

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

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК 004.8

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«___» _____ 2025 р.

Магістерська дисертація

на здобуття ступеня магістра

**за освітньо-професійною програмою
«Інтегровані інформаційні системи»**

зі спеціальності 126 «Інформаційні системи та технології»

**на тему: «Безпілотна система для автоматизованого відбору проб
води»**

Виконав:
студент 2 курсу, групи ІА-41мп
Варварик Микола Костянтинович _____

Керівник:
доцент каф. ІСТ, к.т.н., доц.
Писаренко Андрій Володимирович _____

Рецензент:
доцент каф. ІСТ, к.т.н., доц.
Лісовиченко Олег Іванович _____

Засвідчую, що у цій магістерській
дисертації немає запозичень з праць
інших авторів без відповідних
посилань.
Студент _____

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 — другий (магістерський)

Спеціальність — 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Варварику Миколі Костянтиновичу

1. Тема дисертації «Безпілотна система для автоматизованого відбору проб води», науковий керівник дисертації Писаренко Андрій Володимирович, к.т.н., доц., затвержені наказом по університету від «06» 11 2025 р. № 4841-с
2. Термін подання студентом дисертації «15» 12 2025 р.
3. Об'єкт дослідження: процес автоматизованого відбору проб води з використанням безпілотної платформи.
4. Вихідні дані: точність позиціонування до 1 метра, час безперервної роботи до 40 хвилин, додаткова вага відібраної проби до 50 грам, підтримка одночасного виконання більше 1 місії.
5. Перелік завдань, які потрібно розробити: опис предметної області, проектування системи, реалізація і тестування підсистеми управління оператора, розроблення стартап-проєкту.
6. Орієнтовний перелік графічного (ілюстративного) матеріалу: структурна схема, функціональна схема, блок-схема алгоритму функціонування, блок-схема алгоритму керування, схема компонентів підсистеми управління оператора, діаграма послідовностей підсистеми, схема бази даних, діаграма розгортання підсистеми.

7. Орієнтовний перелік публікацій: не заплановано.

8. Дата видачі завдання 01.09.2025 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Огляд та аналіз існуючих рішень за тематикою завдання магістерської дисертації	до 12.09.25	
2.	Вибір платформи для проєкту	до 12.09.25	
3.	Визначення функціональних, технічних вимог, методів та засобів для вирішення завдання	до 26.09.25	
4.	Проектування стурктурної та функціональної схем системи	до 10.10.25	
5.	Вибір компонентів та програмного забезпечення для керування безпіотною системою	до 31.10.25	
6.	Розроблення системи менеджменту оператора	до 21.11.25	
7.	Побудова UML-діаграм	до 28.11.25	
8.	Розроблення стартап-проєкту	до 05.12.25	
9.	Оформлення основної частини	до 05.12.25	

Студент

Микола ВАРВАРИК

Науковий керівник

Андрій ПИСАРЕНКО

РЕФЕРАТ

Безпілотна система для автоматизованого відбору проб води: 109 с., 29 табл., 35 рис., 9 дод., 31 джерел.

APACHE KAFKA, JAKARTA EE, POSTGRESQL, SPE-КАРТРИДЖ, АВТОМАТИЗАЦІЯ, БПЛА, ЕКОЛОГІЧНИЙ МОНІТОРИНГ, ТЕЛЕМЕТРІЯ, ЯКІСТЬ ВОДИ.

Сучасний стан екологічної безпеки вимагає оперативного та точного контролю якості водних ресурсів. Традиційні методи ручного відбору проб є часозатратними, економічно неефективними та часто пов'язані з ризиками для життя персоналу, особливо у важкодоступних або забруднених зонах. Існуючі комерційні рішення мають високу вартість та обмежену мобільність. Розроблення безпілотної системи дозволяє вирішити проблему оперативного моніторингу, забезпечуючи швидке транспортування даних та зразків.

Метою роботи є підвищення ефективності та безпеки екологічного моніторингу якості води шляхом створення автоматизованого програмно-апаратного комплексу на базі БПЛА для відбору проб води.

Об'єктом дослідження є процес автоматизованого відбору проб води з використанням безпілотної платформи.

Предметом дослідження виступають методи та засоби побудови системи автоматичного відбору проб води.

У роботі використані методи системного аналізу для проєктування архітектури комплексу; об'єктно-орієнтованого проєктування та патерни розроблення для реалізації програмної частини; методи маркетингового аналізу для оцінки комерційного потенціалу.

Запропонована архітектура інформаційної системи на базі монолітного додатку Jakarta EE з використанням брокера повідомлень Apache Kafka, що забезпечує доставку телеметрії в умовах нестабільного мобільного зв'язку.

Створено веб-інтерфейс оператора для планування місій, моніторингу в реальному часі та формування звітів.

ABSTRACT

Unmanned system for automated water sample selection: 109 p., 29 tab., 35 draw., 9 app., 31 sources.

APACHE KAFKA, AUTOMATION, ECOLOGICAL MONITORING, JAKARTA EE, POSTGRESQL, SPE CARTRIDGE, TELEMETRY, UAV, WATER QUALITY.

The current state of environmental safety demands rapid and accurate monitoring of water resource quality. Traditional manual sampling methods are time-consuming, cost-ineffective, and often involve life-threatening risks for personnel, particularly in hard-to-reach or contaminated areas. Existing commercial solutions are characterized by high costs and limited mobility. The development of an unmanned system addresses the problem of rapid monitoring by ensuring the fast transportation of data and samples.

The objective of the work is to enhance the efficiency and safety of environmental water quality monitoring by creating an automated software-hardware complex based on a UAV for water sampling.

The object of the research is the process of automated water sampling using an unmanned platform.

The subject of the research comprises the methods and tools for constructing an automated water sampling system and its management by operators.

The methods are the work utilizes systems analysis methods for designing the complex's architecture; object-oriented design and development patterns for implementing the software component; and marketing analysis methods for assessing commercial potential.

In result is an information system architecture based on a monolithic Jakarta EE application is proposed, utilizing the Apache Kafka message broker to ensure telemetry delivery under unstable mobile connection conditions. An operator web interface has been created for mission planning, real-time monitoring, and report generation.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ	8
ВСТУП.....	10
1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.1 Аналіз комерційних безпілотних систем для моніторингу водойм	12
1.2 Переваги та недоліки існуючих технологій.....	22
1.3 Вибір платформи для проєкту	23
Висновки до розділу 1	24
2 ПРОЄКТУВАННЯ СИСТЕМИ	26
2.1 Розроблення структурної схеми	26
2.2 Розроблення функціональної схеми	27
2.3 Вибір окремих елементів системи.....	30
2.4 Проєктування програмної архітектури	35
2.4.1 Архітектура бортового програмного забезпечення.....	35
2.4.2 Архітектура серверної частини	36
2.4.3 Архітектура підсистеми управління оператора.....	37
2.5 Алгоритм функціонування системи	38
2.5.1 Алгоритм керування БПЛА	38
Висновки до розділу 2	39
3 РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ ПІДСИСТЕМИ УПРАВЛІННЯ ОПЕРАТОРА.....	41
3.1 Вибір технологічного стеку	41
3.2 Реалізація модулів	47
3.3 Тестування функціоналу підсистеми	68
Висновки до розділу 3	74
4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ	77
4.1 Опис ідеї проєкту	77
4.2 Технологічний аудит ідеї проєкту.....	79
4.3 Аналіз ринкових можливостей запуску стартап-проєкту	80

4.4 Розроблення ринкової стратегії проекту.....	87
4.5 Розроблення маркетингової програми стартап-проекту	90
Висновки до розділу 4	93
ВИСНОВКИ	94
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	98
ДОДАТОК А	101
ДОДАТОК Б	102
ДОДАТОК В	103
ДОДАТОК Г	104
ДОДАТОК Д	105
ДОДАТОК Е	106
ДОДАТОК Ж	107
ДОДАТОК И	108
ДОДАТОК К	109

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

БД — база даних

СУБД — система управління базою даних

API — Application Programming Interface — певний контракт, що визначає, які з застосунком можуть взаємодіяти інші

DAO — Data Access Object — абстрактний інтерфейс до бази даних або іншого сховища

NPM — Node Package Manager — менеджер пакетів для NodeJS

REST — Representational State Transfer — стиль написання веб-застосунків, описує як користувачу слід організувати написання коду веб-застосунку

SQL — Structured Query Language — мова написання запитів, за допомогою якої можна взаємодіяти із базами даних

БПЛА — Безпілотний літальний апарат

UAV — Unmanned Aerial Vehicle, безпілотний літальний апарат

ASV — Autonomous Surface Vehicles

АПС — Автономні поверхневі судна

AUV — Autonomous Underwater Vehicles

АПА — Автономні підводні апарати

ROV — Remotely Operated Vehicle, дистанційно керований підводний апарат

ДПА — Дистанційно керований підводний апарат

USV — Unmanned Surface Vehicle, автономний наземний чи надводний транспортний засіб без екіпажу

БНА — Безпілотний надводний апарат

SPE — Solid Phase Extraction, твердофазна екстракція

pH — водневий показник, показник кислотності та лужності води

DO — Dissolved Oxygen, розчинений кисень

ЕС — Electrical Conductivity, електропровідність

IMU — Inertial Measurement Unit, інерційний вимірювальний блок

GNSS — Global Navigation Satellite System, глобальна навігаційна супутникова система

GPS — Global Positioning System, глобальна система позиціонування

RTK — Real-Time Kinematic, кінематика реального часу — технологія точного позиціонування

JDBC — Java Database Connectivity, стандарт взаємодії Jakarta з базами даних

JPA — Java Persistence API, специфікація для збереження об'єктів у базі даних

EJB — Enterprise JavaBeans, компоненти бізнес-логіки

JAX-RS — Java API for RESTful Web Services

JSP — JavaServer Pages, технологія для створення веб-сторінок

OPB — Об'єктно-реляційне відображення

SSE — Server-Sent Events, технологія відправки подій від сервера до клієнта

ACID — Atomicity, Consistency, Isolation, Durability, властивості транзакцій: атомарність, узгодженість, ізольованість, довговічність

JSON — JavaScript Object Notation, текстовий формат обміну даними

ВСТУП

Сучасний стан екологічної безпеки та управління водними ресурсами вимагає впровадження нових, більш ефективних підходів до контролю якості поверхневих вод. Гідросфера зазнає постійного антропогенного впливу, що призводить до забруднення водойм промисловими скидами, агрохімікатами та побутовими відходами. Своєчасне виявлення таких забруднень є необхідним для збереження екосистем та здоров'я населення.

Традиційні методи моніторингу, які базуються на ручному відборі проб із використанням човнів або заборі води з берега, на сьогодні є застарілими. Вони характеризуються високою часозатратністю, економічною неефективністю та, що найважливіше, часто пов'язані з ризиками для життя та здоров'я персоналу, особливо при роботі у важкодоступних, заболочених або токсично забруднених зонах. Окрім того, ручний відбір не дозволяє забезпечити необхідну оперативність отримання даних та їх просторову дискретизацію для побудови точних карт забруднення.

На ринку існують автоматизовані рішення, такі як автономні поверхневі судна та автономні підводні апарати. Проте, аналіз предметної області показує, що вони мають високу вартість, складні в обслуговуванні та мають обмежену мобільність (потребують спеціалізованих місць для спуску на воду). У той же час безпілотні літальні апарати (БПЛА) демонструють високу мобільність і здатність швидко дістатися будь-якої точки водойми, проте їх використання для гідрохімічного аналізу стримується обмеженою вантажопідйомністю та відсутністю спеціалізованих легких систем відбору.

Метою роботи є підвищення ефективності та безпеки екологічного моніторингу якості води шляхом створення автоматизованого програмно-апаратного комплексу на базі БПЛА для відбору проб води.

Для досягнення поставленої мети поставлені і вирішені наступні задачі:

— аналіз існуючих комерційних рішень та методів для автоматизованого відбору проб води;

- обрання апаратної платформи та компонентів для проєкту, визначення функціональних та технічних вимог до системи;

- проєктування структурної та функціональної схеми системи, розроблення алгоритмів керування та функціонування.

- удосконалення методів відбору проб для зниження ваги корисного навантаження;

- проєктування архітектури інформаційної системи.

- реалізація програмного забезпечення для підсистеми управління оператора.

Розв'язання цієї проблеми можливе шляхом створення інтегрованої безпілотної системи, що поєднує мобільність з лабораторною точністю. Актуальність роботи також підсилюється необхідністю розроблення інформаційних систем для передачі телеметрії в реальному часі в умовах нестабільного покриття мобільних мереж над водоймами.

1 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз комерційних безпілотних систем для моніторингу водойм

На ринку присутні різні типи комерційно представлених безпілотних рішень для моніторингу та відбору проб води, які базуються на різних типах безпілотних апаратів.

БПЛА — безпілотні літаючі апарати з вертикальним зльотом і посадкою та мультикоптери. Використовуються для завдань з повітря: дистанційної візуальної інспекції та доставки невеликих проб-контейнерів із важкодоступних місць [1]. Є корисними завдяки швидкості покриття великої площі і доступу до віддалених місць, проте мають малу вантажопідйомність, короткий час польоту та ризик шуму та впливу на дику природу.



Рисунок 1.1 — Типовий вигляд БПЛА [1]

АПС — автономні поверхневі судна: катамарани, середні та великі безпілотні судна. АПС часто можуть бути повністю автономними, напівавтомними або переведеними на ручне керування. Програмовані для слідування поточним параметрам і маршрутним точкам, деякі автономні човни здатні орієнтуватися та уникати перешкод, що робить їх придатними для збору даних, автономної гідрографічної зйомки та військового застосування [2]. Вони зручні відбором проб води з різних глибин насосами, можливістю встановлення датчиків. Є стабільними

на воді, мають більший час роботи та можливість нести важке обладнання, проте доступні тільки по поверхні, залежні від впливу хвиль та течій.



Рисунок 1.2 — Приклад вигляду автономного поверхневого судна [2]

АПА — автономні підводні дрони та апарати, використовуються для занурення на глибину, відбору проб води із глибин, встановлення сенсорних модулів та картографування дна водойми. Перевагою технології АПА є те, що вони можуть працювати без прямого командування з боку оператора. Дослідники розробляють траєкторію та місію АПА перед зануренням, потім розгортають АПА, а пізніше дістають робота та завантажують зібрані дані для обробки та перегляду [3]. Проте технологія потребує складного технічного обслуговування, є високовартісною та обмежена у часі автономної роботи.

Гібридні системи — комбінації з різних типів безпілотних апаратів, є гнучкими та універсальними із можливістю виконувати задачі притаманні різним типам апаратів.

Задля вирішення проблеми існують різні варіанти від виробників БПЛА, автономних водних та підводних суден, які представлені на ринку.

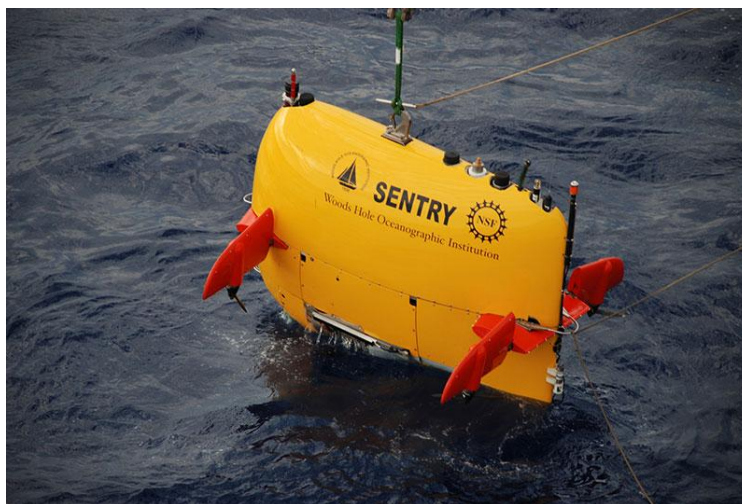


Рисунок 1.3 — Приклад автономного підводного судна [3]

Blue Robotics із судном BlueROV2 — є підводним ДПА з широкими можливостями налаштування, призначений для інспекцій, досліджень і розвідок. Завдяки векторній конфігурації з шістьма двигунами, програмному забезпеченню з відкритим вихідним кодом і можливості модульного розширення, він забезпечує професійні можливості без високої вартості традиційних систем [4]. Маючи багато пристроїв, що використовуються, BlueROV2 є одним з найпопулярніших підводних дронів, здатним адаптуватися як до простих, так і до складних місій [5].

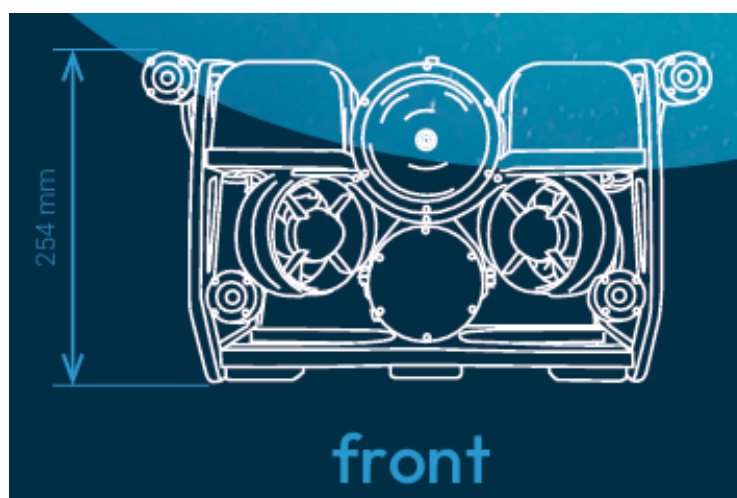


Рисунок 1.4 — Схематичне зображення судна BlueROV2 [5]

Розроблений як робочий або дослідницький ДПА, BlueROV2 має відкриту раму, яка несе стандартні компоненти ДПА. Ця конструкція є міцною та

розширюваною, що дозволяє легко прикріпити додаткові аксесуари, такі як захват або скануючий ехолотатор. Також дрон має виділене місце для корисного навантаження (додаткові корпуси акумулятора, проби води). BlueROV2 призначений для модифікації та розширення. Його можна адаптувати практично до будь-якого підводного завдання.

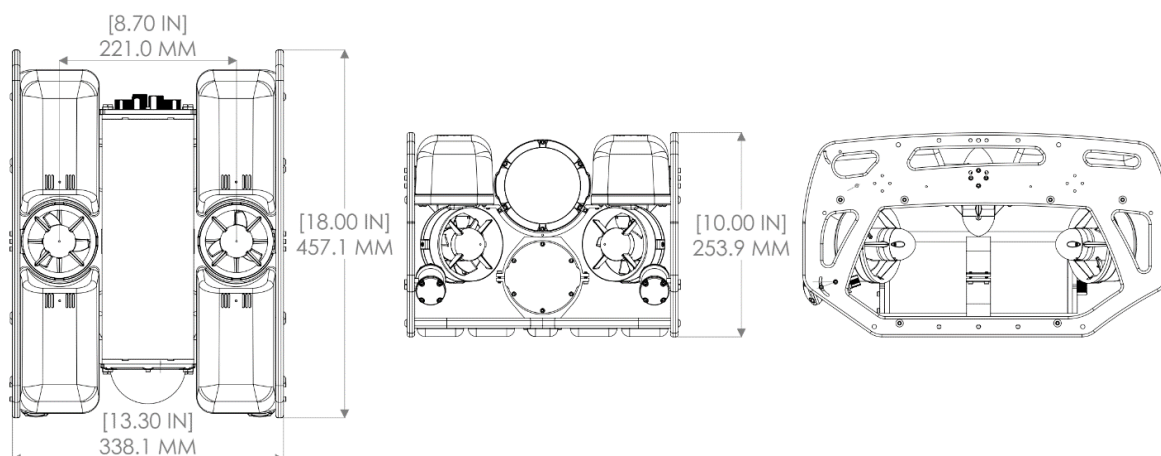


Рисунок 1.5 — Креслення BlueROV2 у трьох проекціях [4]

Управління BlueROV2 здійснюється за допомогою контролера руху Navigator і BlueOS. Navigator розроблений і створений для ДПА та інших робототехнічних додатків. Він має вбудовані датчики, включаючи інерційний вимірювальний блок для вимірювання орієнтації, магнітометр для вимірювання курсу компаса та датчик виток, щоб попереджати про потрапляння води всередину. Він має 16 виходів, які можна підключити до двигунів, ліхтарів, захватів та інших аксесуарів [4].

Sea Robotics мають різні класи АПС рішень в залежності від масштабу та потреб проєкту. SR Utility Class — АПС оснащені набором налаштованих датчиків, включаючи сонар з боковим скануванням і зонд якості води [6].

Розглянемо SR Utility 3.0 який розроблений як АПС для відбору проб води з важким навантаженням у 2019 році.

Велика палуба корисного навантаження та бортовий комп'ютер дозволяють операторам передавати різноманітні гідрографічні дані. Драйвер АПС від

SeaRobotics забезпечує з'єднання, роблячи АПС інтуїтивно зрозумілим, безпечним і ефективним в експлуатації.

Кожен човен оснащений джерелами живлення 5 В, 12 В та 24 В постійного струму для датчиків та обладнання. Також доступні USB та Ethernet, що дозволяє користувачам обмінюватися та інтегрувати різні датчики та обладнання [7].



Рисунок 1.6 — Зовнішній вигляд SR Utility 3.0 [7]

SR Endurance Class клас представлений двома АПС. SR-Endurance 7.0 — багатоцільовий АПС, який підходить для прибережних і морських операцій. SR-Endurance 7.0 має дизель-електричну гібридну силову установку, що дозволяє забезпечити тихіший режим роботи, необхідний для акустичних досліджень, автоматизовану систему запуску і може експлуатуватися як пілотоване або безпілотне судно. Рівень автономності можна налаштувати для кожної місії. Рівні автономності варіюються від базового — мінімальний інтелект, зрілого, тобто пілотований нагляд, та розширеної автономності — повна автоматична реалізація місії [8].

SR-Endurance 8.0 оснащена гідрографічною лебідкою в поєднанні з вузлом для посадки ДПА. Система обладнана для розгортання буксируваних інструментів або корисного навантаження вагою 100 кг. SR-Endurance 8.0 оснащений

алюмінієвим корпусом, що допомагає функціонувати в морі при несприятливих умовах.



Рисунок 1.7 — Вигляд SR-Endurance 7.0 [8]

Електрична силова установка з прямим приводом може досягати швидкості 10 та більше вузлів з ефективною крейсерською швидкістю 5 вузлів, забезпечуючи плавне регулювання швидкості у всьому діапазоні швидкостей, включаючи перехід від переднього до зворотного ходу. SR-Endurance 8.0 може виконувати місії без екіпажу тривалістю до 14 днів, що робить його ідеальною платформою для проведення морських інспекцій, океанських досліджень і поточних операцій зі спостереження за морем [8].

Liquid Robotics пропонують Wave Glider — автономний наземний транспортний засіб без екіпажу, який працює окремо або групою, надаючи дані в режимі реального часу протягом року без додаткового палива [9].

Платформа складається з надводного поплавця та підводного човна, які збирають та доставляють інформацію в режимі реального часу безперервно протягом кількох місяців поспіль [10].

Wave Glider SV5 має додаткову вантажопідйомність і збільшений збір і зберігання енергії. Більша платформа підходить для енергоємних корисних

навантажень і тривалого розгортання у високих широтах і умовах з низьким рівнем сонячної енергії [10].



Рисунок 1.8 — Вигляд SR-Endurance 8.0 [8]

Платформа DJI Matrice 350 RTK нового покоління має нову систему передачі відео та керування, більш ефективну акумуляторну систему, функції безпеки та можливості розширення [11]. Ця платформа має запас польоту 55 хвилин та максимальну вантажопідйомність 2,7 кг, що відповідає потребам різних сценаріїв експлуатації, таких як громадська безпека, інспекція та картографування.

Для проведення моніторингу якості води існують і інші рішення, наприклад, YSI EXO — платформа для моніторингу якості води. Модульна конструкція EXO має можливість кастомізації з цілим рядом багатопараметричних моделей, змінними датчиками, захистом від обростання, а також різними варіантами інтеграції та зв'язку [12].

Пропонуються різні види EXO для різних задач. Так, EXO1 – може підійти для систем моніторингу труб та свердловин підземних вод завдяки своєму невеликому діаметру [13].

EXO2 є платформою для моніторингу якості води, завдяки найширшому спектру комбінацій інтелектуальних датчиків для моніторингу якості води. EXO2 також дозволяє інтегрувати датчики сторонніх виробників [12].

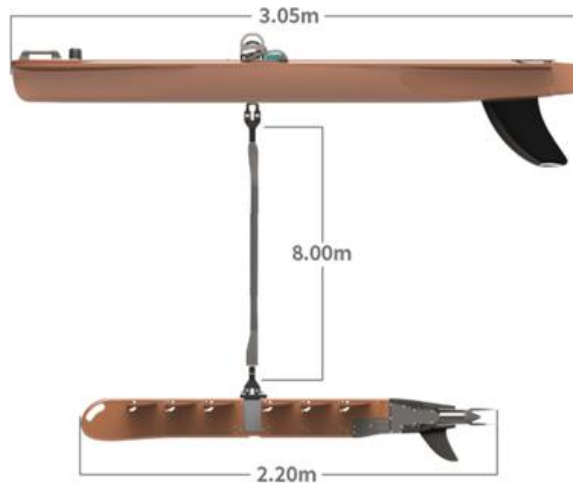


Рисунок 1.9 — Схематичне зображення Wave Glider SV3 [10]

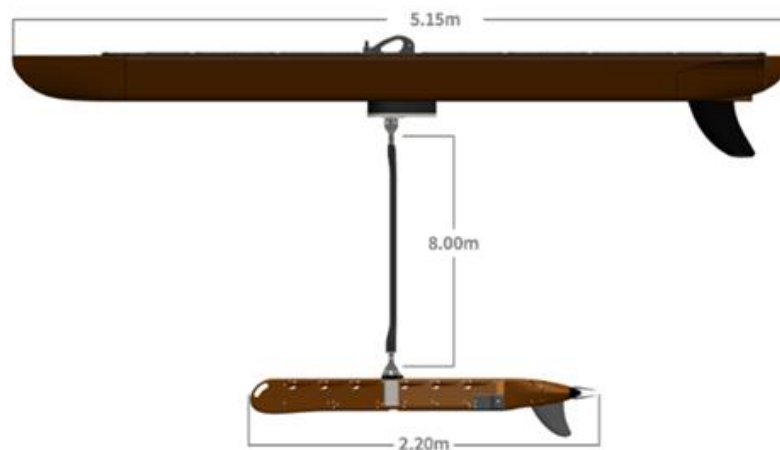


Рисунок 1.10 — Схематичне зображення Wave Glider SV5 [10]

EXO3 поєднує в собі маневреність EXO1 з протиобростаючим склоочисником EXO2. EXO3 включає зв'язок SDI-12 для інтеграції в платформи збору [14].

Порівняння доступних рішень наведено у таблиці 1.1. Згідно таблиці можна визначити що БПЛА – конкурентна опція для швидких, локальних проб, а АПС чи АПА – для великих або глибинних тривалих програм моніторингу.



Рисунок 1.11 — Matrice 350 RTK [11]



Рисунок 1.12 – Вигляд систем моніторингу якості води: EXO1, EXO2, EXO3 [14]

Проаналізувавши представлені рішення можна виокремити функціональні можливості сучасних комерційних платформ:

— мультисенсори для дослідження показників: рН, розчиненого кисню, електропровідності, температури, мутності води, спостереження з використанням спектрометрів;

— автоматичний відбір проб: механізми для збору води в стерильні контейнери, збереження проб при контролі температури та запечатування;

— передача даних у реальному часі: використання технологій GSM, 4G або 5G, супутникового зв'язку, тощо;

— навігація та позиціонування у автономному режимі: використання GPS, інерційних навігаційних систем, локальних систем позиціонування;

— телеметрія та віддалене управління: застосування інтерактивних панелей, API для інтеграції з хмарним середовищем та системами аналізу;

— модульність та розширюваність: можливість швидкої заміни сенсорів та додавання спеціалізованих модулів.

Таблиця 1.1 — Порівняння безпілотних апаратів

Параметр / Платформа	БПЛА	АПС	АПА/ДПА
Час місії	40хв	дні / місяці	декілька годин
Вантажопідйомність	до 5кг	більше 10кг	10-30кг
Відбір глибинних проб	обмежений	можливий	можливий
Розгортання	легке	ускладнене: потребує місце для спуску	складне: потребує носія
Складність обслуговування	низька	середня	висока
Регуляторна політика	сувора	потребує дозволів	потребує дозволів

Основними ринками, зацікавленими у використанні подібних рішень є державні моніторингові служби з екології та органи з контролю довкілля та контролю якості води; промислові підприємства: рибні господарства, енергетика, нафтові платформи, портова інфраструктура; наукові дослідження та міжнародні

екологічні програми із збереження довкілля, органи з управління водними ресурсами та агромоніторинг.

1.2 Переваги та недоліки існуючих технологій

Враховуючи різні типи безпілотних платформ, які використовуються у існуючих комерційних рішеннях, можна виокремити переваги безпілотних рішень:

— покращене просторово-часове покриття: можливість частих і регулярних виїздів до великої кількості точок без залучення людей. Це важливо для раннього виявлення інцидентів та підвищення оперативного реагування на виявлені за допомогою моніторингу проблеми;

— зниження ризиків для персоналу: відбір проб у небезпечних, забруднених або віддалених місцях виконується без залучення людей;

— підвищення відтворюваності та точності: автоматизовані методи відбору та стабільні протоколи дозволяють зменшити людський фактор і похибки;

— швидке транспортування даних: можливість передавати результати в реальному часі, що сприяє швидкому аналізу й прийняттю рішень;

— ефективність при масштабуванні: з часом вартість моніторингу на точку суттєво знижується при масштабуванні операції.

Проте, безпілотні рішення також мають деякі недоліки:

— висока початкова вартість: багато АПС та АПА платформ мають високу вартість, що може бути занадто дорогим рішенням для невеликих організацій;

— обмежена автономність: час роботи платформи, особливо у випадку дронів або малих АПС, обмежений батареями, це обмежує дальність та тривалість операцій;

— чутливість до умов довкілля: хвилі, течії, складні погодні умови, крига або сміття можуть ускладнювати або робити неможливими операції;

— складність сервісного обслуговування та калібрування: сенсори потребують регулярної калібрування, очищення та технічного обслуговування, що підвищує операційні витрати безпілотного рішення;

— проблеми з регулюванням та доступом: у деяких юрисдикціях існують обмеження на використання БПЛА та автономних суден: обмеження на польоти, правила судноплавства; що вимагає погодження використання з регуляторами;

— обмеження по глибині та вантажопідйомності: більшість безпілотних платформ, такі як БПЛА, не мають можливості збирати проби на глибині, а також мають значно обмежену вантажопідйомність;

— надійність та відмовостійкість: складні системи зі значною кількістю електроніки й програмного забезпечення потенційно мають більше точок відмови.

Окремо потрібно зауважити технічні ризики та кібербезпеку, адже такі безпілотні рішення використовують мережу інтернет для зв'язку із сервером. Віддалений контроль і передача даних вимагають захищених каналів, шифрування й механізмів автентифікації, а також механізми захисту від ризиків зіткнення з суднами або людьми, підйомом і падінням дронів, загроз крадіжок обладнання, неправильного процесу відбору проб, що може призвести до забруднення зразків і непридатності для аналізу.

1.3 Вибір платформи для проєкту

Враховуючи дані існуючих комерційних платформ, можна сформулювати вимоги до проєкту:

— платформа: платформа з запасом тяги більше 1.5 від номінально необхідної вантажопідйомності;

— автономність: реальний час місії з урахуванням доставки, збору, повороту більше 25-30 хв із резервом для маневрів;

— вантажопідйомність: можливість нести 0.5-1.5 кг корисного навантаження: картридж, насос, герметичний механізм;

— система відбору: стерильний картридж із дистанційною герметизацією або перистальтичний насос із одноразовою трубкою;

— позиціонування: GNSS із можливістю підключення RTK або точність позиціонування $\leq 1-2$ м при посадках;

— безпека: автоповернення при низькому заряді, система уникнення перешкод, GPS-трекінг і плавучі елементи для картриджа та корисного навантаження.

Грунтуючись на вимогах завдання і існуючих варіантах на ринку безпілотних апаратів для збирання проб води для виконання завдання обрана платформа на базі БПЛА, вони є достатньо мобільними, не потребують складної інфраструктури, та можуть збирати проби із різних водойм без додаткової участі людей. Вантажопідйомність БПЛА є достатньою для переносу однієї проби води і вони мають достатній запас енергії для покриття середніх відстаней для збирання проб.

Для відбору проб використані картриджі типу SPE які відправляються у лабораторію для аналізу. Картридж SPE є найбільш поширеним і є стандартизованим для використання у лабораторіях аналізу води [18].

Висновки до розділу 1

В комерційному середовищі представлена велика кількість безпілотних платформ для моніторингу водних ресурсів, включаючи автономні поверхневі судна, підводні апарати та літальні комплекси. Проте детальний аналіз доступних на ринку рішень вказує на низку суттєвих обмежень, що ускладнюють їх використання для задач оперативного контролю. Більшість водних та підводних систем мають недостатній рівень мобільності та автономності: вони потребують спеціалізованої інфраструктури для спуску на воду, складного технічного обслуговування та людського втручання для транспортування між об'єктами дослідження. Експлуатація надводних дронів суттєво залежить від гідрологічних умов — сильні течії, хвилі або наявність водної рослинності можуть блокувати їх рух чи спотворювати результати вимірювань, чого позбавлені повітряні засоби.

Недоліком існуючих АПС та АПА рішень є обмеження зони покриття моніторингу лише однією водоймою без можливості швидкого перебазування, що робить їх економічно неефективними. Крім того, висока вартість таких комплексів обмежує їх масове впровадження. Водночас, мультироторні системи здатні

виконувати завдання у режимі вертикального зльоту з будь-якої необладнаної точки берега, забезпечуючи вищу оперативність розгортання та швидкість доставки проб до місця аналізу.

Для вирішення завдання, за результатами аналізу, в якості платформи для реалізації проєкту обраний безпілотний літальний апарат (БПЛА). Використання БПЛА забезпечує необхідну гнучкість, дозволяючи швидко дістатися будь-якої точки водойми з берега, та є автономним у роботі, мінімізуючи ризики для персоналу та зменшуючи час на виконання місії. Вибір БПЛА також підкріплений можливістю інтеграції легких систем відбору проб води, що вирішує проблему обмеженої вантажопідйомності, зберігаючи при високу мобільність комплексу.

2 ПРОЄКТУВАННЯ СИСТЕМИ

2.1 Розроблення структурної схеми

Проектowana система являє собою інтегрований програмно-апаратний комплекс, структурна схема якого наведена у додатку Б, розроблений для автоматизації задач відбору проб води. На відміну від традиційних методів ручного відбору, дана система забезпечує безперервність потоку даних та мінімізує вплив людського фактору.

Архітектура системи побудована за багаторівневим ієрархічним принципом, що забезпечує логічне розмежування процесів керування, виконання та обробки даних.

Рівень оператора — клієнтський рівень, точка входу користувача в систему. На цьому рівні відбувається стратегічне планування місії. Оператор через графічний інтерфейс визначає водойму для дослідження, задає маршрутні точки та параметри відбору: глибина, об'єм проби. Функція оператора зводиться до нагляду, тоді як безпосереднє керування польотом здійснює автоматика.

Рівень безпілотного апарату — це фізичний рівень взаємодії з середовищем. БПЛА виступає автономним агентом, який отримує високорівневі команди, наприклад, «летіти в точку А», «відібрати пробу», і самостійно перетворює їх у керуючі сигнали для двигунів та приводів. Цей рівень також відповідає за первинну агрегацію сирих даних із сенсорів та забезпечення безпеки польоту.

Серверний рівень — виступає центральним хабом системи, що забезпечує асинхронний обмін даними між оператором та флотом дронів. Він відповідає за довгострокове зберігання телеметрії, валідацію отриманих даних, виконання ресурсомістких алгоритмів аналізу, наприклад, побудову теплових карт забруднення, та візуалізацію результатів.

Така декомпозиція дозволяє масштабувати систему: підключати декілька БПЛА до одного сервера або надавати доступ до даних різним групам екологів без зміни апаратної частини дронів.

2.2 Розроблення функціональної схеми

На основі структурної схеми розроблена функціональна схема, наведена у додатку В. Одним з важливих завдань є забезпечення електромагнітної сумісності між силовими ланцюгами двигунів та чутливими вимірювальними колами гідрохімічних сенсорів. Для розв'язання цієї проблеми застосоване екранування сигнальних кабелів та просторове рознесення компонентів.

В якості носія для апаратури обрана класична гексакоптерна схема на базі рами F550 (рисунок 2.1). Такий вибір обумовлений тим, що шестироторна конфігурація забезпечує важливу для місій над водою надлишковість тяги: у разі відмови одного з двигунів апарат здатен зберегти керованість та здійснити аварійну посадку. Рама виготовлена із композитного матеріалу на основі скляного волокна, армованого поліамідного нейлону, що робить конструкцію стійкою до деформацій при падіннях та ударах, а також забезпечує достатню жорсткість для мінімізації низькочастотних вібрацій [15].



Рисунок 2.1 — Зовнішній вигляд рами F550 [15]

Роль центрального вузла керування польотом виконує польотний контролер, з трьома дубльованими інерціальними вимірювальними модулями. Ці сенсори працюють у комплексі із розширеним фільтром Калмана EKF3, який дозволяє

оцінювати просторове положення апарату в реальному часі. Така система є важливою для фільтрації високочастотних вібрацій, що виникають при асинхронній роботі шести роторів, забезпечуючи стабільність платформи під час зависання [16].

Система електроживлення безпілотного комплексу реалізована за схемою з розділенням на два незалежні контури для підвищення надійності. Перший, силовий контур, відповідає за подачу струму високої напруги на електронні регулятори ходу та тягові двигуни. Другий, логічний контур, через перетворювачі напруги забезпечує стабілізоване живлення номіналом 5В для бортового комп'ютера та польотного контролера, а також напруги 3.3В для живлення периферійних сенсорів. Таке розділення дозволяє мінімізувати електричні шуми на шині передачі даних, що виникають при різких змінах навантаження на двигуни.

Для забезпечення навігації використовується модуль супутникового позиціонування. Його підключення до контролера реалізоване через промислову шину CAN з використанням протоколу UAVCAN (DroneCAN). Це рішення забезпечує високу завадостійкість передачі навігаційних даних в умовах щільної компоновки електроніки [17]. Налаштування режиму кінематики реального часу (RTK) здійснене через NTRIP-клієнт, що дозволяє бортовій системі отримувати корегуючі поправки від мережі наземних базових станцій через Інтернет. Завдяки частоті оновлення даних у 5 Гц система досягає точності позиціонування до 2 сантиметрів.

Гідравлічна схема модуля відбору проб спроектована для роботи з картриджами SPE стандартного лабораторного типорозміру об'ємом 3–6 мл. Для прокачування води крізь сорбент картриджа використовується прецизійний перистальтичний насос, що приводиться в дію кроковим двигуном. Вибір на користь крокового двигуна замість звичайного колекторного дозволяє реалізувати точне дозування об'єму прокаченої води V . Система розраховує об'єм, базуючись на кількості зроблених кроків N та заздалегідь визначеному калібрувальному коефіцієнту k_{vol} за формулою (2.1):

$$V = N \cdot k_{vol}. \quad (2.1)$$

Такий підхід спрощує конструкцію, оскільки усуває необхідність встановлення додаткового датчика витрати рідини, який міг би засмітитися мулистю водою.

Архітектура підключення вимірювальних сенсорів реалізована шляхом об'єднання їх в єдину мережу з використанням комбінованої топології «зірка» та «шина», вибір якої залежить від типу цифрового інтерфейсу конкретного датчика. Топологія «шина» застосовується для групи сенсорів, що підтримують протоколи з адресним зверненням, зокрема I2C. До цієї групи належать сенсори водневого показника (рН) та електропровідності. Також за шинною топологією через протокол 1-Wire підключений цифровий сенсор температури DS18B20. Завдяки наявності у кожного з цих пристроїв унікальної цифрової адреси, контролер має можливість опитувати їх послідовно, використовуючи одні й ті ж фізичні порти вводу-виводу, що економить ресурси системи.

У свою чергу, топологія «зірка» використовується для тих периферійних пристроїв, які вимагають окремого каналу зв'язку або високої швидкості передачі даних за протоколом UART. У цій схемі польотний контролер виступає центральним вузлом «зірки». Від нього до кожного UART-пристрою прокладена окрема пара проводів (Rx/Tx). Таким чином під'єднані GNSS-приймач, оптичний датчик розчиненого кисню, а також телеметричний модем. Кожен із цих компонентів займає свій фізичний порт на контролері, що гарантує відсутність колізій при передачі великих масивів даних.

Всі сенсори інтегровані в систему відповідно до топології та підтримуваних протоколів (додаток В). Сенсор рН моделі Atlas Scientific EZO-pH використовує інтерфейс I2C. Для забезпечення точності вимірювань програмно реалізований алгоритм триточкового калібрування за буферними розчинами рН 4.01, 7.00 та 10.00. Окрім того, у програмному коді імплементована формула Нернста [18] для автоматичної температурної компенсації показників, оскільки фізико-хімічні властивості електрода змінюються залежно від температури середовища: зміна

температури на 1°C може спричинити похибку вимірювання рН приблизно на 0.003 одиниці, що є суттєвим для лабораторної точності.

Визначення концентрації розчиненого кисню виконується відповідним датчиком, який під'єднаний з використанням послідовного інтерфейсу UART. Перевагою цього датчика є використання оптичного (люмінесцентного) методу вимірювання замість традиційного електрохімічного.

Сенсор електропровідності з'єднаний з системою через інтерфейс I2C. Особливістю його інтеграції є застосування апаратного ізолятора живлення та ліній даних на базі мікросхем. Це необхідно тому, що сенсори провідності є вкрай чутливими до так званих «земляних петель», які виникають у спільній системі живлення БПЛА, де мінусовий провід є спільним для силової та сигнальної електроніки. Без гальванічної розв'язки покази датчика були б спотворені електричними шумами від двигунів.

Вимірювання температури води здійснюється відповідним датчиком, підключеним через цифрову шину 1-Wire. Для забезпечення стабільності цифрового сигналу на довгому кабелі (довжиною до 1 метра), який опускається у воду, використаний підтягуючий резистор.

2.3 Вибір окремих елементів системи

Для автоматизованого збору проб води обрано SPE-картриджну технологію, яка дозволяє герметично зберігати зразки для подальшого лабораторного аналізу. Вода прокачується через спеціальний картридж, що містить сорбент. Розчинені речовини, забруднювачі, затримуються на сорбенті, а очищена вода виводиться назовні — це дозволяє транспортувати лише легкий картридж із сконцентрованими речовинами, а не важкі ємності з водою, що суттєво економить заряд батареї БПЛА. Крім того, герметичність системи SPE-картриджів виключає ризик перехресного забруднення зразка під час транспортування.

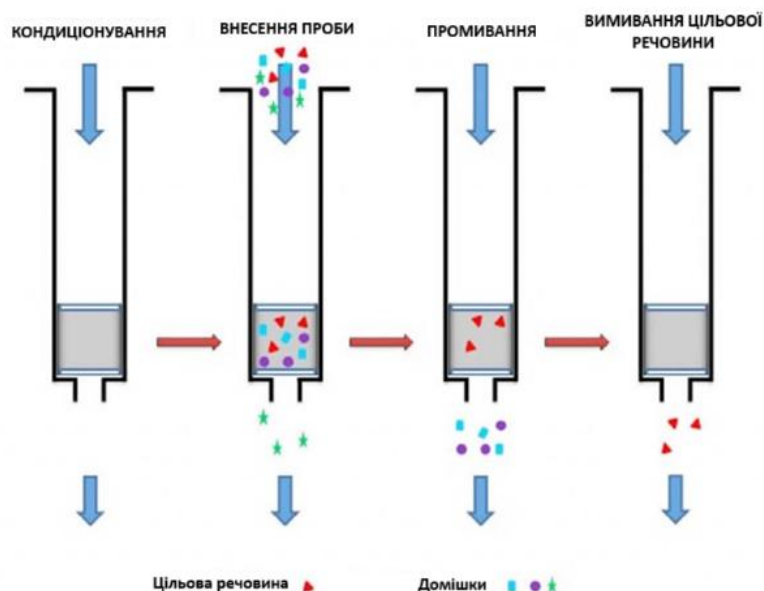


Рисунок 2.2 — Схематичне відображення процедури аналізу зразка з використанням SPE-картриджу [19]

Модуль відбору проб складається з перистальтичного насоса, який забезпечує точне дозування об'єму води, та системи силіконових трубок. Для контролю процесу використовується датчик глибини занурення забірної трубки, а електромагнітний клапан герметизує контур після завершення прокачки, запобігаючи витіканню або потраплянню повітря.

Для отримання миттєвих даних про стан водойми БПЛА оснащений комплексом цифрових та аналогових сенсорів.

Сенсор рН — Atlas Scientific EZO-pH. Цей модуль, який має вбудовану схему температурної компенсації, що є важливим для польових умов, де температура води може змінюватися. Він підключається до головного контролера через цифровий інтерфейс UART, забезпечуючи високу завадостійкість даних [20].

Оптичний датчик розчиненого кисню Hamilton VisiFerm DO. Використання оптичної технології вимірювання на відміну від традиційної електрохімічної дозволяє отримати більш стабільні результати без необхідності калібрування чи заміни електроліту. Датчик стійкий до впливу сірководню та інших речовин, які можуть зіпсувати звичайні електроди [21].



Рисунок 2.3 — Сенсор pH Atlas Scientific EZO-pH [20]



Рисунок 2.4 — Датчик розчиненого кисню VisiFerm Dissolved Oxygen Optical Sensor [21]

Датчик температури DS18B20. Цифровий термометр, що працює за протоколом 1-Wire. Він дозволяє отримувати дані про температуру води, які необхідні не лише як самостійний параметр, але і для температурної корекції показників pH та електропровідності. Датчик має водонепроникний корпус з нержавіючої сталі [22].

Датчик електропровідності Atlas Scientific EZO-EC. Цей сенсор дозволяє оцінити загальну мінералізацію води та її солоність. Вимірювання електропровідності є індикатором наявності розчинених солей, важких металів або

інших іонів-забруднювачів. Модуль EZO-EC забезпечує широкий динамічний діапазон вимірювань, від прісної води до морської [24].



Рисунок 2.5 — Датчик температури DS18B20 [23]



Рисунок 2.6 — Датчик електропровідності Atlas Scientific EZO-EC [24]

Датчик мутності DFRobot SEN0189. Цей сенсор працює за оптичним принципом вимірювання розсіювання світла зваженими частками. Дозволяє оцінити прозорість води, наявність мулу, водоростей або промислових скидів. Датчик може працювати як в аналоговому режимі, видаючи напругу пропорційну мутності, так і в цифровому, що дає гнучкість у налаштуванні системи [25]

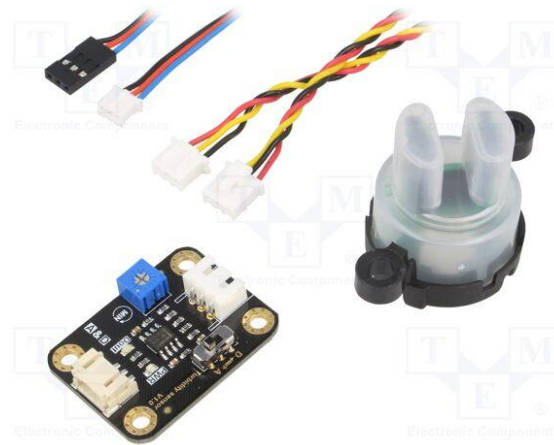


Рисунок 2.7 — Датчик мутності DFRobot SEN0189 [26]

Польотний контролер Pixhawk Cube Orange+. Він є частиною широкої екосистеми модулів автопілотів і плат носіїв. Блок управління польотом та інерціальний блок керування розміщені у компактному форм-факторі. Сумісний з платою CubePilot, всі входи та виходи проходять через 80-контактний роз'єм DF17, що дозволяє використовувати роз'єм для виробників комерційних систем [17].



Рисунок 2.8 — Зовнішній вигляд контролеру Cube Orange+ [17]

Модуль GPS GNSS Here 3+. Він є економічно ефективною GNSS-системою, яка підтримує режим RTK, є стійким до пилу та води. Модуль має високу швидкість передачі даних, можливість оновлення, стійкість до шуму. Він має вбудований інерційний вимірювальний блок, який складається з компасу, гіроскопу і акселерометра, для складних навігаційних потреб [18].



Рисунок 2.9 — Зовнішній вигляд GNSS модуля Here 3+ [18]

2.4 Проектування програмної архітектури

Апаратна складова системи, наведена у додатку Б, базується на двох фізично рознесених вузлах: наземній станції керування — комплексу обладнання на робочому місці оператора і хмарних серверів; та мобільній платформі — спеціалізованого мультикоптера, адаптованого до роботи в умовах підвищеної вологості.

2.4.1 Архітектура бортового програмного забезпечення

Програмний стек розробленого комплексу має ієрархічну структуру та реалізований на трьох взаємопов'язаних рівнях: рівень мікроконтролера польоту, рівень бортового одноплатного комп'ютера та рівень віддаленого сервера. Безпосередньо на контролері розгорнута прошивка ArduCopter 4.x. У конфігурації прошивки здійснене налаштування параметрів для специфічних умов місії. Зокрема, параметр `RTL_ALT` встановлено на рівні 3000 см, що задає безпечну висоту повернення додому. Параметр `WPNAV_SPEED` обмежує горизонтальну швидкість польоту по точках до 500 см/с для плавності руху. Коефіцієнт `PSC_POSXY_P` PID-регулятора налаштований для максимально жорсткого утримання горизонтальної позиції, щоб протидіяти вітровим навантаженням.

Параметр SERIALx_PROTOCOL забезпечує розширені можливості телеметрії при передачі даних на бортовий комп'ютер.

На другому рівні, на бортовому одноплатному комп'ютері Raspberry Pi 4, функціонує основний керуючий сервіс, написаний мовою програмування Python, який виконує роль «менеджера місії». Архітектура цього сервісу є асинхронною та багатопотоковою, що дозволяє виконувати кілька завдань паралельно. Перший потік, використовуючи бібліотеку rpyavlink, підтримує постійний двосторонній зв'язок з польотним контролером, отримуючи дані телеметрії, такі як географічні координати та поточний статус системи. Другий потік відповідає за роботу з вимірювальним обладнанням: він циклічно опитує шину I2C, використовуючи бібліотеку smbus2 для взаємодії з сенсорами Atlas Scientific. Отримані дані зчитуються у сирому форматі ASCII, після чого проходять етап парсингу та конвертації у числовий формат з плаваючою комою для подальшої обробки. Третій незалежний потік керує низькорівневими GPIO-пінами комп'ютера, що необхідно для фізичної активації драйвера крокового двигуна насоса та перемикачів реле електромагнітного клапана в потрібні моменти часу.

2.4.2 Архітектура серверної частини

Серверна інфраструктура являє собою обчислювальний вузол, який реалізований у хмарному середовищі. Він відповідає за зберігання великих масивів даних, виконання ресурсомістких аналітичних алгоритмів та підтримку веб-сервісів.

RESTful API. Слугує шлюзом для обміну JSON-пакетами між дроном, базою даних та клієнтським інтерфейсом.

Реляційна БД PostgreSQL використовується для зберігання даних місій, треків польотів та масивів екологічних даних.

Аналітичний модуль, представлений програмними скриптами, що обробляють вхідні дані, виявляють аномалії (різкі стрибки рН або мутності) та генерують звіти.

Графічне представлення функцій системи для оператора, відображення доступних БПЛА, місій та результат місій разом з їх аналізом реалізований через веб-інтерфейс.

Програмне забезпечення системи об'єднує всі компоненти в єдиний логічний контур. Вбудоване ПЗ БПЛА, забезпечує низькорівневе керування обладнанням. До нього входять алгоритми стабілізації польоту, драйвери для опитування сенсорів, логіка керування насосом та клапанами, а також підсистема безпеки, яка слідкує за станом заряду батареї та якістю зв'язку.

Серверне ПЗ виконує роль диспетчера, приймає телеметрію від дрона, записує її у реляційну базу даних, обробляє запити від оператора та формує команди керування. Важливим елементом є аналітичний модуль, який може автоматично виявляти аномалії у показниках води. Клієнтське ПЗ, реалізоване у вигляді веб-інтерфейсу, робить систему кросплатформною. Оператор отримує інструменти для візуального планування маршрутів на карті, перегляду результатів місій.

2.4.3 Архітектура підсистеми управління оператора

Підсистема управління оператора є важливою складовою контролю виконання місій і складається з кількох доменів, які відповідають за різні аспекти роботи комплексу:

— підсистема управління місіями відповідає за створення, налаштування та збереження параметрів місій та забезпечує доступ до історії завершених місій та результатів аналізу;

— підсистема моніторингу та телеметрії відповідає за обробку потоку даних від дронів у реальному часі і забезпечує візуалізацію стану дрона (заряд, позиція, статус виконання) для оператора;

— підсистема адміністративного керування забезпечує аутентифікацію користувачів та розмежування прав доступу, дозволяє додавати нові дрони у систему.

Архітектура побудована за класичним принципом розподілу відповідальності на рівні. Дані проходять через ці рівні послідовно.

Клієнтський рівень це точки входу в систему. Оператор взаємодіє через веб-інтерфейс та відправляє запити, що виступають у ролі команд. Дрон взаємодіє автоматично та генерує потік даних телеметрії.

Рівень інтерфейсів та буферизації приймає вхідні сигнали та маршрутизує їх.

Рівень бізнес-логіки проводить валідацію даних, обробку транзакцій та управляє станами місій.

Рівень доступу до даних конвертує об'єкти в запити до бази даних і навпаки.

Рівень зберігання представляє собою фізичне сховище, де знаходяться дані.

Архітектура використовує два різних підходи до обміну даними, залежно від задачі:

- запит-відповідь використовується оператором для управління;
- публікація-підписка використовується дроном для телеметрії.

2.5 Алгоритм функціонування системи

Алгоритм функціонування системи наведений у додатку Г. Оператор планує місію, вводячи параметри у веб-інтерфейсі. Запускаючи місію дані параметри передаються до БПЛА та розпочинається стадія ініціалізації та перевірки готовності БПЛА. По завершенню підготовки до польоту БПЛА виконує місію, відбирає зразок води, отримує сенсорні дані, які одразу відправляє на сервер та повертається назад, завершуючи місію. Опісля відібрана проба аналізується та формується звіт.

2.5.1 Алгоритм керування БПЛА

Принцип роботи навігаційної системи полягає у послідовному виконанню завдань наведених у додатку Д. Спочатку БПЛА проходить ініціалізацію, самотест та калібрування: перевіряється калібрування сенсорів, наявність достатньої

кількості супутників GPS, напруга батареї та справність виконавчих механізмів. Тільки при успішному проходженні тестів дозволяється зліт.

Фаза маршрутного польоту — дрон злітає та рухається до цільової точки за заданими координатами. Контролер коригує курс, компенсуючи знесення вітром. Після чого при досягненні точки відбору зразків води БПЛА зависає та стабілізується, вмикається режим утримання позиції.

Фаза виконання робочих операцій — активується насос модуля відбору зразків води, заповнюючи SPE-картридж. Паралельно відбувається опитування сенсорів рН, розчиненого кисню, електропровідності, температури, мутності з частотою кілька разів на секунду для отримання усереднених значень. Дані записуються у внутрішню пам'ять БПЛА та дублюються на сервер через 4G. По завершенню відбору проб БПЛА переходить у режим повернення та розраховує енергетично оптимальний маршрут повернення на точку старту.

У системі запрограмовані алгоритми дій на випадок позаштатних ситуацій: при втраті зв'язку або критичному розряді батареї дрон автоматично переходить у режим повернення додому.

Висновки до розділу 2

В результаті проектування сформована архітектура безпілотної системи, яка розділяє систему на три логічні рівні: рівень оператора, серверний рівень та рівень безпілотної апаратури. Такий розподіл забезпечує масштабованість комплексу, дозволяючи керувати дронами через єдиний серверний хаб, а також мінімізувати вплив людського фактору на процес відбору проб води, залишаючи за оператором лише наглядні функції.

В якості платформи носія обрана гексакоптерна схема на базі рами F550, через необхідність забезпечення надлишковості тяги та безпеки польотів над водою: шість роторів дозволяють зберегти керованість та здійснити аварійну посадку навіть у разі відмови одного двигуна. Для забезпечення надійності роботи

електроніки спроектована система живлення з розділенням на силовий та логічний контури, що мінімізує вплив на чутливі цифрові компоненти.

Використання SPE картриджів дозволяє відмовитись від транспортування великих об'ємів води, замінивши їх легкими картриджами з концентратом забруднюючих речовин, що суттєво підвищує енергоефективність польоту. Впровадження перистальтичного насоса з кроковим двигуном дозволяє реалізувати точне дозування об'єму прокаченої рідини програмним шляхом, усунувши необхідність встановлення ненадійних датчиків потоку.

Для забезпечення точності позиціювання, інтегровано модуль GNSS із підтримкою RTK та використанням завадостійкого протоколу UAVCAN. Використання польотного контролера Cube Orange+ забезпечує стабільне зависання апарату над водою попри вібрації та вітрові навантаження.

3 РЕАЛІЗАЦІЯ І ТЕСТУВАННЯ ПІДСИСТЕМИ УПРАВЛІННЯ ОПЕРАТОРА

3.1 Вибір технологічного стеку

Для роботи системи з відбору води важливою є підсистема управління місіями та відображення стану дронів оператору. Ця підсистема повинна бути стабільною, легкою у користуванні але зберігати достатньо важливих даних і мати змогу переглянути як історію виконання та результати вже завершеної місії, так і спостерігати за виконанням поточних місій у реальному часі.

Ця підсистема базується на серверній архітектурі на основі фреймворку Spring Boot, який використовує мову програмування Java. Він спрощує налаштування та запуск програм, дотримуючись принципу «Convention over Configuration» — угода важливіша за конфігурацію. Це дозволяє створювати автономні додатки, які можна запустити просто як .jar файл, оскільки веб-сервер, наприклад, Apache Tomcat вже вбудований всередину [27].

Фреймворк Django є зручним, адже більшість бібліотек для аналізу масивів даних написані під мову Python. Об'єктно-реляційне відображення для бази даних, панель адміністратора, аутентифікація є предвстановленими і одразу доступні для використання. Головними недоліками є інтерпретована мова Python, тому виконання коду, як правило, повільніше, ніж у компільованої Java та доволі не гнучка структура самого фреймворку [28].

Використання Express.js у середовищі виконання JavaScript Node.js є популярним вибором для систем, де потрібен обмін даними в реальному часі, наприклад для передачі телеметрії дрона. Він використовує асинхронну, подійно-орієнтовану архітектуру. Також використання мови JavaScript спрощує розроблення, адже одна мова використовується як для серверної частини, так і для клієнтської. Проте якщо серверу потрібно виконати складні математичні розрахунки то, один потік, який використовує Node.js, блокується, і весь сервер зависає для інших користувачів [29].

На відміну від Node.js чи Django, Jakarta EE є промисловим стандартом зі строгою специфікацією [30]. Це гарантує надійність та стабільність архітектури, що є надважливим для підсистеми управління дронами та важливістю безперебійного моніторингу виконання місій. Через надійність, достатній рівень простоти створення та налаштування Jakarta EE обрана основою для створення підсистеми управління оператора. Основними компонентами Jakarta EE, які використовувались при створенні системи є JDBC, JPA, EJB та JAX-RS.

JDBC є API для з'єднання Java-додатків з базами даних. Він забезпечує фізичне з'єднання з базою даних, відправку SQL-запитів та отримання результатів, гарантуючи стабільне з'єднання та транзакційну цілісність.

Частиною стандарту JPA є EntityManager який виступає в ролі ОРВ — відображає об'єкти Java в таблиці бази даних. Він є інтерфейсом для роботи з сутністю — звичайним Java-класом, який прив'язаний до таблиці БД за допомогою анотації @Entity. Основною перевагою EntityManager є робота саме з Java-класами замість використання звичайних SQL запитів, що спрощує роботу з базою даних та гарантує надійність та безпеку виконання запитів.

EJB — це класи, які містять логіку додатку, тобто обробку всієї бізнес логіки системи. Контейнер EJB автоматично відкриває і закриває транзакції. Якщо під час запису даних стається помилка, EJB автоматично виконує відкат, щоб зберегти справну роботу бази, захищаючи її від порушення консистентності. Контейнер сам керує потоками, надаючи можливість одночасного звернення великої кількості користувачів до сервера без виникнення конфліктів.

JAX-RS є стандартом для створення веб-сервісів REST. Він використовує анотації, які перетворюють звичайний Java-клас на веб-контролер, який приймає HTTP-запити.

Завдяки використанню цих компонентів виходить потік обробки викликів:

— клієнт, тобто браузер або дрон, звертається до одного з endpoint, який є доступним у системі;

— resource що виступає у ролі контролеру приймає запит та автоматично конвертує JSON об'єкт у Java;

— EJB отримує від контролера Java об'єкт, який перевіряється та відкривається транзакція;

— Repository, який викликається Java біном, використовує EntityManager для виконання записів у базу даних, наприклад функції persist, що є створенням нового запису у таблиці;

— JDBC перетворює команду у SQL-запит і відправляє в базу даних.

Вибір між монолітною архітектурою та мікросервісами впливає на швидкість розроблення, складність підтримки та масштабованість проєкту. Традиційна модель розроблення програмного забезпечення (монолітна архітектура) дозволяє побудувати систему як єдиний, неподільний модуль. У такому додатку всі функціональні компоненти — інтерфейс користувача, бізнес-логіка, доступ до даних та службові сервіси — зібрані разом і розгортаються як один артефакт, наприклад, один файл .war або .jar у випадку Java. Всі частини програми працюють в одному процесі, використовують спільну пам'ять та зазвичай звертаються до однієї централізованої бази даних. Взаємодія між компонентами відбувається через прямі виклики методів всередині коду, що забезпечує високу швидкість обміну даними та транзакційну цілісність без необхідності використання складних мережових протоколів. Розроблення моноліту на початкових етапах є простішою, оскільки вся кодова база знаходиться в одному місці, що спрощує налагодження, тестування та розгортання, адже не потрібно налаштовувати координацію між багатьма незалежними сервісами.

В свою ж чергу мікросервісна архітектура — це підхід, при якому додаток розбивається на набір невеликих, незалежних сервісів, кожен з яких виконує свою специфічну бізнес-функцію. Кожен мікросервіс є автономним: він має власну кодову базу, може бути написаний на іншій мові програмування, використовує власну ізольовану базу даних і розгортається незалежно від інших. Взаємодія між цими сервісами відбувається через мережу, найчастіше за допомогою протоколів, таких як HTTP/REST або через черги повідомлень. Така децентралізація дозволяє великим командам працювати паралельно над різними частинами системи, не блокуючи одна одну, а також дає можливість масштабувати лише ті компоненти,

які перебувають під найбільшим навантаженням. Однак це вносить значну складність в інфраструктуру, вимагає налаштування систем моніторингу, балансування навантаження, обробки розподілених транзакцій та відмовостійкості мережі.

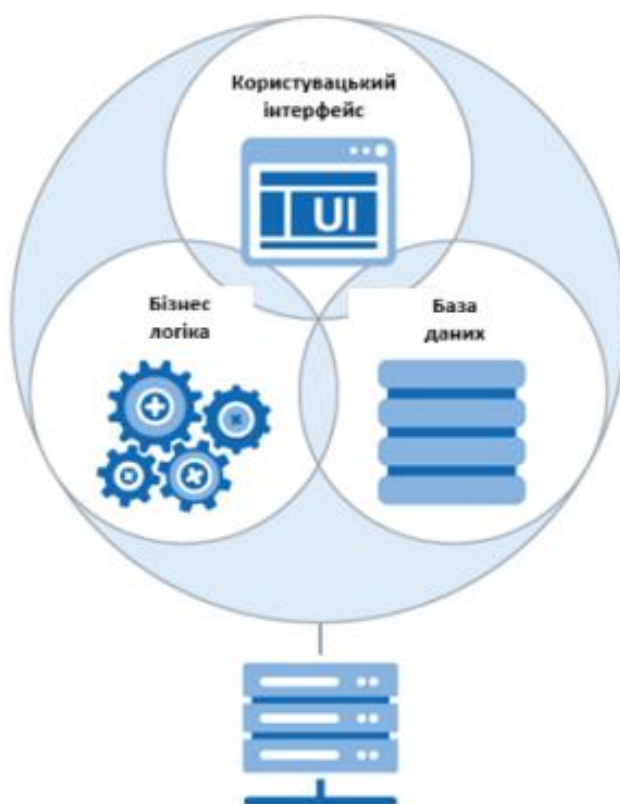


Рисунок 3.1 — Схематичне зображення монолітної архітектури [31]

Для реалізації програмного комплексу підсистеми менеджменту оператора для відбору проб води обрана монолітна архітектура. Цей вибір зумовлений специфікою проєкту, яка не вимагає складності, притаманної розподіленим системам. Система є локалізованою та має чітко окреслену предметну область з обмеженим набором функцій, які тісно пов'язані між собою: налаштування місій, збір телеметрії, відображення результатів місії. Розбиття цих функцій на окремі мережеві сервіси лише сповільнило б роботу системи через затримки мережі.

Проєкт не передбачає високого навантаження від тисяч одночасних користувачів. Система розрахована на роботу з невеликою кількістю операторів та

обмеженим парком дронів, тому горизонтальне масштабування окремих компонентів, яке є головною перевагою мікросервісів, тут є надлишковим. Один потужний сервер здатен ефективно обробити весь потік даних.

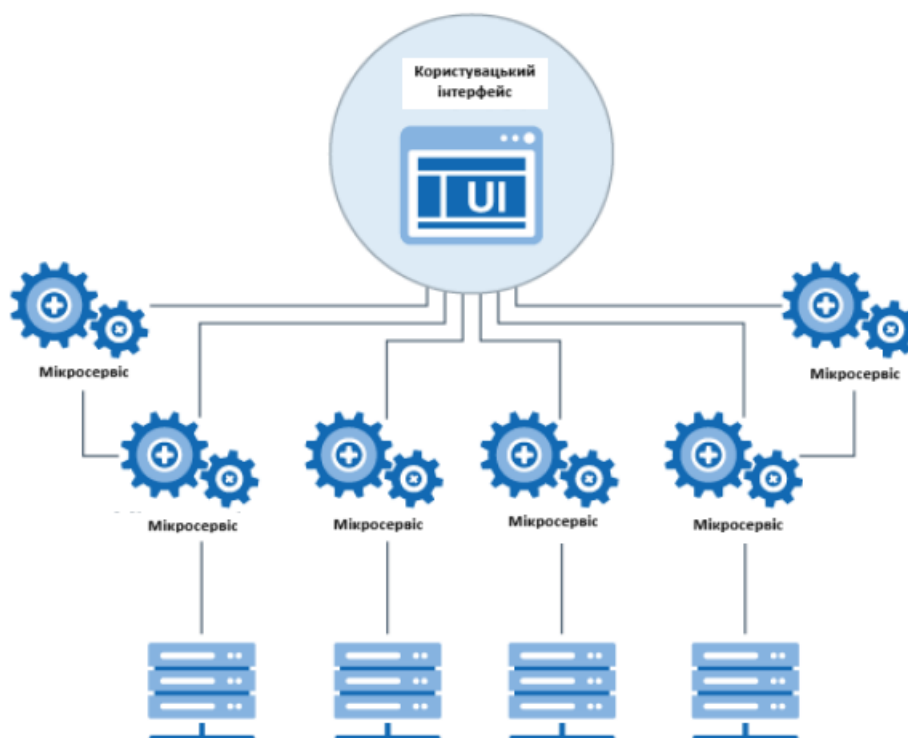


Рисунок 3.2 — Схематичне зображення мікросервісної архітектури [31]

Крім того, вибір моноліту значно спрощує процес розроблення та розгортання. Монолітна архітектура дозволяє зосередитися на реалізації складної бізнес-логіки та алгоритмів, гарантуючи при цьому простоту тестування та цілісність даних — ACID-транзакцій, без додаткових інженерних зусиль.

При проєктуванні архітектури програмного комплексу управління оператора одним із головних завдань є вибір системи зберігання даних, здатної ефективно обробляти різномірну інформацію: від часових рядів телеметрії до геопросторових координат. Аналіз вимог до системи показав, що структура даних у проєкті є чітко визначеною та має виражені зв'язки. Наприклад, кожна «місія» складається з заданої точки відбору, прив'язана до конкретного дрона та містить результати аналізу води. Така реляційна природа даних робить використання SQL-баз даних

більш доцільним порівняно з NoSQL-рішеннями, які, хоч і пропонують гнучкість схеми, часто програють у забезпеченні цілісності даних та складності побудови аналітичних вибірок зі зв'язками багато-до-багатьох.

Серед реляційних СУБД вибір зроблений на користь PostgreSQL, оскільки вона об'єктивно переважає інші популярні рішення, такі як MySQL або MariaDB, у контексті наукоємних та інженерних завдань. Вирішальним фактором стала її надійність та відповідність стандартам SQL. У системі екологічного моніторингу, де ціна втрати або спотворення даних вимірювань є високою, механізми забезпечення цілісності, такі як транзакційність та надійна система журналювання за принципом WAL — Write-Ahead Logging, гарантують збереження інформації навіть у випадку аварійного вимкнення живлення сервера.

Також варто зазначити, що PostgreSQL є об'єктно-реляційною системою, що дозволяє створювати власні типи даних та функції, інтегруючись із мовою Java через JPA/Hibernate. На відміну від комерційних рішень, таких як Oracle Database, PostgreSQL є повністю безкоштовним продуктом з відкритим вихідним кодом, що відповідає економічним вимогам до розроблення дипломного проєкту та дозволяє уникнути проблем із ліцензуванням при потенційному масштабуванні системи.

Для організації надійного та високошвидкісного каналу обміну даними між безпілотним апаратом та серверною частиною комплексу обрана розподілена платформа потокової передачі подій Apache Kafka. В архітектурі системи Kafka виступає в ролі центральної сполучної ланки — message broker, що забезпечує асинхронну взаємодію між бортовим комп'ютером дрона та сервером на базі Jakarta EE.

Вибір цього інструменту обумовлений необхідністю обробки часових рядів телеметрії. Бортовий комп'ютер Raspberry Pi, виступаючи в ролі продюсера, публікує пакети даних у топіки Kafka. На відміну від синхронних HTTP-запитів, де відправник змушений чекати підтвердження отримання кожного пакету сервером, Kafka дозволяє реалізувати патерн fire-and-forget — це важливим для польотного контролера, оскільки звільняє обчислювальні ресурси дрона для пріоритетних задач керування польотом, не блокуючи їх очікуванням мережевої відповіді.

Головним фактором впровадження Kafka стала її здатність забезпечувати буферизацію даних та стійкість до нестабільного мобільного зв'язку. В умовах реальної експлуатації над водоймами 4G-зв'язок може бути переривчастим. Kafka гарантує збереження послідовності повідомлень: якщо сервер тимчасово недоступний або перевантажений обробкою аналітики, дані з дрона не втрачаються, а зберігаються в логах брокера і будуть оброблені сервером одразу після відновлення з'єднання або зниження навантаження. Це забезпечує гарантовану доставку, що є обов'язковою вимогою для систем моніторингу.

Також використання Kafka забезпечує слабку зв'язність компонентів системи. Серверна частина на Jakarta EE підписується на топіки Kafka і обробляє вхідний потік даних у власному темпі, незалежно від швидкості їх генерації дроном. Крім того, архітектура Kafka дозволяє масштабувати систему: у майбутньому до платформи можна підключити додаткові мікросервіси, наприклад, сервіс архівування логів або модуль машинного навчання для аналізу аномалій у поведінці дрона, які будуть читати ті самі дані з Kafka паралельно з основним сервером, не вимагаючи жодних змін у програмному коді самого БПЛА.

3.2 Реалізація модулів

Виходячи з основних вимог системи, оператор повинен мати доступ до реєстрації дронів в системі, створення нових місій, нагляду за активними місіями та перегляду завершених місій, їх результатів та можливості залишити звіт щодо результатів місії після їх аналізу. Також всі сторінки як і запити повинні бути захищеними від несанкціонованого доступу, для виконання цієї вимоги реалізована система авторизації та реєстрації операторів. Для виконання поставленого завдання по створенню системи управління оператора, створені JSP сторінки, які виступають в ролі фронтенду застосунку і з якими безпосередньо взаємодіє оператор системи (таблиця 3.1).

Таблиця 3.1 — JSP сторінки у системі та їх призначення

№ п/п	Назва JSP сторінки	Призначення
1	home.jsp	Домашня сторінка, яку бачить оператор при відкритті сайту, з неї можна перейти до реєстрації або авторизації в системі
2	register.jsp	Сторінка реєстрації оператора, оскільки операторів з однаковим логіном не може бути декілька, при спробі створення дублюючого запису на сторінці з'являється помилка авторизації
3	register_success.jsp	Сторінка вдалої реєстрації, з'являється тільки у випадку якщо було додано нового оператора
4	login.jsp	Сторінка авторизації оператора, вводячи логін на пароль система перевіряє на відповідність введених даних і авторизує чи блокує запит відповідно
5	login_error.jsp	Сторінка помилки авторизації через невідповідність введених даних оператором для авторизації
6	dashboard.jsp	Основне меню навігації системи, з нього можна перейти у три основних блоки: список операторів, управління місіями та управління дронами у системі

№ п/п	Назва JSP сторінки	Призначення
7	dashboard_operators.jsp	Відображає список зареєстрованих операторів, їх логін та час реєстрації в системі
8	dashboard_drones.jsp	Відображає список доданих до системи дронів: їх ідентифікатор, кодову назву, статус та дату реєстрації дрона у системі
9	dashboard_drones_add.jsp	Сторінка реєстрації нового дрона у системі, оскільки, як і з операторами, дронів з однаковою кодовою назвою не може бути декілька, то при реєстрації нового дрона перевіряється наявність вже зареєстрованого дрона з тою самою кодовою назвою й повертає помилку авторизації або створює нового дрона
10	dashboard_drones_telemetry.jsp	Показує останній зареєстрований системою пакет телеметрії дрона, та його місцезнаходження на мапі, в разі якщо дрон ще не експлуатувався, тобто не був відправлений на жодну з місій, дані телеметрії будуть пустими
11	dashboard_missions.jsp	Список місій, створених у системі, оскільки кількість місій може сягати великих обсягів доцільно використано пагінацію списку, розділяючи його на декілька сторінок по дев'ять елементів у кожній, у кожній місії відображається заданий дрон для її виконання,

№ п/п	Назва JSP сторінки	Призначення
		створивший місію оператор, поточний статус місії та час її створення
12	dashboard_missions_add.jsp	Сторінка створення нової місії, оператор повинен ввести кодову назву дрону, якому буде надана місія та обрати місце відбору проби на мапі, оскільки кожен оператор є у системі вона самостійно вводить дані оператора у відповідні поля при створенні нового рядку у базі даних
13	dashboard_missions_view.jsp	Сторінка перегляду місії, дає можливість розпочати місію, що передає її дані на дрон з використанням брокеру Kafka, під час виконання місії на мапі малює маршрут дрону та його місцезнаходження у реальному часі за допомоги SSE, а також веде лог телеметрії дрона під час виконання місії, по завершенню місії дає можливість переглянути її результати
14	dashboard_missions_view_result.jsp	Сторінка результатів заведеної місії, відображає показники сенсорів та має спеціальне поле вводу для звіту, проведеного після аналізу проби, взятої дроном і доставленої до місця розташування

Таким чином система покриває всі аспекти управління оператора, надаючи зручний інтерфейс користування.

Для роботи системи спроектована реляційна база даних PostgreSQL. Вона складається з п'яти таблиць (таблиця 3.2). Структура БД наведена у додатку Е.

Таблиця 3.2 — Структура БД

№ п/п	Таблиця БД	Назва параметру	Призначення
1	operators — таблиця даних операторів	id	Унікальний ідентифікатор оператора
		login	Унікальний логін оператора
		password	Пароль оператора, захищений хеш-алгоритмом Pbkdf2PasswordHash
		createdAt	Часова мітка реєстрації оператора
2	drones — таблиця даних дронів	id	Унікальний ідентифікатор дрона
		name	Унікальна кодова назва дрона
		status	Поточний статус дрона, може бути одним з двох значень: inactive — дрон не є активним на жодній з місій; active — дрон безпосередньо приймає участь у виконанні однієї з місій системи
		createdAt	Час реєстрації дрона у системі
3	missions — таблиця даних місій	id	Унікальний ідентифікатор місії
		latitude	Координата широти обраного оператором місця відбору проби
		longitude	Координата довготи обраного місця відбору проби води
		createdAt	Часова мітка реєстрації місії у системі одним з операторів

№ п/п	Таблиця БД	Назва параметру	Призначення
		startedAt	Часова мітка початку виконання місії, вона відповідає часу, коли дрону була відправлена команда на виконання місії разом з координатами місця відбору проби
		probeAt	Часова мітка взяття дроном проби води та отримання сенсорних даних з заданої координати
		finishedAt	Часова мітка повернення дрона до місця старту місії, тобто час завершення місії
4	results — таблиця результатів взятої проби та сенсорних даних дроном під час виконання місії	id	Унікальний ідентифікатор, відповідної місії з якої була взята проба, оскільки результат у місії може бути лише одним, то така прив'язка ідентифікатору не порушує потребу його унікальності для кожного запису
		createdAt	Часова мітка створення результату місії у системі, приблизно відповідає часу коли дрон зняв сенсорні дані із заданої точки та передав їх через мережу
		reportAt	Часова мітка збереження звіту аналізу проби, введеного оператором
		probeReport	Дані лабораторного звіту проби води, введені оператором
		pH	Значення pH зняте із сенсору дроном
		dissolvenOxygen	Значення сенсору розчиненого кисню

№ п/п	Таблиця БД	Назва параметру	Призначення
		temp	Температура водойми під час забору проби води
		turbidity	Значення сенсору мутності води під час забору проби води
		conductivity	Значення електропровідності водойми у місці відбору
5	telemetries — таблиця отриманих з використанням брокеру Kafka телеметричних даних дронів	id	Унікальний ідентифікатор запису
		createdAt	Час збору телеметричних даних дроном, відрізняється від фактичного часу запису даних, оскільки можуть виникати затримки записів через погане з'єднання дрона з мережею
		batteryLevel	Значення батареї дрона у відсотках на момент збору телеметрії
		speed	Швидкість польоту дрона
		thrust	Сумарна тяга двигунів дрона на момент збору телеметричних даних
		latitude	Координата ширини місцезнаходження дрона
		longitude	Координата довготи місцезнаходження дрона
		altitude	Координата висоти дрона
		pitch	Значення осі тангажу
		roll	Значення осі крену
		yaw	Значення осі рискання дрона

Таблиці також з'єднані одна з одною завдяки використанню зовнішніх ключів:

— `missions` має три ключі: оператора `operatorId`, дрона `droneId` та ключ результатів місії `resultId` який може має значення `null`, у випадку якщо сенсорні дані для даної місії ще не були зібрані;

— `telemetries` має два ключі: місії `missionId` та дрона `droneId`, який передає телеметричні дані;

— `results` мають один зовнішній ключ місії `missionId` який і виступає у ролі головного ідентифікатора таблиці.

Проект структурований на чотири репозиторії, використовуючи інтерфейси наведені у таблиці 3.3.

Таблиця 3.3 — Репозиторії та їх функції

№ п/п	Репозиторій / Інтерфейс	Функція	Результат функції	Опис функції
1	Operator Repository / IOperator Repository	<code>findAll()</code>	<code>List<Operator></code>	Повертає список всіх операторів зареєстрованих у підсистемі управління
		<code>findById(Long id)</code>	<code>Optional<Operator></code>	Шукає оператора з відповідним ідентифікатором і повертає опціональне значення із об'єктом оператора або без об'єкту, якщо оператора із

№ п/п	Репозиторій / Інтерфейс	Функція	Результат функції	Опис функції
				запитуваним ідентифікатором не було знайдено
		findByLogin(String login)	Optional<Operator>	Шукає оператора за його логіном, повертає опціональне значення об'єкту оператора
		save(Operator operator)	Operator	Зберігає наданий об'єкт оператора у базу даних та повертає його
2	DroneRepository / IDroneRepository	findAll()	List<Drone>	Повертає список всіх зареєстрованих у системі дронів
		findById(Long id)	Optional<Drone>	Виконує пошук дрона за допомогою ідентифікатора, наданого у функції, повертає опціональне значення із об'єктом дрона або без нього

№ п/п	Репозиторій / Інтерфейс	Функція	Результат функції	Опис функції
		findByName(String name)	Optional<Drone>	Виконує пошук дрона за наданою кодовою назвою БПЛА, повертає опціональне значення
		save(Drone drone)	Drone	Зберігає наданий об'єкт дрона у базу даних, повертаючи об'єкт дрона
3	Telemetry Repository / ITelemetry Repository	findByDroneId(Long droneId)	Telemetry	повертає останній знайдений запис телеметрії дрона використовуючи наданий ідентифікатор і сортуючи записи згідно з колонкою createdAt від найбільшого, тобто найновішого, до найменшого — найстарішого, запису, якщо у дрона немає жодного запису телеметричних

№ п/п	Репозиторій / Інтерфейс	Функція	Результат функції	Опис функції
				даних повертається null
		save(Telemetry telemetry)	Telemetry	Створює новий запис наданого об'єкту телеметричних даних дрона до бази даних та повертає його
4	Mission Repository / IMission Repository	count()	List<Mission>	Повертає кількість записів у таблиці missions
		findById(Long id)	Optional<Mission>	Проводить пошук і повертає опціональне значення об'єкту місії відповідно до наданого ідентифікатора
		findPage(int page, int size)	List<Mission>	Повертає відповідну до запиту сторінку запитуваної кількості елементів, виконує сортування записів відповідно

№ п/п	Репозиторій / Інтерфейс	Функція	Результат функції	Опис функції
				значення <code>finishedAt</code> , тобто якщо місія не є завершеною, вона буде знаходитись у початку видачі, а опісля відповідно до параметру <code>createdAt</code> за принципом DESC, тобто від найновішого до найстарішого запису
		<code>save(Mission mission)</code>	Mission	Зберігає наданий об'єкт місії, разом із вкладеним до нього об'єктом

Репозиторії в свою чергу використовуються у сервісах, які наведено на схемі компонентів у додатку Ж та представлені у таблиці 3.4 — спеціальних класах, які реалізують бізнес-логіку застосунку.

Таблиця 3.4 — Сервіси та їх функції

№ п/п	Сервіс / Інтерфейс	Функція	Результат функції	Опис функції
1	DroneService / IDroneService	<code>getAllDrones()</code>	List<Drone>	Повертає усі зареєстровані в системі

№ п/п	Сервіс / Інтерфейс	Функція	Результат функції	Опис функції
				дрони у вигляді списку
		getOrThrow(Long id)	Drone	Повертає дрон із запитуваним ідентифікатором, якщо дрона із таким ідентифікатором не існує виникає помилка IllegalArgumentException Exception
		setStatus(Long id, String status)	void	Змінює поточний статус дрона відповідно до наданого ідентифікатора і значення статусу, який являє собою строку, цей метод не повертає жодного значення, оскільки статус змінюється на основі отриманих від дрона даних через брокер повідомлень Kafka
		createDrone(Strin g name)	Drone	реєструє й повертає щойно створений запис нового дрону, перевіряє на наявність дублікатів кової назви дрону, відає помилку за наявності

№ п/п	Сервіс / Інтерфейс	Функція	Результат функції	Опис функції
2	Operator Service / IOperator Service	getAllOperators()	List<Operator>	Повертає всіх зареєстрованих операторів у базі даних системі
		getByLoginOr Throw(String login)	Operator	Повертає оператора із відповідним логіном, якщо оператора не знайдено викликає помилку IllegalArgumentException Exception
		createOperator(String login, String password)	Operator	Створює нового оператора у системі, перевіряє на наявність вже зареєстрованого оператора з тим самим логіном і повертає помилку IllegalArgumentException Exception в разі знаходження, хешує пароль користувача перед збереженням об'єкту оператора до бази даних використовуючи Pbkdf2Password Hash
3	Telemetry Service /	findByDroneId(Long droneId)	Optional<Drone>	Повертає опціональне значення об'єкту

№ п/п	Сервіс / Інтерфейс	Функція	Результат функції	Опис функції
	ITelemetry Service	createTelemetry(Long missionId, String droneName, Float batteryLevel, Float altitude, Float speed, Float latitude, Float longitude, Float thrust, Float roll, Float pitch, Float yaw, LocalDateTime timestamp)	Telemetry	телеметрії відповідного дрона Створює новий запис телеметрії до відповідної таблиці, шукає місію та дрона і повертає помилку у випадку якщо не їх знайдено у базі даних, зберігає об'єкт із відповідними даними: рівнем заряду батареї, швидкістю польоту, координатами широти, довготи та висоти, значеннями осей тангажу, крену та ристання і часовою міткою збору телеметрії дроном
4	Mission Service / IMission Service	countAll()	List<Mission>	Повертає кількість місій збережених до бази даних
		getPage(int page, int size)	List<Mission>	Повертає список відповідної сторінки місій із заданою кількістю елементів
		getOrThrow(Long id)	Mission	Повертає відповідну місію до ідентифікатора, видає помилку у випадку якщо

№ п/п	Сервіс / Інтерфейс	Функція	Результат функції	Опис функції
				місії з таким ідентифікатором не існує
		setFinishedAt(Long id, LocalDateTime finishedAt)	Mission	Встановлює часову мітку завершення місії
		setResult(Long id, Float pH, Float DO, Float temp, Float turbidity, Float conductivity)	Mission	Створює новий запис результатів місії із сенсорними значеннями рН, розчиненого кисню, температури, мутності та електро-провідності
		setProbeReport(Long id, String probeReport)	Mission	Оновлює результат місії, додаючи звіт, наданий оператором, до відповідного запису результату
		createMission(String operatorLogin, String droneName, Float latitude, Float longitude)	Mission	Створює новий запис місії у базі даних із відповідним оператором, дроном та заданими координатами відбору проби
		startMission(Long id)	Mission	Розпочинає місію, у випадку якщо розпочати місію неможливо: місія

№ п/п	Сервіс / Інтерфейс	Функція	Результат функції	Опис функції
				вже розпочата або дрон зайнятий іншою місією, повертає помилку, додає відповідну часову мітку до стовпчику startedAt та передає команду на запуск дрона через брокер Kafka

Загалом для опрацювання запитів від операторів, створені HttpServlets які виконують GET та POST запити, описані у таблиці 3.5.

Таблиця 3.5 — Сервлети застосунку та запити до них

№ п/п	Servlet	GET запит	POST запит
1.	DashboardServlet	Повертає сторінку dashboard.jsp із об'єктом оператора який виконує запит	-
2.	DronesAddServlet	повертає сторінку реєстрації нового дрона dashboard_drones_add.jsp	отримує параметр name з відправленої форми та перенаправляє на /dashboard/drones в разі успішного збереження нового запису
3.	DronesServlet	повертає сторінку dashboard_drones.jsp разом із списком	-

№ п/п	Servlet	GET запит	POST запит
		зареєстрованих у системі дронів	
4.	DroneTelemetry Servlet	передає об'єкт дрона та його останній запис телеметрії або null, якщо даних телеметрії не знайдено і повертає сторінку dashboard_drones_telemetry.jsp	-
5.	HomeServlet	передає сторінку home.jsp разом із об'єктом оператора, якщо він авторизований в системі	-
6.	MissionsAddServlet	відкриває dashboard_missions_add.jsp сторінку	приймає параметри drone, lat, lng та бере логін оператора з його сесійних даних для створення нової місії й переадресування до /dashboard/missions при вдалому збереженні нової місії
7.	MissionsServlet	приймає параметр page для отримання списку об'єктів місій для передачі їх до сторінки dashboard_missions.jsp	-

№ п/п	Servlet	GET запит	POST запит
8.	MissionsStartServlet	-	приймає параметр id якій є ідентифікатором місії, оновлює стан місії — додає часову мітку параметру startedAt, та змінює статус дрону на значення active та переадресовує на /dashboard/missions/view?id= із значенням id — ідентифікатором місії
9.	MissionsViewServlet	запитує дані місії використовуючи переданий параметр id, який є ідентифікатором місії, для загрузки сторінки dashboard_missions_view.jsp	-
10.	OperatorsServlet	передає список всіх зареєстрованих в системі операторів і сторінку, яка їх відображає — dashboard_operators.jsp	-
11.	RegisterServlet	передає сторінку register.jsp	приймає параметри login та password для створення нового запису оператора в

№ п/п	Servlet	GET запит	POST запит
			системі, в разі вдалого виконання запиту передає у відповідь сторінку вдалої реєстрації — <code>register_success.jsp</code>
12.	ReportServlet	-	приймає параметри <code>id</code> , що є ідентифікатором місії, і <code>report</code> який є текстовою строкою, що містить наданий оператором звіт результатів дослідження проби, відкриваючи оператору JSP сторінку <code>dashboard_missions_view_result.jsp</code>
13.	ResultServlet	приймає параметр <code>id</code> , що є ідентифікатором місії та передаючи сторінку <code>dashboard_missions_view_result.jsp</code> разом з об'єктом результатів місії	-

Таким чином застосунок має низку endpoints, деякі з яких наведені у схемі послідовностей (додаток И) та у таблиці 3.6 — адрес, до яких оператор звертається, щоб отримати інформацію з бази даних системи шляхом перегляду контенту через JSP сторінки.

Таблиця 3.6 — Endpoints системи менеджменту

№ п/п	Endpoint	Servlet	Призначення
1.	/	HomeServlet	Основний вхід в систему
2.	/register	RegisterServlet	Відтворює сторінку реєстрації
3.	/dashboard	DashboardServlet	Основне меню навігації оператора
4.	/dashboard/operators	OperatorsServlet	Сторінка списку операторів
5.	/dashboard/drones	DronesServlet	Сторінка списку наявних в системі дронів
6.	/dashboard/drones/add	DronesAddServlet	Сторінка створення нового дрона
7.	/dashboard/drones/telemetry?id=	DroneTelemetryServlet	Сторінка останньої відомої телеметрії дрона
8.	/dashboard/missions	MissionsServlet	Сторінка списку місії
9.	/dashboard/missions/add	MissionsAddServlet	Сторінка створення нової місії
10.	/dashboard/missions/start	MissionsStartServlet	команда запуску місії
11.	/dashboard/missions/view	MissionsViewServlet	Сторінка перегляду даних місії
12.	/dashboard/report/add	ReportServlet	Команда введення звіту проби
13.	/dashboard/missions/view/result	ResultServlet	Сторінка перегляду результатів місії
14.	/missions/{missionId}/stream	-	Адреса підключення до Server Side Events

3.3 Тестування функціоналу підсистеми

Згідно додатку К задля розгортання системи спочатку виконується файл `docker-compose.yml` — він розгортає брокер повідомлень Kafka у окремому docker контейнері. По завершенню розгортання контейнеру потрібно виконати команди на створення двох топіків `drone-commands` для передачі команд від серверу до дрону та `drone-telemetry` для отримання телеметричних даних від дронів для запису у базу даних:

— `docker exec -it kafka kafka-topics --create --topic drone-commands --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1;`

— `docker exec -it kafka kafka-topics --create --topic drone-telemetry --bootstrap-server localhost:9092 --partitions 1 --replication-factor 1.`

Перевірити чи топіки були успішно створені можна за допомогою команди наведеної на рисунку 3.3.

```
C:\Users\User>docker exec -it kafka kafka-topics --list --bootstrap-server localhost:9092
__consumer_offsets
drone-commands
drone-telemetry
```

Рисунок 3.3 — Створені топіки у Kafka

Для роботи бази даних локально встановлена PostgreSQL із доступом до БД через порт 5432. Наступним кроком розгортається Jakarta EE, тобто основний додаток, з використанням сервер додатків Eclipse GlassFish. На відміну від простих веб-контейнерів, таких як Apache Tomcat, які вміють обробляти лише сервлети та JSP-сторінки, GlassFish є повноцінним сервером рівня Enterprise. Він забезпечує середовище виконання для всіх компонентів розподіленого додатку, включаючи веб-клієнти, бізнес-компоненти та конектори до корпоративних інформаційних систем. В реалізованій системі Glassfish керує життєвим циклом компонентів:

— `Web container` обробляє HTTP-запити від браузера оператора, керує сесіями та виконує рендеринг JSP-сторінок;

— EJB container керує бізнес-логікою, забезпечуючи транзакційність, багатопотоковість та безпеку доступу;

— JPA provider забезпечує ORM-відображення Java-об'єктів на таблиці бази даних PostgreSQL.

В свою чергу у GlassFish налаштовується зв'язок із базою даних шляхом відкриття спеціальної сторінки налаштувань, розгорнутої за замовчуванням за шляхом <http://localhost:4848/> де створюється JDBC Connection Pool (рисунок 3.4), в який передаються параметри бази даних.

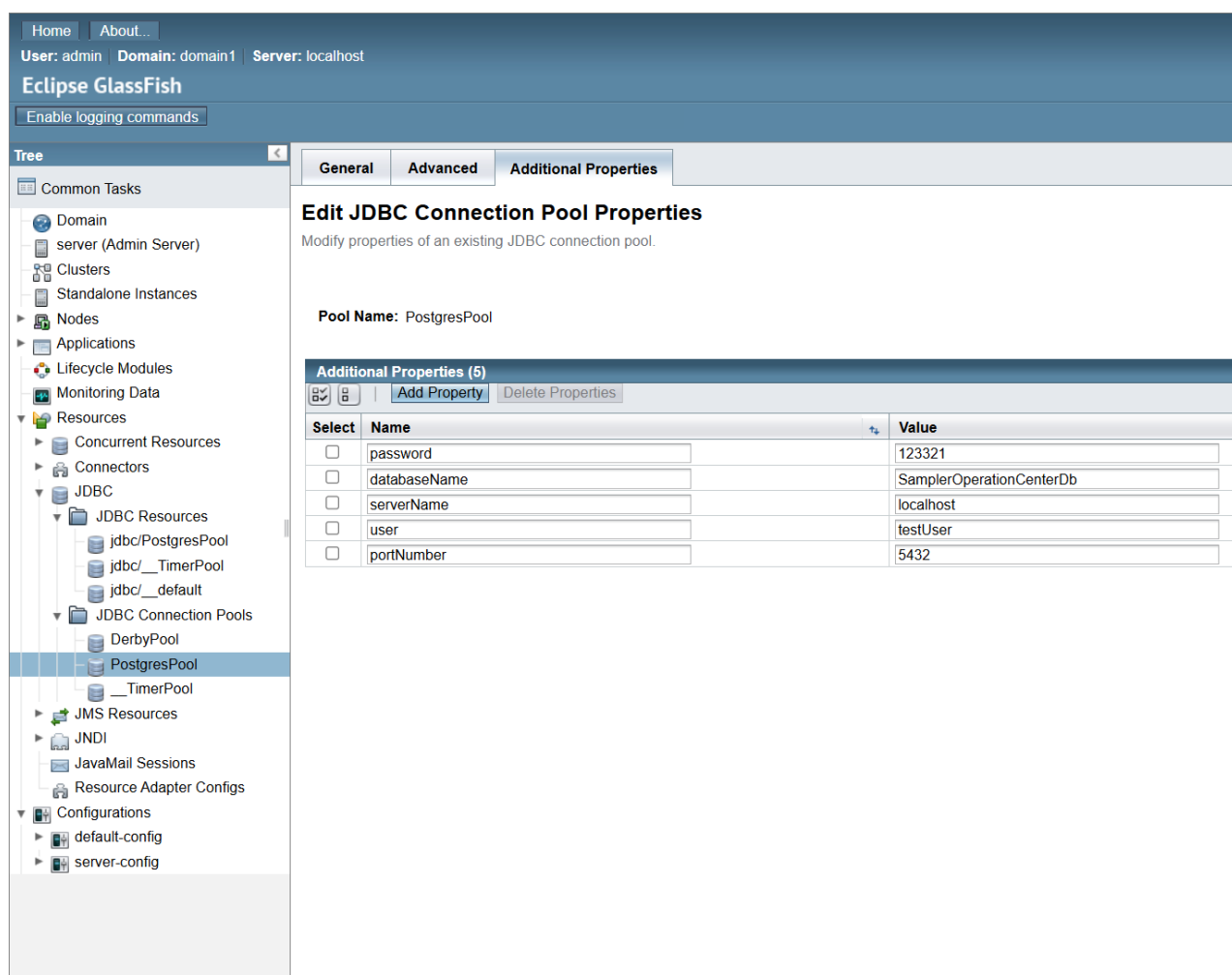


Рисунок 3.4 — Сторінка налаштувань JDBC Connection Pool серверу додатків GlassFish

Також створюється новий JDBC Resource (рисунок 3.5), до якого вже сконфігуроване підключення у файлі persistence.xml (рисунок 3.6).

Задля проведення наглядного тесту системи запускається окремий Java файл, який виступає в ролі моделі поведінки дрона, що генерує схожі до реальних дані, отримує команди та відправляє телеметрію через брокер Kafka.

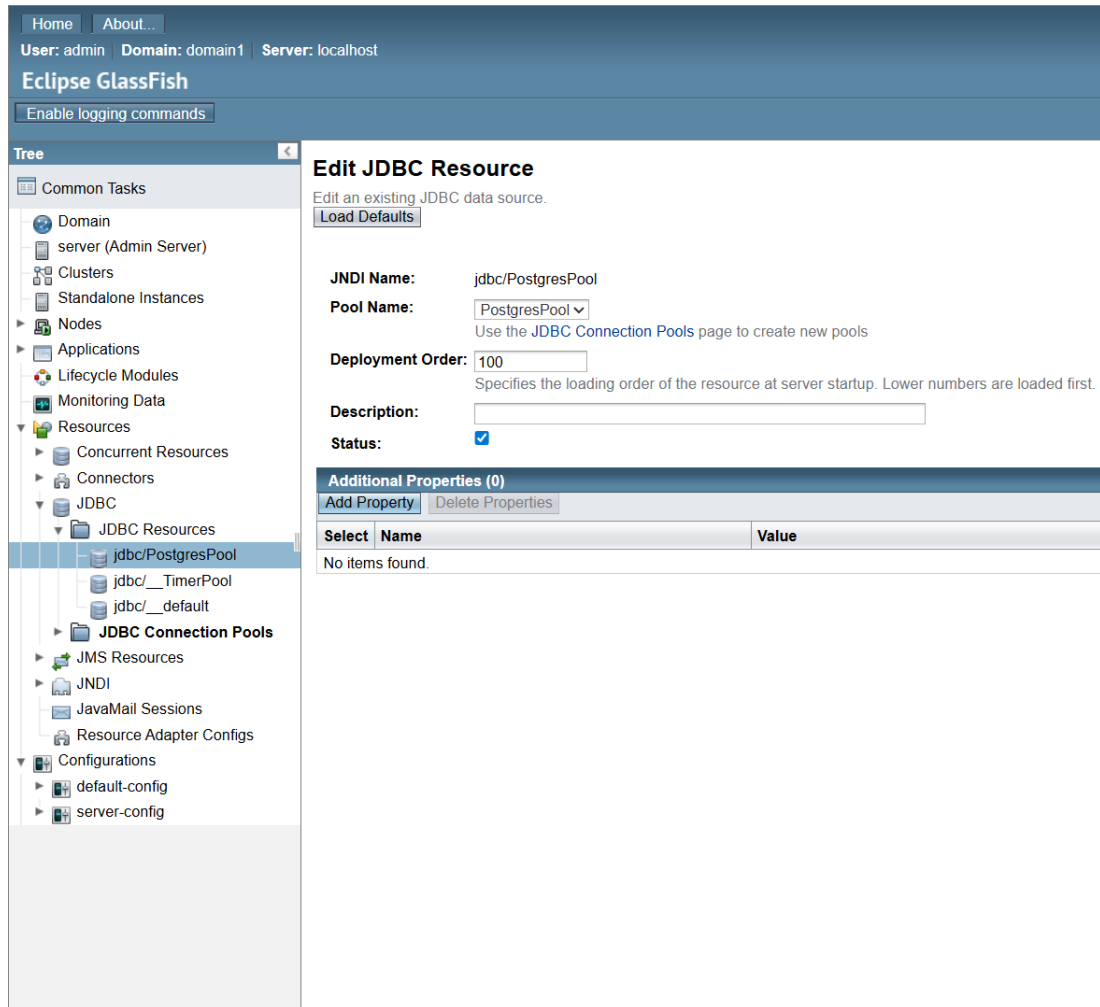


Рисунок 3.5 — Сторінка налаштувань JDBC Resource

```
<persistence xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:persistence="https://jta
  <persistence-unit name="samplerOperationCenterPersistence" transaction-type="JTA">
    <jta-data-source>jdbc/PostgresPool</jta-data-source>
  </persistence-unit>
</persistence>
```

Рисунок 3.6 — Файл конфігурації підключення до бази даних

Після розгортання додатку користувач, відкривши веб сайт вперше, побачить головку сторінку — home.jsp, із кнопками: реєстрації та авторизації. По

завершенню реєстрації, оператор може авторизуватись в системі, використовуючи спеціальні поля вводу у вікні авторизації.

Таким чином, авторизувавшись у системі управління, оператор має дві основні сторінки, доступні через сторінку dashboard (рисунок 3.7): перегляд місій та перегляд дронів. Для виконання запитів задіяні одразу декілька технологій Jakarta EE: JPA для зв'язку з базою даних, EJB для використання сервісів, та Servlets для обробки GET та POST запитів.

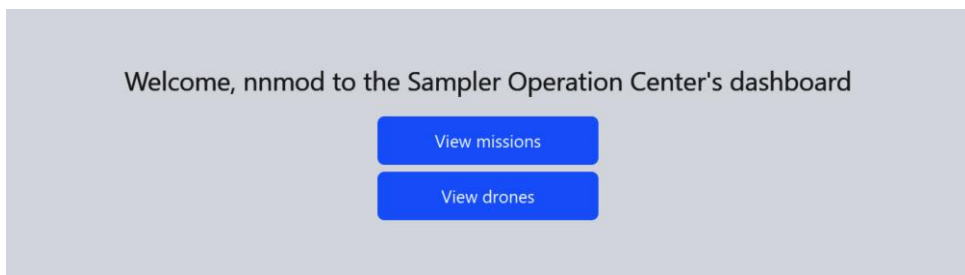


Рисунок 3.7 — Сторінка основного меню системи

Оператор має доступ до додавання нових дронів у додатку (рисунок 3.8), таким чином вони одразу будуть доступні у системі для використання їх у нових місіях. Під час створення нової місії оператор бачить онлайн мапу від Open Street Мар з використанням технологій Leaflet. для зручного пошуку та відображення заданого місця відбору, таким чином оператор безпосередньо взаємодіє з мапою і не потребує вводу координат напряду (рисунок 3.9).

Після початку місії Jakarta EE відправляє команду на запуск дрона разом із даними місії використовуючи топик `drone-commands` (рисунок 3.10).

Створивши нову місію вона стає доступною до перегляду у списку місій (рисунок 3.11) разом з іншими місіями, відображаючи статус «waiting to start», тобто стає в очікування команди на початок виконання місії.

Отримавши відповідну команду спеціальний Java скрипт опрацьовує її і починає імітувати поведінку дрона відправляючи пакети телеметрії, наведені на рисунку 3.12, кожну секунду.

Dashboard		List of operators		Add Drone
Drone Id	Drone Name	Drone Status	Drone Added DateTime	Actions
<u>3</u>	<u>razor</u>	<u>inactive</u>	<u>2025-12-10T20:04:06.985569</u>	View Telemetry
<u>4</u>	<u>spectrum</u>	<u>inactive</u>	<u>2025-12-11T13:01:14.786511</u>	View Telemetry
<u>1</u>	<u>fly</u>	<u>inactive</u>	<u>2025-12-10T17:44:19.620856</u>	View Telemetry
<u>2</u>	<u>cobalt</u>	<u>inactive</u>	<u>2025-12-10T17:46:20.538989</u>	View Telemetry

Рисунок 3.8 — Сторінка списку дронів, зареєстрованих у системі

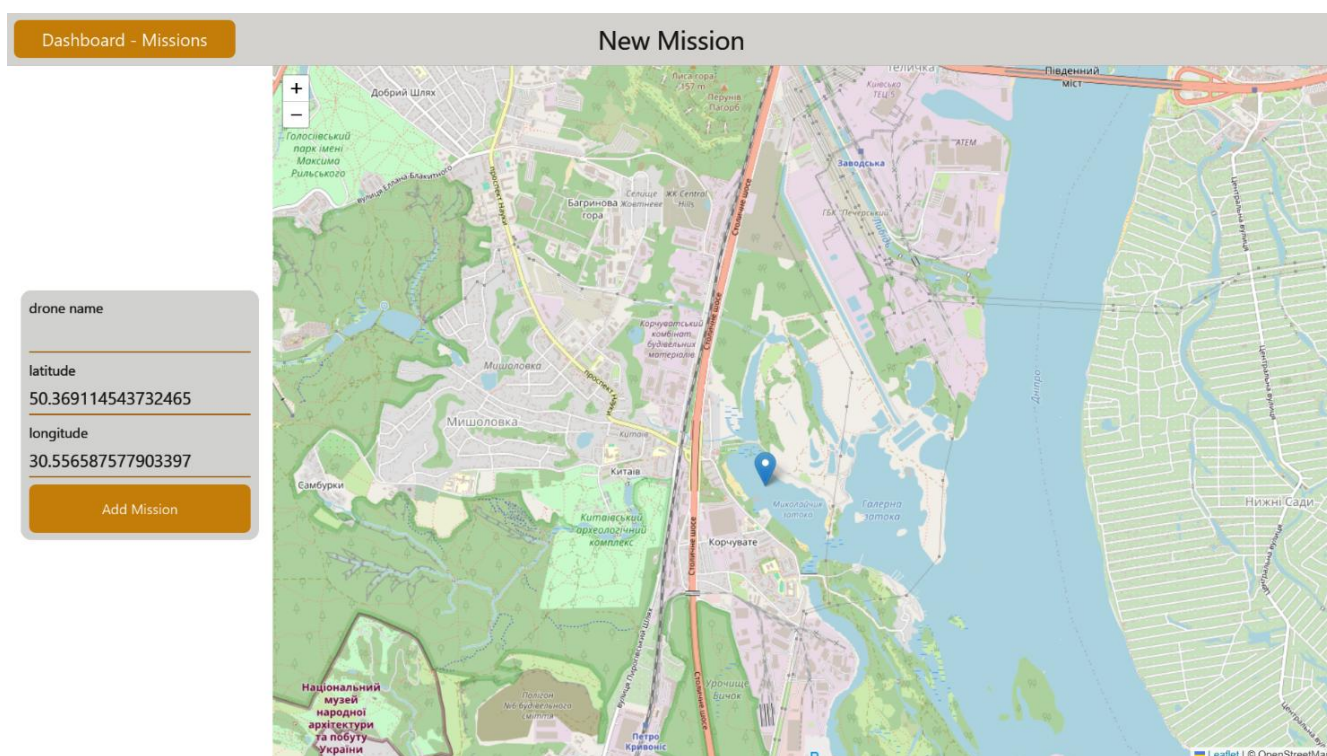


Рисунок 3.9 — Сторінка створення нової місії

Отримуючи пакети даних Jakarta EE зберігає їх у базі даних та відправляє до SSE хабу, який розсилає їх операторам, які мають відкриту сторінку перегляду місії, завдяки чому оператор має можливість переглядати шлях дрону та його позицію у реальному часі що проілюстровано на рисунку 3.13.

По завершенню виконання місії оператор має доступ до перегляду результатів місії: сенсорних даних та має можливість ввести відповідний звіт проаналізованої проби води у лабораторії (рисунок 3.14).

```

1 {
2   "missionId": "12",
3   "drone": "cobalt",
4   "lat": 50.481461,
5   "lng": 30.504917
6 }

```

Рисунок 3.10 — Вигляд пакетів даних при отриманні команди від Kafka

Dashboard		List of missions			Add Mission
Mission Id	Operator Login	Drone Name	Mission Status	Mission Created	Actions
12	nnmod	cobalt	flying to the position	2025-12-17T14:45:06.282126300	View
10	nnmod	razor	waiting to start	2025-12-11T12:19:00.473867	View
11	nnmod	cobalt	finished	2025-12-11T12:19:29.510623	View
9	nnmod	fly	finished	2025-12-11T12:18:47.790474	View
8	nnmod	razor	finished	2025-12-11T12:16:39.961676	View
7	nnmod	fly	finished	2025-12-11T12:16:14.554558	View
6	nnmod	razor	finished	2025-12-11T11:14:43.235645	View
5	nnmod	fly	finished	2025-12-11T11:05:05.809538	View
4	nnmod	razor	finished	2025-12-11T10:56:52.204012	View

< 1 >

Рисунок 3.11 — Відображення списку місій з використанням пагінації

```

1 {
2   "missionId": "12",
3   "drone": "cobalt",
4   "latitude": 50.49295320542213,
5   "longitude": 30.49306747810126,
6   "altitude": 0,
7   "speed": 0,
8   "roll": 0,
9   "pitch": 0,
10  "yaw": -0.618174368058245,
11  "thrust": 0,
12  "batteryLevel": 87.97043707044617,
13  "timestamp": "2025-12-17T14:51:49.5098955",
14  "probe": {
15    "pH": 7.589901274635779,
16    "DO": 10.119843978567271,
17    "temp": 12.373023919684506,
18    "turbidity": 2.5064854868366893,
19    "conductivity": 105.01796921740076
20  },
21  "isLanded": true
22 }

```

Рисунок 3.12 — Приклад вигляду пакетів телеметрії дрона

Dashboard - Missions
Mission View

Mission Data

id: 12
operator: nmod
drone: cobalt
status: flying to the position
created: 2025-12-17T14:45:06.282126300
started: 2025-12-17T14:45:11.515356100
probe:
finished:

-0.007418301, yaw: 2.5967097
[2025-12-17T14:46:18.389956] battery:
90.0086, speed: 10, thrust: 0.5537498, lat:
50.492355, lng: 30.493696, alt:
4.9762354, roll: -0.06630754, pitch:
-0.017871384, yaw: 2.566021
[2025-12-17T14:46:19.3922933] battery:
90.0036, speed: 10, thrust: 0.6221744, lat:
50.49228, lng: 30.493769, alt: 4.675566,
roll: -0.06308103, pitch: -0.035171673,
yaw: 2.5999265
[2025-12-17T14:46:20.3950424] battery:
89.9986, speed: 10, thrust: 0.5509217, lat:
50.492207, lng: 30.493856, alt:
5.0970597, roll: -0.032518536, pitch:
-0.002753601, yaw: 2.4667068
[2025-12-17T14:46:21.4003191] battery:
89.9936, speed: 10, thrust: 0.60933745,
lat: 50.492134, lng: 30.493937, alt:
5.023672, roll: -0.0047194078, pitch:
-0.06343591, yaw: 2.531802

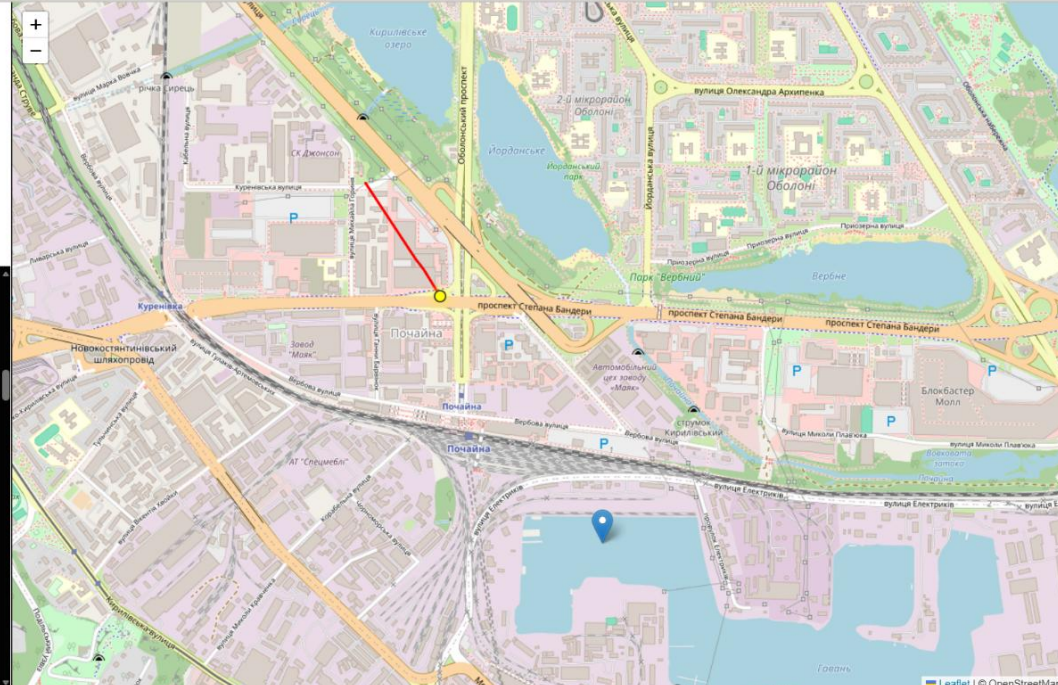


Рисунок 3.13 — Вигляд сторінки перегляду місії під час виконання її дроном

Back to Mission
Mission Result

pH
7.5899014
Dissolved Oxygen
10.119844
Temperature
12.373024
Turbidity
2.5064855
Conductivity
105.01797
report
Add Report

Рисунок 3.14 — сторінка результатів місії

Висновки до розділу 3

Проведений порівняльний аналіз сучасних технологічних рішень, зокрема Spring Boot, Django та Node.js, визначив платформу Jakarta EE як найбільш

збалансований варіант для побудови серверної частини. Цей вибір базується на необхідності дотримання промислових стандартів надійності та ефективного управління системними ресурсами, де Jakarta EE переважає завдяки підтримці специфікацій EJB для бізнес-логіки, JPA для роботи з даними та JAX-RS для веб-сервісів. Платформа демонструє відмінну роботу з багатопотоковістю, потрібну для одночасної обробки даних від багатьох БПЛА, уникаючи при цьому надлишкового споживання оперативної пам'яті.

Архітектура системи побудована за монолітною моделлю, яка є виправданою для локалізованого комплексу з прогнозованим навантаженням та визначеним колом користувачів. Такий підхід оптимізує цикл розроблення програмного забезпечення, спрощуючи процеси налагодження, інтеграційного тестування та розгортання єдиного артефакту додатку. Монолітна архітектура гарантує максимальну швидкість обміну даними між компонентами через виклики в пам'яті та забезпечує транзакційну цілісність, повністю нівелюючи ризики мережеских затримок та проблем узгодженості даних, що часто виникають у розподілених мікросервісних середовищах.

Задачу збереження та обробки різномірних даних вирішує об'єктно-реляційна система управління базами даних PostgreSQL. Її функціональні можливості дозволяють поєднати зберігання реляційної інформації про місії та операторів із масивами часових рядів телеметрії та геопросторовими даними.

Інтеграція розподіленого брокера повідомлень Apache Kafka розв'язує проблему стабільної передачі поточкових даних телеметрії в умовах низької пропускної здатності мобільного зв'язку. Реалізований механізм асинхронного обміну даними за принципом fire-and-forget дозволяє бортовому комп'ютеру дрона відправляти пакети даних, не витрачаючи ресурси на очікування відповіді сервера. Завдяки буферизації повідомлень Kafka гарантує доставку та збереження хронології телеметрії навіть у випадках тимчасових розривів з'єднання, забезпечуючи нульову втрату важливих вимірювань під час польоту.

Результатом роботи є створення повнофункціональної підсистеми управління оператором, що надає веб інтерфейс на базі JSP для керування повним

життєвим циклом місії: від реєстрації апаратів та планування точок відбору проб до візуалізації треку польоту в реальному часі. Реалізована багаторівнева система безпеки з криптографічним хешуванням паролів хеш-алгоритмом Pbkdf2PasswordHash, яка захищає адміністративні функції, а сервер додатків Eclipse GlassFish створює стабільну платформу для відтворення та підтримки системи управління оператора. Використання брокера Apache Kafka, бази даних PostgreSQL та технології Server Side Events у поєднанні з JakartaEE дозволило реалізувати масштабовану архітектуру, здатну підтримувати одночасний моніторинг декількох БПЛА, забезпечуючи індивідуальне відображення даних місій у реальному часі для кожного оператора без конфліктів та переривань.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

В умовах стрімкого розвитку технологій та зростаючих екологічних вимог, створення інноваційних рішень для моніторингу водних ресурсів є актуальним завданням. Метою даного розділу є проведення маркетингового аналізу та розроблення стратегії виведення на ринок безпілотної системи для автоматизованого відбору проб води. Основні завдання включають аналіз ринкового середовища, оцінку конкурентоспроможності розробки, визначення бізнес-моделі та планування маркетингових заходів.

4.1 Опис ідеї проєкту

На першому етапі проводиться аналіз змісту ідеї, можливих напрямків її застосування та основних вигод для користувача. Проєкт передбачає створення програмно-апаратного комплексу на базі гексакоптера, обладнаного системою відбору проб за технологією твердофазної екстракції (SPE) та набором гідрохімічних сенсорів.

У таблиці 4.1 наведений опис ідеї стартап-проєкту, який дозволяє зрозуміти загальну концепцію товару.

Таблиця 4.1 — Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення безпілотної літального апарату з інтегрованим модулем SPE-екстракції та системою передачі телеметрії в	1. Екологічний моніторинг (Державні екоінспекції).	Безпека персоналу: відсутність ризику для життя та здоров'я екологів при роботі на забруднених або небезпечних водоймах.
	2. Контроль показників води в промисловому рибництві (Аквакультура).	Економічна вигода: значне зниження витрат на логістику (паливо для човнів, час

Зміст ідеї	Напрямки застосування	Вигоди для користувача
реальному часі через 4G/Kafka.		персоналу) порівняно з традиційними методами.
	3. Наукові дослідження гідросфери та моніторинг промислових стоків.	Якість даних: забезпечення стерильності проби завдяки автоматизації та отримання аналітики в реальному часі.

Для визначення техніко-економічних переваг ідеї порівняно з конкурентами:

— ручний метод із використанням човна: традиційний спосіб, що вимагає залучення людей та плавзасобів;

— DJI Matrice 350 RTK: промисловий дрон універсального призначення;

— SeaRobotics SR Utility: автономне надводне судно.

Таблиця 4.2 — Визначення сильних, слабких та нейтральних характеристик ідеї проекту

№ п/п	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проект	Ручний відбір з човна	DJI Matrice 350 RTK	Sea Robotics SR Utility			
1	Вартість комплексу	низька собівартість компонентів	Середня вартість човна та палива	висока вартість бренду	дуже висока вартість промислового АПС	-	-	+
2	Автономність (час роботи)	обмежена батареєю до 40 хв	Обмежений лише паливом	до 55 хв	працює годинами та днями	+	-	-

№ п/ п	Техніко- економічні характери- стики ідеї	(потенційні) товари/концепції конкурентів				W	N	S
		Мій проект	Ручний відбір з човна	DJI Matrice 350 RTK	Sea Robotics SR Utility			
3	Спеціалізація під відбір проб	Інтегро- ваний SPE модуль	за наявно- сті обла- днання	потребує сторо- нного навісного обла- днання	спеціа- лізоване судно	-	-	+
4	Мобільність розгортання	Запуск з берега чи недалеко від місця відбору	необхі- дність спуску човна на воду	швидкий запуск	потребує спуску на воду, повільне розго- ртання	-	+	-
5	Програмна інтеграція	власне ПЗ з базою даних та аналіти- кою	відсутня	закрите ПЗ виробни- ка	складне профе- сійне ПЗ	-	-	+

4.2 Технологічний аудит ідеї проекту

Важливим етапом є перевірка технологічної здійсненності ідеї. Необхідно впевнитись, що всі складові проекту можуть бути реалізовані наявними ресурсами, наведеними у таблиці 4.3.

Таблиця 4.3 — Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
		Технологія 1 (технологія виготовлення товару, надання послуги)	Чи вони наявні, або ж необхідно їх розробити/додати?	Чи вони доступні авторам проекту?
1	Платформа-носій	Мультироторна система (рама F550, Pixhawk Cube Orange)	Так, наявні на ринку компоненти	Так, вільно купуються
2	Система відбору проб	Перистальтичний насос NEMA 17, SPE-картриджі	Так, технологія відома в лабораторній практиці	Так, потребує конструкторської адаптації
3	Передача та обробка даних	4G модеми, протоколи MAVLink, Kafka, Jakarta EE	Так, стандартизовані IT-рішення	Так, Open-Source або ліцензії доступні

4.3 Аналіз ринкових можливостей запуску стартап-проекту

Для успішного запуску проаналізоване ринкове середовище у таблиці 4.4. Ринок екологічного моніторингу характеризується стабільним зростанням попиту на автоматизацію.

Визначення портрета потенційного клієнта у таблиці 4.5 дозволяє адаптувати продукт під конкретні потреби.

Таблиця 4.4 — Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	Не застосовується
3	Динаміка ринку	Зростає через вимоги екологічних стандартів
4	Наявність обмежень для входу	Необхідність сертифікації, дозволи на польоти
5	Специфічні вимоги до стандартизації та сертифікації	Високі вимоги до точності вимірювань та захисту даних
6	Середня норма рентабельності в галузі %	25-30%

Таблиця 4.5 — Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Виконання державних програм моніторингу води	Державні екологічні інспекції	Бюрократизовані закупівлі, орієнтація на стандарти	Відповідність ДСТУ або ISO, наявність сертифікатів, офіційна звітність
2	Контроль умов	Приватні рибні господарства	Раціональність витрат, потреба в оперативності	Простота керування, надійність 24/7,

№ п/п	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	розведення риби			низька ціна обслуговування
3	Проведення наукових досліджень	НДІ, Університети	Потреба в «сирих» даних та гнучкості	Можливість зміни сенсорів, доступ до API, експорт даних

Аналіз факторів загроз наведених у таблиці 4.6 та факторів можливостей із таблиці 4.7, що впливають на проєкт, дозволяє передбачити ризики та використати сприятливі обставини.

Таблиця 4.6 — Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1	Законодавче регулювання	Заборона польотів БПЛА у певних зонах	Отримання спеціальних дозволів, розроблення режимів польоту на низькій висоті
2	Погодні умови	Неможливість роботи при сильному вітрі	Розроблення захищеного корпусу, покращення системи стабілізації
3	Технологічні ризики	Втрата зв'язку або аварія над водою	Впровадження аварійних режимів, використання плавучих елементів

Таблиця 4.7 — Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1	Розвиток компонентної бази	Здешевлення сенсорів та акумуляторів	Зниження собівартості продукту, підвищення автономності
2	Цифровізація галузі	Зростання попиту на IoT рішення	Інтеграція з існуючими ERP-системами
3	Грантова підтримка	Доступність грантів для еко-стартапів	Подача заявок на грантове фінансування для масштабування

Аналіз конкурентного середовища для розуміння умов ринку наведений у таблиці 4.8.

Таблиця 4.8 — Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Тип конкуренції: олігополія	Кілька великих гравців	Необхідність пошуку унікальної ніші
2. За рівнем конкурентної боротьби: національний	Невелика кількість представлених рішень на національному рівні	Потреба у побудові сильної репутації на локальному ринку
3. За галузевою ознакою: міжгалузева	Конкуренція з традиційними методами відбору	Пошук та використання інноваційних рішень

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
5. За характером конкурентних переваг: нецінова	Розроблення нових методів відбору проб води	Акцент на інноваційності SPE-технології та зручності ПЗ
6. За інтенсивністю: не марочна	Представлення продукту без відомого бренду	Можливість швидкого захоплення частки ринку

Аналіз конкуренції в галузі неведений у таблиці 4.9.

Таблиця 4.9 — Аналіз конкуренції в галузі за М. Портером

	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
Складові аналізу	Виробники водних дронів (Sea Robotics) та інтегратори DJI	ІТ-компанії, що розробляють агро-рішення	Виробники сенсорів (Atlas Scientific) та електроніки	Державні служби, агрохолдинги	Ручний відбір проб лабораторіями
Висновки	Конкуренція середня. Прямі конкуренти дорогі	Бар'єри входу середні, потрібна експертиза	Постачальники мають вплив через унікальність сенсорів	Клієнти мають високу ринкову силу через тендери	Замінники дешевші на старті, але дорожчі в експлуатації

На основі аналізу конкуренції обґрунтовані ключові фактори конкурентоспроможності наведені у таблиці 4.10.

Таблиця 4.10 — Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1	Вартість рішення	Критичний фактор для українського ринку в умовах обмежених бюджетів
2	Автоматизація обробки даних	Клієнт хоче отримати готовий звіт у базі даних
3	Безпека експлуатації	Мінімізація ризиків втрати дорогого обладнання та безпека персоналу
4	Мобільність	Можливість обслуговування кількох віддалених водойм за один день

У таблиці 4.11 наведена бальна оцінка сильних та слабких сторін проекту відносно конкурентів.

Таблиця 4.11 — Порівняльний аналіз сильних та слабких сторін системи

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з автоматичною системою відбору води						
			-3	-2	-1	0	+1	+2	+3
1	Вартість придбання	20	SR Utility	DJ 350					Ручний відбір
2	Операційна вартість	20		SR Utility , DJ 350					Ручний відбір

№ п/п	Фактор конку- нтоспро- можності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з автоматичною системою відбору води							
			-3	-2	-1	0	+1	+2	+3	
3	Автоно- мність	15	Ручний відбір					DJI 350		SR Utility
4	Автома- тизація даних	20	Ручний відбір				DJI 350, SR Utility			
5	Мобі- льність	20	Ручний відбір			SR Utility	DJI 350			

Підсумком аналізу є SWOT-матриця наведена у таблиці 4.12, що узагальнює стратегічне положення проєкту.

Таблиця 4.12 — SWOT- аналіз стартап-проєкту

<p>Сильні сторони:</p> <p>Низька собівартість та експлуатаційні витрати.</p> <p>Інтегрована інформаційна система (Jakarta EE).</p> <p>Використання інноваційної технології SPE</p>	<p>Слабкі сторони:</p> <p>Обмежений час польоту (залежність від батарей).</p> <p>Залежність від покриття 4G мережі.</p> <p>Обмежена вантажопідйомність.</p>
<p>Можливості:</p> <p>Вихід на ринки агромоніторингу та «розумного міста».</p> <p>Продаж ліцензій на ПЗ (модель SaaS).</p>	<p>Загрози:</p> <p>Посилення регуляції використання повітряного простору.</p> <p>Поява дешевих аналогів з Китаю.</p> <p>Зростання цін на комплектуючі через валютні коливання.</p>

Масштабування на інші платформи (човни).	
---	--

На основі SWOT-аналізу розроблені альтернативи ринкової поведінки представлені у таблиці 4.13.

Таблиця 4.13 — Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива ринкової поведінки	Ймовірність отримання ресурсів	Терміни реалізації
1	Створення власного стартапу	Середня: потрібні інвестиції в «залізо»	6-12 місяців
2	Продаж технології виробникам дронів	Низька: складно захистити інтелектуальну власність	3-6 місяців
3	Надання послуг моніторингу	Висока: потрібен мінімальний парк дронів	1-3 місяці

Обрана альтернатива: надання послуг моніторингу.

4.4 Розроблення ринкової стратегії проєкту

У таблиці 4.14 наведений вибір цільових груп потенційних споживачів системи безпілотного відбору проб води.

Визначена базова стратегія розвитку стартап-проєкту у таблиці 4.15.

У таблиці 4.16 наведено визначення базової стратегії конкурентної поведінки на ринку.

Таблиця 4.14 — Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Приватні рибні господарства	Висока	Середній	Середня	Висока

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
2	Екологічні інспекції	Середня	Високий	Низька	Низька
3	Наукові установи	Висока	Низький	Низька	Середня
Які цільові групи обрано: рибні господарства та екологічні інспекції					

Таблиця 4.15 — Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1	Впровадження власного виробництва та сервісу	Стратегія концентрованого маркетингу	Комплексність, автоматизація, ціна.	Стратегія диференціації

Стратегія позиціонування товару наведена у таблиці 4.17.

Таблиця 4.16 — Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1	Так, у ніші бюджетних БПЛА	Залучення нових споживачів, що	Ні, розробка власної	Стратегія нішера, що

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
	з SPE-модулем в Україні	раніше не використовували автоматизацію	унікальної архітектури	унікає прямої конкуренції з гігантами

Таблиця 4.17 — Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап- проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
1	Точність, безпека, простота	Диференціація	Повна автоматизація циклу «відбір-звіт», відсутність контакту людини з водою	«Лабораторія, що літає». «Безпечний еколог». «Розумний моніторинг води».

4.5 Розроблення маркетингової програми стартап-проєкту

Маркетингова програма деталізує інструменти впливу на ринок, ключові переваги наведені у таблиці 4.18.

Таблиця 4.18 — Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1	Швидкий аналіз якості води	Отримання сенсорних результатів в реальному часі на планшет	Інтеграція з хмарною базою даних та автоматична побудова графіків
2	Відбір проб у важкодоступних місцях	Можливість відбору проб без човна та чобіт	Використання БПЛА з вертикальним зльотом
3	Зменшення ваги проб для транспортування	Використання SPE-картриджів	Унікальна технологія попередньої підготовки проби на борту дрона

Трирівнева модель товару, наведена у таблиці 4.19, допомагає структурувати пропозицію. Цінова політика неведена у таблиці 4.20.

Таблиця 4.19 — Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Отримання достовірної інформації про екологічний стан водойми без ризику для життя та здоров'я людини		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Гексакоптер F550 з захищеною електронікою	-/+	+//+//+//+

Рівні товару	Сутність та складові		
	Модуль автоматизованого відбору проб (насос, клапани, SPE) Веб-інтерфейс оператора на базі Jakarta EE		
	Якість: постійне та інтегроване в розробку тестування, підтримка зворотного зв'язку з клієнтами, відповідність IP54, точність позиціювання RTK		
III. Товар із підкріпленням	До продажу Навчання персоналу замовника. Хмарне зберігання історії всіх місій та звітів		
	Після продажу Технічна підтримка 24/7 та гарантійний ремонт Оновлення прошивки «по повітрю» (OTA)		
За рахунок чого потенційний товар буде захищено від копіювання: програмний код серверної частини, унікальна конструкція вузла відбору			

Таблиця 4.20 — Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	Низький	Дуже високий	Середній (бюджетні установи) та Високий (агрохолдинги)	Нижня: Собівартість + 30% Верхня: 50% від вартості ASV

Таблиця 4.21 — Формування системи збуту

№ п/п	Специфіка поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1	Потребують консультацій та демонстрації роботи	Презентація, навчання, пуско-налагодження, сервіс	Канал нульового рівня	Власний відділ продажів, участь у тендерах

Вибір каналів збуту представлений у таблиці 4.21 та є критичним для доставки цінності клієнту.

Для стартап-проєкту розроблена концепція маркетингових комунікацій, наведена у таблиці 4.22.

Таблиця 4.22 — Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Довіряють технічним характеристикам та тестам	1. Профільні виставки 2. SEO-просування сайту 3. Комерційні пропозиції	Інноваційність, Економія, Безпека	Інформування про існування рішення, формування довіри	«Припиніть ризикувати людьми. Довірте моніторинг безпілотним технологіям»

Висновки до розділу 4

Проведений маркетинговий аналіз підтвердив високий потенціал для комерціалізації розробленого продукту. Ринок екологічного моніторингу та агротехнологій демонструє стійку тенденцію до зростання, що зумовлено посиленням регуляторних вимог до якості водних ресурсів та загальним трендом на цифровізацію процесів. Проєкт успішно займає вільну ринкову нішу між двома рішеннями: дорогими та логістично складними промисловими водними дронами та дешевим, але неефективним і небезпечним ручним відбором проб. Для виходу на ринок обрано стратегію нішевої спеціалізації. Компанія фокусуватиметься на задоволенні специфічних потреб клієнтів, для яких важливою є швидкість розгортання та безпека персоналу. В якості моделі збуту обраний прямий канал продажів, що включає власну систему дистрибуції та участь у державних тендерних закупівлях. Це дозволить забезпечити повний контроль якості комунікації зі споживачем, надавати кваліфіковану технічну підтримку та забезпечити вищу норму рентабельності шляхом виключення посередників.

Таким чином, подальша імплементація проєкту є доцільною. Рекомендовано зосередити зусилля на сертифікації вимірювального обладнання для відповідності державним стандартам та проведенні пілотних впроваджень на базі партнерських рибницьких господарств для формування репутації на ринку.

ВИСНОВКИ

Проведений аналіз існуючих методів контролю якості води та доведена неефективність традиційних ручних заборів. Зокрема, через ризики для життя та здоров'я персоналу, особливо при роботі у важкодоступних, заболочених або токсично забруднених зонах. Аналіз ринку комерційних рішень довів, що існуючі автоматизовані системи, такі як автономні поверхневі судна та підводні апарати, мають високу вартість, складні в експлуатації та потребують спеціалізованої інфраструктури для спуску на воду, що робить їх недоступними для споживачів. Враховуючи це, обґрунтована доцільність створення спеціалізованої безпілотної системи на базі літального апарату (БПЛА), яка поєднує високу мобільність, швидкість розгортання та точність лабораторного обладнання.

В якості носія обрана платформа гексакоптера на базі рами F550. Вибір шестироторної схеми забезпечив необхідну надлишковість тяги та надійність: у разі відмови одного з двигунів апарат здатен зберегти керування та здійснити аварійну посадку, що є надважливим фактором при польоті над водними об'єктами. Використання польотного контролера Pixhawk Cube Orange+ у поєднанні з GNSS-модулем Here 3+ та технологією RTK дозволяє досягати точності позиціонування, що покращує точність місця відбору проб води у заданих координатах.

Основою апаратної частини механізму відбору проб води є впровадження технології твердофазної екстракції. Замість транспортування великих об'ємів води, які зменшують дальність польоту та потребують дорожчих моделей БПЛА, розроблена система прокачує воду через спеціалізований картридж із сорбентом безпосередньо на місці відбору. Зниження злітної маси суттєво розвантажує платформу та збільшує ефективний час польоту дрона. Для реалізації процесу відбору використано прецизійний перистальтичний насос на базі крокового двигуна, що дозволяє точно дозувати об'єм прокаченої рідини.

Комплекс оснащений набором високоточних датчиків для моніторингу параметрів води в режимі реального часу: рН, розчинений кисень, електропровідність, температура та мутність. При проектуванні електронної схеми

вирішена задача із забезпечення електромагнітної сумісності. Оскільки спільна система живлення силових агрегатів дрона та чутливих сенсорів створювала значні електричні шуми, застосована гальванічна розв'язка сигнальних ліній та живлення. Це дозволяє отримати стабільні та достовірні показники вимірювань, безпосередньо під час польоту.

Інженерна значущість роботи полягає в реалізації надійної архітектури взаємодії оператора із парком дронів. На основі порівняльного аналізу доступних фреймворків для написання системи управління оператором обґрунтований вибір платформи Jakarta EE для реалізації серверної частини, забезпечуючи дотримання промислових стандартів надійності, транзакційної цілісності даних (ACID) та ефективного управління потоками завдяки використанню компонентів EJB та JPA. Монолітна архітектура системи є виправданою для даного класу задач, дозволяючи спростити розгортання та забезпечити високу швидкість обміну даними всередині системи.

Для зберігання даних використана об'єктно-реляційну СУБД PostgreSQL, яка забезпечує ефективну роботу як зі структурованою інформацією про місії та операторів, так і з великими масивами часових рядів телеметрії та геопросторовими даними.

Для вирішення задачі використання дронів для моніторингу із нестабільністю мобільного зв'язку в архітектуру системи впроваджений розподілений брокер повідомлень Apache Kafka. Реалізований асинхронний механізм обміну даними за принципом «fire-and-forget» дозволяє бортовому комп'ютеру дрона відправляти пакети телеметрії, не витрачаючи обчислювальні ресурси на очікування підтвердження від сервера. Завдяки буферизації повідомлень у Kafka гарантується повне збереження хронології польоту та вимірювань навіть у випадках тимчасових розривів з'єднання. Це забезпечує нульову втрату критично важливих екологічних даних.

Розроблений повнофункціональний веб інтерфейс, який надає оператору інструменти для повного циклу керування місією. Реалізований функціонал планування маршруту на карті з використанням Leaflet та OpenStreetMap,

реєстрації нових дронів, моніторингу виконання місії в реальному часі за допомогою технології Server-Sent Events (SSE), а також формування звітів після завершення лабораторного аналізу SPE-картриджа. Система безпеки базується на розмежуванні прав доступу та криптографічному хешуванні паролів завдяки алгоритму Pbkdf2PasswordHash, що унеможливорює несанкціоноване втручання в роботу комплексу.

Маркетинговий аналіз підтверджує високу ймовірність успішного впровадження розробки на ринку. Проєкт займає вільну нішу між дорогими автономними суднами та ручною працею, пропонуючи оптимальне співвідношення ціни та функціональності.

Конкурентними перевагами розробленої системи є:

- значно нижча за аналоги типу АПС при збереженні необхідного функціоналу;
- повне виключення людини з небезпечної зони відбору проб;
- автоматизація процесу виключає людський фактор та помилки при консервації проби;
- отримання показників стану водойми у реальному часі ще до повернення дрона на базу.

Потенційними замовниками виступають державні екологічні інспекції та приватні рибницькі господарства, для яких особливо важливою є швидкість розгортання та безпека. Бізнес-стратегія передбачає прямі продажі та участь у тендерах, що дозволить забезпечити контроль якості та технічну підтримку клієнтів.

Незважаючи на успішну реалізацію поставлених завдань, система має потенціал до вдосконалення. Подальші дослідження за цією тематикою доцільно спрямувати на:

- подолання обмежень автономності шляхом розроблення автоматизованих станцій підзарядки або заміни акумуляторів, що дозволить виконувати безперервний моніторинг великих акваторій;

— інтеграцію алгоритмів машинного навчання на серверному рівні для автоматичного виявлення аномалій у показниках якості води, прогнозування поширення забруднень та побудови теплових карт екологічного стану водойм без участі оператора;

— розширення кількості наявних датчиків та адаптацію системи для роботи в умовах низьких температур;

— розроблення механізму спеціалізованої лебідки для забезпечення можливості відбору проб води з різних глибин, а не лише з поверхневого шару водойми;

— впровадження резервних каналів зв'язку для забезпечення надійної передачі телеметрії та керування у віддалених районах із нестабільним або відсутнім покриттям мобільних мереж.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Agricultural UAVs in the U.S.: potential, policy, and hype. URL: <https://www.sciencedirect.com/science/article/abs/pii/S235293851500021X?via%3Dihub>
2. Autonomous Surface Vehicles | ASV Manufacturers | Autonomous Boats. URL: <https://www.unmannedsystemstechnology.com/expo/autonomous-surface-vehicles-asv/>
3. Autonomous Underwater Vehicles - NOAA Ocean Exploration. URL: <https://oceanexplorer.noaa.gov/technology/subs-auvs/>
4. BlueROV2 — Affordable and Capable Underwater ROV. URL: <https://bluerobotics.com/store/rov/bluerov2/>
5. BlueROV2 — Affordable and Capable Underwater ROV. URL: <https://bluerobotics.com/wp-content/uploads/2025/04/BROV2-DATASHEET.pdf>
6. Autonomous Surface Vehicles — SeaRobotics. URL: <https://searobotics.com/products/autonomous-surface-vehicles/>
7. SR Utility Class — SeaRobotics. URL: <https://searobotics.com/products/autonomous-surface-vehicles/sr-utility-class/>
8. SR Endurance Class — SeaRobotics. URL: <https://searobotics.com/products/autonomous-surface-vehicles/sr-endurance-class/>
9. The Wave Glider | Unmanned Surface Vehicle by Liquid Robotics. URL: <https://www.liquid-robotics.com/wave-glider/overview/>
10. The Wave Glider | How It Works. URL: <https://www.liquid-robotics.com/wave-glider/how-it-works/>
11. Matrice 350 RTK — DJI. URL: <https://enterprise.dji.com/matrice-350-rtk>
12. Water Quality Continuous Monitoring System — EXO Sondes. URL: <https://www.yisi.com/exo>
13. Water Quality Sonde Multiparameter 4 Sensor Ports. URL: <https://www.yisi.com/exo1>
14. Multiparameter Water Quality Sonde Monitoring System. URL: <https://www.yisi.com/exo3>

15. WinOut F550 Drone Frame Kit 6-Axis Airframe 550mm Quadcopter Frame Kit with Landing Skid Gear & Mount (Full Set). URL: <https://www.amazon.com/QWinOut-6-Axis-Airframe-Quadcopter-Landing/dp/B08DV4YLWW?th=1>

16. The Cube Orange/+ With ADSB-In Overview — Copter documentation. URL: <https://ardupilot.org/copter/docs/common-thecubeorange-overview.html>

17. Here 3 GNSS GPS India Pixhawk Cube Orange quadcopter hexacopter Hex VTOL Fixed wing Autonomous mission way point navigation. URL: <https://www.rchyderabad.com/here3-gps-gnss-compass-pixhawk-cube-orange.html>

18. Рівняння Нернста - LibreTexts – Ukrayinska URL: [https://ukrayinska.libretexts.org/%D0%A5%D1%96%D0%BC%D1%96%D1%8F/%D0%97%D0%B0%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%85%D1%96%D0%BC%D1%96%D1%8F/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B0%3A_%D0%A5%D1%96%D0%BC%D1%96%D1%8F_\(Lower\)/16%3A_%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%85%D1%96%D0%BC%D1%96%D1%8F/16.04%3A_%D0%A0%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D0%BD%D1%8F_%D0%9D%D0%B5%D1%80%D0%BD%D1%81%D1%82%D0%B0](https://ukrayinska.libretexts.org/%D0%A5%D1%96%D0%BC%D1%96%D1%8F/%D0%97%D0%B0%D0%B3%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%85%D1%96%D0%BC%D1%96%D1%8F/%D0%9A%D0%BD%D0%B8%D0%B3%D0%B0%3A_%D0%A5%D1%96%D0%BC%D1%96%D1%8F_(Lower)/16%3A_%D0%95%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D1%85%D1%96%D0%BC%D1%96%D1%8F/16.04%3A_%D0%A0%D1%96%D0%B2%D0%BD%D1%8F%D0%BD%D0%BD%D1%8F_%D0%9D%D0%B5%D1%80%D0%BD%D1%81%D1%82%D0%B0)

19. Bioanalytical Method Development and Validation for the Estimation of Escitalopram Oxalate in Human Plasma by Using RP-HPLC Method | Semantic Scholar URL: <https://www.semanticscholar.org/paper/Bioanalytical-Method-Development-and-Validation-for-Deepthi/259722b29c56899f39070e23bc6e682fab5edb7d>

20. EZO™ pH Circuit | Atlas Scientific. URL: <https://atlas-scientific.com/embedded-solutions/ezo-ph-circuit/>

21. VisiFerm mA Dissolved Oxygen Sensor. URL: <https://www.hamiltoncompany.com/sensors/dissolved-oxygen-sensors/visiferm/visiferm-ma>

22. DS18B20 Datasheet and Product Info | Analog Devices. URL: <https://www.analog.com/en/products/ds18b20.html>

23. Temperature Sensor DS18B20 – Voltaat URL:
<https://www.voltaat.com/products/digital-thermometer>
24. EZO™ Conductivity Circuit | Atlas Scientific. URL: <https://atlas-scientific.com/embedded-solutions/ezo-conductivity-circuit/>
25. Turbidity sensor SKU SEN0189. URL:
https://wiki.dfrobot.com/Turbidity_sensor_SKU_SEN0189
26. SEN0189 Sensor: turbidity optical analog 5VDC Kit: module,cables
DFROBOT : Amazon.co.uk: Everything Else URL: <https://www.amazon.co.uk/DF-SEN0189-Sensor-turbidity-Interface-Channels1/dp/B06X169PBC>
27. Spring Boot. URL: <https://spring.io/projects/spring-boot>
28. Django overview | Django URL:
<https://www.djangoproject.com/start/overview/>
29. Express - Node.js web application framework URL: <https://expressjs.com/>
30. Learn | Jakarta EE | The Eclipse Foundation URL: <https://jakarta.ee/learn/>
31. E-bebek Microservis Yapısı. Merhabalar bu yazımda e-bebek kullanıcı... | by
Uğur Cem YÜKSEL | Ebebek Tech | Medium URL: <https://medium.com/ebebek-tech/e-bebek-microservis-yap%C4%B1s%C4%B1-11183f017aa9>