КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

До захисту допущено
Завідувача кафедри ІСТ
_____ Олександр Кучанський
(підпис) (ім'я, ПРІЗВИЩЕ)
"___"_____ 2022р.

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

спеціальності 126 «Інформаційні системи та технології»
освітньої програми «Програмні технології інтернет речей»

на тему: «Використання IoT - рішень та дронів для запобігання й усунення надзвичайних ситуацій»

Виконав студент 4 курсу, групи ІР-41

Олександр ШВИДЧЕНКО _____ _____
(ім'я, ПРІЗВИЩЕ) **(підпис)**

Керівник завідувач кафедри, д.т.н., доцент Олександр КУЧАНСЬКИЙ _____
(посада, науковий ступінь, вчене звання, ім'я, ПРІЗВИЩЕ) **(підпис)**

Консультант нормо контроль к.т.н., доцент Ростислав ЛІСНЕВСЬКИЙ _____
(назва розділу) (посада, науковий ступінь, вчене звання, ім'я, ПРІЗВИЩЕ) **(підпис)**

Рецензент к.т.н., доцент Юрій АНДРАШКО _____ _____
(посада, науковий ступінь, вчене звання, ім'я, ПРІЗВИЩЕ) **(підпис)**

Засвідчую, що кваліфікаційна робота не має запозичень з праць інших авторів без відповідних посилань.

Здобувач освіти_____
(підпис)

Київ - 2022 року

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра Інформаційні системи та технології
Освітній рівень Бакалавр
Спеціальність 126 Інформаційні системи та технології
Освітня програма Програмні технології інтернет речей

ЗАТВЕРДЖУЮ
Завідувач кафедри,
д.т.н., доцент
Олександр КУЧАНСЬКИЙ
_____
«__» _____ 2022 року

ЗАВДАННЯ
НА ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ БАКАЛАВРА

Здобувач освіти: Олександр ШВИДЧЕНКО

Група: ІР-41

1. **Тема кваліфікаційна робота бакалавра**: «Використання IoT - рішень та дронів для запобігання й усунення надзвичайних ситуацій».

Затверджена протоколом засідання кафедри ІСТ №18/20 від 01.12.2021 року

2. **Строк подання студентом готової роботи** – «18» червня 2022 р.

3. **Вихідні дані до роботи**: дослідження в області використання технологій БПЛА та IoT для автоматизації процесів та забезпечення безпеки. Програмні рішення для проектування системи та бази даних до даної тематики. Дані про аналоги систем, їх складових, їх характеристики та способи використання.

4. Зміст роботи: РОЗДІЛ 1 ПРОЄКТ ВИКОРИСТАННЯ ІОТ РІШЕНЬ ТА ДРОНІВ ДЛЯ ЗАПОБІГАННЯ ТА УСУНЕННЯ НАДЗВИЧАЙНИХ СИТУАЦІЙ (Опис проєкту, пояснення концепції системи, сфери використання, огляд аналогів та наукових публікацій); РОЗДІЛ 2 ТЕХНІЧНІ ОСОБЛИВОСТІ СИСТЕМИ (Визначення та розбір елементів системи, визначення ролі інтернету речей в проєкті); РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРАКТИЧНОЇЇ ЧАСТИНИ (Проектування системи, проектування та створення бази даних, розробка програмних методів для роботи з нею).

5. **Перелік графічного матеріалу**: Функціональна модель роботи системи автоматизованого реагування; способи використання дронів та їх функцій; архітектура системи екстреної евакуації на базі технологій IoT; види дронів та механіка їх роботи; використання датчиків в роботі дронів; проект “Quadroport”; станції контролю та управління дронів; проектування системи; проектування бази даних; програмна реалізація та тестування методів для роботи з базою данних.

6. **Календарний план виконання роботи:**

Етапи виконання кваліфікаційної роботи бакалавра	**Термін виконання**	**Результат виконання**
1. Вибір тематики кваліфікаційної роботи бакалавра	до 01.12.2021	виконано
2. Наказ про затвердження тем кваліфікаційної роботи бакалавра та призначення керівників	01.12.2021	виконано
3. Розробка плану кваліфікаційної роботи бакалавра і його погодження з керівником	25.12.2021	виконано
4. Написання І розділу кваліфікаційної роботи	20.01.2022	виконано
5. Написання ІІ розділу кваліфікаційної роботи	19.02.2022	виконано
6. Написання ІІІ розділу кваліфікаційної роботи	05.03.2022	виконано
7. Підготовка висновків і пропозицій	05.04.2022	виконано
8. Попередній захист кваліфікаційної роботи	20.04.2022	виконано
9. Перевірка на плагіат	до 15.06.2022	виконано
10. Нормоконтроль	до 17.06.22	виконано
11. Рецензування кваліфікаційної роботи бакалавра і представлення роботи на кафедру в друкованому вигляді	до 21.06.2022	виконано
12. Захист кваліфікаційної роботи бакалавра	24 .06.2022	

Дата видачі завдання «__» _____ 2022 р.

Керівник роботи: д.т.н. завідувач кафедри ІСТ Олександр КУЧАНСЬКИЙ _____
(підпис)

Завдання прийняв до виконання:
Здобувач освіти на освітньому рівні «бакалавр» 4-го курсу групи ІР-41

Олександр ШВИДЧЕНКО _____

(Власне Ім’я, ПРІЗВИЩЕ) (підпис)

АНОТАЦІЯ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

Факультет інформаційних технологій

Кафедра інформаційних систем та технологій

Освітня програма «Програмні технології інтернет речей»

Кваліфікаційна робота бакалавра Олександра ШВИДЧЕНКА

Тема роботи: «Використання IoT - рішень та дронів для запобігання й усунення надзвичайних ситуацій».

Мета кваліфікаційної роботи бакалавра – проектування системи автоматизованої реакції на надзвичайні ситуації, проектування та створення бази даних системи, розробка методів для подальшої роботи з базою даних.

Об'єкт дослідження – проект системи автоматизованого реагування на надзвичайні ситуації.

Предмет дослідження – проектування та створення бази даних системи, разом із програмними методами для роботи з нею.

Апробація результатів. Основні положення і результати досліджень, викладені у проекті, пройшли апробацію на науково-практичній конференції «IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)» (Балі, Індонезія 2019) [1], опубліковані в журналі «International Journal of Electrical and Computer Engineering» №10 за 2020р. [2] та в журналі «International Journal of Advanced Computer Science and Applications» №9 за 2018 р. [3].

Кваліфікаційна робота бакалавра складається зі змісту, вступу, основної частини, яка включає три розділи, висновків, списку використаних джерел та додатків. Всього 85 сторінок.

КЛЮЧОВІ СЛОВА: Дрони, БПЛА автоматизація, IoT, інтернет речей, датчики, UAV, GCS, надзвичайні ситуації (НС).

ABSTRACT

TARAS SHEVCHENKO NATIONAL UNIVERSITY OF KYIV

Faculty of Information Technology

Department of Information Systems and Technology

Educational Program "Software Technologies of the Internet of Things"

Qualification work of master Alexander SHVYDCHENKO.

Work topic: "Use of IoT solutions and drones to prevent and eliminate emergencies."

The purpose of the bachelor's qualification work is to develop software for a microcontroller to repeat the signal from the main router, design and implement the extension of the home Wi-Fi network by repeating the signal from the main router.

The object of research is the project of automated emergency response system.

The subject of research designing and creating a system database, along with software methods for working with it.

Approbation of results. The main provisions and results of the research outlined in the project were tested at the scientific and practical conference "IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)" (Bali, Indonesia 2019) [1], published in the Journal of Electrical and Computer Engineering No. 10 for 2020. [2] and in the 2018 Journal of Advanced Computer Science and Applications [3].

The bachelor's qualification work consists of the content, introduction, main part, which includes four sections, conclusions and a list of sources used. Total 85 pages.

KEY WORDS: Drones, automation, IoT, Internet of Things, sensors, UAV.

ЗМІСТ

ВСТУП

Життя в 21-му столітті, це життя в комфорті, великими можливостями та доступом до технологій, які ще 20 років тому вважались фантастикою. Ми здатні підтримувати контакт один з одним будучи в різних частинах світу, керувати приладами на відстані в кілометри та забезпечувати енергією цілі міста, проте ми досі люди, і людський фактор досі періодично стає причиною великих проблем, в свою чергу ріст технологій веде за собою ріст можливої небезпеки при їх несправності, а природа іноді більш безжальна ніж люди. Усе це забезпечує те, що надзвичайні ситуації, хоч і стали траплятись рідше, проте досі дуже небезпечні.

Цей неприємний але простий факт дає зрозуміти, що людству потрібна система що здатна ефективно запобігати та нейтралізувати надзвичайні ситуації, а події останніх років демонструють необхідність у можливості робити це без наражання людей на небезпеку.

Метою даної бакалаврської роботи є проектування системи реагування на надзвичайні ситуації, разом зі створенням баз даних для системи та розробкою програмних способів для роботи з її базами даних.

Для вирішення мети необхідно вирішити наступні задачі:

- Дослідити аналогічні системи.
- Визначити ключові складові системи.
- Дослідити механізми роботи її компонентів.
- Створити функціональну схему роботи системи.
- Спроектувати принцип роботи системи.
- Спроектувати та створити базу даних системи.
- Розробити та протестувати методи для взаємодії з базами даних.

Задачею даної бакалаврської роботи є дослідження готових рішень використання дронів та IoT технологій для виконання завдань віддалено, опис її компонентів, проектування системи з її базою даних та розробка методів для роботи з нею.

При дослідженні систем безпеки та реагування на надзвичайні ситуації, за допомогою технологій IoT був зроблений висновок, що існують ефективні способи своєчасного попередження та забезпечення безпеки цивільних осіб, особливо у випадках із пожежами. Проте вони не виконують функції боротьби із НС, хоча вони і допомагають рятувати життя цивільних та допомагати у боротьбі із пожежами, мають дуже вузьку спеціалізацію та не гарантують безпеку робітникам відповідних служб. Дані системи мають свої переваги в певних ситуаціях, проте якщо об’єднати технології БПЛА та IoT в систему автоматизованого реагування, можна отримати гнучку систему, що здатна не тільки рятувати життя людей, як цивільних так і робітників служб ДСНС, а і активно та ефективно боротись з різними надзвичайними ситуаціями та їх наслідками.

РОЗДІЛ 1. ПРОЄКТ ВИКОРИСТАННЯ IoT РІШЕНЬ ТА ДРОНІВ ДЛЯ ЗАПОБІГАННЯ Й УСУНЕННЯ НАДЗВИЧАЙНИХ СИТУАЦІЙ

1.1 Опис проєкту

Сучасні технологічні тенденції дозволяють впроваджувати активну цифровізацію та автоматизацію наших життєвих процесів, що поступово переходять у віртуальний простір і відбуваються завдяки високотехнологічним механізмам та приладам, через постійний потік інформації та легкий доступ до неї, дізнаватись про події можливо в лічені хвилини після того як вони трапились, а людина все менше і менше підвергає себе ризику та в цілому є більш захищеною. Не дивлячись на те, що сучасна технологічна епоха дуже молода а деякі її сфери тільки почали свій шлях розвитку, неможливо не помітити зміни між тим, що було 20 років тому і тим, що людство має зараз, а для історії, як і для людей – це феноменальні зміни. Проте світ досі далекий від майбутнього, де людство звільнене від важкої праці, повсякденних нудних завдань, небезпеки та власних помилок. Опираючись на це, сучасні події у вигляді епідемії коронавірусу, що змусила людей працювати на відстані один від одного, війни, яка піддала життя цілої нації постійному ризику та з наслідками якої вона буде зіштовхуватись ще багато років, масові пожежі (Туреччина, кінець липня 2021-го року, Австралія, період 2019 – 2020 роки), аварія на АЕС “Фукусіма – 1” 2011-го року та постійні надзвичайні ситуації, на мою думку необхідним шляхом у вирішенні питання безпеки як звичайних людей, так і тих хто має їх захищати та рятувати, є створення системи автоматичного реагування на НС (надзвичайні ситуації), за допомогою технологій що дозволять мінімізувати ризики для людей, а саме технології інтернету речей та дронів. Надзвичайна ситуація (НС) – представляє собою порушення умов життя і діяльності людей та тварин на певній території або об’єкті, спричинені природніми або техногенними явищами, що можуть призвести до матеріальних та людських втрат, а також нанести шкоду здоров’ю людей та тварин.

Проєкт представляє собою систему автоматизованої реакції на надзвичайні події за допомогою технологій інтернету речей та безпілотних літальних апаратів, або дронів.

Роль людини в ній зведена до фізичного мінімуму, а сам контролю дрону (UAV), спостереженням за виконанням завдання, перехоплення контролю дрона від системи у випадку необхідності (пошкодження дрону, нестандартна ситуація, що не внесена в протокол дії і т.п.) та аналіз отриманих даних після виконання завдання, що дозволяє запобігати та нейтралізувати НС швидше без безпосередньої небезпеки людям.

1.2 Пояснення концепції системи

Детальне пояснення системи варто почати з детального опису того, як саме вона має працювати. Сама концепція системи розроблялась для використання в державній службі України з надзвичайних ситуацій (ДСНС), тому і опис буде проводитись орієнтуючись на її використання в ДСНС.

Блок схема (рис 1.1) демонструє алгоритм роботи системи. В свою чергу функціональна модель системи (рис. 1.2) демонструє процес її роботи з перспективи оператора дронів та самого дрона.

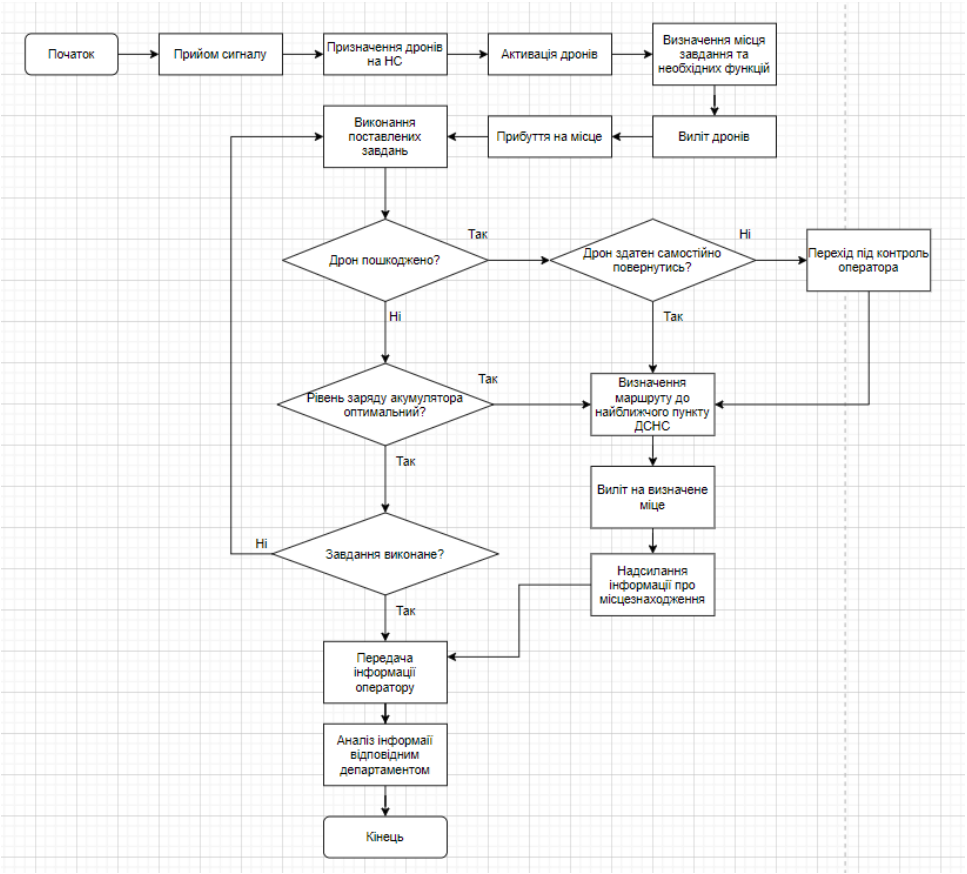


Рисунок 1.1 – блок схема алгоритму роботи

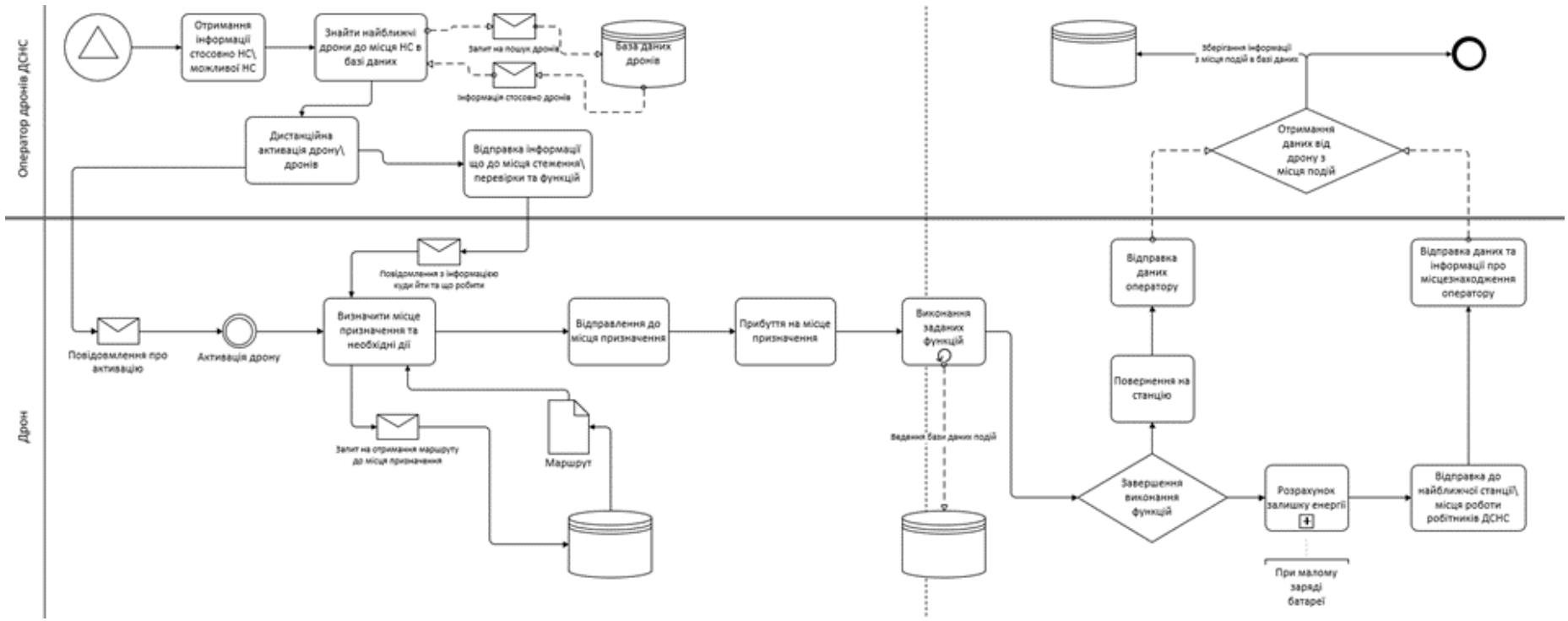


Рисунок 1.2 – Функціональна модель по роботі системи

Перш ніж розбирати модель, варто пояснити чому, не дивлячись на завдання по максимальній автоматизації реакції, людина має приймати участь в її роботі.

1) В чому полягає роль людини в системі?

Відповідь на це питання, насправді, доволі проста. Абсолютна автоматизація, що дозволить повністю позбавити людину необхідності безпосередньої участі в роботі із НС, на даний момент, нема, і найближчі роки не з'явиться. Справа в тому, що при роботі над системою я не зміг знайти, чи навіть вигадати, ефективний спосіб запуску її роботи без участі в цьому оператора.

Другим моментом, що змусив мене замислитись над цим питанням – можливі несправності в самій техніці та програмні помилки при виконанні завдання. Якою б надійною апаратура не була, завжди існує ризик того що щось піде не так, а коли на кону стоять безпека та життя людей, необхідно мати під рукою спосіб виправлення та взяття під контроль ситуації, для цього, у виникненні подібної форс-мажорної ситуації, оператор матиме можливість деактивації протоколів автоматизації та взяти дрон (UAV) під свій безпосередній контроль.

Останній момент, який потребуватиме безпосередньої участі людини, це аналіз ситуації та подій в яких брав участь дрон. Під час виконання завдання, дрони будуть везти запис подій, який потім буде передано на аналіз операторам та робітникам ДСНС для наступних цілей:

1) Передача запису правоохоронним органам на аналіз.
2) Аналіз середовища в якому працював дрон.
3) Пошук причин НС.
4) Оцінка наслідків НС.
5) Оцінка ефективності роботи дрона.
6) Внесення необхідних змін в стек-протоколів дій дрона.

Пояснивши цей момент, можна переходити до безпосереднього пояснення функціонування системи.

2) Як саме працюватиме система?

В системі ключові ролі виконують 3 елементи:

1) Оператор дронів.
2) Дрони.
3) Протоколи дій.

При виникненні НС, попередню інформацію про це отримує оператор дронів. Інформація містить в собі: місце НС, характер НС, загони дронів що приставлені до району де відбувається НС, час виникнення НС. Опираючись на цю інформацію, оператор дистанційно активує необхідні для завдання дрони, згруповані в загін. Разом із протоколом активації, дрони отримують повідомлення про своє завдання, що містить наступну інформацію:

- Місце призначення.
- Інформація про місцевість.
- Необхідні для виконання завдання функції.
- Протоколи що будуть задіяні на завданні.

Отримавши це повідомлення, система GPS будує найкращий шлях до місця призначення. При цьому процес пересування дронів є повністю автономним, проте за необхідністю, оператор має можливість перехопити керування.

Прибувши на місце призначення, загін починає виконувати свої задачі, які базуватимуться на задіяних протоколах, паралельно збираючи інформацію своїми датчиками та ведучи постійний відеозапис для подальшого аналізу.

Після виконання завдання, в силу вступають протоколи повернення на станцію для обслуговування дронів. При цьому є два варіанти розвитку подій, дрон або без повертається на базу, або, при малому залишку енергії чи через пошкодження, передає оператору зібрану інформацію дистанційно, разом зі своїми останніми координатами, або координатами станції, до якої він зміг долетіти.

1.3 Можливі сфери та способи використання

Як було вказано раніше, концепція даної системи розроблялась з оглядом на те, що вона буде використовуватись в першу чергу службою ДСНС України. Однак, вона також може бути використана і в таких сферах, як:

- Сільське господарство.
- Охоронні підприємства.
- Органи національної поліції України.
- Армійські структури.
- Вантажні перевезення.

Виконавча частина системи базується на 3 речах:

- Дрони (UAV).
- Датчики.
- Протоколи.

Однією з головних їх переваг є те, що дані елементи є модульними, тобто їх можна достатньо легко підігнати на певні задачі, замінити і т.д..

1.3.1 Сільськогосподарський напрямок

Наприклад, при використанні датчиків, що будуть надавати інформацію про ґрунт, протоколів, що матимуть сільськогосподарське направлення та відповідних моделей дронів (існує направлення БПЛА що спеціально створене для роботи в сільському господарстві, такі як Agras T16, або можливо замінити чи приєднати до вже існуючих дронів відповідні модулі), і отримаємо ідентичну систему, сільськогосподарського направлення. В ході дослідження були визначені 5 найбільш популярних дронів сільськогосподарського сектору, разом з їхніми характеристиками (таб. 1.1).

Таблиця 1.1 – Характеристики дронів сільськогосподарського сектору

Назва моделі	Запас речовин	Вага дрона	Час роботи при максимальній завантаженості	Середня ціна
1	2	3	4	5
Agras T16	16 л	39,5 кг	35 хвилин	$9 700
EA-2021	20 л	25 – 45 кг	35 хвилин	$14 500
eBee SQ	Моніторинг та аналіз	1,1 кг	60 хвилин	$3 200
Quantix Mapper		2,27 кг	45 хвилин	$8 000
Drone4Agro V3	500 л	20 кг	20 хвилин	$32 000

Рисунок 1.3 – Процес обприскування Agras T16 [4]

На сьогоднішній день, найбільш поширеню функцією дронів даного сектора є оприскування (рис. 1.3), однак вони також використовуються для аналізу грунту, нагляду з аналізом врожаю та проведення посівної, що дозволяє зменшити час,

витрачений на це, збільшити ефективність отримувати інформацію про його стан в режимі реального часу.

1.3.2 Сектор вантажних перевезень

Вантажні перевезення є тою сферою, де технології дронів вже починають активно впроваджуватись та досягають дуже вражаючих результатів. Спосіб використання дуже схожий із тим, як дана система може бути використана в сільськогосподарському секторі, а саме: використання відповідних для завдання протоколів, дронів, здатних пересувати вантажі та датчики, або спеціальні QR – коди. Конкретний приклад використання аналогу системи для здійснення доставки наведено в розділі про аналоги систем.

До найпоширеніших вантажних дронів зараз відносять (таб. 1.2):

Таблиця 1.2 – Характеристики дронів сектору вантажних перевезень

Назва моделі	Максимальна доступна вага	Вага дрона	Час роботи при максимальній завантаженості	Середня ціна
1	2	3	4	5
DJI Spreading Wings S1000+	11 кг	4,4 кг	15 хвилин	$1 200
DJI Matrice 600	15,5 кг	9,5 кг	35 хвилин	$6 000
Freefly Alta 8	18,1 кг	6,2 кг	10 хвилин	$18 200
Yuneec Tornado H920	5 кг	1,19 кг	21 хвилин	$2 800
DJI Agras MG-1P	25 кг	8,8 кг	10 – 20 хвилин	$12 000

Продовження таблиці 1.2

1	2	3	4	5
Aerones drone VertiPod	100 кг	55 кг	30 хвилин	$23 950

На сьогоднішній день, активно використовуються дрони компанії DJI, через їх фінансову доступність для широкого спектра користувачів, високу ефективність, на що вказують характеристики моделей та їх модульність.

1.4 Аналоги системи та моделей

На сьогодні існують певні моделі систем, що використовують дронів (UAV) в поєднанні із технологіями інтернету речей (IoT).

1.4.1. Prime Air

Компанія “Amazon”, активно займається інвестиціями в сферу дронів (UAV), та, починаючи з 2016-го року проводить тестування власної системи доставки в Англії, що базується на використанні повністю автономних дронів. Результати проведених тестів компанія продемонструвала у своєму відео “Amazon Prime Air’s First Customer Delivery”. Сама система має назву “Prime Air” (рис. 1.4).

Рисунок 1.4 – Презентація проекту “Prime Air” [5]

Дрон працює повністю автономно, починаючи від підняття у повітря (рис 1.5), посадки (рис. 1.6) та повернення на базу. Використовуючи технологію GPS, він прямує за побудованим маршрутом на відстані менше 120-ти метрів від землі, маючи на собі груз до 2-х з половиною кілограмів.

Рисунок 1.5 – дрон Prime Air із вантажем готується до відправки [5]

Результати тестування показали, що така система здатна привозити грузи до місця призначення менше ніж за 30 хвилин.

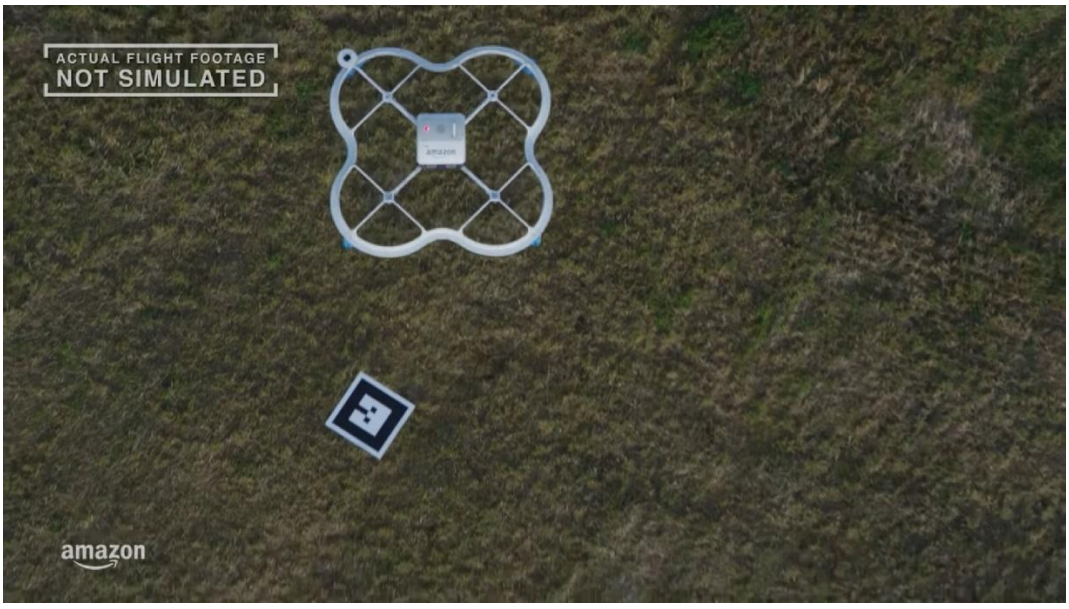


Рисунок 1.6 – дрон Prime Air із вантажем підлітає до спеціальної платформи з кодом, що слугує посадочною платформою та передає сигнал для початку процедури передачі вантажу [5]

На даний момент, дана система досі проходить тестування, проте в більших масштабах, і хоча ліміт ваги вантажу в 2.5 кг комусь здається не серйозним, ведуться розробки нових моделей, у тому числі і вантажних дронів здатних працювати із більшою вагою.

1.4.3 Автоматичне обприскування "BASF"

В свою чергу, компанія "BASF", що спеціалізується на агрокультурному секторі, співпрацюючи з фермерами Колумбії та Еквадору, створює мережу партнерів для покращення умов їх роботи та збільшення ефективності їх ферм, в тому числі і через використання технологій дронів (UAV) для розпилювання хімікатів та добрив (рис 1.7).

Рисунок 1.7 – Процес розпилювання хімікатів Agras T16 [6]

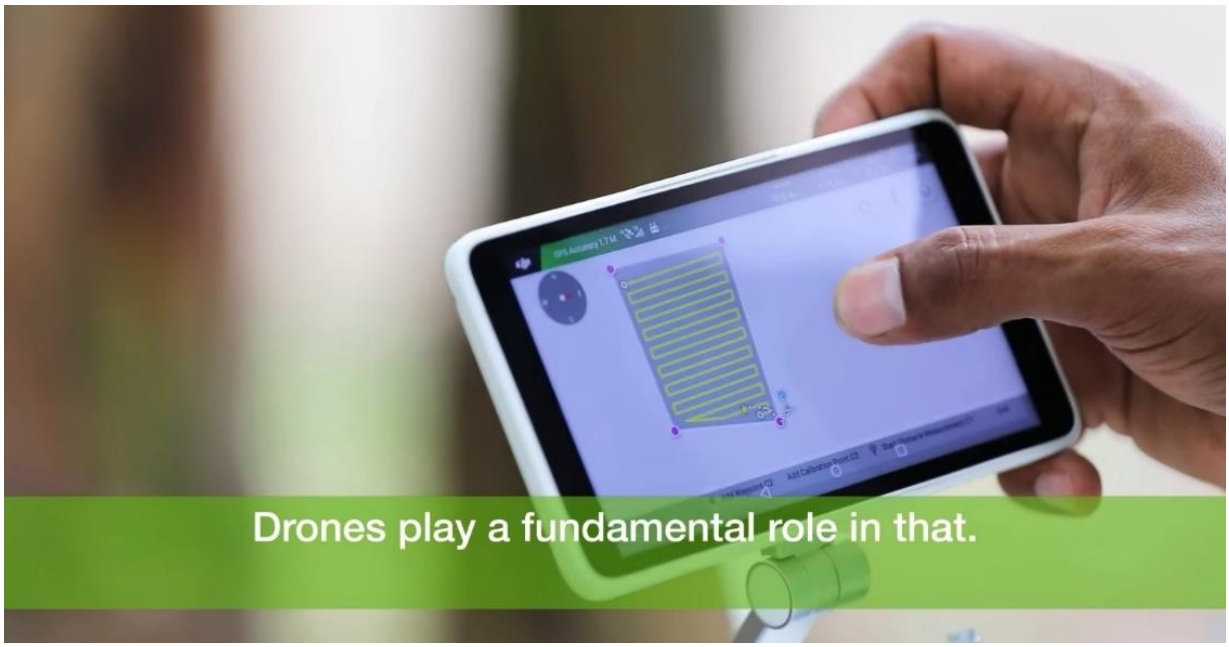


Рисунок 1.8 – Відстежування роботи дрона [6]

Як відмічає один з партнерів “BASF”, що бере участь в цьому проекті: ”Одні з головних переваг використання дронів в агро-секторі – це оптимізація використання хімікатів (рис 1.8), зменшений ризик для життя людей, зменшення шкідливого впливу на оточення та висока ефективність застосування. Усе це призводить до високої конкурентоспроможності.”[7].

Окрім розпилу хімікатів, дрони (UAV) від компанії “BASF” дозволяють фермерам виконувати ряд функцій, такі як:

- Аналіз ґрунту та полів.
- Посів.
- Моніторинг посівів.
- Зрошення.
- Оцінка стану здоров'я врожаю.

1.4.4 Огляд наукових джерел

В листопаді 2019 року була випущена стаття про систему екстреної евакуації, на базі технологій IoT від Американського університету Шарджи. Метою системи є підвищення рівня безпеки приміщень, створення та надання кращого маршруту для

евакуації та допомога в боротьбі із надзвичайною ситуацією. Для цього система використовує маячки на основі технології Blutooth (BLE маячки), мобільний додаток та розумні евакуаційні знаки, що створюють динамічний план евакуації, на основі того як розвивається ситуація.

Наведена архітектура системи (рис 1.9) демонструє яким чином проводиться обмін даними між пристроями системи та її 4 складові рівні IoT, а саме:

1) Сенсорний рівень.
2) Рівень комунікацій.
3) Рівень проміжного ПЗ.
4) Програмний рівень.

Сенсорний рівень використовує два типи кінцевих пристроїв, BLE маячки та мікроконтролери ESP32. BLE маячки має вбудований датчик температури, мікроконтролери під'єднано до MQ-2 сенсорів газу, що здатен зафіксувати від 300 до 10000 ppm газу, що дозволяє перейти від використання сенсорів цього призначення пожежниками до орієнтації в якості носія статичної інфраструктури, що буде передавати інформацію в реальному часі, як пожежникам так, цивільним, разом із можливістю визначення їх місце знаходження.

На рівні комунікацій відбувається зв'язок та передача інформації від BLE маячків до пристроїв мешканців або робітників будівлі та до серверу Raspberry Pi, використовуючи протокол Eddystone Beacon замість BLE. В той же час, Raspberry Pi та ESP32 здатні комунікувати через Wi-Fi, однак у випадку пошкодження WI-Fi роутерів вони переходять на протокол DigiMesh, що перекриває доступ до отримання інформації про місцезнаходження, проте гарантує роботу розумних евакуаційних знаків. На прикладному рівні, ESP32, Raspberry Pi та пристрій користувача обмінюються повідомленнями завдяки протоколу MQTT.

Рівень проміжного ПЗ представляє собою бекенд системи. Він складається з MQTT брокера, головного серверу та двох баз даних системи. MQTT брокер дозволяє

реалізувати архітектуру публікації/підписки, яку використовують мобільний додаток, ESP32 та Raspberry Pi. В той же час база даних CouchDB займається обробкою завдань, що не вимагають рішення в реальному часі, а Remote Dictionary Server (Redis) – розрахунком динамічного плану евакуації.

Програмний рівень представляє собою веб-інтерфейси, один з яких забезпечує інформацією про стан будинку пожежну станцію, до юрисдикції якої він входить, другий використовується для налаштування конфігурації системи певного будинку. Веб інтерфейс вікна конфігурації системи (рис. 1.10) здатен надавати інформацію про розташування BLE маячків та критичних точок безпеки, що використовуються для побудови шляху евакуації. У випадку пожежі, користувач отримує інформацію про безпечний евакуаційний маршрут через додаток на мобільному пристрої (рис. 1.11).

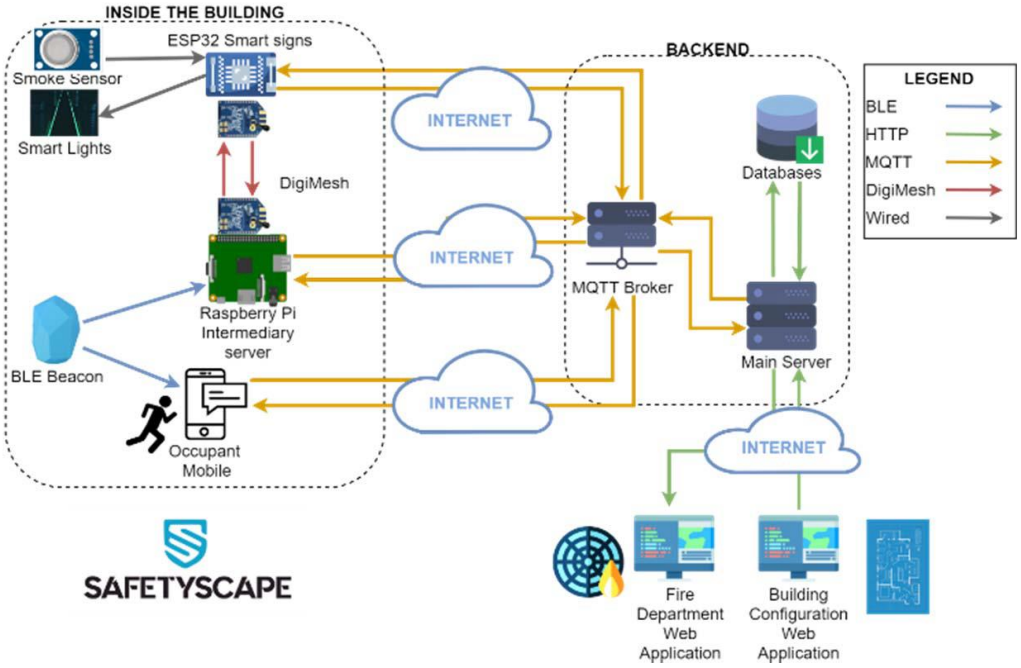


Рисунок 1.9 – Архітектура системи екстреної евакуації на базі технологій IoT

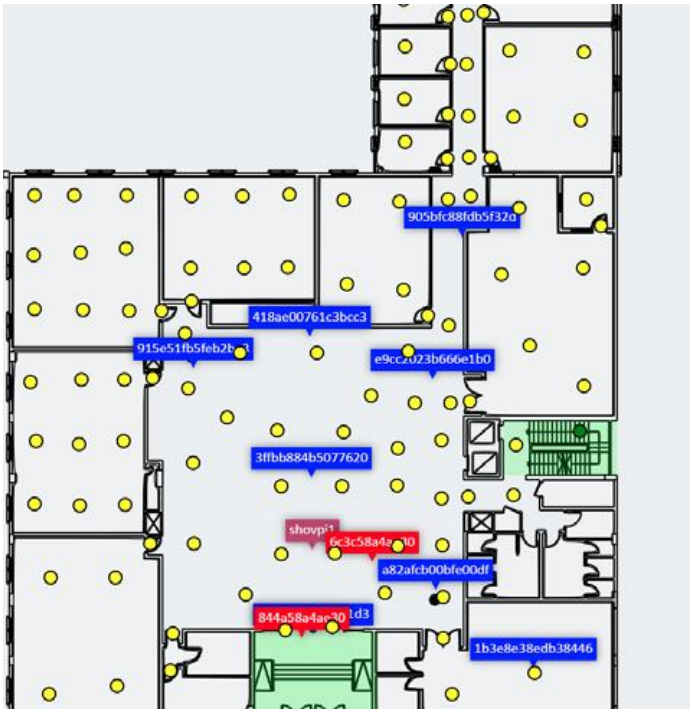


Рисунок 1.10 – Веб інтерфейс вікна конфігурації з розташуванням маячків та точок безпеки

Рисунок 1.11 – Інтерфейс додатку з побудованим динамічним евакуаційним шляхом

Дана система є чудовим прикладом використання IoT технологій для вчасного реагування на надзвичайні ситуації, в даному випадку – пожежі. Проте вона не відповідає повністю поставленій меті даної бакалаврської роботи, вона спеціалізована

під один тип НС, а саме – пожежі. Також, хоча вона здатна забезпечити збереження життя цивільних осіб під час пожежі та надавати інформацію пожежним під час виконання ними роботи, вона не здатна ліквідовувати або запобігати НС.

1.5 Висновки до розділу 1

Даний розділ підіймає проблему ризику для життя людей при виникненні надзвичайних ситуацій, як з боку громадського населення, так і з боку працівників служб, що займаються їх нейтралізацією. В якості рішення даної проблеми була запропонована та описана система автоматичної реакції на НС за допомогою технологій БПЛА, створена та описана її функціональна модель.

Наведені в розділі приклади аналогів систем, що базуються на схожих механізмах, демонструють, що активно ведуться розробки в даному направлені та, навіть, розроблені компаніями рішення активно тестуються, впроваджуються та використовуються, отже, не дивлячись на молодість даних технологій, в них є потужний потенціал, а способи використання зупиняються лише на уяві людей.

РОЗДІЛ 2. ТЕХНІЧНІ ОСОБЛИВОСТІ ПРОЄКТУ

2.1 Основні технічні та програмні компоненти

В розділі 1.3, про можливі сфери та способи використання даної системи, було сказано що ключовими її елементами є: дрони, датчики та проколи дій. Однак це не повний список її компонентів. До її складових відносяться:

- Дрони (UAV).
- Датчики.
- Протоколи.
- Станції базування\обслуговування дронів.
- Оператори.
- Станції роботи операторів.

Кожна складова буде розібрана окремо.

2.1.1 Дрони (UAV)

До дронів, або UAV (unmanned aerial vehicle) відносять безпілотні літальні апарати, скорочено БПЛА, що працюють автоматизовано або контролюються дистанційно. До найбільш поширених, на даний момент, відносять:

- Мультикоптери.
- БПЛА літакного типу.
- БПЛА – гелікоптери.

Найбільш популярним серед них є мультикоптери (рис. 2.1). Унікальний дизайн, фізика та простота в роботі робить з нього фаворита у використанні в різних сферах життя, починаючи від використання в кіноіндустрії та сфері розваг, закінчуючи сільським господарством, рятуванням людей, науковими дослідженнями та гасінням пожеж.

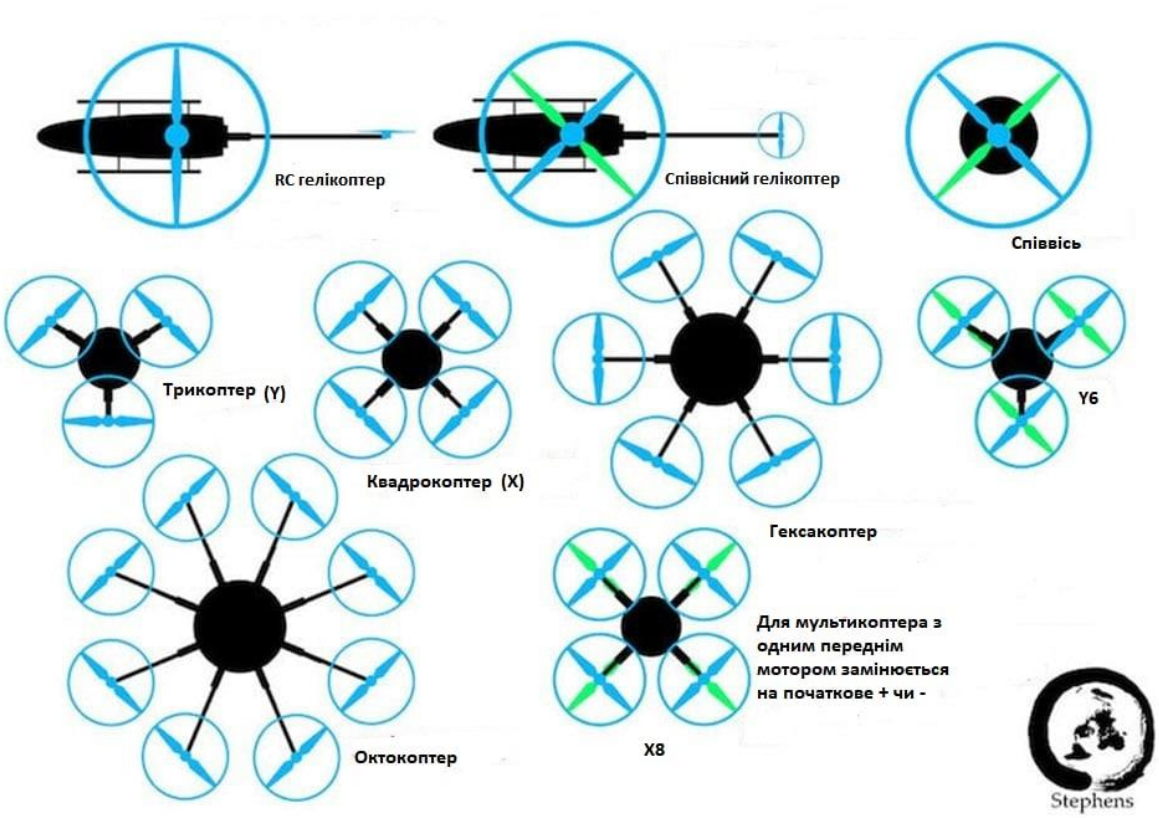


Рисунок 2.1 – Основні види дронів мультикоптерів[9]

Сам дрон складається з 6-ти компонентів:

- Рама – є основою дрона, до якої прикріпляються усі складові пристрою.
- Контролер – є “серцем” та “мозком” дрона.
- Мотори – відповідають за обертання пропелерів та льотні характеристики БПЛА.
- Гвинти – за рахунок синхронного обертання створюють тягу, за рахунок якої БПЛА здатен рухатись у повітрі.
- Акумулятор – надає компонентам БПЛА енергію для роботи. В залежності від її ємкості та рівня споживання енергії компонентів дрона, її може вистачати від 5 до 60 хвилин.
- ESC – регулятор швидкості. Він контролює постачання енергії то мотора (в кожного мотора свій ESC).

Дані компоненти – основа, яка є в кожному дроні, окрім них він може мати додаткові компоненти, в залежності від моделі, або необхідних функцій, починаючи від датчиків для створення мапи приміщення, до камер та модулів для обприскування полів. Однак вбудовані GPS та компас є настільки розповсюдженими в БПЛА та присутні ледь не в кожній моделі, що їх також можна віднести до основних їх компонентів (рис 2.2). GPS дозволяє дрону побудувати шлях до точки призначення, а компас допомагає дрону орієнтуватись в просторі та визначити його точне місце знаходження.

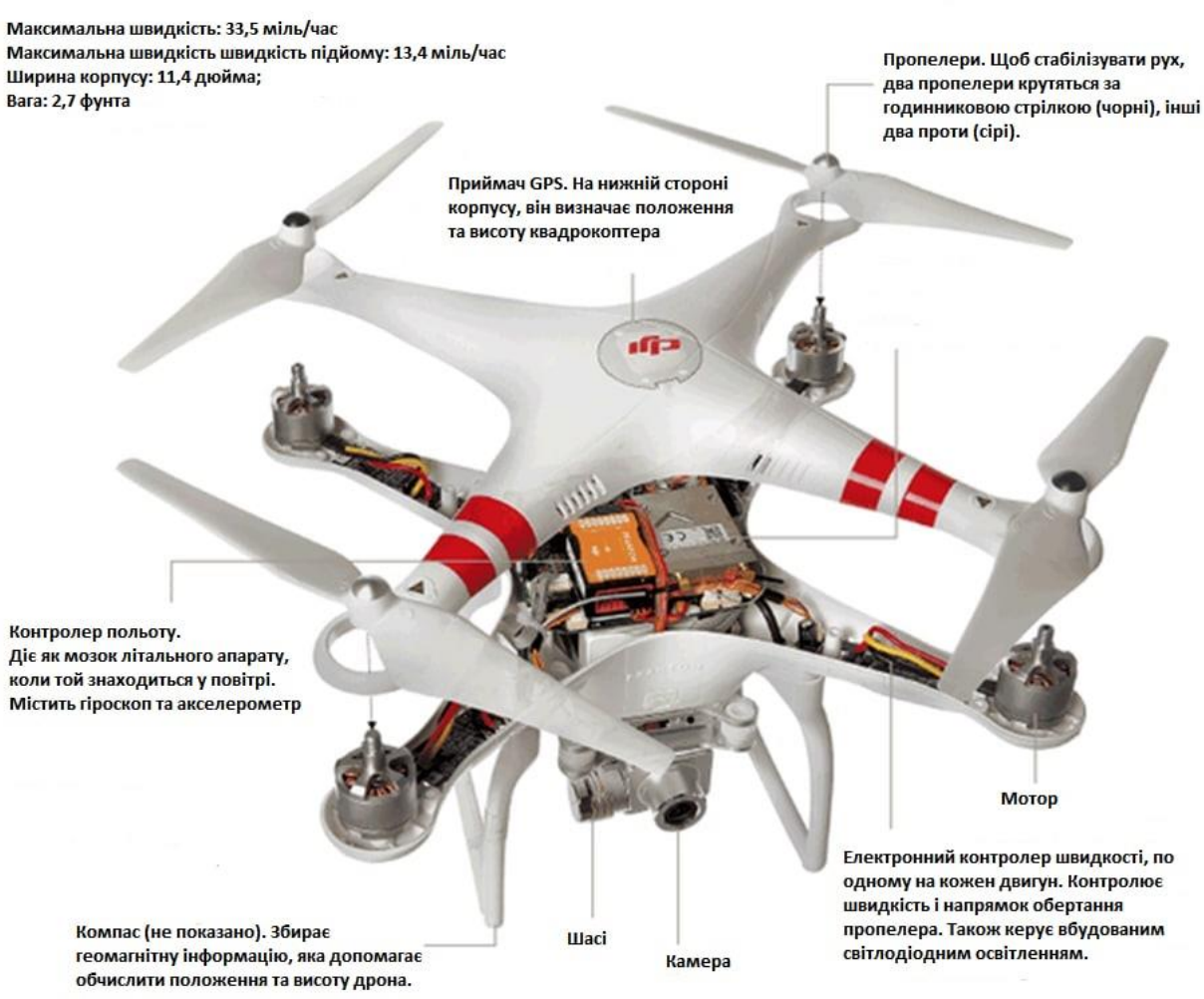


Рисунок 2.2 – Складові частини квадрокоптера з камерою[9]

Базова механіка роботи дронів виглядає наступним чином.

Як було вказано раніше, здатність дрона літати забезпечують гвинти, що генерують при обертанні тягу. При його роботі, кожен з двигунів має підтримувати

частину ваги та генерувати достатню силу тяги для підтримки дрона в повітрі. Ця частина залежить від кількості гвинтів.

Знаючи показник необхідної тяги, ми можемо визначати необхідну швидкість оберту гвинтів для, щоб тримати дрон в повітрі. Однак ця швидкість створює момент супротиву, який необхідно подолати кожному ротору.

Характеристики двигуна підбираються таким чином, щоб він мав можливість створити необхідний обертальний момент, що буде здатний подолати момент супротиву.

Таким чином, при польоті, швидкість гвинтів компенсує вагу дрону. В свою чергу, знаючи його вагу, ми здатні визначити який обертальний момент необхідно задіяти на кожному двигуні.

Щоб тримати дрон в рівновазі, необхідно щоб рівнодіюча сила (Resultant force) та загальний момент (Resultant moment) були рівні нулю. Рівнодіюча сила є сумою усіх тяг, що генерують гвинти дрона, при урахуванні сили тяжіння. В свою чергу, загальний момент це сума реакцій від обертання гвинтів за чи проти годинникової стрілки. Якщо обидві ці сили не дорівнюють нулю, результатом є прискорення.

Прискорення в вертикальній плоскості:

Баланс: при умові що кожна тяга двигунів однакова, вона буде сумуватись для підтримки ваги дрону та його балансу.

Прискорення в гору: при збільшенні швидкості двигунів, дрон прискориться та буде підійматись вище.

Посадка: при зменшенні швидкості двигунів, дрон буде опускатись.

Керування цими процесами бере на себе контроллер. При контролі дрону оператором, він допомагає в стабілізації та контролі швидкості. Однак, при автоматизованому польоті, без безпосередньої участі людини, він здатен повністю взяти на себе контроль та орієнтуватись в просторі, завдяки встановленим GPS, компасу та

датчиків, що дозволяють йому створювати віртуальну картину навколишнього середовища або камер.

2.1.2 Датчики

Датчики або давачі – пристрої, що збираюсть інформацію про навколишнє середовище, його зміни та передають інформацію про це в систему.

В системі використовуються датчики двох направлень:

- Датчики для дронів.
- Датчики інтернету речей (IoT).

Обидва направлення виконують одну функцію – збір та передача даних про оточення. Розберемо перші.

В минулому розділі було вказано, що контролер є серцем та мозком дрону. Датчики, в свою чергу, являються його очима. Вони розташовані на його корпусі та ведуть постійний збір інформації про оточення, починаючи від температури та сили вітру, до будь яких фізичних перешкод на шляху дрона (рис. 2.3 – 2.6), в залежності від того яку інформацію збирає встановлений датчик, що дозволяє йому своєчасно маневрувати в повітрі (рис. 2.5).

Уся ця інформація потрапляє в контролер на аналіз та дозволяє дрону формувати віртуальну динамічну карту місцевості (рис 2.3).

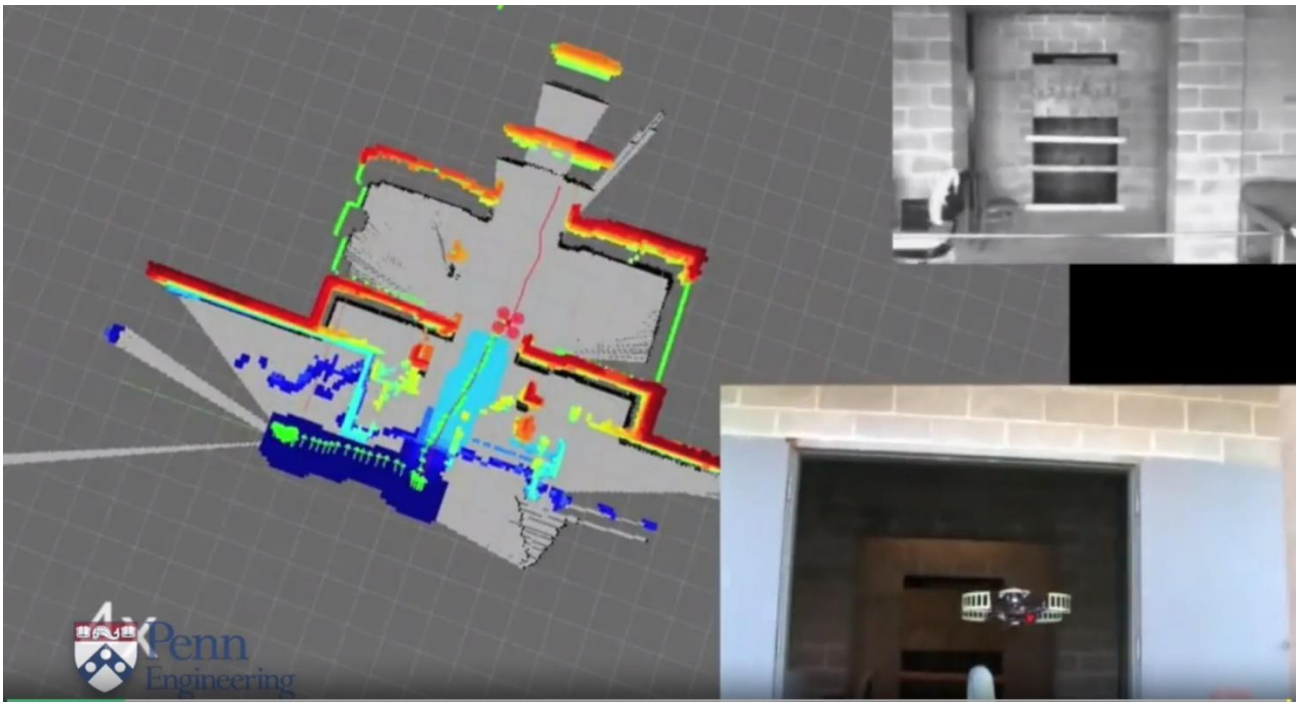


Рисунок 2.3 – Створення віртуальної мапи приміщення [11]

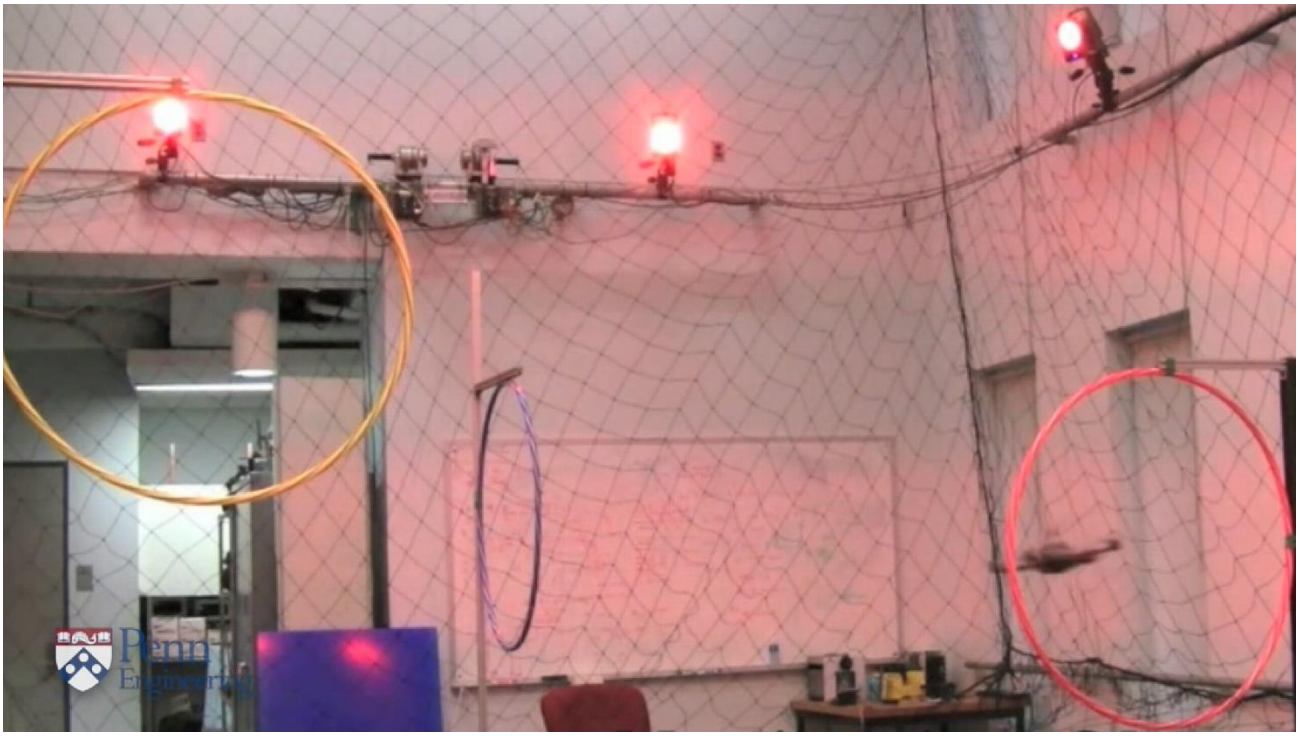


Рисунок 2.4 – Проліт дрону через прегради [11]

Рисунок 2.5 – Проліт дрону через динамічну прегради (кільце кинуто в повітря) [11]

Рисунок 2.6 – Використання камери для зчитування QR кодів з командами [11]

Модульна конструкція дронів дозволяє встановлювати різні датчики та модулі, для виконання великого спектру завдань, що робить його універсальним засобом для виконання широкого спектру функцій та коли необхідне виконання конкретної функції.

Для системи автоматично реагування, збір інформації з місця надзвичайної ситуації є одним з головних завдань. Це дозволяє проводити аналіз наслідків НС, її можливих причин, без безпосереднього знаходження в небезпечній зоні людини. Отримані дані аналізуються контролером, для виконання завдання дрону, буть то тушіння пожежі, пошук та порятунок людей або відстеження ситуації, та зберігаються для подальшого аналізу в центрі управління відповідних департаментів ДСНС. У випадку, якщо дрон, через недостатній запас енергії або через пошкодження, не здатен дістатись до найближчої станції базування для передачі даних та обслуговування, він відправляє зібрану інформацію дистанційно, разом зі своїми координатами та активує маяк, по типу тих що використовують для пошуку людей в лісі.

Датчики інтернету речей, в свою чергу, збираючи інформацію про оточення повідомляють систему про відповідні зміни, і у випадку сильного відбиття від норми, інформують про це систему (наприклад інформацію про пожежу, збільшений рівень радіації і т.п.). Разом із інформацією про зміни в оточенні, відправляється повідомлення з вказаним часом змін та місцем знаходження датчика.

2.1.3 Станції базування\обслуговування дронів

Станції базування дронів представляють собою своєрідний “аеропорт”, на якому розміщені, обслуговуються та тестуються усі робочі БПЛА.

Забезпечення станції енергією відбувається за допомогою сонячних батарей. Більша частина отриманої енергії піде на забезпечення електроенергією робочих приміщень, роботи апаратури та зарядки дронів. Залишки будуть накопичуватись в резерв на випадок аварії в секторі сонячних панелей, або при умові що ці самі панелі, через погодні умови, не здатні забезпечити станцію енергією, згідно з її потребами.

Станція має 4 приміщення для її функціонування:

- Ангар.
- Приміщення для обслуговування дронів.
- Приміщення для роботи операторів.

- Цивільне приміщення для відпочинку.

В ангарі зберігаються готові до вильотів дрони (рис. 2.7). Приміщення для обслуговування дронів окреме від ангару, щоб не виникало плутанини на місці із тим які дрони готові до роботи (також для цього, інформація про стан кожного окремого дрону вноситься до бази даних). В приміщенні для операторів знаходиться станція роботи операторів. Вона уде розглянута в наступному розділі. Цивільне приміщення станції розраховане на зняття стресу працівників станції та працює як місце для відпочинку та розваг.

Для відправлення дронів на завдання використовуються спеціальні платформи для мультикоптерів та злітна смуга для дронів літакного типу. Для безпечнішого та точного руху БПЛА під час посадки, в обох варіантах використовуються спеціальні QR коди з командами для дронів (рис. 2.6).

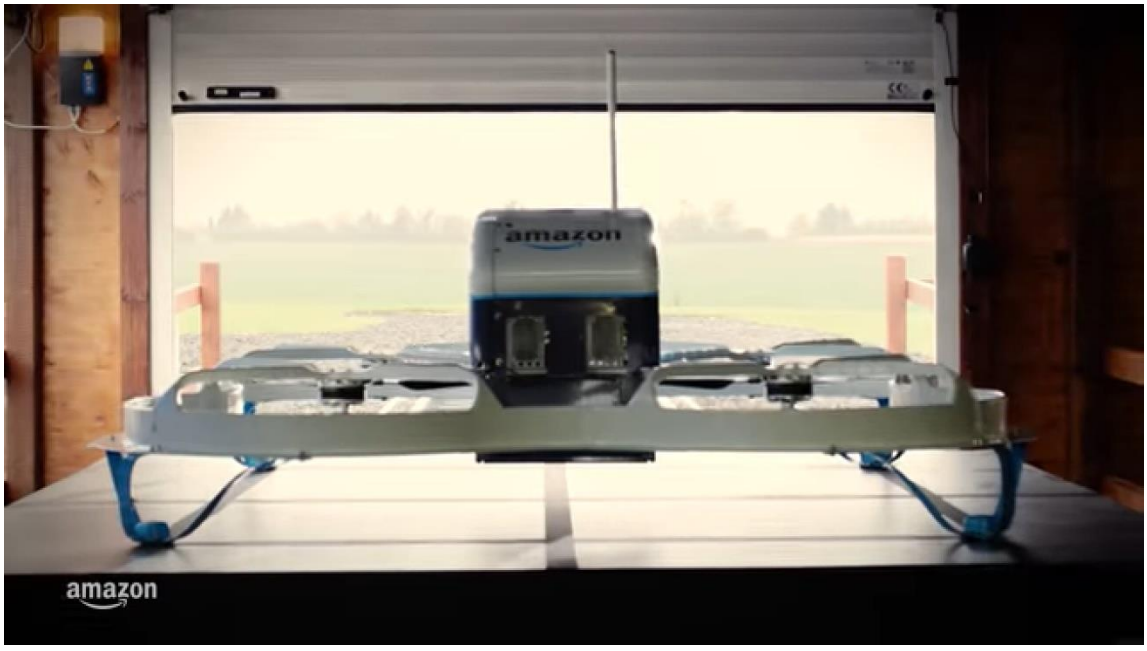


Рисунок 2.7 – Дрон автоматизованої системи доставки Amazon Prime Air з вантажем виїжджає з ангару на спеціальній платформі [5]

На даний момент, над схожим проектом ведуться роботи в “Всеукраїнській федерації власників безпілотників” (рис. 2.8 – 2.9).

Рисунок 2.8 – Візуалізація проекту “Quadroport” аеропорту для БПЛА [12]

Рисунок 2.9 – Вид з гори з кращим видом на злітну смугу [12]

На фотографіях видно злітну смугу (рис. 2.9), що застосовується для підйому в небо крилатих БПЛА, та платформи для здійснення посадки та підйому мультикоптерів. Проект “Quadroport” орієнтовано для використання в сфері розваг, через що в ньому не передбачено спеціальні ангари для масового зберігання дронів, проте він також може бути використаний в якості місця базування дронів ДСНС. Електроенергією його забезпечують 19 сонячних панелей.

2.1.4 Станції роботи операторів

Станції роботи операторів виконують функцію контроля БПЛА та аналізу отриманих даних.

В залежності від того, з яким дроном працює оператор, станція може варіюватись від невеликого джойстика з екраном, що передає вид з камери, до середніх мобільних станцій та спеціалізованого приміщення. Вона надає оператору широкополосний зв’язок з БПЛА через антену стеження, можливість відслідковувати інформацію про його політ, а також відслідковувати та аналізувати інформацію з його датчиків в режимі реального часу, в тому числі отримувати відео-звіт з його камер. Існує багато видів станцій керування дронами, від портативних пультів керування до стаціонарних станцій (рис. 2.10 – 2.12).

Рисунок 2.10 – Пульт керування від DJI [13]

Рисунок 2.11 – Мобільна станція керування Wing GCS [14]

Рисунок 2.12 – Станція керування дроном “Bayraktar TB2” [15]

Обладнання наземної станції керування БПЛА зазвичай складається з пультів дистанційного керування, комп'ютерів, відеодисплеїв, систем живлення, радіостанцій та іншого обладнання. Інструменти планування маршруту використовуються для планування маршрутів польоту дрона та встановлення висоти польоту, швидкості польоту, розташування польоту та польотних завдань. Станція передачі даних, підключена до порту даних, збирає та передає дані місії на контролер польоту.

Радіопередача даних є основним засобом зв'язку між БПЛА та наземними станціями, однак існують і інші варіанти, в тому числі завдяки технологіям Bluetooth та Wi-Fi, проте в нашому випадку вони малоефективні, так як перший варіант здатен забезпечити стабільний зв’язок між GCS та дроном на відстані лише 50 – 100 метрів, а другий – на відстані від 100 до 500 метрів. Протоколи інтерфейсу, що використовуються загальною радіопередачею даних, включають в себе інтерфейс TTL, інтерфейс RS485 та інтерфейс RS232, але є також деякі інтерфейси шини CAN-BUS з частотами 2,4 ГГц, 433 МГц, 900 МГц, 915 МГц, зазвичай більше 433 МГц, через те що 433 МГц є відкритим діапазон частот у поєднанні з перевагами довших хвиль 433 МГц та сильного проникнення, тому більшість цивільних користувачів зазвичай використовують 433 МГц, відстань становить від 5 кілометрів до 15 000 метрів і навіть далі.

Зрештою досягається зв'язок між БПЛА та GCS. GCS видають БПЛА завдання, через нього будуть передаватися висота польоту, швидкість та інші дані що будуть відслідковуватись в режимі реального часу. На сьогоднішній день, основними типами зв’язку з дронами є LOS зв’язку та BLOS зв’язку (рис. 2.13).

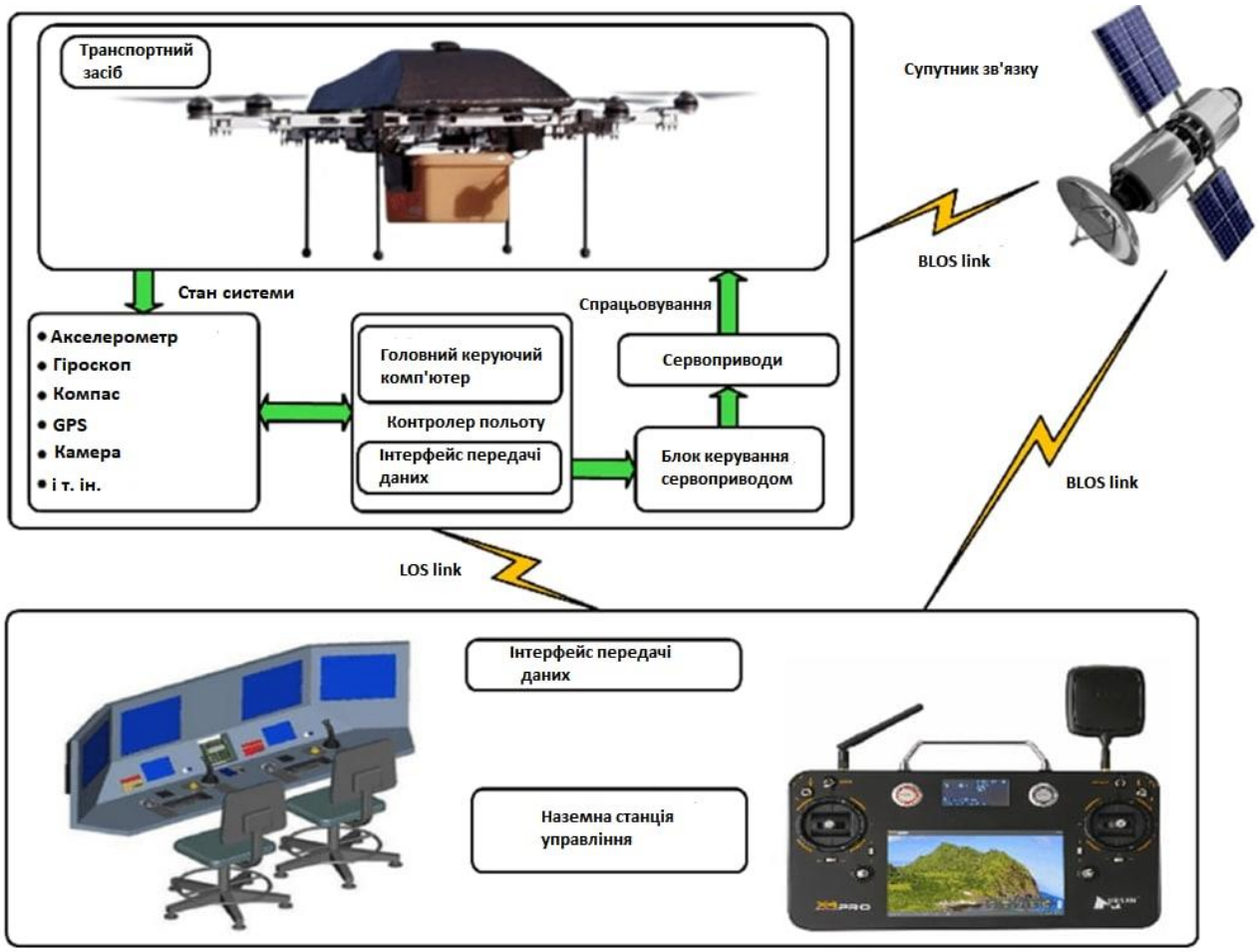


Рисунок 2.13 – Візуалізація зв’язку між станцією та дроном [16]

LOS зв’язок (Line-of-sight propagation) забезпечує зв’язок з БПЛА, поки його приймач сигналу знаходиться в полі зору передавача сигналу станції, в свою чергу BLOS зв’язок (Beyond-Line-of-Sight) дозволяють підтримувати зв’язок та обмінюватись інформацією через супутник.

2.1.5 Протоколи

Протоколи дій представляють собою набір функцій та правил що має виконувати дрон в тій чи іншій ситуації.

Протоколи дій є важливою частиною системи, так як вони вказують дронам як необхідно діяти при виконанні завдань та при певних умовах, що і дозволяє їм працювати автономно без безпосередньої участі людини та адаптуватись до різних ситуацій.

Початкові набори протоколів створюються за участі відповідних служб, озираючись на їхній знання, правила та досвід. Після виконання кожного завдання, проводиться аналіз зібраних даних та ефективності задіяних протоколів. У випадку низької ефективності, в задіяні протоколи будуть внесені відповідні зміни або створено нові для конкретних ситуацій.

2.2 Використання інтернету речей в проекті та його роль в ньому

Постійний потік та обмін даними між пристроями дозволяє максимально швидко отримувати повну картину ситуації, що особливо важливо у випадку надзвичайних ситуацій, коли на кону стоять життя та безпека людей. Потреба в наданні актуальної інформації та її отриманні в режимі реального часу є однією з ключових потреб при роботі служб ДСНС. І інтернет речей здатен змінити баланс сил на користь людей.

Автоматичний збір інформації з можливого місця НС та її аналіз системами розумних датчиків вже сам по собі здатен запобігати аваріям та попереджати людей про небезпеку через аналіз та обробку отримуємих даних, що в свою чергу, при необхідності, запускає систему попередження та надає точну інформацію стосовно характеру, місця розташування причин небезпеки та необхідних дій для упередження НС.

Якщо аварія все ж таки сталася, то завдяки інтернету речей, система здатна передавати дані про середовище прямо з небезпечної території. Однак це не єдиний варіант того, як IoT та розумні пристрої здатні вплинути на безпеку людей і їх можливість реагування та боротьби з НС.

Ведуться розробки та впроваджуються технології та механізми, що роблять людину більш захищеною, дозволяють їй виконувати свої задачі на відстані від небезпеки або отримувати актуальну інформацію знаходячись поза зоною ураження (роботи для роботи в зонах із високою радіацією, дрони для пошуку людей та збору інформації, супутникові знімки і т.д.). Дані технології мають великий потенціал, який може розкрити інтернет речей. Постійний потік інформації стосовно ситуації, оточення та її аналіз здатен забезпечити високий рівень автоматизації процесу боротьби з

надзвичайними ситуаціями та адаптування під ситуацію, що здатне збільшити ефективність усунення та запобігання НС та збільшити рівень безпеки людства в цілому.

Отже інтернет речей відіграє критичну роль в системі та є ключовою її складовою, що дозволяє реалізувати її повний потенціал автоматизації та можливості адаптації під конкретну ситуацію, забезпечити її високий рівень ефективності і безпеки для людей шляхом отримання даних в режимі реального часу без потреби людей знаходитись в епіцентрі небезпечної зони, та забезпечувати максимально повний обсяг інформації з місця подій та виконання відповідних дій згідно протоколів поведінки.

2.3 Висновки до розділу 2

Даний розділ підіймає технічну сторону системи та описує принцип роботи кожного її елемента.

Були описані та пояснені принципи роботи дронів, разом із їхньою механікою польоту та способами їх використання в системі. Були описані два види датчиків, а саме розумні датчики для оцінки середовища на місці потенційної надзвичайної ситуації та датчики що дозволяють дронам орієнтуватись в середовищі під час виконання завдання, адаптуватись до нього та збираи інформацію.

Були описані місце базування дронів та принцип його функціонування, разом із принципом роботи станції керування і способом контролю та підтримки зв’язку між станцією та дроном. Також була пояснена роль протоколів дій дронів, разом із тим як та що на них впливає.

Окремим пунктом було винесено пояснення ролі технологій IoT в системі та зроблено відповідні висновки.

РОЗДІЛ 3. РОБОТА НАД ПРОЄКТОМ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОЇ ЧАСТИНИ

3.1 Реалізація практичної частини

Практичною частиною дипломної роботи було проектування системи реагування, її бази даних та створення методів для роботи з нею.

Для створення бази даних важливо розуміти як саме має працювати система, тому на початку була проведена робота над проектуванням роботи системи за допомогою ПЗ “Process Modeler” та створені серія діаграм для її проектування та пояснення принципів роботи її елементів. Отримані діаграми та схеми наведені в розділі 3.1.1.

Завдяки цьому були розроблені логічна та фізична бази даних, маючи розуміння принципу роботи системи та її основних компонентів, що дозволило перенести її в програмний простір для подальшої роботи.

Останнім етапом практичної реалізації дипломної роботи була розробка програмних методів для можливості взаємодії з базою даних системи та їх тестування.

3.1.1 Проектування системи

Опис системи можна представити у вигляді діаграми типу IDEF0, тобто завдяки контекстній діаграмі. Вона представляє собою найзагальніший опис системи та її зав’язків із зовнішнім середовищем.

У наведеній далі діаграмі, в систему входять інформація із сенсорів, GPS навігатора, інформація від GCS та перелік доступних функцій дронів, керують нею протоколи дій та команди оператора надіслані через GCS, в якості механізмів виступають дрони, оператори, сенсори та GCS та GPS, а на виході як результат – запобігання або ліквідація надзвичайної ситуації та зібрана інформація про надзвичайну ситуацію, що буде використана для внесення необхідних змін або створення нових протоколів та аналізу надзвичайної ситуації.

Дана діаграма (рис. 3.1) представляє собою контекстну діаграму, що описує принцип роботи системи автоматизації реакції на надзвичайні ситуації.

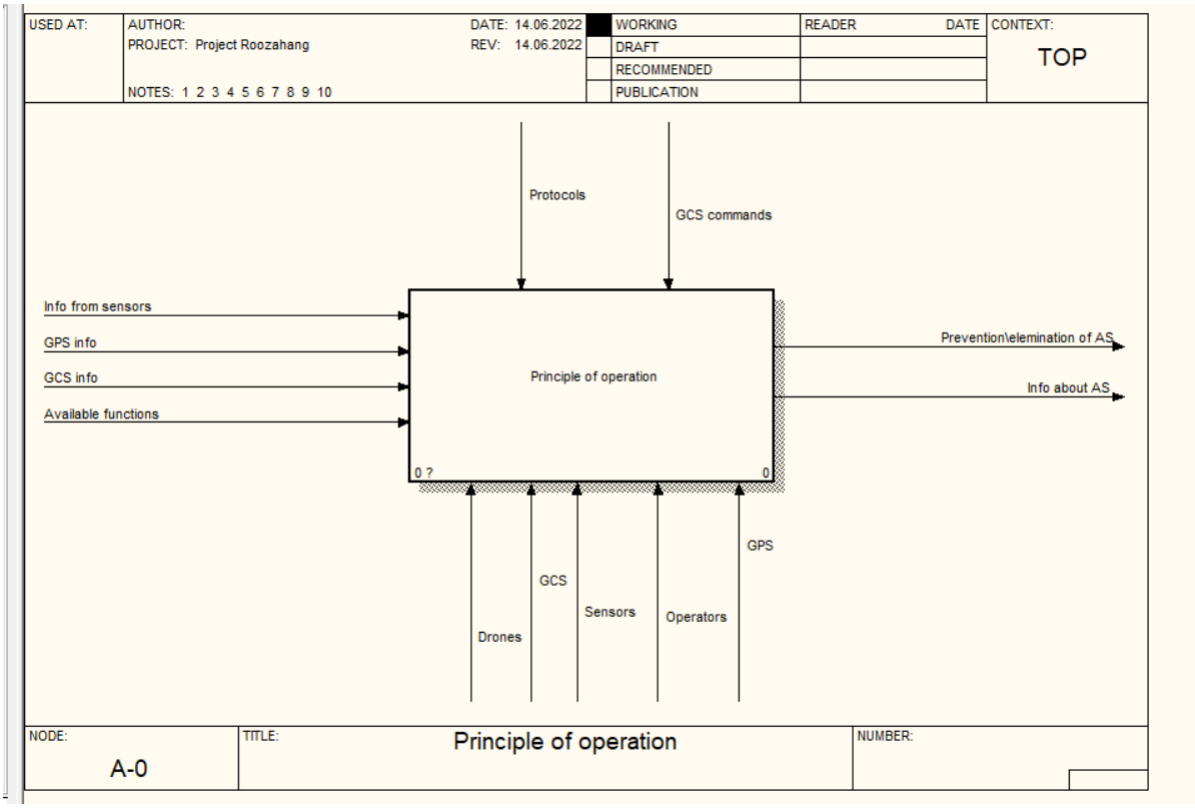


Рисунок 3.1 – Контекстна діаграма процесу роботи системи

Для кращого та більш глибокого розуміння діаграми паралельно була створена пояснювальна таблиця, що включає в себе інформацію стосовно контексту кожної стрілки. Стрілки контекстної діаграми описані та пояснені в пояснювальній таблиці (таб. 3.1).

Таблиця 3.1 – Пояснювальна таблиця до контекстної діаграми

Назва стрілки	Визначення стрілки	Тип Стрілки
1	2	3
Інформація з сенсорів	Інформація, отримана з сенсорів для визначення типу та місця НС	Input

Продовження таблиці 3.1

1	2	3
Інформація GPS	Інформація з GPS локатора дрона.	Input
Інформація GCS	Інформація зі станції керування (GCS)	Input
Доступні функції	Функції доступні для дронів для запобігання та ліквідації НС	Input
Протоколи	Протоколи для контролю дронів та сенсорів	Command
Команди GCS	Команди оператора для контролю дронів	Command
Дрони	Дрони, використані для роботи з НС	Mechanism
Оператори	Співробітники підприємства	Mechanism
Сенсори	Збирають інформацію про навколишнє середовище	Mechanism
Станція оператора (GCS)	Надає оператору контроль дронами	Mechanism
Сенсори	Надають онформацію про оточення в місці НС та оточення дрона	Mechanism

Продовження таблиці 3.1

1	2	3
GPS	Надає інформацію про місцезнаходження дрона та шлях до місця призначення	Mechanism
Запобігання та ліквідація НС	Кінцевий результат	Output
Інформація про НС	Інформація, забрана під час виконання завдання для подальшого завдання та внесення змін\створення нових протоколів дій	Output

Після контекстної діаграми йде її розбиття на діаграму декомпозиції. Даний тип діаграми надає більш детальний опис роботи окремого великого фрагмента. Таким чином ми здатні розбивати крупні фрагменти діаграми до тоді, поки не отримаємо найбільш повного опису процесу. Наступна діаграма є результатом функціональної декомпозиції (рис. 3.2) контекстної діаграми (рис. 3.1). На ній показані підпроцеси роботи оператора дронів, GCS та самого дрона, стрілки використані ті самі що і в контекстній діаграмі та з'єднані з відповідними підпроцесами, що надає більш детальний опис процесу, а також встановлено відповідний зв'язок між ними.

Оператор та GCS отримують інформацію про НС та інформацію з сенсорів про неї. Оператор активує необхідні дрони та передає їм, через GCS, усю необхідну інформацію для виконання завдання (необхідні для виконання функції, місце призначення, інформацію про місцевість і т.д.) та встановлює зв'язок із ним. Це дозволяє йому, при

необхідності, перехопити контроль над дроном, обмінюватись із нм інформацією в режимі реального часу та слідкувати за ситуацією “очима” дрона. БПЛА, в свою чергу, діє згідно протоколам, відповідних до ситуації, та команд, надісланих оператором через GCS. Він також отримує інформацію з сенсорів, проте тих, що розташовані на ньому. Таким чином він має можливість збирати необхідну інформацію про оточення, передавати її на GCS та проводити дії, відповідні до ситуації в якій він знаходиться. Результатом роботи є запобігання та ліквідація надзвичайної ситуації, та зібрана під час роботи системи інформація для аналізу.

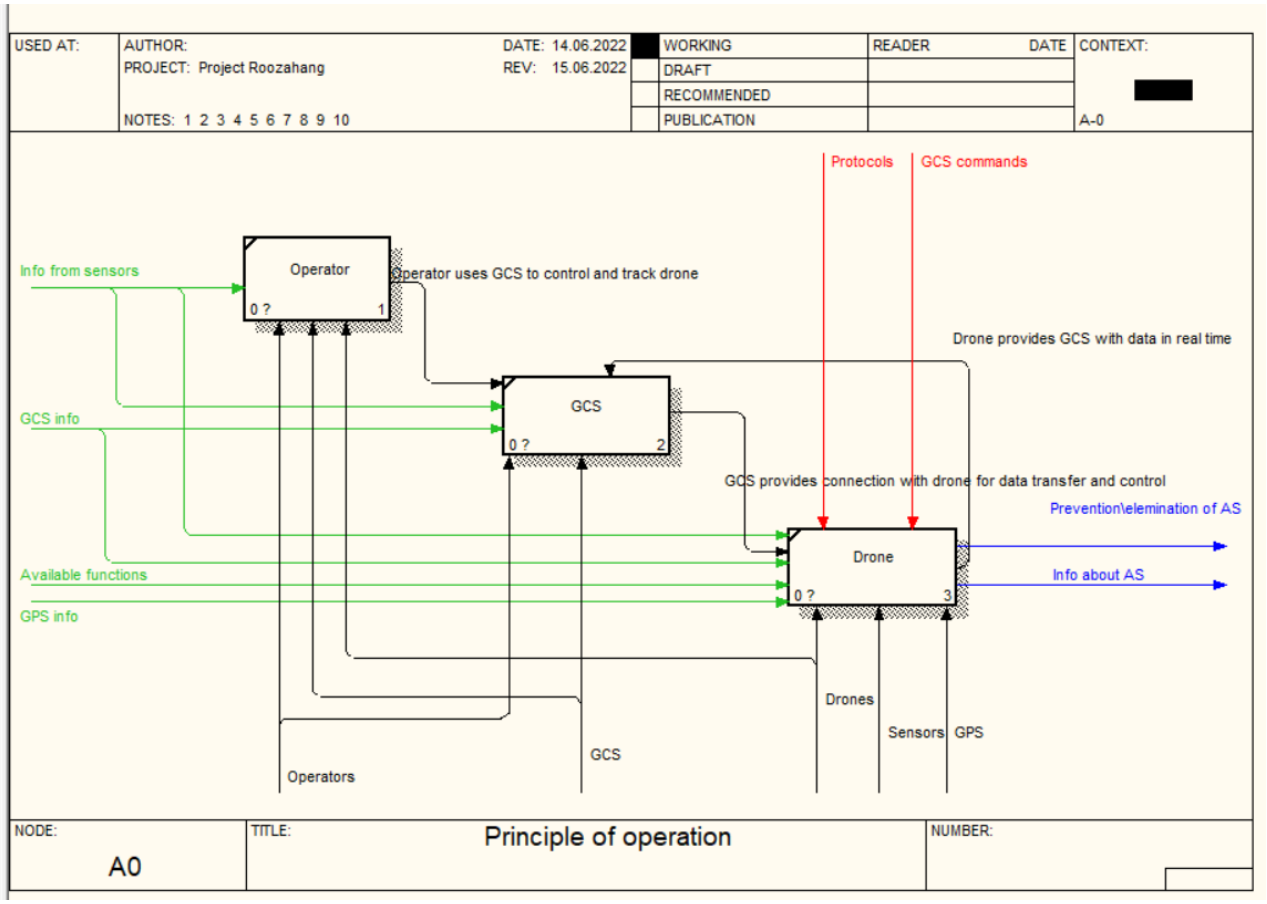


Рисунок 3.2 – Діаграма декомпозиції

Пояснення та опис процесу наведено в пояснювальній таблиці (таб. 3.2).

Таблиця 3.2 – Пояснювальна таблиця до процесів діаграми декомпозиції

Назва процесу	Визначення процесу
1	2

Продовження таблиці 3.2

1	2
Operator	Отримання інформації з сенсорів, надсилання команд та інформації дронам, робота з базами даних та GCS.
GCS	Отримання інформації з сенсорів, отримання та передача інформації від дрону до оператора, надсилання команд дрону, підтримка зв’язку з дроном.
Drone	Отримання інформації від оператора через GCS, отримання команд оператора через GCS, виконання функцій та протоколів, збір інформації.

Для кращого розуміння під процесів існують ще два види діаграм: IDEF3 та DFD. IDEF3 – була створена для нотації одночасно технічних та бізнес процесів, що була використана для дрону (рис. 3.3). DFD – діаграма потоків даних, що була використана для оператора (рис. 3.4). [4] Обидва типи діаграм дозволяють представити детальний опис підпроцесів, однак зі своїми особливостями.

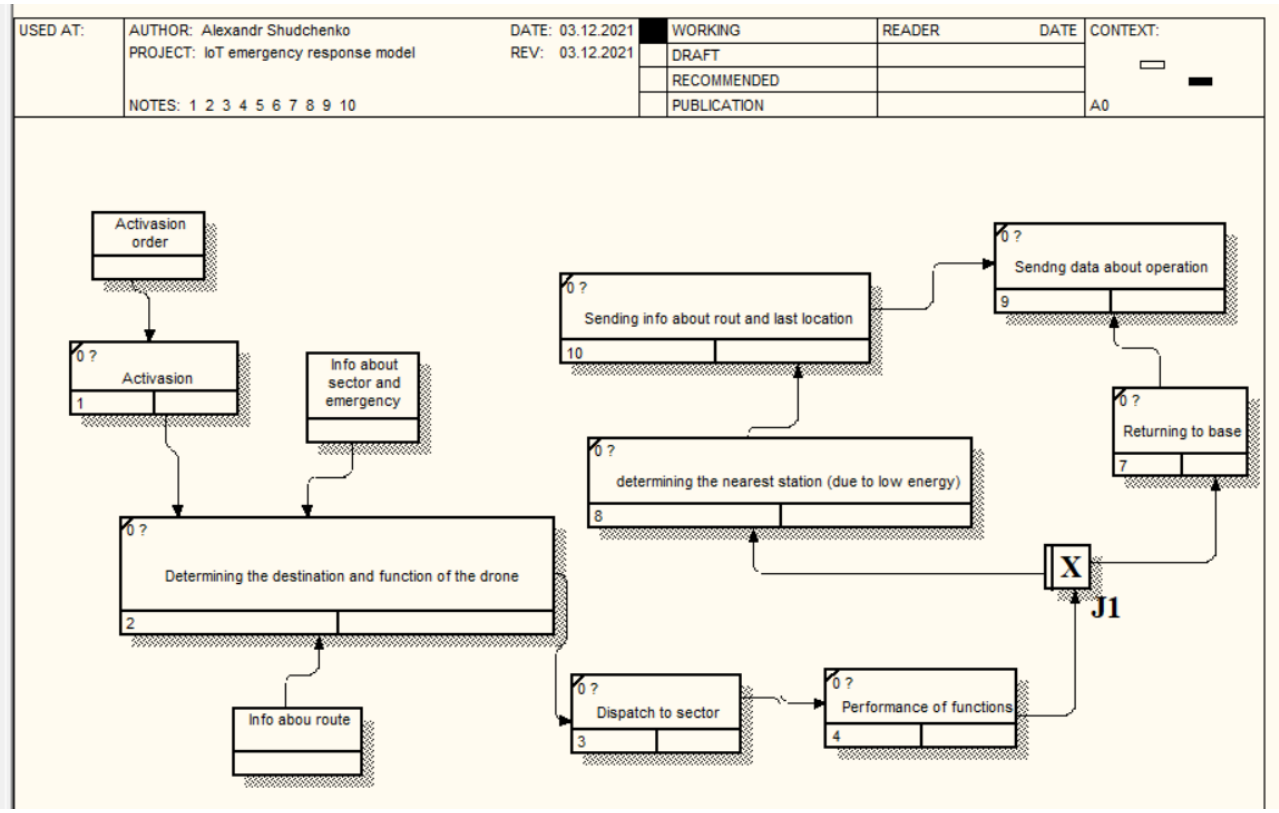


Рисунок 3.3 – IDEF3 діаграма для дрону

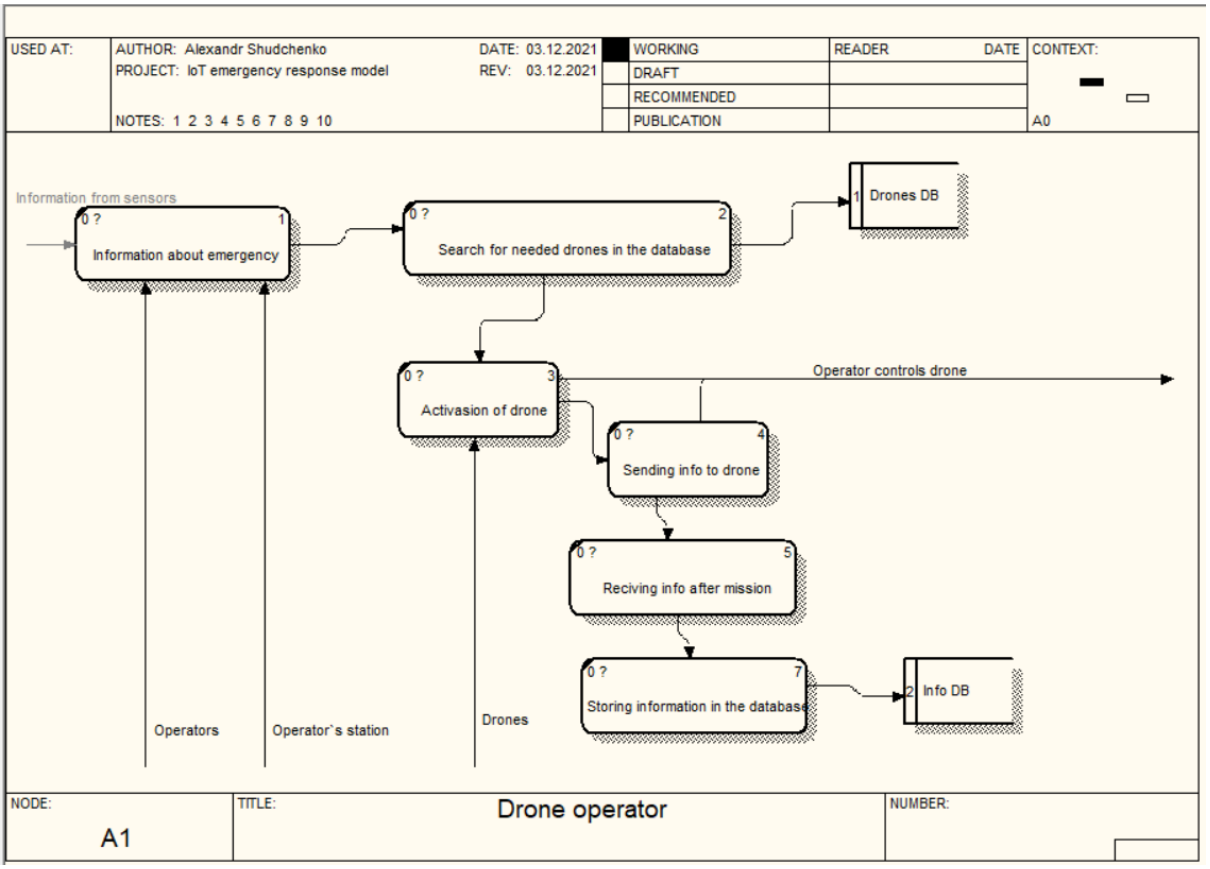


Рисунок 3.4 – DFD діаграма для оператора дронів

Можна доповнити діаграму декомпозиції, додавши в неї показники фінансових витрат та отримавши згенеровані звіти про витрати.

3.1.2 Побудова бази даних

Маючи готовим перечисленні вище діаграми та моделі, ми можемо приступити до розробки логічної моделі бази даних. Для її створення було використане програмне забезпечення ERBuilder Free 4.0. Даний додаток було обрано через його доступність та інтуїтивно зрозумілий інтерфейс, однак він має певні недоліки, увагу на які буде звернено пізніше. При розробці логічної моделі бази даних процес починається зі створення таблиць та внесення в них необхідних полів, налаштування ключів, значень за замовчуванням та типів зв'язків (рис.3.5).

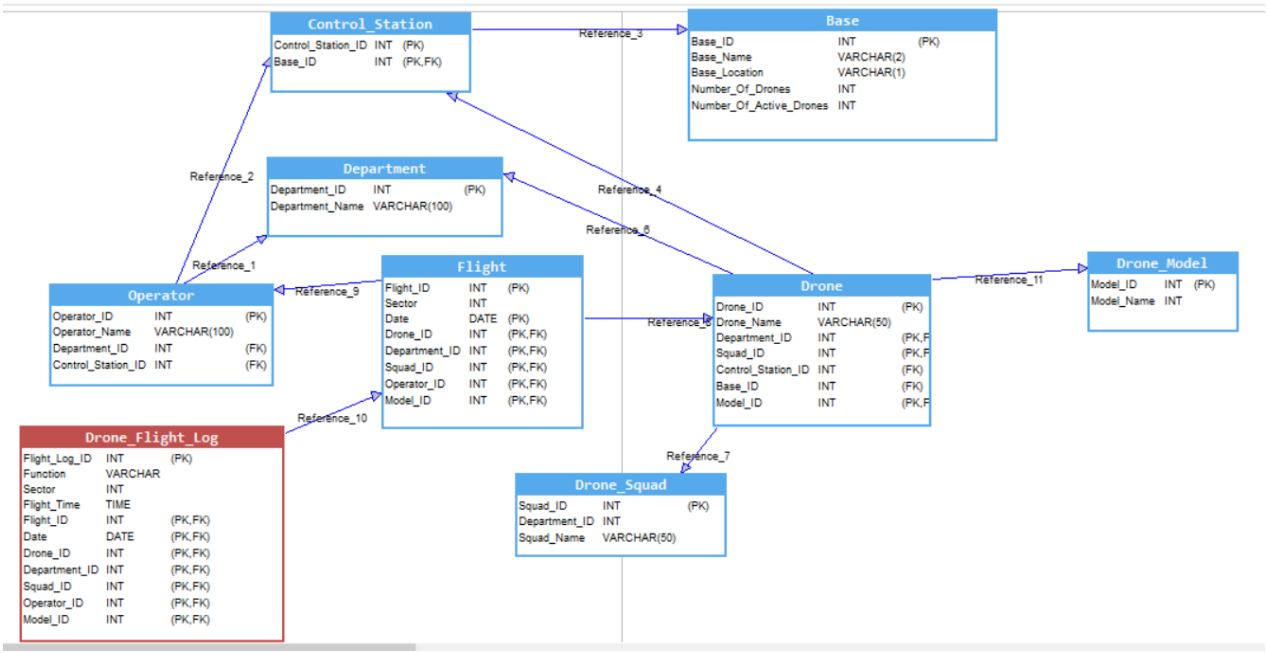


Рисунок 3.5 – Логічна модель бази даних

У зв'язку із відсутністю можливості поділу моделей на логічні та фізичні, фізична модель бази даних була створена вже після перенесення в СУБД (рис. 3.6).

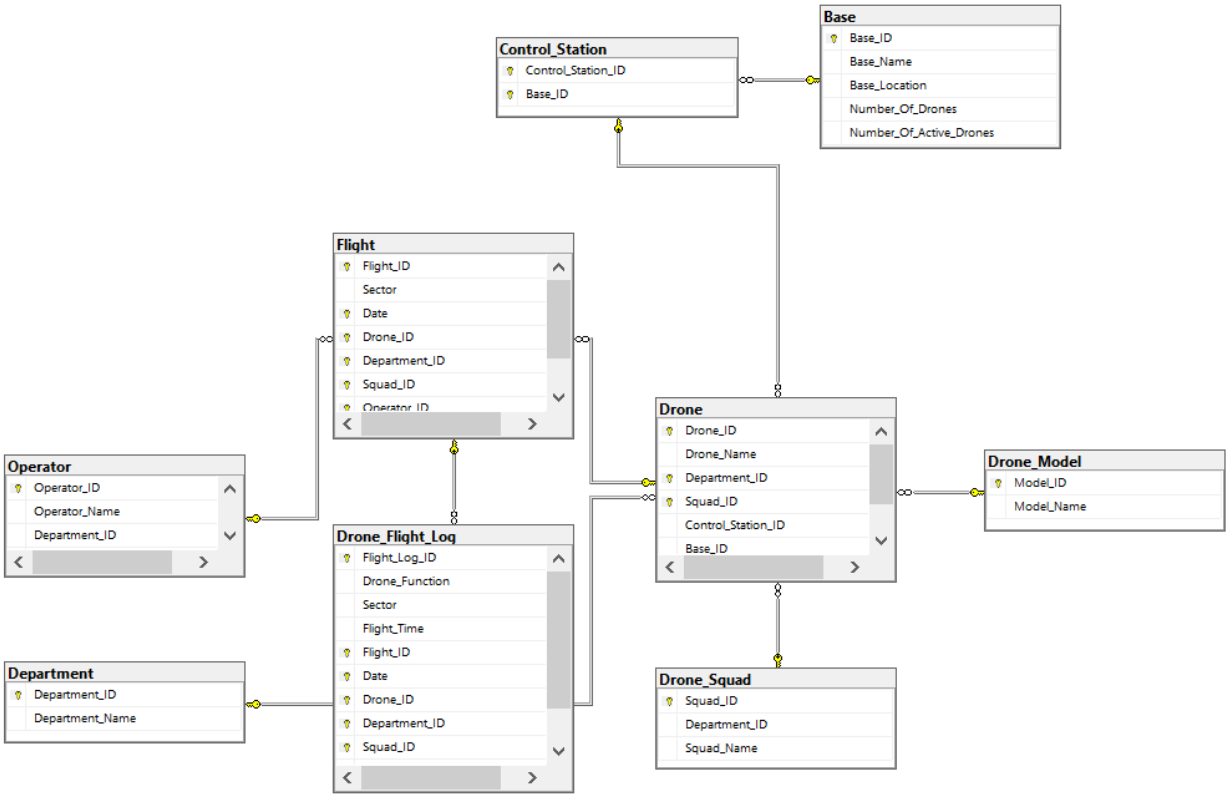


Рисунок 3.6 – Фізична модель бази даних

Для подальшої роботи зі створеною базою даних, необхідно під’єднати її до SQL серверу, щоб мати можливість програмної взаємодії з нею та внесення даних. Для цього, на робочому ПК було створено SQL сервер за допомогою SQL Server installer (рис. 3.7) та підключено до Microsoft SQL Server management studio (рис. 3.8).

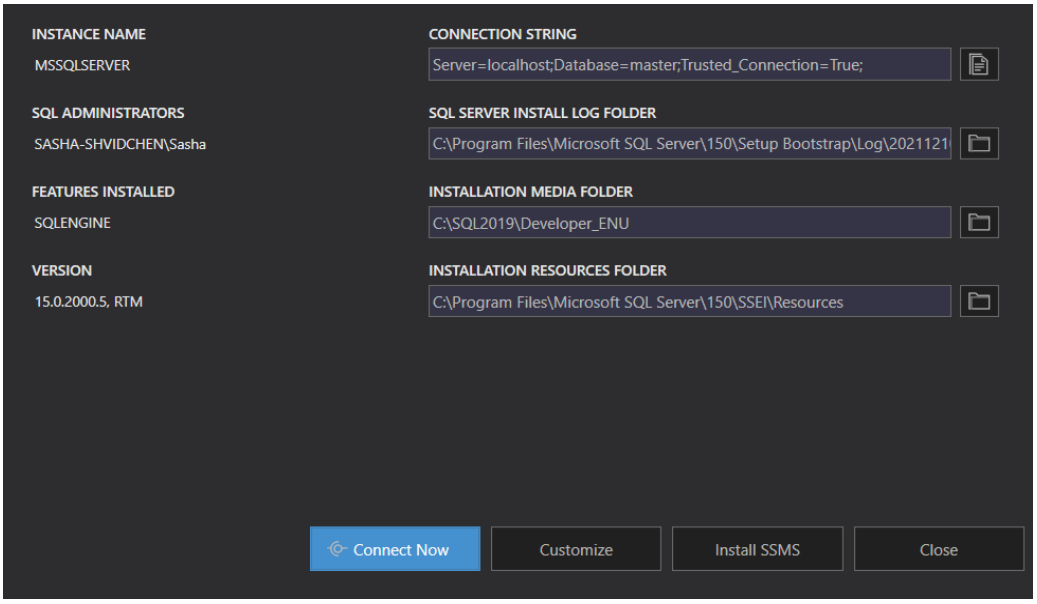


Рисунок 3.7 – створення та запуск SQL серверу на ПК

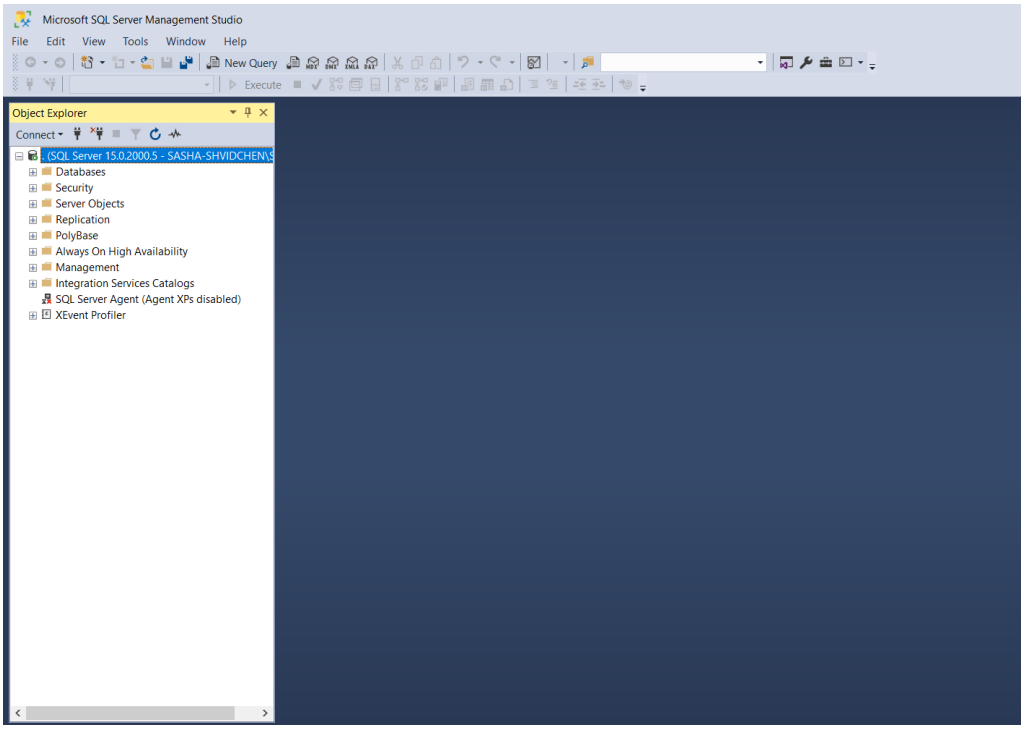


Рисунок 3.8 – підключення до серверу через SQL Server Management

Керування сервером відбувається через SQL Server Configuration Manager (рис. 3.9).

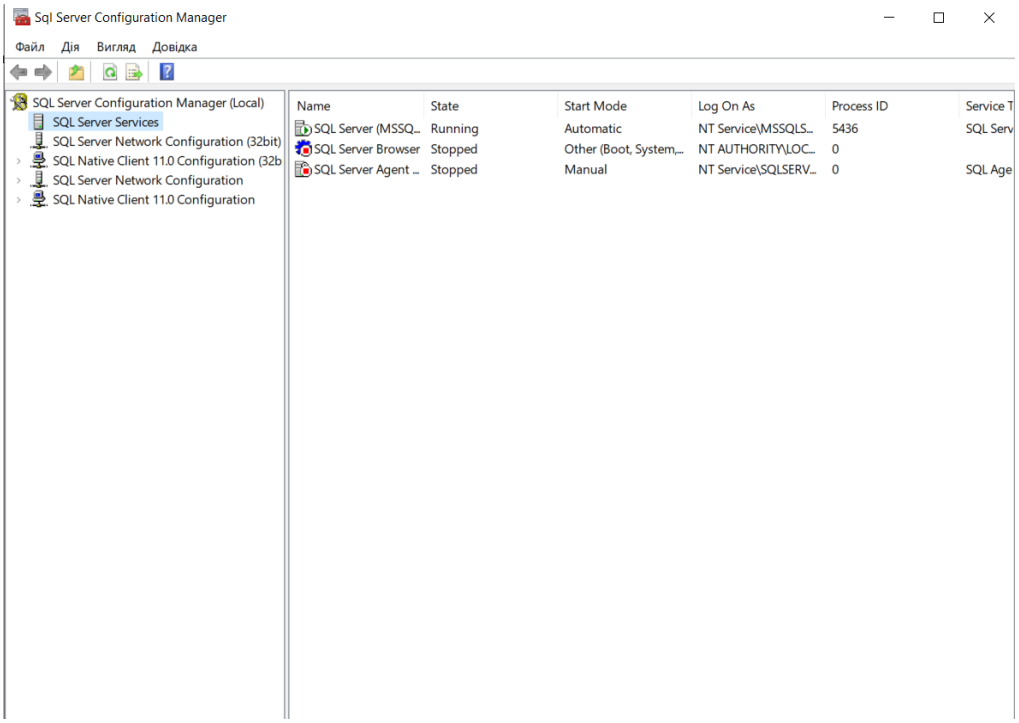


Рисунок 3.9 – Інтерфейс SQL SCM

На початку роботи ми мали першу версію бази даних нашої системи, яку ми взяли за основу для виробничої практики. Однак вона є незакінченою, тому на її основі ми створили нову та меншу базу даних для виконання поставленого завдання, а саме, за основу були взяті та змінені таблиці “Departments” та “Operators” старої бази даних та створено “ Drone_models” в якості доповнення для подальшої роботи (рис. 3.10 – 3.11).

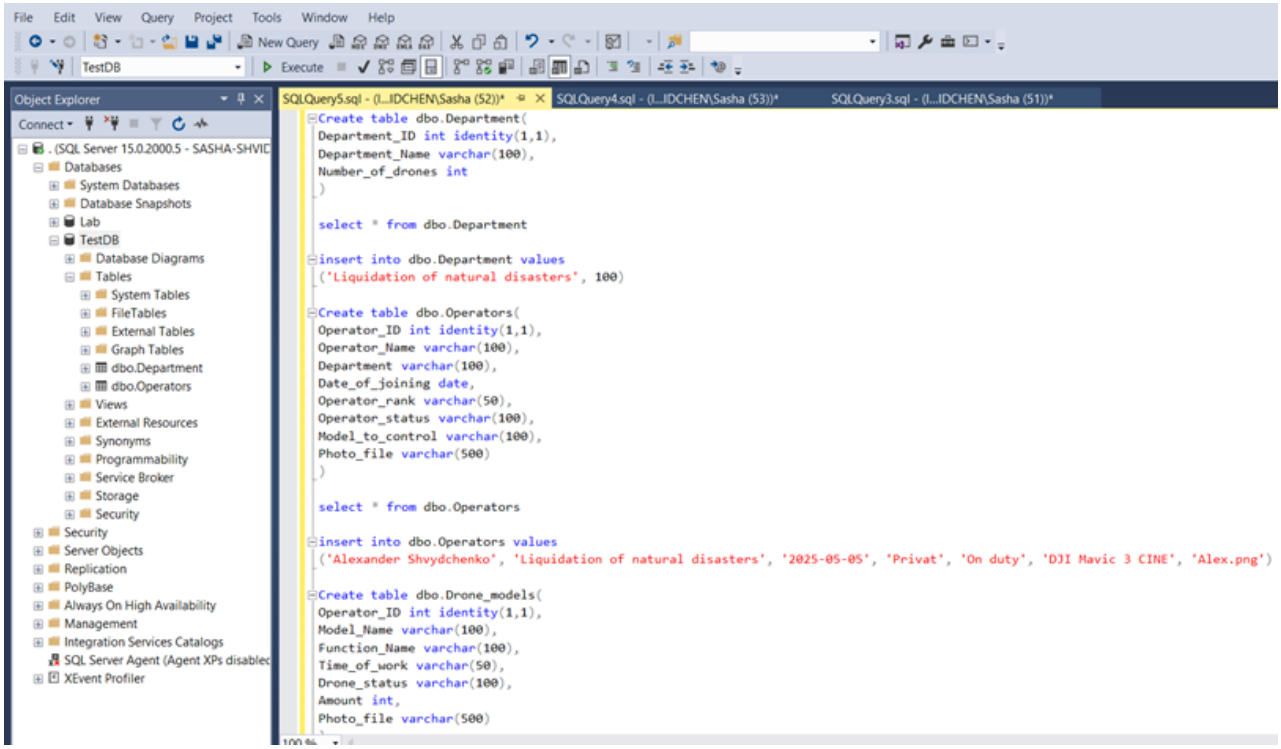


Рисунок 3.10 – Створення таблиць тестової бази даних

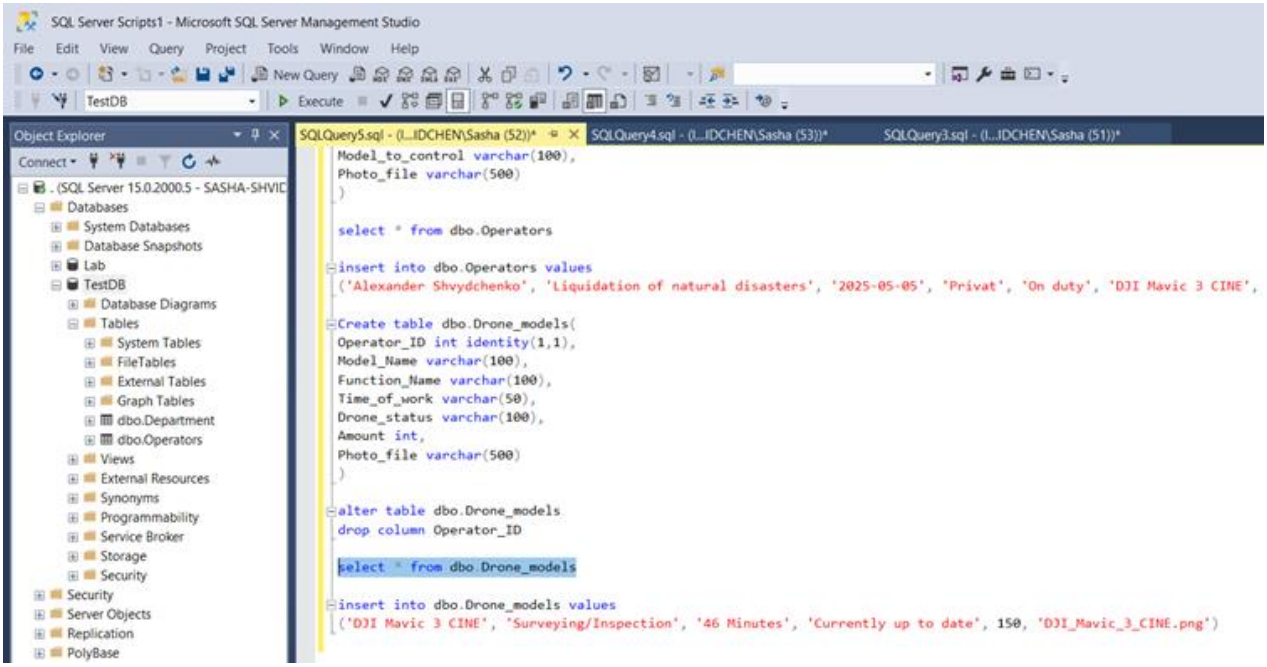


Рисунок 3.11 – Створення таблиць тестової бази даних

В результаті роботи, нові таблиці мали наступний вигляд (таб. 3.3 – 3.5).

Таблиця 3.3 – “Department”

Назва поля	Тип даних	Призначення
1	2	3
Department_ID	int identity(1,1)	Кожен департамент ДСНС в системі має свій унікальний ID.
Department_Name	varchar(100)	Назва департаменту.
Number_of_drones	int	Кількості дронів на службі кожного департаменту.

Таблиця 3.4 – “Operators”

Назва поля	Тип даних	Призначення
1	2	3

Продовження таблиці 3.4

1	2	3
Operator_ID	int identity(1,1)	Кожен оператор дронів на службі ДСНС має свій унікальний ID в системі.
Operator_Name	varchar(100)	Ім’я оператору.
Department	varchar	Департамент, в якому працює оператор.
Date_of_joining	date	Дата надходження на службу.
Operator_rank	varchar(50)	Ранг оператора.
Operator_status	varchar(100)	Статус/стан оператора.
Model_to_control	varchar(100)	Модель дрона, якою здатен керувати оператор.
Photo_file	varchar(500)	Фото оператора.

Таблиця 3.5 – “Drone_models”

Назва поля	Тип даних	Призначення
1	2	3
Model_ID	int identity(1,1)	Кожна модель дронів, що знаходиться на службі ДСНС, має свій власній ID.
Model_Name	varchar(100)	Назва моделі.
Function_Name	varchar(100)	Функція, що виконується дроном.
Time_of_work	varchar(50)	Час роботи дрону, при повному заряді акумулятору.
Drone_status	varchar(50)	Актуальність/застарілість моделі.
Amount	int	Кількість моделей в розпорядженні ДСНС.

Продовження таблиці 3.5

1	2	3
Photo_file	varchar(500)	Фото моделі дрону.

Дані таблиці стали центральними в подальшій роботі.

3.1.3 Створення програмних методів для роботи з базою даних та їх тестування

Наступний етап роботи передбачав роботу в середовищі програмування Visual Studio 2022. Після створення тестової бази даних та основних таблиць, завдання було у створенні методів для роботи з нашою БД та для подальшої роботи над системою.

Даний етап складається з трьох етапів:

1) Підготовка середовища.
2) Розробка методів.
3) Тестування.

Підготовка середовища включає в себе перевірку актуальності версії ПЗ та роботу функцій з якими ми маємо працювати. В процесі була виявлена необхідність оновлення ПЗ Visual Studio 2019 до новішої версії Visual Studio 2022 через його застарілість та некоректну роботу необхідних в роботі функцій. Після встановлення нової версії необхідного ПЗ, було проведено ряд перевірок (рис. 3.12 – 3.15), аби бути певними що ПЗ підходить для виконання завдання:

- Перевірка наявності необхідного формату (рис. 3.12).
- Перевірка можливості створення контролерів (рис. 3.13).
- Перевірка роботи наявних методів в Visual Studio (рис. 3.14 – 3.16).

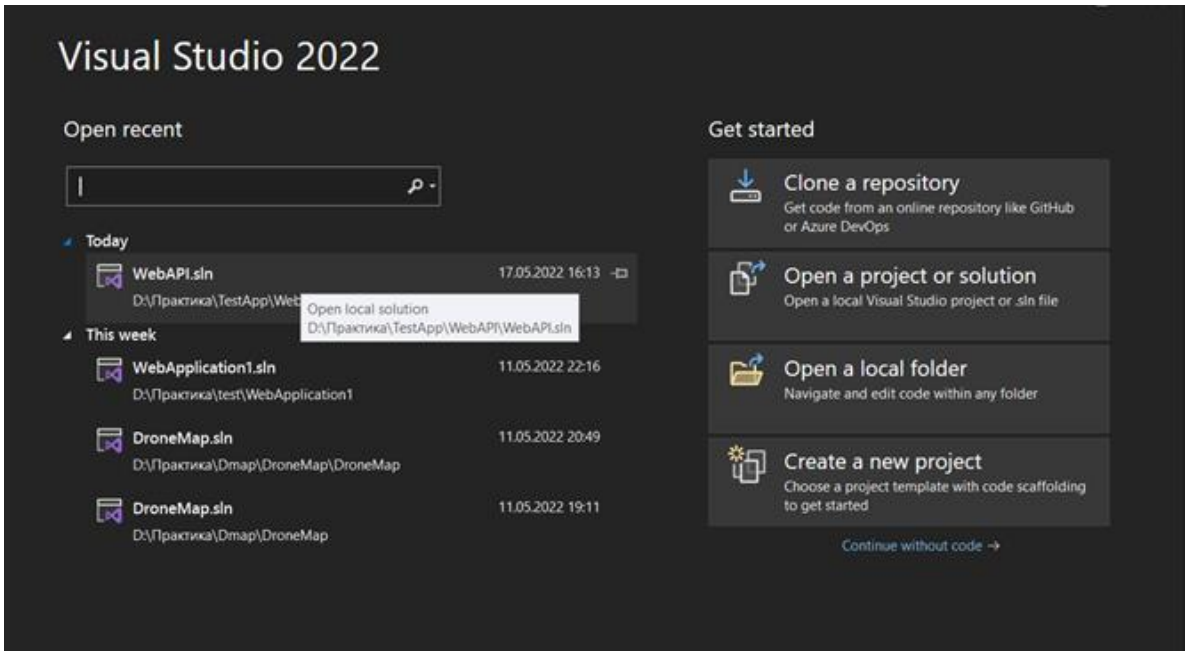


Рисунок 3.12 – Відкриття створеного проекту типу WebAPI

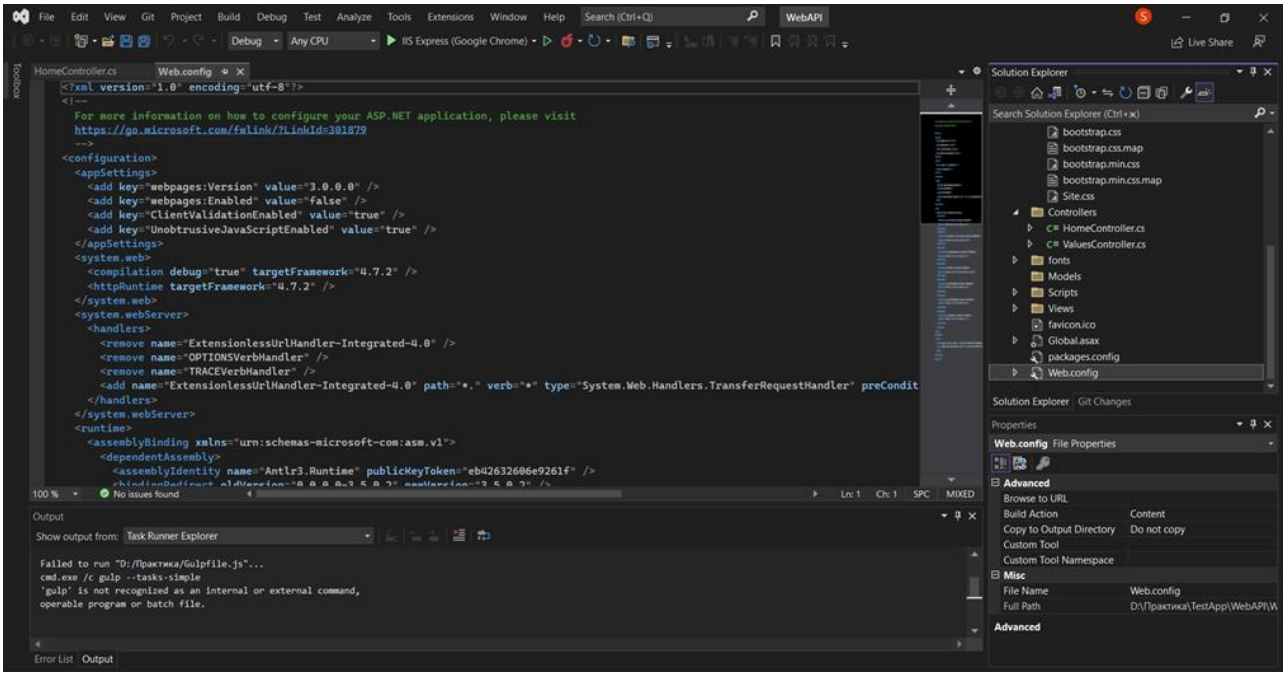


Рисунок 3.13 – Перегляд вікна Web.config

В даній версії ПЗ, функція створення контролерів працює коректно та не видає помилок, одже одна з проблем, з якою ми зіштовхнулись на початку роботи виправлена. В якості наступного етапу перевірки була перевірка роботи методів, створених в Visual Studio 2022. Для цього був використано метод IEnumerable, що має повертати нам значення "value1" та "value2" в веб браузері (рис. 3.14 – 3.16).

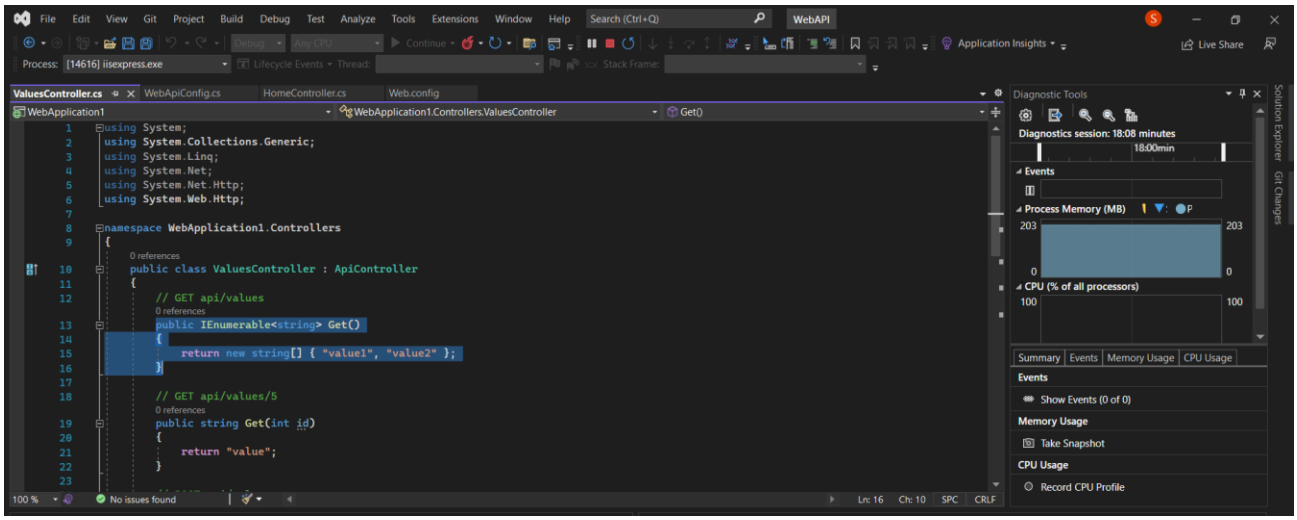


Рисунок 3.14 – Запуск нашого проекту WebAPI

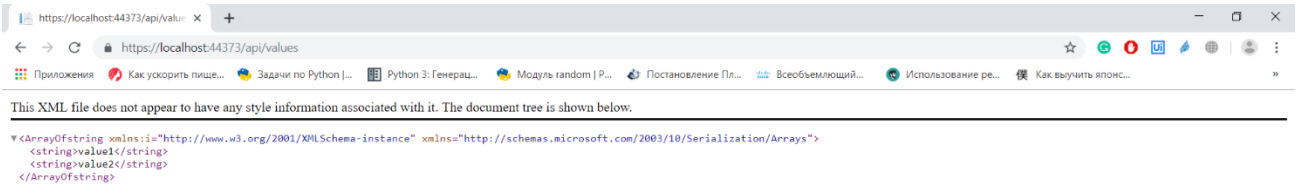


Рисунок 3.15 – Результат виклику значень методу IEnumerable через браузер

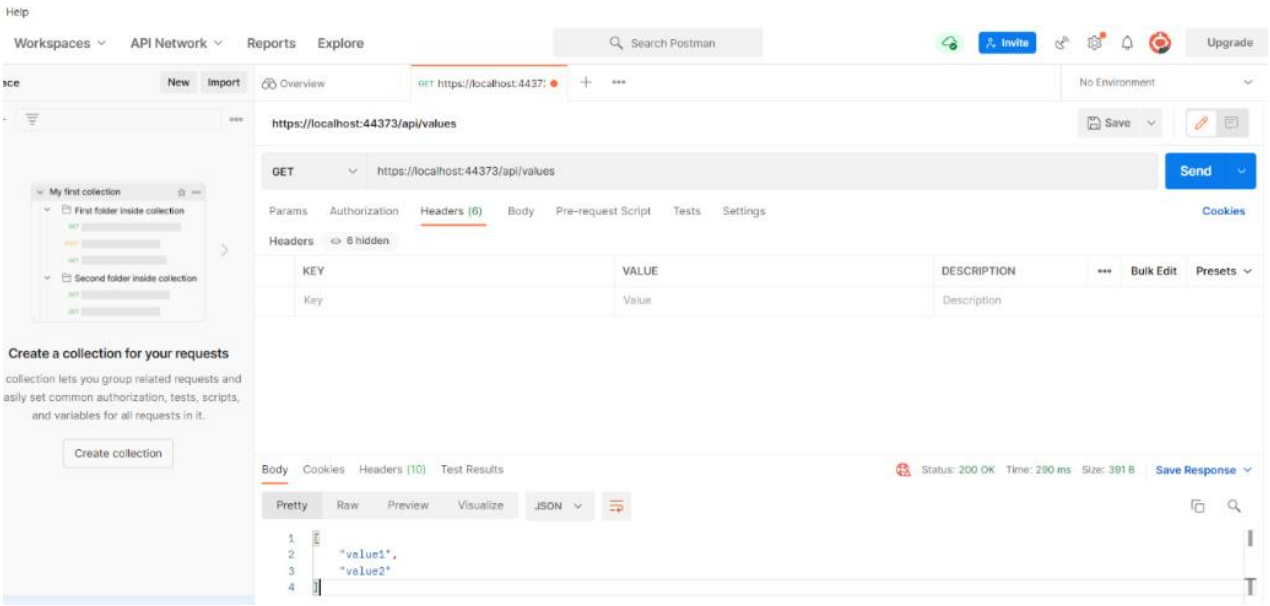


Рисунок 3.16 – Результат виклику методу IEnumerable через “Postmen”

На даному етапі було перевірено роботу усіх необхідних нам можливостей та функцій Visual Studio 2022, отже можна перейти до наступного етапу, програмної реалізації методів.

В першу чергу необхідно встановити зв’язок між проектом WebAPI та локальним SQL сервером. Даний зв’язок прописаний та встановлений у вікні Web.config через функцію connectionStrings (рис. 3.17).

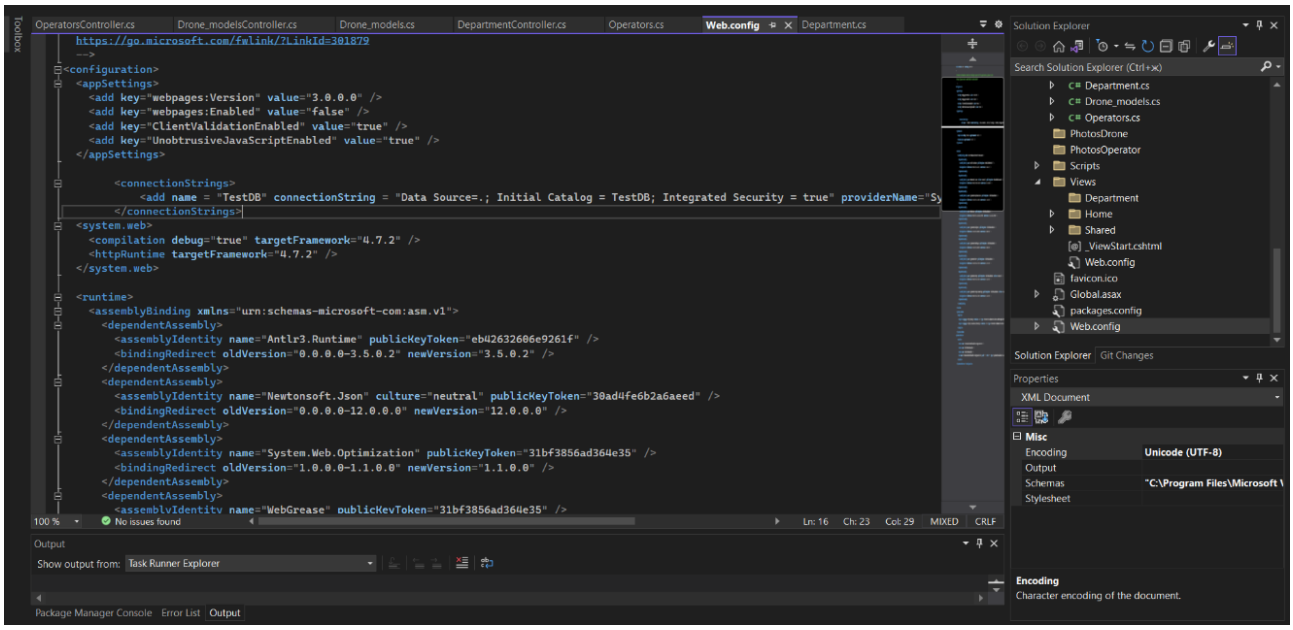


Рисунок 3.17 – Вікно Web.config

<connectionStrings>

<add name = "TestDB" connectionString = "Data Source=.; Initial Catalog = TestDB; Integrated Security = true" providerName="System.Data.SqlClient"/>

</connectionStrings>

Після встановлення зв’язку із локальним SQL сервером, необхідно провести перевірку доступу до створеної бази даних. Для процедури перевірки використано ПЗ “Postman” та додано в таблиці інформацію заздалегідь, або перевірити чи можливо отримати до неї доступ (рис. 3.18).

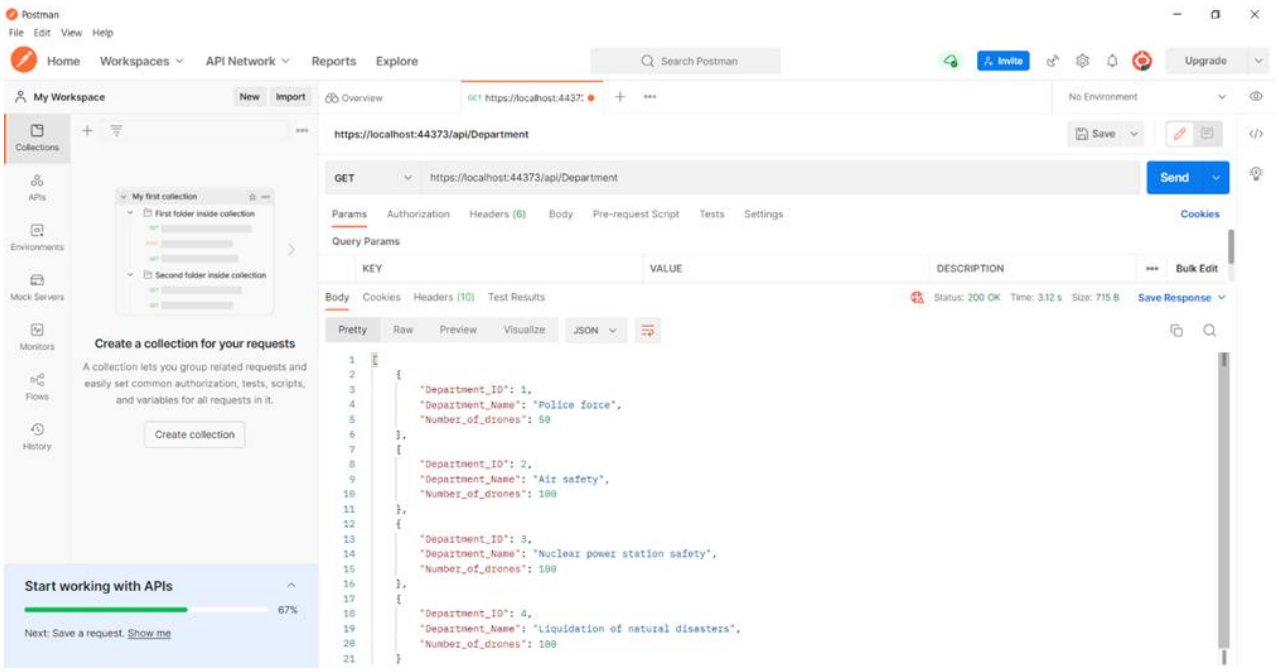


Рисунок 3.18 – Результат перевірки зв’язку із сервером

Після введення запиту на отримання інформації з таблиці “Department” ми отримали інформацію про департаменти, що є частиною нашої системи. Перевірка зв’язку із сервером є вдалою.

Наступним етапом було створення моделей для кожної з таблиць бази даних що ми використовуємо, щоб ми могли застосовувати контролери. Було створено наступні моделі:

- Department.cs (рис. 3.19).
- Drone_models.cs (рис 3.20).
- Operators.cs (рис. 3.21).

Код для моделей наведено окремо в додатку А.

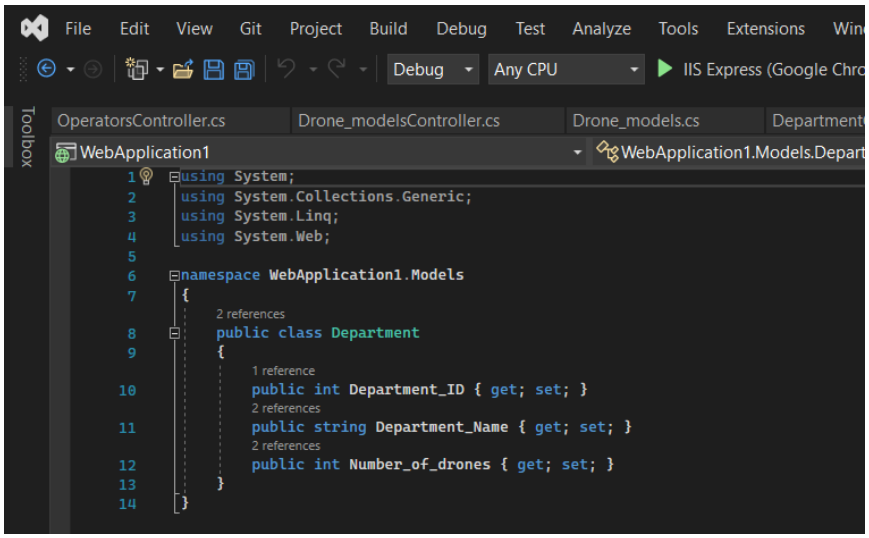


Рисунок 3.19 – Модель для таблиці Департаментів

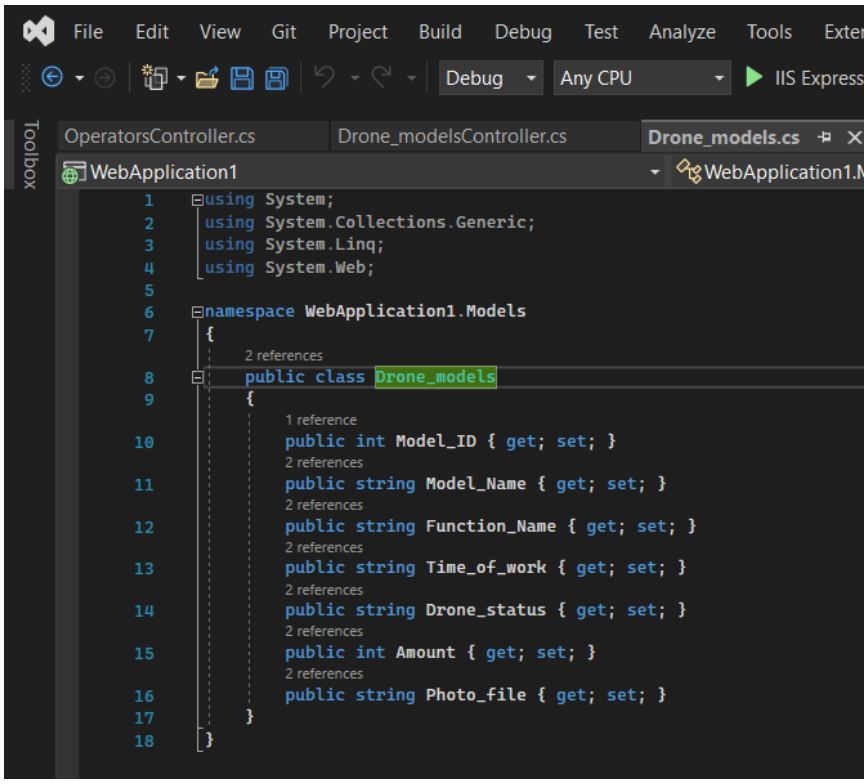


Рисунок 3.20 – Модель для таблиці моделей дронів

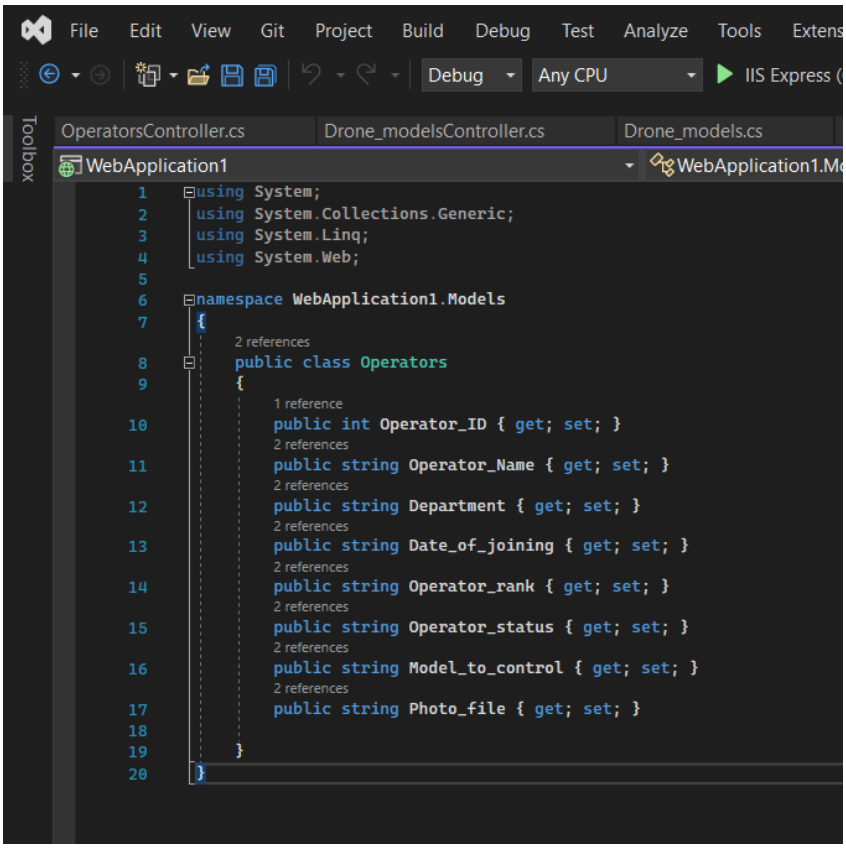


Рисунок 3.21 – Модель для талиці Операторів

Створені моделі є важливим елементом роботи програмної частини, так як вони дозволятимуть методам взаємодіяти із полями таблиць, отже кожна окрема модель має відповідати полям таблиці, до якої вона відноситься. Без них, взаємодія методів з базою даних неможлива.

Наступним етапом є створення самих методів, які будуть взаємодіяти з самою базою даних (рис. 3.22 – 3.25). Код наведено окремо в додатку Б.

Розроблені методі взаємодіють із кожним окремим пунктом таблиці, до яких вони прив’язані. Метод “Get” звертається до існуючих елементів таблиці та виводить їх на екран, метод “Post” додає значення в пусті клітини таблиці, окрім значень ID, так як воно додається автоматично при створенні нових записів, метод “Put” звертається до існуючих записів та редагує їх, метод “Delete” в свою чергу вишукує запис згідно його ID та видаляє (рис. 3.22 – 3.25).

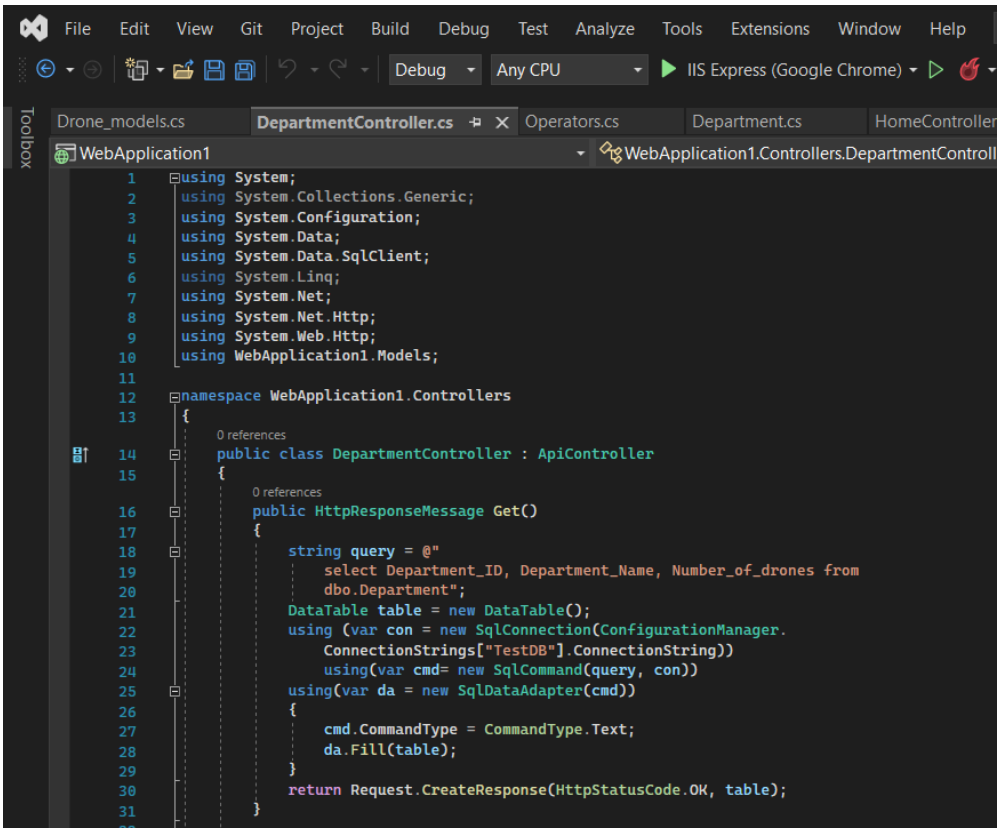


Рисунок 3.22 – Написання методу Get

Рисунок 3.23 – Написання методу Post

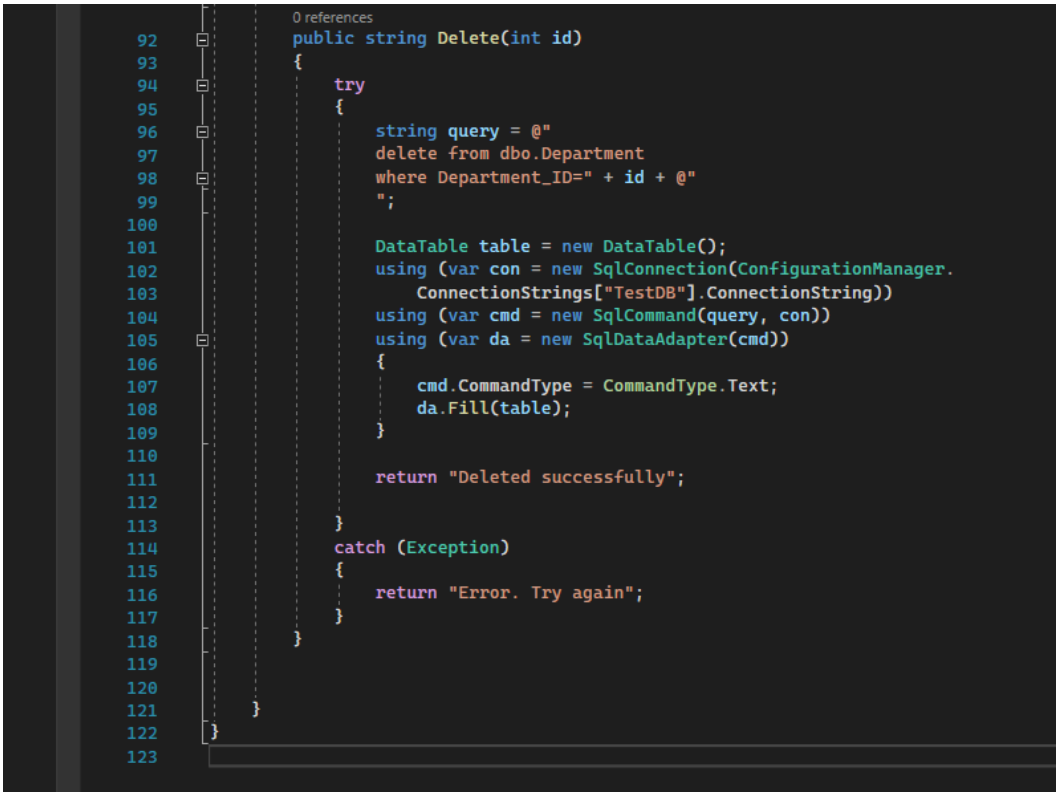
```
0 references
public string Put(Department dep)
{
    try
    {
        string query = @"
        update dbo.Department set
        Department_Name='" + dep.Department_Name + @"',
        Number_of_drones='" + dep.Number_of_drones + @"'
        where Department_ID=" + dep.Department_ID + @"
        ";

        DataTable table = new DataTable();
        using (var con = new SqlConnection(ConfigurationManager.
            ConnectionStrings["TestDB"].ConnectionString))
        using (var cmd = new SqlCommand(query, con))
        using (var da = new SqlDataAdapter(cmd))
        {
            cmd.CommandType = CommandType.Text;
            da.Fill(table);
        }

        return "Updated successfully";

    }
    catch (Exception)
    {
        return "Error. Try again";
    }
}
```

Рисунок 3.24 – Написання методу Put



```
0 references
public string Delete(int id)
{
    try
    {
        string query = @"
        delete from dbo.Department
        where Department_ID=" + id + @"
        ";

        DataTable table = new DataTable();
        using (var con = new SqlConnection(ConfigurationManager.
            ConnectionStrings["TestDB"].ConnectionString))
        using (var cmd = new SqlCommand(query, con))
        using (var da = new SqlDataAdapter(cmd))
        {
            cmd.CommandType = CommandType.Text;
            da.Fill(table);
        }

        return "Deleted successfully";

    }
    catch (Exception)
    {
        return "Error. Try again";
    }
}


}
}
```

Рисунок 3.25 - Написання методу Delete

Після написання методів була проведена серія тестів, аби дізнатись чи працюють вони так як необхідно та чи є помилки в написанні коду. Тести були проведені за допомогою ПЗ “Postman” (рис. 3.26 – 3.32).

1) Тестування методу “GET”

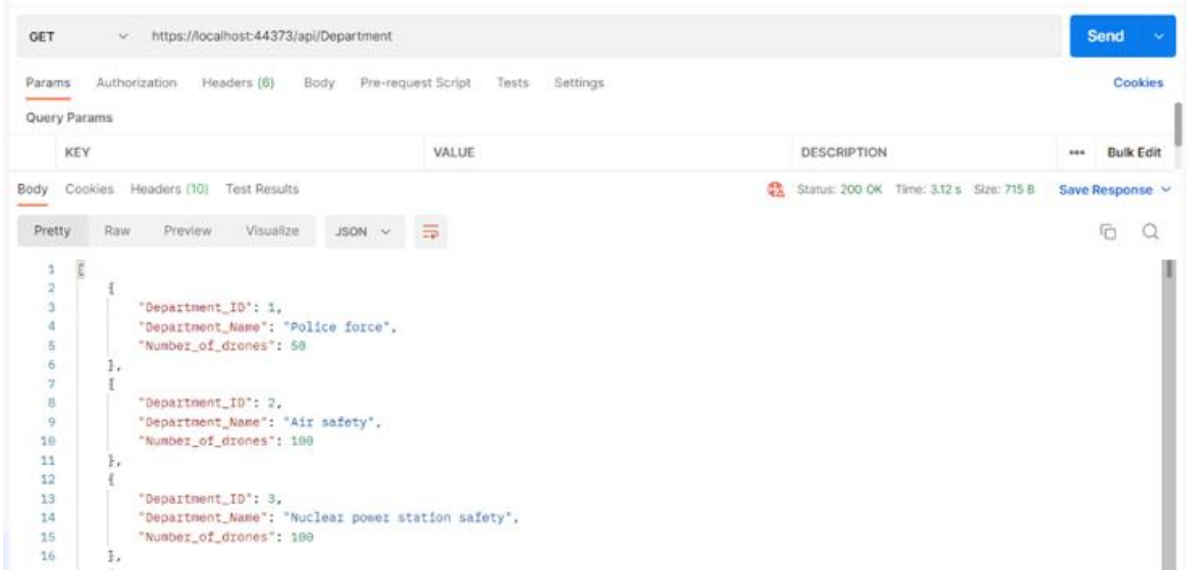


Рисунок 3.26 – Результат тестування методу “GET”

Тестування методу “GET” пройшло вдало з першої спроби. Метод вивів на екран занесену для тесту інформацію.

2) Тестування методу “POST”

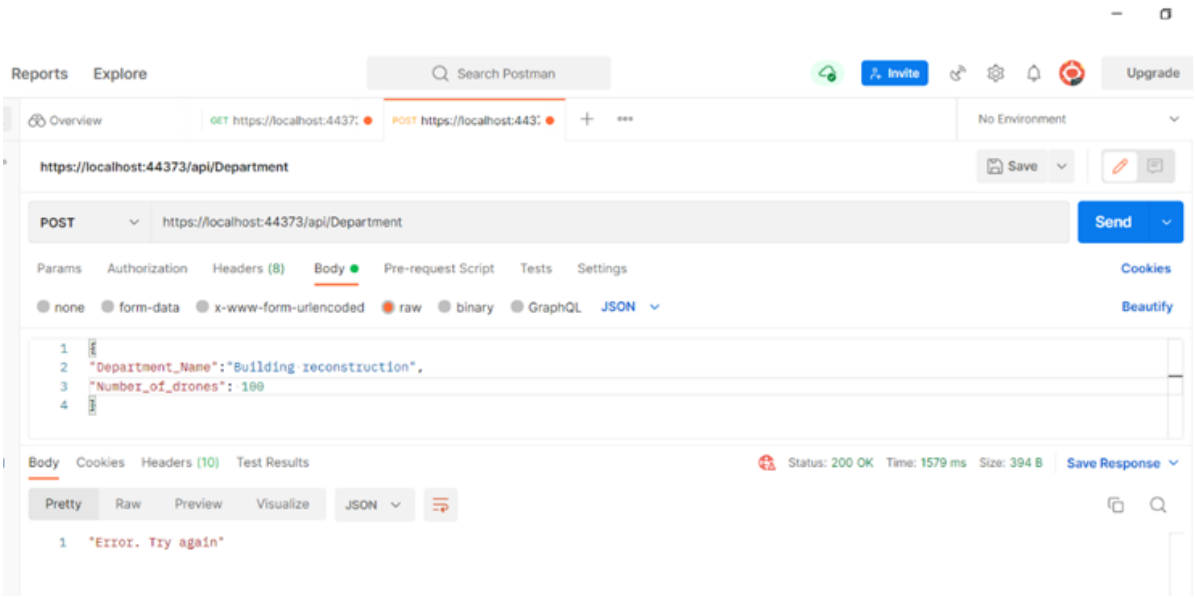


Рисунок 3.27 – Результат тестування методу “POST”, перша спроба

Перша спроба тестування методу “POST” була невдала (рис. 3.27). Виходячи з того, що тест метод “GET” працює, проблема не може бути у зв’язку із сервером і помилку необхідно шукати в коді методу. Як результат була знайдена помилка, а саме пропущений символ “,”. Після виправлення було здійснено другу спробу.

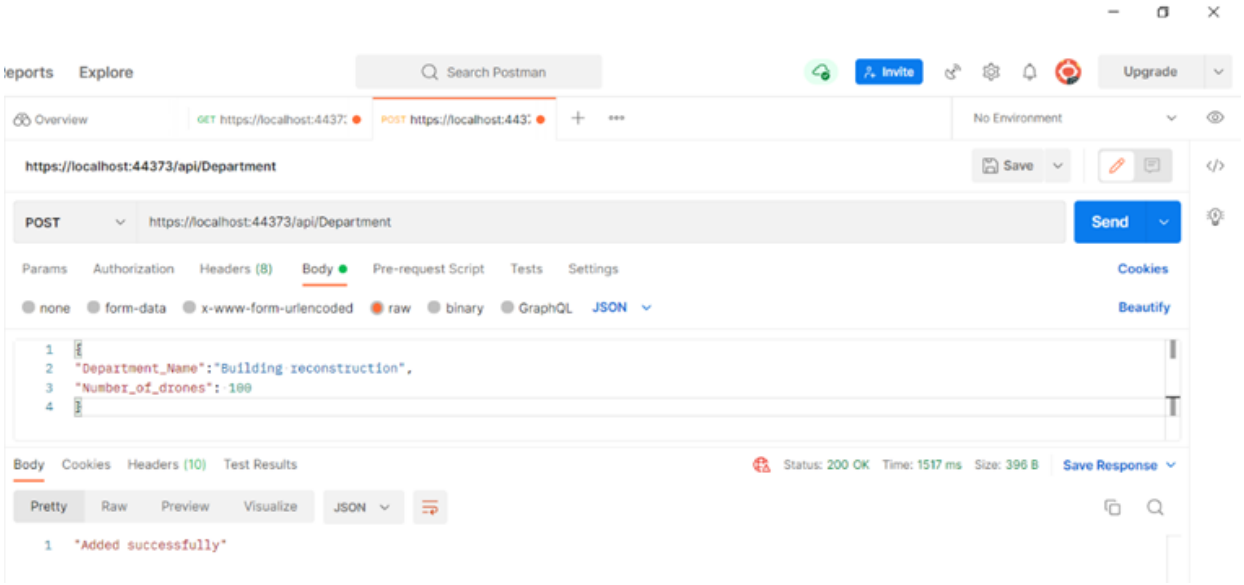


Рисунок 3.28 – Результат тестування методу “POST”, друга спроба

Виправлення даної помилки виправило ситуацію, тестування методу “POST” пройшло вдало (рис. 3.28). Тепер можна додавати нові дані в базу даних не через сервер.

3) Тестування методу “PUT”

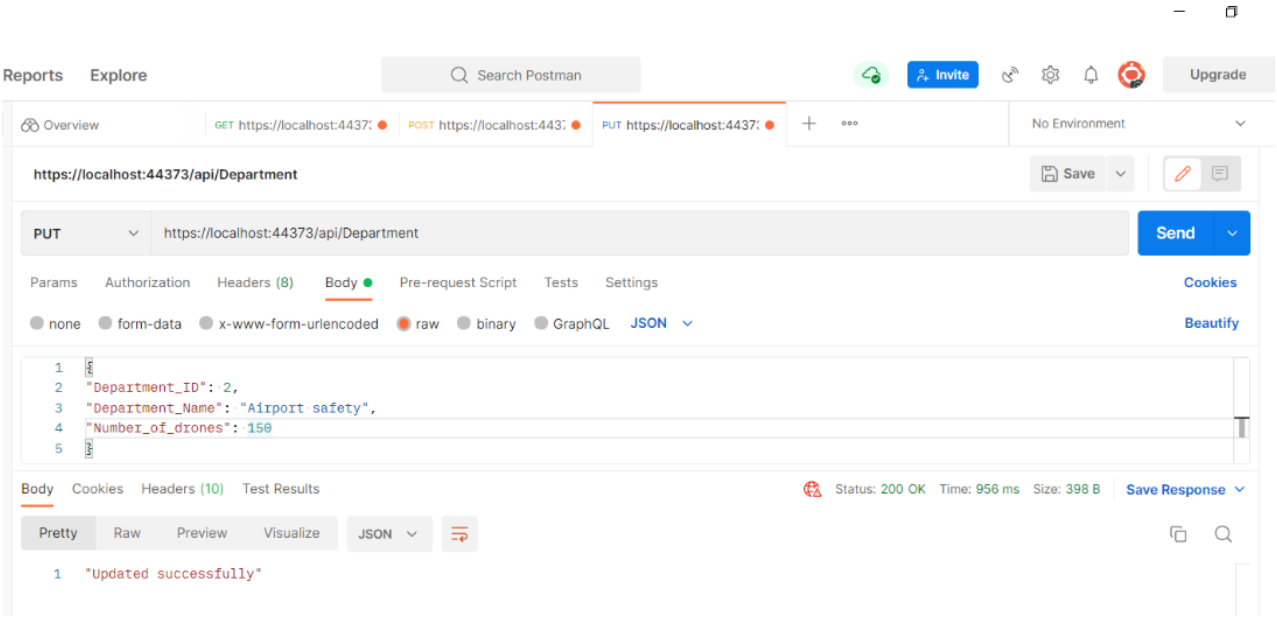


Рисунок 3.29 – Тестування методу редагування “PUT”

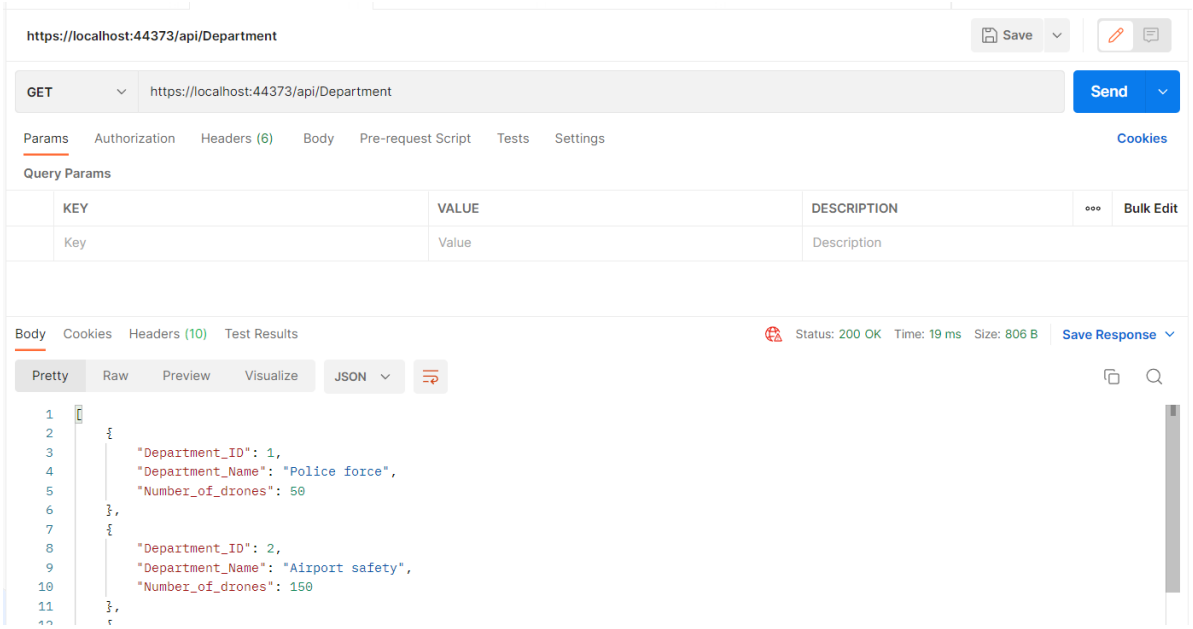


Рисунок 3.30 – Перевірка внесених змін

Тестування методу “PUT” пройшло вдало, отже тепер є можливість редагувати інформацію в базі даних (рис. 3.29 – 3.30).

4) Тестування методу “DELETE”

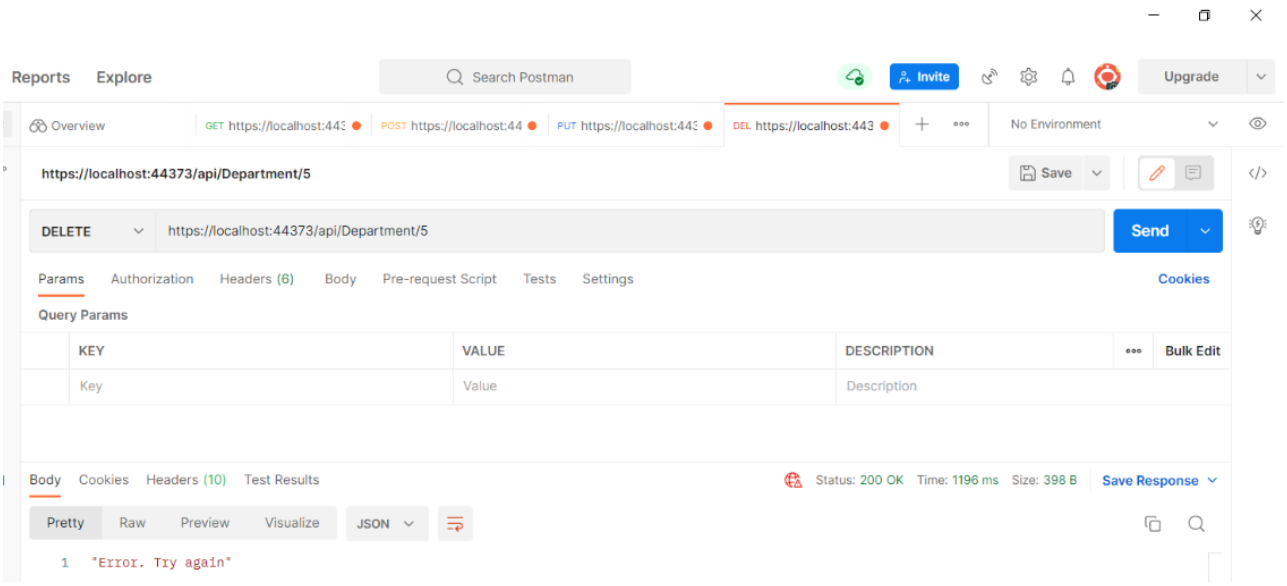


Рисунок 3.31 – Результат тестування методу “DELETE”, перша спроба

Перша спроба тестування методу “DELETE” була невдала (рис. 3.31). Після пошуків помилки в коді методу було виявлено дві помилки, а саме:

- Зайве використання “set”.
- Відсутня умова “Where”.

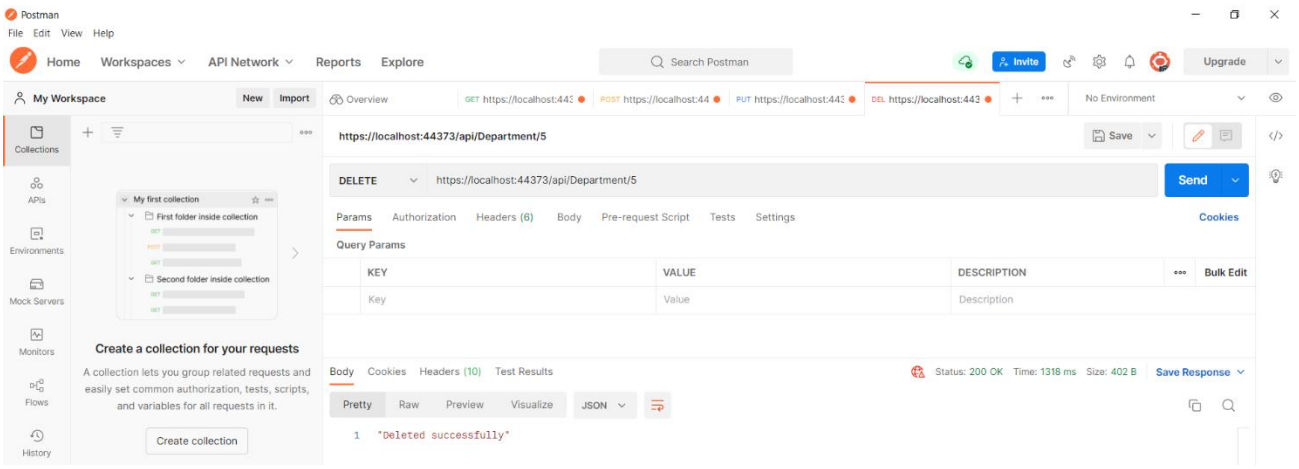


Рисунок 3.32 – Результат тестування методу “DELETE”, друга спроба

Виправлення даних помилок дозволило отримати необхідний результат. Тестування методу “DELETE” пройшло вдало, отже є можливість видаляти зайву інформацію з бази даних (рис. 3.32).

Результатом проведеної роботи є програмна реалізація можливих взаємодії користувача зі створеною базою даних.

3.2 Висновки до розділу 3

Даний розділ описує процес проектування системи, проектування та технічної реалізації її бази даних та розробки бекенд методів для подальших робіт з нею.

В ході робіт над проектуванням системи були створені діаграми, що пояснюють логіку системи та як саме вона працює, які компоненти в неї входять, як вони працюють та яким чином вони співпрацюють один з одним, що надає максимально можливе розуміння її роботи, що в свою чергу, дозволило перейти до проектування бази даних для системи.

Спроектована та програмно реалізована база даних є важливою частиною системи та дозволятиме зберігати необхідну інформацію для ефективної роботи системи. Робота над нею буде продовжуватись, і для цього, як і для можливості взаємодії з нею, були розроблені та протестовані бекенд-методи для взаємодії з нею.

ВИСНОВКИ

В ході роботи над даною кваліфікаційною роботою бакалавра було спроектовано систему автоматизованого реагування на надзвичайні ситуації за допомогою дронів, спроектовано базу даних системи, розроблено та протестовані програмні методі для роботи з нею.

Під час виконання поставлених завдань також були досягнуті наступні результати:

- Розглянуто актуальність задачі. Надзвичайні ситуації, будь то природні чи техногенні катастрофи, будуть існувати як явище та траплятись ще багато років, при цьому рівень небезпеки що вони становлять, зменшуватись не буде, тому людству необхідна система, що здатна нейтралізувати їх без знаходження людей в зоні небезпеки.
- Описано об’єкт дослідження. Проект системи автоматизованого реагування на надзвичайні ситуації представляє собою систему боротьби з НС за допомогою технологій IoT та БПЛА.
- Досліджено та розглянуто системи зі схожою моделлю роботи, а саме система автоматизованої доставки Amazon Prime Air та система автоматичного обприскування від “BASF”. Також було досліджено систему екстреної евакуації на базі технологій інтернету речей. Дані системи слугували орієнтиром при подальшій роботи.
- На основі матеріалів про аналогічні та схожі за принципом роботи технологій, було визначено ключові компоненти системи. В подальшому було досліджено принцип роботи та роль в системі кожного з них.
- На основі досліджених матеріалів та з урахуванням особливостей компонентів системи, була побудована її функціональна модель та спроектовано принцип її роботи та роботи її компонентів за допомогою діаграм.
- Були спроектовані логічна та фізична бази даних системи.

- Для можливості взаємодії зі створеною базою даних, були розроблені та протестовані програмні методи що дозволяють проводити необхідні маніпуляції з інформацією та слугують основою для створення веб-інтерфейсу для її зручного використання.

Поставлена задача виконана, мета кваліфікаційної роботи бакалавра досягнута.

СПИСКИ ВИКОРИСТАНИХ ДЖЕРЕЛ

1. An IoT-based Emergency Evacuation System [Електронний ресурс] / Режим доступу: https://www.researchgate.net/publication/339096717_An_IoT-based_Emergency_Evacuation_System
2. Development in building fire detection and evacuation system-a comprehensive review [Електронний ресурс] / Режим доступу: https://www.researchgate.net/publication/343179923_Development_in_building_fire_detection_and_evacuation_system-a_comprehensive_review
3. Efficient Smart Emergency Response System for Fire Hazards using IoT [Електронний ресурс] / Режим доступу: https://www.researchgate.net/publication/322881482_Efficient_Smart_Emergency_Response_System_for_Fire_Hazards_using_IoT
4. Презентація дрону Agras T16 [Електронний ресурс] / Режим доступу: https://www.youtube.com/watch?v=Dn7Q4oIrP3I&ab_channel=DJI
5. Презентація проекту Amazon Prime Air [Електронний ресурс] / Режим доступу: https://www.youtube.com/watch?v=vNySOrI2Ny8&list=LL&index=11&t=8s&ab_channel=amazon
6. Smart Agriculture with Drones: Experiences from Latin America [Електронний ресурс] / Режим доступу: https://agriculture.basf.com/global/en/media/featured-stories/drones-in-agriculture.html?at_medium=sl&at_campaign=AP_BAW_GLOB_EN_none_TRA_DronesInAgriculture&at_term=drones%20in%20agriculture&at_creation=Search_Google_Search_DronesInAgGeneral&at_platform=google&at_variant=DronesInAgGeneral
7. Basf – Ready For Takeoff: How Drones Bring More Safety To Farming [Електронний ресурс] / Режим доступу:

https://agriculture.basf.com/global/en/media/featured-stories/drones-in-agriculture.html?at_medium=sl&at_campaign=AP_BAW_GLOB_EN_none_TRA_DronesInAgriculture&at_term=drones%20in%20agriculture&at_creation=Search_Google_Search_DronesInAgGeneral&at_platform=google&at_variant=DronesInAgGeneral

8. Розробка квадрокоптера [Електронний ресурс] / Режим доступу: https://klona.ru/blog/promyshlennyy-dizayn/razrabotka-kvadrokoptera-idei-i-primeri
9. Види та типи дронів [Електронний ресурс] / Режим доступу: https://www.computer.org/publications/tech-news/research/flying-iot-toward-low-power-vision-sky
10. Iotworldtoday – Drone Technology Extends Reach of Mobile IoT [Електронний ресурс] / Режим доступу: https://www.iotworldtoday.com/2021/01/05/drone-technology-extends-reach-of-mobile-iot/
11. Robotics: Aerial Robotics [Електронний ресурс] / Режим доступу: https://www.coursera.org/learn/robotics-flight/lecture/XguwZ/quadrotors
12. Аеропорт для дронів, Наталья Максюк [Електронний ресурс] / Режим доступу: https://bzh.life/ua/gorod/v-kieve-planiruyut-postroit-aeroport-dlya-dronov-chto-tam-budet/
13. Як може літати квадрокоптер [Електронний ресурс] / Режим доступу: https://coptermarket.by/kak-upravliat-dronom
14. Drone ground station Wing GCS [Електронний ресурс] / Режим доступу: https://www.aeroexpo.online/prod/worthington-sharpe/product-185931-28625.html
15. Українські військові вчаться керувати турецькими бойовими дронами Bayraktar [Електронний ресурс] / Режим доступу: https://www.ukrinform.ua/rubric-ato/2773318-vijskovi-zsu-vcatsa-keruvati-tureckimi-bojovimi-dronami-bayraktar.html

16. How is drone communicate with UAV ground stations [Електронний ресурс] / Режим доступу: http://www.dronefromchina.com/new/drone-communicate-with-UAV-ground-stations.html

ДОДАТОК А

Код моделей

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication1.Models
{
    public class Department
    {
        public int Department_ID { get; set; }
        public string Department_Name { get; set; }
        public int Number_of_drones { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication1.Models
{
    public class Drone_models
    {
        public int Model_ID { get; set; }
        public string Model_Name { get; set; }
        public string Function_Name { get; set; }
        public string Time_of_work { get; set; }
        public string Drone_status { get; set; }
        public int Amount { get; set; }
        public string Photo_file { get; set; }
    }
}
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace WebApplication1.Models
{
    public class Operators
    {
        public int Operator_ID { get; set; }
        public string Operator_Name { get; set; }
        public string Department { get; set; }
        public string Date_of_joining { get; set; }
        public string Operator_rank { get; set; }
        public string Operator_status { get; set; }
        public string Model_to_control { get; set; }
        public string Photo_file { get; set; }

    }
}
```

ДОДАТОК Б

Код контролерів

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using WebApplication1.Models;

namespace WebApplication1.Controllers
{
    public class DepartmentController : ApiController
    {
        public HttpResponseMessage Get()
        {
            string query = @"
                select Department_ID, Department_Name, Number_of_drones from
                dbo.Department";
            DataTable table = new DataTable();
            using (var con = new SqlConnection(ConfigurationManager.
                ConnectionStrings["TestDB"].ConnectionString))
                using(var cmd= new SqlCommand(query, con))
            using(var da = new SqlDataAdapter(cmd))
            {
                cmd.CommandType = CommandType.Text;
                da.Fill(table);
            }
            return Request.CreateResponse(HttpStatusCode.OK, table);
        }

        public string Post(Department dep)
        {
            try
            {
                string query = @"
                insert into dbo.Department values
                (
                '"+ dep.Department_Name+ @"',
                '"+ dep.Number_of_drones+ @"'
                )";

                DataTable table = new DataTable();
                using (var con = new SqlConnection(ConfigurationManager.
                    ConnectionStrings["TestDB"].ConnectionString))
                using (var cmd = new SqlCommand(query, con))
                using (var da = new SqlDataAdapter(cmd))
                {
                    cmd.CommandType = CommandType.Text;
                    da.Fill(table);
                }

                return "Added successfully";

            }
            catch (Exception)
            {
                return "Error. Try again";
```

```
            }
        }

        public string Put(Department dep)
        {
            try
            {
                string query = @"
                update dbo.Department set
                Department_Name='" + dep.Department_Name + @"',
                Number_of_drones='" + dep.Number_of_drones + @"'
                where Department_ID=" + dep.Department_ID + @"
                ";

                DataTable table = new DataTable();
                using (var con = new SqlConnection(ConfigurationManager.
                    ConnectionStrings["TestDB"].ConnectionString))
                using (var cmd = new SqlCommand(query, con))
                using (var da = new SqlDataAdapter(cmd))
                {
                    cmd.CommandType = CommandType.Text;
                    da.Fill(table);
                }

                return "Updated successfully";

            }
            catch (Exception)
            {
                return "Error. Try again";
            }
        }
        public string Delete(int id)
        {
            try
            {
                string query = @"
                delete from dbo.Department
                where Department_ID=" + id + @"
                ";

                DataTable table = new DataTable();
                using (var con = new SqlConnection(ConfigurationManager.
                    ConnectionStrings["TestDB"].ConnectionString))
                using (var cmd = new SqlCommand(query, con))
                using (var da = new SqlDataAdapter(cmd))
                {
                    cmd.CommandType = CommandType.Text;
                    da.Fill(table);
                }

                return "Deleted successfully";

            }
            catch (Exception)
            {
                return "Error. Try again";
            }
        }


    }
}
```

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using WebApplication1.Models;

namespace WebApplication1.Controllers
{
    public class Drone_modelsController : ApiController
    {
        public HttpResponseMessage Get()
        {
            string query = @"
                select Model_ID, Model_Name, Function_Name, Time_of_work, Drone_status,
Amount, Photo_file from
                dbo.Drone_models";
            DataTable table = new DataTable();
            using (var con = new SqlConnection(ConfigurationManager.
                ConnectionStrings["TestDB"].ConnectionString))
            using (var cmd = new SqlCommand(query, con))
            using (var da = new SqlDataAdapter(cmd))
            {
                cmd.CommandType = CommandType.Text;
                da.Fill(table);
            }
            return Request.CreateResponse(HttpStatusCode.OK, table);
        }

        public string Post(Drone_models dr)
        {
            try
            {
                string query = @"
                insert into dbo.Drone_models values
                (
                '" + dr.Model_Name + @"',
                '" + dr.Function_Name + @"',
                '" + dr.Time_of_work + @"',
                '" + dr.Drone_status + @"',
                '" + dr.Amount + @"',
                '" + dr.Photo_file + @"'
                )";

                DataTable table = new DataTable();
                using (var con = new SqlConnection(ConfigurationManager.
                    ConnectionStrings["TestDB"].ConnectionString))
                using (var cmd = new SqlCommand(query, con))
                using (var da = new SqlDataAdapter(cmd))
                {
                    cmd.CommandType = CommandType.Text;
                    da.Fill(table);
                }

                return "Added successfully";

            }
```

```
    catch (Exception)
    {
        return "Error. Try again";
    }
}

public string Put(Drone_models dr)
{
    try
    {
        string query = @"
        update dbo.Drone_models set
        Model_Name='" + dr.Model_Name + @"',
        Function_Name='" + dr.Function_Name + @"',
        Time_of_work='" + dr.Time_of_work + @"',
        Drone_status='" + dr.Drone_status + @"',
        Amount='" + dr.Amount + @"',
        Photo_file='" + dr.Photo_file + @"'
        where Model_ID=" + dr.Model_ID + @"
        ";

        DataTable table = new DataTable();
        using (var con = new SqlConnection(ConfigurationManager.
            ConnectionStrings["TestDB"].ConnectionString))
        using (var cmd = new SqlCommand(query, con))
        using (var da = new SqlDataAdapter(cmd))
        {
            cmd.CommandType = CommandType.Text;
            da.Fill(table);
        }

        return "Updated successfully";

    }
    catch (Exception)
    {
        return "Error. Try again";
    }
}
public string Delete(int id)
{
    try
    {
        string query = @"
        delete from dbo.Drone_models
        where Model_ID=" + id + @"
        ";

        DataTable table = new DataTable();
        using (var con = new SqlConnection(ConfigurationManager.
            ConnectionStrings["TestDB"].ConnectionString))
        using (var cmd = new SqlCommand(query, con))
        using (var da = new SqlDataAdapter(cmd))
        {
            cmd.CommandType = CommandType.Text;
            da.Fill(table);
        }

        return "Deleted successfully";

    }
    catch (Exception)
    {
```

```
                return "Error. Try again";
            }
        }
    }
}


using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Web.Http;
using WebApplication1.Models;
using System.Web;

namespace WebApplication1.Controllers
{
    public class OperatorsController : ApiController
    {
        public HttpResponseMessage Get()
        {
            string query = @"
                select Operator_ID, Operator_Name, Department, Date_of_joining, Operator_rank,
Operator_status, Model_to_control, Photo_file from
                dbo.Operators";
            DataTable table = new DataTable();
            using (var con = new SqlConnection(ConfigurationManager.
                ConnectionStrings["TestDB"].ConnectionString))
            using (var cmd = new SqlCommand(query, con))
            using (var da = new SqlDataAdapter(cmd))
            {
                cmd.CommandType = CommandType.Text;
                da.Fill(table);
            }
            return Request.CreateResponse(HttpStatusCode.OK, table);
        }

        public string Post(Operators op)
        {
            try
            {
                string query = @"
                insert into dbo.Operators values
                (
                '" + op.Operator_Name + @"',
                '" + op.Department + @"',
                '" + op.Date_of_joining + @"',
                '" + op.Operator_rank + @"',
                '" + op.Operator_status + @"',
                '" + op.Model_to_control + @"',
                '" + op.Photo_file + @"'
                )";

                DataTable table = new DataTable();
                using (var con = new SqlConnection(ConfigurationManager.
                    ConnectionStrings["TestDB"].ConnectionString))
                using (var cmd = new SqlCommand(query, con))
                using (var da = new SqlDataAdapter(cmd))
                {
```

```
            cmd.CommandType = CommandType.Text;
            da.Fill(table);
        }

        return "Added successfully";

    }
    catch (Exception)
    {
        return "Error. Try again";
    }
}

public string Put(Operators op)
{
    try
    {
        string query = @"
        update dbo.Operators set
        Operator_Name='" + op.Operator_Name + @"',
        Department='" + op.Department + @"',
        Date_of_joining='" + op.Date_of_joining + @"',
        Operator_rank='" + op.Operator_rank + @"',
        Operator_status='" + op.Operator_status + @"',
        Model_to_control='" + op.Model_to_control + @"',
        Photo_file='" + op.Photo_file + @"'
        where Operator_ID=" + op.Operator_ID + @"
        ";

        DataTable table = new DataTable();
        using (var con = new SqlConnection(ConfigurationManager.
            ConnectionStrings["TestDB"].ConnectionString))
        using (var cmd = new SqlCommand(query, con))
        using (var da = new SqlDataAdapter(cmd))
        {
            cmd.CommandType = CommandType.Text;
            da.Fill(table);
        }

        return "Updated successfully";

    }
    catch (Exception)
    {
        return "Error. Try again";
    }
}
public string Delete(int id)
{
    try
    {
        string query = @"
        delete from dbo.Operators
        where Operator_ID=" + id + @"
        ";

        DataTable table = new DataTable();
        using (var con = new SqlConnection(ConfigurationManager.
            ConnectionStrings["TestDB"].ConnectionString))
        using (var cmd = new SqlCommand(query, con))
        using (var da = new SqlDataAdapter(cmd))
        {
            cmd.CommandType = CommandType.Text;
```

```
                    da.Fill(table);
                }

                return "Deleted successfully";

            }
            catch (Exception)
            {
                return "Error. Try again";
            }
        }
        [Route("api/Operators/SaveFile")]
        public string SaveFile()
        {
            try
            {
                var httpRequest = HttpContext.Current.Request;
                var postedFile = httpRequest.Files[0];
                string filename = postedFile.FileName;
                var physicalPath = HttpContext.Current.Server.MapPath("~/PhotosOperator/" +
filename);
                postedFile.SaveAs(physicalPath);
                return filename;
            }
            catch (Exception)
            {
                return "amogus.png";
            }
        }
    }
}
```

ДОДАТОК В

Скріншоти презентації

Кваліфікаційна робота бакалавра на тему:

Використання IoT - рішень та дронів для запобігання й усунення надзвичайних ситуацій

Керівник: доктор технічних наук, доцент
Кучанський Олександр Юрійович
Виконав: студент групи ІР-41
Швидченко Олександр Дмитрович

Рисунок 1 – Титульний слайд

Проекти – аналоги

Amazon Prime Air

Safetysape

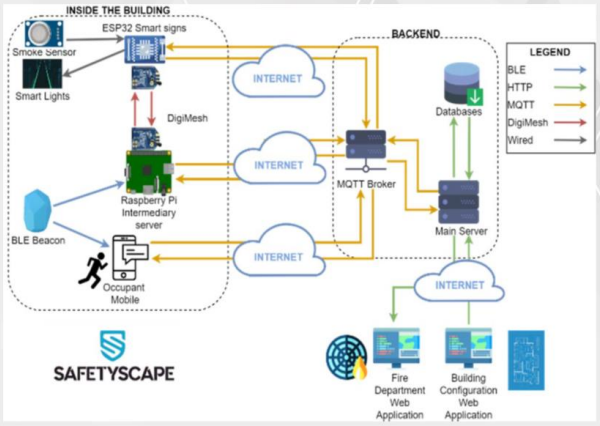


Рисунок 2 – аналоги систем

Рисунок 3 – Складові системи

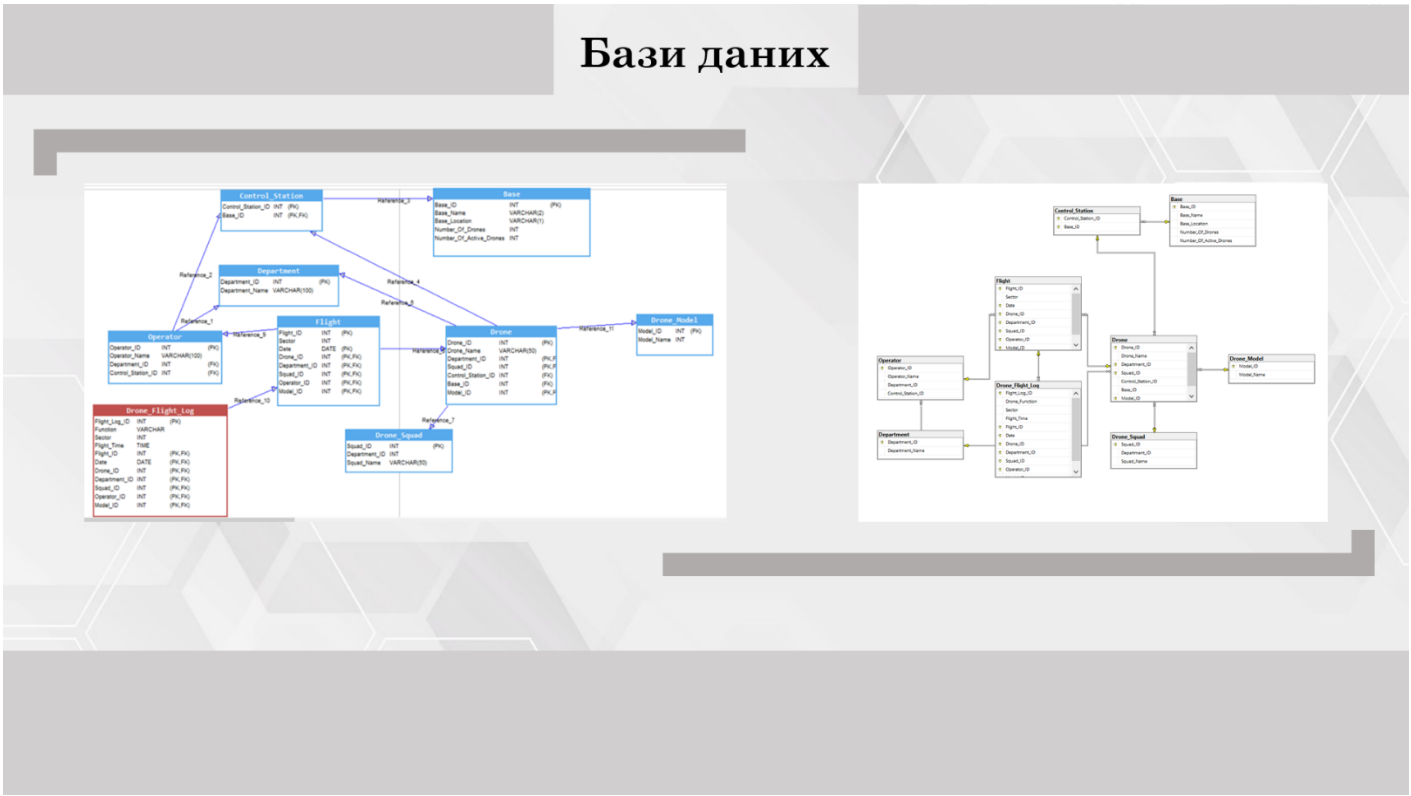


Рисунок 4 – Спроектовані бази даних

Програмна реалізація

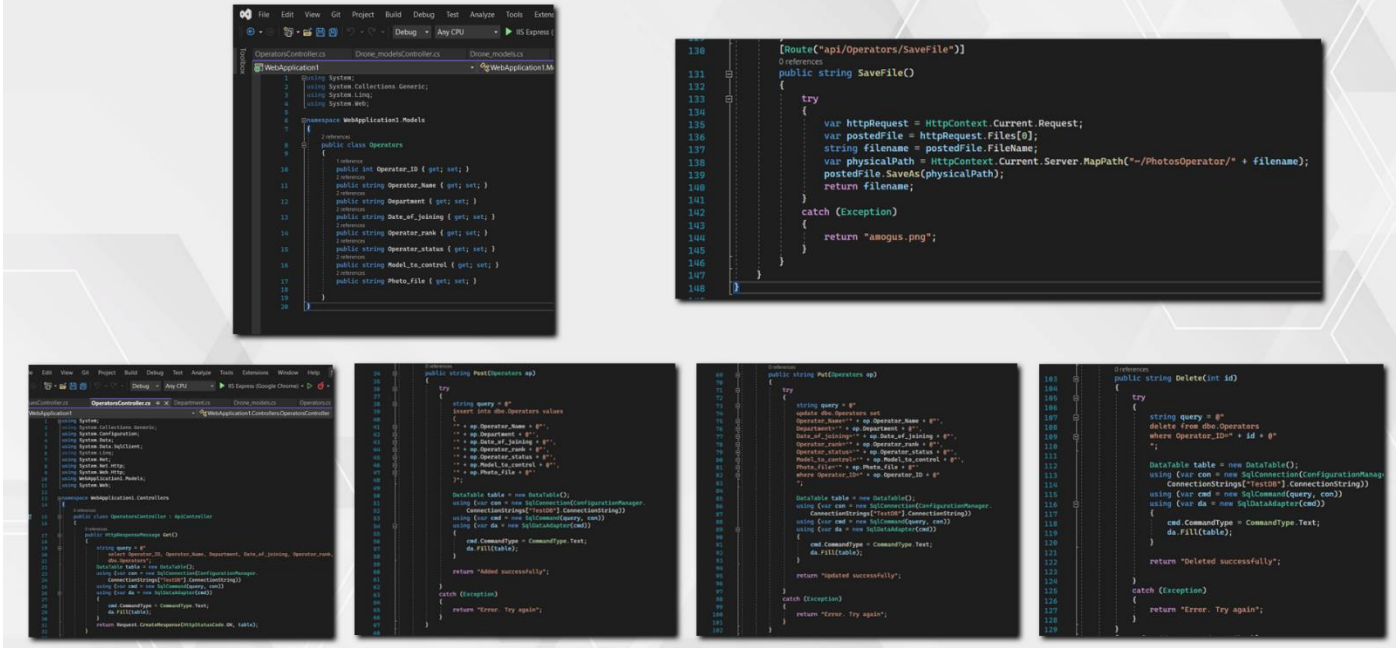


Рисунок 5 – Програмна реалізація

Тестування

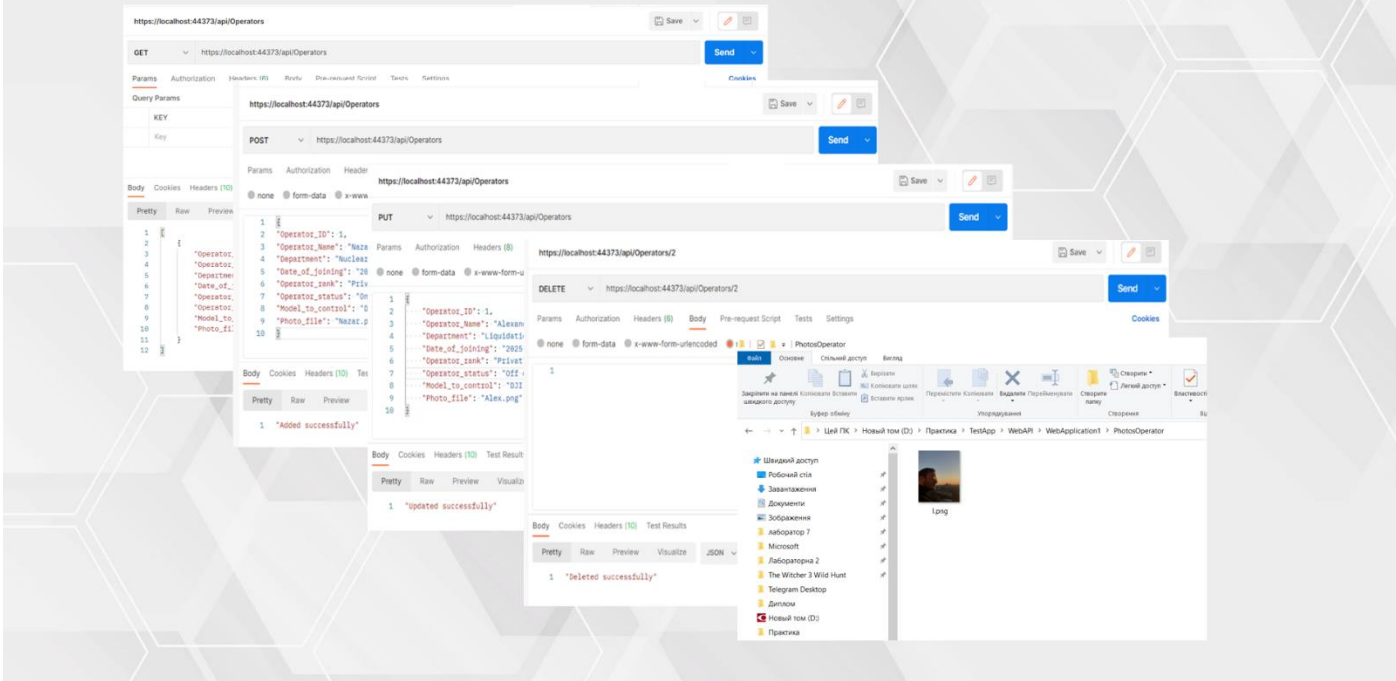


Рисунок 6 – Результати тестування