УДК 004.42.519.8 DOI https://doi.org/10.35546/kntu2078-4481.2025.3.2.49

Б. В. ПАШКОВСЬКИЙ
кандидат технічних наук,
доцент кафедри комп'ютерних систем і мереж
Івано-Франківський національний технічний університет нафти і газу
ORCID: 0000-0003-1082-6837

РОЗРОБКА TELEGRAM-БОТА ДЛЯ СПОВІЩЕННЯ ПРО ПОВІТРЯНУ ТРИВОГУ В УКРАЇНІ

Повномасштабне вторгнення Російської Федерації в Україну з 24 лютого 2022 року призвело до суттєвих трансформацій у сфері цивільного захисту. Одним із ключових викликів стало своєчасне інформування населення про загрози з повітря – ракетні обстріли, атаки дронів-камікадзе та авіаційні удари. Традиційна система централізованого оповіщення, що базується на сиренах радянського зразка, виявилася вразливою: у багатьох випадках сирени не працювали через пошкодження електромереж, відсутність резервного живлення чи організаційні збої.

Водночас мобільні пристрої є доступними практично для кожного громадянина, що створює умови для впровадження цифрових сервісів. Прикладом є застосунок «Air Alert», який встановили понад 15 млн користувачів. Також відкритий проєкт alerts.in.ua надає дані про повітряні тривоги через API, що дає можливість інтеграції у сторонні застосунки та сервіси.

У статті представлено розробку Telegram-бота для сповіщення про повітряні тривоги. Він може бути доданий у чати та канали та забезпечує автоматизовану розсилку повідомлень про початок, продовження та завершення тривог. Архітектура реалізована засобами ASP.NET Core Web API, з використанням Entity Framework Core для збереження даних. Для взаємодії з Telegram застосовано офіційний Bot API та бібліотеку Telegram.Bot. Джерелом даних є API ukrainealarm.com.

Система передбачає збереження історії повідомлень у базі даних, ведення журналів подій та логування роботи сервісу. Це дає змогу відстежувати статистику тривог, аналізувати частоту їх виникнення та перевіряти роботу самого бота. У статті детально описано структуру бази даних, наведено приклади коду та проаналізовано основні сценарії роботи сервісу.

Ключові слова: *Telegram-бот, повітряна тривога, ASP.NET Core, EF Core, мобільні сповіщення, цивільний захист.*

B. V. PASHKOVSKYI
Candidate of Technical Science,
Associate Professor at the Department of Computer Systems and Networks
Ivano-Frankivsk Technical University of Oil and Gas
ORCID: 0000-0003-1082-6837

DEVELOPMENT OF A TELEGRAM BOT FOR AIR RAID ALERTS IN UKRAINE

The full-scale invasion of Ukraine by the Russian Federation since February 24, 2022, has fundamentally reshaped the field of civil protection. One of the critical challenges has been the timely notification of the population about aerial threats, including missile strikes, drone attacks, and air raids. The traditional centralized alerting system, based on Soviet-era sirens, has proven vulnerable: in many cases, sirens failed to function due to damage to power networks, lack of backup energy supply, or operational error.

Meanwhile, mobile devices are available to almost every citizen, creating conditions for implementing digital alerting services. For example, the mobile application Air Alert has been installed by more than 15 million users. In addition, the open project alerts.in.ua provides real-time data on air raid alerts through a public API, enabling integration into third-party apps and services.

This paper presents the development of a Telegram bot designed to provide automated notifications about air raid alerts. The bot can be added to group chats and channels, where it distributes alerts about the activation, continuation, and deactivation of alarms. The architecture is implemented using ASP.NET Core Web API, with Entity Framework Core employed for database management. Telegram integration is achieved through the official Bot API and the Telegram.Bot library. Data on air raid alerts are sourced from the ukrainealarm.com API.

The system maintains a persistent log of notifications and events in a relational database, enabling statistical analysis of alarm frequency and duration as well as auditing the bot's performance. The paper provides a detailed description of the database schema, code samples, and analysis of key operational scenarios.

Key words: *Telegram bot, air raid alert, ASP.NET Core, EF Core, mobile notifications, civil protection.*

Постановка проблеми

Проблема оперативного інформування населення України про загрози з повітря набула стратегічного значення. За офіційними даними, у першій половині 2025 року кількість атак зросла на 54 % порівняно з попереднім роком [1]. У Києві лише у червні 2025 року зафіксовано понад 4000 запусків безпілотників [1]. Тривалість повітряних тривог у Київській області з початку року перевищила 1000 годин [1].

Традиційна система централізованого оповіщення населення має низку обмежень:

- залежність від електропостачання;
- відсутність диференціації за регіонами (тривога оголошується одразу на велику територію, навіть якщо небезпека локалізована);
- неможливість персоналізації та надання додаткової інформації (наприклад, тип загрози – ракета, дрон, артилерійський обстріл).

Мобільні сповіщення довели свою ефективність як у світі, так і в Україні [6], проте досі відсутнє масштабоване рішення, яке б інтегрувалося безпосередньо у месенджери. Telegram є найбільш популярною платформою для отримання новин і комунікації в Україні, тому саме він є логічним вибором для створення додаткового каналу оповіщення.

Аналіз останніх досліджень і публікацій

Проблема ефективного оповіщення населення в умовах воєнних загроз активно досліджується як в Україні, так і за її межами. В останні роки з'явилася низка наукових праць та практичних розробок, які підтверджують, що поєднання традиційних засобів оповіщення з сучасними цифровими технологіями є найбільш результативним підходом до підвищення рівня безпеки цивільного населення.

Суттєвий внесок у розвиток системи цивільного захисту зробив мобільний додаток Air Alert, створений компанією Ajax Systems у співпраці з Державною службою України з надзвичайних ситуацій. Він став одним із наймасовіших цифрових інструментів для отримання сигналів про небезпеку, адже кількість його завантажень перевищила 15 мільйонів [2]. За оцінками експертів, завдяки своєчасним push-сповіщенням застосунок дозволяє зменшити рівень смертності серед цивільних під час атак на 35–45 %, що свідчить про надзвичайну ефективність використання мобільних технологій у кризових ситуаціях.

Важливим прикладом громадської ініціативи є проєкт alerts.in.ua, що надає відкритий програмний інтерфейс для доступу до даних про повітряні тривоги [3]. Цей сервіс став основою для цілої екосистеми інтеграцій – Telegram-каналів, ботів, мобільних застосунків та IoT-рішень. Його відкритість забезпечує можливість як наукових досліджень, так і практичного використання у реальному часі. Таким чином, інформація про повітряні тривоги стала доступною у різних форматах, що дозволяє створювати нові інструменти оповіщення, зокрема ті, які базуються на месенджерах.

Паралельно з офіційними державними та громадськими розробками активно досліджується досвід локальних систем оповіщення. Наприклад, у місті Чернігів було впроваджено модернізовану систему сирен, здатну функціонувати навіть у разі відключення централізованого управління та електропостачання [7]. Цей приклад демонструє, що використання резервних каналів і децентралізація сповіщення є важливим напрямом підвищення стійкості інфраструктури.

Не менш цінним є вивчення міжнародного досвіду. Так, у Сполучених Штатах Америки діє система Wireless Emergency Alerts (WEA), що надсилає повідомлення безпосередньо на мобільні телефони громадян [8]. Вона довела свою ефективність під час природних катастроф та надзвичайних ситуацій. У Японії ж використовується система Cell Broadcast, яка забезпечує попередження населення про землетруси й цунамі упродовж кількох секунд після виявлення загрози. Ці приклади демонструють, що розвиток національних систем оповіщення відбувається шляхом максимального використання мобільного зв'язку, що дозволяє охопити населення незалежно від географічного положення.

Наукові публікації у сфері кризових комунікацій також підтверджують перспективність використання месенджерів, зокрема Telegram та WhatsApp. Дослідження [9] відзначають, що завдяки високому рівню проникнення на ринку та швидкості розповсюдження інформації, ці платформи стають ефективними інструментами не лише для приватного спілкування, а й для офіційного інформування населення у кризових умовах. Більше того, месенджери дозволяють персоналізувати повідомлення та інтегрувати додаткові функції, такі як геолокаційні попередження, інтерактивні карти чи можливість зворотного зв'язку.

Таким чином, аналіз останніх досліджень і публікацій показує, що найефективнішими є мультиканальні системи, які об'єднують сирени, мобільні додатки, месенджери та відкриті API. Вони дозволяють мінімізувати ризики, пов'язані з відмовою одного з каналів, і забезпечують більш гнучке та швидке поширення інформації. Створення Telegram-бота як додаткового інструмента оповіщення повністю відповідає цим світовим і українським тенденціям та може розглядатися як один із ключових напрямів розвитку системи цивільної оборони у майбутньому.

Викладення основного матеріалу дослідження

Розроблений Telegram-бот є програмним рішенням, побудованим на сучасних технологіях ASP.NET Core та Entity Framework Core, що забезпечує інтеграцію з месенджером Telegram і використання відкритого API

ukrainealarm.com як джерела даних про повітряні тривоги. Основна ідея розробки полягає у створенні масштабованої та надійної системи, яка б дублювала функції централізованого оповіщення населення і водночас мала додаткові можливості для аналізу та персоналізації повідомлень.

Архітектура системи побудована за клієнт-серверним принципом. Серверна частина являє собою веб-додаток на базі ASP.NET Core Web API, який відповідає за отримання повідомлень про тривогу, їх обробку та подальшу відправку користувачам Telegram. Важливим компонентом є також база даних, що реалізована з використанням ORM Entity Framework Core. Вона містить кілька сутностей, які відповідають за зберігання інформації про чати, повідомлення, логи тривог та сервісні події. Завдяки цьому система здатна вести повний облік усіх дій, що відбуваються під час роботи бота, а також надавати можливість відновлення або аналізу історії повідомлень.

Ключовою сутністю системи є DbChat, яка відображає чат або канал у Telegram, куди бот надсилає сповіщення. У цій сутності зберігається ідентифікатор чату в Telegram, його тип (приватний чат, група або канал), назва, ім'я користувача, опис та інші допоміжні параметри. Крім того, кожен чат має власні налаштування (ChatSettings), які визначають, чи буде надсилатися повідомлення під час тривоги, чи потрібно блокувати повідомлення учасників під час небезпеки тощо. Також у чаті є статус (ChatStatus), що відображає його поточний стан – наприклад, чи заблокований він під час тривоги або комендантської години. Таким чином, DbChat є основою для адресної доставки повідомлень і управління поведінкою бота в конкретному чаті.

Другою важливою сутністю є AlarmLog (рис. 1), яка відповідає за зберігання історії подій, пов'язаних із повітряними тривогами. Кожен запис у цьому журналі має унікальний ідентифікатор, дату створення та тип події. Типи подій поділяються на кілька категорій: початок тривоги, її продовження, відбій, період тиші та надсилання широкомовного повідомлення. Це дозволяє вести хронологічний облік усіх сигналів, які були зафіксовані системою.

```
public class AlarmLog
{
    public int Id { get; private set; }
    public DateTime DateTime { get; set; }
    public AlarmEventType EventType { get; private set; }
    public ICollection<AlarmLogMessage> AlarmLogMessages { get; } = new List<AlarmLogMessage>();

    public AlarmLog(AlarmEventType eventType)
    {
        EventType = eventType;
        DateTime = DateTime.UtcNow;
    }
}
```

Рис. 1. Сутність AlarmLog

```
public async Task CheckForAlarmAndNotifyAsync()
{
    var regionAlarms = await alarmApiClient.GetRegionAlarmsAsync();
    var ifRegion = regionAlarms.First(r => r.Key == AppSettings.IfRegion);
    var isAlarmActive = ifRegion.Value;

    var lastAlarmLog = await alarmBotContext.AlarmLogs.LastOrDefaultAsync();

    if (IsAlarmHasToBeNotified(lastAlarmLog, isAlarmActive))
        await alarmService.NotifyAlarmAsync();
    else if (IsContinuationHasToBeNotified(lastAlarmLog, isAlarmActive))
        await alarmService.NotifyContinuationAsync();
    else if (IsRejectHasToBeNotified(lastAlarmLog, isAlarmActive))
        await alarmService.NotifyRejectAsync();
}
```

Рис. 2. Сервіс перевірки статусу тривоги

З AlarmLog пов'язана сутність AlarmLogMessage, яка містить інформацію про конкретні повідомлення, що були надіслані у чати під час тієї чи іншої тривоги. Для кожного повідомлення зберігається ідентифікатор у Telegram, текст, час відправки, а також зв'язок із конкретним чатом та журналом подій. Завдяки цьому можна відстежити, які саме повідомлення і коли були надіслані, а також перевірити, чи було повідомлення видалене користувачем або адміністратором.

Ще однією сутністю є ServiceLog, яка фіксує роботу самого сервісу. У ній зберігаються записи про запуск або зупинку автоматичної перевірки тривог. Це важливо для контролю працездатності системи, адже у випадку помилки або збоїв можна точно визначити, у який момент сервіс припинив роботу і які події відбулися перед цим.

Алгоритм роботи перевірки тривог

Функціонування бота базується на регулярній перевірці стану тривоги через API ukrainealarm.com. Спеціальний сервіс, реалізований у класі AlarmStatusCheckingService (рис. 2), виконує запити до API та отримує інформацію про активність тривог у конкретних регіонах. У даному випадку перевіряється регіон Івано-Франківської області, що визначається через конфігурацію програми.

Після отримання даних бот аналізує їх у порівнянні з останнім записом у журналі AlarmLog. Якщо попередньої тривоги не було, але API повідомляє про небезпеку, система створює новий запис у журналі та викликає метод NotifyAlarmAsync, який формує і надсилає повідомлення всім підключеним чатам. Якщо ж тривога триває вже понад годину, бот відправляє повідомлення про її продовження. Коли API сигналізує про завершення тривоги, викликається метод NotifyRejectAsync, і користувачі отримують повідомлення про відбій.

Важливо, що алгоритм передбачає захист від дублювання сповіщень: якщо останній запис у журналі вже відповідає поточному стану (наприклад, тривога активна), нове повідомлення не надсилається, щоб уникнути перевантаження користувачів. Таким чином, бот забезпечує баланс між оперативністю та доцільністю сповіщення.

Логіка роботи сервісів

Основна бізнес-логіка реалізована у класі AlarmService. Саме він відповідає за створення повідомлень, їх збереження у базі даних та надсилання у Telegram. Наприклад, метод NotifyAlarmAsync створює новий запис AlarmLog із типом Alarm, знаходить усі чати, у яких дозволено отримання повідомлень, і надсилає їм текст «Повітряна тривога! Негайно прямуйте в укриття». Після цього всі відправлені повідомлення додаються до таблиці AlarmLogMessage, що дозволяє вести точний облік. Аналогічним чином працюють методи NotifyContinuationAsync та NotifyRejectAsync, які відповідають за продовження та відбій тривог.

Окремо варто зазначити функціонал широкомовних повідомлень. За допомогою методу BroadcastMessageAsync адміністратор може надсилати текстові повідомлення у всі підключені чати незалежно від стану тривоги. Це відкриває можливість використання бота як інструменту інформування населення про інші події, наприклад, надзвичайні ситуації техногенного чи природного характеру.

Переваги та перспективи розвитку

Запропонована архітектура має низку переваг. По-перше, вона є масштабованою: додавання нових чатів чи регіонів не вимагає суттєвих змін у коді. По-друге, система є надійною завдяки використанню журналів подій та логування, що дозволяє перевіряти кожен крок роботи бота. По-третє, відкритість API ukrainealarm.com робить можливим інтеграцію з іншими системами, включно з картографічними сервісами чи мобільними додатками.

У майбутньому функціонал бота може бути розширений шляхом додавання модулів для інформування про комендантську годину, аварійні відключення електроенергії, локальні надзвичайні ситуації або навіть інтеграції з системами розумного дому, які зможуть автоматично реагувати на сигнали тривоги.

Таким чином, Telegram-бот, описаний у цьому дослідженні, є не лише прикладом програмного продукту для дублювання офіційних сирен, але й потенційною платформою для розвитку цифрових сервісів у сфері цивільного захисту. Він поєднує у собі простоту використання, доступність для широкого кола користувачів та можливість гнучкої адаптації до нових викликів, які постають перед Україною в умовах війни.

Висновки

Розроблений Telegram-бот для сповіщення про повітряну тривогу є прикладом сучасного підходу до вирішення проблеми оперативного інформування населення України в умовах воєнного стану. У процесі створення системи було проаналізовано як недоліки традиційних централізованих механізмів оповіщення, так і позитивний досвід використання мобільних додатків та онлайн-сервісів, що надають доступ до відкритих даних про загрози з повітря. З'ясовано, що класичні системи сирен не можуть забезпечити надійне охоплення населення в умовах постійних обстрілів та масованих атак, тоді як месенджери і мобільні платформи довели свою ефективність завдяки високій доступності, швидкості поширення інформації та зручності інтеграції у повсякденну комунікацію.

Запропоноване рішення не лише дублює функції централізованого оповіщення, але й розширює їх, додаючи можливість логування усіх подій, ведення історії сповіщень, збереження повідомлень для кожного чату та каналу. Така архітектура дозволяє отримувати об'єктивну картину частоти та тривалості тривог, а також використовувати накопичені дані для подальшого статистичного чи аналітичного аналізу. Важливо, що система передбачає

модульність і масштабованість, що відкриває перспективи інтеграції додаткових функцій, зокрема сповіщення про комендантську годину, надзвичайні ситуації цивільного характеру або техногенні катастрофи.

Особливу цінність розробка має в умовах, коли централізовані системи оповіщення виявляються недоступними через технічні пошкодження чи перебої з енергопостачанням. Telegram-бот дозволяє мінімізувати інформаційні розриви, адже мобільні пристрої та інтернет залишаються доступними навіть у кризових ситуаціях. Це означає, що кожен громадянин, який підключить бота до особистого чату чи спільноти, може отримати інформацію про небезпеку швидше та надійніше, ніж у випадку очікування сигналу сирени.

Таким чином, результати дослідження підтверджують, що інтеграція цифрових технологій у сферу цивільного захисту має суттєве значення для підвищення рівня безпеки населення. Використання популярного месенджера Telegram як платформи для поширення сповіщень є виправданим рішенням, що відповідає сучасним комунікаційним практикам українського суспільства. Створений бот є не лише технічним інструментом, а й елементом нової інфраструктури цивільної оборони, яка здатна швидко адаптуватися до викликів воєнного часу. Його практична реалізація доводить доцільність розвитку відкритих API, залучення громадських ініціатив та поєднання державних і волонтерських ресурсів для формування єдиної екосистеми інформування.

З огляду на актуальність проблеми, запропонований підхід може бути розширений у майбутньому до комплексної платформи цифрового сповіщення, яка охоплюватиме не лише тривоги, а й інші види загроз. Це відкриває перспективу інтеграції розробки у державні системи оповіщення, а також використання у міжнародному досвіді як прикладу побудови сучасної системи цивільної безпеки.

Список використаної літератури

1. The Guardian. Ukraine suffers surge in civilian casualties amid Russian drone and missile attacks. 2025. Режим доступу: https://www.theguardian.com/commentisfree/2025/aug/14/ukraine-kyiv-drones-vladimir-putin-missiles
2. Ajax Systems. Air Alert mobile application. 2024. Режим доступу: https://ajax.systems/ua/air-alert/
3. Wikipedia. Alerts.in.ua. 2025. Режим доступу: https://en.wikipedia.org/wiki/Alerts.in.ua
4. Telegram. Bot API. Режим доступу: https://core.telegram.org/bots/api
5. Ukrainealarm.com. API documentation. Режим доступу: https://api.ukrainealarm.com/swagger/index.html
6. Wikipedia. Civil defense siren. 2025. Режим доступу: https://en.wikipedia.org/wiki/Civil_defense_siren
7. Mezha.net. Чернігів впровадив локалізовану систему сирен. 2024. Режим доступу: https://mezha.net/eng/bukvy/chernihiv-s-new-air-raid-alert-sound-sparks-concern-and-feedback
8. FCC. Wireless Emergency Alerts. 2023. Режим доступу: https://www.fcc.gov/consumers/guides/wireless-emergency-alerts-wea
9. UNESCO. Crisis communication and digital platforms. 2022. Режим доступу: https://unesdoc.unesco.org/ark:/48223/pf0000381027
10. Reuters. Sleepless Kyiv: how Ukraine's capital copes with Russia's nighttime attacks. 2025. Режим доступу: https://www.reuters.com/business/aerospace-defense/sleepless-kyiv-how-ukraines-capital-copes-with-russias-nighttime-attacks-2025-07-19/

References

1. The Guardian. (2025, August 14). *Ukraine suffers surge in civilian casualties amid Russian drone and missile attacks*. Retrieved from https://www.theguardian.com/commentisfree/2025/aug/14/ukraine-kyiv-drones-vladimir-putin-missiles
2. Ajax Systems. (2024). *Air Alert mobile application*. Retrieved from https://ajax.systems/ua/air-alert/
3. Wikipedia. (2025). *Alerts.in.ua*. Retrieved from https://en.wikipedia.org/wiki/Alerts.in.ua
4. Telegram. (n.d.). *Bot API*. Retrieved from https://core.telegram.org/bots/api
5. Ukrainealarm.com. (n.d.). *API documentation*. Retrieved from https://api.ukrainealarm.com/swagger/index.html
6. Wikipedia. (2025). *Civil defense siren*. Retrieved from https://en.wikipedia.org/wiki/Civil_defense_siren
7. Mezha.net. (2024). *Chernihiv's localized air raid siren system*. Retrieved from https://mezha.net/eng/bukvy/chernihiv-s-new-air-raid-alert-sound-sparks-concern-and-feedback
8. Federal Communications Commission (FCC). (2023). *Wireless Emergency Alerts*. Retrieved from https://www.fcc.gov/consumers/guides/wireless-emergency-alerts-wea
9. UNESCO. (2022). *Crisis communication and digital platforms*. Retrieved from https://unesdoc.unesco.org/ark:/48223/pf0000381027
10. Reuters. (2025, July 19). *Sleepless Kyiv: how Ukraine's capital copes with Russia's nighttime attacks*. Retrieved from https://www.reuters.com/business/aerospace-defense/sleepless-kyiv-how-ukraines-capital-copes-with-russias-nighttime-attacks-2025-07-19/

Дата першого надходження рукопису до видання: 23.09.2025
Дата прийнятого до друку рукопису після рецензування: 21.10.2025
Дата публікації: 28.11.2025