

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри )

## Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОНТРОЛЮ  
ТЕМПЕРАТУРНОГО РЕЖИМУ ЗБЕРІГАННЯ ХАРЧОВИХ ПРОДУКТІВ ТА  
ЛІКАРСЬКИХ ЗАСОБІВ В ХОЛОДИЛЬНИХ УСТАНОВКАХ  
SOFTWARE DEVELOPMENT FOR CONTROLLING THE TEMPERATURE  
CONDITIONS OF FOOD AND MEDICINES STORAGE IN REFRIGERATION  
UNITS

Виконав(ла): студент(ка) 4 курсу, групи КНТ-130

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

РОГОЖКІН Я.А.

(ПРИЗВИЩЕ та ініціали)

Керівник СТЕПАНЕНКО О.О.

(ПРИЗВИЩЕ та ініціали)

Рецензент ПОЛЯКОВ М.О.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний університет «Запорізька політехніка»  
(повне найменування закладу вищої освіти)

Факультет КНТ  
Кафедра програмних засобів  
Ступінь вищої освіти бакалавр  
Спеціальність 121 Інженерія програмного забезпечення  
(код і найменування)  
Освітня програма (спеціалізація) Інженерія програмного забезпечення  
(назва освітньої програми (спеціалізації))

**ЗАТВЕРДЖУЮ**

Завідувач кафедри ПЗ, д.т.н, проф.  
Сергій СУББОТІН  
“ ” 2024 року

**З А В Д А Н Н Я**

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

РОГОЖКІНА Ярослава Андрійовича

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Розробка програмного забезпечення для контролю температурного режиму зберігання харчових продуктів та лікарських засобів в холодильних установках Software Development for Controlling the Temperature Conditions of Food and Medicines Storage in Refrigeration Units  
керівник проєкту (роботи) к.т.н., доцент, СТЕПАНЕНКО Олександр Олексійович,

(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “ 24 ” квітня 2024 року  
№ 168

2. Строк подання студентом проєкту (роботи) 10 червня 2024 року  
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання  
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Вибір та обґрунтування структури системи, яка проєктується. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво оператора. 6. Експерименти та результати.  
5. Перелік графічного матеріалу (з з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів)  
Слайди презентації

## 6. Консультанти розділів проєкту (роботи)

| Розділ              | ПРИЗВИЩЕ, ініціали та посада консультанта | Підпис, дата   |                           |
|---------------------|-------------------------------------------|----------------|---------------------------|
|                     |                                           | завдання видав | прийняв виконане завдання |
| 1-6 Основна частина | СТЕПАНЕНКО О.О., доцент                   |                |                           |
| Нормоконтроль       | ДЕЙНЕГА Л.Ю., ст. викладач                |                |                           |

7. Дата видачі завдання « 15 » квітня 2024 року.

## КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів дипломного проєкту (роботи)                             | Строк виконання етапів проєкту ( роботи ) | Примітка     |
|-------|----------------------------------------------------------------------|-------------------------------------------|--------------|
| 1     | Постановка завдання роботи.                                          | 1 тиждень                                 | Завдання, ТЗ |
| 2     | Аналіз предметної області.                                           | 1 тиждень                                 | Розділ 1     |
| 3     | Вибір засобів розробки програмного продукту.                         | 2 тиждень                                 | Розділ 2     |
| 4     | Розробка архітектури програми.                                       | 2 тиждень                                 | Розділ 3     |
| 5     | Розробка програми                                                    | 3-4 тижні                                 | Розділ 4, 5  |
| 6     | Тестування та експериментальне дослідження програмного забезпечення. | 5 тиждень                                 | Розділ 6     |
| 7     | Оформлення пояснювальної записки та документів до неї.               | 6 тиждень                                 | Додатки      |
| 8     | Нормоконтроль та рецензування.                                       | 7 тиждень                                 |              |
| 9     | Захист роботи.                                                       | 8 тиждень                                 |              |

Студент(ка)

\_\_\_\_\_  
( підпис )      Ярослав РОГОЖКІН  
(Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

\_\_\_\_\_  
( підпис )      Олександр СТЕПАНЕНКО  
(Ім'я ПРИЗВИЩЕ)

## РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра: 117 с., 8 табл., 60 рис., 4 дод., 25 джерела.

ТЕМПЕРАТУРНИЙ ДІАПАЗОН, ПРОДУКТИ, ЛІКИ, ПРОГНОЗУВАННЯ, C#, ТЕРМІН ПРИДАТНОСТІ.

Об'єкт дослідження – процес моніторингу температури зберігання, термінів придатності харчових продуктів та ліків та їх розміщення в холодильниках.

Предмет дослідження – методи моніторингу температури зберігання, термінів придатності харчових продуктів та ліків та їх розміщення в холодильниках.

Мета роботи – дослідження та програмна реалізація методів моніторингу термінів придатності продуктів харчування та температури зберігання продуктів харчування.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування C#, середовище програмування Visual Studio 2022, персональний комп'ютер з процесором Intel Core i5 під управлінням операційної системи Microsoft Windows 10.

Результати. Створено програмний Windows-застосунок, який здійснює моніторинг температурних діапазонів при яких зберігаються продукти харчування та моніторинг термінів придатності продуктів харчування та ліків.

Висновки. Проведено аналіз області використання та створено програмний продукт для моніторингу температурних діапазонів зберігання та термінів придатності продуктів харчування.

Галузь використання – прогнозування передчасного псування продуктів та контроль температурних меж в холодильних установках.

## **ABSTRACT**

Explanatory note to the diploma qualifying work of the master: 117 figures, 8 tables, 60 figures, 4 appendixes, 25 sources.

TEMPERATURE RANGE, PRODUCTS, MEDICINES, FORECASTING, C#, SHELF LIFE.

The object of research is the process of monitoring the storage temperature, shelf life of food and medicines and their placement in refrigerators.

Subject of research - methods of monitoring the storage temperature, shelf life of food and medicines and their placement in refrigerators.

Purpose - to study and implement methods for monitoring the shelf life of food and storage temperature of food.

Materials, methods and equipment: object-oriented programming, C# programming language, Visual Studio 2022 programming environment, a personal computer with an Intel Core i5 processor running the Microsoft Windows 10 operating system.

Results. A Windows application has been created that monitors the temperature ranges at which food is stored and monitors the shelf life of food and

Conclusions. The analysis of the field of application was carried out and a software product for monitoring the temperature ranges of storage and shelf life of food was created.

The field of application is prediction of premature food spoilage and control of temperature limits in refrigeration units.

## ЗМІСТ

С.

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Перелік скорочень та умовних позначень.....                       | 8  |
| Вступ.....                                                        | 9  |
| 1 Аналіз предметної області.....                                  | 13 |
| 1.1 Основні поняття.....                                          | 14 |
| 1.2 Огляд існуючих методів збору даних.....                       | 16 |
| 1.3 Порівняння методів збору даних.....                           | 21 |
| 1.4 Порівняння застосунків зі схожим функціоналом.....            | 23 |
| 1.5 Постановка завдання до роботи.....                            | 24 |
| 1.6 Висновок.....                                                 | 25 |
| 2 Вибір та обґрунтування структури системи, яка проектується..... | 25 |
| 2.1 Вибір мови програмування.....                                 | 26 |
| 2.2 Порівняння мов програмування.....                             | 32 |
| 2.3 Вибір середовища розробки.....                                | 33 |
| 2.4 Порівняння середовищ розробки.....                            | 39 |
| 2.5 Структура програмного продукту.....                           | 40 |
| 2.6 Висновок.....                                                 | 41 |
| 3 Основні рішення щодо реалізації компонентів системи.....        | 41 |
| 3.1 Основні відомі методи.....                                    | 42 |
| 3.2 Висновок.....                                                 | 46 |
| 4 Керівництво програміста.....                                    | 46 |
| 4.1 Призначення й умови використання.....                         | 47 |
| 4.2 Звертання до програмного продукту (файли проєкту).....        | 47 |
| 4.3 Вхідні і вихідні дані.....                                    | 48 |
| 4.4 Повідомлення.....                                             | 52 |

|                                                             |     |
|-------------------------------------------------------------|-----|
| 4.5 Структура бази даних.....                               | 53  |
| 4.6 Повідомлення оператору.....                             | 55  |
| 4.7 Висновок.....                                           | 56  |
| 5 Керівництво оператора.....                                | 56  |
| 5.1 Призначення й умови виконання програмного продукту..... | 57  |
| 5.2 Виконання програмного продукту.....                     | 57  |
| 5.3 Висновок.....                                           | 63  |
| 6 Експерименти та результати.....                           | 63  |
| 6.1 Проведення експериментів.....                           | 64  |
| 6.2 Висновок.....                                           | 69  |
| Висновки.....                                               | 69  |
| Перелік джерел посилання.....                               | 70  |
| Додаток А Технічне завдання.....                            | 74  |
| Додаток Б Опис програмного продукту.....                    | 80  |
| Додаток В Текст програми.....                               | 95  |
| Додаток Г Слайди презентації.....                           | 112 |

## ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

|     |                                       |
|-----|---------------------------------------|
| API | – Application Programming Interface;  |
| IDE | – Integrated Development Environment; |
| ЛП  | – лікарські препарати;                |
| МН  | – машинне навчання;                   |
| ПП  | – програмний продукт;                 |
| ПХ  | – продукти харчування;                |
| ТД  | – температурний діапазон;             |
| ТП  | – термін придатності;                 |
| ХУ  | – холодильна установка.               |

## ВСТУП

**Актуальність.** Потреба правильного зберігання продуктів та дотримання температурного режиму виникає постійно. В наслідок різних непередбачуваних ситуацій(війна, природна катастрофа або різних збоїв у роботі електропостачання) трапляються випадки відключення світла. Для деяких продуктів важливо суворо дотримуватись температурного режиму зберігання. Це такі продукти як м'ясні, молочні, або лікарські препарати. Саме тому важливо вчасно зреагувати на такі ситуації, аби уникнути, або хоча б значно зменшити втрати продуктів харчування або ЛП.

**Новизна.** Звісно, на ринку є пропозиції контролю температури в холодильних установах. Починаючи зі звичайних термометрів, які ви вішаєте в холодильник і закінчуючи Wi-Fi датчиками, які надсилають дані на смартфон. Та ж навіть деякі холодильники оснащені датчиками і міні дисплеєм з заводу. Але в чому ж унікальність нашої розробки. Наша розробка не тільки слідкує за зміною температури, але і може розрахувати для окремого продукту як зміна температури і хід часу впливають на його придатність. Адже кожен продукт старіє по своєму. Саме за зміни в придатності продукту і буде відповідати наша розробка. На разі таке рішення є новинкою на ринку, адже аналогічних систем не вдалося знайти.

**Практична цінність.** Розрахувавши придатність продукції її можна набагато ефективніше застосовувати. Це зменшує ризики того, що продукція пропаде і зіпсується. А якщо таких ризиків буде менше, то більше людей зможуть отримати допомогу, якої потребують. Та і для звичайних магазинів, аптек чи складів дана програма представляє неабияку цінність. Це дасть змогу збільшити придатних до вживання продуктів, а отже і виручка грошей з продуктів стає більшою.

Продукти харчування (надалі ПХ) – це різноманітні товари, які вживаються для задоволення потреб організму в харчуванні. Зберігання ПХ є

важливою складовою безпечного харчування. Коректне зберігання допомагає зберегти якість продуктів, запобігти появі шкідливих мікроорганізмів та зберегти їх смакові та поживні якості.

ПХ мають велику важливість для здоров'я та добробуту людей. Вони є джерелом необхідних поживних речовин, включаючи білки, вуглеводи, жири, вітаміни, мінерали та воду. Їх правильне харчування забезпечує енергію для фізичної та розумової активності, підтримує функціонування організму і сприяє росту та розвитку.

Збереження ПХ у холодильних установках має вирішальне значення для збереження їх якості та безпеки. Ось декілька причин, чому правильне зберігання у холодильнику важливо:

Запобігання розмноженню шкідливих мікроорганізмів. Холодильники забезпечують низьку температуру, що уповільнює розмноження бактерій, вірусів та грибків у продуктах. Це допомагає запобігти харчовим отруєнням та захворюванням, пов'язаним із споживанням збитих продуктів.

Збереження свіжості та смакових якостей. Багато продуктів, таких як молочні продукти, м'ясо, риба, фрукти та овочі, швидко псуються при кімнатній температурі. Зберігання їх у холодильнику допомагає затримати процес псування, зберегти їх свіжість, текстуру та смак.

Ефективне управління запасами. Холодильник дозволяє триваліше зберігати продукти, що дозволяє ефективніше планувати придбання та використання ПХ. Це допомагає уникнути марнотратства та забезпечує економію коштів.

Збереження поживної цінності. Холодильники допомагають зберегти вітаміни та мінерали у продуктах, оскільки довше термін зберігання сприяє збереженню поживних речовин.

Важливо дотримуватись вказівок виробника щодо зберігання продуктів у холодильнику, контролювати температуру та вчасно видаляти

збиті або застарілі продукти. Це забезпечить максимальну безпеку та якість ПХ для споживання.

З розвитком продовольчої промисловості і зростанням споживання ПХ з'явилася необхідність встановлення ТП та рекомендованих температурних режимів зберігання. Це стало можливим завдяки науковим дослідженням та технологічному прогресу.

ТП вказує на той період, протягом якого продукт, за заданих умов зберігання, зберігає свою якість і безпеку для споживання. Встановлення ТП базується на результаті наукових досліджень, які визначають, як швидко продукт може псуватися або стати непридатним для споживання з точки зору якості та безпеки.

Температурні режими зберігання продуктів встановлюються з метою забезпечити оптимальні умови збереження. Холодильники та морозильні камери регулюються на певні температури, які рекомендуються для різних видів продуктів. Наприклад, низькі температури у холодильнику сповільнюють розмноження бактерій і зберігають свіжість продуктів, тоді як низькі температури у морозильних камерах зупиняють процес псування і дозволяють продуктам тривати довше.

Для регулювання температури і збереження продуктів у холодильниках також використовуються різні зони з різними температурами, що дозволяє оптимізувати збереження різних видів продуктів.

Зазначені ТП та температурні режими зберігання є важливими для забезпечення якості і безпеки ПХ. Дотримання цих рекомендацій допомагає запобігти харчовим отруєнням, зберегти поживні речовини і зберігати продукти в оптимальному стані для споживання.

Вчасне використання або передача ПХ з ТП, що наближається до закінчення, є важливим з міркувань безпеки та ефективного використання ресурсів. Споживання продуктів після закінчення ТП може призвести до харчових отруєнь, оскільки вони можуть занести шкідливі бактерії або

токсини до організму. Віддавання таких продуктів на інші потреби, якщо вони ще можуть бути використані, допомагає уникнути марнотратства та забезпечити стале забезпечення харчування.

# 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Основні поняття

Термін придатності – це часовий період, протягом якого ПХ вважається безпечним для споживання та зберігає свою якість за належних умов зберігання. Встановлення терміну придатності є важливим етапом виробництва ПХ та дозволяє споживачам визначати, коли продукт може втратити свою якість або стати небезпечним для споживання [1].

ТП базується на наукових дослідженнях, які враховують фізичні, хімічні та мікробіологічні зміни, що відбуваються в продукті протягом часу. Для встановлення ТП проводяться випробування, включаючи оцінку стійкості продукту до псування, мікробного забруднення, зміни текстури, смаку, запаху тощо [2].

ПХ – це будь-які речовини, які люди споживають для задоволення своїх потреб у харчуванні та забезпечення необхідних поживних речовин для організму. ПХ можуть бути рослинного або тваринного походження, а також перероблені або неперероблені [3].

Ліки – це ЛП, які залежно від рівня розвитку суспільства в різні часи мали різний вигляд і якість, але завжди використовувалися з метою лікування та запобігання хворобам [4].

Ліки використовувалися з сивої давнини і лише на підставі археології, етнографії та інших даних можна робити висновок про те, що відбувалось у ті далекі часи.

Використовуючи досвід попередніх поколінь, людство емпіричним шляхом відкривало лікувальні властивості багатьох рослин. Майже до XIX ст. як Л. використовувалася сировина рослинного, тваринного або мінерального походження, що піддавалася відносно простому обробленню. Згодом з рослин та іншої сировини почали виділяти або концентрувати діючі речовини, встановлювати їх структуру, здійснювати синтез БАР. Значно пізніше, коли

була встановлена певна залежність між хімічною будовою БАР та їх фармакологічною дією, були отримані аналоги і гомологи природних сполук. Нарешті, з вивченням механізму дії Л. багатьох фармакотерапевтичних груп, був започаткований цілеспрямований пошук нових ефективних лікарських речовин. Почали розвиватися нові, більш сучасні методи синтезу лікарських речовин (комбінаторний, комп'ютерний тощо). На різних етапах розвитку фармацевтичної технології в медицині використовувалися ЛП у різних лікарських формах (кашки, збори, порошки, пілюлі, пластирі, настої, примочки, гранули, таблетки, мазі тощо). Нині фармацевтичні фірми пропонують більш складні лікарські системи (див. Терапевтичні системи з адресною доставкою лікарських речовин), які, на відміну від звичайних ЛП, характеризуються пролонгованою дією, контрольованим вивільненням діючої субстанції та її цілеспрямованою (адресною) доставкою до певних органів або тканин організму. Залежно від бажаного фармакологічного ефекту ЛП вводять в організм різними шляхами (див. Шляхи введення ліків). Для кожного шляху введення ЛП розробляється певна лікарська форма, метод застосування, а інколи і спеціальний інструментарій. І.П. Павлов ще в 1895 р. відзначав універсальність лікарської терапії, яка використовується навіть при різних медичних процедурах, що не втратило свого значення донині. Ліки стали могутньою зброєю в боротьбі з найтяжчими захворюваннями; вони прискорюють видужання, поліпшують самопочуття, підвищують працездатність, мобілізують захисні сили організму [4].

ХУ (також відома як холодильник або холодильна камера) – це пристрій або система, призначена для зберігання та охолодження ПХ та інших перетворюваних товарів при низьких температурах, що сприяє збереженню їх свіжості, якості та безпеки [5].

ХУ працює на основі принципу теплового насоса, де використовується компресія та розширення холодоагенту (наприклад, фреону) для перенесення

тепла зсередини приміщення до зовнішнього середовища. Це дозволяє створити низьку температуру всередині холодильника.

ХУ мають усередині розташовані полиці, ящики та дверцята, що дозволяє організувати зручне та організоване зберігання продуктів. Деякі ХУ також мають функції автоматичного розморожування, регулювання вологості, контролю температури тощо.

Основні завдання ХУ полягають у:

- зниженні температури внутрішнього простору для збереження ПХ при оптимальних умовах температури і вологості;
- запобіганні розмноженню бактерій, грибків та інших мікроорганізмів, що призводять до псування продуктів;
- збереженні свіжості, текстури, смаку та якості продуктів на тривалий період.

Важливо правильно використовувати холодильну установку, розміщувати продукти згідно з їх рекомендованими температурними режимами і дотримуватись правил гігієни для запобігання харчовим [5].

## **1.2 Огляд існуючих методів збору даних**

Основними методами збору даних(в нашому випадку фіксація температури в холодильниках) є ручний метод(оператор перевіряє термометри та записує отримані дані в журнал), напівавтоматичний(оператор сидить за пультом моніторингу та веде фіксацію отриманих даних) та автоматичний(електронна система сама веде облік температури в холодильниках, оператор спостерігає за отриманими даними) [6].

### 1.2.1 Ручний збір даних температури холодильника

Ручний метод збору даних температури холодильника передбачає використання простих інструментів, таких як ртутні або цифрові термометри, для зчитування температури та записування цих даних вручну на папері або в електронній таблиці. Цей метод потребує регулярної перевірки і фіксації температурних показників оператором [7].

Розбір процедури збору даних:

а) підготовка інструментів:

1) використання ртутного або цифрового термометра, що відповідає стандартам точності;

2) паперовий журнал або електронна таблиця для фіксації даних;

б) зчитування показників:

1) визначення інтервалу, через який буде проводитися зчитування температури (наприклад, кожну годину);

2) розташування термометра в холодильнику у місці, що точно відображає середню температуру (не біля дверей або вентиляційних отворів);

в) фіксація даних:

1) кожне зчитування повинно супроводжуватися записом часу і дати;

2) записування фактичної температури з термометра;

3) фіксація інформації про відповідального оператора, який проводив зчитування;

г) аналіз даних:

1) регулярний огляд зібраних даних для виявлення тенденцій або відхилень від норми;

2) вжиття негайних заходів у разі виявлення відхилень від встановлених температурних норм [7].

Переваги:

- не потребує великих початкових інвестицій у спеціалізоване обладнання чи програмне забезпечення;
- легкий у впровадженні метод, який не вимагає спеціальних навичок або навчання [7].

Недоліки:

- помилки можуть виникати через неточне зчитування або запис даних;
- метод вимагає регулярного ручного втручання, що може бути виснажливим для оператора;
- відсутність можливості миттєвого виявлення і реагування на відхилення температури [7].

### **1.2.2 Напівавтоматичний збір даних температури холодильника**

Напівавтоматичний метод збору даних температури холодильника поєднує в собі використання цифрових пристроїв для автоматичного вимірювання температури з ручним зчитуванням і аналізом цих даних. Цей метод дозволяє зменшити кількість ручної праці, зберігаючи при цьому можливість контролю з боку оператора [8].

Розбір процедури збору даних:

а) підготовка інструментів:

1) пристрої, які автоматично вимірюють і зберігають показники температури з заданими інтервалами часу;

2) програми або додатки для зчитування та аналізу даних з логерів;

б) зчитування показників:

1) логери даних встановлюються в холодильнику і налаштовуються для автоматичного вимірювання температури з певною частотою (наприклад, кожні 5 хвилин, кожную годину тощо);

2) логери зберігають дані у внутрішній пам'яті, що може зберігати значні обсяги інформації;

в) ручне зчитування даних:

1) оператори періодично зчитують дані з логерів, використовуючи програмне забезпечення або вручну;

2) дані переносяться до центральної бази даних або електронної таблиці для подальшого аналізу;

г) аналіз даних:

1) регулярний перегляд зібраних даних для виявлення тенденцій або відхилень від норми;

2) вжиття необхідних заходів у разі виявлення відхилень від встановлених температурних норм [8].

Переваги:

- автоматичне вимірювання температури зменшує ймовірність помилок, пов'язаних з ручним введенням даних;
- зменшується час, необхідний для збору даних, підвищується ефективність процесу;
- дані збираються безперервно без необхідності постійної участі людини [8].

Недоліки:

- дані все ще потрібно періодично зчитувати і аналізувати вручну;
- часткове автоматизування процесу все ще залишає місце для помилок і затримок у виявленні проблем;
- можлива затримка у виявленні відхилень через періодичність зчитування даних [8].

### 1.2.3 Автоматичний збір даних температури холодильника

Автоматичний метод збору даних температури холодильника використовує сучасні технології, такі як бездротові датчики та інтернет речей (IoT), для постійного моніторингу та запису температури. Ці дані автоматично передаються в реальному часі в центральну систему для аналізу та негайного реагування на будь-які відхилення від встановлених норм [9].

Розбір процедури збору даних:

а) підготовка інструментів:

1) пристрої, що встановлюються всередині холодильника та підключаються до Інтернету або локальної мережі для передачі даних;

2) програмне забезпечення або платформа, яка отримує, обробляє та аналізує дані з датчиків;

б) встановлення датчиків:

1) датчики встановлюються у критичних точках всередині холодильника для забезпечення точного вимірювання температури;

2) попереднє калібрування датчиків для забезпечення їх точності;

в) автоматичний збір даних:

1) датчики постійно вимірюють температуру і передають дані в реальному часі (наприклад, кожні кілька секунд або хвилин);

2) дані автоматично зберігаються в базі даних на центральному сервері або в хмарі;

г) аналіз даних:

1) система обробляє дані в реальному часі, виявляючи відхилення від нормальних температурних значень;

2) у разі виявлення відхилення, система автоматично надсилає сповіщення відповідальним особам через електронну пошту, SMS або мобільний додаток;

г) реагування на відхилення:

1) відповідальні особи можуть швидко реагувати на сповіщення, вживаючи необхідних заходів для виправлення ситуації;

2) автоматична реєстрація всіх подій і дій, що були вжиті у відповідь на сповіщення [9].

Переваги:

- можливість миттєвого реагування на відхилення температури;
- дані збираються автоматично, що знижує ймовірність людських помилок;
- система може легко масштабуватися для великих операцій і забезпечує високу ефективність;
- постійний доступ до історичних даних для аналізу тенденцій та покращення процесів [9].

Недоліки:

- високі початкові витрати на впровадження та технічне обслуговування;
- необхідність стабільного інтернет-з'єднання та електроживлення;
- необхідність технічної підтримки та обслуговування системи [9].

### **1.3 Порівняння методів збору даних**

Проаналізувавши роботи [6]–[9] було проведено порівняльний аналіз досліджених методів збору даних (табл.1.1).

Таблиця 1.1 – Порівняльний аналіз досліджених методів збору даних

| Показники/Методи            | Ручний | Напівавтоматичний | Автоматичний |
|-----------------------------|--------|-------------------|--------------|
| Дешевизна                   | +      | –                 | –            |
| Зручність                   | +      | –                 | +            |
| Ефективність                | –      | +                 | +            |
| Точність                    | –      | +                 | +            |
| Моніторинг в реальному часі | –      | +                 | +            |
| Простота                    | +      | +                 | +            |

Як видно з таблиці 1.1, найкращим методом виявився автоматичний метод збору даних. Він має безліч переваг, що робить найкращим серед розглянутих методів. При застосуванні цього метода від оператора вимагається мінімум зусиль та зменшується такий чинник, як фактор. Саме тому саме цей метод планується взяти за основу при розробці ПП.

#### 1.4 Порівняння застосунків зі схожим функціоналом

Як зазначалося в пункті вище, було обрано за основу автоматичний метод збору даних. Для порівняння аналогів розроблюваному застосунку було обрано такі програми моніторингу температури: RuuviTag [10], SensGuard [11] та Thermotrack Webserve [12].

Порівняльний аналіз застосунків наведено в таблиці 1.2.

Таблиця 1.2 – Порівняльний аналіз застосунків

| Показники/Застосунки                                             | Ruuvitag   | SensGuard  | Thermotrack<br>Webserve | Мій<br>застосунок |
|------------------------------------------------------------------|------------|------------|-------------------------|-------------------|
| Встановлення діапазону<br>температури                            | В<br>ручну | Вр<br>учну | Вруч<br>ну              | Ав<br>томатично   |
| Вимірювання в<br>реальному часі                                  | Т<br>ак    | Та<br>к    | Так                     | Та<br>к           |
| Сповіщення про<br>неполадки                                      | Т<br>ак    | Та<br>к    | Так                     | Та<br>к           |
| Можливість<br>встановлення періоду<br>моніторингу<br>температури | Н<br>і     | Ні         | Так                     | Та<br>к           |
| Кросплатформність                                                | Н<br>і     | Та<br>к    | Так                     | Та<br>к           |
| Моніторинг термінів<br>придатності продуктів                     | Н<br>і     | Ні         | Ні                      | Та<br>к           |
| Управління вмістом<br>холодильників                              | Ні         | Ні         | Ні                      | Так               |
| Дистанційний<br>моніторинг                                       | Так        | Так        | Так                     | Так               |

З таблиці 1.2 видно, що запропонований мною програмний застосунок має деякі переваги в порівнянні з аналоговими застосунками. Саме розширення можливостей застосунку і надає йому перевагу у виборі користувачем.

### 1.5 Постановка завдання до роботи

Метою роботи є програмна реалізація методів моніторингу ТП ПХ та ТД зберігання ПХ.

У роботі розглядається вирішення таких проблем як моніторинг ТП ПХ, моніторинг ТД зберігання ПХ та введення електронної бази даних вмісту кожного холодильника.

Один з основних функціоналів ПП є моніторинг термінів придатності ПХ. 1 раз на день ПП аналізує всі ПХ у всіх ХУ, а саме такі параметри як кінцева дата придатності та кінцевий час придатності. За дві неділі до кінцевого дня ТП ПП сповіщає про наявність таких продуктів. Це дасть змогу працівникам реалізувати даний товар в першу чергу. Якщо ПП виявить протермінований продукт, він також сповістить користувача про його наявність і потребу утилізації. Це допоможе зменшити кількість зіпсованих ПХ, що значно позитивно відобразиться на заробітку від продажу товару, та допоможе уникнути зараження інших продуктів, які знаходяться в одному холодильнику.

Ще одним з основних функціоналів є моніторинг ТД зберігання ПХ в окремому холодильнику. Ця можливість досягається завдяки створенні окремої бази даних для кожного холодильника. При кожному додаванні продуктів в холодильник ПП автоматично визначає мінімальне та максимальне значення зберігання ПХ. Періодичність моніторингу ТД користувач може встановити самостійно (наприклад 1 раз на 5 хвилин). у разі виявлення невідповідності до визначеного ТД, ПП повідомить користувача про вихід за межі ТД. Це дозволить швидко зреагувати на дану ситуацію та максимально вберегти ПХ від дочасного псування.

Оскільки моніторинг температури відбувається індивідуально до кожного холодильника, і вже створено базу даних з його набором ПХ, то

доречно розробити і можливість керувати вмістом холодильників. Це дозволить вирішувати такі питання як додавання та видалення вмісту холодильника та можливість швидко знайти потрібний продукт у холодильниках. Це допоможе краще орієнтуватись у вмісті холодильників та більш ефективніше виконувати роботу з ними.

Отже, з метою вирішення проблем, що виникають при зберігання ПХ було прийнято рішення розробки ПП з описаним вище функціоналом.

## **1.6 Висновок**

У цьому розділі було проведено аналіз предметної області, визначено основні поняття, пов'язані з предметною областю і об'єктом дослідження та розробки. Було розглянуто запропоновані рішення та їх принцип роботи.

Було проведено аналіз існуючих методів збору даних, виявлено найбільш вигідний метод та на його основі було розроблено застосунок для моніторингу температури та термінів придатності продуктів. Було проведено порівняння мого застосунку з аналогами.

Було сформульовано і поставлено завдання до роботи.

## **2 ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРИ СИСТЕМИ, ЯКА ПРОЕКТУЄТЬСЯ**

### **2.1 Вибір мови програмування**

Існує декілька сотень мов програмування, кожна з мов має свої особливості, і та, яка з успіхом працює в одному проекті, може зовсім не підходити для іншого, тому вибір мови програмування для розробки ПП є важливим етапом. Найчастіше використовувані мови програмування для створення застосунку є Python, C#, R, Java, Swift [13].

#### **2.1.1 R**

R – це мова програмування, призначена для статистичної обробки даних та роботи з графікою, а також безкоштовне обчислювальне середовище з відкритим вихідним кодом у рамках проекту GNU. Вона була створена як аналог мови S, розробленої у Bell Labs, і є її альтернативною реалізацією. Хоча між мовами є суттєві відмінності, більша частина коду, написаного мовою S, може виконуватися в середовищі R.

R була спеціально розроблена для аналізу даних та створення додатків і програмних рішень, здатних виконувати статистичний та інтелектуальний аналіз даних. Популярна серед науковців та дослідників у сфері науки про дані, мова R пропонує широкий спектр бібліотек та інструментів для таких задач:

- очищення та підготовка даних;
- створення візуалізацій;
- навчання та оцінка алгоритмів МН та глибокого навчання.

Дослідники часто обирають R через її здатність генерувати графіки, готові для публікації, а також через наявність готових математичних формул та позначень. Загалом, R привертає увагу своїми можливостями візуалізації даних, зокрема графіками та діаграмами, які полегшують інтерпретацію

даних і допомагають виявляти закономірності, аномалії та тенденції у наборах даних [14].

Переваги:

- мова R відома своїми потужними статистичними і аналітичними можливостями, що дозволяє створювати додатки, які можуть виконувати складні обчислення та аналізи;
- існує велика кількість пакетів, які розширюють можливості R, включаючи пакети для створення GUI, такі як Shiny та RGtk2;
- мова R має велику та активну спільноту користувачів, яка постійно розробляє нові пакети та надає підтримку;
- додатки, створені на R, можуть працювати на різних операційних системах, таких як Windows, macOS та Linux [14].

Недоліки:

- хоча R має інструменти для створення настільних додатків, її можливості для створення складних GUI додатків обмежені порівняно з іншими мовами програмування, такими як Python або C#;
- як і у випадку з Python, R є інтерпретованою мовою, що може призводити до меншої швидкості виконання у порівнянні з компільованими мовами;
- розгортання R-додатків може бути складнішим через необхідність встановлення середовища R та залежностей;
- порівняно з іншими мовами, R має менше ресурсів та інструментів, спрямованих на розробку настільних додатків [14].

### 2.1.2 C#

C# (C-sharp) – це сучасна, об'єктно-орієнтована мова програмування, розроблена компанією Microsoft в рамках платформи .NET. Вона була створена для того, щоб забезпечити швидку розробку ефективних і надійних

додатків. С# широко використовується для створення настільних додатків, веб-додатків, ігрових додатків (з використанням Unity) і багатьох інших видів програмного забезпечення [15].

Для створення настільних додатків на С# зазвичай використовуються такі фреймворки, як Windows Forms, WPF (Windows Presentation Foundation), і новіший .NET MAUI (Multi-platform App UI), який дозволяє розробляти крос-платформенні додатки [15].

Переваги:

- мова С# був розроблена спеціально для платформи .NET, яка забезпечує глибоку інтеграцію з операційною системою Windows, що дозволяє легко використовувати всі можливості ОС;
- середовище Visual Studio – потужна IDE від Microsoft, забезпечує широкі можливості для розробки, відлагодження та розгортання додатків на С#;
- платформа .NET пропонує велику кількість вбудованих бібліотек, які спрощують виконання багатьох завдань, таких як робота з базами даних, обробка графіки, мережеві операції та багато іншого;
- завдяки .NET Core та .NET 5/6/7, С# додатки можуть працювати на різних платформах, включаючи Windows, macOS та Linux;
- мова С# постійно оновлюється та включає в себе сучасні мовні функції, такі як асинхронне програмування, шаблони та лінк (LINQ) [15].

Недоліки:

- через свою багатофункціональність та глибокі можливості, С# може бути складною мовою для початківців;
- хоча С# стає все більш кросплатформенним, він все ще сильно пов'язаний з екосистемою Microsoft, що може обмежити використання в деяких сценаріях;
- написання простих програм може вимагати значних зусиль через необхідність дотримання формальностей мови та фреймворків;

- хоча .NET дозволяє створювати високопродуктивні додатки, вони можуть поступатися в швидкості виконання додаткам, написаним на низькорівневих мовах, таких як C++ [15].

### 2.1.3 Python

Python – це високорівнева мова програмування загального призначення, відома своєю простотою та читабельністю. Хоча Python часто асоціюється з веб-розробкою, аналізом даних і МН, він також є потужним інструментом для створення настільних додатків. Для розробки таких програм широко використовуються фреймворки, такі як Tkinter, PyQt, PySide, Kivy та wxPython. Ці фреймворки надають інструменти для створення графічних інтерфейсів користувача (GUI), які можуть працювати на різних операційних системах, включаючи Windows, macOS і Linux [16].

Переваги Python:

- мова Python відома своєю простою та зрозумілою синтаксисом, що робить його ідеальним вибором для розробників-початківців і швидкого прототипування;
- більшість бібліотек для створення GUI на Python підтримують різні операційні системи, що дозволяє створювати кросплатформені додатки;
- мова Python має великий набір сторонніх бібліотек і модулів, що полегшує додавання функціональності до додатків, таких як робота з мережею, обробка даних, інтеграція з базами даних і багато іншого;
- завдяки своїй синтаксичній простоті та великим можливостям, Python дозволяє швидко розробляти та тестувати настільні додатки;
- мова Python має одну з найбільших та найактивніших спільнот розробників, що забезпечує підтримку, документацію та численні ресурси для навчання [16].

Недоліки Python:

- мова Python є інтерпретованою мовою, що може впливати на швидкість виконання додатків у порівнянні з компільованими мовами, такими як C++ або Java;
- створення інсталяційних пакетів для настільних додатків на Python може бути складнішим завданням порівняно з деякими іншими мовами програмування;
- додатки на Python можуть залежати від багатьох зовнішніх бібліотек, що ускладнює процес налаштування середовища для кінцевих користувачів;
- хоча Python підтримує кілька бібліотек для створення GUI, багато з них можуть бути менш функціональними або більш складними у використанні в порівнянні з нативними фреймворками інших мов [16].

#### 2.1.4 Swift

Swift – це мова програмування, розроблена компанією Apple, яка була представлена в 2014 році. Swift спочатку був створений для розробки додатків під iOS і macOS, але з часом здобув популярність і в інших сферах. Використання Swift для створення настільних додатків на macOS дозволяє скористатися всіма перевагами екосистеми Apple, включаючи сучасні фреймворки для розробки графічних інтерфейсів, такі як SwiftUI та AppKit [17].

Переваги:

- мова Swift компілюється в машинний код, що забезпечує високу продуктивність додатків. Це робить його порівняним з іншими високопродуктивними мовами, такими як C++;
- мова Swift має багато вбудованих функцій для безпечного кодування, таких як управління пам'яттю і захист від переповнення буфера;

- мова Swift поєднує в собі потужність і простоту, пропонуючи сучасний синтаксис, який легше читати і писати в порівнянні з Objective-C;
- новий фреймворк для створення UI, який дозволяє швидко і просто створювати інтерфейси за допомогою декларативного синтаксису. SwiftUI значно спрощує процес розробки UI для macOS;
- мова Swift забезпечує легку інтеграцію з іншими продуктами та сервісами Apple, такими як iCloud, Core Data та інші [18].

Недоліки:

- мова Swift переважно використовується для розробки під iOS та macOS. Хоча існують проекти для підтримки Swift на інших платформах, основний фокус залишається на екосистемі Apple;
- у порівнянні з іншими мовами, такими як JavaScript або Python, Swift має менше варіантів для кросплатформної розробки;
- хоча Swift швидко розвивається, він все ще відносно молодий у порівнянні з іншими мовами програмування. Це може призвести до меншої кількості доступних бібліотек та інструментів;
- оскільки Swift активно розвивається, нові версії можуть містити зміни, які не завжди зворотно сумісні, що може вимагати оновлення коду [19].

### 2.1.5 Java

Java – це об'єктно-орієнтована мова програмування загального призначення та обчислювальна платформа. Вона була спочатку створена для вбудованих мережових програм, що працюють на різних платформах. Java призначена для розробки високопродуктивних, портативних програм для різноманітних обчислювальних платформ. Це дозволяє реалізувати основні принципи доступності та міжплатформної взаємодії. Програми, написані на Java, працюють однаково на будь-якому комп'ютері, незалежно від його апаратного забезпечення або операційної системи, за умови наявності

інтерпретатора Java. Окрім портативності, однією з основних переваг Java є набір функцій безпеки, які захищають комп'ютер від проблем, викликаних помилками в коді або шкідливими програмами [20].

Переваги:

- мова Java дозволяє кільком комп'ютерам в мережі працювати спільно над завданням;
- мова Java більш практична та зручна для програмування;
- програми на Java можуть працювати на різних платформах без необхідності повторної компіляції коду;
- мова Java не використовує явні покажчики та має менеджер безпеки, який визначає доступ до класів;
- мова Java підтримує середовище, де великі задачі можуть бути розбиті на окремі потоки, які виконуються паралельно [20].

Недоліки:

- мова Java виконується поверх віртуальної машини, що робить її менш продуктивною та більш ресурсомісткою;
- мова Java забезпечує автоматичне складання сміття, але програмісти не можуть керувати цим процесом вручну, оскільки Java не надає таких методів, як `delete()` та `free()` для звільнення пам'яті [20].

## **2.2 Порівняння мов програмування**

Проаналізувавши роботи [13]–[20] було проведено порівняльний аналіз мов програмування, який наведено у табл. 2.1.

Таблиця 2.1 – Порівняльна характеристика мов програмування

| Критерії порівняння              | R       | C#   | Python                             | Swift   | Java |
|----------------------------------|---------|------|------------------------------------|---------|------|
| Ліцензія                         | GNU GPL | RAND | Python Software Foundation License | GNU GPL | MIT  |
| Простота використання            | –       | +    | +                                  | +       | –    |
| Підтримка кросплатформеності     | +       | +    | +                                  | –       | +    |
| Продуктивність                   | +       | +    | +                                  | +       | +    |
| Розвиток та підтримка            | +       | +    | +                                  | +       | +    |
| Візуальні інтерфейси (GUI)       | +       | +    | –                                  | +       | +    |
| Підтримка великих проєктів       | –       | +    | –                                  | –       | +    |
| Спільнота та ресурси             | +       | +    | +                                  | +       | +    |
| Інтеграція з іншими технологіями | +       | +    | +                                  | –       | +    |

Як видно з таблиці 2.1, мова програмування C# має найбільші переваги. Вона легко адаптується і може ефективно працювати в будь-якому середовищі. З невеликими змінами C# може функціонувати на будь-якій операційній системі та інтегруватися з іншими мовами програмування. Ця мова має багатий набір бібліотек та інструментів, що спрощує виконання завдань з аналізу даних. Крім того, C# має велику спільноту, де інженери та фахівці можуть ставити запитання та отримувати відповіді. Отже, для розробки програмного забезпечення була обрана мова програмування C#.

## 2.3 Вибір середовища розробки

Середовище розробки дозволяє писати, тестувати та налагоджувати код більш простим способом, оскільки воно зазвичай надає автозавершення коду або розуміння коду шляхом виділення, керування ресурсами, інструментів налагодження. Середовище розробки заощаджує час при написанні додатків і дозволяє виправляти поширені помилки у коді, налагоджувати програми та розробляти великі проєкти. Найчастіше

використовувані середовища розробки для створення ПП є Visual Studio, Visual Studio Code, JetBrains Rider, MonoDevelop (Xamarin Studio) [21].

### 2.3.1 Visual Studio

Visual Studio – це інтегроване середовище розробки (IDE), створене Microsoft для операційних систем Windows та macOS. Воно забезпечує підтримку різних мов програмування та платформ, таких як .NET, C++, Python, JavaScript, і багато інших. Visual Studio інтегрується з системами контролю версій, такими як Git, TFS та SVN, що дозволяє користувачам легко керувати своїми проектами, відстежувати зміни та працювати з репозиторіями. Visual Studio пропонує потужну функцію IntelliSense, яка забезпечує автодоповнення коду, підсвічування синтаксису, рефакторинг та інші розширені можливості для покращення ефективності кодування. Крім того, середовище підтримує створення, налагодження та тестування додатків, включаючи можливість виконання модульних тестів [22].

Переваги:

- потужні інструменти для розробки, налагодження та тестування додатків;
- підтримка широкого спектра мов програмування та платформ;
- інтеграція з системами контролю версій, такими як Git, TFS та SVN;
- розширені можливості IntelliSense для автодоповнення, підсвічування синтаксису та рефакторингу коду;
- інтеграція з Azure, що спрощує розгортання та управління хмарними додатками [22].

Недоліки:

- може бути ресурсомістким, що впливає на продуктивність при роботі з великими проектами;

- обмежена кросплатформна підтримка порівняно з іншими середовищами розробки, такими як Visual Studio Code;
- висока вартість ліцензій для професійних та корпоративних версій [22].

### 2.3.2 Visual Studio Code

Visual Studio Code – це безкоштовне середовище розробки з відкритим вихідним кодом, створене Microsoft для роботи на операційних системах Windows, Linux та macOS. Воно інтегрується з локальними та віддаленими репозиторіями Git, включаючи GitHub та Azure Repos, дозволяючи користувачам клонувати репозиторії, фіксувати зміни та керувати гілками. Visual Studio Code має функцію IntelliSense, яка надає доповнення коду на основі типів змінних, функцій та імпортованих модулів, а також забезпечує підсвічування синтаксису та автозаповнення. Крім Python, середовище підтримує багато інших мов програмування, таких як C, C#, JavaScript, HTML, CSS та інші, що є корисним для експериментування з новими API або бібліотеками. Visual Studio Code також підтримує виконання модульних тестів через фреймворки unittest або pytest [23].

Переваги:

- просте управління життєвим циклом проектів у сфері науки про дані;
- широкі можливості налаштування;
- наявність IntelliSense, що допомагає визначати незавершені фрагменти коду;
- легке створення прототипів окремих компонентів з подальшим їх об'єднанням у більші проекти;
- кросплатформна підтримка для Windows, Linux та macOS [23].

Недоліки:

- обмежена функціональність інтерфейсу користувача порівняно з Visual Studio;
- можливі збої при встановленні плагінів;
- відсутність вбудованих функцій для перегляду SVG-файлів [23].

### 2.3.3 JetBrains Rider

JetBrains Rider – це комерційне середовище розробки, створене компанією JetBrains для роботи на операційних системах Windows, Linux та macOS. Воно інтегрується з локальними та віддаленими репозиторіями Git, включаючи GitHub та Bitbucket, дозволяючи користувачам клонувати репозиторії, фіксувати зміни та керувати гілками. JetBrains Rider використовує основу IntelliJ IDEA, забезпечуючи потужну функцію IntelliSense, яка надає автодоповнення коду, підсвічування синтаксису та рефакторинг. Окрім C#, середовище підтримує багато інших мов програмування, таких як Java, JavaScript, TypeScript, HTML, CSS та інші, що є корисним для експериментування з новими API або бібліотеками. JetBrains Rider також підтримує виконання модульних тестів через фреймворки NUnit, xUnit та MSTest [24].

Переваги:

- професійні інструменти для розробки, налагодження та тестування додатків;
- широка підтримка мов програмування та платформ;
- інтеграція з системами контролю версій, такими як Git, GitHub та Bitbucket;
- потужна IntelliJ IDEA-based IntelliSense для автодоповнення, підсвічування синтаксису та рефакторингу коду;
- кросплатформна підтримка для Windows, Linux та macOS [24].

Недоліки:

- висока вартість ліцензії порівняно з безкоштовними альтернативами;
- може бути ресурсомістким, що впливає на продуктивність при роботі з великими проектами;
- складність освоєння для новачків через велику кількість налаштувань та функцій [24].

#### **2.3.4 MonoDevelop (Xamarin Studio)**

MonoDevelop (також відомий як Xamarin Studio) – це безкоштовне середовище розробки з відкритим вихідним кодом, створене для роботи на операційних системах Windows, Linux та macOS. Воно інтегрується з локальними та віддаленими репозиторіями Git, дозволяючи користувачам клонувати репозиторії, фіксувати зміни та керувати гілками. MonoDevelop підтримує функцію IntelliSense, яка надає автодоповнення коду, підсвічування синтаксису та інтелектуальний рефакторинг. Окрім C#, середовище підтримує інші мови програмування, такі як F# та Visual Basic .NET, що є корисним для розробки мультиплатформних додатків. MonoDevelop також підтримує виконання модульних тестів через різні фреймворки, що полегшує тестування та налагодження програм [25].

Переваги:

- просте управління життєвим циклом проектів;
- широкі можливості налаштування;
- підтримка IntelliSense для автодоповнення та рефакторингу коду;
- легке створення прототипів окремих компонентів та їх подальше об'єднання у більші проекти;
- кросплатформна підтримка для Windows, Linux та macOS [25].

Недоліки:

- обмежена функціональність інтерфейсу користувача порівняно з Visual Studio;
- може бути менш стабільним порівняно з іншими середовищами розробки;
- обмежена підтримка деяких мов програмування у порівнянні з іншими IDE [25].

## 2.4 Порівняння середовищ розробки

Проаналізувавши роботи [21]–[25] було проведено порівняльних аналіз середовищ розробки, який наведений у табл. 2.2.

Таблиця 2.2 – Порівняльна характеристика середовищ розробки

| Критерії порівняння                           | Visual Studio | Visual Studio Code | JetBrains Rider | Xamarin Studio |
|-----------------------------------------------|---------------|--------------------|-----------------|----------------|
| Ліцензія                                      | MIT           | MIT                | –               | MIT            |
| Кросплатформність                             | +             | +                  | +               | –              |
| Автозавершення коду                           | +             | +                  | +               | +              |
| Підтримка репозиторіїв                        | +             | +                  | +               | +              |
| Простота встановлення                         | +             | –                  | +               | –              |
| Перевірка та аналіз коду                      | +             | +                  | +               | +              |
| Швидкий та безпечний рефакторинг коду         | +             | +                  | –               | +              |
| Наявність інтерактивної консолі               | +             | +                  | +               | +              |
| Наявність зручного користувацького інтерфейсу | +             | +                  | +               | +              |
| Швидкий та зручний пошук                      | +             | +                  | –               | –              |
| Підсвічування синтаксису                      | +             | +                  | +               | +              |
| Підтримка наукових пакетів                    | +             | +                  | +               | –              |
| Наявність функції статичного аналізу коду     | +             | +                  | +               | +              |

З таблиці 2.2 видно, що середовище розробки Visual Studio має найбільші переваги. Воно вирізняється простотою освоєння, кросплатформністю та спеціалізацією на створенні багатифункціональних

додатків. Visual Studio включає такі можливості, як текстовий редактор з підсвічуванням синтаксису, автозавершенням коду та інспекцією змінних, значення яких можна редагувати через графічний інтерфейс. Тому, для розробки програмного забезпечення було обрано саме Visual Studio.

## 2.5 Структура програмного продукту

Розроблюваний ПП призначений для моніторингу ТП ПХ та ТД зберігання ПХ. Програмний продукт представлений у вигляді окремого віконного додатку завдяки якому можна легко контролювати всі важливі показники. Завдяки розташуванню всього наданого функціоналу на одному вікні програми користувач може одразу з легкістю виконувати всі основні операції з лише 1 вікном, що не перешкоджає роботі запущених вікон інших програм.

Розроблюваний ПП використовує дані з термодатчиків, які за задумом стоять в ХУ. Застосунок порівнює отримані дані з наявними в його базі даних значень ТД, а саме з мінімальною та максимальною температурами зберігання конкретного ПХ. Стосовно ТП, ПП бере з бази даних, де зберігається список ПХ, які зберігаються у всіх холодильниках, кінцевий ТП кожного ПХ та порівнює з актуальною датою.

Модуль MainForm.cs – основний модуль, відповідає за ініціалізацію програмного продукту та елементів користувацького інтерфейсу.

Опис структури модулю:

- temperature\_monitoring – метод, призначений для моніторингу температури кожного холодильника в певний проміжок часу;
- expiration\_date\_control – призначений для моніторингу продуктів, ТП яких закінчується через n днів;
- control\_of\_spoiled\_goods – призначений для моніторингу продуктів, термін придатності яких закінчився на момент перевірки;

- `add_in_fridge` – метод додає до обраного холодильника продукт із вказаними параметрами цього продукту. Додавання відбувається шляхом створення нового запису в базі даних продуктів, які зберігаються;
- `delete_from_fridge` – виймає продукт з обраного холодильника. Виймання відбувається шляхом видалення обраного запису в базі даних продуктів, які зберігаються;
- `product_search` – метод здійснює пошук продукту з бази даних з продуктами, що зберігаються шляхом пошуку запису за унікальним параметром “id”;
- `product_movement` – здійснює переміщення продукту з одного холодильника в інший шляхом пошуку продукту за унікальним параметром “id” в базі даних продуктів, які зберігаються, та зміною значення “fridge” на обраний номер холодильника;
- `setting_frequency_for_temperature` – метод встановлює частоту, з якою ППП буде запускати метод `temperature_monitoring`;
- `setting_the_days_until_the_end_date` – встановлює мінімальну кількість днів, яка може бути між актуальною датою та датою кінцевого ТП.

## 2.6 Висновок

У цьому розділі було проведено аналіз та вибір інструментарію розробки ППП та було спроектовано архітектуру майбутнього ППП. Було проведено порівняльний аналіз мов програмування для розробки ППП та встановлено, що найбільші переваги має мова програмування C#, тому що вона легко адаптується, має багатий набір бібліотек та інструментів, що спрощує виконання завдань з розробки застосунку.

Було проведено порівняльний аналіз середовищ розробки, в ході чого було встановлено, що найбільші переваги має середовище розробки Visual

Studio, тому що воно має такі переваги як простота в освоєнні, кросплатформне, має великий набір бібліотеки для розробки застосунку.

### 3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

Завданням роботи є програмна реалізація методів моніторингу температури зберігання продуктів в холодильниках, моніторингу ТП ПХ та робота з вмістом холодильників.

У роботі розглядається проблема ефективного зберігання ПХ, а саме вчасне реагування на зміну температури зберігання ПХ, вчасне виявлення продуктів у яких ТП підходить до закінчення, або продуктів з прострочений ТП.

Для досягнення поставленої мети необхідно розробити застосунок, який дозволяє одночасно працювати з вмістом холодильників, моніторити температуру в холодильниках та моніторити ТП ПХ.

#### 3.1 Основні відомі методи

Для моніторингу температури в холодильниках використовуються значення, отримані з бази даних(імітація роботи справжніх термодатчиків), і відбувається порівняння отриманого значення температури холодильника з мінімальним та максимальним значенням температури зберігання кожного продукту.

Порівняння відбувається за відношенням (3.1):

$$T_{min} \leq T_{fridge} \leq T_{max} \quad (3.1)$$

де  $T_{min}$  – мінімальна температура зберігання продукту;

$T_{fridge}$  – температура в холодильнику;

$T_{max}$  – максимальна температура зберігання продукту.

Для визначення, чи продукт зіпсований використовується значення актуальної дати та кінцевої дати ТП продукту та порівнюються між собою за відношенням (3.2):

$$D_{today} \geq D_{last}, \quad (3.2)$$

де  $D_{today}$  – актуальна дата;

$D_{last}$  – кінцева дата ТП.

Визначення, чи закінчується термін придатності продукту відбувається при порівнянні різниці кінцевої дати ТП продукту та актуальної дати з заданим мінімальним числом кількості днів проводиться за відношенням (3.3):

$$(D_{last} - D_{today}) \leq N_{days}, \quad (3.3)$$

де  $D_{last}$  – кінцева дата ТП;

$D_{today}$  – актуальна дата;

$N_{days}$  – кількість днів до кінцевого ТП.

Для виймання продукту з холодильника реалізовано запит, який визначає запис за параметром  $id$  та видаляє запис за запитом(3.4):

$$DELETE FROM products WHERE id = ID, \quad (3.4)$$

де  $ID$  – унікальний номер присвоєний кожному продукту.

Для пошуку продукту реалізовано запит, який визначає запис за параметром *id* за запитом(3.5).

$$SELECT * FROM products WHERE id = ID, \quad (3.5)$$

де *ID* – унікальний номер присвоєний кожному продукту.

Переміщення продуктів між холодильниками реалізовано запитом, який визначає запис за параметром *id*, та в параметр *fridge\_№* записуємо номер обраного холодильника за запитом(3.6).

$$UPDATE products SET id = Nfr WHERE id = ID, \quad (3.6)$$

де *Nfr* – номер холодильника;

*ID* – унікальний номер присвоєний кожному продукту.

### 3.2 Висновок

У цьому розділі було створено та проаналізовано методи моніторингу температури в холодильниках, визначення протермінованих продуктів та визначення продуктів, ТП як підходить до кінця. Розроблено методи роботи з вмістом холодильника, а саме додавання товару, виймання товару, пошук товару та переміщення товару між холодильниками.

## **4 КЕРІВНИЦТВО ПРОГРАМІСТА**

### **4.1 Призначення й умови використання**

Розроблюваний ПП призначений для моніторингу температури зберігання ПХ в ХУ. Програмний продукт представлений у вигляді окремого віконного додатку завдяки якому можна переглядати температуру в холодильниках, моніторити продукти, які вже протерміновані, або у яких ТП закінчується. Також можна керувати розміщенням продуктів в холодильнику, а саме додавати, забирати, пошук та переміщення між холодильниками продукту.

Для коректної роботи ПП необхідні такі мінімальні системні вимоги:

- платформа x64-based;
- операційна система Microsoft Windows 11/10/8/7 (рекомендується встановити всі останні оновлення);
- процесор з мінімальною тактовою частотою 1,8 ГГц, рекомендується використовувати як мінімум двоядерний процесор;
- оперативна пам'ять від 2 ГБ (рекомендується 4 ГБ та мінімум 2,5 ГБ при використанні на віртуальній машині);
- роздільна здатність екрану 1920x1080 пікселів з 16-бітовим кольором;
- мінімум 2 ГБ вільного місця на жорсткому диску.

### **4.2 Звертання до програмного продукту (файли проєкту)**

Розроблюваний ПП складається з таких модулів:

Модуль MainForm.cs – основний модуль, відповідає за ініціалізацію програмного продукту та елементів користувацького інтерфейсу.

Опис структури модулю:

- `temperature_monitoring` – метод, призначений для моніторингу температури кожного холодильника в певний проміжок часу;
- `expiration_date_control` – призначений для моніторингу продуктів, ТП яких закінчується через  $n$  днів;
- `control_of_spoiled_goods` – призначений для моніторингу продуктів, термін придатності яких закінчився на момент перевірки;
- `add_in_fridge` – метод додає до обраного холодильника продукт із вказаними параметрами цього продукту. Додавання відбувається шляхом створення нового запису в базі даних продуктів, які зберігаються;
- `delete_from_fridge` – виймає продукт з обраного холодильника. Виймання відбувається шляхом видалення обраного запису в базі даних продуктів, які зберігаються;
- `product_search` – метод здійснює пошук продукту з бази даних з продуктами, що зберігаються шляхом пошуку запису за унікальним параметром “id”;
- `product_movement` – здійснює переміщення продукту з одного холодильника в інший шляхом пошуку продукту за унікальним параметром “id” в базі даних продуктів, які зберігаються, та зміною значення “fridge” на обраний номер холодильника;
- `setting_frequency_for_temperature` – метод встановлює частоту, з якою ППП буде запускати метод `temperature_monitoring`;
- `setting_the_days_until_the_end_date` – встановлює мінімальну кількість днів, яка може бути між актуальною датою та датою кінцевого ТП.

### 4.3 Вхідні і вихідні дані

В якості вхідних даних, ППП приймає дані про налаштування, холодильники та продукти, які зберігаються в холодильниках. Ці дані

представляють собою таблиці створені на локальному сервері до яких звертається застосунок за допомогою сформованих запитів мовою SQL.

Застосунок отримує як вхідні дані:

- temp\_fridge – температура в холодильнику (°C);
- temp\_min – мінімальна температура зберігання продукту (°C);
- temp\_max – максимальна температура зберігання продукту (°C);
- current\_date – поточна дата (dd.mm.yyyy);
- end\_date – дата закінчення ТП продукту (dd.mm.yyyy);
- n\_days – кількість днів до кінцевого ТП (доба);
- temp\_frequency – частота перевірки температури зберігання продуктів в холодильниках (хв).

На рис. 4.1 – 4.3 представлено приклад вхідних даних для моніторингу застосунком температури і ТП.

| id | temp_frequency | n_days |
|----|----------------|--------|
| 1  | 5              | 14     |

Рисунок 4.1 – Вхідні дані для моніторингу застосунком температури і ТП з таблиці “settings”

| id | temperature |
|----|-------------|
| 1  | 0           |
| 2  | 2           |
| 3  | -1          |

Рисунок 4.2 – Вхідні дані для моніторингу застосунком температури і ТП з таблиці “fridge”

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-15 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |

Рисунок 4.3 – Вхідні дані для моніторингу застосунком температури і ТП з таблиці “products”

Вихідними даними у разі виявлення невідповідності умові в термінах придатності є список продуктів, які не пройшли перевірку.

Для здійснення додавання, виймання, пошуку та переміщення між холодильниками продуктів ПП отримує дані з таблиці продуктів.

Застосунок отримує як вхідні дані:

- fridge\_No – номер холодильника;
- name – назва продукту;
- type – тип продукту;
- date\_of\_manufacture – дата виготовлення продукту;
- date\_last – кінцевий ТП продукту;
- Tmin – мінімальна температура зберігання продукту;

– Tmax – максимальна температура зберігання продукту.

На рис. 4.4 представлено приклад вхідних даних для додавання, виймання, пошуку та переміщення між холодильниками продуктів.

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-15 | 1    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |

Рисунок 4.4 – Вхідні дані для додавання, виймання, пошуку та переміщення між холодильниками продуктів з таблиці “products”

Вихідними даними є результати роботи з вмістом холодильників, а саме записи в базі даних продуктів які зберігаються як результат таких методів як додавання, виймання та переміщення між холодильниками продукту.

4.4 Повідомлення

Під час роботи створений ПП виводить текстові повідомлення, які описані у табл. 4.1.

Таблиця 4.1 – Повідомлення під час роботи «Temperature and shelf life control»

| Повідомлення                                                                                                                                                                       | Опис                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| Невірні дані!                                                                                                                                                                      | Невірно введені дані.                                                                        |
| Зіпсованих товарів не знайдено.                                                                                                                                                    | Повідомлення з’являється, при успішному проходженні перевірки на кінцевий ТП.                |
| Товарів, ТП яких закінчується не знайдено.                                                                                                                                         | Повідомлення з’являється, при успішному проходженні перевірки товарів, ТП яких закінчується. |
| Продукт успішно додано.                                                                                                                                                            | Повідомлення появляться при успішному виконанні операції додавання.                          |
| Продукт успішно вийнято.                                                                                                                                                           | Повідомлення появляться при успішному виконанні операції виймання.                           |
| Продукт успішно переміщено.                                                                                                                                                        | Повідомлення означає, що завантажені дані користувачем не є коректними.                      |
| Холодильник: 2<br>Назва: Масло<br>Вид: Молочні продукти<br>Дата виготовлення: 2024-05-15<br>Кінцева дата :2024-06-20<br>(в текстовому полі “Скоро закінчиться термін придатності”) | Виявлено продукти, у яких ТП скоро закінчується.                                             |

Продовження таблиці 4.1

| Повідомлення                                                                                                                                                       | Опис                                    |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------|
| Холодильник: 1<br>Назва: Яблуко<br>Вид: Фрукти<br>Дата виготовлення: 2024-05-01<br>Кінцева дата :2024-06-01<br>(в текстовому полі “Термін придатності закінчився”) | Виявлено продукт, у яких ТП закінчився. |

4.5 Структура бази даних

ПП для роботи використовує 3 таблиці. З них 2 між собою пов’язані спільним ключем.

Таблиця “settings” призначена для встановлення налаштувань в програму. Складається з 3 колонок: id, temp\_frequency, n\_days.

Опис колонок таблиці:

- id – індекс, нумерує рядки таблиці;
- temp\_frequency – частота моніторингу температури холодильників;
- n\_days – кількість днів до кінцевого ТП.

Типи колонок наведено в таблиці 4.2.

Таблиця 4.2 – Типи колонок в таблиці “settings”

| Назва колонки  | Тип колонки |
|----------------|-------------|
| id             | Число       |
| temp_frequency | Число       |
| n_days         | Число       |

Таблиця “fridge” призначена для зберігання номерів холодильників та температури всередині них. Складається з 2 колонок: id, temperature.

Опис колонок таблиці:

- id – індекс, нумерує рядки таблиці та відповідає номеру холодильника. Це поле є ключем для зв’язку з таблицею “products”;
- temperature – температура холодильника.

Типи колонок наведено в таблиці 4.3.

Таблиця 4.3 – Типи колонок в таблиці “fridge”

| Назва колонки | Тип колонки |
|---------------|-------------|
| id            | Число       |
| temperature   | Число       |

Таблиця “products” призначена для зберігання даних про товари, які зберігаються в холодильниках. Складається з 8 колонок: id, fridge\_№, name, type, date\_of\_manufacture, date\_last, Tmin, Tmax.

Опис колонок таблиці:

- id – індекс, нумерує рядки таблиці та є унікальним номером для продукту;
- fridge\_№ – номер холодильника, в якому зберігається продукт. Це поле є ключем для зв’язку з таблицею “fridge”;
- name – назва продукту;
- type – тип продукту;
- date\_of\_manufacture – дата виготовлення продукту;
- date\_last – кінцевий ТП продукту;
- Tmin – мінімальна температура зберігання продукту;
- Tmax – максимальна температура зберігання продукту.

Типи колонок наведено в таблиці 4.4.

Таблиця 4.4 – Типи колонок в таблиці “fridge”

| Назва колонки       | Тип колонки |
|---------------------|-------------|
| id                  | Число       |
| temperature         | Число       |
| name                | Рядок       |
| type                | Рядок       |
| date_of_manufacture | Дата        |
| date_last           | Дата        |
| Tmin                | Число       |
| Tmax                | Число       |

Зв’язок між таблицями “fridge” та “products” наведено на рисунку 4.5.

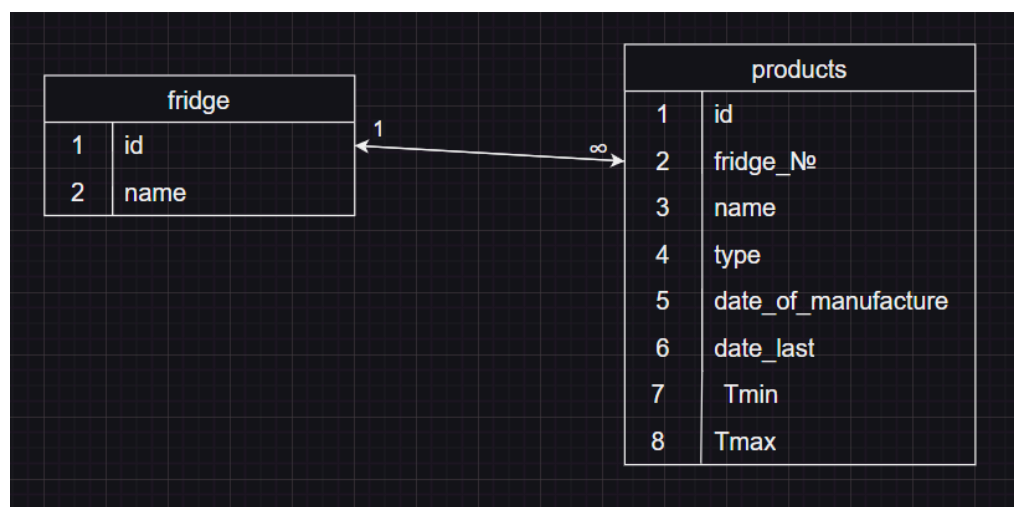


Рисунок 4.5 – Зв’язок між таблицями “fridge” та “products”

#### 4.6 Повідомлення оператора

Під час роботи оператора з розробленим ПП виводяться різні текстові повідомлення, вони інформують оператора про помилки або його неправильні дії, а також про успішність виконання різних маніпуляцій.

Список всіх текстових повідомлень, описання їх змісту та дії оператора, які він повинен виконати при отриманні цих повідомлень представлені у розділі “Керівництво програміста” в таблиці 4.1.

#### **4.7 Висновок**

У цьому розділі було представлено призначення та функції розробленого програмного продукту ПП, а також мінімальні системні вимоги для його коректної роботи. Докладно описані модулі, з яких складається ПП. Створений ПП має відкритий вихідний код, що дозволяє іншим розробникам переглядати, вивчати та змінювати його. Також у цьому розділі міститься опис вхідних і вихідних даних, а також список усіх текстових повідомлень, які можуть з'являтися під час роботи користувача з ПП. Описано структуру бази даних та її таблиць.

## **5 КЕРІВНИЦТВО ОПЕРАТОРА**

### **5.1 Призначення й умови виконання програмного продукту**

Розроблюваний програмний продукт призначений для моніторингу температури зберігання ПХ в ХУ. Програмний продукт представлений у вигляді окремого віконного додатку (Windows-застосунок) завдяки якому можна переглядати температуру в холодильниках, моніторити продукти, які вже протерміновані, або у яких ТП закінчується. Також можна керувати розміщенням продуктів в холодильнику, а саме додавати, забирати, пошук та переміщення між холодильниками продукту.

Для коректної роботи ПП необхідні такі мінімальні системні вимоги:

- платформа x64-based;
- операційна система Microsoft Windows 11/10/8/7 (рекомендується встановити всі останні оновлення);
- процесор з мінімальною тактовою частотою 1,8 ГГц, рекомендується використовувати як мінімум двоядерний процесор;
- оперативна пам'ять від 2 ГБ (рекомендується 4 ГБ та мінімум 2,5 ГБ при використанні на віртуальній машині);
- роздільна здатність екрану 1920x1080 пікселів з 16-бітовим кольором;
- мінімум 2 ГБ вільного місця на жорсткому диску.

### **5.2 Виконання програмного продукту**

Після вдалого запуску ПП відкривається головне вікно, яке представлено на рис. 5.1.

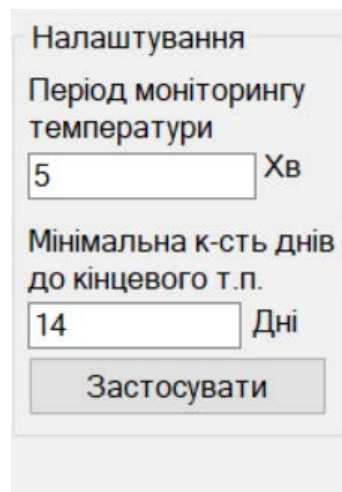
Рисунок 5.1 – Головне вікно ПП

На рисунку 5.2 зображено блок “Температура в холодильниках”. В цьому блоці оператор може переглядати поточну температуру всіх холодильників. ПП виконує перевірку в заданий в блоці “Налаштування” в полі “Період моніторингу температури”(рис. 5.3).

Рисунок 5.2 – Блок застосунку “Температура в холодильниках”

На рисунку 5.3 зображено блок “Температура”. В цьому блоці оператор може встановити такі параметри як “Період моніторингу температури” та “Мінімальна кількість днів до кінцевого терміну

придатності”. Значення поля “Період моніторингу температури” означає період, з яким ПП буде перевіряти, чи немає відхилень у температурному режимі зберігання ПХ холодильнику(в даному випадку 5 хв.). Значення поля “Мінімальна кількість днів до кінцевого терміну придатності” означає мінімальну різницю між кінцевою датою придатності та поточною датою(в даному випадку 14 днів).



Налаштування

Період моніторингу температури

5 Хв

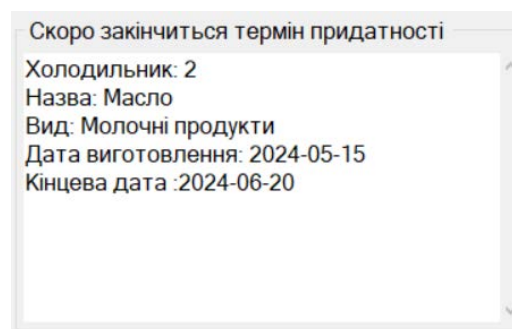
Мінімальна к-сть днів до кінцевого т.п.

14 Дні

Застосувати

Рисунок 5.3 – Блок застосунку “Налаштування”

На рисунку 5.4 зображено блок “Скоро закінчиться термін придатності”. В цьому блоці оператор може переглянути продукти, у яких різниця між кінцевою датою придатності та поточною датою менша зазначеного терміну(рис. 5.3).



Скоро закінчиться термін придатності

Холодильник: 2

Назва: Масло

Вид: Молочні продукти

Дата виготовлення: 2024-05-15

Кінцева дата :2024-06-20

Рисунок 5.4 – Блок застосунку “Скоро закінчиться термін придатності”

На рисунку 5.5 зображено блок “Термін придатності закінчився”. В цьому блоці оператор може переглянути продукти, у яких ТП вже закінчився(поточна дата більша за кінцеву дату придатності продукту).

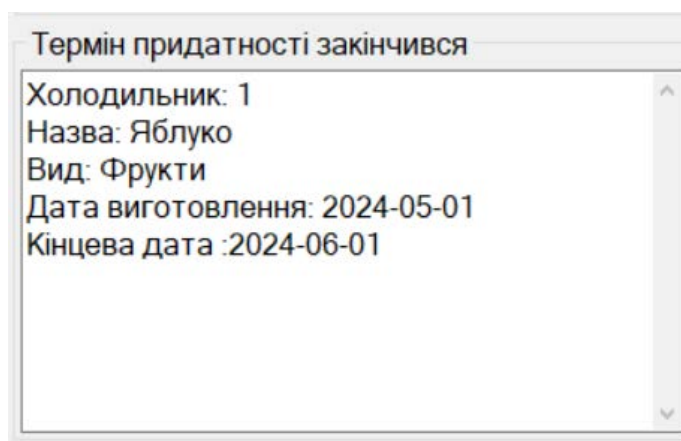


Рисунок 5.5 – Блок застосування “Термін придатності закінчився”

На рисунку 5.6 зображено блок “Додавання продуктів в холодильник”. В цьому блоці оператор може додати продукт у відповідний холодильник. Блок складається з таких полів:

- № холодильника – номер холодильника, в який поміщають продукт;
- назва продукту – повна назва продукту, який поміщають в холодильник;
- вид продукту – вид продукту(наприклад молочні вироби, м’ясні вироби, фрукти і т.д.), який поміщають в холодильник;
- дата виготовлення – дата виготовлення ПХ, який поміщають в холодильник;
- кінцева дата – кінцева дата ТП ПХ, який поміщають в холодильник;
- $\min t^{\circ}$  зберігання – мінімальна температура зберігання ПХ, який поміщають в холодильник;

–  $\max t^{\circ}$  зберігання – мінімальна температура зберігання ПХ, який поміщають в холодильник.

Після заповнення всіх полів треба натиснути кнопку “Додати продукт”, для додавання запису в базу даних з продуктами. Тільки тоді продукт вважається доданим в холодильник.

Додавання продуктів в холодильник

№ холодильника

|                   |                            |
|-------------------|----------------------------|
| Назва продукту    | Кінцева дата               |
| Вид продукту      | Min $t^{\circ}$ зберігання |
| Дата виготовлення | Max $t^{\circ}$ зберігання |

Додати продукт

Рисунок 5.6 – Блок застосунку “Додавання продуктів в холодильник”

На рисунку 5.7 зображено блок “Переміщення продукту між холодильниками”. В цьому блоці оператор може перемістити продукт з одного холодильника в інший. Блок складається з таких полів:

- з холодильника № – номер холодильника, з якого переміщують продукт;
- перенести – індивідуальний номер продукту, який переміщують в інший холодильник;
- в холодильник № – номер холодильника, в який переміщують продукт.

Після заповнення всіх полів треба натиснути кнопку “Перенести”, для переміщення продукту. Тільки тоді продукт вважається переміщеним.

Рисунок 5.7 – Блок застосунку “Переміщення продукту між холодильниками”

На рисунку 5.8 зображено блок “Пошук продуктів”. В цьому блоці оператор може зробити пошук продукту за унікальним номером. Блок складається з таких елементів:

- текстове поле “Введіть id продукту” – текстове поле, де треба ввести індивідуальний номер(id) для пошуку продукту;
- порожнє текстове поле – в цьому текстовому полі відображаються результати пошуку продукту.

Після заповнення поля ”Введіть id продукту” треба натиснути кнопку “Пошук”. Після цього, якщо продукт буде знайдено, дані про нього з’являться в текстовому полі нижче.

Рисунок 5.8 – Блок застосунку “Пошук продуктів”

На рисунку 5.9 зображено блок “Вилучення продукту з холодильника”. В цьому блоці оператор може зробити вилучення продукту з

обраного холодильника за унікальним номером. Блок складається з таких елементів:

- текстове поле “Введіть id продукту” – текстове поле, де треба ввести індивідуальний номер(id) для вилучення продукту;
- порожнє текстове поле з випадаючим списком – в цьому текстовому полі необхідно обрати номер холодильника, з якого треба вилучити продукт.

Після заповнення поля ”Введіть id продукту” треба натиснути кнопку “Вилучити”. Після цього, продукт вважається вилученим.

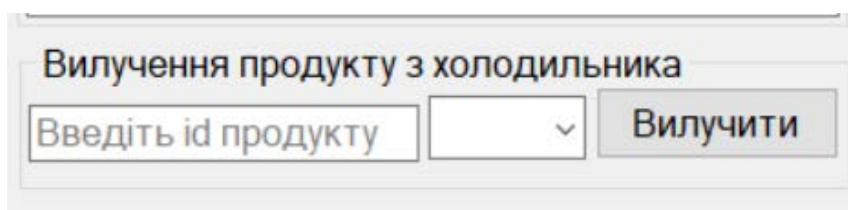


Рисунок 5.9 – Блок застосунку “Вилучення продукту з холодильника”

### 5.3 Висновок

У цьому розділі було представлено призначення та функції розробленого програмного продукту ПП та мінімальні системні вимоги для його коректної роботи. Описано покрокову інструкцію для оператора щодо роботи з ПП. Перелічено всі текстові повідомлення, які можуть з'являтися під час роботи оператора з ПП. Надано детальну інструкцію по роботі оператора з ПП, включаючи приклади додавання, вилучення, пошуку та переміщення між холодильниками. Інтерфейс ПП є інтуїтивно зрозумілим і простим, що не потребує додаткового навчання оператора, а у разі виникнення питань по роботі з ПП, доступна довідка.

6 ЕКСПЕРИМЕНТИ ТА РЕЗУЛЬТАТИ

6.1 Проведення експериментів

Спроектувавши та створивши ПП було проведено випробування функціоналу. Було перевірено та детально описано дії при перевірці працездатності таких методів як додавання, виймання, переміщення та пошук продуктів.

На рисунку 6.1 зображено вигляд бази даних “products” до виконання операції додавання.

| id | fridge_№ | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1        | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2        | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1        | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3        | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3        | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1        | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2        | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3        | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2        | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1        | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2        | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3        | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1        | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1        | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2        | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3        | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2        | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1        | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2        | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3        | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |

Рисунок 6.1 – база даних “products” до виконання операції додавання

На рисунках 6.2 – 6.4 зображено результати додавання продукту до холодильника.

Додавання продуктів в холодильник

№ холодильника

1

HELL

2024-07-05

Енергетик

0

2024-05-20

4

Додати продукт

Рисунок 6.2 – Заповнення полів блоку “Додавання продуктів в холодильник”

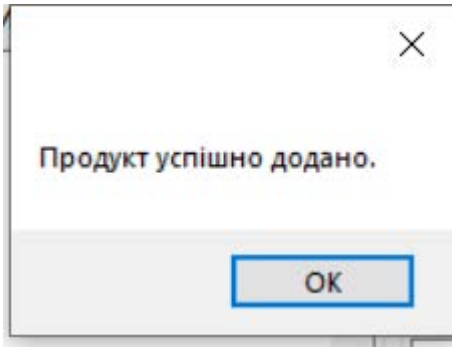


Рисунок 6.3 – Повідомлення про успішне додавання

| id | fridge_Nr | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |
| 21 | 1         | HELL         | Енергетик        | 2024-05-20          | 2024-07-05 | 0    | 4    |

Рисунок 6.4 – База даних “products” після операції додавання

На рисунку 6.5 – 6.8 зображено результати переміщення продукту між холодитильниками.

| id | fridge_Nr | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Брокколі     | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |
| 21 | 1         | HELL         | Енергетик        | 2024-05-20          | 2024-07-05 | 0    | 4    |

Рисунок 6.5 – База даних “products” до операції переміщення

Переміщення продукту між холодитильниками

З холодитильника №

Перенести

В холодитильник №

Рисунок 6.6 – Введення даних в блок “Переміщення продукту між холодитильниками”

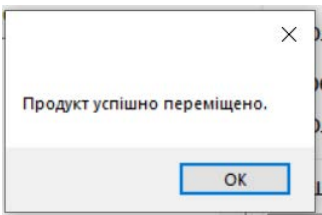


Рисунок 6.7 – Повідомлення про успішне переміщення

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |
| 21 | 3         | HELL         | Енергетик        | 2024-05-20          | 2024-07-05 | 0    | 4    |

Рисунок 6.8 – База даних “products” після операції переміщення

На рисунку 6.9 зображено результати пошуку продукту.

Пошук продуктів

Холодильник: 3  
Назва: HELL  
Вид: Енергетик  
Дата виготовлення: 2024-05-20  
Кінцева дата :2024-07-05  
Min t° зберігання: 0  
Max t° зберігання 4

Рисунок 6.9 – Результати пошуку продукту

На рисунку 6.10 – 6.13 зображено результати виймання продукту з холодильника.

| id | fridge_Nr | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |
| 21 | 3         | HELL         | Енергетик        | 2024-05-20          | 2024-07-05 | 0    | 4    |

Рисунок 6.10 – База даних “products” до операції виймання

Рисунок 6.11 – Введення даних в блок “Вилучення продукту з холодильника”

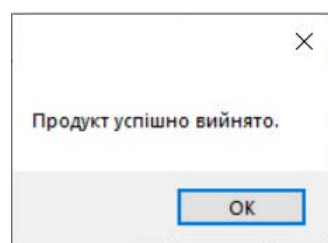


Рисунок 6.12 – Повідомлення про успішне вийняття продукту

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |

Рисунок 6.13 – База даних “products” після операції переміщення

## 6.2 Висновок

У цьому розділі було детально показано етапи використання методів додавання, виймання, переміщення та пошуку продуктів. Результати експерименту цілком відповідають заявленим вимогам.

## ВИСНОВКИ

Під час виконання роботи було проведено аналіз предметної області, визначено основні поняття, пов'язані з предметною областю і об'єктом дослідження та розробки, визначено проблематику поставленого завдання, наведено її актуальність та доцільність вирішення.

Було досліджено основні методи моніторингу температурних діапазонів та ТП ПХ, визначено їх переваги та недоліки, та проведено порівняльний аналіз цих методів.

Наукова новизна роботи полягає у тому, що запропоновано створення застосунку, який в своєму функціоналі об'єднує, автоматизує та спрощує виконання таких завдань, як моніторинг температурних діапазонів зберігання ПХ, моніторинг ТП ПХ, додавання продуктів в холодильник, переміщення продуктів між холодильниками, пошук продуктів та виймання продуктів з холодильника. Це допоможе значно зменшити витрати від псування ПХ та невчасного реагування на надзвичайні ситуації, особливо в період, коли відбуваються часті вимкнення світла.

Розроблюваний програмний продукт призначений для моніторингу температури зберігання ПХ в ХУ. Програмний продукт представлений у вигляді окремого віконного додатку (Windows-застосунок) завдяки якому можна переглядати температуру в холодильниках, моніторити продукти, які вже протерміновані, або у яких ТП закінчується. Також можна керувати розміщенням продуктів в холодильнику, а саме додавати, забирати, пошук та переміщення між холодильниками продукту.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Термін придатності [Електрон. ресурс]. – Режим доступу:  
<https://www.pharmencyclopedia.com.ua/article/2161/termin-pridatnosti>.
2. Термін придатності [Електрон. ресурс]. – Режим доступу:  
<https://ua.nesrakonk.ru/expiration-date/>.
3. Продукти харчування [Електрон. ресурс]. – Режим доступу:  
<https://ua-etymology.livejournal.com/257295.html>.
4. Ліки [Електрон. ресурс]. – Режим доступу:  
<https://www.pharmencyclopedia.com.ua/article/2082/liki>.
5. Основні типи холодильних установок і циклів [Електрон. ресурс].  
– Режим доступу:  
[https://elearning.sumdu.edu.ua/free\\_content/lectured:a431440315da9066760766d0c931a08acfc9ca64/latest/5465/index.html](https://elearning.sumdu.edu.ua/free_content/lectured:a431440315da9066760766d0c931a08acfc9ca64/latest/5465/index.html).
6. Data Collection Methods and Tools for Performance Monitoring –  
[Electronic resource]. – Access mode:  
[https://usaidlearninglab.org/system/files/resource/files/data\\_collection\\_perf\\_monitoring\\_final2021.pdf](https://usaidlearninglab.org/system/files/resource/files/data_collection_perf_monitoring_final2021.pdf).
7. The Showdown: Manual vs. Automated Temperature Monitoring  
[Electronic resource]. – Access mode:  
<https://www.compliancemate.com/blog/showdown-manual-vs-automated-temperature-monitoring>.
8. Choosing the Right Temperature Monitoring Solution: Data Logger vs  
Automated Monitoring System [Electronic resource]. – Access mode:  
<https://safetyculture.com/topics/temperature-data-logger/>.
9. Manual data collection: manufacturing's biggest problem [Electronic  
resource]. – Access mode:  
<https://www.machinemetrics.com/blog/manual-data-collection>.
10. RuuviTag [Electronic resource]. – Access mode:

<https://ruuvi.com/ruuvitag/>.

11. SensGuard [Electronic resource]. – Access mode: <https://sensmax.eu/software/online-temperature-and-humidity-monitoring-software/>.

12. Thermotrack Webserve [Electronic resource]. – Access mode: <https://www.plugandtrack.com/fr/sondes-enregistreurs-temperature-logiciels/surveillance-temperature-distance/>.

13. Найпопулярніші мови програмування у 2024 році [Електрон. ресурс]. – Режим доступу: <https://dan-it.com.ua/uk/blog/samyje-populjarnye-jazyki-programmirovanija-v-2024-godu/>.

14. Кофанов О. Є. Програмування із використанням R у статистичних та маркетингових дослідженнях / О. Є. Кофанов, С. О. Солнцев, О. В. Зозульов. – Київ: КПІ ім. Ігоря Сікорського, 2023. – 204 с.

15. Як вивчити мову програмування C# та стати .NET розробником [Електрон. ресурс]. – Режим доступу: <https://edu.cbsystematics.com/ua/blog/learn-csharp-become-dnet>.

16. Lee, W. Python machine learning / W. Lee. – London : Wiley, 2019. – 320 p.

17. Swift (мова програмування): нова епоха універсального коду та неймовірних шляхів до інновацій [Електрон. ресурс]. – Режим доступу: <https://bdut.co.ua/pro-nas/swift-mova-programuvannya/>.

18. Чому варто програмувати на мові Swift? [Електрон. ресурс]. – Режим доступу: <https://novograd.city/articles/268916/chomu-var-to-programuvati-na-movi-sift>.

19. Нова мова програмування swift: коментарі розробників [Електрон. ресурс]. – Режим доступу: <https://www.cn.ua/news/apple/16717-nova-mova-programyvannia-swift-komentari-rozrobnikiv.html>.

20. Сьерра, К. Изучаем Java. Лекции и упражнения / К. Сьерра, Б. Бэйтс. – М. : Эксмо, 2017. – 720 с.

21. Найкращі IDE для програмування на C# [Електрон. ресурс]. – Режим доступу: <https://deveducation.com/blog/luchshie-ide-dlya-razrabotki-na-c/>.

22. Microsoft Visual Studio Reviews [Electronic resource]. – Access mode:

<https://www.softwareadvice.com/game-development/microsoft-visual-studio-profile/reviews/>.

23. Stetsyuk, V. Method of choosing the programming environment for software / V. Stetsyuk, T. Hovorushchenko // International Scientific Journal Computer Systems and Information Technologies. – 2021. – Vol. 2. – P. 6–10.

24. Rider [Electronic resource]. – Access mode: <https://www.trustradius.com/products/jetbrains-rider/reviews?qs=pros-and-cons#overview>.

25. Xamarin vs. Android Studio – A Comparison Guide with Pros and Cons [Electronic resource]. – Access mode: <https://medium.com/nerd-for-tech/xamarin-vs-android-studio-a-comparison-guide-with-pros-and-cons-dab30cb11008>.

**ДОДАТОК А**  
**Технічне завдання**

## **Вступ**

Розроблюваний програмний продукт використовується для моніторингу ТП ПХ, моніторингу ТД зберігання ПХ та управління складом холодильників. ПП «Temperature and shelf life control» представлений у вигляді окремого віконного додатку завдяки якому можна зручно моніторити ТП ПХ та моніторити температуру в ХУ. Також застосунок дає змогу виконувати такі функції, як додавання продукту в холодильник, виймання продукту з холодильника, пошук продукту та переміщення продукту між холодильниками.

### **A.1 Мета розробки і її призначення**

Об'єкт дослідження – процес моніторингу температури зберігання, термінів придатності харчових продуктів та ліків та їх розміщення в холодильниках.

Предмет дослідження – методи моніторингу температури зберігання, термінів придатності харчових продуктів та ліків та їх розміщення в холодильниках.

Мета роботи – дослідження та програмна реалізація методів моніторингу термінів придатності продуктів харчування та ліків від часу та температури.

Метод дослідження – метод порівняння значень з еталонними, метод моніторингу термінів придатності.

Актуальність роботи полягає в тому, що розробка застосунку для моніторингу температури в холодильниках та термінів придатності продуктів харчування має важливе значення з кількох ключових причин. По-перше, такий застосунок сприяє забезпеченню безпеки харчових продуктів, оскільки контроль за температурним режимом дозволяє уникнути розмноження шкідливих мікроорганізмів і запобігти харчовим отруєнням. Він також

допомагає зберегти якість продуктів, оскільки оптимальні умови зберігання зберігають їх харчову цінність і смак.

По-друге, застосунок дозволяє відповідати санітарним нормам і стандартам, що є критично важливим для підприємств харчової промисловості. Він спрощує процес дотримання регуляторних вимог і підвищує ефективність під час аудитів та інспекцій.

## **А.2 Стислий зміст основних теоретичних посилок до розробки на основі проведених пошуків**

Потреба правильного зберігання продуктів та дотримання температурного режиму виникає постійно. В наслідок різних непередбачуваних ситуацій(війна, природна катастрофа або різних збоїв у роботі електропостачання) трапляються випадки відключення світла. Для деяких продуктів важливо суворо дотримуватись температурного режиму зберігання. Це такі продукти як м'ясні, молочні, або лікарські препарати. Саме тому важливо вчасно зреагувати на такі ситуації, аби уникнути, або хоча б значно зменшити втрати ПХ або ЛП.

## **А.3 Підстави для розробки**

Підставою для виконання розробки є завдання на дипломний проект на тему “Розробка програмного забезпечення для контролю температурного режиму зберігання харчових продуктів та лікарських засобів в холодильних установках”, затверджене наказом по НУ «Запорізька політехніка» № 168 від 24 квітня 2024 року.

## **А.4 Основні вимоги до програми, що розробляється, початковим даним і результату**

### **А.4.1 Вимоги до функціональних характеристик**

Вимоги до функціональних характеристик:

- повинна бути реалізована операція моніторингу зміни температури в холодильниках;
- повинна бути реалізована операція моніторингу протермінованих продуктів харчування, які зберігаються в холодильниках;
- повинна бути реалізована операція моніторингу продуктів харчування з термінами придатності, які скоро закінчуються які зберігаються в холодильниках;
- повинна бути реалізована можливість додавання товарів в холодильник;
- повинна бути реалізована можливість виймання продуктів харчування з холодильника;
- повинна бути реалізована можливість пошуку продуктів харчування, які зберігаються в холодильниках;
- повинна бути реалізована можливість переміщення продуктів між холодильниками.

### **А.4.2 Вимоги до надійності**

Програмне забезпечення має відповідати певним вимогам щодо надійності, серед яких такі:

- захист від спроб завантаження файлів, тип яких не відповідає вимогам системи;
- захист від помилкових дій користувача, наприклад, у випадку введення невірних даних.

Додатково, програмне забезпечення повинно надійно працювати при безперервній роботі комп'ютера. У разі виникнення неполадок у роботі апаратної частини, відновлення нормальної функціональності програмного забезпечення має здійснюватися після перезавантаження системи.

#### **А.4.3 Умови експлуатації**

Умови експлуатації:

- програмне забезпечення не потребує обслуговуючого персоналу;
- для користування програмним забезпеченням не потрібна додаткова кваліфікація.

#### **А.4.4 Вимоги до складу та параметрів технічних засобів**

Вимоги до складу та параметрів технічних засобів:

- платформа x64-based;
- операційна система Microsoft Windows 11/10/8/7 (рекомендується встановити всі останні оновлення);
- процесор з мінімальною тактовою частотою 1,8 ГГц, рекомендується використовувати як мінімум двоядерний процесор;
- оперативна пам'ять від 2 ГБ (рекомендується 4 ГБ та мінімум 2,5 ГБ при використанні на віртуальній машині);
- роздільна здатність екрану 1920x1080 пікселів з 16-бітовим кольором;
- мінімум 2 ГБ вільного місця на жорсткому диску.

#### **A.4.5 Вимоги до маркування та упакування**

Програмне забезпечення не потребує додаткових дій при маркуванні та упакуванні.

#### **A.4.6 Вимоги до транспортування та збереження**

Вимоги до транспортування й зберігання аналогічні вимогам, які пропонуються до носія, на якому записана програма.

#### **A.5 Вимоги до програмної документації**

Склад програмної документації повинен включати в себе:

- технічне завдання;
- керівництво програміста;
- керівництво оператора.

#### **A.6 Основні обмеження на установку і використання програмного продукту**

Розроблюваний програмний продукт потребує інсталяції. Програмний продукт повинен використовуватись на комп'ютерах які задовільняють вимогам та параметрам технічних засобів. Програмний продукт може використовуватись користувачем у будь-який момент часу, не конфліктуючи з іншими програмами.

**ДОДАТОК Б**  
**Опис програмного продукту**

## **Б.1 Загальні відомості**

Назва програмного продукту – «Temperature and shelf life control».

Розроблюваний ПП призначений для моніторингу температури зберігання ПХ в ХУ. Програмний продукт представлений у вигляді окремого віконного додатку завдяки якому можна переглядати температуру в холодильниках, моніторити продукти, які вже протерміновані, або у яких ТП закінчується. Також можна керувати розміщенням продуктів в холодильнику, а саме додавати, забирати, пошук та переміщення між холодильниками продукту.

Для коректної роботи ПП необхідні такі мінімальні системні вимоги:

- платформа x64-based;
- операційна система Microsoft Windows 11/10/8/7 (рекомендується встановити всі останні оновлення);
- процесор з мінімальною тактовою частотою 1,8 ГГц, рекомендується використовувати як мінімум двоядерний процесор;
- оперативна пам'ять від 2 ГБ (рекомендується 4 ГБ та мінімум 2,5 ГБ при використанні на віртуальній машині);
- роздільна здатність екрану 1920x1080 пікселів з 16-бітовим кольором;
- мінімум 2 ГБ вільного місця на жорсткому диску.

## **Б.2 Функціональне призначення**

Вимоги до функціональних характеристик:

- повинна бути реалізована операція моніторингу зміни температури в холодильниках;
- повинна бути реалізована операція моніторингу протермінованих продуктів харчування, які зберігаються в холодильниках;

- повинна бути реалізована операція моніторингу продуктів харчування з термінами придатності, які скоро закінчуються які зберігаються в холодильниках;
- повинна бути реалізована можливість додавання товарів в холодильник;
- повинна бути реалізована можливість виймання продуктів харчування з холодильника;
- повинна бути реалізована можливість пошуку продуктів харчування, які зберігаються в холодильниках;
- повинна бути реалізована можливість переміщення продуктів між холодильниками;
- повинна бути реалізована можливість обробки помилок під час виконання програми та виведення повідомлення про помилку;
- повинна бути реалізована можливість виводу інформації про програмний продукт у окремому вікні.

### **Б.3 Опис логічної структури**

Модуль MainForm.cs – основний модуль, відповідає за ініціалізацію програмного продукту та елементів користувацького інтерфейсу.

Опис структури модулю:

- temperature\_monitoring – метод, призначений для моніторингу температури кожного холодильника в певний проміжок часу;
- expiration\_date\_control – призначений для моніторингу продуктів, ТП яких закінчується через n днів;
- control\_of\_spoiled\_goods – призначений для моніторингу продуктів, термін придатності яких закінчився на момент перевірки;

- `add_in_fridge` – метод додає до обраного холодильника продукт із вказаними параметрами цього продукту. Додавання відбувається шляхом створення нового запису в базі даних продуктів, які зберігаються;
- `delete_from_fridge` – виймає продукт з обраного холодильника. Виймання відбувається шляхом видалення обраного запису в базі даних продуктів, які зберігаються;
- `product_search` – метод здійснює пошук продукту з бази даних з продуктами, що зберігаються шляхом пошуку запису за унікальним параметром “id”;
- `product_movement` – здійснює переміщення продукту з одного холодильника в інший шляхом пошуку продукту за унікальним параметром “id” в базі даних продуктів, які зберігаються, та зміною значення “fridge” на обраний номер холодильника;
- `setting_frequency_for_temperature` – метод встановлює частоту, з якою ПП буде запускати метод `temperature_monitoring`;
- `setting_the_days_until_the_end_date` – встановлює мінімальну кількість днів, яка може бути між актуальною датою та датою кінцевого ТП.

#### **Б.4 Вибір інструментарію розробки програмного продукту**

Для розробки програмного продукту було обрано мову програмування C#, бо вона легко адаптується і може ефективно працювати в будь-якому середовищі. З невеликими змінами C# може функціонувати на будь-якій операційній системі та інтегруватися з іншими мовами програмування. Ця мова має багатий набір бібліотек та інструментів, що спрощує виконання завдань з аналізу даних. Крім того, C# має велику спільноту, де інженери та фахівці можуть ставити запитання та отримувати відповіді.

Для розробки ПП було обрано середовище Visual Studio, бо воно вирізняється простотою освоєння, кросплатформністю та спеціалізацією на

створенні багатфункціональних додатків. Visual Studio включає такі можливості, як текстовий редактор з підсвічуванням синтаксису, автозавершенням коду та інспекцією змінних, значення яких можна редагувати через графічний інтерфейс.

## Б.5 Виклик і завантаження

Для запуску ПП необхідно запустити на робочому столі файл Temperature and shelf life control.exe, після чого відкриється головне вікно, яке представлено на рис. Б.1.

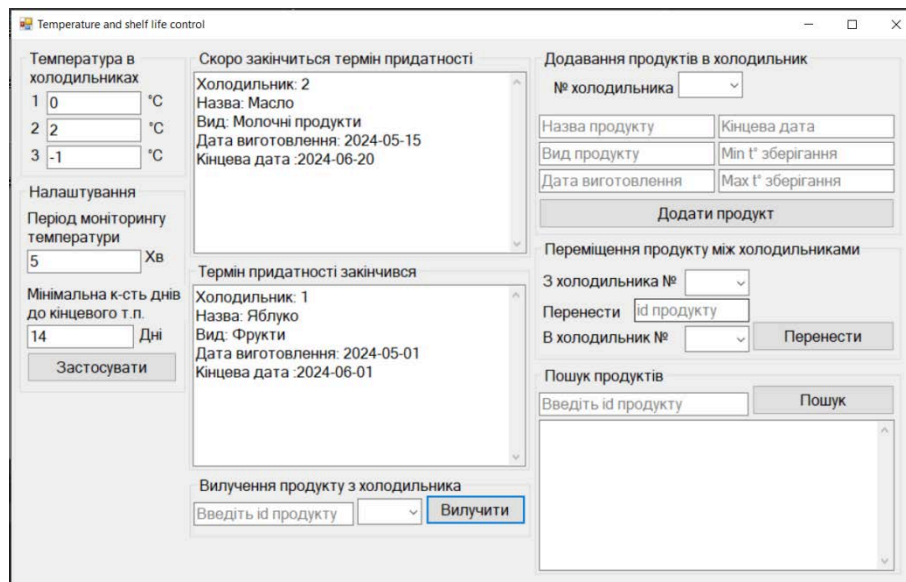


Рисунок Б.1 – Головне вікно ПП

## Б.6 Приклади реалізації

На рисунку Б.2 зображено блок “Температура в холодильниках”. В цьому блоці оператор може переглядати поточну температуру всіх холодильників. ПП виконує перевірку в заданий в блоці ”Налаштування” в полі “Період моніторингу температури”(рис. Б.3).

Рисунок Б.2 – Блок застосунку “Температура в холодильниках”

На рисунку Б.3 зображено блок “Температура”. В цьому блоці оператор може встановити такі параметри як “Період моніторингу температури” та “Мінімальна кількість днів до кінцевого терміну придатності”. Значення поля “Період моніторингу температури” означає період, з яким ПП буде перевіряти, чи немає відхилень у температурному режимі зберігання ПХ холодильнику(в даному випадку 5 хв.). Значення поля “Мінімальна кількість днів до кінцевого терміну придатності” означає мінімальну різницю між кінцевою датою придатності та поточною датою(в даному випадку 14 днів).

Рисунок Б.3 – Блок застосунку “Налаштування”

На рисунку Б.4 зображено блок “Скоро закінчиться термін придатності”. В цьому блоці оператор може переглянути продукти, у яких різниця між кінцевою датою придатності та поточною датою менша зазначеного терміну(рис. Б.3).

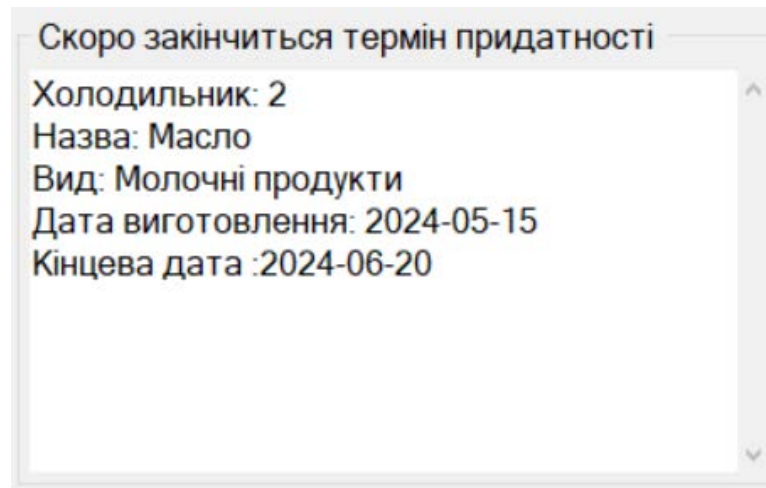


Рисунок Б.4 – Блок застосунку “Скоро закінчиться термін придатності”

На рисунку Б.5 зображено блок “Термін придатності закінчився”. В цьому блоці оператор може переглянути продукти, у яких ТП вже закінчився(поточна дата більша за кінцеву дату придатності продукту).

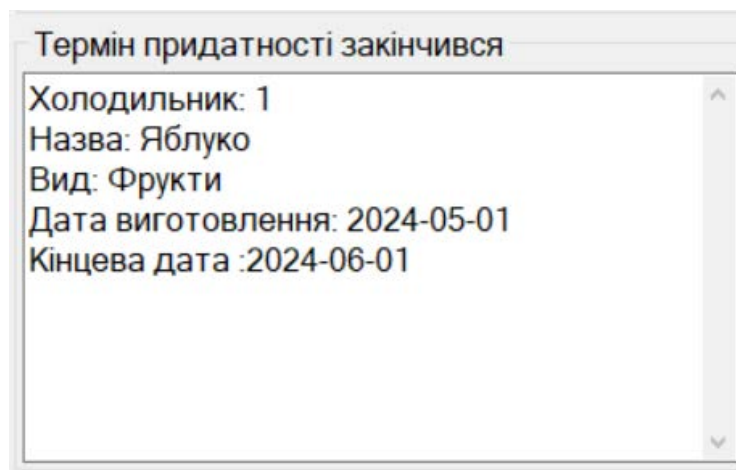


Рисунок Б.5 – Блок застосунку “Термін придатності закінчився”

На рисунку Б.6 зображено блок “Додавання продуктів в холодильник”. В цьому блоці оператор може додати продукт у відповідний холодильник. Блок складається з таких полів:

- № холодильника – номер холодильника, в який поміщають продукт;
- назва продукту – повна назва продукту, який поміщають в холодильник;
- вид продукту – вид продукту(наприклад молочні вироби, м’ясні вироби, фрукти і т.д.), який поміщають в холодильник;
- дата виготовлення – дата виготовлення ПХ, який поміщають в холодильник;
- кінцева дата – кінцева дата ТП ПХ, який поміщають в холодильник;
- $\min t^{\circ}$  зберігання – мінімальна температура зберігання ПХ, який поміщають в холодильник;
- $\max t^{\circ}$  зберігання – мінімальна температура зберігання ПХ, який поміщають в холодильник.

Після заповнення всіх полів треба натиснути кнопку “Додати продукт”, для додавання запису в базу даних з продуктами. Тільки тоді продукт вважається доданим в холодильник.

Додавання продуктів в холодильник

№ холодильника

|                   |                   |
|-------------------|-------------------|
| Назва продукту    | Кінцева дата      |
| Вид продукту      | Min t° зберігання |
| Дата виготовлення | Max t° зберігання |

Додати продукт

Рисунок Б.6 – Блок застосунку “Додавання продуктів в холодильник”

На рисунку Б.7 зображено блок “Переміщення продукту між холодильниками”. В цьому блоці оператор може перемістити продукт з одного холодильника в інший. Блок складається з таких полів:

- з холодильника № – номер холодильника, з якого переміщують продукт;
- перенести – індивідуальний номер продукту, який переміщують в інший холодильник;
- в холодильник № – номер холодильника, в який переміщують продукт.

Після заповнення всіх полів треба натиснути кнопку “Перенести”, для переміщення продукту. Тільки тоді продукт вважається переміщеним.

Переміщення продукту між холодильниками

З холодильника №

Перенести

В холодильник №

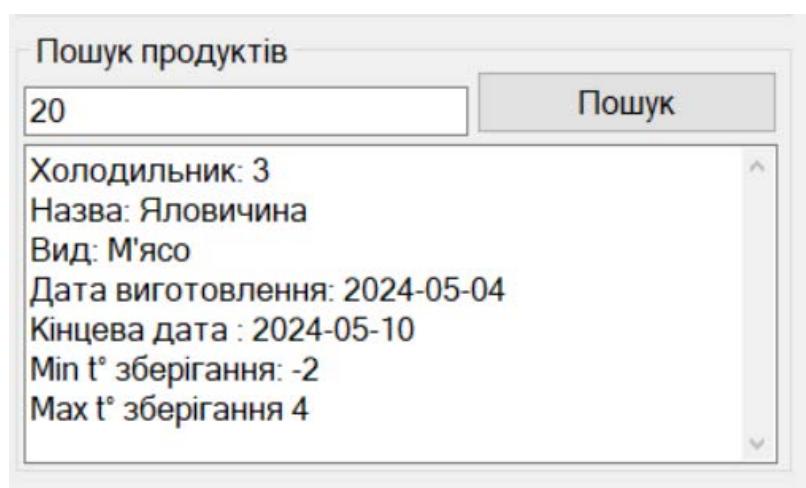
Перенести

Рисунок Б.7 – Блок застосунку “Переміщення продукту між холодильниками”

На рисунку Б.8 зображено блок “Пошук продуктів”. В цьому блоці оператор може зробити пошук продукту за унікальним номером. Блок складається з таких елементів:

- текстове поле “Введіть id продукту” – текстове поле, де треба ввести індивідуальний номер(id) для пошуку продукту;
- порожнє текстове поле – в цьому текстовому полі відображаються результати пошуку продукту.

Після заповнення поля ”Введіть id продукту” треба натиснути кнопку “Пошук”. Після цього, якщо продукт буде знайдено, дані про нього з’являться в текстовому полі нижче.



Пошук продуктів

20

Пошук

Холодильник: 3  
Назва: Яловичина  
Вид: М'ясо  
Дата виготовлення: 2024-05-04  
Кінцева дата : 2024-05-10  
Min t° зберігання: -2  
Max t° зберігання 4

Рисунок Б.8 – Блок застосунку “Пошук продуктів”

На рисунку Б.9 зображено блок “Вилучення продукту з холодильника”. В цьому блоці оператор може зробити вилучення продукту з обраного холодильника за унікальним номером. Блок складається з таких елементів:

- текстове поле “Введіть id продукту” – текстове поле, де треба ввести індивідуальний номер(id) для вилучення продукту;

– порожнє текстове поле з випадаючим списком – в цьому текстовому полі необхідно обрати номер холодильника, з якого треба вилучити продукт.

Після заповнення поля ”Введіть ід продукту” треба натиснути кнопку “Вилучити”. Після цього, продукт вважається вилученим.

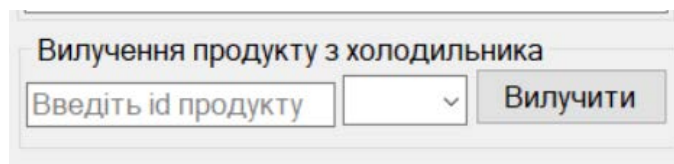


Рисунок Б.9 – Блок застосунку “Вилучення продукту з холодильника”

### Б.7 Вхідні та вихідні дані

В якості вхідних даних, ПП приймає дані про налаштування, холодильники та продукти, які зберігаються в холодильниках. Ці дані представляють собою таблиці створені на локальному сервері до яких звертається застосунок за допомогою сформованих запитів мовою SQL.

Застосунок отримує як вхідні дані:

- temp\_fridge – температура в холодильнику (°C);
- temp\_min – мінімальна температура зберігання продукту (°C);
- temp\_max – максимальна температура зберігання продукту (°C);
- current\_date – поточна дата (dd.mm.yyyy);
- end\_date – дата закінчення ТП продукту (dd.mm.yyyy);
- n\_days – кількість днів до кінцевого ТП (доба);
- temp\_frequency – частота перевірки температури зберігання продуктів в холодильниках (хв).

На рис. Б.10 – Б.12 представлено приклад вхідних даних для моніторингу застосунком температури і ТП.

| id | temp_frequency | n_days |
|----|----------------|--------|
| 1  | 5              | 14     |

Рисунок Б.10 – Вхідні дані для моніторингу застосунком температури і ТП з таблиці settings

| id | temperature |
|----|-------------|
| 1  | 0           |
| 2  | 2           |
| 3  | -1          |

Рисунок Б.11 – Вхідні дані для моніторингу застосунком температури і ТП з таблиці fridge

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-15 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |

Рисунок Б.12 – Вхідні дані для моніторингу застосунком температури і ТП з таблиці products

Вихідними даними у разі виявлення невідповідності умові в термінах придатності є список продуктів, які не пройшли перевірку.

Для здійснення додавання, виймання, пошуку та переміщення між холодильниками продуктів ПП отримує дані з таблиці продуктів.

Застосунок отримує як вхідні дані:

- fridge\_№ – номер холодильника;
- name – назва продукту;
- type – тип продукту;
- date\_of\_manufacture – дата виготовлення продукту;
- date\_last – кінцевий ТП продукту;
- Tmin – мінімальна температура зберігання продукту;
- Tmax – максимальна температура зберігання продукту.

На рис. Б.13 представлено приклад вхідних даних для додавання, виймання, пошуку та переміщення між холодильниками продуктів.

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-15 | 1    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |

Рисунок Б.13 – Вхідні дані для додавання, виймання, пошуку та переміщення між холодильниками продуктів з таблиці products

Вихідними даними є результати роботи з вмістом холодильників, а саме записи в базі даних продуктів які зберігаються як результат таких методів як додавання, виймання та переміщення між холодильниками продукту.

**ДОДАТОК В**  
**Текст програми**

### **В.1 Файл Form1.cs**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Data.SqlClient;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Temperature_and_shelf_life_control
{
    public partial class MainForm : Form
    {
        public MainForm()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            textBox8.Text = "Назва продукту"; //підказка
            textBox8.ForeColor = Color.Gray;
            textBox9.Text = "Вид продукту"; //підказка
            textBox9.ForeColor = Color.Gray;
        }
    }
}
```

```

textBox10.Text = "Дата виготовлення"; //підказка
textBox10.ForeColor = Color.Gray;
textBox11.Text = "Кінцева дата"; //підказка
textBox11.ForeColor = Color.Gray;
textBox12.Text = "Min t° зберігання"; //підказка
textBox12.ForeColor = Color.Gray;
textBox13.Text = "Max t° зберігання"; //підказка
textBox13.ForeColor = Color.Gray;
textBox14.Text = "id продукту"; //підказка
textBox14.ForeColor = Color.Gray;
textBox16.Text = "Введіть id продукту"; //підказка
textBox16.ForeColor = Color.Gray;
textBox17.Text = "Введіть id продукту"; //підказка
textBox17.ForeColor = Color.Gray;
}

private void buttonAdd_Click(object sender, EventArgs e)
{
    textBox8.Text = "Назва продукту"; //підказка
    textBox8.ForeColor = Color.Gray;
    textBox9.Text = "Вид продукту"; //підказка
    textBox9.ForeColor = Color.Gray;
    textBox10.Text = "Дата виготовлення"; //підказка
    textBox10.ForeColor = Color.Gray;
    textBox11.Text = "Кінцева дата"; //підказка
    textBox11.ForeColor = Color.Gray;
    textBox12.Text = "Min t° зберігання"; //підказка
    textBox12.ForeColor = Color.Gray;
    textBox13.Text = "Max t° зберігання"; //підказка

```

```
        textBox13.ForeColor = Color.Gray;  
        add_in_fridge();  
    }
```

```
private void textBox8_Enter(object sender, EventArgs e)  
{  
    textBox8.Text = "";  
    textBox8.ForeColor = Color.Black;  
}
```

```
private void textBox9_Enter(object sender, EventArgs e)  
{  
    textBox9.Text = "";  
    textBox9.ForeColor = Color.Black;  
}
```

```
private void textBox10_Enter(object sender, EventArgs e)  
{  
    textBox10.Text = "";  
    textBox10.ForeColor = Color.Black;  
}
```

```
private void textBox11_Enter(object sender, EventArgs e)  
{  
    textBox11.Text = "";  
    textBox11.ForeColor = Color.Black;  
}
```

```
private void textBox12_Enter(object sender, EventArgs e)
```

```
{
    textBox12.Text = "";
    textBox12.ForeColor = Color.Black;
}

private void textBox13_Enter(object sender, EventArgs e)
{
    textBox13.Text = "";
    textBox13.ForeColor = Color.Black;
}

private void textBox14_Enter(object sender, EventArgs e)
{
    textBox14.Text = "";
    textBox14.ForeColor = Color.Black;
}

private void buttonMove_Click(object sender, EventArgs e)
{
    textBox14.Text = "id продукту"; //підказка
    textBox14.ForeColor = Color.Gray;
    product_movement();
}

private void textBox16_Enter(object sender, EventArgs e)
{
    textBox16.Text = "";
    textBox16.ForeColor = Color.Black;
}
```

```
private void buttonSearch_Click(object sender, EventArgs e)
{
    textBox16.Text = "Введіть id продукту"; //підказка
    textBox16.ForeColor = Color.Gray;
}
```

```
private void buttonDelete_Click(object sender, EventArgs e)
{
    textBox17.Text = "Введіть id продукту"; //підказка
    textBox17.ForeColor = Color.Gray;
    delete_from_fridge();
}
```

```
private void textBox17_Enter(object sender, EventArgs e)
{
    textBox17.Text = "";
    textBox17.ForeColor = Color.Black;
}
```

```
private void buttonApply_Click(object sender, EventArgs e)
{
    if (textBox5.Text != "" && textBox6.Text != "")
    {
        setting_frequency_for_temperature(textBox5.Text);
        setting_the_days_until_the_end_date(textBox6.Text);
    }
    else MessageBox.Show("Невірні дані!");
}
```

```
private void temperature_monitoring()
{
    textBox1.Text = "0";
    textBox2.Text = "2";
    textBox3.Text = "-1";
}
```

```
private void expiration_date_control()
{
    textBox6.Text = "14";
}
```

```
private void control_of_spoiled_goods()
{
    textBox5.Text = "5";
}
```

```
private void add_in_fridge()
{
```

```
    MessageBox.Show("Продукт успішно додано.");
```

```
    //////////////////////////////////
```

```
    string connectionString =
```

```
"Server=localhost;Database=your_database_name;Uid=root;Pwd=your_password;";
```

```
    MySqlConnection connection = new
```

```
MySqlConnection(connectionString);
```

```
    MySqlCommand command = connection.CreateCommand();
```

```

try
{
    connection.Open();

    // SQL-запит INSERT
    string sql = "INSERT INTO products (fridge_№, name, type,
date_of_manufacture, date_last, Tmin, Tmax) " +
        "VALUES (@fridgeNumber, @name, @type,
@dateOfManufacture, @dateLast, @Tmin, @Tmax)";

    command.CommandText = sql;

    // Параметри для вставки
    command.Parameters.AddWithValue("@fridgeNumber",
comboBox1.Text);
    command.Parameters.AddWithValue("@name", textBox8.Text);
    command.Parameters.AddWithValue("@type", textBox9.Text);
    command.Parameters.AddWithValue("@dateOfManufacture",
textBox10.Text);
    command.Parameters.AddWithValue("@dateLast",
textBox11.Text);
    command.Parameters.AddWithValue("@Tmin", textBox12.Text);
    command.Parameters.AddWithValue("@Tmax",
textBox13.Text);

    // Виконання запиту
    int rowsAffected = command.ExecuteNonQuery();
    Console.WriteLine($"Додано записів: {rowsAffected}");

```

```

    }
    catch (Exception ex)
    {
        Console.WriteLine($"Помилка: {ex.Message}");
    }
    finally
    {
        connection.Close();
    }
    //////////////////////////////////
}

private void delete_from_fridge()
{
    MessageBox.Show("Продукт успішно вийнято.");
    //////////////////////////////////
    string connectionString =
"Server=localhost;Database=your_database_name;Uid=root;Pwd=your_password;";

    MySqlConnection connection = new
MySqlConnection(connectionString);
    MySqlCommand command = connection.CreateCommand();

    try
    {
        connection.Open();

        // SQL-запит DELETE

```

```

string sql = "DELETE FROM products WHERE id = @id";

command.CommandText = sql;

// Параметри для видалення
command.Parameters.AddWithValue("@id", textBox17.Text);

// Виконання запиту
int rowsAffected = command.ExecuteNonQuery();
Console.WriteLine($"Видалено записів: {rowsAffected}");
}
catch (Exception ex)
{
    Console.WriteLine($"Помилка: {ex.Message}");
}
finally
{
    connection.Close();
}

//////////

}

private void product_search()
{
    //////////

    string connectionString =
"Server=localhost;Database=your_database_name;Uid=root;Pwd=your_password;
";

```

```

        MySqlConnection connection = new
MySqlConnection(connectionString);
        MySqlCommand command = connection.CreateCommand();

        try
        {
            connection.Open();

            // SQL-запит SELECT для пошуку за id
            string sql = "SELECT * FROM products WHERE id = @id";

            command.CommandText = sql;

            // Параметри для пошуку
            command.Parameters.AddWithValue("@id", 1);

            // Виконання запиту і отримання результатів
            MySqlDataReader reader = command.ExecuteReader();

            while (reader.Read())
            {
                // Отримання даних з результатів запиту
                int id = reader.GetInt32("id");
                string fridgeNumber = reader.GetString("fridge_number");
                string name = reader.GetString("name");
                string type = reader.GetString("type");
                DateTime dateOfManufacture =
reader.GetDateTime("date_of_manufacture");
                DateTime dateLast = reader.GetDateTime("date_last");

```

```

        int tMin = reader.GetInt32("tmin");
        int tMax = reader.GetInt32("tmax");

        // Виведення результатів на консоль
        Console.WriteLine($"ID: {id}, Fridge Number:
{fridgeNumber}, Name: {name}, Type: {type}, Date of Manufacture:
{dateOfManufacture}, Date Last: {dateLast}, Tmin: {tMin}, Tmax: {tMax}");
    }

    reader.Close();
}
catch (Exception ex)
{
    Console.WriteLine($"Помилка: {ex.Message}");
}
finally
{
    connection.Close();
}
//////////
}

private void product_movement()
{
    MessageBox.Show("Продукт успішно переміщено.");
    //////////
    string connectionString =
"Server=localhost;Database=your_database_name;Uid=root;Pwd=your_password;
";

```

```

        MySqlConnection connection = new
MySqlConnection(connectionString);
        MySqlCommand command = connection.CreateCommand();

        try
        {
            connection.Open();

            // SQL-запит UPDATE для заміни id
            string sql = "UPDATE products SET id = @newId WHERE id =
@oldId";

            command.CommandText = sql;

            // Параметри для заміни id
            command.Parameters.AddWithValue("@newId", 100); // Нове
значення id
            command.Parameters.AddWithValue("@oldId", 1); // Старе
значення id, яке потрібно замінити

            int rowsAffected = command.ExecuteNonQuery();

            Console.WriteLine($"Кількість змінених рядків:
{rowsAffected}");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Помилка: {ex.Message}");

```

```

    }
    finally
    {
        connection.Close();
    }
    //////////////////////////////////

}

private void setting_frequency_for_temperature(String temperature)
{
    //////////////////////////////////

    string connectionString =
"Server=localhost;Database=your_database_name;Uid=root;Pwd=your_password;";

    MySqlConnection connection = new
MySqlConnection(connectionString);

    MySqlCommand command = connection.CreateCommand();

    try
    {
        connection.Open();

        // SQL-запит UPDATE для заміни id
        string sql = "UPDATE products SET id = @newId WHERE id =
@oldId";

        command.CommandText = sql;

```

```

        // Параметри для заміни id
        command.Parameters.AddWithValue("@newId", 100); // Нове
значення id

        command.Parameters.AddWithValue("@oldId", 1); // Старе
значення id, яке потрібно замінити

        int rowsAffected = command.ExecuteNonQuery();

        Console.WriteLine($"Кількість змінених рядків:
{rowsAffected}");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Помилка: {ex.Message}");
    }
    finally
    {
        connection.Close();
    }
    //////////////////////////////////////

}

private void setting_the_days_until_the_end_date(String days)
{
    //////////////////////////////////////

    string connectionString =
"Server=localhost;Database=your_database_name;Uid=root;Pwd=your_password;
";

```

```

        MySqlConnection connection = new
MySqlConnection(connectionString);
        MySqlCommand command = connection.CreateCommand();

        try
        {
            connection.Open();

            // SQL-запит UPDATE для заміни id
            string sql = "UPDATE products SET id = @newId WHERE id =
@oldId";

            command.CommandText = sql;

            // Параметри для заміни id
            command.Parameters.AddWithValue("@newId", 100); // Нове
значення id
            command.Parameters.AddWithValue("@oldId", 1); // Старе
значення id, яке потрібно замінити

            int rowsAffected = command.ExecuteNonQuery();

            Console.WriteLine($"Кількість змінених рядків:
{rowsAffected}");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Помилка: {ex.Message}");
        }
    } finally { connection.Close(); }

```

////////////////////////////////////  
}  
}  
}

**ДОДАТОК Г**  
**Слайди презентації**

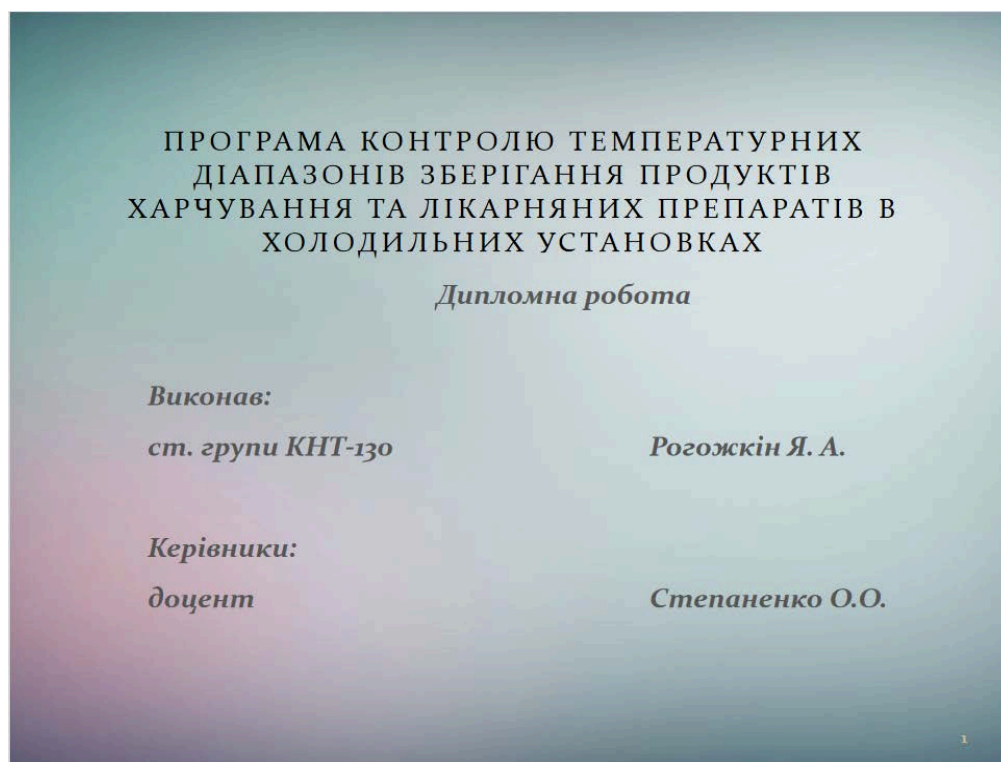


Рисунок Г.1 – Слайд № 1

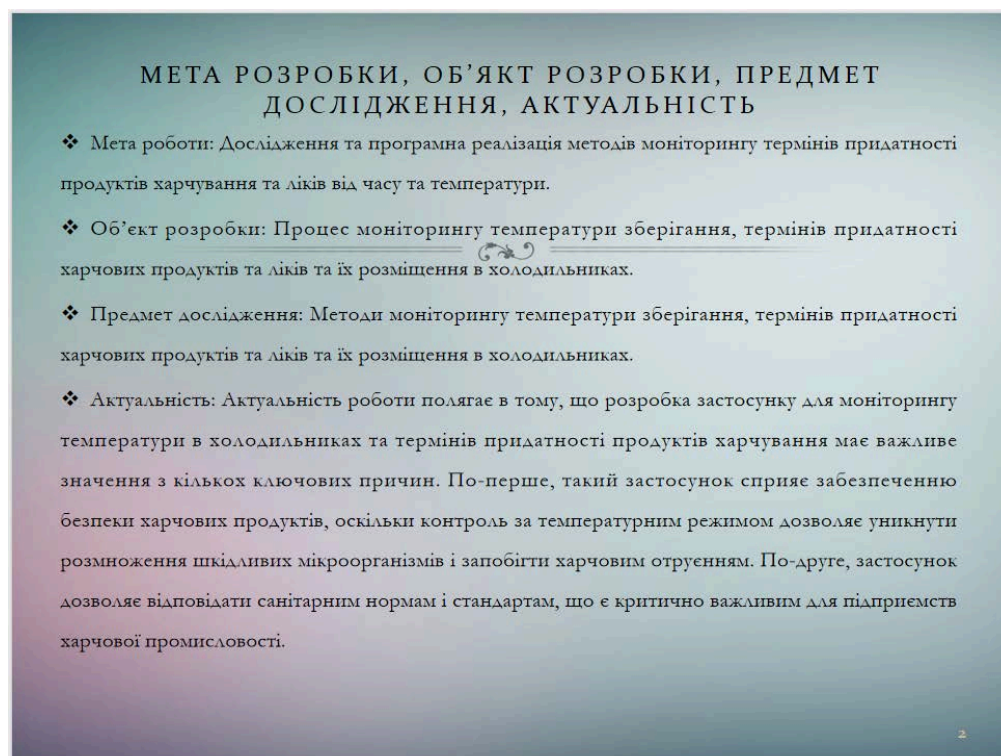


Рисунок Г.2 – Слайд № 2

## ОГЛЯД ІСНУЮЧИХ МЕТОДІВ

| Показники/Застосунок                                    | Ruuvitag | SensGuard | ThermotrackWebService | Мій застосунок |
|---------------------------------------------------------|----------|-----------|-----------------------|----------------|
| Встановлення діапазону температури                      | Вручну   | Вручну    | Вручну                | Вручну         |
| Вимірювання в реальному часі                            | Так      | Так       | Так                   | Так            |
| Сповіщення про неполадки                                | Так      | Так       | Так                   | Так            |
| Можливість встановлення періоду моніторингу температури | Ні       | Ні        | Так                   | Так            |
| Кросплатформність                                       | Ні       | Так       | Так                   | Так            |
| Моніторинг термінів придатності продуктів               | Ні       | Ні        | Ні                    | Так            |
| Управління вмістом холодильників                        | Ні       | Ні        | Ні                    | Так            |
| Дистанційний моніторинг                                 | Так      | Так       | Так                   | Так            |

3

Рисунок Г.3 – Слайд № 3

## ЗАПРОПОНОВАНЕ РІШЕННЯ



Інноваційний підхід у вигляді застосунку поєднає функції моніторингу термінів придатності продуктів, температури в холодильнику та організації вмісту холодильників. Застосунок дозволяє користувачам організовувати вміст холодильника, отримувати сповіщення про закінчення терміну придатності та моніторити зміну температурного режиму. Для зручності користувачів розроблено методи додавання, виймання, пошуку продуктів та переміщення їх між холодильниками, що полегшує роботу з вмістом та розміщенням продукції.

4

Рисунок Г.4 – Слайд № 4

## ВИБІР МОВИ ПРОГРАМУВАННЯ

| Критерії порівняння              | R       | C#   | Python                             | Swift   | Java |
|----------------------------------|---------|------|------------------------------------|---------|------|
| Ліцензія                         | GNU GPL | RAND | Python Software Foundation License | GNU GPL | MIT  |
| Простота використання            | -       | +    | +                                  | +       | -    |
| Підтримка крос-платформеності    | +       | +    | +                                  | -       | +    |
| Продуктивність                   | +       | +    | +                                  | +       | +    |
| Розвиток та підтримка            | +       | +    | +                                  | +       | +    |
| Візуальні інтерфейси (GUI)       | +       | +    | -                                  | +       | +    |
| Підтримка великих проєктів       | -       | +    | -                                  | -       | +    |
| Спільнота та ресурси             | +       | +    | +                                  | +       | +    |
| Інтеграція з іншими технологіями | +       | +    | +                                  | -       | +    |

Рисунок Г.5 – Слайд №5

## ВИБІР СЕРЕДОВИЩА РОЗРОБКИ

| Критерії порівняння                           | Visual Studio | Visual Studio Code | JetBrains Rider | Xamarin Studio |
|-----------------------------------------------|---------------|--------------------|-----------------|----------------|
| Ліцензія                                      | MIT           | MIT                | -               | MIT            |
| Кросплатформність                             | +             | +                  | +               | -              |
| Автозавершення коду                           | +             | +                  | +               | +              |
| Підтримка репозиторіїв                        | +             | +                  | +               | +              |
| Простота встановлення                         | +             | -                  | +               | -              |
| Перевірка та аналіз коду                      | +             | +                  | +               | +              |
| Швидкий та безпечний рефакторинг коду         | +             | +                  | -               | +              |
| Наявність інтерактивної консолі               | +             | +                  | +               | +              |
| Наявність зручного користувацького інтерфейсу | +             | +                  | +               | +              |
| Швидкий та зручний пошук                      | +             | +                  | -               | -              |
| Підсвічування синтаксису                      | +             | +                  | +               | +              |
| Підтримка наукових пакетів                    | +             | +                  | +               | -              |
| Наявність функції статичного аналізу коду     | +             | +                  | +               | +              |

Рисунок Г.6 – Слайд № 6

## СТРУКТУРА БАЗИ ДАНИХ

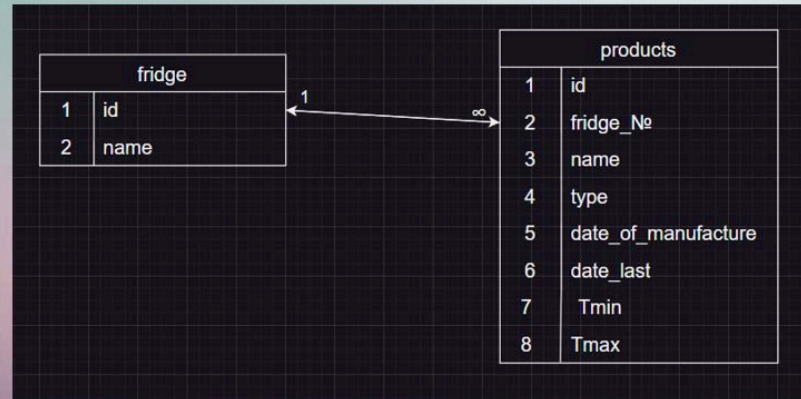


Рисунок Г.7 – Слайд № 7

## ІНТЕРФЕЙС ЗАСТОСУНКУ

**Temperature and shelf life control**

**Температура в холодильниках**

|   |    |    |
|---|----|----|
| 1 | 0  | °C |
| 2 | 2  | °C |
| 3 | -1 | °C |

**Налаштування**

Період моніторингу температури: 5 Хв

Мінімальна к-сть днів до кінцевого т.п.: 14 Дні

**Застосувати**

**Скоро закінчиться термін придатності**

Холодильник: 2  
Назва: Масло  
Вид: Молочні продукти  
Дата виготовлення: 2024-05-15  
Кінцева дата: 2024-06-20

**Термін придатності закінчився**

Холодильник: 1  
Назва: Яблуко  
Вид: Фрукти  
Дата виготовлення: 2024-05-01  
Кінцева дата: 2024-06-01

**Вилучення продукту з холодильника**

Введіть id продукту:  **Вилучити**

**Додавання продуктів в холодильник**

№ холодильника:

|                   |                   |
|-------------------|-------------------|
| Назва продукту    | Кінцева дата      |
| Вид продукту      | Min t° зберігання |
| Дата виготовлення | Max t° зберігання |

**Додати продукт**

**Переміщення продукту між холодильниками**

З холодильника №:

Перенести:  id продукту

В холодильник №:  **Перенести**

**Пошук продуктів**

Введіть id продукту:  **Пошук**

Рисунок Г.8 – Слайд № 8

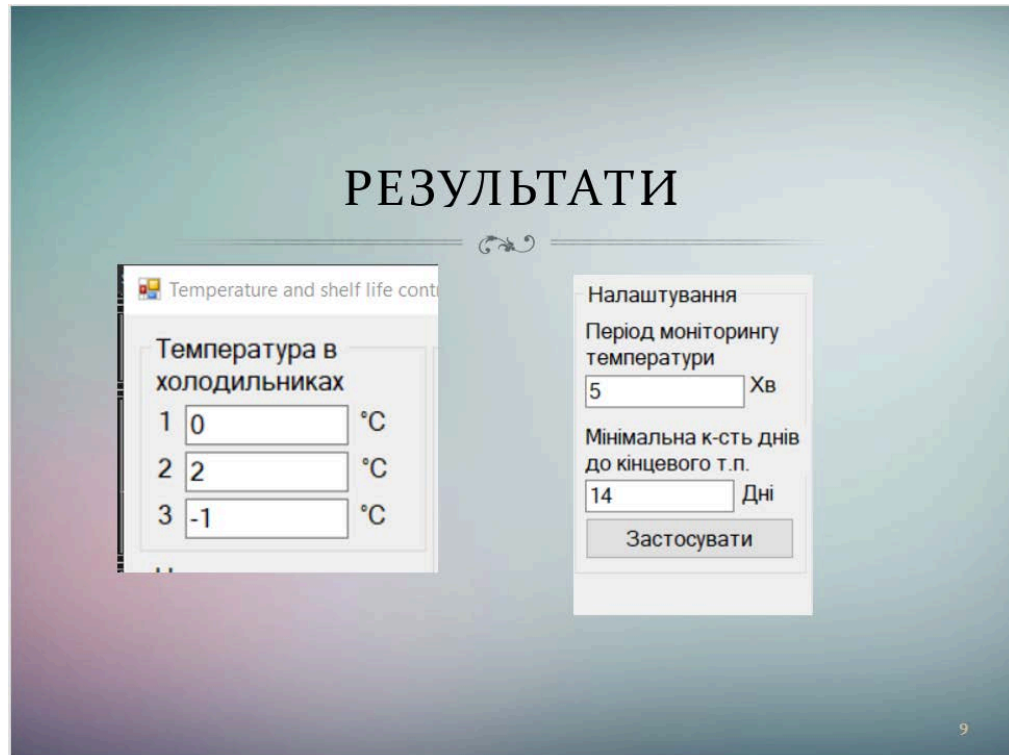


Рисунок Г.9 – Слайд № 9

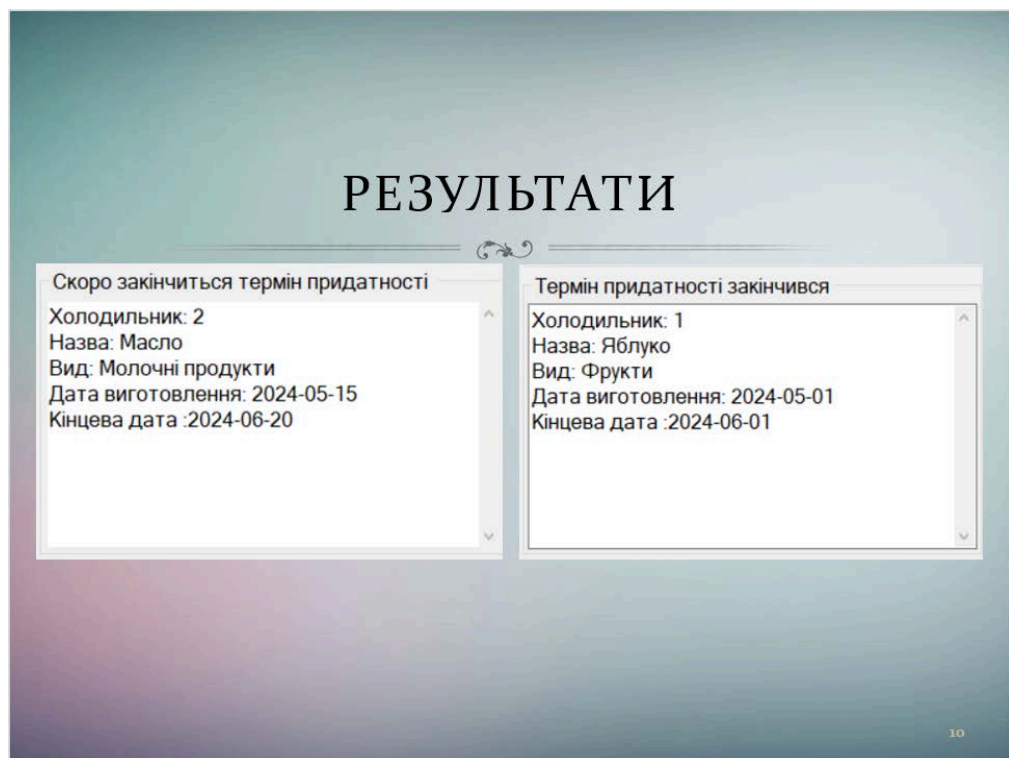


Рисунок Г.10 – Слайд № 10

# РЕЗУЛЬТАТИ

Додавання продуктів в холодильник

№ холодильника1

HELL2024-07-05

Енергетик0

2024-05-204

Додати продукт

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |
| 21 | 1         | HELL         | Енергетик        | 2024-05-20          | 2024-07-05 | 0    | 4    |

Рисунок Г.11 – Слайд № 11

# РЕЗУЛЬТАТИ

Переміщення продукту між холодильниками

З холодильника №1

Перенести21

В холодильник №3

Перенести

| id | fridge_No | name         | type             | date_of_manufacture | date_last  | Tmin | Tmax |
|----|-----------|--------------|------------------|---------------------|------------|------|------|
| 1  | 1         | Яблуко       | Фрукти           | 2024-05-01          | 2024-06-01 | 0    | 4    |
| 2  | 2         | Молоко       | Молочні продукти | 2024-05-10          | 2024-05-20 | 1    | 4    |
| 3  | 1         | Морква       | Овочі            | 2024-05-05          | 2024-06-05 | 0    | 5    |
| 4  | 3         | Куряче філе  | М'ясо            | 2024-05-12          | 2024-05-18 | -2   | 4    |
| 5  | 3         | Сир          | Молочні продукти | 2024-05-07          | 2024-05-27 | 1    | 5    |
| 6  | 1         | Виноград     | Фрукти           | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 7  | 2         | Йогурт       | Молочні продукти | 2024-05-09          | 2024-05-19 | 1    | 4    |
| 8  | 3         | Броколі      | Овочі            | 2024-05-03          | 2024-05-13 | 0    | 4    |
| 9  | 2         | Свиняче філе | М'ясо            | 2024-05-11          | 2024-05-17 | -2   | 4    |
| 10 | 1         | Апельсин     | Фрукти           | 2024-05-06          | 2024-05-16 | 0    | 4    |
| 11 | 2         | Масло        | Молочні продукти | 2024-05-15          | 2024-06-20 | 1    | 4    |
| 12 | 3         | Груша        | Фрукти           | 2024-05-10          | 2024-05-20 | 0    | 4    |
| 13 | 1         | Печінка      | М'ясо            | 2024-05-09          | 2024-05-15 | -2   | 4    |
| 14 | 1         | Огірок       | Овочі            | 2024-05-08          | 2024-05-18 | 0    | 4    |
| 15 | 2         | Сметана      | Молочні продукти | 2024-05-12          | 2024-05-22 | 1    | 4    |
| 16 | 3         | Шинка        | М'ясо            | 2024-05-07          | 2024-05-13 | -2   | 4    |
| 17 | 2         | Помідор      | Овочі            | 2024-05-14          | 2024-05-24 | 0    | 4    |
| 18 | 1         | Кефір        | Молочні продукти | 2024-05-16          | 2024-05-26 | 1    | 4    |
| 19 | 2         | Персик       | Фрукти           | 2024-05-13          | 2024-05-23 | 0    | 4    |
| 20 | 3         | Яловичина    | М'ясо            | 2024-05-04          | 2024-05-10 | -2   | 4    |
| 21 | 3         | HELL         | Енергетик        | 2024-05-20          | 2024-07-05 | 0    | 4    |

Рисунок Г.12 – Слайд № 12

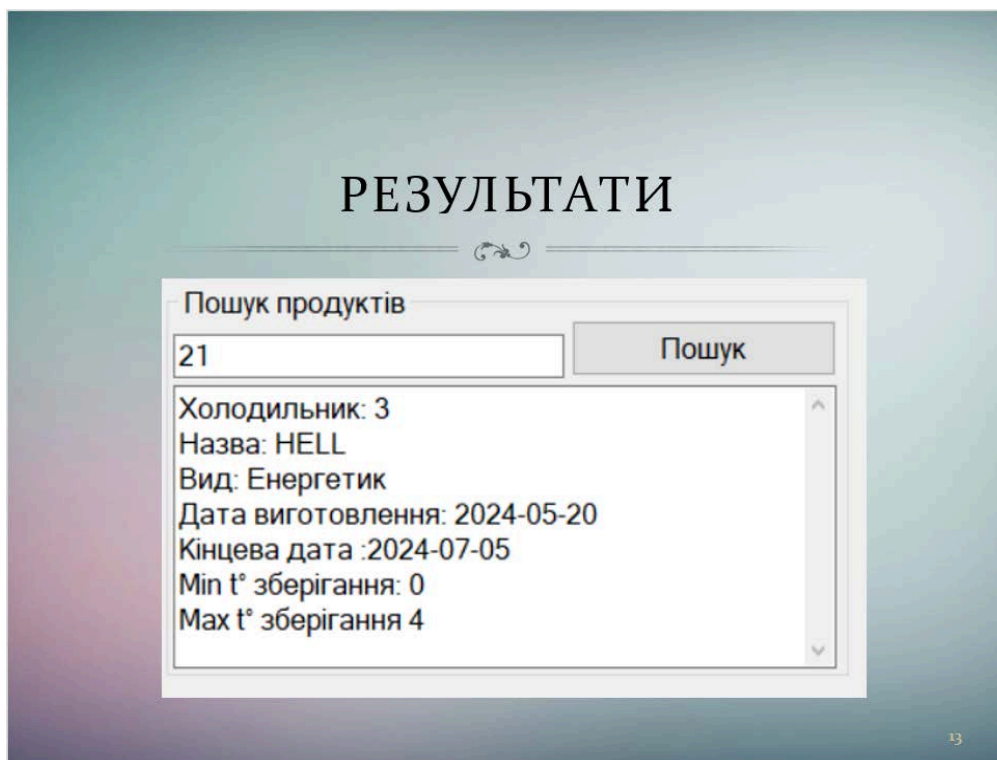


Рисунок Г.13 – Слайд № 13

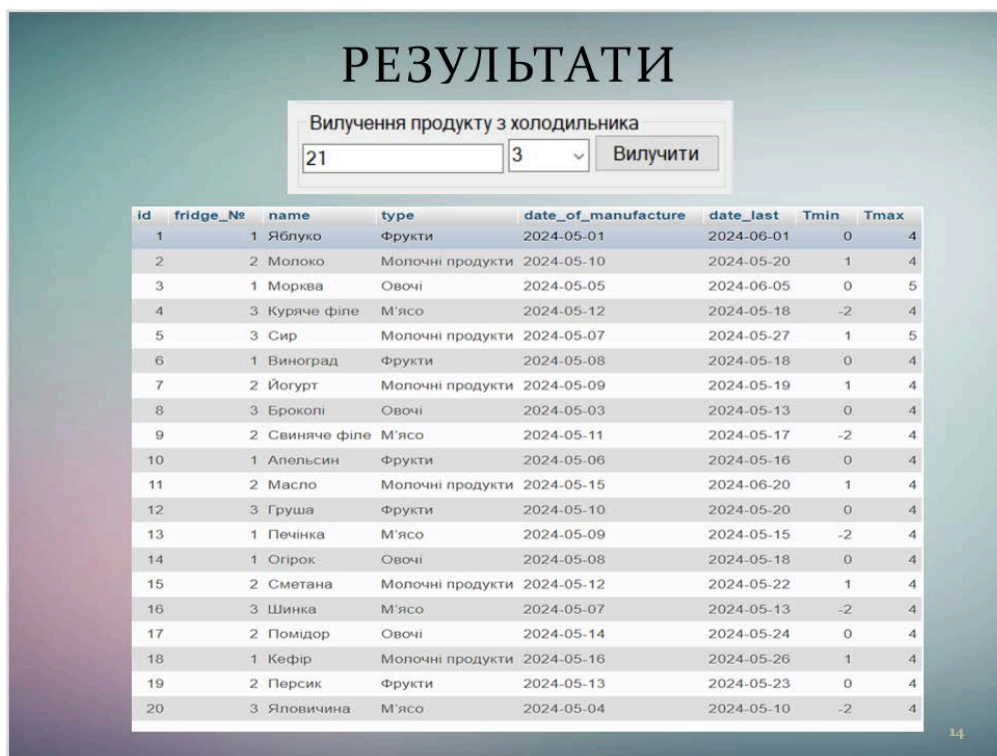


Рисунок Г.14 – Слайд № 14

## ВИСНОВОК

Під час роботи було проаналізовано предметну область, визначено основні поняття та проблематику завдання, обґрунтовано його актуальність та доцільність вирішення. Досліджено методи моніторингу температури та термінів придатності продуктів харчування, їхні переваги та недоліки, проведено порівняльний аналіз. Розроблений програмний продукт призначений для моніторингу температури зберігання продуктів у холодильниках, представлений як окремий Windows-застосунок. Він дозволяє переглядати температуру, моніторити протерміновані продукти та керувати розміщенням продуктів: додавати, вилучати, шукати та переміщувати між холодильниками.

15

Рисунок Г.15 – Слайд № 15