсистеми безпеки:

- Безпека доступу: Застосування параметра hd (Hosted Domain) у конфігурації OAuth дозволило реалізувати фільтрацію користувачів на рівні сервісу ідентифікації Google, що виключає можливість випадкового використання сторонніх акаунтів.
- Цілісність даних: Тестування підтвердило ефективність обробки конфліктів у таблиці push_tokens. Використання операції UPSERT замість простого додавання запису дозволило повністю усунути помилки цілісності бази даних при повторній реєстрації пристроїв.
- Відмовостійкість: Додаток зберігає здатність надавати критичну інформацію про розташування укриттів навіть за умов повної відсутності інтернет-з'єднання завдяки механізмам кешування даних.

4.3 Виявлення недоліків і заходи з їх усунення

У процесі ітеративного тестування підсистеми безпеки було виявлено та успішно усунуто низку технічних недоліків, що дозволило значно підвищити стабільність та надійність програмного продукту. Однією з перших критичних проблем, зафіксованих у системних логах, став конфлікт первинних ключів у базі даних при повторній авторизації користувачів або перевстановленні застосунку. Помилка дублювання значень у таблиці push_tokens блокувала процес реєстрації пристрою для отримання критичних сповіщень. Цей недолік було ліквідовано шляхом заміни стандартної операції вставки на логіку UPSERT із параметром onConflict: 'id', що забезпечило коректне оновлення токенів без порушення цілісності бази даних.

Важливим технічним аспектом, що потребував виправлення, стала синхронізація часових поясів при відображенні часу тривог. Через те, що хмарна інфраструктура Supabase зберігає часові мітки у форматі UTC, дані в інтерфейсі

додатка відображалися з різницею у три години порівняно з реальним часом в Україні. Для розв'язання цієї проблеми реалізовано механізм локальної конвертації часу на рівні клієнтської частини додатка. Тепер дані, отримані з бази, автоматично трансформуються згідно з налаштуваннями часового поясу пристрою користувача (Europe/Kyiv), що гарантує точність інформування студентів.

З метою посилення безпеки було обмежено доступ лише до акаунтів домену @nubip.edu.ua у налаштуваннях Google Auth. Початкова конфігурація, яка дозволяла вхід через будь-який особистий акаунт Google, була обмежена впровадженням параметра hd: 'nubip.edu.ua'. Це дозволило реалізувати жорстку фільтрацію облікових записів безпосередньо на рівні інтерфейсу вибору пошти, надаючи доступ до внутрішніх сервісів безпеки кампусу виключно верифікованим студентам та співробітникам університету.

Крім того, враховуючи специфіку експлуатації додатка в умовах «мертвих зон» зв’язку, характерних для захисних споруд із товстими бетонними перекриттями, було розроблено механізм підвищення відмовостійкості. Замість повної залежності від онлайн-запитів, у систему було інтегровано механізми локального кешування даних та вбудовані статичні план-схеми корпусів НУБіП. Це дає студентам базовий доступ до навігаційної інформації та інструкцій з безпеки навіть за умов повної відсутності інтернет-сигналу. Після виправлення виявлених проблем система почала працювати стабільніше та надійніше в реальних умовах використання

4.4 Оцінка користі, продуктивності й зручності системи

Після проведення функціонального тестування було виконано аналіз ефективності підсистеми безпеки. Основна увага приділялася трьом ключовим метрикам: швидкості доставки інформації (продуктивність), практичній цінності для студентів (користь) та ергономіці інтерфейсу (зручність). Продуктивність та технічна ефективність

Завдяки використанню хмарної інфраструктури Supabase та реляційної бази даних PostgreSQL, швидкість обробки запитів до списку укриттів становить менше ніж 500 мс, що є критично важливим для мобільних мереж. Використання Edge-функцій для синхронізації з API alerts.in.ua дозволило автоматизувати моніторинг загроз, забезпечуючи актуальність даних у режимі 24/7 без необхідності тримати додаток постійно активним на пристрої. Це дозволяє системі працювати з мінімальним навантаженням на акумулятор смартфона студента, зберігаючи при цьому високу готовність до інформування. Практична користь для безпеки кампусу

Основною перевагою системи є те, що вся інформація про укриття та сповіщення знаходиться в одному застосунку. Впровадження підсистеми безпеки дозволяє скоротити час на пошук укриття в незнайомих територіях університету за рахунок автоматизованого прокладання маршрутів через Google Maps API. Наявність офлайн-схем корпусів значно підвищує шанси на успішну орієнтацію студента в умовах відсутності стабільного зв'язку, що часто трапляється в підвальних приміщеннях захисних споруд. Зручність користувацького інтерфейсу (UX)

Інтерфейс застосунку, розроблений на базі React Native, відповідає сучасним стандартам мобільного дизайну та забезпечує високу контрастність елементів, що важливо для легкого зчитування інформації в стресових ситуаціях. Мінімалістична структура головного меню дозволяє отримати доступ до мапи укриттів або останніх сповіщень всього в два дотика. Автоматична фільтрація акаунтів через корпоративний домен @nubip.edu.ua позбавляє користувача необхідності запам'ятовувати додаткові паролі, що робить процес взаємодії із системою максимально безшовним.

4.5 Перспективи масштабування

Першим кроком розвитку проєкту буде забезпечення повної кросплатформенності на рівні дистрибуції. Поточна версія продукту повністю

готова для використання на базі операційної системі Android. Подальше розширення екосистеми застосунку передбачає реєстрацію в Apple Developer Program (що є платною послугою і вимагає щорічного фінансового забезпечення). Також обов'язково треба інтегрувати зі службою Apple Push Notification service (APNs). Після цих кроків можна буде виконувати аналогічне EAS-збирання для платформи IOS. Згодом опублікувати застосунок в AppStore, оскільки це забезпечить 100% охоплення мобільних пристроїв усіх членів академічної спільноти.

Другим кроком є запуск додатка в інших університетах та містах. Використання СУБД PostgreSQL на базі хмари Supabase та чіткому розділенню логіки сповіщень завдяки унікальному ідентифікатору регіонів, архітектуру підсистеми можна легко адаптувати для потреб будь-якого іншого ЗВО України. Для розгортання платформи треба виконати кілька кроків:

- оновити просторові координати, фотоматеріали та місткість сховищ у таблиці shelter під інфраструктуру конкретного кампусу;
- змінити трохи логіку функції, яка при створенні акаунта по дефолту буде ставити регіон ЗВО від якого реєструється користувач.
- змінити програмний фільтр доменного імені на рівні шлюзу Google OAuth для верифікації корпоративних пошт нового навчального закладу.

Третім, перспективним кроком є побудова API-взаємодії з розкладом занять та електронними журналами університету. Це дозволить алгоритмам додатка превентивно моделювати ймовірне місцеперебування студента (номер навчального корпусу, поверх та аудиторію) відповідно до поточного розкладу пар, дня тижня та часу. У разі оголошення повітряної тривоги система зможе автоматично розрахувати й запропонувати оптимальний евакуаційний маршрут до найближчого сховища ще до моменту, коли мобільний пристрій оновить супутникові GPS-координати.

Четвертим кроком стане внутрішня навігація всередині корпусу (Indoor Navigation). Оскільки система вже знає з розкладу, в якій саме аудиторії зараз

перебуває студент, вона може миттєво побудувати для нього точний маршрут евакуації. Від дверей аудиторії – через коридори, найближчі сходи й виходи – прямо до бомбосховища. Для цього використовуватимуться заздалегідь оцифровані плани поверхів. Навіть якщо інтернету не буде, додаток буде працювати в офлайн-режимі і вести людину покроково, як звичайний навігатор, тільки всередині будівлі.

П’ятим кроком є використання штучного інтелекту для розумного прогнозування навантаження на бомбосховища. Точно порахувати, скільки людей насправді знаходиться в укритті в момент тривоги, досить складно. Тому ми пропонуємо інший підхід: ШІ, який вміє добре прогнозувати. Система аналізує розклад занять, розуміє, які групи та викладачі мають бути в конкретному корпусі в певний час, враховує середню відвідуваність і приблизну кількість персоналу. На основі цих даних нейромережа прогнозує, наскільки завантаженим буде сховище під час пари. Наприклад, якщо алгоритм бачить, що в корпусі №3 укриття може заповнитися на 90 % від максимуму, додаток одразу почне скеровувати частину користувачів до іншого найближчого сховища.

ВИСНОВКИ

У межах бакалаврської кваліфікаційної роботи було успішно реалізовано поставлену мету, яка полягала в розробці підсистеми безпеки інтегрованої мобільної системи підтримки студентів НУБіП України. Першим кроком було проведено системний аналіз предметної області. На ньому вже було видно критичну необхідність зібрати один додаток, який поєднає в собі всі необхідні процеси які людина виконує для створення безпечних умов. А порівняльний аналіз існуючих програмних аналогів тільки ще більше підтвердив це.

Систему реалізовано за допомогою сучасного технологічного стеку, і цей вибір було обґрунтовано. Використання Supabase (PostgreSQL) дозволило ефективно організувати зберігання даних, автентифікацію та розсилку критичних сповіщень. Особливу увагу було приділено проєктуванню архітектури, де за допомогою UML-моделювання було детально описано логіку взаємодії компонентів, зокрема інтеграцію з картографічними сервісами Google Maps та API державних систем оповіщення.

Програмна реалізація підсистеми базується на використанні Edge-функцій (Deno) для безперервного моніторингу статусу загроз у реальному часі. Створений алгоритм забезпечує ідентифікацію нових тривог, автоматичну фіксацію «відбоїв» та таргетовану розсилку push-повідомлень користувачам згідно з їхніми регіональними налаштуваннями. Крім того, впроваджено функціонал побудови найкоротшого маршруту до найближчого доступного укриття на основі GPS-координат, що значно скорочує час реакції студента в екстреній ситуації.

Надійність системи забезпечується багаторівневою політикою безпеки, яка включає обов'язкову авторизацію через корпоративні акаунти домену @nubip.edu.ua та механізми Row Level Security (RLS) на рівні бази даних. Результати комплексного тестування підтвердили високу продуктивність підсистеми та її відмовостійкість. У разі втрати інтернет-з’єднання мобільний

застосунок зберігає доступ до локально кешованих даних про укриття, раніше завантажених план-схем та останньої синхронізованої інформації, що дозволяє користувачам орієнтуватися в межах кампусу навіть в офлайн-режимі..

Перспективи подальшого розвитку проєкту полягають у масштабуванні системи на інші заклади вищої освіти та інтеграції з внутрішніми системами розкладу занять НУБіП України. Це дозволить автоматично пропонувати оптимальний маршрут до сховища залежно від того, в якому корпусі перебуває студент згідно з графіком навчання, що ще більше підвищить загальний рівень цивільного захисту академічної спільноти.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України «Про правовий режим воєнного стану» : від 12.05.2015 № 389-VIII. Верховна Рада України : офіційний вебпортал. [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/389-19 – Назва з екрана. – Дата звернення: 14.01.2026.

2. Кодекс цивільного захисту України : від 02.10.2012 № 5403-VI. Верховна Рада України : офіційний вебпортал. [Електронний ресурс]. – Режим доступу: https://zakon.rada.gov.ua/laws/show/5403-17 – Назва з екрана. – Дата звернення: 22.01.2026.

3. Офіційна документація фреймворку React Native. React Native : офіційний сайт. [Електронний ресурс]. – Режим доступу: https://reactnative.dev/docs/getting-started – Назва з екрана. – Дата звернення: 11.02.2026.

4. Офіційна документація платформи Expo. Application services and push notifications : офіційний сайт. [Електронний ресурс]. – Режим доступу: https://docs.expo.dev/ – Назва з екрана. – Дата звернення: 19.02.2026.

5. Офіційна документація хмарної платформи Supabase. Database, Auth, and Edge Functions : офіційний сайт. [Електронний ресурс]. – Режим доступу: https://supabase.com/docs – Назва з екрана. – Дата звернення: 04.03.2026.

6. Офіційна документація СУБД PostgreSQL 16. The PostgreSQL Global Development Group : офіційний сайт. [Електронний ресурс]. – Режим доступу: https://www.postgresql.org/docs/16/index.html – Назва з екрана. – Дата звернення: 12.03.2026.

7. Довідник розширення PostGIS. Spatial and Geographic objects for PostgreSQL : офіційний сайт. [Електронний ресурс]. – Режим доступу: https://postgis.net/docs/ – Назва з екрана. – Дата звернення: 26.03.2026.

8. Документація картографічного сервісу Google Maps Platform. Maps SDK for Android/iOS : вебсайт розробників. [Електронний ресурс]. – Режим доступу: https://developers.google.com/maps/documentation – Назва з екрана. – Дата звернення: 07.04.2026.

9. Довідник середовища виконання Deno. Secure runtime for JavaScript and TypeScript : офіційний сайт. [Електронний ресурс]. – Режим доступу: https://docs.deno.com/runtime/manual – Назва з екрана. – Дата звернення: 16.04.2026.

10. Специфікація RFC 7519: JSON Web Token (JWT). Internet Engineering Task Force (IETF) : офіційний вебпортал. [Електронний ресурс]. – Режим

доступу: https://datatracker.ietf.org/doc/html/rfc7519 – Назва з екрана. – Дата звернення: 23.04.2026.

11. Специфікація протоколу OAuth 2.0 Authorization Framework. IETF : офіційний сайт. [Електронний ресурс]. – Режим доступу: https://oauth.net/2/ – Назва з екрана. – Дата звернення: 29.04.2026.

12. Документація API сервісу alerts.in.ua. Потоковий моніторинг повітряних загроз : портал розробників. [Електронний ресурс]. – Режим доступу: https://devs.alerts.in.ua/ – Назва з екрана. – Дата звернення: 06.05.2026.

Додаток А

Декларація про академічну доброчесність та використання ШІ

Я, Глущенко Владислав Русланович, здобувач НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота на тему «Розробка підсистеми безпеки інтегрованої мобільної системи підтримки студентів НУБІП України» є результатом моєї особистої праці.

2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.

3. Щодо використання інструментів ШІ зазначаю, що:

інструменти ШІ: ChatGPT та Gemini були використані для:

- перекладу іншомовних джерел;
- стилістичного редагування та перевірки граматики;
- допомоги у написанні/відлагодженні програмного коду;
- стилістичного редагування авторського тексту та його приведення до норм академічного письма;
- структурування інформації на основі наданих технічних даних та результатів самостійного проєктування системи;
- верифікації оформлення пояснювальної записки на відповідність методичним вказівкам університету;
- підготовки перекладу анотації англійською мовою

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«15» травня 2026 р. _____ **Владислав ГЛУЩЕНКО**