

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ВАСИЛЯ СТУСА

ЧЕРНЕГА ВІКТОРІЯ МИХАЙЛІВНА

Допускається до захисту:

в.о. завідувача кафедри
інформаційних технологій

канд. техн. наук, доцент

_____ О.В. Зелінська

« _____ » _____ 2024 р.

**МОБІЛЬНИЙ ДОДАТОК ДЛЯ ПОБУДОВИ КАРТИ WIFI МЕРЕЖ З
ВБУДОВАНОЮ РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ ДЛЯ
ПОКРАЩЕННЯ РОБОТИ ОБРАНОЇ МЕРЕЖІ**

Спеціальність 122 Комп'ютерні науки

Кваліфікаційна (магістерська) робота

Науковий керівник:

Антонов Ю. С., кан. фіз.-мат. наук,

доцент, доцент кафедри

інформаційних технологій

(підпис)

Оцінка: _____ / _____ /

(бали/за шкалою ЄКТС/за національною шкалою)

Голова ЕК: _____
(підпис)

Вінниця 2024

АНОТАЦІЯ

Метою кваліфікаційної роботи є закріплення на практиці отриманих протягом курсу теоретичних знань проходження освітньої програми «Комп'ютерні технології обробки даних» та здобування нових в процесі її реалізації. Результатом роботи є проектування, реалізація та тестування мобільного застосунку для побудови карти Wi-Fi мереж з вбудованою рекомендаційною системою для покращення роботи обраної мережі.

У вступі наведено актуальність, мету, предмет, об'єкт дослідження та розробки програмного засобу для моніторингу погодних умов, застосування даного програмного засобу та аналоги вітчизняного та зарубіжного виробника.

Перший розділ містить загальну інформацію про використані ресурси та теоретичний матеріал відносно теми, існуючі рішення, опис вимог, призначення та цілі створення розробленого продукту.

Другий розділ описує етапи проектування мобільного застосунку, інфологічну модель розробленого мобільного застосунку, обґрунтування вибору технологій.

Третій розділ присвячений реалізації програмного засобу за описаними в минулому розділі етапами, а також процес тестування програмного засобу.

У висновку підбитий підсумок проведеної роботи, а також обґрунтована значимість розробленого продукту.

Ключові слова: мобільна розробка, мобільний додаток, Wi-Fi, карта, Android, Kotlin, Compose.

78 с., 6 табл., 13 рис., 56 джерел.

ABSTRACT

The purpose of the qualification work is to consolidate in practice the theoretical knowledge gained during the course the passage of the educational program "Computer technologies of data processing" and the acquisition of new ones in the process of its implementation. The result of the work is the design, implementation and testing of an information system for building a map of Wi-Fi networks with a built-in recommendation system to improve the operation of the selected network.

The introduction shows the relevance of developing a weather monitoring mobile application and the use of it. Also it contains the discussion about the relevance, purpose, subject and object of the research and information about analogue systems developed in our country and all over the world.

The first section contains general information about used resources and theoretical material on the topic, existing solutions, a description of the requirements, purpose and goals of creating the developed product.

The second section describes the stages of the system design, infological model of the developed information system, the rationale for the choice of used technologies.

The third section describes the implementation of the information system in the stages specified in the previous section and the process of application testing.

The conclusion summarizes the accomplished work, as well as substantiated the importance of the developed product.

Keywords: mobile development, mobile application, Wi-Fi, map, Android, Kotlin, Compose.

ЗМІСТ

СПИСОК ПОЗНАЧЕНЬ І ПРИЙНЯТИХ СКОРОЧЕНЬ	5
ВСТУП	6
РОЗДІЛ 1 ТЕОРЕТИЧНІ СКЛАДОВІ БЕЗДРОТОВИХ МЕРЕЖ	9
1.1 Аналіз процесів та складових WI-FI мереж.....	9
1.2 Парадигми та загальні положення у розробці програмного забезпечення.....	24
1.3 Огляд існуючих рішень	35
РОЗДІЛ 2 ПРОЕКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПОБУДОВИ КАРТИ WI-FI МЕРЕЖ З ВБУДОВАНОЮ РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ПОКРАЩЕННЯ РОБОТИ ОБРАНОЇ МЕРЕЖІ	36
2.1 Етапи вирішення поставленого завдання.....	36
2.2 Загальна інформація про розроблюваний додаток	41
2.3 Обґрунтування вибору технології і мови програмування для розробки додатка/системи.	48
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ ДЛЯ ПОБУДОВИ КАРТИ WIFI МЕРЕЖ З ВБУДОВАНОЮ РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ПОКРАЩЕННЯ РОБОТИ ОБРАНОЇ МЕРЕЖІ	52
3.1 Опис авторських класів, функцій класів, що використані в реалізації	52
3.2 Опис інструкції користувачеві.....	59
3.3 Тестування програмного засобу	68
ВИСНОВКИ.....	72
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ	74

СПИСОК ПОЗНАЧЕНЬ І ПРИЙНЯТИХ СКОРОЧЕНЬ

WLAN	Wireless local area network
WIFI	Wireless Fidelity
IDE	Інтегроване середовище розробки
ООП	Об'єктно орієнтоване програмування
MVVM	Model – View – View Model
OS	Операційна система
API	Інтерфейс програмування додатків
REST	Передача репрезентативного стану
UI	Інтерфейс користувача
UX	Досвід користувача
QoS	Quality of Service
NAT	Network Address Translation

ВСТУП

На даному етапі розвитку інформаційних технологій запит на бездротові з'єднання стрімко зростає по всьому світу, особливо серед бізнес-сегменту. Користувачі з можливістю бездротового доступу до інформації можуть працювати більш продуктивно та ефективно, у порівнянні з колегами, які прив'язані до провідних мереж. Зі зростанням кількості мобільних користувачів, виникає потреба у швидких та ефективних засобах комунікації та обміну даними. Це призводить до активного розвитку технологій бездротового зв'язку, зокрема, бездротових локальних мереж (WLAN).

WLAN, або бездротова локальна мережа, забезпечує можливість з'єднання пристроїв без застосування дротів, використовуючи радіохвилі. Це дозволяє уникнути обмежень, пов'язаних з прокладанням кабелів, і забезпечує гнучкість та мобільність в налаштуванні мережі. Бездротові мережі дозволяють легко розгортати складні проєкти там, де проводи були б неефективними або неможливими з економічної точки зору. Крім того, вони забезпечують високу доступність і мобільність, і стають основою для розробки різноманітних пристроїв, включаючи IoT-рішення для корпоративних та приватних користувачів.

Сьогодні для створення WLAN та організації бездротових комунікацій переважно використовується технологія Wi-Fi. Це широко використовувана технологія, яка забезпечує надійну та швидку бездротову зв'язок для різних пристроїв у різних сферах життя. Саме через вище зазначені причини розробка інформаційних систем для побудови карти Wi-Fi мереж з вбудованою рекомендаційною системою для покращення роботи обраної мережі.

Метою роботи є закріплення знань, отриманих протягом проходження освітньої програми «Комп'ютерні технології обробки даних» в процесі розробки мобільного додатку для побудови карти Wi-Fi мереж з вбудованою рекомендаційною системою для покращення роботи обраної мережі.

В даній роботі буде розв’язана така задача: можливість аналізувати Wi-Fi мережі, що знаходяться в зоні дії девайсу, будувати карту Wi-Fi мереж із виведенням даних про кожную з точок доступу на ній з мобільного пристрою на операційній системі Android. Для досягнення цієї мети необхідно вирішити кілька ключових завдань:

1. Розробити архітектуру додатку з інтуїтивно зрозумілим та зручним для користувачів інтерфейсом;
2. Реалізувати функціонал для сканування Wi-Fi мереж, зокрема визначення їхніх параметрів, таких як сила сигналу, частота, протоколи безпеки тощо;
3. Розробити модуль для відображення інформації про мережі на інтерактивній карті з можливістю взаємодії;
4. Впровадити систему аналізу та рекомендацій для вибору найкращої мережі;
5. Спростити вибір оптимальної мережі для користувачів та забезпечити зручний спосіб взаємодії з інформацією про мережі
6. Провести комплексне тестування програмного забезпечення для перевірки його надійності, ефективності та коректності в різних сценаріях.

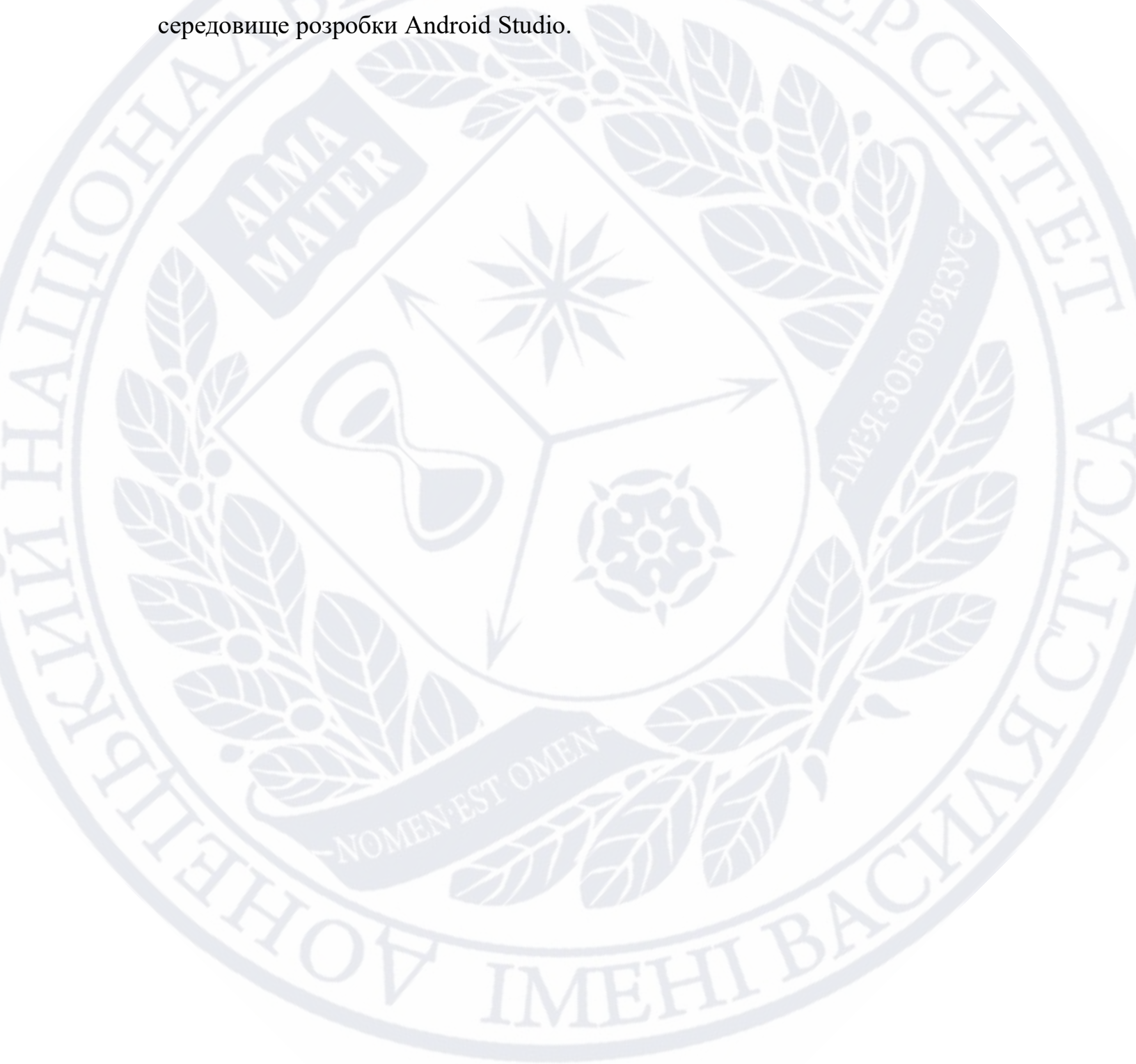
Об’єктом дослідження є процес інформатизації оптимізації локальної мережі у локації довкола користувача.

Предметом дослідження є методичні та практичні основи проектування мобільних застосунків для побудови карти Wi-Fi мереж з вбудованою рекомендаційною системою для покращення роботи обраної мережі.

Теоретична значимість даної роботи полягає в тому, що результати дослідження та проектування можуть використовуватись для реалізації інформаційних систем.

Практична значимість даної роботи полягає в можливості використання розробленого програмного засобу у повсякденному житті будь якої людини.

В даній кваліфікаційній роботі спроектовано та реалізовано мобільний застосунок для побудови карти Wi-Fi мереж з вбудованою рекомендаційною системою для покращення роботи обраної мережі. Для реалізації програмного коду та інтерфейсу користувача були використані мова програмування Kotlin та середовище розробки Android Studio.



РОЗДІЛ 1

ТЕОРЕТИЧНІ СКЛАДОВІ БЕЗДРОТОВИХ МЕРЕЖ

1.1 Аналіз процесів та складових WI-FI мереж

WLAN (wireless local area network) [1] – це форма мережі, яка побудована на бездротових технологіях, спрямованих на об'єднання комп'ютерів та інших пристроїв у межах обмеженої території, такої як будинок, офіс або кампус. Ця технологія дозволяє передавати дані між пристроями без необхідності проведення кабелів, замість цього використовуючи радіохвилі для забезпечення зв'язку та обміну інформацією.

У бездротових мережах передача інформації здійснюється за допомогою радіочастотних технологій [2], які використовуються для створення зв'язку між пристроями. Роботу бездротових локальних мереж (WLAN) забезпечують специфічні частоти в електромагнітному спектрі, які асоціюються з поширенням радіохвиль [3]. У цих мережевих пристроях встановлені як приймачі, так і передавачі, що дозволяє здійснювати двонаправлену комунікацію між ними, створюючи можливість обміну даними.

Серед ключового обладнання [4], що використовується для побудови бездротових мереж, можна виділити наступні пристрої:

1. Мережеві карти, що встановлюються в комп'ютерах та інших мережевих пристроях для забезпечення бездротового зв'язку.
2. Точки доступу, які служать центральними вузлами бездротової мережі, забезпечуючи з'єднання між бездротовими пристроями та провідною мережею.
3. Комутатори, які розширюють мережеву інфраструктуру, дозволяючи підключати до бездротової мережі більше пристроїв і керувати їхнім трафіком.
4. Ретранслятори, які покращують зону покриття мережі, підсилюючи та пересилаючи сигнал.

5. Мости та інше спеціалізоване обладнання, призначене для розвитку та оптимізації функцій бездротової мережі з урахуванням конкретних потреб користувачів.

При налаштуванні Wi-Fi-мереж передача радіосигналів здійснюється у радіодіапазоні 2,4 ГГц або 5 ГГц, в залежності від обраного стандарту та налаштувань мережі [5]. Точки доступу, розташовані у цих мережах, активно транслюють свій SSID (Service Set Identifier) [6], що представляє собою ідентифікатор мережі, який дозволяє пристроям користувачів знаходити та встановлювати зв'язок із мережею. Пристрої користувачів, які потрапляють у зону покриття Wi-Fi, автоматично надсилають запити на налаштування підключення до цієї мережі, ініціюючи процес встановлення зв'язку.

Мережева карта [7] – пристрій, який встановлюється в комп'ютерах або інших мережевих пристроях для забезпечення його підключення до мережі, будь то провідна або бездротова. Мережеві карти отримують, обробляють і передають дані через мережу.

Комутатор [8] – пристрій, який використовується в мережах для з'єднання різних пристроїв і керування передачею даних між ними. Комутатори дозволяють ефективно розподіляти трафік у мережі, відправляючи дані тільки до необхідних пристроїв.

Ретранслятори [9] – пристрій, який використовується для підсилення і пересилання бездротових сигналів, щоб забезпечити краще покриття мережі та підвищити якість зв'язку для бездротових пристроїв.

Міст [10] – це пристрій, який використовується для з'єднання двох або більше мереж на фізичному рівні для передачі даних між ними.

Маршрутизатор (роутер) [11] — це ключовий мережевий пристрій, що виконує важливу роль у процесі маршрутизації даних між різними мережами. Його основна функція полягає в прийомі пакетів даних, що надходять з однієї мережі, та їх подальшій обробці. Після прийому маршрутизатор аналізує заголовки цих

пакетів, щоб з'ясувати їх призначення та параметри. На основі отриманої інформації маршрутизатор визначає найбільш оптимальний та ефективний маршрут для передачі даних, враховуючи різні фактори, такі як завантаженість мережі, швидкість з'єднання та можливі затримки.

Основні функції маршрутизаторів [12]:

1. Маршрутизація даних: аналіз пакетів даних і визначення найефективнішого маршруту для їх передачі. Роутери використовують таблиці маршрутизації та протоколи (таких як OSPF, BGP), щоб забезпечити оптимальну передачу даних між мережами.

2. Перетворення IP-адрес: роутери виконують функції присвоєння та управління IP-адресами, а також забезпечують NAT [13] (Network Address Translation), що дозволяє кільком пристроям у локальній мережі використовувати одну публічну IP-адресу для виходу в Інтернет.

3. Безпека мережі: сучасні маршрутизатори оснащені брандмауерами, функціями шифрування, а також можливостями налаштування VPN (віртуальних приватних мереж) для захисту даних та запобігання несанкціонованому доступу.

4. Керування трафіком: маршрутизатори можуть реалізовувати механізми управління трафіком, наприклад, пріоритизувати трафік для різних видів даних, щоб забезпечити високу якість обслуговування QoS (Quality of Service) [14] для критичних додатків, таких як VoIP та потокове відео.

Види маршрутизаторів [15]:

1. Доступні маршрутизатори (використовуються для підключення локальних мереж до Інтернету, надаючи базові функції маршрутизації та безпеки).

2. Корпоративні маршрутизатори (призначені для використання в організаціях, забезпечують складні функції маршрутизації, управління трафіком, VPN та безпеки).

3. Мобільні маршрутизатори (використовуються для забезпечення доступу до Інтернету на ходу, часто за допомогою 4G або 5G з'єднань).

4. Безпроводні маршрутизатори або Wi-Fi маршрутизатори (забезпечують бездротовий доступ до мережі, використовуючи технології Wi-Fi для передачі даних).

Загальні характеристики та порівняння різних видів маршрутизаторів представлені у табл. 1.1.

Таблиця 1.1 – Порівняльна характеристика типів маршрутизаторів

Характеристика	Домашній маршрутизатор	Корпоративний маршрутизатор	Безпроводний маршрутизатор	Віртуальний маршрутизатор
Призначення	Підключення до інтернету для домашніх користувачів	Підключення офісних мереж та управління трафіком	Підключення бездротових пристроїв до мережі	Віртуалізація маршрутизаторів для хостингових мереж
Кількість портів	4-8 LAN	8 та більше	4-8 LAN та бездротовий модуль	Залежить від віртуальної архітектури
Швидкість з'єднання	До 1 Гбіт/с	Від 1 Гбіт/с до 100 Гбіт/с	До 1 Гбіт/с	Залежить від серверних ресурсів
Протоколи безпеки	WPA2, WPA3	Різноманітні VPN, IPSec, MPLS	WPA2, WPA3, WEP	Залежить від віртуальної архітектури
Управління	Просте управління через веб-інтерфейс	Розширене управління, SNMP, CLI	Просте управління через веб-інтерфейс	Управління через віртуалізаційні платформи
Ціна	Доступні (до 100\$)	Висока вартість (500\$<)	Доступні (до 150\$)	Вартість залежить від ресурсів віртуалізації
Підтримка QoS	Обмежена	Розширена	Обмежена	Залежить від реалізації

Приклад моделей	TP-Link Archer A7, Asus RT- AC66U	Cisco ISR 4000, Juniper MX Series	Netgear Nighthawk, Linksys EA8300	Cisco CSR 1000V, VyOS
--------------------	---	---	---	--------------------------

Точка доступу (Access Point) [16] – це пристрій, який виступає в ролі приймача та передавача в бездротовій мережі. В основному, цей пристрій дозволяє підключати бездротові пристрої до дротової мережі або до Інтернету. Він може бути також базовою станцією для організації мережі та керувати передачею даних між бездротовими пристроями.

Основні функції точки доступу включають [17]:

1. Передача сигналу: точка доступу отримує радіосигнали від бездротових пристроїв та маршрутизаторів і пересилає їх, щоб вони могли взаємодіяти між собою та з'єднуватися з дротовою мережею або Інтернетом.
2. Прийом сигналу: точка доступу також приймає радіосигнали від бездротових пристроїв і пересилає їх до маршрутизатора або іншої мережі для подальшої обробки та передачі даних.
3. Керування мережею: деякі точки доступу можуть мати функції керування мережею, такі як розподіл IP-адрес, налаштування безпеки, контроль доступу тощо.
4. Розширення покриття мережі: декілька точок доступу можуть бути розміщені в мережі для розширення зони покриття та покращення якості сигналу в областях з поганою доступністю.
5. Підтримка різних пристроїв: точка доступу може підтримувати різні стандарти бездротового зв'язку, такі як 802.11n, 802.11ac тощо, щоб забезпечити сумісність з різними бездротовими пристроями.

В залежності від дальності, на яку розповсюджується їх покриття, бездротові мережі можна класифікувати на чотири типи [18], кожен з яких має свої унікальні особливості:

- WPAN, або бездротові персональні мережі (Wireless Personal Area Networks), що призначені для підключення пристроїв в межах обмеженої особистої зони, такої як робочий стіл або робоча область.

- WLAN, або бездротові локальні мережі (Wireless Local Area Networks), які охоплюють більші території, такі як будинки, офіси або кампуси, і забезпечують бездротовий доступ до мережевих ресурсів у межах цих областей.

- WMAN, або бездротові міські мережі (Wireless Metropolitan Area Networks), які покривають великі території в межах міста, надаючи бездротовий доступ до мережі для мешканців та підприємств у міських середовищах.

- WWAN, або бездротові глобальні мережі (Wireless Wide Area Networks), що охоплюють великі географічні області та забезпечують бездротовий зв'язок у різних регіонах, включаючи навіть віддалені та важкодоступні місцевості.

Для створення бездротових мереж широко використовуються різноманітні технології, які відрізняються за своїми особливостями та застосуванням:

- ZigBee (стандарт IEEE 802.15.4) [19]: Ця технологія використовується для створення персональних мереж, особливо ефективно в системах "розумний дім". Завдяки своєму низькому енергоспоживанню та радіусу покриття від 1 до 100 метрів, вона забезпечує надійний зв'язок для різних пристроїв.

- Bluetooth (стандарт IEEE 802.15.1) [20]: Ця технологія володіє невеликим радіусом дії, що робить її ідеальним вибором для створення персональних мереж, зокрема, для підключення бездротових пристроїв до комп'ютерів та інших пристроїв.

- Wi-Fi (стандарт IEEE 802.11) [21]: Найпоширеніша технологія для створення бездротових локальних мереж (WLAN). Wi-Fi забезпечує швидкий та надійний зв'язок між пристроями в межах обмеженої території, такої як будинок, офіс або кафе.

- WiMAX (стандарт IEEE 802.16) [22]: Ця технологія призначена для створення бездротових міських мереж (WMAN) і забезпечує зв'язок на відстанях до

декількох кілометрів. Вона дозволяє підключати користувачів у великих містах до мережі з високою швидкістю передачі даних та стабільним зв'язком.

Wi-Fi мережа [23] – це бездротова мережа, яка дозволяє пристроям обмінюватися даними та отримувати доступ до Інтернету без потреби в провіднику. Ця технологія базується на радіохвилях для передачі сигналу між пристроями та маршрутизаторами. Поняття "Wi-Fi" відображає не лише аббревіатуру, що походить від "Wireless Fidelity" (що дослівно перекладається як "висока точність бездротової передачі даних"), але і відзначає широкий спектр стандартів передачі цифрових потоків даних через радіоканали. Цей термін став символом бездротового зв'язку, що забезпечує зручність і ефективність у спілкуванні та обміні інформацією.

Технологія Wi-Fi, вперше представлена стандартом 802.11 (відома як Wi-Fi 1) у 1997 році, відзначалася пропускнуою здатністю до 54 Мбіт/с. Протягом років розвитку цієї технології, максимальна швидкість передачі даних і надійність з'єднання постійно зростали. Нові версії стандартів, такі як 802.11ac, 802.11ax і 802.11be, вводили різні покращення для забезпечення ще кращої ефективності і продуктивності бездротових мереж. Наприклад, стандарт 802.11ac працює на частоті 5 ГГц і може підтримувати ширину каналу до 160 МГц, досягаючи максимальної швидкості передачі даних 3,5 Гбіт/с. 802.11ax та його додаткові варіанти (802.11ax та 802.11ax) розширюють діапазон частот, підтримують більшу ширину каналу і досягають значно вищої максимальної швидкості передачі даних - 9,6 Гбіт/с. Остання версія, 802.11be, надає ще більшу швидкість і пропускну здатність, працюючи на частотах 2,4 ГГц, 5 ГГц і 6 ГГц, з максимальною швидкістю передачі даних, яка сягає 46 Гбіт/с.

Різні стандарти Wi-Fi [24] визначають специфікації технічних характеристик та функціональних можливостей бездротової мережі. Кожен стандарт має свої унікальні особливості, які визначають його ефективність, швидкість та сумісність з іншими пристроями.

Стандарт 802.11b є одним із перших стандартів Wi-Fi, що був введений у 1999 році Інститутом інженерів електротехніки та електроніки (IEEE). Цей стандарт визначає технічні характеристики та протоколи для бездротового зв'язку у частотному діапазоні 2,4 ГГц. Характеристики та особливості стандарту 802.11b:

1. Швидкість передачі даних: стандарт 802.11b підтримує максимальну швидкість передачі даних до 11 Мбіт/с. Це була значна поліпшення порівняно з попередніми технологіями і дозволило швидше обмінюватися даними в бездротових мережах.

2. Частотний діапазон: 802.11b працює в частотному діапазоні 2,4 ГГц. Це частота, яка дозволяє пристроям бездротового зв'язку працювати на значній відстані один від одного та проникати через перешкоди, такі як стіни або перекриття.

3. Технологія модуляції: 802.11b використовує технологію Direct Sequence Spread Spectrum (DSSS). Ця технологія дозволяє розділяти сигнали на багато широкосмугових каналів, що забезпечує більшу стійкість до перешкод та завдяки цьому забезпечує стабільний зв'язок.

4. Модуляційна схема: стандарт використовує BPSK (Binary Phase Shift Keying) та QPSK (Quadrature Phase Shift Keying) для передачі даних. Ці схеми дозволяють кодувати біти даних у фазі сигналу, що забезпечує ефективну передачу інформації.

5. Дальність зв'язку: при нормальних умовах 802.11b може працювати на відстані близько 38 метрів. Однак ця відстань може змінюватися в залежності від конкретних умов середовища та наявності перешкод.

6. Сумісність: однією з переваг 802.11b була його сумісність з попередніми версіями Wi-Fi та пристроями, що підтримували ці стандарти. Це дозволило плавний перехід на нову технологію без повного заміщення існуючого обладнання.

7. Популярність: стандарт 802.11b був дуже популярним у перший час після введення через свою високу швидкість передачі даних та розповсюдження в багатьох сферах, включаючи домашнє використання, бізнес та освіту.

Хоча стандарт 802.11b вже застарів, він є важливим етапом у розвитку бездротових технологій, який поклав початок широкому поширенню Wi-Fi мереж та відкрив дорогу для подальшого розвитку швидших та ефективніших стандартів.

Стандарт 802.11g є наступним поколінням Wi-Fi після 802.11b та був впроваджений у 2003 році. Він вносить значні поліпшення порівняно з 802.11b, зокрема щодо швидкості передачі даних та дальності зв'язку. Характеристики стандарту 802.11g:

1. Швидкість передачі даних: однією з головних переваг 802.11g є здатність передавати дані зі швидкістю до 54 Мбіт/с, що вдвічі перевищує максимальну швидкість 802.11b. Це дозволяє більш швидко обмінюватися великими обсягами даних, такими як мультимедійні файли та потокове відео.

2. Частотний діапазон: 802.11g також працює в частотному діапазоні 2,4 ГГц, так само як 802.11b. Це дозволяє йому бути сумісним зі старішими пристроями, які підтримують 802.11b, а також забезпечує стійкий зв'язок на значній відстані.

3. Технологія модуляції: 802.11g використовує технологію Orthogonal Frequency Division Multiplexing (OFDM), яка дозволяє передавати дані по багатьом неспівмірних підканалах. Це робить стандарт більш стійким до шумів та перешкод і дозволяє досягати високої швидкості передачі даних.

4. Розширений діапазон: однією з переваг 802.11g є його здатність працювати на відстані до 38 метрів при нормальних умовах, що дозволяє користувачам отримувати доступ до бездротової мережі на великих територіях, таких як будинки чи офісні приміщення.

5. Сумісність: 802.11g є зворотно сумісним зі стандартом 802.11b, тобто він може працювати з пристроями, що підтримують попередній стандарт. Це дозволяє плавний перехід на нову технологію без повного заміщення існуючого обладнання.

6. Захист і безпека: стандарт 802.11g підтримує широкий набір методів захисту, включаючи WEP (Wired Equivalent Privacy), WPA (Wi-Fi Protected Access) та WPA2, що забезпечує безпеку передачі даних через бездротову мережу.

Стандарт 802.11g виявився значним проривом у розвитку Wi-Fi технологій, забезпечивши високу швидкість передачі даних та більшу стабільність зв'язку. Він став популярним у домашніх та бізнес-мережах і відкрив шлях для подальшого розвитку бездротових технологій.

Стандарт 802.11n є значним кроком у передачі бездротових даних, оскільки він приніс значні поліпшення порівняно зі своїми попередниками. Введений у 2009 році, 802.11n відкрив нові можливості для бездротових мереж, забезпечуючи вищу швидкість передачі даних, більшу дальність зв'язку та покращену якість сигналу. Характеристики та особливості стандарту 802.11n:

1. Швидкість передачі даних: однією з найбільших переваг 802.11n є його здатність досягати високої швидкості передачі даних. Стандарт може підтримувати швидкості до 600 Мбіт/с, що є значним покращенням порівняно з попередніми стандартами. Це дозволяє ефективно передавати великі обсяги даних, включаючи HD-відео та потокову музику.

2. Мінімізація втрат даних: 802.11n використовує технологію Multiple Input Multiple Output (MIMO), яка дозволяє використовувати кілька антен для передачі та отримання даних. Це поліпшує якість сигналу та мінімізує втрату даних через перешкоди та відбиття.

3. Частотний діапазон: 802.11n працює як в частотному діапазоні 2,4 ГГц, так і в діапазоні 5 ГГц, що забезпечує більшу гнучкість у виборі каналу та зменшує перевантаження на частотах 2,4 ГГц.

4. Розширений діапазон: стандарт 802.11n забезпечує покращену дальність зв'язку порівняно з попередніми стандартами. Він може працювати на відстань до 70 метрів в приміщенні та до 250 метрів на відкритому повітрі.

5. Сумісність: 802.11n є зворотно сумісним з попередніми стандартами, такими як 802.11a/b/g, тому він може працювати з існуючим обладнанням без необхідності повного оновлення.

6. Захист і безпека: стандарт 802.11n підтримує широкий спектр захисних механізмів, включаючи WEP, WPA та WPA2, що забезпечує безпеку передачі даних через бездротову мережу.

Стандарт 802.11n виявився значним кроком у розвитку Wi-Fi технологій, забезпечуючи вищу швидкість передачі даних, більшу дальність зв'язку та покращену стійкість сигналу. Він став популярним в домашніх, корпоративних та громадських мережах і стимулював розвиток нових застосувань, які вимагають великих обсягів даних та високої швидкості передачі.

Стандарт 802.11ac є значним кроком в еволюції Wi-Fi технологій, принісши значні покращення у швидкості передачі даних, дальності зв'язку та продуктивності мережі. Характеристики та особливості стандарту 802.11ac:

1. Швидкість передачі даних: однією з найважливіших переваг 802.11ac є його здатність досягати високої швидкості передачі даних. Стандарт може підтримувати швидкості до 3,5 Гбіт/с, що є значним покращенням порівняно з попередніми стандартами. Це дозволяє ефективно передавати великі обсяги даних, включаючи відео високої роздільної здатності та потокову музику, без затримок або втрати якості.

2. Частотний діапазон: 802.11ac працює тільки в діапазоні 5 ГГц, що дозволяє уникнути перевантаження та інтерференції, які можуть виникнути в діапазоні 2,4 ГГц. Це також дозволяє використовувати більш широкі канали для передачі даних, що сприяє збільшенню пропускної здатності мережі.

3. MIMO технологія: 802.11ac використовує технологію Multiple Input Multiple Output (MIMO) з більшим числом потоків даних, ніж попередні стандарти. Це дозволяє пристроям використовувати кілька антен для передачі та отримання даних одночасно, підвищуючи швидкість передачі та продуктивність мережі.

4. Beamforming: 802.11ac підтримує технологію Beamforming, яка дозволяє точково направляти сигнал до конкретного пристрою, замість того, щоб відправляти

його по всій області покриття. Це покращує якість зв'язку та ефективність передачі даних, зменшуючи інтерференцію та сприяючи збільшенню швидкості передачі.

5. Сумісність з попередніми стандартами: 802.11ac є зворотно сумісним з попередніми стандартами Wi-Fi, такими як 802.11a/b/g/n. Це означає, що він може працювати з існуючим обладнанням без необхідності повного оновлення.

6. Безпека: 802.11ac підтримує різноманітні захисні механізми, такі як WPA2, що забезпечує безпечну передачу даних через мережу Wi-Fi.

Стандарт 802.11ac виявився значним проривом у бездротових технологіях, забезпечуючи велику швидкість передачі даних, покращену якість зв'язку та ефективність мережі. Він знайшов широке застосування в домашніх, корпоративних та громадських мережах, стимулюючи розвиток нових застосувань, які вимагають високої швидкості передачі даних та надійного зв'язку.

Стандарт 802.11ax, також відомий як Wi-Fi 6, є останнім етапом розвитку бездротових мереж Wi-Fi, призначений для покращення швидкості передачі даних, продуктивності та ефективності мережі. Ось детальний огляд характеристик та особливостей стандарту 802.11ax:

1. Швидкість передачі даних: Стандарт забезпечує максимальну швидкість до 9,6 Гбіт/с, що вдвічі більше, ніж у попереднього стандарту 802.11ac. Це дозволяє передавати великі обсяги даних з вражаючою швидкістю, що робить його ідеальним вибором для сучасних вимог до мереж передачі даних.

2. Частотний діапазон: 802.11ax працює як в діапазоні 2,4 ГГц, так і в діапазоні 5 ГГц, що дозволяє максимально використовувати доступний спектр радіочастот для покращення продуктивності мережі.

3. OFDMA (Orthogonal Frequency Division Multiple Access): ця технологія дозволяє одночасно передавати дані до кількох пристроїв в одному каналі, зменшуючи час очікування та збільшуючи ефективність мережі. OFDMA розбиває канал на менші підканали, що можуть бути використані для передачі даних різним

пристроєм, що дозволяє оптимізувати використання доступного пропускнуго здатності.

4. MU-MIMO (Multi-User Multiple Input Multiple Output): 802.11ax підтримує більшу кількість потоків MU-MIMO, що дозволяє одночасно передавати дані до кількох пристроїв. Це робить мережу більш ефективною в умовах високого навантаження, коли багато пристроїв одночасно користуються мережею.

5. BSS Colouring: ця функція дозволяє розділити сусідні базові станції на різні кольори, щоб уникнути інтерференції між ними. Це покращує ефективність мережі та дозволяє краще використовувати доступний спектр радіочастот.

6. Зворотна сумісність: 802.11ax є зворотно сумісним з попередніми стандартами Wi-Fi, що означає, що він може працювати з існуючим обладнанням, що використовує старіші версії стандартів.

Стандарт 802.11ax є суттєвим кроком у розвитку Wi-Fi технологій, принісши значні покращення у швидкості передачі даних, ефективності та продуктивності мережі. Його висока швидкість передачі даних та покращена мультиплексація дозволяють впроваджувати нові застосування, що вимагають великої пропускнуго здатності та стабільного зв'язку.

Таблиця 1.2 – Порівняння стандартів Wi-Fi [25].

Характеристика	802.11b	802.11g	802.11n	802.11ac	802.11ax
Швидкість передачі даних	До 11 Мбіт/с	До 54 Мбіт/с	До 600 Мбіт/с	До 3,5 Гбіт/с	До 9,6 Гбіт/с
Діапазон частот	2.4 ГГц	2.4 ГГц, 5 ГГц	2.4 ГГц, 5 ГГц	5 ГГц	2.4 ГГц, 5 ГГц
Підтримувані функції	-	-	MIMO, OFDM, MU-MIMO, Channel Bonding	MIMO, OFDM, MU-MIMO, Channel Bonding	MIMO, OFDM, MU-MIMO, Channel Bonding, OFDMA, BSS Colouring

Зворотна сумісність	Так	Так	Так	Так	Так
Використання технологій	DSSS	OFDM	OFDM	OFDM, MU-MIMO	OFDM, MU-MIMO, OFDMA
Робочий діапазон	До 100 м	До 100 м	До 100 м	До 100 м	До 100 м

За допомогою Wi-Fi мереж можна побудувати карти таких мереж із детальною інформацією та характеристиками покриття.

Карта Wi-Fi мережі – це візуальне зображення бездротових точок доступу та їх покриття в певній області, такі як будинок, офіс, кампус або громадське місце. Вона відображає місцезнаходження точок доступу, сигнали, які вони випромінюють, та інші характеристики мережі. Головною метою карт Wi-Fi мереж є надати користувачам візуальне уявлення про покриття бездротового сигналу в певній області. Це дозволяє адміністраторам мережі та користувачам аналізувати ефективність та якість покриття Wi-Fi, виявляти слабкі місця та вдосконалювати мережу.

Основні функції карт Wi-Fi мережі включають:

1. Візуалізація покриття: Карти Wi-Fi дозволяють відобразити області, які покриваються сигналом від кожної точки доступу. Це допомагає визначити місця зі слабким сигналом або зони перекриття між точками доступу.
2. Оптимізація розташування точок доступу: Аналіз карти Wi-Fi дозволяє встановити оптимальне розташування точок доступу для забезпечення рівномірного та надійного покриття сигналом у всій області.
3. Виявлення перешкод: Карти Wi-Fi можуть допомогти виявити можливі джерела перешкод або інтерференції, такі як металеві конструкції, стіни, меблі або електричні пристрої, які можуть впливати на якість бездротового сигналу.

4. Планування розширення мережі: Адміністратори мережі можуть використовувати карти Wi-Fi для планування розширення мережі та встановлення нових точок доступу в стратегічних місцях.

5. Вирішення проблем: Карти Wi-Fi допомагають виявляти проблеми зі зв'язком, такі як зони зі слабким сигналом або перебої в роботі мережі, і забезпечують вихідні дані для їх вирішення.

Карти Wi-Fi мережі є важливим інструментом для аналізу, відладки та оптимізації бездротових мереж, що допомагає покращити якість зв'язку та ефективність роботи мережі.

Побудова карти Wi-Fi мережі вимагає використання різних ресурсів для збору та аналізу даних про бездротові сигнали. Деякі з найбільш поширених ресурсів:

1. Програмне забезпечення для аналізу Wi-Fi. Існують різні програмні засоби для аналізу бездротових мереж, такі як NetSpot, Ekahau HeatMapper, Acrylic Wi-Fi, InSSIDer, VisiWave Site Survey тощо. Ці програми дозволяють сканувати навколишні Wi-Fi мережі, вимірювати сигнали, визначати їх потужність, канали та інші параметри для побудови карти покриття.

2. Апаратне забезпечення для збору даних. Деякі спеціалізовані пристрої, такі як Wi-Fi адаптери, Wi-Fi аналізатори та Wi-Fi детектори, можуть використовуватися для збору даних про бездротові сигнали. Ці пристрої можуть мати більшу чутливість та функціональність, ніж звичайні смартфони або ноутбуки.

3. Карти побудови приміщень або територій. Для візуалізації та побудови карти покриття Wi-Fi часто використовуються карти приміщень, офісів, кампусів або інших областей. Ці карти можуть бути створені вручну або з використанням спеціалізованих програм для дизайну приміщень.

4. Дані від власників Wi-Fi мереж. Деякі власники Wi-Fi мереж можуть надавати дані про свої мережі для використання в побудові карт покриття. Це може включати інформацію про SSID, канали, потужність сигналу та інші параметри.

5. Геопросторові дані. Додаткові геопросторові дані, такі як координати місця розташування точок доступу та їхніх характеристик, можуть бути використані для створення точних карт покриття Wi-Fi.

Ці ресурси разом допомагають побудувати детальні та точні карти Wi-Fi покриття, які дозволяють користувачам оптимізувати розташування точок доступу та забезпечити надійний бездротовий зв'язок.

1.2 Парадигми та загальні положення у розробці програмного забезпечення

Об'єктно орієнтоване програмування (ООП) [26] – одна з парадигм програмування, що розглядає програму як сукупність взаємодіючих об'єктів, кожен з яких має свої властивості та можливості. Ця методологія базується на таких концепціях, як інкапсуляція, успадкування, поліморфізм та абстракція, які визначають спосіб організації та взаємодії об'єктів у програмі. Однією з переваг ООП є підвищена модульність програмного забезпечення. Там, де традиційні процедурні мови можуть вимагати значної кількості функцій, в ООП можна замінити цей обсяг кількома класами, кожен з яких містить в собі відповідні методи та функції для здійснення певного функціоналу. Класи в ООП служать шаблонами для створення об'єктів, які можуть мати свої власні властивості та методи, що відображає реальний світ та логіку роботи програми. Цей підхід дозволяє реалізувати принципи інкапсуляції, спадкування та поліморфізму, що значно підвищує гнучкість і можливості повторного використання коду.

Android [27] – операційна система, велика екосистема, розроблена компанією Google для різноманітних мобільних пристроїв, включаючи смартфони, планшети, носимі пристрої, телевізори та інше. Побудована на базі ядра Linux, Android відкритий за своєю природою, що дозволяє розробникам адаптувати та розширювати його функціональність. Однією з основних переваг Android є його велика гнучкість та наявність безлічі додатків, що забезпечують різноманітність

можливостей для користувачів, включаючи веб-перегляд, мультимедійні можливості, інтеграцію з сервісами Google, а також широкий спектр персоналізації.

Операційна система Android має інтуїтивний і легко налаштовуваний інтерфейс користувача, що дозволяє користувачам швидко і зручно працювати з пристроями [28]. Вона надає доступ до широкого спектру додатків через Google Play Store, де користувачі можуть знаходити та встановлювати різноманітні програми для використання на своїх пристроях. Також, Android має вбудовану підтримку для багатьох сервісів Google, таких як Gmail, Google Maps, Google Drive, YouTube та інші, що робить його ідеальним вибором для користувачів, які використовують екосистему Google. Операційна система постійно оновлюється та покращується, надаючи користувачам нові функції, збільшуючи продуктивність та покращуючи безпеку.

Kotlin [29] – це сучасна статично типізована мова програмування, яка створена для розробки на платформі Java Virtual Machine (JVM) [30] компанією JetBrains. Вона була створена з метою полегшення розробки мобільних, веб- та серверних застосунків. Однією з важливих особливостей Kotlin є те, що вона є 100% сумісною з Java, що означає, що ви можете використовувати Kotlin вже в існуючих Java-проектах. Крім того, Kotlin також може бути компільований в JavaScript, що розширює його використання та надає більші можливості для розробки веб-застосунків.

Kotlin надає розробникам багато переваг, особливо в контексті розробки для платформи Android [31], саме тому в останні роки вона стала дедалі популярнішою серед розробників Android-додатків.

По-перше, Kotlin є 100% сумісною з Java, що означає, що розробники можуть використовувати його в існуючому Java-кодi, або навпаки – Java-код у Kotlin. Це полегшує початок роботи з Kotlin, особливо для тих, хто вже має досвід розробки на Java.

По-друге, Kotlin пропонує чистий і сучасний синтаксис, що робить код більш зрозумілим і підтримуваним. Крім того, мова забезпечує високий рівень безпеки завдяки концепції null-safety [32], яка допомагає уникати частих помилок, пов'язаних із посиланнями на null. Це дозволяє значно зменшити кількість аварійних завершень програми, підвищуючи стабільність та надійність нашого додатку. З ясністю та чіткістю синтаксису, Kotlin дозволяє писати код, який є більш зрозумілим та легким для читання, що полегшує підтримку та оновлення в майбутньому.

По-третє, Kotlin дозволяє писати менше коду, щоб досягти того ж результату порівняно з Java. Це зменшує кількість помилок та робить розробку більш продуктивною.

Однією з ключових характеристик Kotlin є підтримка функціональності розширення, яка дозволяє додавати нові методи до існуючих класів без потреби створення підкласи [33]. Це означає, що є інструмент для інтегрування нових можливостей в уже наявні компоненти, зберігаючи при цьому чистоту та зрозумілість коду. У випадку мобільного застосунку, що працює з Wi-Fi мережами, така функціональність є надзвичайно корисною. Наприклад, при роботі з нативними компонентами, такими як слухачі Wi-Fi, розробники можуть легко додавати специфічні методи, що забезпечують отримання та обробку інформації про доступні мережі, без необхідності змінювати самі класи. Це не лише спрощує процес розробки, але й підвищує модульність коду, роблячи його більш зручним для тестування та подальшого обслуговування. Таким чином, вибір Kotlin для розробки додатку є обґрунтованим, оскільки його можливості розширення сприяють ефективній інтеграції нових функцій, що є критично важливим для реалізації комплексних рішень у сфері управління інформацією про Wi-Fi мережі.

Особливо актуальним функціоналом для додатку, що аналізує Wi-Fi мережі, є підтримка корутин в Kotlin [34], оскільки така система часто потребує виконання численних мережевих запитів і обробки великих обсягів даних у реальному часі.

Використання корутин дозволяє зберігати структуру і ясність коду, уникаючи при цьому проблем, пов'язаних із так званим "callback hell", що може суттєво ускладнити процес читання та підтримки коду. Завдяки корутинам можна реалізувати асинхронні операції без блокування основного потоку, що забезпечує більш плавний і чуйний інтерфейс користувача. Це не лише підвищує загальну продуктивність додатку, але й покращує досвід взаємодії користувачів з системою, оскільки вони можуть отримувати інформацію про стан Wi-Fi мережі в режимі реального часу без затримок та без блокування основного потоку, що відповідає за інтерфейс користувача.

Також, у мову програмування Kotlin вбудовано сучасні функції, які суттєво підвищують продуктивність розробки. Наприклад, data-класи [35] дозволяють легко створювати прості класи для зберігання даних з автоматичною генерацією методів equals(), hashCode() та toString(), що спрощує роботу з даними про Wi-Fi мережі, такими як їхні параметри та характеристики. Деструктуризація [36], у свою чергу, дозволяє зручно витягувати дані з об'єктів у читабельному вигляді, що підвищує зрозумілість коду. Оператори розширення [37] забезпечують можливість додавання нових функцій до існуючих класів без необхідності успадкування, що надає гнучкість у розширенні функціональності додатку, наприклад, для інтеграції з нативними компонентами Android, такими як сенсори Wi-Fi. Завдяки цим сучасним можливостям, Kotlin не лише зменшує обсяг коду, але й робить його більш зрозумілим і легким у підтримці, що є критично важливим для розробки складних систем, таких як мобільний застосунок для побудови карти Wi-Fi мереж.

Kotlin має велику спільноту розробників і отримує постійну підтримку від Google. Це створює сприятливе середовище для розробників, оскільки забезпечує доступ до численних ресурсів, бібліотек та інструментів, що можуть значно спростити процес розробки. Активна спільнота надає можливість швидкого отримання допомоги при виникненні питань, а також сприяє обміну досвідом між розробниками, що, в свою чергу, може підвищити якість продукту. В умовах

розробки мобільного додатку для побудови карти Wi-Fi мереж, наявність такого потужного ресурсу є незаперечною перевагою, адже це дозволяє легко знаходити готові рішення для інтеграції з різними API, включаючи Google Maps, а також впроваджувати нові функції без значних витрат часу на вирішення типових завдань.

Інструментальна підтримка Kotlin також є важливим аспектом, оскільки вона інтегрується з такими середовищами, як Android Studio, забезпечуючи потужні інструменти для налагодження, тестування та профілювання додатків. Використання Kotlin дає можливість активно застосовувати сучасні парадигми програмування, такі як функціональне програмування, що сприяє створенню елегантного та ефективного коду.

Clean Architecture [38] – це методологія проектування програмного забезпечення, яка покликана забезпечити чітке розділення відповідальностей між різними компонентами додатку та забезпечити його простоту, гнучкість та легкість тестування. У контексті Android-додатків, Clean Architecture стає особливо важливою, оскільки дозволяє створювати додатки, які легко розширюються, мають чистий та організований код і забезпечують швидку реакцію на зміни в бізнес-логіці.

Clean Architecture складається з різних шарів, які розділяють функціональність додатку та забезпечують незалежність між ними. Основні компоненти Clean Architecture включають:

1. **Presentation Layer (Презентаційний Шар):** В цьому шарі знаходяться компоненти, які відповідають за відображення даних та взаємодію з користувачем, такі як Activity, Fragment, ViewModel в архітектурі MVVM тощо.

2. **Domain Layer (Доменний Шар):** Цей шар містить бізнес-логіку додатку, яка не повинна залежати від конкретної реалізації інтерфейсів користувача або зовнішніх джерел даних. Він визначає основні правила та об'єкти, які характеризують додаток.

3. Data Layer (Шар Даних): Цей шар відповідає за доступ до даних додатку. Він може включати реалізації інтерфейсів для роботи з локальними та віддаленими джерелами даних, такими як бази даних SQLite, RESTful API, файлова система тощо.

Головна ідея Clean Architecture [39] полягає в тому, щоб забезпечити, щоб кожен шар був незалежним від інших, щоб можливість змінювати один шар без впливу на інші. Наприклад, зміни в бізнес-логіці не повинні впливати на інтерфейс користувача або спосіб отримання даних. У контексті Android-додатків, використання Clean Architecture дозволяє розробникам створювати додатки, які мають чистий та організований код, що полегшує його розуміння та підтримку. Крім того, такий підхід сприяє тестовій сумісності додатку, що дозволяє виявляти та виправляти помилки в ранніх стадіях розробки.

Model-View-ViewModel (MVVM) [40] – це архітектурний шаблон проектування, який сприяє чіткому розділенню об'єктів в додатку на три основні складові: Model, View і ViewModel. Він дозволяє покращити організацію коду та підвищити його повторне використання. У цій архітектурі кожен об'єкт має свій рівень відповідальності:

- Model – відповідає за збереження та обробку даних;
- View – відповідає за відображення графічного інтерфейсу користувача та взаємодію з користувачем;
- ViewModel – відповідає за підготовку даних, необхідних для відображення відповідного View.

Модульність [41] у додатку відіграє важливу роль, оскільки дозволяє ефективно розподілити функціональність програмного забезпечення на різні категорії, які відповідають за різні елементи його роботи. Це підходить не лише для покращення організації коду, а й для полегшення його подальшої підтримки та розвитку. У даному випадку додаток складається з єдиного модуля — app, який автоматично створюється під час ініціалізації проекту в середовищі розробки. Цей

модуль містить усі ключові файли, необхідні для збирання та компіляції програми, включаючи сценарії Gradle, файл маніфесту та інші важливі компоненти, які забезпечують належну роботу програми. Зосередженість на одному модулі спрощує управління проектом, дозволяє зосередитися на розробці основного функціоналу без зайвих ускладнень. Це також полегшує відлагодження та тестування, оскільки всі ресурси та залежності зосереджені в одному місці. Залежно від потреб у майбутньому, при необхідності можна легко розширити структуру проекту, додавши нові модулі для реалізації специфічних функцій або компонентів, що підвищить загальну гнучкість і масштабованість додатку.

RESTful API [42] – є стандартом архітектури програмного забезпечення, який використовується для створення веб-сервісів, які забезпечують доступ до даних та функцій через Інтернет. У контексті Android-додатків, використання RESTful API дозволяє забезпечити взаємодію між мобільним додатком та сервером, обмін даними та виконання операцій, таких як отримання, оновлення, створення або видалення ресурсів.

Основні принципи RESTful API [43] включають використання стандартних HTTP-методів (GET, POST, PUT, DELETE) для взаємодії з ресурсами, а також ідентифікацію ресурсів за допомогою унікальних URI (Uniform Resource Identifier). Клієнтська програма, наприклад, Android-додаток, може надсилати HTTP-запити до сервера за допомогою цих методів, отримуючи або надсилаючи дані відповідно до потреб додатку.

Для реалізації взаємодії з RESTful API в Android-додатках часто використовуються бібліотеки, такі як Retrofit або Volley. Retrofit, наприклад, є популярною бібліотекою для створення HTTP-клієнтів в Android, яка дозволяє визначити інтерфейси, які відображають структуру запитів і відповідей на сервері, і автоматично генерує код для взаємодії з цими запитами. Це спрощує процес роботи з RESTful API та забезпечує швидку та ефективну інтеграцію з веб-сервісами.

Jetpack Compose [44] – це інноваційний інструментарій, який рекомендується компанією Google для розробки нативного інтерфейсу користувача в Android додатках. Цей інструментарій змінює підхід до побудови інтерфейсу, використовуючи мову програмування Kotlin замість стандартних .xml файлів, які використовувалися раніше. Використання Kotlin дозволяє розробникам створювати інтерфейси швидше і ефективніше, не потребуючи додаткових розширень чи інструментів. Jetpack Compose пропонує декларативний підхід до розробки інтерфейсів, де користувальницький інтерфейс описується як набір компонентів, що полегшує розуміння та підтримку коду. Цей підхід дозволяє розробникам створювати динамічні, відзеркалені інтерфейси з меншою кількістю коду, що підвищує продуктивність та забезпечує більш гнучку розробку. Jetpack Compose також інтегрований з іншими компонентами Jetpack, що дозволяє розробникам легко використовувати його разом з іншими функціональностями платформи Android.

Compose є не просто новою бібліотекою, а справжнім проривом у розробці інтерфейсів користувача для платформ Android. Вона використовує декларативний підхід, що дозволяє розробникам описувати, як має виглядати інтерфейс користувача, зосереджуючись на логіці, а не на процесі оновлення елементів. Це означає, що програма автоматично адаптується до змін, що суттєво спрощує код і робить його більш читабельним.

Окрім того, Compose має потужну інтеграцію з Kotlin [45], що дозволяє використовувати всі переваги цієї мови, такі як нульова безпека та лямбда-функції, для створення інтуїтивно зрозумілого та ефективного інтерфейсу. Згідно з рекомендаціями Google, використання Compose зменшує кількість необхідного коду та скорочує час на реалізацію функціональності, що є особливо важливим у контексті проектування програмного забезпечення для побудови карти Wi-Fi мереж.

Compose дозволяє безперешкодно створювати адаптивні до різних розмірів екрану інтерфейси користувача. Бібліотека забезпечує використання гнучких

компонентів (наприклад, Column, Row, Box), які дозволяють легко розміщувати елементи один над одним, поруч чи в один контейнер. Це дозволяє автоматично адаптуватися до розмірів екрана, завдяки чому інтерфейс виглядає привабливо на будь-якому пристрої — від смартфонів до планшетів. За допомогою модифікаторів (наприклад, `Modifier.fillMaxWidth()` або `Modifier.size()`) розробники можуть створювати інтерфейси, які змінюють свій вигляд залежно від розміру екрана. Це означає, що інтерфейс може легко підлаштовуватися під різні розміри і формати, наприклад, вертикальне і горизонтальне положення. Завдяки декларативному стилю, будь-яка зміна в даних автоматично відображається на екрані без необхідності вручну оновлювати UI. Наприклад, якщо користувач змінює значення у формі, інші елементи інтерфейсу можуть динамічно реагувати на ці зміни, додаючи інтерактивність. Також, Compose дозволяє зберігати стан інтерфейсу у ViewModel, що робить його динамічним. Якщо дані змінюються, інтерфейс автоматично оновлюється, щоб відобразити ці зміни. Це особливо корисно при розробці таких функцій, як фільтри або сортування, коли користувач може змінити умови відображення даних. Гнучка архітектура Compose полегшує процес тестування інтерфейсу, оскільки можна ізолювати окремі компоненти і перевіряти їх функціональність в різних сценаріях.

Android Studio [46] — комплексне інтегроване середовище розробки, розроблене Google для створення програмного забезпечення для платформи Android. Це інструмент, який надає розробникам усе необхідне для розробки, налагодження та тестування Android-додатків. Android Studio є високопродуктивним інструментом, який дозволяє розробникам працювати на різних операційних системах, включаючи Windows, Mac та Linux. Це середовище розробки базується на популярному Інтегрованому середовищі розробки IntelliJ IDEA [47] і має широкий набір функцій та інструментів, призначених спеціально для розробки Android-додатків. Окрім того, Android Studio надає підтримку для

розробки додатків для різних пристроїв, таких як смартфони, планшети, Android TV, Android Wear і багато інших.

Крім того, Android Studio забезпечує інтеграцію з системою контролю версій Git [48], що дозволяє ефективно співпрацювати над проектом і відстежувати зміни у коді. Середовище також містить потужні інструменти для створення інтерфейсу користувача, зокрема підтримка Jetpack Compose.

Android Studio пропонує ряд потужних вбудованих інструментів, які значно полегшують процес налагодження та тестування мобільних додатків. Серед них:

1. Emulator [49]

Емулятор Android у складі Android Studio дозволяє тестувати додатки на віртуальних пристроях з різними конфігураціями, версіями Android та розмірами екранів. Це дозволяє перевіряти, як додаток виглядатиме і працюватиме на різних пристроях, не вимагаючи фізичного доступу до них. Емулятор підтримує всі функції реального пристрою, включаючи GPS, камеру, сенсори, і можливість підключення до мережі. Завдяки емулятору можна швидко тестувати та налагоджувати свої додатки в різних умовах.

2. Logcat [50]

Logcat є важливим інструментом для моніторингу та запису журналів подій у додатку. Він відображає всі повідомлення, які генеруються в додатку, включаючи повідомлення про помилки, попередження та інформаційні повідомлення. Це дозволяє швидко відстежувати, що відбувається під час виконання програми, виявляти помилки та аналізувати їх причини. Logcat надає можливість фільтрувати повідомлення за рівнем серйозності (наприклад, ERROR, WARN, INFO) або за тегами, що робить його надзвичайно зручним для глибокого аналізу роботи програми.

3. Debugger [51]

Вбудований налагоджувач (debugger) Android Studio дозволяє детально досліджувати код програми під час його виконання. З його допомогою можна

встановлювати точки зупинки (breakpoints), що дозволяє зупинити виконання програми на певній лінії коду і аналізувати стан змінних, стек викликів та інші параметри в реальному часі. Це є безцінним інструментом для виявлення логічних помилок і неправильних даних, оскільки розробники можуть покроково проходити через код і спостерігати, як змінюється стан програми.

4. Layout Inspector [52]

Layout Inspector — це інструмент, який дає можливість переглядати ієрархію макета інтерфейсу користувача в реальному часі. Завдяки цьому інструменту можна бачити, як різні елементи інтерфейсу взаємодіють між собою, які атрибути використовуються для кожного елемента та як вони розташовані на екрані. Це особливо корисно для виявлення проблем з відображенням інтерфейсу, таких як перекриття елементів, неправильні відступи або недоступність певних елементів для взаємодії. Layout Inspector також дозволяє перевіряти атрибути стилів і ресурсів, що спрощує налаштування дизайну.

5. Performance Profiler [53]

Профайлер продуктивності є ще одним важливим інструментом, який дозволяє аналізувати використання ресурсів додатком. Він надає візуалізацію даних про використання пам'яті, процесора, мережі та інших ресурсів, що дозволяє виявляти проблеми з продуктивністю. За допомогою цього інструмента можна моніторити, які частини коду споживають найбільше ресурсів, і, таким чином, оптимізувати їх для покращення швидкості роботи програми.

Android Studio постійно оновлюється та покращується, щоб забезпечити розробникам доступ до останніх технологій та інструментів, необхідних для ефективної роботи. Також воно має широкий спектр плагінів та розширень, які допомагають розробникам розширювати його функціональність відповідно до їхніх потреб і вимог.

1.3 Огляд існуючих рішень

Додаток для побудови карти Wi-Fi мереж з вбудованою рекомендаційною системою для покращення роботи обраної мережі – це програмне забезпечення для смартфонів або планшетів, призначене для сканування та аналізу бездротових мереж у певній області та створення детальної карти покриття Wi-Fi сигналу. Основною функцією такого додатку є надання користувачам інформації про якість сигналу, швидкість передачі даних, потужність сигналу та інші характеристики Wi-Fi мережі в різних точках простору. При дослідженні існуючих рішень було виявлено дві системи, які вирішують задачу побудову карти Wi-Fi мережі та рекомендаційної системи.

WiFi Analyzer [54] – мобільний додаток, що дозволяє користувачам аналізувати та проводити моніторинг бездротових мереж. Додаток може проводити сканування мереж, відображати графіки та графічні представлення з показниками якості покриття сигналу в різних частинах простору, перевіряти швидкість мережі, використані канали тощо.

NetSpot [55] – додаток, для аналізу та оптимізації бездротових мереж, шляхом аналізу ефективності поточної мережі та виявлення слабких місць сигналу. Додаток має функції створення карти покриття мережі Wi-Fi, аналіз безпеки мережі, виявлення інтерференції іншими пристроями, визначення швидкості та експорт із збереженням даних для обміну з іншими користувачами.

РОЗДІЛ 2

ПРОЕКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПОБУДОВИ КАРТИ WI-FI МЕРЕЖ З ВБУДОВАНОЮ РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ПОКРАЩЕННЯ РОБОТИ ОБРАНОЇ МЕРЕЖІ

2.1 Етапи вирішення поставленого завдання

Під час розробки мобільного додатку я обираю водоспадну модель, яка передбачає послідовну реалізацію проекту відповідно до чітко визначених етапів. Ця модель дозволяє структурувати процес розробки, забезпечуючи ясність і контроль на кожному етапі. Вона включає наступні ключові етапи вирішення поставленого завдання:

1. Формування вимог – на цьому етапі здійснюється детальний аналіз потреб користувачів та формування вимог до системи, що забезпечує розуміння мети та функціональності мобільного застосунку.

2. Проектування (інфологічна модель) – тут розробляється архітектура системи, визначаються основні компоненти та їх взаємозв'язки, що дозволяє створити інтуїтивно зрозумілу структуру даних і забезпечити логічну організацію інформації.

3. Реалізація – на цьому етапі програмний код фактично реалізується, а обрані рішення втілюються в життя, що є критично важливим для досягнення поставлених цілей проекту.

4. Тестування – завершує цикл розробки, забезпечуючи виявлення та виправлення можливих помилок, а також підтвердження відповідності системи визначеним вимогам та стандартам якості.

У процесі проектування програмного засобу для створення карти Wi-Fi мереж з вбудованою рекомендаційною системою особливу увагу буде приділено архітектурі додатку, що дозволить забезпечити гнучкість та можливість подальшого розширення функціональності. Передбачається, що додаток зможе легко

адаптуватися до збільшення кількості екранів з додатковими можливостями, такими як налаштування профілю користувача, перегляд історії підключень або управління доступними мережами.

Архітектура розроблюваного програмного засобу:

Архітектура розроблюваного додатку базується на принципі MVVM, що є рекомендованою Google для створення Android-додатків, дозволяє забезпечити гнучкість та можливість подальшого розширення функціональності. Цей підхід дозволяє ефективно розподілити обов'язки між трьома основними компонентами: Model, View та ViewModel, забезпечуючи при цьому чітке розмежування логіки програми та відображення даних. Передбачається, що додаток зможе легко адаптуватися до збільшення кількості екранів з додатковими можливостями. Відповідальність кожного класу:

1. Model – компонент, що відповідає за обробку, збереження та отримання даних. У контексті даного додатку, Model міститиме структури даних, які зберігають інформацію про Wi-Fi мережі, їх характеристики (такі як SSID, сила сигналу, тип шифрування) та рекомендації щодо оптимізації з'єднання. Model також взаємодіє з базою даних (можливо, з використанням Room) для зберігання та отримання даних, а також виконує запити до API для отримання актуальної інформації про мережі.

2. View – компонент, що відповідає за відображення інформації на екрані користувача. View використовує Jetpack Compose для створення інтуїтивно зрозумілого інтерфейсу, що дозволяє відображати карту Wi-Fi мереж з використанням Google Maps. З його допомогою користувач може взаємодіяти з картою мережі, натискати на маркери для отримання детальної інформації про доступні точки доступу, переглядати рекомендації та здійснювати налаштування. View в основному орієнтований на представлення даних, не містячи бізнес-логіки, що забезпечує простоту тестування та зміни в інтерфейсі.

3. ViewModel – компонент, що слугує посередником між Model та View, обробляючи дані, які будуть представлені в інтерфейсі. ViewModel адаптує

інформацію з Model для відображення у View, забезпечуючи збереження стану програми під час змін конфігурації, таких як обертання екрана. Завдяки використанню LiveData або StateFlow, ViewModel може сповіщати View про зміни в даних, що дозволяє автоматично оновлювати інтерфейс без ручного управління. У нашому випадку, ViewModel також реалізує бізнес-логіку, пов'язану з обробкою даних про Wi-Fi мережі, наприклад, фільтрацію доступних мереж за різними критеріями або надання рекомендацій на основі аналітики підключень.

Завдяки такій архітектурі, що базується на MVVM, мобільний додаток зможе ефективно обробляти дані, швидко реагувати на зміни стану, а також забезпечувати плавний та зручний користувацький досвід. Це дозволить користувачам легко орієнтуватися у додатку, взаємодіючи з картою Wi-Fi мереж та отримуючи актуальну інформацію про доступні мережі, а також рекомендації щодо їх оптимізації.

Для відображення інформації про предметну область в незалежному від способу реалізації форматі зручно використовувати інформаційно-логічну модель. Ця модель забезпечує інформаційно-логічний рівень абстрагування, який пов'язаний з описом об'єктів предметної області, їх властивостей та взаємозв'язків. У контексті розроблюваного мобільного засобу для побудови карти Wi-Fi мереж з вбудованою рекомендаційною системою важливо не лише врахувати різноманітність даних, а й структурувати їх так, щоб забезпечити ефективний доступ та обробку.

У даному мобільному застосунку системі необхідно зберігати дані про характеристики Wi-Fi мереж та їх параметри, а також взаємодію користувачів з цими мережами. Це дозволить не лише створити візуалізацію, але й надавати рекомендації щодо оптимізації підключення. Можна виділити основні об'єкти та побудувати інфологічну модель (рис 2.1):

1. Об'єкт, що зберігає дані про Wi-Fi мережі. Цей об'єкт міститиме всю необхідну інформацію про доступні мережі, такі як SSID, MAC-адреса, тип безпеки, а також інформацію про потужність сигналу.

2. Об'єкт, що зберігає дані про локацію та координати. У цьому об'єкті будуть зберігатися дані про географічні координати, а також інформація про назву локації. Це дозволить точно визначити, де саме розташовані Wi-Fi точки доступу.

3. Об'єкт, що містить інформацію про параметри мережі: Цей об'єкт зберігатиме дані про швидкість з'єднання, пропускну здатність, тип технології (2.4GHz, 5GHz) тощо.

4. Об'єкт, що містить короткий опис умов використання мережі. Цей об'єкт надасть інформацію про загальний стан мережі в даній локації, включаючи рекомендації щодо поліпшення з'єднання.

5. Об'єкт, що зберігає інформацію про технічні параметри пристроїв. Цей об'єкт буде зберігати дані про типи пристроїв, які підключаються до мережі, їх характеристики, такі як версія операційної системи, що вплине на сумісність і продуктивність додатку.

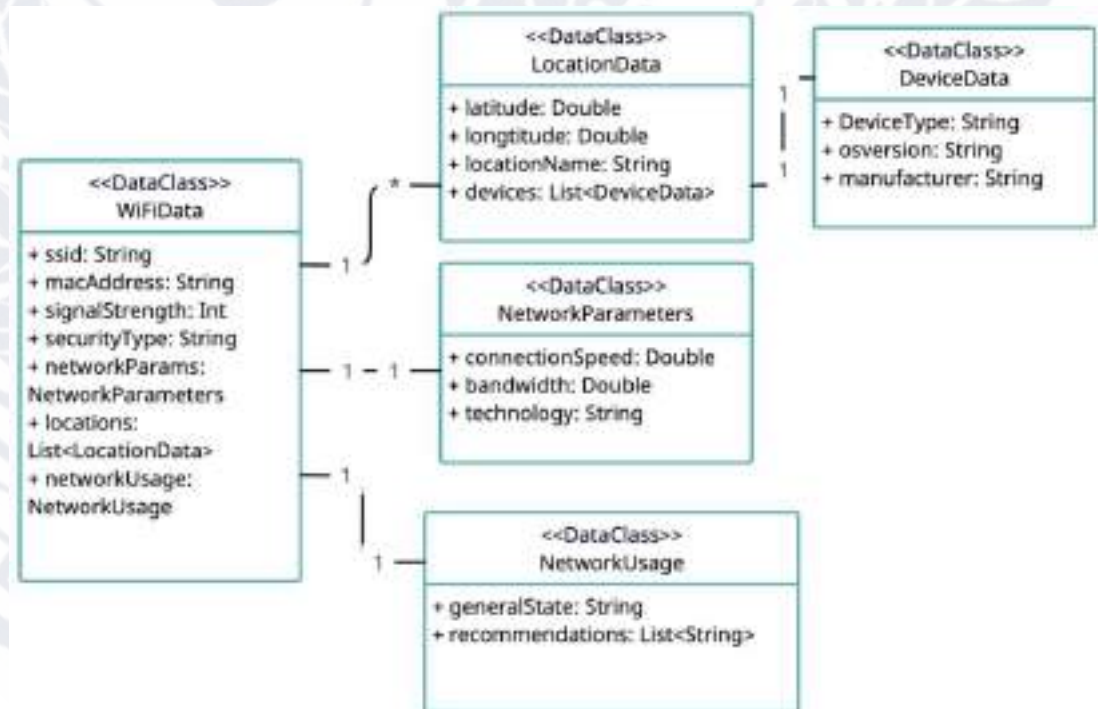


Рисунок 2.1 – Інфологічна модель розроблюваного додатку

Крім того, визначення мінімальних характеристик пристроїв, на яких буде можливість запускати даний додаток, є важливим етапом у процесі проектування мобільного застосунку. Це питання безпосередньо вплине на потенційну аудиторію користувачів, адже незадовільні характеристики можуть суттєво звузити коло тих, хто зможе скористатися розробкою. Важливо, щоб обрані параметри відповідали сучасним стандартам, оскільки це забезпечить кращу продуктивність, стабільність та зручність роботи додатку. Для досягнення максимального охоплення цільової аудиторії було враховано різноманітність використовуваних пристроїв, а також їх технічних характеристик, таких як процесор, обсяг оперативної пам'яті та ємність батареї, щоб усі користувачі мали можливість ефективно взаємодіяти з програмним продуктом.

Характеристики обраної операційної системи, необхідних додатків і служб: для коректної роботи програмного засобу, призначеного для побудови карти Wi-Fi мереж, необхідною умовою є доступ до Інтернету. Це дозволить користувачам отримувати актуальну інформацію про наявні мережі, їхні характеристики та стан. Додаток має бути оптимізований для роботи на пристроях з Android OS 9.0 або новішими версіями. Це рішення базується на тому, що такі версії ОС забезпечують доступ до сучасних API, які необхідні для реалізації функцій геолокації, роботи з картами та інтеграції з іншими сервісами, такими як Google Maps. Це не лише спростить реалізацію функціональності, а й дозволить забезпечити плавність роботи системи. Крім того, такі обмеження сприятимуть підвищенню продуктивності програми, оскільки нові версії ОС оптимізовані для ефективної роботи з ресурсами пристроїв, а отже, підвищать загальний користувацький досвід, зменшуючи ризик виникнення зависань чи інших проблем, пов'язаних із продуктивністю.

2.2 Загальна інформація про розроблюваний додаток

У рамках розробки мобільний додаток для побудови карти Wi-Fi мереж, яка включає в себе вбудовану рекомендаційну систему, здатна аналізувати параметри роботи мережі та пропонувати відповідні налаштування або оптимізації для покращення її функціонування, може бути використана відповідно до свого функціоналу, сприяючи підвищенню стабільності та продуктивності обраної мережі на основі даних про її поточний стан та навколишнє середовище.

Важливою складовою етапу проектування програмного забезпечення є визначення призначень і цілей його створення. Розробка мобільного додатку призначена для забезпечення ефективної обробки, зберігання та аналізу даних, автоматизації рутинних процесів, покращення взаємодії між користувачами та системою, а також для надання інструментів, що дозволяють приймати обґрунтовані рішення на основі отриманих результатів, сприяючи підвищенню продуктивності та оптимізації робочих процесів у визначеній сфері діяльності.

Розробка мобільного додатку призначена для:

1. Збирання та обробки даних про Wi-Fi мережі: система покликана автоматично виявляти доступні Wi-Fi мережі в певному географічному регіоні, збирати інформацію про їхні параметри, такі як сила сигналу, пропускна здатність, тип шифрування, а також інші технічні характеристики, що дозволяють оцінити якість мережі та її придатність для різних типів використання.

2. Побудови інтерактивної карти Wi-Fi мереж: на основі зібраних даних система здатна формувати карту Wi-Fi покриття, яка буде візуалізувати місця розташування точок доступу, їхні характеристики та якість сигналу в різних зонах, що допоможе користувачам знайти найбільш підходящі точки для підключення та уникати зон зі слабким покриттям.

3. Вбудованої рекомендаційної системи: система розроблена для аналізу зібраних даних та надання користувачам індивідуальних рекомендацій щодо налаштування або вибору Wi-Fi мережі, з урахуванням специфічних потреб

користувача, наприклад, для забезпечення стабільного з'єднання при потоковому відео чи онлайн-іграх.

4. Оптимізації роботи обраної мережі: мобільний додаток дозволяє не лише виявити слабкі місця в роботі Wi-Fi мережі, але й пропонувати рішення для їх усунення, таких як зміна каналу, налаштування частоти, або оптимізація розташування роутерів, що призведе до підвищення якості та стабільності інтернет-з'єднання.

5. Моніторингу продуктивності мереж: система також передбачає можливість постійного моніторингу якості Wi-Fi мереж у реальному часі, що дає змогу відстежувати зміни в продуктивності мережі, попереджати користувачів про можливі проблеми або збої, а також своєчасно реагувати на непередбачені зміни у стані мережі.

6. Забезпечення безпеки та конфіденційності: важливим призначенням мобільного застосунку є аналіз рівня безпеки Wi-Fi мереж, оцінка рівня шифрування та захисту, а також надання користувачам рекомендацій щодо підвищення безпеки підключення, що мінімізує ризики несанкціонованого доступу до мережі.

7. Покращення користувацького досвіду: розробка також спрямована на забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для користувачів, що дозволяє легко взаємодіяти з системою, отримувати корисну інформацію про мережі, слідкувати за рекомендаціями, і приймати оптимальні рішення щодо підключення до Wi-Fi мереж у будь-якому місці.

Мета створення програмного забезпечення:

Проектування та реалізація ефективного інструменту для збирання, аналізу та візуалізації даних про Wi-Fi мережі, з можливістю надання користувачам рекомендацій щодо оптимізації та покращення продуктивності обраної мережі. Основним завданням системи є створення інтерактивної карти покриття Wi-Fi мереж, яка допоможе користувачам вибирати найбільш підходящі точки доступу

для підключення, а також впровадження вбудованої рекомендаційної системи, що дозволить автоматично оцінювати та покращувати роботу мережі, пропонуючи оптимальні рішення на основі аналізу її технічних параметрів. Цей програмний засіб спрямована на підвищення зручності, якості інтернет-з'єднання та забезпечення безпеки при користуванні Wi-Fi мережами, що в результаті покращить загальний користувацький досвід на мобільному девайсі з операційною системою Android.

Важливо сформувати детальні та чіткі вимоги до програмного засобу, оскільки це є ключовим етапом, який дозволяє не лише забезпечити правильне розуміння функціональних і технічних аспектів майбутньої системи, але й дає змогу провести точні розрахунки щодо необхідних фінансових витрат, ресурсів, а також часу, який буде витрачено на розробку, тестування та впровадження програмного продукту, зокрема, для забезпечення його відповідності встановленим бізнес-цілям та технічним специфікаціям.

Програмний засіб повинен представляти собою мобільний додаток на платформі Android, що дозволяє користувачу переглядати інтерактивну карту Wi-Fi мереж у певній локації, а також отримувати рекомендації для покращення продуктивності та стабільності обраної мережі.

Вимоги до функціональних характеристик:

1. Мобільний застосунок повинен забезпечувати виконання наступних функцій:

- Можливість автоматичного сканування доступних Wi-Fi мереж у поточній локації користувача.
- Можливість введення користувачем конкретної локації для аналізу Wi-Fi мереж у цьому регіоні.
- Відображення інтерактивної карти з доступними Wi-Fi мережами, включно з їх точками доступу, рівнем сигналу та характеристиками.

- Відображення детальної інформації про кожну мережу, такої як назва мережі (SSID), тип шифрування, рівень сигналу, пропускна здатність, а також поточне навантаження на мережу.

- Аналіз якості Wi-Fi сигналу в різних точках на карті та візуалізація зон з поганим покриттям або перевантаженням.

- Вбудована рекомендаційна система, яка надає користувачам поради для покращення роботи мережі, наприклад зміна каналу, налаштування частот або розміщення додаткових точок доступу.

- Інформація про можливі перешкоди для сигналу (наприклад, від інших мереж або фізичних об'єктів) та рекомендації для їх усунення.

- Можливість збереження карт мереж для подальшого використання або аналізу.

- Надання рекомендацій щодо безпеки Wi-Fi мережі на основі аналізу рівня шифрування та доступних налаштувань.

2. Вихідні дані:

- Результат сканування доступних Wi-Fi мереж у поточній локації або локації, введеній користувачем.

- Інтерактивна карта з позначенням зон покриття та характеристикою Wi-Fi мереж.

- Рекомендації для оптимізації роботи Wi-Fi мережі на основі зібраних даних.

3. Вхідні дані:

- Поточна або задана локація користувача.

- Дані про доступні Wi-Fi мережі, зібрані під час сканування.

- Параметри мережі для аналізу та надання рекомендацій (наприклад, пропускна здатність, частота, рівень шифрування).

У застосунку реалізовано збереження та відображення вхідних, вихідних і оброблених даних у LogCat (журналі логів), що дозволяє візуалізувати процеси роботи з інформацією (див. табл. 2.1).

Таблиця 2.1 – Оброблені дані, що записуються в журнал логів

Назва об'єкта	Дані
Мережа Wi-Fi	Назва (SSID), Унікальний ідентифікатор точки доступу (BSSID), поточний рівень сигналу, частота, підтримувані протоколи, відстань до точки доступу, широта та довгота, рекомендації
Рекомендації	Найгірша мережа, найкраща мережа та посередня мережа (якість)
Стан карти	Характеристики бездротової мережі, локація пристрою

Одним із ключових аспектів у процесі планування програмного засобу є ретельне визначення дизайну інтерфейсу користувача, оскільки він безпосередньо впливає на зручність та ефективність взаємодії користувача з програмним забезпеченням. У розробці інтерфейсу особливу роль відіграє UI/UX-дизайнер, який спеціалізується на створенні інтуїтивно зрозумілих та зручних елементів взаємодії, що мають відповідати як технічним, так і естетичним вимогам. Для даного мобільного застосунку, який призначений для побудови карти Wi-Fi мереж із вбудованою рекомендаційною системою, надзвичайно важливим є забезпечення простого та інтуїтивного доступу до ключових функцій програми.

Опис графічного інтерфейсу користувача передбачає структурування застосунку навколо однієї основної активності (Activity), яка буде містити кілька взаємопов'язаних екранів. В рамках цієї активності планується створити два головних екрани з різними візуальними елементами для зручної навігації та взаємодії з програмою. Перший екран буде відображати карту Wi-Fi мереж у реальному часі, де користувач зможе переглядати точки доступу та їх

характеристики. Другий екран дозволить користувачу взаємодіяти з рекомендаційною системою, отримуючи поради щодо оптимізації вибраної мережі та її налаштувань.

Для полегшення користування будуть використані такі елементи, як кнопки, текстові поля, випадаючі меню та інтерактивні графічні компоненти. Інтерфейс повинен бути адаптивним і підтримувати різні розміри екранів, що є важливим для забезпечення зручного користування програмою на будь-якому пристрої. Важливо також забезпечити швидкий доступ до ключових функцій через мінімалістичний та логічно структурований дизайн, що дозволить користувачу зосередитися на основних задачах — побудові карти Wi-Fi мереж та оптимізації її роботи, не витрачаючи зайвий час на пошук потрібних функцій або елементів керування.

Швидкодія інтерфейсу користувача є важливим критерієм при розробці мобільних застосунків, і для забезпечення високої продуктивності програмного засобу буде використовуватись Jetpack Compose — сучасний інструмент для побудови UI на платформі Android. Завдяки декларативному підходу Compose значно спрощує процес створення інтерфейсу та підвищує його швидкодію. Це досягається за рахунок того, що оновлення UI відбувається лише при зміні стану елементів, що мінімізує кількість необхідних перерахунків та рендерів. У контексті проєктування мобільного застосунку для побудови карти Wi-Fi мереж, використання Compose дозволить створити легкий та динамічний інтерфейс, який швидко реагуватиме на дії користувача. Швидкодія Jetpack Compose також відзначається оптимізованим управлінням ресурсами, що допоможе уникнути затримок при відображенні інтерактивної карти Wi-Fi мереж або рекомендаційної системи. Також такий підхід допоможе забезпечити плавні переходи між екранами, що є ключовим для позитивного користувацького досвіду.

Для реалізації карти Wi-Fi мережі планується використовувати Google Maps API. Інтеграція програмного засобу із Google Maps є ключовим елементом, що значно підвищує функціональність та зручність використання додатку для побудови

карти Wi-Fi мереж. Завдяки використанню API Google Maps, застосунок може відображати точну карту місцевості, на якій будуть позначатися всі доступні Wi-Fi мережі у реальному часі. Це дозволяє користувачам швидко орієнтуватися у просторі, знаходити найближчі точки доступу та вибирати оптимальне місце для підключення, враховуючи інтенсивність сигналу та інші параметри мережі.



Рисунок 2.2 – Рекомендації, що можуть надаватись з використанням Google Maps API [56]

Крім того, інтеграція з Google Maps забезпечує можливість використання функцій геолокації, що дозволяє автоматично визначати поточне місцезнаходження користувача та відображати найближчі Wi-Fi мережі. Це особливо зручно для цільової аудиторії програмного засобу – мобільних користувачів, які пересуваються і потребують оперативної інформації про доступні підключення в різних районах. Завдяки цій функції, система стає більш інтерактивною, що значно покращує користувацький досвід. Для реалізації вимог до мобільного додатку буде застосовуватись функціонал з використання геопросторових даних для аналітики та рекомендацій, що дозволить системі надавати точні рекомендації щодо покращення

роботи обраної мережі залежно від її розташування, щільності підключень і інших географічних факторів (див. рис. 2.2).

Окрім цього, Google Maps API надає додаткові інструменти для кастомізації карти, що дозволяє налаштувати відображення необхідних елементів, таких як точки доступу Wi-Fi, маршрути до них, або інформаційні значки з додатковими даними. Таким чином, інтеграція з цим сервісом не лише підвищує функціональність додатку, але й дозволяє легко масштабувати та адаптувати інтерфейс під потреби користувачів у різних регіонах або ситуаціях.

2.3 Обґрунтування вибору технології і мови програмування для розробки додатка/системи.

Для розробки додатку була обрана об'єктно-орієнтована технологія, оскільки вона дозволяє розділити функціональність на класи, кожен з яких має свій об'єкт. Використання об'єктно-орієнтованого програмування (ООП) сприяє кращій організації коду, що робить його більш структурованим, зрозумілим і легким для підтримки та розширення.

Крім того, вибір об'єктно-орієнтованої технології надає можливість ефективно управляти складністю програми, що є критично важливим при розробці додатку, який має інтегрувати різноманітні функції, такі як побудова карти Wi-Fi мереж, використання Google Maps для відображення локацій, а також взаємодію з нативними компонентами Android, такими як слухачі Wi-Fi. Цей підхід дозволяє швидше реалізувати нові функції, зменшуючи час, необхідний для тестування та виправлення помилок, що, в свою чергу, позитивно вплине на загальний процес розробки.

Для реалізації мобільного застосунку, що забезпечує побудову карти Wi-Fi мереж з рекомендаційною системою, була обрана мова програмування Kotlin. Це сучасний інструмент, який ідеально підходить для розробки програмного забезпечення під платформу Android, завдяки своїй простоті, функціональності та

сумісності з Java. Kotlin дозволяє легко інтегруватися у вже існуючі Java-проекти, що дає змогу використовувати переваги обох мов та забезпечує плавний перехід до більш сучасних підходів у розробці. Мобільний застосунок покликаний реалізувати функціонал для побудови карти Wi-Fi мереж з інтерактивною взаємодією. Завдяки використанню Kotlin розробники отримують доступ до широкого спектра бібліотек, таких як бібліотеки для роботи з Google Maps, що забезпечують гнучкість та високу продуктивність системи. Крім того, використання Kotlin сприяє скороченню обсягу коду, підвищенню його читабельності та зменшенню ризику виникнення помилок, що особливо важливо для проектів із багаторівневою архітектурою.

Jetpack Compose – це сучасна бібліотека для розробки інтерфейсів користувача (UI) на платформі Android, розроблена компанією Google. Вона використовує декларативний підхід, що означає, що замість традиційного підходу до програмування інтерфейсу, де ви повинні вручну описувати, як і коли змінювати UI, ви описуєте, як повинен виглядати інтерфейс у певний момент часу, а бібліотека автоматично займається оновленням UI відповідно до змін у даних.

Однією з ключових переваг Jetpack Compose є його зручна інтеграція з існуючими бібліотеками Android, що робить його надзвичайно ефективним інструментом для розробки комплексних додатків. Це забезпечує можливість створювати сучасні інтерфейси, зберігаючи при цьому архітектурну цілісність та ефективність:

1. Взаємодія з ViewModel:

Jetpack Compose ідеально працює з основною архітектурою програмного засобу – MVVM (Model-View-ViewModel), що дозволяє розробникам зберігати стан UI окремо від логіки програми. ViewModel слугує як зв'язуючий елемент між UI (View) та даними (Model), забезпечуючи легкий доступ до даних, адже ViewModel дозволяє зберігати та керувати даними, які переживають зміни конфігурації, такі як повороти екрана, без втрати стану (це особливо корисно для динамічних інтерфейсів, де дані можуть часто змінюватись), а також швидко

оновлення інтерфейсу користувача, адже за допомогою LiveData або StateFlow в ViewModel, зміни в даних автоматично оновлюють інтерфейс користувача, що дозволяє створювати інтерактивні елементи, які реагують на зміни без необхідності додаткового коду для оновлення UI.

2. Використання LiveData та StateFlow

Jetpack Compose може безпосередньо підписуватися на LiveData або StateFlow, що є потужним способом реагування на зміни в даних, оскільки існують інструменти з використання декларативного стилю для опису UI компонентів, які автоматично реагують на зміни у LiveData або StateFlow, а також є можливість виконання операцій та отримання даних з результатів API запитів асинхронно.

3. Інтеграція з іншими фреймворками

Jetpack Compose також підтримує інтеграцію з багатьма іншими бібліотеками та фреймворками Android, такими як:

- Coroutines для виконання асинхронних запитів дозволяє спростити роботу з фоновими завданнями та підвищити ефективність виконання програм.
- Hilt або Dagger для впровадження залежностей, ці бібліотеки можуть бути легко використані разом з Compose, що дозволяє організувати код у зрозумілій та підтримуваній формі.

Зважаючи на складність та динамічність сучасних додатків, а також потребу в інтеграції з різними сервісами, такими як Google Maps, вибір Compose як основного засобу для створення інтерфейсу користувача робить процес розробки більш зручним та ефективним. В результаті, мобільний додаток для створення карти Wi-Fi мереж з вбудованою рекомендаційною системою отримує інструмент, який дозволяє швидше реалізувати ідеї, зберігаючи при цьому високу якість користувацького досвіду.

Отже для реалізації мобільного застосунку було вибрано та обґрунтовано вибір ОПП, Kotlin, Jetpack Compose та Android Studio як основних використовуваних технологій під час проектування та розробки.



РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАСОБУ ДЛЯ ПОБУДОВИ КАРТИ WIFI МЕРЕЖ З ВБУДОВАНОЮ РЕКОМЕНДАЦІЙНОЮ СИСТЕМОЮ ДЛЯ ПОКРАЩЕННЯ РОБОТИ ОБРАНОЇ МЕРЕЖІ

Додаток побудований як єдиний модуль, що структуровано розподілений на окремі підмодулі. Кожен підмодуль виконує конкретну роль у функціонуванні додатка, зокрема обробку запитів до сервера, управління даними, виконання операцій з ними та забезпечення взаємодії з користувачем через інтерфейс.

3.1 Опис авторських класів, функцій класів, що використані в реалізації

Авторські класи поділяються на 6 груп:

1. Activity – головна активність, яка виступає стартовою точкою додатку, забезпечуючи взаємодію між компонентами.
2. ViewModel – клас, що відповідає за обробку даних і підготовку їх до відображення у користувацькому інтерфейсі, а також за управління станами додатку.
3. Repository – проміжний клас, який забезпечує взаємодію з сервером, отримання, обробку та передачу даних іншим компонентам системи.
4. Model – локальний об'єкт, що використовується для зберігання даних, а також для їх виведення у зрозумілому для користувача форматі. Моделі створюються на основі Entity об'єктів.
5. Entity – спеціальний об'єкт, який приймає та зберігає дані, отримані із сервера, забезпечуючи їх структурованість.
6. Composable функції – функції, які відповідають за динамічне формування користувацького інтерфейсу, керування елементами та їх відображення.

Опис авторських класів та методів:

1. MainActivity – розширює ComponentActivity і є головною активністю додатку. Використовується для ініціалізації інтерфейсу та управління життєвим циклом. Анотація @AndroidEntryPoint вказує, що клас використовує Dagger Hilt для залежностей.

- onCreate – функція, що використовує Jetpack Compose для рендерингу інтерфейсу через функцію setContent. Вмикає режим edge-to-edge для розширення екрану. Перевіряє стан сканування мереж та керує навігацією через AppWithBottomNavigation.

2. LoadingNetworksScreen – функція, що відповідає за відображення екрану завантаження під час сканування Wi-Fi мереж. Використовує компонент LottieAnimationComponent для анімації завантаження та текст "Scanning...".

3. PermissionsHandler – функція для роботи з runtime-дозволами, використовує бібліотеку Accompanist для управління кількома дозволами. Обробляє три стани:

- Усі дозволи надані — викликається content.
- Відмова у дозволах із поясненням — відображається повідомлення із кнопкою "Try again".
- Повна відмова у дозволах — пропонує користувачу перейти до налаштувань через GoToSettings.

4. GoToSettings – функція, що відкриває налаштування додатку для надання дозволів. Використовує Intent для переходу до налаштувань додатку через URI.

5. AppWithBottomNavigation – функція, що відображає основний інтерфейс з нижньою панеллю навігації (BottomNavigationBar). Використовує Scaffold для компоновання інтерфейсу.

6. AppNavigation – функція, що реалізує навігацію між екранами за допомогою NavHost. Визначає три маршрути:

- NetworksScreen — відображає список Wi-Fi мереж.
- MapScreen — показує карту з мережею Wi-Fi.
- RecommendationsScreen — надає рекомендації на основі мереж.

7. BottomNavigationBar – функція, що відображає нижню панель навігації з кнопками для перемикання між екраном мереж, картою та рекомендаціями.

Обробляє стан активного маршруту (`currentRoute`) для візуального виділення активної кнопки. Кожна кнопка запускає відповідний маршрут через `NavController`.

8. `BottomNavItem` – клас, що визначає елементи нижньої панелі навігації та містить три вкладені об'єкти, для кожного з яких визначені маршрут (`route`), іконка (`icon`) та мітка (`label`):

- `Networks` — маршрут для екрану мереж.
- `Map` — маршрут для карти.
- `Recommendations` — маршрут для рекомендацій

9. `MainState` — це `data class`, який розширює `BaseState`. Він використовується для управління станом у `MainViewModel` і надає інформацію про поточний стан додатка, зокрема щодо сканування Wi-Fi мереж, отриманих мереж та місця розташування пристрою. Поля класу:

- `isLoadingNetworks` — вказує, чи відбувається сканування Wi-Fi мереж. Використовується для відображення стану завантаження у UI (`LoadingNetworksScreen`).
- `wifiNetworks` — список знайдених Wi-Fi мереж. Використовується для передачі списку мереж до екрану `NetworksScreen` і інших компонентів.
- `deviceLocation` — координати місця розташування пристрою у форматі (широта, довгота). Використовується на екрані `MapScreen` для відображення поточного місця розташування на карті.

10. `MainViewModel` — це `ViewModel`, що керує основною логікою додатка, включаючи пошук Wi-Fi мереж та визначення місця розташування пристрою. Клас використовує залежності для роботи з Wi-Fi, GPS та підтримує структуру `BaseViewModel` із системою станів. Анотація `HiltViewModel` вказує на використання `Hilt` для впровадження залежностей. Функції:

- `startScanningForNetworksAndLocation` — відповідає за ініціалізацію сканування Wi-Fi мереж і отримання геолокації. Вмикає Wi-Fi та реєструє `wifiScanReceiver`.

- `getScanResults` – перевіряє наявність дозволів на доступ до геолокації. Запитує останнє місце розташування пристрою та обробляє результати сканування Wi-Fi мереж через `processScanResults()`.

- `processScanResults` — обробляє список знайдених Wi-Fi мереж (групує мережі за SSID, визначає мережу з найсильнішим сигналом для кожного SSID, обчислює приблизні координати мереж).

- `calculateDistance` — розраховує відстань до точки доступу на основі RSSI (рівня сигналу).

- `calculateDestinationPoint` – обчислює приблизні координати точки доступу на основі відстані та напрямку.

- `showLoadingNetworks` – оновлює стан `isLoadingNetworks` у `MainState`.

- `onCleared` – скасовує реєстрацію `wifiScanReceiver`.

11. `LottieAnimationComponent` — це `composable` функція, яка відповідає за відображення анімацій у форматі Lottie. Ця функція абстрагує основну логіку відтворення анімацій, забезпечуючи простий інтерфейс для використання Lottie у Compose.

12. `MapScreen` – це `composable` функція, яка реалізує екран карти та відповідає за взаємодію з `ViewModel`, передачу стану (`state`) в компонент UI та забезпечує зручність навігації. Аргументи:

- `viewModel` (`MapViewModel`) – керує станом та бізнес-логікою для цього екрану.

- `wifiNetworksList` — список доступних Wi-Fi мереж, отриманий з іншого екрану або джерела даних.

- `deviceLocation` — координати пристрою у вигляді широти та довготи. Може бути `null`, якщо локація невідома.

- `onNavigate` – Лямбда-функція, яка забезпечує навігацію. Приймає рядок як параметр для визначення маршруту.

13. `MapState` – це клас даних, який представляє стан для екрану карти в `Jetpack Compose`. Він є дочірнім від `BaseState` і використовується для зберігання інформації про доступні Wi-Fi мережі та місцезнаходження пристрою.

- `wifiNetworksList` — список доступних Wi-Fi мереж, представлених об'єктами типу `WifiNetwork`, за замовчуванням — порожній список.

- `deviceLocation` — координати пристрою у вигляді пари чисел (`Pair<Double, Double>`), що представляють широту та довготу.

14. `MapLayout` – компонент для відображення інтерактивного інтерфейсу з інформацією про Wi-Fi мережі та карту. Параметри:

- `State` – стан екрану, що містить список мереж (`wifiNetworks`) і місцезнаходження пристрою (`deviceLocation`).

- `onIntent` – функція-обробник намірів користувача для взаємодії з `ViewModel`.

15. `WifiNetworkItem` – відповідає за відображення одного елементу списку Wi-Fi мереж.

16. `WifiNetworksMap` – відображає карту Google з маркерами для місцезнаходження пристрою та координат Wi-Fi мереж.

17. `NetworksScreen` – відповідає за відображення екрану, що взаємодіє зі станом `NetworksViewModel` та відображає інформацію про список Wi-Fi мереж через `NetworksLayout`.

18. `NetworksLayout` – є основним макетом для відображення списку мереж.

19. `WifiNetworkItem` – відповідає за відображення інформації про окрему Wi-Fi мережу. Відображає SSID, BSSID, рівень сигналу (у dBm), формований рівень сигналу, частоту, характеристики мережі та іконку сигналу, яка вибирається за допомогою функції `getWifiSignalIcon`.

20. `getWifiSignalIcon` – функція, що повертає ідентифікатор ресурсу для відповідної іконки сигналу Wi-Fi залежно від рівня сигналу (level). Логіка:

- Якщо рівень ≥ 4 — найсильніший сигнал (повна шкала).
- Якщо рівень ≥ 3 — середній сигнал.
- Якщо рівень ≥ 2 або ≥ 1 — слабкий сигнал.
- Якщо рівень < 1 — дуже слабкий або відсутній сигнал.

21. `NetworksViewModel` — це `ViewModel`, що керує станом екрану, який відображає список доступних бездротових мереж. Анотація `HiltViewModel` вказує на використання `Hilt` для впровадження залежностей.

22. `NetworksState` представляє стан екрану або функціоналу, який працює з Wi-Fi мережами. Цей клас є `data class`, що дозволяє зручно зберігати дані стану та забезпечувати їхню імутабельність. Він успадковується від базового класу `BaseState`, ймовірно, для інтеграції з існуючою архітектурою.

- `isLoadingNetworks` – вказує, чи зараз виконується процес завантаження списку Wi-Fi мереж. Використовується для керування відображенням індикатора завантаження в інтерфейсі.
- `wifiNetworks` – список об'єктів `WifiNetwork`, які представляють знайдені Wi-Fi мережі. Може відображатися у вигляді списку в інтерфейсі користувача.
- `deviceLocation` – місцезнаходження пристрою, представлене у вигляді пари географічних координат: широта (latitude) і довгота (longitude), використовується для прив'язки Wi-Fi мереж до географічної точки.

23. `RecommendationsScreen` – функція, `composable` елемент, який відповідає за відображення екрану рекомендацій. Вона взаємодіє з `ViewModel` для обробки стану та логіки, а також приймає список мереж Wi-Fi і `callback` для навігації.

24. `RecommendationsLayout` – функція, основний компонент, який рендерить макет списку рекомендацій. Вона динамічно відображає список Wi-Fi мереж із можливістю розгортання додаткової інформації для кожної мережі

- state – стан, який містить дані для відображення, включаючи список Wi-Fi мереж.

- onIntent – колбек для обробки подій або намірів, що виникають у макеті.

- rememberedNetworks – збережена копія списку Wi-Fi мереж із state.wifiNetworks, використовує remember для забезпечення реактивного оновлення під час зміни стану.

- expandedSsid – поточний SSID розгорнутої мережі.

25. RecommendationsViewModel — це ViewModel, що керує станом та логікою екрану, який відображає рекомендаційну систему. Анотація HiltViewModel вказує на використання Hilt для впровадження залежностей.

- consumeIntent – метод для обробки намірів (Intent) від UI.

- setAndSortAndClassifyWifiNetworks – функція, що сортує список Wi-Fi мереж та коасифікує мережі залежно від їх характеристик.

26. RecommendationsState – клас, який описує стан для екрану рекомендацій.

27. RecommendationType – використовується для класифікації Wi-Fi мереж за трьома категоріями:

- BEST_CHOICE — Найкращий вибір.

- AVERAGE — Середній варіант.

- WORST_CHOICE — Найгірший варіант.

Функції:

- getCardColor — визначає колір межі картки Wi-Fi мережі залежно від типу рекомендації (зелений, сірий або червоний).

- getRecommendationSignalText – повертає текст для рівня сигналу.

- getRecommendationImageRes – визначає ресурс зображення для типу рекомендації.

3.2 Опис інструкції користувачеві

Додаток для аналізу Wi-Fi мереж підтримує англійську мову та забезпечує інтуїтивно зрозумілу взаємодію. Інтерфейс програми побудований на основі кількох екранів, які користувач може вільно переглядати завдяки зручній навігації. Переходи між екранами виконуються через натискання кнопок, кожна з яких має зрозумілий підпис, що пояснює її функцію.

Програма активно реагує на дії користувача, включно з некоректними. У разі помилок система надає зрозумілі повідомлення у відповідних полях, щоб користувач міг швидко виправити їх. Для операцій, що потребують часу, таких як завантаження даних, передбачено відображення анімованого індикатора (лоадера), що допомагає уникнути непорозумінь під час очікування.

Додаток має сучасний і зручний для використання інтерфейс із такими функціями:

- Сортування Wi-Fi мереж за рівнем сигналу.
- Класифікація мереж на категорії: "Найкращий вибір", "Середній", "Найгірший вибір".
- Інтерактивний дизайн карток для кожної мережі, який дозволяє розкривати додаткову інформацію, як-от частота, відстань до мережі або координати точки доступу.

Вимоги до програмного та апаратного забезпечення

Щоб додаток функціонував належним чином, пристрій повинен відповідати таким вимогам:

- Вимоги до апаратного забезпечення:
 - Обсяг внутрішньої пам'яті: не менше 250 МБ.
 - Обсяг оперативної пам'яті: не менше 250 МБ.
- Вимоги до програмного забезпечення:
 - Операційна система: Android 10.0 або новіша.

Для взаємодії з інтерфейсом користувачу доступно три основних екрани: список та деталі просканованих мереж, карта мереж та екран з рекомендаціями.

Після запуску додатку користувач потрапляє на головний екран із можливістю розпочати сканування Wi-Fi мереж. Для цього достатньо натиснути на кнопку “Start scanning” (рис. 3.1). Додаток одразу запустить процедуру аналізу доступних мереж, що включає збір даних про рівень сигналу, частоту, відстань до точки доступу тощо.

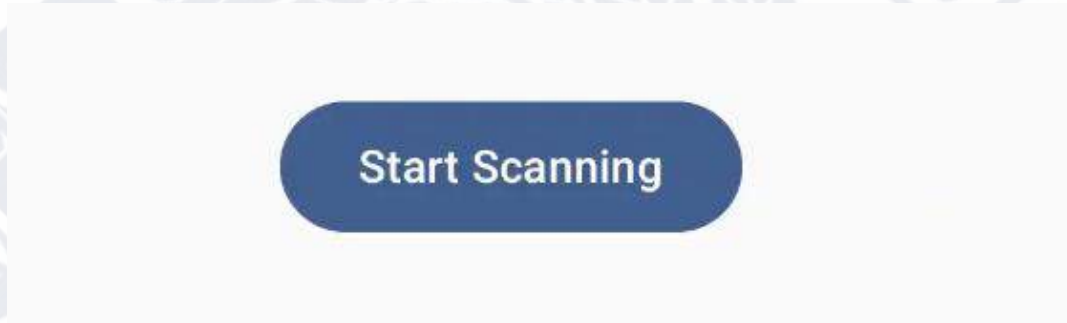


Рисунок 3.1 – Екран з кнопкою Start scanning

Після натискання кнопки “Start scanning” на екрані з’явиться спливаюче вікно з запитом на надання дозволу для доступу до геолокації користувача (рис. 3.2). Це необхідно для коректного визначення місцезнаходження точки доступу та побудови точних рекомендацій.

Для повного функціонування додатку важливо надати доступ саме до точної геолокації. Якщо користувач вибере лише доступ до приблизної локації, функціонал програми буде обмежений. У цьому випадку на екрані з’явиться попередження, що для продовження необхідно надати дозвіл на точне визначення місця розташування (рис. 3.3). Крім того, користувачу буде запропоновано перейти в розділ “Налаштування” свого пристрою, де можна змінити права доступу.



Рисунок 3.2 – Вікно із запитом на геолокацію користувача

У разі відсутності дозволу на геолокацію додаток не зможе:

- Правильно відобразити карту покриття мереж.
- Надати рекомендації, засновані на відстані до точки доступу.
- Забезпечити точне сортування мереж за рівнем сигналу в залежності від фізичної віддаленості.

Таким чином, надання дозволу є ключовим етапом для початку роботи з додатком і забезпечення його повної функціональності. Додаток завчасно інформує про це та допомагає швидко внести необхідні зміни в налаштування.

Permissions have been denied
permanently. Please enable them in
settings to use this app.

Рисунок 3.3 – Екран з помилкою надання дозволу до геолокації

Після надання користувачем дозволу на доступ до геолокації розпочинається процес сканування доступних Wi-Fi мереж. На екрані відображається анімована індикація, яка інформує про виконання цього процесу (рис. 3.4). Анімація створена для того, щоб користувач міг спостерігати за станом виконання завдання та розуміти, що програма працює.

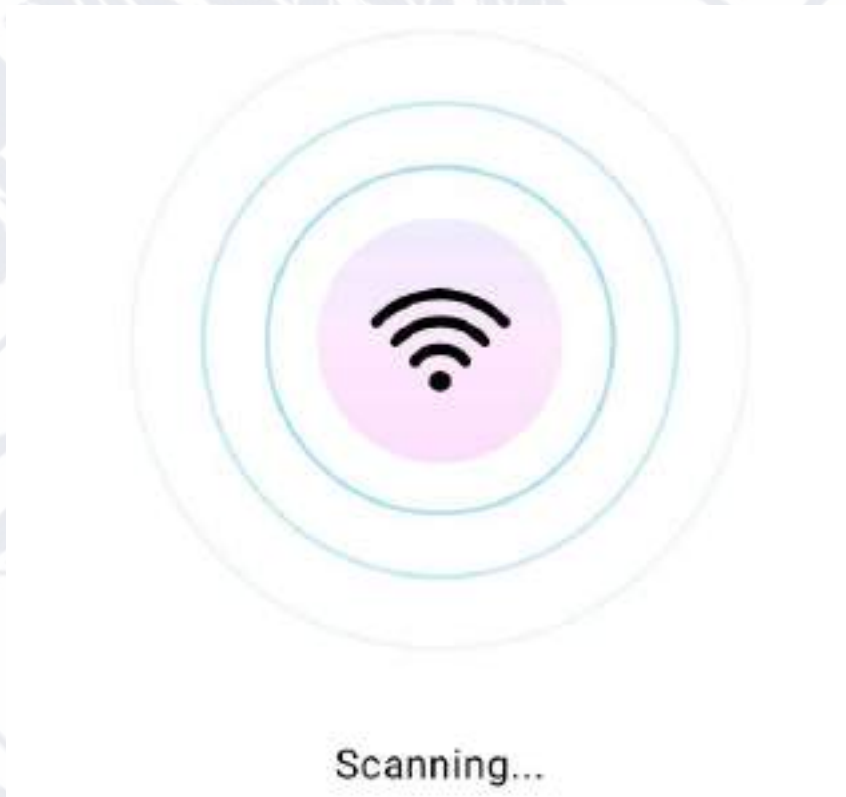
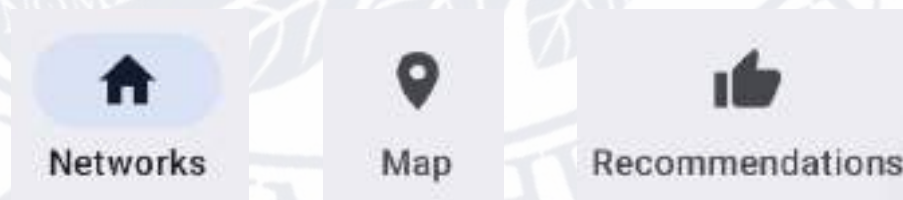


Рисунок 3.4 – Сканування мережі

Після завершення сканування користувач автоматично перенаправляється до основного інтерфейсу додатку, де відкриваються всі можливості для взаємодії з результатами. Для зручності орієнтування між різними розділами програми передбачено панель навігації, яка розташована в нижній частині екрана.



а

б

в

Рисунок 3.5 – Нижня панель навігації

Панель навігації дозволяє швидко переключатися між трьома основними розділами програми: Networks – список знайдених під час сканування мереж із детальною інформацією про кожну з них (рис. 3.5.а); Map – карта, яка демонструє розташування знайдених точок доступу (рис. 3.5.б); Recommendations – список Wi-Fi мереж, які система класифікувала за якістю сигналу та іншими характеристиками (рис. 3.5.в)

Екран Networks у додатку представляє користувачу список усіх доступних Wi-Fi мереж, які були виявлені під час сканування (рис. 3.6). Цей розділ розроблений для зручного перегляду та аналізу інформації про кожну з доступних точок доступу.



Рисунок 3.6 – Екран Networks

Кожна Wi-Fi мережа відображається у вигляді окремої картки, яка містить розширену інформацію, необхідну для оцінки її характеристик. Зокрема, у картці представлена така інформація:

1. Назва мережі (SSID) – це загальнодоступне ім'я, яке дозволяє користувачу ідентифікувати конкретну точку доступу серед інших.
2. Унікальний ідентифікатор точки доступу (BSSID) – це унікальний MAC-адрес пристрою, який забезпечує доступ до цієї мережі. Цей параметр корисний для точнішої ідентифікації мережі у випадках, коли кілька точок доступу мають однакове ім'я.
3. Сила сигналу – визначається у відсотках і показує, наскільки добре мережа доступна у поточній локації користувача.
4. Рівень сигналу (RSSI) – кількісний показник потужності сигналу, що допомагає оцінити якість зв'язку.
5. Частота сигналу – частотний діапазон, у якому працює точка доступу (наприклад, 2.4 ГГц або 5 ГГц). Вказівка частоти дозволяє користувачу зрозуміти, наскільки стабільною може бути мережа залежно від її дальності та перешкод.
6. Протоколи безпеки – інформація про типи шифрування, які підтримує мережа (наприклад, WEP, WPA, WPA2). Це параметр, що дозволяє визначити рівень захищеності підключення.

Окрім текстової інформації, на картці кожної мережі розміщена відповідна іконка, яка візуально відображає рівень сигналу. Це полегшує швидку оцінку якості мережі без необхідності аналізувати числові показники.

Другим екраном, доступним через нижню панель навігації, є Мар (карта мереж). Цей розділ додатку пропонує інтерактивний досвід для користувача, дозволяючи не лише переглядати розташування мереж, але й активно взаємодіяти з елементами карти. Завдяки цьому користувач може легко знаходити потрібні

точки доступу та отримувати інформацію про них у зручному форматі.



Рисунок 3.7 – Екран Мар

Екран Мар поділено на дві секції (рис. 3.7):

1. Список доступних мереж – ця частина екрана містить текстову інформацію про виявлені мережі. У списку представлено:

- Сила сигналу у вигляді іконки, що візуально демонструє якість підключення.
- Назва мережі (SSID) – для зручності користувача в ідентифікації мережі.
- Приблизна відстань до джерела сигналу, що допомагає оцінити, наскільки близько знаходиться точка доступу.

2. Карта з інтерактивними елементами – візуальна частина, що відображає поточне розташування пристрою та мереж із виявленого списку.

- Синім кольором позначено поточну геолокацію пристрою користувача. Це дозволяє орієнтуватися в просторі та розуміти своє місцезнаходження відносно мереж.
- Червоним кольором відмічено точки доступу.

Карта дозволяє користувачу вільно переміщуватися по ній, наближати чи віддаляти зображення для детальнішого перегляду. У разі кліку на червоний маркер, що позначає точку доступу, відкривається спливаюче вікно із детальною інформацією про цю мережу. У ньому зазначено:

- Назву мережі.
- Відстань до точки доступу.

Ця функціональність особливо корисна для аналізу розташування мереж у реальному часі та допомагає користувачам приймати рішення, наприклад, яку мережу обрати для підключення або як покращити своє розташування для кращої якості сигналу.

На екрані Recommendations (рис. 3.8) користувач може побачити результати аналізу якості доступних мереж Wi-Fi. У цьому розділі додаток автоматично оцінює всі доступні мережі, присвоюючи їм одну з трьох категорій якості на основі сили сигналу та інших технічних параметрів.

Категорії якості мереж:

1. Найкращі мережі (Best signal):

- Позначаються зеленим кольором картки.
- Супроводжуються зеленою іконкою, що вказує на позитивний рівень сигналу.
- Відображають текст "Best signal".

2. Середні мережі (Average signal):

- Позначаються сірим кольором картки.

- Не містять додаткових іконок.
 - Відображають текст "Average signal".
3. Погані мережі (Worst signal):
- Позначаються червоним кольором картки.
 - Містять червону іконку, що вказує на низький рівень сигналу.
 - Відображають текст "Worst signal".



Рисунок 3.8 – Екран Recommendations

Картка мережі має два стани – скорочена та повна.

Інформація на картках в скороченому стані:

- Назва мережі: допомагає користувачу ідентифікувати доступні точки доступу.
- Рівень сигналу.

Картка в повному стані дублює параметри з скороченого стану, а також додатково відображає (рис. 3.9):

- Назва мережі (SSID).
- Унікальний ідентифікатор точки доступу (BSSID).
- Сила сигналу.
- Координати (широта та довгота)
- Частота сигналу.
- Дистанція до точки доступу.
- Протоколи безпеки – інформація про типи шифрування, які підтримує мережа (наприклад, WEP, WPA, WPA2). Це параметр, що дозволяє визначити рівень захищеності підключення.



Рисунок 3.9 – Картка в повному вигляді

3.3 Тестування програмного засобу

Процес тестування є невід'ємною частиною розробки програмного забезпечення, адже саме він дозволяє перевірити коректність роботи функціоналу, виявити потенційні помилки та гарантувати стабільність програми для кінцевого користувача. У випадку програмного засобу для аналізу та рекомендацій Wi-Fi мереж, тестування має вирішальне значення, оскільки від надійності програми залежить здатність користувача отримати точну та своєчасну інформацію про доступні мережі, їх якість та параметри.

Для забезпечення високої якості роботи додатку проводилось тестування за наступними критеріями:

1. Функціональність: перевірка коректної роботи основних функцій, таких як сканування Wi-Fi мереж, аналіз сигналу, відображення інформації про мережі та рекомендації.

Таблиця 3.1 – Тестування функціональності.

Тест	Очікуваний результат	Фактичний результат	Результат
Сканування Wi-Fi мереж	Відображення анімації сканування, список доступних мереж після завершення процесу	Відображення анімації сканування, список доступних мереж після завершення процесу	Успіх
Відмова в доступі до локації	Виведення повідомлення про необхідність надати доступ до точної локації та перенаправлення в налаштування	Виведення повідомлення про необхідність надати доступ до точної локації та перенаправлення в налаштування	Успіх
Відображення списку мереж	Коректне відображення назв мереж, рівня сигналу, частоти та протоколів безпеки у вигляді карток	Коректне відображення назв мереж, рівня сигналу, частоти та протоколів безпеки у вигляді карток	Успіх
Навігація між екранами	Перехід між екраном "Networks", "Map" та "Recommendations" за допомогою нижньої панелі навігації	Перехід між екраном "Networks", "Map" та "Recommendations" за допомогою нижньої панелі навігації	Успіх
Карта мереж	Відображення поточної локації пристрою та джерел сигналів, інтерактивна взаємодія з точками на карті	Відображення поточної локації пристрою та джерел сигналів, інтерактивна взаємодія з точками на карті	Успіх

Рекомендації мереж	Відображення статусу мереж (найкраща, середня, погана) із кольоровим маркуванням та відповідними іконками	Відображення статусу мереж (найкраща, середня, погана) із кольоровим маркуванням та відповідними іконками	Успіх
-----------------------	--	--	-------

2. Інтерфейс: тестування зручності взаємодії з користувацьким інтерфейсом, перевірка логіки переходів між екранами та коректного відображення інформації.

Таблиця 3.2 – Тестування інтерфейсу.

Тест	Очікуваний результат	Фактичний результат	Результат
Коректність переходів	Після натискання на іконку "Мар" на панелі навігації відображається екран із картою мереж	Після натискання на іконку "Мар" на панелі навігації відображається екран із картою мереж	Успіх
Відображення екрану "Мар"	Карта завантажується, відображаються поточна локація пристрою та джерела сигналів червоним кольором	Карта завантажується, відображаються поточна локація пристрою та джерела сигналів червоним кольором	Успіх
Коректність списку мереж	Відображення назв мереж, рівня сигналу, частоти, протоколів безпеки та іконок у списку мереж на екрані "Networks"	Відображення назв мереж, рівня сигналу, частоти, протоколів безпеки та іконок у списку мереж на екрані "Networks"	Успіх
Інтерактивність карти	На карті доступна взаємодія: пересування, збільшення, натискання на точку відкриває деталі мережі	На карті доступна взаємодія: пересування, збільшення, натискання на точку відкриває деталі мережі	Успіх
Маркування рекомендацій	Мережі мають кольорові позначки: зелений – найкраща, сірий – середня, червоний – погана	Мережі мають кольорові позначки: зелений – найкраща, сірий – середня, червоний – погана	Успіх

3. Продуктивність: оцінка швидкості виконання операцій сканування та обробки даних, а також оптимізація використання ресурсів пристрою.

Таблиця 3.3 – Тестування продуктивності.

Тест	Очікуваний результат	Фактичний результат	Результат
Швидкість сканування мереж	Сканування завершується за 2–3 секунд при наявності 10 мереж у радіусі дії	Сканування завершується за 2 секунди при наявності 10 мереж у радіусі дії	Успіх
Швидкість обробки результатів	Обробка даних завершується за 1–2 секунди після завершення сканування	Обробка даних завершується менш за 1 секунду після завершення сканування	Успіх
Завантаження карти	Карта з мережами завантажується за 2–4 секунди при середній швидкості інтернет-з'єднання (10 Mbps)	Карта з мережами завантажується за 3 секунди при середній швидкості інтернет-з'єднання (10 Mbps)"	Успіх
Стабільність роботи під навантаженням	Додаток коректно працює при скануванні 50+ мереж та одночасній навігації по екрану "Map"	Додаток коректно працює при скануванні 50+ мереж та одночасній навігації по екрану "Map"	Успіх

У процесі тестування додатка було перевірено його функціональність, продуктивність, стабільність та відповідність вимогам. Усі тестові кейси пройдені успішно, і додаток продемонстрував коректну роботу в різних сценаріях використання. Основні функції, такі як сканування мереж, відображення даних на карті та надання рекомендацій, виконуються відповідно до очікувань. Тестування підтвердило, що додаток готовий до використання кінцевими користувачами.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було розроблено мобільний додаток для аналізу та вибору Wi-Fi мереж, який успішно поєднує функціональність, зручність використання та сучасні технологічні рішення. Робота охоплювала повний цикл створення програмного забезпечення: від проєктування архітектури та розробки основних модулів до тестування готового продукту. Основною метою було спрощення процесу вибору оптимальної Wi-Fi мережі для користувачів, що досягнуто завдяки інтеграції функцій сканування, візуалізації та аналізу даних про мережі.

У рамках виконання кваліфікаційної роботи були успішно реалізовані всі поставлені завдання, спрямовані на розробку мобільного додатку для аналізу та вибору Wi-Fi мереж.

1. Розроблено архітектуру додатку з інтуїтивно зрозумілим інтерфейсом, який забезпечує простоту використання та доступність для кінцевого користувача. Особливу увагу було приділено зручності навігації між основними функціональними екранами, такими як список мереж, інтерактивна карта та рекомендації.

2. Реалізовано функціонал сканування Wi-Fi мереж. Додаток дозволяє отримувати детальну інформацію про доступні мережі, включаючи такі параметри, як сила сигналу, частота, протоколи безпеки та інші характеристики.

3. Розроблено модуль інтерактивної карти, який відображає візуальну інформацію про місцезнаходження мереж. Користувач може взаємодіяти з картою, отримуючи доступ до детальної інформації про конкретну мережу за допомогою простих дій.

4. Впроваджено систему аналізу та рекомендацій, яка забезпечує зручний спосіб вибору оптимальної мережі. Рекомендації базуються на показниках сили сигналу, безпеки та доступності, що дозволяє користувачам легко знайти найкращу мережу для підключення.

5. Спрощено процес вибору мережі завдяки інтеграції аналізу та відображення інформації в єдиному інтерфейсі, що відповідає потребам сучасних користувачів і підвищує їх задоволеність роботою з додатком.

6. Проведено комплексне тестування програмного забезпечення, яке підтвердило його надійність, ефективність і коректність роботи в різних умовах. Тестування включало як перевірку функціоналу, так і оцінку взаємодії з інтерфейсом.

Таким чином, виконання роботи дозволило створити сучасний, функціональний та зручний мобільний додаток, який допомагає користувачам швидко та легко вибрати найкращу Wi-Fi мережу, підвищуючи якість підключення до інтернету. Отримані результати можуть бути використані для подальшого вдосконалення додатку та розширення його функціональних можливостей.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ТА ЛІТЕРАТУРИ

1. WLAN та Wi-Fi [Електронний ресурс]. – Режим доступу: <https://wiki.cusu.edu.ua/index.php/Wi-FiWlan>
2. Бездротові технології [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/wirelessTechnologies>
3. Б.Ю. Жураковський, І.О. Зенів. Комп'ютерні мережі. Частина 2 – Київю КПІ ім. Ігоря Сікорського. 2020 – 372 с.
4. Ключове обладнання для побудови бездротових мереж. [Електронний ресурс]. – Режим доступу: <https://wiki.cusu.edu.ua/index.php/devices>
5. Що впливає на роботу бездротових мереж Wi-Fi [Електронний ресурс] Режим доступу: <https://keenetic.biz/whatAffectsWirelessNetworks>
6. SSID [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/SSID>
7. Мережева карта [Електронний ресурс] Режим доступу: <https://kovelpost.com/blogs/110>
8. Комутатор [Електронний ресурс] Режим доступу: <https://ntools.com.ua/uk/information/faq/chto-takoe-kommutator-switch>
9. Ретранслятор [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/Ретранслятор>
10. Мережевий міст [Електронний ресурс] Режим доступу: <https://kovelpost.com/blogs/112>
11. Маршрутизатор [Електронний ресурс] Режим доступу: <https://uk.wikipedia.org/wiki/Маршрутизатор>
12. Основні функції маршрутизаторів [Електронний ресурс] Режим доступу: https://web.posibnyky.vntu.edu.ua/fitki/3yarovijk_komp_merezhi/2.5.1.html
13. Network Address Translation [Електронний ресурс] Режим доступу: <https://www.vmware.com/topics/network-address-translation>

14. Quality of Service [Електронний ресурс] Режим доступу: https://en.wikipedia.org/wiki/Quality_of_service
15. Класифікація маршрутизаторів [Електронний ресурс] Режим доступу: <https://lookhouse.com.ua/klasifikatsiya-marshrutizatoriv-yakiy-kupiti-dlya-domashnogo-vikoristannya/>
16. Що таке точка доступу та її функції [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/wirelessAccessPoint>
17. Функції точки доступу [Електронний ресурс]. – Режим доступу: <https://www.cisco.com/c/en/us/solutions/small-business/resource-center/networking/what-is-access-point.html>
18. Типи мереж за радіусом дії. [Електронний ресурс]. – Режим доступу: <https://wiki.cusu.edu.ua/index.ph/types>
19. ZigBee. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/ZigBee>
20. Bluetooth. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Bluetooth>
21. Wi-Fi. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Wi-Fi>
22. WiMax. [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/WiMAX>
23. Wi-Fi мережа. [Електронний ресурс]. – Режим доступу: <https://nashprostir.com/wi-fi-merezha>
24. Стандарти Wi-Fi. [Електронний ресурс]. – Режим доступу: <https://wiki.cusu.edu.ua/index.php/wifiStandarts>
25. Порівняння стандартів Wi-Fi. [Електронний ресурс]. – Режим доступу: <https://netwave.ua/vse-shcho-potribno-znaty-pro-bezdrotovi-merezhi-wlan-pobudova-bezpeka-ta-keruvannya/>

26. ООП. [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/OOP>
27. Android. [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/Android>
28. Історія та переваги Android. [Електронний ресурс]. – Режим доступу:
<https://gsmhub.com.ua/glossary/androyid>
29. Kotlin. [Електронний ресурс]. – Режим доступу:
<https://uk.wikipedia.org/wiki/Kotlin>
30. Mastering the Java Virtual Machine: An in-depth guide to JVM internals and performance optimization – Packt Publishing Ltd. 2024 – 234 с.
31. Kotlin features [Електронний ресурс]. – Режим доступу:
<https://kotlinlang.org/docs/comparison-to-java.html>
32. Null safety [Електронний ресурс]. – Режим доступу:
<https://kotlinlang.org/docs/null-safety.html>
33. Наслідування в Kotlin [Електронний ресурс]. – Режим доступу:
<https://kotlinlang.org/docs/inheritance.html>
34. Kotlin Coroutines [Електронний ресурс]. – Режим доступу:
<https://kotlinlang.org/docs/coroutines-overview.html>
35. Data class in Kotlin [Електронний ресурс]. – Режим доступу:
<https://kotlinlang.org/docs/data-classes.html>
36. Деструктуризація в Kotlin [Електронний ресурс]. – Режим доступу:
<https://kotlinlang.org/docs/destructuring-declarations.html>
37. Функції розширення Kotlin [Електронний ресурс]. – Режим доступу:
<https://kotlinlang.org/docs/extensions.html>
38. R.C. Martin. Clean Architecture – Prentice Hall. 2018 – 420 с.
39. R.C. Martin. Clean Architecture: A Craftsman's Guide to Software Structure and Design – Pearson. 2017 – 432 с.

40. MVVM. [Электронный ресурс]. – Режим доступа: <https://www.raywenderlich.com/34-design-patterns-by-tutorials-mvvm>
41. Guide to Android app modularization. [Электронный ресурс]. – Режим доступа: <https://developer.android.com/topic/modularization>
42. REST API. [Электронный ресурс]. – Режим доступа: <https://www.redhat.com/en/topics/api/what-is-a-rest-api>
43. M. Masse. REST API Design Rulebook – O'Reilly Media. 2011 – 112 с.
44. Jetpack Compose [Электронный ресурс] – Режим доступа: <https://developer.android.com/jetpack/compose>
45. Kotlin for Jetpack Compose [Электронный ресурс] – Режим доступа: <https://developer.android.com/develop/ui/compose/kotlin>
46. Android Studio [Электронный ресурс] – Режим доступа: <https://qagroup.com.ua/publications/android-studio-perevagy-ta-osoblyvosti/>
47. IntelliJ IDEA [Электронный ресурс] – Режим доступа: https://en.wikipedia.org/wiki/IntelliJ_IDEA
48. GIT [Электронный ресурс] – Режим доступа: <https://git-scm.com/>
49. Emulator [Электронный ресурс] – Режим доступа: <https://developer.android.com/studio/run/emulator>
50. Logcat [Электронный ресурс] – Режим доступа: <https://developer.android.com/tools/logcat>
51. Debugger [Электронный ресурс] – Режим доступа: <https://developer.android.com/studio/debug>
52. Layout Inspector [Электронный ресурс] – Режим доступа: <https://developer.android.com/studio/debug/layout-inspector>
53. Performance Profile [Электронный ресурс] – Режим доступа: <https://developer.android.com/studio/profile>
54. WiFi Analyzer [Электронный ресурс] – Режим доступа: <https://vremsoftwaredevelopment.github.io/WiFiAnalyzer/>

55. NetSpot [Електронний ресурс] – Режим доступу:
<https://www.netspotapp.com/>

56. Wi-Fi Mapping Apps [Електронний ресурс] – Режим доступу:
<https://bg.androidhow.eu/wifiMappingApps>



ДЕКЛАРАЦІЯ

про дотримання академічної доброчесності

Я, Чернега Вікторія Михайлівна, студентка, ОП Комп'ютерні технології обробки даних, тема: «Мобільний додаток для побудови карти wіfі мереж з вбудованою рекомендаційною системою для покращення роботи обраної мережі», що нижче підписалась, розуміючи та підтримуючи загальновизнані засади справедливості, доброчесності та законності,

ЗОБОВ'ЯЗУЮСЬ:

дотримуватися принципів та правил академічної доброчесності, що визначені законодавством України, локальними нормативними актами Донецького національного університету імені Василя Стуса, положеннями, правилами, умовами, визначеними іншими суб'єктами, та не допускати їх порушення.

ПІДТВЕРДЖУЮ:

що мені відомі положення статті 42 Закону України «Про освіту»;

що у даній роботі не представляла чийсь роботи повністю або частково як свої власні. Там, де я скористалася працею інших, я зробила відповідні посилання на джерела інформації;

що дана робота не передавалась іншим особам і подається вперше, не порушує авторських та суміжних прав закріплених статтями 21-25 Закону України «Про авторське право та суміжні права», а дані та інформація не отримувались в недозволений спосіб.

УСВІДОМЛЮЮ:

що ця робота може бути перевірена університетом на плагіат або інші порушення академічної доброчесності, в тому числі з використанням спеціалізованих сервісів;

що у разі порушення академічної доброчесності, до мене можуть бути застосовані процедури, передбачені законодавством України та Кодексом академічної доброчесності та корпоративної етики Донецького національного університету імені Василя Стуса, іншими локальними нормативними актами університету, та я можу бути притягнута до академічної відповідальності.

(дата)

(підпис)