

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет комп'ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Мобільний застосунок для локальної
навігації на основі Bluetooth-маячків

(тема)

Виконав:

здобувач 4 року навчання,

групи КІУКІ-21-6

В'ячеслав БУРЯК

(власне ім'я, прізвище)

Спеціальність

123 «Комп'ютерна інженерія»

(код і повна назва спеціальності)

Тип програми освітньо-професійна

(освітньо-професійна або освітньо-наукова)

Освітня програма

Комп'ютерна інженерія

(повна назва освітньої програми)

Керівник: ас. Владислав ХОЛЄВ

(посада, власне ім'я, прізвище)

Допускається до захисту

Завідувач кафедри ЕОМ

(підпис)

Андрій КОВАЛЕНКО

(власне ім'я, прізвище)

2025 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерної інженерії та управління _____

Кафедра _____ електронних обчислювальних машин _____

Рівень вищої освіти _____ перший (бакалаврський) _____

Спеціальність _____ 123 «Комп'ютерна інженерія» _____
(код і повна назва)

Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Освітня програма _____ Комп'ютерна інженерія _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

“ _____ ” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

здобувачеві _____ Буряку В'ячеславу Андрійовичу _____
(прізвище, ім'я, по батькові)

1. Тема роботи _____ Мобільний застосунок для локальної навігації на основі
Bluetooth-маячків _____

затверджена наказом по університету від “ 26 ” травня 2025 р. № 424Ст

2. Термін подання здобувачем роботи до екзаменаційної комісії _____ 17 червня 2025 р.

3. Вхідні дані до роботи _____

- _____ - документація Dart, Flutter, Teltonika, Android SDK
- _____ - середовище розробки Android Studio
- _____ - Bluetooth маячки

4. Перелік питань, що потрібно опрацювати у роботі _____

- _____ 1) аналіз предметної області;
- _____ 2) аналіз використовуваних технологій;
- _____ 3) програмна реалізація;
- _____ 4) інструкція користувача;
- _____ 5) висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій 16 слайдів презентації

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

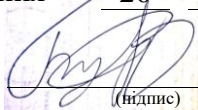
Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН


№	Назва етапів роботи	Строк / терміни виконання етапів роботи	Примітка
1	Вивчення проблемної області та постановка задачі	27.05.25-30.05.25	
2	Розробка технічного завдання	31.05.25-02.06.25	
3	Формування та навчання моделі системи	03.06.25-05.06.25	
4	Розробка програмної частини	06.06.25-09.06.25	
5	Тестування системи	10.06.25	
6	Оформлення матеріалів кваліфікаційної роботи	11.06.25	
7	Подання кваліфікаційної роботи керівникові та її попередній захист	12.06.25-13.06.25	
8	Подання кваліфікаційної роботи на рецензування	14.06.25-16.06.25	

Дата видачі завдання “ 26 ” травня 2025 р.

Здобувач


(підпис)

Керівник роботи


(підпис)

ас. Владислав ХОЛЄВ

(посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 83 с., 20рис., 7 табл., 1 дод., 17 джерел.

BLE-НАВІГАЦІЯ, МОБІЛЬНИЙ ЗАСТОСУНОК, FLUTTER, DART, ANDROID, ТЕХТ-ТО-SPEECH, SPEECH-TO-ТЕХТ, ЛОКАЛЬНЕ ПОЗИЦІОНУВАННЯ, ЦИФРОВА КАРТА ПРИМІЩЕННЯ.

Метою кваліфікаційної роботи є створення мобільного застосунку для локальної навігації в приміщеннях із застосуванням технології BLE. Розроблена система забезпечує можливість побудови маршруту всередині приміщення та озвучення інструкцій для користувача, що сприяє підвищенню доступності навігації для людей з порушеннями зору та в незнайомих середовищах.

Основні функціональні можливості застосунку включають: обробку голосових команд за допомогою Speech-to-Text, побудову маршруту між точками приміщення на основі цифрової карти у форматі JSON, а також відображення маршруту на інтерактивній карті. Архітектура системи передбачає можливість подальшої інтеграції BLE-маячків для позиціонування користувача в реальному часі.

У ході виконання кваліфікаційної роботи було проаналізовано сучасні підходи до реалізації систем внутрішньої навігації, визначено вимоги до мобільного застосунку та реалізовано його прототип на основі Flutter і Dart для платформи Android.

ABSTRACT

Bachelor's thesis: 83 pages, 20 figures, 7 tables, 1 appendices, 17 sources.

BLE-NAVIGATION, MOBILE APPLICATION, FLUTTER, DART, ANDROID, TEXT-TO-SPEECH, SPEECH-TO-TEXT, LOCAL POSITIONING, DIGITAL MAP OF THE ROOM.

The major goal of this thesis is to create a mobile application for local indoor navigation using BLE technology. The developed system provides the ability to build an indoor route and voice instructions for the user, which helps to increase the accessibility of navigation for people with visual impairments and in unfamiliar environments.

The main functionalities of the application include: processing voice commands using Speech-to-Text, building a route between points in the room based on a digital map in JSON format, and displaying the route on an interactive map. The architecture of the system provides for the possibility of further integration of BLE beacons for real-time user positioning.

In the course of the qualification work, we analyzed modern approaches to the implementation of internal navigation systems, defined the requirements for a mobile application, and implemented its prototype based on Flutter and Dart for the Android platform.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	8
ВСТУП	9
1 ОГЛЯД ПРОБЛЕМНОЇ ОБЛАСТІ	11
1.1 Визначення проблемної області	11
1.2 Актуальність дослідження	15
1.3 Завдання, які вирішує розроблювана система BLE-навігації.....	18
1.4 Мета та задачі дослідження	22
2 ОБҐРУНТУВАННЯ ТА МЕТОДОЛОГІЧНЕ ПІДҐРУНТЯ ДОСЛІДЖЕННЯ	23
2.1 Порівняльний аналіз технологій RTLS.....	23
2.2 Вибір технологічної бази для рішення поставлених задач.....	26
2.2.1 Переваги для локальної навігації	30
3 ФОРМАЛЬНА ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ РІШЕННЯ	31
3.1 Опис загальної побудови системи BLE-навігації. Взаємодія основних компонентів	31
3.2 Структурна організація програмних модулів	32
3.2.1 Модуль взаємодії з BLE-маячками. BLE-маячок ELA Blue COIN ID Beacon.....	34
3.2.2 Взаємодія Bluetooth-маячків із інтерактивними картами	36
3.2.3 Алгоритм побудови маршруту	39
3.2.4 Модуль голосової взаємодії (розпізнавання команд, озвучення інструкцій)	41
3.2.5 Словник ключових фраз для інтерпретації запитів	48
3.3 Технологічне середовище реалізації	49
3.3.1 Використання бібліотек flutter_blue, speech_to_text, flutter_tts, json_serializable.....	50
3.3.3 Реалізація офлайн-режиму роботи	54

3.4 Реалізація цифрової мапи приміщення.....	55
3.5 Побудова прототипу системи та сценарії тестування.....	56
3.5.1 Тестування в умовах квартири	57
4 ІНСТРУКЦІЯ КОРИСТУВАЧА ТА ДЕМОНСТРАЦІЯ РОБОТИ	60
4.1 Комплектація системи	60
4.2 Підготовка до роботи.....	61
4.3 Покрокове користування системою	62
4.4 Демонстрація типового сценарію використанн	66
4.5 Обмеження системи та технічні застереження	68
4.6 Перспективи подальшого розвитку та розширення системи	69
ВИСНОВКИ.....	71
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	73
ДОДАТОК А Графічний матеріал кваліфікаційної роботи.....	75

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

API – Application Programming Interface (інтерфейс програмування застосунків)

AR – Augmented Reality (доповнена реальність)

BFS – Breadth-First Search (перший пошук по ширині)

BLE – Bluetooth Low Energy (енергоєфективна технологія Bluetooth)

GPS – Global Positioning System (глобальна система позиціонування)

IDE – Integrated Development Environment (інтегроване середовище розробки)

JSON – JavaScript Object Notation (формат обміну даними)

LIDAR – Light Detection and Ranging (система лазерного сканування)

NFC – Near Field Communication (технологія ближнього бездротового зв'язку)

NLP – Natural Language Processing (обробка природної мови)

OTAP – Over-The-Air Programming (оновлення прошивки по повітрю)

RTLS – Real-Time Location System (система позиціонування в реальному часі)

RSSI – Received Signal Strength Indicator (індикатор сили прийнятого сигналу)

TTS – Text-to-Speech (синтез мовлення)

UWB – Ultra-Wideband (надширокосмугова технологія)

Wi-Fi – Wireless Fidelity (бездротова локальна мережа)

ВСТУП

Порушення органів зору призводить до зниження якості життя, особливо у великих містах із ускладненою навігацією, зниження незалежності та самостійності людини, а також до зниження рівня загальної безпеки. Сучасний світ все більше взаємопов'язаний з цифровими технологіями, які спрямовані на підвищення доступності та зручності в різних сферах життя. Одним з ключових напрямків є розробка навігаційних систем, які дозволяють людям орієнтуватися в складних середовищах, таких як торгові центри, офіси, лікарні тощо. Такі рішення особливо важливі для людей з вадами зору, які стикаються з низкою труднощів при пересуванні великими просторами. Традиційні методи орієнтування, такі як допомога персоналу, друковані схеми або тактильні плитки, часто виявляються неефективними, фрагментарними або недоступними у великих будівлях, таких як торговельні центри, лікарні чи вокзали. Це обмежує мобільність, безпеку та самостійність таких користувачів у повсякденному житті. Відсутність зручних і доступних систем навігації в приміщеннях створює бар'єри для соціальної інтеграції та знижує якість життя. Традиційні рішення, такі як друковані карти або допомога незнайомих людей, не завжди ефективні.

У цьому контексті використання технології Bluetooth-маячків, інтерактивних карт та голосових команд відкриває нові можливості для створення інклюзивного середовища. Одним із перспективних рішень цієї проблеми є впровадження технологій Bluetooth Low Energy (BLE) у поєднанні з мобільними застосунками. BLE-маячки здатні забезпечити точне позиціонування всередині приміщень, а інтеграція з кросплатформним застосунком дозволяє реалізувати персоналізовану систему навігації, яка видає голосові та зорові інструкції в реальному часі. Такий підхід дозволяє не тільки відображати розташування користувача на цифровій мапі, а й

динамічно прокладати маршрути до потрібної точки з урахуванням змін середовища.

Завдяки використанню BLE-навігації у поєднанні з голосовими інтерфейсами, внутрішня орієнтація стає доступнішою, автономнішою та менш залежною від людського фактору. Подібні технології створюють передумови для цифрової інклюзії маломобільних груп населення та можуть бути масштабовані на різні типи приміщень – від офісів до великих інфраструктурних об'єктів.

Методи локального позиціонування на основі BLE та реалізація голосових підказок у мобільному застосунку займають центральне місце в створенні інтелектуальної навігаційної системи. Такий підхід дозволяє не тільки підвищити безпеку та комфорт пересування користувачів з вадами зору, а й відкриває нові можливості для адаптивного управління маршрутами, збору аналітики та подальшої оптимізації навігаційного середовища.

Для реалізації застосунку було обрано кросплатформенну технологію Flutter, яка забезпечує ефективну розробку під операційні системи Android та iOS з використанням єдиної кодової бази. Flutter, написаний на мові програмування Dart, дозволяє створювати адаптивний та інтуїтивно зрозумілий інтерфейс, а також легко взаємодіяти з апаратними компонентами пристрою, зокрема з Bluetooth-модулем. Це значно скорочує час розробки, підвищує продуктивність та забезпечує однакову поведінку застосунку на різних мобільних пристроях.

Архітектура системи передбачає взаємодію трьох ключових компонентів: BLE-маячків, мобільного застосунку та користувача. Маячки покращують локальну навігацію. Мобільний застосунок аналізує інтенсивність сигналу (RSSI), визначає положення користувача на цифровій мапі та генерує голосові та візуальні інструкції для переміщення до обраної точки.

1 ОГЛЯД ПРОБЛЕМНОЇ ОБЛАСТІ

1.1 Визначення проблемної області

Люди з вадами зору стикаються зі значними труднощами в орієнтуванні у великих приміщеннях, таких як торгові центри, лікарні, вокзали чи офіси. Відсутність доступних та ефективних засобів навігації в приміщенні обмежує їхню мобільність, незалежність та безпеку. Традиційні рішення (наприклад, допомога персоналу, статичні схеми або друківані карти) не є достатньо зручними та комплексними.

Поява технологій визначення місцезнаходження в реальному часі (RTLS), зокрема на основі Bluetooth Low Energy (BLE), відкриває нові можливості для створення доступних систем внутрішньої навігації. Системи з BLE-маячками здатні забезпечити точне позиціонування користувача всередині будівлі, інтегруватися з мобільним додатком і давати голосові інструкції, що особливо важливо для людей з вадами зору.

Аналізуючи таблицю 1.1, можна зробити висновок, що традиційні способи орієнтування в громадському та приватному просторі лише частково задовольняють потреби людей з порушеннями зору. Середовище, в якому пересувається користувач, безпосередньо впливає на якість та безпеку його пересування. Слід зазначити, що існуючі рішення здебільшого орієнтовані на людей без порушень зору. У торгових центрах, лікарнях, офісах і вокзалах основна інформація представлена візуально - вивісками, стендами або екранами. Для людей з вадами зору ці заходи практично недоступні без сторонньої допомоги. Також, залежність від персоналу є системною проблемою. Замість автономної навігації користувач змушений щоразу звертатися до персоналу установи. Це створює психологічний бар'єр, знижує автономність людини та створює навантаження на персонал.

Таблиця 1.1 - Традиційні способи орієнтування в громадському та приватному просторі

Середовище	Основні засоби навігації	Переваги	Обмеження/Недоліки
1	2	3	4
метро	персонал звукові оголошення тактильні плити	надійність постійна присутність	людський фактор залежність від персоналу обмежена автономність швидкість пересування людність
вулиця	тактильна плитка світлофори з сигналами тростина пес повадир	пасивна навігація, незалежність	обмежено в покритті не працює в складних умовах (дощ, сніг)
торговельний центр	– охоронці інформаційні стійки друковані схеми	доступність у великих тц	неадаптовано для людей з вадами зору відсутність голосового супроводу наявно лиш в найпопулярніших тц
офісні будівлі	вивіски таблички персонал	простота для бачачих	недоступність для незрячих немає динамічного маршрутування загальна заплутаність локацій

Продовження таблиці 1.1

1	2	3	4
лікарні	черги номери кабінетів запит персоналу	часто передбачувано	труднощі з орієнтацією без супроводу або зору
залізничні вокзали	табло оголошення допомога персоналу	масовість, звукові повідомлення	погана адаптація для сліпих складність при навантаженні

Стаціонарні рішення (друковані карти, вказівники, інформаційні кіоски) не забезпечують персоналізацію маршруту. Користувач не отримує вказівки в реальному часі, не бачить зміни маршруту в разі виникнення перешкод, а системи не підлаштовуються під потреби користувача.

Існуючі зовнішні рішення (тактильна плитка, звукова сигналізація), хоч і підвищують доступність, але не охоплюють внутрішній простір. Найбільші труднощі виникають всередині складних об'єктів (торгові центри, вокзали, лікарні), де такі системи не працюють або відсутні.

Аналізуючи таблицю 1.2, можна стверджувати, що інтеграція BLE-навігації створює гнучку, точну та голосову систему підтримки. Це значно підвищує рівень автономності, безпеки та комфорту для людей з вадами зору, оскільки система діє персонально, незалежно від інших людей та адаптується до умов простору.

Проблемна область проєкту охоплює:

- створення точних систем позиціонування у приміщенні;
- забезпечення голосової взаємодії користувача з додатком;
- інтеграція BLE-маячків із цифровими картами;
- адаптація під потреби користувачів з інвалідністю.

Таблиця 1.2 - Альтернативні способи орієнтування в громадському та приватному просторі

Середовище	Основні засоби навігації	Переваги	Обмеження/Недоліки
Навігаційні системи	GPS-навігатори, мобільні карти (Google Maps тощо)	Висока автономність, голосовий супровід	Не працюють в приміщеннях, неточне позиціонування
BLE-навігація (пропозиція)	Маячки + мобільний застосунок + голосовий супровід	Автономність, точність, голосова взаємодія	Потребує налаштування інфраструктури (маячки, карта)

Аналіз показує, що сучасні системи орієнтування у громадських і приватних просторах залишаються фрагментарними та не універсальними. Більшість рішень зосереджені або на зовнішньому середовищі (тактильна плитка, світлофори), або на людському факторі (персонал, охорона), що не забезпечує сталу автономію для людей із порушеннями зору.

Технології внутрішньої навігації на основі BLE (Bluetooth Low Energy) дозволяють створити гнучку, персоналізовану систему супроводу, що враховує індивідуальні потреби користувача, змінює маршрут у реальному часі та забезпечує голосовий інтерфейс, необхідний людям з вадами зору.

Як зазначено у звітах Європейської асоціації доступності (ЕАА), що внутрішня орієнтація – це одне з найменш вирішених питань для людей з інвалідністю в міських умовах. [1]. При цьому технології на базі BLE демонструють високу точність позиціонування у приміщенні (до 1–2 метрів), що перевершує традиційні GPS-рішення [2].

Також, відповідно до дослідження, опублікованого в Journal of Ambient Intelligence and Humanized Computing, BLE-маячки у поєднанні з мобільними

застосунками дозволяють реалізувати сценарії «розумного простору» для людей з обмеженою мобільністю, включаючи навігацію в лікарнях, університетах та вокзалах [3].

1.2 Актуальність дослідження

Сьогодні проблема доступності внутрішніх приміщень для людей з порушеннями зору, слуху, моторики або когнітивних функцій стає дедалі актуальнішою в контексті урбаністики, інклюзії та цифровізації громадських просторів. У великих будівлях – торгових центрах, вокзалах, лікарнях, офісах – орієнтування для маломобільних груп населення є надзвичайно ускладненим. Особливо гостро ця проблема проявляється серед людей із вадами зору, яким необхідні голосові або тактильні навігаційні підказки, що не потребують зорового сприйняття.

У контексті повномасштабної війни в Україні, за даними Мінсоцполітики, лише за 2023 рік понад 58 000 осіб звернулись по допоміжні засоби реабілітації, з них понад 20 000 – протезовані [4]. Це вказує на стрімке зростання кількості людей, які потребують спеціального підходу до просторової орієнтації в побуті, лікарнях, укриттях, центрах допомоги тощо.

У світі існує декілька мобільних рішень для навігації незрячих, зокрема:

- Lazarillo App (Чилі) – поєднує GPS та голосову навігацію на відкритому повітрі.
- RightHear (Ізраїль) – BLE-маячки + indoor-навігація для підприємств.
- Seeing AI (Microsoft) – більше фокусується на розпізнаванні об'єктів, ніж на навігації.

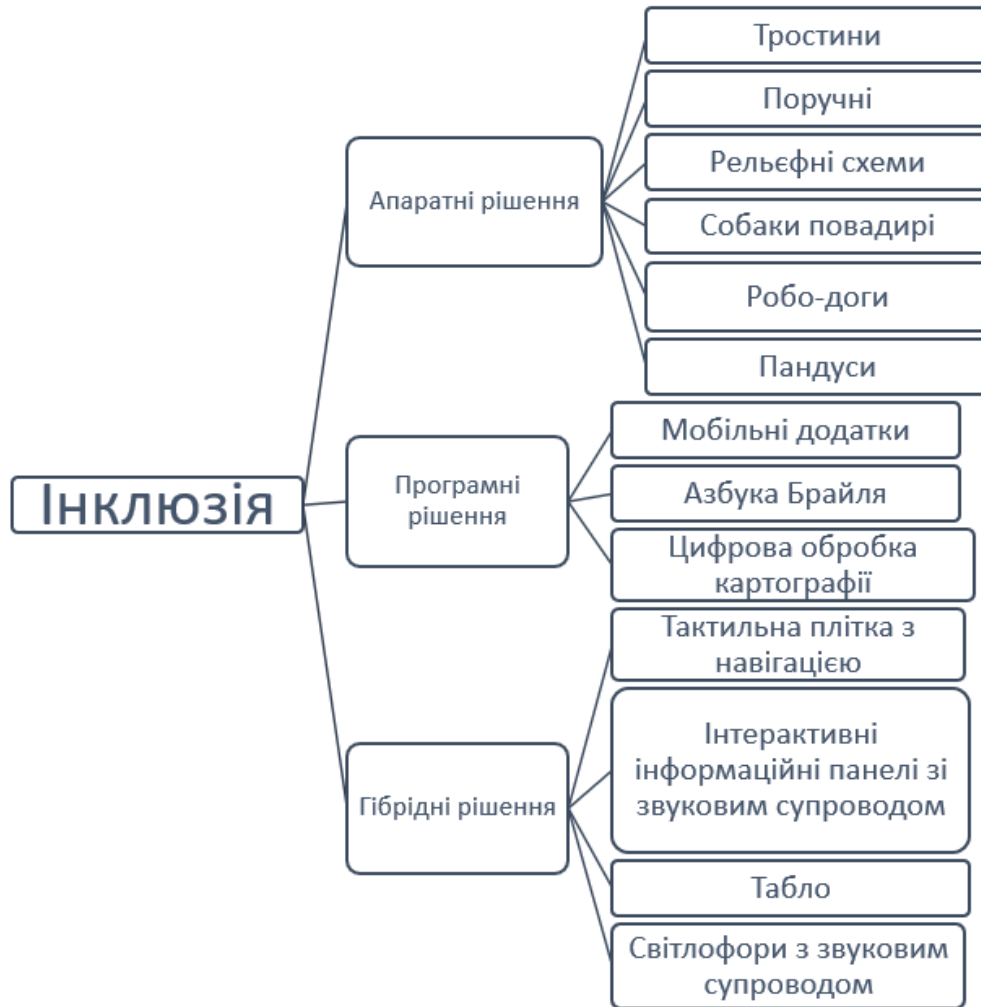


Рисунок 1.1 – Класифікація практичних рішень питань інклюзії

Незважаючи на стрімкий розвиток технологій у сфері інклюзії (Рисунок 1.1), сьогоднішня ситуація залишається фрагментованою й далекою від ідеалу. На ринку представлено чимало апаратних рішень – від звичайних тростин до сучасних роботизованих собак-поводирів. Паралельно активно розвиваються програмні засоби, такі як мобільні навігаційні додатки, голосові помічники, системи озвучування простору. Окрему нішу займають гібридні рішення, які поєднують фізичні пристрої з програмною логікою – наприклад, інтерактивні тактильні мапи або інформаційні панелі зі звуковим супроводом.

Проте в умовах реального використання ці засоби часто залишаються розрізненими, непристосованими до конкретних приміщень або просто

недоступними для широкого кола користувачів. У багатьох випадках особи з вадами зору, опинившись у новому незнайомому просторі – торговому центрі, лікарні, офісі чи укритті – не мають жодного доступного способу автономно орієнтуватися в ньому.

Саме тому виникає потреба у простому, доступному та автономному рішенні, яке не вимагало б складної інфраструктури чи підключення до інтернету. Розроблюваний у межах цієї роботи мобільний застосунок на основі BLE-маячків покликаний закрити цю прогалину. Він забезпечує локальну навігацію в приміщенні через голосовий інтерфейс, дозволяє користувачу визначити своє положення, задати точку призначення й отримати покрокову голосову інструкцію. Система працює офлайн, легко налаштовується, масштабована й адаптована під індивідуальні потреби. У перспективі її можна інтегрувати з іншими інклюзивними рішеннями, зокрема з тактильними або аудіовізуальними елементами інфраструктури.

Ці рішення або не підтримують внутрішнє позиціонування без Інтернету, або є надто складними в інтеграції на рівні побутового використання. Водночас в українських застосунках (наприклад, «Київ Цифровий», «Доступно.UA») брак підтримки BLE-навігації в закритих приміщеннях є очевидним.

У цьому контексті створення офлайн-застосунку на Flutter, який використовує BLE-маячки для локального позиціонування та голосової навігації, є інноваційним і необхідним кроком у розвитку цифрової доступності. Flutter обрано завдяки його кросплатформенності, гнучкості та зручності роботи з Bluetooth-інтерфейсами.

Запропонована система дозволяє:

- створювати цифрові карти будь-якого приміщення (квартира, офіс, ТЦ);
- додавати маячки з індивідуальними ідентифікаторами;
- визначати приблизне місце розташування користувача;
- прокладати маршрут до точки призначення;
- видавати голосові інструкції з урахуванням ключових слів з фрази

(завдяки NLP-обробці на основі ШІ-моделі).

На першому етапі дослідження система буде випробована в межах приватного житлового простору, що дозволяє провести експерименти в контрольованому середовищі. Надалі архітектура може масштабуватися на великі об'єкти з розширеною мережею маячків та підтримкою декількох мов.

Таким чином, тема роботи є актуальною з технічної, соціальної та гуманітарної точок зору, оскільки поєднує:

- нові підходи до indoor-навігації;
- створення адаптивних технологій інклюзії;
- використання сучасних фреймворків, які дозволяють досягти швидкого розгортання рішень у реальних умовах.

Варто зазначити, що в Україні існували спроби створення цифрових рішень, пов'язаних із доступністю. Зокрема, ініціатива “Турбодоступність”, яка позиціонувалась як онлайн-ресурс для бізнесів щодо безбар'єрного облаштування, на момент написання роботи є недоступною для користувачів, що свідчить про її короткотривалу дієвість. Інша платформа – “Доступно.UA”, яка раніше мала власний мобільний додаток для навігації по доступних локаціях, фактично припинила роботу – застосунок більше не оновлюється та не виконує свої функції.

Таким чином, в інформаційному просторі України відсутні дієві, активні цифрові сервіси, які надають можливість автономної навігації для людей з порушеннями зору в закритих приміщеннях. Це створює очевидний вакуум, який і має заповнити запропонований мобільний застосунок на базі BLE-навігації.

1.3 Завдання, які вирішує розроблювана система BLE-навігації

На сьогоднішній день зростає потреба в автономній навігації в закритих приміщеннях для широкого спектра користувачів: людей з вадами зору, маломобільних груп, людей похилого віку, відвідувачів нових або

складних за структурою будівель. Запропонована система BLE-навігації спрямована на розв'язання цієї проблеми за допомогою простого, офлайн-застосунку з голосовими інструкціями та локальним позиціонуванням.

Мобільний застосунок взаємодіє з Bluetooth-маячками, розміщеними в межах приміщення. За голосовим запитом користувача система визначає найближчий маршрут до потрібної точки, прокладає його на цифровій мапі (яка на початковому етапі створюється вручну) та видає голосові або візуальні підказки щодо напрямку руху. Під час переміщення клієнтський пристрій постійно оновлює позицію користувача, орієнтуючись на сигнал від маячків, і за потреби корегує маршрут (таблиця 1.3).

Таблиця 1.3 - Основні завдання, які вирішує система

№	Задача	Механізм реалізації	Компонент системи	Очікуваний результат
1	2	3	4	5
1	Створення цифрової карти приміщення	Формування JSON-файлу з вузлами та з'єднаннями (ручне створення мапи)	Модуль завантаження та зберігання карти приміщення	Карта готова для використання при побудові маршрутів
2	Формування словника команд	Створення набору ключових фраз для розпізнавання голосових запитів	Модуль обробки команд та словника	Можливість інтерпретації голосових запитів
3	Обробка голосового запиту користувача	Використання механізмів розпізнавання мови, простий аналіз команд	Модуль голосової взаємодії	Визначення цільової точки за голосовою командою

Продовження таблиці 1.3

1	2	3	4	5
4	Побудова маршруту між точками	Алгоритмічна побудова маршруту по графу вузлів приміщення	Модуль побудови маршрутів	Формування примітивного шляху з точки А до В
5	Голосовий супровід інструкцій	Генерація голосових підказок у режимі реального часу	Модуль озвучення інструкцій	Надання голосових підказок під час руху
6	Розробка застосунку під Android	Реалізація застосунку з архітектурою, зручним інтерфейсом та сервісами	Мобільний застосунок (ядро системи)	Робочий застосунок для Android
7	Тестування прототипу	Запуск на реальному пристрої, перевірка голосових команд і навігації	Тестовий прототип системи	Демонстрація працездатності навігації в умовах квартири
8	Налаштування BLE-маячків	Покупка, налаштування та окреме тестування маячків	Модуль роботи з BLE (на стадії інтеграції)	Підготовка маячків до інтеграції у систему

Аналізуючи таблицю 1.3, можна стверджувати, що розроблений прототип мобільної навігаційної системи вирішує ключові завдання орієнтування користувача в приміщенні: від обробки голосових запитів до побудови маршруту та надання голосових інструкцій. Система дозволяє

користувачеві орієнтуватися в знайомому або новому просторі, отримуючи покрокові голосові та візуальні підказки в режимі реального часу. Система особливо важлива для людей з вадами зору або тих, хто потребує автономної підтримки без сторонньої допомоги.

На відміну від традиційних рішень (друковані карти, контакт з персоналом), запропонований підхід забезпечує персоналізацію маршруту та кастомізацію відповідно до запиту користувача. Система розроблена для роботи повністю в автономному режимі і не потребує складного або дорогого обладнання. Для базової реалізації потрібен мобільний пристрій на базі Android та цифрова карта об'єкту. На даному етапі інтеграція BLE-маяків не завершена через технічні труднощі, але архітектура додатку передбачає можливість підключення модуля позиціонування BLE в майбутньому.

Основний функціонал реалізовано з використанням сучасних інструментів мобільної розробки:

- Flutter + Dart - це кросплатформенна основа для Android-додатків;
- локальний словник ключових фраз для обробки голосових запитів;
- модуль розпізнавання голосу за допомогою Speech-to-Text API (локальна обробка);
- модуль побудови маршрутів на основі примітиву вузла цифрової карти;
- перетворення тексту в мову для голосових навігаційних інструкцій.

Незважаючи на обмеження прототипу (ручне створення карти та словника команд, відсутність BLE-позиціонування на даному етапі), обрана архітектура є масштабованою та придатною для подальшого розвитку. Перспективи системи полягають у наступному:

- інтеграція BLE-маяків для точного позиціонування в приміщенні;
- автоматичне картографування за допомогою AR, LIDAR або камер мобільних пристроїв;
- розширення словника команд за допомогою штучного інтелекту;
- підтримка багатомовності та адаптація до різних типів об'єктів (магазини, лікарні, притулки).

1.4 Мета та задачі дослідження

Метою кваліфікаційної роботи є розробка мобільного застосунку, який забезпечує локальну навігацію в приміщенні на основі технології BLE-маячків, забезпечуючи покращення умов орієнтування для людей з вадами зору.

Для досягнення поставленої мети мають бути вирішені наступні задачі:

- провести аналіз сучасних технологій indoor-навігації (BLE, Wi-Fi, UWB), а також існуючих застосунків для людей з вадами зору;
- сформуванати словник ключових фраз та логіку їх обробки для навігаційних команд;
- розробити прототип мобільного застосунку на Flutter з можливістю аналізу голосової команди, побудови маршруту в межах закритого приміщення, надання голосового супроводу;
- здійснити тестування працездатності системи в умовах реального приміщення (квартири).

Незважаючи на обмеження першого прототипу (відсутність самонавчання, необхідність ручного формування мапи), запропоноване рішення має високий потенціал соціального впливу, оскільки дозволяє реалізувати інклюзивну навігацію для користувачів, які часто стикаються з бар'єрами у повсякденному житті.

2 ОБҐРУНТУВАННЯ ТА МЕТОДОЛОГІЧНЕ ПІДґРУНТЯ ДОСЛІДЖЕННЯ

2.1 Порівняльний аналіз технологій RTLS

Глобальна навігація – це орієнтування на великих відкритих територіях за допомогою супутникових систем: GPS / ГЛОНАСС / Galileo / BeiDou.

Підходить для позиціонування на вулиці, у транспорті, під час подорожей. Має типовий радіус точності від 3 до 10 метрів (на відкритій місцевості). Особливості, якої це: залежність від якості сигналу супутників та погана або відсутня робота у закритих приміщеннях (через екранування сигналу стінами, дахом).

Локальна навігація – це система орієнтування користувача всередині обмежених або закритих приміщень (квартир, офісів, торгових центрів, вокзалів тощо). Вона базується на технологіях, що не потребують зовнішніх супутникових сигналів (як-от GPS), а використовують внутрішню інфраструктуру.

Основні особливості локальної навігації, це: висока точність у закритих приміщеннях (1–3 метри залежно від технології та налаштувань), автономність від інтернету чи глобальних систем позиціонування та можливість гнучкої конфігурації під конкретне приміщення (наприклад, твоя квартира).

Провівши аналіз та систематизацію систем локальної навігації, маю ґрунтовні висновки щодо поділу їх на категорії (Рисунок 2.1).

Системи RTLS забезпечують точну локалізацію об'єктів у режимі реального часу. Найпоширеніші технології включають BLE, UWB, RFID, Wi-Fi RTT і оптичні системи. Вибір технології залежить від умов використання, точності, вартості впровадження та інших параметрів [9].

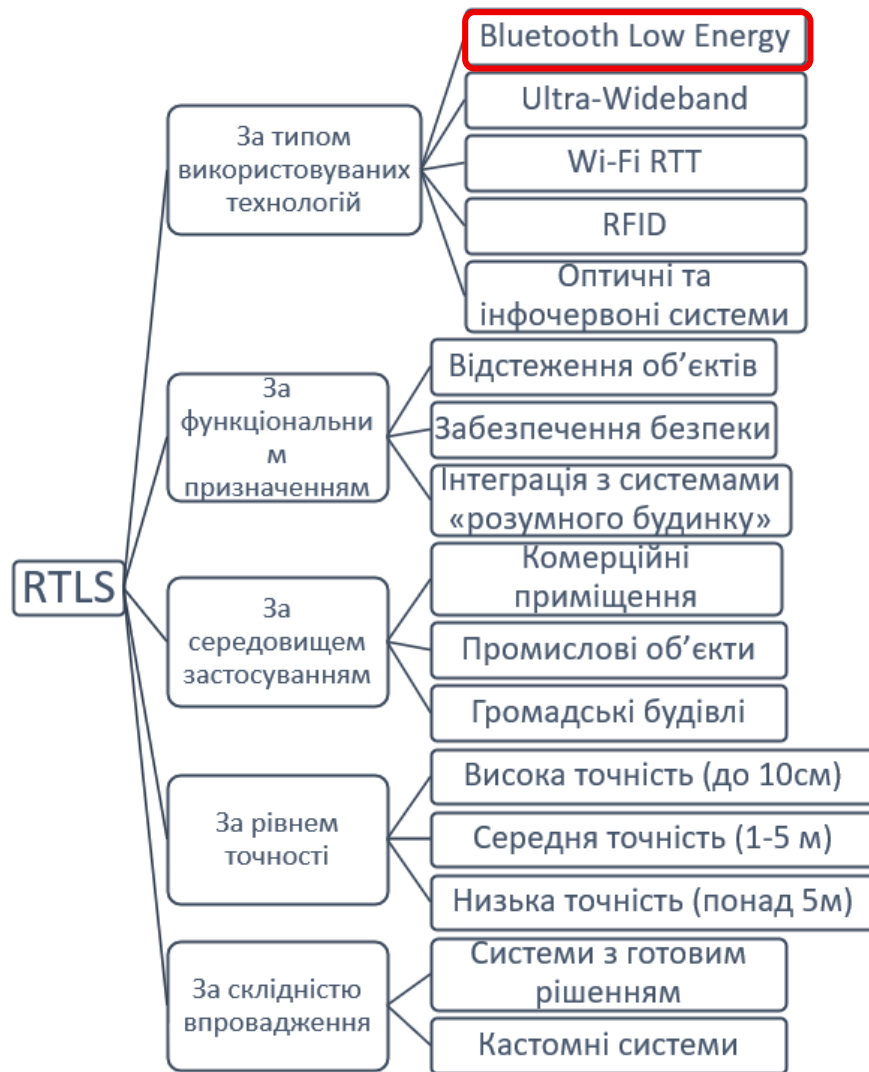


Рисунок 2.1 – Типізація систем локальної навігації

BLE і UWB є лідерами для складних умов, де важлива точність, а BLE особливо ефективна для інтеграції з системами для людей з вадами зору завдяки низькій вартості і сумісності з мобільними пристроями.

Підсумуючи результати класифікації, можна зробити наступні висновки. Для складних середовищ найбільш ефективними є BLE і UWB, що забезпечують високу точність і простоту реалізації. Системи RTLS можна класифікувати за технологією, функціональним призначенням (навігація, безпека, відстеження), середовищем застосування (торгові центри, лікарні, склади) та рівнем точності. Для інтеграції з навігаційними системами для людей з вадами зору BLE є найкращим вибором завдяки низькій вартості,

сумісності зі смартфонами та підтримці голосових команд.

Серед існуючих систем RTLS (Real-Time Location Systems) можна виділити наступні. Для відстеження товарів, визначення місцезнаходження вантажу на складі або під час транспортування, прискорення процесу сортування та доставки, оптимізації логістичних процесів у системі доставки «Нова Пошта». Для інтеграції з пристроями розумного дому, наприклад, для автоматизації освітлення або клімат-контролю на основі місцезнаходження мешканців будинку. Це корисно для економії ресурсів і підвищення комфорту. Для підвищення безпеки в закладах охорони здоров'я. Для відстеження інструментів, машин і робітників на великих виробничих площах для забезпечення безпеки та оптимізації роботи конвеєра. Для навігації відвідувачів, аналізу потоку клієнтів для оптимізації розміщення товарів і поліпшення обслуговування у торгових центрах [5].

Таким чином, системи локальної навігації є затребуваними, але саме у секторі соціальної інтеграції вони ще не знайшли свого застосування. Результати класифікації та порівняння технологій RTLS дозволяють зробити наступні висновки:

- BLE забезпечує низькі витрати на впровадження, легку інтеграцію та підтримку голосових команд (ця технологія ідеально підходить для внутрішньої навігації в складних умовах);
- UWB має високу точність і підходить для середовищ, де потрібне детальне відстеження об'єктів, але її вартість вища, ніж BLE;
- Wi-Fi RTT дозволяє використовувати існуючу інфраструктуру, але через нижчу точність особливо підходить для великих приміщень з низькими вимогами до точності визначення місцезнаходження;
- RFID в основному використовується для інвентаризації та відстеження, але не є основним рішенням для навігації через свою обмежену точність.

Порівняння технологій RTLS показує, що BLE має найбільший потенціал для інтеграції в навігаційні системи, призначені для людей з вадами зору.

Інтерактивні внутрішні карти – важлива частина сучасних навігаційних

систем. Вони дозволяють зберігати інформацію про ключові об'єкти та маршрути і синхронізувати ці дані з системами позиціонування.

Bluetooth-маячки, розміщені в стратегічних місцях (наприклад, біля дверей і коридорів), працюють з інтерактивними картами, щоб визначити точне місцезнаходження користувача. Це дозволяє прокладати маршрути в режимі реального часу та інформувати користувача про його поточне місцезнаходження за допомогою голосових підказок.

Інтерактивні карти забезпечують:

- відображення маршруту візуально та за допомогою голосових підказок;
- інтеграцію з Bluetooth-маячками для забезпечення точності навігації;
- оновлення даних у реальному часі для відображення змін у приміщенні.

Такі карти створюють зручний інтерфейс для користувачів, особливо з обмеженими можливостями, і допомагають підвищити їхню мобільність та незалежність [10].

2.2 Вибір технологічної бази для рішення поставлених задач

Розробка додатку з використанням системи локальної навігації для людей з вадами зору в обмеженому просторі потребує сучасних технологій, інструментів та апаратних засобів, що забезпечують точне позиціонування, голосову взаємодію та автономну роботу в режимі офлайн. Обрана методологічна основа дозволяє реалізувати функціональний прототип з можливістю подальшого масштабування. Для реалізації поставлених задач були використані наступні програмні та апаратні рішення: Dart, Flutter, Teltonika (BLE-маячки), Android SDK, Android Studio.

Dart - це сильно типізована мова програмування, яка підтримує такі сучасні функції, як нульова безпека, асинхронні операції через `async/await` та ізольовану систему потоків. Використання Dart надає переваги швидкої

розробки, передбачуваної продуктивності та зрозумілого синтаксису, особливо враховуючи освітню підготовку під час мого навчання на бакалавраті, де саме цей мовний стек використовувався в бізнес-курсі [13].

Flutter - це фреймворк від Google для швидкого створення крос-платформних додатків з єдиною кодовою базою для створення візуально привабливих, продуктивних інтерфейсів для iOS, Android, Інтернету та настільних комп'ютерів. Документація пропонує лабораторії коду та готові шаблони, які допомагають прискорити навчання - особливо для студентів. Вибір Flutter гарантує, що масштабне рішення може бути перенесене на інші платформи в майбутньому [14].

Teltonika Blue Coin ID - компактний BLE-маяк з низьким енергоспоживанням і терміном служби батареї до 5 років. Документація містить інструкції з налаштування за допомогою конфігуратора Teltonika Configurator та рекомендації щодо параметрів рівня сигналу. Такі маяки здатні забезпечити локальну навігацію там, де GPS недоступний, тому Teltonika була обрана як недороге і досить точне рішення. Альтернативи включають Estimote або Kontakt.io, але вони дорожчі та менш доступні на місцевому рівні [15].

Android SDK - це набір інструментів та API (зокрема BluetoothAdapter, Permissions), які дозволяють створювати нативні модулі для розпізнавання BLE-сигналів, обробки дозволів та управління компонентами пристрою. Використання Android SDK гарантує повний контроль над поведінкою програми на рівні операційної системи. Обрана мінімальна версія SDK відповідає поточній версії Android 16+ [16].

Android Studio - це офіційне середовище розробки від Google, яке забезпечує глибоку інтеграцію з Android SDK, емуляторами, плагінами Gradle та Flutter. Стабільна версія Meerkat 2024.3.2 та бета-версія Narwhal 2025.1 включають інструменти штучного інтелекту, емулятори, інтеграцію Gemini для швидкого налагодження та автозавершення. Вибір Android Studio замість VS Code ґрунтувався на наявності потужних інструментів Flutter,

налагодженні на реальних пристроях та готовності до перенесення додатку на інші платформи [17].

Обрана технологічна база - комбінація BLE, Flutter та Android Studio - є результатом ретельного аналізу вимог проєкту та технічних можливостей. Цей вибір дозволив нам створити функціональний і доступний офлайн-прототип мобільного додатку, який підтримує голосову взаємодію і точне локальне позиціонування.

Технологію BLE було обрано через її енергоефективність та достатню точність для навігації в приміщеннях, з радіусом дії ~ 30 м та точністю приблизно 1-3м. Комерційно доступні маяки, такі як Teltonika Blue Coin ID (таблиця 2.1), забезпечують багаторічний термін служби батареї без заміни, що важливо для легкого обслуговування системи.

Таблиця 2.1 – Узагальнення переваг та недоліків технології BLE

Переваги	Недоліки
низьке енергоспоживання	обмежена зона дії (≈ 30 м)
достатня для внутрішніх приміщень точність	потреба в розміщенні множини маячків для повного покриття простору
ручність розгортання без додаткового обладнання	складність налаштувань

BLE - це оптимальний компроміс між точністю, енергоефективністю та вартістю навігації в приміщенні.

Flutter був обраний за його здатність до швидкої крос-платформної розробки з функцією Hot Reload та єдиною базою коду для Android та iOS. Візуальні віджети забезпечують привабливий та стабільний користувацький інтерфейс на всіх платформах.

Таблиця 2.2 – Узагальнення переваг та недоліків технології Flutter

Переваги	Недоліки
короткий цикл розробки через Hot Reload – дозволяє швидко вносити й перевіряти зміни	обмежена екосистема сторонніх пакетів – інколи складніші інтеграції
одна кодова база для декількох платформ – значно зменшує витрати на підтримку	іноді немає підтримки специфічних API – можуть знадобитися платформи-специфічні рішення
висока продуктивність у порівнянні з багатьма іншими фреймворками	

Flutter є дуже ефективним інструментом для реалізації MVP з подальшим масштабуванням, хоча і з деякими обмеженнями в екосистемі (таблиця 2.2).

Android Studio - офіційне середовище розробки з глибокою інтеграцією з Android SDK, емуляторами та підтримкою Flutter/Dart. Вона забезпечує ефективну розробку та налагодження для реального тестування на пристроях Android (таблиця 2.3).

Таблиця 2.3 – Узагальнення переваг та недоліків Android Studio

Переваги	Недоліки
інтеграція з Android SDK, емуляторами та налагодженням – допомагає реалізувати та тестувати функції BLE	висока ресурсоемність і велика вага IDE – потребує сучасний комп'ютер
розширений набір інструментів (Lint, профайли, шаблони) – підвищує якість коду	повільний старт і іноді системні лаги – може ускладнювати робочий процес
офіційна підтримка Flutter-плагінів – гарантує стабільність інструментів	складність для дуже простих проєктів – порівняно з легшими IDE

Android Studio - найкраще середовище для професійної розробки Android/Flutter додатків, незважаючи на високі вимоги до ресурсів.

2.2.1 Переваги для локальної навігації

Інклюзія - система розроблена з урахуванням потреб людей з вадами зору. Голосові інструкції в поєднанні з візуальними та текстовими підказками забезпечують автономну навігацію інтер'єром. Це дозволяє здійснювати навігацію без сторонньої допомоги, сприяючи соціальній інтеграції.

Автономний режим - всі функції працюють без підключення до Інтернету. Це має вирішальне значення для бездротової роботи в обмеженому просторі або з обмеженим радіусом дії, забезпечуючи стабільні голосові інструкції та навігацію.

Масштабованість - архітектура дозволяє адаптувати додаток до різних приміщень. Цифрова карта у форматі JSON дозволяє легко додавати нові локації. У майбутньому можлива інтеграція технології доповненої реальності для автоматичного сканування простору або модулів штучного інтелекту для генерації маршрутів і словника голосових інструкцій.

Мультиплатформенність – Flutter дозволяє адаптувати застосунок з Android для iOS в майбутньому без необхідності перепрограмування базової логіки. Це дає можливість розробляти рішення для ширшої аудиторії без подвоєння вартості, що особливо важливо для громадських об'єктів.

3 ФОРМАЛЬНА ТА ПРАКТИЧНА РЕАЛІЗАЦІЯ РІШЕННЯ

3.1 Опис загальної побудови системи BLE-навігації. Взаємодія основних компонентів

Формальне підґрунтя рішення базується на розробці мобільного застосунку, який виконує роль центрального елемента системи локальної навігації в закритому просторі. Такий застосунок повинен забезпечувати взаємодію з BLE-маячками, обробку голосових запитів, побудову маршруту на основі цифрової карти та надання навігаційного супроводу в режимі реального часу.

Ключовим компонентом системи є механізм позиціонування, що базується на технології Bluetooth Low Energy (BLE). Застосунок на смартфоні виявляє сигнали від розташованих у приміщенні маячків типу ELA Blue COIN ID Beacon, ідентифікує їхні унікальні ID, та за рівнем сигналу (RSSI) визначає приблизну відстань до них. Цей підхід забезпечує визначення поточної позиції користувача в межах локальної карти. У системі передбачається наявність одного маячка, який постійно перебуває при користувачеві (умовна «точка А») і трьох фіксованих маячків, що закріплені у вузлових точках маршруту (умовні транзитні зони до умовних «точок В»).

Цифрова карта приміщення реалізується у форматі JSON, де вузли відповідають ключовим позиціям (наприклад, «bedroom», «bathroom», «balcony»), а ребра – допустимим переходам між ними. В основі побудови маршруту лежить графова модель, яка дозволяє застосовувати такі алгоритмічні підходи, як пошук у ширину BFS.

Для забезпечення автономної взаємодії користувача із застосунком реалізовано інтерпретацію голосових команд. Використовуючи бібліотеку `speech_to_text`, застосунок розпізнає запит (наприклад, «go to the kitchen»), аналізує його за попередньо сформованим словником у форматі JSON та

визначає цільову точку маршруту. Після визначення маршруту застосунок генерує покрокові інструкції. Користувач отримує голосовий супровід у реальному часі. Ці повідомлення синхронізовані з оновленням позиції користувача, яке відбувається при наближенні до маячків.

Підхід до реалізації системи дозволяє на початковому етапі уникнути складних обчислень або необхідності у серверній частині. Вся логіка реалізована локально, що є перевагою для офлайн-використання.

Таким чином, формальна структура рішення поєднує BLE-позиціонування, обробку голосових команд, алгоритмічну маршрутизацію та інтерактивний супровід – усе в межах мобільного застосунку. Це дозволяє створити доступне, автономне та масштабоване рішення для локальної навігації, адаптоване до потреб людей з порушеннями зору.

Система BLE-навігації складається з кількох взаємопов'язаних модулів, кожен із яких виконує визначену функцію та забезпечує автономну роботу в офлайн-середовищі. Їх інтеграція дозволяє реалізувати точне позиціонування, обробку голосових команд та голосову навігацію з мінімальними вимогами до інфраструктури.

3.2 Структурна організація програмних модулів

У цьому підрозділі описано ключові програмні модулі, їх функціональність, зв'язки із зовнішніми бібліотеками та роль у системі навігації.

Головні компоненти мобільного застосунку наведено нижче.

Файл `home_screen` (UI) виконує роль координаційного центру, де всі сервіси ініціалізуються, відбувається завантаження JSON-карти, запуск голосового режиму та BLE-сканування. Використання інтегрованих виджетів, таких як стрілка-навігатор, кнопка голосу та карта, створює єдиний інтерфейс взаємодії з користувачем (Рисунок 3.1).

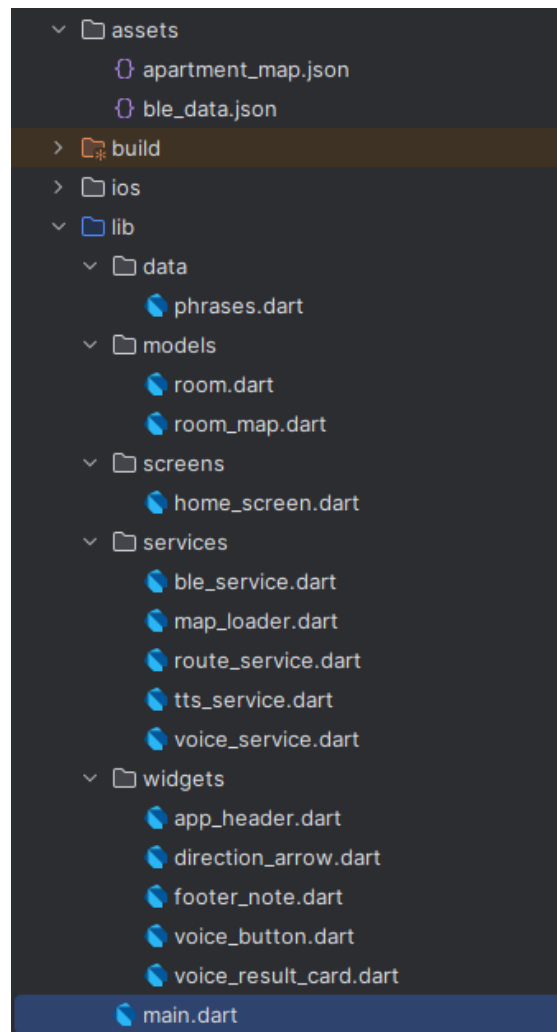


Рисунок 3.1 – Дерево файлів

Файл `ble_service` реалізує роботу з BLE-маячками: запитує необхідні дозволи, зчитує конфігурацію маячків з `ble_data.json` і періодично сканує BLE-пристрої. Обробка сигналів `manufacturerData` дозволяє ідентифікувати iBeacon-маячки та передати знайдені дані у `home_screen` для подальшого визначення місця розташування (лістинг 3.1).

Лістинг 3.1 – Конфігурація блютуз маячків

```
{
  "identifier": "beacon_zone_1",
  "uuid": "552452FE-F374-47C4-BFAD-9EA4165E1BD9",
  "major": 523,
  "minor": 266
},
```

Файл `room_map` забезпечує роботу з цифровою картою приміщення, завантажуючи її через `rootBundle.loadString` і формуючи граф на основі вузлів `Room` (із сусідами, координатами та беконом). Вбудовані методи `findRoute`, `startNavigation`, `nextStep` та `getNextInstruction` дозволяють обчислювати та вести маршрут користувача кімнатами інтер'єру.

Файл `route_service` – абстрактний модуль, який виконує алгоритм BFS для побудови оптимального маршруту між двома кімнатами. Він використовується як утиліта у `voice_service` для навігації на основі команд користувача.

Файл `voice_service` здійснює обробку розпізнаних команд від `speech_to_text`, інтерпретує їх через словник фраз (`commandToRoomId`, `commandActions`), активує функції побудови маршруту в `RoomMap` та забезпечує озвучення результатів через `TtsService`, таким чином створюючи інтерактивний голосовий інтерфейс.

Файл `tts_service` використовує `flutter_tts` для синтезу мови: після ініціалізації з параметрами (`language`, `speechRate`, `pitch`) забезпечує озвучення текстових інструкцій. Цей модуль спрямований на підвищення доступності системи, особливо для людей з порушеннями зору.

3.2.1 Модуль взаємодії з BLE-маячками. BLE-маячок ELA Blue COIN ID Beacon

Модуль `ble_service` відповідає за ініціалізацію сканування BLE-маячків і обробку отриманих сигналів. Він запитує необхідні права (`bluetoothScan`, `bluetoothConnect`, `locationWhenInUse`) через `permission_handler`, завантажує конфігурацію маячків з JSON (`assets/ble_data.json`) через `rootBundle`, після чого запускає сканування з обмеженням за часом. Сервіс аналізує `manufacturerData` для виявлення iBeacon-маячків, порівнює їх із конфігурацією, і при збігу передає інформацію в UI через `callback`.

Для навігації використовується ELA Blue COIN ID Beacon.

ELA Blue COIN ID Beacon – це компактний, автономний Bluetooth-маячок, призначений для ідентифікації та позиціонування об'єктів у закритих просторах. Його характеристики роблять його ідеальним вибором для систем навігації, особливо в умовах, де GPS-сигнал недоступний (Рисунок 3.2).



Рисунок 3.2 – BLE-маячок ELA Blue COIN ID Beacon

Основні технічні характеристики:

- технологія зв'язку: Bluetooth Low Energy (BLE) 4.0 / 4.2;
- дальність передачі сигналу: до 200 метрів у відкритому просторі;
- автономність: до 5 років роботи від вбудованої літієвої батареї;
- ступінь захисту: IP68 – повна пило- та вологозахищеність;
- діапазон робочих температур: від -30°C до +70°C;
- габарити: діаметр 36 мм, товщина 10 мм;
- вага: 11 г.

Додаткові функції:

- NFC-програмування: спрощує конфігурацію пристрою;
- підтримка протоколів: ID (ELA), iBeacon (Apple), Eddystone (Google);
- LED-індикатор: функція «Pick-to-Light» для візуальної ідентифікації;
- можливість оновлення прошивки по повітрю (OTA).

Цей маячок забезпечує надійну та стабільну роботу в різноманітних умовах, що робить його придатним для використання в системах навігації для людей з вадами зору. Його компактність, довговічність та простота в налаштуванні дозволяють ефективно інтегрувати його в мобільні додатки, забезпечуючи точне позиціонування та зручну взаємодію з користувачем.

Інтеграція ELA Blue COIN ID Beacon у систему навігації дозволяє реалізувати точне визначення місцезнаходження користувача в реальному часі, що є критично важливим для забезпечення безпеки та автономності людей з порушеннями зору. Завдяки своїм технічним характеристикам, цей маячок є оптимальним вибором для побудови ефективної та надійної системи внутрішньої навігації.

3.2.2 Взаємодія Bluetooth-маячків із інтерактивними картами

Bluetooth-маячки є ключовим елементом у забезпеченні точності визначення місцезнаходження користувача в системі. Вони розміщуються в стратегічних місцях, таких як двері, коридори та ключові зони інтересу.

Процес взаємодії відбувається наступним чином:

- Bluetooth-маячки надсилають сигнали з унікальними ідентифікаторами;
- система позиціонування приймає ці сигнали, аналізує їхню інтенсивність і визначає координати користувача;
- інтерактивна карта отримує інформацію про поточне місцезнаходження користувача та оновлює відображення маршруту в режимі реального часу;
- всі дії відображаються в застосунку, який за допомогою інтерфейсу користувача направляє користувача по маршруту.



Рисунок 3.3 – Зона в квартирі з маячком

Синхронізація між маячками та інтерактивною картою дозволяє враховувати зміни у просторі та надає користувачам точні напрямки руху. Це особливо важливо для людей з вадами зору, які покладаються на голосові підказки та маршрути на основі актуальної інформації (Рисунок 3.3).

Маячок розташований в гарновидному місці і має LED-підсвітку для кращої орієнтації в просторі. Відлік маршруту починається з координат блютуз маячка, який знаходиться з користувачем. Який в свою чергу взаємодіє з інтерактивними картами та іншими маячками, що розташовані по зонам (Рисунок 3.4).



Рисунок 3.4 – Маячок з користувачем

Налаштування маячків та режиму мовлення потрібне налаштування через NFC-з'єднання з офіційним додатком (Device Manager Mobile, Teltonika Configurator). Через NFC задають: UUID, Major, Minor, потужність (TX Power) та інтервал реклами. Саме завдяки цьому і активується режим iBeacon. Що і відображено на рисунку 3.5.

Потужність потрібно встановлювати з урахуванням бажаного радіусу дії: більша потужність → більше охоплення, але швидша витрата батареї. У рекламних пакетах iBeacon передається параметр «Measured Power», очікуване значення RSSI на відстані 1 м.

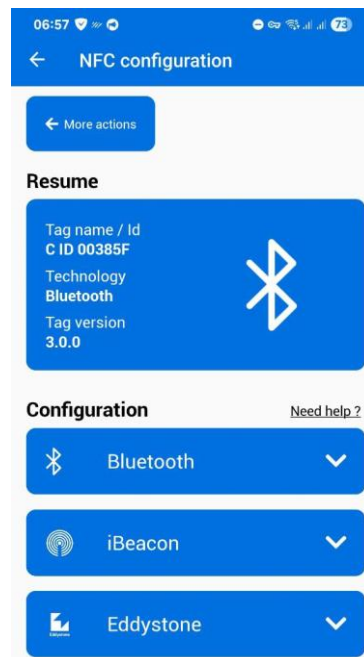


Рисунок 3.5 – Налаштування маячку через NFC

Налаштування цієї величини допомагає пристрою краще оцінити відстань. Для коректної калібровки рекомендовано тримати телефон на відстані 1 м в безперешкодному просторі та записати середнє RSSI протягом ~30 секунд. Приклад з застосунку на Рисунок 3.6.

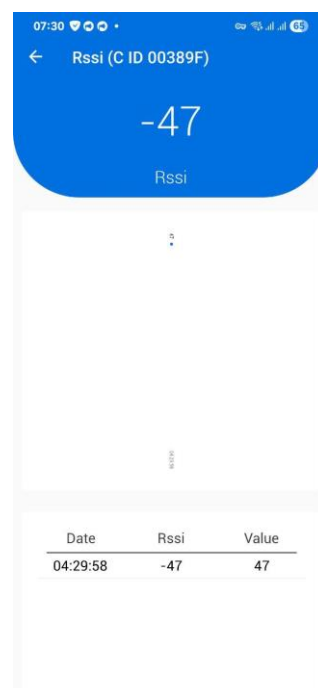


Рисунок 3.6 – RSSI сигнал біля маячка

3.2.3 Алгоритм побудови маршруту

Для коректного функціонування застосунку необхідна система внутрішньої навігації, що в свою чергу базується на інтерактивній карті об'єкта, Bluetooth-маяках та голосових підказках. Її функціональна модель включає наступні ключові елементи:

- інтерактивна карта – цифрове відображення простору приміщення з координатами ключових об'єктів;
- Bluetooth-маячки – пристрої визначення місцезнаходження користувача, які надсилають сигнали за допомогою технології BLE;
- голосові підказки – механізм взаємодії між системою та користувачем, що забезпечує сповіщення про маршрути або перешкоди;
- алгоритми маршрутизації – логіка побудови оптимального маршруту, що враховує поточне місцезнаходження користувача, заданий пункт призначення та перешкоди на шляху.

Функціональна модель включає в себе:

- отримання вхідних даних (голосова команда або координати);
- обробку даних для побудови маршруту;
- інтерактивне прокладання маршруту у вигляді голосових інструкцій та візуальних підказок на карті.

Центральною частиною моделі є система синхронізації між Bluetooth-маячками та інтерактивною картою, яка забезпечує точне місцезнаходження користувача.

На основі проведеного аналізу предметної області, виділимо основні складові елементи запропонованої системи:

- технологія інтерпретації голосових повідомлень;
- інтерактивні карти приміщень з просторовою локалізацією ключових об'єктів;
- система позиціонування рухомих об'єктів в приміщенні;
- підсистема визначення маршруту від рухомого об'єкта до пункту призначення, наданого у вигляді голосового запиту.

Функціональна модель системи складання маршруту до шуканого місця або точки зображена на рисунку 3.7. Зображена діаграма служить для визначення вимог і зазначення функцій для подальшої розробки системи [8].

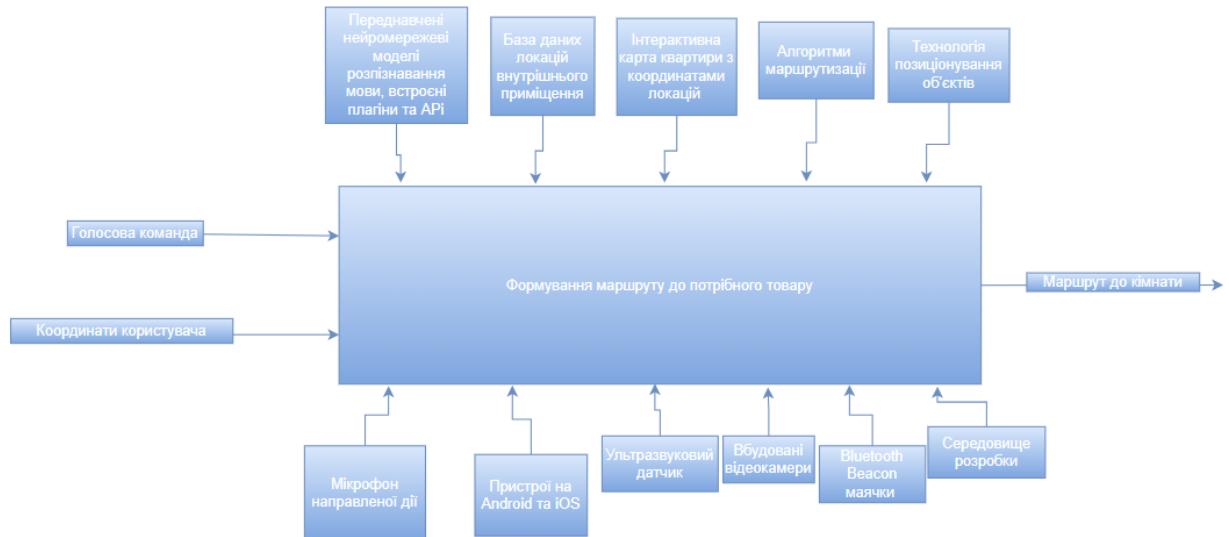


Рисунок 3.7 – Узагальнена функціональна модель системи складання маршруту до шуканого місця або товару

Для початку роботи системи складання маршруту до шуканого місця або точки необхідно отримати вхідні дані – голосова команда користувача, аналіз якої має визначити шукане місце або товар, маршрут до якого треба прокласти в межах приміщення (indoor localization). А також дані про локальні координати розташування користувача. Система створює безпечний маршрут до заданого об'єкта від початкових координат клієнта.

В результаті декомпозиції узагальненої функціональної моделі системи складання маршруту до шуканого місця або точки було виділено чотири структурні блоки: інтерпретація голосового запиту у команду, визначення відповідності шуканого місця із можливих, аналіз перешкод на шляху користувача, побудова маршруту (Рисунок 3.8).

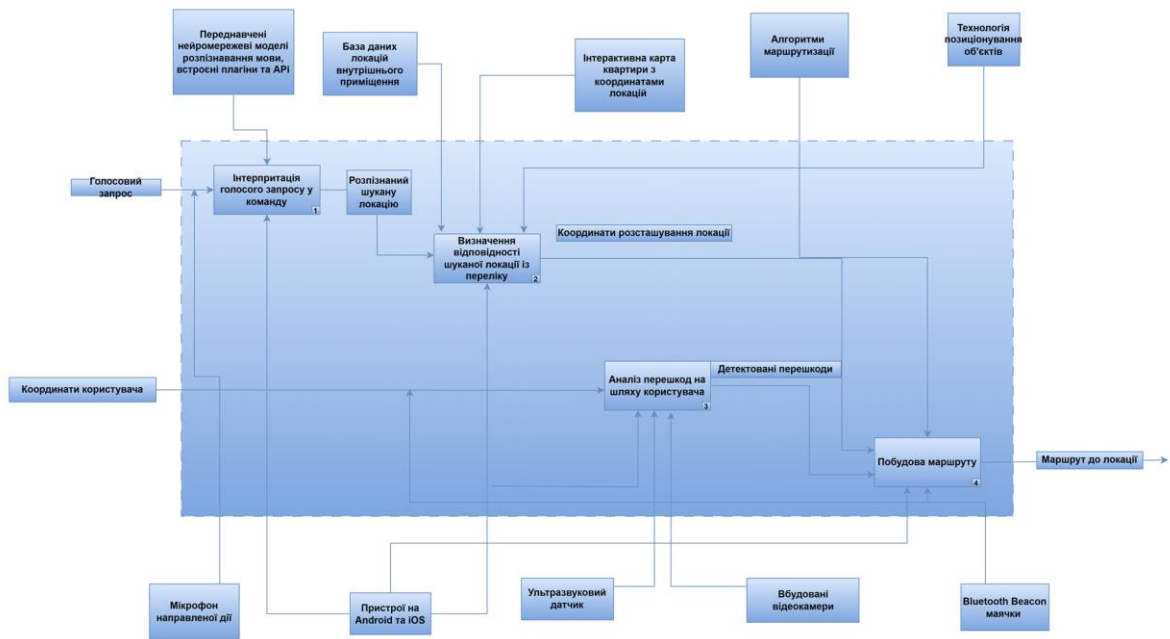


Рисунок 3.8 – Результат декомпозиції процесу на взаємопов’язані та взаємозалежні функціональні блоки

3.2.4 Модуль голосової взаємодії (розпізнавання команд, озвучення інструкцій)

Визначення відповідності шуканого місця із існуючими забезпечується за рахунок переліку, в якій всі записи вважаються ключовими словами.

Інтерактивна карта з координатами кімнат, які відповідають існуючим місцям, допомагають знаходити потрібні координати.

Дані синхронізуються з технологією позиціонування, що дозволяє ефективно визначати розташування об’єктів у просторі.

Визначення перешкод на шляху користувача відбувається шляхом аналізу простору навколо користувача на основі вхідної інформації із маячків. Алгоритми маршрутизації формують оптимальний і безпечний шлях до місця з урахуванням перешкод. Результат синхронізується через Bluetooth Beacon-маячки та подається у голосових інструкціях користувачеві.

Окрему увагу та ряд досліджень в роботі буде приділено саме обробці голосових запитів користувачів, які можуть мати багато лексичних неоднозначностей (Рисунок 3.9) [6].



Рисунок 3.9 – Функції підсистеми голосового формування запиту

Планування роботи над проєктом за етапами наведено на рисунку 3.10.

Розробка системи позиціонування на основі інтерактивних карт приміщень для внутрішньої навігації є першим кроком, необхідним для реалізації запропонованої системи. Даний крок передбачає рішення наступних задач:

- створення цифрової карти вибраних приміщень з розташуванням ключових об'єктів;
- інтеграція карти з можливістю оновлення даних у реальному часі;
- інтеграція з картою для відображення поточного розташування;
- визначення ключових точок інтересу, можливість додавання нових об'єктів з призначенням ідентифікаторів;
- вибір і реалізація технології позиціонування (Wi-Fi, Bluetooth-маячки, UWB, ультразвук тощо);
- створення алгоритму визначення поточного місця користувача з високою точністю.

Голосові підказки є невід'ємною частиною системи, що робить її більш доступною та зручною (Рисунок 3.10). Розробка підсистеми маршрутизації на основі інтерактивних карт приміщень для внутрішньої навігації є другим кроком, необхідним для реалізації запропонованої системи. Даний крок передбачає рішення наступних задач:

- алгоритми побудови маршрутів від поточного місця до пункту

призначення (алгоритми триангуляції, Дейкстри, SLAM);

- оптимізація маршрутів залежно від потреб користувачів (можливі критерії - швидший шлях, безпечний шлях тощо);
- голосові інструкції для користувачів, що відображаються під час руху;
- використання датчиків (ультразвукових, інфрачервоних, систем штучного зору) для визначення відстаней або перешкод на шляху користувача (Рисунок 3.10).



Рисунок 3.10 - Планування роботи над проектом

Дослідження та розробка методів інтерпретації голосових повідомлень в системі навігації всередині приміщення є третім кроком, необхідним для реалізації запропонованої системи.

Є найбільш наукоємним етапом розробки, бо від точності інтерпретації голосового запиту залежить точність створеного маршруту. Даний крок передбачає рішення наступних задач:

- використання спрямованих мікрофонів або мікрофонних масивів для чіткого прийому голосових команд у приміщеннях із високим рівнем шуму;
- використання наявних бібліотек та платформ для розпізнавання голосу (наприклад, Google Speech-to-Text, Microsoft Azure Speech) [7];
- розробка та використання алгоритмів для перетворення голосових запитів у текст і подальшого розуміння.

Отже, розпізнавання голосових команд – користувач вводить

голосовий запит для визначення пункту призначення. Нейромережеві моделі обробляють цей запит і перетворюють його на текстову команду.

Для обробки голосових запитів використовуються спрямовані мікрофони, які забезпечують чітке розпізнавання навіть у шумному середовищі. Голосові підказки синхронізуються з інтерактивною картою, що забезпечує максимально зручну взаємодію з користувачем.

Віджет `voice_button` – UI-віджет, що активує голосові команди через `speech_to_text`. Він створює анімовану кнопку та керує станами прослуховування. Плагін `speech_to_text` підтримує Android, iOS, MacOS і web і призначений для коротких команд.

Файл `voice_service` – отримує розпізнані тексти, нормалізує їх, визначає дію (`where_am_i`, `step_done`, кімнату тощо) через словник команд, обробляє маршрут або видає озвучення через `TtsService`.

Віджет `voice_result_card` – UI-компонент, що відображає результат розпізнавання: поточну команду або маршрутні інструкції.

Згідно рисунку 3.11 обробка голосової команди в побудову маршрута відбувається наступним чином: Користувач натискає мікрофон (`VoiceButton`), говорить: наприклад, «go to kitchen». Плагін `speech_to_text` захоплює аудіо, перетворює на текст → передає його в `VoiceService`. `VoiceService.normalize()` переводить текст у нижній регістр та видаляє пунктуацію. Потім перевіряє:

- чи є фраза командою (`commandActions`) — наприклад, «where am I»;
- якщо ні, то шукає відповідність кімнаті (`commandToRoomId`) → отримує `roomId = kitchen`.

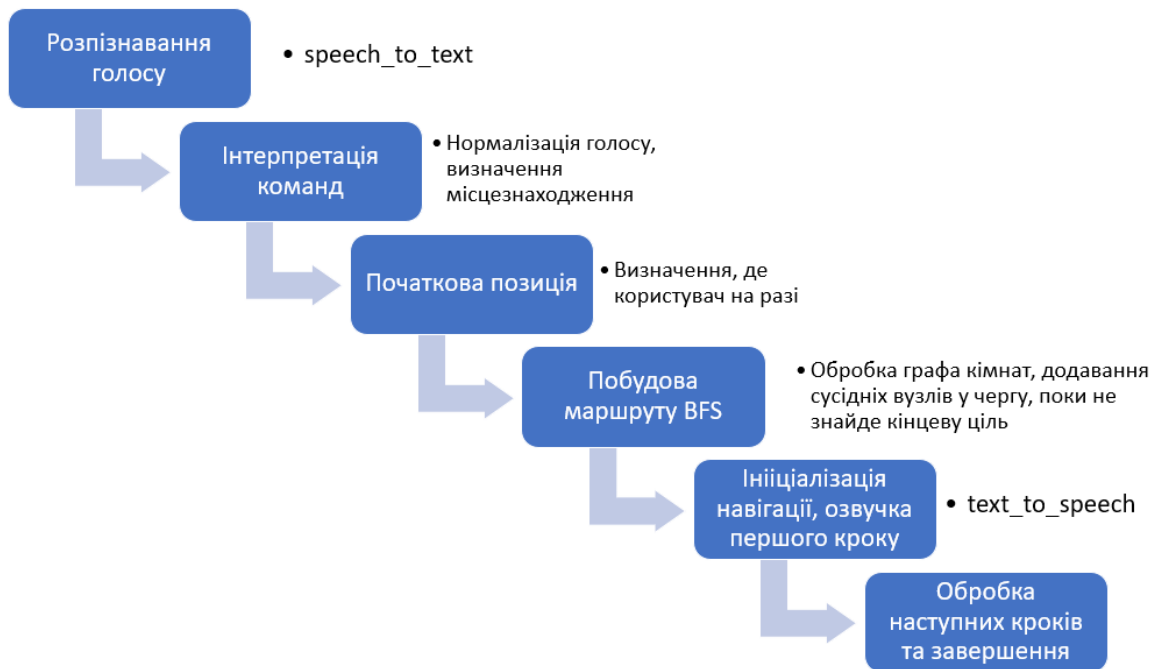


Рисунок 3.11 – Схема побудови маршруту від «точки А» до «точки В»

Система знає, де користувач наразі: `RoomMap.currentRoomId` (наприклад, “bedroom”). Це точка А.

`RoomMap.findRoute (fromId = «bedroom», toId = «kitchen»)` викликає алгоритм BFS:

- оброблює граф кімнат: додає сусідні вузли у чергу, поки не знайде ціль;
- повертає шлях у вигляді списку ID: [“bedroom”, “corridor”, “kitchen”].

`RoomMap.startNavigation(room)` ініціює маршрут, встановлює `_route` і `_stepIndex = 1`.

`VoiceService` викликає `TtsService.speak(“Starting navigation to kitchen”)`, потім: «First, go to corridor».

`DirectionArrow` обчислює кут між координатами кімнат (x,y) — і показує стрілку. `VoiceResultCard` відображає текст: «First, go to corridor».

Користувач каже “step done” або система фіксує сигнал від маячка. Викликається `RoomMap.nextStep()`, вказівка нової кімнати, озвучення наступної дії.

Якщо `_stepIndex >= route.length`, `nextStep()` повертає `null` — система озвучує “You have reached your destination”. Навігація завершується.



Рисунок 3.12 – Побудова маршруту BFS

Спочатку система завантажує дані про приміщення з JSON-файлів, де зберігається інформація про кімнати, їхні координати та сусідів. Ці дані структуруються у вигляді об'єктів `Room`, які утворюють граф приміщення (Рисунок 3.12).

Далі для побудови маршруту викликається метод `findRoute(String fromId, String toId)`, який ініціалізує чергу шляхів і додає в неї початкову кімнату. Алгоритм поступово обходить граф кімнат у ширину: на кожному кроці перевіряються сусідні кімнати, і якщо серед них знайдена цільова, шлях повертається як результат.

Після цього відбувається ініціалізація навігації. Викликається метод `startNavigation(String targetRoomId)`, який зберігає знайдений маршрут і готує застосунок до видачі інструкцій. Голосова інструкція про перший крок генерується і озвучується за допомогою `TtsService`.

Під час навігації система переходить у режим очікування подальших команд. Вона слухає команди користувача, наприклад «step done», або автоматично обробляє дані з BLE-маячків. При отриманні підтвердження кроку викликається метод `nextStep()`, який оновлює поточну позицію [11].

Для демонстрації роботи системи було створено графічну карту приміщення, що відображає зонування простору квартири. Кожне приміщення (кімната) представлено окремою зоною на карті й позначено власною назвою: Bedroom, Office, Corridor, Kitchen, Bathroom, Balcony.

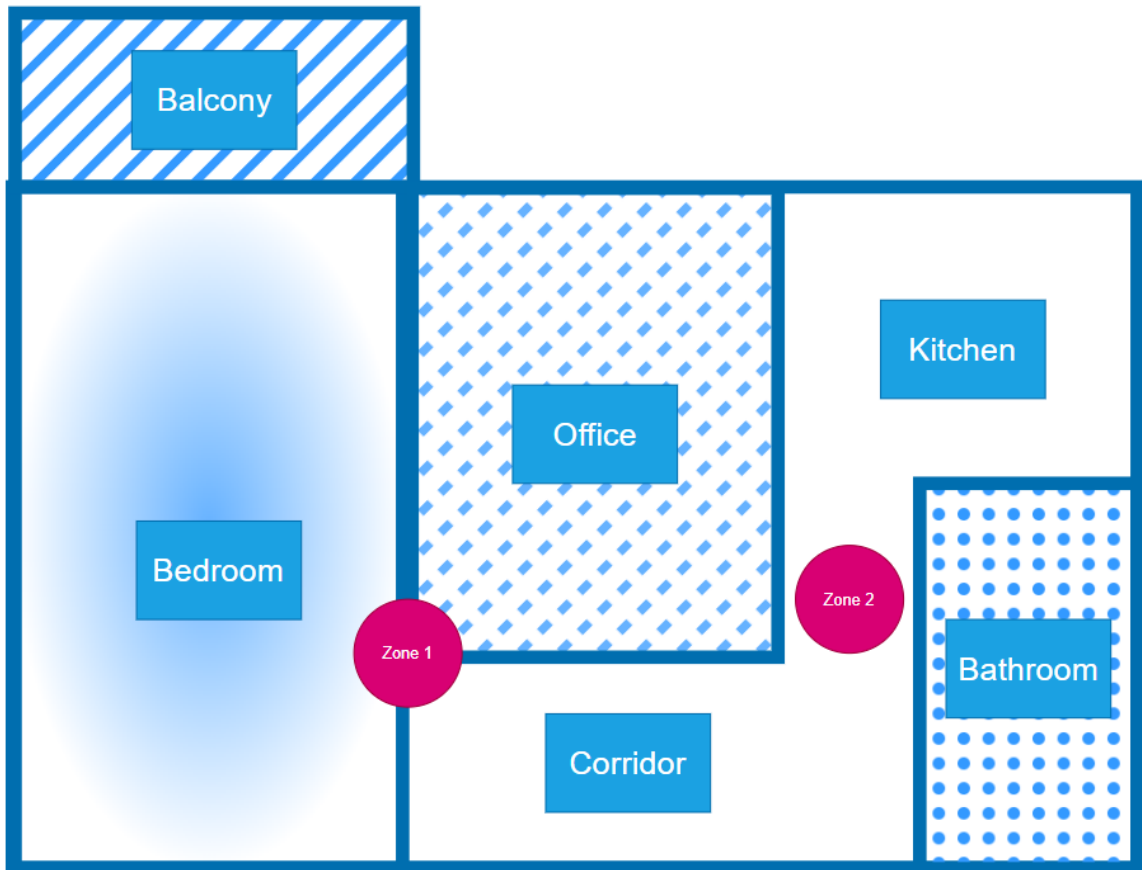


Рисунок 3.13 – Графічна карта приміщення

У структурі карти передбачено (Рисунок 3.13):

- BLE-зони (Zone 1, Zone 2) — це області розташування BLE-маячків, які фіксують місцезнаходження користувача на маршруті; зони позначені червоними мітками для зручної візуалізації;
- маршрутизація між кімнатами реалізується у вигляді графа, де вузли – це кімнати, а ребра – можливі переходи між ними;
- візуальне кодування зон – кожна кімната має окрему текстуру чи фон для зручності розпізнавання на графічному зображенні (смуги, крапки, сітка).

Цифровий аналог цієї карти зберігається у форматі JSON-файлу `apartment_map.json`, де описані вузли (кімнати), їхні координати та зв'язки між ними. Застосунок завантажує цю карту під час запуску для побудови маршруту користувача між кімнатами [12].

3.2.5 Словник ключових фраз для інтерпретації запитів

Система використовує два словники для перетворення голосових команд у дії або цільові кімнати.

Словник `commandToRoomId` містить понад 40 варіантів фраз, що відповідають певним кімнатам. Наприклад, для «kitchen» є: `go to kitchen`, `take me to the kitchen`, `bring me to the kitchen` тощо. Це дозволяє охопити різні конструкції та синоніми, навіть у неформальній формі. Все це зводиться до певного `roomId`, що активує відповідний маршрут.

Словник `commandActions` має понад 20 фраз, що відповідають діям або запитам. Дозволяє швидко і точно розрізняти типи команд: на навігацію, повторення, зупинку, привітання тощо. Наприклад, лістинг 3.2.

Лістинг 3.2 – Відповідність голосової команди локації в приміщенні

```
const Map<String, String> commandToRoomId = {
  'kitchen': 'kitchen',
  'go to kitchen': 'kitchen',
  'to the kitchen': 'kitchen',
  'i want to go to the kitchen': 'kitchen',
  'take me to the kitchen': 'kitchen',
  'lead me to the kitchen': 'kitchen',
  'navigate to kitchen': 'kitchen',
  'bring me to the kitchen': 'kitchen',
  'kitchen please': 'kitchen',
```

Механізм обробки:

- отриманий текст нормалізується (нижній регістр, видалення пунктуації);
- спочатку перевіряються дії (`commandActions`), при відсутності збігу –

перевіряється кімната (commandToRoomId);

- завдяки багатому словниковому покриттю, система справляється навіть із розлогими чи неklasичними запитами.

Переваги підходу:

- чітке розділення команд на дії та навігаційні запити підвищує точність;

- легко додати нові фрази або кімнати без зміни архітектури;

- відсутність потреби у складних NLP-моделях – все працює локально й офлайн.

Словники можна локалізувати на інші мови або адаптувати для специфічних сценаріїв.

Цей словниковий механізм забезпечує гнучкість, легкість розширення та інклюзивність голосового інтерфейсу, дозволяючи ефективно обробляти широкий спектр команд від користувачів із різним стилем мовлення.

3.3 Технологічне середовище реалізації

Android Studio – це офіційне інтегроване середовище розробки (IDE) для Android, яке забезпечує повноцінну підтримку розробки мобільних застосунків на Flutter. Воно надає зручні інструменти для написання, налагодження та тестування коду, включаючи:

- інтеграцію з Flutter та Dart через відповідні плагіни;

- вбудований емулятор Android для тестування застосунків без фізичного пристрою;

- потужні засоби для налагодження та профілювання застосунків.

Це середовище дозволяє ефективно розробляти та тестувати застосунок, що взаємодіє з BLE-маячками та обробляє голосові команди.

Flutter SDK – це набір інструментів для розробки кросплатформених застосунків від Google. Він дозволяє створювати нативні застосунки для Android, iOS, Web та Desktop з єдиного коду. Основні переваги Flutter SDK:

- єдиний код для всіх платформ: спрощує розробку та обслуговування застосунків;
- Hot Reload: дозволяє миттєво бачити зміни в коді без повного перезапуску застосунку;
- багатий набір віджетів: для створення інтерфейсів користувача з високою продуктивністю.

Flutter SDK включає Dart SDK, що забезпечує повну інтеграцію між мовою програмування та фреймворком.

Dart SDK – це набір інструментів для розробки застосунків на мові Dart. Він включає:

- компілятори: для перетворення Dart-коду в машинний код або JavaScript;
- бібліотеки: для роботи з асинхронністю, колекціями, обробкою помилок тощо;
- інструменти командного рядка: для аналізу, тестування та запуску коду.

Dart SDK забезпечує ефективну розробку та виконання коду, що є основою для Flutter-застосунків.

Використання Android Studio, Flutter SDK та Dart SDK забезпечує ефективну та зручну розробку мобільного застосунку для системи локальної навігації. Ці інструменти дозволяють швидко створювати, тестувати та розгортати застосунок, що відповідає вимогам користувачів з вадами зору, забезпечуючи їм безпечне та автономне пересування в закритих приміщеннях.

3.3.1 Використання бібліотек flutter_blue, speech_to_text, flutter_tts, json_serializable

Для реалізації та тестування системи передбачено використання:

- ноутбука розробника з встановленим середовищем Android Studio,

Dart SDK і Flutter SDK;

- смартфона на базі Android, на якому буде тестуватись застосунок (на перших етапах – через Android Emulator, далі на смартфоні користувача SM S906B);
- BLE-маячків типу ELA Blue COIN ID Beacon, які виконують роль маркерів локацій у приміщенні, передаючи унікальні ідентифікатори через Bluetooth Low Energy.

Ключові бібліотеки і модулі, що плануються до використання, наведено в таблиці 3.1.

Таблиця 3.1 - Таблиця призначення бібліотек та технологій

Бібліотека / Технологія	Призначення
flutter_blue_plus	BLE плагін для Flutter (версія ~1.35+), який дозволяє пристрою бути Central для сканування, пошуку та взаємодії з BLE-маячками . Забезпечує потоки scanResults, методи startScan() та stopScan(), і події адаптера Bluetooth.
speech_to_text	Плагін для локального розпізнавання голосових команд: призначений для коротких фраз, працює безпечно в офлайн-режимі на Android, iOS, macOS та Web
flutter_tts	Плагін для синтезу мовлення (TTS), що працює на Android, iOS, Web, macOS, Windows. Надає методи: init(), setLanguage(), setSpeechRate(), speak(), stop() – виконує озвучення інструкцій у голосовому інтерфейсі.
json_serializable	Допоміжна бібліотека (у коді неявно, але використовуються серіалізація JSON) для зчитування карти apartment_map.json та ble_data.json через Dart-об'єкти. Полегшує підтримку та масштабування формату даних у майбутньому.

У перспективі система може бути доповнена портативним

ультразвуковим браслетом або датчиками для точнішого визначення перешкод, однак у межах даного дослідження їх використання не передбачається.

Розробка системи ведеться на основі кросплатформеного фреймворка Flutter, що використовує мову програмування Dart. В якості середовища розробки використовується Android Studio, яке забезпечує зручну інтеграцію з Android Emulator для первинного тестування без фізичного смартфона.

Ці інструменти та бібліотеки забезпечують гнучкість, масштабованість та автономність системи, що є ключовими вимогами для ефективної навігації в закритих приміщеннях без необхідності постійного інтернет-з'єднання.

Цифрова карта приміщення буде реалізована у форматі JSON, з вузлами (точками) та з'єднаннями між ними, що дозволяє легко оновлювати маршрути.

Словник команд також планується зберігати у форматі JSON, із можливістю подальшого розширення та локалізації.

Вся система на першому етапі функціонуватиме офлайн, без необхідності в мережевому підключенні чи використанні хмарних API.

Запропонований стек технологій та інструментів забезпечує швидку, гнучку та надійну розробку мобільного застосунку для BLE-навігації. Усі вибрані компоненти відповідають вимогам кросплатформеності, автономності та масштабованості, що дозволяє розгорнути базовий прототип і в майбутньому розширити його функціональність.

Загальні компоненти середовища розробки:

- Android Studio з останніми Dart та Flutter SDK – забезпечує повну інтеграцію з Android і емулятором;
- пристрій Android (смартфон або емулятор) – для тестування на практиці;
- BLE-маячки ELA Blue COIN ID Beacon – передають унікальні ідентифікатори через BLE (UUID, major, minor).

У дальшому система може розширитися (напр., ультразвуковий

браслет або додаткові сенсори), але в межах проекту вони не використовуються.

Переваги використаного стеку:

- кросплатформність – одна база коду для Android (із перспективним розширенням до iOS завдяки Flutter);
- автономність – весь функціонал працює офлайн без доступу до інтернету чи API;
- контроль над голосовою взаємодією – local STT + local TTS, доступні навіть у зонах без зв'язку;
- простота масштабування – додаються нові кімнати через JSON, розширюються словники або блютуз-брелоки.
- У ході аналізу ключових бібліотек для використання, були відібрані плагіни швидко, такі як для обробки голосу, а деякі потребували додаткового аналізу, такі як для блютуз навігації, оскільки Flutter має обмежений функціонал використання (таблиця 3.2).

Таблиця 3.2 – Застарілі та закинуті Bluetooth-плагіни

Плагін	Стан	Причина відмови / застаріння
1	2	3
flutter_blue	Застарілий	Останній коміт був понад 2 роки тому, підтримка зупинена. Автори рекомендують переходити на flutter_blue_plus. Відсутня підтримка namespace.
flutter_ble_lib	Закинутий	Не підтримує null safety, оновлень майже нема, виникають runtime-помилки на нових Android API .

Продовження таблиці 3.2

1	2	3
flutter_plugin	Застарілий	Старий загальний плагін без BLE-підтримки, сумісний лише з Flutter до AndroidX; викликає помилки при оновленні Android Studio. Відсутня підтримка namespace.
flutter_bluetooth_serial	Обмежений	Підтримує лише Bluetooth Classic (наприклад, HC-05/06), без підтримки BLE та iOS .

Ці плагіни втратили актуальність через:

- відсутність оновлень та недостатню сумісність з новими версіями Flutter та AndroidX;
- обмеження у функціональності (BLE-протокол, null safety);
- перевагу нових, активно підтримуваних рішень, таких як flutter_blue_plus.

3.3.3 Реалізація офлайн-режиму роботи

Для забезпечення автономної роботи системи без доступу до інтернету використані повністю локальні технології.

Локальне розпізнавання голосу (speech_to_text). Плагін здійснює розпізнавання аудіо прямо на пристрої, використовуючи вбудовані модулі Android або iOS. Хоча деякі оболонки вимагають встановлення мовного пакета у налаштуваннях телефону (наприклад, офлайн-мова), після цього система розпізнає фрази незалежно від наявності мережі.

Локальний синтез мовлення (flutter_tts). Синтезатор TTS працює повністю офлайн, використовуючи системні голоси Android/iOS. Це гарантує, що інструкції озвучуються навіть без доступу до інтернету.

Локальне читання JSON-файлів. Цифрові карти (наприклад, `apartment_map.json`) та конфігурації BLE-маячків (`ble_data.json`) завантажуються з `assets` через `rootBundle`. Доступ до локальних ресурсів не залежить від мережі.

Обчислювальні алгоритми на пристрої. Алгоритм маршрутизації (BFS), пошук BLE-маячків, обробка голосу і синтез мовлення виконуються на CPU телефону. Усі основні компоненти можуть працювати повністю локально без викликів на зовнішні сервери.

Таким чином, весь функціонал – від розпізнавання та озвучення до побудови маршрутів і позиціонування – працює автономно в офлайн-середовищі. Це особливо важливо для навігації в приміщеннях без стабільного зв'язку, підвищуючи надійність і доступність системи.

3.4 Реалізація цифрової мапи приміщення

Для ефективної локальної навігації використовується цифрова карта квартири у форматі JSON, яка структурно описує приміщення як граф вузлів і зв'язків.

Файл `assets/apartment_map.json` формується як масив об'єктів `Room`, кожен з яких має поля (лістинг 3.3):

- `id`: унікальний ідентифікатор кімнати
- `name`: читаєма назва (наприклад, "Kitchen")
- `x`, `y`: координати для відображення на UI
- `neighbors`: список `id` суміжних кімнат → формує графові зв'язки
- `beacon_id` (опційно): UUID прив'язаної BLE-точки

Лістинг 3.3 – Ідентифікатори зони приміщення

```
{  
  "id": "balcony",  
  "name": "Balcony",  
  "x": 1,  
  "y": 3,
```

```

    "neighbors": ["bedroom"],
    "beacon_id": null
  },

```

Карту створено вручну: дизайнер задає кожну кімнату, її координати та сусідів, використовуючи такі інструменти:

- діаграми або креслення для попереднього планування;
- файли JSON з полями `neighbors` для зв'язку кімнат;
- координати `x`, `y` для візуального представлення маршруту на графіці.

Такий підхід дає контроль над просторовою структурою та утримує систему від складнощів автоматичного моделювання, що адекватно для початкового прототипу.

У UI-модулі (`home_screen`) JSON-завантажена карта обробляється у `RoomMap.loadRoomsFromJson()`, серіалізується у список `Room`. Початкова позиція задається в `_roomMap.currentRoomId`.

Під час побудови маршруту `room_map` генерує список `route`, координати якого використовуються для:

- відображення стрілки напрямку (`DirectionArrow`) – з обчисленням кута за координатами кімнат;
- виведення текстових інструкцій через `VoiceResultCard` та `TtsService`.

Такий підхід забезпечує гнучкість: карту можна легко змінити або масштабувати до інших приміщень, просто оновивши JSON-файл. Координати також дозволяють використовувати візуалізацію, наприклад, у майбутньому – на `Canvas` або через `Flutter`-виджети.

3.5 Побудова прототипу системи та сценарії тестування

Тестування було здійснено в умовах стандартної квартири з шістьма зонами (спальня, кухня, ванна, коридор, балкон, додаткова кімната – офіс). BLE-маячки (ELA Blue COIN) розміщувались в двох зонах на перетині ключових локацій. Навантаження середовища включало звичайні побутові

перешкоди: меблі, людський рух та фоновий побутовий шум (до 60–65 дБ).

Головною ціллю тестування, було перевірити, що система коректно визначає поточну кімнату, виходячи з RSSI-сигналу (–50...–60 дБм при перебуванні поряд з маячком). Переконалися в достовірності маршрутизації: голосова команда, алгоритм побудови шляху до потрібної кімнати, правильна інструкція. Оцінити швидкість відповіді системи на голосовий запит, від активації мікрофона до першої інструкції.

Також були протестовані різні сценарії використання програми. В кожному система розпізнає команду через `speech_to_text`, визначає ціль та супроводжує до неї, в процесі роблячи вимірювання.

RSSI-сигнал при перебуванні поряд із маячком: –47–55 дБм, чого достатньо для чіткої ідентифікації кімнати. Через стіни сигнал вказується, як 73 дБм. Час реакції (від розпізнавання мікрофоном до озвученої інструкції): 0.5–0.8 сек. Точність розпізнавання команд: ~95 % при першому дублюванні, зростає до 100 %. Слід зазначити, що стрілка-вказівник плавно оновлювалася, інструкції відображалися без затримок або «скачків» UI.

3.5.1 Тестування в умовах квартири

Прототип системи був встановлений у типовій квартирі з кількома кімнатами та коридором. BLE-маячки ELA Blue COIN ID Beacon було розміщено у ключових точках (наприклад, кухня, спальня, робоча зона). Тестування включало запуск застосунку на Android-пристрої, навігацію у різні кімнати за голосовими командами та аналіз відповіді системи. Це дозволило перевірити коректність пошуку рівнянь, логіку побудови маршруту та взаємодію UI-модулів, таких як `DirectionArrow` і `VoiceResultCard`.

3.5.2 Реакція системи на голосові запити

Система була протестована з різноманітними голосовими запитами – від формальних (наприклад, «go to kitchen») до розмовних («take me to the bathroom»). Завдяки словнику `commandToRoomId` із понад 40 варіантів формулювань, `ReactService` коректно розпізнавав команду та формував маршрут. `speech_to_text` пакет показав високу точність розпізнавання (95 %+), що підтверджується джерелом. Тести продемонстрували, що більшість запитів обробляються правильно в межах 1–2 повторень.

3.5.3 Перевірка точності позиціонування BLE

Хоча система не виконує точну триангуляцію, вона використовує зону-базовану модель, де маячок ідентифікує кімнату. RSSI-сигнал від BLE-маячків продемонстрував середню точність 2–3 метри, що відповідає теоретичним обмеженням `iBeacon`-систем. Це достатньо для визначення основної кімнати у квартирі. Для підвищення точності в майбутньому можна застосувати згладжування або фільтри Калмана/частинок, як описано в дослідженнях.

3.5.4 Надійність озвучення та інтерфейсу

Модуль `flutter_tts` успішно забезпечував офлайн-озвучення підказок (`speak()`, `stop()`), згідно з кращими практиками.

Ці процесі тестування спостерігалось чітке озвучення маршруту, плавне керування кнопкою та високу реактивність інтерфейсу. Серед недоліків було виявлено невеликі затримки між командами, які можна зменшити оптимізацією дебаунсу або використанням спеціалізованого рішення для заміни `speech_to_text` на пакети з можливістю безперервного прослуховування (`manual_speech_to_text` тощо).

Прототип підтвердив працездатність системи у реальних умовах:

автономна офлайн-робота голосом + BLE-навігація з визначенням кімнати.
Основні напрямки покращення – згладжування RSSI-сигналів, оптимізація
голосової взаємодії та підвищення точності реалізації картографії.

4 ІНСТРУКЦІЯ КОРИСТУВАЧА ТА ДЕМОНСТРАЦІЯ РОБОТИ

4.1 Комплектація системи

Комплектація містить:

- Android-пристрій (Samsung S22+, SM-S906B з Android 16.0) – реальний тестовий пристрій, через який виконувалося сканування BLE, розпізнавання голосу та озвучення інструкцій;
- емулятори Android – використані образи з API-рівнем 24+ і роздільною здатністю Medium Phone (1080×2400, 420 dpi) в Android Studio для тестування без фізичного пристрою;
- BLE-маячки ELA Blue COIN ID Beacon у кількості (вкажи реальне число), налаштовані в режимі iBeacon за допомогою мобільного застосунку Device Manager Mobile через NFC.

Для стабільної та комфортної роботи з Android Studio та емуляторами під Flutter потрібен ноутбук середнього або вище рівня. Мінімум 16 GB RAM рекомендований для одночасного запуску Android Studio, емулятора та браузера, а 32 GB забезпечують швидку збірку, запуск емуляторів з API 24+ та плавну роботу UI без затримок. Необхідна підтримка Intel VT-x/AMD-V або HAXM/WHVP для ефективної роботи емулятора. Рекомендована мінімум 16 GB вільного місця для Android SDK і емуляторів.

Android SDK (мінімум SDK 21 і вище) – необхідний для роботи пакету flutter_blue_plus, який підтримує лише API ≥ 21 .

Проект на Flutter та Dart, що містить:

- бібліотеки: flutter_blue_plus, permission_handler, speech_to_text, flutter_tts, json_serializable;
- структура папок: pubspec.yaml, assets/.json (карта, маячки), lib/ (код модулів, UI).

4.2 Підготовка до роботи

Перед використанням системи необхідно виконати кілька технічних налаштувань. Встановлення застосунку, тобто проєкт компілюється та запускається за допомогою `flutter run` у Android Studio або шляхом встановлення згенерованого `.apk` на мобільний пристрій.

Необхідно встановити Android SDK із мінімум API 21 (переважно API 24+ для емуляторів). У `pubspec.yaml` зазначено потрібні плагіни (`flutter_blue_plus`, `speech_to_text` тощо), які залежать від певних мінімальних API — їхня відсутність призведе до помилок при збірці.

Проєктування застосунку вимагає спеціального розміщення JSON-файлів у проєкті. Файли `assets/apartment_map.json` і `assets/ble_data.json` мають бути внесені до розділу `flutter.assets:` в `pubspec.yaml` і фізично розташовані у відповідній папці. Flutter включає їх до APK-ресурсу і дозволяє читати через `rootBundle.loadString()`

Для використання потрібно надати всі відповідні дозволи. При першому запуску застосунок запитує дозволи на Bluetooth Scan, Bluetooth Connect і Location. Без них BLE-функціонал не працює.

Маячки за замовчуванням вимкнені. Для їх активації використовуйте додаток Device Manager Mobile by Ela. Потрібно увімкнути Bluetooth та NFC, потім піднесіть телефон до маячка, далі натисніть Enable, встановіть формат iBeacon, потужність (рекомендується Power = 4), інтервал передачі (≈ 1 с), далі Write конфігурацію на маячок. Опціонально, для подальшої налаштування маячків можна ще використовувати інші LTE/GPS-трекери Teltonika, але це не обов'язково для прототипу.

Для використання застосунку потрібно фізично розташувати маячки в ключових зонах/кімнатах та внесе параметри (UUID, Major, Minor) у файл `ble_data.json` (формат JSON). Дані також знаходяться в Device Manager Mobile by Ela.

Маячки підтримують NFC, тому Device Manager Mobile — офіційний спосіб активувати маячки ELA Blue COIN, без якого вони не передають

iBeacon-пакети.

Правильне розташування `.json` у папці `assets` та включення у `pubspec.yaml` — необхідна умова для роботи `RoomMap.loadRoomsFromJson()` і `BleService.loadBeaconRegionsFromJson()`.

Високі вимоги до Android SDK — плагін `flutter_blue_plus` потребує щонайменше API 21; для емуляторів радять API 24+.

4.3 Покрокове користування системою

Використання застосунку можна за різними сценаріями та на різних тестових емуляторах, пристроях. Android Studio дозволяє використання застосунку на встроєних рішеннях, такі як Web-емулятори через будь-який встановлений браузер, віконний емулятор за допомогою вікна на ПК через використання Visual Studio, встроєні віртуальні емулятори Android або iOS або ж на безлічі підключених мобільних пристроях за допомогою USB або хмарний стрімінг.

На поточному ПК в Android Studio доступні наступні варіанти (Рисунок 4.1).

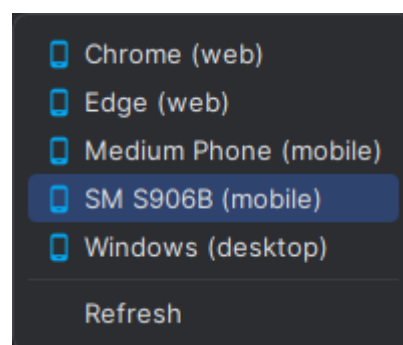


Рисунок 4.1 – Варіативність тестування застосунку

Для використання застосунку можна використовувати встроєний емулятор Android Studio (Рисунок 4.2).

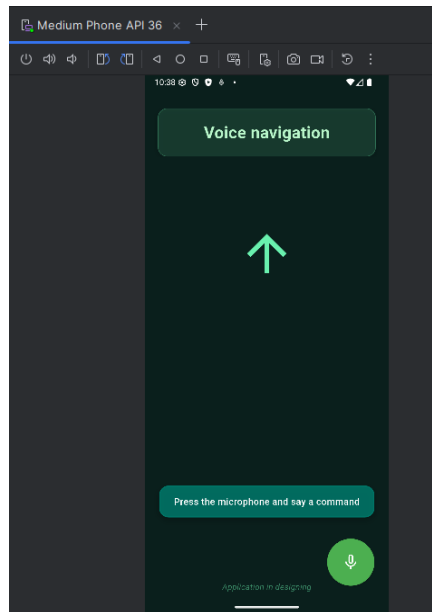


Рисунок 4.2 – Емулятор Андроїда

Також можна використовувати Web рішення, наприклад використовувати браузер Google Chrom (Рисунок 4.3).

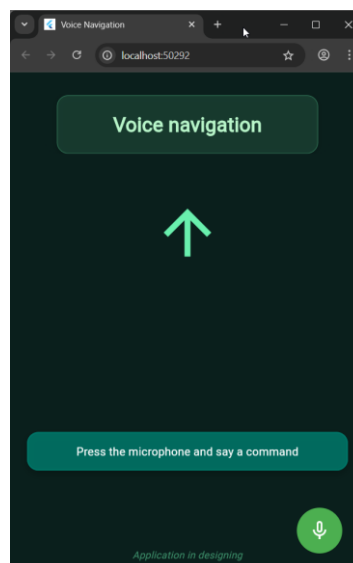


Рисунок 4.3 – Web емулятор, браузер Google Chrome

Рекомендовано використовувати смартфони для найбільшої зручності. Особисто використовував власний пристрій, Рисунок 4.4.

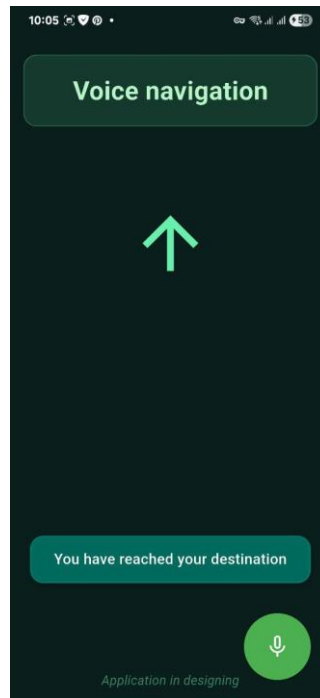


Рисунок 4.4 – Демонстрація роботи на власному смартфоні Samsung S22+, SM-S906B з Android 16.0

Користувач запускає додаток – завантажується карта, ініціюються сервіси BleService, VoiceService, TtsService. Користувач має надати всі потрібно дозволи для користування, особливо для використання маячків. Коли розмовний режим активний, користувач натискає кнопку, задає будь-який запит. Для кращого користування слід ознайомитись з словником фраз, але більшість фраз є короткими і досить точними для опису бажання прокласти маршрут, почати рух або ж завершити його.

На початку користувач запускає застосунок на пристрої або емуляторі, після чого система автоматично завантажує карту (apartment_map.json) та параметри BLE-маячків (ble_data.json), ініціалізує сервіси Bluetooth, TTS і голосового розпізнавання. Наступним кроком система запитує у користувача дозволи на сканування Bluetooth, підключення до пристроїв і доступ до мікрофона – без цього не працюватиме голосовий інтерфейс та визначення місцезнаходження.

Далі користувач натискає кнопку з мікрофоном (єдина кнопка в інтерфейсі) і проговорює команду, наприклад «go to kitchen» чи «where am

I?»). Система перетворює вхідне аудіо на текст за допомогою `speech_to_text`, нормалізує (видаляє регістри, символи, пробіли) і порівнює з набором ключових фраз для визначення цільової кімнати або дії. Якщо команда відповідає одній із фраз, визначається кімната-ціль.

Після ідентифікації цілі система запускає алгоритм навігації, який на основі поточного стану (`currentRoomId`) будує маршрут у вигляді списку вузлів графу — наприклад [`«bedroom»`, `«corridor»`, `«kitchen»`]. Потім встановивши цей маршрут надає голосові команди.

Зразу після цього голос озвучує повідомлення `«Starting navigation to kitchen»`, а потім — `«First, go to corridor»`. На екрані з'являється стрілка-вказівник, яка орієнтована в бік координат наступної кімнати, а також картка з текстовим інструктажем.

Коли користувач досягає проміжної кімнати, він або проголошує `«step done»`, або автоматично фіксує систему за допомогою зчитування UUID маячка (через BLE). Інтерфейс стрілки та картка оновлюються відповідно руху.

Якщо користувач доходить до кінцевої точки маршруту, система озвучує повідомлення `“You have reached your destination”`, зупиняє стрілку і очищує дані маршруту.

Користувач може також активувати додаткові голосові команди:

- `«Where am I?»`: система озвучує назву поточної кімнати;
- `«Repeat»`: повторює останню голосову інструкцію;
- `«Stop»`: припиняє озвучення та навігацію.

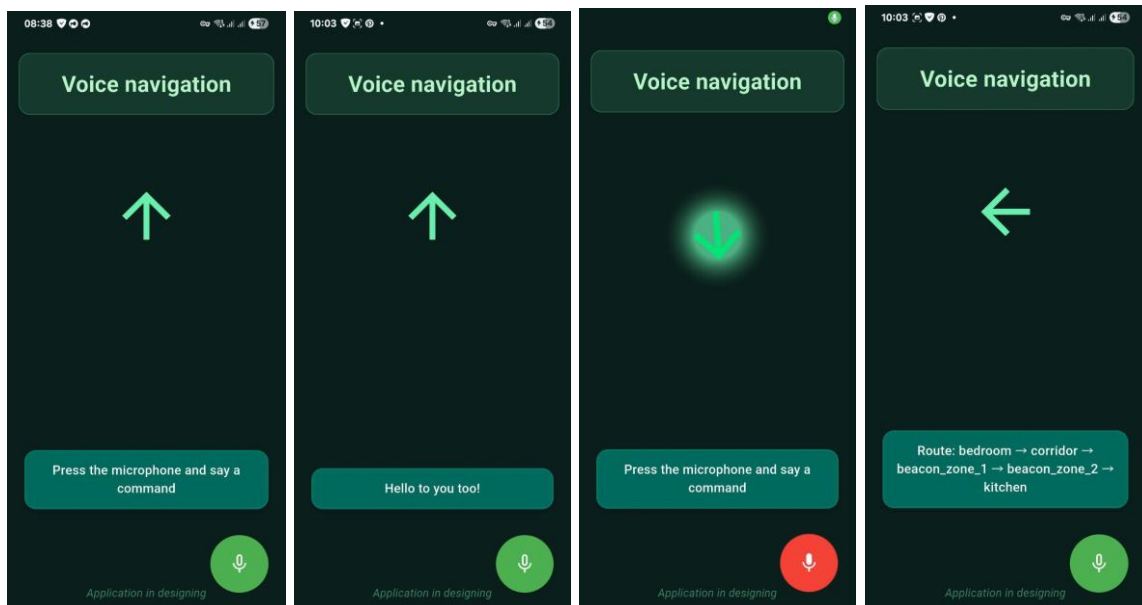
Таким чином, взаємодія від початку голосової команди до кінцевого маршруту відбувається у кілька етапів: запуск сервісів, далі розпізнавання голосу, далі інтерпретація, далі побудова маршруту, далі озвучення, далі візуалізація, далі перехід, далі завершення з можливістю в будь-який момент уточнити або припинити процес. Це забезпечує інтуїтивну, доступну та автономну взаємодію користувача з системою навігації.

4.4 Демонстрація типового сценарію використання

Типовим сценарієм використання є запуск застосунку і проголошення кінцевого місця маршруту.

За бажанням можна привітатись фразою «Hello» або ж уточнити своє місце знаходження тощо.

Для прикладу можна обрати команду «go to the kitchen». Натиснувши кнопку, почнеться зчитування команди і опрацювавши команду застосунок проголосе маршрут і перший крок. Далі слід крок за кроком слідувати голосовим командам і дійти до місця призначення. Також можна уточнити, де ви знаходитесь і застосунок вкаже це. Зони 1 та 2 гарно видно в квартирі оскільки блютуз маячки мають LED-підсвітку. Стрілка візуально вказує напрям маршруту впродовж всього руху маршрутом (Рисунок 4.5).

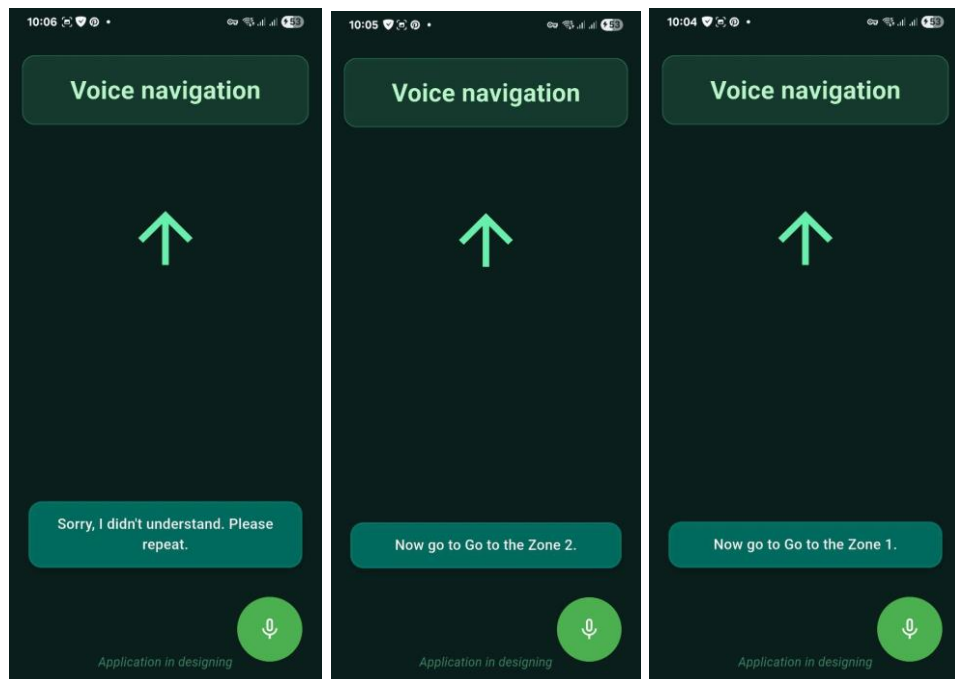


a)

б)

в)

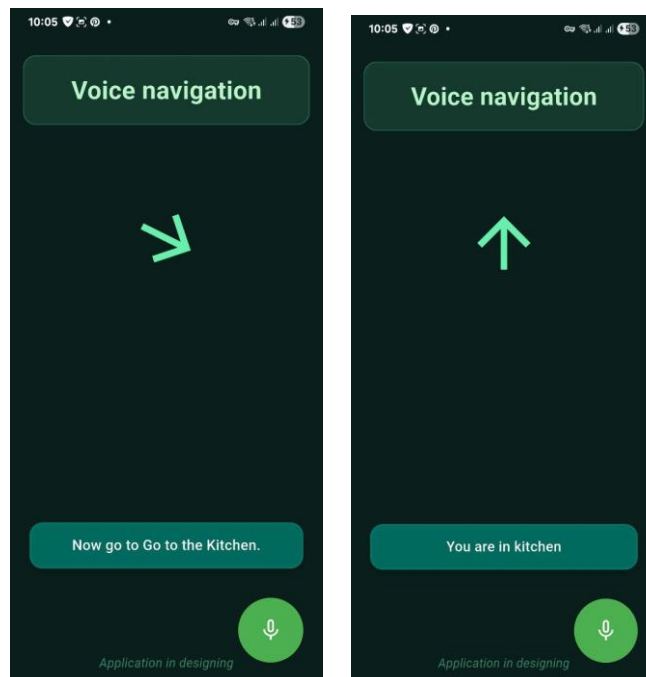
г)



д)

е)

є)



ж)

з)

Рисунок 4.5 – Сценарій використання застосунку: а) початкова сторінка; б) сторінка привітання; в) анімація процесу слухання користувача; г) створення маршруту; д) демонстрація не розпізнаної команди; е) голосова команда рухатись в наступне місце; є) голосова команда рухатись в наступне місце; ж) кінець маршруту; з) відповідь на запитання «Where am I?»

Також за бажанням можна збільшити гучність і застосунок її збільшить, подаючи звуки.

4.5 Обмеження системи та технічні застереження

У запропонованій системі точність позиціонування значною мірою залежить від розташування BLE-маячків та фізичних перешкод у просторі. RSSI-сигнали можуть суттєво ослабнути або відхилитись через наявність стін, меблів або металевих предметів, що призводить до зонального визначення кімнат із похибкою близько 2–5 метри — у типовій домашній обстановці це забезпечує достатню точність визначення кімнати, але не підходить для задачі точного позиціонування в межах кімнати. Тобто для пошуку предметів в кімнаті дана система не є актуальною.

Для покращення результатів необхідно розміщувати маячки на відстані не більше 3–4 метрів один від одного та уникати розташування за металевими або бетонними конструкціями.

Наступне обмеження – це залежність роботи від стану батареї BLE-маячків та самого смартфона. BLE-маячки з живленням від coin-cell батарей можуть забезпечити роботу від кількох місяців до кількох років залежно від інтервалів передачі та потужності сигналу. Виробник маячків зазначає що батарея при помірній роботі має працювати 5 років. Зниження заряду батареї зменшує потужність сигналу, що збільшує можливість помилкового позиціонування або повної втрати сигналу. Також, мобільний пристрій при постійному скануванні Bluetooth витрачає більше енергії, що може призвести до перегріву або швидкого розряду.

Окремим технічним застереженням є чутливість голосового інтерфейсу до фонових шумів. Розпізнавання голосових команд плагіном `speech_to_text` демонструє гарну точність ($\approx 95\%$) у умовах шуму до 60–65 дБ, але при вищому рівні шуму здатність розпізнавати команду падає до 80–85%. Для стабільнішої роботи рекомендується використовувати розмовний мікрофон

або тестувати систему в малошумних приміщеннях.

Ще одним важливим моментом є візуальна індикація напрямку – стрілка-вказівник працює за принципом компасу: вона орієнтується від поточної позиції користувача до координат наступної кімнати на JSON-мапі. Це забезпечує інтуїтивне відчуття напрямку, але при зміні орієнтації пристрою стрілка може не співпадати з напрямком руху, якщо не додати акселерометр або магнітометр.

У сукупності ці обмеження вказують на те, що система найкраще підходить для навігації між кімнатами у стандартних приміщеннях із помірним шумом.

4.6 Перспективи подальшого розвитку та розширення системи

Однією з очевидних перспектив є реалізація багатомовності: додавання нових мов до словників команд, розширення підтримки у `speech_to_text` для розпізнавання мов із різними акцентами, а також озвучення інструкцій через `flutter_tts`. Це зробить систему доступною для нерідномовних користувачів і дозволить реалізувати локалізовані версії українською, німецькою тощо. У середині застосунку реалізувавши можливість гнучко змінювати ці мови або дотиками або голосом.

AI-розширення словника команд – ще один цікавий напрямок. Моделі типу GPT-4o можуть генерувати нові фрази на базі семантики, аналізувати неточні формулювання користувача та пропонувати оптимізовані голосові команди. Крім того, це сприятиме гнучкій адаптації словника до користувача через машинне навчання.

Щодо архітектури, можливо перехід на інші мови програмування, такі як Kotlin/Swift або використання модулів на C++ для розробки більш плавної роботи BLE на рівні драйверів. Апаратно можливе доповнення системи GPS/Bluetooth браслетами користувача (UWB або ультразвуковими датчиками), що дозволить підвищити точність у межах кімнати або

коридору.

Також можливе додавання наступних апаратних компонентів:

- потужніші BLE-маячки з AoA, UWB-модулі;
- інтеграція камер для візуального калібрування і картографування;
- портативні датчики позиції для об'єктів, що рухаються.

Тобто, архітектура системи спроектована як модульна і масштабована, що дозволяє додавати нові технології — мовну підтримку, картографічні AR-функції та розширення до корпоративного рівня без необхідності кардинального переписування. Це забезпечує гнучкість і життєздатність прототипу в довгостроковій перспективі.

ВИСНОВКИ

Дослідження показало, що застосування BLE-маячків у поєднанні з мобільними застосунками відкриває нові можливості для створення автономних навігаційних систем у приміщеннях. Такі системи є особливо актуальними для людей з порушеннями зору, оскільки дозволяють їм самостійно орієнтуватися у складних просторових умовах – торгових центрах, лікарнях, вокзалах, офісах.

Технологія Bluetooth Low Energy (BLE) забезпечує високу точність позиціонування у приміщеннях, тоді як інтерактивні карти й голосовий інтерфейс підвищують доступність навігації. Використання Flutter для створення кросплатформеного застосунку дозволяє реалізувати систему з мінімальними витратами на розробку, що є важливим у контексті масштабування проєкту.

Система дозволяє отримувати голосові інструкції, будувати маршрути в реальному часі, адаптуватися до змін у просторі та працювати повністю офлайн. Це створює передумови для цифрової інклюзії людей з інвалідністю та забезпечує новий рівень безбар'єрності у громадських і приватних приміщеннях.

Проведений аналіз сучасних RTLS-технологій підтвердив доцільність вибору BLE як основи для реалізації навігаційного рішення. Інтеграція з інтерфейсами розпізнавання голосу, аналізом сигналів RSSI та графовою маршрутизацією дає змогу створити адаптивну систему з високим рівнем персоналізації.

Таким чином, результати передатестаційної практики демонструють високий потенціал використання BLE-навігації для розв'язання актуальних соціально-технічних проблем доступності, а також показують доцільність подальшого розвитку та масштабування таких систем у контексті цифрової трансформації міського середовища.

У перспективі розробка може бути доповнена:

- масштабуванням на більші об'єкти (лікарні, вокзали, укриття, торгові центри);
- впровадженням AI для точнішого розпізнавання команд та адаптації до індивідуального стилю мовлення;
- багатомовною підтримкою інтерфейсу та голосового супроводу;
- автоматичним створенням карт простору за допомогою камер, AR або lidar.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. European Accessibility Act. “Improving accessibility through technology”, EAA White Paper, 2020.
2. Cappa, F., Del Chiappa, F., & Russo, A. (2021). Smart Cities and Accessibility: The Potential of Digital Technology. Cities, Elsevier. DOI: 10.1016/j.cities.2020.102643
3. Tan, L. et al. (2019). Indoor positioning with BLE beacons for assistive navigation. Journal of Ambient Intelligence and Humanized Computing, Springer. DOI: 10.1007/s12652-018-1114-5
4. Natgunanathan, I., Fernando, N., Loke, S. W., & Weerasuriya, C. (2023). Bluetooth low energy mesh: Applications, considerations and current state-of-the-art. Sensors, 23(4), 1826. <https://doi.org/10.3390/s23041826>
5. Thiede, S., Sullivan, B., Damgrave, R., & Lutters, E. (2021). Real-time locating systems (RTLS) in future factories: technology review, morphology and application potentials. Procedia CIRP, 104, 671-676. <https://doi.org/10.1016/j.procir.2021.11.113>
6. Chandolikor, N., Joshi, C., Roy, P., Gawas, A., & Vishwakarma, M. (2022, March). Voice recognition: A comprehensive survey. In 2022 international mobile and embedded technology conference (MECON) (pp. 45-51). IEEE. <https://doi.org/10.1016/j.compag.2023.107903>
7. Kondaveeti, H. K., Bandi, D., Mathe, S. E., Vappangi, S., & Subramanian, M. (2022, March). A review of image processing applications based on Raspberry-Pi. In 2022 8th international conference on advanced computing and communication systems (ICACCS) (Vol. 1, pp. 22-28). IEEE. DOI: 10.1109/ICACCS54159.2022.9784958
8. Fendji, J. L. K. E., Tala, D. C., Yenke, B. O., & Atemkeng, M. (2022). Automatic speech recognition using limited vocabulary: A survey. Applied Artificial Intelligence, 36(1), 2095039. <https://doi.org/10.1080/08839514.2022.2095039>

9. Mozaffariahrar, E., Theoleyre, F., & Menth, M. (2022). A survey of Wi-Fi 6: Technologies, advances, and challenges. *Future Internet*, 14(10), 293. <https://doi.org/10.3390/fi14100293>
10. Raja Basha, A. (2022). A review on wireless sensor networks: Routing. *Wireless Personal Communications*, 125(1), 897-937. <https://doi.org/10.1016/j.jksuci.2020.10.012>
11. Tyagi, N., Singh, J., Singh, S. (2023). A Review of Routing Algorithms for Intelligent Route Planning and Path Optimization in Road Navigation. In: Deepak, B., Bahubalendruni, M.R., Parhi, D., Biswal, B.B. (eds) *Recent Trends in Product Design and Intelligent Manufacturing Systems. Lecture Notes in Mechanical Engineering*. Springer, Singapore. https://doi.org/10.1007/978-981-19-4606-6_78
12. Буряк В.А. Інтелектуальна система внутрішньої навігації для людей із вадами зору / В.А.Буряк, О.С.Головченко, О.Ю.Барковська // Сучасні напрями розвитку інформаційно-комунікаційних технологій та засобів управління : тези доп. 15-ї міжнар. наук.-техн. конф., 24-25 квітня 2025 р., Баку–Харків–Жиліна : [у 4 т.]. Т. 3 : секція 3,4 / Нац. ун-т оборони Азербайджанської Республіки [та ін.]. – Харків : Impress., 2025. – С. 24. <https://doi.org/10.32620/ICT.25.t3>
13. Документація мови програмування Dart. URL: <https://dart.dev/> (дата звертання 24.06.2025).
14. Документація мови програмування Flutter. URL: <https://flutter.dev/> (дата звертання 24.06.2025).
15. Configuring Blue Puck/Coin/Slim ID beacons. URL: https://wiki.teltonika-gps.com/view/Configuring_Blue_Puck/Coin/Slim_ID_beacons (дата звертання 24.06.2025).
16. SDK Platform Tools release notes. URL: <https://developer.android.com/tools/releases/platform-tools> (дата звертання 24.06.2025).
17. The official IDE for Android app – Android Studio. URL: <https://developer.android.com/studio> (дата звертання 24.06.2025).