

Міністерство освіти і науки України  
Харківський національний університет радіоелектроніки

Факультет Автоматики та комп'ютеризованих технологій  
(повна назва)

Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
(повна назва)

## КВАЛІФІКАЦІЙНА РОБОТА

### Пояснювальна записка

Рівень вищої освіти перший (бакалаврський)

Розробка системи автоматизації для доставки медикаментів за допомогою дронів  
(тема)

Виконав:

студент 4 курсу, групи АКТАКІТ-20-2

Чеснаков В. С.

(прізвище, ініціали)

Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Автоматизація та комп'ютерно-інтегровані технології

(повна назва освітньої програми)

Керівник доцент Жарікова І. В.

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри

\_\_\_\_\_ (підпис)

\_\_\_\_\_ (прізвище, ініціали)

2024 р.

## Харківський національний університет радіоелектроніки

Факультет Автоматики і комп'ютеризованих технологій  
 Кафедра Комп'ютерно-інтегрованих технологій, автоматизації та робототехніки  
 Рівень вищої освіти перший (бакалаврський)  
 Спеціальність 151 – Автоматизація та комп'ютерно-інтегровані технології  
 (код і повна назва)  
 Тип програми освітньо-професійна  
 Освітня програма Автоматизація та комп'ютерно-інтегровані технології  
 (повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри \_\_\_\_\_  
 (підпис)  
 « \_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_ р.

**ЗАВДАННЯ****НА КВАЛІФІКАЦІЙНУ РОБОТУ**

студентові \_\_\_\_\_ Чеснаков Владислав Сергійович  
 (прізвище, ім'я, по батькові)

1. Тема роботи Розробка системи автоматизації для доставки медикаментів за допомогою дронів  
 затверджена наказом університету від 03 червня 2024 р. № 544 Ст
2. Термін подання студентом роботи до екзаменаційної комісії 17.06.2024 р.
3. Вихідні дані до роботи 3.1 Середовище розробки – Android Studio.  
3.2 Мова програмування – Kotlin, Python 3.3 Реалізація додатку для системи доставки  
 медикаментів. 3.4 Реалізація алгоритму автоматичного планування маршруту.
4. Перелік питань, що потрібно опрацювати в роботі
  - 4.1 Аналіз вимог та функціональних можливостей системи автоматизованої доставки  
 медикаментів;
  - 4.2 Аналіз алгоритмів автоматичного планування маршрутів;
  - 4.3 Розробка додатку для автоматизованої системи доставки;
  - 4.4 Вибір платформи та реалізація алгоритму планування маршруту для автоматизованої  
 системи доставки.
  - 4.5 Забезпечення умов охорони праці;
  - 4.6 Висновки.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) Графічний демонстраційний матеріал в форматі PowerPoint(\*.ppt) формату А4 – 12 сторінок та відеоматеріал

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

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

### КАЛЕНДАРНИЙ ПЛАН

| №  | Назва етапів роботи   | Терміни виконання етапів роботи | Примітка |
|----|---|---------------------------------|----------|
| 1  | Отримання завдання до кваліфікаційної роботи  | 13.05.24                        | виконано |
| 2  | Вступ   | 15.05.24                        | виконано |
| 3  | Аналіз вимог та функціональних можливостей системи автоматизованої доставки медикаментів          | 18.05.24                        | виконано |
| 4  | Аналіз алгоритмів автоматичного планування маршрутів  | 20.05.24                        | виконано |
| 5  | Розробка додатку для автоматизованої системи доставки   | 27.05.24                        | виконано |
| 6  | Вибір платформи та реалізація алгоритму планування маршруту для автоматизованої системи доставки. | 31.05.24                        | виконано |
| 7  | Забезпечення умов охорони праці   | 02.06.24                        | виконано |
| 8  | Висновки  | 03.06.24                        | виконано |
| 9  | Оформлення пояснювальної записки  | 05.06.24                        | виконано |
| 10 | Подання роботи на нормоконтроль   | 11.06.24                        | виконано |
| 11 | Перевірка роботи у системі StrikePlagiarism   | 11.06.24                        | виконано |
| 12 | Отримання відгуку та рецензії   | 13.06.24                        | виконано |
| 13 | Подання роботи до ЕК  | 14.06.24                        | виконано |
| 14 | Підготовка презентації та доповіді  | 14.06.24                        | виконано |

Дата видачі завдання 13.05.2024 р.

Студент \_\_\_\_\_  
(підпис)

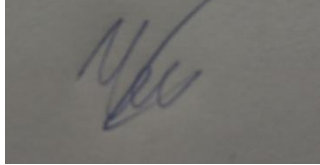
Чеснаков В. С.  
(прізвище та ініціали)

Керівник роботи \_\_\_\_\_  
(підпис)

доц. Жарікова І. В.  
(посада, прізвище, ініціали)

Я, як студент ХНУРЕ, розумію і підтримую політику закладу із академічної доброчесності. Я не надавав та не одержував недозволену допомогу під час підготовки кваліфікаційної роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

10.06.2024

A rectangular area containing a handwritten signature in blue ink, which appears to be 'V. Chesnakov'.

Владислав ЧЕСНАКОВ

## РЕФЕРАТ

Кваліфікаційна робота містить: 49 с., 14 рис., 2 дод., 18 джерел.

### ДРОН, АВТОМАТИЗАЦІЯ, ДОСТАВКА МЕДИКАМЕНТІВ, АЛГОРИТМ ДЕЙКСТРИ.

Мета роботи – розробка системи автоматизації для доставки медикаментів за допомогою дронів, що включає в себе реалізацію алгоритмів планування маршрутів, розробку додатка для створення замовлення та інтеграцію з хмарним середовищем для забезпечення ефективного та безпечного виконання поставлених завдань.

Об'єкт розробки – процес автоматизованої доставки медикаментів за допомогою дронів.

Предмет розробки – алгоритми та програмні засоби, що використовуються для автоматизації процесу доставки медикаментів дронами.

Методи розробки – системний аналіз, програмування для реалізації алгоритмів та інтерфейсів користувача.

У кваліфікаційній роботі виконано аналіз вимог та функціональних можливостей автоматизованої системи доставки медикаментів; аналіз алгоритмів автоматичного планування маршрутів для дронів.

Вирішені практичні задачі: розроблення автоматизованої системи доставки медикаментів; вибір платформи та реалізація алгоритму планування маршруту для автоматизованої системи доставки.

У розробленому додатку реалізовано можливість реєструватися в системі, обирати товар, робити та замовлення та підтверджувати його отримання.

## ABSTRACT

Qualification paper includes: 49 p., 14 fig., 2 adj., 18 sources

DRONE, AUTOMATION, MEDICATION DELIVERY, DIJKSTRA'S ALGORITHM.

Objective of the work – to develop an automated system for the delivery of medications using drones, which includes the implementation of route planning algorithms, the development of an application for order creation, and integration with a cloud environment to ensure efficient and safe task execution.

Object of development – the process of automated delivery of medications using drones.

Subject of development – algorithms and software tools used to automate the process of medication delivery by drones.

Development methods – systems analysis, programming for the implementation of algorithms and user interfaces.

In the qualification work, an analysis of the requirements and functional capabilities of the automated medication delivery system was performed; an analysis of automatic route planning algorithms for drones was conducted.

Practical tasks solved: the development of an automated medication delivery system; selection of a platform and implementation of a route planning algorithm for the automated delivery system.

The developed application includes the ability to register in the system, select products, place orders, and confirm receipt of the orders.

## ЗМІСТ

|   |    |
|---|----|
| Перелік умовних скорочень.....  | 9  |
| Вступ.....  | 10 |
| 1 Аналіз вимог та функціональних можливостей системи автоматизованої доставки медикаментів.....         | 12 |
| 1.1 Робота в умовах обмеженого доступу .....  | 12 |
| 1.2 Стійкість до впливу факторів бойової обстановки .....   | 13 |
| 1.3 Умови забезпечення швидкої реакції та доставки .....  | 14 |
| 1.4 Шляхи забезпечення конфіденційності системи доставки медикаментів ....                              | 15 |
| 1.5 Робота в умовах екстремальних температур та кліматичних умовах .....                                | 15 |
| 1.6 Запас енергії та тривалість польоту.....  | 16 |
| 1.7 Можливість використання відео- та тепловізійної зйомки.....   | 17 |
| 2 Аналіз алгоритмів автоматичного планування маршрутів .....  | 18 |
| 2.1 Алгоритм А* (A - star).....   | 18 |
| 2.2 Алгоритм Дейкстри.....  | 20 |
| 2.3 Алгоритм Флойда-Уоршела.....  | 22 |
| 2.4 Мурашиний алгоритм .....  | 24 |
| 3 Розробка додатку для автоматизованої системи доставки .....   | 29 |
| 3.1 Вибір платформи для створення додатку користувача та оператора .....                                | 29 |
| 3.2 Опис функціоналу додатку.....   | 33 |
| 3.3 Розроблення алгоритму дій користувача/оператора в системі .....                                     | 34 |
| 3.4 Розроблений інтерфейс користувача та приклад його використання .....                                | 36 |
| 3.5 Вибір хмарного середовища.....  | 42 |
| 4 Вибір платформи та реалізація алгоритму планування маршруту для автоматизованої системи доставки..... | 44 |
| 4.1 Огляд технічних характеристик дрона.....  | 44 |
| 4.2 Реалізація алгоритму автоматичного планування маршруту.....   | 45 |
| 4.3 Система автоматизованого управління для дрона DJI Matrice 300 RTK .....                             | 46 |

|     |  |    |
|-----|--|----|
| 4.4 | Опис пропорційно-інтегрально-диференціального (ПІД) регулятора ..... | 47 |
| 5   | Забезпечення умов охорони праці .....                                | 50 |
| 5.1 | Робоче середовище та організація праці.....                          | 50 |
| 5.2 | Безпека експлуатації дронів .....                                    | 51 |
| 5.3 | Правила безпеки при польотах .....                                   | 52 |
| 5.4 | Екологічна безпека .....   | 53 |
|     | Висновки .....   | 55 |
|     | Перелік джерел посилань .....  | 57 |
|     | Додаток А Лістинг коду .....   | 59 |
|     | Додаток Б Демонстраційний матеріал.....                              | 91 |

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД – бойові дії;

ІВП – інерціальні вимірювальні пристрої;

МП – мова програмування;

ПЗ – програмне забезпечення;

ПІД – пропорційно-інтегральний-диференційний;

САУ – система автоматизованого управління;

API – application programming interface;

GPS – global positioning system;

VPN – virtual private network.

## ВСТУП

Нинішній світ стрімко розвивається в плані впровадження новітніх технологій у різні сфери людської діяльності. Одним із найважливіших напрямків впровадження таких технологій є медична галузь, яка відіграє ключову роль у забезпеченні здоров'я та добробуту суспільства. Останнім часом, спостерігається необхідність в оперативній та безпечній доставці медикаментів, особливо в умовах пандемій, надзвичайних ситуацій. Звичні нам методи доставки медикаментів часто не здатні забезпечити необхідну швидкість та надійність, що може мати критичні наслідки для життя та здоров'я людей.

Одним із перспективних рішень даної проблеми є використання дронів, для автоматизованої системи доставки медикаментів. Дрони мають ряд переваг у порівнянні зі звичними нам способами транспортування, тому що вони здатні швидко долати великі відстані, легко оминати транспортні затори, доставляти вантажі у важкодоступні місця для людини. Крім того, використання дронів може значно знизити витрати на логістику та підвищити ефективність медичних послуг.

Виходячи із опрацьованої інформації в світі вже існують приклади успішного впровадження дронів для доставки медикаментів. В країнах Африки їх використовують для швидкої доставки крові та вакцин у важкодоступні регіони, де звичайний транспорт добратися не зможе. У США та Європі дрони також знаходять застосування у медицині, доставляючи ліки та медичні зразки між лікарнями та лабораторіями. Це свідчить про те, що дрони мають значний потенціал у медичній логістиці.

Мета роботи – розробка системи автоматизації для доставки медикаментів за допомогою дронів, що включає в себе реалізацію алгоритмів планування маршрутів, розробку додатка для створення замовлення та інтеграцію з хмарним середовищем для забезпечення ефективного та безпечного виконання поставлених завдань.

Об'єкт розробки – процес автоматизованої доставки медикаментів за допомогою дронів.

Предмет розробки – алгоритми та програмні засоби, що використовуються для автоматизації процесу доставки медикаментів дронами.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- аналіз вимог та функціональних можливостей системи автоматизованої доставки медикаментів;
- аналіз алгоритмів автоматичного планування маршрутів для дронів;
- розробка додатку для автоматизованої системи доставки;
- вибір платформи та реалізація алгоритму планування маршруту для автоматизованої системи доставки;
- оформити кваліфікаційну роботу згідно ДСТУ 3008:2015 [1], а також за допомогою методичних вказівок [2 - 5], та підтримує політику академічної доброчесності університету [6].

# 1 АНАЛІЗ ВИМОГ ТА ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СИСТЕМИ АВТОМАТИЗОВАНОЇ ДОСТАВКИ МЕДИКАМЕНТІВ

У військових конфліктах та в умовах важкодоступних регіонів, де доступ до медичної допомоги може бути обмеженим або навіть не можливим, доставка медикаментів за допомогою дронів стає необхідністю, яка може врятувати життя. Аналізування вимог щодо розробки системи є критично необхідним етапом, щоб визначити необхідні параметри та функціональні можливості, які необхідно врахувати. Розглянемо вимоги та функціональні можливості до системи автоматизованої доставки медикаментів детальніше.

## 1.1 Робота в умовах обмеженого доступу

Однією з не менш важливих деталей нашої системи повинна бути автономність. Таким чином:

- по-перше, система мусить бути здатна до роботи без прямого доступу до цивільних мереж зв'язку, котре може бути обмеженим або навіть відсутнім у деяких військових та важкодоступних регіонах;

- по-друге, дрони повинні мати вбудовані системи навігації та автопілоту, для змоги працювати без необхідності постійного зв'язку з оператором.

Також не менш важливим треба забезпечити системі стійкість до електромагнітних перешкод, адже:

- у військових операціях, коли противник використовує засоби радіоелектронної боротьби, виникають електромагнітні перешкоди, які впливають на роботу радіочастотного зв'язку. Система повинна бути стійкою до цих перешкод;

- необхідним може бути використання захисту від перешкод або альтернативні системи зв'язку такі, як зашифровані канали зв'язку або супутниковий зв'язок.

Для забезпечення ефективної роботи системи вона має здатність працювати

при низькій освітленості та погодних умовах, тому:

- система повинна мати здатність працювати у важкодоступних регіонах та зонах бойових дій (БД) де можуть виникнути умови з обмеженням освітленості або вночі. Для роботи в таких умовах система можливо може використовувати інфрачервону або нічну зйомку;

- дрони можна оснастити системами штучного зору та датчиками, що дозволить їм надійно працювати в умовах обмеженої видимості.

Система повинна мати стійкість до фізичних перешкод, адже у важкодоступних регіонах завжди присутні перешкоди, такі як гори, ліси або будівлі. Дрони повинні мати системи уникнення перешкод та здатність змінювати маршрут в разі необхідності.

## 1.2 Стійкість до впливу факторів бойової обстановки

Захист від зовнішніх впливів не менш важливіший у нашій системі. Вона має бути влаштована таким чином:

- систему треба конструювати таким чином, щоб вона була стійкою до впливу фізичних пошкоджень, які можуть виникати через обстріли, вибухи або інші фактори, які спричинені БД;

- треба використовувати матеріали та конструкції, які забезпечують захист від ударних хвиль, вогнепальних пошкоджень та фрагментів, може бути важливим для забезпечення цілісності системи.

Також критично важливим завданням є забезпечення безпеки посилок. Перерахуємо деякі аспекти:

- в разі впливу БД, система повинна забезпечити безпеку медичної посилки, щоб запобігти пошкодженню або втраті;

- слід використовувати спеціальне упакування для захисту медичних посилок, воно необхідне для забезпечення цілісності під час транспортування.

Також важливо враховувати автоматизацію та резервність. Розглянемо

необхідні підпункти:

- системі слід мати вбудовані механізми автоматизації та резервність, щоб в разі впливу різних факторів, таких як втрата зв'язку або пошкодження дрону, вона забезпечила продовження роботи;

- для забезпечення стійкості системи до негативних впливів, можна використовувати автономні системи управління та механізми автоматичного відновлення.

### 1.3 Умови забезпечення швидкої реакції та доставки

Швидка реакція та доставка вкрай необхідна для ефективності системи, особливо в надзвичайних умовах де кожна секунда грає вирішальне значення.

Розглянемо основні аспекти:

- система повинна мати здатність оперативного реагування на ситуації надзвичайних обставин, такі як травми, масові катастрофи або воєнні дії;

- дрони повинні бути готовими до швидкого вильоту та швидко реагувати на команди оператора або автоматичні сигнали про терміновість доставки;

- система повинна забезпечити швидку доставку медичної посилки до місця потреби;

- використання автоматизованих систем управління та моніторингу може допомогти зменшити час обробки замовлень та забезпечити швидку реакцію на потреби.

Також наша система повинна швидко монтуватися та розгортатися, тобто мати швидку мобільність. Розглянемо детальніше по пунктах:

- система повинна легко монтуватися та розгортатися швидко для оперативної реакції на екстрені ситуації;

- використання компактних і переносних дронів, які можна швидко встановити та запустити в роботу, може значно прискорити процес реагування.

## 1.4 Шляхи забезпечення конфіденційності системи доставки медикаментів

Шифрування для розроблюваної системи є важливим аспектом в плані конфіденційності та безпеки інформації. Розглянемо деякі аспекти:

- усі дані, які передаються між операторами та дроном повинні бути шифровані, для того, щоб захистити їх від несанкціонованого доступу;
- використання криптографічних протоколів та шифрування забезпечить конфіденційність даних, зокрема, маршрутів та медичних відомостей.

Безпека комунікацій в нашій системі критично важлива для захисту від несанкціонованого доступу. Її основні аспекти включають:

- захищені канали зв'язку, щоб можна було уникнути перехоплення або зміни команд та даних, які передаються між операторами та дронами;
- використання захищених протоколів зв'язку, таких як VPN, може допомогти забезпечити конфіденційність.

Анонімність даних є критичним аспектом в нашій системі. Розглянемо більш детально:

- медичні дані пацієнтів, які транспортуються за допомогою системи, повинні бути анонімними та захищеними від несанкціонованого доступу;
- забезпечення анонімізації медичних даних перед їхнім транспортуванням може зменшити ризик витоку особистої інформації.

Забезпечення фізичної безпеки також один із головних факторів в розробці нашої системи. Використання фізичних заходів безпеки, таких як блокування доступу до обладнання та мережевих підключень, може забезпечити захист від фізичних атак.

## 1.5 Робота в умовах екстремальних температур та кліматичних умовах

Робота в таких умовах вимагає ретельного проектування та вибору матеріалів, для забезпечення надійної та ефективної роботи системи. Розглянемо по

таких аспектах:

- система повинна бути стійкою до екстремальних температур, які можуть виникати в різних регіонах, зокрема в гарячих пустелях або холодних гірських місцевостях;

- дрони та їх обладнання повинні бути пристосованими до роботи в різних кліматичних умовах, включаючи вологість, дощ та сильний вітер;

- батареї, які використовуються для живлення дронів, повинні бути оптимізованими для роботи в екстремальних умовах;

- також важливо аби система у випадку роботи в умовах високих температур мала систему вентиляції та охолодження, для уникнення перегріву електронних компонентів та забезпечення стабільної роботи;

- ще важливо мати систему управління та моніторингу, яка буде в змозі відслідковувати стан дронів, включаючи температуру, заряд батарей, стан електроніки та інші параметри, які мають вплив на роботу в екстремальних умовах.

## 1.6 Запас енергії та тривалість польоту

Запас енергії та тривалість польоту є ключовим параметром для нашої системи. Розглянемо по аспектах:

- важливим аспектом є забезпечення дронів достатнім запасом енергії для тривалого польоту, особливо в умовах довгих відстаней та обмежених місцевостей;

- розробка ефективних акумуляторів або систем зарядки може значно покращити тривалість польоту та забезпечити безперервну роботу системи;

- також важливо мінімізувати вагу дронів та оптимізувати їх аеродинаміку, щоб забезпечити ефективне використання енергії під час польоту. Це може включати в себе використання легких матеріалів або зменшення опору повітря із-за його дизайну.

## 1.7 Можливість використання відео- та тепловізійної зйомки

Важливою функцією для системи доставки за допомогою дронів також є можливість використання відео- та тепловізійної зйомки. Розглянемо переваги:

- використання зйомки може в рази полегшити пошук потерпілих та визначення їхнього стану в умовах обмеженої видимості або вночі;
- додаткове обладнання на дронах для зйомки відео або тепловізійної картографії може забезпечити додаткові можливості для оперативного реагування на екстрені ситуації.

Після аналізу вимог та функціональних можливостей для розробки автоматизованої системи доставки медикаментів можна зробити наступні висновки:

- комплексний підхід: для успішної розробки системи необхідно враховувати всі аспекти, такі як автономність, стійкість до електромагнітних та фізичних перешкод, здатність працювати в погодніх умовах та при обмеженій видимості, а також швидку реакцію та конфіденційність;
- технологічна надійність: розробка та використання високотехнологічних рішень таких шифрування даних, алгоритми автономної навігації та стійкі системи зв'язку, є важливими для забезпечення надійності та ефективності системи;
- швидка реакція: система повинна мати здатність оперативно реагувати на надзвичайні ситуації та швидко доставляти медичні посилки до місць потреб, щоб максимально зменшити час реакції та покращити шанси на порятунок;
- безпека та конфіденційність: безпека медичних посилок та конфіденційності даних є критично важливими аспектами, щоб уникнути витоку особистої інформації та забезпечити надійність системи в умовах БД.

## 2 АНАЛІЗ АЛГОРИТМІВ АВТОМАТИЧНОГО ПЛАНУВАННЯ МАРШРУТІВ

### 2.1 Алгоритм A\* (A - star)

Алгоритми автоматичного планування маршруту для розроблюваної системи є ключовим етапом проєкту. Розберемо декілька алгоритмів, які можна застосувати.

Алгоритм Алгоритм A\* (A-star) користується популярністю серед алгоритмів для пошуку шляху в графі з вагами. Це комбінація алгоритмів «Greedy Best First Search» і «Shortest Path» (алгоритм Дейкстри) для пошуку найкращого шляху між двома точками у графі [7].

Принцип роботи алгоритму полягає у тому, щоб обрати наступний вузол для розгляду на основі суми двох значень. Обчислення проводиться за формулою:

$$f(x) = g(x) + h(x), \quad (3.1)$$

де  $g(x)$  – вартість найкоротшого шляху від початкового вузла до поточного вузла;

$h(x)$  – оцінка відстані від поточного вузла до кінцевого вузла.

Отже, алгоритм A\* обирає вузол з найменшою сумою  $g(x) + h(n)$ , що дозволяє йому ефективно обходити граф та шукати оптимальний маршрут.

Переваги:

- швидкий та ефективний для багатьох типів графів;
- гарантує знаходження оптимального маршруту, якщо евристична функція оцінює відстань правильно.

Розглянемо ще приклад реалізації алгоритму на мові програмування (МП) Python:

```
import heapq
```

```
def astar(graph, start,
goal):
    open_set = [(0,
start)]
    came_from = {}

    g_score = {node: float('inf') for node in graph}
    g_score[start] = 0

    f_score = {node: float('inf') for node in graph}
    f_score[start] = heuristic(start, goal)

    while open_set:
        current_f, current = heapq.heappop(open_set)
        if current == goal:
            return reconstruct_path(came_from, current)

        for neighbor in graph[current]:
            tentative_g = g_score[current] + graph[current][neighbor]
            if tentative_g < g_score[neighbor]:
                came_from[neighbor] = current
                g_score[neighbor] = tentative_g
                f_score[neighbor] = tentative_g + heuristic(neighbor, goal)
                heapq.heappush(open_set,
(f_score[neighbor], neighbor))

    return None
```

```

def heuristic(node, goal):

# Це може бути будь-яка евристична функція, наприклад, евклідова відстань

return ((node[0] - goal[0]) ** 2 + (node[1] - goal[1]) ** 2) ** 0.5

def reconstruct_path(came_from,
current):path = [current]

while current in came_from:

current = came_from[current]

path.append(current)

return path[::-1]

```

Найчастіше даний алгоритм використовується для знаходження найшвидшого маршруту між точками доставки.

## 2.2 Алгоритм Дейкстри

Алгоритм Дейкстри – класичний алгоритм пошуку найкоротшого шляху в графі з неорієнтованими або орієнтованими ребрами. Він широко використовується в різних сферах, включаючи транспорт, мережеву маршрутизацію та системи доставки, включаючи системи доставки безпілотниками. Принцип роботи даного алгоритму полягає в наступному [8]:

- ініціалізується відстань від початкової вершини до всіх інших вершин як нескінченна, окрім самої початкової вершини, відстань до якої встановлюється як 0;
- обирається вершина з найменшою відстанню серед вершин, що ще не були відвідані;
- оновлюється відстань до всіх сусідніх вершин вибраної вершини, якщо нова

відстань менше поточної;

- повторюється кроки 2 та 3, поки всі вершини не будуть відвідані.

Переваги алгоритму:

– гарантує знаходження найкоротшого шляху від початкової вершини до всіх інших вершин у напіввагованому графі;

– має поліноміальну складність у порівнянні з багатьма іншими алгоритмами.

Недоліки:

- не працює з графами, у яких є ваги з від'ємними значеннями;
- може бути неефективним на графах з великою кількістю вершин та ребер.

Приклад реалізації алгоритму на МП Python:

```
import heapq

def dijkstra(graph, start):
    distances = {vertex: float('infinity') for vertex in graph}
    distances[start] = 0
    queue = [(0, start)]

    while queue:
        current_distance, current_vertex = heapq.heappop(queue)

        if current_distance > distances[current_vertex]:
            continue

        for neighbor, weight in graph[current_vertex].items():
            distance = current_distance + weight
```

```

if distance < distances[neighbor]: distances[neighbor] =
distance
heapq.heappush(queue, (distance, neighbor))

return distances

```

# Приклад

```

графу:graph = {
'A': {'B': 1, 'C': 4},
'B': {'A': 1, 'C': 2, 'D': 5},
'C': {'A': 4, 'B': 2, 'D': 1},
'D': {'B': 5, 'C': 1}
}
start_vertex = 'A'

shortest_distances = dijkstra(graph, start_vertex)

print("Найкоротші відстані від вершини", start_vertex, "до інших вершин:",
shortest_distances)

```

### 2.3 Алгоритм Флойда-Уоршела

Даний алгоритм використовується для знаходження найкоротших шляхів між усіма парами вершин у напіввагованому або зваженому графі. Він працює навіть у випадку, коли в графі присутні цикли з від'ємними вагами [9].

Принцип роботи алгоритму полягає у наступному:

- ініціалізується матриця відстаней, в якій початкові відстані між усіма парами вершин дорівнюють вагам ребер або нескінченності;
- за допомогою трьох вкладених циклів обчислюються всі можливі шляхи

між парами вершин, враховуючи можливі шляхи через інші вершини;

- кожен раз, коли знаходиться коротший шлях між вершинами через іншу вершину, оновлюється значення відстані.

Переваги алгоритму Флойда-Уоршелла:

- гарантує знаходження найкоротших шляхів між усіма парами вершину напіввагованому графі;
- працює з графами з вагами з від'ємними значеннями та знаходить коректний результат.

Недоліки:

- має кубічну складність за часом виконання, тому не ефективний на великих графах;
- вимагає більше пам'яті для зберігання матриці відстаней.

Приклад реалізації алгоритму на МП Python:

```
def floyd_warshall(graph):
    # Кількість вершин в графі
    num_vertices = len(graph)

    # Ініціалізуємо матрицю відстаней
    dist = [[float('inf')] * num_vertices for _ in range(num_vertices)]

    # Ініціалізуємо матрицю відстаней з графа
    for i in range(num_vertices):
        for j in range(num_vertices):
            if i == j:
                dist[i][j] = 0
            elif graph[i][j] != 0:
                dist[i][j] = graph[i][j]

    # Основний алгоритм
    for k in range(num_vertices):
```

```

for i in range(num_vertices):
    for j in range(num_vertices):
        if dist[i][j] > dist[i][k] + dist[k][j]:
            dist[i][j] = dist[i][k] + dist[k][j]

return dist

```

## 2.4 Мурашиний алгоритм

Мурашиний алгоритм – метаевристичний алгоритм оптимізації, який імітує поведінку мурашок при пошуку їжі. Він використовує принципи колективної поведінки мурашок для взаємодії між ними для знаходження найкращих рішень у просторі пошуку. Мурашиний алгоритм часто використовується для розв’язання задач комбінаторної оптимізації, включаючи задачі комівояжера, маршрутизації транспорту та інші.

Принцип роботи можна узагальнити наступним чином [10]:

- ініціалізація феромонів: кожне ребро графа має початкове значення феромону;
- рух мурашок: кожна мурашка переходить з однієї вершини графа в іншу, вибираючи наступну вершину на основі комбінації феромонів та евристичних оцінок;
- оновлення феромонів: після кожного кроку кожна мурашка оновлює значення феромону на ребрах свого маршруту відповідно до його ефективності;
- глобальне оновлення феромонів: після кінцевого кроку глобально оновлюються значення феромонів на всіх ребрах відповідно до ефективності усіх маршрутів.

Переваги алгоритму:

- ефективний у розв’язанні задач комбінаторної оптимізації з великою кількістю можливих рішень;
- легко паралелізується для роботи з більшими обсягами даних та

обчислювальними завданнями.

Недоліки:

- може займати багато часу для збіжності до оптимального рішення;
- вимагає налаштування параметрів, таких як швидкість випаровування феромону та значення параметрів альфа і бета, щоб забезпечити збіжність та якість рішення.

Приклад реалізації на МП Python:

```
import random
```

```
class AntColony:
```

```
    def __init__(self, distances, n_ants, n_best, n_iterations, decay, alpha=1, beta=1):
        self.distances = distances
        self.pheromone = [[1 / len(distances) for _ in range(len(distances))] for _ in
range(len(distances))]
        self.all_inds = range(len(distances))
        self.n_ants = n_ants
        self.n_best = n_best
        self.n_iterations = n_iterations
        self.decay = decay
        self.alpha = alpha
        self.beta = beta

    def run(self):
        shortest_path = None
        all_time_shortest_path = ("placeholder", float('inf'))
        for i in range(self.n_iterations):
            all_paths = self.gen_all_paths()
            self.spread_pheromone(all_paths, self.n_best)
            shortest_path = min(all_paths, key=lambda x: x[1])
```

```

    if shortest_path[1] < all_time_shortest_path[1]:
        all_time_shortest_path = shortest_path
    self.pheromone_evaporation()
return all_time_shortest_path

```

```

def gen_path_dist(self, path):
    total_dist = 0
    for i in range(len(path)):
        total_dist += self.distances[path[i-1]][path[i]]
    return total_dist

```

```

def gen_all_paths(self):
    all_paths = []
    for i in range(self.n_ants):
        path = self.gen_path(0)
        all_paths.append((path, self.gen_path_dist(path)))
    return all_paths

```

```

def gen_path(self, start):
    path = []
    visited = set()
    visited.add(start)
    prev = start
    for i in range(len(self.distances) - 1):
        move = self.pick_move(self.pheromone[prev], self.distances[prev], visited)
        path.append(move)
        prev = move
        visited.add(move)
    path.append(start)
    return path

```

```

def pick_move(self, pheromone, dist, visited):
    pheromone = [p ** self.alpha for p in pheromone]
    inv_dist = [1 / d if d > 0 else 0 for d in dist]
    inv_dist = [d ** self.beta for d in inv_dist]
    numerators = [pheromone[i] * inv_dist[i] if i not in visited else 0 for i in
range(len(pheromone))]
    denominator = sum(numerators)
    probabilities = [n / denominator for n in numerators]
    move = random.choices(self.all_inds, probabilities)[0]
    return move

def spread_pheromone(self, all_paths, n_best):
    sorted_paths = sorted(all_paths, key=lambda x: x[1])
    for path, dist in sorted_paths[:n_best]:
        for move in path:
            self.pheromone[move[0]][move[1]] += 1.0 / self.distances[move[0]][move[1]]

def pheromone_evaporation(self):
    for i in range(len(self.pheromone)):
        for j in range(len(self.pheromone[i])):
            self.pheromone[i][j] *= self.decay

# Приклад використання
distances = [
    [0, 2, 2, 5, 7],
    [2, 0, 4, 8, 2],
    [2, 4, 0, 1, 3],
    [5, 8, 1, 0, 2],
    [7, 2, 3, 2, 0]

```

```
]
ant_colony = AntColony(distances, 3, 1, 100, 0.95, alpha=1, beta=2)
shortest_path = ant_colony.run()
print("Коротший шлях:", shortest_path)
```

Кожен з цих розглянутих алгоритмів має свої унікальні переваги та застосування:

- алгоритм A\* використовує евристичну функцію для пошуку найкоротшого шляху в графі з вагами. Він є ефективним для пошуку оптимальних маршрутів у навігації та штучному інтелекті;

- алгоритм Дейкстри також призначений для пошуку найкоротшого шляху, але працює лише з неорієнтованими або напіввагованими графами. Цей алгоритм широко використовується у мережевих та транспортних задачах;

- алгоритм Флойда-Уоршелла дозволяє знаходити найкоротші шляхи між усіма парами вершин у графі. Він добре підходить для використання в транспортних системах та інших задачах, де необхідно знати відстані між усіма парами вершин;

- мурашиний алгоритм – це метаевристичний алгоритм, який ефективно розв'язує складні оптимізаційні задачі. Він може бути застосований до широкого спектру проблем, таких як маршрутизація, планування та навіть машинне навчання.

### 3 РОЗРОБКА ДОДАТКУ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДОСТАВКИ

#### 3.1 Вибір платформи для створення додатку користувача та оператора

Створення додатку з інтерфейсом користувача в проекті є невід’ємною частиною. Адже автоматизована система доставки медикаментів може значно забезпечити ефективне та своєчасне постачання необхідних ліків. Тому виникає неабияка потреба у створенні мобільного додатку, який допоміг би забезпечити ефективне управління процесом доставки медикаментів.

Вибір платформи для створення додатку є важливим кроком, оскільки є багато факторів, такі як швидкість розробки, інтеграцію з іншими системами та продуктивність самого додатку. Отже, виконаємо порівняння сучасних платформ та фреймворків для розробки.

React Native – фреймворк інтерфейсу користувача з відкритим кодом, створений Meta Platforms Inc [11]. Його використовують для розробки програм як для Android так і для iOS що дозволяє розробникам використовувати фреймворк React разом із нативними можливостями платформи. Також його використовують для розробки програм віртуальної реальності в Oculus.

Для створення програм React Native існує два популярних засоби EXPO CLI та React Native. Розберемо їх детальніше та окремо.

EXPO – набір інструментів та сервісів, які допомагають розробникам створювати мобільні додатки на основі React Native. З ним можна швидко почати роботу над мобільним додатком, не звертаючи увагу на налаштування середовища розробки. Також тут є зручні інструменти для розробки, які дозволяють бачити зміни в реальному часі без перезавантаження їх назва: Live Reload та Hot Reload. Також Expo постачається з багатьма готовими до використання бібліотеками та інструментами для розробки, які допомагають зосередитися на функціональності додатку. Але в деяких випадках, особливо для додатків з високими вимогами до

налаштування, Expo може обмежувати можливості розробника в порівнянні з чистим React Native.

React Native – використовує для створення JavaScript та React. Використання нативних компонентів дає можливість досягти високої продуктивності та швидкості роботи додатків. Також має велику кількість готових компонентів, що дозволяє прискорити розробку. Один його код може бути використаний, як для Android так і для iOS. Ще він має більшу гнучкість та контроль. Розробник має більший контроль над додатком, що може бути корисним для складних або унікальних функцій.

Переваги МП React Native:

- надає розробникам недорогий спосіб створення кросплатформових додатків за допомогою React Native;
- один код для розробки для Android та iOS, що дозволяє знизити витрати на розробку та підтримку;
- використання нативних компонентів, забезпечує високу продуктивність і плавність роботи;
- велика кількість бібліотек та компонентів;
- Hot Reload дозволяє миттєво бачити зміни в коді, що прискорює розробку.

Недоліки:

- відлагодження нативних модулів може бути складним;
- обмеження нативних функцій;
- часто оновлення фреймворку потребує значних змін в коді.

Xamarin – платформа з відкритим вихідним кодом, призначена для побудови сучасних продуктивних програм для iOS, Android та Windows з .NET. Платформа являє собою новий рівень абстракції, який керує зв'язком між загальним кодом та з базовим кодом платформи. Xamarin працює в керованому середовищі, яке забезпечує такі зручності, як розподіл пам'яті та збирання сміття [12].

Xamarin дозволяє розробникам спільно використовувати в середньому 90 % своїх програм на різних платформах. Цей шаблон дозволяє розробникам писати всю свою бізнес-логіку однією мовою (або повторно використовувати існуючий код

програми), але досягати нативної продуктивності, зовнішнього вигляду на кожній платформі.

Переваги платформи Xamarin:

- один код для Android, iOS та Windows;
- інтеграція з Microsoft;
- використання нативних API та компонентів.

Недоліки:

- додатки можуть бути більшими за розміром через включення .NET Runtime;
- для доступу до деяких функцій може знадобитися підписка на Visual Studio;
- складніше у налаштуванні та розробці в порівнянні з іншими фреймворками.

Android Studio – інтегроване середовище розробки, вироблене Google, за допомогою якого розробникам стають доступними інструменти для створення програм на платформі Android [13].

Studio містить інструменти для розробки для Android, Android TV, Android Wear, Android Auto, Glass.

На сьогодні основною мовою програмування, яка використовується на даній платформі є Kotlin.

Kotlin – це сучасна багатоцільова мова програмування. Вона призначена для створення додатків на платформі Java, але вона також підтримує безліч інших цілей, включно з Android – розробкою, веб – розробкою та ін. Ця МП вирізняється високою виразністю, безпекою типів, і зручністю у використанні, що робить його привабливим вибором для широкого кола завдань і проєктів.

Переваги .МП Kotlin:

- Kotlin надає змогу писати код компактніше та виразніше, що прискорює процес розробки. Завдяки статичній типізації та функціональним можливостям, також він сприяє збільшенню продуктивності додатків;
- надає безліч інструментів для забезпечення безпеки коду;

- сприяє написанню чистішого та читабельнішого коду завдяки своєму синтаксису та функціональним конструкціям;

- має багату екосистему бібліотек та інструментів, що спрощує розробку.

Недоліки:

- перехід з Java на Kotlin може викликати певний страх у деяких розробників. Це пов'язано з тим, що зміна мови програмування, яка використовується, завжди супроводжується перенавчанням і адаптацією до нового синтаксису;

- у деяких галузях розробки, як-от системне програмування або розробка ігор, Kotlin може бути менш популярним вибором у порівнянні з іншими мовами.

NativeScript – фреймворк з відкритим вихідним кодом, що розробляється компанією Telerik, для розробки додатків на платформах Android і iOS. Додатки NativeScript розробляються на платформонезалежних мовах, таких як Javascript або TypeScript. У NativeScript реалізована повна підтримка фреймворку AngularJS. Мобільні додатки, побудовані з NativeScript, мають повний доступ до API платформи так, ніби вони були розроблені в XCode або в Android Studio. Також розробники можуть включати в свої додатки сторонні бібліотеки з таких ресурсів, як Cocoapods, Android Arsenal, Maven і npm.js, без створення додаткових прошарків [14].

Переваги NativeScript:

- використання нативних компонентів для забезпечення високої продуктивності;

- можливість використання JavaScript, TypeScript або Angular;

- один код для Android та iOS.

Недоліки:

- може бути складніше у налаштуванні порівняно з іншими фреймворками;

- може потребувати написання нативного коду для доступу до деяких функцій.

### 3.2 Опис функціоналу додатку

Для системи було розроблено два додатки, один для користувачів, через який вони зможуть зробити замовлення, а інший – для оператора. Для розробки було використано Android Studio, МП Kotlin.

Додаток клієнта:

- реєстрація та авторизація: клієнт може створити обліковий запис через email, а потім авторизуватися через login, який він вказав під час реєстрації;
- профіль користувача: можна редагувати особисту інформацію (ім'я, адресу, номер телефону), також зберігати цю інформацію для швидкого вибору замовлення;
- перегляд асортименту: можна переглянути асортимент доступних ліків та почитати про них більш детальну інформацію;
- замовлення медикаментів: додавання товару до кошика, оформлення замовлення з вибором адреси доставки, або за геолокацією, також отримання статусу доставки та підтвердження.

Додаток для оператора:

- авторизація: оператор не має необхідності створювати обліковий запис адже вся інформація про нього є базі даних, тому йому треба лише авторизуватися;
- управління замовленнями: перегляд всіх наявних замовлень, моніторинг статусу кожного замовлення;
- профіль оператора: буде відображатися інформація про оператора;
- контроль дронів: відстеження місцезнаходження дрону, контроль маршруту, оновлення статусу доставки;

Також в майбутньому буде додана система, контролю технічного стану та система підтримки.

### 3.3 Розроблення алгоритму дій користувача/оператора в системі

Коли користувач вперше заходить до додатку, де потрібна авторизація, він має зареєструватися в системі, тому алгоритм створення облікового запису буде мати такий вигляд, як показано на рис. 3.1.

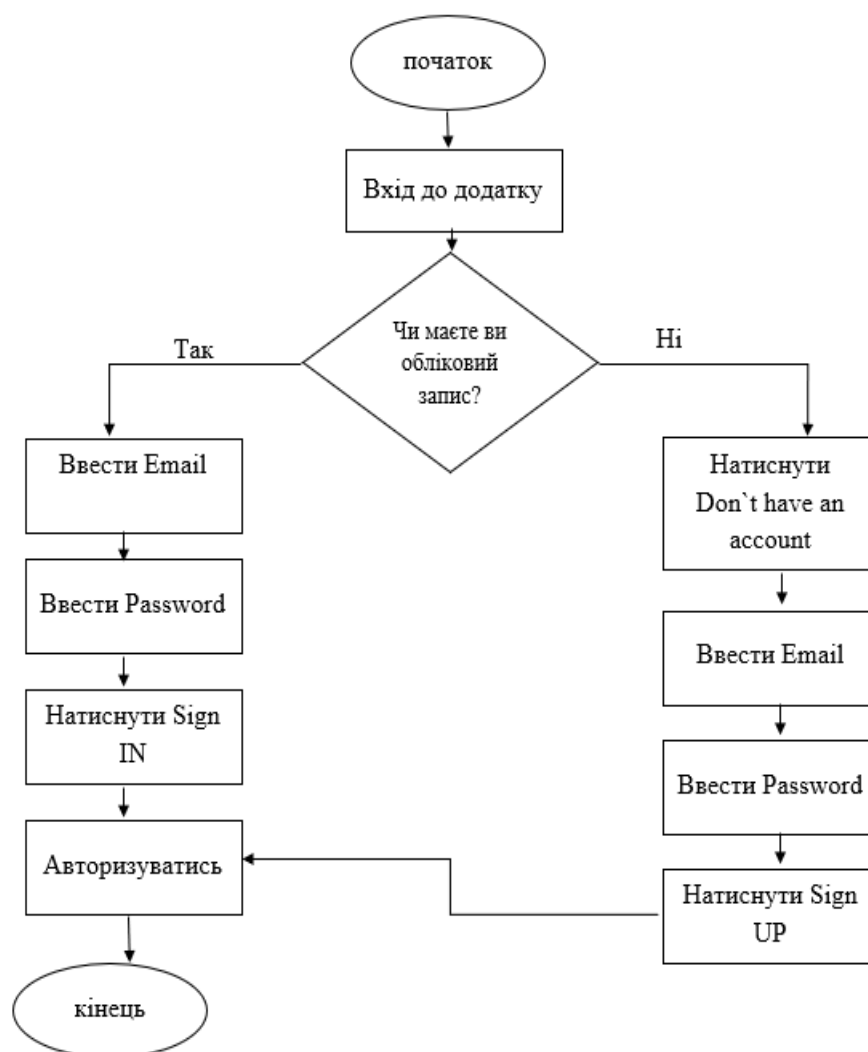


Рисунок 3.1 – Алгоритм авторизації/реєстрації користувача

Даний алгоритм відображає дії користувача в разі створення облікового запису або авторизації додатку.

Наступний алгоритм відображає створення замовлення (рис. 3.2).

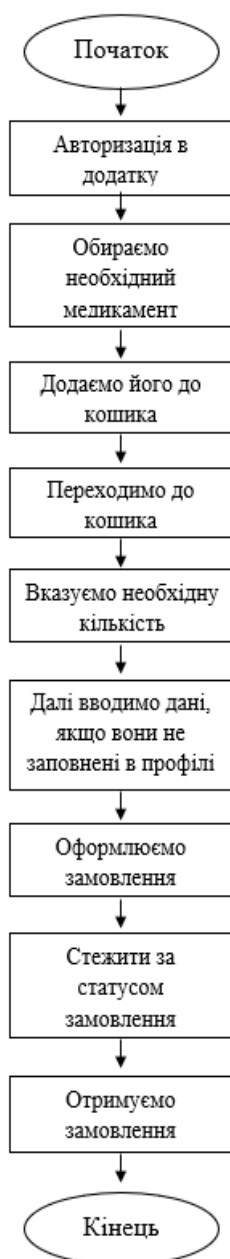


Рисунок 3.2 – Алгоритм створення, відстежування та отримання змовлення

Даний алгоритм відображає етапи створення замовлення користувачем.

Наступний алгоритм буде орієнтованим вже на додаток оператору, він авторизується в системі без реєстрації, адже інформація про нього вноситься в БД до цього. Тому даний алгоритм буде відображати дії при отриманні замовлення (рис. 3.3).



Рисунок 3.3 – Алгоритм доставки замовлення

Даний алгоритм відображає дії оператора під час замовлення.

### 3.4 Розроблений інтерфейс користувача та приклад його використання

Додаток складається з двох типів форм:

- реєстрація облікового запису;
- сам мобільний додаток.

Форма реєстрації вимагає від користувача введення даних для авторизації в системі. Користувачу доведеться ввести Email та Password, для того щоб продовжити працювати з додатком. Форма набуває такого вигляду, який наведено на рис. 3.4.

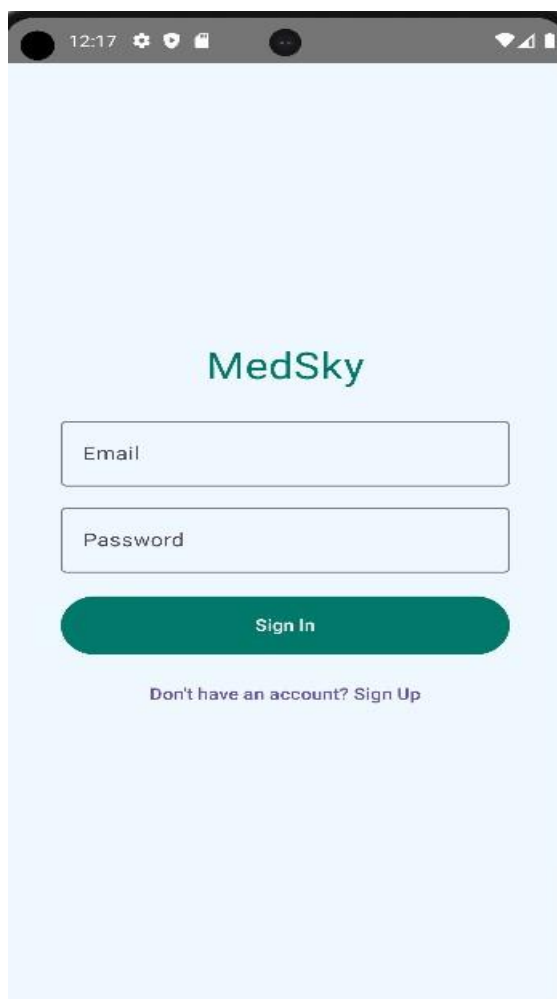


Рисунок 3.4 – Авторизація в системі

Якщо користувач в системі вперше, то йому треба буде створити аккаунт для того, щоб в подальшому взаємодіяти з додатком. Форма реєстрації має такий же самий вигляд.

Після успішної авторизації/реєстрації перед користувачем відкривається каталог в якому він може ознайомитися з ліками, дізнатися більше детальнішу інформацію про них або додати їх до кошика. Дана сторінка має вигляд, наведений на рис. 3.5.

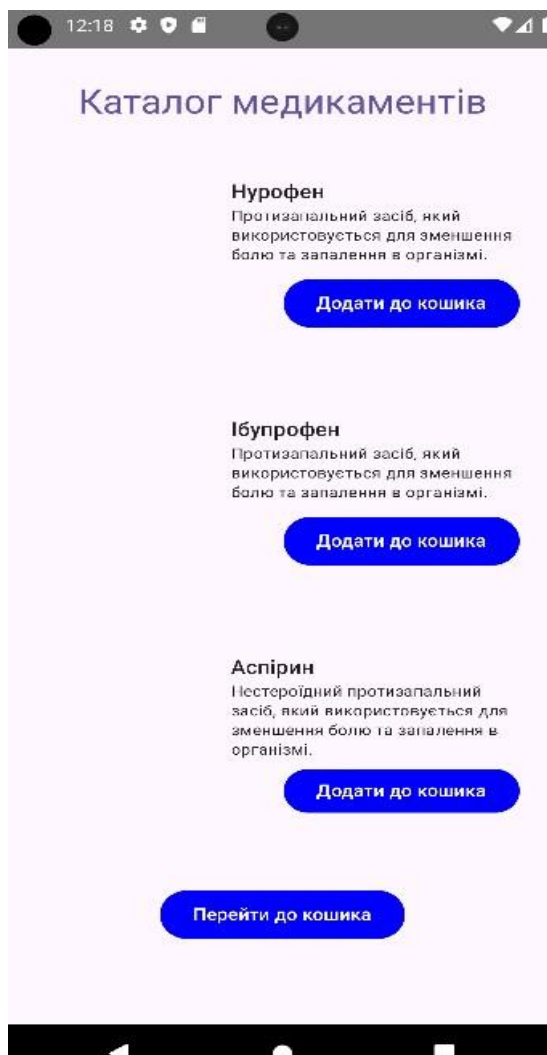


Рисунок 3.5 – Каталог медикаментів

Після того, як користувач додав медикаменти до кошика, він може вказати їх кількість або видалити їх із кошика та повернутися до каталогу, щоб обрати інші. Коли користувач обрав необхідну кількість медикаментів, то він натискає «Продовжити», далі заповнює інформацію про себе. Замість адреси можна ввести координати або натиснути кнопку «Визначити геолокацію» (буде інтегрована в додаток згодом) та після заповнення натискати кнопку «Продовжити». Увесь процес продемонстровано на рис. 3.6 та рис. 3.7.



Рисунок 3.6 – Кошик

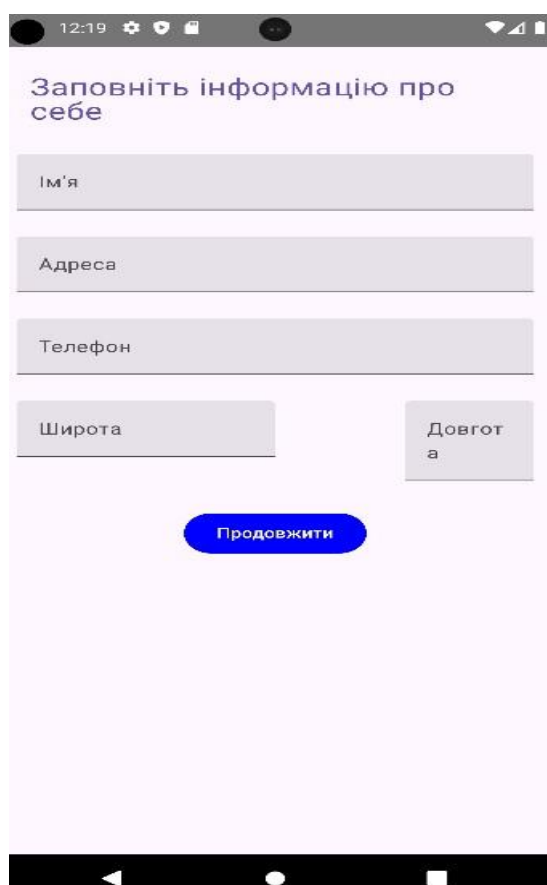


Рисунок 3.7 – Інформація про користувача

Після оформлення замовлення користувач звіряє свої дані, замовлення та підтверджує замовлення. Подалі буде інтегрований процес відслідковування замовлення. Форма деталей замовлення продемонстрована на рис. 3.8.



Рисунок 3.8 – Деталі замовлення

Підтвердивши замовлення, з'являється форма, в якій після його виконання треба буде вже підтвердити замовлення після його отримання. Після ознайомлення з додатком користувача перейдемо до додатку оператора.

Авторизація в ньому відбувається таким же чином, як і в користувачів. Після авторизації в додатку з'являється екран з наявними замовленнями, обравши замовлення оператор відкриває його деталі, які були відправленні користувачем.

Далі оператор споряджає дрона, вводить дані про маршрут в систему навігації та натискає кнопку «Надіслати замовлення». Сам процес продемонстрований нижче на рис. 3.9 та рис. 3.10.



Рисунок 3.9 – Екран наявних замовлень

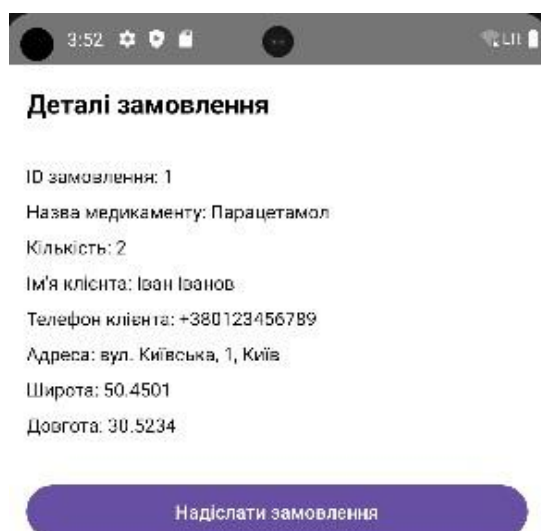


Рисунок 3.10 – Екран підтвердження

Отже, в цьому підрозділі було розглянуто, як проходить процес отримання/надсилання замовлення завдяки додатку. Багато змін, як в плані інтерфейсу, так в плані функціональності, ще буде внесено.

### 3.5 Вибір хмарного середовища

Під час проєктування та реалізації додатку було прийняте рішення про використання хмарного середовища для того, щоб забезпечити збереження та обробку даних, а ще для реалізації додаткових функцій.

Вибір був зроблений на користь платформи розробки мобільних та веб застосунків Firebase [15], адже вона легко інтегрується з середовищем розробки Android Studio, на платформі якого і було створено додаток. Також наведено ще декілька причин:

- простий та інтуїтивно зрозумілий інтерфейс;
- широкий спектр інструментів, таких як аутентифікація, база даних хмарні функції та ін.;
- легке масштабування від прототипу до великомасштабного додатка.

Firebase інтегровано з додатком для забезпечення наступного функціоналу:

- Firebase Realtime Database використовується для збереження даних про медикаменти, інформації про користувачів та статус замовлення;
- Firebase Authentication використовується для аутентифікації користувачів, щоб забезпечити доступ до додатку та в майбутньому до історії замовлень.

Саме з вищенаведених причин було обрано платформу розробки мобільних та веб застосунків Firebase, адже вона дозволить додатку забезпечити ефективну та надійну роботу.

Приклад використання Firebase наведено на рис. 3.11 та рис. 3.12.

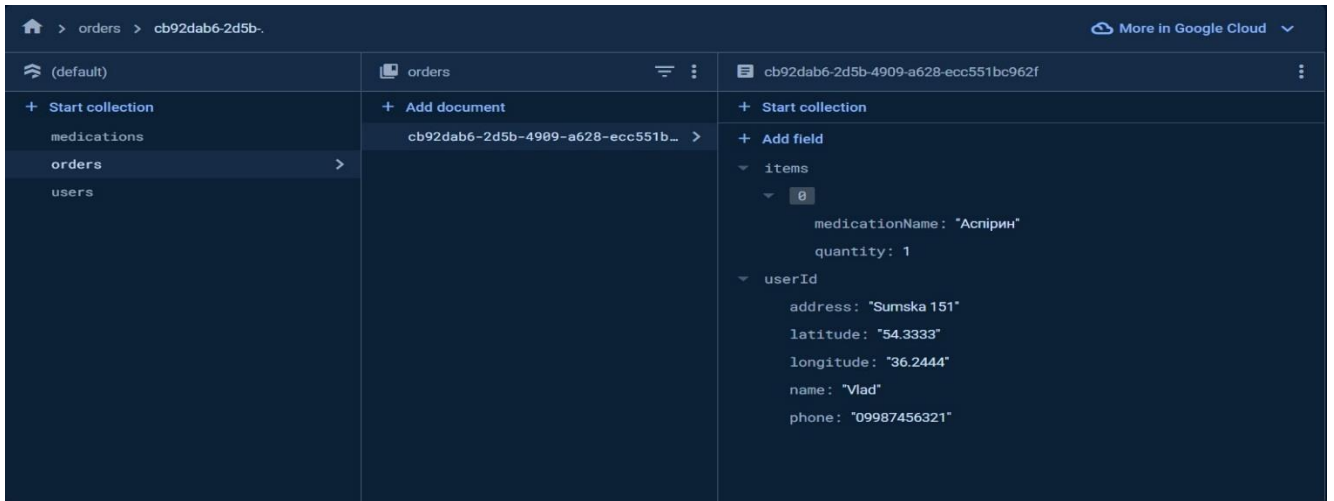


Рисунок 3.11 – База даних, яка використовується в додатку

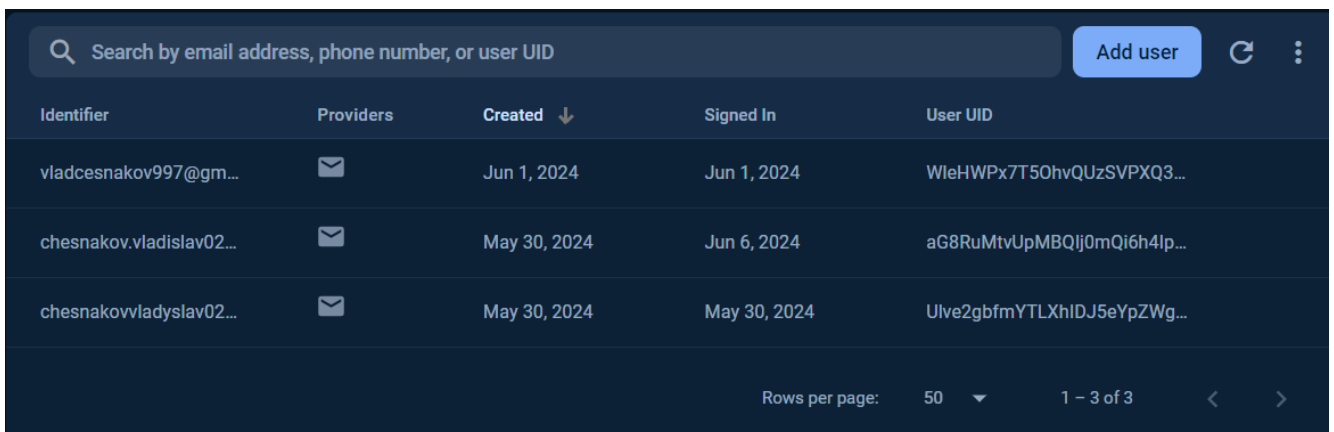


Рисунок 3.12 – Система аутентифікації з вже зареєстрованими користувачами

## 4 ВИБІР ПЛАТФОРМИ ТА РЕАЛІЗАЦІЯ АЛГОРИТМУ ПЛАНУВАННЯ МАРШРУТУ ДЛЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ ДОСТАВКИ

### 4.1 Огляд технічних характеристик дрона

Провівши аналіз вимог до системи та проаналізувавши багато дронів вибір був зроблений на користь DJI Matrice 300 RTK. Ось чому:

- дальність його польоту може досягати 15 км, вона дозволяє робити доставку в межах міста та в передмістя;
- максимальне навантаження – до 2,7 кг, цього цілком вистачить для доставки медикаментів;
- автономність до 55 хв польоту, чого цілком вистачає для більшості завдань доставки в межах міста;
- система навігації має високу точність;
- наявність сенсорів уникнення перешкод, функції повернення додому. Наявні сенсори забезпечують безпеку польоту;
- легка система управління, інтелектуальне планування маршрутів;
- стійкість до погодних умов.

Отже, даний варіант підходить для подальшої реалізації системи доставки медикаментів. Наступним кроком виконаємо реалізацію алгоритму автоматичного планування маршруту. Модель обраного дрона наведено на рис. 4.1.



Рисунок 4.1 – DJI Matrice 300 RTK

## 4.2 Реалізація алгоритму автоматичного планування маршруту

Обрана модель дрона має досить гарну вбудовану систему навігації, а також в нього можна інтегрувати алгоритми автоматичного планування маршруту.

Для того, щоб забезпечити ефективність та точність доставки, обрано алгоритм Дейкстри [8]. Даний алгоритм відомий своєю швидкістю та можливістю знаходити найкоротший шлях між вузлами в графі. Цей алгоритм дозволяє оптимально планувати маршрути для дронів, тим самим забезпечуючи ефективну доставку до місця призначення. В майбутньому він обов'язково буде інтегрований в платформу системи.

Нижче буде наведений приклад реалізації цього алгоритму на мові python на рис. 4.2.

```
Граф після ініціалізації:
{'A': {'B': 2, 'C': 3, 'G': 12}, 'B': {'A': 2, 'D': 4, 'E': 5, 'H': 13}, 'C': {'A': 3, 'D': 6, 'I': 14}, 'D': {'B': 4, 'C': 6, 'E': 7}, 'E': {'B': 5, 'D': 7, 'F': 8}, 'F': {'E': 8, 'I': 11, 'G': 15}, 'G': {'H': 9, 'A': 12, 'F': 15}, 'H': {'G': 9, 'I': 10, 'B': 13}, 'I': {'H': 10, 'F': 11, 'C': 14}}
Введіть початкову точку: A
Введіть кінцеву точку: I
Дрон рухається з A до A.
Маршрут: C -> I
Дрон рухається з A до C.
Маршрут: I
Дрон рухається з C до I.
Маршрут: завершено.
Граф після переміщення дрона:
{'A': {'B': 2, 'C': 3, 'G': 12}, 'B': {'A': 2, 'D': 4, 'E': 5, 'H': 13}, 'C': {'A': 3, 'D': 6, 'I': 14}, 'D': {'B': 4, 'C': 6, 'E': 7}, 'E': {'B': 5, 'D': 7, 'F': 8}, 'F': {'E': 8, 'I': 11, 'G': 15}, 'G': {'H': 9, 'A': 12, 'F': 15}, 'H': {'G': 9, 'I': 10, 'B': 13}, 'I': {'H': 10, 'F': 11, 'C': 14}}
```

Рисунок 4.2 – Реалізація алгоритму Дейкстри

Основними складовими цього коду є два класа «Graph» (Граф), «Drone» (дрон):

- клас «Graph» відповідає за представлення графа, на основі якого і відбувається планування маршруту. Він містить в собі методи для додавання вершин і ребер до графа;

- алгоритм Дейкстри реалізований у функціях «dijkstra» та «generate\_path»;

– клас «Drone» використовується для створення дрона. Має метод «move», який саме відповідальний за переміщення від однієї точки до іншої. Також під час руху враховуються можливі перешкоди.

Дана реалізація алгоритму є лише симуляцією роботи дрона та планування маршруту доставки. Вона використана для тестування та демонстрації концепції системи автоматизованої доставки.

Для майбутньої інтеграції з ПЗ дрона буде забезпечений зв'язок між програмним кодом та апаратною платформою дрона, також буде враховано специфічні можливості та обмеження дрона, такі як максимальна швидкість, обмеження на максимальну вагу тощо.

#### 4.3 Система автоматизованого управління для дрона DJI Matrice 300 RTK

У даному підрозділі зробимо детальний опис системи автоматизованого управління (CAU), що використовуються для оптимізації роботи обраного дрона DJI Matrice 300 RTK. Розглянемо детальніше компоненти CAU:

– сенсори: дрон використовує GPS для того, щоб визначати своє географічне положення. Також він має інерціальні вимірювальні пристрої (ІВП), такі як акселерометр та гіроскоп, що вимірюють прискорення та кутову швидкість. Ще є камери та сенсори обмеженого зору, які використовуються для того, щоб виявляти перешкоди та працювати в умовах обмеженої видимості;

– керуючі елементи: електроніка керування двигунами, яка регулює швидкість обертання пропелерів для керування рухом дрона. Також наявна система стабілізації, вона здійснює корегування руху для підтримки стабільного польоту забезпечуючи стабільність на всіх трьох осях;

– алгоритми керування: PID-регулятори, які використовуються для стабілізації руху та керування орієнтацією дрона. PID-контролери генерують керуючі сигнали, залежно від різниці між бажаним і фактичним станом системи. Також в системі є адаптивні керуючі алгоритми, які дозволяють системі адаптуватися до таких зміни умов, як маса дрона, або зміна атмосферних умов;

– алгоритми планування маршруту та уникнення перешкод: система використовує дані алгоритми для того, щоб визначити оптимальний маршрут до місця доставки, беручи до уваги відстань, швидкість вітру та обмеження маршруту. Також використовується алгоритм, який обробляє зображення та дані з сенсорів для того, щоб виявляти перешкоду та уникати зіткнення з нею.

Робота системи відбувається наступним чином:

- система отримує дані від сенсорів, таких як GPS, ІВІ, камери та сенсори обмеженого зору;
- отримавши дані, після цього на основі них, алгоритми керування аналізують ситуацію та приймають рішення щодо керування дроном;
- наступним кроком система генерує керуючі сигнали, які починають передаватися на керуючі елементи, такі як електроніка керування моторами та система стабілізації. Ці сигнали регулюють швидкість обертання пропелерів та коригують кут нахилу дрона для того, щоб підтримувати задану траєкторію;
- далі, отримавши керуючі сигнали, дрон виконує рух за заданою траєкторією. Система стабілізації виправляє будь-які відхилення в орієнтації дрона для підтримки стабільності та точності польоту;
- під час польоту система продовжує моніторити дані від сенсорів і коригувати керуючі сигнали в реальному часі для уникнення можливих перешкод.

#### 4.4 Опис пропорційно-інтегрально-диференціального (ПІД) регулятора

ПІД-регулятор часто використовується для управління дроном. Це програма в дроні, яка оцінює дані датчиків і відповідно керує двигунами через контролер двигуна [16]. Саме це робить його основним компонентом багатьох автоматизованих систем.

ПІД-регулятор використовує три основні компоненти для обчислення [17]:

- пропорційний (П) компонент: відповідає пропорційно поточній помилці. Даний компонент визначається компонентом  $K_p$ ;

– інтегральний (І) компонент: враховує суму попередніх помилок.

Визначається коефіцієнтом інтеграції  $K_i$ ;

– диференціальний (Д) компонент: враховує швидкість зміни помилки.

Визначається коефіцієнтом диференціації  $K_d$ .

ПД-закон управління має наступний вигляд [18]:

$$u(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt}, \quad (4.1)$$

де  $u(t)$  – керуючий сигнал;

$e(t)$  – поточна помилка;

$K_p$  – коефіцієнт пропорційності;

$K_i$  – коефіцієнт інтеграції;

$K_d$  – коефіцієнт диференціації.

В системі управління дроном ПД-регулятори застосовуються для стабілізації та контролю наступних параметрів:

– висота (Z-вісь): П-компонент регулює висоту, забезпечуючи швидку реакцію на відхилення від бажаного значення. І-компонент компенсує постійні відхилення. А Д-компонент пригнічує коливання, забезпечуючи стабільний підйом або спуск;

– нахил: П-компонент швидко реагує на відхилення від горизонтального положення. І-компонент усуває постійні нахили, що можуть виникнути через зовнішні впливи, такі як вітер. Д-компонент зменшує коливання стабілізуючи рух дрона;

– курс: П-компонент забезпечує точне підтримання напрямку. І-компонент усуває постійні відхилення в напрямку. Д-компонент пригнічує раптові зміни напрямку, забезпечуючи плавний поворот.

Необхідно правильно налаштувати параметри  $K_p$ ,  $K_i$ ,  $K_d$ , ПД-регулятора для оптимального його функціонування. Це виконується зазвичай методом проб і

помилки або за допомогою спеціальних алгоритмів налаштування. Налаштування включає такі основні етапи:

- налаштування П-компонента: збільшуємо  $K_p$  до тих пір, поки система не почне коливатись, потім трохи зменшуємо для досягнення стабільності;
- налаштування І-компонента: додаємо  $K_i$ , щоб усунути постійне зміщення, але треба бути обережним, щоб не викликати нестабільність;
- налаштування Д-компонента: додаємо  $K_d$ , щоб зменшити коливання та забезпечити плавність реакції системи на зміни.

Використання ПІД-регуляторів у САУ для дрона дозволяє забезпечити точне та стабільне керування польотом. Завдяки поєднанню П-, І-, Д-компонентів система може ефективно реагувати на зміни умов польоту та підтримувати бажану траєкторію, що є критично важливим для доставки медикаментів.

## 5 ЗАБЕЗПЕЧЕННЯ УМОВ ОХОРОНИ ПРАЦІ

Забезпечення умов охорони є надважливим аспектом в розробці системи автоматизації доставки медикаментів. Дотримання вимог охорони праці сприяє безпеці та здоров'ю працівників, підвищуючи ефективність та надійність роботи системи. Розглянемо детальніше умови для створення безпечних умов праці.

### 5.1 Робоче середовище та організація праці

Організація робочих місць розробників повинна відповідати ергономічним вимогам для зменшення навантаження на опорно-руховий апарат та очі [19]:

- монітори на рівні очей: монітори слід розташовувати на рівні очей або трохи нижче, щоб уникнути перенапруження шії та очей;
- регульовані робочі столи та крісла: робочі столи повинні мати здатність регулюватися по висоті, для забезпечення комфортної робочої позиції незалежно від росту працівника. Крісла повинні бути ергономічними з підтримкою поперекового відділу спини, регульованою висотою сидіння;
- перерви для очей та вправи: працівники повинні робити регулярні перерви для очей. Також слід робити вправи для розминки та зняття напруги з м'язів.

Забезпечення належного освітлення, яке не створює відблисків на моніторах:

- природне та штучне освітлення: за можливості треба використовувати природне освітлення. Штучне освітлення повинно бути рівномірним і достатньо яскравим, але не сліпучим. Використання світлодіодних ламп з регульованою яскравістю може забезпечити комфортне освітлення;
- антиблікові екрани та фільтри: використання антиблікових екранів або фільтрів на моніторах допоможе зменшити відблиски та підвищити комфорт для очей.

Зниження рівня шуму в приміщеннях до допустимих норм для того, щоб забезпечити комфортні умови праці та зменшити ризик стресу:

- розташування робочих зон: робочі місця повинні бути розташовані так, щоб уникати джерел шуму, таких як принтери та сервера. За можливості треба організувати окремі кімнати для обладнання, що генерує шум;

- контроль рівня шуму: регулярне вимірювання рівня шуму в робочих зонах для забезпечення відповідності до встановлених норм.

Дотримання цих вимог є критично важливим для підтримання здоров'я та продуктивності працівників, а також допоможе створити комфортні умови праці та сприятиме ефективній роботі команди.

## 5.2 Безпека експлуатації дронів

Критично важливим для забезпечення надійної та безпечної доставки медикаментів є безпека експлуатації дронів. Розглянемо основні заходи для забезпечення безпечної експлуатації.

Забезпечення якісного навчання операторів дронів з акцентом на безпечне управління та реагування на позаштатні ситуації:

- теоретичне навчання: оператори повинні отримати знання про принципи роботи дронів, їх характеристики та можливі ризики, пов'язані з їх використанням;

- практичне навчання: операторам слід пройти практичні заняття з управління дронами, включаючи симуляції позаштатних ситуацій, таких як відмова зв'язку, втрата зв'язку, несприятливі погодні умови;

- підвищення кваліфікації: операторам слід проходити регулярно курси підвищення кваліфікації, щоб бути в курсі новітніх технологій та методів безпечного управління дронами.

Проведення регулярних технічних оглядів та обслуговування дронів для запобігання технічним несправностям:

- планові перевірки: дрони повинні проходити планові огляди відповідно встановленого графіка, що включає в себе перевірку всіх основних систем і компонентів;

- заміна зношених деталей: всі пошкоджені або зношені деталі слід замінювати своєчасно для запобігання несправностям під час польотів;
- аналіз даних: використання ПЗ для моніторингу стану дронів і аналізу даних про їх роботу для своєчасного виявлення проблем.

Оператори повинні використовувати засоби індивідуального засобу (ЗІЗ) під час обслуговування дронів, зокрема рукавички, захисні окуляри та додаткові засоби:

- рукавички: захищають руки від механічних пошкоджень та контакту з хімічними речовинами, які можуть бути використані під час технічного обслуговування;
- захисні окуляри: використовуються для захисту очей від дрібних частинок, які можуть виникнути під час обслуговування дронів;
- додаткові засоби захисту: за необхідності використовуються додаткові ЗІЗ, спецодяг або взуття.

Дотримання даних заходів допоможе забезпечити безпеку експлуатації дронів, знизити ризики технічних несправностей та аварій, а також захистити здоров'я та безпеку працівників, які займаються обслуговуванням дронів.

### 5.3 Правила безпеки польотів

Забезпечення безпеки під час польотів дронів є не менш критично важливим для мінімізації ризиків пошкодження обладнання та запобігання нещасним випадкам. Розберемо правила які треба дотримуватися для забезпечення безпечної експлуатації.

Вибір безпечних місць для запуску та посадки дронів, щоб уникнути ризику пошкоджень та нещасних випадків:

- відкриті простори: слід обирати відкриту місцевість для посадки або запуску дронів, вони повинні бути вільними від перешкод, таких як дерева, будівлі тощо;

- плоскі поверхні: також слід використовувати плоскі та стабільні поверхні для зменшення ризику перекидання дрона під час взльоту або посадки;
- безпечні відстані: слід забезпечити безпечну відстань від людей, транспорту та інших джерел потенційної небезпеки.

Обмеження доступу сторонніх осіб до зон запуску та посадки дронів:

- зони обмеженого доступу: треба позначати зони запуску та посадки, як зони обмеженого доступу з використанням знаків та бар'єрів;
- контроль доступу: слід залучати персонал для контролю доступу до зон запуску та посадки дронів, щоб запобігти проникненню сторонніх осіб.

Оцінка погодних умов перед вильотом та ухвалення належного рішення при несприятливій погоді:

- прогнози погоди: регулярний моніторинг погодних умов за допомогою джерел прогнозу погоди;
- критичні умови: визначення критичних та погодних умов, таких як сильний вітер, дощ, що може вплинути на безпеку польоту дрона.

#### 5.4 Екологічна безпека

Дотримання екологічних норм та правил є однією з важливих складових відповідальної експлуатації дронів. Врахування екологічної безпеки допоможе зменшити негативний вплив на навколишнє середовище.

Організація безпечної утилізації використаних батарей та інших компонентів дронів для зменшення впливу на навколишнє середовище:

- збір та сортування: використанні батареї та інші компоненти дронів повинні збиратися окремо та сортуватися для подальшої утилізації;
- утилізаційні центри: співпраця з ліцензованими утилізаційними центрами, які мають відповідні дозволи на переробку та утилізацію електронних відходів;
- безпека процесу: створення умов безпечного зберігання використаних батарей до їх передачі на утилізацію, щоб уникнути ризику займання або витікання небезпечних речовин.

Використання дронів з низьким рівнем шуму для зменшення впливу на довкілля та людей:

- вибір дронів: треба надавати перевагу моделям дронів з низьким рівнем шуму, що відповідають вимогам шумового забруднення;
- шумознижувальні технології: використання дронів, обладнаних сучасними технологіями шумозниження, такими як спеціальні пропелери або матеріали корпусу.

Забезпечення умов охорони праці є невід'ємною частиною розробки та впровадження системи автоматизації доставки медикаментів. Дотримання зазначених вимог сприятимуть ефективності, безпеці та надійності роботи системи.

## ВИСНОВКИ

У результаті аналізу різних джерел інформації за темою кваліфікаційної роботи було розглянуто основні вимоги та функціональні можливості до автоматизованої системи доставки медикаментів. Система повинна забезпечувати стабільну та ефективну роботу у важкодоступних місцях. Дрони повинні мати гарну автономність, а також мати вбудовану систему навігацію, а також протидіяти радіоелектронним та іншим видам перешкод. Повинні мати конструкцію, яка буде стійкою до фізичних пошкоджень. Мати достатній запас енергії для забезпечення довготривалого польоту. І на останок бути готовими працювати в різних кліматичних умовах.

Проведено аналіз існуючих алгоритмів автоматичного планування маршруту, було проаналізовано такі алгоритми, як: A\* (A-star), алгоритм Дейкстри, алгоритм Флойда-Уоршела та мурашиний алгоритм. Було прийняте рішення, щодо використання алгоритму Дейкстри в системі. Цей алгоритм має гарну швидкість та дозволяє оптимально планувати маршрути для дронів, завдяки можливості знаходити найкоротший шлях. Його інтеграція в систему забезпечить ефективну доставку до місця призначення, враховуючи оптимізацію маршрутів.

Створено додаток для користувача та оператора, для зручності взаємодії із системою. Для створення було використано середовище Android Studio, МП Kotlin, адже ця мова надає змогу писати код компактніше та виразніше, що прискорює процес розробки. Також він сприяє збільшенню продуктивності додатків, завдяки статичній типізації та функціональним можливостям. Також було обране хмарне середовище Firebase, воно легко інтегрується в проекти та дозволяє розширити їх функціонал.

Здійснено вибір платформи для системи. Обраний дрон DJI Matrice 300 RTK, він має вбудовану систему навігації, що дозволяє інтегрувати алгоритм автоматичного планування маршруту. Описано систему автоматизованого

управління. А також було здійснено опис ПД-регулятора, який використовується для стабілізації польоту.

Розроблений додаток та інтегрований алгоритм забезпечать ефективну та надійну роботу системи автоматизованої доставки, що є дуже важливим для медичних потреб.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. ДСТУ 3008-15. Документація. Звіти у сфері науки та техніки. структура та правила оформлення. Введ. 2015-06-22. К. Держстандарт України, 2017. – 29 с.
2. Методичні вказівки з підготовки кваліфікаційної роботи бакалавра для студентів усіх форм навчання спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» / Упоряд.: І. Ш. Невлюдов, А. О. Андрусевич, О. В. Токарева, С. П. Новоселов, О. В Сичова. Харків: ХНУРЕ, 2022. – 55 с.
3. Закон України «Про вищу освіту» зі змінами 2022 рік №1556-VII від 01.07.2014, редакція від 01.01.2022. [Електронний ресурс]: Режим доступу:<https://zakon.rada.gov.ua/laws/show/1556-18#Text>.
4. Положення про організацію освітнього процесу у ХНУРЕ [Електронний ресурс] : Режим доступу: [https://nure.ua/wp-content/uploads/Main\\_Docs\\_NURE/polozhennja-proorganiza-ciju-osvitnogo-procesu-vhnure.pdf](https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennja-proorganiza-ciju-osvitnogo-procesu-vhnure.pdf).
5. Дипломне проектування для студентів усіх форм навчання спеціальностей 151 «Автоматизація та комп'ютерно-інтегровані технології»: довід. / І.Ш. Невлюдов, А.О. Андрусевич, О.В. Токарева, Г.В. Пономарьова..: Київ-56, пр. Космонавта Комарова, 1, 2016. 320 с.
6. Положення про протидію академічному плагіату в ХНУРЕ / nure.ua. URL: [https://nure.ua/wp-content/uploads/Main\\_Docs\\_NURE/polozhennya-pro-protidiyu-akademichnomu-plagiatu-v-hnure-290-vid-28.04.2017.pdf](https://nure.ua/wp-content/uploads/Main_Docs_NURE/polozhennya-pro-protidiyu-akademichnomu-plagiatu-v-hnure-290-vid-28.04.2017.pdf)
7. Ami`ts A\* Pages [Електронний ресурс]. – Режим доступу: [www/URL:https://theory.stanford.edu/~amitp/GameProgramming/](http://www.URL:https://theory.stanford.edu/~amitp/GameProgramming/).
8. Programiz [Електронний ресурс]. – Режим доступу: [www/URL:https://www.programiz.com/dsa/dijkstra-algorithm](http://www.URL:https://www.programiz.com/dsa/dijkstra-algorithm).
9. Programiz [Електронний ресурс]. – Режим доступу: [www/URL:https://www.programiz.com/dsa/floyd-warshall-algorithm](http://www.URL:https://www.programiz.com/dsa/floyd-warshall-algorithm).
10. GeegksforGeeks [Електронний ресурс]. – Режим доступу: [www/URL:](http://www.URL:)

<https://www.geeksforgeeks.org/introduction-to-ant-colony-optimization/>.

11. OREILLY [Електронний ресурс]. – Режим доступу: [www/URL:https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html](http://www.URL:https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html).

12. Microsoft Learn [Електронний ресурс]. – Режим доступу: [www/URL:https://learn.microsoft.com/en-us/previous-versions/xamarin/get-started/what-is-xamarin](http://www.URL:https://learn.microsoft.com/en-us/previous-versions/xamarin/get-started/what-is-xamarin).

13. GeeksforGeeks[Електронний ресурс]. – Режим доступу: [www/URL:https://www.geeksforgeeks.org/overview-of-android-studio/](http://www.URL:https://www.geeksforgeeks.org/overview-of-android-studio/).

14. ВІКІ ЦДУ [Електронний ресурс]. – Режим доступу: [www/URL:https://wiki.cusu.edu.ua/index.php/Nativescript](http://www.URL:https://wiki.cusu.edu.ua/index.php/Nativescript).

15. Avada-Media [Електронний ресурс] – Режим доступу: <https://avada-media.ua/ua/services/firebase/>.

16. OscarLiand [Електронний ресурс] – Режим доступу: <https://oscarliang.com/pid/>.

17. Mituhiko Araki Control Systems, Robotics, and Automation – Vol. VII - PID control [Електронний ресурс] / Mituhiko Araki. – 2009. – Режим доступу: <https://www.eolss.net/ebooks/Sample%20Chapters/C18/E6-43-03-03.pdf>.

18. Невлюдов І. Ш. Теорія автоматичного управління : навч. посіб. / І. Ш. Невлюдов, О. В. Токарева; Харків. нац. ун-т радіоелектроніки. – Харків: Панов А. М., 2020. – 240 с.

19. Інструкція з охорони праці для головного спеціаліста з інформаційних технологій та оператора комп'ютерного набору [Електронний ресурс] – Режим доступу: <https://services.uteka.ua/ua/publication/zrazky-34-trudovi-vidnosyny-ta-oplata-pratsi-138-instrukciya-po-oxrane-truda-dlya-glavnogo-specialista-po-informacionnym-texnologiyam-i-operatora-kompyuternogo-nabora>