DOI 10.15276/imms.v14.no3.191
УДК 004.428.4

Informatics and Mathematical Methods in Simulation
Vol.14 (2024), No. 3, pp. 191-198

РОЗРОБКА IOS ЗАСТОСУНКУ ДЛЯ РІШЕННЯ ЗАДАЧ БЕЗПЕКИ МЕРЕЖЕВИХ ПІДКЛЮЧЕНЬ

А. М. Макарова, І. А. Ярова

Національний університет «Одеська політехніка»
1, Шевченка пр., Одеса, 65044, Україна
Emails: anastasia.mikki@gmail.com, yarova@op.edu.ua

Для підтримки мобільної мережі в ефективному та безпечному стані розроблено iOS застосунок, який являє собою спеціалізований комплекс інтегрованих мережевих інструментів для моніторингу, аналізу та діагностики мережевої активності. По результатах аналізу існуючих застосунків та досліджень в сфері проектування застосунків для мобільних пристроїв iOS показано, що більшість застосунків має вузьку спеціалізацію і доведено необхідність інтеграції мережевих інструментів в одному застосунку. В роботі визначено основну комплектацію для iOS застосунку інтегрованих мережевих інструментів, який має забезпечувати функції моніторингу мережі, виявлення вразливостей та оптимізації продуктивності. Обґрунтовано вибір стеку технологій для розроблюваного iOS застосунку. За допомогою діаграми послідовності визначено основні завдання системи, описано взаємодію користувача з нею, а також представлено компоненти системи iOS застосунку – об'єкти користувацького інтерфейсу та сервіси – та взаємодію між ними. Описано розробку логіки та користувальницького інтерфейсу iOS застосунку. Дизайн iOS застосунку виконано у стилі неоморфізм. Показано, що подальші можливості розширення функціональності розробленого iOS застосунку пов'язані із застосуванням технологій машинного навчання і штучного інтелекту.

Ключові слова: iOS застосунок, мобільні пристрої iOS, інтегровані мережеві інструменти, безпека мережевих підключень, моніторинг мережі, діагностика мережі

Вступ. Тенденція активної діджиталізації усіх сфер життя ставить перед користувачами завдання моніторингу та ефективного управління мережею. Мобільні пристрої, зокрема ті, що працюють з iOS, вже давно перетворилися на невід'ємну частину сучасного життя. Гаджети забезпечують постійний доступ до інформації та комунікацій. Але разом зі зростанням масштабів використання мобільних пристроїв зростають і ризики щодо безпеки та стабільності мережі [1, 2]. Гарантування безпеки та продуктивності власної мережі є критично важливим як для IT-фахівців, так і для звичайних користувачів [3, 4]. Існує багато різноманітних інструментів і способів моніторингу, аналізу та підтримки безпеки мережі, але мобільний застосунок, що об'єднує в собі набір функцій для роботи з мережею, є ефективним рішенням з точки зору доступності і економії часу. Компактний застосунок мережевих інструментів в смартфоні здатний замінити десктопні застосунки і сайти (web API), надати інструменти командного рядка тощо.

Програмне забезпечення для моніторингу та діагностики мережевих підключень є важливим інструментом щодо гарантування безпечного функціонування окремих користувачів і корпоративних мереж. Операційна система iOS підтримує високі стандарти безпеки, оскільки має високий рівень закритості екосистеми. Це дозволяє захищати пристрої від шкідливого програмного забезпечення і підвищувати їх стійкість до зловмисного втручання, але одночасно створює певні обмеження щодо доступу до глибоких мережевих параметрів. Як результат, виникає потреба в створенні спеціалізованих інструментів для аналізу мережевої активності [5]. Тому актуальним завданням є розробка iOS застосунку, який являє собою комплект інтегрованих мережевих інструментів.

Необхідність завантаження на мобільний пристрій певної кількості вузькоспеціалізованих застосунків, які досить часто не взаємодіють між собою, в кінцевому результаті підвищує вразливість пристрою [6]. Навпаки, інтеграція мережевих інструментів для моніторингу, аналізу та діагностики мережі в одному застосунку значно підвищує ефективність використання і швидкість реагування на можливі небезпеки. Користувачеві не потрібно перемикатися між різними застосунками і сайтами для виконання різних завдань – усе необхідне доступне в одному місці. Більш того, уніфікований інтерфейс та спільні дані між різними інструментами в одному застосунку забезпечують більш точний та повний аналіз мережевої активності.

Набір функцій застосунків для мобільних пристроїв є досить різноманітним. Досить часто вони передбачають можливість використання утиліти *ping*, призначеної для перевірки доступності мережевих вузлів, і утиліти *traceroute*, яка відстежує маршрут даних через інтернет. Таким чином визначається вузол, на якому відбувається затримка або втрата пакетів, що фактично є початком виявлення та усунення мережевих збоїв. Деякі мобільні застосунки мають функцію сканування локальної мережі із пошуком під'єднаних пристроїв. Це допомагає користувачеві виявити неавторизовані підключення, наприклад, зловмисника або неавторизованого користувача Wi-Fi. Зазвичай подібні застосунки виявляють і надають IP-адресу пристрою, іноді марку, тип, модель, ім'я, MAC-адресу. Деякі з них мають не тільки функцію виявлення неавторизованих пристроїв, але й виконують їх блокування. Більшість мобільних застосунків для сканування Wi-Fi надає можливість аналізу стану мережі, якості сигналу, наявності перешкод і каналів зв'язку [7, 8]. Подібний аналіз виконується з метою зменшення перешкод, виявлення неполадок, вибору оптимального каналу мережі, підвищення продуктивності мережі. В застосунках можна також зустріти функцію перевірки швидкості інтернету *Speed Test*. Вимірювання затримки, швидкості завантаження і вивантаження даних допомагають оцінити продуктивність інтернет-з'єднання і таким чином виявити недоліки в роботі інтернет-провайдера або налаштування мережі. Деякі застосунки надають інструменти для перевірки безпеки мережі: пошук відкритих портів, перевірка конфігурації маршрутизатора, аналіз рівнів шифрування Wi-Fi, а також формування рекомендацій щодо усунення потенційних загроз, виявлених за допомогою цих інструментів [9, 10].

Метою дослідження є розробка iOS застосунку у вигляді комплексу інтегрованих мережевих інструментів із функціями моніторингу мережі, виявлення вразливостей та оптимізації продуктивності. В якості функцій моніторингу мережі розглядається можливість отримання актуальної інформації про стан мережі: визначення переліку підключених пристроїв, маршрутизація та якість інтернет-з'єднання. Обрані функції діагностики призначені для виявлення потенційних загроз для мережі, перш за все, несанкціонованого доступу та недостатньо ефективного налаштування мережевих елементів. Функції аналізу мережевої продуктивності – перевірка швидкості інтернету та пінг – призначені для ідентифікації та усунення перешкод під час підключення до мережі.

Визначення основних функцій застосунку. Аналіз існуючих застосунків для мобільних пристроїв показує, що iOS застосунок мережевих інструментів повинен виконувати наступні функції: відображення загальної інформації про мережу, сканування мережевих інтерфейсів, відображення таблиці маршрутів, сканування локальної мережі для пошуку несанкціонованих підключень, утиліти *ping*, *traceroute*, *internet speed test*. Також мобільний застосунок має містити у собі функції надання загальної інформації щодо поточної мережі, відображення мережевих інтерфейсів, виведення таблиці маршрутизації, сканування локальної мережі для пошуку неавторизованих підключень, утиліти *ping*, *traceroute*, і тестування швидкості інтернету. Застосунок повинен мати зручний користувацький інтерфейс і локалізацію – застосовувати прийнятні для користувача мови. На кожному етапі розробки продукту

слід перевіряти відповідність користувацького сценарію реальній користувацькій взаємодії. Усі процеси створення продукту мають бути безперервними і циклічними [11].

Стек технологій. Оптимальними мовами програмування для розробки застосунка iOS, на наш погляд, є мови Swift, C, Objective C. В якості середовища розробки використовувалось інтегроване середовище розробки Xcode. Для розробки користувацького інтерфейсу був обраний фреймворк UIKit, який в даному випадку є більш придатним, ніж SwiftUI. Незважаючи на те, що фреймворк SwiftUI є більш декларативним і простішим для написання коду, він має певні обмеження для деяких задач [12]. Перевага UIKit над альтернативними фреймворками полягає в тому, що він є основним фреймворком для розробки користувацького інтерфейсу на iOS, має значну кількість інструментів і ресурсів, а також широку спільноту користувачів.

Діаграма послідовності. Для наочного відтворення процесів взаємодії між клієнтом і об'єктами та компонентами системи було побудовано діаграму послідовності (рис. 1). Аналіз діаграми послідовності дозволяє виявити потенційні неефективності у взаємодії об'єктів, зайві ускладнення, не оптимальні сценарії виконання [13].

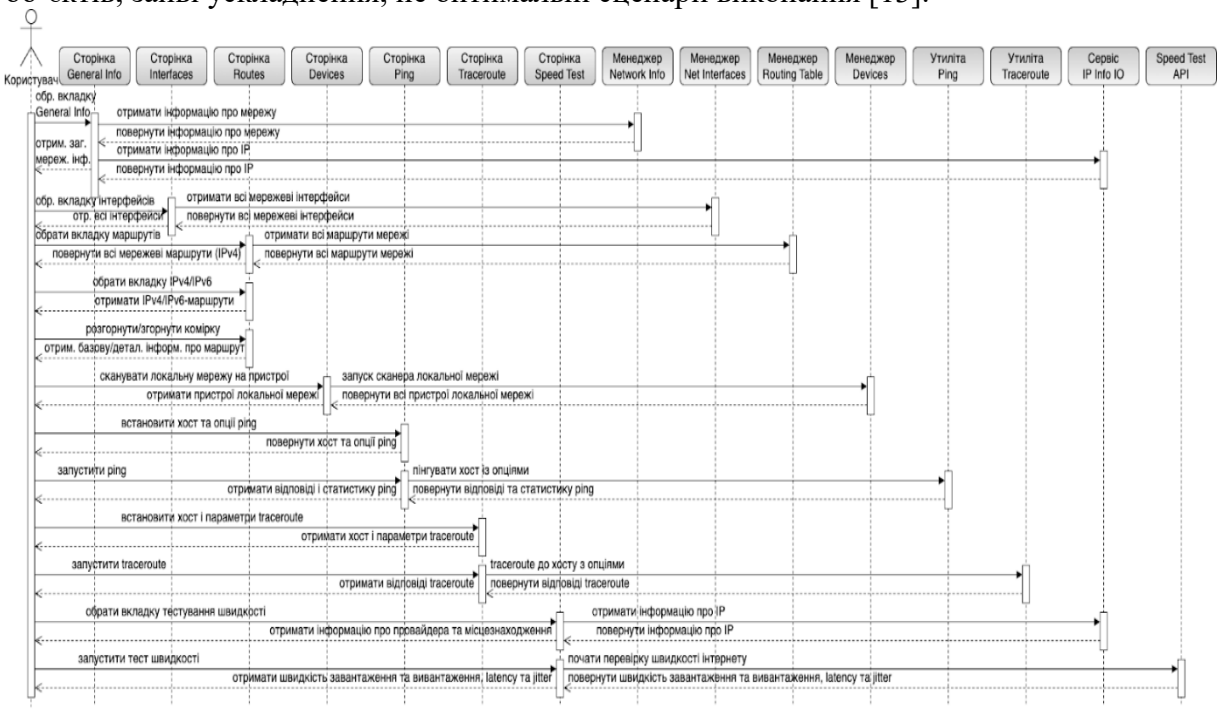

Рис. 1. Діаграма послідовності для розроблюваного iOS застосунку

Діаграма послідовності відображає об'єкти користувацького інтерфейсу та сервіси розроблюваного iOS застосунку. Користувацький інтерфейс – це сторінки програми і контролери. Через контролери відбувається звернення до сервісів для відображення інформації в користувацькому інтерфейсі і для відгуку на дії користувача. При переході користувача на сторінку *General Info* контролер запитує інформацію про мережу у сервісу. Сервіс повертає певні мережеві дані, наприклад IP-адресу маршрутизатора або мережевий шлюз. Таким чином у контролері створюються об'єкти мережевих даних, які впорядковуються і відображаються в користувацькому інтерфейсі. Під час переходу на сторінку мережевих інтерфейсів контролер запитує всі мережеві інтерфейси у менеджера. У контролері отримані інтерфейси групуються за протоколами IPv4 та IPv6 і відображаються в користувацькому інтерфейсі. При переході на сторінку мережевих маршрутів контролер запитує таблицю маршрутів у менеджера. У контролері отримані маршрути групуються за протоколами IPv4 та IPv6 і відображаються в різних вкладках. Під час зміни вкладок IPv4/IPv6 повторного запиту даних не відбувається. Об'єкт маршруту представлений у вигляді комірки, яка може бути розгорнути і згорнути.

Користувач переходить на сторінку *Devices* і натисканням на кнопку починає сканування мережі для отримання списку пристроїв у локальній мережі. Після цього відбувається звернення до сервісу щодо отримання списку пристроїв, контролер отримує об'єкти пристроїв і відображає їх у користувацькому інтерфейсі.

На сторінках *Ping* та *Traceroute* користувач має можливість встановити хост, на який буде виконуватися перевірка якості з'єднання, і опції, з якими він буде виконуватися. Операція (звернення до утиліти) починається після натискання на кнопку користувачем. Після завершення операції контролер отримує і відображає в користувацькому інтерфейсі ICMP-відповіді і статистику.

На сторінці *Speed Test* користувач має можливість отримати інформацію про швидкісні характеристики мережі. Під час переходу на сторінку контролер запрошує у сервісу інформацію про IP-адресу користувача. Таким чином користувач отримує інформацію щодо інтернет-провайдера і геолокації. Після натискання на кнопку запуску *Run Speed Test* відбувається вимірювання швидкості інтернету, в користувацькому інтерфейсі відображається швидкість завантаження і вивантаження даних, затримка і джитер – фазові спотворення сигналу, що передається.

Розробка дизайну застосунку. Перед початком розробки дизайну мобільного iOS застосунку результати проєктування були структуровані за компонентами проєкту і перенесені в платформу корпоративного програмного забезпечення *Atlassian Confluence* як документація проєкту. Відповідно до проєктної документації було реалізовано дизайн у веб-застосунку *Figma*, за яким розроблено iOS застосунок. Для дизайну iOS застосунку обрано стиль неоморфізм [14]. Цей стиль створює візуально-психологічний комфорт завдяки пастельним тонам кольорів, м'якості тіней і згладженості ліній. У дизайні iOS застосунку були використані векторні іконки, внутрішні та зовнішні тіні, градієнти.

Навігація по застосунку. Для навігації по iOS застосунку у ньому передбачене бічне меню з двома секціями: *Network Info* та *Utilities*, яке на екрані пристрою відображається ліворуч. Кожна секція містить панель вкладок, яка забезпечує навігацію між сторінками застосунка. У секції *Network Info* розміщені сторінки *General Info*, *Interfaces*, *Routes*, *Devices*. У секції *Utilities* розміщені сторінки *Ping*, *Traceroute*, *Speed Test*. Управління відображенням контенту виконує кореневий контролер. У кореневому контролері налаштовується кастомізований заголовок застосунка, який містить кнопки «Меню» та «Оновлення», а також відображає назву сторінки. Крім кнопки в заголовку iOS застосунка передбачено відкривання меню за допомогою свайп-технології. Контролер вкладок, який реалізує кастомізовану панель вкладок, призначений для відображення сторінок секцій через вкладки.

Реалізація логіки сторінок застосунку. Логіка сторінки загальної мережевої інформації реалізована в мережевому менеджері. Мережевий менеджер надає методи для отримання інформації про мережеві інтерфейси, IP-адреси, MAC-адреси, маску підмережі, адреси DNS-серверів, стан VPN-інтерфейсу та інші мережеві дані. Клас мережевого менеджера в основному використовує системні функції, бібліотеки і функції мовою C (*getifaddrs*, *resolv.h*, *inet_ntop*, *CFNetwork*).

На цій сторінці також відображаються дані про публічну IP-адресу, місцезнаходження, провайдера і ім'я хосту користувача. Для цього було розроблено сервіс, який реалізує взаємодію з API IPInfo.io. Структура даних API про публічну IP-адресу має властивості для IP, ім'я хосту, міста, регіону, країни, місця розташування, назви провайдера, поштового коду, часового поясу.

Логіка для сторінки *Interfaces* реалізована за допомогою сервісу мережевих інтерфейсів. Цей клас дає можливість ітерування за всіма доступними інтерфейсами і надає методи для їхньої фільтрації та отримання докладних характеристик. Серед властивостей об'єкту інтерфейсу: ім'я, IP-адреса, маска мережі, адреса призначення, перевірка належності до VPN. В структурі передбачені властивості для прапорів IFF_UP,

IFF_BROADCAST, IFF_LOOPBACK, IFF_POINTOPOINT, IFF_RUNNING, IFF_NOARP, IFF_MULTICAST, а також тип IP-адреси, тип адреси призначення, тип маски мережі.

Для реалізації логіки сторінки *Routes* було реалізовано класи менеджера маршрутів та об'єкту маршруту. Менеджер маршрутів надає методи для сканування маршрутів, отримання нещодавніх маршрутів і формування таблиці маршрутів. Серед властивостей моделі маршруту: адреса призначення маршруту, ім'я призначення маршруту, адреса шлюзу, ім'я шлюзу, прапори маршруту, посилання на маршрут, використання маршруту, ім'я мережевого інтерфейсу, сімейство маршруту, час закінчення маршруту. Інтерфейс користувача для сторінки *Routes* було реалізовано в контролері маршрутів. Контролер відображає маршрути мережі за вкладками IPv4/IPv6 у вигляді таблиці і реалізує перемикання між цими вкладками.

Сторінка *Devices* призначена для виявлення пристроїв, що підключені до локальної мережі. Для сканування мережі використовується бібліотека *MMLanScan*. Класичний мережевий сканер виконує пінг кожного хосту у мережі для побудови ARP-таблиці, після чого відбувається спроба отримання MAC-адреси кожного хосту. Якщо MAC-адресу знайдено, то вважається, що хост існує у мережі. Для пінгу використано бібліотеку *Apple SimplePing*. У контролері девайсів відбувається сканування та відображення пристроїв, підключених до локальної мережі. Якщо мережа стільникова або відсутнє підключення до мережі, користувачеві не надається можливість сканування. Для цієї функції також реалізовано ідентифікацію поточного пристрою.

Для перевірки цілісності і якості з'єднань в мережах використовується *ping.c* – реалізація стандартної утиліти *ping* на рівні коду C, що забезпечує повний контроль над ICMP-пакетами і дозволяє змінювати будь-які параметри, наприклад, розмір пакетів, тривалість існування пакетів в мережі (TTL), час між відправками. Для інтеграції *ping.c* у проєкт був створений клас, який включає необхідні властивості та методи для ініціалізації та виконання ICMP-запитів, таких як ім'я хосту, кількість пакетів, інтервал часу та обробка помилок. Була визначена структура для зберігання результатів, а також перелік можливих помилок при пінгуванні. Контролер реалізує інтерфейс користувача та логіку для виконання операцій *ping* у розробленому iOS застосунку. Він надає користувачеві можливість задавати наступні параметри для тестування підключення до мережі: хост, інтервал між пінгами, таймаут, розмір пакету та кількість пакетів.

Для реалізації операції визначення маршруту проходження даних від мобільного пристрою до сервера для зазначеного хосту було реалізовано окремий клас для утиліти *traceroute*. В розробленому класі доменне ім'я перетворюється на IP-адресу, після чого створюються сокети для відправки та отримання пакетів, ініціалізується значення TTL та інші змінні для вимірювання часу. Цикл *while* збільшує TTL на кожному кроці, до досягнення цільового хосту, або до перевищення максимального значення TTL. На кожному кроці на цільовий хост відправляються UDP-пакети зі збільшуваним TTL і очікується ICMP-відповідь від проміжних маршрутизаторів. Пакети надсилаються задану кількість разів для кожного TTL. При отриманні відповіді ICMP фіксується час проходження пакета і записується лог з інформацією про маршрутизатор. Якщо час очікування закінчився, або не отримано відповіді, у лог записується символ зірочки. При досягненні цільового хосту виконання припиняється. Через опції контролера встановлюються максимальна кількість переходів (*Bad Hops Limit*) та порт для визначення маршруту проходження даних.

Для тестування швидкості інтернету використовується *Speedchecker* SDK iOS, який дозволяє визначити такі параметри як затримка, швидкість завантаження та вивантаження з'єднання. SDK використовується для стільникових, бездротових та локальних мереж, надаючи деталі тестування, такі як поточна швидкість та прогрес. Методологія вимірювання швидкості інтернет-з'єднання ґрунтується на активних тестах з використанням локальних серверів та комерційних CDN, які передають великий обсяг інтернет-трафіку. Тест починається з відправки десяти ICMP-пакетів на сервер для

вимірювання затримки, після чого починається процес тестування швидкості завантаження та вивантаження, використовуючи два або десять потоків передачі HTTP-запитів, в залежності від можливостей з'єднання. Пакети даних передаються з високою швидкістю для повного залучення пропускної спроможності мережі, результати семплюються кожні 100 мс. Наприкінці тесту середня швидкість обчислюється як із семплів, так і з необроблених даних, після чого як остаточний результат обирається максимальне з двох значень.

Контролер для вимірювання швидкості передачі даних відображає процес тестування за допомогою кастомізованого елемента спідометру. Він візуально відображає поточну швидкість інтернет-з'єднання. Основними елементами є текстова мітка для швидкості передачі даних, стрілка та кругова шкала прогресу.

Висновки і подальші перспективи. За результатами аналізу спектру можливостей існуючих мобільних застосунків спроектовано iOS застосунок для мобільних пристроїв, який являє собою комплекс інтегрованих iOS засобів моніторингу та виявлення неполадок мобільної мережі. До набору функцій спроектованого iOS застосунку включено: сканування мережі, аналіз Wi-Fi, перевірку конфігурації маршрутизатора, аналіз рівнів шифрування Wi-Fi, перевірку швидкості інтернету, перевірку цілісності і якості з'єднань в мережах, визначення маршруту проходження даних тощо.

Розроблений iOS застосунок повністю відповідає поставленим функціональним вимогам. Усі функції розглянутого застосунку мають високу точність і швидкість виконання порівняно з еталонними інструментами. Завдяки широкому набору функцій та зручному інтерфейсу розроблений iOS застосунок може використовуватися як професіоналами, так і звичайними користувачами. Розроблений iOS застосунок з компактним набором мережевих інструментів підвищує швидкість реагування на мережеві події, що своєю чергою сприяє підвищенню рівня безпеки мережевого підключення з пристроїв iOS.

В цілому розроблений iOS застосунок має широкі можливості для розширення у вигляді застосункових мережевих інструментів для перевірки вразливостей мережі, технології VPN, брандмауера, сніфера трафіку. Серед застосункових функцій безпеки слід звернути увагу на функції виявлення компрометації даних користувача.

Подальші перспективи удосконалення розробленого iOS застосунку лежать в сфері застосування технологій машинного навчання і штучного інтелекту, які пропонують інноваційні рішення для усунення неполадок у мережах [15, 16]. Перспективним напрямком є автоматичне виявлення та діагностика помилок мережних підключень на основі моделей аномальної поведінки або систем попереджень, що використовують шаблони. Впровадження методів машинного навчання для прогнозування вразливостей і аналізу інцидентів в режимі реального часу є передумовою для підвищення ефективності управління мережевою безпекою. Для аналізу вразливостей можливе застосування методів обробки даних і методів NLP (обробка природної мови), які дають змогу збирати дані про загрози з відкритих джерел, оцінювати їх і класифікувати. Методи контрольованого навчання можуть бути впроваджені для пошуку шаблонів вразливостей в історичних даних.

Список літератури

1. Zihan Zhou. Distributed WSN Vulnerability Remediation System Based on Mobile-N Policy. 2023. URL: https://doi.org/10.21203/rs.3.rs-3740423/v1
2. Abdellaoui A., Elmhamdi J. Network Stability Based Multicriteria Weighted MPRs Selection Algorithm for Mobile Ad Hoc Networks. *International Journal of Communication Networks and Information Security*. 2024. vol. 16, № 2. 17 p. URL: https://doi.org/10.17762/ijcnis.v16i2.6668.
3. Goggin G. Apps: From mobile phones to digital lives. John Wiley & Sons, 2021. 154 p.
4. Rahul Ranjan, Ram Keshwar Prasad Yadav. A Decision Framework for Enhancing Adhoc Network Stability and Security. *International Journal of Innovative Science and Research*

Technology. 2024. V. 9, Iss. 10. P. 55–61. URL: https://doi.org/10.38124/ijisrt/IJISRT24OCT246

5. Faria Nawshin, Radwa Gad, Devrim Unal, Abdulla Khalid Al-Ali, Ponnuthurai N. Suganthan. Malware detection for mobile computing using secure and privacy-preserving machine learning approaches: A comprehensive survey. *Computers and Electrical Engineering*. 2024. Vol. 117. 24 p. URL: https://doi.org/10.1016/j.compeleceng.2024.109233.
6. Hohenegger V. Developing a Vulnerability Assessment Concept for eHealth iOS Applications. Diss. Wien, 2021. 157 p. URL: https://web.archive.org/web/20220115233717id_/https://repositum.tuwien.at/bitstream/20.500.12708/18842/1/Hohenegger%20Vanessa%20-%202021%20-%20Developing%20a%20Vulnerability%20Assessment%20Concept%20for...pdf
7. Jivthesh M. R., Gaushik M. R., Adarsh P., Niranga G. H., Rao N. S. A Comprehensive survey of WiFi Analyzer Tools. *2022 IEEE 3rd Global Conference for Advancement in Technology, Bangalore, India*. 2022. P. 1–8. URL: 10.1109/GCAT55367.2022.9972040 https://ieeexplore.ieee.org/abstract/document/9972040/metrics#metrics
8. 10 Best Wi-Fi analyzer apps for iPhone and iPad. URL: https://www.igeeksblog.com/best-iphone-ipad-wifi-analyzer-apps/
9. Masum M. R. iOS App Development Training. 2018. URL: 10.13140/RG.2.2.10443.23847
10. Iversen J., Eierman M. Learning Mobile App Development: A Hands-on Guide to Building Apps with iOS and Android. Prospect Press, 2017. 352 p.
11. Резнік Р. Ю., Антонов Ю. С. Розробка застосунків під платформи Android та iOS. *Прикладні інформаційні технології*: мат. всеукр. науково-практ. конф. Вінниця : Донецький національний університет імені Василя Стуса. 2020. С. 132–135. URL: https://www.researchgate.net/publication/355370862_ROZROBKA_DODATKIV_PID_PLATFORMI_ANDROID_TA_IOS
12. Mohamed Ahmed Eltaher. SwiftUI vs UIKit: A Comprehensive Comparison. URL: https://medium.com/@mohamed.ahmedeltaher/swiftui-vs-uikit-a-comprehensive-comparison-92f58507495f#:~:text=SwiftUI%20and%20UIKit%20are%20both,framework%20with%20extensive%20customization%20options
13. System Sequence Diagram Used in Software Development. URL: https://www.researchgate.net/publication/371904764_System_Sequence_Diagram_Used_in_Software_Development
14. Bjork S. Flat and neumorphic design: aesthetic preferences compared between age groups. *21st Student Conference in Interaction Technology and Design*. Umea University, 2021. P. 71–78. URL: https://www.diva-portal.org/smash/get/diva2:1574853/FULLTEXT01.pdf#page=75
15. Cimitile A., Martinelli F., Mercaldo F. Machine Learning Meets iOS Malware: Identifying Malicious Applications on Apple Environment. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy*. Porto, Portugal, 2017. P. 487–492. URL: 10.5220/0006217304870492
16. Amster A. Automating Vulnerability Detection in Networks with AI. URL: https://www.allstarsit.com/blog/automating-vulnerability-detection-in-networks-with-ai

А. М. Макарова, І. А. Ярова

THE DEVELOPMENT OF IOS MOBILE APP FOR SOLVING THE NETWORK CONNECTION SECURITY PROBLEM

A. M. Makarova, I. A. Yarova

Odesa National Polytechnic University
1, Shevchenko Ave., 65044, Odesa, Ukraine
Emails: anastasia.mikki@gmail.com, yarova@op.edu.ua

To maintain mobile network in an efficient and secure state, an iOS mobile app has been developed, which is a specialized set of integrated network tools for monitoring, analyzing, and diagnosing network activity. The functions of up-to-date mobile apps were analyzed and the researches in the field of designing of applications for iOS mobile devices. It is shown, that most apps are of narrow specialization. The necessity to integrate network tools in one mobile app is proven. The basic set of integrated network tools for the developed iOS mobile app has been determined, which should provide network monitoring, vulnerability detection, and efficiency optimization functions. The technology stack for the developed iOS mobile app has been selected. Using the sequence diagram, the main functions of iOS app system have been determined, the interaction between the user and the system is described, and interaction between the components of iOS app system, which are user interface objects and services, is presented. The development of iOS mobile app logic and user interface of iOS mobile app is described. The design of developed iOS mobile app is inspired by neomorphism style. It is shown that further possibilities for expanding the functionality of the developed iOS app are associated with the use of machine learning and artificial intelligence technologies.

Keywords: iOS mobile app, iOS mobile devices, integrated network tools, network security, network monitoring, network diagnostics