

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

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 20__ р.

**Дипломний проєкт
на здобуття ступеня бакалавра
за освітньо-професійною програмою «Програмне забезпечення інформаційно-
комунікаційних систем»
спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Автоматизована система вирощування рослин, чутливих до
абіотичних факторів»**

Виконав:

студент IV курсу, групи ІТ-61

Тарасенко Михайло Валерійович _____

Керівник:

асистент кафедри АУТС

Бердник Юрій Михайлович _____

Рецензент:

доцент кафедри АСОіУ, к.т.н., доцент

Жданова Олена Григорівна _____

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

Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – перший (бакалаврський)

Напрямок підготовки – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Олександр РОЛІК

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студенту

Тарасенку Михайлу Валерійовичу

1. Тема проєкту «Автоматизована система вирощування рослин, чутливих до абіотичних факторів», керівник проєкту Бердник Юрій Михайлович, асис., затверджені наказом по університету від «7» травня 2020 р. №1081-с
2. Термін подання студентом проєкту 9.06.2020 _____
3. Вихідні дані до проєкту: аутентифікація, авторизація, можливість спостереження за гроубоксом в реальному часі, перегляд статистики по гроубоксу, зміна налаштувань гроубокса, створення профілів гроубокса.
4. Зміст пояснювальної записки: аналіз інсуючих рішень, постанова вимог до системи, вибір технологій, розробка архітектури системи, реалізація системи
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма класів серверу, діаграма прецедентів, схема реляційної бази даних, діаграма послідовності
7. Дата видачі завдання 6.03.2020 _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд та аналіз засобів керування та методів моніторингу абіотичних факторів при вирощуванні рослин	13.04 – 19.04	
2	Розроблення структурної схеми системи керування та моніторингу абіотичних факторів при вирощуванні рослин	20.04 – 21.04	
3	Розроблення функціональної схеми системи керування та моніторингу абіотичних факторів при вирощуванні рослин	22.04 – 25.04	
4	Вибір технологій, які будуть використовуватися в побудові системи	26.04 – 29.04	
5	Розроблення діаграми послідовності системи керування та моніторингу абіотичних факторів при вирощуванні рослин	30.04 – 01.05	
6	Розроблення структури та проєктування бази даних	01.05 – 06.05	
7	Розроблення програмного забезпечення системи керування абіотичних факторів при вирощуванні рослин	06.05 – 20.05	
8	Тестування готового програмного забезпечення на макеті	20.05 – 24.05	
9	Передзахист дипломного проєкту	25.05 – 26.05	
10	Доопрацювання пояснювальної записки	27.05 – 15.06	
11	Захист дипломного проєкту	15.06 – 19.06	

Студент

Михайло ТАРАСЕНКО

Керівник проєкту

Юрій БЕРДНИК

АНОТАЦІЯ

Тарасенко М.В. Автоматизована система вирощування рослин, чутливих до абіотичних факторів. КПІ ім. Ігоря Сікорського, Київ, 2020. Проект містить 65 с. тексту, 8 рисунків, 22 таблиці, посилання на 16 літературних джерел та 4 конструкторські документи.

Ключові слова: сервер, Android застосунок, автоматизована система, абіотичний фактор, клієнт-серверна архітектура, REST.

Об'єкт дипломного проекту: вирощування рослин чутливих до абіотичних факторів.

Предмет дипломного проекту: автоматизована система вирощування рослин, чутливих до абіотичних факторів.

Мета дипломного проекту: спростити процес вирощування рослин.

У дипломному проекті розроблено систему керування будь якою кількістю гроубоксів з можливістю керування параметрами внутрішнього середовища гроубоксів та керування групами параметрів (профілями).

Також система оснащена функцією збору та збереження статистики гроубокса для подальшого аналізу та виявлення аномалій в роботі кожного конкретного гроубокса.

Практичне значення дипломного проекту полягає в застосуванні системи для домашнього використання окремими користувачами або для керування вирощуванням рослин на фермах в промислових масштабах. Цей проект може бути використаний учасниками гуртка «ES&IoT – Вбудовані системи та інтернет речей» кафедри АУТС.

ABSTRACT

Tarasenko M.V. Automated system for growing plants sensitive to abiotic factors. Igor Sikorsky Kyiv Polytechnic Institute, Kiev, 2020. The project contains 63 p. text, 8 figures, 22 tables, a message for 16 literature sources and 4 design documents.

Keywords: server, Android device, automated system, network factor, server, REST.

Object of the diploma project: growing plants sensitive to abiotic factors.

Subject of the diploma project: automated system of growing plants sensitive to abiotic factors.

The purpose of the diploma project: to simplify the process of growing plants.

In the diploma project the system that controls massive amount of growboxes with opportunity of growbox's enviroment paramaters control and the grouping parameters control was developed.

The system is also equipped with the function of gathering and storing growboxes statistics for further analysis and revealing anomalies in each growbox operation.

The practical significance of the degree of polarization in closed systems for household use or for growing plants on farms on an industrial scale. This project can be presented by participants of the club "ES&IoT – Implementation of systems and the Internet of Things" of department of ACTS.

Номер рядка	Формат	Позначення	Найменування	Кільк. арк.	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IT61.230БАК.004 ПЗ	Пояснювальна записка	65		
6	A3	IT61.230БАК.004 Д1	Автоматизована система	1		
7			вирощування рослин, чутливих			
8			до абіотичних факторів			
9			Діаграма прецедентів			
10	A3	IT61.230БАК.004 Д2	Автоматизована система	1		
11			вирощування рослин, чутливих			
12			до абіотичних факторів			
13			Діаграма послідовності			
14	A3	IT61.230БАК.004 Д3	Автоматизована система	1		
15			вирощування рослин, чутливих			
16			до абіотичних факторів			
17			Діаграма класів серверу			
18	A3	IT61.230БАК.004 Д4	Автоматизована система	1		
19			вирощування рослин, чутливих			
20			до абіотичних факторів			
21			Схема реляційної бази даних			
22						
23						
24						
25						
26						
27						
28						

					IT61.230БАК.004 ТП		
Зм.	Арк.	№ докум.	Підпис	Дата			
		Тарасенко М.В.					
Перевір.		Бердник Ю.М.					
Реценз.							
Н. Контр.							
Затв.							
					Автоматизована система вирощування рослин, чутливих до абіотичних факторів. Відомість технічного проєкту		
					Літ.	Аркуш	Аркушів
					Т	1	1
					КПІ ім. Ігоря Сікорського ФІОТ Група IT-61		

Пояснювальна записка
до дипломного проєкту
на тему: «Автоматизована система вирощування
рослин, чутливих до абіотичних факторів»

Київ – 2020 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, ТЕРМІНІВ.....	4
ВСТУП.....	5
1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	7
1.1 Загальний огляд предметної області	7
1.1.1 Промислові теплиці.....	7
1.1.2 Міні-теплиці.....	9
1.1.3 Гроубокси.....	10
1.2 Огляд існуючих програмних засобів керування теплицями.....	10
1.2.1 Система керування теплицею Atom.....	11
1.2.2 Система керування кліматом теплиці	12
Висновок до розділу	13
2 ПОСТАНОВКА РОЗГОРНУТИХ ВИМОГ ДО СИСТЕМИ.....	14
2.1 Вимогу до серверної частини.....	14
2.2 Вимоги до Android застосунку	15
3 ВИБІР КОМПОНЕНТІВ СИСТЕМИ	16
3.1 Обґрунтування вибору компонентів сервера	17
3.1.1 Обґрунтування вибору фреймворку ASP.NET Core.....	18
3.1.2 Обґрунтування вибору СКБД PostgreSQL.....	18
3.1.3 Обґрунтування вибору фреймворка Entity Framework	19
3.1.4 Обґрунтування вибору мови парсингу бінарних структур.....	19
3.2 Вибір середовища розроблення Android-застосунку	20
4 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	22
4.1 Загальна структура системи	22
4.2 Розроблення архітектури сервера	23
4.3 Розроблення структури бази даних	25

					IT61.230БАК.004 ПЗ			
	Лист	№ докум.	Підпис	Дата				
Розробив	Тарасенко М.В.				Автоматизована система вирощування рослин, чутливих до абіотичних факторів	Літ.	Аркуш	Аркушів
Перевірив	Бердник Ю.М.					Т	2	63
Реценз.								
Н. Контр.								
Затв.								
						КПІ ім. І. Сікорського ФІОТ Група ІТ-61		

4.4	Розроблення архітектури Android-застосунка	31
5	РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	34
5.1	Реалізація сервера	34
5.2	Реалізації бази даних	47
5.3	Реалізація протоколу зв'язку між гроубоксом та сервером	49
	ВИСНОВКИ.....	63
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	64

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ, ТЕРМІНІВ

HTTP – протокол передачі гіпер тексту.

REST – підхід до створення архітектури доступу к даним.

API – програмний інтерфейс.

CodeFirst – підхід до побудови БД, в якому спочатку описуються моделі, з яких за допомогою EntityFramework генеруються скрипти для бази даних.

DB – база даних.

Growbox – теплиця маленьких розмірів.

ПЗ – програмне забезпечення.

Технологічний стек – технології та інструменти, що були використані при розробці програмного продукту.

					IT61.230БАК.004 ПЗ	Арк.
						4
Зм.	Лист	№ докум.	Підпис	Дата		

ВСТУП

Сьогодні все більшу популярність набирає тепличний спосіб вирощування рослин. Вже існують великі промислові ферми побудовані лише на теплицях для вирощування великої кількості рослин для різноманітних підприємств та для задоволення потреб населення у їжі. Проаналізувавши ринок можна досить легко побачити, що систем по віддаленому керуванню теплицями майже нема. Тому задача розроблення автоматизованої системи керування вирощуванням рослин чутливих до абіотичних факторів складна та актуальна.

Об'єкт дипломного проєкту: вирощування рослин чутливих до абіотичних факторів.

Предмет дипломного проєкту: автоматизована система вирощування рослин, чутливих до абіотичних факторів.

Мета дипломного проєкту – спростити процес вирощування рослин.

Для досягнення поставленої мети необхідно вирішити наступні задачі:

- виконання аналізу існуючих рішень;
- виконання огляду предметної області;
- створення формату повідомлення;
- розроблення діаграми прецедентів;
- розроблення діаграми послідовності;
- розроблення діаграми класів серверу;
- розроблення схема реляційної бази даних;
- розроблення архітектури серверу;
- розроблення архітектури Android-застосунку;
- реалізація серверу;
- реалізація Android-застосунку.

Цей дипломний проєкт призначений для вирішення цієї проблеми шляхом створення застосунку для смартфонів. Таким чином, власник теплиць зможе в будь-який час за допомогою свого телефона перевірити стан теплиці та виставити

					IT61.230БАК.004 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

нові налаштування для теплиці без необхідності їхати до самої теплиці. Це було досягнуто за допомогою клієнт-серверного застосунку за основі ASP.NET Core та Android додатку в ролі клієнта на основі Kotlin.

Проєкт призначений для промислових ферм та просто людей, які захотіли розмістити в кімнаті екзотичну рослину. Система дозволить їм швидко дізнаватися про стан всередині теплиці та змінювати параметри середовища всередині теплиці у разі потреби. Цей проєкт також може бути використаний учасниками гуртка «ES&IoT – Вбудовані системи та інтернет речей» кафедри АУТС.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

1 ОГЛЯД ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

1.1 Загальний огляд предметної області

Аналіз ринку сучасних автоматизованих теплиць показав, що на сучасному ринку існує великий вибір теплиць спеціалізованих під різні цілі.

Теплиці можна поділити на промислові теплиці, міні-теплиці та гроубокси.

Таким чином промислові теплиці використовуються для вирощування рослин. Вони призначені для вирощування рослин в несприятливих умовах. Зазвичай такі теплиці мають засоби вентиляції та регулювання температури. Такі теплиці використовуються на фермах та підприємствах для вирощування рослин в великих масштабах.

Міні-теплиці, це зменшений варіант промислових теплиць і здебільшого використовуються на приватних земельних ділянках садоводами. Зазвичай їх використовують для вирощування екзотичних рослин та створення маленьких городів.

Гроубокси є ще більш малою версією теплиць та призначені для розміщення в кімнаті. Зазвичай вони мають розмір невеликого акваріума та призначені для пророщування/вирощування невеликої кількості рослин, зазвичай від однієї до 4. Такі теплиці в більшості випадків обладнанні системою провітрювання, регуляції температури та автополиву.

1.1.1 Промислові теплиці

Головним призначенням промислових теплиць є вирощування та пророщування рослин у великих кількостях з метою забезпечити сировиною різні промислові галузі, такі як, наприклад, харчові. Основним функціональними можливостями таких теплиць є автополив, підтримка необхідних для вирощування навколишніх умов, таких як: температура, вологість, кількість світла.

Теплиці можна поділити на дві групи – ті які використовують землю та гідропонні. Гідропонні теплиці вирощують рослини на спеціальній поверхні або у

					ІТ61.230БАК.004 ПЗ	Арк.
						7
Зм.	Лист	№ докум.	Підпис	Дата		

розчину, що містять всі необхідні для здорового та швидкого росту рослин речовини.

Також теплиці можна поділити за формою: багатокутні, односкілі, краплеподібні, подвійні та голландські теплиці. Різні форми обираються в залежності від середовища в якому будуть знаходитися теплиці. У теплиць різної форми по різному розподіляється тепло і це треба враховувати при створенні власної теплиці, бо може виявитися, що встановлена теплиця може бути не енергоефективною та приносити через це збитки[1].



Рисунок 1.1 – приклад теплиці від компанії ООО «Теплица Люкс»[2]

Гарним прикладом промислових теплиць, є теплиця зображена на рисунку 1.1 від компанії «Теплица Люкс»[2]. Ця теплиця використовується фермерам для вирощування великої кількості продукції та пропонує для цього багато опцій для налаштування параметрів середовища. Ця теплиця оснащена системами вентиляції, підігріву ґрунту, системами освітлювання, системами поливу.

Підігрів цієї теплиці забезпечується системами нижнього, верхнього, зонального та водяного підігріву. Контурні системи підігріву працюють

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

автоматично в залежності від датчиків температури і завдяки цьому економлять до 15% тепла на відміну систем з підгрівом без регуляції.

Вентиляція теплиці виконується завдяки квартирок, що відкриваються та закриваються в автоматичному режимі, при цьому площа цих квартирок є 25% від площі стелі теплиці. Також додатково є можливість встановити вентилятор для створення активної вентиляції та перемішування мас повітря всередині теплиці.

1.1.2 Міні-теплиці

Міні-теплиці зазвичай використовуються садоводами на приватних ділянках для вирощування рослин в малих масштабах і здебільшого є хобі і не мають цінності для промислових підприємств.

Одним з типів теплиць середнього розміру є парники, вони використовуються здебільшого для пророщування рослин або для збереження їх від снігу та низьких температур взимку.

Зазвичай це прості конструкції які не мають засобів автоматизації та систем підтримки стану середовища всередині.



Рисунок 1.2 – модель «Супер парник» [3]

Їх конструкція дозволяє підтримувати температуру всередині достатню для життя рослини.

					ІТ61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		9

1.1.3 Гроубокси

Гроубокс – це частково або повністю закрита система для вирощування рослин у приміщенні або на невеликих площах. Ящики для вирощування використовуються з кількох причин, включаючи відсутність наявного відкритого простору або бажання вирощувати овочі, трави або квіти протягом несприятливих погодних умов. Вони також можуть допомогти захистити рослини від шкідників або хвороб.

Гроубокси здебільшого невеликого розміру і призначенні для розміщенні в кімнаті. Вони призначенні для вирощування невеликої кількості рослин(від 1 до 4). Зазвичай вони використовуються для вирощування овочів в домашніх умовах. Також гроубокси використовуються для вирощування екзотичних рослин, які не можуть рости в кліматі навколишнього середовища.[4]

Гроубокси можуть бути обладнанні різноманітними системи по контролю середовищем всередині, до таких відносяться системи по контролю вологості повітря, вологості ґрунту, рівня освітленості, системи вентиляції, системи поливу, систем удобрення ґрунту та інші.

Також гроубокси можуть бути обладнанні системами по контролю темпу росту рослин на кшталт генератора CO₂ та розвинутих систем освітлення, що можуть регулювати кількість випромінюваного світла та його спектр.

1.2 Огляд існуючих програмних засобів керування теплицями

На сьогодні існує багато систем автоматичного керування теплицями, деякі з них будуть розглянуті в цьому розділі.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

1.2.1 Система керування теплицею Atom

Як можна побачити на рисунку 1.3, система керування теплицею Atom має зручний інтерфейс.



Рисунок 1.3 – Інтерфейс системи керування теплицею[17]

Інтерфейс теплиці забезпечує зручну взаємодію користувача з системою та відображає поточний стан внутрішнього середовища теплиці. На дисплеї відображаються такі параметри:

- температура;
- вологість;
- активність вентиляції.

До кожного параметра надається окремий графік, який вказує на зміни конкретного параметра внутрішнього середовища теплиці протягом заданого періоду часу. Також ця система дозволяє виставляти параметри системи, яких буде дотримуватися теплиця одразу після встановлення.

					IT61.230BAK.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

Але ця система має значний недолік: вона має бути розміщена безпосередньо біля теплиці. Такий значний недолік в автономності може заважати користувачам подорожувати на значні відстані від теплиці через неможливість спостерігати за її станом в режимі онлайн.

1.2.2 Система керування кліматом теплиці

На рисунку 1.4 зображено екрани мобільного застосунку керування теплицями, як можна побачити, у цієї системи є велика кількість функцій.

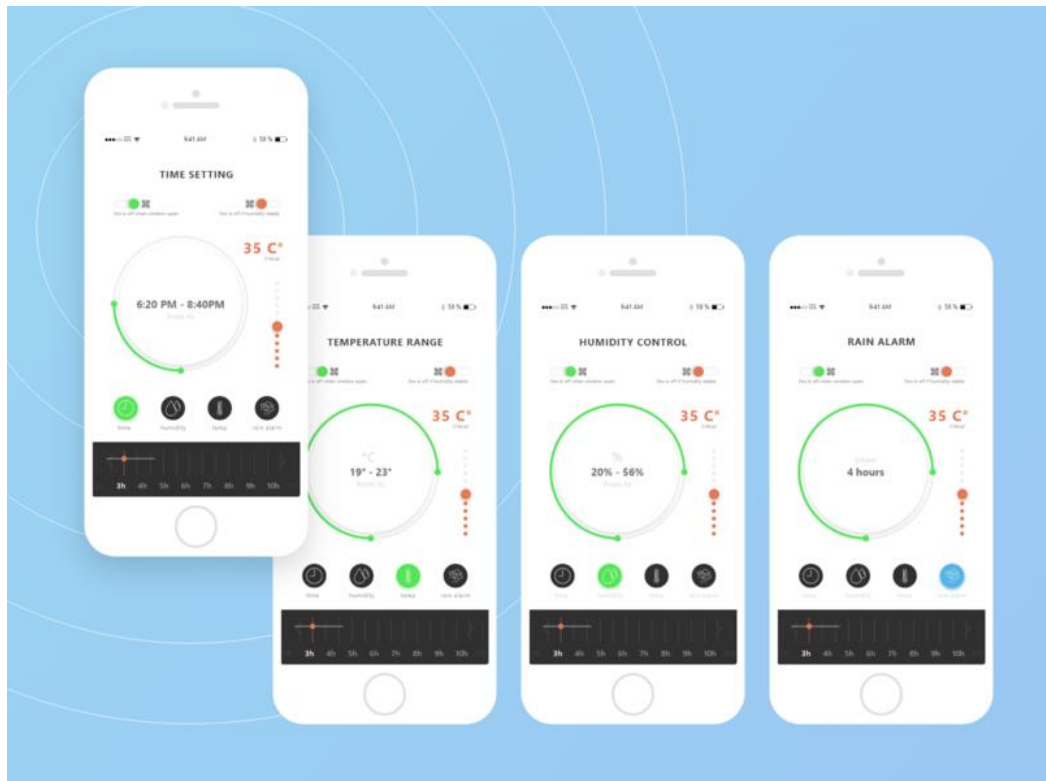


Рисунок 1.4 – Екрани системи керування параметрами теплиці

Як можна побачити на рисунку 1.4 ця система має дуже зручний інтерфейс та великий спектр функцій, а саме:

- регулювання температури;
- регулювання вологості;
- виставлення часових проміжків;

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

— сигналізація на випадок дощу.

Також як і в системі описаній в підрозділі 1.5.1, ця система надає графіки того як змінювалися параметри системи протягом вказаного інтервалу часу.

Висновок до розділу

У цьому розділі було розглянуто типи існуючих теплиць та специфіку кожного з них. В залежності від типу теплиці змінюються системи керування цими теплицями та кількість параметрів, що вони підтримують.

Як показав аналіз існуючих рішень програмних засобів керування вирощуванням рослин, чутливих до абіотичних факторів, вони надають широкі функціональні можливості зі збору статистики за параметрами всередині гроубоксів та дозволяють забезпечити віддалене керування теплицями.

Сучасні програмні засоби керування вирощуванням рослин забезпечують моніторинг наступних параметрів в середині теплиці:

- вологість;
- світловий потік;
- температура;
- вологість ґрунту.

Після аналізу ринку стало зрозуміло, що на ринку існує мало мобільних систем керування теплицями та ті які є не відповідають усім вимогам.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		13

2 ПОСТАНОВКА РОЗГОРНУТИХ ВИМОГ ДО СИСТЕМИ

2.1 Вимогу до серверної частини

Перед специфікацією вимог до серверної частини треба вести нове поняття – профіль.

Профіль гроубоксу – це збірна інформація щодо бажаного стану середовища всередині гроубоксу. Профілі включають в себе таку інформацію як вологість, температуру, інтенсивність вентиляції, вологість ґрунту, рівень освітленості. Профілі використовуються у системі для виставлення в гроубоксі параметрів внутрішнього середовища бажаних клієнтом. Клієнт може створити декілька профілів та міняти їх в Android додатку в будь-який час.

Сервер повинен містити сховище даних для статистики профілів, користувачів та гроубоксів.

Сервер повинен мати систему авторизації для забезпечення безпеки чутливих даних від несанкціонованого доступу. Система авторизації повинна враховувати декілька ролей для ефективного розмежування обов'язків користувачів системи та не дозволити операції які не мають сенсу в конкретному контексті. Таким чином в системі виділені ролі:

— адміністратор – не має обмежень у функціоналі і має доступ до усіх функцій системи. Він обслуговує систему та вирішує проблеми, які можуть з'явитися у інших користувачів;

— гроубокс – має обмеження по функціоналу та має доступ тільки до функцій, які необхідні для операцій з гроубоксами, що йому належать;

— користувач – має доступ тільки до інформації створеної гроубоксами та профілів створених користувачем.

Функції, які повинен мати користувач ролі гроубокс:

— запит активного профілю;

— відправлення статистики на сервер.

Функції, які повинен мати користувач:

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		14

- додавання гроубоксу;
- видалення гроубоксу;
- створення профілю;
- видалення профілю;
- редагування профілю;
- запит статистики по конкретному гроубоксу;
- редагування інформації облікового запису.

Користувач функції Admin має доступ до усіх функцій систем та інформації.
 Діаграма прецедентів наведена в кресленику IT61.230БАК.004 Д4.

2.2 Вимоги до Android застосунку

Android застосунок (далі просто застосунок) має підтримувати авторизацію сервера та використовувати її.

За допомогою REST запитів на сервер застосунок повинен відображати отриману інформацію та надавати графічний інтерфейс для взаємодії з нею.

Застосунок повинен мати зручний інтерфейс та надавати можливість користуватися функціоналом серверу, таким чином застосунок повинен:

- давати можливість авторизуватися на сервері;
- давати можливість переглянути гроубокси користувача;
- давати можливість переглянути статистику по кожному гроубоксу;
- давати можливість додати новий профіль;
- давати можливість змінити існуючий профіль;
- давати можливість видалити профіль;
- давати можливість створити новий профіль;
- давати можливість додати новий гроубокс;
- давати можливість видалити існуючий гроубокс.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

3 ВИБІР КОМПОНЕНТІВ СИСТЕМИ

При виборі компонентів системи треба враховувати безліч різних аспектів. І перше з чого треба почати – це мова програмування.

При підборі мови програмування треба мати на увазі усі аспекти кінцевої системи, для того щоб результат задовольняв усім потребам користувачів і при цьому був стабільним і мав запас для можливого майбутнього розширення новими функціями.

При підборі мови програмування треба перш за все виділити такі аспекти:

- тип мови програмування;
- можливості мови програмування;
- додаткові існуючі інструменти мови програмування.

Під типами мов програмування розуміються такі аспекти:

- система типізації;
- реалізовані парадигми;
- середовище виконання;
- режим доступу до оперативної пам'яті комп'ютеру.

В мовах програмування розрізняють статичну та динамічну типізації. Статична типізація дозволяє перевіряти більшість перевірок типів ще на етапі компіляції програми, що зменшує шанси на помилку в програмі при написанні. Динамічна типізація передбачає, що розробник сам буде перевіряти типи, що використовуються операція. Динамічна типізація дозволяє значно прискорити створення простих програмних продуктів за рахунок спрощеного процесу написання коду.

Зараз нараховують безліч парадигм програмування, наприклад:

- об'єктно-орієнтоване програмування;
- узагальнене програмування;
- функціональне програмування;
- узагальнене програмування;

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		16

— процедурне програмування.

Вибір парадигми програмування має велике значення, оскільки саме обрана парадигма в багатьох аспектах вирішує архітектуру застосунку.

Також мови програмування можна поділити на типи: ті що використовують віртуальну машину та ті що виконуються у середовищі операційної системи.

Ті, які компілюються в байткод, що виконуються в віртуальній машині мають велику перевагу над мовами, що виконуються у середовищі операційної системи. Ця перевага полягає в тому, що код таких мов програмування може бути без змін виконаний на будь якій іншій платформі на яких реалізована віртуальна машина. Але такі мови програмування мають суттєвий недолік – програми, які написані на цих мовах, зазвичай на порядок повільніші ніж аналогічні програми написані на мовах програмування, що виконуються у середовищі операційних систем.

3.1 Обґрунтування вибору компонентів сервера

Виходячи з вимог до проєкту, сервер повинен підтримувати HTTP запити для повноцінної реалізації REST.

Сервер повинен мати централізоване сховище даних, для цього використовується реляційна база даних з системою сутностей.

Проєкт не повинен працювати на багатьох платформах і це означає, що можна зосередитися на конкретній платформі, в даному випадку Windows.

Для швидкої розробки під Windows можна використати C#, мову програмування від компанії Microsoft розроблену спеціально для створення програм на операційну систему Windows та надають для цього .NET Framework, що регулярно оновлюється додаючи нові функції.

C# є об'єктно-орієнтованою мовою програмування з статичною типізацією створений з ціллю зробити «краще Java» але по факту це Java з іншою системою виконання програм.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		17

Головною причиною вибору мови С# є функціонал який пропонує компанія Microsoft за допомогою .NET Framework, що надає велику кількість вже готових функцій та структур даних для різноманітних потреб.

3.1.1 Обґрунтування вибору фреймворку ASP.NET Core

В якості основного фреймворку для сервера було обрано ASP.NET Core.

ASP.NET Core — вільне та відкрите програмне забезпечення каркаса вебзастосунків, з продуктивністю вищою ніж у ASP.NET, розроблена корпорацією Microsoft і співтовариством. Це модульна структура, яка працює як на повній платформі .NET Framework, так і на платформі .NET Core [8].

ASP.NET Core модульна структура розподіляється як Nuget пакунки, що дозволяє легко вбудовувати та видаляти додаткові бібліотеки та фреймворки, які знадобляться при створенні сервера.

ASP.NET Core був обраний як основний фреймворк для серверу через його великий вбудований функціонал та можливість легко будувати REST сервіси. Він надає можливість використати «з коробки» механізми авторизації, аутентифікації, використання CORS та інші.

3.1.2 Обґрунтування вибору СКБД PostgreSQL

В якості бази даних було обрано PostgreSQL.

PostgreSQL — об'єктно-реляційна система керування базами даних (СКБД). Є альтернативою як комерційним СКБД (Oracle Database, Microsoft SQL Server, IBM DB2 та інші), так і СКБД з відкритим кодом (MySQL, Firebird, SQLite).[10]

Порівняно з іншими проектами з відкритим кодом, такими як Apache, FreeBSD або MySQL, PostgreSQL не контролюється якоюсь однією компанією, її розробка можлива завдяки співпраці багатьох людей та компаній, які хочуть використовувати цю СКБД та впроваджувати у неї найновіші досягнення.[10]

					IT61.230БАК.004 ПЗ	Арк.
						18
Зм.	Лист	№ докум.	Підпис	Дата		

PostgreSQL була обрана в якості бази даних через більшу кількість функціональних можливостей ніж MySQL. PostgreSQL є об'єктно реляційною базою даних на відміну від MySQL, яка є реляційною базою даних, це означає що таблиці в базі даних можуть мати дочірні таблиці.

Також PostgreSQL дозволяє створювати часткові індекси, які наприклад дозволяють реалізувати soft delete даних.

3.1.3 Обґрунтування вибору фреймворка Entity Framework

Entity Framework обраний в якості засобу взаємодії з базою даних.

ADO.NET Entity Framework (EF) — об'єктно-орієнтована технологія доступу до даних, є object-relational mapping (ORM) рішенням для .NET Framework від Microsoft. Дає можливість взаємодії з об'єктами за допомогою LINQ в вигляді LINQ to Entities, так і з допомогою Entity SQL. Для спрощення створення web-застосунків використовується як ADO.NET Data Services (Astoria), так і зв'язка з Windows Communication Foundation та Windows Presentation Foundation, які дозволяють створювати багаторівневі застосунки, виконуючи один з шаблонів проектування MVC, MVP чи MVVM.[9]

Entity Framework дозволяє абстаргуватися від реалізації конкретних баз даних та при необхідності перейти на іншу, якщо наявна перестане відповідати вимогам проекту. Також Entity Framework надає зручні інструменти для маніпулювання даними в базах даним і робить прості операції як редагування, зчитування, додавання даних та видалення даних набагато простішими.

3.1.4 Обґрунтування вибору мови парсингу бінарних структур

Для парсингу структур даних, які прибувають з гроубоксів, мова програмування C# підходить погано.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		19

C++ буде зручним за декількох причин, одна з яких що дані надходять до серверу у вигляді бінарних даних, що потрібно розшифрувати.

Для парсингу бінарних структур даних, які використовуються у повідомленнях, якими обмінюються сервер та гроубокс, будуть зручними такі функції C++:

— `union` – спеціальний тип структур даних, який дозволяє тримати в одній самій комірці пам'яті дані різних типів, що буде дуже корисним якщо прийняти до уваги те, що бінарні дані, що надходять з гроубоксу можуть бути різного розміру та типу;

— `bitfields` – бітові поля, дозволяють створювати структури даних, які мають в собі дані з нетрадиційним для цього типу кількістю бітів.

— `bitwise operations` – операції над окремими бітами знадобляться при парсингу бінарних даних завдяки функціоналу зсувів та побітових логічних операцій.

Код C++ буде скомпільовану в окрему бібліотеку та підключено до проєкту C#.

3.2 Вибір середовища розроблення Andoid-застосунку

Для створення Android застосунку було використано середовище розробки Android Studio.

Android Studio — інтегроване середовище розробки (IDE) для платформи Android, представлене 16 травня 2013 року на конференції Google I/O менеджером по продукції корпорації Google — Еллі Паверс (англ. Ellie Powers).[13]

Середовище розробки адаптоване для виконання типових завдань, що вирішуються в процесі розробки застосунків для платформи Android. У тому числі у середовище включені засоби для спрощення тестування програм на сумісність з різними версіями платформи та інструменти для проєктування застосунків, що працюють на пристроях з екранами різної роздільності (планшети, смартфони,

					IT61.230БАК.004 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

ноутбуки, годинники, окуляри тощо). Крім можливостей, присутніх в IntelliJ IDEA, в Android Studio реалізовано кілька додаткових функцій, таких як нова уніфікована підсистема складання, тестування і розгортання застосунків, заснована на складальному інструментарії Gradle і підтримуюча використання засобів безперервної інтеграції.[13]

Для розроблення мобільного застосунку була використана мова програмування Kotlin.

					IT61.230БАК.004 ПЗ	Арк.
						21
Зм.	Лист	№ докум.	Підпис	Дата		

4 ПРОЄКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 Загальна структура системи

На рисунку 4.1 зображена загальна схема системи.

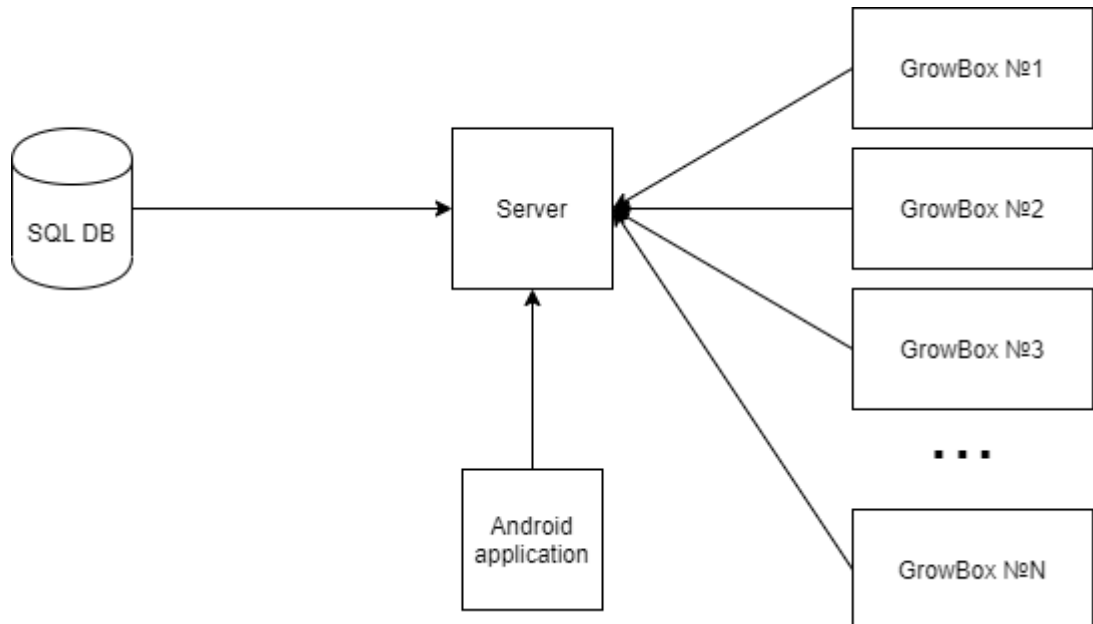


Рисунок 4.1 – Структурна схема системи

Система складається з декількох елементів, ці елементи:

- сервер;
- гроубокс;
- реляційна база даних;
- Android застосунок.

Сервер приймає головну роль в системі, він приймає роль посередника між клієнтом даними та гроубоксами. Велика кількість гроубоксів може бути підключена до сервера і керована через Android застосунок користувачем.

Дані зберігаються в базах даних та керуються сервером. Дані користувачів, профілів та гроубоксів зберігаються в реляційній базі даних PostgreSQL.

Android застосунок слугує інтерфейсом між користувачем та сервером. Він відправляє на сервер запити і, після обробки сервером, отримуючи результат

відображає на екрані смартфона користувача дані в вигляді діаграм та спеціальних графіків інформацію щодо змін в обраному грубоксі.

4.2 Розроблення архітектури сервера

Сервер використовує підхід до створення архітектури мережевих протоколів по доступу до даних REST. На рисунку 4.2 зображена загальна схема серверу.

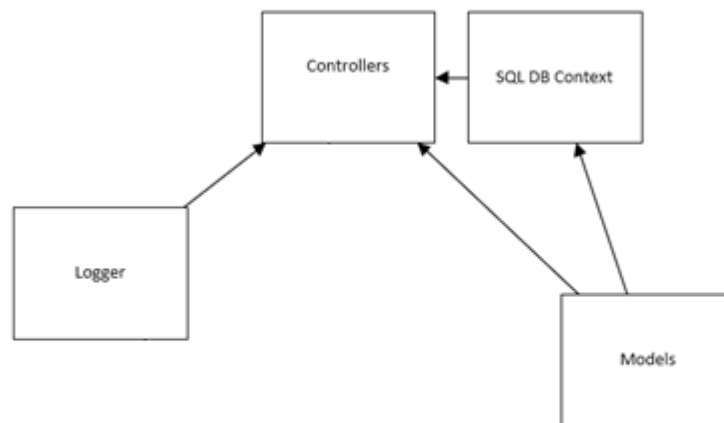


Рисунок 4.2 – Структурна схема сервера

На рисунку 4.2 можна побачити декілька основних модулів з яких складається сервер, це:

- Controllers;
- Logger;
- Models;
- SQL DB Context.

Призначенням контролерів є приймання запитів які надходять з відкритої мережі Internet в вигляді HTTP запитів. Контролери виконують керуючу функцію

серверу, вони використовують інші компоненти системи для формування відповіді клієнтам на HTTP запити в залежності від типу запиту.

Призначенням модуля Logger є логування в документоорієнтовану базу даних процесів на сервері, таких як:

- надходження запиту;
- зчитування даних користувача;
- зміна даних користувача;
- зчитування даних гроубокс;
- зміна даних гроубоксів;
- зчитування профілів;
- зміна даних профілів;
- запит статистики гроубоксу;
- запит логів серверу.

Під час роботи модуль Logger відправляє залоговані дані на документоорієнтовану базу даних, на якій вони можуть бути переглянуті та оброблені в будь який час.

Функцією моделей є описання даних якими маніпулює сервер та їх взаємодію між ними. Кожна з моделей представлених в системі має задачу репрезентації в системі користувачів, профілів, гроубоксів та статистики в вигляді абстракцій структур даних. Також моделі описують відношення між абстракціями та їх можливі взаємодії.

Головною функцією контексту реляційної бази даних є абстрагувати усі взаємодії з базою даних основними таким функціями є:

- створення нових кортежів;
- редагування існуючих кортежів;
- видалення існуючих кортежів;
- зчитування існуючих кортежів.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		24

Також додатковими функціями контексту реляційної бази даних є автоматичним створення нових таблиць та зміна їх в залежності до змін моделей, які використовуються в системі.

Запити після надходження на сервер оброблюються та логуються. Під час обробки запиту, перевіряється користувач на наявність прав на користування запрошеною функцією. Після перевірки прав користувача, запит починає оброблюватися. Під час обробки виконуються запити до реляційною бази даних в разі необхідності змін або зчитування даних.

4.3 Розроблення структури бази даних

На рисунку показано схему таблиць реляційної бази даних. На схемі можна побачити декілька сутностей які описують дані якими маніпулює сервер під час роботи, нижче опис полів сутностей системи.

У таблиці 4.1 описано поля сутності Profile.

Таблиця 4.1 – Поля сутності Profile

Назва	Тип	Опис
ID	INT	унікальний ідентифікатор профілю, головний ключ
owner_id	INT	унікальний ідентифікатор облікового запису користувача, зовнішній ключ
last_change_time	DATETIME	штамп часу під час якого було зроблено

		останню зміну в параметрах у профілі
binary_data	BYTEA	дані даного профілю в бінарному вигляді
growbox_received	BOOLEAN	поле вказує на те чи отримав гроубокс новий профіль, де true означає, що отримав, та false означає, що профіль потрібно переслати на гроубокс

Сутність User описує користувача в системі, головна роль цієї сутності - збереження особистих даних користувача. В таблиці 4.2 наведено деталі щодо полів сутності User.

Таблиця 4.2 – Поля сутності User

Поле	Тип	Опис
ID	INT	унікальний ідентифікатор облікового запису користувача, головний ключ
username	TEXT	унікальний ідентифікатор даний обліковому засобу користувачем для виконання процедури авторизації в системі
Поле	Тип	Опис

password	TEXT	секретна комбінація цифр, букв та спеціальних символів, які використовує користувач під час процедури авторизації в системі в вигляді тексту
registration_date	DATETIME	дата реєстрації облікового запису користувача
role_id	INT	унікальний ідентифікатор ролі, яку виконує користувач, зовнішній ключ
last_sign_in_date	DATETIME	дата останнього входження в обліковий запис користувачем

Сутність GrowBox тримає в собі детальну інформацію щодо стану гроубокса. В таблиці 4.3 наведено деталі щодо структури сутності.

Таблиця 4.3 – Поля сутності Growbox.

Поле	Тип	Опис
ID	INT	унікальний ідентифікатор даних гроубоксу, головний ключ

Поле	Тип	Опис
label	TEXT	ідентифікатор даний користувачем гроубоксу у вигляді тексту
last_stat_report_time	DATETIME	дата останнього звіту по статистиці гроубокса відправленого на сервер
owner_id	INT	унікальний ідентифікатор облікового запису користувача, що є власником гроубокса
active_profile_id	INT	унікальний ідентифікатор профілю параметрів внутрішнього середовища гроубокса
account_id	INT	унікальний ідентифікатор облікового запису гроубоксу, зовнішній ключ
first_report_time	DATETIME	дата першого звіту по статистиці гроубокса відправленого на сервер

Сутність Role тримає в собі інформації щодо ролей, що використовуються в системі. В таблиці 4.4 наведено деталі щодо сутності Role.

Таблиця 4.4 – Поля сутності Role

Поле	Тип	Опис
ID	INT	унікальний ідентифікатор ролі, створеної адміністратором, ГОЛОВНИЙ КЛЮЧ
name	TEXT	унікальний назва ролі задана адміністратором у вигляді тексту
description	TEXT	опис ролі заданий адміністратором
creation_date	DATETIME	дата створення ролі адміністратором системи

Сутність Message створена для зберігання повідомлень що надходять до серверу з гроубокса. В таблиці 4.5 наведено деталі щодо структури сутності в базі даних.

До сутності входять такі поля:

- ID;
- type_id;
- data;
- growbox_id;

Таблиця 4.5 – Поля сутності Message

Поле	Тип	Опис
ID	INT	унікальний ідентифікатор повідомлення отриманного гроубоксу 3
type_id	INT	унікальний ідентифікатор типу повідомлення, що прибуло
data	BYTEA	бінарні дані, що вказують додаткову інформацію до повідомлення
growbox_id	INT	унікальний ідентифікатор гроубоксу з якого прийшло повідомлення

Сутність MessageType створена для зберігання типів повідомлень. В таблиці 4.6 наведено деталі щодо структури сутності в базі даних.

Таблиця 4.6 – Поля сутності MessageType

Поле	Тип	Опис
ID	INT	унікальний ідентифікатор типу повідомлення, головний ключ

Поле	Тип	Опис
name	ТЕХТ	унікальна назва типу повідомлення у вигляді тексту
description	ТЕХТ	опис типу повідомлення заданий адміністратором

4.4 Розроблення архітектури Android-застосунка

На рисунку 4.3 зображено загальну схему Android застосунку.

Застосунок умовно можна розділити на три основних модулі:

- Models;
- ServerConnection;
- Activities;

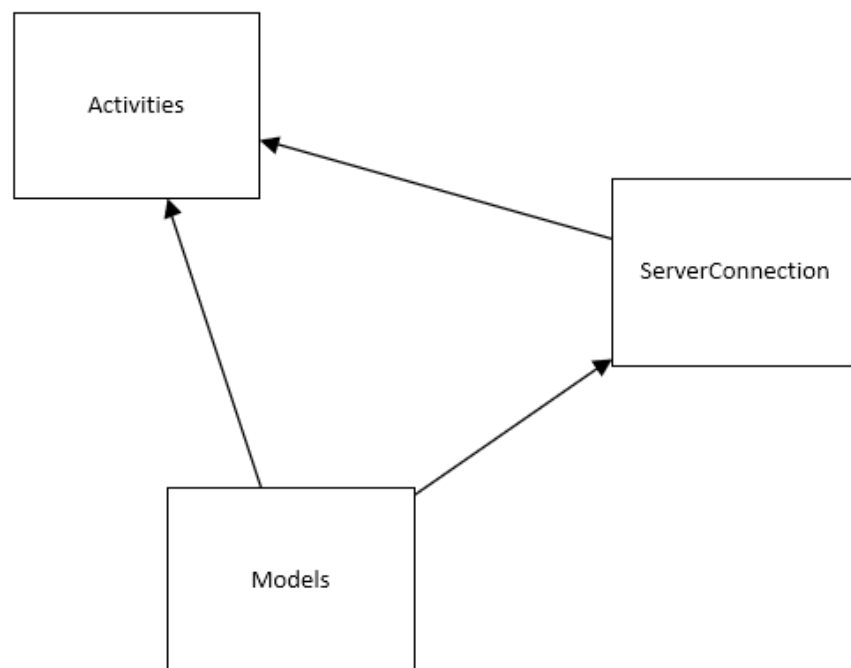


Рисунок 4.3 – Загальна структура Android застосунку

Перший модуль відповідає за об'єкти якими керує застосунок. Об'єкти представляють собою презентацію абстракцій які використовуються в системі, наприклад:

- профілі;
- гроубокси;
- обліковий запис.

Кожен з цих типів об'єктів використовується в модулі Activities для подальшого відображення в інтерфейсі застосунку з метою подальшої взаємодії користувача з ними.

Модуль ServerConnection приймає на себе роль керування зв'язком з'єднання застосунку з сервером. Модуль вбирає в себе такі функції:

- авторизація користувача на сервері;
- запити інформації про гроубокси;
- запити інформації про профілі;
- запити про зміну параметрів гроубоксу;
- запити про зміну параметрів профілю;
- запити про видалення гроубоксу;
- запити про видалення профілю;
- запити статистики по конкретному гроубоксу.

Модуль Activities інкапсулює в собі уся необхідну логіку для побудування інтерфейсу взаємодії користувача с застосунком. Таким чином модуль описує в собі усі параметри необхідні для розміщення елементів інтерфейсу на екрані користувача.

Модуль Activities відповідає за побудову таких екранів:

- відображення всіх гроубоксів;
- редагування гроубокса;
- додавання гроубокса;
- видалення гроубокса;
- відображення всіх профілів;

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

- редагування профілів;
- видалення профілів;
- перегляд статистики конкретного гроубокса;
- перегляд даних облікового запису.

Таким чином модуль відповідає за всі способи взаємодії з системою користувача завдяки системі екранів та елементів керування.

					ІТ61.230БАК.004 ПЗ	Арк.
						33
Зм.	Лист	№ докум.	Підпис	Дата		

5 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Реалізація сервера

Як було вказано в 3 частині пояснювальної записки, мовою розробки серверу була обрана мова С#, тож головною парадигмою програмування серверу була обрана об'єктно-орієнтована.

В цьому розділі буде описана реалізація серверу на основі ASP.NET Core. В системі реалізовано декілька класів для виконання задач поставлених системі.

Класи реалізовані в системі:

- ProfileController;
- AccountController;
- GrowBoxController;
- UserController;
- ApplicationContext;
- AuthOptions;
- Statistics;
- Sensor;
- Role;
- GrowBox;
- Account;
- Profile;
- Message;
- MessageType.

У таблиці 5.1 наведено детальний опис методів класу ProfileController.

В обов'язки об'єктів класу ProfileController входить обробка запитів користувачів, які надходять до серверу.

					IT61.230БАК.004 ПЗ	Арк.
						34
Зм.	Лист	№ докум.	Підпис	Дата		

Таблиця 5.1 – Методи класу ProfileController

Повертає	Назва	Аргументи	Опис
IEnumerable<Profile>	GetAll		Повертає колекцію даних типу Profile у форматі JSON
Profile	GetProfile	int id	Повертає профіль в залежності від даного в аргументах параметра унікального ідентифікатора в форматі JSON
IActionResult	AddProfile	Profile	Додає профіль надісланого на сервер у форматі JSON до колекції профілів користувача
IActionResult	DeleteProfile	int id	Видаляє профіль користувача з колекції профілів, унікальний ідентифікатор якого дорівнює надісланому

UserController відповідає за операції які використовує користувач. Взаємодія користувача з системою виконується за допомогою запитів до конкретних ендпоінтів. В таблиці 5.2 детально описано кожен функцію яку може використати користувач.

Таблиця 5.2 – Методи класу UserController

Повертає	Назва	Аргумент	Опис
IEnumerable<GrowBox>	GetAllGrowBoxes		Повертає дані гроубоксів які належать користувачу, який проводить запит до серверу
GrowBox	GetGrowBox	int id	Повертає дані конкретного гроубоксу, який належить користувачу, залежності від даного унікального ідентифікатора
IActionResult	AddGrowBox	GrowBox	Виконує додавання гроубоксу до колекції користувача, повертає код результату операції.

Повертає	Назва	Аргумент	Опис
IActionResult	EditGrowBox	GrowBox	Виконує редагування гроубоксу з ідентифікатором отриманого гроубоксу, повертає код результату операції
IActionResult	DeleteGrowBox	int id	Виконує видалення гроубоксу з колекції в залежності від отриманого в аргументах унікального ідентифікатору
Statistics	GetGrowBoxStatistics	int id	Повертає статистику змін параметрів по конкретному гроубоксу, ідентифікатор якого отриманий в аргументах
IEnumerable<Message>	GetGrowBoxMessages	int id	Повертає усі повідомлення з конкретного

Зм.	Лист	№ докум.	Підпис	Дата

			гроубокса, що належать цьому користувачеві
IEnumerable<MessageTypes>	GetMessageTypes		Повертає усі можливі типи повідомлення

Методи класу `GrowBoxController`, наведені у таблиці 5.3, відповідають за взаємодію гроубоксу з сервером. Гроубокс взаємодіє з сервером шляхом відправлення запиту на потрібну йому кінцеву точку.

Таблиця 5.3 – Методи класу `GrowBoxController`

Повертає	Назва	Аргумент	Опис
Profile	GetActiveProfile		Повертає гроубоксу, що запитує, його активний профіль
ActionResult	SendStatistics	Statistics stat, GrowBox gb	Записує в базу даних статистику отриману від гроубоксу в JSON форматі

Методи класу `AccountController`, наведені у таблиці 5.4, відповідають за аутентифікацію користувача в системі. Це відбувається шляхом запиту на кінцеву точку, при цьому запит повинен мати в собі ім'я та пароль користувача.

Таблиця 5.4 – Методи класу AccountController

Повертає	Назва	Аргументи	Опис
IActionResult	Token	string, string	повертає JSON WebToken в разі знаходження совпадіння логін/пароль в базі даних
IActionResult	Register	string,string	реєструє нового користувача перевіряючи на унікальність логін, та в випадку успіху повертає JSON WebToken
ClaimsIdentity	GetIdentity	string,string	повертає ClaimsIdentity користувача за його логіном та паролем

ApplicationContext представляє собою посередника між сервером та реляційною базою даних. У таблиці 5.5 детально наведено, які сутності збирає ApplicationContext з бази даних.

Таблиця 5.5 – Поля класу ApplicationContext

Назва	Тип	Опис
Accounts	DbSet<Account>	Колекція об'єктів отриманих з бази даних

Назва	Тип	Опис
GrowBoxes	DbSet<GrowBox>	Колекція об'єктів класу GrowBox отриманих з бази даних
Roles	DbSet<Roles>	Колекція об'єктів класу Roles отриманих з бази даних
Profiles	DbSet<Profile>	Колекція об'єктів класу Profile отриманих з бази даних
Messages	DbSet<Messages>	Колекція об'єктів класу Messages отриманих з бази даних
MessageTypes	DbSet<MessageType>	Колекція об'єктів класу MessageType отриманих з бази даних

ApplicationContext має спеціальний метод для опису спеціальної стрічки з'єднання з базою даних. Детальний опис наведено в таблиці 5.6.

Таблиця 5.6 – Методи класу ApplicationContext

Повертає	Назва	Аргументи	Опис
void	OnConfiguring	DbContextOptionsBuilder	Функція, яка викликається під час конфігурації ApplicationContext та задає стрічку з'єднання з базою даних.

Клас Role відповідає за презентацію ролей в системі, об'єкти цього класу мають в собі детальну інформацію щодо ролі, опис цієї структури наведено в таблиці 5.7.

Об'єкти цього класу мають інформацію про ім'я користувача та опис.

Таблиця 5.7 – Поля класу Role

Назва	Тип	Опис
ID	int	Унікальний ідентифікатор об'єкта
name	string	Дана адміном назва ролі у вигляді тексту
description	string	Повний опис ролі даний адміном для ролі
creation_date	DateTime	Дата створення ролі

Об'єкти класу GrowBox мають в собі детальну інформацію щодо конкретних гроубоксів користувачів. Детальну інформацію про поля класу GrowBox наведено в таблиці 5.8.

Таблиця 5.8 – Поля класу GrowBox

Назва	Тип	Опис
ID	int	унікальний ідентифікатор об'єкта, який використовуються базою даних
label	string	Назва дана користувачем гроубоксом

Назва	Тип	Опис
owner_id	int	Унікальний ідентифікатор облікового запису користувача якому належить гроубокс
active_profile_id	int	Унікальний ідентифікатор профілю налаштуванням якого слідує гроубокс
account_id	int	Унікальний ідентифікатор облікового запису який використовує гроубокс для взаємодії з сервером
last_stat_report_time	DateTime	Дата останнього звіту гроубоксів щодо стану всередині гроубоксу
first_report_time	DateTime	Дата першого звіту гроубоксу щодо стану всередині гроубоксу

Об'єкти класу Account мають в собі інформацію щодо облікових записів користувачів. Детальну інформацію щодо структури класу Account наведено в таблиці 5.9.

Об'єкти класу Account мають в собі інформації щодо унікального ідентифікатора, імені користувача, пароля, ролі та дати створення.

Таблиця 5.9 – Поля класу Account

Назва	Тип	Опис
ID	int	Унікальний ідентифікатор об'єкта, використовується базою даних
username	string	Унікальний ідентифікатор користувача у вигляді тексту, використовується для логіну в системі
password	string	Секретна фраза користувача, яка використовується для логіну в системі.
role_id	int	Унікальний ідентифікатор ролі, яка використовується в системі
registration_date	DateTime	Дата реєстрації облікового запису користувача
last_sign_in_date	DateTime	Дата останнього входу в обліковий запис користувача

Об'єкти класу Profile мають в собі інформацію щодо профілей, які використовуються в грубоксах для контролю за параметрами внутрішнього

середовища гроубокса. Детальну інформацію щодо структури класу Profile наведено в таблиці 5.10.

Таблиця 5.10 – Поля класу Profile

Назва	Тип	Опис
ID	int	Унікальний ідентифікатор об'єкту, використовується в базі даних
owner_id	int	Унікальний ідентифікатор облікового запису користувача, що створив цей профіль
label	string	Умовне позначення користувачем цього профілю у вигляді тексту
last_change_time	DateTime	Дата останньої зміни зробленої користувачем в параметри профілю
creation_date	DateTime	Дата створення профілю користувачем
binary_data	byte[]	дані профілю записані в бінарному вигляді

Клас AuthOptions має в собі інформацію щодо глобальних налаштувань аутентифікації в системі, на кшталт часу дії токена авторизації. В таблиці 5.11 наведено детальну інформацію щодо полів класу AuthOptions.

Таблиця 5.11 – Поля класу AuthOptions

Назва	Тип	Опис
KEY	string	Секретна комбінація даних на основі якої будуються токени симетричні ключі користувачів
LIFETIME	int	Час який буде дійсним токен авторизації користувача

Для генерації секретних ключів, які використовуються при аутентифікації, використовується метод класу AuthOptions. Детальний опис цього методу наведений в таблиці 5.12.

Таблиця 5.12 – Методи класу AuthOptions

Повертає	Назва	Аргументи	Опис
string	GetSymmetricSecurityKey		Повертає унікальний для кожного користувача симетричний ключ

Об'єкти класу Message представляють в системі історію повідомлень, якими обмінюється сервер та гроубокс. Більш детально структуру класу GrowBox наведено в таблиці 5.13.

Таблиця 5.13 – поля класу Message

Назва	Тип	Опис
ID	int	Унікальний ідентифікатор отриманого з гроубокса повідомлення
data	byte[]	Дані, які йдуть в доповненні до повідомлення з гроубокса
type_id	int	Унікальний ідентифікатор типу повідомлення, що надійшло з гроубокса
message_received_date	DateTime	Дата надходження гроубокса на сервер з гроубокса
growbox_id	int	Унікальний ідентифікатор гроубокса з якого надійшло повідомлення

Тип повідомлень виділено в окрему абстракцію MessageType, об'єкти цього класу використовуються для визначення типу повідомлення, що надходять на сервер. Більш детально наведено структуру класу MessageType в таблиці 5.14.

Таблиця 5.14 – Поля класу MessageType

Назва	Тип	Опис
ID	int	Унікальний ідентифікатор типу повідомлення, що надходять з гроубокса
name	string	Унікальний ідентифікатор повідомлення у вигляді тексту
description	string	Опис типу повідомлення отриманого з гроубокса

5.2 Реалізації бази даних

Невід’ємною частиною системи є база даних, саме завдяки ній вдається організувати зручне та ефективне сховище даних, яке забезпечує безвідмовний та швидкий доступ до даних користувачів.

База даних будується за допомогою спеціального SQL коду, який виконуючись базою даних створює в середині неї спеціальні сутності, що є репрезентацією даних, які будуть там знаходитися.

Таким чином створити таблиці в середині бази даних можна створити за допомогою ключових слів «CREATE TABLE» та додаткового синтаксису, які визначають які таблиці будуть стовпці та додаткові властивості для них, цими додатковими властивостями можуть бути наприклад:

— унікальність – значення в кожному з кортежів в цьому стовпці повинно бути унікальним або в протилежному випадку база даних не дозволить додати новий кортеж з дублікатом, така властивість даних завжди використовується для

головного ключа сутності, для забезпечення унікальності кожного кортежа. Також унікальність даних дозволяє пришвидшити пошук даних по цьому параметру завдяки розумним варіаціям хеш-таблиць.

— автогенерація – значення в кожному з кортежів в цьому стовпці будується за допомогою автогенерації на стороні бази даних. Це може виконуватися в PostgreSQL за допомогою спеціальних сутностей реалізованих в базі даних за назвою `sequence`(послідовність), в найпростішому випадку послідовність виглядає як деяке число, яке збільшується при кожному виклику базою даних послідовності. Автогенерація найчастіше використовується для генерування унікального ідентифікатора в базі даних. Також автогенерація може бути використана наприклад для автогенерування хешів даних.

— NOT NULL (не нуль) – значення в кожному з кортежів в цьому стовпці має мати значення, тобто забороняється ситуація при якій стовпцю не надається значення. Це зазвичай використовується для критично важних даних для функціонування системи, наприклад: паролів, унікальних ідентифікаторів, логінів та інших даних, потреба в яких може з'являтися в залежності від ситуації.

Маніпулювання даними виконується за допомогою команд:

— SELECT – використовується для отримання даних з бази даних з конкретними умовами, які задаються за допомогою ключового слова `WHERE` та об'єднання таблиць за допомогою ключового слова `JOIN`;

— INSERT – використовується для внесення нових даних в таблиці;

— DELETE – використовується для видалення конкретного кортежа таблиці або масиву кортежів за допомогою ключового слова `WHERE`;

— UPDATE – використовується для зміни даних в вже існуючих кортежах по одному або в масиві кортежів за допомогою ключового слова `WHERE`.

Усю складну логіку з цими командами інкапсулює в собі Entity Framework.

В проекті використовується підхід до створення бази даних, який називається Code First. Цей підхід, реалізований в Entity Framework, дозволяє спочатку написати код моделей які будуть використовуватися в базі даних, а далі фреймворк

					IT61.230БАК.004 ПЗ	Арк.
						48
Зм.	Лист	№ докум.	Підпис	Дата		

сам сформує усі необхідні запити в базу даних на створення нових таблиць та налаштування їх стовпців з особливими модифікаторами такими як автогенерування, зовнішні ключі, головний ключ та інші.

5.3 Реалізація протоколу зв'язку між гроубоксом та сервером

Оскільки гроубокси обладнанні слабкими мікроконтролерами, спеціально для цього було розроблено спеціальний формат повідомлень яким будуть користуватися гроубокс та сервер для обміну повідомленнями.

Обмін повідомленнями можна умовно поділити на два етапи:

1. запит зі сторони гроубокса;
2. відповідь зі сторони серверу.

Під час етапу 1 гроубокс надсилає запит на сервер, який має структуру наведену в таблиці 5.1.

Таблиця 5.15 – Структура запиту з гроубокса до сервера

Призначення	Розмір поля у бітах	Тип
Унікальний ідентифікатор типу повідомлення	32	ціле число
Додаткові дані повідомлення	64	залежить від типу повідомлення
Унікальний ідентифікатор активного налаштування	32	ціле число

Призначення	Розмір поля у бітах	Тип
Кількість часових проміжків для яких надана статистика	16	ціле число
Розмір одного проміжка даних у байтах	16	ціле число
Унікальний ідентифікатор типу датчика	16	ціле число
Дані датчику	64	залежить від типу датчика

Формат виглядає складним а отже треба навести приклад.

Нехай у нас наступні вихідні дані:

- Тип повідомлення «звіт»;
- Додаткова інформація до повідомлення «лампа вийшла з ладу»;
- Унікальний ідентифікатор активного налаштування 42;
- Кількість часових проміжків 1;
- Кількість даних в одному проміжку 140 (14 датчиків).

Тож дані запиту будуть виглядати чином наведеним в таблиці 5.2.

Таблиця 5.16 – Приклад повідомлення з гроубоксу до серверу

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Унікальний ідентифікатор типу повідомлення	Унікальний ідентифікатор типу повідомлення «звіт»	32	1

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Додаткові дані повідомлення	Унікальний ідентифікатор помилки «лампа вийшла з ладу»	64	1
Унікальний ідентифікатор активного налаштування	Унікальний ідентифікатор активного налаштування	32	42
Кількість часових проміжків для яких надана статистика		16	1
Розмір одного проміжка даних у байтах		16	140
Показання датчиків	Унікальний ідентифікатор датчику внутрішньої вологості	16	1
Показання датчиків	Кількісне значення вологості у відсотках	64	56
Показання датчиків	Унікальний ідентифікатор датчику	16	2

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Показання датчиків	Кількісне значення вологості у відсотках	64	46
Показання датчиків	Унікальний ідентифікатор датчику вологості ґрунту	16	3
Показання датчиків	Кількісне значення вологості у відсотках	64	
Показання датчиків	Унікальний ідентифікатор датчику зовнішньої температури	16	4
Показання датчиків	Кількісне значення температури у градусах Цельсію	64	
Показання датчиків	Унікальний ідентифікатор датчику вимірювання внутрішньої температури	16	5

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Показання датчиків	Кількісне значення температури у градусах Цельсію	64	
Показання датчиків	Унікальний ідентифікатор дати останнього поливу	16	6
Показання датчиків	Час Unix	64	1591621299
Показання датчиків	Ідентифікатор дати останнього удобрювання	16	7
Показання датчиків	Час Unix	64	1591621299
Показання датчиків	Ідентифікатор потужності роботи вентилятора	16	8
Показання датчиків	Кількісне значення оборотів вентилятора в секунду	64	j
Показання датчиків	Ідентифікатор потужності роботи елемента Пельтьє	16	9

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Показання датчиків	Кількісне значення потужності	64	10
Показання датчиків	Унікальний ідентифікатор стану елемента Пельтьє	16	1
Показання датчиків	Охолоджує чи нагріває внутрішнє середовище	64	1
Показання датчиків	Унікальний ідентифікатор датчику наявності води в баку	16	11
Показання датчиків	Булеве значення, вода наявна чи ні	64	1
Показання датчиків	Унікальний ідентифікатор датчику наявності розчину в баку	16	12
Показання датчиків	Булеве значення, розчин наявний чи ні	64	1
Показання датчиків	Ідентифікатор датчику стану освітлення	16	13

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Показання датчиків	Ідентифікатор стану освітленості гроубокса	64	2
Показання датчиків	Ідентифікатор дати та часу взяття даних з датчиків	16	14
Показання датчиків	Час Unix	64	1591621299

Таким чином розмір повідомлення в цьому форматі буде 160 байтів, що набагато нижче аналогічного повідомлення в форматі JSON. Тепер, коли на сервер надійшло повідомлення з гроубоксу у відповідь сервер відправляє повідомлення в якому описано які дії потрібно виконати та новий профіль, якщо такий з'явився.

У таблиці 5.3 показана структура початку повідомлення, в якому в типі повідомлення вказується інформація щодо дій які потрібно виконати на гроубоксі. Наприклад поставити новий профіль.

Таблиця 5.17 – Заголовок відповіді гроубоксу з сервера.

Призначення	Розмір поля у бітах	Тип
Унікальний ідентифікатор типу повідомлення	32	ціле число
Додаткові дані отриманого повідомлення	64	залежить від тип повідомлення

Призначення	Розмір поля у бітах	Тип
Унікальний ідентифікатор наданого налаштування	32	ціле число

У таблиці 5.18 наведено-формат, у якому передається профіль до гроубоксу.

Таблиця 5.18 – Формат запису профілю у повідомленні

Призначення	Розмір поля у бітах	Тип
Кількість різних параметрів у профілі	32	ціле число
Розмір опису наступного параметра в профілі	16	ціле число
Унікальний ідентифікатор типу параметру	16	ціле число
День початку дії конфігурації параметру k	16	ціле число
Значення конфігурації параметру k	64	ціле число
...
День початку дії конфігурації параметру k + N	16	ціле число
Значення конфігурації параметру k + N	64	ціле число
...		

Призначення	Розмір поля у бітах	Тип
Унікальний ідентифікатор типу параметру k_n	16	ціле число
День початку дії конфігурації параметру k_n	16	ціле число
Значення конфігурації параметру k_n	64	ціле число
...
День початку дії конфігурації параметру $k_n + N$	16	ціле число
Значення конфігурації параметру $k_n + N$	64	ціле число

Таким чином в повідомленні можна вказати велику кількість параметрів профілю (більше 1000). Усі параметри складаються з проміжків часу які активуються на потрібний день.

Для того що більш докладно показати як будуть виглядати повідомлення, в таблиці 5.19 наведено приклад повідомлення.

Таблиця 5.19 – Приклад відповіді сервера гроубоксу

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Унікальний ідентифікатор типу повідомлення	Повідомлення типу «новий профіль»	32	2

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Додаткові дані повідомлення	Додаткових даних з повідомленням не передається	64	0
Унікальний ідентифікатор наданого налаштування	Ідентифікатор нового налаштування	32	2
Кількість різних параметрів у профілі	У повідомленні будуть 5 різних параметрів	32	5
Розмір опису наступного параметра в профілі у байтах	Розмір опису наступного параметра вологості	16	42
Унікальний ідентифікатор типу параметру	Унікальний ідентифікатор типу параметру вологості	16	1
День початку дії конфігурації параметру вологості	Починає діяти з самого початку	16	0
Значення бажаного в системі параметру вологості	Значення вологості у відсотках	64	56

Назва поля	Опис даних	Розмір даних	Значення записане в даних
День початку дії конфігурації параметру вологості	Починає діяти з дня	16	2
Значення конфігурації параметру вологості	Значення вологості у відсотках	64	80
День початку дії конфігурації параметру вологості	Починає діяти з дня	16	4
Значення конфігурації параметру вологості	Значення вологості у відсотках	64	40
День початку дії конфігурації параметру вологості	Починає діяти з дня	16	6
Значення конфігурації параметру вологості	Значення вологості у відсотках	64	90
Розмір опису параметра в профілі у байтах	Розмір опису параметра температури	16	52

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Унікальний ідентифікатор типу параметру	Унікальний ідентифікатор типу параметру вологості	16	2
День початку дії конфігурації параметру температури	Починає діяти з самого початку	16	0
Значення конфігурації параметру температури	Значення температури градусах Цельсію	64	24
День початку дії конфігурації параметру температури	Починає діяти з дня	16	2
Значення конфігурації параметру температури	Значення температури градусах Цельсію	64	30
День початку дії конфігурації параметру температури	Починає діяти з дня	16	5
Значення параметру температури	Значення температури градусах Цельсію	64	15

Назва поля	Опис даних	Розмір даних	Значення записане в даних
День початку дії конфігурації параметру температури	Починає діяти з дня	16	6
Значення конфігурації параметру температури	Значення температури градусах Цельсію	64	24
День початку дії конфігурації параметру температури	Починає діяти з дня	16	8
Значення конфігурації параметру температури	Значення температури градусах Цельсію	64	32
Розмір опису наступного параметра в профілі у байтах	Розмір опису наступного параметра вологості ґрунту	16	42
Унікальний ідентифікатор типу параметру	Унікальний ідентифікатор типу параметру вологості ґрунту	16	3
День початку дії параметру вологості ґрунту	Починає діяти з самого початку	16	0

Назва поля	Опис даних	Розмір даних	Значення записане в даних
Значення конфігурації параметру вологості ґрунту	Значення вологості ґрунту у відсотках	64	13
День початку дії конфігурації параметру вологості ґрунту	Починає діяти з самого початку	16	2
Значення конфігурації параметру вологості ґрунту	Значення вологості ґрунту у відсотках	64	30
День початку дії конфігурації параметру вологості ґрунту	Починає діяти з самого початку	16	4
Значення конфігурації параметру вологості ґрунту	Значення вологості ґрунту у відсотках	64	25
День початку дії конфігурації параметру вологості ґрунту	Починає діяти з самого початку	16	5
Значення параметру вологості ґрунту	Значення вологості ґрунту у відсотках	64	29

ВИСНОВКИ

У результаті роботи над дипломним проектом було досягнуто поставленої мети: спрощено процес вирощування рослин, чутливих до абіотичних факторів, у домашніх умовах.

Для досягнення цієї мети було вирішено ряд задач:

- виконано огляд існуючих рішень;
- виконано вибір компонентів;
- розроблено формат повідомлень;
- розроблено архітектуру сервера;
- реалізовано сервер;
- розроблено архітектуру Android-застосунку;
- реалізовано Android-застосунок.

Розроблена система відповідає поставленим вимогам:

- можливість маніпуляції даними користувача через застосунок;
- зміна налаштувань гроубокса через застосунок;
- зміна профілів гроубоксів через застосунок;
- перегляд статистики гроубоксів через застосунок.

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		63

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Всі основні види теплиць і їх конструкції характеристика за типами та креслення. URL: <https://valest.com.ua/vsi-osnovni-vidi-teplic-i-ih-konstrukcii/>
2. ПРОМЫШЛЕННЫЕ ТЕПЛИЦЫ. URL: <https://teplitca.kiev.ua/p907706-promyshlennye-teplitsy.html>
3. ТЕПЛИЦА "СУПЕР ПАРНИК". URL: <https://teplitca.kiev.ua/p1134124689-teplitsa-super-parnik.html>
4. Grow box – Wikipedia. URL: https://en.wikipedia.org/wiki/Grow_box
5. Мини гроубокс с комплектом оборудования Джин. URL: <https://fito.store/komplekty-grouboksov-dlya-grovinga/groubok-s-oborudovaniem-dzhin-600-600-1400>
6. Гроубокс Hydro Shoot 100. URL: http://growing.com.ua/growbox/secret_jardin_category/hydro_shoot/hydro_shoot_100.html
7. Климатическая камера Фитотрон ЛиА-2. URL: <https://bobrujsk.deal.by/p105685532-klimaticheskaya-kamera-fitotron.html>
8. ASP.NET Core – Wikipedia. URL: https://uk.wikipedia.org/wiki/ASP.NET_Core
9. ADO.NET Entity Framework – Wikipedia. URL: https://ru.wikipedia.org/wiki/ADO.NET_Entity_Framework
10. PostgreSQL – Wikipedia. URL: <https://uk.wikipedia.org/wiki/PostgreSQL>
11. Elasticsearch – Wikipedia. URL: <https://uk.wikipedia.org/wiki/Elasticsearch>
12. Elasticsearch.Net. URL: <https://www.elastic.co/guide/en/elasticsearch/client/net-api/7.x/index.html>
13. Android Studio – Wikipedia. URL: https://uk.wikipedia.org/wiki/Android_Studio
14. React Native – Wikipedia. URL: https://en.wikipedia.org/wiki/React_Native

15. Flask – Wikipedia. URL:

[https://en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework))

16. MySQL – Wikipedia. URL: <https://uk.wikipedia.org/wiki/MySQL>

17. MONITOR&CONTROLLER. URL: <https://ceresgs.com/climate-control/monitor-and-controller/>

18. Smart Greenhouse App. URL: <https://dribbble.com/shots/7710718-Smart-greenhouse-app>

					IT61.230БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		65