

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

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

До захисту допущено:

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційне забезпечення
робототехнічних систем»**

спеціальності 126 «Інформаційні системи та технології»

**на тему: «Автоматизована система оцінки якості харчових продуктів на
основі комп'ютерного зору в робототехнічних системах візуального кон-
тролю»**

Виконав:

студент IV курсу, групи ІК-11
Яценко Аркадій Станіславович _____

Керівник:

Старший викладач ІСТ,
Густера Олег Михайлович _____

Рецензент:

Доцент, к. е. н., доцент кафедри ІТ
КНУ імені Тараса Шевченка,
Подскребко Олександр Сергійович _____

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

Студент _____

Київ – 2025 року

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

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«__» _____ 2025 р.

ЗАВДАННЯ

на дипломний проєкт студенту
Яценку Аркадію Станіславовичу

1. Тема проєкту «Автоматизована система оцінки якості харчових продуктів на основі комп'ютерного зору в робототехнічних системах візуального контролю», керівник проєкту Густера Олег Михайлович, старший викладач кафедри ІСТ, затверджені наказом по університету від «23» травня 2025 р. № 1705-с
2. Термін подання студентом проєкту: «12» червня 2025 року;
3. Вихідні дані до проєкту: мова програмування Python, середовище розробки PyCharm;
4. Зміст пояснювальної записки: аналіз готових систем, порівняння наявних рішень, вибір необхідних технологій для розробки, опис особливостей технічної реалізації, огляд готової системи;
5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо): діаграма компонентів, блок-схема алгоритму, діаграма прецедентів, діаграма класів;
6. Дата видачі завдання: «6» березня 2025 року.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд та аналіз існуючих рішень	До 23.03.2025	Виконано
2	Формування вимог до проєкту	До 06.04.2025	Виконано
3	Вибір технологій та інструментів розробки	До 20.04.2025	Виконано
4	Проектування архітектури системи	До 27.04.2025	Виконано
5	Проектування і розробка бази даних	До 04.05.2025	Виконано
6	Розробка клієнтської частини системи	До 11.05.2025	Виконано
7	Інтеграція модуля комп'ютерного зору	До 18.05.2025	Виконано
8	Налагодження взаємодії всіх систем	До 25.05.2025	Виконано
9	Передзахист дипломного проєкту	26.05.2025	Виконано
10	Тестування розробленої системи	До 09.06.2025	Виконано
11	Захист дипломного проєкту	3 16.06.2025	

Студент

Аркадій ЯЦЕНКО

Керівник

Олег ГУСТЕРА

АНОТАЦІЯ

Яценко А. С. Автоматизована система оцінки якості харчових продуктів на основі комп'ютерного зору в роботизованих засобах візуального контролю.

Проект містить 60 сторінок тексту, 38 рисунків, 3 таблиці, посилання на 21 літературне джерело, 4 кресленика та додатки.

КОНТРОЛЬ ЯКОСТІ, ХАРЧОВІ ПРОДУКТИ, КОМП'ЮТЕРНИЙ ЗІР, YOLOV8, TELEGRAM-БОТ, PYTHON, SQLITE.

Об'єкт дослідження – процеси контролю якості харчових продуктів у харчовій промисловості.

Метою роботи є створення інтерактивної системи, яка дозволяє користувачам здійснювати оцінку якості харчових продуктів через надсилання зображень до Telegram-бота, з подальшим аналізом за допомогою моделі глибокого навчання.

Для реалізації проекту було використано мову програмування Python та фреймворк YOLOv8. Система реалізована як Telegram-бот із графічним інтерфейсом, побудованим на бібліотеці "python-telegram-bot". Для зберігання історії аналізів та користувачів застосовано реляційну базу даних SQLite, роботу з якою організовано через ORM-бібліотеку SQLAlchemy. Для створення власного датасету та анотацій використано сервіс Roboflow. Розробка здійснювалася в середовищі PyCharm.

Створено повноцінну автоматизовану систему, яка забезпечує розпізнавання дефектів на фотографіях продуктів, формування текстових звітів, рекомендацій, збереження результатів, а також розмежування доступу користувачів за ролями.

Розроблена система є адаптивною, масштабованою, має інтуїтивно зрозумілий інтерфейс, працює швидко та точно завдяки використанню сучасних моделей глибокого навчання. Отримані результати можуть бути використані в харчовій промисловості, роздрібній торгівлі та аграрному секторі для підвищення ефективності контролю якості продукції.

SUMMARY

Yatsenko A. S. Automated System for Food Quality Assessment Based on Computer Vision in Robotic Visual Inspection Tools.

The project contains 60 pages of text, 38 figures, 3 tables, references to 21 literary sources, 4 technical drawings, and appendices.

QUALITY CONTROL, FOOD PRODUCTS, COMPUTER VISION, YOLOV8, TELEGRAM BOT, PYTHON, SQLITE.

Object of the research – quality control processes of food products in the food industry.

The aim of the work is to develop an interactive system that enables users to assess the quality of food products by sending images to a Telegram bot, followed by analysis using a deep learning model.

To implement the project, the Python programming language and the YOLOv8 framework were used. The system is realized as a Telegram bot with a graphical interface based on the “python-telegram-bot” library. SQLite, a relational database, is used for storing analysis history and user data, managed through the SQLAlchemy ORM library. The Roboflow service was used to create a custom dataset and annotations. Development was carried out in the PyCharm environment.

A fully functional automated system was developed that ensures the detection of defects on product images, generation of text reports and recommendations, storage of results, as well as user access differentiation by roles.

The developed system is adaptive, scalable, has an intuitive interface, and operates quickly and accurately due to the use of modern deep learning models. The obtained results can be applied in the food industry, retail, and agriculture sectors to improve the efficiency of product quality control.

№ рядка	Формат	Позначення	Найменування	Кіл. аркушів	№ екз.	Примітка
1			<u>Документація загальна</u>			
2			Знову розроблена			
3	A4	ІК11.310БАК.006 ПЗ	Пояснювальна записка	60		
4	A3	ІК11.310БАК.006 Д1	Автоматизована система оцінки	1		
5			якості харчових продуктів на			
6			основі комп'ютерного зору в			
7			робототехнічних системах.			
8			візуального контролю.			
9			Діаграма компонентів			
10	A3	ІК11.310БАК.006 Д2	Автоматизована система оцінки	1		
11			якості харчових продуктів на			
12			основі комп'ютерного зору в			
13			робототехнічних системах			
14			візуального контролю.			
15			Блок-схема алгоритму			
16	A3	ІК11.310БАК.006 Д3	Автоматизована система оцінки	1		
17			якості харчових продуктів на			
18			основі комп'ютерного зору в			
19			робототехнічних системах			
20			візуального контролю.			
21			Діаграма прецедентів			
22	A3	ІК11.310БАК.006 Д4	Автоматизована система оцінки	1		
23			якості харчових продуктів на			
24			основі комп'ютерного зору в			
25			робототехнічних системах			
26			візуального контролю.			
27			Діаграма класів			

				ІК11.310БАК.006 ТП					
Зм.	Лист	№ докум.	Підпис						
Розробив	Яценко А. С			Автоматизована система оцінки якості харчових продуктів на основі комп'ютерного зору в робототехнічних засобах візуального контролю.			Літ.	Арк.	Аркушів
Перевірив	Густера О.М						Т		1
Затв.				Відомість дипломного проекту			КПІ ім. Ігоря Сікорського Група ІК-11		

Пояснювальна записка
до дипломного проєкту
на тему: «Автоматизована система оцінки якості харчових продуктів на основі комп'ютерного зору в робототехнічних системах візуального контролю»

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП.....	5
1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Загальні положення.....	9
1.2 Огляд існуючих рішень.....	12
1.2.1 Класифікація сервісів за призначенням та способом використання.....	12
1.2.2 Веб-платформа Odoо Inventory.....	13
1.2.3 Мобільний застосунок WithoutWire.....	14
1.2.4 Рішення SAP ERP QM.....	15
1.2.5 Порівняння сервісів.....	16
Висновок до розділу 1.....	17
2 ФОРМУВАННЯ ВИМОГ ТА ВИБІР ТЕХНОЛОГІЙ І СЕРВІСІВ ДЛЯ РОЗРО- БКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ.....	18
2.1 Вибір технологій та сервісів.....	18
2.1.1 Мова програмування Python.....	18
2.1.2 Фреймворк комп'ютерного зору YOLOv8.....	19
2.1.3 Бібліотека python-telegram-bot(Telegram Bot API).....	19
2.1.4 СУБД SQLite.....	20
2.1.5 Бібліотека ORM SQLAlchemy.....	20
2.1.6 Сервіс Roboflow.....	20
2.1.7 Середовище розробки PyCharm.....	21
2.1.8 Інші бібліотеки та сервіси.....	22
2.2 Функціональні вимоги.....	22
2.3 Нефункціональні вимоги.....	25
Висновок до розділу 2.....	26

				ІК11.310БАК.006 ПЗ			
Зм.	Лист	№ докум.	Підпис				
Розробив	Яценко А.С.			Автоматизована система оцінки якості харчових продуктів на основі комп'ютерного зору в робототехнічних засобах візуального контролю.			Літ.
Перевірив	Густера О.М						Т
							2
							71
Затв.				Пояснювальна записка			КПІ ім. Ігоря Сікорського Група ІК-11

3 РОЗРОБКА ТА АНАЛІЗ ПРОГРАМНОГО ПРОДКТУ.....	28
3.1 Архітектура автоматизованої системи.....	28
3.1.1 Загальне визначення архітектури систем.....	28
3.1.2 Поширені архітектурні підходи у даній сфері.....	28
3.1.3 Основні компоненти архітектури системи.....	33
3.1.4 Взаємодія між компонентами архітектури.....	33
3.2 Визначення ролей користувачів.....	34
3.3 Розробка клієнтської частини.....	35
3.4 Розробка бази даних.....	38
3.4.1 Технології для роботи з базою даних.....	39
3.4.2 Структура бази даних.....	39
3.4.3 Підключення до бази даних та взаємодія з ORM.....	42
3.4.4 Взаємодія з ORM.....	43
Висновки до розділу 3.....	44
4 ДЕМОНСТРАЦІЯ РОБОТИ СИСТЕМИ.....	45
4.1 Звичайний користувач.....	45
4.2 Адміністратор.....	55
Висновки до розділу 4.....	57
ВИСНОВКИ.....	59
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	61
ДОДАТОК А.....	63
ДОДАТОК Б.....	64
ДОДАТОК В.....	66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД – база даних.

СУБД – система управління базою даних.

ШІ – штучний інтелект.

API – Application Programming Interface, сукупність засобів та правил, що вможливають взаємодію між окремими складниками програмного забезпечення..

CV – Computer Vision, напрямок ШІ, на якому базується логіка аналізу фото..

ORM – Object-Relational Mapping.

SDK – Software Development Kit.

SQL – Structured Query Language, мова запитів до бази даних SQLite..

SQLAlchemy – ORM-бібліотека для Python.

SQLite – вбудована СУБД.

YOLOv8 – You Only Look Once v8.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

ВСТУП

У сучасному світі автоматизація різних процесів стає невід'ємною частиною будь-якої галузі знань, особливо в сфері харчової промисловості та логістики. Ринок харчових продуктів вимагає високої якості продукції, а також контролю за станом кожної одиниці товару. Традиційні методи контролю якості, які переважно базуються на ручному огляді, супроводжуються низкою проблем, таких як суб'єктивність оцінки, низька продуктивність, помилки людини та значні фінансові витрати. Саме тому впровадження автоматизованих систем контролю якості харчових продуктів, які використовують комп'ютерний зір та технології штучного інтелекту, стає надзвичайно актуальним.

Особливого значення набуває застосування таких систем саме на робототехнічних платформах візуального контролю, де вони можуть виконувати функції перевірки якості безперервно, швидко та не залежачи від впливу людини, її фактору. Використання комп'ютерного зору дає змогу не лише виявляти дефекти, а й надавати рекомендації щодо подальшого використання продуктів харчування, що підвищує ефективність, швидкість обробки і безпеку виробничих процесів.

Нині, в епоху технологій, стартапів та вільного доступу людей до технологій розробки додатків та сервісів, особливо тих, що не потребують особливих ресурсо-вкладень, існує серйозна конкуренція. Ринок пропонує найрізноманітніші рішення для будь-яких задач. Це гонка, яку виграє найвправніший, той, хто зможе підлаштуватися під поточну ситуацію. Потрібно боротися за кожного покупця, бо при невдачі він піде до прямих конкурентів, що невідворотно призведе до матеріальних втрат.

Об'єктом дослідження в даній роботі є процес візуального контролю якості харчових продуктів у робототехнічних системах на основі комп'ютерного зору. Це стосується автоматизованих рішень, які інтегруються в лінії сортування, пакування, переробки та інших етапах виробництва.

Предметом дослідження є технології та методи автоматизованої обробки зображень харчових продуктів для виявлення дефектів та формування рекомендацій

					ІК11.310БАК.006 ПЗ	Арк.
						5
Зм.	Лист	№ докум.	Підпис	Дата		

щодо їх стану. Особлива увага приділяється використанню глибоких нейронних мереж для підвищення точності та швидкодії процесу контролю.

Метою даної роботи є розробка та впровадження автоматизованої системи контролю якості харчових продуктів на базі комп'ютерного зору в робототехнічних системах візуального контролю. Така система має бути здатна обробляти зображення в реальному часі, точно виявляти дефекти, формувати звіти та мати змогу інтегруватися з робототехнічними платформами.

Для того, щоб досягнути цієї мети необхідно вирішити такі задачі:

- 1) проаналізувати існуючі рішення у сфері комп'ютерного зору та автоматизації контролю якості продуктів харчування;
- 2) дослідити сучасні моделі глибокого навчання та їх застосування для виявлення дефектів на харчових продуктах;
- 3) розробити архітектуру системи, яка включатиме модулі комп'ютерного зору, інтерфейс користувача та систему зберігання даних;
- 4) реалізувати базу даних для зберігання результатів аналізу та історії перевірок користувачів;
- 5) створити користувацький інтерфейс, який дозволить користувачам взаємодіяти із системою;
- 6) провести тестування та оцінку ефективності роботи системи, визначити точність виявлення дефектів, швидкість роботи та зручність використання.

Практичне значення одержаних результатів полягає у можливості впровадження розробленої системи в реальні виробничі лінії підприємств харчової галузі, агросектору, логістики та суміжних сфер. Це дозволить автоматизувати процеси перевірки якості продукції, зменшити вплив людського фактору, підвищити швидкість та точність контролю, а також забезпечити безперервний моніторинг стану харчових продуктів у режимі реального часу. Система також може бути масштабована та адаптована для роботи з різними видами харчових продуктів та різними типами робототехнічних платформ. Виконавши всі заплановані етапи розробки, можна досягти таких результатів:

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

1) підвищення точності контролю якості харчових продуктів за рахунок використання сучасних моделей глибокого навчання. Це призведе до зменшення помилок, та, відповідно, збільшення якості продукції;

2) зменшення часу перевірки та швидке прийняття рішень про придатність продукції. У результаті буде збільшено об'єми обробленої продукції за одиницю часу;

3) можливість інтеграції системи в роботизовані виробничі лінії, що забезпечить автоматизацію процесу та мінімізацію людського втручання. Це призведе до значного зменшення витрат на персонал, та, відповідно, збільшить прибутки компанії;

4) забезпечення безпеки та збереження історії перевірок користувачів. Це дозволить краще контролювати процес роботи;

5) гнучкість та масштабованість, що дозволить адаптувати систему під потреби конкретного підприємства. У результаті індивідуальний підхід до кожного клієнта призведе до зростання кількості замовлень на систему.

У першому розділі пояснювальної записки розглядається важливість автоматизованого контролю якості у харчовій промисловості та застосування комп'ютерного зору для цього завдання. Аналізуються ключові проблеми галузі та потреба у впровадженні автоматизації. Проводиться порівняння, аналіз та оцінка найвідоміших готових рішень у даній сфері.

У другому розділі досліджуються сучасні технології та бібліотеки, які використовуються при створенні системи контролю якості продуктів харчування. Зокрема, приділяється увага технологіям глибокого навчання та інструментам комп'ютерного зору. Після цього вже буде складене уявлення про майбутню систему.

У третьому розділі проводиться безпосередня розробка системи. Описується архітектура програмного забезпечення, розглядаються компоненти клієнтської та серверної частини, а також база даних та взаємодія між модулями системи. До всього розглядається призначення окремих компонентів даного проєкту на прикладі програми, що розробляється.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

У четвертому розділі демонструється готовий програмний продукт. Показуються результати тестування, аналізуються можливості системи, її переваги та обмеження. Додатково наводиться інструкція для користувачів, яка пояснює принципи роботи із системою.

Таким чином, ця робота охоплює всі етапи від дослідження та аналізу проблеми до розробки, впровадження та тестування системи, забезпечуючи всебічний підхід до вирішення задачі автоматизованого контролю якості харчових продуктів у сучасних робототехнічних системах.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		8

1 ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальні положення

У наш час автоматизовані системи проникають практично в усі галузі людської діяльності. Їх застосування значно пришвидшує виконання повторюваних завдань, мінімізує людський фактор, знижує витрати та підвищує загальну ефективність роботи підприємств. Особливо активно автоматизація розвивається у сфері харчової промисловості, де ключовим є забезпечення високої якості та безпеки продукції на всіх етапах — від виробництва до постачання споживачу.

Сучасна харчова промисловість, агросектор і логістика потребують постійного моніторингу стану продукції. Традиційні методи, засновані на ручній перевірці, виявляються неефективними в умовах масштабного виробництва. Людина може втомитися, допустити помилку, не помітити дефект або неправильно класифікувати товар. У таких умовах на перший план виходять автоматизовані системи візуального контролю, які здатні швидко, точно й безперервно аналізувати зображення продуктів, виявляючи на них характерні ознаки дефектів.

Контроль якості харчових продуктів завжди був однією з ключових ланок у виробничих процесах підприємств агропромислового комплексу, логістичних компаній, рітейлу та інших суміжних галузей. Основна мета цього етапу – забезпечити споживачеві безпечний та якісний продукт, який відповідає всім встановленим стандартам та вимогам.

У класичній моделі організації виробництва перевірка якості продуктів здійснюється переважно візуально та вручну, шляхом інспектування кожної партії або вибіркової перевірки. Така система базується на роботі кваліфікованих працівників, інспекторів або операторів, які проводять аналіз зовнішнього вигляду, кольору, запаху, текстури та інших органолептичних характеристик. У деяких випадках застосовуються ручні або напівавтоматичні вимірювальні прилади, зокрема на овочевих та фруктових виробництвах інспектори перевіряють рівномірність забарвлення, відсутність механічних пошкоджень, цвілі та гнилі. У молочній промисло-

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		9

вості проводиться перевірка кольору, запаху та консистенції продукції. На м'ясо-переробних підприємствах — контроль кольору, структури, наявності сторонніх включень, а також забруднень або порушень пакування.

Проте така система перевірки має низку критичних недоліків, це може бути висока ймовірність людського фактору: людина може пропустити дрібні дефекти або через втому або неуважність прийняти неправильне рішення. Також додається низька пропускна здатність: при великих обсягах виробництва перевірити кожен продукт вручну стає практично неможливо, а отже значна частина браку потрапляє до споживача. Проблемою є висока собівартість перевірки: підприємства змушені утримувати великий штат контролерів якості, що збільшує витрати на заробітну плату, навчання та управління персоналом [1]. Недоліком можна назвати ще складність стандартизації: оцінка дефектів часто має суб'єктивний характер і залежить від досвіду інспектора. Це створює неоднозначність у рішенні про якість партії товару. Врешті можуть виникнути затримки у виробництві: ручний контроль вимагає зупинки конвеєра або виробничої лінії для відбору зразків, що призводить до втрати часу, а іноді — і до псування партій продуктів.

За даними досліджень FAO (Продовольча та сільськогосподарська організація ООН), приблизно 30% від усіх харчових продуктів на глобальному ринку втрачаються або відбувається псування саме через неналежний контроль якості на етапах збирання, переробки або транспортування. Це завдає колосальних економічних збитків як самим виробникам, так і споживачам [2].

Як приклад з практики можна навести такий випадок: на підприємствах із виробництва яблук працівники оглядають кожне яблуко візуально, перевіряють його колір, наявність пошкоджень або цвілі. За день працівник може перевірити близько 500–700 кг продукції, при цьому точність виявлення дефектів становить лише 70–80% через втому та монотонність роботи. При автоматизованій лінії комп'ютерного зору ця продуктивність може зрости до 3000–5000 кг/год, а точність — до 95% і більше [3].

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		10

У зв'язку зі зростанням масштабів виробництва харчових продуктів у світі, підприємства стикаються з дедалі більш жорсткими вимогами до якості, зростанням конкуренції, посиленням санітарних норм та збільшенням обсягів продукції. Ці фактори ускладнюють ручний контроль якості, вимагаючи переходу на нові, більш ефективні методи.

Аналітики прогнозують, що до 2030 року понад 70% підприємств харчової промисловості перейдуть на автоматизовані системи контролю якості на основі комп'ютерного зору та штучного інтелекту [4]. Це дозволить значно знизити ризики потрапляння дефектної продукції на полиці магазинів і одночасно оптимізувати виробничі витрати.

Для ілюстрації проблеми можна навести дані останніх звітів(рис. 1.1).



Рисунок 1.1 — Розподіл типів дефектів у харчовій продукції за результатами вибіркового контролю [5]

Як видно з діаграми, більшість дефектів пов'язані з механічними пошкодженнями та зміною кольору, які найважче відстежувати вручну.

Таким чином, традиційна модель перевірки якості є застарілою та недостатньо ефективною в сучасних умовах. Для розв'язання цих проблем та підвищення якості продукції підприємства повинні впроваджувати сучасні системи автоматизованого контролю на основі комп'ютерного зору. Саме ця технологія дозволить нашій системі значно підвищити ефективність та надійність перевірки харчових продуктів, що буде розглянуто у наступних підрозділах.

1.2 Огляд існуючих рішень

1.2.1 Класифікація сервісів за призначенням та способом використання

У сучасному світі існує широкий спектр сервісів та програмних рішень, призначених для автоматизації контролю якості харчових продуктів. В залежності від призначення та способу використання ці рішення можна умовно поділити на такі категорії:

1) вебплатформи: призначаються для централізованого контролю якості продуктів через браузер. Використовуються таким способом: користувачі завантажують фото чи дані, система аналізує їх і повертає результати через вебінтерфейс. В якості прикладів можна навести такі сервіси як: Odoc Inventory, Fishbowl Inventory;

2) мобільні застосунки: призначаються для забезпечення мобільного доступу до сервісу перевірки якості продуктів у режимі реального часу. Під час користування користувачі встановлюють застосунок на телефон, роблять фото чи сканують продукт і отримують результат. Приклади застосунків: WithoutWire, Inspecto App;

3) ERP-рішення з модулем контролю якості: призначені для комплексного управління бізнес-процесами, включаючи контроль якості продукції. Використовуються за допомогою інтеграції модуля контролю якості у всю ERP-систему(склади, логістика, закупівлі). Приклади: SAP ERP QM, Oracle Quality Management;

4) хмарні сервіси для обробки зображень: призначаються для надання API або вебсервісів для автоматичного аналізу зображень харчових продуктів. Спосіб

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		12

використання: інтеграція з іншими системами через API або прямий доступ для завантаження зображень. В якості прикладів можна навести Clarifa та AWS Rekognition.

Ця класифікація дозволяє визначити основні напрями розвитку ринку та підкреслює важливість вибору саме того типу рішення, який найкраще відповідає конкретним бізнес-потребам.

1.2.2 Веб-платформа Odoo Inventory

Odoo Inventory - це застосунок, який дозволяє ефективно керувати запасами на складах та забезпечує управління складською логістикою. Він є складовою частиною ERP-системи Odoo та використовується для відстеження руху товарів, контролю запасів та планування складських процесів [6]. Зображення інтерфейсу веб-застосунку зображено нижче(рис. 1.2)

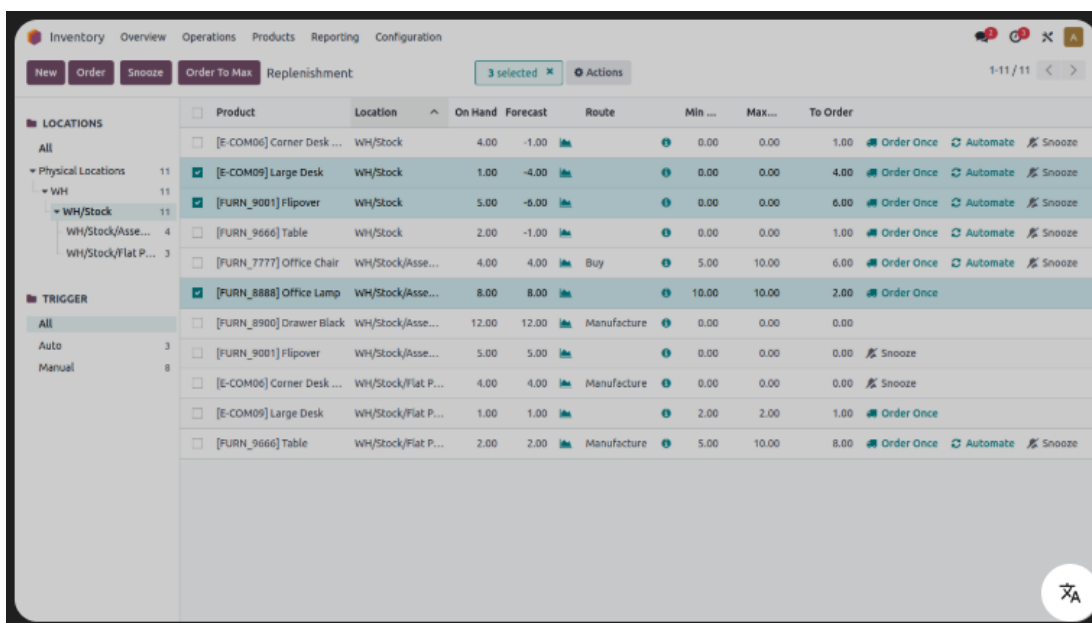


Рисунок 1.2 – Інтерфейс веб-застосунку Odoo Inventory

Розробник: Odoo S.A. (Бельгія).

Технології: Python (базовий стек), PostgreSQL, веб-фреймворк Odoo.

Функціональність:

- 1) управління запасами та інвентаризація;
- 2) відстеження партій продуктів;
- 3) підтримка контролю якості з фіксацією дефектів;
- 4) автоматичне формування звітів.

Застосування: харчова промисловість, дистрибуція, логістика.

Переваги: інтеграція з іншими модулями ERP (закупівлі, продажі, виробництво) та гнучке налаштування бізнес-процесів.

Недоліки: висока вартість ліцензії та потребує навчання персоналу.

1.2.3 Мобільний застосунок WithoutWire

WithoutWire — це мобільний додаток для управління складом, розроблений для підприємств із складними потребами в інвентаризації [7]. Зображення інтерфейсу сервісу зображено нижче(рис. 1.3).

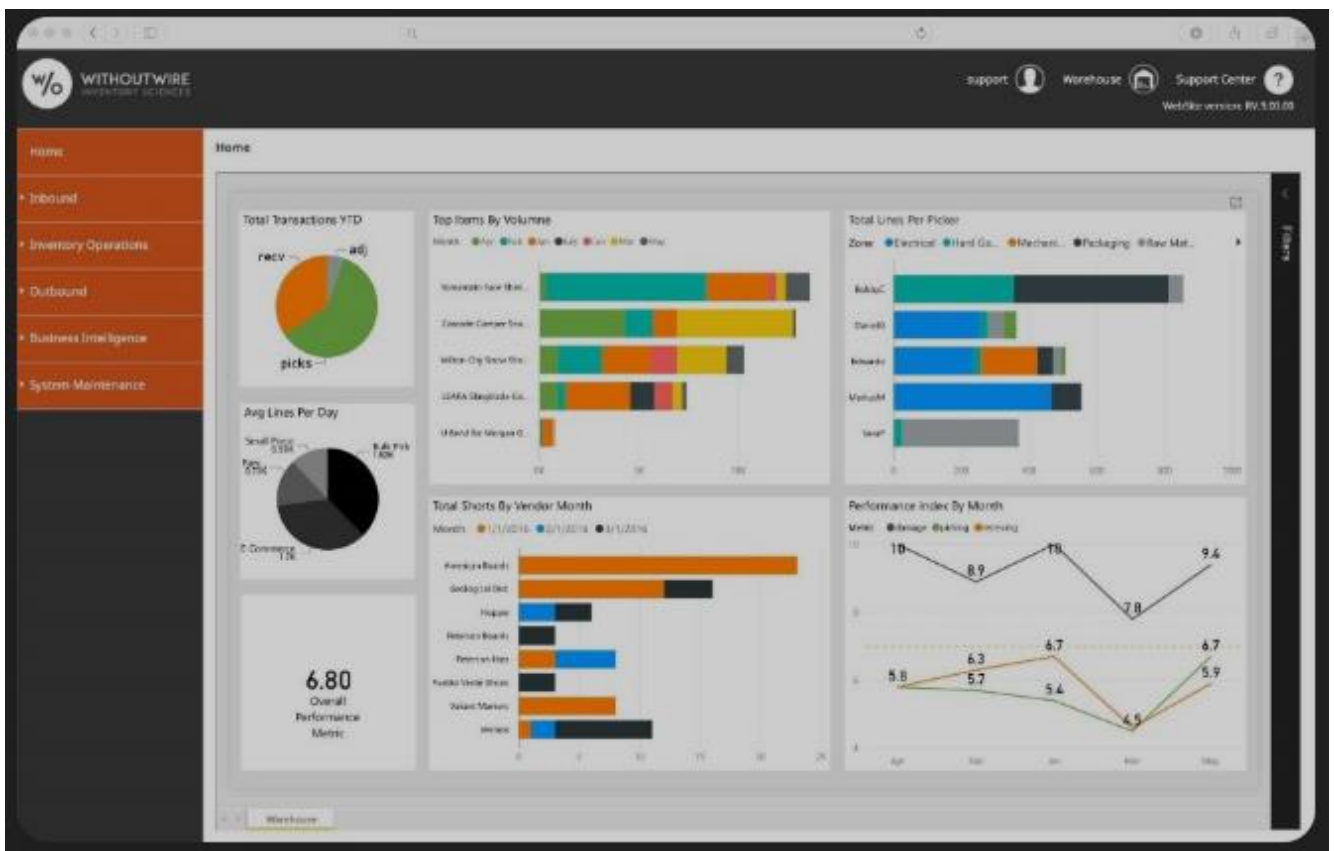


Рисунок 1.3 – Інтерфейс мобільного застосунку WithoutWire

Зм.	Лист	№ докум.	Підпис	Дата

Розробник: WithoutWire Inventory Sciences, Inc. (США).

Технології: Xamarin, .NET, Microsoft Azure.

Функціональність:

- 1) сканування штрихкодів;
- 2) контроль партій;
- 3) можливість аудиту якості продукції.

Застосування: мобільний аудит складів, виробництво, харчова промисловість.

Переваги: мобільність і швидке реагування, а також підтримка роботи офлайн.

Недоліки: обмежений інтерфейс для налаштувань і можливість вимагання інтеграції з іншими системами.

1.2.4 Рішення SAP ERP QM

SAP ERP QM (Quality Management) - це модуль SAP ERP, який забезпечує управління процесами контролю якості на підприємстві, від планування та випробувань до сертифікації та звітності. Він допомагає запобігти дефектам, забезпечити безперервне вдосконалення процесів та встановити сталі програми контролю якості [8]. Зображення інтерфейсу сервісу зображено нижче(рис. 1.4).

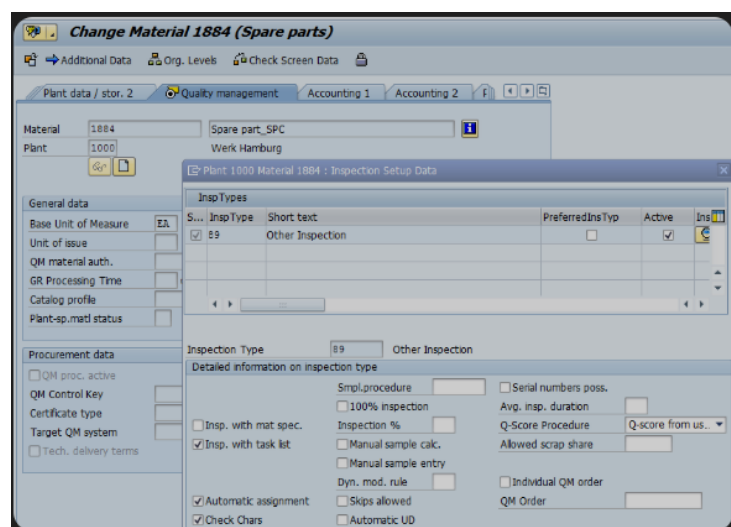


Рисунок 1.4 – Інтерфейс модулю SAP ERP QM

									Арк.
									15
Зм.	Лист	№ докум.	Підпис	Дата					

Розробник: SAP SE (Німеччина).

Технології: ABAP, SAP HANA, інтеграція через API.

Функціональність:

- 1) управління якістю продукції;
- 2) планування перевірок;
- 3) документування відхилень.

Застосування: великі виробничі підприємства.

Переваги глибока інтеграція в бізнес-процеси і підтримка міжнародних стандартів.

Недоліки: висока складність налаштування, а також висока вартість впровадження.

1.2.5 Порівняння сервісів

Тепер для наочнішого аналізу можна порівняти ці сервіси між собою за різними критеріями, такими як технології, використані при роботі та інші, найкраще це зробити у вигляді таблиці(табл. 1.1).

Таблиця 1.1 – Порівняння сервісів

Назва	Розробник	Технології	Основні функції	Переваги	Недоліки
Odoo Inventory	Odoo S.A. (Бельгія)	Python, PostgreSQL	Інвентаризація, контроль якості, ERP-інтеграція	Гнучкість, інтеграція з ERP	Вартість, потреба навчання
WithoutWire	WithoutWire Sciences (США)	Xamarin, .NET, Azure	Мобільний аудит, штрихкоди	Мобільність, офлайн-режим	Обмежена гнучкість

Назва	Розробник	Технології	Основні функції	Переваги	Недоліки
SAP ERP QM	SAP SE (Німеччина)	ABAP, SAP HANA, API	Планування перевірок, управління відхиленнями	Інтеграція в бізнес, підтримка стандартів	Складність, вартість впровадження

З проведеного порівняння можна зробити висновок, що кожен із розглянутих сервісів має власну специфіку, переваги та недоліки, що робить їх більш або менш придатними для конкретних сценаріїв використання.

Висновок до розділу 1

У даному розділі були більш детально розглянуті робочі процеси контролю якості харчових продуктів у харчовій промисловості, як у ручному, так і в автоматизованому режимі. Після аналізу традиційних підходів до перевірки якості продуктів харчування, було встановлено, що вони вимагають значних людських, матеріальних та часових ресурсів, часто є недостатньо точними і суб'єктивними. Це підтверджується статистичними даними, які свідчать про високий відсоток браку та додаткових витрат на виробництво.

Було проведено класифікацію існуючих рішень та розглянуто ряд сучасних застосунків, що забезпечують автоматизацію контролю якості. Для кожного з них визначено функціональність, використані технології та можливості їх застосування у харчовій промисловості.

В результаті огляду наявних аналогів виявлені основні проблеми та недоліки, які мають бути вирішені у даній розробці: відсутність універсальності, висока вартість рішень, складність інтеграції з робототехнічними системами та обмежена гнучкість налаштувань під потреби конкретного підприємства.

2 ФОРМУВАННЯ ВИМОГ ТА ВИБІР ТЕХНОЛОГІЙ І СЕРВІСІВ ДЛЯ РОЗРОБКИ АВТОМАТИЗОВАНОЇ СИСТЕМИ

У цьому розділі буде детально розглянуто функціональні та нефункціональні вимоги до розроблюваної системи автоматизованого контролю якості харчових продуктів, а також обґрунтовано вибір технологій та інструментів для системи.

2.1 Вибір технологій та сервісів

Для успішної реалізації автоматизованої системи контролю якості харчових продуктів було обрано сучасні та перевірені технології, які відповідають функціональним і нефункціональним вимогам, визначеним у попередніх підпунктах. У цьому підпункті розглянемо кожну технологію окремо та обґрунтуємо її використання у системі.

2.1.1 Мова програмування Python

Python — високорівнева мова програмування з відкритим вихідним кодом, яка підтримує об'єктно-орієнтоване програмування [9]. Завдяки своїй простоті, гнучкості та великій кількості бібліотек, Python є однією з найпопулярніших мов для розробки систем штучного інтелекту та комп'ютерного зору.

Python використовується для розробки бекенду системи, реалізації нейронних мереж, обробки зображень та взаємодії з базами даних. Також саме на ньому написаний фронтенд системи. Основні переваги: простота синтаксису, що пришвидшує розробку та наявність великої кількості бібліотек для обробки зображень, машинного навчання та інтеграції з сервісами Telegram Bot API.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		18

2.1.2 Фреймворк комп'ютерного зору YOLOv8

YOLOv8 (You Only Look Once) — сучасний фреймворк для розпізнавання об'єктів у режимі реального часу, розроблений компанією Ultralytics [10]. YOLOv8 відзначається високою швидкістю роботи, точністю та можливістю донавчання під конкретні завдання.

YOLOv8 використовується для автоматичного виявлення дефектів на фотографіях продуктів харчування. Це дозволяє швидко аналізувати зображення, виявляти плями, механічні пошкодження та інші дефекти, що впливають на якість продукції. Також YOLOv8 підтримує роботу з кастомним датасетом.

Схематичне зображення архітектури YOLOv8 зображено нижче(рис. 2.1)

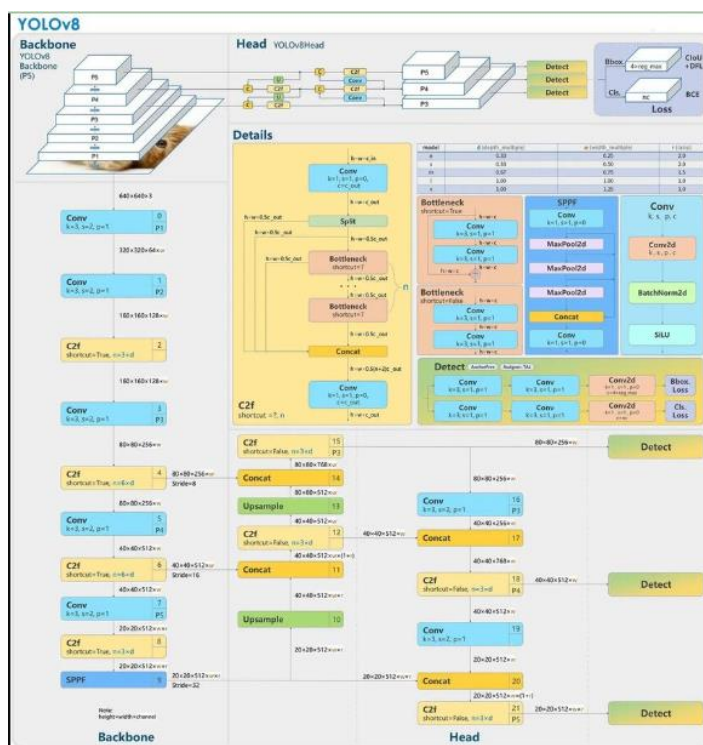


Рисунок 2.1 – Архітектура YOLOv8 [11]

2.1.3 Бібліотека python-telegram-bot(Telegram Bot API)

Telegram Bot API — набір інструментів для розробки ботів у месенджері Telegram. Бібліотека python-telegram-bot є популярною обгорткою для інтеграції

Python-програм із Telegram Bot API, що дозволяє швидко створювати чатботи та додавати інтерактивність [12].

Бот забезпечує зручний інтерфейс для користувачів, дозволяючи надсилати фотографії, отримувати результати аналізу, переглядати історію перевірок та взаємодіяти з системою. Це мінімізує потребу у веб-інтерфейсі та спрощує використання для кінцевого користувача.

2.1.4 СУБД SQLite

SQLite — реляційна база даних, яка працює як вбудована в додаток. Не потребує окремого сервера для зберігання даних та відзначається високою продуктивністю для невеликих та середніх проєктів [13].

SQLite використовується для зберігання результатів аналізу зображень, історії перевірок та даних користувачів. Це рішення дозволяє знизити складність розгортання та пришвидшити прототипування.

2.1.5 Бібліотека ORM SQLAlchemy

SQLAlchemy — популярна бібліотека Python для реалізації Object-Relational Mapping (ORM). Дозволяє працювати з базою даних як з об'єктами Python, спрощуючи управління транзакціями та бізнес-логікою [14].

SQLAlchemy використовується для взаємодії з базою даних SQLite, полегшуючи написання запитів і управління даними. Це прискорює розробку та тестування, а також дозволяє легко масштабувати проєкт за потреби.

2.1.6 Сервіс Roboflow

Roboflow — популярна платформа для підготовки, анотування та керування датасетами комп'ютерного зору. Вона дозволяє швидко анотувати зображення,

експортувати датасети у потрібному форматі та інтегрувати їх із YOLOv8(рис. 2.2) [15].

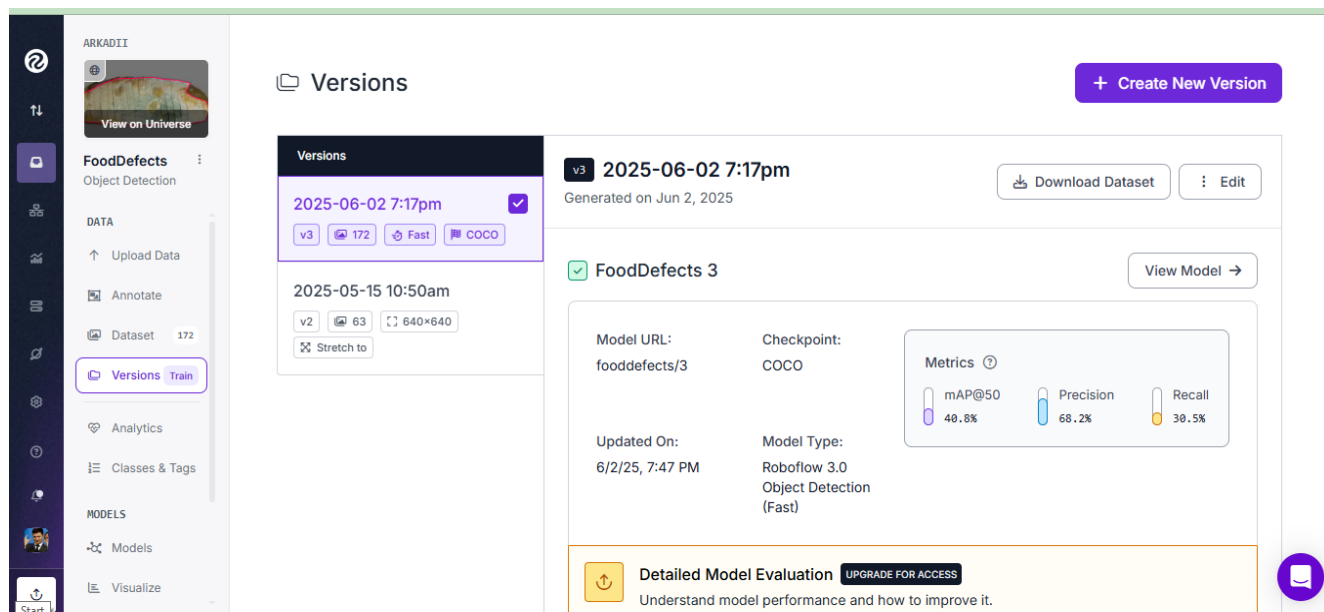


Рисунок 2.2 – Інтерфейс сервісу Roboflow

Roboflow використовується для створення та управління власним датасетом дефектів харчових продуктів. Це значно спрощує процес навчання моделі YOLOv8 та дозволяє швидко адаптувати систему до нових типів продуктів.

2.1.7 Середовище розробки PyCharm

PyCharm — інтегроване середовище розробки (IDE) для Python від JetBrains. Воно забезпечує потужні інструменти для відлагодження, тестування та керування проектами Python, підтримує інтеграцію з Git та іншими системами керування версіями(рис. 2.3) [16].

PyCharm використовується для розробки коду, управління проектом, створення віртуальних середовищ та забезпечує зручний графічний інтерфейс для налагодження системи. Це полегшує розробку та супровід проекту на всіх етапах.

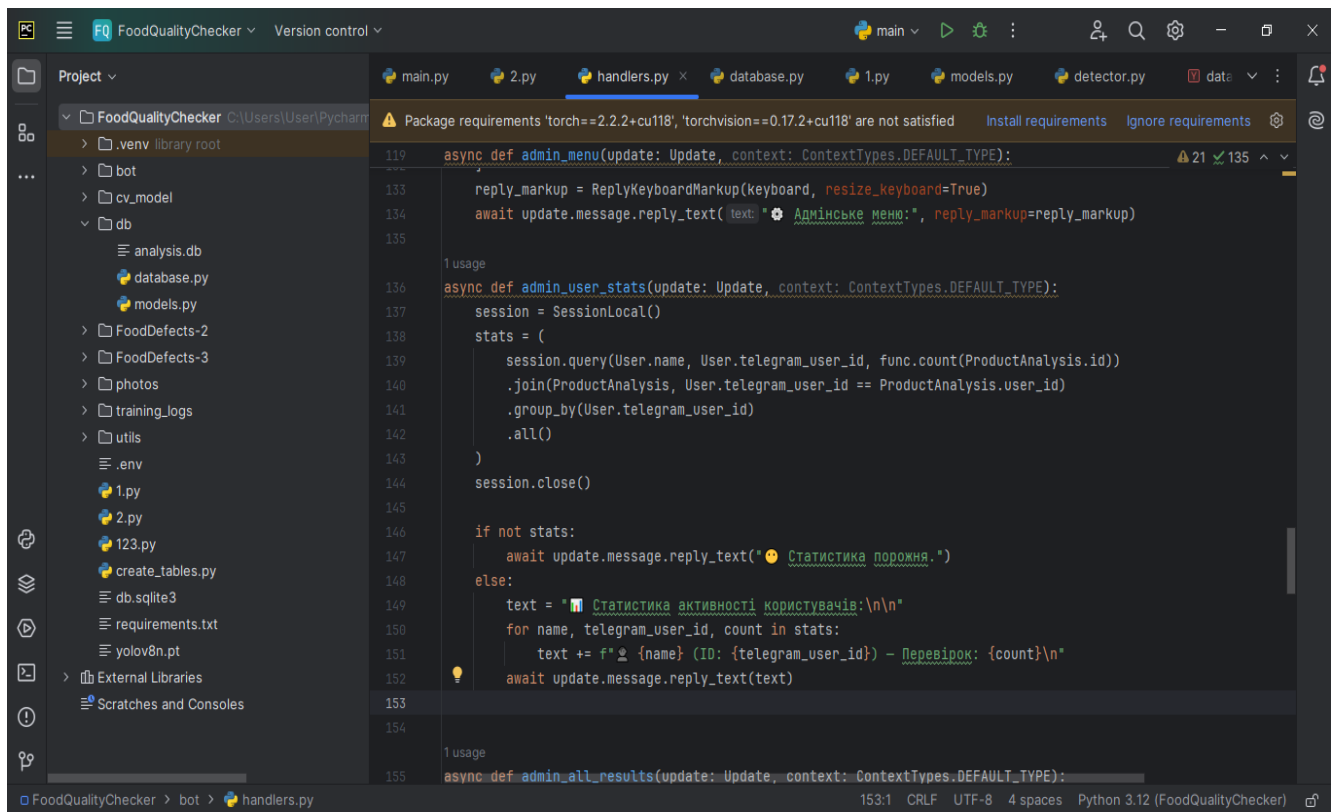


Рисунок 2.3 – Інтерфейс IDE PyCharm

2.1.8 Інші бібліотеки та сервіси

Для роботи над проектом застосовувалися й інші технології, такі як:

- 1) Pillow – бібліотека для обробки зображень у Python (зміна розміру, форматування, анотації);
- 2) Ultralytics Hub – для спрощеної інтеграції та керування моделями YOLOv8, підтримки кастомних датасетів і управління хмарними інстанціями.

2.2 Функціональні вимоги

Функціональні вимоги — це опис того, що має робити система, щоб задовольнити потреби користувачів та замовників. Ці вимоги відображають функціонал, необхідний для вирішення завдань, покладених на систему, і визначають її архітектуру та вибір технологій. У контексті розробки автоматизованої системи конт-

						Арк.
						22
Зм.	Лист	№ докум.	Підпис	Дата		

ролю якості харчових продуктів на основі комп'ютерного зору функціональні вимоги є ключовими для досягнення основної мети — забезпечення швидкого, об'єктивного та надійного контролю якості продукції. Нижче наведено перелік та пояснення кожної функціональної вимоги:

1) автоматизований аналіз якості харчових продуктів. Система повинна автоматично аналізувати якість харчових продуктів за допомогою комп'ютерного зору та алгоритмів машинного навчання. Це дозволяє мінімізувати вплив людського фактору (суб'єктивності), забезпечити стабільність та повторюваність результатів, а також значно знизити час, необхідний для перевірки кожного продукту. Така функція особливо важлива для підприємств, де великий обсяг продукції потребує оперативного контролю;

2) надсилання фотографій продуктів користувачами через Telegram-бота. Для зручності використання система повинна дозволяти користувачам надсилати фотографії продуктів безпосередньо через інтерфейс Telegram-бота. Це спрощує процес збору даних, під час якого користувач робить фото продукту за допомогою смартфона, потім фото автоматично передається системі для аналізу, а вже після цього користувач отримує результат прямо у чаті бота. Таким чином, мінімізуються вимоги до встановлення додаткових програм та налаштування обладнання;

3) автоматичне визначення дефектів. Система повинна вміти автоматично виявляти дефекти на фотографії продукту, це можуть бути механічні пошкодження (тріщини, вм'ятини), плями або зміна кольору, пліснява та інші ознаки псування. Ця функція має базуватися на алгоритмах глибокого навчання та передбачати високу точність, щоб уникати помилкових спрацювань;

4) формування текстових висновків. На основі виявлених дефектів система має формувати короткі та зрозумілі текстові висновки для користувача. Наприклад: «Продукт має допустимий стан для споживання» чи «Виявлено значні дефекти — продукт потребує утилізації». Це дозволяє користувачу одразу прийняти рішення про подальше використання чи вибракування продукту;

5) відображення обробленого зображення. Для кращого сприйняття результатів користувачеві необхідно відображати фото з виявленими дефектами у вигляді

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		23

bounding boxes (рамки навколо дефектів), підсвічування проблемних зон, а також додавання підписів до кожного дефекту. Це дозволить користувачу візуально оцінити результат і переконатися у правильності аналізу;

6) формування рекомендацій. На основі результатів аналізу система має формувати персоналізовані рекомендації, такі як придатність до споживання, потреба у сортуванні або повторному контролі, необхідність утилізації. Це допоможе підприємствам автоматизувати процес прийняття рішень;

7) виконання адміністраторських функцій. Окремим користувачам, що виконують адміністраторські функції, має бути надано додатковий функціонал, а саме перегляд всіх результатів перевірок користувачів і моніторинг завантаженості системи, статистики використання. Це допоможе краще контролювати діяльність системи та швидко виправляти знайдені прогалини;

8) збереження результатів у базі даних. Всі результати аналізів повинні зберігатися у базі даних із такими атрибутами як дата та час перевірки, фото оригінальне та оброблене, висновки, виявлені дефекти, рекомендації. Це дозволяє користувачам переглядати історію та здійснювати контроль якості продукції з плином часу;

9) перегляд історії перевірок. Користувачі повинні мати змогу переглядати історію своїх перевірок через Telegram-бота. Це передбачає доступ до останніх 5-10 перевірок, можливість перегляду фото та результатів та фільтрацію за датою. Це важливо для аналізу динаміки якості продукції;

10) розділ FAQ. Для покращення користувацького досвіду у боті має бути передбачений розділ «Часті запитання», де дається відповідь на те як користуватися ботом, як правильно фотографувати продукт, як зберігаються дані та чи безпечний бот для конфіденційності. Це дозволить користувачам швидко орієнтуватися у функціоналі;

11) базова інструкція користування. Бот має включати коротку інструкцію користування системою для нових користувачів, у ній має бути інформація, про те, де знайти кнопку для відправки фото, як отримати результат, як переглядати історію. Це значно підвищує зручність використання системи.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		24

2.3 Нефункціональні вимоги

Нефункціональні вимоги визначають характеристики системи, що впливають на її експлуатаційні якості та користувацький досвід. Вони не описують конкретні функції, а визначають «як» система виконує свої функції. Це стосується зручності використання, безпеки, швидкодії, масштабованості, надійності та інших важливих аспектів роботи системи. Нижче наведено основні нефункціональні вимоги до розроблюваної системи автоматизованого контролю якості харчових продуктів на основі комп'ютерного зору.

Інтерфейс Telegram-бота має бути простим, зрозумілим і не перевантаженим зайвими елементами. Це забезпечить користувачам комфортну взаємодію з системою, дозволяючи їм швидко отримати необхідний результат. Інтуїтивно зрозумілий інтерфейс підвищує ефективність використання системи та зменшує потребу у навчанні користувачів.

За даними Nielsen Norman Group, простота та мінімалізм інтерфейсу зменшують кількість помилок користувачів на 40% порівняно з перевантаженими інтерфейсами [17]. Це критично важливо для підвищення продуктивності та зручності використання. Саме простий і зрозумілий інтерфейс має стати тим, що втримає користувача на цьому продукті.

Згідно зі звітом Statista, користувачі очікують отримати результат у межах 3-5 секунд після запиту. Це пояснюється тим, що понад 65% користувачів мобільних додатків покидають додаток, якщо він відповідає повільно [18]. Це підтверджує необхідність забезпечення обробки фотографій у межах 5 секунд.

Система має функціонувати автономно, без постійної участі людини-оператора. Це дозволить знизити витрати на обслуговування та забезпечити більш стабільну роботу системи. Автоматизація передбачає, що користувач може взаємодіяти з ботом без залучення фахівця або адміністратора для кожної перевірки.

Система повинна забезпечувати безпечне зберігання даних користувачів (фото, результати перевірок, історію). Передача даних має відбуватися за захищеним протоколом HTTPS, а база даних має бути захищена від несанкціонованого

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		25

доступу. Також необхідно забезпечити дотримання загальних стандартів конфіденційності (GDPR).

У сучасних умовах безпеки даних приділяється особлива увага через зростання кіберзагроз. Наприклад, дослідження IBM показують, що витрати на ліквідацію наслідків витоку даних сягають в середньому 4,35 млн доларів США для однієї організації [19]. Саме тому захист прав споживачів має стати важливим завданням при розробці цієї системи.

Система повинна бути побудована таким чином, щоб у майбутньому можна було легко збільшувати кількість користувачів, підключати нові моделі комп'ютерного зору, додавати нові сервіси та функціонал. Не менш необхідною є можливість підготовки до переходу з засобу оцінювання якості харчових продуктів для звичайного користувача до промислового, який вже буде використовувати роботів для цього завдання.

Масштабованість дозволяє системі адаптуватися до зростання бізнесу або до підвищених навантажень під час пікових періодів (наприклад, сезонні перевірки продуктів). За даними Gartner, зростання обсягу даних вимагає масштабованих рішень у 78% випадків для систем, що працюють саме із великими обсягами зображень [20].

Висновок до розділу 2

У даному розділі було проведено детальний аналіз функціональних та нефункціональних вимог до системи автоматизованого контролю якості харчових продуктів, а також вибору технологій, які необхідні для розробки.

Спочатку було визначено ключові функціональні вимоги, такі як автоматизація процесу перевірки якості продуктів, інтеграція з Telegram-ботом для зручної взаємодії користувача, можливість перегляду історії перевірок та надання рекомендацій.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		26

Також було сформульовано нефункціональні вимоги до системи, ними стали: висока швидкість обробки, інтуїтивний інтерфейс користувача, безпека даних, масштабованість та автономність.

Щодо вибору технологій та сервісів, які дозволяють реалізувати всі функціональні та нефункціональні вимоги. Були обґрунтовані переваги використання Python як основної мови програмування, фреймворку YOLOv8 для реалізації модуля комп'ютерного зору, бібліотеки python-telegram-bot для інтеграції з Telegram, а також бази даних SQLite у зв'язці з ORM SQLAlchemy. Додатково до цього було розглянуто використання сервісу Roboflow для створення та керування датасетами та середовища розробки PyCharm для зручного управління кодом і налаштування проекту.

Отже, сформульовані вимоги та обґрунтований вибір технологій створюють міцну основу для подальшої розробки автоматизованої системи, що забезпечить високу якість перевірки харчових продуктів, ефективність та надійність роботи програми.

3 РОЗРОБКА ТА АНАЛІЗ ПРОГРАМНОГО ПРОДУКТУ

3.1 Архітектура автоматизованої системи

3.1.1 Загальне визначення архітектури систем

Архітектура автоматизованої системи визначає загальний спосіб організації та взаємодії її основних компонентів. У сфері автоматизованих систем візуального контролю якості харчових продуктів архітектура виконує важливу роль у забезпеченні ефективності, масштабованості та гнучкості системи. Вона визначає, як модулі програмного забезпечення, апаратні засоби та користувацький інтерфейс взаємодіють між собою для досягнення поставлених цілей.

У сучасній практиці архітектура таких систем включає в себе не лише програмну логіку, а й питання інтеграції з базами даних, хмарними сервісами, засобами глибинного навчання та користувацькими інтерфейсами.

3.1.2 Поширені архітектурні підходи у даній сфері

У розробці автоматизованих систем контролю якості харчових продуктів на основі комп'ютерного зору найбільш поширеними та вживаними є такі архітектурні підходи:

1) монолітна архітектура. Вона передбачає об'єднання всіх компонентів системи в один програмний блок. У такій архітектурі база даних, сервер додатку, інтерфейс користувача та бізнес-логіка функціонують як єдиний модуль (рис. 3.1) [21].

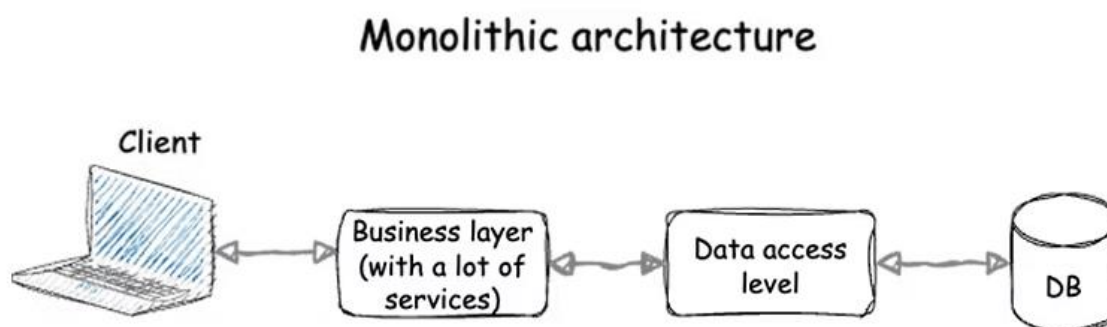


Рисунок 3.1 – Схема монолітної архітектури

Зм.	Лист	№ докум.	Підпис	Дата

Монолітна архітектура має такі переваги, як:

- простота реалізації та підтримки для невеликих проєктів;
- відсутність складної конфігурації для взаємодії між компонентами.

Натомість вона має і свої недоліки:

- обмежена масштабованість;
- складність оновлень (зміна однієї частини впливає на всю систему);
- важче забезпечити надійність (помилка в одному модулі може вплинути на всю систему);

2) клієнт-серверна архітектура. Ця архітектура базується на розділенні системи на клієнтську і серверну частини. Клієнт надсилає запити до сервера, а сервер обробляє їх і повертає відповіді (рис. 3.2).

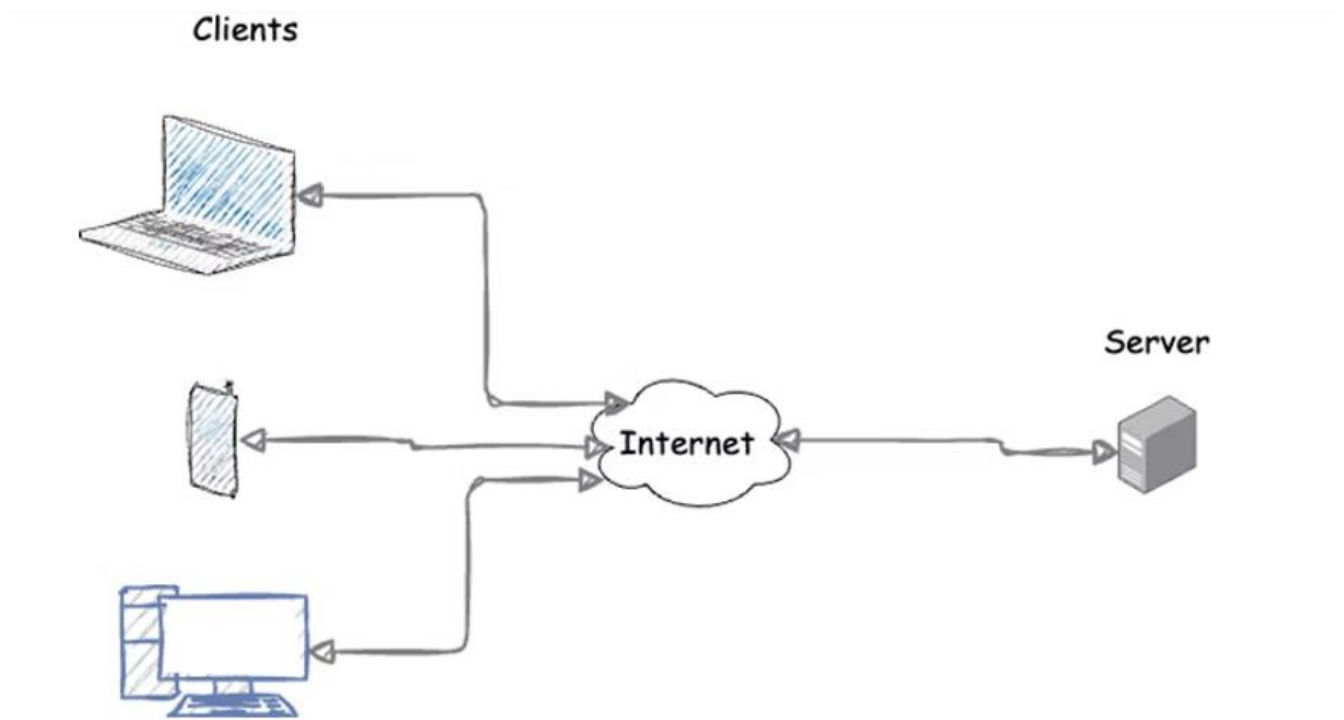


Рисунок 3.2 – Схема клієнт-серверної архітектури

Клієнт-серверна архітектура має такі переваги:

- гнучкість та масштабованість;
- можливість використання різних клієнтів (мобільний додаток, веб-інтерфейс);

Зм.	Лист	№ докум.	Підпис	Дата

- централізоване управління даними та безпекою.

Але вона має і свої недоліки, такі як:

- потреба у надійному сервері та мережевому з'єднанні;
- ускладнення при високому навантаженні (необхідне балансування);

3) мікросервісна архітектура. Ця архітектура розділяє систему на окремі незалежні сервіси (наприклад, сервіс виявлення дефектів, сервіс управління користувачами, сервіс бази даних). Кожен сервіс виконує конкретну функцію та взаємодіє з іншими через API(рис. 3.3).

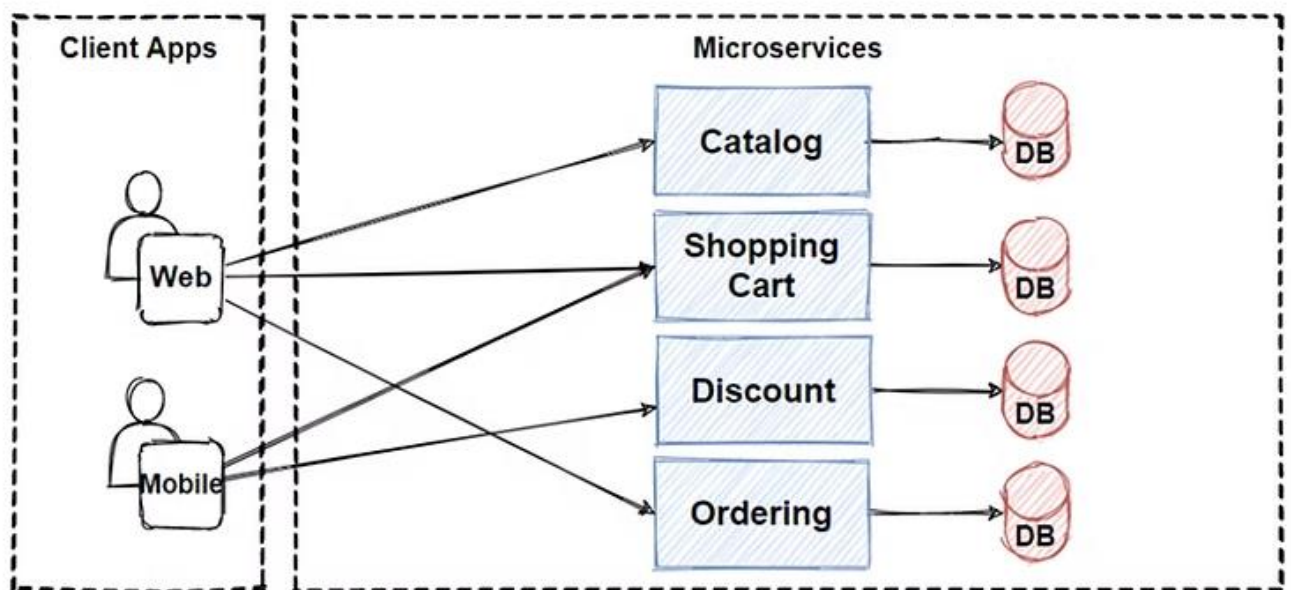


Рисунок 3.3 – Схема мікросервісної архітектури

Перевагами цієї архітектури можна назвати:

- масштабованість кожного окремого модуля;
- гнучкість у розробці та оновленні (можна оновлювати окремі сервіси без зупинки всієї системи);
- краща надійність (відмова одного сервісу не обов'язково впливає на інші).

Серед недоліків можна виділити таке:

- ускладнена розробка та підтримка (потрібна координація між сервісами);
- необхідність організації безпечної взаємодії між сервісами;
- потреба в додатковій інфраструктурі (Docker, Kubernetes);

Зм.	Лист	№ докум.	Підпис	Дата

SOA активно застосовується у великих розподілених системах, де потрібна інтеграція різних платформ та технологій. Сервіси можуть взаємодіяти між собою через протоколи HTTP, REST API або за допомогою брокерів повідомлень (рис. 3.5).

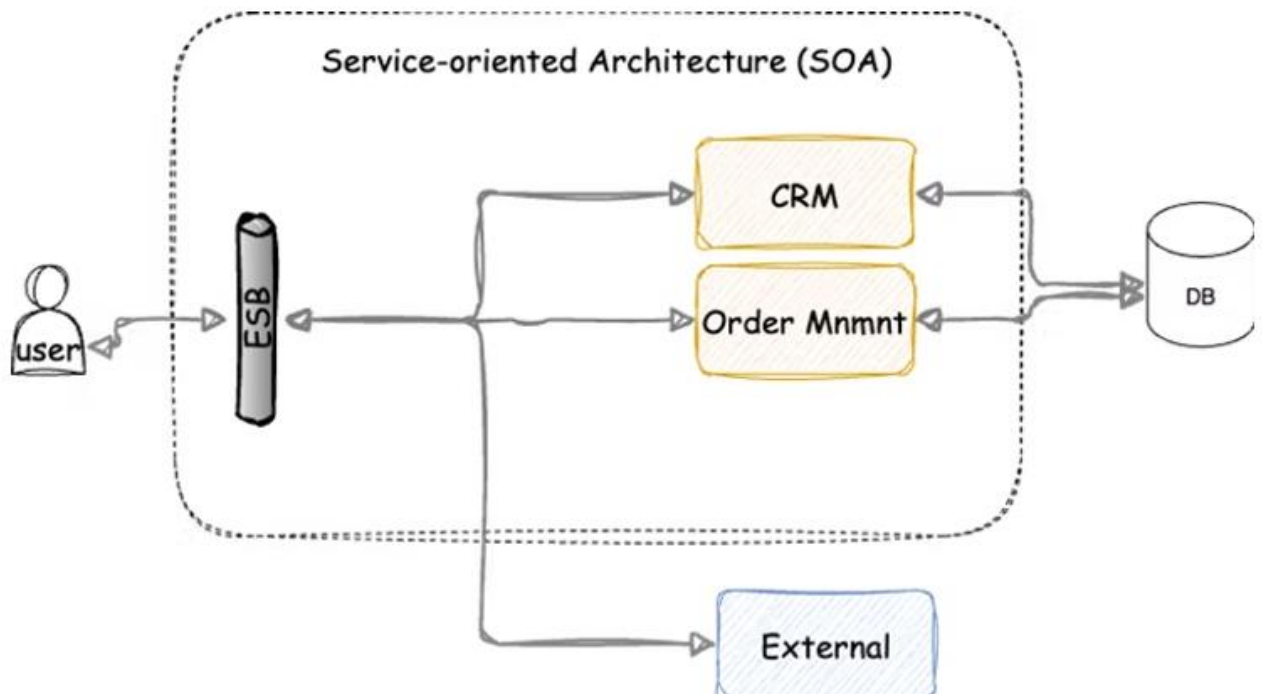


Рисунок 3.5 – Схема сервісно-орієнтованої архітектури

Переваги цієї архітектури:

- чіткий розподіл обов'язків між сервісами;
- масштабованість та можливість повторного використання сервісів;
- незалежна розробка та тестування кожного сервісу.

Недоліки сервісно-орієнтованої архітектури:

- може ускладнювати моніторинг та налагодження через розподілений характер системи;
- потребує стандартизації інтерфейсів та протоколів взаємодії.

3.1.3 Основні компоненти архітектури системи

З урахуванням завдань та обраного підходу, дана система контролю якості харчових продуктів базується на клієнт-серверній архітектурі, що поєднує простоту реалізації та гнучкість масштабування. Основні компоненти:

1) клієнтський інтерфейс (Telegram-бот), який забезпечує зручний доступ користувачів до системи через популярний месенджер. Дозволяє надсилати фотографії продуктів для аналізу, переглядати результати, переглядати історію перевірок та отримувати відповіді на запитання.;

2) сервер додатка, який містить бізнес-логіку системи, включаючи обробку зображень, виклик алгоритмів комп'ютерного зору, формування результатів, управління доступом, запис історії перевірок та інше;

3) база даних, яка призначена для зберігання інформації про користувачів, результати перевірок, шляхи до зображень, налаштування та інші дані. Реалізована на базі SQLite через простоту інтеграції з Python та Telegram-ботом;

4) модуль комп'ютерного зору, що виконує основну задачу виявлення дефектів на зображеннях харчових продуктів, використовуючи алгоритм YOLOv8 та інші бібліотеки (OpenCV, Pillow).

Діаграма компонентів системи наведена на кресленнику ІК11.310БАК.006 Д1.

3.1.4 Взаємодія між компонентами архітектури

Блок-схема алгоритму системи наведена на кресленнику ІК11.310БАК.006 Д2.

Взаємодія між компонентами відбувається наступним чином:

1) користувач через Telegram-бота надсилає фотографію продукту;
2) telegram-бот приймає фотографію та передає її разом з ідентифікатором користувача на Сервер додатка;

3) сервер додатка отримує фото, викликає Модуль YOLOv8 для виконання аналізу зображення;

4) модуль YOLOv8 проводить аналіз фотографії, виявляє дефекти та формує звіт із результатами, після чого повертає його на сервер;

5) сервер додатка записує отримані результати у базу даних, що дає можливість зберігати історію перевірок користувача та аналітичні дані;

б) сервер додатка передає результати перевірки назад до Telegram-бота, а той, у свою чергу, відображає їх користувачу.

Ця архітектура дозволяє забезпечити розподіл завдань між окремими модулями та досягти високої швидкості обробки запитів завдяки паралельній взаємодії між компонентами.

3.2 Визначення ролей користувачів

У сучасних автоматизованих системах контролю якості харчових продуктів важливу роль відіграє чітке визначення ролей користувачів. Це дозволяє не тільки регламентувати права доступу, а й забезпечити ефективне управління інформаційними потоками, розмежування обов'язків та підвищення безпеки даних.

В рамках проєкту передбачається впровадження двох основних ролей користувачів: Звичайний користувач та Адміністратор. Кожна роль має свій набір функцій та рівень доступу до системи.

Звичайний користувач - це основна роль, призначена для працівників підприємства, інспекторів або операторів, які безпосередньо взаємодіють із системою. Основна мета звичайного користувача — завантажувати фотографії продуктів для перевірки якості та отримувати результати аналізу.

Адміністратор – це користувач із розширеними правами доступу. Роль адміністратора призначена для керівників підрозділів, інженерів з якості, а також ІТ-фахівців, які відповідають за налаштування та обслуговування системи.

Діаграма прецедентів наведена на кресленику ІК11.310БАК.006 ДЗ.

Діаграма прецедентів показує, перелік яких можливостей має кожна роль.

Можливості користувача:

1) завантаження фотографій продуктів – користувач може завантажувати фотографії для автоматизованого аналізу дефектів (пліснява, механічні пошкодження, гниль);

					ІК11.310БАК.006 ПЗ	Арк.
						34
Зм.	Лист	№ докум.	Підпис	Дата		

2) перегляд результатів перевірки – користувач отримує звіт про стан продукту та рекомендації щодо подальших дій (наприклад, «Придатний до споживання» або «Потребує утилізації»);

3) перегляд історії перевірок – користувач може ознайомитися з результатами попередніх перевірок продуктів, щоб аналізувати динаміку якості;

4) отримання відповідей на поширені запитання – система надає користувачу пояснення щодо роботи сервісу та можливих дій;

5) перегляд інструкцій – користувач має доступ до текстових інструкцій щодо роботи з ботом.

Можливості адміністратора:

1) отримання всіх результатів перевірок – адміністратор має доступ до всіх результатів перевірок, завантажених будь-якими користувачами, включаючи фотографії та звіти. Це дозволяє здійснювати загальний контроль якості продукції;

2) аналіз активності користувачів – адміністратор може отримувати статистику використання системи, включаючи кількість завантажених фото, кількість виконаних перевірок, періодичність використання системи тощо. Це важливо для моніторингу ефективності роботи працівників та підрозділів.

3.3 Розробка клієнтської частини

Програмний код клієнтської частини застосунку розміщується у вигляді окремих структурних елементів і зберігається у директоріях, кожна з яких відповідає за окрему функціональність системи. Така організація дозволяє легко масштабувати та підтримувати проєкт у майбутньому. Основна структура клієнтської частини представлена у директорії кореневого каталогу проєкту та складається з кількох основних директорій: “bot”, “db”, “cv_model”, “photos” та основного файлу запуску “main.py” (рис. 3.6).

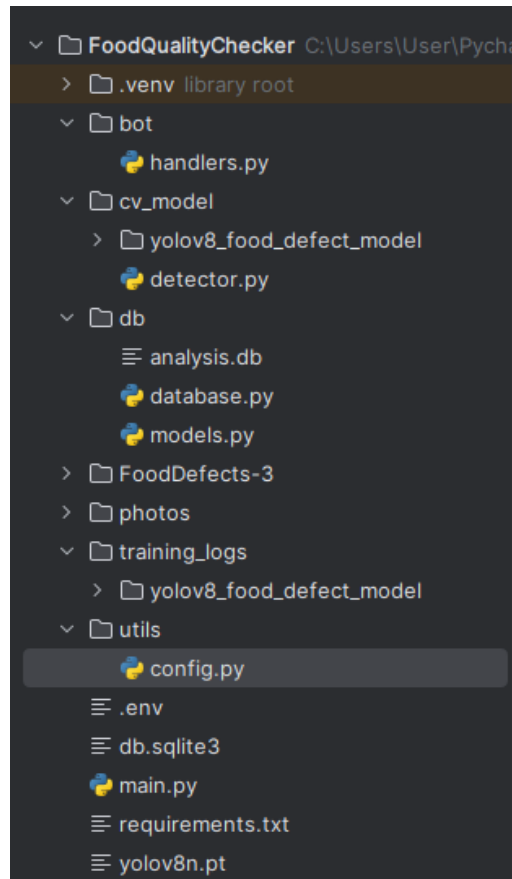


Рисунок 3.6 – Структура файлів проєкту

Компоненти розташовані у вигляді модулів, що спрощує управління кодом та робить його більш зрозумілим для розробника.

Діаграма класів наочно демонструє взаємозв'язок між основними модулями системи: модуль обробки команд користувача взаємодіє з моделями бази даних та з детектором комп'ютерного зору. Такий підхід дозволяє централізовано керувати логікою та забезпечує гнучкість при розширенні функціональності.

Діаграма класів наведена на кресленику ІК11.310БАК.006 Д4.

Архітектура клієнтської частини Telegram-бота для автоматизованої перевірки якості харчових продуктів включає кілька компонентів:

- 1) “bot” — обробка логіки взаємодії користувача через Telegram API;
- 2) “db” — взаємодія з базою даних через ORM SQLAlchemy;
- 3) “cv_model” — взаємодія з модулем комп'ютерного зору для виявлення дефектів продуктів;

4) “photos” — директорія для збереження отриманих та оброблених зображень;

5) `main.py` — головна точка запуску застосунку.

Розглянемо і опишемо детальніше кожен із компонентів клієнтської частини:

1) директорія bot, у ній розміщується файл handlers.py, для взаємодії користувача з Telegram-ботом. Він має такий функціонал:

- містить основні обробники команд, таких як “/start”, “/history”, “/faq”, а також обробку фото користувача. Також тут реалізовані команди адміністратора;

- забезпечує логіку роботи з Telegram API, обробку повідомлень, виклик модуля комп’ютерного зору для аналізу фото та збереження результатів у базі даних;

- реалізує функції відправки повідомлень, фото та результатів аналізу користувачеві;

- обробляє команди для історії перевірок користувача та FAQ;

2) директорія db, вона відповідає за роботу з базою даних, де зберігаються результати аналізу фото та інформація про користувачів. Розглянемо всі файли даної директорії:

- database.py, цей файл містить функції для підключення до бази даних SQLite через SQLAlchemy, реалізує створення таблиць у базі даних за допомогою методу “Base.metadata.create_all(engine)”, а також забезпечує підключення сесій для взаємодії з даними;

- models.py, він містить ORM-моделі таблиць “ProductAnalysis”, яка берігач інформацію про перевірені фото, результат, дефекти, рекомендації та дату перевірки та “User”, яка зберігач користувачів. Завдяки ORM можна швидко виконувати CRUD-операції без використання сирого SQL;

3) директорія cv_model, у ній знаходиться код для аналізу фотографій за допомогою модуля комп’ютерного зору YOLOv8 та навчена модель. У ній знаходяться:

- файл detector.py, який містить функцію “detect_defects”, яка приймає зображення, виконує його обробку та повертає результат аналізу, список виявлених дефектів, рекомендацію та анотоване зображення. У ньому використовується бібліотека Ultralytics YOLOv8 та Pillow для візуалізації;

- директорія yolov8_food_defect_model, що містить файли best.pt та args.yaml, які потрібні для детекції об’єктів;

4) директорія photos, вона містить усі отримані та оброблені фотографії, названі у форматі “{user_id}_{timestamp}.jpg”. Це дозволяє зберігати історію перевірок для кожного користувача та зручно їх відображати при повторних запитах;

5) файл main.py, головний файл застосунку, він імпортує та підключає всі обробники з “handlers.py”, ініціалізує Telegram Application для запуску бота, налаштовує сесії для взаємодії з базою даних та запускає прослуховування нових повідомлень від користувачів у режимі асинхронної обробки;

б) файл requirements.txt, який містить перелік залежностей проєкту, а саме:

- “python-telegram-bot”,

- “sqlalchemy”,

- “ultralytics”,

- “pillow”,

- інші бібліотеки, необхідні для стабільної роботи бота.

3.4 Розробка бази даних

База даних є серцем будь-якої інформаційної системи, оскільки вона зберігає всі необхідні дані, забезпечує їхню цілісність, доступність та взаємодію між компонентами системи. Для автоматизованої системи контролю якості харчових продуктів база даних виконує критично важливу роль: у ній зберігаються результати аналізів, фотографії продуктів, анотації дефектів та історія взаємодії користувачів із системою.

3.4.1 Технології для роботи з базою даних

У процесі розробки бази даних для Telegram-бота було обрано реляційну базу даних SQLite у поєднанні з бібліотекою ORM (Object-Relational Mapping) SQLAlchemy. Такий вибір зумовлений кількома факторами:

1) SQLite – це легка, вбудована база даних, яка не потребує розгортання окремого сервера, що особливо корисно для Telegram-бота, який працює як автономний сервіс. SQLite має відкритий вихідний код, підтримує транзакції та забезпечує достатню продуктивність для невеликих і середніх проєктів. До переваг SQLite належать простота використання, швидкість налаштування та надійність при розробці MVP-рішень;

2) SQLAlchemy – це потужний ORM-фреймворк для Python, який дозволяє розробнику працювати з базою даних на рівні об'єктів замість сирих SQL-запитів. Він підтримує як декларативний підхід для опису моделей таблиць, так і більш низькорівневі SQL-запити, що робить його універсальним інструментом для будь-яких завдань. SQLAlchemy дозволяє легко виконувати CRUD-операції (Create, Read, Update, Delete) та управляти транзакціями у зрозумілій формі.

Завдяки поєднанню SQLite та SQLAlchemy вдалося створити гнучку архітектуру бази даних, яка легко масштабується та дозволяє інтегрувати нові функції в майбутньому.

3.4.2 Структура бази даних

Розроблена база даних відповідає основним функціональним вимогам системи. Основна увага приділялася збереженню результатів перевірок фотографій харчових продуктів та даних користувачів Telegram-бота.

Діаграма бази даних наведена нижче(рис. 3.7)

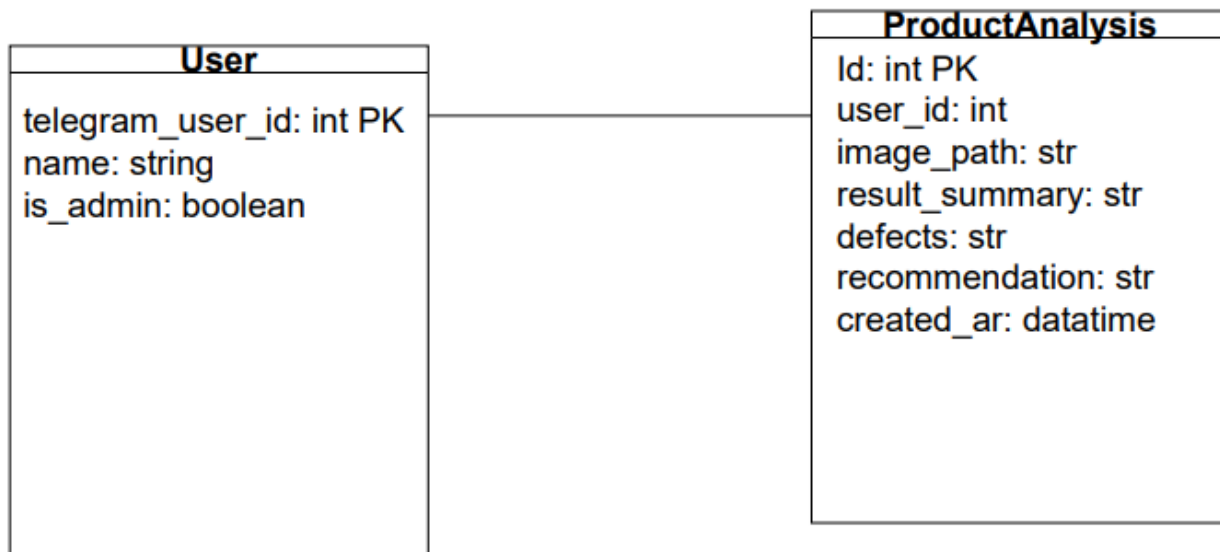


Рисунок 3.7 – Діаграма бази даних

Основні таблиці бази даних:

- 1) ProductAnalysis – таблиця, яка призначена для зберігання результатів перевірки продуктів;
- 2) User – таблиця для зберігання даних про користувачів Telegram-бота.

Розглянемо кожну таблицю детально, потрібно визначити атрибути та показати створені класи бази даних.

Таблиця “ProductAnalysis” призначена для збереження результатів аналізу фотографій продуктів харчування, оброблених за допомогою комп’ютерного зору. Вона містить інформацію про користувача, зображення, результати перевірки та рекомендації (табл. 3.1).

Таблиця 3.1 – Структура таблиці «Аналіз продукту»

Атрибут	Опис
id	Унікальний ідентифікатор запису
user_id	Telegram ID користувача, який надіслав фото
image_path	Шлях до збереженого зображення на диску
result_summary	Результат аналізу фото (короткий текстовий висновок)
defects	Перелік виявлених дефектів (якщо є)


```

11 usages
class User(Base):
    __tablename__ = "users"

    telegram_user_id = Column(Integer, primary_key=True, index=True)
    name = Column(String)
    is_admin = Column(Integer, default=0)

```

Рисунок 3.9 – Клас "User"

3.4.3 Підключення до бази даних та взаємодія з ORM

Для підключення до бази даних використовується модуль SQLAlchemy, який створює сесію взаємодії із СУБД. Нижче(рис. 3.10) наведено приклад коду з "db/database.py", де налаштовується підключення до SQLite.

```

DATABASE_URL = "sqlite:///db.sqlite3"

engine = create_engine(DATABASE_URL)
SessionLocal = sessionmaker(autocommit=False, autoflush=False, bind=engine)
Base = declarative_base()

```

Рисунок 3.10 – Підключення до SQLite

"create_engine" – створює підключення до бази даних SQLite;

"SessionLocal" – відповідає за створення сесій для виконання запитів;

"Base" – батьківський клас для всіх моделей.

Нижче(рис. 3.11) наведено код, який потрібно викликати для створення таблиць у базі даних.

```

Base.metadata.create_all(bind=engine)

```

Рисунок 3.11 – Створення таблиць

Цей метод створює всі таблиці, описані через ORM.

3.4.4 Взаємодія з ORM

Взаємодія з базою даних здійснюється через сесію SQLAlchemy. Запис у БД здійснюється способом, зображеним нижче(рис. 3.12)

```
session = SessionLocal()
analysis = ProductAnalysis(
    user_id=user_id,
    image_path=detected_path,
    result_summary=result_summary,
    defects=defects,
    recommendation=recommendation
)
session.add(analysis)
session.commit()
session.close()
```

Рисунок 3.12 – Запис у базу

Оновлення даних у БД відбувається способом, показаним на рисунку 3.13

```
record = session.query(User).filter_by(id=1).first()
record.is_admin = 1
session.commit()
```

Рисунок 3.13 – Оновлення даних

Отримання даних у БД реалізовано способом, показаним нижче(рис. 3.14)

```
session = SessionLocal()
records = session.query(ProductAnalysis).filter_by(user_id=user_id).all()
session.close()
```

Рисунок 3.14 – Отримання даних

Використання ORM значно спрощує роботу з даними та забезпечує безпечну обробку запитів.

Висновки до розділу 3

У даному розділі було детально розглянуто процес розробки та аналізу програмного продукту для автоматизації контролю якості харчових продуктів. Спочатку було досліджено та обґрунтовано вибір архітектури системи, визначено основні підходи до побудови архітектури та здійснено порівняння різних підходів. На основі цього було сформовано архітектуру системи, що враховує сучасні вимоги до продуктивності, масштабованості та зручності у використанні.

Далі було визначено ролі користувачів у системі, описано їх функціонал та права доступу, що дозволило чітко структурувати роботу додатку відповідно до потреб користувачів. Це забезпечило гнучкість та безпеку використання системи.

Особливу увагу приділено розробці клієнтської частини, де описано структуру каталогів та файлів додатку, їх призначення та взаємодію. Такий підхід дав можливість забезпечити логічну організацію коду, спростити процес тестування та подальшого супроводу системи.

Важливою частиною роботи стало проектування та розробка бази даних. Було описано вибір технологій та інструментів, структуру бази даних, а також взаємодію з ORM. Це дозволило досягти ефективного збереження та обробки даних, забезпечуючи швидкий доступ до інформації та захищеність даних користувачів.

					ІК11.310БАК.006 ПЗ	Арк.
						44
Зм.	Лист	№ докум.	Підпис	Дата		

4 ДЕМОНСТРАЦІЯ РОБОТИ СИСТЕМИ

Станом на зараз бот можна знайти в самому пошуку Телеграму, ввівши назву бота в пошукову стрічку(рис. 4.1). Бот працює на всіх платформах, які підтримує Телеграм, а це: мобільний застосунок, web-сторінка та додаток на ПК. Для зручності демонстрація роботи буде здійснюватися на додатку в комп'ютері.



Рисунок 4.1 – Пошук бота

У даній системі, згідно діаграми прецедентів, присутні 2 ролі, а саме: звичайний користувач та адміністратор. Кожна з цих ролей має свій власний набір функцій, тому для наглядності краще розібрати їх окремо.

4.1 Звичайний користувач

У цій підсекції розглянемо роботу системи для звичайного користувача, який має змогу перевіряти якість харчових продуктів та отримувати результати.

Після того, як бота було знайдено в пошуку користувач заходить в нього. Якщо користувач користується системою вперше, то переходить на передпочаткову сторінку. Будемо розглядати саме такий сценарій.

					ІК11.310БАК.006 ПЗ	Арк.
						45
Зм.	Лист	№ докум.	Підпис	Дата		

На цій сторінці присутні кілька елементів, які одразу бачить користувач, це:

- 1) назва («FoodQualityChecker») та позначка, яка показує, що цей сервіс є ботом (вгорі);

- 2) короткий опис того, для чого був створений цей бот (по центру);

- 3) кнопка («START»), яка власне починає чат з цим ботом та авторизує користувача.

Далі, при натисканні кнопки «START» автоматично відкривається чат та починається взаємодія з системою (рис. 4.2).



Рисунок 4.2 – Початкова сторінка бота

На екрані з'являється початкове повідомлення від бота з вітанням та з переліком основних функцій, якими може скористатися користувач. Це означає, що бот авторизував користувача та готовий працювати з ним.

Окрім вступного повідомлення внизу з'явилося базове меню, яке реалізовує основні функції. Воно виконано в простому мінімалістичному оформленні, що спрощує навігацію по системі.

Щоб згорнути меню необхідно натиснути на відповідну кнопку поруч з ним, щоб його повернути необхідно натиснути на кнопку меню.

Перед демонстрацією основної функції, а саме детекції неякісних об'єктів, перевіримо інші можливості системи, які не прив'язані до неї. Нічого не надсилаючи натиснемо на кнопку «Історія», що має викликати історію перевірок і переглянемо результат(рис. 4.3).

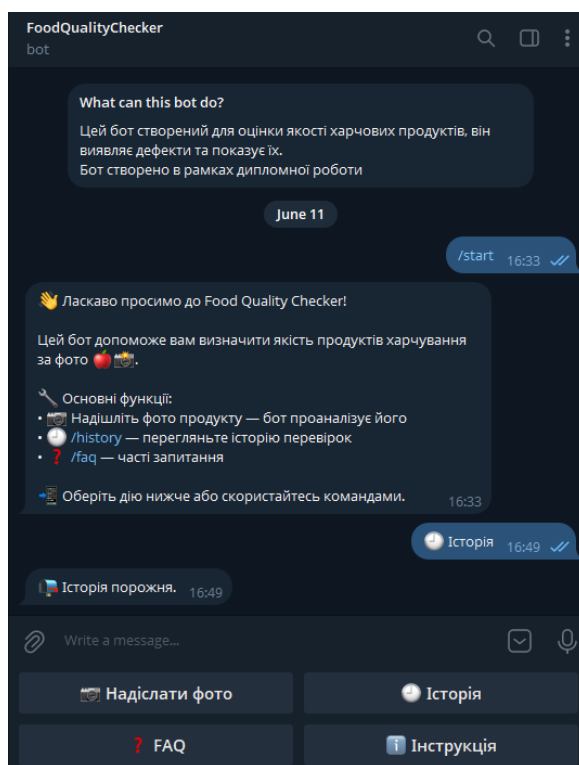


Рисунок 4.3 – Перший виклик історії перевірок

Як видно, бот повідомляє про те, що наразі історія пошуку порожня, що є вірним, адже користувач ще нічого не відправляв на перевірку.

Далі перевіримо іншу функцію, а саме отримання відповідей на найголовніші та найпопулярніші запитання, які можуть виникнути у користувача. Для цього потрібно натиснути кнопку «FAQ», після цього бот надішле відповідні повідомлення в чат, як і зображено на рисунку 4.4.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		47



Рисунок 4.4 –Відповіді на FAQ

На рисунку видно, що одразу бот вивів 5 запитань і дав на них відповідь. Це були:

- 1) як працює бот? Надішліть фото продукту, і бот визначить наявність дефектів за допомогою комп'ютерного зору;
- 2) яке фото надсилати? Надсилайте чітке фото одного продукту на нейтральному фоні.
- 3) чи зберігаються результати? Так, бот зберігає історію ваших перевірок. Команда /history виведе останні;
- 4) чи безпечно це? Так, бот не передає ваші фото третім особам поза системою;
- 5) хто розробник? Цей бот створено в межах дипломної роботи. Розробник - Яценко Аркадій, група ІК-11.

Це основне, що потрібно знати користувачеві для коректного та впевненого користування ботом.

Далі переглянемо яку інструкцію пропонує бот для користувача, для цього потрібно натиснути на кнопку «Інструкція» і переглянути результат(рис. 4.5).

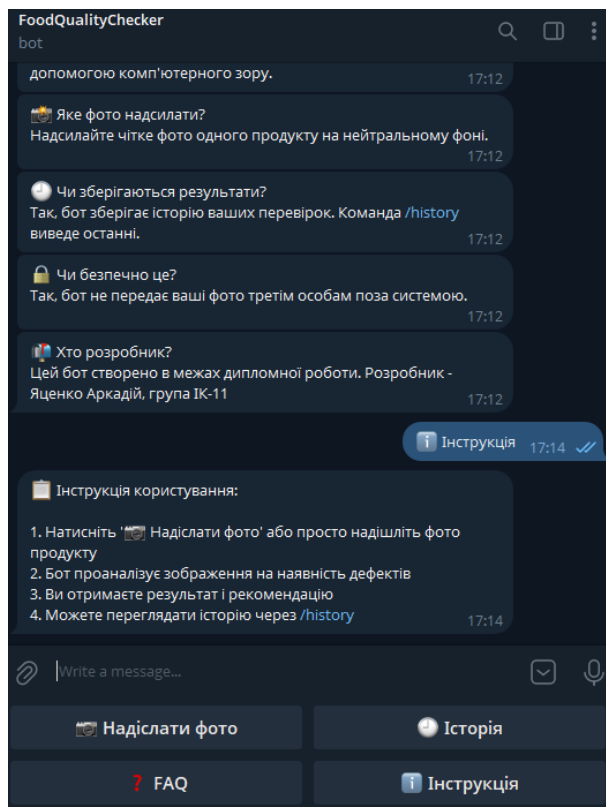


Рисунок 4.5 – Інструкція користування

Інструкція, надана ботом в легкому і доступному форматі у вигляді одного повідомлення, має кілька пунктів, це часткове повторення вступного повідомлення. Вона може бути корисна тоді, коли користувач давно не користувався ботом або загубив вступне повідомлення.

Тепер перейдемо до основної функції системи, а саме оцінки якості харчових продуктів шляхом детекції неякісних об'єктів. Натиснемо в меню на кнопку «Надіслати фото». Результат на рисунку 4.6.

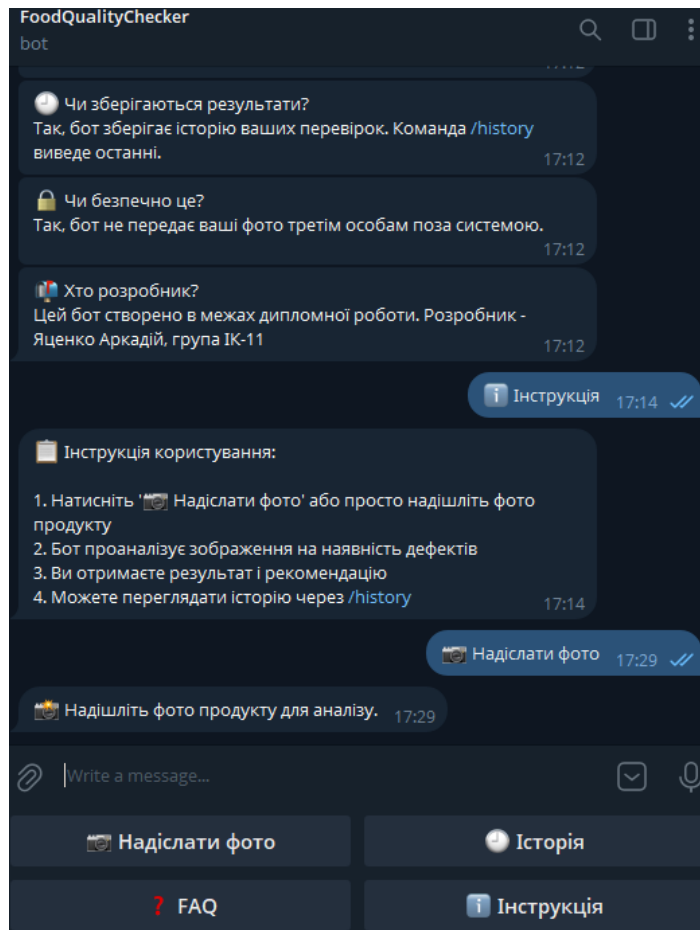


Рисунок 4.6 – Кнопка «Надіслати фото»

Далі необхідно натиснути на «скріпку» зліва біля меню(рис. 4.7) і обрати фото для аналізу з галереї, або зробити нове, та відправити прямо в чат з ботом. Після цього зображення відправиться на обробку, система проведе аналіз та виведе результат, як показано на рисунку 4.8.

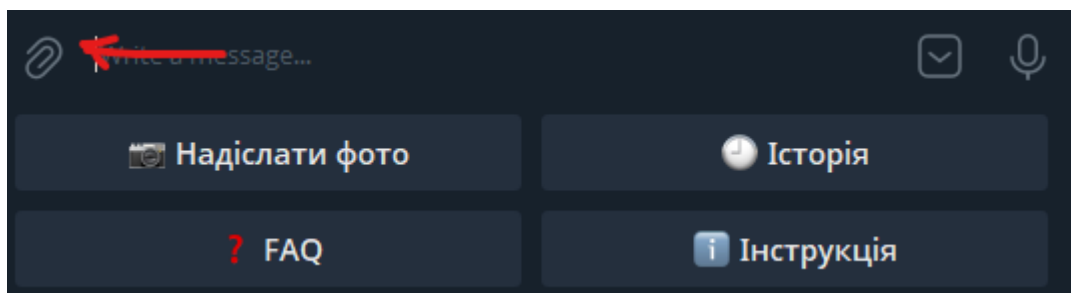


Рисунок 4.7 – Кнопка для вибору фото



Рисунок 4.8 – Результат аналізу фото

Як можна побачити на рисунку, аналіз було здійснено успішно. З отриманого від бота повідомлення можна зробити кілька висновків:

- 1) система успішно отримала зображення для обробки;
- 2) початкове зображення було відфільтровано для проведення детекції;
- 3) на зображення поміщено bounding boxes (рамки навколо дефектів);
- 4) проведено аналіз і створено звіт про виявлені дефекти;
- 5) створено рекомендацію на основі звіту.

Далі необхідно впевнитися, що система коректно працює на всіх визначених в вимогах типах дефектів, а також чи валідно працює система для об'єктів, у яких немає дефектів. Для цього проведемо ще кілька тестів цієї функції. Спочатку проведемо тест з повністю цілим, здоровим об'єктом(рис. 4.9).



Рисунок 4.9 – Тестування системи на нормальному об’єкті

Як бачимо, система вірно розпізнала недефектний продукт та повідомляє, що він придатний до споживання.

Проведемо ще один тест, цього разу візьмемо продукт з цвіллю і переглянемо як система його розпізнає(рис. 4.10).



Рисунок 4.10 – тестування на об’єкті з цвіллю

Зм.	Лист	№ докум.	Підпис	Дата

Система вірно розпізнала дефекти як цвіль і повідомила про це користувача, ураження були розкидані в кількох місцях, тому система зафіксувала їх там як окремі і вивела в результаті 6 об'єктів.

А зараз проведемо тестування на ще одній фотографії, вона цікава тим, що на ній шукаються механічні пошкодження. Результат зображено на рисунку 4.11.



Рисунок 4.11 – Детекція механічних пошкоджень

Система правильно розпізнала об'єкт з механічними пошкодженнями, як і вимагалось від неї.

На кінець проведемо тест, де потрібно буде розпізнати кілька різних видів дефектів на одному зображенні, результат зображено на рисунку 4.12.

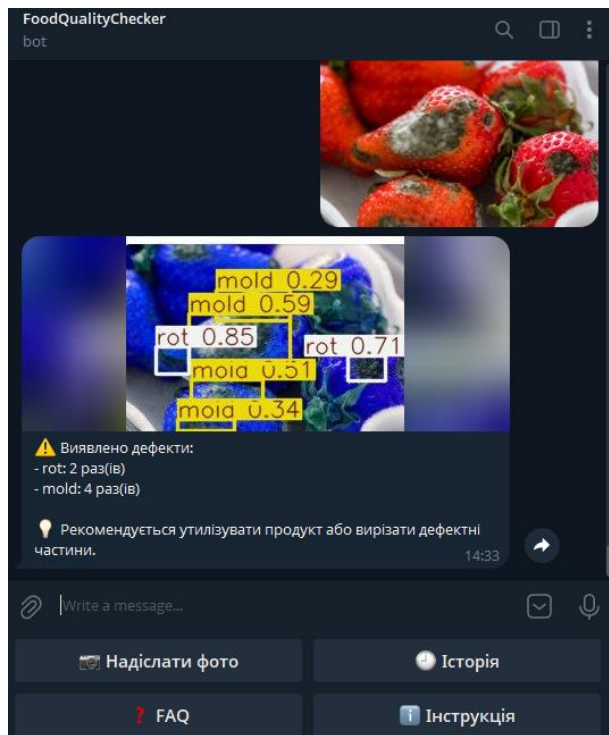


Рисунок 4.12 – Детекція різних типів дефектів

Система правильно різні типи дефектів: гниль і цвіль.

Тепер після кількох тестів детекції неякісних об'єктів можна повернутися до тестування функції перегляду історії. Натиснемо знову на кнопку «Історія» і перевіримо результат(рис. 4.13).

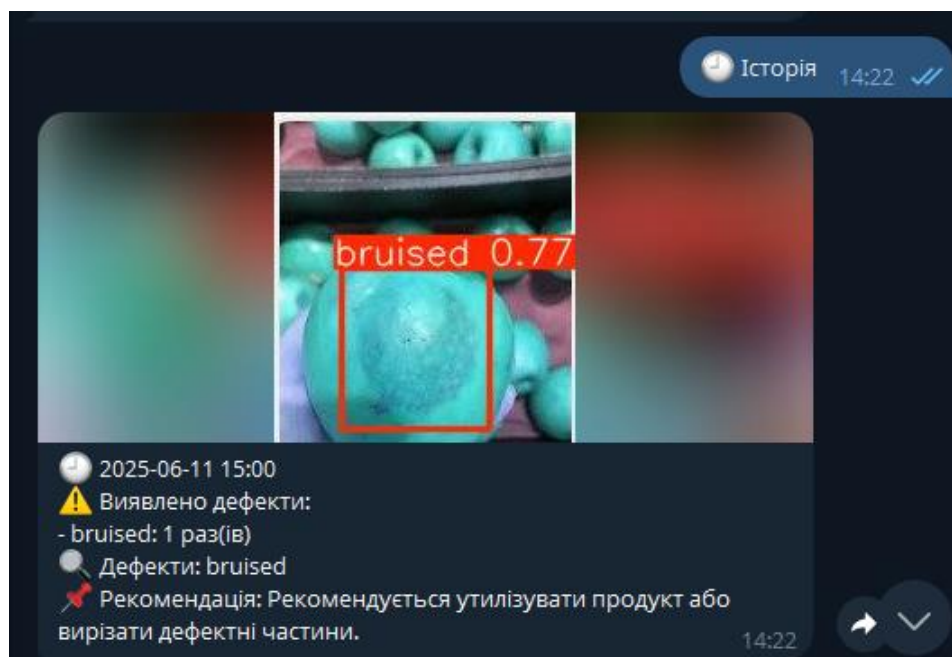


Рисунок 4.13 – Перегляд історії

Зм.	Лист	№ докум.	Підпис	Дата

Ця функція коректно виводить останні результати детекції об'єктів у форматі: 1 запис – 1 повідомлення в чаті. Також для зручності у повідомленні окрім звичних вже показників по кількості й типах дефектів вказана ще й дата та час аналізу. Оскільки історія складається в кількох повідомлень, то інші результати команди знаходяться в Додатку Б.

Отже, було описано весь функціонал системи, доступний звичайному користувачеві. Загалом інтерфейс системи є простим та інтуїтивно зрозумілим для будь якого користувача, що дозволить йому економити час та зусилля в користуванні.

4.2 Адміністратор

Адміністратор має найбільше можливостей з усіх можливих користувачів. Його функціонал є розширеним функціоналом звичайного користувача з деякими додатковими можливостями.

Аутентифікація адміністратора є такою ж, як у звичайного користувача. Початок його роботи з ботом також є аналогічним. Але він має трохи інший інтерфейс. Відмінність проявляється у наявності додаткової кнопки в меню(рис. 4.14).

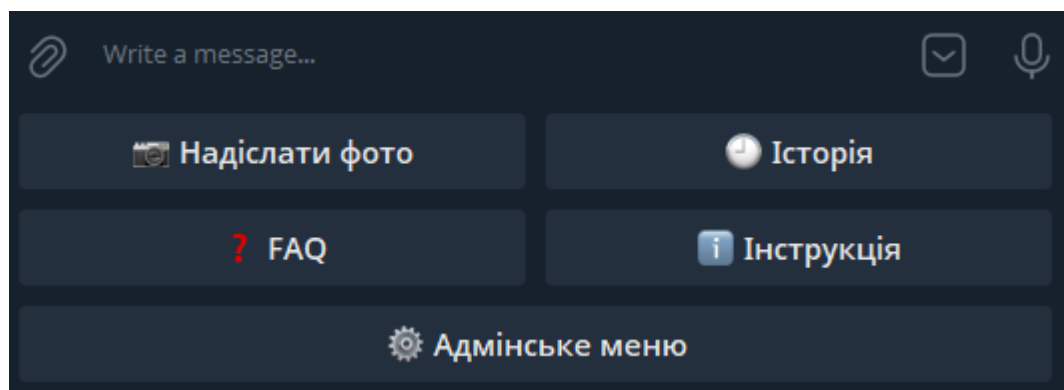


Рисунок 4.14 – Загальне меню у адміністраторів

Ця кнопка дозволяє перейти до адміністраторського меню зі своїми кнопками, як показано на рисунку 4.15.

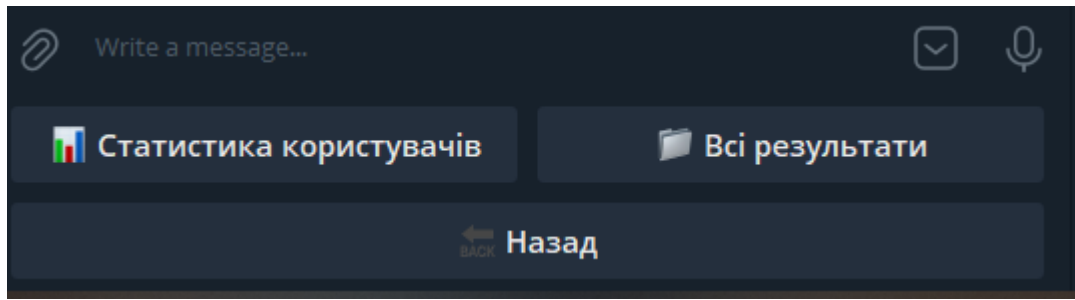


Рисунок 4.15 – Адміністраторське меню

Кнопка «Назад» повертає до початкового меню, а 2 інші мають свої власні функції. Натиснемо на кнопку «Статистика користувачів» і перевіримо виконання, результат на рисунку 4.16.

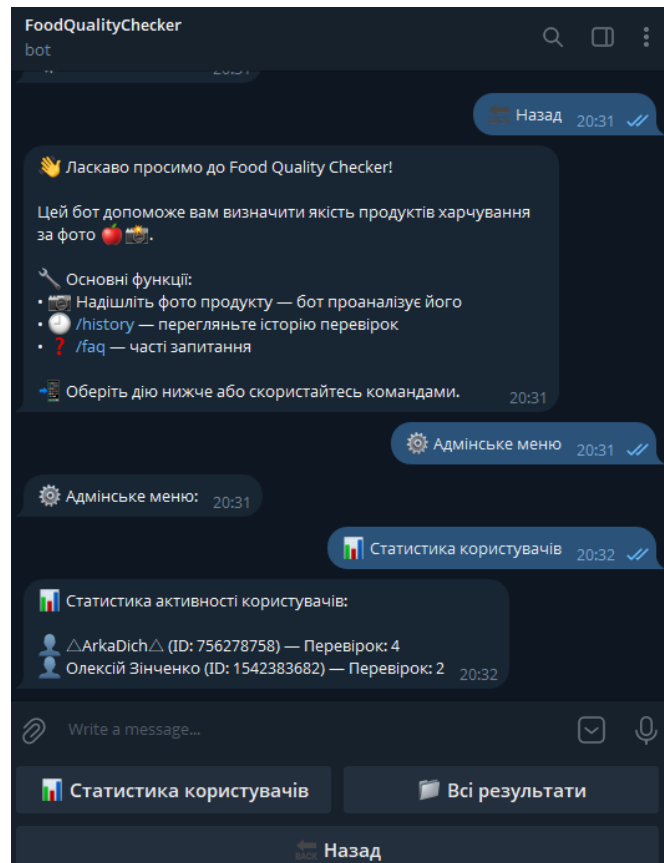


Рисунок 4.16 – Перегляд статистики користувачів

Як можна бачити на рисунку, система успішно зібрала статистику перевірок усіма користувачами системи. Отримана статистика показує який користувач скільки виконав перевірок.

Тепер перевіримо функцію перегляду всіх результатів, для цього потрібно натиснути кнопку «Всі результати», виконання функції можна переглянути на рисунку 4.17.

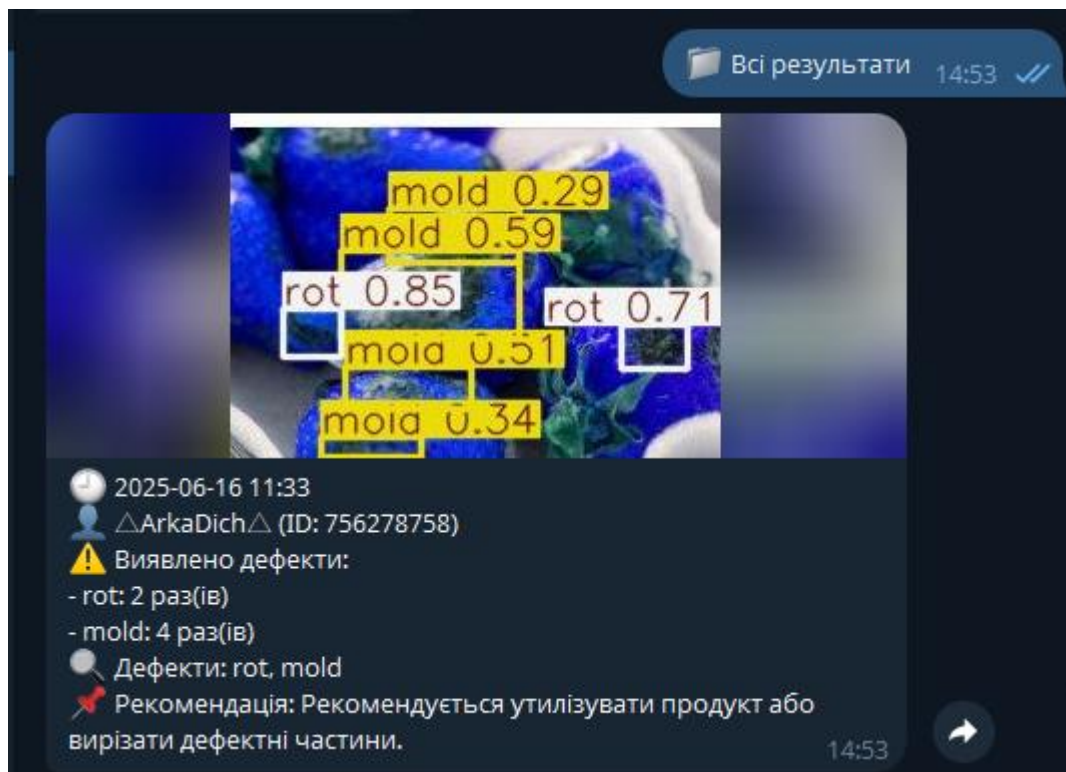


Рисунок 4.17 – Перегляд всіх результатів

Отже, дана функція правильно виводить всі запити, що робили користувачі, які збережені в БД. Система видає інформацію в форматі: 1 результат – 1 повідомлення. Вигляд повідомлення для виводу інформації схожий на вивід історії, але відрізняється наявністю інформації про користувача системи, який цей запит робив. Повністю всі результати, які видає система можна переглянути в Додатку В.

Завдяки такій простій навігації та інтуїтивно зрозумілому інтерфейсу адміністратор може легко переглядати і збирати необхідну йому інформацію.

Висновки до розділу 4

У даному розділі було проведено демонстрацію роботи розробленої автоматизованої системи контролю якості харчових продуктів на основі комп'ютерного

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

зору та чат-бота Telegram. Було детально описано всі основні сценарії взаємодії користувачів з системою, а саме: перевірка якості продуктів, перегляд історії перевірок, ознайомлення з поширеними запитаннями та інструкцією користування. Для кожного кроку були надані пояснення щодо роботи інтерфейсу, доступних кнопок та відповідей від системи. Перевірена також робота системи при нелогічних запитах(наприклад, перегляд історії запитів ще до будь яких запитів), система показала свою підготовленість до таких кроків.

Окрему увагу було приділено демонстрації можливостей адміністратора, де було показано як адміністратор може отримати доступ до статистики користувачів, а також перегляду всіх результатів перевірок. Було детально описано особливості кожного етапу взаємодії адміністратора з системою.

У ході демонстрації було підтверджено відповідність функціональності розробленої системи встановленим вимогам, визначеним у попередніх розділах. Кожен сценарій роботи користувачів та адміністратора було успішно реалізовано та перевірено у процесі тестування. Таким чином, можна зробити висновок, що розроблений програмний продукт відповідає сучасним вимогам до систем автоматизації контролю якості продуктів, забезпечує зручність користування та високу продуктивність.

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

ВИСНОВКИ

Під час виконання дипломного проєкту було проведено дослідження проблем автоматизації контролю якості харчових продуктів із застосуванням комп'ютерного зору. У процесі роботи було здійснено аналіз існуючих рішень, визначено недоліки та переваги кожного з них, а також обґрунтовано необхідність розробки власної автоматизованої системи для визначення дефектів харчових продуктів.

Було проведено всебічний аналіз існуючих методів контролю якості харчових продуктів, що застосовуються в підприємствах харчової промисловості. Було виявлено, що більшість підприємств досі покладаються на ручний або напівручний контроль якості, що призводить до значного людського фактору, помилок та неефективності. Це підтвердило доцільність створення автоматизованої системи контролю якості продуктів із застосуванням комп'ютерного зору.

У процесі роботи було сформульовано як функціональні, так і нефункціональні вимоги до системи, що дозволило визначити чіткі критерії успішної реалізації проєкту. Зокрема, було визначено вимоги щодо безпеки даних користувачів, швидкодії, зручності інтерфейсу та можливості масштабування. Для реалізації проєкту було обрано такі технології: мову програмування Python, бібліотеку комп'ютерного зору YOLOv8, фреймворк python-telegram-bot, ORM SQLAlchemy, а також SQLite як СУБД. Додатково були використані Pillow для роботи із зображеннями та сервіс Roboflow для підготовки та анотації датасету. Розробка та тестування відбувалася в середовищі розробки PyCharm.

Була розроблена архітектура системи, яка базується на модульному підході та забезпечує чітке розділення обов'язків між компонентами: клієнтська частина у вигляді Telegram-бота, серверна частина для обробки фотографій та збереження результатів, база даних для зберігання інформації. Було реалізовано ключовий функціонал: обробка фотографій користувачів, визначення дефектів продуктів, формування рекомендацій, ведення історії перевірок та FAQ. Додатково була реалізована

					ІК11.310БАК.006 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

можливість адміністрування: адміністратор отримує доступ до всіх результатів перевірок, може переглядати активність користувачів та змінювати налаштування моделі.

Система була протестована у двох режимах: користувацькому та адміністративному. Було продемонстровано всі основні сценарії використання, включаючи надсилання фотографій для перевірки, отримання результатів та рекомендацій, перегляд історії перевірок, а також адміністративні функції — перегляд усіх перевірок, доступ до фото, звітів та активності користувачів. Було підтверджено відповідність функціоналу встановленим вимогам.

У результаті виконання проекту було створено прототип автоматизованої системи контролю якості харчових продуктів із застосуванням комп'ютерного зору. Система відповідає сучасним вимогам, забезпечує ефективну обробку фотографій та виведення результатів у реальному часі. Впровадження системи дозволить підприємствам значно підвищити якість контролю продуктів, зменшити вплив людського фактору та оптимізувати роботу персоналу. Це також сприяє цифровій трансформації харчової промисловості, що відповідає сучасним тенденціям розвитку.

Всі поставлені завдання були виконані в повному обсязі, а результати відповідають завданню на дипломне проектування. Розроблена система є універсальною та гнучкою у використанні, забезпечуючи як індивідуальні перевірки користувачів, так і адміністрування, що дозволяє ефективно впроваджувати її на практиці.

Отже, виконана робота довела свою доцільність і практичну значущість. Розроблена система відповідає всім функціональним та нефункціональним вимогам, демонструє стабільну роботу та має перспективи для подальшого розвитку та вдосконалення.

					ІК11.310БАК.006 ПЗ	Арк.
						60
Зм.	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. IFPRI. The Costs and Benefits of Quality Control in the Food Industry [Електронний ресурс] – Режим доступу: <https://www.ifpri.org/publication/costs-and-benefits-quality-control-food-industry>
2. FAO. The State of Food and Agriculture 2019. Moving Forward on Food Loss and Waste Reduction [Електронний ресурс] – Режим доступу: <https://www.fao.org/publications/sofa/2019/en/>
3. FAO. Food Loss and Food Waste [Електронний ресурс] – Режим доступу: <https://www.fao.org/food-loss-and-food-waste/en/>
4. Food Engineering Magazine. Advances in Visual Inspection Systems for Food Quality Control [Електронний ресурс] – Режим доступу: <https://www.foodengineeringmag.com/articles/100835-advances-in-visual-inspection-systems-for-food-quality-control>
5. MarketsandMarkets. AI in Food & Beverages Market — Global Forecast to 2030 [Електронний ресурс]. – Режим доступу: <https://www.marketsandmarkets.com/Market-Reports/ai-in-food-beverages-market-125432465.html>
6. Odoo Inventory [Електронний ресурс] – Режим доступу: <https://www.odoo.com/page/inventory>
7. WithoutWire [Електронний ресурс] – Режим доступу: <https://www.withoutwire.com/>
8. SAP ERP QM [Електронний ресурс] – Режим доступу: <https://help.sap.com/docs/SAP\ ERP>
9. Python Software Foundation. What is Python? [Електронний ресурс] – Режим доступу: <https://www.python.org/doc/essays/blurb/>
10. Ultralytics. YOLOv8 Documentation [Електронний ресурс] – Режим доступу: <https://docs.ultralytics.com/>
11. ResearchGate. Architecture of YOLOv8 [Електронний ресурс] – Режим доступу: <https://www.researchgate.net/figure/Architecture-of-YOLOv8\ fig2\ 380603155>

12. Python Telegram Bot Library. Documentation [Електронний ресурс] – Режим доступу: <https://docs.python-telegram-bot.org/en/stable/>

13. SQLite. About SQLite [Електронний ресурс] – Режим доступу: <https://www.sqlite.org/about.html>

14. SQLAlchemy. SQLAlchemy ORM Documentation [Електронний ресурс] – Режим доступу: <https://docs.sqlalchemy.org/en/20/orm/>

15. Roboflow. Getting Started with Roboflow [Електронний ресурс] – Режим доступу: <https://docs.roboflow.com/>

16. JetBrains. PyCharm Overview [Електронний ресурс] – Режим доступу: <https://www.jetbrains.com/pycharm/>

17. Nielsen Norman Group. Simplicity vs. Complexity in User Interfaces [Електронний ресурс] – Режим доступу: <https://www.nngroup.com/articles/simplicity-vs-complexity/>

18. Statista. Average user expectations for response time in mobile apps (2023) [Електронний ресурс] – Режим доступу: <https://www.statista.com/statistics/1255842/mobile-user-expectations-response-time/>

19. IBM Security. Cost of a Data Breach Report 2023 [Електронний ресурс] – Режим доступу: <https://www.ibm.com/reports/data-breach>

20. Gartner. Scalable architectures in computer vision systems (2022) [Електронний ресурс] – Режим доступу: <https://www.gartner.com/en/documents/4000123>

21. Основні типи архітектур ПЗ [Електронний ресурс] – Режим доступу: <https://www.artofba.com/uk/post/main-types-of-software-architecture>