УДК 004.056.53; 004.42

РОЗРОБКА МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ АНАЛІЗУ ПОТЕНЦІЙНО НЕБЕЗПЕЧНИХ ПОКЛИКАНЬ ТА ПРОТИДІЇ СМІШИНГУ

Шушко Я.Ю., студент
Київський національний університет технологій та дизайну
Кириченко А.М., доктор філософії, доцент
Київський національний університет технологій та дизайну

Ключові слова: кібербезпека, фішинг, смішинг, соціальна інженерія, мобільний застосунок, Android-розробка, багаторівнева валідація, кібергігієна.

Аналіз існуючих рішень у сфері кібербезпеки демонструє, що більшість комплексних антивірусних систем є надмірно ресурсоємними та перевантаженими функціоналом, що підвищує поріг їх використання для звичайних користувачів [1-2]. Водночас спеціалізовані сервіси для перевірки репутації URL-адрес здебільшого представлені у форматі вебплатформ, що ускладнює оперативну перевірку посилань, отриманих у мобільних месенджерах чи SMS-повідомленнях [3-4].Отже, розробка спеціалізованого мобільного застосунку «Anti-Phishing» для швидкого аналізу текстових повідомлень та виявлення шкідливих покликань є актуальним науково-практичним завданням.

Головною перевагою запропонованого програмного рішення є автоматизація процесу перевірки. Користувачу достатньо скопіювати підозріле повідомлення в буфер обміну та вставити його в застосунок. Система самостійно здійснює парсинг тексту, виділяє URL-адреси та проводить їх багаторівневу валідацію: первинно – за локальною базою даних (Honeypot-даних) відомих фішингових патернів, а вторинно — через інтеграцію з відкритим API (наприклад, Google Safe Browsing API) для перевірки актуальних загроз у реальному часі.

Основним актором системи є Користувач, для якого передбачено такі ключові прецеденти (Use Cases): ініціалізація перевірки тексту, перегляд результатів аналізу (маркування посилання як безпечного, підозрілого або критично небезпечного) та перегляд рекомендацій щодо протидії виявленій схемі шахрайства. Архітектура взаємодії користувача із системою наведена на рис. 1.

Технологічний стек розробки базується на сучасних стандартах створення мобільного програмного забезпечення. Застосунок реалізовано мовою Kotlin у середовищі Android Studio. В основу архітектури покладено патерн MVVM (Model-View-ViewModel) [3], що забезпечує чітке відокремлення бізнес-логіки аналізу посилань від візуального інтерфейсу, гарантуючи високу масштабованість та простоту тестування коду.

Для забезпечення безперебійної роботи інтерфейсу під час мережевих запитів до API перевірки використано бібліотеку Retrofit 2 у комбінації з механізмом асинхронного програмування Kotlin Coroutines. Локальне збереження історії перевірок та оновлення бази шкідливих доменів реалізовано за допомогою реляційної системи керування базами даних SQLite з використанням ORM-обгортки Room.

Створений програмний продукт дозволяє пересічним користувачам ефективно ідентифікувати загрози соціальної інженерії на етапі доставки повідомлення. Інтуїтивно зрозумілий інтерфейс та висока швидкість обробки даних роблять застосунок надійним інструментом для захисту особистої цифрової ідентичності та підвищення загального рівня кібергігієни в суспільстві.

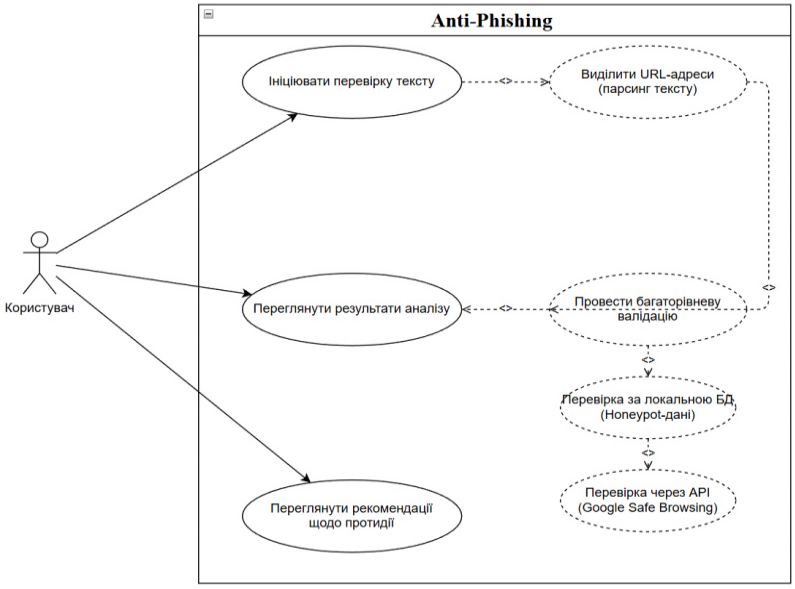

Рисунок 1 – UML-діаграма прецедентів, яка описує взаємодію користувача із системою

Окрему увагу приділено безпеці збереження локальних даних. Використання бібліотеки Room дозволяє реалізувати абстракцію над SQLite, що забезпечує перевірку запитів на етапі компіляції та підтримку міграцій бази даних. Для запобігання витоку пам'яті (memory leaks) під час аналізу довгих текстових повідомлень, обробка здійснюється в межах жизненного циклу ViewModel із використанням архітектурних компонентів Android Architecture Components.У подальшому планується впровадження алгоритмів машинного навчання безпосередньо на мобільному пристрої (On-device ML) для попередньої класифікації підозрілих повідомлень без необхідності звернення до зовнішніх серверів, що підвищить рівень приватності користувачів та швидкість відгуку системи у випадку нестабільного інтернет-з'єднання.

Список використаних джерел

1. Карпенко В. О. Аналіз сучасних векторів фішингових атак у мобільних комунікаційних мережах. Кібербезпека: освіта, наука, техніка. 2025. № 4. С. 112–120. Guide to app architecture / Android Developers. (дата звернення: 19.04.2026).

2. Mahmud T., Prince M. A. H., Ali M. H. Enhancing Cybersecurity: Hybrid Deep Learning Approaches to Smishing Attack Detection. Systems. 2024. Vol. 12, № 11. P. 490.

3. Schwarz S. F., Fonseca P., Rocha A. Smishing Detection From a Messaging Platform View. IEEE Access. 2025.

4. Mehmood M. K., Arshad H., Alawida M., Mehmood A. Enhancing Smishing Detection: A Deep Learning Approach for Improved Accuracy and Reduced False Positives. IEEE Access. 2024.