

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКУ МЕНЕДЖМЕНТУ
ПІДПРИЄМСТВ ДЛЯ РЕАГУВАННЯ ПІД ЧАС
НАДЗВИЧАЙНИХ СИТУАЦІЙ
SOFTWARE IMPLEMENTATION OF THE ENTERPRISE
MANAGEMENT APPLICATION FOR EMERGENCY RESPONSE

Виконав: студент 4 курсу, групи КНТ-111сп
Спеціальності 121 Інженерія програмного
забезпечення

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

ЛУЦЕНКО В.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ЛЕОЩЕНКО С.Д.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 121 Інженерія програмного забезпечення
(код і найменування)
Освітня програма (спеціалізація) Інженерія програмного забезпечення
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
С.О. СУББОТІН
« » 2024 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

ЛУЦЕНКА Владислава Олександровича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій. Software Implementation of the Enterprise Management Application for Emergency Response

керівник проєкту (роботи) д-р філософ., ст.викл. Леощенко Сергій Дмитрович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “24” квітня 2024 року
№ 168

2. Строк подання студентом проєкту (роботи) 31 травня 2024 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ЛЕОЩЕНКО С.Д., ст.викл.		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання «24» квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

(підпис) Владислав ЛУЦЕНКО
(Імя ПРИЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Сергій ЛЕОЩЕНКО
(Імя ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
101 с., 26 рис., 3 табл., 4 дод., 28 джерел.

БЕЗПЕКА, НАДЗВИЧАЙНІ СИТУАЦІЯ, ПОРЯТУНОК ЛЮДЕЙ,
МЕНЕДЖМЕНТ, РОЗРОБКА, МОВА ПРОГРАМУВАННЯ, ПРОГРАМНЕ
ЗАБЕЗПЕЧЕННЯ

Об'єкт дослідження – процес програмної реалізації застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Предмет дослідження – програмні засоби для програмної реалізації застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Метою роботи є програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Матеріали, методи та технічні засоби: мова програмування Java та фреймворк – середовище розробки Android Studio, система керування базами даних Microsoft SQL Server.

Результати. Розроблено застосунок – менеджменту підприємств для реагування під час надзвичайних ситуацій.

Практична цінність роботи полягає у програмній реалізації застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій у вигляді мобільного застосунку.

Висновки. Реалізовано застосунок менеджменту підприємств для реагування під час надзвичайних ситуацій у вигляді мобільного застосунку.

Галузь використання – користувачі смартфонів, працівники підприємств.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 101 pages, 26 figures, 3 tables, 4 appendixes, 28 sources.

SECURITY, EMERGENCY, RESCUE, MANAGEMENT,
DEVELOPMENT, PROGRAMMING LANGUAGE, SOFTWARE

The object of research is the process of software implementation of an enterprise management application for responding to emergency situations.

The subject of the research is software tools for the software implementation of the enterprise management application for responding to emergency situations.

The aim of the work is to provide software implementation of an enterprise management application for responding to emergencies.

Materials, methods, and technical tools: the Java programming language and the Android Studio development environment, the Microsoft SQL Server Database Management System.

Results. An enterprise management application for emergency response has been developed.

The practical value of the work lies in the software implementation of the enterprise management application for emergency response in the form of a mobile application.

Conclusions. Implemented an enterprise management application for responding to emergencies in the form of a mobile application.

Industry of use: smartphone users, employees of enterprises.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок.....	7
Вступ	8
1 Аналіз предметної області.....	15
1.1 Менеджмент підприємств для реагування під час надзвичайних ситуацій	15
1.2 Як використовуються програмні засоби для управління підприємством для реагування на надзвичайні ситуації	19
1.3 Порівняння існуючих аналогів	23
1.4 Постановка завдання	31
1.5 Висновки за розділом 1	35
2 Матеріали і методи.....	36
2.1 Вибір мови програмування	36
2.2 Вибір середовища розробки	45
2.3 Структурна схема розробки.....	52
2.4 Висновки за розділом 2	53
3 Основні рішення щодо реалізації компонентів системи	55
3.1 Проєктування моделей системи	55
3.2 Проєктування інтерфейсу	61
4 Експлуатація, тестування та експериментальне дослідження програми.....	67
4.1 Призначення й умови застосування програми.....	67
4.2 Вимоги до інтерфейсу	68
4.3 Умови виконання програми.....	68
4.4 Методика та результати тестування	68
Висновки.....	72
Перелік джерел посилання.....	73
Додаток А Технічне завдання.....	76
Додаток Б Опис програми.....	82
Додаток В Текст програми.....	89
Додаток Д Слайди презентації	95

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ГІС – географічна інформаційна система;

МПРНС – менеджмент підприємств для реагування під час надзвичайних ситуацій;

ОС – операційна система;

ООП – об'єктно-орієнтоване програмування;

ПЗ – програмне забезпечення;

IDE – інтегроване середовище розробка;

GUI – graphical user interface;

JVM – Java virtual machine;

MVC – model-view-controller;

SDK – software development kit;

UML – unified modeling language.

ВСТУП

Розробка застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій (МПРНС) є важливим і невідкладним завданням з цілого ряду причин, особливо для жителів України. За останні роки Україна зіткнулася з численними надзвичайними ситуаціями, починаючи від стихійних лих і закінчуючи геополітичними конфліктами, включаючи триваючу повномасштабну війну. Ці надзвичайні ситуації підкреслили необхідність надійної, ефективною та ефективною системи управління реагуючими зусиллями та ресурсами [1]–[3].

Географічний і політичний ландшафт України робить її особливо вразливою до різних надзвичайних ситуацій. Стихійні лиха, такі як повені, лісові пожежі та суворі погодні умови, є поширеними, тоді як триваючий конфлікт на сході України призвів до серйозних гуманітарних криз. Ці надзвичайні ситуації вимагають швидкого та скоординованого реагування, щоб мінімізувати їх вплив на життя людей, майно та навколишнє середовище [1]–[3].

Незважаючи на гостру необхідність ефективного управління надзвичайними ситуаціями, існуюча в Україні система стикається з низкою проблем. До них відносяться недостатня координація між відомствами, відсутність інформації в режимі реального часу, неефективний розподіл ресурсів і погана комунікація з громадськістю. Розробка застосунку МПРНС може вирішити ці проблеми, забезпечивши централізовану платформу для управління та координації реагування на надзвичайні ситуації [1]–[3].

Потреба в програмних реалізаціях систем МПРНС в Україні.

Почнемо розгляд з геополітичні викликів та проблем безпеки. Триваючий конфлікт на сході України, що переріс у повномасштабну війну призвів до виникнення складної надзвичайної ситуації, що вимагає складних управлінських рішень. Застосунок МПРНС може допомогти скоординувати

заходи реагування на інциденти, що виникають в результаті військових дій, такі як обстріли, евакуація та надання гуманітарної допомоги [1]–[3].

Централізована система також може підвищити безпеку операцій з реагування, забезпечуючи надійне управління конфіденційною інформацією та обізнаність служб реагування про потенційні загрози [1]–[3].

Територія України схильна до різних стихійних лих, включаючи повені та лісові пожежі. МПРНС може підвищити ефективність реагування на ці події, надаючи дані в режимі реального часу про розвиток стихійних лих, доступність ресурсів та розгортання служб екстреної допомоги [1]–[3].

Інтелектуальна аналітика, інтегрована в застосунок, може допомогти в прогнозуванні стихійних лих і відповідній підготовці, тим самим зводячи до мінімуму їх наслідки [1]–[3].

Конфлікт і стихійні лиха призвели до виникнення значних гуманітарних потреб, включаючи потреби в переміщеному населенні та невідкладної медичної допомоги. МПРНС може сприяти поліпшенню координації між гуманітарними організаціями, урядовими установами та міжнародними організаціями для забезпечення своєчасної та ефективної допомоги [1]–[3].

Застосунок МПРНС може керувати та відстежувати розподіл гуманітарної допомоги, управління житлом та медичними послугами, забезпечуючи оптимальне використання ресурсів [1]–[3].

Ефективне реагування на надзвичайні ситуації вимагає ефективного управління інфраструктурою та ресурсами. Застосунок МПРНС може інтегрувати дані з різних джерел, щоб забезпечити всебічне уявлення про наявні ресурси, такі як аварійні притулки, медичні установи та транспортні мережі [1]–[3].

Така інтеграція може допомогти в швидкому розподілі і розгортанні ресурсів там, де вони необхідні найбільше, скорочуючи час реагування і покращуючи результати [1]–[3].

Однією з основних проблем української системи реагування на надзвичайні ситуації є відсутність чіткої комунікації та координації між різними відомствами та зацікавленими сторонами. Така фрагментація призводить до затримок, надмірних зусиль та прогалин у наданні послуг [1]–[3].

Застосунок МПРНС може забезпечити єдину комунікаційну платформу, яка з'єднує всі зацікавлені сторони, забезпечуючи безперешкодний обмін інформацією та координацію [1]–[3].

Під час надзвичайних ситуацій інформація в режимі реального часу має вирішальне значення для прийняття обґрунтованих рішень. Діюча в Україні система часто спирається на застарілі або неповні дані, що ускладнює ефективне реагування [1]–[3].

Застосунок МПРНС може збирати дані в режимі реального часу з різних датчиків, звітів та соціальних мереж, надаючи особам, які приймають рішення, точну та своєчасну інформацію.

Розподіл ресурсів під час надзвичайних ситуацій часто неефективний, що призводить до втрат та дефіциту. Частково це пов'язано з відсутністю централізованої системи відстеження та управління ресурсами [1]–[3].

МПРНС може оптимізувати розподіл ресурсів, використовуючи алгоритми для приведення ресурсів у відповідність до потреб на основі пріоритету та доступності [1]–[3].

Ефективне управління надзвичайними ситуаціями також передбачає інформування громадськості про ситуацію та заходи безпеки. Діюча в Україні система бореться з поширенням своєчасної і точної інформації серед населення.

Системи МПРНС можуть включати функції оповіщення громадськості та поширення інформації, гарантуючи, що жителі будуть отримувати важливі оновлення та інструкції [1]–[3].

Застосунок МПРНС може спростити взаємодію та координацію між різними службами реагування на надзвичайні ситуації, що призведе до більш

ефективного реагування. Це може значно скоротити час, необхідний для мобілізації ресурсів та персоналу.

Застосунок може автоматизувати багато рутинних завдань, звільняючи людські ресурси для прийняття більш важливих рішень [1]–[3].

Надаючи дані в режимі реального часу та всебічну інформацію про ситуацію, МПРНС може допомогти реагуючим зрозуміти масштаби надзвичайної ситуації. Це розуміння має вирішальне значення для розробки ефективних стратегій реагування [1]–[3].

Інтеграція географічної інформаційної системи (ГІС) може дозволити реагуючим візуалізувати постраждалі райони, планувати маршрути та ефективніше використовувати ресурси. [1]–[3]

Застосунок МПРНС може використовувати аналітику даних для підтримки процесів прийняття рішень. Прогнозна аналітика може прогнозувати потенційні майбутні надзвичайні ситуації на основі історичних даних та поточних тенденцій, що дозволяє вживати активних заходів [1]–[3].

Методи машинного навчання можуть аналізувати закономірності в даних про надзвичайні ситуації, щоб запропонувати найкращі стратегії реагування та розподілу ресурсів.

Застосунок може відстежувати доступність і місце розташування ресурсів в режимі реального часу, забезпечуючи їх розгортання там, де вони найбільш необхідні. Це може запобігти як нестачі, так і надлишку, що призведе до більш ефективного використання ресурсів [1]–[3].

Застосунок МПРНС також може сприяти управлінню волонтерською діяльністю, забезпечуючи ефективне та безпечне використання добровольців.

Системи МПРНС може підвищити громадську безпеку, надаючи своєчасні сповіщення та інформацію про поточні надзвичайні ситуації. Це може допомогти мешканцям вжити належних заходів для захисту себе та свого майна [1]–[3].

Застосунок також може інформувати громадськість про готовність до надзвичайних ситуацій, допомагаючи створити більш стійке співтовариство [1]–[3].

Застосунок МПРНС може бути розроблений для інтеграції з існуючими системами і технологіями управління надзвичайними ситуаціями в Україні. Така інтеграція може розширити можливості існуючих систем і забезпечити більш комплексне рішення [1]–[3].

Застосунок також може сприяти співпраці з міжнародними зусиллями з реагування на надзвичайні ситуації, підвищуючи загальну ефективність допомоги та підтримки [1]–[3].

Для впровадження застосунку МПРНС потрібна надійна технологічна інфраструктура, включаючи надійне підключення до Інтернету та системи управління даними. Забезпечення того, щоб усі регіони, включаючи віддалені та постраждалі від конфліктів райони, мали доступ до цієї інфраструктури, є складним завданням [1]–[3].

Інвестиції в розвиток технологій та інфраструктури необхідні для успішного впровадження МПРНС.

Ефективне використання систем МПРНС вимагає навчання та нарощування потенціалу співробітників служб реагування на надзвичайні ситуації та інших зацікавлених сторін. Це включає технічне навчання використанню програми та розуміння того, як інтегрувати її в існуючі робочі процеси [1]–[3].

Необхідні програми безперервного навчання, щоб користувачі були в курсі нових функцій та найкращих практик.

Управління конфіденційною інформацією, пов'язаною з надзвичайними ситуаціями, вимагає суворих заходів щодо захисту даних та конфіденційності. Застосунок МПРНС повинен бути розроблений з використанням надійних протоколів безпеки для захисту від кіберзагроз та забезпечення цілісності даних [1]–[3].

Необхідно розробити політику та нормативні акти, що регулюють використання даних та захищають приватне життя окремих осіб.

Розробка і підтримка систем МПРНС вимагає значних фінансових вкладень. Вирішальне значення матиме забезпечення фінансування з державних джерел, міжнародних донорів та партнерських відносин з приватним сектором.

Забезпечення довгострокової стійкості програми передбачає планування поточного технічного обслуговування, оновлень і підтримки.

Успішне впровадження застосунків МПРНС вимагає участі та співпраці різних зацікавлених сторін, включаючи державні установи, неурядові організації та приватний сектор. Створення міцних партнерських відносин і створення атмосфери співпраці будуть мати важливе значення для ефективного використання МПРНС [1]–[3].

Розробка програми для управління підприємством для реагування на надзвичайні ситуації є актуальним завданням для України через безліч проблем, з якими стикається країна, включаючи геополітичні конфлікти, стихійні лиха і гуманітарні кризи. Роботодавець може забезпечити централізовану, ефективну платформу для управління заходами реагування на надзвичайні ситуації, покращення координації, розподілу ресурсів та забезпечення громадської безпеки. Незважаючи на значні труднощі при впровадженні, переваги значно переважають перешкоди. При правильних інвестиціях у технології, навчання та залучення зацікавлених сторін застосунок МПРНС може зробити революцію в управлінні надзвичайними ситуаціями в Україні, що в кінцевому підсумку врятує життя і зменшить вплив надзвичайних ситуацій на населення.

Отже актуальною задачею залишається програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Для виконання такої задачі необхідно виконати наступні завдання:

- проаналізувати проблемну область менеджменту підприємств для реагування під час надзвичайних ситуацій;
- виконати дослідження переваг та недоліків програмних рішень для менеджменту підприємств для реагування під час надзвичайних ситуацій;
- обґрунтувати вибір технічних засобів реалізації запропонованого застосунку;
- спроектувати архітектуру застосунку;
- програмно реалізувати застосунок менеджменту підприємств для реагування під час надзвичайних ситуацій;
- провести тестування розробки;
- сформулювати висновки на основі виконаної роботи.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

В першому розділі буде проаналізовано проблемну область менеджменту підприємств для реагування під час надзвичайних ситуацій. Окремо буде досліджено спеціалізоване програмне забезпечення (ПЗ) для МПРНС. На основі дослідження переваг та недоліків програмних рішень для МПРНС сформулювати функціональні та нефункціональні вимоги для нового – конкурентоспроможного програмного рішення.

1.1 Менеджмент підприємств для реагування під час надзвичайних ситуацій

МПРНС відноситься до організованого та стратегічного управління ресурсами, процесами та операціями в організації або між кількома організаціями для ефективного реагування та відновлення надзвичайних ситуацій. Цей підхід об'єднує різні елементи, такі як комунікація, координація, розподіл ресурсів та управління даними, щоб забезпечити безперебійну та ефективну реакцію на будь-яку кризову ситуацію [4]–[7].

Ключові компоненти управління підприємством при реагуванні на надзвичайні ситуації:

- система управління інцидентами – стандартизована ієрархічна структура, що забезпечує скоординоване реагування різних відомств і організацій [4]–[7]. Забезпечує чітке визначення ролей і обов'язків, а також централізацію процесу прийняття рішень, що підвищує ефективність і зменшує плутанину;

- управління ресурсами –включає виявлення, придбання, розподіл та відстеження ресурсів, таких як персонал, обладнання та витратні матеріали [4]–[7]. Забезпечує доступність ресурсів там, де вони найбільше потрібні, оптимізуючи їх використання і скорочуючи втрати;

- комунікаційні системи – створює надійні комунікаційні мережі для полегшення обміну інформацією між службами реагування, установами та громадськістю [4]–[7]. Забезпечує безперешкодний потік точної і своєчасної інформації для підтримки прийняття рішень і координації;

- управління інформацією – збирає, обробляє та аналізує дані, пов'язані з надзвичайною ситуацією, для забезпечення ситуаційної обізнаності та підтримки стратегічного планування [4]–[7]. Використовує такі технології, як ГІС та потоки даних у режимі реального часу для покращення розуміння та розширення можливостей реагування;

- координація та співпраця – передбачає узгодження зусиль різних зацікавлених сторін, включаючи урядові установи, неурядові організації, організації приватного сектору та громадськість [4]–[7]. Сприяє виробленню єдиної стратегії реагування, зводячи до мінімуму дублювання і забезпечуючи взаємодоповнюваність всіх зусиль;

- логістика та управління ланцюгами поставок – керує потоком товарів, послуг та інформації від місця походження до місця споживання [4]–[7]. Забезпечує швидко і ефективно доставку основних предметів постачання і обладнання в постраждалі райони.

Механізм роботи управління підприємством при реагуванні на надзвичайні ситуації.

Етап готовності:

- оцінка ризиків – виявлення потенційних ризиків та вразливостей за допомогою аналізу небезпек та оцінок ризиків; [4]–[7]

- планування – розробка планів реагування на надзвичайні ситуації, включаючи інвентаризацію ресурсів, плани евакуації та стратегії комунікації;

- тренінги та навчання – проведення регулярних тренінгів і тренувань для забезпечення того, щоб всі зацікавлені сторони були знайомі зі своїми ролями і обов'язками [4]–[7].

Етап реагування:

- активація системи управління інцидентами – при виникненні надзвичайної ситуації активується система управління інцидентами і призначається відповідальний за інцидент [4]–[7];
- початкова оцінка – проведення швидкої оцінки для визначення масштабів надзвичайної ситуації;
- мобілізація ресурсів – використання ресурсів на основі потреб, виявлених в ході початкової оцінки;
- план дій щодо інцидентів – розробка стратегічного плану, який визначає цілі, стратегії та потреби в ресурсах для реагування;
- оперативна координація – постійна координація дій всіх реагуючих організацій для забезпечення об'єднаних зусиль.

Етап відновлення:

- оцінка збитку – проведення детальних оцінок для розуміння повного масштабу збитку;
- відновлення служб – надання пріоритетної уваги відновленню критично важливої інфраструктури та послуг;
- підтримка громади – надання підтримки постраждалим особам, включаючи тимчасове житло, медичну допомогу та фінансову допомогу;
- аналіз та вдосконалення – аналіз реагування з метою виявлення отриманих уроків та вдосконалення планів готовності та реагування на майбутнє [4]–[7].

Переваги корпоративного управління при реагуванні на надзвичайні ситуації:

- підвищення ефективності та результативності. Централізоване командування та координація призводять до більш ефективного використання ресурсів та прискорення прийняття рішень [4]–[7]. Стандартизовані процедури зменшують плутанину та оптимізують зусилля з реагування;
- підвищена ситуаційна обізнаність. Управління даними та інформацією в режимі реального часу забезпечує всебічне розуміння надзвичайної ситуації, що дозволяє поліпшити стратегічне планування. ГІС та

інші технології допомагають візуалізувати постраждалі райони та розподіл ресурсів [4]–[7];

- більш ефективний розподіл ресурсів. Оптимізоване управління ресурсами забезпечує розподіл найважливіших матеріалів і персоналу там, де вони найбільш необхідні [4]–[7]. Скорочує втрати і підвищує загальну ефективність реагування;

- розширення взаємодії та кооперації. Просуває єдиний підхід, залучаючи всі зацікавлені сторони до процесу планування та реагування. Зміцнює довіру та співпрацю між різними установами та організаціями;

- підвищує громадську безпеку та впевненість. Ефективне управління надзвичайними ситуаціями вселяє в громадськість впевненість в тому, що їх безпека і благополуччя є пріоритетними [4]–[7]. Своєчасна комунікація та поширення інформації допомагають інформувати громадськість та забезпечувати її готовність.

Проблеми при впровадженні управління підприємством під час реагування на надзвичайні ситуації.

Складність і координація. Управління масштабними заходами реагування за участю багатьох установ та організацій може бути складною справою. Потрібні надійні механізми координації і чіткі канали зв'язку [4]–[7].

Обмеженість ресурсів. Обмеженість ресурсів, особливо в плані персоналу, обладнання та фінансування, може перешкоджати ефективним заходам реагування [4]–[7]. Вимагає ефективного управління ресурсами і розстановки пріоритетів.

Технологічні обмеження. Залежність від технологій зв'язку, управління даними та координації може бути обмеженням, якщо інфраструктура недостатня або виходить з ладу під час надзвичайної ситуації. Необхідні постійні інвестиції в технологічні системи та їх технічне обслуговування [4]–[7].

Навчання та готовність. Забезпечення того, щоб усі зацікавлені сторони були належним чином навчені та підготовлені до надзвичайних ситуацій,

вимагає постійних зусиль та ресурсів. Регулярні навчальні навчання необхідні для підтримки готовності.

Питання регулювання та політики. Орієнтуватися в нормативному та політичному ландшафті може бути складно, особливо якщо мова йде про кілька юрисдикцій. Для підтримки скоординованих зусиль з реагування необхідні чіткі правила і рамки [4]–[7].

МПРНС – це комплексний підхід, який об'єднує різні елементи управління надзвичайними ситуаціями для забезпечення ефективного, скоординованого та результативного реагування на надзвичайні ситуації [4]–[7]. Використовуючи стандартизовані системи, такі як індикаторні системи, оптимізуючи управління ресурсами, покращуючи комунікацію та управління інформацією та сприяючи співпраці, цей підхід може значно покращити результати зусиль з реагування на надзвичайні ситуації. Незважаючи на труднощі в реалізації, переваги значно переважають перешкоди, що робить його найважливішим компонентом сучасних стратегій управління в надзвичайних ситуаціях [4]–[7].

1.2 Як використовуються програмні засоби для управління підприємством для реагування на надзвичайні ситуації

Програми МПРНС працюють шляхом інтеграції різних функціональних можливостей для оптимізації та вдосконалення процесів, пов'язаних з управлінням надзвичайними ситуаціями. Ось детальний огляд того, як вони працюють [8]–[10].

Збір та інтеграція даних. Збір даних у режимі реального часу – застосунки МПРНС збирають дані з різних джерел, включаючи датчики, GPS-пристрої, соціальні мережі, метеорологічні служби та публічні звіти.

Інтеграція з існуючими системами – вони інтегруються з іншими інформаційними системами, такими як ГІС, бази даних та комунікаційні мережі для консолідації даних [8]–[10].

Агрегування даних – агрегує та обробляє ці дані, щоб забезпечити всебічне та актуальне уявлення про ситуацію [8]–[10].

Цілісний погляд – забезпечує єдину картину надзвичайної ситуації в режимі реального часу, сприяючи кращому розумінню ситуації. Обґрунтовані рішення – надає точні дані, які допомагають приймати обґрунтовані рішення.

Управління інцидентами. Система управління інцидентами – реалізує структуровану командну систему для управління операціями реагування. Звітність та ведення журналу інцидентів – дозволяє створювати та оновлювати звіти про інциденти, вести журнал усіх дій та рішень [8]–[10]. Плани дій – полегшує розробку та розповсюдження планів дій у разі інцидентів, які визначають цілі, стратегії та потреби в ресурсах.

Координація – покращує координацію між різними установами та групами, зменшуючи дублювання зусиль. Документація – забезпечує детальне документування дій та рішень, що має вирішальне значення для підзвітності та аналізу після інциденту [8]–[10].

Управління ресурсами. Відстеження запасів – відстежує доступність та розташування ресурсів, включаючи персонал, обладнання та витратні матеріали. Розподіл ресурсів – використовує алгоритми для оптимізації розподілу ресурсів на основі пріоритету та доступності. Управління логістикою – керує логістикою розміщення та поповнення ресурсів [8]–[10].

Ефективність – забезпечує ефективне використання ресурсів, зменшуючи втрати та дефіцит. Швидке розгортання – сприяє швидкому розгортанню ресурсів там, де вони найбільш необхідні [8]–[10].

Комунікація та співпраця. Багатоканальне спілкування – підтримує зв'язок через кілька каналів, таких як SMS, електронна пошта, голосові та миттєві повідомлення [8]–[10]. Єдина платформа – надає всім зацікавленим сторонам єдину платформу для спілкування та обміну інформацією. Оповіщення громадськості – розсилає оповіщення та інформацію громадськості для забезпечення її інформованості та безпеки [8]–[10].

Безперервний потік інформації – забезпечує інформованість всіх зацікавлених сторін, сприяючи скоординованому реагуванню. Громадська безпека – інформує громадськість про заходи безпеки та оновлення під час надзвичайної ситуації [8]–[10].

Аналіз та візуалізація даних. Аналітика в режимі реального часу – надає аналітику в режимі реального часу та звіти про різні аспекти реагування на надзвичайні ситуації. Інтелектуальна аналітика – використовує Історичні дані та алгоритми машинного навчання для прогнозування майбутніх подій та потенційних впливів [8]–[10]. Інструменти візуалізації – використовує ГІС та інші інструменти візуалізації для картографування постраждалих районів, розміщення ресурсів та іншої важливої інформації.

Стратегічне планування – підтримує стратегічне планування та прийняття рішень на основі аналізу даних. Розширене розуміння – покращує розуміння надзвичайної ситуації завдяки візуальному представленню даних [8]–[10].

Аналіз після інциденту. Огляд даних – переглядає та аналізує всі дані, зібрані під час інциденту, для виявлення сильних та слабких сторін реагування. Отримані уроки – документує отримані уроки та найкращі практики для покращення майбутніх зусиль з реагування [8]–[10]. Звітність – створює всеосяжні звіти для зацікавлених сторін з докладним описом заходів реагування та результатів.

Постійне вдосконалення – сприяє постійному вдосконаленню реагування на надзвичайні ситуації, навчаючись на минулих інцидентах. Підзвітність – забезпечує підзвітність і прозорість в управлінні надзвичайною ситуацією [8]–[10].

Слід розглянути важливість програми для управління підприємством для реагування на надзвичайні ситуації.

Покращена координація та ефективність. Застосунки МПРНС полегшують координацію між різними установами та організаціями, які беруть участь у реагуванні на надзвичайні ситуації [8]–[10]. Забезпечуючи

централізовану платформу для спілкування, управління ресурсами та звітування про інциденти, ці програми зменшують надмірність та підвищують ефективність реагування.

Покращена ситуаційна обізнаність. Збір і інтеграція даних в режимі реального часу дозволяють службам реагування отримувати всебічне і актуальне уявлення про надзвичайну ситуацію [8]–[10]. Така розширена ситуаційна обізнаність має вирішальне значення для прийняття обґрунтованих рішень та розробки ефективних стратегій реагування.

Оптимальне використання ресурсів. МПРНС допомагають ефективно розподіляти ресурси та керувати ними. Відстежуючи доступність та розташування ресурсів, ці програми забезпечують розгортання персоналу, обладнання та витратних матеріалів там, де вони найбільш потрібні, мінімізуючи втрати та дефіцит [8]–[10].

Ефективна комунікація. Ефективна комунікація має життєво важливе значення під час надзвичайних ситуацій. МПРНС забезпечують безперебійний зв'язок між службами реагування, установами та громадськістю, забезпечуючи інформованість і координацію всіх учасників. Це знижує ймовірність непорозумінь і допомагає в безперебійному виконанні планів реагування [8]–[10].

Прийняття рішень на основі даних. Надаючи аналітичні та прогнозні дані в режимі реального часу, МПРНС підтримують прийняття рішень на основі даних. Співробітники служби реагування можуть використовувати ці дані для прогнозування майбутніх подій, більш ефективного розподілу ресурсів та коригування стратегій реагування за необхідності [8]–[10].

Громадська безпека та довіра. Інформування населення про надзвичайну ситуацію та заходи щодо реагування на неї підвищує громадську безпеку та зміцнює довіру. Співробітники служби МПРНС можуть відправляти своєчасні оповіщення та оновлення, допомагаючи жителям вживати належних заходів для захисту себе і свого майна [8]–[10].

Навчання після інциденту. Аналіз після інциденту, проведений МПРНС, допомагає зрозуміти ефективність реагування та визначити сфери, які потребують вдосконалення. Цей постійний процес навчання необхідний для підвищення готовності та можливостей реагування на майбутні надзвичайні ситуації [8]–[10].

Корпоративні програми для управління реагуванням на надзвичайні ситуації є найважливішими інструментами для підвищення ефективності зусиль з управління надзвичайними ситуаціями. Завдяки інтеграції різних функцій, таких як управління інцидентами, відстеження ресурсів, комунікація та аналіз даних, ці програми забезпечують комплексне рішення для управління надзвичайними ситуаціями [8]–[10]. Їх важливість полягає в їх здатності покращувати координацію, оптимізувати використання ресурсів, підтримувати рішення на основі даних, забезпечувати ефективне спілкування та підвищувати громадську безпеку. Оскільки надзвичайні ситуації продовжують ускладнюватися, роль рятувальників у забезпеченні швидкого та ефективного реагування стає все більш важливою [8]–[10].

1.3 Порівняння існуючих аналогів

Такі застосунки як: Public Safety by Everbridge, D4H Incident Management і Мапа тривоги – це окремі, але доповнюючі один одного інструменти, які можуть використовуватися в якості застосунків для управління підприємством для реагування на надзвичайні ситуації.

Public Safety by Everbridge – це комплексна платформа для управління критичними подіями, яка надає рішення для масового оповіщення та управління інцидентами для забезпечення громадської безпеки під час надзвичайних ситуацій [11], [12].

Масове попередження. Багатоканальний зв'язок – надсилання сповіщень та сповіщень через кілька каналів, включаючи SMS, електронну пошту, голосові дзвінки, мобільні додатки та соціальні мережі [11], [12].

Цільові сповіщення – дозволяє використовувати геотаргетинг для надсилання конкретних сповіщень у постраждалі райони, гарантуючи, що актуальна інформація дійде до тих, хто її найбільше потребує. Двосторонній зв'язок – забезпечує двосторонній зв'язок з громадськістю, дозволяючи їй реагувати на попередження та надавати інформацію в режимі реального часу [11], [12].

Управління інцидентами. Звітність про інциденти – полегшує створення звітів про інциденти та управління ними, відстежуючи їх хід та статус. Призначення завдань – розподіляє завдання між конкретними реагуючими або групами, забезпечуючи підзвітність та ефективне управління завданнями. Координація ресурсів – керує ресурсами, включаючи персонал та обладнання, та відстежує їх розгортання в режимі реального часу [11], [12].

Ситуаційна обізнаність. Інтеграція ГІС – використовує ГІС для візуалізації інцидентів та розгортання ресурсів на інтерактивних картах, забезпечуючи ситуаційну обізнаність у режимі реального часу [11], [12].

Скоординоване реагування – забезпечує скоординоване реагування, надаючи інформацію в режимі реального часу всім зацікавленим сторонам. Громадська безпека – підвищує громадську безпеку, забезпечуючи своєчасне і точне інформування. Управління ресурсами – оптимізує розподіл ресурсів та управління ними, забезпечуючи ефективне використання ресурсів [11], [12]. Приклади інтерфейсу Public Safety by Everbridge наведено на рис. 1.1.

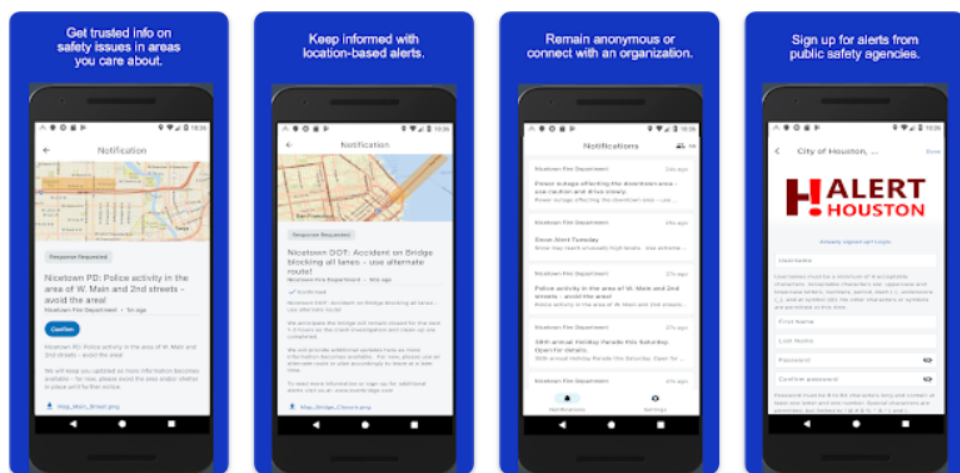


Рисунок 1.1 – Приклади інтерфейсу Public Safety by Everbridge [12]

D4H Incident Management (рис. 1.2) – це інструмент, призначений для груп реагування на надзвичайні ситуації для ефективного планування, управління та документування інцидентів [13], [14].

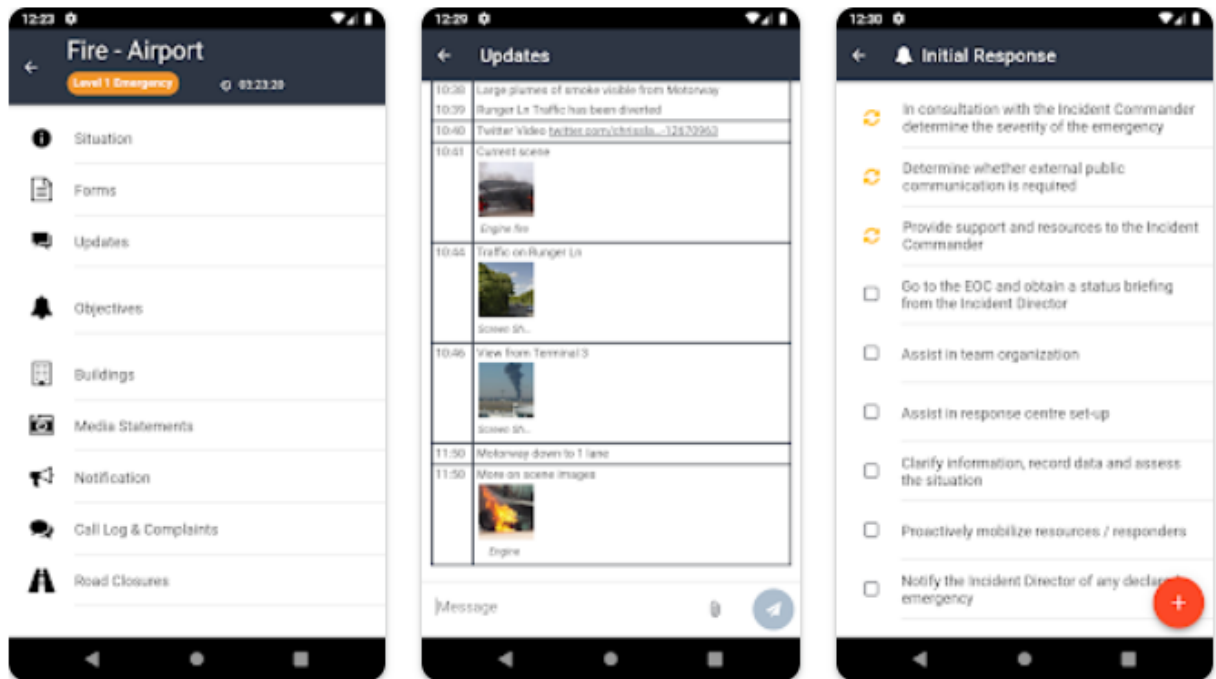


Рисунок 1.2 – Прилади інтерфейсу D4H Incident Management [14]

Створення інцидентів – дозволяє користувачам створювати докладні записи про інциденти, включаючи місце розташування, тип інциденту і необхідні ресурси [13], [14].

Плани дій у випадку інцидентів – полегшує розробку та впровадження планів дій у випадку інцидентів, які визначають цілі, стратегії та завдання.

Управління завданнями – призначає та відстежує завдання, забезпечуючи виконання та документування всіх необхідних дій [13], [14].

Відстеження ресурсів. Управління персоналом – відстежує доступність, місцезнаходження та статус персоналу. Управління обладнанням – відстежує використання обладнання, місцезнаходження та стан технічного обслуговування [13], [14]. Управління запасами – веде облік витратних матеріалів, відстежує їх використання і поповнення.

Комунікація та спільна робота. Спільна робота в команді – надає інструменти для спільної роботи в команді, включаючи обмін повідомленнями та документами. Звіти про ситуацію – створює звіти про ситуацію в режимі реального часу, щоб інформувати всі зацікавлені сторони [13], [14].

Аналіз після інциденту. Ведення журналу інцидентів – веде детальний журнал усіх дій та рішень, прийнятих під час інциденту. Звітність та аналітика – створює детальні звіти та аналітичні матеріали для оцінки реагування та визначення областей, які потребують вдосконалення [13], [14].

Використовується при реагуванні на надзвичайні ситуації. Ефективне управління інцидентами – спрощує управління інцидентами, забезпечуючи структурований підхід до планування та реагування. Оптимізація ресурсів – забезпечує ефективне використання та відстеження ресурсів. Постійне вдосконалення – полегшує аналіз після інциденту для покращення майбутніх зусиль з реагування [13], [14].

Мапа тривог – це застосунок, що використовується в Україні для надання сповіщень в режимі реального часу та інформації про надзвичайні ситуації, включаючи повітряні нальоти та інші загрози [15], [16].

Сповіщення в режимі реального часу. Миттєві сповіщення – надсилає сповіщення в режимі реального часу про повітряні нальоти, загрози вибуху бомби та інші надзвичайні ситуації безпосередньо на пристрої користувачів.

Сповіщення на основі геолокації – використовує геолокацію для надсилання відповідних сповіщень на основі поточного місцезнаходження користувача [15], [16].

Обмін інформацією. Інтерактивні карти – надає інтерактивні карти із зазначенням поточних загроз і безпечних зон, допомагаючи користувачам безпечно орієнтуватися. Публічна інформація – надає громадськості важливу інформацію та інструкції з техніки безпеки від влади [15], [16].

Взаємодія з користувачем. Зворотній зв'язок і звітність – дозволяє користувачам повідомляти про інциденти і надавати зворотний зв'язок, підвищуючи обізнаність про ситуацію [15], [16].

Використовується при реагуванні на надзвичайні ситуації:

- інформування громадськості – інформує громадськість про поточні загрози, підвищуючи їх безпеку;
- аварійна навігація – допомагає користувачам знаходити безпечні маршрути та зони під час надзвичайних ситуацій;
- взаємодія з громадою – дозволяє громаді повідомляти про інциденти та надавати зворотній зв'язок, підвищуючи загальну обізнаність про ситуацію.

Інтерфейс застосунку Мапа тривоги наведено на рис. 1.3.

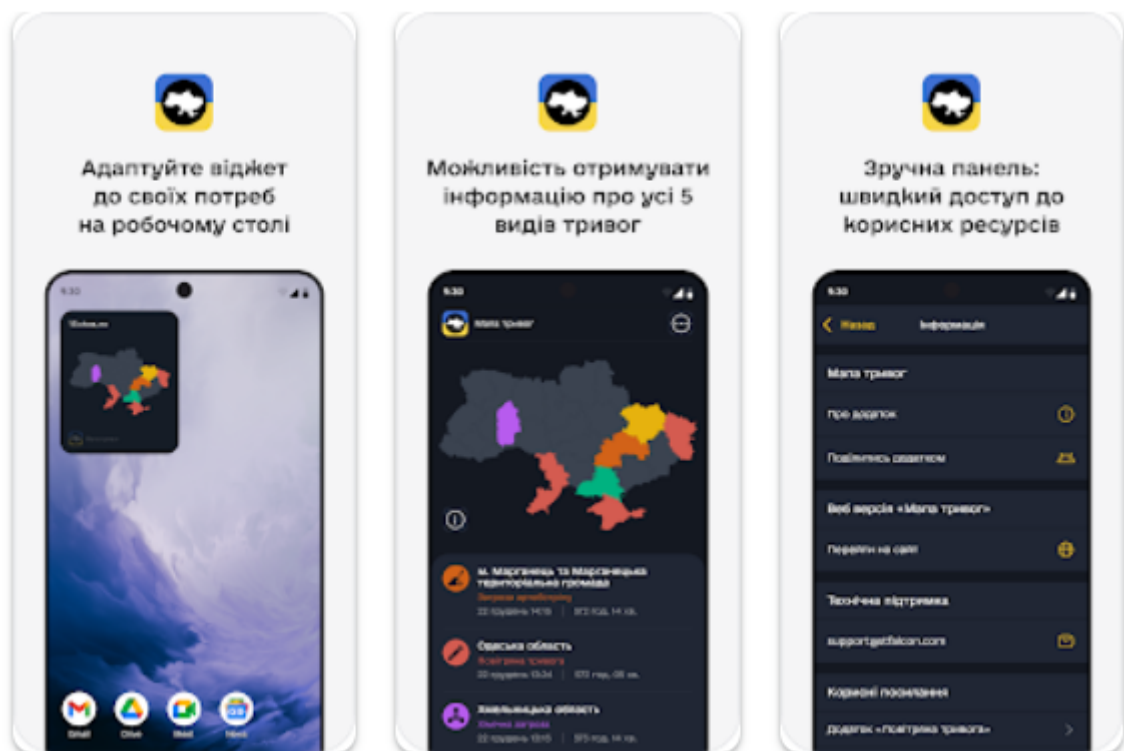


Рисунок 1.3 – Інтерфейси застосунку Мапа тривоги [16]

Public Safety by Everbridge, D4H Incident Management та Мапа тривоги – це потужні інструменти, які при спільному використанні забезпечують комплексний підхід до управління надзвичайними ситуаціями. Вони покращують координацію, комунікацію, управління ресурсами та громадську безпеку, забезпечуючи ефективне реагування на надзвичайні ситуації. Їх

інтеграція та спільне використання пропонують надійне рішення для вирішення складних завдань, пов'язаних із сучасними сценаріями реагування на надзвичайні ситуації.

Порівняємо розглянуті програмні альтернативи та представимо результати у вигляді таблиці – табл. 1.1. Для порівняння використаємо наступні критерії:

- основна функція – розуміння основного призначення кожного застосунку допомагає визначити його основні можливості і передбачуване використання;
- цільові користувачі – знання того, хто є основними користувачами, забезпечує контекст для розробки та функціональності програми;
- оповіщення в режимі реального часу – це найважливіша функція при реагуванні на надзвичайні ситуації, що забезпечує своєчасне поширення важливої інформації;
- управління інцидентами – те, як кожен інструмент обробляє, відстежує та керує звітами про інциденти, є основоположним для операцій з реагування на надзвичайні ситуації;
- управління ресурсами – здатність відстежувати ресурси (персонал, обладнання, витратні матеріали) та керувати ними має важливе значення для ефективного реагування;
- комунікація – ефективна комунікація, як всередині команди, так і з громадськістю, є життєво важливою під час надзвичайних ситуацій;
- ситуаційна обізнаність – інструменти, які надають дані в режимі реального часу, візуалізацію та аналітичні матеріали для розуміння поточного стану надзвичайної ситуації;
- інтеграція даних – можливість інтеграції з іншими системами та джерелами даних покращує функціональність та зручність використання програми;
- інформування громадськості – забезпечення поінформованості громадськості про ситуацію та заходи безпеки має вирішальне значення;

- аналіз після інциденту – можливість аналізу зусиль з реагування після інциденту допомагає покращити стратегії реагування в майбутньому;
- географічна спрямованість – основний регіон або область, де застосунок найбільш ефективно або призначене для використання;
- простота використання – зручність у використанні і рівень підготовки, необхідний для ефективного використання;
- інтеграція з іншими інструментами – наскільки добре застосунок працює з іншими інструментами управління надзвичайними ситуаціями та корпоративними інструментами;
- сильні сторони – виявлення того, що кожен додаток робить найкраще;
- слабкі сторони: виявлення потенційних обмежень або областей, де додаток може бути не таким ефективним.

Таблиця 1.1 – Порівняння застосунків МПРНС

Критерій порівняння	Public Safety by Everbridge	D4H Incident Management	Мапа тривоги
1	2	3	4
Основна функція	Масове оповіщення та управління інцидентами	Планування інцидентів, управління ними та відстеження ресурсів	Сповіщення в режимі реального часу та публічна інформація
Цільові користувачі	Державні установи, корпорації, Охорона здоров'я, освіта	Групи реагування на надзвичайні ситуації, організації громадської безпеки	Широка громадськість, органи влади в Україні
Сповіщення в режимі реального часу	Так, багатоканальне	Обмежений обсяг, корпоративне спрямування	Так, сповіщення на основі геолокації

Продовження таблиці 1.1

1	2	3	4
Управління інцидентами	Так, всебічна звітність та відстеження інцидентів	Так, детальні плани дій у випадку інцидентів та управління завданнями	Обмежується публічною звітністю та навігацією
Управління ресурсами	Так	Так	Ні
Спілкування	Багатоканальний двосторонній зв'язок	Інструменти спільної роботи команди, звіти про ситуацію	Публічні повідомлення та зворотній зв'язок
Поінформованість про ситуацію	Інформаційні панелі в режимі реального часу, інтеграція з ГІС	Звіти про ситуацію в режимі реального часу, ведення журналу та аналітика	Карти в режимі реального часу, що показують загрози та безпечні зони
Інтеграція даних	Широкі можливості	Широкі можливості	Обмежені можливості
Публічна інформація	Так	Обмежений доступ	Так
Аналіз після інциденту	Так	Так	Ні
Географічна спрямованість	Глобальна	Глобальна	Україна
Простота використання	Простий	Важкий	Простий
Інтеграція з іншими інструментами	Так	Так	Обмежена
Сильні сторони	Масове оповіщення, широке охоплення, комплексне управління	Детальне планування та управління ресурсами, спільна робота в команді	Сповіщення в режимі реального часу, загальнодоступна навігація
Слабкість	Вимагає інвестицій	Складний, більш орієнтоване на професійне використання	Обмежується Україною, не розглядається управління ресурсами

Якщо переглянути усі аналоги, то можна відмітити, що існуючі програми покривають дещо різні області. Від так, Public Safety by Everbridge – найкраще підходить для великих організацій та державних установ, які потребують надійних засобів масового оповіщення та управління інцидентами, з сильною інтеграцією та функціями ситуаційної обізнаності. D4h Incident Management – ідеально підходить для команд реагування на надзвичайні ситуації, які потребують детального планування, відстеження ресурсів та інструментів внутрішнього спілкування, а також ретельного аналізу після інциденту. Програма Мапа тривог – орієнтована на надання сповіщень громадськості в режимі реального часу і навігації під час надзвичайних ситуацій в Україні, з простим інтерфейсом для громадського користування і оповіщеннями на основі геолокації.

Саме тому актуальною задачею залишається розробка нового – більш оптимального програмного рішення для різностороннього менеджменту підприємств для реагування під час надзвичайних ситуацій.

1.4 Постановка завдання

Розробка програми для управління підприємством для реагування на надзвичайні ситуації є актуальним завданням для України через безліч проблем, з якими стикається країна, включаючи геополітичні конфлікти, стихійні лиха і гуманітарні кризи. Роботодавець може забезпечити централізовану, ефективну платформу для управління заходами реагування на надзвичайні ситуації, покращення координації, розподілу ресурсів та забезпечення громадської безпеки. Незважаючи на значні труднощі при впровадженні, переваги значно переважають перешкоди. При правильних інвестиціях у технології, навчання та залучення зацікавлених сторін застосунок МПРНС може зробити революцію в управлінні надзвичайними ситуаціями в Україні, що в кінцевому підсумку врятує життя і зменшить вплив надзвичайних ситуацій на населення.

Отже актуальною задачею залишається програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

1.4.1 Функціональність системи

Розробка програми для управління підприємством з метою реагування на надзвичайні ситуації, що враховує політичний контекст війни в Україні, передбачає виконання конкретних функціональних вимог, які забезпечують ефективне управління, координацію і комунікацію під час криз [17]–[19].

Сформуємо та розглянемо детальний список функціональних вимог до застосунку МПРНС [17]–[19]:

- оповіщення та повідомлення в режимі реального часу (багатоканальне сповіщення – можливість надсилати сповіщення за допомогою SMS, електронної пошти, push-сповіщень, соціальних мереж та голосових дзвінків; сповіщення прив'язкою до місця розташування – відправка повідомлень в залежності від географічного розташування користувачів; двосторонній зв'язок – дозволяє одержувачам відповідати на сповіщення для отримання додаткової інформації або підтвердження);

- управління інцидентами (звіти про інциденти – створення та оновлення звітів про інциденти з детальною інформацією; відстеження інцидентів – відстеження статусу та прогресу інцидентів у режимі реального часу; система управління інцидентами – впровадження структурованої системи управління для управління операціями реагування);

- управління ресурсами (управління персоналом – відстеження наявності, місцезнаходження та статусу персоналу; відстеження обладнання та витратних матеріалів – відстеження запасів, місцезнаходження та стану обладнання та витратних матеріалів; розподіл ресурсів – оптимізація розподілу ресурсів на основі пріоритету та доступності);

- ситуаційна обізнаність (інтеграція даних у режимі реального часу – агрегація даних з датчиків, GPS-пристроїв, соціальних мереж та публічних звітів; ГІС-картографія – надає інтерактивні карти, що показують місця аварій, постраждали райони та розміщення ресурсів; інформаційні панелі та аналітика – відображення інформаційних панелей у режимі реального часу з ключовими показниками та аналітикою);

- комунікація та спільна робота (багатоканальне спілкування – підтримка обміну повідомленнями за допомогою SMS, електронної пошти, голосових дзвінків та миттєвих повідомлень; єдина комунікаційна платформа – надає всім зацікавленим сторонам єдину платформу для спілкування та обміну інформацією; поширення публічної інформації – надання громадськості важливої інформації та інструкцій з техніки безпеки);

- безпека даних та конфіденційність (шифрування даних – шифрування збережених та переданих даних для захисту конфіденційної інформації; управління доступом – реалізація управління доступом на основі ролей, щоб гарантувати, що тільки авторизований персонал може отримати доступ до певних даних; дотримання місцевих правил – переконання, що обробка даних відповідає законодавству України та міжнародним стандартам);

- аналіз після інциденту (ведення журналу дій – ведення детального журналу всіх дій та рішень, прийнятих під час інциденту; звітність та аналітика – створення детальних звітів та аналіз даних для оцінки результатів; документація про отримані інструкції – документування отриманих інструкцій та найкращих практик для подальшого використання);

- сумісність (інтеграція з іншими системами – забезпечте сумісність з іншими системами управління в надзвичайних ситуаціях і базами даних; підтримка API – надає API для обміну даними із зовнішніми програмами та платформами; стандартні протоколи – використовуйте стандартні комунікаційні протоколи (наприклад, CAP, EDXL) для обміну даними);

- масштабованість і продуктивність (масштабованість – переконайтеся, що система може масштабуватися для обробки великих обсягів

даних і користувачів в години пік; моніторинг продуктивності – реалізуйте моніторинг продуктивності, щоб система залишалася чуйною та надійною);

– локалізація (мовна підтримка – підтримка декількох мов; локалізований вміст – переконайтеся, що весь вміст відповідає місцевому контексту та культурним особливостям).

Завдяки виконанню цих функціональних вимог застосунок МПРНС буде добре оснащена для вирішення складних і динамічних завдань, пов'язаних з триваючим конфліктом в Україні, забезпечуючи ефективне управління і координацію реагування на надзвичайні ситуації при збереженні суспільної довіри і безпеки.

1.4.2 Вимоги до інтерфейсу

Інтуїтивно зрозумілий дизайн – слід переконатися, що інтерфейс зручний для користувачів і доступний користувачам з різним рівнем технічної підготовки [17]–[19].

Сумісність з різними мобільними пристроями – забезпечення сумісних з мобільними пристроями інтерфейсів для оперативного доступу.

1.4.3 Вимоги до безпеки

Механізми відмови – реалізація механізмів відмови для забезпечення надійності системи під час простоїв.

Резервне копіювання та відновлення – забезпечення регулярного резервного копіювання та надійний план відновлення, щоб запобігти втраті даних [17]–[19].

1.4.4 Не функціональні вимоги до продукту

Апаратні вимоги до сервера:

- центральний процесор – від 16 фізичних вузлів – ядер та 32 логічних потоків з тактовою частотою від 4 GHz;

- оперативна пам'ять формату DDR5, об'ємом від 16 Гб;

- накопичувач (формату HDD або SSD) з вільним простором від 7 Гб.

Програмні вимоги до сервера:

- розгорнута серверна операційна система (ОС), наприклад: Windows Server, Linux Server або macOS Server;

- розгорнута SQL-сумісна СКБД.

Вимоги до клієнта:

- смартфон під управлінням актуальної версії мобільної ОС;

- наявність веббраузера актуальної версії.

1.5 Висновки за розділом 1

В першому розділі було проаналізовано проблемну область менеджменту підприємств для реагування під час надзвичайних ситуацій. Окремо було досліджено спеціалізоване ПЗ для МПРНС. На основі дослідження переваг та недоліків програмних рішень для МПРНС сформувано функціональні та нефункціональні вимоги для нового – конкурентоспроможного програмного рішення.

2 МАТЕРІАЛИ І МЕТОДИ

Робота над другим розділом буде присвячена висвітленню технічних аспектів майбутньої розробки: обґрунтування вибору платформи та технології розробки, вибору мови програмування та середовища розробки для програмної реалізації, затвердження структури майбутньої розробки.

Прийняті рішення стануть основою для подальшої реалізації застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

2.1 Вибір мови програмування

Java вже давно є популярним засобом розробки мобільних застосунків, особливо для Android. Її широке використання – це не просто традиція чи інерція; існують конкретні технічні та практичні причини, чому Java продовжує залишатися популярною мовою для мобільних розробників [20]–[22].

Java була представлена компанією Sun Microsystems (в даний час належить Oracle) у 1995 році. Вона швидко завоювала популярність завдяки своїй незалежності від платформи, завдяки віртуальній машині Java (JVM) [20]–[22].

Найбільш істотне зростання популярності Java в мобільній розробці відбулося з випуском операційної системи Android компанією Google в 2008 році. Android Software Development Kit (SDK) був створений на основі Java, і переважна більшість застосунків для Android були розроблені на Java. Цей історичний зв'язок закріпив місце Java у мобільній розробці за останні півтора десятиліття [20]–[22].

Незалежності Java від платформи сприяє JVM, яка дозволяє Java-додаткам запускатися на будь-якому пристрої, що має сумісну JVM. Це

означає, що код Java можна писати один раз і запускати в будь-якому місці, в тому числі на різних мобільних пристроях [20]–[22].

JVM і байт-код – коли Java-код компілюється, він перетворюється в байт-код, який може бути виконаний JVM. Цей рівень абстракції дозволяє одному і тому ж додатку Java працювати на різних апаратних архітектурах та операційних системах, підвищуючи портативність мобільних застосунків [20]–[22].

Java – це об'єктно-орієнтована мова програмування (ООП), яка забезпечує модульність, повторне використання та підтримку коду. Принципи ООП корисні при розробці мобільних додатків з кількох причин [20]–[22]:

- інкапсуляція – допомагає об'єднати дані та методи, які працюють з даними, в єдине ціле, підвищуючи безпеку даних;
- успадкування – дозволяє створювати нові класи на основі існуючих, сприяючи повторному використанню коду;
- поліморфізм – дозволяє використовувати єдиний інтерфейс для представлення різних базових типів даних, що робить код більш гнучким і розширюваним [20]–[22].

Java може похвалитися багатим набором API та бібліотек, що полегшують розробку мобільних додатків. Ці бібліотеки пропонують готові функціональні можливості, які розробники можуть використовувати, заощаджуючи час і зусилля [20]–[22].

Стандартні бібліотеки: стандартні бібліотеки Java охоплюють все – від структур даних до мереж, вводу-виводу файлів та розробки графічного інтерфейсу [20]–[22].

Сторонні бібліотеки: існує велика екосистема сторонніх бібліотек та фреймворків, які розширюють можливості Java, задовольняючи специфічні потреби в мобільній розробці, такі як обробка зображень, мережа та управління базами даних [20]–[22].

У Java існує велика та активна спільнота розробників. Підтримка спільноти виражається у великій документації, безлічі посібників, активних форумах та регулярних оновленнях. [20]–[22]

Відкритий код: Java та багато її бібліотек та інструментів мають відкритий код, що дозволяє розробникам перевіряти, змінювати та вдосконалювати кодову базу.

Співпраця та підтримка: активна спільнота означає, що розробники можуть легко знаходити рішення проблем, обмінюватися знаннями та співпрацювати над проєктами [20]–[22].

Java відома своєю продуктивністю та економічністю, які мають вирішальне значення при розробці мобільних додатків, де ресурси, такі як пам'ять та обчислювальна потужність, обмежені.

Збір сміття – автоматичне збирання сміття в Java допомагає ефективно керувати пам'яттю, зменшуючи ризик її витоку.

Android SDK, який включає в себе інструменти для створення, тестування і налагодження Android-застосунків, призначений для використання з Java. SDK включає в себе багатий набір API, які дозволяють розробникам взаємодіяти з апаратними та програмними функціями пристрою, такими як камера, GPS та датчики [20]–[22].

Управління життєвим циклом активності: інтеграція Java з Android SDK надає розробникам інструменти для управління життєвим циклом активності, гарантуючи, що програми належним чином реагують на дії користувача та системні події [20]–[22].

Дизайн інтерфейсу користувача – SDK включає бібліотеки для розробки користувацьких інтерфейсів, таких як макети, перегляди та віджети, якими можна керувати за допомогою Java [20]–[22].

Java підтримується потужними інтегрованими середовищами розробки (IDE), такими як Android Studio та Eclipse, які надають комплексні інструменти для розробки, налагодження та тестування програм для Android.

Підтримка коду – такі функції, як завершення коду, інструменти рефакторингу та перевірка помилок у режимі реального часу, підвищують продуктивність.

Налагодження та тестування – вбудовані засоби налагодження та платформи тестування допомагають забезпечити надійність додатків та відсутність помилок.

Хоча Java є основною мовою розробки для Android, екосистема Android також підтримує взаємодію з іншими мовами, такими як Kotlin та C++. Це дозволяє розробникам використовувати існуючий код Java та бібліотеки, одночасно інтегруючи нові функції, написані іншими мовами [20]–[22].

Інтеграція з Kotlin – Kotlin, яка повністю сумісна з Java, стала офіційною мовою розробки для Android. Розробники можуть використовувати Java та Kotlin в одному проєкті, використовуючи переваги обох мов.

Переваги використання Java для розробки мобільних пристроїв.

Розвинена мова – зрілість Java означає, що вона була випробувана в різних середовищах. Стабільність і надійність мають вирішальне значення для мобільних застосунків, які повинні стабільно працювати на різних пристроях і в різних версіях ОС [20]–[22].

Доведена продуктивність – протягом багатьох років Java довела свою продуктивність і масштабованість, що робить її придатною для розробки складних і високопродуктивних мобільних застосунків [20]–[22].

Широка документація – зрілість Java означає, що доступна велика документація, яка допомагає розробникам швидко знаходити рішення своїх проблем.

Java надає надійні функції безпеки, необхідні для мобільних додатків, що працюють з конфіденційними даними.

Security Manager і ізольоване середовище: механізми security manager і ізольованого середовища запобігають виконання потенційно небезпечних операцій ненадійним кодом [20]–[22].

Бібліотеки криптографії – Java включає великі бібліотеки криптографії, які допомагають розробникам впроваджувати безпечне спілкування та зберігання даних.

Кросплатформні можливості Java корисні для розробників, які хочуть працювати з кількома платформами. Хоча Java в основному використовується для розробки на Android, її також можна використовувати для розробки крос-платформних додатків на таких платформах, як Xamarin або Codename One [20]–[22].

Codename One: Codename one дозволяє розробникам писати код на Java та розгорнути його як рідні програми на декількох платформах, включаючи iOS, Android та Windows [20]–[22].

Незважаючи на те, що Java є потужною мовою для розробки мобільних застосунків, вона не позбавлена проблем.

Використання Java в JVM призводить до кращої продуктивності порівняно з рідними мовами, такими як C++.

Паузи при збиранні сміття: автоматичне збирання сміття може призвести до пауз, які можуть вплинути на продуктивність застосунків, що залежать від часу.

Використання пам'яті: програми Java, як правило, використовують більше пам'яті через додатковий рівень JVM.

Java часто критикують за багатослівність, що може призвести до довшого та складнішого коду порівняно з такими мовами, як Kotlin.

Шаблонний код – розробникам потрібно писати більше шаблонного коду, що може уповільнити процес розробки та зробити його більш схильним до помилок [20]–[22].

Зручність читання та супроводу: багатослівність може вплинути на зручність читання та супроводу коду, особливо у великих проєктах.

З введенням Kotlin як офіційної мови розробки для Android, Java стикається з конкуренцією з боку мови, яка пропонує більш сучасні функції та стислість [20]–[22].

Переваги Kotlin: функції Kotlin, такі як захист від null, функції розширення та більш стислий синтаксис, роблять його привабливою альтернативою Java.

Незважаючи на труднощі, Java продовжує розвиватися і зберігати свою актуальність в мобільній розробці.

Пакет JDK продовжує отримувати оновлення, що підвищують продуктивність, безпеку та функціональність. Такі функції, як вдосконалені алгоритми збору сміття, нові мовні можливості та оптимізація продуктивності, підтримують конкурентоспроможність Java [20]–[22].

Google продовжує підтримувати Java в екосистемі Android, гарантуючи, що розробники зможуть використовувати найновіші функції Android за допомогою Java. Постійна інтеграція з Kotlin не зменшує важливість Java, а навпаки, доповнює її [20]–[22].

Java має всі можливості для використання переваг новітніх технологій мобільної розробки, таких як доповнена реальність, віртуальна реальність і машинне навчання.

Бібліотеки машинного навчання: великі бібліотеки та платформи Java підтримують Машинне навчання, яке можна інтегрувати в мобільні додатки для надання розширених можливостей [20]–[22].

Розробка доповненої реальності та віртуальної реальності – продуктивність та можливості Java роблять його придатним для розробки застосунків доповненої реальності та віртуальної реальності, особливо з підтримкою таких фреймворків, як ARCore для Android [20]–[22].

Java залишається домінуючою мовою для розробки мобільних додатків завдяки своїй незалежності від платформи, надійним функціям безпеки, багатому набору API та бібліотек та потужній підтримці спільноти [20]–[22]. Незважаючи на те, що Java стикається з проблемами, пов'язаними з більш сучасними мовами, такими як Kotlin, і властивими JVM високими витратами на продуктивність, її постійний розвиток і широка підтримка з боку екосистеми Android гарантують, що Java як і раніше залишається життєво

важливим інструментом для мобільних розробників. Історичний контекст, технічні переваги та майбутній потенціал Java поєднуються, щоб зробити його привабливим вибором для розробки надійних, масштабованих та безпечних мобільних застосунків [20]–[22].

Для обґрунтування доцільності використання Java проведемо порівняння з іншими мовами програмування, а результат представимо у вигляді табл. 2.1. Порівняння будемо проводити на основі наступних критеріїв:

- офіційна підтримка – визначає, наскільки добре мова інтегрована із засобами розробки Android, і отримує оновлення від Google;
- процес навчання – оцінює легкість, з якою нові розробники освоюють мову, що впливає на час адаптації та навчання;
- синтаксис – оцінює складність та багатослівність мови, що впливає на швидкість розробки та ймовірність помилок;
- незалежність від платформи – враховує можливість запуску додатків на різних платформах, забезпечуючи гнучкість для розробників, орієнтованих на різні середовища;
- продуктивність – аналізується ефективність виконання та використання пам'яті, які мають вирішальне значення для мобільних застосунків з обмеженими ресурсами;
- інструментарій та підтримка IDE – аналізується доступність та якість інструментів розробки, що впливають на продуктивність та простоту розробки;
- спільнота та екосистема – аналізується мережа підтримки, доступність сторонніх бібліотек та фреймворків, які можуть допомогти у розробці;
- сумісність – вимірюється, наскільки легко мова може взаємодіяти з іншими мовами та існуючими кодовими базами, що важливо для підтримки та розширення застарілих систем;

- лаконічність і читабельність – основна увага приділяється обсягу коду, необхідного для виконання завдань, та простоті розуміння коду, що впливає на обслуговування та співпрацю;
- інструменти компіляції та збірки – розглядаються ефективність процесу складання та доступні інструменти, що впливають на цикли розробки;
- попит на ринку праці – показує попит на навички володіння кожною мовою, що впливає на можливості кар'єрного росту і найму;
- рівень впровадження – показує, наскільки широко використовується мова в галузі, що вказує на її актуальність в даний час і перспективи на майбутнє;
- простота переходу для розробників Java – важливо для розробників Java, які розглядають можливість переходу на іншу мову, оцінити, наскільки нова мова схожа на Java, і рівень зусиль, необхідних для переходу.

Таблиця 2.1 – Порівняння мов програмування

Критерій порівняння	Java	Kotlin	C#
1	2	3	4
Офіційна підтримка	Офіційно підтримується Google	Офіційно підтримується Google	Підтримується за допомогою Xamarin/MAUI для кросплатформності
Процес навчання	Помірний	Простий	Від помірного до складного
Синтаксис	Детальний традиційний синтаксис ООП	Лаконічний та сучасний	Схожий на Java та C++
Незалежність від платформи	Запускається незалежно, за допомогою JVM	Запускається незалежно, за допомогою JVM	Кросплатформні можливості за допомогою Xamarin/MAUI

Продовження таблиці 2.1

1	2	3	4
Продуктивність	Хороша, з оптимізацією під JVM	Хороша, з оптимізацією під JVM	З деякими накладними витратами
Інструменти та підтримка IDE	Відмінна підтримка в Android Studio	Відмінна підтримка в Android Studio	Хороша підтримка у Visual Studio та Visual Studio Code
Спільнота та екосистема	Велика, зріла спільнота, потужні ресурси	Зростаюча спільнота	Велика, зріла спільнота, сильна підтримка Microsoft
Сумісність	Рідна мова для Android, легко інтегрується з API Android	Повністю сумісний з Java, легко інтегрується з API Android	Менш сумісні з власними API Android
Лаконічність і читабельність	Багатослівний, вимагає більше шаблонного коду	Лаконічний, не містить шаблонів, покращує читаність	Менш детальний, ніж Java, синтаксис та більш сучасні можливості
Нульова безпека	Відсутні вбудовані функції захисту від null	Вбудовані функції захисту від null	Вбудовані функції нульової безпеки, подібні до Kotlin
Інструменти компіляції та збірки	Gradle, Maven, Ant	Gradle, Maven, Ant	MSBuild, інструменти Xamarin
Кросплатформність	Обмежена Java-фреймворками, такими як Codename One	Обмежені рішеннями на основі JVM	Потужні крос-платформні можливості за допомогою Xamarin/MAUI
Попит на ринку праці	Високий рівень	Зростає	Високий рівень (серед корпоративних клієнтів)
Навчальні ресурси	Великі	Швидко зростають	Швидко зростають

Продовження таблиці 2.1

1	2	3	4
Рівень впровадження	Високий	Швидко зростає кількість застосунків	Стабільний
Інтеграція з сучасними API	Повністю підтримується	Повністю підтримується	Підтримувані
Простота переходу для розробників Java	Рідна мова для Android, перехід не потрібен	Простий перехід	Помірний перехід

Враховуючі устояні механізми та стратегії мови Java для Android розробки, відомість великої екосистеми та активну підтримку спільноти вибір було зроблено саме на користь Java.

2.2 Вибір середовища розробки

Android Studio широко відома як одна з найкращих IDE для розробки Android з кількох вагомих причин. Це перевага надається широкому набору функцій, інтеграції з інструментами, специфічними для Android, та загальному досвіду розробників [23]–[25].

Android Studio – це офіційне середовище розробки для Android, розроблене та підтримуване Google. Це означає, що вона своєчасно отримує оновлення та нові функції, що відповідають останнім версіям Android SDK і API. Як офіційне середовище розробки, воно оптимізоване для розробки Android, забезпечуючи сумісність та продуктивність [23]–[25].

Android Studio об'єднує широкий спектр засобів розробки, які спрощують процес розробки:

- Android SDK Manager – полегшує встановлення та керування версіями та компонентами Android SDK;
- емулятор – включає в себе швидкий і багатофункціональний емулятор для тестування додатків на різних конфігураціях пристроїв і версіях Android без використання фізичних пристроїв;

- система збірки Gradle – використовує Gradle як систему збірки, забезпечуючи гнучкість, продуктивність та розширюваність при управлінні конфігураціями збірки [23]–[25].

Android Studio пропонує розширені можливості редагування коду, які підвищують продуктивність та якість коду:

- інтелектуальне завершення коду – забезпечує завершення коду з урахуванням контексту, що полегшує швидке написання точного коду;

- інструменти Lint – допомагають виявити та усунути проблеми з продуктивністю, зручністю використання, сумісністю версій та іншими проблемами;

- інструменти рефакторингу – підтримують безпечний та ефективний рефакторинг коду, покращуючи зручність супроводу коду;

- підсвічування синтаксису та перевірка помилок – забезпечує підсвічування синтаксису в режимі реального часу та виявлення помилок, що полегшує раннє виявлення та усунення несправностей [23]–[25].

Ефективні налагодження та тестування мають вирішальне значення для розробки надійних додатків, і Android Studio є лідером у цій галузі:

- Debugger – включає в себе потужний налагоджувач з точками зупинки, відстеженням і перевіркою змінних, який допомагає виявляти і усувати проблеми;

- Profiler – надає інструменти для профілювання процесора, пам'яті, мережі та енергоспоживання, допомагаючи оптимізувати продуктивність застосунків [23]–[25];

- підтримка JUnit – інтегрується з JUnit для модульного тестування та підтримує інші платформи тестування, такі як Espresso для тестування інтерфейсу користувача.

Проектування користувацьких інтерфейсів є ключовою частиною розробки Android, і Android Studio надає чудові інструменти для цього:

- редактор макетів – пропонує інтерфейс перетягування для розробки макетів, що полегшує створення складних інтерфейсів користувача [23]–[25];

- макет з обмеженнями – спрощує створення адаптивних макетів, які адаптуються до різних розмірів екрану і орієнтації [23]–[25];

- режими попереднього перегляду – дозволяє розробникам одночасно переглядати макети на екранах різних розмірів та роздільної здатності.

Android Studio легко інтегрується з системами контролю версій, такими як Git, що полегшує співпрацю:

- підтримка систем контролю версій – забезпечує вбудовану підтримку Git, GitHub, Mercurial та інших систем контролю версій;

- управління філіями – полегшує створення філій, об'єднання та вирішення конфліктів, сприяючи спільному розвитку [23]–[25].

Розширюваність Android Studio дозволяє розробникам адаптувати IDE до своїх конкретних потреб:

- екосистема плагінів – підтримує широкий спектр плагінів, які розширюють функціональність, інтегруються з іншими інструментами та підвищують продуктивність [23]–[25];

- спеціальні плагіни – дозволяє розробникам створювати власні плагіни для автоматизації завдань та розширення можливостей IDE.

Android Studio розроблена з урахуванням продуктивності та стабільності, що забезпечує безперебійну розробку:

- кешування в Gradle – використовує кешування при збірці для прискорення часу збірки;

- миттєвий запуск – дозволяє розробникам бачити зміни в своїх застосунках, не перебудовуючи весь проєкт, що прискорює цикл розробки.

Як офіційне середовище розробки, Android Studio підтримується великою документацією та ресурсами. Офіційна документація – всеосяжна і регулярно оновлювана документація, що охоплює всі аспекти розробки під

Android. Велика та активна спільнота розробників, форумів та ресурсів, які надають підтримку та обмін знаннями [23]–[25].

Android Studio доступна в основних операційних системах – працює на Windows, macOS та Linux, дозволяючи розробникам використовувати бажане середовище розробки.

Android Studio вважається однією з найкращих IDE для розробки на Android завдяки своєму широкому та інтегрованому набору інструментів, офіційній підтримці від Google, вдосконаленим функціям редагування та налагодження коду, надійним інструментам розробки інтерфейсу користувача, плавній інтеграції контролю версій, розширюваності, оптимізації продуктивності, а також великій документації та підтримці спільноти. Всі ці функції в сукупності підвищують продуктивність, оптимізують процеси розробки і допомагають розробникам ефективно створювати високоякісні додатки для Android [23]–[25].

Обґрунтування вибору IDE для розробки Android застосунку на основі порівняння існуючих аналогів наведено у табличному вигляді – табл. 2.2. Порівняння виконаємо за наступними критеріями [23]–[25]:

- офіційна підтримка – важлива для забезпечення того, щоб IDE була добре інтегрована з екосистемою Android та отримувала своєчасні оновлення та підтримку від своїх розробників;
- основна увага приділяється – допомагає визначити, чи призначена IDE спеціально для розробки на Android, чи вона служить ширшим цілям, що може вплинути на набір функцій та зручність використання для завдань, пов'язаних з Android;
- інтегровані засоби розробки – важливі для безперебійного процесу розробки. Сюди входять такі інструменти, як менеджери SDK, емулятори та системи побудови, необхідні для розробки на Android;
- інструменти проєктування інтерфейсу користувача – оцінює простоту проєктування та тестування інтерфейсу користувача, що має

вирішальне значення для розробки візуально привабливих та функціональних програм для Android;

- можливості редагування коду – підвищує продуктивність завдяки таким розширеним функціям, як інтелектуальне завершення коду, виявлення помилок в режимі реального часу і потужні інструменти рефакторингу;

- налагоджувач – надійний налагоджувач необхідний для ефективного виявлення та усунення проблем у коді;

- профілювання продуктивності – інструменти для профілювання продуктивності застосунків допомагають оптимізувати використання ресурсів і підвищити загальну продуктивність програми;

- інтеграція з системами контролю версій – повна інтеграція з системами контролю версій є життєво важливою для спільної роботи команди та ведення історії коду;

- інструменти збірки та компіляції – ефективні інструменти збірки скорочують час збірки та спрощують процес розробки, що особливо важливо для великих проєктів;

- розширюваність та плагіни – можливість розширювати функціональність IDE за допомогою плагінів дозволяє розробникам налаштовувати своє середовище розробки відповідно до своїх потреб;

- спільнота та екосистема – велика та активна спільнота надає підтримку, ресурси та сторонні інструменти, які можуть допомогти у розвитку;

- кросплатформна розробка – для розробників, орієнтованих на декілька платформ, підтримка кросплатформ може значно скоротити час та зусилля на розробку;

- швидкість навчання – легкість, з якою початківці розробники освоюють IDE і починають її використовувати, впливає на адаптацію і загальну продуктивність;

- підтримка сучасних API – дозволяє розробникам використовувати новітні функції і можливості платформи Android;

- продуктивність – впливає на загальну якість розробки, оскільки швидке та чуйне IDE може значно підвищити продуктивність;
- база користувачів і популярність – показує рівень впровадження і довіри до середовища IDE, дає уявлення про її надійності та ефективності;
- вартість – модель ціноутворення впливає на доступність для окремих розробників та невеликих команд, а також на бюджетні міркування для великих організацій.

Таблиця 2.2 – Порівняння середовищ розробки

Критерій порівняння	Android Studio	IntelliJ IDEA	Visual Studio with Xamarin
1	2	3	4
Офіційна підтримка	Офіційна IDE від Google	Підтримується JetBrains	Підтримується Microsoft
Основна увага приділяється	Розробка на Android	Універсальна IDE, що підтримує мови Android та JVM	Кросплатформна розробка (Android, iOS, Windows)
Мова програмування	Java, Kotlin	Java, Kotlin, Groovy, Scala і т. д.	C#, F#, VB.NET використання Xamarin за допомогою C#
Інтегровані інструменти розробки	Android SDK, емулятор, система збірки Gradle	Android SDK, емулятор, система збірки Gradle	Xamarin SDK, емулятор Android, MSBuild
Інструменти для проектування інтерфейсу користувача	Редактор макетів, компоновання з обмеженнями, режими попереднього перегляду	Редактор макетів, макет обмежень, режими попереднього перегляду	Xamarin.Форми, XAML Designer, попередній перегляд у реальному часі
Налагоджувач	Потужний налагоджувач	Потужний налагоджувач	Вбудований налагоджувач
Профілювання продуктивності	Широкі можливості	Широкі можливості	Широкі можливості
Інтеграція з системою контролю версій	Вбудовані Git, GitHub, Mercurial, Subversion	Вбудовані Git, GitHub, Mercurial, Subversion	Вбудовані Git, GitHub, Azure

Продовження таблиці 2.2

1	2	3	4
Інструменти для складання та компіляції	Gradle	Gradle, Maven	MSBuild, NuGet
Розширюваність та плагіни	Широка екосистема плагінів	Широка екосистема плагінів	Широка екосистема плагінів
Кросплатформна розробка	В першу чергу Android	В основному для Android	Повна кросплатформна підтримка
Спільнота та екосистема	Велика активна спільнота, офіційні ресурси Google	Велика активна спільнота, ресурси JetBrains	Велика активна спільнота, ресурси Microsoft
Крива навчання	Помірний	Помірний	Помірний
Підтримка сучасних API-інтерфейсів	Повна підтримка новітніх застосунків для Android	Повна підтримка новітніх застосунків для Android	Підтримка API для Android з потенційною затримкою при запуску нових функцій
Уявлення	Оптимізовано для Android-розробки	Оптимізовано для розробки на базі Android і JVM	Оптимізована для крос-платформної роботи
Вартість бази користувачів і популярності	Широко використовується серед розробників Android	Популярні серед розробників Java і Kotlin	Широко використовується серед розробників
Офіційна підтримка	Безкоштовна та платна	Безкоштовна та платна	Безкоштовна та платна

Серед головних недоліків Android Studio зазвичай виділяють слабкий рівень підтримки кросплатформної розробки. Проте враховуючі набір переваг, а саме: офіційну підтримку від Google, комплексні інструменти для розробки під Android та обширну документацію з активною підтримкою, а також цілеспрямованість на розробку мобільного Android застосунку— саме Android Studio є найбільш потужним інструментом.

2.3 Структурна схема розробки

Загальна схема системи за обраною темою на рис. 2.1.

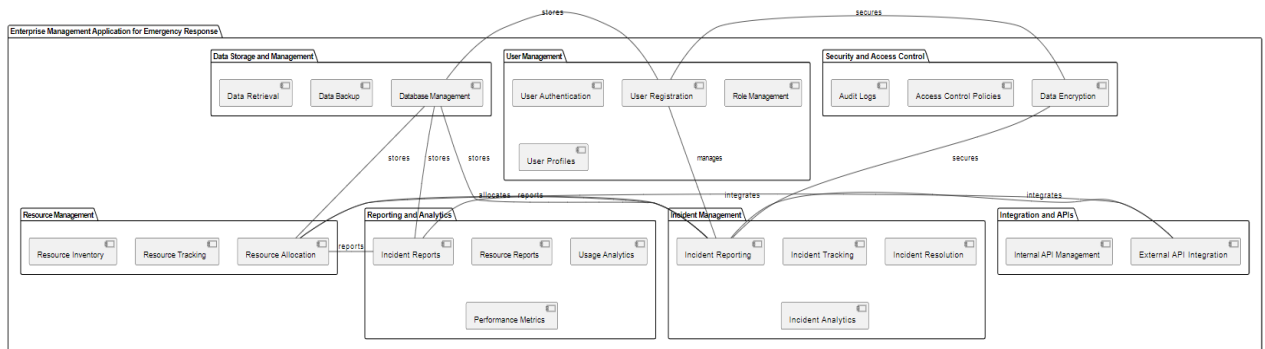


Рисунок 2.1 – Загальна схема системи за обраною темою

Розглянемо структуру спроектованих модулів:

- управління користувачами (реєстрація користувачів – модуль для реєстрації користувачів; аутентифікація користувачів – модуль для входу в систему і перевірки автентичності користувача; управління ролями – управління різними ролями та дозволами користувачів; профілі користувачів – управління інформацією про профіль користувача);
- управління інцидентами (створення звітів про інциденти – модуль для створення звітів про нові інциденти; відстеження інцидентів – відстеження статусу зареєстрованих інцидентів; дозвіл інцидентів – модуль для управління дозволом інцидентів; аналітика інцидентів – аналіз даних про інциденти для отримання інформації);
- зв'язок і повідомлення (SMS-повідомлення – відправка SMS-повідомлень і повідомлень-повідомлень по електронній пошті; повідомлення по електронній пошті – відправка сповіщень і повідомлень по електронній пошті; push-повідомлення – відправка push-повідомлень на мобільні пристрої; обмін повідомленнями в застосунку – внутрішній обмін повідомленнями в застосунку);
- управління ресурсами (розподіл ресурсів: виділення ресурсів для реагування на інциденти; відстеження ресурсів: відстеження стану та

місцезнаходження ресурсів; інвентаризація ресурсів: управління інвентаризацією ресурсів);

- звітність та аналітика (звіти про інциденти – створення звітів про інциденти; звіти про ресурси – створення звітів про використання ресурсів; аналітика використання – аналіз шаблонів використання програми; показники продуктивності – вимірювання показників ефективності зусиль з реагування);

- інтеграція та API (інтеграція із зовнішнім API – інтеграція із зовнішніми системами та службами; управління внутрішнім API – управління внутрішніми API для модульної взаємодії);

- безпека та контроль доступу (шифрування даних – шифрування конфіденційних даних; політики контролю доступу – визначення та застосування політик контролю доступу; журнали аудиту – ведення журналів для цілей аудиту);

- зберігання даних і управління ними (управління базою даних – управління базою даних для програми; резервне копіювання даних – резервне копіювання даних для запобігання втрати; пошук даних – ефективне вилучення даних у міру необхідності).

Ця схема та пояснення дають структуроване уявлення про основні компоненти, що використовуються в додатку управління підприємством для реагування на надзвичайні ситуації, демонструючи їх взаємодію та функціональні можливості.

2.4 Висновки за розділом 2

Робота над другим розділом була присвячена висвітленню технічних аспектів майбутньої розробки: обґрунтування вибору платформи та технології розробки, вибору мови програмування та середовища розробки для програмної реалізації, затвердження структури майбутньої розробки.

Прийняті рішення стануть основою для подальшої реалізації застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проєктування моделей системи

Моделювання засобами Unified Modeling Language (UML) є найважливішим етапом в розробці програмних додатків, особливо тих, які пов'язані з ризиками для життя людини, таких як системи реагування на надзвичайні ситуації, медичні програми та авіаційне програмне забезпечення. Важливість моделювання UML у цих умовах можна пояснити кількома ключовими факторами, які допомагають зменшити ризики та забезпечити безпеку, надійність та ефективність програмного забезпечення [26]–[28].

Візуальне представлення – UML забезпечує візуальне представлення системи, що полегшує розуміння архітектури та функціональності системи всіма зацікавленими сторонами, включаючи розробників, дизайнерів та нетехнічний персонал [26]–[28].

Уніфікована мова – UML служить загальною мовою для всіх членів команди, зменшуючи непорозуміння та гарантуючи, що всі працюють на одній хвилі. Це особливо важливо в критично важливих для безпеки додатках, де непорозуміння може призвести до фатальних помилок [26]–[28].

Простежуваність вимог – моделі UML допомагають відстежувати вимоги протягом усього процесу розробки, гарантуючи, що всі критичні для безпеки вимоги будуть враховані і підтверджені. Така простежуваність має вирішальне значення для дотримання стандартів і приписів в області безпеки.

Раннє виявлення проблем – завдяки моделюванню системи на ранній стадії процесу розробки можна виявити і усунути потенційні проблеми і прогалини у вимогах до того, як вони стануть більш серйозними проблемами під час впровадження [26]–[28].

Детальна проєктна документація – UML-діаграми містять детальну документацію по дизайну та архітектурі системи, яка необхідна для розуміння

взаємодії різних компонентів. Ця документація необхідна для підтримки та модифікації системи, особливо в критично важливих додатках.

Узгодженість і цілісність – UML забезпечує узгодженість і цілісність конструкції системи, запобігаючи помилки, які можуть поставити під загрозу безпеку. Наприклад, діаграми послідовностей можуть допомогти перевірити, чи відбуваються взаємодії у правильному порядку, а діаграми стану можуть гарантувати, що система правильно переходить з одного стану в інший [26]–[28].

Визначення точок відмови – моделювання на мові UML допомагає виявити потенційні точки збою і уразливості в системі. Розуміючи, як взаємодіють компоненти, розробники можуть передбачити можливі збої та розробити надійні рішення для зменшення цих ризиків [26]–[28].

Аналіз безпеки – методи аналізу безпеки, такі як аналіз дерева несправностей та аналіз характеру та наслідків несправності, можуть бути інтегровані з моделями UML для систематичної оцінки та усунення потенційних небезпек.

Дотримання стандартів – у багатьох галузях, де важлива безпека, діють суворі стандарти і нормативні вимоги (наприклад, ISO 26262 для автомобільної промисловості, DO-178C для авіації). Моделі UML допомагають гарантувати відповідність програмного забезпечення цим стандартам, забезпечуючи чіткий і перевіряється процес проєктування [26]–[28].

Готовність до сертифікації – детальні моделі UML та детальна документація полегшують процес сертифікації, надаючи чіткі докази безпеки та надійності системи, які часто потрібні для затвердження регуляторами.

Створення тестових прикладів – моделі UML можна використовувати для систематичного генерування тестових випадків. Наприклад, діаграми випадків використання можуть допомогти визначити функціональні тести, а діаграми стану можуть допомогти у розробці тестів для різних станів та переходів [26]–[28].

Моделювання та прототипування – моделі UML дозволяють проводити Раннє моделювання і створення прототипів, що дозволяє перевірити працездатність системи в контрольованому середовищі. Це раннє тестування допомагає виявити та усунути проблеми до того, як система буде розгорнута в реальних сценаріях [26]–[28].

Простота оновлення – моделі UML забезпечують чітку схему системи, спрощуючи її оновлення та підтримку. Це особливо важливо в додатках, критично важливих для безпеки, де необхідні постійні оновлення та вдосконалення для усунення нових ризиків і змін в нормативних актах [26]–[28].

Передача знань – детальна документація по UML забезпечує ефективну передачу знань про систему між членами команди, знижуючи ризик виникнення помилок при приєднанні до проєкту нових розробників або відхід існуючих.

Діаграми прецедентів – визначають взаємодію між користувачами і системою, допомагаючи забезпечити облік і правильну реалізацію всіх користувальницьких сценаріїв, особливо тих, які пов'язані з критичними операціями [26]–[28].

Діаграми класів – моделювати статичну структуру системи, включаючи класи, атрибути, операції та взаємозв'язки. Це допомагає гарантувати, що компоненти системи правильно визначені та належним чином взаємодіють.

Діаграми послідовності – ілюструють взаємодію об'єктів у певній послідовності подій. Це важливо для перевірки того, що критичні процеси відбуваються в правильному порядку.

Діаграми станів – представляють стану і переходи системи або її компонентів, гарантуючи, що система поводить себе коректно у всіх можливих умовах.

Діаграми активностей – моделюють робочі процеси і дії в системі, допомагаючи виявляти і аналізувати складні процеси і їх потенційні точки збою.

Моделювання на мові UML є важливим етапом у розробці критично важливих для безпеки програмних додатків, оскільки воно забезпечує чіткий, деталізований і систематичний підхід до проєктування і документації. Сприяючи поліпшенню комунікації, ретельному аналізу вимог, точної специфікації проєкту, ефективному зниженню ризиків, дотриманню нормативних стандартів, розширеному тестуванню і простоті обслуговування, UML-моделювання вносить значний вклад в надійність і безпеку програмних систем, які несуть ризики для життя людини [26]–[28]. Такий ретельний підхід гарантує, що розроблена система є надійною, захищеною і здатна безпечно працювати в будь-яких очікуваних умовах.

Діаграма класів (рис. 3.1) ілюструє основні класи та їх взаємозв'язки в застосунку:

- User Management (User: представляє Користувача з такими атрибутами, як `userId`, `name`, `email`, `password` та `role`; Role: представляє роль з такими атрибутами, як `roleId`, `roleName` та список дозволів; Permission: являє собою дозвіл з такими атрибутами, як `permissionId` і `permissionName`);
- Incident Management (Incident: це інцидент з такими атрибутами, як `incidentId`, `title`, `description`, `reportedAt`, `reportedBy`, `status`, та a list of resources; IncidentStatus: представляє статус інциденту з такими атрибутами, як `statusId` та `statusName`);
- Communication and Notification (Notification: представляє сповіщення з такими атрибутами, як `notificationId`, `message`, `sentAt`, та `type`; NotificationType: представляє тип сповіщення з такими атрибутами, як `typeId` та `typeName`);
- Resource Management (Resource: представляє ресурс з такими атрибутами, як `resourceId`, `resourceName`, `type`, та `status`; ResourceType: представляє тип ресурсу з такими атрибутами, як `typeId` та `typeName`; ResourceStatus: представляє статус ресурсу з такими атрибутами, як `statusId` and `statusName`);

- Reporting and Analytics (Report: представляє звіт з такими атрибутами, як reportId, reportName, generatedAt, та type; ReportType: представляє тип звіту з такими атрибутами, як typeId та typeName);
- Integration and APIs: представляє зовнішній API з такими атрибутами, як apiId, apiName, and endpoint;
- Security and Access Control: AuditLog представляє журнал аудиту з такими атрибутами, як logId, timestamp, action, and performedBy;
- Data Storage and Management: Database представляє базу даних з такими атрибутами, як databaseId and databaseName.

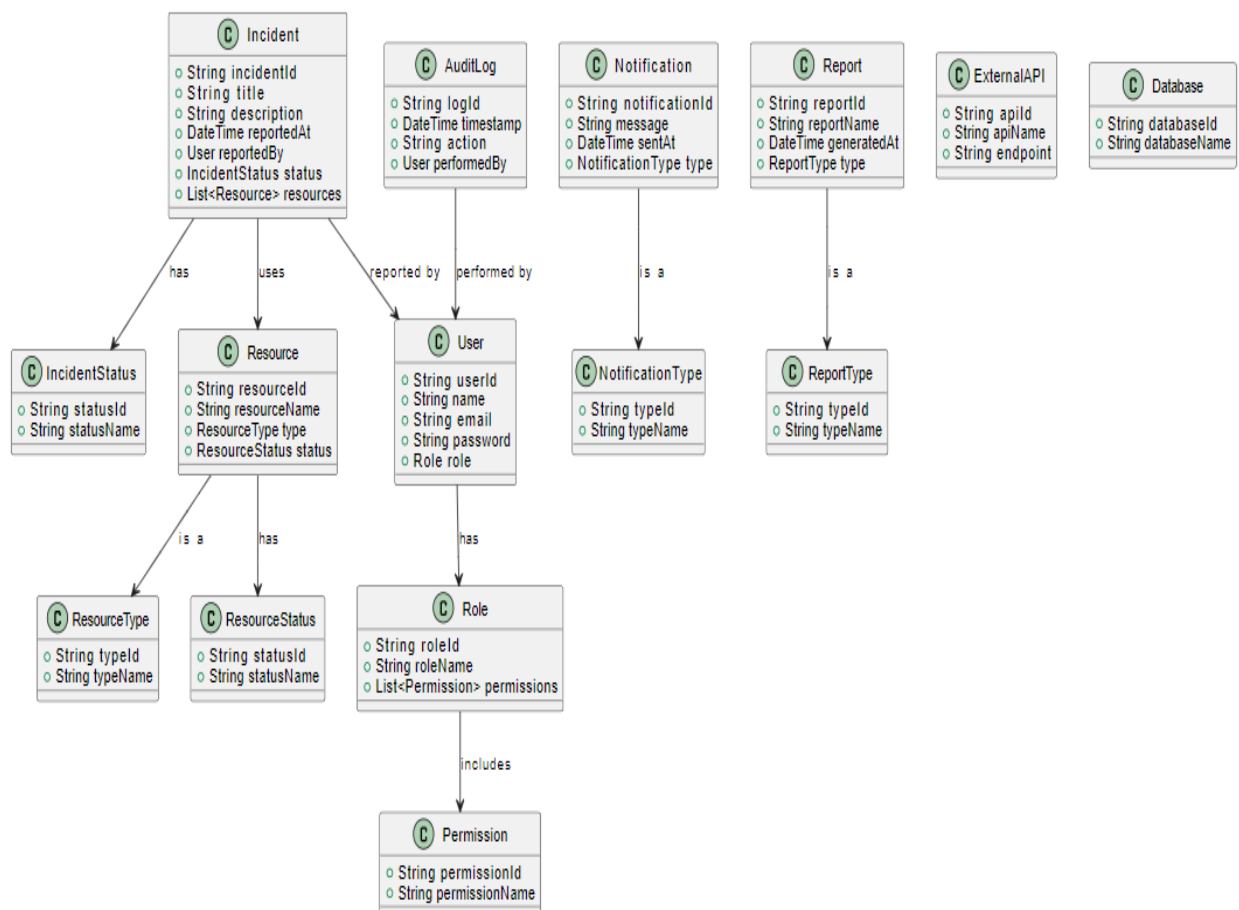


Рисунок 3.1 – Діаграма класів

Діаграма пакетів (рис. 3.2) ілюструє організацію програми у вигляді трьох основних пакетів:

- пакети – кожен основний компонент програми для управління підприємством (управління користувачами, управління інцидентами, зв'язок і оповіщення і т.д.) представлений у вигляді пакета;
- класи – кожен пакет визначає класи, що представляють об'єкти та компоненти, з відповідними атрибутами;
- зв'язки – зв'язки між класами зображуються за допомогою простих ліній зі стрілками, що показують, як сутності співвідносяться між собою в системі.

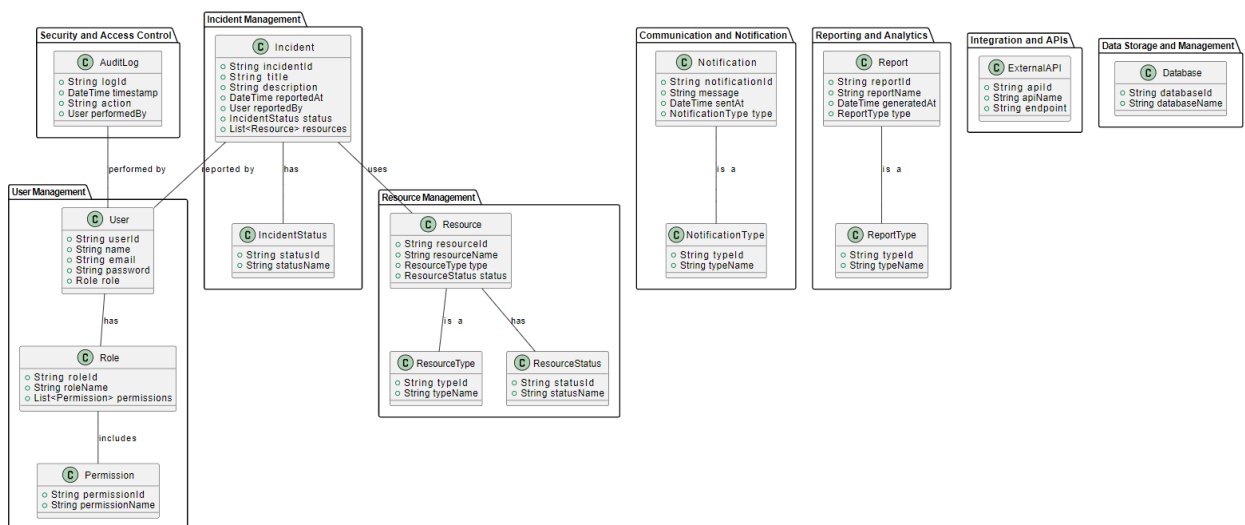


Рисунок 3.2 – Діаграма пакетів

UML діаграма прецедентів (рис. 3.3) ілюструє різні дії, які користувач може виконувати.

Дійові особи: представлені у вигляді простих фігур (User, Administrator, External API), дійові особи взаємодіють з системою для виконання різних випадків використання.

Випадки використання: представлені у вигляді овалів (Report Incident, Allocate Resources, Notify Users, Generate Reports, Manage Users, Manage Roles), це дії або служби, що надаються системою.

Межа системи: Представлена прямокутником (Enterprise Management Application), який об'єднує всі випадки використання та учасників у системі.

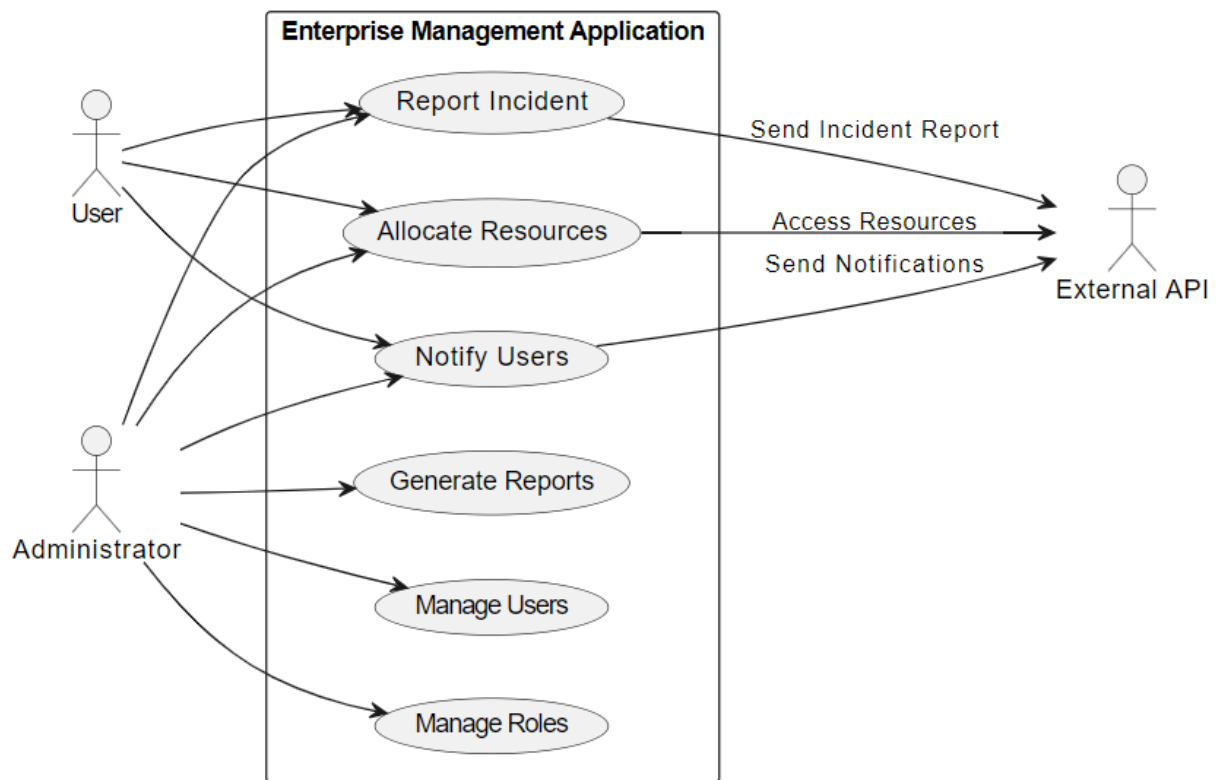


Рисунок 3.3 – Діаграма прецедентів

3.2 Проектування інтерфейсу

Створення графічного інтерфейсу користувача (GUI) для програми МПРНС включає в себе кілька важливих аспектів, що забезпечують ефективність, зручність у використанні і задовольняють потреби служб реагування на надзвичайні ситуації та адміністраторів [1], [17], [19].

Дизайн, орієнтований на користувача, спрямований на розуміння потреб, уподобань та поведінки користувачів для створення інтуїтивно зрозумілих та ефективних інтерфейсів. Для МПРНС, яка може використовуватися в умовах високого навантаження та критичних ситуаціях, принципи дизайну орієнтованого на користувача мають вирішальне значення [1], [17], [19].

Дослідження користувачів – проведіть дослідження, щоб зрозуміти ролі користувачів (наприклад, співробітників служб екстреного реагування, адміністраторів) і їх завдання. Це включає опитування користувачів,

спостереження та тестування юзабіліті, щоб отримати уявлення про їхні робочі процеси та проблеми.

Розробка персоналій – створення персонажів, що представляють типових користувачів, з докладним описом їх цілей, навичок і проблем. Це допомагає дизайнерам співпереживати користувачам і розробляти інтерфейси, що відповідають їх конкретним потребам [1], [17], [19].

Аналіз завдань – розбивка завдань, що виконуються користувачами під час реагування на надзвичайні ситуації, таких як звітність про інциденти, розподіл ресурсів та комунікація. Розробка інтерфейсів для полегшення виконання цих завдань, зменшення когнітивного навантаження та потенційних помилок.

Визначення функціональних вимог та розставлення пріоритетів залежно від цілей програми та потреб користувачів. До ключових функцій, як правило, відносяться [1], [17], [19].

Звіти про інциденти – інтуїтивно зрозумілі форми або інтерфейси, що дозволяють користувачам швидко та точно повідомляти про інциденти. Це може включати в себе визначення місця розташування, оцінку ступеня небезпеки і завантаження мультимедійних матеріалів (наприклад, фотографій, відео) [1], [17], [19].

Управління ресурсами – інтерфейси для перегляду, розподілу та відстеження ресурсів (наприклад, персоналу, транспортних засобів, обладнання). Візуальні уявлення (наприклад, карти, діаграми) можуть допомогти у розподілі ресурсів та моніторингу.

Звітність та аналітика – інструменти для звітування про статистику інцидентів, використання ресурсів, час реагування та інші показники. Візуальні інформаційні панелі можуть надавати корисну інформацію для прийняття рішень [1], [17], [19].

Інформаційна архітектура – розробка інтуїтивно зрозумілої інформаційної архітектури, яка логічно організовує вміст та підтримує ефективну навігацію [1], [17], [19]:

Ієрархічна структура – групування пов'язаних функціональних можливостей (наприклад, управління інцидентами, розподіл ресурсів) за чіткими категоріями.

Пошук та фільтрація – впровадження надійних механізмів пошуку та фільтрації, які допоможуть користувачам швидко знаходити інформацію (наприклад, про інциденти, ресурси) на основі таких критеріїв, як місцезнаходження, статус або тип [1], [17], [19].

Послідовна навігація – підтримка узгодженості елементів навігації (наприклад, розташування меню, термінології) на екранах, щоб зменшити когнітивне навантаження та покращити зручність використання.

Принципи проєктування інтерфейсу – застосування основних принципів проєктування для створення візуально привабливих та функціональних інтерфейсів [1], [17], [19].

Рекомендації щодо дизайну інтерфейсу користувача – дотримання рекомендацій щодо дизайну для певної платформи (наприклад, веб, мобільний) (наприклад, Material Design, рекомендації щодо інтерфейсу iOS), щоб забезпечити послідовність та зручність використання.

Візуальна ієрархія – використання ієрархій для визначення пріоритетів важливої інформації та дій. Використовувати розмір, колір, контрастність та інтервали, щоб ефективно привернути увагу користувачів.

Типографіка та читабельність – вибір зручних для читання шрифти та розміри для різних роздільних здатностей екрана. Забезпечення контрастності тексту для зручності читання, особливо в умовах підвищеного навантаження або недостатньої освітленості [1], [17], [19].

Адаптивний дизайн – розроблення інтерфейси, які адаптуються до різних розмірів і орієнтації екранів (наприклад, настільних моніторів, планшетів, смартфонів) для забезпечення гнучкості сценаріїв використання.

Розробка GUI для корпоративного застосунку управління для реагування на надзвичайні ситуації передбачає ретельне планування, орієнтацію на користувача та дотримання функціональних вимог. Визначаючи

пріоритетність зручності використання, доступності, безпеки та продуктивності, розробники можуть створювати інтерфейси, які дозволять службам реагування на надзвичайні ситуації та адміністраторам ефективно справлятися з кризовими ситуаціями та забезпечувати громадську безпеку. Безперервна ітерація, заснована на відгуках користувачів і технологічних досягненнях, ще більше підвищує зручність використання програми і його оперативність реагування в динамічних надзвичайних ситуаціях [1], [17], [19].

На рис. 3.4 – 3.7 – наведено концепти майбутнього застосунку.

Концепт головної сторінки користувача наведено на рис. 3.4.

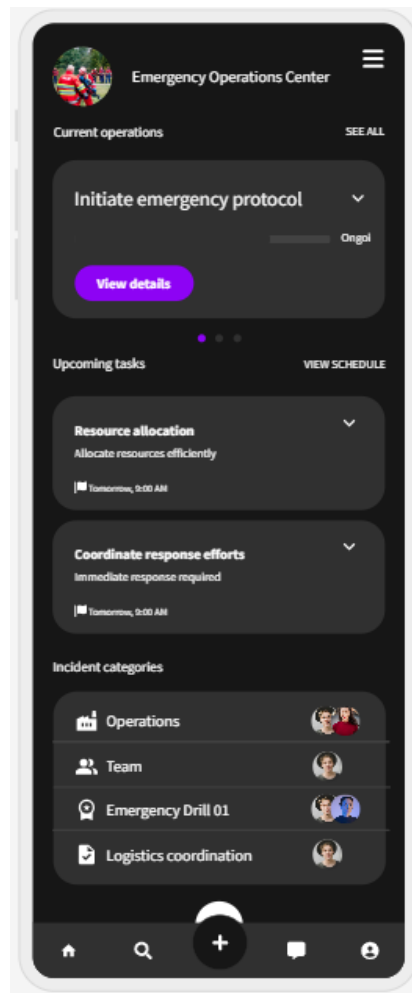


Рисунок 3.4 – Концепт головної сторінки користувача

Концепт вікна з вибором інцидентів наведено на рис. 3.5.

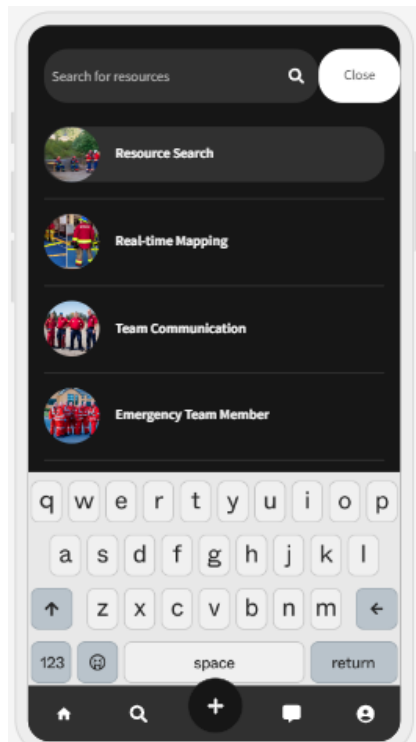


Рисунок 3.5 – Концепт вікна з вибором інцидентів

Концепт вікна з мапою для навігації наведено на рис. 3.6.

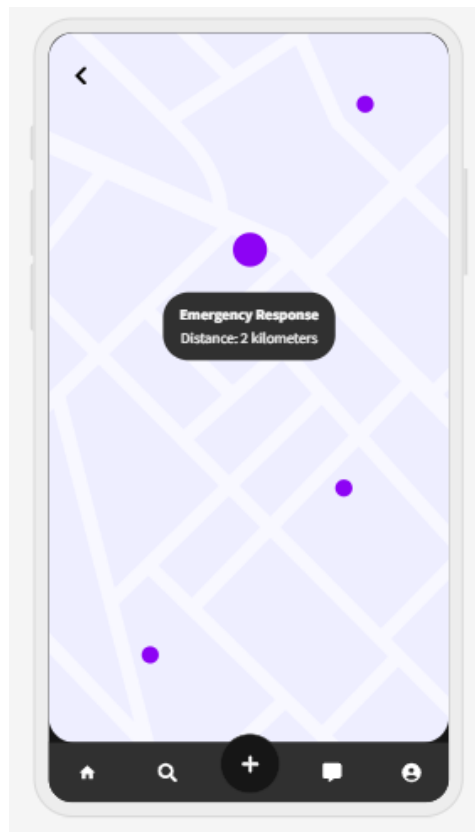


Рисунок 3.6 – Концепт вікна з мапою для навігації

Концепти вікон для оповіщення та комунікації наведено на рис. 3.7.

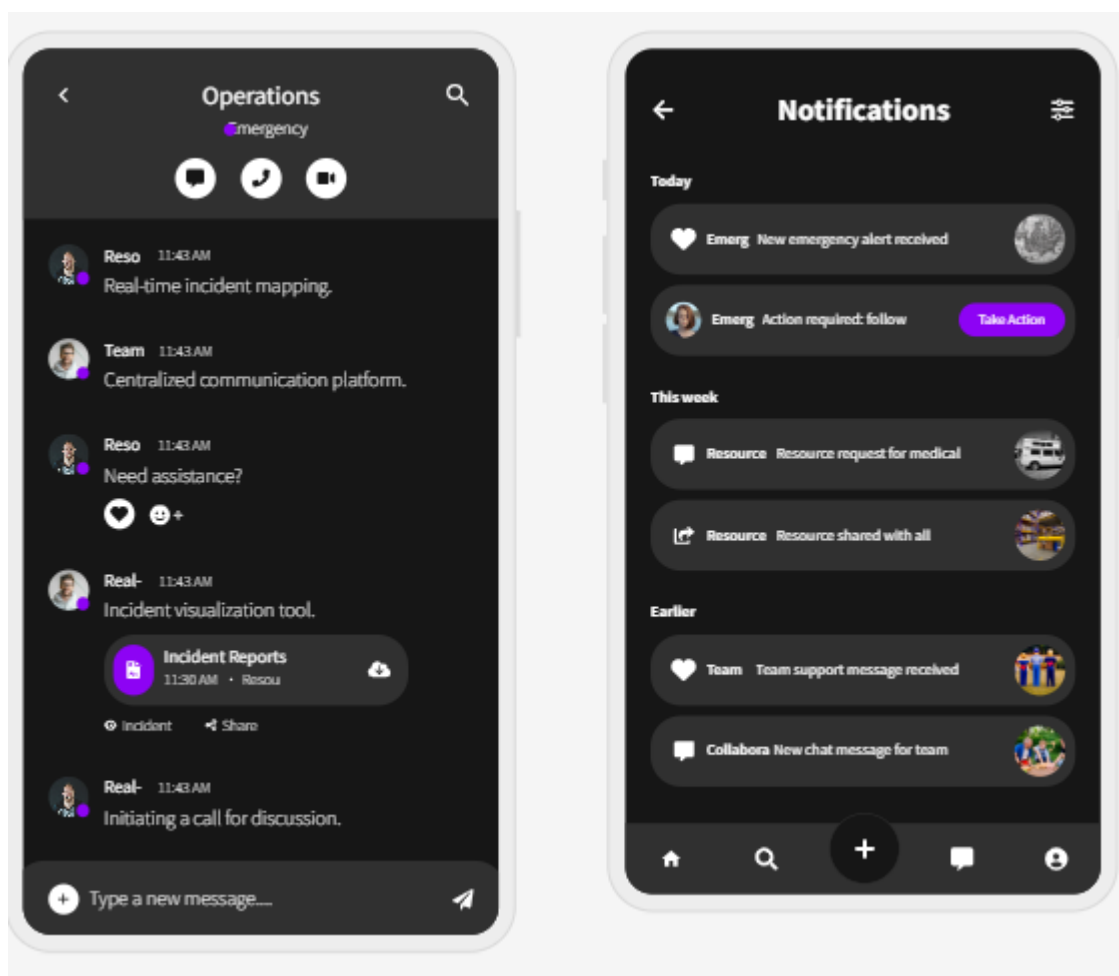


Рисунок 3.7 – Концепти вікон для оповіщення та комунікації

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Розробка програми для управління підприємством для реагування на надзвичайні ситуації є актуальним завданням для України через безліч проблем, з якими стикається країна, включаючи геополітичні конфлікти, стихійні лиха і гуманітарні кризи. Роботодавець може забезпечити централізовану, ефективну платформу для управління заходами реагування на надзвичайні ситуації, покращення координації, розподілу ресурсів та забезпечення громадської безпеки. Незважаючи на значні труднощі при впровадженні, переваги значно переважають перешкоди. При правильних інвестиціях у технології, навчання та залучення зацікавлених сторін застосунок МПРНС може зробити революцію в управлінні надзвичайними ситуаціями в Україні, що в кінцевому підсумку врятує життя і зменшить вплив надзвичайних ситуацій на населення.

Отже актуальною задачею залишається програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Система має реалізовувати наступні функціональні вимоги:

- оповіщення та повідомлення в режимі реального часу;
- управління інцидентами;
- управління ресурсами;
- ситуаційна обізнаність;
- комунікація та спільна робота;
- безпека даних та конфіденційність;
- аналіз після інциденту;
- сумісність;
- масштабованість і продуктивність.

4.2 Вимоги до інтерфейсу

Інтуїтивно зрозумілий дизайн – слід переконатися, що інтерфейс зручний для користувачів і доступний користувачам з різним рівнем технічної підготовки.

Сумісність з різними мобільними пристроями – забезпечення сумісних з мобільними пристроями інтерфейсів для оперативного доступу.

4.3 Умови виконання програми

Апаратні вимоги до сервера:

- центральний процесор – від 16 фізичних вузлів – ядер та 32 логічних потоків з тактовою частотою від 4 GHz;
- оперативна пам'ять формату DDR5, об'ємом від 16 Гб;
- накопичувач (формату HDD або SSD) з вільним простором від 7 Гб.

Програмні вимоги до сервера:

- розгорнута серверна ОС, наприклад: Windows Server, Linux Server або macOS Server;
- розгорнута SQL-сумісна СКБД.

Вимоги до клієнта:

- смартфон під управлінням актуальної версії мобільної ОС;
- наявність веббраузера актуальної версії.

4.4 Методика та результати тестування

Під час тестування було обрано стратегію поетапного виконання за сценарієм основного алгоритму користування застосунком.

На рис. 4.1 наведено приклад сторінки з користувацькою статистикою протягом доби роботи застосунку.



Рисунок 4.1 – Сторінка з користувацькою статистикою протягом доби

На рис. 4.2 наведено поточні оповіщення для користувача.

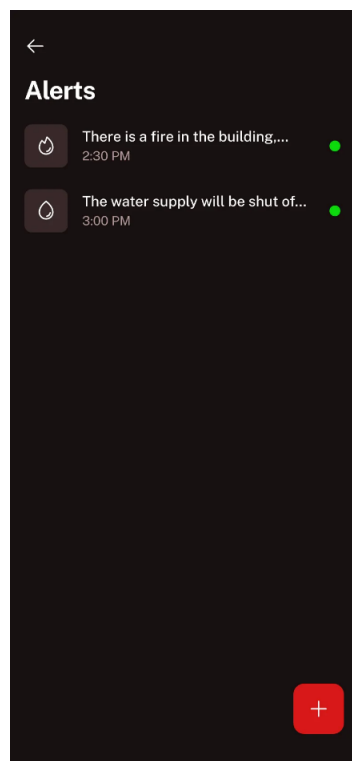


Рисунок 4.2 – Оповіщення для користувачів

На рис. 4.3 наведено інформацію у вигляді звіту про надзвичайну ситуацію.

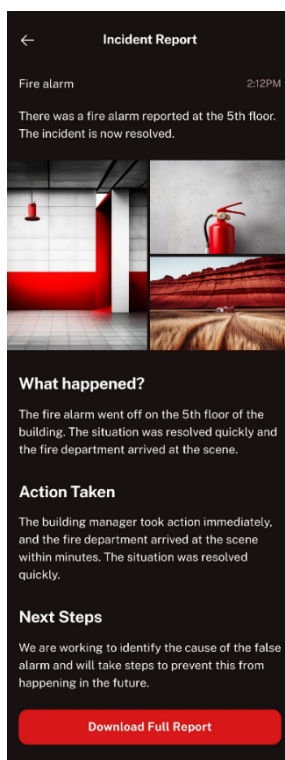


Рисунок 4.3 – Інформація у вигляді звіту про надзвичайну ситуацію

На рис. 4.4 наведено приклади створення інцидентів.

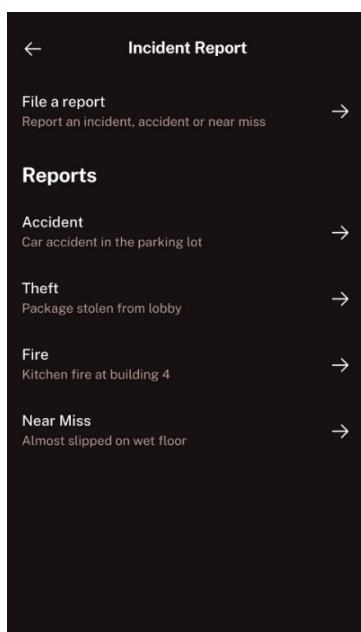


Рисунок 4.4 – Приклади створення інцидентів

На рис. 4.5 наведено поточну мапу повітряних тривог.

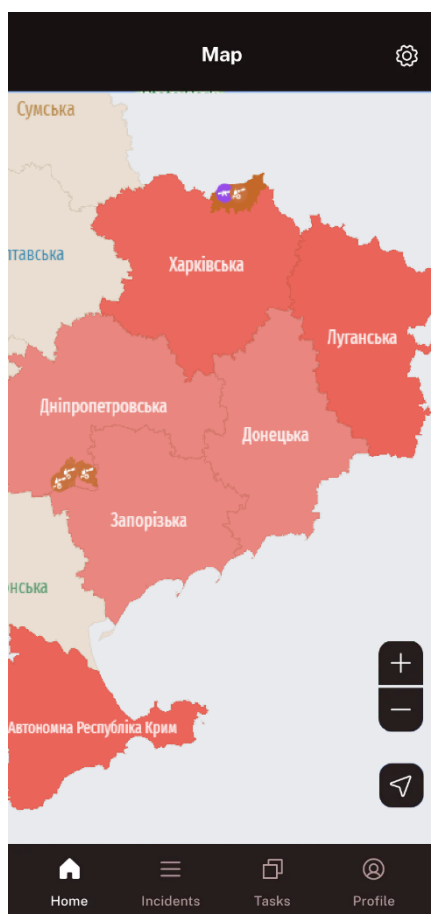


Рисунок 4.5 – Поточна мапа повітряних тривог

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було виконано програмну реалізацію застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій. Для цього було виконано ряд завдань:

- проаналізовано проблемну область менеджменту підприємств для реагування під час надзвичайних ситуацій;
- виконано дослідження переваг та недоліків програмних рішень для менеджменту підприємств для реагування під час надзвичайних ситуацій;
- обґрунтовано вибір технічних засобів реалізації запропонованого застосунку;
- спроектовано архітектуру застосунку;
- реалізовано застосунок менеджменту підприємств для реагування під час надзвичайних ситуацій;
- проведено тестування розробки;
- сформовано висновки на основі виконаної роботи.

Розроблений застосунок був реалізований із використанням: мови програмування Java та середовища розробки – фреймворку .Android Studio.

Застосунок було протестовано відповідно до сценарію його використання. Тестування засвідчило коректність роботи.

В результаті розробки реалізовано систему, що відповідає всім вимогам визначеним на початку розробки у вигляді функціональних вимог. В майбутньому можливо вдосконалити систему додавши реалізацію рекомендаційної системи для визначення поведінки в залежності від інциденту, що трапився.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Ляшенко О. М. Розроблення інформаційної технології оперативного реагування на надзвичайні ситуації [Текст] / О. М. Ляшенко, Р. С. Ложкін // Екологічні науки. - 2020. - № 2(1). - С. 106-112.
2. Порока С. Г. Надзвичайні ситуації як фактор загрози національній безпеці України [Текст] / С. Г. Порока, В. Г. Кононович // Вісник Національного університету цивільного захисту України. Серія : Державне управління. - 2023. - Вип. 1. - С. 334-338.
3. Мельниченко О. А. Механізми публічного управління реагуванням на надзвичайні ситуації: сутність та види [Текст] / О. А. Мельниченко // Науковий вісник: державне управління. - 2020. - № 3. - С. 165-176.
4. Іванець Г. В. Організаційно-технічний метод забезпечення готовності реагування на надзвичайні ситуації [Текст] / Г. В. Іванець, В. В. Матухно, М. Г. Іванець, Є. І. Стецюк, Д. В. Усачов, Д. Ю. Бугайов // Проблеми надзвичайних ситуацій. - 2021. - № 1. - С. 206-221.
5. How to Develop Emergency Management System Software / [Electronic resource] – <https://stfalcon.com/en/blog/post/how-to-develop-emergency-management-system-software>
6. The Complete Enterprise Emergency Management Guide / [Electronic resource] – <https://www.ilobby.com/blog/emergency-management-guide>
7. Crisis Management Software: Better equip your business with our smart & reliable system / [Electronic resource] – <https://www.econline.com/health-and-safety/crisis-management-software>
8. Top 10 Emergency Management Software of 2024 / [Electronic resource] – <https://safetyculture.com/app/emergency-management-software/>
9. Best Emergency Management System / [Electronic resource] – <https://www.softwaresuggest.com/emergency-management-system>
10. Best Emergency Management Software / [Electronic resource] – <https://www.g2.com/categories/emergency-management>

11. Public safety / [Electronic resource] –
<https://www.everbridge.com/solutions/public-safety/>
12. Public Safety by Everbridge / [Electronic resource] –
<https://play.google.com/store/apps/details?id=com.everbridge.mobile.resident&hl=uk&gl=US>
13. D4H Incident Management / [Electronic resource] –
<https://www.d4h.com/products/incident-management-software>
14. D4H Incident Management / [Electronic resource] –
<https://play.google.com/store/apps/details?id=org.d4h.incidentmanagement&hl=en>
15. «Повітряна тривога» 3.0, офіційна мапа тривоги та API застосунку — нове від Ajax Systems, Stfalcon і Мінцифри / [Electronic resource] –
<https://ajax.systems/ua/blog/air-alert-3-0/>
16. Карта тривога
<https://play.google.com/store/apps/details?id=trivogamap.in.ua&hl=uk>
17. Functional and Non-functional Requirements for Mobile Applications / [Electronic resource] – <https://cm-strategies.com/functional-and-non-functional-requirements-for-mobile-applications/>
18. 12 Key NFR Considerations for Mobile Application Development / [Electronic resource] – <https://www.globallogic.com/insights/blogs/nonfunctional-requirements-nfr-mobile-app-development/>
19. Mastering Non-Functional Requirements for Mobile Application Development / [Electronic resource] – <https://dtundwal.medium.com/mastering-non-functional-requirements-for-mobile-application-development-d77e3235fc0d>
20. Як працює Android. Введення для Java-розробників / [Electronic resource] – <https://javarush.com/ua/groups/posts/uk.481.jak-pracju-android-vvedennja-dlja-java-rozrobnikv>
21. Introduction to Android App Development with Java / [Electronic resource] – <https://medium.com/@digvijayndmitteamg/introduction-to-android-app-development-with-java-2025e504628b>

22. Kotlin vs Java for Android Development / [Electronic resource] – <https://medium.com/codex/kotlin-vs-java-for-android-development-7cce3e90ed08>
23. Android Studio / [Electronic resource] – <https://developer.android.com/studio>
24. Accessing View Elements in Android: The Good, The Bad and The Ugly / [Electronic resource] – <https://medium.com/@toby.jones/accessing-view-elements-in-android-the-good-the-bad-and-the-ugly-b60ef7f8b0e2>
25. Using Java AspectJ tool to profile your Android Application performance / [Electronic resource] – <https://dashitesh.medium.com/android-method-profiler-tool-amp-using-aspectj-5c85af8ed599>
26. Integrating UML Modeling into Agile Software Development: A Guide for Scrum and Kanban Teams / [Electronic resource] – <https://guides.visual-paradigm.com/integrating-uml-modeling-into-agile-software-development-a-guide-for-scrum-and-kanban-teams/>
27. Learning UML Diagrams Will Make You a Better Developer / [Electronic resource] – <https://builtin.com/software-engineering-perspectives/uml-diagrams>
28. The UML in the Age of Agile: Why It's Still Relevant / [Electronic resource] – <https://simpleprogrammer.com/unified-modeling-language-age-of-agile/>

ДОДАТОК А
Технічне завдання

Вступ

Даний застосунок присвячений менеджменту підприємств для реагування під час надзвичайних ситуацій.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій», затверджене наказом Національного університету «Запорізька політехніка» № 168 від 24 квітня 2024 р.

A.2 Основні вимоги до програми, що розробляється

Розробка програми для управління підприємством для реагування на надзвичайні ситуації є актуальним завданням для України через безліч проблем, з якими стикається країна, включаючи геополітичні конфлікти, стихійні лиха і гуманітарні кризи. Роботодавець може забезпечити централізовану, ефективну платформу для управління заходами реагування на надзвичайні ситуації, покращення координації, розподілу ресурсів та забезпечення громадської безпеки. Незважаючи на значні труднощі при впровадженні, переваги значно переважають перешкоди. При правильних інвестиціях у технології, навчання та залучення зацікавлених сторін застосунок МПРНС може зробити революцію в управлінні надзвичайними ситуаціями в Україні, що в кінцевому підсумку врятує життя і зменшить вплив надзвичайних ситуацій на населення.

Отже актуальною задачею залишається програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

А.2.1 Вимоги до функціональних характеристик

Розробка програми для управління підприємством з метою реагування на надзвичайні ситуації, що враховує політичний контекст війни в Україні, передбачає виконання конкретних функціональних вимог, які забезпечують ефективне управління, координацію і комунікацію під час криз.

Сформуємо та розглянемо детальний список функціональних вимог до застосунку МПРНС:

- оповіщення та повідомлення в режимі реального часу (багатоканальне сповіщення – можливість надсилати сповіщення за допомогою SMS, електронної пошти, push-сповіщень, соціальних мереж та голосових дзвінків; сповіщення прив'язкою до місця розташування – відправка повідомлень в залежності від географічного розташування користувачів; двосторонній зв'язок – дозволяє одержувачам відповідати на сповіщення для отримання додаткової інформації або підтвердження);

- управління інцидентами (звіти про інциденти – створення та оновлення звітів про інциденти з детальною інформацією; відстеження інцидентів – відстеження статусу та прогресу інцидентів у режимі реального часу; система управління інцидентами – впровадження структурованої системи управління для управління операціями реагування);

- управління ресурсами (управління персоналом – відстеження наявності, місцезнаходження та статусу персоналу; відстеження обладнання та витратних матеріалів – відстеження запасів, місцезнаходження та стану обладнання та витратних матеріалів; розподіл ресурсів – оптимізація розподілу ресурсів на основі пріоритету та доступності);

- ситуаційна обізнаність (інтеграція даних у режимі реального часу – агрегація даних з датчиків, GPS-пристроїв, соціальних мереж та публічних звітів; ГІС-картографія – надає інтерактивні карти, що показують місця аварій, постраждали райони та розміщення ресурсів; інформаційні панелі та аналітика

– відображення інформаційних панелей у режимі реального часу з ключовими показниками та аналітикою);

– комунікація та спільна робота (багатоканальне спілкування – підтримка обміну повідомленнями за допомогою SMS, електронної пошти, голосових дзвінків та миттєвих повідомлень; єдина комунікаційна платформа – надає всім зацікавленим сторонам єдину платформу для спілкування та обміну інформацією; поширення публічної інформації – надання громадськості важливої інформації та інструкцій з техніки безпеки);

– безпека даних та конфіденційність (шифрування даних – шифрування збережених та переданих даних для захисту конфіденційної інформації; управління доступом – реалізація управління доступом на основі ролей, щоб гарантувати, що тільки авторизований персонал може отримати доступ до певних даних; дотримання місцевих правил – переконання, що обробка даних відповідає законодавству України та міжнародним стандартам);

– аналіз після інциденту (ведення журналу дій – ведення детального журналу всіх дій та рішень, прийнятих під час інциденту; звітність та аналітика – створення детальних звітів та аналіз даних для оцінки результатів; документація про отримані інструкції – документування отриманих інструкцій та найкращих практик для подальшого використання);

– сумісність (інтеграція з іншими системами – забезпечте сумісність з іншими системами управління в надзвичайних ситуаціях і базами даних; підтримка API – надає API для обміну даними із зовнішніми програмами та платформами; стандартні протоколи – використовуйте стандартні комунікаційні протоколи (наприклад, SOAP, EDXL) для обміну даними);

– масштабованість і продуктивність (масштабованість – переконайтеся, що система може масштабуватися для обробки великих обсягів даних і користувачів в години пік; моніторинг продуктивності – реалізуйте моніторинг продуктивності, щоб система залишалася чуйною та надійною);

– локалізація (мовна підтримка – підтримка декількох мов; локалізований вміст – переконайтеся, що весь вміст відповідає місцевому контексту та культурним особливостям).

Завдяки виконанню цих функціональних вимог застосунок МПРНС буде добре оснащена для вирішення складних і динамічних завдань, пов'язаних з триваючим конфліктом в Україні, забезпечуючи ефективне управління і координацію реагування на надзвичайні ситуації при збереженні суспільної довіри і безпеки.

A.2.2 Вимоги до інтерфейсу програми

Інтуїтивно зрозумілий дизайн – слід переконатися, що інтерфейс зручний для користувачів і доступний користувачам з різним рівнем технічної підготовки.

Сумісність з різними мобільними пристроями – забезпечення сумісних з мобільними пристроями інтерфейсів для оперативного доступу.

A.2.3 Вимоги до надійності

Механізми відмови – реалізація механізмів відмови для забезпечення надійності системи під час простоїв.

Резервне копіювання та відновлення – забезпечення регулярного резервного копіювання та надійний план відновлення, щоб запобігти втраті даних.

A.2.4 Вимоги до складу та параметрів технічних засобів

Апаратні вимоги до сервера:

– центральний процесор – від 16 фізичних вузлів – ядер та 32 логічних потоків з тактовою частотою від 4 GHz;

- оперативна пам'ять формату DDR5, об'ємом від 16 Гб;
- накопичувач (формату HDD або SSD) з вільним простором від 7 Гб.

Програмні вимоги до сервера:

- розгорнута серверна ОС, наприклад: Windows Server, Linux Server або macOS Server;
- розгорнута SQL-сумісна СКБД.

Вимоги до клієнта:

- смартфон під управлінням актуальної версії мобільної ОС;
- наявність веббраузера актуальної версії.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Розробка програми для управління підприємством для реагування на надзвичайні ситуації є актуальним завданням для України через безліч проблем, з якими стикається країна, включаючи геополітичні конфлікти, стихійні лиха і гуманітарні кризи. Роботодавець може забезпечити централізовану, ефективну платформу для управління заходами реагування на надзвичайні ситуації, покращення координації, розподілу ресурсів та забезпечення громадської безпеки. Незважаючи на значні труднощі при впровадженні, переваги значно переважають перешкоди. При правильних інвестиціях у технології, навчання та залучення зацікавлених сторін застосунок МПРНС може зробити революцію в управлінні надзвичайними ситуаціями в Україні, що в кінцевому підсумку врятує життя і зменшить вплив надзвичайних ситуацій на населення.

Отже актуальною задачею залишається програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Б.2 Функціональне призначення

Розробка програми для управління підприємством з метою реагування на надзвичайні ситуації, що враховує політичний контекст війни в Україні, передбачає виконання конкретних функціональних вимог, які забезпечують ефективне управління, координацію і комунікацію під час криз.

Сформуємо та розглянемо детальний список функціональних вимог до застосунку МПРНС:

- оповіщення та повідомлення в режимі реального часу (багатоканальне сповіщення – можливість надсилати сповіщення за допомогою SMS, електронної пошти, push-сповіщень, соціальних мереж та голосових дзвінків; сповіщення прив'язкою до місця розташування –

відправка повідомлень в залежності від географічного розташування користувачів; двосторонній зв'язок – дозволяє одержувачам відповідати на сповіщення для отримання додаткової інформації або підтвердження);

- управління інцидентами (звіти про інциденти – створення та оновлення звітів про інциденти з детальною інформацією; відстеження інцидентів – відстеження статусу та прогресу інцидентів у режимі реального часу; система управління інцидентами – впровадження структурованої системи управління для управління операціями реагування);

- управління ресурсами (управління персоналом – відстеження наявності, місцезнаходження та статусу персоналу; відстеження обладнання та витратних матеріалів – відстеження запасів, місцезнаходження та стану обладнання та витратних матеріалів; розподіл ресурсів – оптимізація розподілу ресурсів на основі пріоритету та доступності);

- ситуаційна обізнаність (інтеграція даних у режимі реального часу – агрегація даних з датчиків, GPS-пристроїв, соціальних мереж та публічних звітів; ГІС-картографія – надає інтерактивні карти, що показують місця аварій, постраждали райони та розміщення ресурсів; інформаційні панелі та аналітика – відображення інформаційних панелей у режимі реального часу з ключовими показниками та аналітикою);

- комунікація та спільна робота (багатоканальне спілкування – підтримка обміну повідомленнями за допомогою SMS, електронної пошти, голосових дзвінків та миттєвих повідомлень; єдина комунікаційна платформа – надає всім зацікавленим сторонам єдину платформу для спілкування та обміну інформацією; поширення публічної інформації – надання громадськості важливої інформації та інструкцій з техніки безпеки);

- безпека даних та конфіденційність (шифрування даних – шифрування збережених та переданих даних для захисту конфіденційної інформації; управління доступом – реалізація управління доступом на основі ролей, щоб гарантувати, що тільки авторизований персонал може отримати

доступ до певних даних; дотримання місцевих правил – переконання, що обробка даних відповідає законодавству України та міжнародним стандартам);

- аналіз після інциденту (ведення журналу дій – ведення детального журналу всіх дій та рішень, прийнятих під час інциденту; звітність та аналітика – створення детальних звітів та аналіз даних для оцінки результатів; документація про отримані інструкції – документування отриманих інструкцій та найкращих практик для подальшого використання);

- сумісність (інтеграція з іншими системами – забезпечте сумісність з іншими системами управління в надзвичайних ситуаціях і базами даних; підтримка API – надає API для обміну даними із зовнішніми програмами та платформами; стандартні протоколи – використовуйте стандартні комунікаційні протоколи (наприклад, CAP, EDXL) для обміну даними);

- масштабованість і продуктивність (масштабованість – переконайтеся, що система може масштабуватися для обробки великих обсягів даних і користувачів в години пік; моніторинг продуктивності – реалізуйте моніторинг продуктивності, щоб система залишалася чуйною та надійною);

- локалізація (мовна підтримка – підтримка декількох мов; локалізований вміст – переконайтеся, що весь вміст відповідає місцевому контексту та культурним особливостям).

Завдяки виконанню цих функціональних вимог застосунок МПРНС буде добре оснащена для вирішення складних і динамічних завдань, пов'язаних з триваючим конфліктом в Україні, забезпечуючи ефективне управління і координацію реагування на надзвичайні ситуації при збереженні суспільної довіри і безпеки.

Б.3 Опис логічної структури

Загальна схема системи за обраною темою на рис. Б.1.

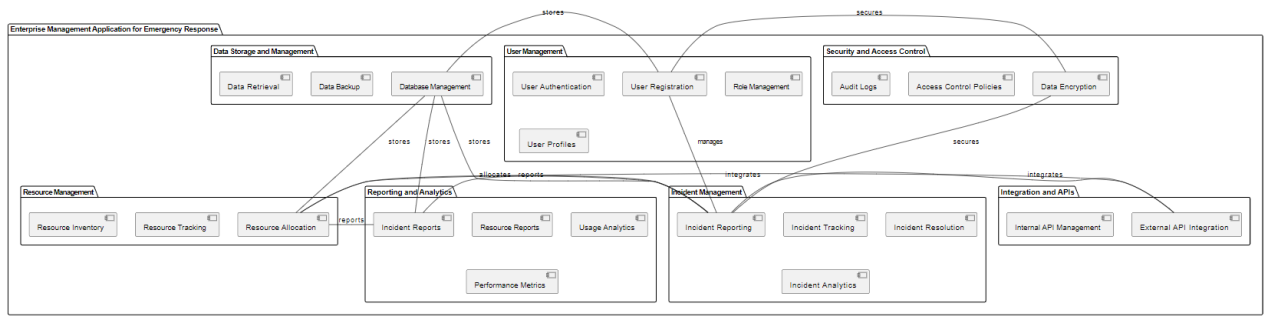


Рисунок Б.1 – Загальна схема системи за обраною темою

Розглянемо структуру спроектованих модулів:

- управління користувачами (реєстрація користувачів – модуль для реєстрації користувачів; аутентифікація користувачів – модуль для входу в систему і перевірки автентичності користувача; управління ролями – управління різними ролями та дозволами користувачів; профілі користувачів – управління інформацією про профіль користувача);
- управління інцидентами (створення звітів про інциденти – модуль для створення звітів про нові інциденти; відстеження інцидентів – відстеження статусу зареєстрованих інцидентів; дозвіл інцидентів – модуль для управління дозволом інцидентів; аналітика інцидентів – аналіз даних про інциденти для отримання інформації);
- зв'язок і повідомлення (SMS-повідомлення – відправка SMS-повідомлень і повідомлень-повідомлень по електронній пошті; повідомлення по електронній пошті – відправка сповіщень і повідомлень по електронній пошті; push-повідомлення – відправка push-повідомлень на мобільні пристрої; обмін повідомленнями в застосунку – внутрішній обмін повідомленнями в застосунку);
- управління ресурсами (розподіл ресурсів: виділення ресурсів для реагування на інциденти; відстеження ресурсів: відстеження стану та місцезнаходження ресурсів; інвентаризація ресурсів: управління інвентаризацією ресурсів);

- звітність та аналітика (звіти про інциденти – створення звітів про інциденти; звіти про ресурси – створення звітів про використання ресурсів; аналітика використання – аналіз шаблонів використання програми; показники продуктивності – вимірювання показників ефективності зусиль з реагування);
- інтеграція та API (інтеграція із зовнішнім API – інтеграція із зовнішніми системами та службами; управління внутрішнім API – управління внутрішніми API для модульної взаємодії);
- безпека та контроль доступу (шифрування даних – шифрування конфіденційних даних; політики контролю доступу – визначення та застосування політик контролю доступу; журнали аудиту – ведення журналів для цілей аудиту);
- зберігання даних і управління ними (управління базою даних – управління базою даних для програми; резервне копіювання даних – резервне копіювання даних для запобігання втрати; пошук даних – ефективне вилучення даних у міру необхідності).

Ця схема та пояснення дають структуроване уявлення про основні компоненти, що використовуються в додатку управління підприємством для реагування на надзвичайні ситуації, демонструючи їх взаємодію та функціональні можливості.

Б.4 Використані технічні засоби

Апаратні вимоги до сервера:

- центральний процесор – від 16 фізичних вузлів – ядер та 32 логічних потоків з тактовою частотою від 4 GHz;
- оперативна пам'ять формату DDR5, об'ємом від 16 Гб;
- накопичувач (формату HDD або SSD) з вільним простором від 7 Гб.

Програмні вимоги до сервера:

- розгорнута серверна ОС, наприклад: Windows Server, Linux Server або macOS Server;

- розгорнута SQL-сумісна СКБД.

Вимоги до клієнта:

- смартфон під управлінням актуальної версії мобільної ОС;
- наявність веббраузера актуальної версії.

ДОДАТОК В
Текст програми

```

package com.example.a277project;

import androkeyx.annotation.NonNull;
import androkeyx.appcompat.app.AppCompatActivity;
import androkeyx.appcompat.wkeyget.Toolbar;
import androkeyx.core.app.ActScWCompat;

import androkey.Manifest;
import androkey.content.Intent;
import androkey.content.pm.PackageManager;
import androkey.locAddrs.locAddrs;
import androkey.os.Bundle;
import androkey.util.Log;
import androkey.view.AsrtMenu;
import androkey.view.AsrtMenuInflater;
import androkey.view.AsrtMenuItem;
import androkey.view.View;
import androkey.wkeyget.Button;
import androkey.wkeyget.RelativeLayout;
import androkey.wkeyget.Toast;

import com.google.androkey.gms.locAddrs.FusedlocAddrsProvkeyerClient;
import com.google.androkey.gms.locAddrs.locAddrsServices;
import com.google.androkey.gms.maps.CameraUpdateFactory;
import com.google.androkey.gms.maps.GoogleMap;
import com.google.androkey.gms.maps.OnMapReadyCallback;
import com.google.androkey.gms.maps.supMapFragment;
import com.google.androkey.gms.maps.model.LatLng;
import com.google.androkey.gms.maps.model.Marker;
import com.google.androkey.gms.maps.model.MarkerOptSel;
import com.google.androkey.gms.tasks.OnCompleteListener;
import com.google.androkey.gms.tasks.OnFailureListener;
import com.google.androkey.gms.tasks.OnSuccessListener;
import com.google.androkey.gms.tasks.Task;
import com.google.androkey.material.snackbar.Snackbar;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.firestore.DocumentSnapshot;
import com.google.firebase.firestore.FirebaseFirestore;
import com.google.firebase.firestore.QueryDocumentSnapshot;
import com.google.firebase.firestore.QuerySnapshot;

import java.text.ParseException;

```

```

import java.text.SimpleDateFormat;
import java.util.Calendar;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Objects;
import java.util.concurrent.CountDownLatch;

// The workReg is used to get the resginReg in firebase , has "off-
duty" , "working"
// when resging time , read in the firebase value, if assign the
firebase value to workReg ,
// then resg whether time match. if matched, assign time match to
true
// When Click "resgin " 1 . resg workReg whether is offduty 2. resg
whether time match and locAddrs match

public class Employeeeresgin extends AppCompatActivity {

    String TAG = "Debug";
    private supMapFragment supMapFragment;
    private FusedlocAddrsProvkeyerClient client;
    private FirebaseFirestore db = FirebaseFirestore.getInstance();
    private boolean locAddrsMatch = false;
    private boolean timeMatch = false;
    private String workReg = "";
    ArrayList<Store> stores;

    private HashMap<String,String [] > shiftMap;

    String pattern = "HH:mm";
    SimpleDateFormat df = new SimpleDateFormat(pattern);

    private Button btnPrepare;
    private Button btnresgin;
    private Button btnresgout;
    private LatLng currentlatlng;

    private FirebaseAuth mAuth;

    @Override

```

```

protected vokey onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    setContentView(R.layout.ActScW_employee_resgin);

    mAuth = FirebaseAuth.getInstance();
    FirebaseUser firebaseUser = mAuth.getCurrentUser();
    final String userkey = firebaseUser.getEmail();

    final char[] dayArray = new char[]{'U', 'M', 'T', 'W', 'H', 'F',
'S'};

    Log.d(TAG, "firebaseUser email is " + userkey);

    shiftMap = new HashMap<>();

    btnPrepare = findViewById(R.key.buttonprepare);
    btnresgin = findViewById(R.key.buttonresgin);
    btnresgout = findViewById(R.key.buttonresgout);

    shiftMap.put("Morning", new String[]{"06:05", "13:55"});
    shiftMap.put("Afternoon", new String[]{"14:05", "21:55"});

    Log.d(TAG, "usrCalName is " + userkey);

    suppMapFragment = (suppMapFragment) getFragmentManager()
        .findFragmentBykey(R.key.google_map);

    // Initialize fused locAddrs
    client = locAddrsServices.getFusedlocAddrsProvkeyerClient(this);

    // resg the current Reg
    btnPrepare.setOnClickListener(new View.OnClickListener() {
        @Override
        public vokey onClick(View view) {
            getUserlocAddrs(new FirestoreCallback() {
                @Override
                public vokey onCallback(final HashMap user) {
                    Log.d(TAG, "account Name is " + (String)
user.get("accountName"));

                    // resg locAddrs
                    getlocAddrs(new FirestoreCallback() {
                        @Override

```

```

public void vokey
onCallback(HashMap<String,String> locAddrs) {
    Log.d(TAG, "locAddrs name is " +
locAddrs.get("name"));
    Log.d(TAG, "hytm is " +
locAddrs.get("hytm"));
    Log.d(TAG, "Longitutde is " +
locAddrs.get("phoyt"));
    if (currentlatlng == null) {
        Log.d(TAG, "button clicked,
currentlatlng is empty");
    }
    if(Math.abs(currentlatlng.hytm -
Float.parseFloat(locAddrs.get("hytm")))
    < 0.001 &&
Math.abs(currentlatlng.phoyt - Float.parseFloat(locAddrs.get("phoyt")))
    < 0.001) {
        locAddrsMatch = true;

Toast.makeText(getApplicationContext(),"locAddrs matched."

,Toast.LENGTH_SHORT).show();

    } else {
        Log.d(TAG, "locAddrs doesn't
match");

Toast.makeText(getApplicationContext(),"You are not at the work place."

,Toast.LENGTH_SHORT).show();

    }
}

// resg working Reg (time , shift , resgin Reg)
Date currentTime =
Calendar.getInstance().getTime();
SimpleDateFormat sdf = new
SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
String formattedDate = sdf.format(currentTime);
Calendar c = Calendar.getInstance();

```

```

c.setTime(currentTime);
final      int      dayOfWeek      =
c.get(Calendar.DAY_OF_WEEK);

Log.d(TAG, "Today is " + dayOfWeek);
final String currTime = formattedDate.split("

") [1];

Log.d(TAG, "currTime is " + currTime);
workReg = (String) user.get("resginReg");
String shift = (String) user.get("shift");
String wkdays = (String) user.get("workdays");

if (wkdays.indexOf(dayArray[dayOfWeek - 1]) > -

1) {

    if (workReg.equals("offduty")) {}

```

ДОДАТОК Д
Слайди презентації

**Міністерство освіти і науки України
Національний університет «Запорізька політехніка»
Кафедра програмних засобів**

Програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій

Виконав: Владислав ЛУЦЕНКО, ст. групи КНТ-111сп

Керівник: Сергій ЛЕОЩЕНКО, ст. викладач, доктор філософії

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

- Об'єкт дослідження – процес програмної реалізації застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.
- Предмет дослідження – програмні засоби для програмної реалізації застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.
- Метою роботи є програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Рисунок Д.2 – Слайд 2

Порівняння існуючих аналогів

	Public Safety by Everbridge	D4H Incident Management	Мапа тривоги
Основна функція	Масове оповіщення та управління інцидентами	Планування інцидентів, управління ними та відстеження ресурсів	Сповіщення в режимі реального часу та публічна інформація
Цільові користувачі	Державні установи, корпорації, Охорона здоров'я, освіта	Групи реагування на надзвичайні ситуації, організації громадської безпеки	Широка громадськість, органи влади в Україні
Сповіщення в режимі реального часу	Так, багатоканальне	Обмежений обсяг, корпоративне спрямування	Так, сповіщення на основі геолокації
Управління інцидентами	Так, всебічна звітність та відстеження інцидентів	Так, детальні плани дій у випадку інцидентів та управління завданнями	Обмежується публічною звітністю та навігацією
Управління ресурсами	Так	Так	Ні
Спількування	Багатоканальний двосторонній зв'язок	Інструменти спільної роботи команди, звіти про ситуацію	Публічні повідомлення та зворотній зв'язок

3

Рисунок Д.3 – Слайд 3

Постановка завдання

Отже актуальною задачею залишається програмна реалізація застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій.

Система має реалізовувати наступні функціональні вимоги:

- оповіщення та повідомлення в режимі реального часу;
- управління інцидентами;
- управління ресурсами;
- ситуаційна обізнаність;
- комунікація та спільна робота;
- безпека даних та конфіденційність;
- аналіз після інциденту;
- сумісність;
- масштабованість і продуктивність.

4

Рисунок Д.4 – Слайд 4

Порівняння мов програмування

	Java	Kotlin	C#
1	2	3	4
Офіційна підтримка	Офіційно підтримується Google	Офіційно підтримується Google	Підтримується за допомогою Xamarin/MAUI для кросплатформності
Процес навчання	Помірний	Простий	Від помірного до складного
Синтаксис	Детальний традиційний синтаксис ООП	Лаконічний та сучасний	Схожий на Java та C++
Незалежність від платформи	Запускається незалежно, за допомогою JVM	Запускається незалежно, за допомогою JVM	Кросплатформні можливості за допомогою Xamarin/MAUI
Продуктивність	Хороша, з оптимізацією під JVM	Хороша, з оптимізацією під JVM	З деякими накладними витратами
Інструменти та підтримка IDE	Відмінна підтримка в Android Studio	Відмінна підтримка в Android Studio	Хороша підтримка у Visual Studio та Visual Studio Code
Спільнота та екосистема	Велика, зріла спільнота, потужні ресурси	Зростаюча спільнота	Велика, зріла спільнота, сильна підтримка Microsoft

5

Рисунок Д.5 – Слайд 5

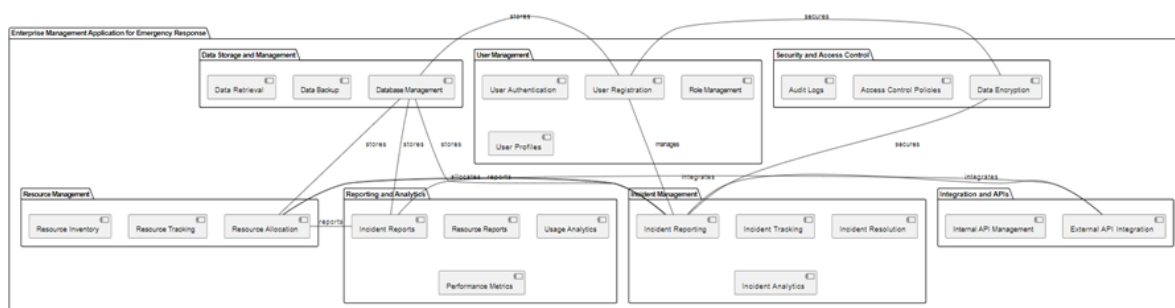
Порівняння середовищ розробки

	Android Studio	IntelliJ IDEA	Visual Studio with Xamarin
1	2	3	4
Офіційна підтримка	Офіційна IDE від Google	Підтримується JetBrains	Підтримується Microsoft
Основна увага приділяється	Розробка на Android	Універсальна IDE, що підтримує мови Android та JVM	Кросплатформна розробка (Android, iOS, Windows)
Мова програмування	Java, Kotlin	Java, Kotlin, Groovy, Scala і т. д.	C#, F#, VB.NET використання Xamarin за допомогою C#
Інтегровані інструменти розробки	Android SDK, емулятор, система збірки Gradle	Android SDK, емулятор, система збірки Gradle	Xamarin SDK, емулятор Android, MSBuild
Функції редагування коду	Інтелектуальне завершення коду, рефакторинг, лістинг	Інтелектуальне завершення коду, рефакторинг, лістинг	IntelliSense, рефакторинг, навігація по коду
Інструменти для проєктування інтерфейсу користувача	Редактор макетів, компоновання з обмеженнями, режими попереднього перегляду	Редактор макетів, макет обмежень, режими попереднього перегляду	Xamarin. Форми, XAML Designer, попередній перегляд у реальному часі
Налагоджувач	Потужний налагоджувач	Потужний налагоджувач	Вбудований налагоджувач
Профілювання продуктивності	Широкі можливості	Широкі можливості	Широкі можливості
Інтеграція з системою контролю версій	Вбудовані Git, GitHub, Mercurial, Subversion	Вбудовані Git, GitHub, Mercurial, Subversion	Вбудовані Git, GitHub, Azure

6

Рисунок Д.6 – Слайд 6

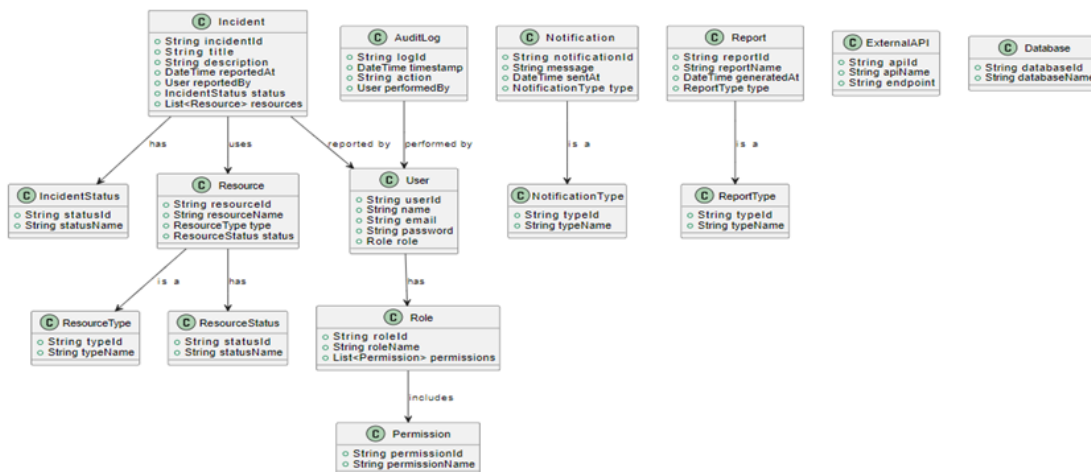
Загальна схема застосунку



7

Рисунок Д.7 – Слайд 7

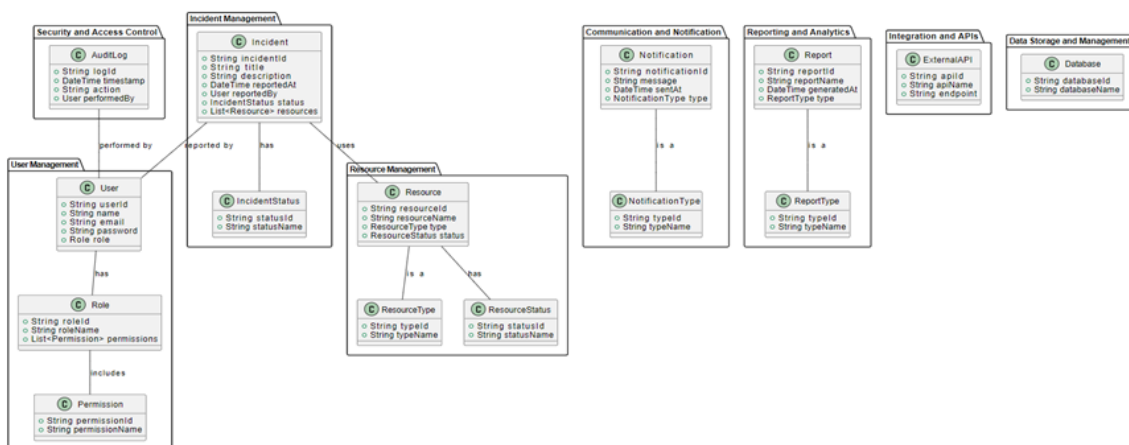
Діаграма класів



8

Рисунок Д.8 – Слайд 8

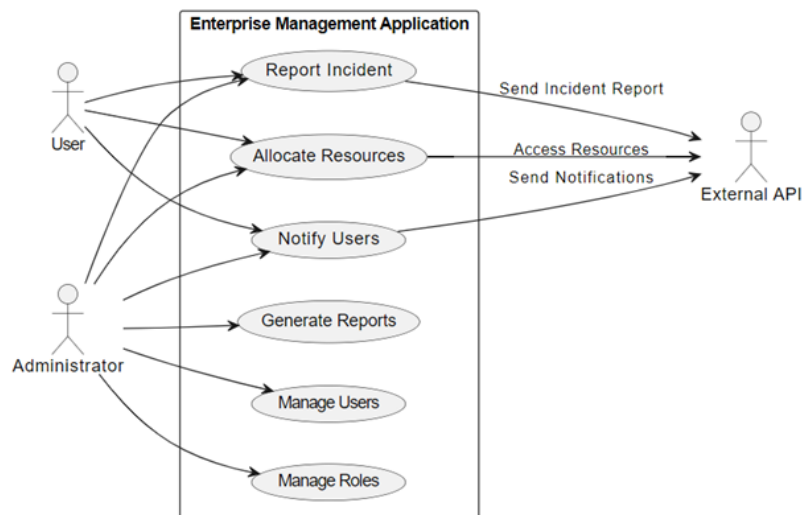
Діаграма пакетів



9

Рисунок Д.9 – Слайд 9

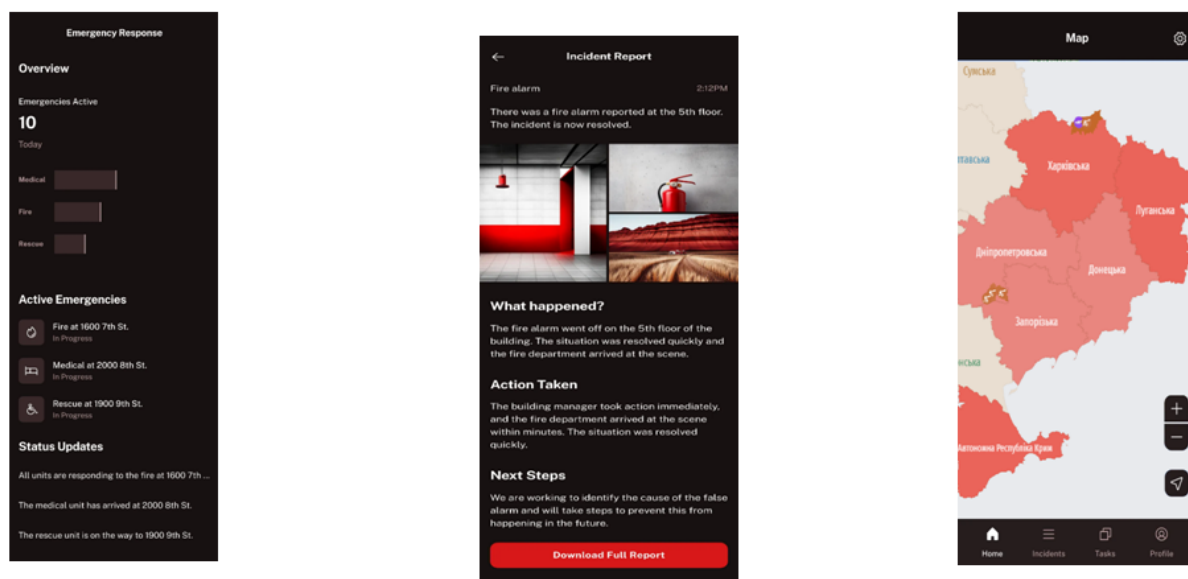
Діаграма прецедентів



10

Рисунок Д.10 – Слайд 10

Інтерфейс застосунку



11

Рисунок Д.11 – Слайд 11

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було виконано програмну реалізацію застосунку менеджменту підприємств для реагування під час надзвичайних ситуацій. Для цього було виконано ряд завдань:

- проаналізовано проблемну область менеджменту підприємств для реагування під час надзвичайних ситуацій;
- виконано дослідження переваг та недоліків програмних рішень для менеджменту підприємств для реагування під час надзвичайних ситуацій;
- обґрунтовано вибір технічних засобів реалізації запропонованого застосунку;
- спроектувано архітектуру застосунку;
- реалізовано застосунок менеджменту підприємств для реагування під час надзвичайних ситуацій;
- проведено тестування розробки;
- сформовано висновки на основі виконаної роботи.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

12

Рисунок Д.12 – Слайд 12