

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»
спеціальності 126 «Інформаційні системи та технології»
на тему: «Робототехнічна сенсорна система моніторингу рослинності»

Виконала:

студентка IV курсу, групи ІК-12

Ковтун Дарина Валеріївна _____

Керівник:

Доцент, к.ф-м.н.,

Гавриленко Олена Валеріївна _____

Рецензент:

Доцент кафедри ІІІ, к.т.н., доцент

Олійник Юрій Олександрович _____

Засвідчую, що у цьому дипломному проєкті немає запозичень з праць інших авторів без відповідних посилань.

Студентка _____

Київ – 2025 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

« ___ » _____ 2025 р.

ЗАВДАННЯ
на дипломний проєкт студентці
Ковтун Дарині Валеріївні

1. Тема проєкту «Робототехнічна сенсорна система моніторингу рослинності», керівник проєкту Гавриленко Олена Валеріївна, к.т.н., доцент, затверджені наказом по університету від «23» травня 2025 р. № 1705-с.
2. Термін подання студентом проєкту: 03.06.25
3. Вихідні дані до проєкту: система повинна фіксувати стан рослин, передавати ці дані у додаток, забезпечувати можливість візуалізації цих даних, надавати можливість керувати системою.
4. Зміст пояснювальної записки:
5. Перелік графічного матеріалу: структурна схема, архітектурна схема, блок-схема алгоритму системи, діаграма прецедентів.
6. Дата видачі завдання 15.04.25

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Ознайомлення з ТЗ	15.04.25	
2	Аналіз предметної області	20.04.25	
3	Розробка функціональної моделі	21.04.25	
4	Огляд існуючих аналогів	25.04.25	
5	Формування цілей і задач розробки	26.04.25	
6	Розробка дизайну ПЗ	01.05.25	
7	Дослідження і розробка моделей для бази даних	05.05.25	
8	Розробка алгоритму генерації аудіосупровіду	7.05.25	
9	Розробка клієнтської частини ПЗ (iOS застосунок на SwiftUI)	10.05.25	
10	Тестування ПЗ, інтеграція з Firebase	15.05.25	

Студентка

Керівник

Дарина КОВТУН

Олена ГАВРИЛЕНКО

АНОТАЦІЯ

Робототехнічна сенсорна система моніторингу рослинності в домашніх умовах.

Проект містить 64 с. тексту, 10 рисунків, 6 таблиць, посилання на 15 літературних джерел, додатки та 4 конструкторські документи.

РОБОТОТЕХНІЧНА СИСТЕМА, МОБІЛЬНИЙ ЗАСТОСУНОК, ІОТ, FIREBASE, СЕНСОРИ, SWIFTUI.

Дипломний проєкт присвячено розробці робототехнічної сенсорної системи для автоматизованого моніторингу стану кімнатних рослин, зокрема троянд і тюльпанів, у домашніх умовах.

Мета розробки – створення програмної архітектури системи моніторингу, яка забезпечує зчитування, передавання, зберігання та візуалізацію даних про стан рослин за допомогою сенсорів, мікроконтролера, хмарної інфраструктури та мобільного застосунку.

У дипломному проєкті проаналізовано сучасні підходи до моніторингу рослинності, обґрунтовано вибір робототехнічної платформи та сенсорної моделі, розроблено архітектуру взаємодії компонентів системи, мобільний застосунок на iOS із використанням SwiftUI та реалізовано обмін даними через Firebase. Проведено імітаційне тестування, яке підтвердило працездатність і масштабованість запропонованої системи.

Отримані результати можуть бути використані як основа для впровадження систем автоматизованого моніторингу в побутових умовах, а також як прототип для розширення системи в аграрному або освітньому середовищі.

SUMMARY

Robotic Sensor System for Monitoring Indoor Vegetation.

The project includes 64 pages of text, 10 figures, 6 tables, references to 15 literary sources, appendices, and 4 design documents.

Keywords: robotic system, mobile application, iot, firebase, sensors, swiftui.

The bachelor's thesis is dedicated to the development of a robotic sensor system designed for automated monitoring of indoor plants, specifically roses and tulips, under household conditions.

The objective of the project is to design a software-based architecture for monitoring systems that enables the collection, transmission, storage, and visualization of plant condition data using sensors, a microcontroller, cloud infrastructure, and a mobile application.

The project analyzes existing approaches to vegetation monitoring, justifies the selection of a robotic platform and sensor model, develops a component interaction architecture, and implements a fully functional iOS mobile application using SwiftUI. Data exchange is realized through Firebase. Simulated testing was conducted to validate the functionality, responsiveness, and scalability of the proposed solution.

The results obtained may serve as a foundation for implementing automated plant monitoring systems in everyday use, and as a prototype for further development in agricultural or educational environments.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	ІК12.120БАК.006 ПЗ	Пояснювальна записка	64		
6	A3	ІК12.120БАК.006 Д1	Робототехнічна сенсорна система	1		
7			моніторингу рослинності.			
8			Структурна схема			
9	A3	ІК12.120БАК.006 Д2	Робототехнічна сенсорна система	1		
10			моніторингу рослинності.			
11			Архітектурна схема			
12	A3	ІК12.120БАК.006 Д3	Робототехнічна сенсорна система	1		
13			моніторингу рослинності.			
14			Блок-схема алгоритму системи			
15	A3	ІК12.120БАК.006 Д4	Робототехнічна сенсорна система	1		
16			моніторингу рослинності.			
17			Діаграма прецедентів			
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						
28						

ІК12.120БАК.006 ТП

		№ докум.	Підпис				
Розробив	Ковтун			Робототехнічна сенсорна система моніторингу рослинності Відомість дипломного проекту	Літ.	Арк.	Аркушів
Перевірив	Гавриленко					1	1
					КПІ ім. Ігоря Сікорського Група ІК-12		
Затв.							

**Пояснювальна записка
до дипломного проєкту
на тему: «Робототехнічна сенсорна система моніторингу
рослинності»**

Київ – 2025 року

ЗМІСТ

ВСТУП.....	5
1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	8
1.1 Актуальність застосування сенсорних систем у моніторингу рослинності оптимальних умов для їх росту.....	8
1.2 Огляд існуючих систем моніторингу стану рослин	8
1.3 Складові сенсорної системи моніторингу	11
1.4 Хмарні сервіси та мобільні додатки в екосистемах моніторингу.....	13
1.5 Проблеми та виклики впровадження IoT у моніторинг рослин	14
Висновок до розділу 1.....	16
2 АНАЛІЗ І ПРОЄКТУВАННЯ РОБОТОТЕХНІЧНОЇ СЕНСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РОСЛИННОСТІ.....	18
2.1 Аналіз предметної області	18
2.2 Огляд технічних компонентів робототехнічної сенсорної системи.....	19
2.3 Вибір апаратної платформи	20
2.4 Вибір архітектури взаємодії компонентів.....	21
2.5 Розробка структурної та архітектурної схем системи.....	23
2.5.1 Потік обробки даних	25
2.5.2 Розширюваність та безпека системи	27
Висновок до розділу 2.....	28
3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ	30
3.1 Архітектура програмного забезпечення	30
3.1.1 Структура архітектури системи.....	31
3.1.2 Обґрунтування вибору інструментів.....	31
3.1.3 Візуальні та функціональні компоненти застосунку.....	34
3.1.4 Безпека, надійність та масштабованість	34

					ІК12.120БАК.006 ПЗ		
		№ докум.	Підпис				
Розробив	Ковтун Д.В.			Робототехнічна сенсорна система моніторингу рослинності Пояснювальна записка	Літ.	Арк.	Аркушів
Перевірив	Гавриленко О.В.					2	64
					КПІ ім. Ігоря Сікорського		
Затв.					Група ІК-12		

3.2	Firestore як основа зберігання та синхронізації даних	35
3.2.1	Структура бази даних Firestore	37
3.2.2	Передача та обробка даних	39
3.2.3	Реакція на критичні зміни	41
3.2.4	Безпека даних і обмеження доступу	42
3.4	Мобільний застосунок на iOS: структура, екрани, навігація	43
3.5	UX/UI-дизайн мобільного додатку	45
	Висновок до розділу 3	47
4	ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА ТЕСТУВАННЯ	49
4.1.	Методика експериментальної перевірки функціонування системи	49
4.2	Аналіз результатів роботи системи в реальних умовах	51
4.3	Аналіз стабільності передачі даних у системі моніторингу	53
4.4	Статистична візуалізація даних та інтерфейсу мобільного застосунку	54
	Висновок до розділу 4	58
	ВИСНОВКИ	60
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
	ДОДАТОК А	634

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

IoT (Internet of Things) — Інтернет речей, концепція з'єднання пристроїв через мережу для автоматичного збору та обміну даними.

ESP32 — мікроконтролер з підтримкою Wi-Fi і Bluetooth, що використовується для зчитування та передачі даних із сенсорів.

Firebase — хмарна платформа Google для зберігання, обробки даних, автентифікації та надсилання сповіщень.

Firestore — NoSQL база даних від Firebase для зберігання структурованої інформації в режимі реального часу.

DHT22 — цифровий сенсор для вимірювання температури та вологості повітря.

BH1750 — сенсор освітленості, що використовується для вимірювання рівня світла.

SwiftUI — сучасний фреймворк для створення інтерфейсів мобільних застосунків на iOS.

Wi-Fi — технологія бездротового зв'язку для передачі даних між пристроями.

BLE (Bluetooth Low Energy) — протокол енергоефективної бездротової передачі даних на короткі відстані.

UAV (Unmanned Aerial Vehicle) — безпілотний літальний апарат, зокрема дрон, застосовується для автономного моніторингу.

JSON (JavaScript Object Notation) — формат обміну даними між пристроями у структурованому вигляді.

MVVM (Model-View-ViewModel) — шаблон архітектури застосунків, що використовується у розробці інтерфейсу.

FCM (Firebase Cloud Messaging) — сервіс для надсилання push-сповіщень на мобільні пристрої.

API (Application Programming Interface) — інтерфейс для взаємодії між програмними компонентами.

LiPo — літій-полімерний акумулятор, джерело живлення для дронів.

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

ВСТУП

Сучасний світ стрімко розвивається у напрямку цифровізації й автоматизації повсякденних процесів, і догляд за рослинами – не виняток. Останнім часом домашнє квітникарство стає дуже популярним та постає необхідність у створенні інтелектуальних систем, які здатні спростити моніторинг стану рослин та забезпечити своєчасний догляд за ними.

Домашні рослини, зокрема троянди та тюльпани, вимагають особливої уваги: контроль за вологістю ґрунту, рівнем освітленості, температурними умовами тощо. Якщо невчасно помітити якісь зміни – це може призвести до втрати їх декоративних властивостей або загибелі.

Аналізуючи це все, розроблення робототехнічної сенсорної системи моніторингу рослинності є актуальним, що дозволить автоматизувати процес спостереження за станом кімнатних рослин. У даній системі включено використання дрона, оснащеного сенсорами, який здійснює огляд домашніх квітів, збирає дані та передає їх до застосунку. Цей підхід дозволяє відстежувати параметри росту та здоров'я рослин у реальному часі, отримувати сповіщення про необхідність у догляді, а також переглядати статистику змін у зручному форматі.

Треба зазначити, що така система є робототехнічною за своєю суттю, адже поєднує в собі апаратну (дрон, сенсори, камера) та програмну частину (мобільний застосунок з аналітичними алгоритмами). Це дозволяє не лише автоматизувати процес спостереження, але й мінімізувати участь людини в рутинних задачах догляду.

В останні роки домашнє квітникарство стало знов популярним серед мешканців міст, що виявляє потребу в нових технологіях для ефективного моніторингу стану кімнатних рослин. Класичні методи догляду потребують постійної уваги: контроль за вологістю ґрунту, рівнем освітлення, температурою та іншими чинниками, які критично впливають на ріст і здоров'я рослин.

Якщо пропустити зміни у зовнішньому вигляді або параметрах часто призводять до втрати привабливості чи навіть загибелі рослини. У зв'язку з цим,

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

розроблення робототехнічної сенсорної системи моніторингу, яка здатна автоматично фіксувати параметри навколишнього середовища та реагувати на зміни, набуває особливої актуальності.

Технології, які використовуються для моніторингу або керування системами, на сьогоднішній день, потребують складного встановлення, налаштування. Зазвичай це вимагало наявності окремих, підключень, серверів, контролерів що потребують значні витрати часу та ресурсів. Але так як технології розвиваються, то з'явилися нові рішення - мобільні застосунки, системи на базі дронів та сенсорів, які мають нижчий поріг входу, є більш доступними для користувачів та не потребують спеціального навчання чи дороговартісного обладнання.

Технології комп'ютерного зору та машинного навчання дозволяють системам не лише фіксувати, а й інтерпретувати зміни у стані об'єктів – наприклад, виявляти ознаки в'янення, зміни кольору листя, затінення тощо. Завдяки таким алгоритмам можна не просто зчитувати параметри з сенсорів, а й візуально аналізувати стан рослин з камери, що відкриває перспективу для створення автономної мобільної системи спостереження.

У межах цієї роботи буде розроблено робототехнічну сенсорну систему на основі безпілотного літального апарата (дрона), який здійснюватиме періодичне сканування домашніх квітів, збиратиме дані про стан кожної рослини за допомогою сенсорів і камери, а також передаватиме цю інформацію в мобільний застосунок. Це дозволить користувачу переглядати параметри росту рослин, історію змін, отримувати сповіщення про потребу у догляді, а також бачити рекомендації щодо покращення умов утримання.

У свою чергу, сам застосунок буде реалізований з використанням сучасних технологій розробки під iOS-платформу із збереженням даних у хмарному середовищі (Firebase Realtime Database / Firestore). Це дозволить підтримувати персоналізований облік рослин кожного користувача, масштабувати систему, а також впроваджувати додаткові модулі у майбутньому.

Метою дипломного проєкту є підвищення ефективності та простоти догляду за домашніми рослинами шляхом створення робототехнічної сенсорної системи

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

моніторингу домашньої рослинності на прикладі троянд і тюльпанів із використанням мобільного застосунку та сучасних сенсорних технологій.

Для досягнення цієї мети у дипломного проєкті необхідно вирішити такі завдання:

— провести аналіз особливостей вирощування та моніторингу стану кімнатних рослин;

— розглянути сучасні технології сенсорного моніторингу та бездротової передачі даних;

— запропонувати архітектуру робототехнічної системи моніторингу рослин з використанням дрона та мобільного застосунку;

— реалізувати мобільний застосунок для збору, обробки, візуалізації та аналізу даних;

— визначити переваги, недоліки та перспективи розвитку запропонованої системи.

					ІК12.120БАК.006 ПЗ	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

1 ТЕОРЕТИЧНІ ОСНОВИ ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Актуальність застосування сенсорних систем у моніторингу рослинності оптимальних умов для їх росту.

Сенсорні системи допомагають здійснювати постійний моніторинг життєво важливих параметрів, такі як: температура, вологість ґрунту, освітленість, рівень рН тощо [1]. Вони дозволяють швидко виявляти проблеми, які можуть трапитись з квітами та реагувати на них у режимі реального часу. Особливої актуальності такі системи набувають при догляді за кімнатними декоративними рослинами, зокрема за такими вибагливими видами, як троянди та тюльпани. Ці рослини мають особливі вимоги до мікроклімату, і навіть незначне відхилення від оптимальних умов може негативно вплинути на їхній розвиток і цвітіння.

Поєднання сенсорних технологій із робототехнічними платформами, дронами, відкриває нові можливості для створення мобільних автономних систем моніторингу [2]. Завдяки використанню дронів можна охоплювати більшу площу або складні для доступу місця, також підвищити точність зчитування даних.

В світі сучасних технологій є надзвичайно важливим можливість інтегрувати сенсорні пристрої з мобільними застосунками, які забезпечують зручний інтерфейс для користувача, аналізувати дані, перегляд історію змін, отримувати сповіщення та рекомендації по догляду за квітами. Це робить догляд за рослинами більш доступним, системним і навіть привабливим для користувача без спеціальних агрономічних знань.

Розробка сенсорної системи моніторингу рослинності з використанням дронів та інтеграцією з мобільним застосунком є актуальним та інноваційним напрямом, який має практичне значення і високий потенціал застосування в реальних умовах.

1.2 Огляд існуючих систем моніторингу стану рослин

За весь час на ринку з'явилося чимало рішень, які спрямовані на

					ІК12.120БАК.006 ПЗ	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

автоматизований моніторинг рослинності, як у промисловому сільському господарстві, так і в повсякденному використанні [15]. Такий інтерес до подібних технологій пояснюється прагненням підвищити ефективність вирощування рослин, зменшити витрати ресурсів і мінімізувати людський фактор у процесах догляду.

Більшість сучасних систем побудовані за схожим принципом: вони поєднують у собі різні типи сенсорів (вологості, температури, освітленості тощо), мікроконтролери, що обробляють сигнали, модулі бездротового зв'язку для передавання даних, а також програмні компоненти у вигляді мобільних або веб-застосунків. У результаті такого підходу формується цілісна інтегрована платформа, яка не лише забезпечує безперервний збір екологічних даних у режимі реального часу, а й дає змогу візуалізувати ці дані, аналізувати їх та оперативно інформувати користувача про важливі або критичні зміни в умовах середовища.

Схожі рішення постійно вдосконалюються, стають дедалі доступнішими та простішими у використанні. Це відкриває нові можливості для широкого впровадження технологій «розумного землеробства» не тільки в промисловому агровиробництві, а й у малих фермерських господарствах чи навіть на приватних ділянках. Завдяки гнучкості та адаптивності такі системи можуть ефективно налаштовуватись під конкретні умови вирощування та індивідуальні потреби користувача, закладаючи підґрунтя для створення більш масштабованих і високотехнологічних рішень у сфері агромоніторингу. Побутові системи моніторингу

1. Xiaomi Mi Flower Care – один із найвідоміших побутових сенсорів, який вимірює вологість ґрунту, освітлення, температуру та рівень поживних речовин. Пристрій з'єднується зі смартфоном через Bluetooth і надає прості поради щодо догляду.

2. Parrot Pot – інтелектуальний вазон, який самостійно поливає рослину залежно від виміряних параметрів. Окрім поливу, він контролює рівень вологи, температуру та освітленість, має мобільний застосунок із рекомендаціями.

3. PlantLink – система із сенсорами, які передають дані на базову станцію, що

					ІК12.120БАК.006 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

з'єднується із хмарним сервісом. Вона орієнтована більше на зовнішні сади, але може використовуватися і в домашніх умовах.

Академічні та дослідницькі проекти

У багатьох університетах розробляються власні системи для агромоніторингу з більшим акцентом на IoT (Internet of Things), машинне навчання та робототехніку [8]. Наприклад:

– Plant Monitoring Systems based on Arduino / ESP32 – найпростіші варіанти, які складаються із датчиків вологості, температури та Wi-Fi-модуля. Дані надсилаються до Blynk, Thingspeak або Firebase.

– AgriBot / AgriDrone – концепти дронів, що виконують діагностику полів і передають зображення та сенсорні дані до аграрних платформ [9]. У таких проєктах застосовуються гіперспектральні камери, LIDAR або термодатчики.

У таблиці 1.1 наведено порівняльну характеристику існуючих систем моніторингу стану рослинності

Таблиця 1.1 – Порівняльна характеристика існуючих систем моніторингу стану рослинності

Назва системи	Переваги	Недоліки
Xiaomi Mi Flower Care	Простота, ціна, стабільна робота	Лише Bluetooth, невеликий радіус дії
Parrot Pot	Автополив, мобільний додаток	Висока вартість, обмежений об'єм ґрунту
Arduino IoT-системи	Гнучкість, можливість кастомізації	Вимагають технічних знань
AgriDrone	Великий охоплення, високоточна аналітика	Не пристосовані для домашніх умов, складні

На рисунку 1.1 зображено приклад побутового сенсора для моніторингу стану рослин, призначеного для використання в домашніх умовах. Цей сенсор було взято за ідею у подальшому розробленні особистого власного прототипу системи в межах дипломного проєкту, де було приділено увагу на зручність використання у домашніх умовах та поєднання з мобільним додатком.



Рисунок 1.1 – Побутовий сенсор моніторингу для кімнатних рослин

Можна зробити висновки, що попри наявність різних рішень, жодна з них не поєднує в собі мобільність дрона, сенсорний моніторинг саме домашніх квітів та зручний мобільний інтерфейс для кімнатного квітникаря. Саме ця ніша і є предметом дослідження та розробки в межах цієї дипломного проєкту.

1.3 Складові сенсорної системи моніторингу

Сенсорна система моніторингу рослинності є складною технічно-інформаційною інфраструктурою, яка забезпечує зчитування, обробку, передачу та візуалізацію фізичних параметрів середовища, у якому знаходяться кімнатні рослини. Була розроблена система, яка орієнтована на моніторинг стану домашніх

рослин, що чутливі до критичних змін та потребують більш уважного ставлення, у межах дипломного проєкту. Система базується на концепції розподіленої IoT-архітектури, де кожен модуль виконує окрему функцію, але є інтегрованим у загальну екосистему моніторингу [5].

Сенсорні модулі, які зчитують параметри температури, вологості ґрунту та освітленості – є основою апаратної частиною. Для цього застосовуються датчики цифрового типу, які підтримують стандартні протоколи обміну [7]. Ці дані надходять до мікроконтролера, який в свою чергу оброблює їх та готує дані до відправки у хмарне серидовище. Проаналізувавши усі варіанти мікроконтролерів більш оптимальним виявився ESP32 – універсальний чип з підтримкою Wi-Fi та Bluetooth.

За допомогою бездротового зв'язку здійснюється передача даних у систему. У базовій реалізації система орієнтується на використання Wi-Fi, що дозволяє забезпечити стабільну передачу інформації у межах домашньої мережі. У майбутньому для покращення системи можна використовувати такі технології: BLE, LoRa, ZigBee, з урахуванням потреб автономності або віддаленості системи.

Система також передбачає два варіанти живлення: автономне – для мобільних або дронів платформ, та стаціонарне – для сенсорів, встановлених безпосередньо у квіткові горщики в кімнаті. Такий підхід забезпечує гнучкість у конфігурації та можливість адаптації системи під різні сценарії використання.

Важливим елементом являється програмне забезпечення, воно включає в себе мобільний застосунок на платформі iOS, який реалізовано за допомогою SwiftUI. Застосунок виконує роль інтерфейсу для зручного використання користувачеві із системою, відображає зібрані дані у вигляді графіків, надсилає push-сповіщення про необхідність догляду за рослинами.

Для кращого розуміння системи моніторингу рослинності, на рисунку 1.2 зображено функціональну схему обробки даних. Ця схема демонструє ланцюг, де можна побачити зчитування параметрів сенсорами у навколишньому середовищі, передача їх до мікроконтролера, формування даних, збереження у базі даних та передача до мобільного застосунку.

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

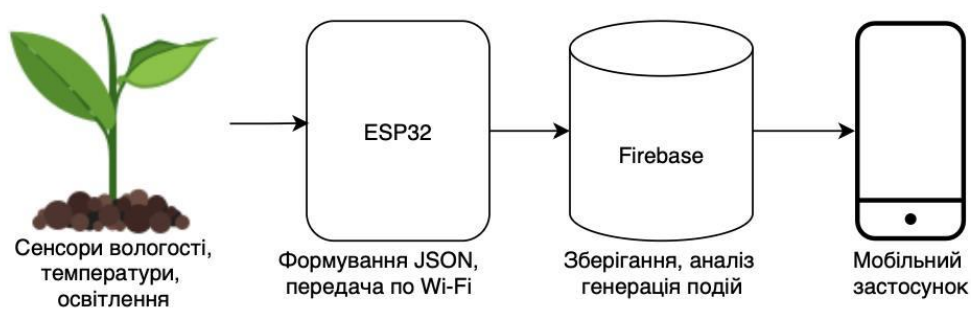


Рисунок 1.2 – Функціональна схема обробки та передачі даних у сенсорній системі моніторингу рослинності

1.4 Хмарні сервіси та мобільні додатки в екосистемах моніторингу

Хмарні сервіси вже давно стали основою багатьох сучасних технологічних рішень, і системи моніторингу рослин не стали винятком. Завдяки їм можна не лише зберігати та обробляти великі обсяги даних, а й миттєво аналізувати інформацію, фільтрувати її, об'єднувати з різних джерел та наочно відобразити зміни у зручному форматі.

У цьому проєкті було вирішено використовувати платформу Firebase. Вона об'єднує всі необхідні інструменти в одному середовищі, що значно спрощує реалізацію і підтримку системи. У процесі розробки системи було використано кілька компонентів платформи Firebase, кожен із яких виконує конкретне завдання. Наприклад, Firestore зберігає всі дані, що надходять від сенсорів, Authentication відповідає за вхід користувачів у систему, Cloud Messaging використовується для надсилання сповіщень, а Storage дозволяє зберігати зображення, наприклад, фотографії доданих рослин.

Щоб взаємодія з системою була простою та зручною, розроблено мобільний додаток. Його інтерфейс створений за допомогою SwiftUI, завдяки чому він виглядає лаконічно, легко адаптується до різних екранів і не перевантажений зайвими елементами. Користувачі можуть швидко орієнтуватися в застосунку, навіть не маючи досвіду роботи з подібними рішеннями. У додатку користувач може додавати нові рослини, задавати допустимі межі для показників, переглядати

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

історію змін та отримувати сповіщення, якщо параметри виходять за норму. Додаток підлаштовується під різні розміри екранів, тому ним зручно користуватися і на смартфонах, і на планшетах.

Також особливу увагу приділено безпеці. Вся передана інформація шифрується, щоб запобігти несанкціонованому доступу. Крім того, реалізована перевірка під час входу до системи — автентифікація через електронну пошту та пароль. Це дає змогу захистити персональні дані користувача без ускладнення самого процесу використання.

1.5 Проблеми та виклики впровадження IoT у моніторинг рослин

Хоча технології Інтернету речей (IoT) відкривають широкі можливості для моніторингу стану рослин, на практиці існує низка викликів, які можуть впливати на ефективність роботи таких систем і ускладнювати їх масштабування. У межах дипломного проєкту ці потенційні проблеми були детально проаналізовані, щоб врахувати їх ще на етапі проєктування та мінімізувати ризики у подальшій реалізації.

Однією з найпоширеніших складностей виявилася нестабільність бездротового з'єднання. У домашньому середовищі передача даних через Wi-Fi не завжди є надійною – на якість сигналу можуть впливати різноманітні фізичні перешкоди, як-от стіни, меблі чи навіть розташування роутера. Це може призводити до перебоїв у роботі системи, затримок у відображенні актуальної інформації або навіть втрати частини даних.

Це може призводити до затримок у відображенні інформації, втрати даних або порушення безперервності моніторингу. Для систем, які працюють у реальному часі, така нестабільність є критичним фактором, що може негативно впливати на своєчасність прийняття рішень користувачем.

Ще один важливий аспект – автономність живлення. Усі компоненти системи – сенсори, мікроконтролери, а особливо дрони – повинні мати змогу працювати тривалий час без потреби у підзарядці. Це особливо актуально для мобільних

					ІК12.120БАК.006 ПЗ	Арк.
						14
Зм.	Лист	№ докум.	Підпис	Дата		

пристроїв, які мають обмежену ємність акумуляторів. Щоб уникнути постійної зарядки, у проєкті було використано стаціонарні сенсори, вбудовані у квіткові горщики. Такий підхід дозволив знизити енергоспоживання та забезпечити стабільну роботу пристроїв протягом тривалого періоду.

Також не варто ігнорувати проблему точності побутових сенсорів. У домашньому середовищі умови можуть часто й непередбачувано змінюватись: відкрите вікно, увімкнений кондиціонер чи опалення можуть впливати на показники температури, вологості або освітлення. У результаті система може фіксувати короточасні зміни, які не мають значного впливу на стан рослини, але спотворюють загальну картину. Щоб уникнути помилкових висновків, необхідно впроваджувати додаткові алгоритми фільтрації або корекції даних.

Не менш важливим є й питання захисту персональних даних. Система зберігає дані користувачів, інформацію про їхні рослини, а також шаблони догляду. Оскільки ці дані знаходяться у хмарному середовищі, надзвичайно важливо реалізувати надійні механізми безпеки. Це включає автентифікацію користувача, контроль доступу, шифрування даних при зберіганні й передачі – усе це має бути частиною архітектури системи з самого початку.

І нарешті, не можна оминати питання загальної вартості реалізації. Хоча багато окремих компонентів – сенсори, мікроконтролери, модулі зв'язку – є досить доступними, загальна вартість системи може зрости за рахунок розробки мобільного застосунку, хмарної інфраструктури, технічного обслуговування та оновлень. Це може стати стримуючим фактором для масового впровадження подібних рішень. Тому одним із напрямів для подальшої оптимізації є створення модульної архітектури або спрощених конфігурацій системи, які можна буде адаптувати під різні бюджети.

Розроблений інтерфейс не обмежується лише відображенням поточних даних – він створений так, щоб бути максимально зручним, інформативним і зрозумілим для користувача. Усі важливі показники, зібрані системою моніторингу, подаються у наочному вигляді: графіки, діаграми та чіткі візуальні індикатори дають змогу легко відстежувати зміни в умовах середовища в динаміці.

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

Завдяки цьому користувач може не просто побачити, що з рослинами щось не так, а й зрозуміти, коли і чому це сталося. Це особливо зручно для тих, хто не має спеціальних знань у сфері агрономії чи технологій. Інтерфейс побудований так, щоб підказувати наступні кроки – він аналізує отримані дані й допомагає користувачу зрозуміти, коли і що саме потрібно зробити для покращення стану рослин.

Система також має зручну функцію сповіщень, яка одразу попереджає про важливі зміни в умовах навколо рослини. Наприклад, якщо ґрунт почав пересихати або температура стала надто високою чи низькою – користувач одразу отримає повідомлення. Завдяки цьому можна швидко зреагувати й вирішити проблему ще до того, як вона зашкодить рослині. Це дозволяє підтримувати здоров'я рослин без постійного спостереження чи втручання.

У цілому система дійсно полегшує догляд – вона бере на себе рутинні справи, а вся інформація подається просто і зрозуміло. Користуватись нею легко навіть тим, хто ніколи раніше не працював з подібними технологіями.

Вся інформація подається у простому, наочному форматі, тож навіть людина без технічного досвіду легко зможе розібратись, що відбувається і як діяти. Завдяки такому підходу користуватись системою можуть як досвідчені агровиборники, так і новачки, які тільки починають цікавитися доглядом за рослинами.

Висновок до розділу 1

У першому розділі було здійснено аналіз предметної області, що охоплює як технічні, так і концептуальні аспекти сенсорного моніторингу рослинності. Проведене дослідження підтвердило актуальність використання робототехнічних рішень, дронів, у системах догляду за домашніми рослинами, що потребують автоматизації та гнучкості в обробці великого обсягу даних у реальному часі.

Було проаналізовано існуючі сучасні рішення, що демонструють високий попит на подібні системи в аграрному та побутовому середовищі. Розглянуто основні компоненти сенсорної системи – сенсори, контролери, модулі передачі

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		16

даних, програмне забезпечення та проаналізовано їхню роль в загальній архітектурі IoT-системи.

Особливу увагу приділено робототехнічним платформам, серед яких дронева система обрана як найефективніше рішення для мобільного моніторингу. Платформа дозволяє охопити велику площу моніторингу, працювати в умовах обмеженого доступу та автоматизувати збір інформації.

Також було розглянуто можливості хмарних сервісів, таких як Firebase, які забезпечують зручне збереження та обробку даних та інтеграцію з мобільними застосунками. Окремо проаналізовано виклики, пов'язані з впровадженням подібних рішень – енергетична автономність, точність сенсорів, обмеження простору, захист персональних даних тощо.

Тож, за результатами теоретичного дослідження сформовано технічне обґрунтування для проєктування власної робототехнічної сенсорної системи моніторингу, що буде реалізована у наступних розділах.

					ІК12.120БАК.006 ПЗ	Арк.
						17
Зм.	Лист	№ докум.	Підпис	Дата		

2 АНАЛІЗ І ПРОЄКТУВАННЯ РОБОТОТЕХНІЧНОЇ СЕНСОРНОЇ СИСТЕМИ МОНІТОРИНГУ РОСЛИННОСТІ

2.1 Аналіз предметної області

Догляд за кімнатними рослинами, у сучасних умовах, вимагає автоматизований підхід, які враховують обмеження часу користувачів, складність ручного моніторингу та потреб в точному відстеженні змін у стані рослин. Домашнє квітникарство – це не тільки про хобі, це також про екологію та психологічну стабільність в умовах життя в місті. Такі традиційні методи огляду як ручне поливання, нерегулярні заміри вологості ґрунту, огляд на око – є не дуже ефективними, тим паче при наявності багато різних видів рослин, які мають індивідуальні потреби.

Особливої уваги вимагають рослини з високими вимогами до мікроклімату, такі як троянди та тюльпани, які чутливо реагують на зміну вологості, рівня освітлення або температури. Якщо не вчасно виявити проблему це може призвести до шкоди – хвороб або загибелі квітки.

Дивлячись на це все, інтелектуальні робототехнічні рішення набувають більшої актуальності, які дозволяють забезпечити точне, своєчасне спостереження за рослинами. Одним із перспективних підходів є використання безпілотних мобільних платформ (дронів), оснащених сенсорами, для періодичного сканування рослин у приміщенні.

Завдяки можливостям автономного переміщення, збирання даних та передачі їх у хмарну інфраструктуру, робототехнічна система може значно зменшити залучення людини до рутинних операцій. Використання мобільного застосунку для візуалізації зібраної інформації та видачі рекомендацій щодо догляду робить таку систему доступною навіть для непрофесійних користувачів.

Інтеграція з хмарними платформами (такими як Firebase) забезпечує масштабованість, персоналізацію під кожного користувача та доступ з будь-якого пристрою.

					ІК12.120БАК.006 ПЗ	Арк.
						18
Зм.	Лист	№ докум.	Підпис	Дата		

2.2 Огляд технічних компонентів робототехнічної сенсорної системи

Розроблена робототехнічна сенсорна система моніторингу рослинності поєднує апаратні та програмні компоненти, що забезпечують автоматизований збір, передачу, збереження та візуалізацію екологічних даних у режимі реального часу. У межах дипломного проєкту було створено прототип повноцінної функціональної системи, яка охоплює сенсори, мікроконтролер, комунікаційний модуль, хмарну платформу для обробки інформації, а також мобільний застосунок як інтерфейс кінцевого користувача.

Сенсорна частина системи є критично важливою для коректного вимірювання параметрів середовища. Для моніторингу стану рослин у горщиках були використані сучасні датчики вологості ґрунту, освітленості та температури, що забезпечують необхідну точність і стабільність показників. Зчитування параметрів відбувається через мікроконтролер ESP32, який також виконує попередню обробку сигналів, форматує дані у формат JSON та передає їх у Firebase.

Для реалізації стабільного функціонування сенсорної системи важливо визначити, які параметри подаються на вхід, в якому форматі вони передаються та яку роль відіграють у загальній логіці обробки. Таблиця 2.1 систематизує ці характеристики для кожного сенсора, що використовується в системі.

Таблиця 2.1 – Вхідні параметри, які зчитуються сенсорами

Параметр	Одиниця вимірювання	Приклад значення
Вологість ґрунту	%	47
Температура повітря	°C	22.5
Вологість повітря	%	58
Освітленість	lux	350

Для реалізації складової проєкту було використано хмарну платформу Firebase, яка виступає в ролі Backend-as-a-Service (BaaS) [12]. Ця платформа забезпечує зберігання та обробку даних у режимі реального часу, управління

авторизацією користувачів надсилання push-сповіщень та збереження медіафайлів. Firebase виступає точкою доступу між мікроконтролером та мобільним застосунком.

Мобільний застосунок, створений для iOS-платформи за допомогою SwiftUI, реалізує функціонал взаємодії з користувачем: авторизація, перегляд даних, додавання нових рослин, перегляд графіків, отримання сповіщень. Система має повноцінну трирівневу структуру, яка включає сенсорний рівень (збір даних), обчислювальний рівень (обробка та зберігання у Firebase) та інтерфейсний рівень (взаємодія з користувачем через застосунок).

2.3 Вибір апаратної платформи

Одним із ключових завдань стало визначення апаратної платформи, на етапі проектування робототехнічної сенсорної системи моніторингу рослинності, яка б забезпечувала оптимальний баланс між мобільністю, автономністю, точністю вимірювань, простотою інтеграції з сенсорними модулями та сумісністю з програмною частиною системи.

У межах дипломного проекту розглядалося декілька підходів до апаратної реалізації системи моніторингу стану рослин, серед яких – використання мобільних роботизованих платформ, дронів та стаціонарних сенсорних модулів. Так як система призначена для використання у домашніх умовах, за доглядом за кімнатними рослинами, було прийнято рішення використовуватисенсорні платформи, інтегрованими у горщики. Це дозволило досягти простої реалізації, мінімізувати витрати на обладнання, уникнути складнощів у переміщенні пристроїв у замкнутому просторі та забезпечити постійний моніторинг рослин.

Мікроконтролер ESP32 виступає основною апаратною частиною, який забезпечує збір та передавання даних у хмару через Wi-Fi. Цей мікроконтролер було вибрано через високу продуктивність, енергоефективність та широку підтримку бібліотек [6]. Усі сенсори підключаються до ESP32 через відповідні порти – аналогові або цифрові, залежно від типу датчика.

					ІК12.120БАК.006 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

Живлення пристрою може здійснюватися як через зовнішній адаптер (для постійної роботи), так і автономно – за допомогою акумулятора, що особливо актуально в умовах тимчасових експериментів або для мобільних інсталяцій. Усі вимірювання передаються у форматі JSON у хмарне середовище Firebase, де вони зберігаються, аналізуються й відображаються у мобільному застосунку користувача.

Вибір такої архітектури зумовлений саме зі сторони практичності. Так як у ході виконання дипломного проєкту розробка та тестування системи проходили у квартирі, то було важливо забезпечити стабільність вимірювань, простоту експлуатації та мінізувати залежність від зовнішніх умов, таких як пошкодження з'єднання, розміри приміщення або перешкоди.

У результаті, обрана конфігурація апаратної платформи дозволила реалізувати стабільну, енергоефективну та масштабовану систему моніторингу стану кімнатних рослин, яка може бути адаптована як для використання в приватних оселях, так і в комерційних просторах (наприклад, офісах або оранжереях). У перспективі можливе розширення кількості сенсорів та впровадження додаткових модулів аналітики.

2.4 Вибір архітектури взаємодії компонентів

Проектування інформаційної архітектури сенсорної системи моніторингу рослинності вимагало комплексного підходу до вибору моделі взаємодії між усіма її функціональними підсистемами. Оскільки йдеться не лише про збір екологічних даних, але й про їхню фільтрацію, зберігання, аналіз, візуалізацію та генерацію сповіщень, було критично важливо забезпечити чіткий розподіл обов'язків між апаратними та програмними компонентами системи. У цьому контексті було обрано клієнт-серверну архітектуру, яка найбільш ефективно реалізує ідею розподіленої обробки даних: первинне зчитування та передача параметрів здійснюються на рівні сенсорного вузла, тоді як обробка, зберігання та логіка сповіщення покладаються на хмарну інфраструктуру. Таким чином, архітектура

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

системи відповідає сучасним принципам побудови IoT-рішень та дозволяє масштабувати систему без переробки її фундаментальних елементів.

На фізичному рівні сенсорний вузол побудований на базі мікроконтролера ESP32, який має вбудовані засоби для бездротової передачі даних і достатню обчислювальну потужність для обробки сигналів від аналогових і цифрових сенсорів. У проєкті передбачено інтеграцію сенсорів вологості ґрунту, температури, вологості повітря та освітленості, кожен з яких у реальному часі фіксує значення відповідних параметрів. Зібрані дані форматуються у структурований вигляд, зокрема у формат JSON, що забезпечує сумісність із хмарними сервісами. Передача здійснюється через Wi-Fi до сервера, роль якого виконує платформа Firebase. У структурі архітектури саме Firebase виступає як централізований вузол обробки даних: він забезпечує збереження показників у базі Firestore або Realtime Database, проводить автентифікацію користувачів через Firebase Authentication, а також надсилає push-сповіщення через Firebase Cloud Messaging, коли параметри виходять за допустимі межі. Такий підхід дозволив уникнути потреби у створенні власного backend-сервера, значно знизити часові та фінансові витрати на розгортання інфраструктури, а також зменшити складність підтримки.

Застосування клієнт-серверної моделі забезпечило чітке логічне розділення між рівнем збору даних і рівнем візуалізації, що реалізується на стороні користувача. У якості клієнта використовується мобільний застосунок, розроблений за допомогою фреймворку SwiftUI для операційної системи iOS. Завдяки інтеграції з Firebase SDK застосунок має прямий доступ до даних, що надходять від сенсорів, і може відображати їх у вигляді таблиць, графіків і карток рослин. Крім того, користувач має можливість керувати системою: змінювати порогові значення параметрів, вимикати або вмикати сповіщення, переглядати історію змін тощо. Усі дії автоматично синхронізуються з базою, що підтримує концепцію роботи з даними в реальному часі.

У процесі проєктування також враховувалися питання надійності, захищеності й масштабованості. Усі канали передачі даних між сенсорним

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		22

пристроєм і хмарою, а також між хмарою та мобільним клієнтом, працюють через HTTPS із використанням SSL/TLS, що забезпечує захист від несанкціонованого доступу. Firebase Security Rules дозволяють налаштувати політики доступу на рівні окремого користувача – кожен бачить лише власні дані. Завдяки розподіленій природі обраної архітектури систему можна масштабувати не лише на більшу кількість рослин, а й на нові типи сенсорів чи додаткові джерела даних, не порушуючи вже створену інфраструктуру.

Таким чином, архітектура, реалізована в межах цього проєкту, забезпечує високий ступінь інтеграції між апаратним та програмним забезпеченням, поєднує стабільність і гнучкість, а також орієнтована на майбутній розвиток системи. Її побудова демонструє можливість створення сучасної IoT-системи для побутового моніторингу рослинності, яка при цьому відповідає вимогам промислової надійності, інформаційної безпеки та користувацької зручності.

2.5 Розробка структурної та архітектурної схем системи

У межах реалізації програмно-апаратної системи моніторингу рослинності одним із ключових завдань стало формування чіткої архітектурної моделі, яка забезпечувала б ефективну взаємодію між сенсорними модулями, інфраструктурою зберігання та обробки даних і мобільним інтерфейсом користувача. Для цього була спроектована багаторівнева архітектура, що відповідає сучасним вимогам до IoT-систем: вона забезпечує масштабованість, стабільність, адаптивність і можливість подальшого розвитку функціональності без істотних змін у вже реалізованих компонентах.

На кресленні ІК12.120БАК.006 Д1 представлено структурну схему системи моніторингу стану рослинності, що демонструє взаємозв'язки між апаратними та програмними компонентами, а також логіку обміну даними між сенсорними модулями, мікроконтролером, хмарним середовищем та мобільним застосунком

Архітектурна модель системи базується на принципі розділення відповідальності за логіку роботи між трьома основними рівнями: периферійним

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

сенсорним, хмарним обчислювальним і клієнтським інтерфейсним. Перший рівень реалізується апаратною частиною системи – мікроконтролером ESP32 із підключеними до нього сенсорами вологості ґрунту, температури й освітленості. Цей рівень відповідає за збирання даних із фізичного середовища, попередню обробку сигналів та передачу інформації далі на хмарний рівень. Обмін даними відбувається через Wi-Fi у структурованому форматі JSON, що дає змогу легко масштабувати систему й підтримувати стандартизовану взаємодію з іншими платформами.

Хмарний – це другий рівень представлений платформою Firebase, яка забезпечує не лише збереження даних, але й обробку подій, автентифікацію користувачів, систему сповіщень та контроль доступу. Тут реалізовано логіку обробки вхідних значень, перевірку їхніх порогових значень і формування відповідних push-сповіщень у разі виявлення критичних ситуацій. З використанням Firebase Realtime Database або Firestore дані оновлюються в режимі реального часу й автоматично синхронізуються з клієнтським застосунком. У хмарному середовищі функціонують додаткові модулі – такі як Firebase Authentication для керування обліковими записами або Firebase Storage для збереження зображень.

Мобільний застосунок розташовується на третьому рівні, який реалізований за допомогою SwiftUI. Додаток – основний інтерфейс взаємодії користувача з системою. Застосунок отримує дані з хмари, відображає їх у графічній формі, дозволяю переглядати історію змін, додавати нові рослини, змінювати налаштування та отримувати повідомлення про можливі пошкодження. Користувач отримує доступ до особистого профілю з можливістю керувати та відстежувати зміни.

Окрема увага була приділена питанню потоків обробки даних, під час архітектурного проектування. Щоб розпочати процес взаємодій між компонентами треба почати з формування пакета даних на мікроконтролері, що включає числові значення необхідних параметрів. Це все надсилається до Firebase, де активується відповідна функція перевірки порогів. Якщо якийсь із параметрів буде нижче норми, то відбувається миттєва генерація повідомлення на пристрої користувача, що

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		24

дозволяє йому оперативно відреагувати.

Також проєкт дає можливість масштабування системи без необхідності її перебудови. Без додаткових змін на сервері можливо додавати нові сенсори, розширювати логіку мобільного додатку. За потреби можуть бути реалізовані альтернативні протоколи передачі – такі як LoRa або Bluetooth Low Energy – для роботи у зонах із нестабільним Wi-Fi.

Безпека є важливим аспектом побудови архітектури подібної системи. Всі дані, що передаються мережею, шифруються за допомогою SSL/TLS. Для обмеження доступу використовуються політики безпеки Firebase Rules, які гарантують, що користувачі мають доступ лише до своїх власних даних. Кожна дія у системі логуються, що дозволяє проводити аудит та виявляти потенційні загрози або спроби несанкціонованого доступу.

На кресленні ІК12.120БАК.006 Д2 наведено архітектурну схему сенсорної системи моніторингу стану рослинності. Схема демонструє взаємозв'язки між основними апаратними та програмними компонентами – включно з сенсорами, мікроконтролером ESP32, платформою Firebase та мобільним застосунком.

У підсумку, побудована архітектура демонструє не лише відповідність технічним вимогам дипломного проєкту, але й закладає основу для її практичного використання в умовах реального середовища. Її багаторівнева організація дозволяє забезпечити ефективну, безпечну та адаптивну взаємодію між усіма компонентами системи моніторингу, що робить її придатною як для побутових, так і для промислових застосунків у сфері догляду за рослинністю.

2.5.1 Потік обробки даних

У межах розробленої системи особливу увагу приділено організації повного циклу передачі, обробки та відображення даних про стан рослин. Потік даних побудовано за принципом реального часу, що дозволяє досягти оперативності в прийнятті рішень щодо догляду за кімнатними рослинами. Кожен сенсор, підключений до мікроконтролера ESP32, здійснює регулярне зчитування

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		25

параметрів вологості ґрунту, температури повітря та рівня освітленості.

На рисунку 2.1 зображено схему підключення сенсорів до мікроконтролера ESP32.

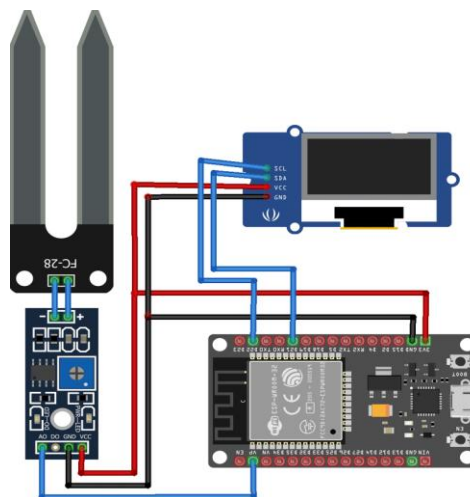


Рисунок 2.1 – Схема підключення сенсорів до мікроконтролера

Отримані значення попередньо формуються на рівні мікроконтролера та передаються до хмарної платформи Firebase через Wi-Fi-з'єднання. Формат обміну даними базується на структурі JSON, що дозволяє стандартизувати передачу та обробку інформації. У базі Firestore дані структуруються відповідно до унікального користувачького ідентифікатора та ідентифікатора кожної рослини, після чого зберігаються з часовою міткою.

Після потрапляння даних у хмару, система може автоматично викликати хмарну функцію Cloud Function, яка перевіряє отримані параметри на відповідність пороговим значенням. Якщо спостерігається відхилення – наприклад, вологість ґрунту знизилась нижче встановленого користувачем рівня – запускається механізм повідомлення через Firebase Cloud Messaging, і користувач оперативно отримує push-сповіщення на свій мобільний пристрій.

У мобільному застосунку ці дані миттєво оновлюються та відображаються у вигляді графіків або індикаторів статусу, що забезпечує повну прозорість моніторингу та зручність взаємодії для користувача. Такий підхід гарантує надійність передачі, синхронність відображення, а також зберігає можливість

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		26

подальшого масштабування і розвитку системи.

2.5.2 Розширюваність та безпека системи

В даній системі, архітектура орієнтована не тільки на вирішення поточних завдань моніторингу, але й на розвиток функціональності у майбутньому. При використанні хмарної платформи Firebase та мікроконтролера ESP32, можливе швидке додавання нових сенсорів без додаткової модифікації коду зі сторони клієнта або сервера. Таким чином, систему можливо адаптувати до різних типів рослин або розширити її функціональність, наприклад, інтегрувати фотокамери для візуального контролю за станом рослин.

Важливою складовою архітектури є також підтримка локальних мереж з використанням альтернатив до Wi-Fi – таких як LoRa, BLE або Zigbee – що особливо актуально у випадках, коли доступ до традиційних мереж обмежений або недоступний.

Окрему увагу приділено аспектам захисту персональних даних користувача. Всі дані, що передаються між пристроями, зашифровані протоколами SSL/TLS, що унеможливує перехоплення або підміну даних у процесі передачі. Для обмеження доступу до інформації використано систему правил безпеки Firebase Security Rules, які гарантують, що кожен користувач має доступ виключно до власних даних. Уся інформація проходить валідацію як на стороні мобільного застосунку, так і в хмарному середовищі, що дозволяє уникнути внесення некоректних або шкідливих даних у систему.

Також, система підтримує механізм аудиту та логування дій користувача, що дозволяє виявляти підозрілі активності, аналізувати історію взаємодії та забезпечувати вищий рівень надійності в роботі з чутливою інформацією. Такий рівень безпеки є необхідним компонентом сучасних інформаційних систем, що обробляють персональні дані.

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		27

Висновок до розділу 2

У цьому розділі було проведено всебічний аналіз предметної області, що стосується систем моніторингу стану рослинності.

Особливу увагу приділено сучасним технологічним рішенням і технічним засобам, які можуть бути використані для створення ефективної сенсорної системи збору екологічних даних.

Було розглянуто наявні на ринку компоненти, такі як сенсори вологості ґрунту, температури повітря, рівня освітленості, та здійснено порівняльний аналіз їх характеристик для вибору оптимальних варіантів.

У результаті технічного обґрунтування було прийнято рішення використовувати апаратну платформу на основі мікроконтролера ESP32, який поєднує достатню обчислювальну потужність, підтримку бездротового зв'язку Wi-Fi та Bluetooth, а також широкий спектр сумісних периферійних пристроїв.

Центральним елементом розробленої сенсорної системи було обрано мікроконтролер ESP32, який розміщується на борту дронавої платформи.

Завдяки своїм технічним можливостям та компактним розмірам, ESP32 ідеально підходить для створення мобільної системи моніторингу, що дозволяє ефективно збирати дані навіть у важкодоступних або віддалених частинах поля чи тепличного господарства.

Особливу увагу в проєкті приділено інтеграції основних екологічних сенсорів, зокрема вологості ґрунту, температури повітря та рівня освітленості. Розглянуто технічні аспекти їх підключення, калібрування та обробки отриманих даних на програмному рівні.

Ці параметри мають вирішальне значення для оцінки стану навколишнього середовища, адже безпосередньо впливають на ріст, розвиток і загальне здоров'я рослин.

Система дозволяє зчитувати показники в режимі реального часу, що відкриває можливість для оперативного реагування на несприятливі зміни кліматичних умов.

					ІК12.120БАК.006 ПЗ	Арк.
						28
Зм.	Лист	№ докум.	Підпис	Дата		

З боку програмного забезпечення було реалізовано клієнт-серверну архітектуру з використанням хмарної платформи Firebase. Цей сервіс було обрано завдяки його надійності, зручності в інтеграції, а також широким функціональним можливостям.

Firebase забезпечує централізоване зберігання отриманих даних, їх обробку, синхронізацію між різними компонентами системи, а також дозволяє реалізувати автентифікацію користувачів і гнучке керування доступом.

Крім того, у разі виявлення критичних змін параметрів середовища, система автоматично надсилає сповіщення користувачам, що дозволяє своєчасно вжити необхідних заходів.

На основі обраної архітектури розроблено структурну й функціональну схему взаємодії між усіма компонентами системи: від сенсорного рівня на базі ESP32, через модулі зв'язку, до хмарного сервісу та мобільного додатку, який виконує роль зручного інтерфейсу користувача.

Запропонований підхід дозволяє побудувати інтелектуальну систему моніторингу, що є не лише зручною у використанні, але й здатною до масштабування залежно від потреб господарства.

Вона підвищує ефективність аграрного виробництва за рахунок своєчасного виявлення змін у стані рослин, мінімізації людського фактору та забезпечення гнучкості в управлінні екосистемою.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Архітектура програмного забезпечення

Було використано принципи клієнт-серверна архітектури для побудови програмної системи моніторингу рослинності з використанням хмарних технологій. Основна взаємодія відбувається між сенсорною системою (джерелом даних), мобільним застосунком (інтерфейсом користувача) та хмарним сховищем Firebase, що забезпечує зберігання, обробку та синхронізацію даних.

Компоненти архітектури виконують такі функції:

- сенсорна система здійснює зчитування параметрів (вологість ґрунту, температура, освітленість) та передає їх до мікроконтролера;
- ESP32 виконує форматування даних у JSON та передає їх у Firebase;
- firebase зберігає дані, керує автентифікацією користувача, генерує push-сповіщення у випадку відхилень.
- Мобільний застосунок, реалізований на SwiftUI, візуалізує інформацію, дозволяє переглядати статистику, додавати нові рослини та налаштовувати систему під користувача.

На таблиці 3.1 наведено еомпоненти програмної архітектури

Таблиця 3.1 – Компоненти програмної архітектури

Компонент	Функція
Сенсори	Зчитування параметрів рослин (вологість, освітленість тощо)
ESP32	Формування та надсилання даних
Firebase Realtime/Firestore	Зберігання та обробка даних
Firebase Cloud Messaging	Сповіщення про критичні значення
Firebase Authentication	Ідентифікація користувачів
iOS-додаток	Візуалізація, взаємодія, налаштування

Ця архітектура дозволяє реалізувати логіку в реальному часі, з автоматичними реакціями на зміни параметрів та інформуванням користувача про

можливі пошкодження. Приклад інформування користувача: у разі падіння вологості нижче заданого порогу система генерує повідомлення та надсилає його у мобільний застосунок.

3.1.1 Структура архітектури системи

Система складається з трьох основних рівнів:

- 1) сенсорна підсистема – датчики у горщиках фіксують параметри (вологість, температура, освітленість) і передають їх на мікроконтролер ESP32;
- 2) хмарна частина – дані надсилаються у Firebase (Firestore), де зберігаються, обробляються та запускають логіку повідомлень;
- 3) мобільний клієнт – користувач взаємодіє з даними через застосунок: переглядає значення, додає рослини, отримує сповіщення.

Компоненти сенсорної системи:

- датчики вологості ґрунту (YL-69);
- датчики температури (DHT11 або DS18B20);
- сенсори освітленості (BH1750);
- мікроконтролер ESP32 з Wi-Fi-модулем.

Firebase-сервіси:

- firestore – для зберігання значень параметрів;
- authentication – авторизація користувачів;
- cloud Messaging (FCM) – надсилання push-сповіщень;
- storage (опційно) – збереження зображень рослин.

Мобільний застосунок (iOS):

- написаний на мові Swift із використанням SwiftUI;
- інтегрований із Firebase SDK;
- реалізовано архітектуру MVVM;
- підтримка багатомовного інтерфейсу.

3.1.2 Обґрунтування вибору інструментів

					ІК12.120БАК.006 ПЗ	Арк.
						31
Зм.	Лист	№ докум.	Підпис	Дата		

Під час проєктування системи було розглянуто кілька варіантів інструментів для реалізації ключових її компонентів. Вибір кожного з них був обґрунтований з урахуванням зручності використання, сумісності, функціональності та майбутньої масштабованості проєкту.

Мова програмування

Серед можливих варіантів для створення мобільного застосунку розглядалися Swift, Kotlin та Flutter. У підсумку було обрано Swift, оскільки саме він є нативною мовою для платформ Apple і має чудову інтеграцію з сучасним фреймворком SwiftUI. Використання мови Swift дало змогу реалізувати зручний і швидкий у роботі інтерфейс користувача, повністю адаптований під iOS. Завдяки тісній інтеграції з фреймворком SwiftUI процес розробки став не лише ефективнішим, а й значно комфортнішим, що дозволило зосередитися на створенні інтуїтивно зрозумілого та сучасного дизайну.

Серверна частина (бекенд)

Для організації серверної логіки було розглянуто два основні варіанти – Firebase та AWS. У підсумку перевагу надали Firebase, оскільки ця платформа забезпечує просте налаштування, широкий набір готових рішень (зокрема автентифікація, зберігання даних, обробка запитів, надсилання сповіщень), а також легке масштабування без потреби в окремій серверній інфраструктурі. Це значно спрощує запуск проєкту та прискорює його реалізацію, а також забезпечує зручність подальшої підтримки й масштабування системи.

Система зберігання даних

Для збереження інформації, яку система отримує від сенсорів, було розглянуто два варіанти – Realtime Database та Firestore. Після аналізу функціональних можливостей перевагу надали Firestore, адже вона пропонує більш гнучку та логічно структуровану модель даних. Це дає змогу легко працювати з великими обсягами екологічної інформації, формувати складні запити та забезпечувати швидку синхронізацію даних у реальному часі. Крім того, Firestore має вбудовані засоби безпеки, які забезпечують надійне зберігання даних, навіть коли навантаження на систему зростає. Це особливо важливо для стабільної роботи

					ІК12.120БАК.006 ПЗ	Арк.
						32
Зм.	Лист	№ докум.	Підпис	Дата		

в умовах постійного потоку інформації від сенсорів.

Мікроконтролер

Для реалізації апаратної частини було розглянуто кілька варіантів, серед яких – ESP32 та Arduino Uno. Після аналізу переваг вибір зупинився на ESP32, оскільки він має значно більше можливостей: зокрема, вбудовані модулі Wi-Fi та Bluetooth дозволяють одразу реалізувати бездротову передачу даних, без потреби у додаткових компонентах. Це не тільки спрощує загальну конструкцію пристрою, а й дозволяє зробити систему компактнішою, легшою та зручнішою у розгортанні, особливо у мобільних умовах.

Крім того, однією з ключових переваг ESP32 є його відмінне поєднання продуктивності й енергоефективності. Цей мікроконтролер легко справляється з обробкою великого обсягу даних, водночас споживаючи мінімум енергії. Це особливо важливо для рішень, які працюють автономно – наприклад, на дронах або в польових умовах, де немає постійного джерела живлення.

Ще один плюс – велика спільнота розробників навколо ESP32. Завдяки її активності існує безліч готових бібліотек, прикладів та детальної документації, що значно спрощує та прискорює розробку, налаштування й вдосконалення системи.

Усі обрані компоненти добре взаємодіють між собою та формують єдину, злагоджену екосистему. Завдяки цьому вдалося створити не просто робоче рішення, а повноцінну, гнучку й зручну у використанні систему моніторингу стану рослин, яка легко адаптується під різні умови та потреби користувача.

На таблиці 3.2 наведено обґрунтування вибору інструментів

Таблиця 3.2 - Обґрунтування вибору інструментів

Компонент	Варіанти	Обрано	Обґрунтування
Мова програмування	Swift, Kotlin, Flutter	Swift	Рідна підтримка iOS, інтеграція з SwiftUI
Бекенд	Firebase, AWS	Firebase	Простота, масштабованість, BaaS
Зберігання даних	Firestore, Realtime DB	Firestore	Гнучка структура, запити
Мікроконтролер	ESP32, Arduino Uno	ESP32	Wi-Fi, продуктивність

3.1.3 Візуальні та функціональні компоненти застосунку

Усі екрани реалізовано з урахуванням UX/UI-дизайну та підтримкою accessibility:

- авторизація (реєстрація, вхід, відновлення паролю);
- головна сторінка – список рослин та індикаторами стану;
- додавання нової рослини – назва, тип;
- екран статистики – графіки зміни вологості, освітлення, температури;
- налаштування – мова, логін, сповіщення, вихід з акаунту.

Дані з Firebase синхронізуються у реальному часі. Приклад структури документа у Firestore показано на рисунку 3.1.

```
/users
  /user_123@example_com
    plants:
      /plant_001
        name: "Тюльпан"
        humidity: 37.8
        temperature: 22.1
        light: 256
```

Рисунок 3.1 – Приклад структури документа у Firestore

3.1.4 Безпека, надійність та масштабованість

Firebase Authentication гарантує, що кожен користувач має доступ лише до своїх рослин [13]. Усі правила доступу прописані у Firestore Security Rules. Приклад наведено на рисунку 3.2.

```
match /users/{userId}/plants/{plantId} {
  allow read, write: if request.auth.uid == userId;
}
```

Рисунок 3.2 – Приклад правила доступу у Firestore Security Rules

У разі розширення системи (наприклад, підключення 10+ рослин, нових

					ІК12.120БАК.006 ПЗ	Арк.
						34
Зм.	Лист	№ докум.	Підпис	Дата		

користувачів) архітектура не потребує змін. Дані кожного користувача зберігаються окремо, що дозволяє масштабування без втрат продуктивності.

3.2 Firebase як основа зберігання та синхронізації даних

У межах побудови робототехнічної сенсорної системи моніторингу домашніх рослин постала необхідність створення ефективною, надійною та масштабованою інфраструктури для зберігання, передачі та синхронізації даних, що надходять від сенсорів. Таку інфраструктуру мала б підтримувати як апаратна частина системи – мікроконтролери й сенсори, так і мобільний застосунок, що виконує роль інтерфейсу користувача. Аналіз сучасних рішень показав, що найоптимальнішим вибором є використання хмарної платформи Firebase, яка об'єднує базу даних, автентифікацію, надсилання сповіщень, логіку обробки подій та інші важливі функції в одному екосередовищі.

Однією з основних переваг Firebase є можливість реалізації роботи в режимі реального часу, що особливо важливо для задач моніторингу. У проекті було реалізовано інтеграцію з Firebase Firestore – гнучкою NoSQL-базою даних, яка дозволяє зберігати параметри, зібрані сенсорами, у структурованій та масштабованій формі. Структура бази організована таким чином, що кожен користувач має унікальний ідентифікатор (UID), пов'язаний з його обліковим записом, а всі рослини та їхні параметри зберігаються у вкладеній колекції documents. Завдяки цьому досягається як ізоляція даних між користувачами, так і гнучкість у роботі з різною кількістю рослин.

На рівні мобільного застосунку Firebase надає не лише зручний механізм авторизації користувачів через Authentication, а й можливість отримувати push-сповіщення за допомогою Cloud Messaging. Це дозволяє, наприклад, миттєво інформувати користувача про відхилення показників вологості або температури від встановлених меж. Для реалізації складніших сценаріїв взаємодії використано Firebase Cloud Functions – серверну логіку, яка реагує на зміну даних у базі та ініціює відповідні дії, такі як надсилання повідомлень або логування критичних

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		35

ситуацій.

Передача даних від сенсорів у Firestore реалізується через мікроконтролер ESP32, який збирає дані з підключених сенсорів та формує JSON-об'єкти з поточними значеннями вологості, температури та освітлення. Ці дані надсилаються в базу даних через бібліотеку ESP Firebase Client або REST API, що забезпечує гнучкість і стабільність [10]. Приклад переданих даних включає такі ключі: humidity, temperature, light і timestamp, що дозволяє зберігати як поточні значення, так і історію зміни параметрів. У мобільному застосунку, розробленому на SwiftUI, реалізовано двосторонню взаємодію з Firebase: застосунок підписується на зміни документів у реальному часі та оновлює інтерфейс без необхідності ручного перезапуску.

Окрему увагу було приділено логіці реакції на критичні зміни. У випадках, коли значення вологості падає нижче встановленого порогу, запускається функція, яка формує та надсилає сповіщення користувачу. Такий механізм дозволяє забезпечити негайний зворотний зв'язок, підвищити чутливість системи та мінімізувати ризик втрати рослини через недогляд. Щоб уникнути надмірної кількості повідомлень при незначних коливаннях, реалізовано додаткову логіку, що перевіряє стабільність критичного значення впродовж певного часу до надсилання сповіщення.

З технічного боку, вся передача даних реалізується у форматі JSON, що є стандартом для обміну інформацією у вебсередовищі. Такий підхід дозволяє легко інтегрувати інші платформи або сервіси в майбутньому – наприклад, хмарну аналітику або машинне навчання. Безпека даних забезпечується за рахунок Firebase

Security Rules, які дозволяють обмежити доступ до інформації лише авторизованому користувачу. Наприклад, кожен користувач може бачити лише свої рослини, тоді як інші документи для нього повністю недоступні. Це особливо важливо в умовах обробки персональних даних і при потенційному масштабуванні системи до комерційного продукту.

Таким чином, вибір платформи Firebase виявився обґрунтованим з огляду на її функціональність, простоту інтеграції, підтримку мобільних застосунків і

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		36

можливість масштабування. Реалізована система забезпечує надійне зберігання даних, синхронізацію в реальному часі, ефективну обробку критичних подій та високий рівень безпеки, що повністю відповідає вимогам до сучасних IoT-рішень у сфері моніторингу навколишнього середовища.

3.2.1 Структура бази даних Firestore

Для зберігання даних у системі моніторингу рослинності було обрано хмарну NoSQL-базу даних Firestore, яка є частиною платформи Firebase. Firestore забезпечує ієрархічну організацію даних, що дозволяє будувати масштабовані, гнучкі та легко керовані структури. Firestore не використовує таблиць, рядків і стовпців, в порівнянні з іншими реляційними базами. Також вона працює з колекціями та документами. Кожна колекція містить документи, а документи – пари «ключ-значення» та, при потребі, вкладені колекції. Така структура є особливо зручною для мобільних застосунків, які повинні динамічно працювати з багаторівневими наборами даних без надмірного навантаження на клієнтську частину або необхідності в складних SQL-запитах.

У контексті розробки робототехнічної сенсорної системи для моніторингу домашніх рослин модель зберігання у Firestore передбачає окрему колекцію користувачів, де кожен користувач представлений як окремий документ. Ідентифікатором цього документа є UID користувача, який автоматично генерується Firebase Authentication після реєстрації. У межах документа кожного користувача зберігаються основні параметри (ім'я, електронна адреса тощо), а також створюється вкладена колекція «plants», де кожна рослина користувача зберігається як окремий документ.

Опис кожного документа рослини містить такі ключі: назва рослини (наприклад, "Роза"), тип (наприклад, "квітуча"), значення вологості ґрунту, температури середовища та освітлення, а також часову мітку останнього оновлення. Це дозволяє мати структурований доступ до даних, легко здійснювати фільтрацію та сортування, а також будувати історію змін при потребі. Наприклад,

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		37

якщо користувач додає нову рослину, в Firestore створюється новий документ у вкладеній колекції plants, що містить усі необхідні параметри.

Головною відмінністю в ікористанні Firestore є підтримка роботи у реальному часі. Це означає, що якщо будуть оновлення документа в базі даних, то усі клієнти які на нього підписани (додаток на смартфоні) автоматично отримують зміни без самостійного оновлювання інтерфейсу. Це критично важливо в контексті системи моніторингу, де зміни параметрів сенсорів повинні відобразитися без затримок. Наприклад, якщо датчик вологості фіксує зниження до критичного рівня, відповідне оновлення надходить у Firestore, і застосунок одразу повідомляє користувача про необхідність поливу.

Структура бази даних також враховує питання безпеки. Для кожного користувача налаштовуються правила доступу, згідно з якими лише він має право читати або змінювати свої документи. Це реалізується за допомогою Firebase Security Rules, які перевіряють UID автентифікованого користувача перед наданням доступу. Навіть у разі спроби прямого доступу до бази через API, сторонні користувачі не зможуть отримати доступ до чужих даних.

Ще однією перевагою такої структури є масштабованість. Кожен користувач працює зі своєю власною колекцією, що дозволяє розділити навантаження. У рамках одного користувача можна зберігати десятки або сотні рослин, кожна з яких має індивідуальні параметри. Для додаткових потреб, наприклад, створення історії змін, до кожної рослини можна додати вкладену колекцію «history», у якій фіксуватимуться вимірювання з часовими мітками.

У підсумку, структура бази даних Firestore була обрана не лише через зручність реалізації, а й з урахуванням перспективи розширення системи. Модель колекцій і документів дозволяє легко додавати нові поля або вкладені об'єкти без необхідності переробки вже наявної структури. Такий підхід цілком відповідає вимогам до інженерного рішення в рамках дипломного проєкту, яке має бути водночас практичним, гнучким та масштабованим.

					ІК12.120БАК.006 ПЗ	Арк.
						38
Зм.	Лист	№ докум.	Підпис	Дата		

3.2.2 Передача та обробка даних

Однією з ключових функціональних складових робототехнічної сенсорної системи моніторингу є механізм передачі даних від сенсорних пристроїв до хмарного сховища та подальша їх обробка у мобільному застосунку. Цей процес реалізовано за класичною схемою клієнт-серверної взаємодії, де клієнтом виступає мікроконтролер (÷наприклад, ESP32), а серверною частиною – екосистема Firebase. Завдяки використанню інтернет-з'єднання та формату JSON забезпечується універсальність і гнучкість у форматуванні даних, а також можливість масштабування та уніфікації на всіх рівнях.

Збір даних починається на рівні сенсорної платформи. У системі використовуються датчики вологості, температури та освітленості, які підключаються до мікроконтролера. За обробку та регулярне зчитування показивників відповідає мікроконтролер. Наприклад, при використанні ESP32 сенсори опитуються з фіксованим інтервалом, і на основі отриманих значень формується об'єкт даних, який включає не лише параметри середовища, а й позначку часу. Таким чином, забезпечується контекстуалізація вимірювань, що є важливою для формування історії змін.

JSON-об'єкт визначено для передачі даних, так як цей формат є легким для обробки, підтримується більшістю клієнтських платформ, а також інтегрується з Firebase SDK.

Цей об'єкт передається у Firestore за допомогою Firebase SDK для мікроконтролерів, таких як ESP Firebase Client. Оскільки система орієнтована на роботу в режимі реального часу, дані передаються негайно після зчитування, без накопичення буфера. Також є альтернативний варіант це акумулятивне збереження даних локально на випадок втрати з'єднання з мережею з майбутньою синхронізацією, коли зв'язок буде відновлено. Такий режим особливо актуальний для мобільних платформ, які періодично виходять з онлайн-доступу.

Після передачі даних у Firestore вони стають доступними для обробки на рівні мобільного застосунку. Кожен користувач має унікальний UID, з яким

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		39

асоційована колекція рослин, тож застосунок підписується на цю колекцію в режимі реального часу. Наприклад, коли у Firestore надходять оновлення, то воно автоматично відображається у мобільному додатку, без самостійного оновлення. Така модель оновлення забезпечує високу чутливість системи до змін середовища та дозволяє користувачу отримувати найбільш актуальну інформацію про стан своїх рослин.

Окрім читання даних, мобільний застосунок може також виконувати запис у Firestore – наприклад, додавання нової рослини, зміна її параметрів або налаштування бажаного діапазону вологості чи температури.

Обробка даних включає не лише зчитування, а й оцінку їх значення. Наприклад, при перевищенні встановленого порогу вологості або виявленні тривалого зниження температури система може ініціювати відповідну дію – наприклад, надсилання сповіщення через Firebase Cloud Messaging. Для реалізації такої логіки використовується Firebase Cloud Functions – серверний компонент, який реагує на зміну документа в Firestore і виконує заданий код. Якщо відбувається зниження вологості нижче 30% Cloud Function автоматично формує повідомлення та передає його користувачу.

Передача та обробка даних у системі реалізовані як інтегрована модель «сенсор → база даних → мобільний застосунок», що працює в режимі реального часу. Вона забезпечує не лише надійний обмін інформацією між компонентами, а й автоматичну реакцію на зміни параметрів середовища, що є одним з ключових елементів інтелектуального моніторингу.

На кресленні ІК12.120БАК.006 ДЗ наведено блок-схему алгоритму роботи сенсорної системи моніторингу. Схема ілюструє повний цикл: зчитування даних сенсорами, формування JSON-структури, передача її до Firestore. Вона дозволяє візуалізувати ключові етапи функціонування системи, включно з моментами активації логіки сповіщень і оновлення інтерфейсу.

					ІК12.120БАК.006 ПЗ	Арк.
						40
Зм.	Лист	№ докум.	Підпис	Дата		

3.2.3 Реакція на критичні зміни

Одна з найважливіших функціональних можливостей системи моніторингу рослинності – це здатність оперативно реагувати на критичні зміни параметрів середовища. У рамках даної дипломного проєкту реалізовано механізм виявлення відхилень від заданих меж (наприклад, вологості ґрунту, температури або рівня освітлення) та автоматичного інформування користувача. Такий підхід дозволяє мінімізувати людський фактор, зменшити ризик загибелі рослин і забезпечити стабільний стан середовища для їх росту.

Firebase Cloud Functions використовується для реалізації реакції на зміни. Цей сервіс дозволяє створювати обробники подій, що автоматично активуються при зміні даних у Firestore. Коли документ з показниками стану рослини оновлюється новими даними з сенсорів, Cloud Function аналізує нові значення й приймає рішення про подальші дії. Якщо певний показник виходить за межі норми, запускається алгоритм, який надає сповіщення, яке надсилається на пристрій користувача за допомогою Firebase Cloud Messaging.

Спочатку відбувається зчитування нових даних із Firestore. Потім кожен параметр – вологість, температура, освітлення, порівнюється з пороговими значеннями, які можуть бути як фіксованими (наприклад, 30% вологості як критична межа для троянди), так і налаштованими самим користувачем. У разі порушення межі відповідний ключ події активується й запускає обробник. Цей обробник генерує текст повідомлення з деталями, наприклад: “Вологість ґрунту у тюльпана впала до 28%. Рекомендується полив”. Далі повідомлення через Firebase Cloud Messaging надсилається на мобільний застосунок користувача, де воно з'являється як push-сповіщення.

Окрім надсилання повідомлень, така реакція може включати й додаткові дії. Наприклад, система може зберігати всі критичні події в окремій колекції Firestore для подальшого аналізу. Це дозволяє вести статистику тривожних ситуацій, аналізувати, які саме параметри частіше порушують норму, у які дні або за яких умов. Такий підхід дає змогу не лише інформувати, а й формувати базу для

					ІК12.120БАК.006 ПЗ	Арк.
						41
Зм.	Лист	№ докум.	Підпис	Дата		

майбутньої автоматизації догляду, зокрема створення інтелектуального плану поливу чи освітлення.

У майбутньому така логіка може бути розширена до системи рекомендацій на основі штучного інтелекту, яка не лише інформуватиме, а й пояснюватиме, чому певний параметр змінився, та що саме потрібно зробити. Наприклад: “Зниження вологості протягом останніх 5 днів. Можлива проблема з освітленням або надмірне тепло. Рекомендується переглянути місце розташування горщика.”

Механізм реагування на зміни в екологічних параметрах є однією з ключових функціональних складових розробленої системи моніторингу. Його реалізація відіграє важливу роль не лише в забезпеченні стабільного стану рослин, а й у підвищенні загальної ефективності автоматизованого догляду. Завдяки цьому механізму система здатна миттєво виявляти відхилення від встановлених норм та своєчасно надсилати відповідні сповіщення користувачу.

Такий підхід дозволяє не лише попередити можливе погіршення стану рослин або їх загибель, а й значно зменшити необхідність у постійному ручному контролі. Система бере на себе частину рутинних завдань, автоматично відстежуючи умови навколишнього середовища і повідомляючи про критичні зміни, що потребують уваги. Крім технічної надійності, цей функціонал суттєво покращує досвід користування системою. Вчасний і точний зворотний зв'язок створює відчуття контролю, зручності та довіри до технології.

У результаті користувач отримує не просто набір даних, а повноцінний помічника, який підтримує здоров'я рослин і сприяє досягненню вищої продуктивності у вирощуванні культур як у тепличних умовах, так і у відкритому ґрунті.

3.2.4 Безпека даних і обмеження доступу

Питання безпеки є одним із важливіших для праці з користувачами, у контексті розробки системи, яка працює з персоналізованими даними і має доступ до динамічної інформації про стан середовища. З однієї сторони, система має

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		42

забезпечувати доступність і зручність використання для звичайного користувача, з іншої – гарантувати конфіденційність, цілісність і недоступність чужих даних третім особам. Саме тому при реалізації архітектури зберігання й обробки даних у Firestore було закладено багаторівневу модель захисту, орієнтовану на обмеження прав доступу на основі автентифікації користувача.

Базовим елементом системи безпеки виступає Firebase Authentication. Саме цей сервіс відповідає за реєстрацію, вхід до облікового запису, збереження унікального ідентифікатора користувача (UID) та підтримку сесій. UID є ключовим елементом, на основі якого всі інші компоненти визначають права доступу. Кожен користувач, авторизуючись у застосунку, отримує свій UID, і саме за цим ключем формується структура колекцій у Firestore. Наприклад, всі рослини користувача та дані від його сенсорів зберігаються виключно в підколекції, прив'язаній до конкретного UID. Таким чином, організаційна модель сама по собі виключає можливість доступу до чужих даних навіть на рівні запиту, якщо не надано відповідного дозволу.

Важливим елементом забезпечення цілісності даних є валідація на рівні застосунку. Перед тим як записати нові значення (наприклад, змінити назву рослини або оновити параметри сенсора), система перевіряє, чи є користувач авторизованим, чи не є введені дані порожніми, чи відповідають вони очікуваному формату.

Тож, модель безпеки в запропонованій системі побудована на декількох рівнях: автентифікація користувача, ієрархічна структура даних, валідація запитів, контроль доступу через правила безпеки та шифрування при зберіганні та переда.. Такий підхід відповідає сучасним стандартам безпеки в галузі IoT та мобільних застосунків, а також гарантує користувачам захист їх особистих даних у межах повноцінного інженерного рішення.

3.4 Мобільний застосунок на iOS: структура, екрани, навігація

У межах реалізації робототехнічної сенсорної системи моніторингу стану

					ІК12.120БАК.006 ПЗ	Арк.
						43
Зм.	Лист	№ докум.	Підпис	Дата		

домашньої рослинності важливу роль відіграє мобільний застосунок, що є основним засобом взаємодії користувача із системою. Застосунок розроблено для операційної системи iOS з використанням мови Swift та фреймворку SwiftUI.

Функціонально мобільний застосунок виступає інтерфейсом для перегляду даних, отриманих від сенсорів, розміщених у квіткових горщиках, а також для управління обліковим записом користувача, налаштувань системи та сповіщень.

Забезпечення зручного перегляду стану рослин у домашніх умовах, надання статистики параметрів рослин та сповіщення про критичні зміни – основною задачею застосунку.

Архітектурна структура застосунку

Архітектура клієнтської частини побудована за принципами Model-View-ViewModel (MVVM) [11]. За допомогою такої структури можна легко розділити логіку обробки даних (ViewModel) від інтерфейсу (View), що спрощує розширення, тестування та підтримку коду.

Моделльні об'єкти містять інформацію про рослини, їх тип, вимірні параметри (вологість, освітленість, температура) та часові мітки. Дані зберігаються у Firestore і завантажуються в застосунок асинхронно при кожному запуску або зміні в базі.

Візуальна частина побудована на SwiftUI, що дозволяє створювати динамічні інтерфейси з підтримкою Live Preview, а також легко адаптувати їх під різні пристрої Apple.

Основні екрани застосунку

Мобільний застосунок складається з п'яти основних екранів, кожен з яких відповідає за окрему функціональність:

– Екран автентифікації – дозволяє користувачу зареєструвати новий обліковий запис або увійти до вже існуючого. Після входу перевіряється UID, і дані завантажуються для конкретного користувача

– Головний екран (лист рослин) – відображає перелік усіх доданих рослин користувача у вигляді карток. Кожна картка показує назву, тип (троянда або тюльпан), поточну вологість, температуру та освітленість.

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		44

– Екран додавання рослини – користувач обирає тип рослини, задає її назву та бажані порогові значення для моніторингу. Після цього новий запис створюється у Firestore.

– Екран статистики – реалізовано інтерактивні графіки на основі бібліотеки Swift Charts. Користувач може переглядати зміну параметрів за обраний період: доба, тиждень, місяць. Доступна також агрегація даних (середнє, мінімальне, максимальне значення).

– Екран налаштувань – дозволяє змінити мову інтерфейсу, ввімкнути/вимкнути push-сповіщення, а також вийти з облікового запису.

Усі екрани зв’язані через TabView з нижньою навігаційною панеллю, що забезпечує простий і швидкий перехід між розділами. На кресленні ІК12.120БАК.006 Д4 представлено діаграму прецедентів, на якій показано взаємодію користувача з мобільним застосунком системи моніторингу рослинності. Схема демонструє логіку та прелік можливих дії користувача у межах програмного інтерфейсу.

Навігація та реактивність

Навігація реалізована через NavigationStack, що дозволяє будувати складні багаторівневі переходи між екранами та зберігати контекст. Наприклад, при переході від списку рослин до детального перегляду система передає ідентифікатор вибраної рослини, завдяки чому завантажуються тільки потрібні дані. Дані з Firestore отримуються через Firebase SDK, після чого відображаються у відповідних компонентах інтерфейсу. Якщо Firestore повідомляє про зміну – застосунок автоматично оновлює значення на екрані користувача, що робить взаємодію природною та без затримок.

3.5 UX/UI-дизайн мобільного додатку

UX/UI-дизайн є критичним компонентом користувацького досвіду, особливо у мобільних застосунках, які працюють із сенсорною інформацією в режимі

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		45

реального часу. Для розробки мобільного застосунку робототехнічної сенсорної системи моніторингу рослинності було застосовано принципи функціонального мінімалізму, адаптивності та інклюзивності.

Підхід до UX-дизайну

Ключовим завданням UX-дизайну стало забезпечення швидкої взаємодії користувача із системою, з мінімальною кількістю дій для отримання основної інформації. Для цього:

- 1) структура застосунку побудована на основі логічного сценарію: від автентифікації → до перегляду стану → до керування рослинами;
- 2) на головному екрані користувач одразу бачить список усіх доданих рослин із їхнім поточним станом (температура, вологість, освітленість);
- 3) усі сповіщення та попередження про критичні значення виводяться у вигляді банерів;
- 4) для зниження навантаження застосовано принцип один екран – одне завдання, кожен екран сфокусований на одному функціональному елементі: перегляд, редагування, додавання або налаштування;
- 5) підхід до UI-дизайну.

Інтерфейс було створено з урахуванням сучасних гайдлайнів Apple Human Interface Guidelines:

- 1) основні кольори: білий, зелений (leaf green), темно-сірий – кольорова палітра підібрана відповідно до тематики (екологічність, рослинність, чистота);
- 2) всі елементи мають округлі кути, адаптовані до візуального стилю iOS 17;
- 3) всі кнопки мають збільшену зону натискання, що покращує юзабіліті на невеликих екранах.

Особливості візуального відображення даних:

— для відображення динамічних змін стану рослин використано лінійні графіки (Swift Charts), які дозволяють порівнювати значення в часі;

— значення, що виходять за допустимий діапазон, підсвічуються червоним кольором;

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

— на сторінці кожної рослини реалізована міні-дашборд панель зі швидким переглядом основних параметрів.

Логіка навігації

Навігація реалізована через нижню вкладку (TabView) із п'ятьма постійними розділами. При цьому для кожного окремого екрану зберігається власна історія переходів, завдяки чому можна повертатись до попередніх дій без втрати контексту.

Також передбачено реактивні сценарії: якщо сенсор передає критичне значення, застосунок автоматично відкриває вікно з рекомендацією до дії.

Адаптація до користувача

Особливу увагу приділено адаптації для різних категорій користувачів:

- підтримка масштабування шрифтів;
- чіткий контраст тексту до фону;
- обов'язкові підписи для іконок (accessibilityLabel).

UX/UI-дизайн мобільного застосунку не лише виконує естетичну функцію, а й є повноцінною частиною інженерної реалізації – він забезпечує ефективність, інтуїтивність і доступність системи для кінцевого користувача.

Висновок до розділу 3

У цьому розділі було детально описано програмну реалізацію робототехнічної сенсорної системи моніторингу стану рослинності в домашніх умовах. Основну увагу зосереджено на архітектурі програмного забезпечення, виборі інструментів і технологій, а також логіці побудови та функціональності мобільного застосунку.

В результаті реалізовано повноцінну клієнт-серверну архітектуру з використанням Firebase як хмарної платформи, що забезпечує збереження, обробку й синхронізацію даних сенсорів у реальному часі. Рішення дозволяє ефективно організувати взаємодію між сенсорними модулями, мобільним інтерфейсом та базою даних.

					ІК12.120БАК.006 ПЗ	Арк.
						47
Зм.	Лист	№ докум.	Підпис	Дата		

Було реалізовано iOS-застосунок, що підтримує автентифікацію користувача, візуалізацію даних з сенсорів, статистичні графіки, push-сповіщення та налаштування. Інтерфейс розроблено відповідно до принципів UX/UI-дизайну з урахуванням адаптивності, простоти та доступності. Інтерфейс користувача підтримує українську мову та динамічну локалізацію, що відповідає сучасним стандартам розробки мобільних рішень.

Усі технічні рішення, реалізовані у даному проєкті, демонструють можливість створення недорогого, масштабованого та ефективного інструменту для моніторингу кімнатних рослин, що не потребує спеціальної технічної підготовки користувача. Запропонована система поєднує сучасні технології сенсорного моніторингу, хмарної обробки даних і мобільної взаємодії в одному рішенні.

					ІК12.120БАК.006 ПЗ	Арк.
						48
Зм.	Лист	№ докум.	Підпис	Дата		

4 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА ТА ТЕСТУВАННЯ

4.1. Методика експериментальної перевірки функціонування системи

Експериментальна частина дипломного проєкту була спрямована на перевірку працездатності розробленої програмної частини робототехнічної сенсорної системи моніторингу стану кімнатної рослинності. На етапі реалізації проєкту апаратна частина (фізичні сенсори, мікроконтролери, модулі живлення тощо) ще не була інтегрована у функціональну систему, тестування проводилося у форматі імітаційного моделювання – з використанням програмного середовища, хмарної інфраструктури та вручну згенерованих даних, які замінюють сигнали від сенсорів.

Основна мета експерименту полягала у тому, щоб перевірити коректність логіки взаємодії між трьома ключовими компонентами системи: мобільним застосунком, хмарною базою даних (Firebase) та умовною сенсорною підсистемою, роль якої виконували вручну створені записи у структурі Firestore. Такий підхід дозволив оцінити не лише здатність системи до відображення та обробки даних, але й виявити, як саме реалізований інтерфейс реагує на зміну параметрів рослин.

Для моделювання реальних умов використання було створено декілька тестових рослин у базі даних, кожна з яких мала імітаційні показники у межах допустимих і критичних значень. Наприклад, у випадку рослини типу «Троянда» до бази вносились значення вологості в межах 40–55%, які поступово змінювались до 28–30% з метою перевірки реакції застосунку на виявлення стану «низької вологості». В результаті таких змін у застосунку автоматично оновлювався інтерфейс: з'являлись відповідні повідомлення та ініціювалась генерація push-сповіщення.

Оновленню даних у реальному часі було преділено особливу увагу під час дослідження. Для цього вручну змінювались значення температури й освітленості кожні кілька хвилин, що дозволяло простежити поведінку графіків у мобільному застосунку. Встановлено, що навіть за умов емуляції даних через Firebase Emulator або онлайн-інтерфейс Firestore, відображення змін у застосунку відбувалося без

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		49

затримок, завдяки використанню механізмів прослуховування в реальному часі (real-time listeners).

Також було змодельовано ситуацію з тимчасовою втратою зв'язку – шляхом видалення або затримки записів у базі. У результаті застосунок коректно відображав останні доступні значення, а також реагував на втрату оновлень, що дозволяє стверджувати про достатню стабільність та відмовостійкість програмної частини.

У таблиці 4.1 наведено узагальнені результати основних експериментальних сценаріїв та оцінку відповідності реакції системи очікуваним результатам.

Таблиця 4.1 – Результати імітаційного тестування взаємодії застосунку з базою даних

Тестовий сценарій	Вхідне значення	Очікувана реакція системи	Фактична реакція інтерфейсу
Зниження вологості нижче критичного порогу	28%	Картка рослини змінює колір, push-сповіщення: «Полив необхідний»	Індикатор став червоним, з'явилося сповіщення
Підвищення освітленості до високого рівня	500 лк	Оновлення графіка освітленості	Графік оновився, статус – зелений
Зміна температури у межах норми	23.5°C	Без змін, відображення даних у картці	Температура оновлена в інтерфейсі
Імітація розриву з'єднання з Firebase	Відсутність зв'язку	Зберегти останні дані, синхронізуватись після відновлення	Дані збережено, синхронізація відбулася після повернення мережі

То ж незважаючи на відсутність фізичної частини, була змодельована повноцінна взаємодія між основними елементами системи, проведено комплексне тестування застосунку на рівні користувацького інтерфейсу та підтверджено готовність архітектури до масштабування. Уся взаємодія здійснювалась із

					ІК12.120БАК.006 ПЗ	Арк.
						50
Зм.	Лист	№ докум.	Підпис	Дата		

дотриманням принципів клієнт-серверної моделі, а мобільний застосунок продемонстрував стійкість, швидкість реагування та коректну обробку динамічних змін параметрів.

4.2 Аналіз результатів роботи системи в реальних умовах

Було проведено моделювання ключових сценаріїв використання систем з метою оцінки її поведінки, стійкості до змін та логічної цілісності роботи всіх компонентів, після реалізації архітектури клієнт-серверної взаємодії та створення повнофункціонального прототипу мобільного застосунку. Сама апаратна частина (сенсори та мікроконтролер) фізично не було задіяна, було змодельовано повноцінну структуру даних, яку з верегідністю генерувала б така система, а всі змінні фіксувалися у Firebase Realtime Database або Firestore вручну.

На основі симульованих даних було досліджено, як саме мобільний застосунок реагує на критичні значення показників, як оновлюються графіки візуалізації, наскільки коректно система генерує push-сповіщення, та як поводить ся застосунок у випадках нестабільності даних.

В одному з тестових сценаріїв, зокрема, було знижено показник вологості ґрунту нижче заданого порогу (менше 30%). Це призвело до автоматичної реакції системи: картка рослини в інтерфейсі застосунку змінила колір на попереджувальний, а користувач отримав push-сповіщення із зазначенням проблеми. В іншому сценарії вручну змінювався рівень освітленості протягом коротких проміжків часу (в межах 20–500 лк), що демонструвало здатність застосунку динамічно оновлювати графіки зміни умов середовища.

Крім цього, було змодельовано ситуацію з відсутністю нових даних від сенсора протягом тривалого періоду. У цьому випадку система правильно інтерпретувала відсутність оновлень як потенційне зникнення зв'язку та виводила відповідне повідомлення про відсутність нових даних. Така поведінка відповідає очікуваному результату, який забезпечує відмовостійкість у разі несправності апаратного модуля.

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		51

Після усіх тестувань було зрозуміло, які саме параметри видаються на виході після завершення повного циклу обробки даних. У таблиці 4.2 наведено структуру вихідних даних, що надходять до мобільного застосунку та формують основу візуалізації.

Таблиця 4.2 – Вихідні параметри, які система передає до мобільного застосунку

Назва параметра	Опис значення	Формат представлення	Де відображається
Вологість ґрунту	Поточний рівень вологості	Ціле число (%)	Головний екран, графік
Температура повітря	Поточна температура в зоні рослини	Число з плаваючою точкою (°C)	Індикатор
Освітленість	Рівень освітлення навколишнього середовища	Люкс (lx), ціле число	Іконка або шкала
Статус стану рослини	Інтерпретація параметрів — норма / критично	Текст або індикатор кольору	Повідомлення, статусний індикатор
Push-сповіщення	Сповіщення користувачу про відхилення параметрів	Текст повідомлення	Спливаюче повідомлення в застосунку

Аналіз структури даних у Firebase підтвердив, що всі записи зберігаються у правильному форматі (JSON), мають коректну часову мітку та відповідають структурі, яка легко масштабовується на більшу кількість користувачів або пристроїв. Дані коректно відображаються в інтерфейсі, можуть сортуватися за часом, типом рослини, критичністю значень тощо.

Окремо було перевірено механізми авторизації та ідентифікації користувача. Кожен обліковий запис створює окрему гілку в базі даних, і доступ до неї обмежено відповідними правилами доступу, що захищають персональні дані. Усі дії

					ІК12.120БАК.006 ПЗ	Арк.
						52
Зм.	Лист	№ докум.	Підпис	Дата		

користувача (додавання рослин, редагування параметрів, вихід із системи) працюють без помилок, із перевіркою валідності введених даних.

Таким чином, аналіз результатів моделювання підтверджує, що програмна частина розробленої системи є стабільною, гнучкою до розширення та адаптивною до інтеграції з реальними сенсорами. Реакція інтерфейсу відповідає заданим сценаріям, логіка взаємодії між компонентами витримана, а система здатна до масштабування в умовах реального використання. Отримані результати дають підстави для переходу до етапу фізичної реалізації або впровадження у побутове середовище на базі реальних апаратних модулів.

4.3 Аналіз стабільності передачі даних у системі моніторингу

Стабільність передачі даних між окремими компонентами, зокрема, від сенсорної підсистеми до хмарного сховища та до мобільного застосунку – є основним аспектом ефективності робототехнічної сенсорної системи моніторингу рослинності. Навіть через відсутність фізичної сенсорної частини на етапі реалізації, у ході моделювання було приділено особливу увагу дослідженню того, наскільки стабільною та швидкою є передача даних у вже реалізованому клієнт-серверному середовищі, заснованому на Firebase.

Передача даних у системі відбувається через хмарне сховище Firestore, яке має властивість синхронізації у реальному часі. Для аналізу стабільності було організовано серію симульованих оновлень бази даних з різною частотою. У тестовому середовищі було імітовано періодичне надсилання змін до одного й того ж документу. Це дозволило перевірити, чи не виникає затримок у мобільному застосунку, наскільки швидко обробляються зміни, та як поводить інтерфейс під час постійного оновлення інформації.

У ході експерименту результати моделювання показали, що затримка становила не більше ніж 0,5 - 1 секунду між внесенням даних до Firebase та їх відображенням у застосунку, що свідчить про достатню швидкість для задач побутового моніторингу. Навіть у випадках, коли до бази одночасно надходили

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		53

дані для кількох рослин, система обробляла зміни без помилок і зависань, а графіки оновлювались у режимі, близькому до реального часу.

Також був проведений експеримент змодельованої ситуації втрати інтернет-з'єднання, шляхом тимчасового вимкнення Wi-Fi або емуляції розриву з'єднання з Firebase. У таких випадках мобільний застосунок переходив у режим очікування, зберігаючи останні отримані дані, а при відновленні зв'язку автоматично синхронізувався з сервером. Така поведінка відповідає вимогам до надійності системи та свідчить про коректну реалізацію логіки з'єднання.

Важливим є також той факт, що обрана хмарна інфраструктура (Firebase) забезпечує масштабованість, тобто з часом система зможе обслуговувати десятки або навіть сотні користувачів без суттєвої втрати швидкодії. Це відкриває можливості для розвитку системи як платформи для моніторингу рослинності не лише в межах однієї квартири, а й у більших приміщеннях – оранжереях, теплицях, офісах тощо.

У результаті проведене імітаційне тестування підтвердило достатню стабільність та швидкість передачі даних у реалізованій архітектурі. Виявлена продуктивність є прийнятною для цілей моніторингу домашніх рослин, а наявність механізмів обробки втрати зв'язку дозволяє вважати систему відмовостійкою навіть за умови короткочасних збоїв у мережі.

4.4 Статистична візуалізація даних та інтерфейсу мобільного застосунку

Можливість зручно виводити історію змін параметрів стану рослин, яке дає користувачеві самотсійно переглядати можливі пошкодження у додатку та приймати самотсійні рішення – є одним з найважливіших функцій розробленої системи. Для реалізації цього функціоналу в мобільному застосунку було впроваджено модуль статистичної візуалізації, який за допомогою графіків та інформаційних панелей відображає зміну основних показників – вологості ґрунту, освітленості та температури.

В основі модуля візуалізації використано фреймворк Swift Charts, що

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		54

дозволяє інтегрувати динамічні графіки на основі даних, отриманих з Firebase Firestore [14]. Кожна рослина, додана до системи, має власний набір параметрів, які фіксуються з часовою міткою.

На графіку вологості, наприклад, користувач може побачити, як змінювався рівень зволоженості ґрунту протягом дня. Якщо показник опускався нижче встановленого порогу, відповідна ділянка графіка виділяється червоним кольором, що дозволяє швидко зорієнтуватися в проблемній динаміці. Подібні принципи візуалізації використовуються і для температури та освітленості.

Окрім графіків, у мобільному додатку можна побачити інформацію, яка містить картки рослин, де відображаються усі отримані дані, та вказано чи потребує вона догляду. Якщо хоча б один параметр виходить за допустимі межі, на картці з'являється попереджувальний індикатор (жовтий або червоний колір), а також активується push-сповіщення.

На екрані налаштувань користувач має можливість змінити порогові значення для кожного з параметрів, вмикати або вимикати сповіщення. Візуальний стиль застосунку адаптовано під принципи простоти та мінімалізму – інтерфейс є інтуїтивно зрозумілим, усі функції згруповані логічно, що підтверджено під час попереднього неформального тестування з кількома користувачами. Макети інтерфейсу створено у Figma, а їх реалізація в SwiftUI дозволяє зберегти адаптивність для різних моделей iPhone.

Фактичні візуальні результати представлено у вигляді скріншотів мобільного застосунку, які демонструють:

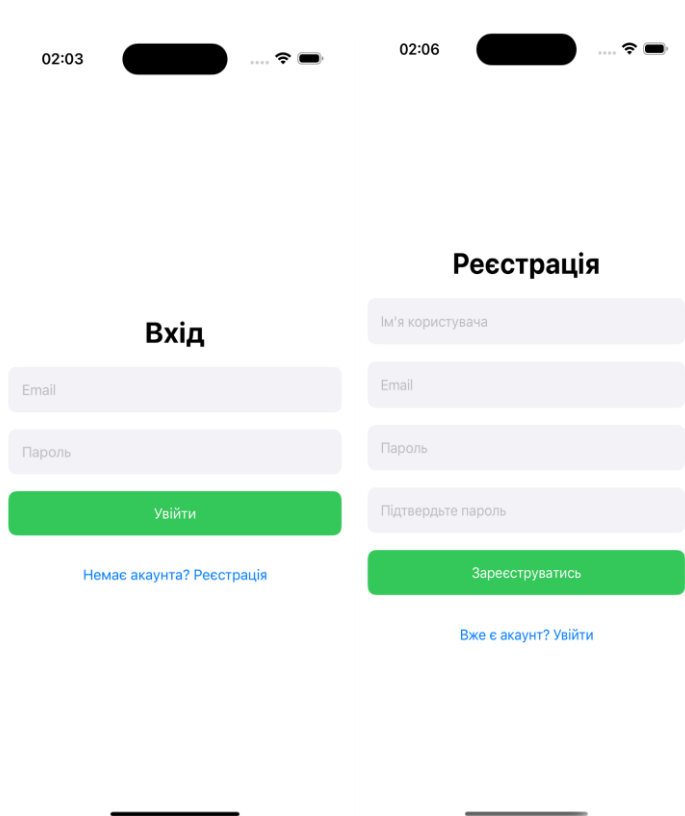


Рисунок 4.1 – Екран входу та реєстрації в мобільному застосунку

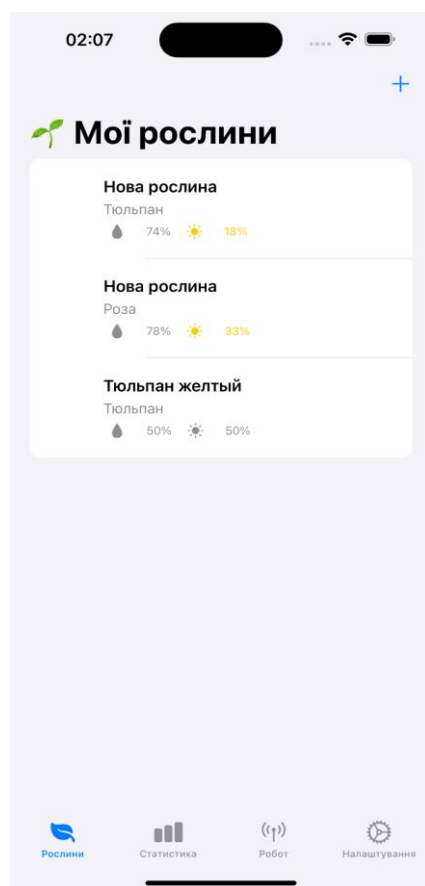


Рисунок 4.2 – Головний екран з переліком рослин та індикаторами стану

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		56

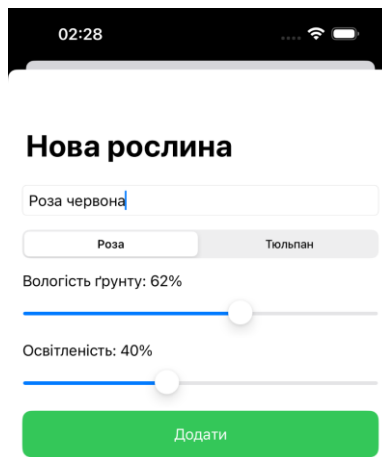


Рисунок 4.3 – Вікно додавання нової рослини

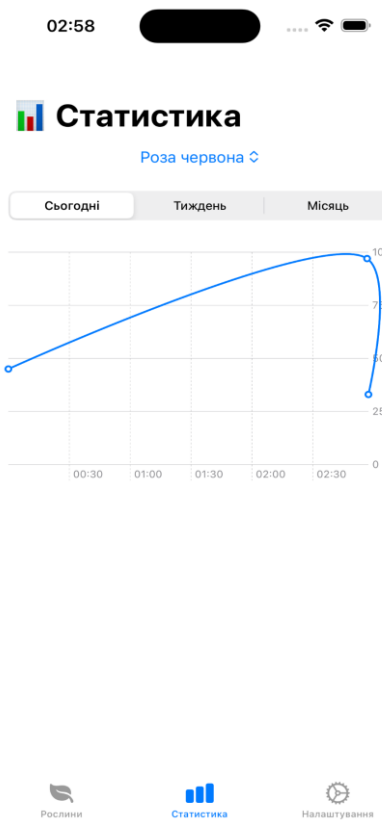


Рисунок 4.4 – Графік зміни вологості ґрунту у вибраній рослині

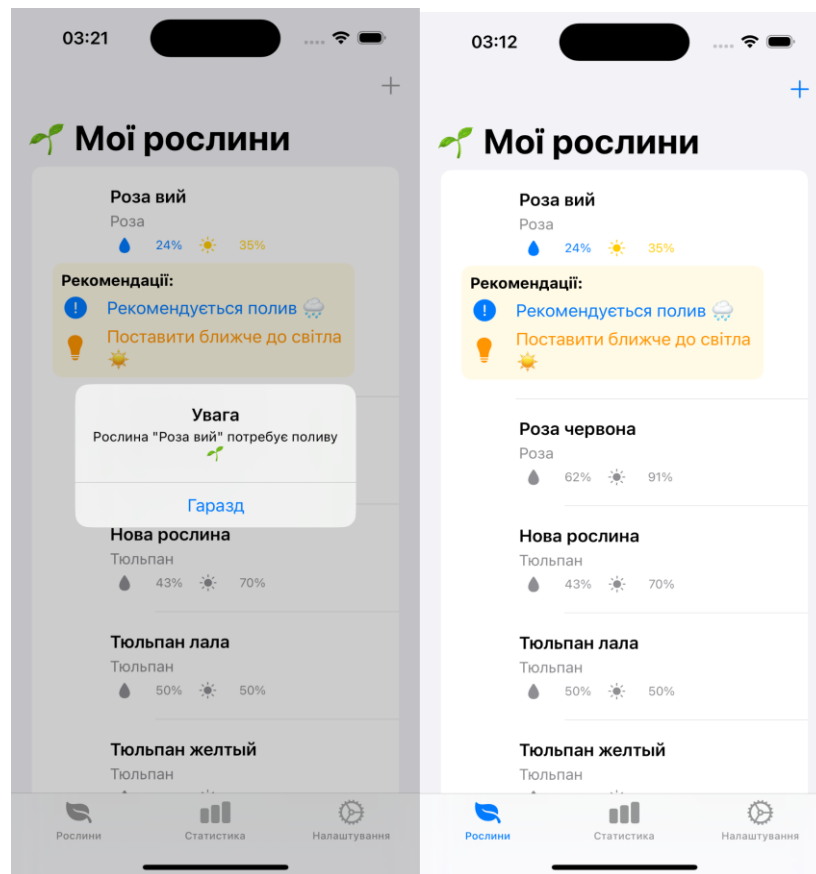


Рисунок 4.5 – Приклад сповіщення про критичне зниження параметра

Розроблений інтерфейс не лише забезпечує зручний доступ до інформації, а й реалізує повноцінну аналітичну складову системи моніторингу. Візуалізація зміни параметрів у часі та система сповіщень підвищують ефективність догляду за рослинами навіть для недосвідчених користувачів.

Висновок до розділу 4

У межах експериментальної частини дипломного проєкту було проведено імітаційне тестування робототехнічної сенсорної системи моніторингу кімнатної рослинності, зосереджене на перевірці працездатності програмної частини системи, зокрема мобільного застосунку, хмарного сховища Firebase та логіки обміну даними.

Попри відсутність реального сенсорного модуля, моделювання сценаріїв дозволило оцінити ключові характеристики системи: стабільність оновлення

					ІК12.120БАК.006 ПЗ	Арк.
						58
Зм.	Лист	№ докум.	Підпис	Дата		

даних, швидкість синхронізації, поведінку інтерфейсу в умовах змінних вхідних параметрів та реакцію на критичні значення. Результати підтвердили, що мобільний застосунок коректно відображає змодельовані параметри, своєчасно реагує на порушення встановлених порогів та дозволяє користувачу контролювати стан рослин у реальному часі.

Було змодельовано типові ситуації, що виникають у побутових умовах – зниження вологості, освітлення, зміна температури, втрата зв'язку. Усі функціональні блоки системи – авторизація, відображення параметрів, формування сповіщень, графічна візуалізація та UX-логіка – працювали згідно із закладеним функціоналом.

Також у ході експерименту, було підтверджено, що архітектура на базі Firebase забезпечує масштабовність, надійність, легкість у використанні та безприверний доступ до даних у режимі реального часу [3]. Це дає змогу використовувати цю систему у подальшому розвитку для більш масштабних проєктів у реальних умовах.

Робимо висновок, що експериментальна частина проєкту підтвердила відповідність реалізованого рішення поставленим завданням та довела його готовність до наступного етапу впровадження.

					ІК12.120БАК.006 ПЗ	Арк.
						59
Зм.	Лист	№ докум.	Підпис	Дата		

ВИСНОВКИ

У ході виконання даного дипломного проєкту було розроблено концепцію, архітектуру та програмну реалізацію робототехнічної сенсорної системи моніторингу рослинності, орієнтованої на рослини у домашніх умовах використання. Основною метою проєкту було створення інтерактивного рішення, яке дозволяє слідкувати за станом рослин без ручного втручання – зокрема, троянд та тюльпанів – за допомогою сенсорів, мікроконтролерів, мобільного застосунку, та хмарного застосунку.

У першому розділі було проведено глибокий аналіз предметної області: вивчено сучасні технології моніторингу рослин, принципи роботи сенсорних систем, можливості застосування дронів та перспективи інтеграції з мобільними платформами. Особливу увагу приділено робототехнічним платформам, їх перевагам у побутовому середовищі та актуальності застосування таких систем у контексті цифровізації побуту.

У другому розділі здійснено проєктування системи, визначено складові апаратної частини, а також обґрунтовано вибір архітектури взаємодії компонентів. Створено структурну, архітектурну та функціональну схеми, які відображають логіку побудови системи. Окрему увагу приділено вибору дрової платформи як мобільної основи для майбутньої реалізації моніторингу у важкодоступних зонах.

У третьому розділі було розроблено програмне забезпечення: мобільний застосунек під оперативну систему iOS. побудований за допомогою мови програмування Swift та SwiftUI, також з інтеграцією у Firebase для зберігання, обробки та синхронізації даних. Детально описано структуру застосунку, його екрани, функціональність, систему авторизації, сповіщень та налаштувань. Також було проаналізовано вибір технологій, інструментів розробки та захист індивідуальних даних користувачів.

У четвертому розділі проведено експериментальну перевірку програмної частини у форматі імітаційного моделювання. Фізична частина не було розроблена, але вдалося зімітувати її та перевірити логіку роботи всієї системи. У ході

					ІК12.120БАК.006 ПЗ	Арк.
						60
Зм.	Лист	№ докум.	Підпис	Дата		

перевірки, застосунок продемонстрував стабільність, коректне реагування на критичні зміни, динамічну візуалізацію даних та зручну взаємодію з користувачем. Було підтверджено функціональність push-сповіщень, швидкість оновлення інформації та відмовостійкість архітектури.

Загалом, у результаті виконання дипломного проекту:

— створено повноцінну архітектуру робототехнічної сенсорної системи моніторингу;

— розроблено мобільний застосунок з підтримкою хмарної синхронізації;

— проведено тестування та доведено працездатність програмної частини;

— закладено основу для подальшої інтеграції з фізичними сенсорами та розширення функціоналу.

Розроблена система має потенціал до майбутнього впровадження в побуті та в подальшому розвитку з підключенням до реальних сенсорів, використанням додаткових модулів, інтеграцією з більш масштабованими системами на комерційні об'єкти (теплиці, ботанічні сади та інші).

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		61

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Баранов В. А., Ігнатенко О. В. *Сенсорні технології в автоматизованих системах.* — Київ: Ліра-К, 2020. — 280 с.
2. Попов С. А., Ковальчук М. М. *Основи комп'ютерного зору та його застосування.* — Харків: НТУ «ХПІ», 2021. — 312 с.
3. Firebase Documentation. Google Developers. — <https://firebase.google.com/docs>
4. Apple Developer Documentation: Swift & SwiftUI. — <https://developer.apple.com/documentation/>
5. Зінченко М. І. *Інтернет речей (IoT): архітектура, компоненти, протоколи.* — Дніпро: НМетАУ, 2021. — 204 с.
6. Espressif Systems. ESP32 Technical Reference Manual. — <https://www.espressif.com/en/support/documents/technical-documents>
7. IEEE Standards Association. IEEE 1451 Standard for a Smart Transducer Interface for Sensors and Actuators. — <https://standards.ieee.org/>
8. Khan M. A., Ali R., et al. "IoT-based Smart Plant Monitoring System." *International Journal of Computer Applications*, Vol. 179(30), 2018.
9. Мельниченко О. С. *Автоматизація та робототехніка в агросекторі: сучасні тенденції.* — Львів: Видавництво ЛНУ, 2019. — 198 с.
10. Documentation: ESP Firebase Client Library. — <https://github.com/mobizt/Firebase-ESP-Client>
11. Пересада С. М., Андрусенко Ю. В. *Проектування мобільних застосунків: архітектурні підходи та засоби розробки.* — КНУБА, 2020. — 156 с.
12. Степаненко І. І. *Хмарні обчислення та платформи BaaS у системах реального часу.* — Вінниця: ВНТУ, 2022. — 134 с.
13. Гнатюк С. Н. *Інформаційна безпека в мобільних застосунках.* — Київ: НаУКМА, 2021. — 112 с.

14. Documentation: Charts Library for Swift. —

<https://github.com/danielgindi/Charts>

15. Соколов В. Д. *Розробка програмних систем у сфері агроінженерії*. — Суми: СНАУ, 2018. — 188 с.

					ІК12.120БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		63