

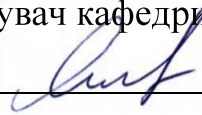
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ



к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2024 р.

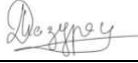
КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему:

Дослідження технологій і розробка інформаційної системи
для овочесховища

Виконав: студент групи 6141 м



(підпис)

Мазурець Д.О.

Керівник роботи:

доцент кафедри ІУСТ, к.т.н.,

(посада, науковий ступінь вчене звання)



(підпис)

Казимиренко Ю.О.

м. Миколаїв – 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 122 "Комп'ютерні науки"

Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

В. Ажищев

к.т.н, доц. Ажищев В.Ф.

"___" _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ на здобуття ступеня вищої освіти "магістр"

Студенту Мазурцю Дмитру Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження технологій і розробка інформаційної системи для овочесховища

Керівник роботи Казимиренко Юлія Олексіївна

Затверджені наказом ректора № 1055-уч. від "08" жовтня 2024 року.

2. Термін подання роботи: 15 грудня 2024 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів)

1. Розробка концепції інформаційної системи обміну інформації на овочесховищі; 2. Проектування інформаційної системи; 3. Реалізація інформаційної системи для овочесховища; 4. Розрахунок економічної ефективності впровадження інформаційної системи для овочесховища; 5. Охорона праці; 6. Охорона навколишнього середовища.

5. Перелік презентаційних матеріалів

6. Консультанти розділів роботи

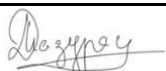
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Казимиренко Ю.О.	05.09.2023	22.12.2023
2	Казимиренко Ю.О.	23.12.2023	15.06.2024
3	Казимиренко Ю.О.	15.06.2024	25.09.2024
4	Казимиренко Ю.О.	25.09.2024	14.11.2024
5	Казимиренко Ю.О.	25.09.2024	14.11.2024
6	Казимиренко Ю.О.	25.09.2024	14.11.2024

7. Дата видачі завдання 05 вересня 2022 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2024	
2	Розробка концепції	15.03.2024	
3	Розділ з розробки проектних рішень по системі (та її частинах)	15.06.2024	
4	Реалізація проекту та розробка робочої документації	25.09.2024	
5	Обчислення економічної ефективності	14.11.2024	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2024	
7	Оформлення пояснювальної записки	14.11.2024	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2024	
9	Подання магістерської роботи рецензенту	20.11.2024	
10	Подання магістерської роботи на державний захист	15.12.2024	

Студент



(підпис)

Мазурець Д.О.

(ПІБ)

Керівник роботи



(підпис)

Казимиренко Ю.О.

(ПІБ)

АНОТАЦІЯ

Магістерська робота «Дослідження технологій і розробка інформаційної системи для овочесховища» за спеціальністю 122 – комп'ютерні науки, освітня програма комп'ютерні науки.

Робота присвячена розробці та впровадженню нової інформаційної системи, призначеної для ефективного управління овочесховищем.

Мета роботи полягає у дослідженні технологій та створенні нової інформаційної системи для овочесховища, яка забезпечить систематизацію обліку продукції та зменшення часу, необхідного для обміну й обробки даних.

Наукова новизна полягає в удосконаленні підходу до розробки моделі даних інформаційної системи, яка, на відміну від існуючих, враховує дані утилізованої, усохшої продукції та мікроклімату.

Практичне значення роботи полягає в розробці інформаційної системи, застосування якої підвищує ефективність інформаційних процесів на існуючих та новостворених овочесховищах.

Методи дослідження: моделювання інформаційних потоків даних, принципи об'єктно-орієнтовного програмування, класична технологія проєктування бази даних.

Робота написана на 123 сторінках машинописного тексту, містить 36 рисунків, 40 таблиць, 2 додатки і список використаних джерел з 31 найменування.

Ключові слова: інформаційна система, база даних, потоки даних, обмін інформацією, овочесховище.

ABSTRACT

Master's thesis: "Research of technologies and development of an information system for a vegetable storage facility" in specialty 122 – computer science, computer science educational program.

The thesis is dedicated to the development and implementation of a new information system designed for efficient management of a vegetable storage facility.

Objective: the research aims to study technologies and create a new information system for vegetable storage, ensuring systematic product accounting and reducing the time required for data exchange and processing.

Scientific novelty: the novelty lies in improving the approach to designing the data model of the information system. Unlike existing models, it considers data on utilized, shriveled products and microclimatic conditions.

Practical significance: the developed information system increases the efficiency of information processes in both existing and newly established vegetable storage facilities.

Research methods: data flow modeling, principles of object-oriented programming, and classical database design techniques.

The thesis comprises 123 pages of typed text, including 36 figures, 40 tables, 2 appendices, and a list of 31 references.

Keywords: information system, database, data flows, information exchange, vegetable storage facility.

ЗМІСТ

ВСТУП.....	5
РОЗДІЛ 1. Розробка концепції Інформаційної системи обміну інформації на овочесховищі.....	7
1.1. Загальна характеристика підприємства.....	7
1.2. Аналіз організаційної структури овочесховища	10
1.3. Аналіз процесів обміну інформацією	17
1.4. Формування вимог до інформаційної системи.....	21
Висновки до розділу 1 та постановка завдань досліджень.....	25
РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	27
2.1. Розробка концепції інформаційної системи	27
2.2. Моделювання та конструювання	31
2.2.1. Розробка діаграми варіантів використання.....	31
2.2.2. Розробка концептуальної моделі даних	35
2.2.3. Розробка діаграми станів	38
2.3. Технічний проект.....	39
2.3.1. Побудова логічної моделі даних	39
2.3.2. Визначення складу модулів програми.....	46
2.3.3. Специфікація модулів програми	47
2.3.4. Розробка алгоритму роботи програми.....	50
2.4. Робочий проект	52
2.4.1. Вибір засобів розробки програмного забезпечення.....	52
2.4.2. Розробка фізичної моделі даних	55
2.4.3. Кодування.....	62
2.4.4. Налаштування програмного забезпечення	63
Висновки до розділу 2.....	64
РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОВОЧЕСХОВИЩА.....	65
Висновки до розділу 3.....	70
РОЗДІЛ 4. РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОВОЧЕСХОВИЩА	71
4.1. Розрахунок вартості матеріально-енергетичних витрат.....	71
4.2. Розрахунок фонду оплати праці за виконану роботу.....	73

4.3. Розрахунок ціни розробки	75
4.4. Розрахунок економічної ефективності та строку окупності розробленої інформаційної системи.....	78
Висновки до розділу 4.....	79
РОЗДІЛ 5. Охорона праці	80
5.1. Охорона праці, мета та основні аспекти.....	80
5.2. Аналіз шкідливих та небезпечних факторів на робочому місці програміста.....	81
5.3. Розрахунок системи штучного освітлення приміщення роботи програміста.....	82
5.4. Заходи щодо зменшення впливу шкідливих та небезпечних факторів при роботі за комп'ютером	83
Висновки до розділу 5.....	85
РОЗДІЛ 6. Охорона навколишнього середовища.....	86
6.1. Охорона навколишнього середовища та його мета	86
6.2. Забруднення навколишнього середовища.....	86
6.3. Розробка заходів щодо зменшення забруднення навколишнього середовища	89
Висновки до розділу 6.....	90
ЗАГАЛЬНІ ВИСНОВКИ.....	91
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	92
ДОДАТОК А – КОД ПРОГРАМИ	95
ДОДАТОК Б – ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ	116

ВСТУП

Актуальність дослідження зумовлена необхідністю побудови у південних регіонах України нових овочесховищ. Одним із ключових завдань інформаційного забезпечення овочесховищ є розробка інтегрованої інформаційної системи, яка охоплює всі процеси, пов'язані з закупкою, продажем, відправкою, зберіганням та обліком товарів. Ефективність виконання цих операцій значною мірою залежить від якісної роботи інформаційної системи овочесховища. Більшість існуючих сучасних інформаційних системи для овочесховища, які використовуються, не враховують умови зберігання та мікроклімату овочесховищ і не можуть бути повністю пристосованими для овочесховищ, тому розробка інформаційної системи є актуальною.

Об'єкт дослідження: процес обміну інформацією стосовно закупівлі та продажу, доставки, зберігання та обліку.

Предмет дослідження: модель даних інформаційної системи овочесховища.

Мета роботи полягає у дослідженні технологій та створенні нової інформаційної системи для овочесховища, яка забезпечить систематизацію обліку продукції та зменшення часу, необхідного для обміну й обробки даних.

Завдання досліджень:

- 1) Провести аналіз організаційної структури овочесховища, процесів обміну інформацією та виявити їх недоліки; провести аналіз готових рішень інформаційних систем та дослідити їх.
- 2) Спроекувати, розробити та налагодити роботу бази даних та програмного забезпечення.
- 3) Застосувати програмне забезпечення на робочих місцях працівників.
- 4) Надати економічне обґрунтування щодо раціональності впровадження інформаційної системи.
- 5) Провести аналіз шкідливих та небезпечних факторів на робочому місці програміста та розробити заходи щодо зменшення їх впливу;

6) Провести аналіз чинників забруднення довкілля через сучасні технології та розробити заходи щодо їх зменшення.

Наукова новизна полягає в удосконаленні підходу до розробки моделі даних інформаційної системи, яка, на відміну від існуючих, враховує дані утилізованої, усохшої продукції та мікроклімату.

Практичне значення роботи полягає в розробці інформаційної системи, застосування якої підвищує ефективність інформаційних процесів на існуючих та новостворених овочесховищах.

РОЗДІЛ 1. РОЗРОБКА КОНЦЕПЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБМІНУ ІНФОРМАЦІЇ НА ОВОЧЕСХОВИЩІ

1.1. Загальна характеристика підприємства

Овочі є одним із ключових елементів у харчуванні сучасної людини, забезпечуючи організм необхідними вітамінами, мінералами, клітковиною та антиоксидантами. Завдяки великому вмісту вітамінів, овочі знижують розвиток злоякісних хвороб та допомагають організму боротися зі стресами та інфекціями[1].

Так як овочі це ресурс, то овочесховища це місця, де подібний ресурс зберігається, подовжуючи час свого існування.

У давнину люди використовували природні методи зберігання овочів, орієнтуючись на кліматичні умови та особливості місцевості. Зазвичай раніше викопували і землі льохи та підземні ями, які забезпечували постійну низьку температуру та оптимальну вологість, що забезпечувало досить довге зберігання. Для запобігання псуванню таких овочів як морква, буряк, ріпа, картопля, капуста тощо їх зверху додатково вкривали товстим шаром землі, соломи або листа, що допомагало уникати висихання, гниття та забезпечувало захист від морозів. Якими б винахідливими раніше не були люди, все одно у такого виду зберігання були недоліки: великий ризик псування, поступова втрата якості, високі затрати робочої сили, неефективний захист від шкідників, малий простір для зберігання овочів та висока залежність від клімату. У давнину зберігання овочів сильно залежало від клімату, бо вищезгадані методи працювали лише в певних умовах з наявним холодом та сухістю[2]. За допомогою сучасних технологій, недоліки були виправлені і науковці працюють лише над тим, як би подовжити термін зберігання овочів.

Овочесховище — це стратегічно важливе рішення, яке відповідає на численні виклики сучасного аграрного, продовольчого та економічного секторів. Овочесховища виконують критично важливу функцію в ланцюгу постачання

продуктів харчування, забезпечуючи збереження врожаїв, стабільність постачання та ефективну логістику[3]. Ключові фактори побудови овочесховищ:

1. Мінімізація втрат урожаю. Відсутність належних умов для зберігання призводить до втрати значної частини врожаю через псування. Овочесховища забезпечують оптимальні умови для збереження свіжості овочів.
2. Підтримання якості продукції. Сховища дозволяють підтримувати необхідну температуру, вологість і вентиляцію, що запобігає гниттю, пересиханню та втраті поживних властивостей.
3. Резерви для міжсезоння. У зимовий або весняний період, коли вирощування овочів неможливе, овочесховища забезпечують постачання продукції з резервів, зберігаючи продовольчу стабільність.
4. Готовність до надзвичайних ситуацій. Наявність запасів на овочесховищах допомагає уникнути дефіциту продуктів у випадку природних катастроф, неврожаю або інших кризових ситуацій.
5. Збільшення доходів фермерів. Фермери отримують можливість продавати більші обсяги продукції, зберігаючи її якість в сховищі, замість втрати через псування.
6. Робота з великими обсягами. Великі овочесховища дозволяють обробляти та зберігати великі партії продукції, що вигідно для постачальників і клієнтів.
7. Покращення умов для фермерів. Доступ до овочесховищ сприяє зростанню кількості фермерських господарств, адже вони мають де зберігати свою продукцію.
8. Створення робочих місць. Будівництво й обслуговування овочесховищ забезпечує робочі місця для місцевого населення, включаючи робітників сховища, водіїв, бухгалтерів, та інший персонал.

Овочесховища відіграють ключову роль життєвого циклу овочів. Вони є проміжною точкою у логістиці, між фермерами та прилавками магазинів, де споживачі, у свою чергу, можуть придбати їх по чудовій ціні та у свіжому стані у будь-яку пору року.

В овочесховищах є всі умови для оптимального та довготривалого зберігання великих обсягів різних видів аграрної продукції:

1. Контроль температури. Овочесховища оснащені холодильними камерами з температурним режимом, який можна налаштувати під кожен вид аграрної продукції.
2. Контроль вологості. Підтримується оптимальний рівень вологості, що запобігає псуванню та зберігає свіжість продуктів.
3. Наявність спеціального газового середовища. Деяка продукція зберігається довше завдяки використанню середовища зі зниженим вмістом кисню (наприклад, томати та яблука).
4. СОРТУВАННЯ ТА КОНТРОЛЬ ЯКОСТІ. Вся наявна продукція сортується та видаляються деякі зіпсовані овочі, запобігаючи поширенню псування.
5. Захист від шкідників. Використовуються засоби для захисту від гризунів, комах та інших шкідників, які можуть зіпсувати продукцію.
6. Санітарні умови. Використовуються сучасні технології і суворий санітарний контроль, який запобігає зараженню овочів.

Зберігання аграрної продукції на овочесховищі — це головний елемент у системі постачання продовольства, що гарантує якість, доступність і свіжість продукції[4]. Завдяки сучасним умовам зберігання, продукція зберігається довше, не втрачаючи своєї свіжості, але якими б функціональними і добрими не були овочесховища, їх прибуток напряду залежить від їх продажів, але не всі овочесховища мають змогу бути в цьому ефективними.

1.2. Аналіз організаційної структури овочесховища

Організаційна структура (рис 1.2) представляє собою внутрішню організацію та розподіл відповідальності, повноважень та обов'язків між працівниками компанії з метою досягнення загальних цілей. Вона складається з різних рівнів та підрозділів, які взаємодіють між собою залежно від функцій та обов'язків. Розглянемо організаційну структуру овочесховища[5].

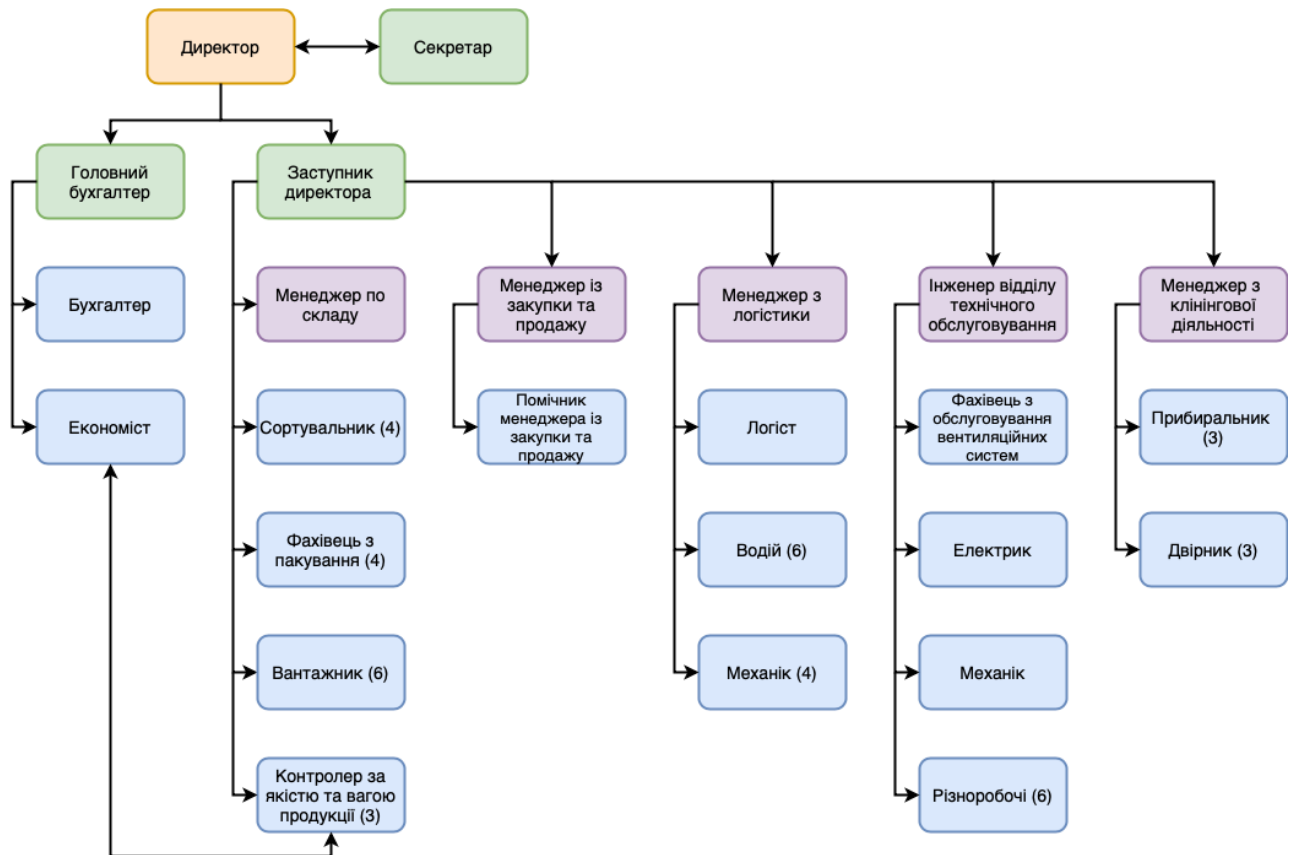


Рис. 1.2. Організаційна структура овочесховища [створено автором]

Директор овочесховища є ключовою фігурою, яка відповідає за повне управління діяльністю підприємства, що включає організацію процесів зберігання, обліку, продажу, закупки і логістики. Він забезпечує стратегічний розвиток компанії, злагоджену роботу всіх відділів, проводить регулярні зустрічі з керівниками підрозділів, контролює виконання поставлених завдань та надає оцінку результатам роботи підрозділів.

Секретар директора забезпечує ефективну роботу керівника, виконуючи широкий спектр обов'язків. До обов'язків секретаря входить організація та

управлінням розкладу директора, підготовка документів і звітів, ведення комунікації з зовнішніми контактами через телефон або електронну пошту, координування проведення зустрічей і нарад, а також створює презентації й інші матеріали відповідно до запитів директора.

Заступник директора овочесховища допомагає директорові керувати роботою овочесховища та виконує його обов'язки у випадку його відсутності. Крім того, заступник директора має власні обов'язки та відповідальність. Він координує роботу підрозділів, забезпечує належні умови зберігання овочів. Керує роботою менеджера по складу, менеджера із закупки та продажу, менеджера з логістики, керує роботою інженера відділу технічного обслуговування та керує відділом комп'ютерних систем. Також контролює дотримання санітарних норм і стандартів якості, та підтримує виконання стратегічних завдань підприємства.

Головний бухгалтер відповідає за ведення фінансового обліку та складання звітності овочесховища. Організовує роботу бухгалтерського відділу, контролює коректність проведення фінансових операцій, займається підготовкою звітів і податкових декларацій. Крім того, співпрацює з іншими відділами та зовнішніми організаціями з питань фінансової діяльності. Головний бухгалтер також аналізує фінансовий стан овочесховища та пропонує рекомендації для його покращення.

Бухгалтер займається веденням обліку фінансових операцій овочесховища. Забезпечує точність облікових записів і звітів, оформлює первинну документацію, здійснює облік операцій із готівковими коштами, розрахунками з клієнтами та постачальниками. Також готує фінансову звітність і надає підтримку головному бухгалтеру у питаннях фінансової діяльності овочесховища.

Економіст співпрацює з іншими відділами овочесховища у питаннях фінансового планування та управління бюджетом, особливо співпрацює з контролером з якості та ваги продукції для подальших коригувань ціни продукції. Контролює виконання бюджету, аналізує показники ефективності

діяльності овочесховища, займається плануванням і прогнозуванням фінансових процесів. Економіст веде облік доходів і витрат та складає фінансові звіти.

Менеджер по складу відповідає за ефективне управління всіма складськими операціями. Веде облік товару, що надходить до сховища, включаючи реєстрацію кількості, якості та терміну зберігання. Проводить контроль за рівнем запасів і своєчасно інформує менеджера із закупки та продажу про необхідність поповнення запасів. Забезпечує порядок на складі, зокрема організовує правильне розміщення товару для зручності доступу. Керує роботою контролера з якості та ваги, сортувальника, фахівця з пакування та вантажника.

Сортувальник проводить сортування нових партій овочів за розміром, кольором, вагою та іншими показниками відповідно до вимог або стандартів якості. Відокремлює продукцію, яка не відповідає стандартам для подальшого зберігання, та доповідає про це контролеру з якості та ваги. Допомогає проводити оцінку овочів на наявність дефектів, пошкоджень або ознак псування.

Фахівець з пакування підбирає відповідні матеріали для пакування овочів (коробки, ящики, плівка, сітки тощо) з урахуванням виду та характеристик продукції. Проводить перевірку якості пакувальних матеріалів на відповідність стандартам (міцність, вентиляція, захист від зовнішніх факторів). Пакує овочі у відповідності до вимог щодо розміру, форми та кількості продукції в упаковці та наносить маркування на упаковку. Перед відправкою перевіряє коректність інформації на етикетках.

Вантажник займається розвантаженням та завантаженням овочів в транспортні засоби. Також може розміщувати продукцію у спеціально відведених для них місцях в сховищі та може брати участь у сортуванні та пакуванні продукції.

Контролер за якістю та вагою продукції на овочесховищі проводить регулярний огляд та оцінює овочі на предмет відповідності стандартам якості. Щотижнево зважує різні типи овочів на втрату ваги (усохлість), докладає про це економісту та фіксує це у звіті. Вимірює температуру, вологість та інші важливі

параметри в складських приміщеннях, щоб забезпечити оптимальні умови для зберігання аграрної продукції. Також перевіряє дотримання термінів зберігання та терміну придатності продукції. Слідкує за дотриманням всіх санітарних та гігієнічних норм і стандартів.

Менеджер із закупки та продажу відповідає за організацію процесів постачання і продажу продукції, допомагаючи овочесховищу ефективно працювати, мінімізувати витрати та максимізувати прибуток. Більш того, він забезпечує стабільність постачань, задоволення клієнтів і підвищення конкурентоспроможності підприємства. Основна його робота полягає у пошуку нових постачальників овочів (фермерів, аграрних підприємств), утриманні наявних та розширенні клієнтської бази (пошук нових партнерів серед роздрібних магазинів, супермаркетів, ринків, переробних підприємств) для збуту продукції. Менеджер проводить переговори щодо умов постачання (ціна, обсяги, строки доставки, якість продукції) та укладає контракти як з постачальниками, так і з клієнтами. Планує обсяги закупок і продажів. Веде архів документації з постачання, збуту продукції, самих постачальників і клієнтів. Контролює терміни доставки від постачальників та клієнтам. Організовує акції, знижки або спеціальні пропозиції для стимулювання продажів. Взаємодіє з іншими відділами (склад, логістика, бухгалтерія) для забезпечення своєчасного виконання замовлень.

Помічник менеджера із закупки та продажу підтримує роботу менеджера, виконуючи всі необхідні завдання, спрямовані на забезпечення безперебійного процесу закупки та продажу овочів. При відсутності менеджера, бере всі його обов'язки на себе. Помічник допомагає у підготовці необхідних документів для закупок і продажу (контракти, заявки, рахунки, звіти). Перевіряє правильність оформлення контрактів, накладних та товаросупровідних документів. Організовує зустрічі, переговори з постачальниками та клієнтами. При великій завантаженості менеджера веде ділову переписку або телефонні переговори з клієнтами, уточнюючи необхідні деталі. Проводить моніторинг ринку, збираючи актуальні інформацію про ціни, конкуренцію, постачальників і клієнтів та

проводить аналіз ринку для забезпечення вигідних закупівельних умов. Також підготовлює таблиць, графіків і звітів про закупку та продажі.

Менеджер з логістики відповідає за організацію, контроль та оптимізацію процесів транспортування. Його робота спрямована на забезпечення своєчасної доставки продукції, збереження її якості та мінімізацію витрат. Менеджер з логістики розробляє маршрути транспортування овочів від постачальників до сховища та від сховища до клієнтів. Планує завантаження і розвантаження для уникнення затримок у роботі складу. Оформлює товаро-транспортні накладні та договори на перевезення, а також веде звітність про виконані логістичні операції. Координує роботу логіста, водіїв і механіків. Взаємодіє з відділом складу та з відділом закупки та продажу.

Логіст допомагає менеджеру з логістики по його справах та виконує його обов'язки у випадку його відсутності, а також має свої обов'язки. Робота логіста спрямована проведення аналізу ефективності логістичних операцій (час доставки, витрати, якість перевезень), створених менеджером з логістики, та пропонування і розробку кращих, більш оптимізованих маршрутів. Також перевіряє відповідність умов транспортування стандартам зберігання овочів, перевіряє належне пакування та фіксацію вантажу для уникнення пошкоджень. При серйозній несправності транспортного засобу вирішує питання оренди транспорту.

Водій на овочесховищі відповідає за транспортування продукції між постачальниками, складом і замовниками, забезпечуючи швидку та безпечну доставку без порушення графіку. Також його робота включає ведення постійного контролю за температурним режимом в морозильній камері транспортного засобу, щоб запобігти псуванню продукції. Проводить базове обслуговування транспорту (заправка, миття). При виявленні несправності транспортного засобу повідомляє механіка про необхідність технічного обслуговування чи ремонту.

Механік відповідає за технічне обслуговування, ремонт та діагностику транспортних засобів, що використовуються підприємством. Механік проводить щоденну перевірку справності транспортних засобів, зокрема перевіряє

гальмівну систему, тиск в шинах, освітлення в кабіні, справність морозильного обладнання тощо. Проводить профілактичне обслуговування, таке як заміна мастил, охолоджувальних рідин, фільтрів, підкачування шин та інше. При несправностях механік проводить ремонт пошкоджених деталей та оцінює якість виконаних робіт. Також веде облік замінених деталей.

Інженер з комп'ютерних систем відповідає за забезпечення безперебійної роботи ІТ-інфраструктури підприємства (комп'ютери, інтернет-з'єднання, веб-сайт). Встановлює, налаштовує та підтримує роботу комп'ютерного обладнання та забезпечує безперебійну роботу локальної мережі, інтернет-з'єднання. Впроваджує заходи для забезпечення кібербезпеки (антивірусні програми, брандмауери, VPN). Підтримує, вдосконалює та розвиває веб-сайт підприємства та програмне забезпечення до бази даних. Регулярно збирає статистику про відвідувачів та аналізує ефективність веб-сайту для впровадження покращень. Створює резервні копії даних, а також надає консультації та технічну допомогу у використанні веб-сайту та офісного обладнання.

Інженер відділу технічного обслуговування відповідає за технічну підтримку, контроль роботи всіх систем та обладнання сховища, забезпечує безперебійну роботу інфраструктури сховища. Також аналізує ефективність роботи технічних систем і обладнання, оформлює акти виконаних робіт, звітів про стан обладнання, планів обслуговування та ремонту. Шукає шляхи для удосконалень системи та обладнання спрямовані на зменшення витрат енергії та підвищення продуктивності. Проводить моніторинг ринку на більш ефективне обладнання для модернізації інфраструктури овочесховища. При оновленні обладнання веде технічну документацію. Веде графік технічного обслуговування і контроль за його дотриманням. Координує роботу фахівця з обслуговування вентиляційної системи, електрика, механіка, різноробочих та прибиральників, а також організує та проводить інструктажі із техніки безпеки для співробітників, які працюють із обладнанням.

Фахівець з обслуговування вентиляційних систем виконує перевірку і моніторинг роботи вентиляційних установок для забезпечення необхідних

температурних умов та умов вологості у складських приміщеннях. Регулює потоки повітря і системи охолодження для підтримання оптимального мікроклімату, необхідного для зберігання продукції. Проводить регулярне обслуговування та очищення вентиляційних установок від пилу, бруду та інших забруднень, які можуть вплинути на їх ефективність. При поломці проводить ремонт та заміну пошкоджених або зношених частин вентиляційних систем (фільтрів, вентиляторів, насосів тощо).

Електрик проводить регулярний моніторинг та підтримує в належному стані електричні мережі, розподільчі щити, вимикачі та електричні панелі. Ремонтує або замінює електричні компоненти, такі як проводка, розетки, освітлення, знижуючи ризик виникнення коротких замикань або інших аварій. Також, при оновленні обладнання, встановлює та необхідним чином підключає його до системи електричної мережі.

Механік проводить регулярний моніторинг та перевірку стану механічного обладнання. Проводить профілактичне обслуговування для запобігання несправностей, поломок або аварійних ситуацій. При несправності, ремонтує та замінює зношені або пошкоджені частини механічного обладнання (підшипників, ременів, моторів, насосів тощо). Також проводить установку нового механічного обладнання, підключає та налаштовує його для подальшої експлуатації.

Різноробочий виконує різні види робіт, залежно від потреб. Займається прибиранням та підтримувати чистоту в приміщеннях. Допмагає у прийомі нового товару на склад або у завантаженні продукції до відправки. Перевіряє кількості і якості овочів при доставці та відправці. Сортує, пакує продукцію. Бере участь у моніторингу, перевірці або ремонті будь-якого виду обладнання. Також виконує інші допоміжні завдань, які можуть виникнути в процесі роботи овочесховища.

Менеджер з клінінгової діяльності відповідає за контроль чистоти та порядку на території та в приміщеннях овочесховища та забезпечує стерильні робочі умови для персоналу підприємства. Займається координацією та

контролем роботи прибиральників та двірників. Забезпечує необхідними матеріалами своїх підлеглих (побутова хімія, обладнання для прибирання, тощо). Також займається розробкою та вдосконаленням інструкцій та стандартів для забезпечення ефективної роботи підлеглих.

Прибиральник проводить регулярне прибирання, підтримуючи чистоту в складських приміщеннях, де зберігаються овочі, включаючи підлогу, стіни, полиці та інше обладнання. Проводить дезінфекцію робочих поверхонь, де відбувається обробка або пакування овочів. Збирає та сортує сміття та відходи (упаковка, пластикові або паперові контейнери, тощо).

Двірник проводить регулярне прибирання території навколо території підприємства. Очищує подвір'я, доріжки, тротуари, паркування, проїзди до паркування від сміття, пилу та листя, а в зимовий період прибирає територію від снігу та льоду. Займається очисткою контейнерів для сміття, ведучи контроль за заповненням сміттєвих баків та забезпечує їх своєчасну очистку. Підтримує особливо суворий порядок біля зон розвантаження та розвантаження.

У результаті дослідження встановлено, що стан інформаційної системи овочесховища є прийнятним, проте його ефективність залишається недостатньо високою[6]. Щоб визначити причини недостатньої продуктивності овочесховища, слід виконати аналіз процесів обміну інформації.

1.3. Аналіз процесів обміну інформацією

Розглянемо детальніше процес закупки та продажу продукції

Процесом закупки овочів займається менеджер із закупки та продажу. Перед закупкою менеджер надає запит до менеджера зі складу про перелік наявної продукції. Визначивши необхідну продукцію для поповнення, менеджер підіймає свій архів з усією необхідною інформацією про постачальників та проводить телефонне опитування кожного з них на наявність необхідної продукції для поповнення, також заходить на спеціальні сайти закупки/продажу овочевої продукції, аналізує пропозиції та проводить телефонні розмови з

власниками поданих оголошень. Знайшовши необхідний товар по вигідній ціні, замовляє в продавця товар та домовляється про терміни підготовки товару та дати транспортування. Домовившись про товар, менеджер шукає у архіві даного постачальника, ставить необхідну помітку про замовлення та записує які саме овочі було замовлено та у якій кількості. У випадку якщо постачальник виявився новим, додає у архів дані нового постачальника. Також веде архів репутації постачальників. Наступним кроком менеджера є передача усіх необхідних даних про постачальника та продукцію до менеджера по логістиці для поставки товару до сховища.

Процесом продажу овочів займається менеджер із закупки та продажу. Менеджер приймає дзвінки від постійних та нових клієнтів, уточнюючи вид овочів, сорт, обсяги замовлення, терміни доставки та інше. Після дзвінка надає запит до менеджера зі складу про перелік наявної продукції, враховуючи всі побажання клієнта та надає запит на продукцію, у якої скоро вийде термін придатності або яка зберігається досить довго на складі. Отримавши свіжу інформацію по товару на складі, телефонує клієнту та надає інформацію щодо запиту клієнта та пропонує продукцію, яка довго лежить на складі по акційній ціні. Визначивши фінальний обсяг замовлення, домовляються про терміни підготовки товару та дати транспортування. Поспілкувавшись з клієнтом, записує до архіву номер клієнта (якщо новий), вид його підприємницької діяльності та місце доставки. Наступним кроком менеджера є передача усіх необхідних даних про замовлення до менеджера по логістиці для поставки замовленої продукції клієнту.

Менеджер із закупки та продажу веде паперовий архів постачальників та клієнтів, що супроводжується незахищеністю інформації. Простежується великий ризик втрати, випадкового знищення або поставлення помилкового запису не в належну картку клієнта з його замовленнями, через що присутній великий ризик відправити не тому клієнту, через це паперовий архів потребує повторної ретельної перевірки. Пошук в архіві займає певний час, а його об'єм з кожним новим замовленням до постачальника або клієнта тільки збільшується.

Довгострокове зберігання даних виливається у проблему, а проводити аналітичну роботу стає все складніше. Більш того, при кожному запиті до менеджера складу на перевірку наявності продукції затрачається дорогий час під час розмови з клієнтом, що може призвести до втрати клієнта через малу пропускну здатність інформації підприємства[7].

Розглянемо процес обліку та утилізації зіпсованих овочів на складі

За процес зберігання та утилізації зіпсованих овочів відповідає менеджер по складу. При поставці товару на склад, веде облік нового товару, записує у спеціальний бланк тип, сорт, середню вагу та вагу усієї поставки одного типу овочів, а також записує номер складу та конкретний осередок зберігання. Проводячи щотижневий огляд та зважування різних типів овочів, записує відсоток усохлості продукції та кількість утилізованої продукції у грамах після чого корегує дані у головному бланку обліку продукції. При запиті менеджера з покупок та продажу виписує необхідні дані на окремий бланк та надає йому інформацію.

Менеджер по складу веде паперовий архів з обліку та утилізації зіпсованої продукції, що є критичною проблемою для роботи підприємства. Архів такого виду має великий ризик незахищеності інформації, обмеженої безпеки, низьку швидкість доступу та помилки через людський фактор. З кожним новим бланком підвищується ймовірність втрати даних, випадкового їх знищення або внесення неправильного запису до певного виду продукції. Така ситуація може спричинити катастрофічні проблеми з репутацією підприємства та відтоку клієнтів, тому кожний новий запис вимагає подвійної перевірки з боку менеджера по складу. Пошук у паперовому архіві займає значний час, і з моменту запиту менеджера із закупки та продажу може пройти не мала кількість часу до того моменту, поки відповідь дійде спершу до менеджера із закупок та продажів, а потім вже до клієнта[7].

Розглянемо процес доставки

За процес доставки відповідає менеджер з логістики. Він приймає запит на доставку від менеджера із закупок та продажу та шукає в архіві вже розроблений

маршрут, якщо постачальник або клієнт є постійним (наявний в архіві). В іншому випадку розробляє та записує новий маршрут на спеціальний бланк. Розробивши маршрут, робить копію бланку та віддає водію. Наступним кроком створює товаро-транспортні накладну, вписуючи номер, дату складання, інформацію про транспортний засіб (марка, модель, реєстраційний номер), який буде виконувати доставку, параметри ТЗ (довжина, ширина, висота), вагу з вантажем та без нього, інформацію про водія, відправника та замовника, пункт завантаження та розвантаження та всю суму вантажу з урахуванням ПДВ та просто суму ПДВ. В накладній вписує також відомості про вантаж: тип і сорт овочів, одиницю виміру, ціну без ПДВ за одиницю продукції, загальну суму з ПДВ, вид пакування, масу в тоннах (разом з пакуванням) та місце складання накладної. Додатково вписує інформацію про вантажно-розвантажувальні операції: ПІБ осіб, які завантажили та розвантажили товар, маса товару (кг), час прибуття, вибуття та час простою. Далі зв'язується з менеджером по складу для підготовки до транспортування необхідної аграрної продукції. Після виконаної доставки, кладе на зберігання товарно-транспортну накладну до архіву. При виявленні незбіжностей у доставці товару або самого товару, шукає необхідну накладу та зв'язує причину незбіжності.

Менеджер з логістики використовує паперовий архів для зберігання розроблених логістичних шляхів та товаро-транспортні накладні. Так як паперова інформація не захищена від впливу зовнішніх факторів, тому існує ймовірність втрати, випадкового пошкодження або помилкового внесення невірних даних до бланків, ураховуючи об'єми інформації до запису, що може призвести до неприємних ситуацій та пониження репутації підприємства. Через це архів потребує частих перевірок, а пошук потрібної інформації займає багато часу. Зі збільшенням кількості доставок обсяг архіву зростає, що ускладнює його зберігання та використання. Крім того, паперова інформація від менеджера із закупок та продажу може доходити через деякий час, беручи на увагу людський фактор, що призводить до низької швидкості обміну інформацією[7].

Проаналізувавши предметну область виявлено, що інформаційна система овочесховища працює, але вже застаріла. Треба вирішити проблему ефективності інформаційних процесі закупівлі та продажу продукції, доставки, обліку усохшої та утилізації зіпсованої продукції для кращих продажів і збільшення прибутку. Перейдемо до складання вимог інформаційної системи.

1.4. Формування вимог до інформаційної системи

Підвищення ефективності роботи овочесховища полягає у необхідному вдосконаленні інформаційних процесів підприємства.

Необхідно розробити програмне забезпечення для вдосконалення внутрішніх процесів обміну інформації на підприємстві, що забезпечує такі функції:

- перегляд, внесення, видалення та коригування інформації щодо аграрної продукції у сховищі;
- перегляд, внесення та коригування інформації щодо списаної продукції;
- перегляд та внесення інформації щодо доставленої та відправленої продукції;
- перегляд, внесення та видалення інформації щодо продукції, яка прямує до складу;
- перегляд, внесення та видалення інформації щодо продукції, яка прямує до клієнтів;
- перегляд, внесення, видалення та коригування інформації щодо постійних клієнтів та постачальників;
- перегляд, внесення та коригування інформації щодо товарно-транспортних накладних;
- перегляд, внесення, видалення та коригування інформації щодо водіїв та вантажників, які працюють на підприємстві;

- перегляд, внесення, видалення та коригування інформації щодо транспортних засобів, які має підприємство;
- перегляд створених базою даних аналітичних діаграм.

Це програмне забезпечення повинно стати надійним інструментом для менеджера по складу, менеджера з логістики, а також ефективним помічником для менеджера із закупок та продажу. Завдяки функціоналу аналітики, воно дозволяє прогнозувати майбутні закупівлі аграрної продукції з урахуванням сезонного зростання попиту.

Необхідно розробити базу даних, у якій має зберігатися інформація про наявну у сховищі продукцію, утилізовану зіпсовану та усохшу продукцію, постачальників, клієнтів, доставлену та відправлену продукцію, товаро-транспортні накладні, водіїв підприємства, транспортні засоби. Також база даних має зберігати інформацію про продукцію, яка перебуває у дорозі як від постачальників до сховища, так і від сховища до клієнта. Крім того, база даних повинна підтримувати функціонування програмного забезпечення.

Необхідно найняти системного адміністратора баз даних для підтримки стабільного функціонування бази даних.

Інформаційна система овочесховища повинна функціонувати на базі ліцензійної версії Windows 10 або новішої.

Для автоматизації інформаційної системи овочесховища можна використовувати різноманітні готові рішення, які забезпечують ефективне управління інформаційними процесами. Серед них можна виділити такі програми як Microsoft Dynamic 365, Odoo та SAP Warehouse Management. Переглянемо їх переваги та недоліки.

Порівняльна таблиця готових рішень [створено автором]

Програмне забезпечення	Переваги	Недоліки
Microsoft Dynamics 365	1. Поєднує функціональність управління ресурсами підприємства та управління взаємодією з клієнтами. 2. Хмарна інфраструктура забезпечує доступ до інформаційної системи дистанційно з будь-якої точки світу. 3. Інтегрується з програмами Microsoft Office. 4. Оснащений інструментами для аналітики даних. 5. Дані захищені за допомогою шифрування та багатофакторною аутентифікацією.	1. Висока ціна преміального продукту та висока ціна впровадження програмного забезпечення. 2. Додаткові програми програми Microsoft Office купляються окремо. 3. Початкове налаштування системи потребує значних технічних знань та може зайняти багато часу. 4. Складний інтерфейс і багатофункціональність вимагають часу на навчання працівників.
Odoo	1. Забезпечує модульну структуру, що дозволяє обирати лише ті модулі, які необхідні інформаційній системи (наприклад, модуль для складу, закупок, продажів).	1. Базова версія програми є безкоштовною, при зростанні бізнесу потрібно буде переходити на платну версію продукту. 2. Відсутність аналітичного блоку та блоку прогнозування.

Продовження таблиці 1.1

	2. Інтуїтивно зрозумілий інтерфейс, що скорочує час на навчання працівників. 3. Можливість роботи через хмару, що забезпечує доступ до інформаційної системи з будь-якої точки світу.	3. Деякі функції можуть не відповідати потребам підприємства та потрібно наймати розробника для доопрацювання програмного продукту.
SAP Warehouse Managment	1. Пропонує великий набір функцій для управління складом від обліку до оптимізації використання простору складу. 2. Дозволяє автоматизувати складські операції, такі як приймання, зберігання, відвантаження та інвентаризація. 3. Надає детальні звіти та аналітику щодо складських операцій. 4. Забезпечує точність у пошуку при великому обсягу інформації.	1. Програмне забезпечення є дорогим рішенням особливо для малого та середнього бізнесу через високу вартість ліцензій. 2. Складний інтерфейс і багатофункціональність системи вимагають тривалого навчання співробітників. 3. Система потребує потужного обладнання та якісного технічного обслуговування.

Проаналізувавши та взваживши переваги та недоліки готових рішень з інформаційних систем, прийнято рішення створити власне програмне забезпечення, яке матиме інтуїтивно зрозумілий інтерфейс, аналітичний блок, малі затрати по обчислювальному обладнанню та короткий термін навчання працівників.

Програмне забезпечення та веб-сайт мають забезпечуватись встановленою платформою .NET Framework версії 8.0 або вище.

База даних повинна функціонувати на базі встановленої платформи MySQL.

Технічне забезпечення інформаційної системи вимагає наявності 5 стаціонарних комп'ютерів із однаковими характеристиками, які будуть використовуватися менеджером із закупки та продажу, менеджером по складу, менеджером з логістики, інженером інформаційних систем, системним адміністратором баз даних.

Висновки до розділу 1 та постановка завдань досліджень

В результаті всебічного аналізу літературних джерел, присвячених вирішенню проблемних питань інформаційного забезпечення на підприємствах зберігання сільськогосподарської продукції, зроблено наступні висновки:

1. Визначено та проведено детальних аналіз організаційної структури сучасних овочесховищ, визначено робочі фігури.
2. Проведено аналіз процесів обміну інформації таких дійових осіб як менеджера із закупок та продажу, менеджера з логістики, менеджера по складу та визначено ключові недоліки цих процесів.
3. Сформульовано вимоги до проекту інформаційної системи.
4. Проаналізовано та оформлено вимоги до програмного та технічного забезпечення.

Мета роботи полягає у дослідженні технологій та створенні нової інформаційної системи для овочесховища, яка забезпечить систематизацію обліку продукції та зменшення часу, необхідного для обміну й обробки даних.

Завдання досліджень:

- 1) Провести аналіз організаційної структури овочесховища, процесів обміну інформацією та виявити їх недоліки; провести аналіз готових рішень інформаційних систем та дослідити їх.

- 2) Спроекувати, розробити та налагодити роботу бази даних та програмного забезпечення.
- 3) Застосувати програмне забезпечення на робочих місцях працівників.
- 4) Надати економічне обґрунтування щодо раціональності впровадження інформаційної системи.
- 5) Провести аналіз шкідливих та небезпечних факторів на робочому місці програміста та розробити заходи щодо зменшення їх впливу.
- 6) Провести аналіз чинників забруднення довкілля через сучасні технології та розробити заходи щодо їх зменшення.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1. Розробка концепції інформаційної системи

На основі сформованих вимог виконана розробка концепції інформаційної системи овочесховища (рис 2.1).

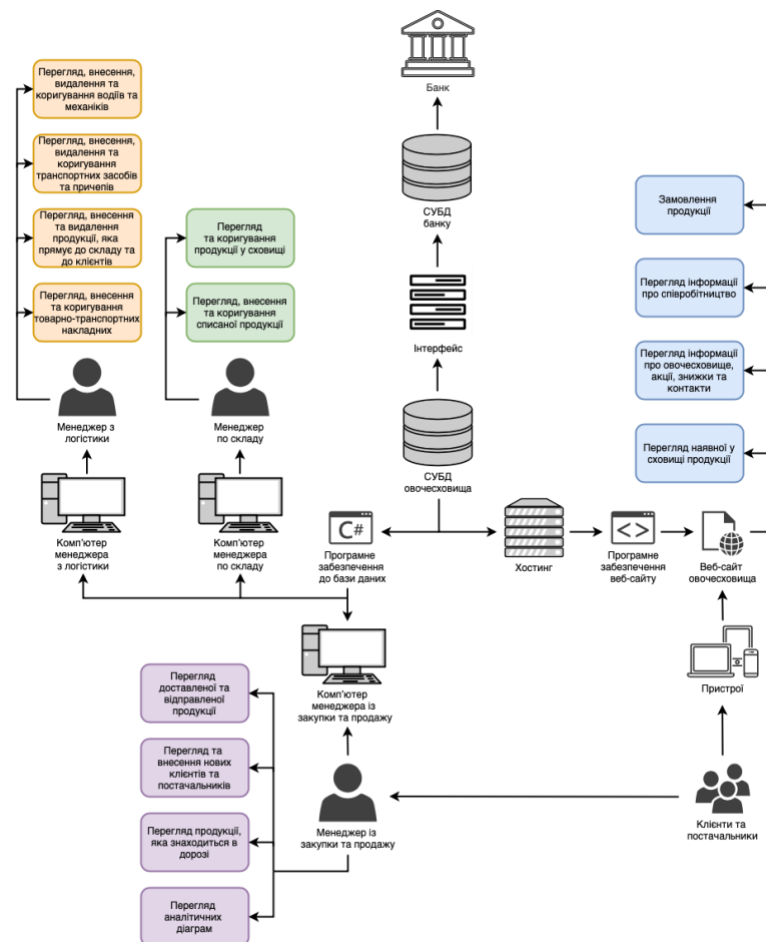


Рис 2.1. Концепція інформаційної системи овочесховища [створено автором]

Програмне забезпечення розгорнуте на комп'ютерах менеджера із закупок та продажів, менеджера по складу та менеджера з логістики. База даних розгорнута на комп'ютері системного адміністратора бази даних.

Клієнт, звернувшись по номеру телефона до менеджера із закупок та продажів, надає інформацію про зацікавлену його продукцію. Менеджер за допомогою програмного забезпечення без складних зусиль переглядає усю наявну продукцію у сховищі та корегує запит клієнта за допомогою спеціальних фільтрів: вид овочів, сорт, зрілість, акції, знижки. За допомогою фільтрів

менеджер може надавати консультацію клієнту швидко та ефективно. Обравши необхідну продукцію, програмне забезпечення автоматично рахує вартість усього замовлення, а менеджер із закупок та продажів тільки оголошує фінальну вартість після чого переглядає перелік вільних водіїв та транспортних засобів та домовляється про дату поставки продукції.

Клієнт, потрапивши на веб-сайт овочесховища має змогу ознайомитись з усією необхідною йому інформацією щодо наявної у сховищі продукції, видів, сортів, цін, якості, об'ємів, акцій, знижок та продукції, яка незабаром з'явиться у продажу. Більш того, може додатково подивитись інформацію щодо інформації про сховище, персонал, час роботи, репутацію, відгуки інших замовників та номеру менеджера з закупок та продажу для додаткової консультації. Також, за допомогою сайту клієнт може самостійно обрати необхідний товар та створити замовлення, увійшовши в систему як постійний клієнт, або зареєструвавшись як новий, надав пошту, номер телефона та придумавши логін та пароль. Після створення, замовлення одразу ж фіксується в базі даних та менеджер з закупки та продажу зв'язується з клієнтом по телефону для підтвердження замовлення та домовляються про дату доставки.

Системний адміністратор бази даних відповідає за налаштування та підтримку бази даних, що використовується для обліку продукції, планування поставок, звітності тощо. Проводить моніторинг продуктивності бази даних та забезпечує можливість швидкого відновлення інформації у разі технічних збоїв або інших надзвичайних ситуацій. Проводить щотижневе резервне копіювання. У разі зміни у бізнес-процесах, адаптує та оптимізує запити до бази даних відповідно до нових вимог. Також надає технічну підтримку працівникам з використання додатка до бази даних у разі потреби.

Менеджер із закупок та продажу звертається до програмного забезпечення з метою ознайомлення з наявними запасами. При зменшенні кількості певного виду продукції, програмне забезпечення автоматично надає менеджеру інформацію щодо продукції, яку необхідно поповнити в залежності від сезону та автоматично надає перелік постачальників, які потенційно можуть задовольнити

цю потребу. Також менеджер із закупки та продажу може самостійно швидко переглядати потрібну йому інформацію щодо постачальників, номерів телефонів, репутації, короткого опису та кількості зроблених поставок, а також, за потреби, може додати або корегувати інформацію.

Менеджер з логістики за допомогою програмного забезпечення автоматично приймає запит на доставку від менеджера із закупок та продажу і створює товаро-транспортні накладну в електронному вигляді, вводячи всі необхідні дані, які пропонує програмне забезпечення, враховуючі дані про замовника або постачальника, водіїв, транспортні засоби, продукцію, вантажників тощо, роблячи процес заповнення накладної малозатратним. Створивши товаро-транспортну накладну, програмне забезпечення автоматично інформує клієнта по електронній пошті, що його замовлення скомплектоване та готове до відправки.

Менеджер по складу веде електронний облік продукції. Проводячи щотижневий огляд та зважування різних типів овочів, записує відсоток усохлості продукції та кількість утилізованої продукції у грамах за допомогою програмного забезпечення, а воно, в свою чергу, автоматично вносить зміни головної таблиці обліку продукції. Більш того, з кожним зважуванням може корегувати дані щодо усохлості продукції з легкістю. На основі раніше введених даних програмне забезпечення прогнозує майбутню усохлість продукції в залежності від виду та сорту, що допомагає економісту краще орієнтуватись у майбутніх прибутках підприємства. При доставці або відправці продукції програмне забезпечення автоматично заводить інформацію про поповнення сховища або його зменшенні, зменшуючи навантаження на менеджера по складу.

Продаючи, закупаючи, зберігаючи та утилізуючи зіпсовану продукцію, програма збирає статистику та через деякий проміжок часу, коли назбирається необхідна кількість даних можна переглянути статистику, аналітичні діаграми за необхідні періоди часу в електронному вигляді у програмному застосунку. Менеджер з закупки та продажу може будувати стратегії до майбутньої закупівлі нової продукції за допомогою даних статистики про сезонні продажі та

закупляти продукцію, яка приносить найбільший прибуток у певний сезон. Менеджер по складу може орієнтуватись на статистику усохлих та зіпсованої продукції та даних про умови їх зберігання та робити висновки щодо майбутніх покращень умов зберігання продукції.

База даних MySQL розташована на комп'ютері системного адміністратора бази даних, який відповідає за її обслуговування та роботу. Комп'ютери менеджера із закупок та продажу, менеджера складу, менеджера з логістики, механіка, директора та бухгалтера об'єднані в локальну мережу для спільної роботи з єдиною базою даних MySQL. Взаємодія з базою здійснюється за допомогою SQL-запитів, які дозволяють додавати, видаляти, редагувати, шукати, сортувати та фільтрувати дані[8].

Для розробки діаграм обрано сервіс Diagrams.net.

Для розробки програмного забезпечення обрано мову програмування C# на платформі .NET Framework версії 8.0, а для написання коду використовуватиметься інтегроване середовище розробки Microsoft Visual Studio 2024 для написання коду.

Реляційну СКБД MySQL обрано для розробки бази даних веб-сайту та програмного забезпечення. Вона створюється програмно з використанням Microsoft Entity Framework Core версії 8.0, застосовуючи принципи об'єктно-орієнтованого програмування. Функціонування бази даних забезпечуватиметься на комп'ютері системного адміністратора бази даних, який здійснюватиме її обслуговування та моніторинг коректності її роботи.

Обрано операційну систему Windows 10 версії 22H2 для належного функціонування інформаційної системи.

Технічне функціонування інформаційної системи має забезпечуватись стаціонарними комп'ютерами у розмірі чотирьох штук, які будуть використовуватися менеджером із закупок та продажу, менеджером по складу, менеджером з логістики та адміністратором бази даних[9].

Комп'ютери мають бути оснащені процесором Intel Core i3 12-го покоління (або вище) із 4 ядрами та вбудованим графічним ядром (або без графічного, але

з використанням належної відеокарти), оперативною пам'яттю DDR4 об'ємом не менше 8 ГБ на основі материнської плати з сокетом 1700 і підтримкою DDR4. Материнська плата повинна мати HDMI або VGA вихід для підключення монітора. Кожен комп'ютер має бути обладнаний жорстким диском або SSD (для більш приємного та швидкого користування системою, на вибір) з об'ємом не менше 256 ГБ[10].

Для відображення зображення на монітор комп'ютери оснащуються відеокартами з об'ємом відеопам'яті не менше 4 ГБ типу DDR3 та розрядністю шини пам'яті 64 біт, які мають HDMI або VGA вихід для підключення. Установлення відеокарти потрібне лише в разі відсутності інтегрованого графічного ядра в процесорі[10].

Для забезпечення користування комп'ютерів необхідно на кожний комп'ютер мати такі пристрої виведення та введення інформації як RGB-монітор з HDMI або VGA на вхід, звичайні мишку та клавіатуру. Також комп'ютери мають забезпечуватись підключенням до мережі Інтернет.

На основі розробленої концепції виконано розробку проекту, що дозволить підвищити ефективність маркетингової стратегії овочесховища та вдосконалить внутрішні процеси підприємства, що й відповідає поставленій меті дипломної роботи.

2.2. Моделювання та конструювання

2.2.1. Розробка діаграми варіантів використання

Для відображення взаємодії функціональних можливостей системи створено діаграму варіантів використання (use-case діаграму). Вона є графічним інструментом для опису функціональності системи з точки зору її взаємодії з користувачами (акторами) та є важливим інструментом аналізу та проектування систем у рамках методології Unified Modeling Language (UML). Діаграма відображає, які дії (функції) може виконувати система та як з нею взаємодіють зовнішні учасники. Use-case діаграма допомагає зосередитись на вимогах до

системи на високому рівні абстракції та полегшує розуміння структури системи[11].

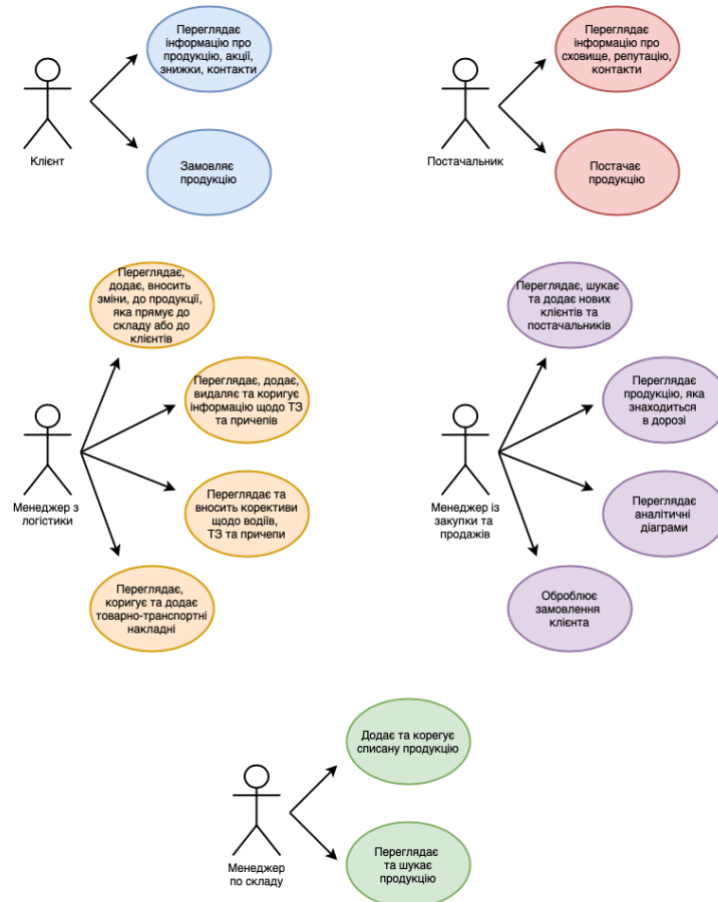


Рис. 2.2. Use-case діаграма інформаційної системи овочесховища [створено автором]

Use-case діаграма має наступні основні елементи:

1. Актори. Сутності, які взаємодіють із системою. Це можуть бути люди, інші системи або пристрої. Актори зображаються у вигляді піктограми людини.
2. Варіанти використання (прецеденти). Описують функції або дії, які система виконує у відповідь на запити акторів. Варіанти використання зображаються у вигляді овала з назвою дії всередині.
3. Відношення між акторами та функціями. Вказують на взаємодію між акторами та варіантами використання або між самими варіантами. Відношення зображаються у вигляді стрілок, вказуючи напрям від того, хто ініціює взаємодію, до функції, яка виконується.

Опис use-case діаграми [створено автором]

Актори	Прецеденти	Опис
Клієнт	Переглядає інформацію про продукцію, акції, знижки, контакти	Клієнт має змогу ознайомитись з необхідною йому інформацією про овочесховище, репутацію, персонал, контакти, наявну у продажі продукцію, акції та знижки
	Замовляє продукцію	Клієнт може замовити цікавлячу його продукцію через контакти менеджера із продажу та закупки, попередньо ознайомившись з наявною продукцією на сайті підприємства
Постачальник	Переглядає інформацію про сховище, репутацію, контакти	Постачальник має змогу ознайомитись з необхідною йому інформацією про овочесховище, репутацію, персонал та контакти для співробітництва
	Постачає продукцію	Постачальник може поставити наявну в нього продукцію попередньо зв'язавшись із менеджером із продажу та закупки та обговоривши всі деталі співробітництва
Менеджер логістики з	Переглядає, коригує та додає товарно-транспортні накладні	Менеджер з логістики має змогу створювати, переглядати, корегувати та виконувати пошук цікавих йому товарно-транспортних накладних у базі даних
	Переглядає, додає, вносить зміни, до продукції, яка прямує до складу або до клієнтів	Менеджер з логістики має змогу створювати, переглядати, корегувати та виконувати пошук поточних замовлень, які прямують до складу овочесховища або до клієнтів у базі даних
	Переглядає, додає, видаляє та коригує інформацію щодо ТЗ	Менеджер з логістики має змогу створювати, переглядати, корегувати, видаляти та виконувати пошук інформації у базі даних щодо наявних у підприємства транспортних засобів для подальшого внесення їх даних до товарно-транспортних накладних
	Переглядає та вносить корективи щодо водіїв, ТЗ та вантажників	Менеджер з логістики має змогу переглядати, корегувати та виконувати пошук інформації у базі

Продовження таблиці 2.1

		даних щодо наявних у підприємства транспортних засобів, водіїв та вантажників для подальшого внесення їх даних до товарно-транспортних накладних
Менеджер із закупки та продажів	Переглядає, шукає та додає нових клієнтів та постачальників	Менеджер із закупки та продажів має змогу додавати, переглядати, корегувати, видаляти та виконувати пошук інформації щодо постійних клієнтів та постачальників
	Переглядає продукцію, яка знаходиться в дорозі	Менеджер із закупки та продажів має змогу переглядати інформацію щодо замовленої продукції, яка прямує на склад або до клієнта та в подальшому надавати клієнтам та постачальникам інформацію щодо їх вантажу
	Переглядає аналітичні діаграми	Менеджер із закупки та продажів має змогу переглядати аналітичні діаграми, які розробляє база даних за рахунок наявної у ній інформації про продукцію, закупки тощо
	Обробляє замовлення клієнта	Менеджер із закупки та продажів обробляє замовлення клієнта, переглядаючи наявну у сховищі продукцію та надаючи консультацію на цікавлячу клієнта продукцію та пропонує акційну продукцію та продукцію по знижкам, швидко беручи інформацію із бази даних
Менеджер по складу	Переглядає та шукає продукцію	Менеджер по складу, має змогу переглядати та виконувати пошук наявної на складі продукції у базі даних
	Додає та корегує утилізовану продукцію	Менеджер по складу, має змогу додавати, переглядати, корегувати та виконувати пошук утилізованої зіпсованої продукції у базі даних

2.2.2. Розробка концептуальної моделі даних

Розробка концептуальної моделі даних є першим кроком до розробки бази даних (рис 1.5). Насамперед, концептуальна модель даних є високорівневим представленням даних, яке використовується для опису основних об'єктів, понять та їх взаємозв'язків у певній предметній області. Вона відображає логічну структуру інформації, яку потрібно зберігати, але не враховує технічні деталі реалізації бази даних. Основний акцент у концептуальній моделі робиться на тому, що зберігати, а не як це реалізувати[12]. Модель має створюватися у зрозумілому форматі, щоб бізнес-аналітики, розробники та замовники могли спільно обговорювати вимоги. Також, у подальшій розробці дає змогу уникнути помилок у проєктуванні бази даних, забезпечуючи чітке розуміння структури даних і зв'язків між ними.

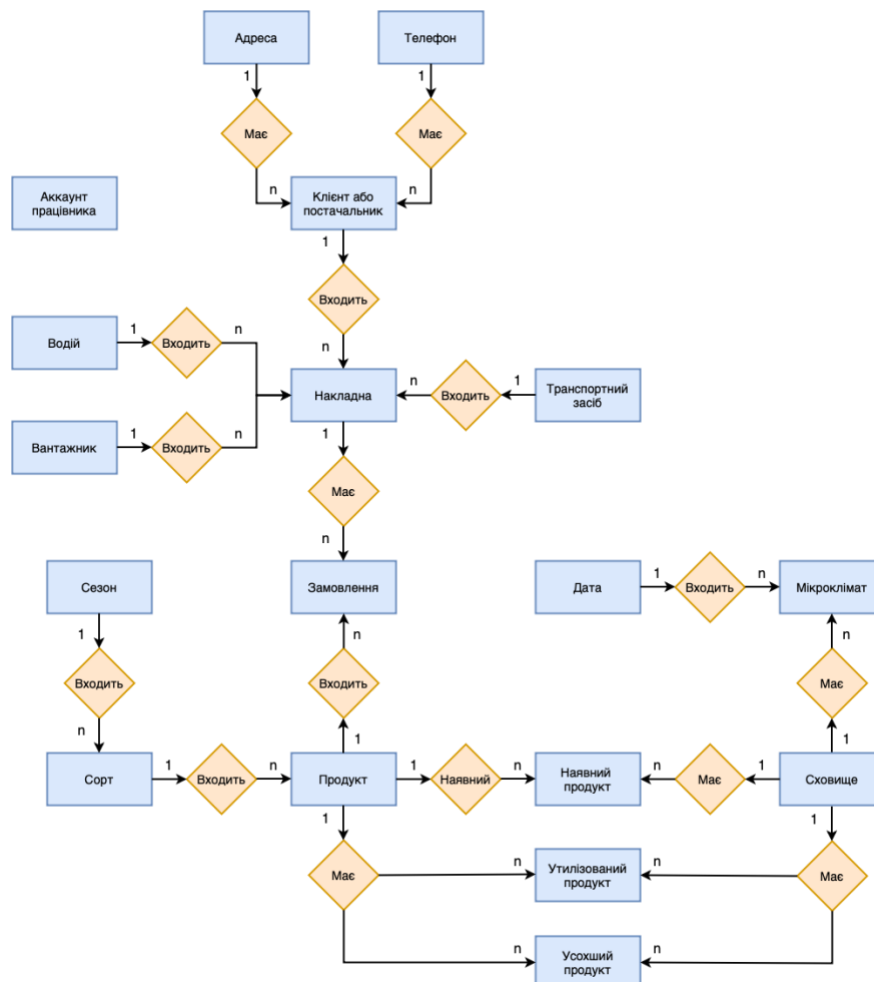


Рис. 2.3. Концептуальна модель даних інформаційної системи овочесховища
[створено автором]

Дослідивши предметну область, виявлено сутності: адреса, телефон, клієнт або постачальник, водій, вантажник, транспортний засіб, накладна, замовлення, продукт, сезон, сорт, наявний продукт, утилізований продукт, усохший продукт, сховище, мікроклімат, дата, аккаунт працівника.

Короткий опис зв'язків між сутностями.

«Адреса клієнта або постачальника»-«Клієнт або постачальник» - 1:Б – обов'язковий-обов'язковий. Одному клієнту або постачальнику може мати одну або більше адрес. Одна або більше адрес мають одного клієнта або постачальника. Видалення – каскадне. Оновлення – каскадне.

«Телефон клієнта або постачальника»-«Клієнт або постачальник» - 1:Б – обов'язковий-обов'язковий. Одному клієнту або постачальнику може відповідати один або більше номерів телефонів. Один або більше номерів телефонів належить до одного клієнта або постачальника. Видалення – каскадне. Оновлення – каскадне.

«Клієнт або постачальник»-«Накладна» - 1:Б – обов'язковий-обов'язковий. Одному клієнту або постачальнику може входити до однієї або більше товарно-транспортних накладних. Одна або більше товарно-транспортних накладних належать до одного клієнта або постачальника. Видалення – каскадне. Оновлення – каскадне.

«Водій»-«Накладна» - 1:Б – обов'язковий-обов'язковий. Один водій може входити до однієї або більше товарно-транспортних накладних. Одна або більше товарно-транспортних накладних мають одного водія. Видалення – каскадне. Оновлення – каскадне.

«Вантажник»-«Накладна» - 1:Б – обов'язковий-обов'язковий. Один вантажник може входити до однієї або більше товарно-транспортних накладних. Одна або більше товарно-транспортних накладних мають одного вантажника. Видалення – каскадне. Оновлення – каскадне.

«Транспортний засіб»-«Накладна» - 1:Б – обов'язковий-обов'язковий. Один транспортний засіб може входити до однієї або більше товарно-

транспортних накладних. Одна або більше товарно-транспортних накладних мають один транспортний засіб. Видалення – каскадне. Оновлення – каскадне.

«Накладна»-«Замовлення» - 1:Б – обов’язковий-обов’язковий. Одна товарно-транспортна накладна має одне або більше замовлень. Одне або більше замовлень наявні в одній товарно-транспортній накладній. Видалення – каскадне. Оновлення – каскадне.

«Продукт»-«Замовлення» - 1:Б – обов’язковий-обов’язковий. Один продукт може входити до одного або більше замовлень. Одне або більше замовлень мають один продукт. Видалення – каскадне. Оновлення – каскадне.

«Сезон»-«Сорт» - 1:Б – обов’язковий-обов’язковий. Один сезон може входити до одного або більше сортів. Один або більше сорт має один сезон. Видалення – каскадне. Оновлення – каскадне.

«Сорт»-«Продукт» - 1:Б – обов’язковий-обов’язковий. Один сорт може входити до одного або більше продуктів. Один або більше продукт має один сорт. Видалення – каскадне. Оновлення – каскадне.

«Продукт»-«Наявний продукт» - 1:Б – обов’язковий-обов’язковий. Один продукт може мати один або більше однакових видів продукції. Один або більше однакових видів продукції належать до одного продукту. Видалення – каскадне. Оновлення – каскадне.

«Продукт»-«Утилізований продукт» - 1:Б – обов’язковий-обов’язковий. Один продукт може мати до одну або більше утилізованої продукції. Одна або більше утилізованої продукції належить до одного продукту. Видалення – каскадне. Оновлення – каскадне.

«Продукт»-«Усохший продукт» - 1:Б – обов’язковий-обов’язковий. Один продукт може мати до одну або більше усохшої продукції. Одна або більше усохшої продукції належить до одного продукту. Видалення – каскадне. Оновлення – каскадне.

«Сховище»-«Наявний продукт» - 1:Б – обов’язковий-обов’язковий. Одне сховище може мати один або більше продукції одного виду. Один або більше

видів продукції мають одне сховище. Видалення – каскадне. Оновлення – каскадне.

«Сховище»-«Мікроклімат» - 1:Б – обов’язковий-обов’язковий. Одне сховище може мати до один або більше мікрокліматів сховища. Один або більше мікрокліматів сховища мають одне сховище. Видалення – каскадне. Оновлення – каскадне.

«Мікроклімат»-«Дати» - 1:Б – обов’язковий-обов’язковий. Один мікроклімат сховища може мати до одну або більше дат. Одна або більше дат мають один мікроклімат сховища. Видалення – каскадне. Оновлення – каскадне.

«Аккаунт працівника». Кожен працівник, який входить до інформаційної системи овочесховища має свій аккаунт для входу в систему.

2.2.3. Розробка діаграми станів

Діаграма станів (рис. 2.6) є одним із типів діаграм в методології Unified Modeling Language (UML), яка моделює поведінку об’єкта або системи, показуючи всі можливі стани, у яких може перебувати об’єкт, а також події, що викликають перехід між цими станами, ґрунтуючись на теорії автоматів.

Ця діаграма використовується для візуалізації змін об’єкта протягом його життєвого циклу залежно від дій або впливів зовнішнього середовища. Використовується для опису, як об’єкт або система реагує на зовнішні події, переходячи з одного стану в інший. Діаграма станів допомагає чітко зрозуміти і описати життєвий цикл об’єкта, що особливо важливо для складних систем із багатьма можливими станами і є потужний інструментом для візуалізації поведінки систем або об’єктів у залежності від подій і дій[13].

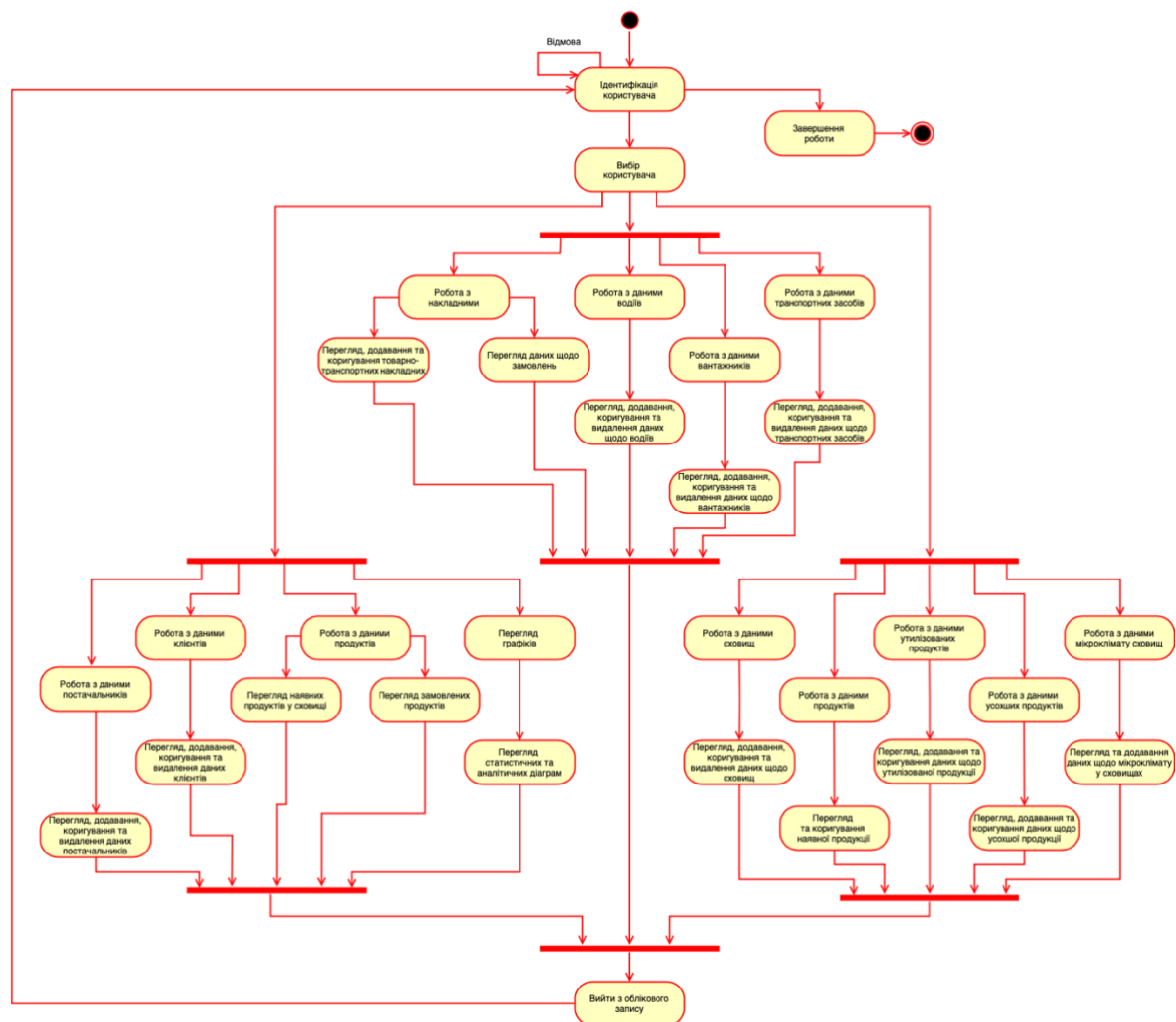


Рис. 2.4. Діаграма станів [створено автором]

2.3. Технічний проект

2.3.1. Побудова логічної моделі даних

Розробка логічної моделі даних є проміжним етапом між розробкою концептуальної та фізичної моделей даних. Логічна модель даних є абстрактним представленням структури даних, яке описує, що дані є і яким чином вони пов'язані між собою, при цьому неприв'язуючись до конкретної системи управління базами даних[14]. Дана модель детально описує сутності (таблиці), їх атрибути (поля) та взаємозв'язки між сутностями (рис 2.4).

Виділивши основні сутності з концептуальної моделі, встановим ключові характеристики кожної сутності, визначивши назву полів, їх типи даних, ключі та визначимо типи зв'язків.

Сутності та поля логічної моделі даних наведено у таблицях 2.2-2.19

Таблиця 2.2

Адреса [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Код клієнта або постачальника	Числовий	Зовнішній ключ
Адреса	Текстовий	Нема

Таблиця 2.3

Телефон [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Код клієнта або постачальника	Числовий	Зовнішній ключ
Телефон	Текстовий	Нема
Опис	Текстовий	Нема

Таблиця 2.4

Клієнт або постачальник [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Особа	Текстовий	Нема
Прізвище	Текстовий	Нема
Ім'я	Текстовий	Нема
По-батькові	Текстовий	Нема
Логін	Текстовий	Нема
Пошта	Текстовий	Нема
Пароль	Текстовий	Нема
Опис	Текстовий	Нема

Таблиця 2.5

Накладна [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Дата	Дата	Нема
Код клієнта або постачальника	Числовий	Зовнішній ключ
Код транспортного засобу	Числовий	Зовнішній ключ
Код водія	Числовий	Зовнішній ключ
Адреса клієнта або постачальника	Текстовий	Нема
Загальна маса продукції	Числовий	Нема
Загальна сума продукції з ПДВ	Числовий	Нема
Сума ПДВ	Числовий	Нема
Код вантажника	Числовий	Зовнішній ключ
Час відбуття	Час	Нема
Час прибуття	Час	Нема
Час простою	Час	Нема
Статус	Текстовий	Нема

Таблиця 2.6

Водій [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Прізвище	Текстовий	Нема
Ім'я	Текстовий	Нема
По-батькові	Текстовий	Нема
Телефон	Текстовий	Нема

Таблиця 2.7

Вантажник [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Прізвище	Текстовий	Нема
Ім'я	Текстовий	Нема
По-батькові	Текстовий	Нема
Телефон	Текстовий	Нема

Таблиця 2.8

Транспортний засіб [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Тип	Текстовий	Нема
Марка	Текстовий	Нема
Модель	Текстовий	Нема
Реєстраційний номер	Текстовий	Нема
Колір	Текстовий	Нема
Довжина	Числовий	Нема
Ширина	Числовий	Нема
Висота	Числовий	Нема
Маса	Числовий	Нема

Таблиця 2.9

Замовлення [створено автором]

Назва поля	Тип даних	Ключ
Код накладної	Числовий	Первинний ключ, Автоматичне збільшення
Код продукту	Числовий	Зовнішній ключ
Кількість	Числовий	Нема
Ціна за одиницю	Числовий	Нема

Продовження таблиці 2.9

Загальна сума з ПДВ	Числовий	Нема
Маса	Числовий	Нема

Таблиця 2.10

Продукт [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Продукт	Текстовий	Нема
Код сорту	Числовий	Зовнішній ключ
Одиниця виміру	Текстовий	Нема
Опис	Текстовий	Нема

Таблиця 2.11

Сорт [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Сорт	Текстовий	Нема
Код сезону	Числовий	Зовнішній ключ

Таблиця 2.12

Сезон [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Сезон	Текстовий	Нема

Таблиця 2.13

Наявний продукт [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Код сховища	Числовий	Зовнішній ключ
Код продукту	Числовий	Зовнішній ключ

Продовження таблиці 2.13

Кількість	Числовий	Нема
Ціна	Числовий	Нема
Маса	Числовий	Нема
Дата завозу	Дата	Нема
Дата кінця зберігання	Дата	Нема

Таблиця 2.14

Утилізований продукт [створено автором]

Назва поля	Тип даних	Ключ
Код сховища	Числовий	Зовнішній ключ
Код продукту	Числовий	Зовнішній ключ
Кількість	Числовий	Нема
Маса	Числовий	Нема
Дата	Дата	Нема

Таблиця 2.15

Усохший продукт [створено автором]

Назва поля	Тип даних	Ключ
Код сховища	Числовий	Зовнішній ключ
Код продукту	Числовий	Зовнішній ключ
Відсоток усихання	Числовий	Нема

Таблиця 2.16

Сховище [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Тип	Текстовий	Нема
Опис	Текстовий	Нема

Таблиця 2.17

Мікроклімат [створено автором]

Назва поля	Тип даних	Ключ
Код дати	Числовий	Зовнішній ключ
Код сховища	Числовий	Зовнішній ключ
Середня температура за день	Числовий	Нема
Середня вологість за день	Числовий	Нема

Таблиця 2.18

Дата [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Дата для мікроклімату	Дата	Нема

Таблиця 2.19

Аккаунт працівника [створено автором]

Назва поля	Тип даних	Ключ
Код	Числовий	Первинний ключ, Автоматичне збільшення
Логін	Текстовий	Нема
Пароль	Текстовий	Нема
Прізвище	Текстовий	Нема
Ім'я	Текстовий	Нема
По-батькові	Текстовий	Нема
Спеціальне поле	Текстовий	Нема

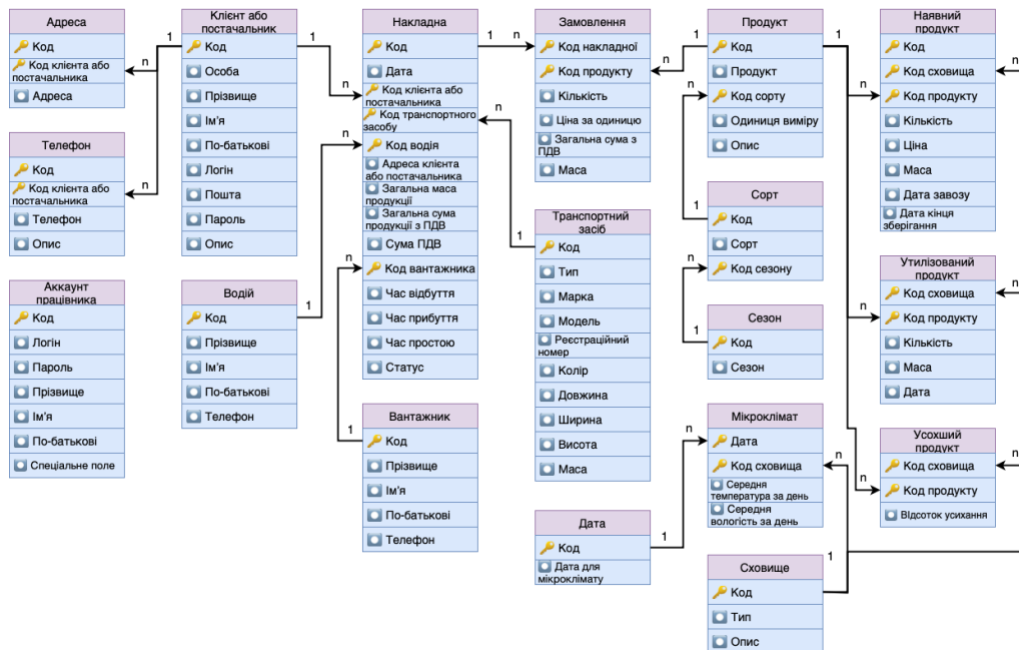


Рис. 2.5. Логічна модель даних інформаційної системи овочесховища [створено автором]

Розроблена логічна модель дозволяє детально зрозуміти структуру даних і зв'язки між ними, не враховуючи технічних обмежень, організовуючи дані в зрозумілу і логічну форму та допомагаючи уникнути дублювання, неоднозначностей і конфліктів у базі даних, що критично шкодить базі даних. Наступним кроком розробки бази даних інформаційної системи буде перетворення логічної моделі в фізичну модель даних, яка враховує специфіку обраної СУБД.

2.3.2. Визначення складу модулів програми

Архітектура програмного забезпечення — це структурний дизайн системи, який визначає, як її компоненти взаємодіють, взаємопов'язані та об'єднуються для виконання певних функцій. Це високорівнева концепція, що враховує всі аспекти роботи системи: функціональність, надійність, продуктивність, безпека, масштабованість. Архітектура є основою для створення програмного забезпечення, забезпечуючи план, якого дотримуються під час розробки, впровадження та підтримки.

Структурний дизайн архітектури програми організований в окремі модулі, які формуються на основі раніше розробленої діаграми варіантів використання. Кожен модуль відображає функції, які виконують актори в системі (менеджер із закупки та продажів, менеджер з логістики, менеджер по складу). Керування модулями здійснюється через головний модуль, який виконує ідентифікацію користувачів і надає їм доступ до відповідного модуля в системі[15].

Перелік модулів: модуль ідентифікації користувача, модуль роботи з постачальниками, модуль роботи з клієнтами, модуль роботи з продуктами, модуль роботи з графіками, модуль роботи з товарно-транспортними накладними, модуль роботи з водіями, модуль роботи з вантажниками, модуль роботи з транспортними засобами, модуль роботи зі сховищами, модуль роботи з утилізованими продуктами, модуль роботи з усохшими продуктами, модуль роботи з мікрокліматом сховищ.

2.3.3. Специфікація модулів програми

Визначивши архітектурні модулі програми, перейдемо до опису їх специфікації.

Модуль ідентифікації користувача

Модуль призначений для ідентифікації користувача в системі та його подальшого доступу.

Вхідні дані: логін та пароль користувача.

Дія: перевірка коректності логіну та паролю користувача.

Вихідні дані: повідомлення про помилку при вводі ідентифікації користувача або надання користувачу доступу до робочого місця при правильності введення ідентифікаційних даних.

Модуль роботи з постачальниками

Модуль призначений для перегляду, додавання, видалення та корегування інформації щодо постачальників.

Вхідні дані: необхідна інформація щодо постачальника.

Дія: перегляд, додавання нового постачальника до таблиці, видалення з таблиці або редагування інформації щодо постачальників.

Вихідні дані: інформація про постачальників.

Модуль роботи з клієнтами

Модуль призначений для перегляду, додавання, видалення та корегування інформації щодо клієнтів.

Вхідні дані: необхідна інформація щодо клієнта.

Дія: перегляд, додавання нового клієнта до таблиці, видалення з таблиці або редагування інформації щодо клієнтів.

Вихідні дані: інформація про клієнтів.

Модуль роботи з продуктами

Модуль призначений для перегляду наявних продуктів у сховищі та замовлених клієнтом або направлених постачальником.

Вхідні дані: дані, які обирає користувач для пошуку.

Дія: перегляд інформації щодо продукції

Вихідні дані: інформація про наявну продукцію та інформація про продукцію, яка замовлена клієнтом або направлена постачальником.

Модуль роботи з графіками

Модуль призначений для перегляду графіків продажів в залежності від овочу, його сорту та сезону закупки.

Вхідні дані: інформація щодо продажу овочів.

Дія: побудова графіків.

Вихідні дані: графіки продажу продукції певного сорту та на певний сезон.

Модуль роботи з товарно-транспортними накладними

Модуль призначений для перегляду, додавання та корегування товарно-транспортними накладними.

Вхідні дані: інформація щодо замовлення.

Дія: перегляд, додавання, корегування накладних.

Вихідні дані: інформація щодо накладної та замовлення.

Модуль роботи з водіями

Модуль призначений для перегляду, додавання, видалення та корегування інформації щодо водіїв.

Вхідні дані: необхідна інформація щодо водія.

Дія: перегляд, додавання нового водія до таблиці, видалення з таблиці або редагування інформації щодо водіїв.

Вихідні дані: інформація про водіїв.

Модуль роботи з вантажниками

Модуль призначений для перегляду, додавання, видалення та корегування інформації щодо вантажників.

Вхідні дані: необхідна інформація щодо вантажника.

Дія: перегляд, додавання нового вантажника до таблиці, видалення з таблиці або редагування інформації щодо вантажників.

Вихідні дані: інформація про вантажників.

Модуль роботи з транспортними засобами

Модуль призначений для перегляду, додавання, видалення та корегування інформації щодо транспортних засобів.

Вхідні дані: необхідна інформація щодо транспортного засобу.

Дія: перегляд, додавання нового транспортного засобу до таблиці, видалення з таблиці або редагування інформації щодо транспортних засобів.

Вихідні дані: інформація про транспортні засоби.

Модуль роботи зі сховищами

Модуль призначений для перегляду, додавання, видалення та корегування інформації щодо сховищ.

Вхідні дані: необхідна інформація щодо сховища.

Дія: перегляд, додавання нового сховища до таблиці, видалення з таблиці або редагування інформації щодо сховищ.

Вихідні дані: інформація про сховища.

Модуль роботи з утилізованими продуктами

Модуль призначений для перегляду, додавання та корегування інформації щодо утилізованих продуктів.

Вхідні дані: необхідна інформація щодо утилізованого продукту.

Дія: перегляд, додавання нового утилізованого продукту до таблиці або редагування інформації щодо утилізованих продуктів.

Вихідні дані: інформація про утилізовані продукти.

Модуль роботи з усохшими продуктами

Модуль призначений для перегляду, додавання та корегування інформації щодо усохших продуктів.

Вхідні дані: необхідна інформація щодо усохшого продукту.

Дія: перегляд, додавання нового усохшого продукту до таблиці або редагування інформації щодо усохших продуктів.

Вихідні дані: інформація про усохші продукти.

Модуль роботи з мікрокліматом сховищ

Модуль призначений для перегляду та додавання та інформації щодо мікроклімату сховищ.

Вхідні дані: необхідна інформація щодо мікроклімату сховищ.

Дія: перегляд, додавання запису про мікроклімат сховища на певний день до таблиці.

Вихідні дані: інформація про мікроклімат сховища на певний день.

2.3.4. Розробка алгоритму роботи програми

Алгоритм — це впорядкована та кінцева послідовність дій, яка визначає процес розв'язання задачі або досягнення певного результату. Алгоритми використовуються у програмуванні для створення послідовності дій, яка допомагає вирішити поставлену задачу за допомогою комп'ютера. Він є основою програмного коду, визначаючи, як обробляти дані та досягати бажаного результату[16]. Алгоритми у програмуванні важливі, бо правильно побудований алгоритм дає змогу точно визначити направлення роботи, зменшує витрати ресурсів, підвищуючи швидкість роботи програми, та розбиває складні задачі в малі під задачі, які легше зрозуміти та вирішити.

Основні характеристики алгоритму:

- кожна дія має бути точно визначена;
- алгоритм повинен завершуватися за скінченний час;
- виконання алгоритму призводить до отримання результату;
- алгоритм може мати вихідні дані, які потрібні для його виконання;
- кожен крок може бути чітко визначений або мати кілька варіантів виконання залежно від умов.

Блок-схема — це графічне подання алгоритму у вигляді послідовності блоків, що з'єднані між собою стрілками, які вказують напрямок виконання. Кожен блок представляє певну операцію або дію. Візуальне зображення процесів алгоритму спрощує розуміння алгоритму, забезпечуючи зрозумілий і структурований спосіб аналізу в цілому. Більш того, блок-схеми спрощують планування додаткових процесів та спрощують пошук помилок та оптимізації процесів.

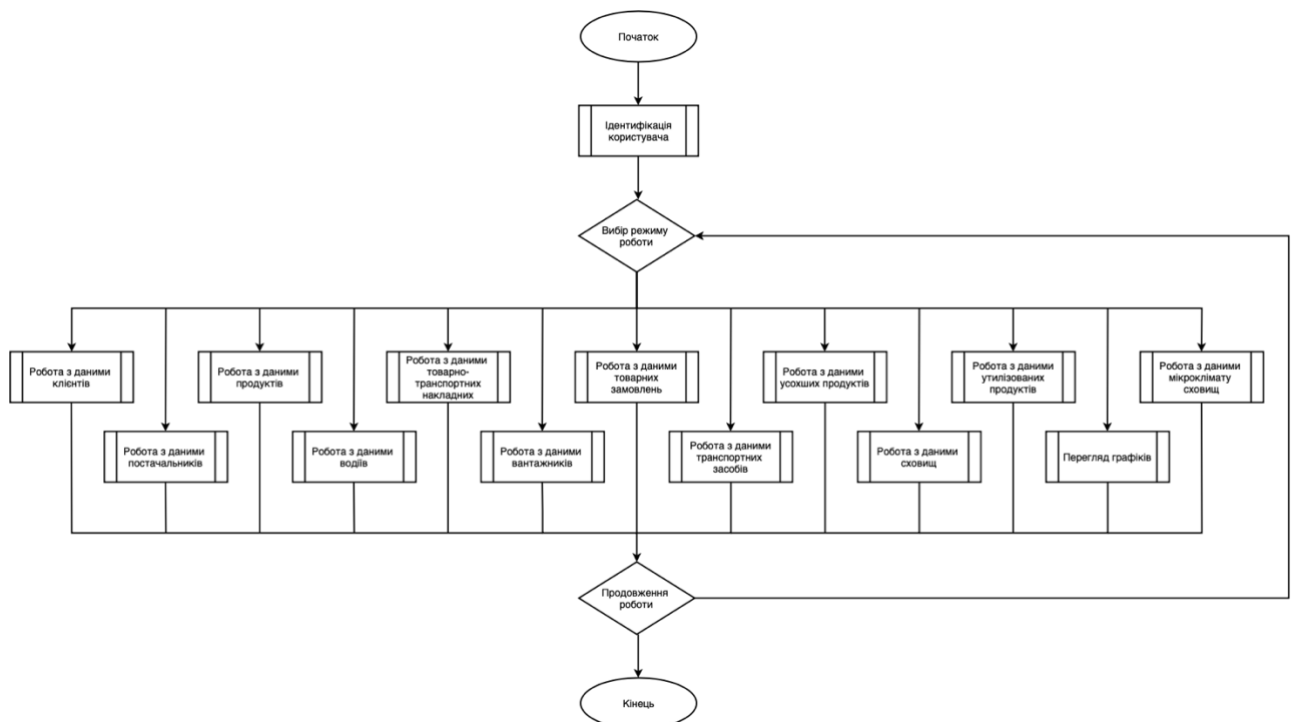


Рис. 2.6. Алгоритм роботи програми [створено автором]

2.4. Робочий проект

2.4.1. Вибір засобів розробки програмного забезпечення

Вибір інструментів розробки є важливим етапом життєвого циклу розробки програмного забезпечення, оскільки він значно впливає на успіх, ефективність і масштабованість проекту, тому для цього етапу треба підійти надзвичайно відповідально.

Середовище розробки Visual Studio 2024 — це інтегроване середовище розробки (IDE) від Microsoft, яке пропонує передові інструменти для програмування, тестування та розгортання програмного забезпечення. Воно підтримує різні мови програмування, включаючи C#, C++, Python, JavaScript та багато інших, забезпечуючи широкий спектр функцій для розробників[17].

Розглянемо переваги використання програмного середовища Visual Studio 2024.

Штучний інтелект і автоматизація. Visual Studio інтегрує GitHub Copilot, що забезпечує автоматичне створення коду, рефакторинг, і підказки, адаптовані до контексту проекту. Це значно підвищує ефективність роботи розробників, особливо при роботі з великими проектами або спадковим кодом.

Надзвичайна продуктивність. Середовище розробки оптимізоване для роботи з великими проектами, знижуючи час завантаження і компіляції. Інструменти для віддаленого налагодження дозволяють працювати з хмарними серверами чи контейнерами без втрати продуктивності.

Чудова робота з багатофайловими (багатокласовими) проектами. Оновлені функції для аналізу та рефакторингу коду значно спрощують підтримку великих проектів.

Інтуїтивний інтерфейс. Обширний користувацький досвід з додатковими кастомізаціями, зокрема теми, розташування панелей, і гарячі клавіші.

Visual Studio 2024 є ідеальним середовищем для розробки інформаційної системи овочесховища, яке має найновітніші технології для створення складних програмних рішень.

Мова програмування C# (C-Sharp) є чудовим вибором для розробки програмного забезпечення завдяки її потужності, універсальності та інтеграції з сучасними технологіями. Це мова загального призначення, розроблена Microsoft, яка поєднує простоту синтаксису, схожу на Java та C++, із широкими можливостями для створення складних програмних систем. Ось детальний аналіз причин, чому C# є чудовим вибором для розробників[18].

Розглянемо переваги використання мови C#.

Простота та зручність. Чіткий синтаксис мови C# є інтуїтивно зрозумілим, що полегшує написання коду та підвищує швидкість його написання. Наявне автоматичне управління пам'яттю, що мінімізує ризик помилок, пов'язаних із ручним видаленням об'єктів. Підтримує об'єктно-орієнтовне програмування, що дозволяє легко організувати код у вигляді модулів, класів та об'єктів.

Інтеграція з .NET Framework. Мова C# тісно інтегрована з платформами .NET Framework і .NET Core, що забезпечує доступ до величезної кількості бібліотек і функцій для роботи з базами даних, мережею, інтерфейсами користувача, хмарними технологіями тощо.

Універсальність застосування. Дана мова використовується для створення додатків з графічним інтерфейсом на основі Windows Presentation Foundation (WPF) або Windows Forms, використовується для створення веб-додатків, мобільної розробки, хмарні додатки тощо.

Потужні можливості мови. Мова C# є строго типізованою мовою, що мінімізує ризик помилок типів під час виконання. Наявність лямбда-виразів та LINQ дозволяє легко працювати з колекціями даних, роблячи їх фільтрацію та трансформацію максимально простою. Мова також має вбудовану підтримку асинхронного програмування (async/await), що полегшує створення асинхронних додатків для підвищення продуктивності.

Велика спільнота та підтримка. Компанія Microsoft пропонує велику кількість ресурсів, документації та навчальних матеріалів для C#, що завжди стане у пригоді при написанні програмного забезпечення. Також мова C# має

активну глобальну спільноту, яка допомагає вирішувати проблеми та вдосконалювати мову.

Таким чином мова C# ідеально підходить для створення різних видів програмних додатків, а його багатофункціональність, інтеграція з .NET та широкі можливості роблять його потужним інструментом, що надає усі необхідні умови для створення програмного додатка для інформаційної системи.

База даних MySQL є одним із найпопулярніших виборів для створення баз даних завдяки її продуктивності, надійності та гнучкості. Вона підходить для широкого спектра проектів — від невеликих додатків до масштабних корпоративних систем[19].

Розглянемо переваги використання бази даних MySQL.

Простота у використанні. База даних MySQL має простий процес налаштування, що робить її зрозумілою навіть для початківців, а додаток MySQL Workbench надає графічний інтерфейс для управління базами даних, проектування та написання запитів.

Висока швидкість та продуктивність. База даних MySQL оптимізована для виконання запитів, тому вона ідеально підходить для додатків, що вимагають високої швидкості обробки даних. Через велику масштабованість, база підтримує великі обсяги даних, що робить її придатною для масштабних програм, веб-додатків і хмарних сервісів.

Кросплатформність. MySQL працює на різних операційних системах, включаючи Windows, macOS, Linux та Unix, що забезпечує гнучкість у виборі середовища розробки.

Гнучкість і сумісність. База даних MySQL працює з широким спектром типів даних, включаючи текстові, числові, часові та двійкові та має сумісність з обширним переліком мов програмування (C#, Java, Python, PHP, Ruby тощо).

Безпека. База MySQL пропонує потужні функції безпеки, включаючи багаторівневу систему доступу та шифрування даних.

Велика спільнота. База даних MySQL має широку спільноту користувачів і розробників, через що початковому та професійному розробнику легко знайти

рішення для більшості технічних питань. MySQL також має офіційну документацію, яка регулярно оновлюється та доступна кількома мовами.

База даних MySQL є чудовим вибором для створення бази даних завдяки її продуктивності, простоті використання та доступності. Вона чудово підійде для створення бази даних для інформаційної системи овочесховища у поєднанні з програмним середовищем Visual Studio 2024 та програмною мовою C#.

2.4.2. Розробка фізичної моделі даних

Фізична модель даних є детальним представленням структури бази даних, яке визначає, як саме дані зберігатимуться у конкретній системі управління базами даних. Вона містить всі технічні деталі, такі як таблиці, стовпці, типи даних, зв'язки між таблицями, обмеження цілісності, а також оптимізацію для конкретного середовища. На відміну від концептуальної та логічної моделей, фізична модель є безпосередньою специфікацією, яка реалізується в базі даних[20].

Для створення фізичної моделі даних треба провести аналіз розробленої логічної моделі даних, визначити фінальну назву поля, визначити його оптимальний тип та довжину, а також визначити обов'язковість поля (NULL або Not NULL).

Фізична модель даних наведена у таблицях 2.20-2.37

Таблиця 2.20

Адреса (Addresses) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Код клієнта або постачальника	CoSID	INT	Not NULL
Адреса	Address	VARCHAR(70)	Not NULL

Таблиця 2.21

Телефон (Phones) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Код клієнта або постачальника	CoSID	INT	Not NULL
Телефон	Phone	VARCHAR(13)	Not NULL
Опис	Description	TINYTEXT	NULL

Таблиця 2.22

Клієнт або постачальник (ClientsOrSuppliers) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Особа	Person	VARCHAR(30)	Not NULL
Прізвище	LastName	VARCHAR(30)	Not NULL
Ім'я	FirstName	VARCHAR(30)	Not NULL
По-батькові	Patronymic	VARCHAR(30)	NULL
Логін	Login	VARCHAR(20)	NULL
Пошта	Mail	VARCHAR(30)	NULL
Пароль	Password	VARCHAR(30)	NULL
Опис	Description	TINYTEXT	NULL

Таблиця 2.23

Накладна (Waybills) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Дата	Date	DATE	Not NULL
Код клієнта або постачальника	CoSID	INT	Not NULL
Код транспортного засобу	VehicleID	INT	Not NULL
Код водія	DriverID	INT	Not NULL

Продовження таблиці 2.23

Адреса клієнта або постачальника	CoSAddress	VARCHAR(70)	Not NULL
Загальна маса продукції	TotalMass	DOUBLE	Not NULL
Загальна сума продукції з ПДВ	TotalSumWithVAT	DECIMAL	Not NULL
Сума ПДВ	VAT	DECIMAL	Not NULL
Код вантажника	LoaderID	INT	Not NULL
Час відбуття	LeavingTime	TIME	Not NULL
Час прибуття	ArrivingTime	TIME	Not NULL
Час простою	StandingTime	TIME	Not NULL
Статус	Status	VARCHAR(10)	Not NULL

Таблиця 2.24

Водій (Drivers) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Прізвище	LastName	VARCHAR(30)	Not NULL
Ім'я	FirstName	VARCHAR(30)	Not NULL
По-батькові	Patronymic	VARCHAR(30)	Not NULL
Телефон	Phone	VARCHAR(13)	Not NULL

Таблиця 2.25

Вантажник (Loaders) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Прізвище	LastName	VARCHAR(30)	Not NULL
Ім'я	FirstName	VARCHAR(30)	Not NULL
По-батькові	Patronymic	VARCHAR(30)	Not NULL
Телефон	Phone	VARCHAR(13)	Not NULL

Таблиця 2.26

Транспортний засіб (Vehicles) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Тип	Type	VARCHAR(20)	Not NULL
Марка	Brend	VARCHAR(30)	Not NULL
Модель	Model	VARCHAR(30)	Not NULL
Реєстраційний номер	RegNum	VARCHAR(8)	Not NULL
Колір	Color	VARCHAR(20)	Not NULL
Довжина	Length	DOUBLE	Not NULL
Ширина	Width	DOUBLE	Not NULL
Висота	Height	DOUBLE	Not NULL
Маса	Mass	DOUBLE	Not NULL

Таблиця 2.27

Замовлення (Orders) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код накладної	WaybillID	INT	Not NULL
Код продукту	ProductID	INT	Not NULL
Кількість	Amount	DOUBLE	Not NULL
Ціна за одиницю	PriceForPiece	DECIMAL	Not NULL
Загальна сума з ПДВ	TotalSumWithVAT	DECIMAL	Not NULL
Маса	Mass	DOUBLE	Not NULL

Таблиця 2.28

Продукт (Products) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Продукт	Product	VARCHAR(30)	Not NULL
Код сорту	VarietyID	INT	Not NULL

Продовження таблиці 2.28

Одиниця виміру	MeasureUnit	VARCHAR(5)	Not NULL
Опис	Description	TINYTEXT	NULL

Таблиця 2.29

Сорт (Varieties) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Сорт	Variety	VARCHAR(30)	Not NULL
Код сезону	SeasonID	INT	Not NULL

Таблиця 2.30

Сезон (Seasons) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Сезон	Season	VARCHAR(20)	Not NULL

Таблиця 2.31

Наявний продукт (AvailableProducts) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Код сховища	VaultID	INT	Not NULL
Код продукту	ProductID	INT	Not NULL
Кількість	Amount	DOUBLE	Not NULL
Ціна	Price	DECIMAL	Not NULL
Маса	Mass	DOUBLE	Not NULL
Дата завантаження	DeliveryDate	DATE	Not NULL
Дата кінця зберігання	HoldingEndDate	DATE	NULL

Таблиця 2.32

Утилізований продукт (UtilizedProducts) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код сховища	VaultID	INT	Not NULL
Код продукту	ProductID	INT	Not NULL
Кількість	Amount	DOUBLE	Not NULL
Маса	Mass	DOUBLE	Not NULL
Дата	Date	DATE	Not NULL

Таблиця 2.33

Усохший продукт (DriedUpProducts) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код сховища	VaultID	INT	Not NULL
Код продукту	ProductID	INT	Not NULL
Відсоток усихання	DriedUpPercent	DOUBLE	Not NULL

Таблиця 2.34

Сховище (Vaults) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Тип	Type	VARCHAR(30)	Not NULL
Опис	Description	TINYTEXT	NULL

Таблиця 2.35

Мікроклімат (MicroclimateData) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код дати	DateID	INT	Not NULL
Код сховища	VaultID	INT	Not NULL
Середня температура за день	AvgTempPerDay	FLOAT	Not NULL
Середня вологість за день	AvgHumdPerDay	FLOAT	Not NULL

Таблиця 2.36

Дата (Dates) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	DateID	INT	Not NULL
Дата для мікроклімату	DateForMicroclimate	DATE	Not NULL

Таблиця 2.37

Аккаунт працівника (EmployeeAccounts) [створено автором]

Логічна назва поля	Фізична назва поля	Тип та довжина даних	Обов'язковість поля
Код	ID	INT	Not NULL
Логін	Login	VARCHAR(20)	Not NULL
Пароль	Password	VARCHAR(30)	Not NULL
Прізвище	LastName	VARCHAR(30)	Not NULL
Ім'я	FirstName	VARCHAR(30)	Not NULL
По-батькові	Partonymic	VARCHAR(30)	Not NULL
Спеціальне поле	SpecialField	VARCHAR(10)	Not NULL

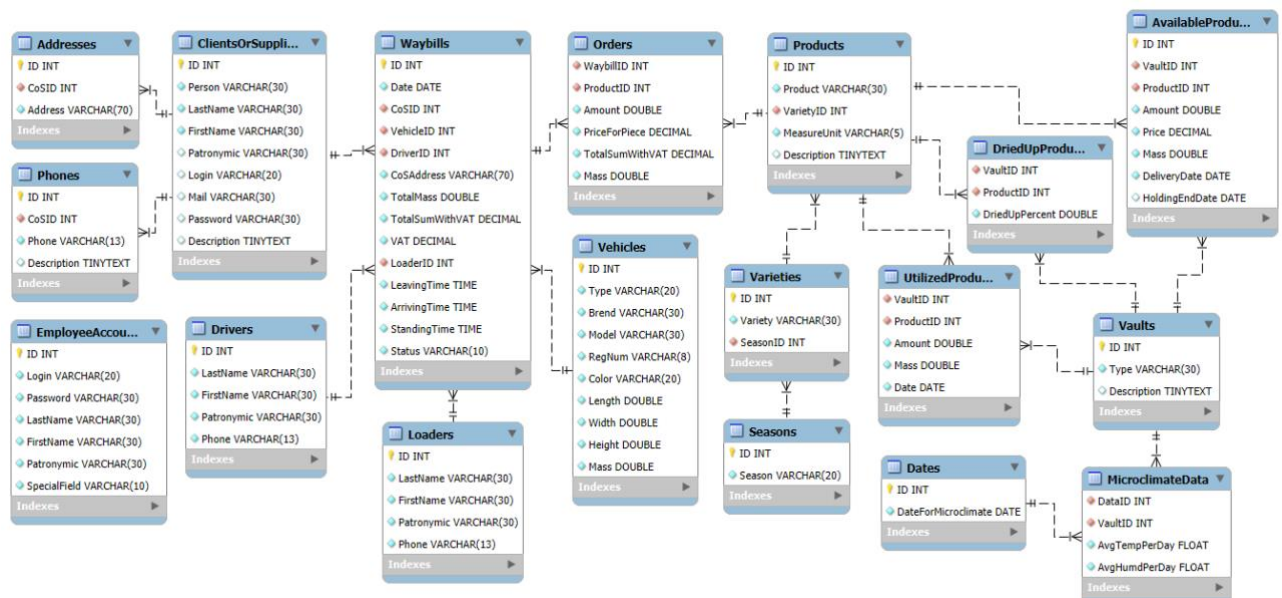


Рис. 2.7. Фізична модель даних інформаційної системи овочесховища [створено автором]

Розроблена фізична модель даних є кінцевим етапом проєктування бази даних, який безпосередньо впливає на її роботу в реальному середовищі. Вона є критичною для досягнення високої продуктивності, надійності та сумісності бази даних із обраною СУБД, а також забезпечує зручність в управлінні даними.

2.4.3. Кодування

Програмний код — це текстова інструкція, створена за правилами певної мови програмування, що визначає, як комп'ютер повинен виконувати певні дії. Код є основою всіх програм і додатків, він дозволяє автоматизувати процеси, обробляти дані, взаємодіяти з апаратним забезпеченням і виконувати різні функції[21]. Головна мета кодування полягає в реалізації вже описаних алгоритмів функціонування інформаційної системи.

Наприклад, розглянемо частину програмного коду ідентифікації користувача.

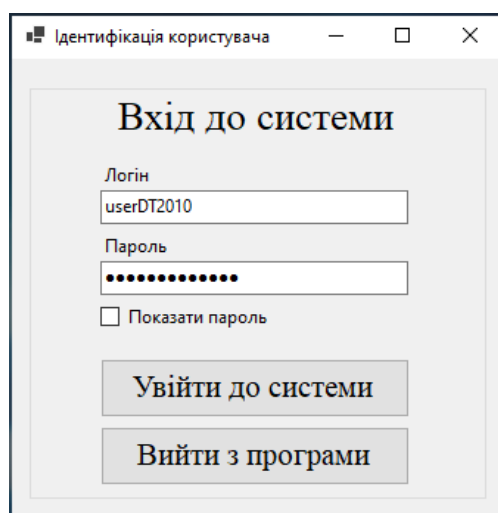


Рис. 2.8. Форма ідентифікація користувача у системі [створено автором]

Програмний код

```
private void buttonEnterSystem_Click(object sender, EventArgs e)
{
    MySqlConnection connection = new MySqlConnection(Services.MySQL.GetConnectionString());
    connection.Open();
    MySqlCommand command = new
    MySqlCommand(Services.MySQL.Queries.LoginDataBase(textBoxLogin.Text.ToString(),
    textBoxPassword.Text.ToString()), connection);
    MySQLDataReader? reader = command.ExecuteReader();
    if (reader == null) {
```

```

        MessageBox.Show("Невірний логін або пароль! \nСпробуйте ще раз.", "Помилка ідентифікації користувача!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    } else {
        DataTable table = new DataTable();
        table.Load(reader);
        string ans = table["SpecialField"][0];
        switch (ans) {
            case "a92nf":
                LogisticsManagerForm logisticsManagerForm = new LogisticsManagerForm();
                this.Hide();
                logisticsManagerForm.Show();
                break;
            case "j45kd":
                PurchasingSalesManagerForm purchasingSalesManagerForm = new PurchasingSalesManagerForm();
                this.Hide();
                purchasingSalesManagerForm.Show();
                break;
            case "q78pd":
                WarehouseManagerForm warehouseManagerForm = new WarehouseManagerForm();
                this.Hide();
                warehouseManagerForm.Show();
                break;
        }
    }
}

```

Форма звертається до бази даних із запитом на перевірку наявності аккаунта користувача за допомогою даних логіну та паролю. При успішній перевірці, користувач входить до свого робочого місця.

2.4.4. Налагодження програмного забезпечення

Налагодження програмного забезпечення є ще одною важливою частиною розробки програмного забезпечення. Налагодження програмного забезпечення необхідне для забезпечення коректної та безперебійної роботи програми. Наприклад, приведемо приклад з формою робочого місця менеджера з логістики. При роботі менеджера зі створенням нової товарно-транспортної накладної для швидкого вводу даних використовуються випадаючі переліки (ComboBox) (рис. 2.7).

Рис. 2.9. Форма ідентифікація користувача у системі [створено автором]

В кожний елемент ComboBox за допомогою запитів до бази даних вписуються необхідні дані з різних таблиці.

Так як в кожному модулі програмного застосунку наявна велика кількість даних елементів, виконано детальну перевірку та ретельне налагодження кожного елементу ComboBox для правильної роботи кожного з них.

Висновки до розділу 2

1. Розроблено концепцію інформаційної системи овочесховища, яка враховує взаємодію роботи менеджера із закупки та продажу, менеджера з логістики та менеджера по складу.
2. Особливістю інформаційної системи є відстеження мікроклімату сховищ, який впливає на якість та вагові параметри при завантаженні, відвантаженні та кінцеву ціну продукції.
3. Розроблено концептуальну, логічну та фізичну моделі даних.
4. Обрано мову програмування C# та середовище програмування Visual Studio 2024, а для створення бази даних обрано СКБД MySQL.
5. Виконано кодування та налагодження у відповідності до поставлених вимог.

РОЗДІЛ 3. РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОВОЧЕСХОВИЩА

В результаті розробки інформаційної системи овочесховища розроблено базу даних та програмне забезпечення до неї. Для початку роботи з інформаційною системою користувачу треба запустити програмний додаток.

Запустивши прогнаний додаток (рис 3.1), користувача зустріне форма ідентифікації користувача, в якій йому потрібно ввести свій логін та пароль.

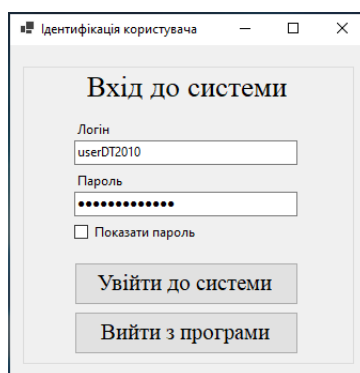


Рис. 3.1. Форма ідентифікація користувача у системі [створено автором]

Натиснувши на кнопку «Увійти до системи», до бази даних формується запит на перевірку відповідності даних до наявних аккаунтів користувачів після чого база даних надає відповідь у вигляді коду робочого місця користувача, до якого його слід підключити та відкриється відповідне робоче місце користувача для його подальшої роботи.

При введенні невірного значення логіну або паролю, користувач отримає відповідне повідомлення у програмі, що змусить його повторити процес ідентифікації користувача та бути більш уважнішим.

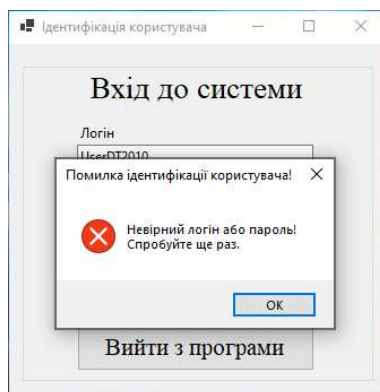


Рис. 3.2. Невірний логін або пароль [створено автором]

Авторизувавшись, користувачу відкривається його форма для роботи з даними. Менеджер з логістики керує даними щодо товарно-транспортних накладних, водіїв, вантажників та транспортних засобів, маючи змогу додавати, видаляти, коригувати та видаляти дані. Обравши для роботи таблицю «Водії», можна подивитись дані наявних водіїв та виконати пошук (рис. 3.3).

Код	Прізвище	Ім'я	По-батькові	Телефон
0	Алайчев	Анатолій	Олегович	+380685269052
1	Квач	Михайло	Васильович	+380973961021
2	Махота	Сергій	Вікторович	+380508834719
3	Андрюшко	Дмитро	Олександрович	+380662901147
4	Біляк	Артур	Семенович	+380686754552
5	Ткач	Віктор	Костянтинович	+380999904632

Рис. 3.3. Перегляд даних в таблиці «Водії» [створено автором]

Для виконання пошуку на формі слід поставити галочку навпроти слова «Пошук», після чого стануть доступні фільтри пошуку, які можна переключати за потребою (рис. 3.4).

Код	Прізвище	Ім'я	По-батькові	Телефон
2	Махота	Сергій	Вікторович	+380508834719

Рис. 3.4. Пошук за прізвищем в таблиці «Водії» [створено автором]

Менеджер з логістики, обравши роботу зі створення товарно-транспортних накладних, має змогу заповнювати накладну за допомогою вже наявних даних для вибору за допомогою елемента випадаючого переліку (рис. 3.5).

Рис. 3.5. Створення нової товарно-транспортної накладної [створено автором]

Заповнивши форму потрібними даними, менеджер натискає на кнопку «Створити накладну». Програма оповістить менеджера з логістики, що товарно-транспортна накладна успішно створена (рис 3.6).

Рис. 3.6. Успішне додання накладної до бази даних [створено автором]

Менеджер із закупки та продажів, авторизувавшись, має змогу керувати даними щодо постачальників, клієнтів, продуктів та графіків, маючи змогу додавати, видаляти, коригувати та видаляти дані. Наприклад, обравши для роботи таблицю «Клієнти», можна подивитись дані наявних клієнтів та виконати пошук їх додаткової інформації (рис. 3.7).

Робоче місце менеджера з закупки та продажів

Вітаємо у системі, Ірина Костянтинівна

Таблиці

- Постачальники
- Клієнти**
- Продукти
- Графіки

Вийти з облікового запису

Клієнти

Переглянути Додати Коригувати Видалити

Код	Прізвище	Ім'я	По-батькові	Опис
4	Деха	Марія	Вікторівна	продуктовий магазин 'Гастроном'

Пошук та додаткова інформація ☒

Код Прізвище

Телефони

Телефон

+380979080107

+380685269052

Адреси

Адреса

Київська обл. вул. Поларна 76

Рис. 3.7. Пошук за кодом клієнта в таблиці «Клієнти» [створено автором]

Менеджер по складу, авторизувавшись, має змогу керувати даними щодо сховищ, продуктів, утилізованих продуктів, усохших продуктів та мікроклімату, маючи змогу додавати, видаляти та коригувати дані. Обравши для роботи таблицю «Продукти», можна подивитись дані наявні продукти у сховищі та виконати їх пошук, використовуючи різні типи фільтрів (рис. 3.8).

Робоче місце менеджера по складу

Вітаємо у системі, Валентина Олександрівна

Таблиці

- Сховища
- Продукти**
- Утилізовані продукти
- Усохші продукти
- Мікроклімат

Вийти з облікового запису

Продукти

Переглянути Додати Коригувати

Код	Код сховища	Продукт	Сорт	Одиниця виміру	Кількість	Ціна за одиницю	Дата завою	Дата кінця зберіг
35769	2	Кукурудза	Мармурова	кг	142	402	16.10.2024	18.04.2025
02385	2	Кукурудза	Дейнеріс	кг	85	278	26.10.2024	20.03.2025
30758	1	Кукурудза	Ракель	кг	263	543	6.11.2024	2.05.2025
83571	3	Кукурудза	Орландо	кг	70	390	3.10.2024	27.04.2025
03758	1	Кукурудза	Растлер	кг	97	342	28.11.2024	22.05.2025
67648	0	Кукурудза	Оверленд	кг	114	436	19.10.2024	25.03.2025

Пошук ☒

Код сховища Продукт та сорт

Код продукту

Маса

Від: ДО:

Дата завою

Від: ДО:

Ціна

Від: ДО:

Рис. 3.8. Пошук продукту за його назвою [створено автором]

Менеджер по складу, обравши роботу з даними усохших продуктів, має змогу відкоригувати певний вид продуктів на відсоток усохшості, наприклад, картоплю. Для цього менеджеру необхідно обрати певні дані за допомогою елемента випадаючого переліку та ввести поточних відсоток усихання продукту (рис. 3.9).

The screenshot shows a web application window titled 'Робоче місце менеджера по складу'. On the left is a sidebar with a welcome message and a list of menu items: 'Сховища', 'Продукти', 'Утилізовані продукти', 'Усохші продукти' (selected), and 'Мікроклімат'. At the bottom of the sidebar is a 'Вийти з облікового запису' button. The main area is titled 'Усохші продукти' and contains a form with tabs 'Переглянути', 'Додати', and 'Коригувати'. The 'Коригувати' tab is active. The form includes a 'Код сховища' dropdown (value: 3), a 'Код продукту' dropdown (value: Картопля), and a 'Сорт' dropdown (value: Пікассо). Below these are two input fields for 'Відсоток усихання': the first shows '0' and the second shows '0,11'. An 'Оновити' button is at the bottom right.

Рис. 3.9. Оновлення відсотку усохшості продукту [створено автором]

Обравши роботу з даними утилізованих продуктів, менеджер має змогу додати інформацію про утилізований продукт, частково обравши деякі дані з елемента випадаючого переліку та ввівши додаткові дані щодо утилізованого продукту (рис. 3.10).

The screenshot shows the 'Утилізовані продукти' form. The sidebar is identical to the previous screenshot, with 'Утилізовані продукти' selected in the menu. The main area is titled 'Утилізовані продукти' and contains a form with tabs 'Переглянути', 'Додати', and 'Коригувати'. The 'Додати' tab is active. The form includes a 'Код сховища' dropdown (value: 0), a 'Код продукту' dropdown (value: Цибуля порей), and a 'Сорт' dropdown (value: Голіас). Below these are three input fields: 'Кількість' (value: 1, unit: шт), 'Маса' (value: 216, unit: кг), and 'Дата' (value: 08.12.2024). A 'Додати' button is at the bottom right.

Рис. 3.10. Додання даних про утилізований продукт [створено автором]

Висновки до розділу 3

Застосування програмного забезпечення та бази даних дозволяє ефективно та без зайвих зусиль працювати менеджеру із закупки та продажу, менеджеру з логістики та менеджеру по складу, швидко обмінюючись даними, використовуючи інтуїтивно зрозумілий та простий інтерфейс, та надійно зберігаючи всю важливу інформацію у базі даних.

Перспективи подальших досліджень будуть пов'язані з практичною реалізацією розробленої інформаційної системи на існуючих та нових овочесховищах.

РОЗДІЛ 4. РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ДЛЯ ОВОЧЕСХОВИЩА

4.1. Розрахунок вартості матеріально-енергетичних витрат

Матеріальні витрати — це витрати підприємства або організації, пов'язані з придбанням, використанням і споживанням матеріальних ресурсів, необхідних для здійснення господарської діяльності. До них належать усі витрати на сировину, матеріали, енергію, паливо та інші ресурси, що безпосередньо або опосередковано використовуються у виробничому процесі або для забезпечення послуг[22,23].

Основні категорії матеріальних витрат:

1. Сировина і матеріали. Основні та допоміжні матеріали, які використовуються у виробництві продукції (наприклад, метал для виготовлення машин, текстиль для швейного виробництва).
2. Паливо та енергія. Витрати на електроенергію, газ, воду, мазут тощо.
3. Комплектуючі та запчастини. Готові частини, які використовуються для складання виробів.
4. Тара і пакувальні матеріали. Витрати на упаковку продукції (наприклад, коробки, плівки, ящики).
5. Інструменти та приладдя. Витратні матеріали для виробничого процесу, що мають короткий термін служби.
6. Транспортно-заготівельні витрати. Витрати на транспортування і зберігання матеріалів до моменту їх використання.

Розрахунок матеріальних витрат [складено автором]

Найменування матеріалу	Кількість	Одиниця виміру	Ціна за одиницю	Загальна сума в грн.
1. Ручка гелева Buromax Erase Slim	1	пачка	229,00	229,00
2. Офісний папір A4 Maestro Standart+	1	пачка.	179,00	179,00
3. Папка-швидкозшивач Buromax Job PP A4	1	набір.	78,00	78,00
4. Флеш-накопичувач Verico Cordial 4 GB	2	шт.	255,00	310,00
5. Чорнило ColorWay Epson L	1	шт.	99,00	99,00
6. Оплата за інтернет	2	міс.	150,00	300,00
7. Непередбачувані витрати			650,00	650,00
Разом:				1 845,00
Транспортно-заготівельні витрати (20% від загальної суми матеріальних витрат)				369,00
Всього:				2 214,00

Розрахувавши матеріальні витрати, перейдемо до розрахунків енергетичних витрат.

Енергетичні витрати вираховуються за формулою:

$$V_{\text{ел}} = D * \gamma * P * W * A, \quad (4.1)$$

де: $V_{\text{ел}}$ – вартість електроенергії; D – кількість робочих днів (37); γ – кількість робочих годин; P – ціна за кВт/год; W – потужність принтера, комп'ютера; A – кількість принтерів, комп'ютерів.

Енергетичні витрати прямо залежать від тривалості розробки програмного продукту: чим більше часу займає розробка, тим вищими будуть витрати. На момент розрахунків вартість однієї кВт/год становить 4,32 грн.

Кількість принтерів – 1 шт. Потужність принтера по технічному паспорту складає 0,14 кВт/год. Середня тривалість роботи принтера за день 2 години. Розрахуємо витрати електроенергії для принтера:

$$V_{\text{ел/принт}} = 37 * 2 * 4,32 * 0,14 * 1 = 44,76 \text{ грн}$$

Кількість комп'ютерів – 2 шт. Потужність комп'ютера по технічному паспорту складає 0,36 кВт/год. Середня тривалість роботи комп'ютера за день 8 годин. Розрахуємо витрати електроенергії для комп'ютерів:

$$V_{\text{ел/комп}} = 37 * 8 * 4,32 * 0,36 * 2 = 920,68 \text{ грн}$$

Обчислимо загальні витрати електроенергії за формулою:

$$V_{\text{загал}} = V_{\text{ел/комп}} + V_{\text{ел/принт}}; \quad (4.2)$$

$$V_{\text{загал}} = 920,68 + 44,76 = 965,44 \text{ грн}$$

Останнім кроком необхідно вичислити загальні матеріально-енергетичні витрати ($MEB_{\text{загал}}$) за формулою:

$$MEB_{\text{загал}} = V_{\text{загал}} + MB, \quad (4.3)$$

де: MB – матеріальні витрати;

Обчислимо загальні матеріально-енергетичні витрати:

$$MEB_{\text{загал}} = 965,44 + 2\,214 = 3\,179,44 \text{ грн}$$

Отже, загальні матеріально-енергетичні витрати на розробку інформаційної системи овочесховища становлять 3 179,44 грн.

4.2. Розрахунок фонду оплати праці за виконану роботу

Заробітна плата — це винагорода, яку працівник отримує від роботодавця за виконану роботу, відповідно до умов трудового договору чи законодавства. Вона є основним джерелом доходу для більшості працюючих людей і визначається в грошовій формі. Види заробітної плати: основна, додаткова, заохочувальні виплати[22,24].

Основна заробітна плата видається за виконання нормованих завдань і обов'язків, встановлених трудовим договором та визначається за формулою:

$$Z_{\text{осн}} = T * C_{\text{сер.г.}} \quad (4.4)$$

де: T – загальна трудомісткість виконання робіт; $C_{\text{сер.г.}}$ – середньо-годинна тарифна ставка.

Середньо-годинна тарифна ставка визначається за формулою:

$$C_{\text{сер.г.}} = (C_{\text{r1p}} + C_{\text{r2p}}) / n, \quad (4.5)$$

де: $C_{г1p}$ – середня годинна тарифна ставка першого виконавця роботи;
 $C_{г2p}$ – середня годинна тарифна ставка другого виконавця роботи; n – кількість працівників.

Додаткова заробітна плата це надбавки, премії, або оплата за понаднормову роботу, роботу в нічний час, святкові чи вихідні дні та визначається за формулою:

$$З_{дод} = B * З_{осн} / 100, \quad (4.6)$$

де: $З_{осн}$ – основна заробітна плата; B – відсоток виплат премій на підприємстві (25%).

Проаналізувавши ринок праці, встановлено, що середня годинна ставка для прикладного програміста становить 153,00 грн/год, а адміністратора баз даних – 144,00 грн.

Обчислимо середньогодинну ставку робітників за формулою 4.5:

$$C_{сер.г} = (153,00 + 144,00) / 2 = 148,50 \text{ грн}$$

Трудомісткість становить 690 годин.

Розрахуємо основну заробітну плату за формулою 4.4:

$$З_{осн} = 690 * 148,50 = 102\,465,00 \text{ грн}$$

Розрахуємо додаткову заробітну плату за формулою 4.6:

$$З_{дод} = 29 * 102\,465,00 / 100 = 29\,714,85 \text{ грн}$$

Фонд оплати праці (ФОП) — це сукупність коштів, які роботодавець виділяє для виплати заробітної плати працівникам за виконану ними роботу. Він включає всі види виплат: основна зарплата, додаткова та заохочувальні виплати. Визначається за формулою:

$$ФОП = З_{осн} + З_{дод}, \quad (4.7)$$

де: $З_{осн}$ – основна заробітна плата; $З_{дод}$ – додаткова заробітна плата.

Єдиний соціальний внесок (ЄСВ) — це обов'язковий платіж, який сплачується до системи загальнообов'язкового державного соціального страхування. Він забезпечує фінансування соціальних виплат, таких як пенсії, допомога з тимчасової непрацездатності, виплати у разі безробіття, материнства

тощо. ЄСВ в Україні регулюється відповідним законодавством[25]. На момент написання ЄСВ складає 22% від ФОП та розраховується за формулою:

$$\text{ЄСВ} = \text{ФОП} * 20 / 100 \quad (4.8)$$

$$\text{ЄСВ} = (102\,465,00 + 29\,714,85) * 20 / 100 = 26\,435,97 \text{ грн}$$

Отже, підприємство має заплатити ЄСВ в сумі 26 435,97 грн.

Наступним кроком визначимо середньоденну заробітну плату працівників за формулою:

$$З_{\text{сер.дн.}} = \text{ФОП} / (D * n), \quad (4.9)$$

де: D – загальна кількість днів праці; n – кількість працівників.

$$З_{\text{сер.дн.}} = 132\,179,85 / (37 * 2) = 1\,786,22 \text{ грн}$$

Формула для розрахунку середньої плати працівника за годину:

$$З_{\text{сер.г}} = З_{\text{сер.дн.}} / \gamma, \quad (4.10)$$

де: $Z_{\text{осн}}$ – основна заробітна плата; $Z_{\text{сер.дн.}}$ – середньоденна заробітна плата; γ – тривалість робочої зміни.

$$З_{\text{сер.г}} = 1\,786,22 / 8 = 223,28 \text{ грн}$$

Середня зарплата одного працівника за годину становить 223,28 грн.

4.3. Розрахунок ціни розробки

Собівартість продукції — це сукупність витрат підприємства, пов'язаних із виробництвом і реалізацією продукції, виконанням робіт або наданням послуг. Вона відображає всі ресурси, які були витрачені для створення конкретного продукту, та є важливим показником для оцінки ефективності діяльності підприємства[26].

Собівартість включає в себе матеріали та сировину, які були витрачені на створення продукту, амортизацію обладнання та податки.

Наступним кроком необхідно розрахувати амортизацію обладнання (принтер, комп'ютери). Здійснити розрахунок амортизації вирішено за допомогою кумулятивного методу. Цей метод належить до прискорених,

оскільки дозволяє списувати більшу частину вартості активу у перші роки його експлуатації, у порівнянні з подальшими періодами.

Основною перевагою методу є те, що значна частина вартості активу амортизується в перші роки, коли інтенсивність його використання найвища, а його знецінення відбувається швидше. Це збігається з періодом, коли витрати на ремонт і технічне обслуговування зазвичай менші, але з часом вони поступово зростають.

Таблиця 4.2

Розрахунок ціни розробки інформаційної системи овочесховища [складено автором]

Статті калькуляцій	Методика розрахунку	Сума в грн.
1. Матеріальні витрати (враховуючи транспортно-заготівельні витрати)	За обчисленнями в таблиці 4.1	2 214,00
2. Енергетичні витрати	За обчисленнями формули 4.2	965,44
3. Основна заробітна плата	За обчисленнями формули 4.4	102 465,00
4. Додаткова заробітна плата	29% від статті №3 (формула 4.6)	29 714,85
5. Єдиний соціальний внесок	За обчисленнями формули 4.8	26 435,97
6. Амортизаційні відрахування	За обчисленнями формули 4.14	7 760,84
7. Загальновиробничі витрати	95% від статті №3	97 341,75
8. Виробнича собівартість	Сума статей №1-7	266 897,85
9. Позавиробничі витрати	13% від статті №8	34 696,72
10.Адміністративні витрати	80% від статті №3	81 972
11.Повна собівартість	Сума статей №8-10	383 566,57
12.Рівень рентабельності	20% від статті №11	76 713,31
13.Ціна розробки	Сума статей №11-12	460 279,88

Розрахунок амортизації за кумулятивним методом здійснюється для мінімального терміну експлуатації — у нашому випадку перший рік (мінімальний строк використання таких засобів, як ПК і принтери) — за формулою:

$$A_{\text{рік}} = K_{\text{кум}} * C_A, \quad (4.11)$$

де: $K_{\text{кум}}$ — кумулятивний коефіцієнт; C_A — це різниця первісної та ліквідаційної вартостей.

Кумулятивний коефіцієнт визначається шляхом ділення кількості років, що залишаються до завершення строку корисного використання об'єкта основних засобів, на загальну суму років його експлуатації.

$$K_{1 \text{ рік кум}} = 2 / (1 + 2) = 0,67$$

Наявно два однакових комп'ютера. Їх загальна вартість становить 57 300 грн. Ліквідаційна вартість обох комп'ютерів становить 2 100 грн. Також наявний принтер, вартість якого становить 15 000 грн, а ліквідаційна — 700 грн. Тобто:

$$C_{A/\text{комп}} = 57\,300 - 2\,100 = 55\,200 \text{ грн}$$

$$C_{A/\text{принт}} = 15\,000 - 700 = 14\,300 \text{ грн}$$

Обчислення амортизаційного відрахування за рік за формулою 4.11:

$$A_{\text{рік/комп}} = 55\,200 * 0,67 = 36\,984 \text{ грн}$$

$$A_{\text{рік/принт}} = 14\,300 * 0,67 = 9\,581 \text{ грн}$$

Обчислення амортизаційного відрахування за місяць:

$$A_{\text{міс/комп}} = A_{\text{рік/комп}} / 12 = 36\,984 / 12 = 3\,082 \text{ грн} \quad (4.12)$$

$$A_{\text{міс/принт}} = A_{\text{рік/принт}} / 12 = 9\,581 / 12 = 798,42 \text{ грн}$$

Обчислення амортизаційного відрахування за два місяці:

$$A_{\text{комп}} = A_{\text{міс/комп}} * 2 = 3\,082 * 2 = 6\,164 \text{ грн} \quad (4.13)$$

$$A_{\text{принт}} = A_{\text{міс/принт}} * 2 = 798,42 * 2 = 1\,596,84 \text{ грн}$$

$$A_{\text{заг}} = A_{\text{комп}} + A_{\text{принт}} = 6\,164 + 1\,596,84 = 7\,760,84 \text{ грн} \quad (4.14)$$

Отже, загальна амортизація складає 7 760,84 грн.

4.4. Розрахунок економічної ефективності та строку окупності розробленої інформаційної системи

Ефективність — це співвідношення між досягнутими результатами та витраченими ресурсами, зусиллями чи часом. Вона показує, наскільки успішно досягаються поставлені цілі за заданих умов[27].

Економічна ефективність — це показник, що характеризує результативність використання економічних ресурсів (капіталу, праці, матеріалів) для досягнення певних цілей. Її метою є максимізація результатів за мінімальних витрат. Її формула:

$$ЕФ = dC - K_{\text{еф}} * B_{\text{св}}, \quad (4.15)$$

де: dC — вивільнені кошти після впровадження інформаційної системи овочесховища; $K_{\text{еф}}$ — коефіцієнт економічної ефективності ($K_{\text{еф}} = 0,6$); $B_{\text{св}}$ — виробнича собівартість (266 897,85 грн).

$$dC = \Phi ЗП - EB, \quad (4.16)$$

де: $\Phi ЗП$ — фонд заробітної плати; EB — експлуатаційні витрати.

Обчислимо експлуатаційні витрати за формулою:

$$EB = A_{\text{відр}} + MB + B_{\text{ел}} = 7\,760,84 + 2\,214,00 + 965,44 = 10\,940,28 \quad (4.17)$$

Після впровадження інформаційної системи овочесховища з'являється можливість оптимізувати штат, що дозволяє скоротити посаду прикладного програміста. Такий підхід сприятиме зменшенню витрат на фонд оплати праці. Середньомісячна заробітна плата одного працівника становить $1\,786,22 * 22 = 39\,296,84$ грн, отже, річна економія фонду заробітної плати складе 471 562,08 грн

Тепер обчислимо вивільнені кошти після впровадження інформаційної системи овочесховища за формулою 4.16:

$$dC = 471\,562,08 - 10\,940,28 = 460\,621,80 \text{ грн}$$

Тепер обчислимо економічну ефективність за формулою 4.15:

$$ЕФ = 460\,621,80 - 0,6 * 266\,897,85 = 300\,483,09 \text{ грн}$$

Фінальним етапом стане визначення періоду окупності інформаційної системи овочесховища за формулою:

$$T_{OK} = B_{cv} / dC = 266\,897,85 / 460\,621,80 = 0,58 \quad (4.18)$$

Отже, судячи зі зроблених обчислень, можна побачити, що розроблена інформаційна система овочесховища окупиться майже за півроку.

Висновки до розділу 4

Визначено та розраховано матеріальні, енергетичні витрати на розробку інформаційної системи овочесховища. Розраховано фонд оплати праці. Розраховано амортизацію двох комп'ютерів та принтера за допомогою кумулятивного методу та розраховано повну ціну розробки інформаційної системи овочесховища. Розраховано економічну ефективність та період окупності після впровадження інформаційної системи, який складає майже півроку.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

5.1. Охорона праці, мета та основні аспекти

Охорона праці — це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних, профілактичних і реабілітаційних заходів та засобів, спрямованих на забезпечення безпечних і здорових умов праці[28].

Метою охорони праці є збереження життя, здоров'я та працездатності працівників у процесі виконання ними трудових обов'язків.

Основні аспекти охорони праці:

1. Правове регулювання — закони, нормативно-правові акти, що визначають права і обов'язки роботодавців та працівників у сфері охорони праці.
2. Організація роботи — створення системи управління охороною праці на підприємстві.
3. Технічна безпека — використання безпечного обладнання, машин і механізмів.
4. Санітарно-гігієнічні заходи — дотримання умов, що сприяють здоров'ю працівників.
5. Навчання та інструктажі — підготовка працівників до безпечного виконання робіт.
6. Розслідування нещасних випадків — аналіз причин та запобігання аналогічним інцидентам.

Охорона праці охоплює всі види діяльності, де існують потенційні ризики для здоров'я або життя працівників.

5.2. Аналіз шкідливих та небезпечних факторів на робочому місці програміста

При роботі програміста, хоча професія здебільшого вважається безпечною, існують певні шкідливі та небезпечні фактори, які можуть впливати на здоров'я програміста. Роздивимось їх поближче.

Тривале сидіння за комп'ютером. Тривале сидіння може призводити до проблем із хребтом (сколіоз, остеохондроз) та підвищує ризик розвитку варикозного розширення вен.

Зорове напруження. Постійна робота з монітором викликає втому очей, сухість, зниження гостроти зору, може розвинутисть комп'ютерний зоровий синдром.

Незручне робоче місце. Незручне крісло, неергономічна клавіатура чи миша сприяють виникненню болю в шиї, плечах і кистях. При особливо незручній комп'ютерній миші є великий ризик розвитку тунельного синдрому.

Психоемоційні навантаження. Високі вимоги до продуктивності з боку керівника по роботі, дедлайни, стрес можуть серйозно впливати на психіку та спричиняти вигорання, тривожність, депресію.

Електромагнітне випромінювання. Хоча наразі сучасна техніка має низький рівень випромінювання завдяки новим технологіям виготовлення моніторів, тривалий контакт все ж таки може мати кумулятивний вплив.

Недостатня фізична активність. Недостатня активність призводить до зниження обміну речовин, проблем із серцево-судинною системою та надмірної ваги.

Погана вентиляція приміщення. Недостатній рівень вентиляції приміщення призводить до зменшення рівня кисню у приміщенні та викликає швидку втому та головний біль.

Погане освітлення. Недостатнє чи занадто яскраве світло може сприяти травмам (спотикання, падіння) або збільшувати навантаження на очі, призводячи до поступової втрати зору.

Постійний голосний шум. Занадто голосний шум у приміщенні є небезпечним та погано впливає на вушні перетинки, спричиняючи малу глухоту або втрату слуху.

Неправильне використання техніки. Халатний догляд за технічним обладнанням або несправне обладнання може спричинити коротке замикання та може виникнути вогонь. Також є ризик удару струмом.

5.3. Розрахунок системи штучного освітлення приміщення роботи програміста

Розрахунок системи штучного освітлення для приміщення програміста виконується для забезпечення комфортних умов роботи з комп'ютером, що мінімізує зорове напруження та підвищує продуктивність[29].

Для розрахунку системи штучного освітлення, використаємо формулу:

$$N = E * S / (F * K * Z), \quad (5.1)$$

де: N – кількість світильників; E – нормативна освітленість (400 люкс);
 S – площа приміщення (m^2); F – світловий потік одного світильника (лм);
 K – коефіцієнт використання світлового потоку (залежить від типу світильника, кольору поверхонь); Z – коефіцієнт запасу (зазвичай 1.1–1.5, обрано 1.3).

Перш за все, необхідно знайти площу робочого приміщення. Для знаходження площі необхідно провести заміри приміщення, визначити довжину та ширину. Провівши заміри ($d = 7$ метрів, $ш = 4$ метри), визначимо площу приміщення:

$$S = 7.3 * 4.1 = 29.93 (m^2).$$

Наступним кроком, для належного освітлення приміщення, обрано світлодіодну панель Ardero потужністю 40 Ватт та світовим потіком у 3600 Лм.

Належно оцінивши поверхню стін, стелі та світильника, визначено коефіцієнт використання світлового потоку ($K = 0.55$).

Отримавши всі необхідні дані, виконаємо обчислення кількості світильників, які необхідні для належного освітлення приміщення, підставивши всі необхідні значення у формулу:

$$N = 400 * 29.93 / (3600 * 0.55 * 1.3) = 4.65$$

Отже, для забезпечення комфортних умов праці у приміщенні з комп'ютером та запобіганню сильної втоми очей, необхідно мати 5 світлодіодних панелей Ardero потужністю 40 Ватт та світовим потіком у 3600 Лм.

5.4. Заходи щодо зменшення впливу шкідливих та небезпечних факторів при роботі за комп'ютером

Робота програміста передбачає тривале перебування за комп'ютером, тому важливо створити певні правила для мінімізації впливу шкідливих та небезпечних факторів, дотримуючись розроблених правил безпеки, щоб запобігти різним видам захворюванням та створити комфортні умови праці[28].

Основні правила для зменшення впливу шкідливих та небезпечних факторів.

Для початку роботи за комп'ютером потрібно відрегулювати висоту робоче крісла та його спинку для зручності подальшого сидіння. Висоту сидіння потрібно відрегулювати таким чином, щоб клавіатура та миша були на рівні ліктів, а руки розташовувались під кутом 90 градусів. Монітор необхідно розташувати на рівні очей на відстані 50-70 сантиметрів та відрегулювати таким чином, щоб на екрані монітори не було видно відблисків. Також програмісту слід використовувати спеціальні підставки під зап'ястя, щоб уникнути тунельного синдрому.

При роботі за комп'ютером потрібно робити короткі перерви кожні 50-60 хвилин. Під час перерви виконати прості вправи для очей (поруhati очима в різні напрямки, змінювати фокус близько-далеко). Також потрібно зробити невеличку розминку для ніг, рук, шиї та спини.

Для збереження здоров'я очей слід налаштувати екран монітора на частоту оновлення 60 герц або вище та налаштувати комфортну для очей яскравість та контрастність монітору. У налаштування монітора слід зменшити кількість випромінюваного синього світла, перевівши монітор у теплий жовтий тон. За змогою, необхідно використовувати нічний режим у кожній програмі, де є така функція, навіть у світлу годину доби.

Задля забезпечення електробезпеки, усі пристрої мають бути підключені через стабілізатори напруги або джерела безперебійного живлення (UPS) та необхідно періодично перевіряти стан та справність кабелів і роз'ємів. Також, ні в якому разі не можна розташовувати спиртні, газовані чи інші види напоїв поблизу комп'ютерного обладнання.

Для забезпечення чистоти робочого місця, необхідно регулярно протирати монітор, клавіатуру, мишу та інше додаткове обладнання від пилу та бруду.

Для забезпечення стабільного психоемоційного стану необхідно правильно управляти стресом. Задля цього потрібно розбивати великі завдання на більш дрібні, щоб їх можна було швидко виконувати та бачити поточний прогрес. Потрібно встановлювати реалістичні терміни виконання роботи, чітко оцінюючи свої навички та беручи трохи часу з запасом на випадок виникнення несподіваної додаткової роботи. Чергування роботи з виконанням малих фізичних активностей під час перерви допоможе тілу та мозку на незначний час перемикнути фокус уваги та перезавантажити розумову діяльність.

Для забезпечення достатнього рівня кисню у приміщенні, слід кожні дві години провітрювати приміщення на 5-10 хвилин, запускаючи свіже повітря до приміщення та забезпечуючи працівників киснем, або використовувати спеціальні вентиляційні системи.

При недостатньому освітленні, необхідно використовувати настільну лампу для відрегулювати яскравість таким чином, щоб очам було зручно працювати за комп'ютером. Також, необхідно відрегулювати потік світла від лампи таким чином, щоб на екрані не було видно відблисків.

Дотримання вищеперерахованих правил допоможе програмісту підтримувати як фізичне, так і психологічне здоров'я, зберігати працездатність і комфортно виконувати свої обов'язки.

Висновки до розділу 5

Проаналізовано шкідливі та небезпечні фактори на робочому місці програміста. Розглянуто спеціальні заходи та правила для зменшення їх впливу для комфортної та безпечної роботи програміста за комп'ютером, а також виконано розрахунок системи штучного освітлення приміщення, визначивши кількість необхідних світлодіодних ламп, для запобігання сильної втоми очей у приміщенні.

РОЗДІЛ 6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1. Охорона навколишнього середовища та його мета

Охорона навколишнього середовища — це система заходів, спрямованих на збереження, відновлення та раціональне використання природних ресурсів, захист екосистем від руйнівного впливу людської діяльності та забезпечення екологічної рівноваги для сталого розвитку суспільства[30].

Головною метою охорони навколишнього середовища є забезпечення екологічної рівноваги та сталого розвитку суспільства, що дозволяє:

1. Зберігати природні ресурси для нинішніх і майбутніх поколінь.
2. Забезпечувати здоров'я і добробут людини через чисте довкілля.
3. Захищати біорізноманіття та екосистеми, які підтримують життя на Землі.
4. Мінімізувати шкоду, яку завдає природа через антропогенну діяльність, і запобігати подальшому руйнуванню природного середовища.

6.2. Забруднення навколишнього середовища

Забруднення навколишнього середовища виникає в результаті впливу природних та антропогенних (людських) чинників, які вводять у довкілля шкідливі речовини, змінюють його властивості або порушують природну рівновагу екосистем. Найбільший внесок у забруднення робить діяльність людини через промисловість, сільське господарство, транспорт та споживання ресурсів.

Сучасний світ наповнений різними електронними видами технологій, які не тільки стали невід'ємною частиною нашого життя, а й новим етапом розвитку цивілізації. Але якими б корисними та зручними технології не були, вони несуть

за собою масу неприємних наслідків і один з них це забруднення навколишнього середовища.

На жаль, вся електроніка забруднює навколишнє середовище на різних етапах свого життєвого циклу, починаючи від виробництва та закінчуючи потенційним викиданням електронного пристрою на смітник. Основні джерела забруднення пов'язані з видобутком сировини, споживанням енергії та неправильним поводженням з електронними відходами. Розглянемо життєвий цикл електронних приладів[30].

Видобуток. Для створення електронних пристроїв потрібні метали, такі як літій, кобальт, золото, мідь і рідкоземельні елементи. Видобуток цих матеріалів спричиняє забруднення ґрунтів та вод токсичними речовинами (ціаніди та інші важкі метали), вирубку лісів і втрату біорізноманіття, а також спричиняє викиди парникових газів через використання важкої техніки для добування матеріалів.

Виробництво. Процес виробництва електронні пристрої спричиняє викиди вуглекислого газу (CO_2) та інших парникових газів в атмосферу через використання енергоємних виробничих технологій. Відбувається забруднення вод через використання хімікатів, наприклад, під час очищення кремнію для виробництва мікросхем. Утворюються промислові відходи, багато з яких є токсичними.

Експлуатація. Користувачі, експлуатуючі електричні прилади споживають значну кількість електроенергії, яка часто виробляється з викопного палива, що сприяє викидам парникових газів.

Кінець експлуатації. При поломці або застарілості електронного пристрою, власники викидають їх на смітник, а з урахуванням політик компаній, які зацікавлені в швидкому зносі техніки для вироблення нової та зароблені на цьому додаткових коштів. Такі часті оновлення електронних пристроїв та їх технологій збільшують обсяги електронних відходів.

Життєвий цикл електронного продукту має купу своїх недоліків, що має свій відбиток на забрудненні навколишнього середовища, але найнебезпечніший

та найсерйозніший вплив на навколишнє середовище має кінець експлуатації електронного продукту. Це спричиняє:

Забруднення ґрунту. При розкладанні або спалюванні електроніки важкі метали (свинець, кадмій, арсен) потрапляють у ґрунт. Це призводить до руйнування структури ґрунту та накопичення токсинів у харчовому ланцюгу через вирощування сільськогосподарських культур. Використаний пластин в електронних пристроях виділяє шкідливі хімічні речовини, такі як фталати й бромовані антипірени. Ці речовини довго залишаються в ґрунті, ускладнюючи його очищення.

Забруднення води. При утворенні електронного звалища відходів та зламаної техніки токсичні речовини такі як рідини з батарей та конденсаторів, які були використанні у створенні пристроїв, можуть проникати у підземні води, забруднюючи її та шкодячи водним організмам, роблячи воду непридатною для споживання.

Забруднення повітря. В деяких країнах світу мають тенденцію до спалення електронної техніки, що спричиняє до забруднення повітря токсичними газами, такими як хлоровані органічні сполуки.

Мікропластик і мікрочастки. Електронні пристрої містять пластикові компоненти, які при руйнуванні утворюють мікропластик. Це призводить до накопичення мікропластику в організмі людей через воду та їжу. Також мікрочастинки потрапляють у водні системи, що негативно впливає на морську флору та фауну.

Розглянувши та проаналізувавши небезпечність забруднення навколишнього середовища електронною технікою, перейдемо до розробки заходів щодо зменшення забруднення навколишнього середовища.

6.3. Розробка заходів щодо зменшення забруднення навколишнього середовища

Для зменшення негативного впливу електроніки на навколишнє середовище необхідно впроваджувати системні заходи на рівні держав, бізнесу та громадян[31]. Ці заходи можуть бути поділені на кілька ключових напрямів.

Розробка екологічно чистих технологій. Необхідно перейти на використання матеріалів, які легко переробляти, та проєктувати електронні пристрої, які будуть мати змогу легко ремонтуватись. Необхідно максимально мінімізувати використання токсичних матеріалів або перейти на більш безпечні матеріали. Необхідно спроектувати електричні пристрої таким чином, щоб вони не втрачали своєї потужності але мали низьке енергоспоживання.

Удосконалення системи переробки електронних відходів. Необхідно створити спеціальні інфраструктури, організувати пункти збору в кожному населеному пункті для збору старих пристроїв, техніки за грошову винагороду. Необхідно розвинути технології переробки та утилізації шляхом використання більш екологічно безпечних методів для вилучення дорогоцінних металів і пластику, використовуючи спеціальні сучасні машини як робототехники та використовувати спеціальні хімічні процеси для більш ефективної переробки компонентів. Необхідно повністю заборонити звалища електроніки, вести строгий контроль за їх створенням та при порушенні стягувати великі штрафи з громадян, які не слідують даній забороні.

Регулювання на рівні законодавства. Необхідно встановити певні нормативи для виробників. Наприклад, вимагати підприємства забезпечувати ремонт пристроїв, які вони вироблять, за мінімальну ціну. Накласти зобов'язання на підприємства використовувати екологічні матеріали. Необхідно стимулювати бізнес розробляючи екологічно чисту продукцію, надаючи компаніям пільги.

Екологічна освіта та підвищення обізнаності. Необхідно провести інформування дорослих людей та молоді, на тему шкідливого впливу викидання електронної техніки у смітник та пояснити чому всі повинні нести старі пристрої

до компаній, які займаються утилізацією та позитивний вплив цих дій на навколишнє середовище.

Зменшення швидкості зносу техніки. Необхідно запровадити суворий контроль за виробниками, які штучно скорочують термін служби їх пристроїв, та вимагати виробників забезпечувати не тільки тривалу підтримку програмного забезпечення, а й належне проектування техніки для її більш тривалого використання. Також необхідно заохочувати людей для більш тривалого використання пристроїв, які вони мають, зменшуючи навантаження на центри переробки та утилізації електронних продуктів.

Перехід на кругову економіку. Необхідно впровадити процеси, які забезпечують повне повернення відходів у виробництво. Необхідно запровадити переробку матеріалів зі старих електронних продуктів, таких як пластик і метали, для створення нової продукції. Необхідно зменшити кількість використаних матеріалів у новій техніці.

Зменшення руйнуючого впливу електронних пристроїв на навколишнє середовище потребує комплексного та відповідального підходу, що включає технічні інновації, вдосконалення законодавства та зміни в поведінці споживачів. Відповідальні дії на кожному рівні допоможуть мінімізувати шкоду довкіллю та сприяти розвитку чистого навколишнього середовища.

Висновки до розділу 6

Проведено аналіз чинників забруднення навколишнього середовища через сучасні технології, електронні пристрої та їх виробництво. Життєвий цикл електронних пристроїв, який складається з видобутку матеріалів, виробництва, експлуатації та кінця експлуатації, має купу своїх недоліків, що має свій відбиток на забрудненні навколишнього середовища, тому розроблено заходи для зменшення забруднення навколишнього середовища, які потребують не тільки залучення виробників, а й самих людей, як споживачів.

ЗАГАЛЬНІ ВИСНОВКИ

- 1) Проаналізовано організаційну структуру, інформаційні процеси обміну та визначено їх недоліки. Розглянуто переваги та недоліки готових рішень інформаційних систем.
- 2) Пройдено всі етапи підготовки для розробки інформаційної системи: моделювання, конструювання, технічний проєкт та робочий проєкт.
- 3) Використовуючи середовище розробки Visual Studio 2024, на мові C# розроблено програмне забезпечення. Розроблено базу даних використовуючи СКБД MySQL.
- 4) Проведено аналіз заходів для покращення економічних показників, здійснено розрахунки строку окупності та складено розрахунок витрат на виробництво й реалізацію проєкту.
- 5) Розглянуто заходи щодо охорони праці та навколишнього середовища.

Наукова новизна: удосконалено підхід до розробки моделі даних інформаційної системи, яка, на відміну від існуючих, враховує дані утилізованої, усохшої продукції та мікроклімату.

Практичне значення: розроблено інформаційну систему, застосування якої підвищує ефективність інформаційних процесів на існуючих та новостворених овочесховищах.

Перспективи подальших досліджень будуть пов'язані з практичною реалізацією розробленої інформаційної системи на існуючих та нових овочесховищах.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Користь овочів. URL: <https://abetka-healthykids.com.ua/koryst-ovochiv-blog>, 5.11.2024
2. Історія зберігання продуктів від давнини до наших днів. URL: <https://yak.bono.odessa.ua/articles/istorija-zberigannja-produktiv-vid-davnini-do.php>, 6.11.2024
3. В.М. Геєць. Економіка аграрного сектора. – 2018. – 412 с.
4. Овочесховища. Вплив умов на тривалість зберігання і якість продукції. URL: <https://vseosvita.ua/lesson/ovocheskhoverishcha-vplyv-umov-na-tryvalist-zberihannia-i-iakist-produktsii-204093.html>, 6.11.2024
5. Організаційна структура та її візуалізація. URL: <https://hurma.work/blog/organizacijna-struktura-ta-yiyi-vizualizaciya/#:~:text=Організаційна%20структура%20–%20%E2%80%8B%E2%80%8Bце,і%20зрозумілою%20для%20всіх%20учасників>, 8.11.2024
6. О.І. Підоричев, І.А. Лещенко, В.П. Дмитренко. Інформаційні системи та технології. – 2018. – 416 с.
7. І.А. Лещенко. Системний аналіз та проєктування інформаційних систем. – 2020. – 368 с.
8. О.М. Шклярчук. Архітектура інформаційних систем. – 2019. – 300 с.
9. Г.І. Волкова. Інформаційні системи в управлінні. – 2019. – 290 с.
10. Як вибрати комп'ютер. URL: https://gazik.ua/blog/porady-pokuptsyam/yak-vibrati-komp-yuter/?srsltid=AfmBOoq-349fWodylRqzW3uasJuXXoTIpU33yE-SwNmfrSg6Ngtg_YQp, 15.11.2024
11. Інструкція, як будувати UML-діаграми – DOU. URL: [https://dou.ua/forums/topic/40575/#:~:text=Діаграма%20варіантів%20використання,зробити\)%20з%20точки%20зору%20користувача](https://dou.ua/forums/topic/40575/#:~:text=Діаграма%20варіантів%20використання,зробити)%20з%20точки%20зору%20користувача), 16.11.2024

12. Концептуальна схема - Вікіпедія. URL:
https://uk.wikipedia.org/wiki/Концептуальна_схема#:~:text=7%20Посилання-,Огляд,серію%20тверджень%20про%20її%20природу, 18.11.2024
13. Створення схеми діаграми станів UML. URL:
<https://support.microsoft.com/uk-ua/office/створення-схеми-діаграми-станів-uml-2e46fd66-e861-4e8c-9188-36255395ebf3>, 20.11.2024
14. Моделювання даних (Data Modelling) . URL:
<https://www.maxzosim.com/data-modelling/>, 23.11.2024
15. М.В. Сидоренко. Основи архітектури програмних систем. – 2021. – 320 с.
16. Алгоритм - Вікіпедія. URL: <https://uk.wikipedia.org/wiki/Алгоритм>,
25.11.2024
17. Visual Studio 2024: What's New in Microsoft's IDE for Developers. URL:
<https://cyberogism.com/visual-studio-2024-whats-new/>, 26.11.2024
18. Середовище програмування сі шарп та його переваги. URL:
<https://foxminded.ua/seredovyshche-prohramuvannia-si-sharp/>, 26.11.2024
19. MySQL - Вікіпедія. URL: <https://uk.wikipedia.org/wiki/MySQL>, 26.11.2024
20. Концептуальні, логічні та фізичні моделі даних. URL: <https://data-life-ua.com/db/kontseptualni-lohichni-ta-fizychni-modeli-danykh/>, 26.11.2024
21. П. Коваль. Мистецтво програмування: чистий код. – 2019. – 320 с.
22. Б.Й. Пасхавер. Економіка аграрних підприємств. – 2018. – 392 с.
23. Склад витрат та їх характеристика. URL:
https://elib.lntu.edu.ua/sites/default/files/elib_upload/Тлучкевич%203/page7.html#:~:text=Матеріальні%20витрати%20—%20це%20елемент%20собівартості,%2C%20виконання%20робіт%2C%20надання%20послуг, 27.11.2024
24. Поняття заробітної плати. Порядок і строки виплати заробітної плати.
URL:
https://wiki.legalaid.gov.ua/index.php/Поняття_заробітної_плати._Порядок_і_строки_виплати_заробітної_плати, 27.11.2024

25. Єдиний соціальний внесок: важливо знати. URL: [https://www.pfu.gov.ua/cv/411925-yedynyj-sotsialnyj-vnesok-vazhlyvo-znaty-2/#:~:text=Єдиний%20соціальний%20внесок%20\(ЄСВ\)%20на,порядку%20та%20на%20регулярній%20основі](https://www.pfu.gov.ua/cv/411925-yedynyj-sotsialnyj-vnesok-vazhlyvo-znaty-2/#:~:text=Єдиний%20соціальний%20внесок%20(ЄСВ)%20на,порядку%20та%20на%20регулярній%20основі), 27.11.2024
26. Що таке собівартість продукції та як її розрахувати. URL: <https://online.novaposhta.education/blog/shho-take-sobivartist-produkcii-ta-yak-ii-rozrahuvati>, 27.11.2024
27. Економічна ефективність. URL: <https://esu.com.ua/article-18769>, 27.11.2024
28. Я. Серіков. Основи охорони праці. – 2020. – 384 с.
29. О.В. Войналович. Охорона праці. Практикум. – 2020. – 448 с.
30. Ю. Бойчук, О. Бугай, Е. Солошенко. Екологія і охорона навколишнього середовища. – 2020. – 380 с.
31. В. Джигирей. Екологія і охорона навколишнього середовища. – 2019. – 422 с.

ДОДАТОК А – КОД ПРОГРАМИ

Лістинг 1. Модуль ідентифікації користувача

```
private void buttonEnterSystem_Click(object sender, EventArgs e)
{
    MySqlConnection connection = new MySqlConnection(Services.MySQL.GetConnectionString());
    connection.Open();
    MySqlCommand command = new
    MySqlCommand(Services.MySQL.Queries.LoginDataBase(textBoxLogin.Text.ToString(),
    textBoxPassword.Text.ToString()), connection);
    MySQLDataReader? reader = command.ExecuteReader();
    if (reader == null) {
        MessageBox.Show("Невірний логін або пароль! \nСпробуйте ще раз.", "Помилка ідентифікації
    користувача!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    } else {
        DataTable table = new DataTable();
        table.Load(reader);
        string ans = table["SpecialField"][0];
        switch (ans) {
            case "a92nf":
                LogisticsManagerForm logisticsManagerForm = new LogisticsManagerForm();
                this.Hide();
                logisticsManagerForm.Show();
                break;
            case "j45kd":
                PurchasingSalesManagerForm purchasingSalesManagerForm = new PurchasingSalesManagerForm();
                this.Hide();
                purchasingSalesManagerForm.Show();
                break;
            case "q78pd":
                WarehouseManagerForm warehouseManagerForm = new WarehouseManagerForm();
                this.Hide();
                warehouseManagerForm.Show();
                break;
        }
    }
}
```

Лістинг 2. Модуль перегляду та пошуку товарно-транспортної накладної

```
private void ClearSearchInvoiceForm()
```



```

{
    dataGridViewSerachInvoices.DataTable = null;
    searchInvoiceCheckButton.Checked = false;
    searchInvoiceRadioButtonID.Enable = false;
    searchInvoiceComboBoxID.SelectedIndex = -1;
    searchInvoiceComboBoxID.Enable = true;
    searchInvoiceRadioButtonName.Enable = false;
    searchInvoiceTextBoxName.Text = "";
    searchInvoiceTextBoxName.Enable = false;
    searchInvoiceRadioButtonLastName.Enable = false;
    searchInvoiceTextBoxLastName.Text = "";
    searchInvoiceTextBoxLastName.Enable = false;
    searchInvoiceRadioButtonPhone.Enable = false;
    searchInvoiceTextBoxPhone.Text = "";
    searchInvoiceTextBoxPhone.Enable = false;
}

private void FillAllSearchInvoiceForm()
{
    FillDataGridViewSearchInvoices(Services.MySQL.Queries.GetDataDataGridViewSearchInvoices())
    FillSearchInvoiceComboBox(Services.MySQL.Queries.GetDataSearchInvoiceComboBoxID());
}

private void FillDataGridViewSearchInvoices(string sqlQuery)
{
    MySqlCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    dataGridViewSerachInvoices.DataTable = table.Data;
}

private void FillSearchInvoiceComboBoxID(string sqlQuery)
{
    MySqlCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] ans = table["ID"];
    searchInvoiceComboBoxID.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        searchInvoiceComboBoxID.Items.Add(table.Rows[i]["ID"]);
}

```

```

}

private void searchInvoiceCheckButton_CheckedChanged(object sender, EventArgs e)
{
    if (searchInvoiceCheckButton.Checked)
    {
        searchInvoiceRadioButtonID.Enable = true;
        searchInvoiceRadioButtonName.Enable = true;
        searchInvoiceRadioButtonLastName.Enable = true;
        searchInvoiceRadioButtonPhone.Enable = true;
    }
    else
    {
        searchInvoiceRadioButtonID.Enable = false;
        searchInvoiceRadioButtonName.Enable = false;
        searchInvoiceRadioButtonLastName.Enable = false;
        searchInvoiceRadioButtonPhone.Enable = false;
    }
}

private void searchInvoiceRadioButtonID_CheckedChanged(object sender, EventArgs e)
{
    if (searchInvoiceRadioButtonID.Checked)
    {
        searchInvoiceComboBoxID.Enable = true;
    }
    else
    {
        searchInvoiceComboBoxID.SelectedIndex = -1;
        searchInvoiceComboBoxID.Enable = false;
    }
}

private void searchInvoiceRadioButtonName_CheckedChanged(object sender, EventArgs e)
{
    if (searchInvoiceRadioButtonName.Checked)
    {
        searchInvoiceTextBoxName.Enable = true;
    }
    else
    {
        searchInvoiceTextBoxName.SelectedIndex = -1;
    }
}

```

```

        searchInvoiceTextBoxName.Enable = false;
    }
}

private void searchInvoiceRadioButtonLastName_CheckedChanged(object sender, EventArgs e)
{
    if (searchInvoiceRadioButtonLastName.Checked)
    {
        searchInvoiceTextBoxLastName.Enable = true;
    }
    else
    {
        searchInvoiceTextBoxLastName.SelectedIndex = -1;
        searchInvoiceTextBoxLastName.Enable = false;
    }
}

private void searchInvoiceRadioButtonPhone_CheckedChanged(object sender, EventArgs e)
{
    if (searchInvoiceRadioButtonPhone.Checked)
    {
        searchInvoiceTextBoxPhone.Enable = true;
    }
    else
    {
        searchInvoiceTextBoxPhone.SelectedIndex = -1;
        searchInvoiceTextBoxPhone.Enable = false;
    }
}

private void searchInvoiceComboBoxID_SelectedValueChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchInvoiceComboBoxID.SelectedValue.ToString());

    FillDataGridViewSearchInvoices(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchInvoicesByID(query.ToString()));
}

private void searchInvoiceTextBoxName_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchInvoiceTextBoxName.Text.ToString());

```

```

FillDataGridViewSearchInvoices(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchInvoicesByName(
query.ToString()));
}

private void searchInvoiceTextBoxLastName_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchInvoiceTextBoxLastName.Text.ToString());
    FillDataGridViewSearchInvoices(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchInvoicesByLastNa
me(query.ToString()));
}

private void searchInvoiceTextBoxPhone_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchInvoiceTextBoxLastPhone.Text.ToString());
    FillDataGridViewSearchInvoices(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchInvoicesByPhone(
query.ToString()));
}

```

ЛІСТИНГ 3. Модуль створення нової товарно-транспортної накладної

```

private void ClearSearchInvoiceForm()
{
    addInvoiceDateTime.Enable = true;
    addInvoiceDateTime.SetDay = Services.Date.GetCurrentDateTime();
    addInvoiceRadioButtonClients.Enable = true;
    addInvoiceComboBoxClients.SelectedIndex = -1;
    addInvoiceRadioButtonSuppliers.Enable = false;
    addInvoiceComboBoxSuppliers.Enable = false;
    addInvoiceComboBoxSuppliers.SelectedIndex = -1;
    addInvoiceComboBoxDrivers.SelectedIndex = -1;
    addInvoiceComboBoxLoaders.SelectedIndex = -1;
    addInvoiceComboBoxVehicles.SelectedIndex = -1;
    addInvoiceComboBoxAddresses.SelectedIndex = -1;
    comboBoxesArray = new ComboBox[3];
    comboBoxesList = new List<comboBoxesArray>();
    comboBoxesArray[0] = new ComboBox("addInvoiceComboBoxProducts0");
    comboBoxesArray[1] = new ComboBox("addInvoiceComboBoxVarieties0");
    comboBoxesArray[2] = new ComboBox("addInvoiceComboBoxMeasureUnit0");
    comboBoxesList.Add(comboBoxesArray);
    comboBoxesList[0][0].SelectedIndex = -1;
    comboBoxesList[0][1].SelectedIndex = -1;
}

```

```

        comboBoxesList[0][2].SelectedIndex = -1;
        addInvoiceTextBox.Text = "";
    }

    ComboBox[] comboBoxesArray;
    List<comboBoxesArray> comboBoxesList;

    private void FillAllAddInvoiceForm()
    {
        FillAddInvoiceComboBoxClients(Services.MySQL.Queries.GetDataAddInvoiceComboBoxClients());
        FillAddInvoiceComboBoxSuppliers(Services.MySQL.Queries.GetDataAddInvoiceComboBoxSuppliers());
        FillAddInvoiceComboBoxLoaders(Services.MySQL.Queries.GetDataAddInvoiceComboBoxLoaders());
        FillAddInvoiceComboBoxVehicles(Services.MySQL.Queries.GetDataAddInvoiceComboBoxVehicles());
        FillAddInvoiceComboBoxAddresses(Services.MySQL.Queries.GetDataAddInvoiceComboBoxAddresses());
        FillAddInvoiceAllComboBoxProducts(Services.MySQL.Queries.GetDataAddInvoiceComboBoxProducts());
    }

    private void FillAddInvoiceComboBoxClients(string sqlQuery)
    {
        MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
        MySqlDataReader? reader = mySqlCommand.ExecuteReader();
        DataTable table = new DataTable();
        table.Load(reader);
        string[] names = table["FirstName"];
        string[] lastNames = table["LastName"];
        string[] patronymics = table["Patronymic"];
        addInvoiceComboBoxClients.Items.Clear();
        for (int i = 0; i < table.Rows.Count; i++)
            addInvoiceComboBoxClients.Items.Add((names[i] + " " + lastNames[i] + " " + patronymics[i]).ToString());
    }

    private void FillAddInvoiceComboBoxSuppliers(string sqlQuery)
    {
        MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
        MySqlDataReader? reader = mySqlCommand.ExecuteReader();
        DataTable table = new DataTable();
        table.Load(reader);
        string[] names = table["FirstName"];
        string[] lastNames = table["LastName"];
        string[] patronymics = table["Patronymic"];
        addInvoiceComboBoxSuppliers.Items.Clear();
        for (int i = 0; i < table.Rows.Count; i++)

```

```

        addInvoiceComboBoxSuppliers.Items.Add((names[i] + " " + lastNames[i] + " " + patronymics[i]).ToString());
    }

private void FillAddInvoiceComboBoxLoaders(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] names = table["FirstName"];
    string[] lastNames = table["LastName"];
    string[] patronymics = table["Patronymic"];
    addInvoiceComboBoxLoaders.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        addInvoiceComboBoxLoaders.Items.Add((names[i] + " " + lastNames[i] + " " + patronymics[i]).ToString());
}

private void FillAddInvoiceComboBoxVehicles(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] brends = table["Brend"];
    string[] models = table["Model"];
    string[] regNums = table["RegNum"];
    addInvoiceComboBoxVehicles.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        addInvoiceComboBoxVehicles.Items.Add((brends[i] + " " + models[i] + " | " + regNums[i]).ToString());
}

private void FillAddInvoiceComboBoxAddresses(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] addresses = table["Address"];
    addInvoiceComboBoxAddresses.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        addInvoiceComboBoxAddresses.Items.Add(addresses[i].ToString());
}

```

```

private void FillAddInvoiceAllComboBoxProducts(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] products = table["Product"];
    string[] varieties = table["Variety"];
    string[] measureUnits = table["MeasureUnit"];
    addInvoiceComboBoxProducts.Items.Clear();
    addInvoiceComboBoxVariety.Items.Clear();
    addInvoiceComboBoxMeasureUnit.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
    {
        addInvoiceComboBoxProducts.Items.Add(products[i].ToString());
        addInvoiceComboBoxVariety.Items.Add(varieties[i].ToString());
        addInvoiceComboBoxMeasureUnit.Items.Add(measureUnits[i].ToString());
    }
}

```

```

private void addInvoiceRadioButtonClients_CheckedChanged(object sender, EventArgs e)
{
    if (addInvoiceRadioButtonClients.Checked)
    {
        addInvoiceComboBoxClients.Enable = true;
    }
    else
    {
        addInvoiceComboBoxClients.SelectedIndex = -1;
        addInvoiceComboBoxClients.Enable = false;
    }
}

```

```

FillAddInvoiceComboBoxAddresses(Services.MySQL.Queries.GetDataAddInvoiceComboBoxAddressesByClient());
}

```

```

private void addInvoiceRadioButtonSuppliers_CheckedChanged(object sender, EventArgs e)
{
    if (addInvoiceRadioButtonSuppliers.Checked)
    {
        addInvoiceComboBoxSuppliers.Enable = true;
    }
}

```

```

    }
    else
    {
        addInvoiceComboBoxSuppliers.SelectedIndex = -1;
        addInvoiceComboBoxSuppliers.Enable = false;
    }
    FillAddInvoiceComboBoxAddresses(Services.MySQL.Queries.GetDataAddInvoiceComboBoxAddressesBySupplier())
;
}

private void addInvoiceButtonAddProduct_Click(object sender, EventArgs e)
{
    comboBoxesArray = new ComboBox[3];
    comboBoxesArray[0] = new ComboBox("addInvoiceComboBoxProducts1");
    comboBoxesArray[1] = new ComboBox("addInvoiceComboBoxVarieties1");
    comboBoxesArray[2] = new ComboBox("addInvoiceComboBoxMeasureUnit1");
    comboBoxesList.Add(comboBoxesArray);
    for (int i = 0; i < 3; i++)
    {
        LoginForm.CreateNewObject(comboBoxesArray[1][i]);
    }
    FillAddInvoiceComboBoxProducts(Services.MySQL.Queries.GetDataAddInvoiceComboBoxProducts());
    FillAddInvoiceComboBoxVarieties(Services.MySQL.Queries.GetDataAddInvoiceComboBoxVarieties());
    FillAddInvoiceComboBoxMeasureUnit(Services.MySQL.Queries.GetDataAddInvoiceComboBoxMeasureUnit());
}

private void addInvoiceButtonCreateWaybill_Click(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder();
    query.Add(addInvoiceDateTime.ToString());
    query.Add(addInvoiceComboBoxClients.SelectedItem.ToString());
    query.Add(addInvoiceComboBoxSuppliers.SelectedItem.ToString());
    query.Add(addInvoiceComboBoxDrivers.SelectedItem.ToString());
    query.Add(addInvoiceComboBoxLoaders.SelectedItem.ToString());
    query.Add(addInvoiceComboBoxVehicles.SelectedItem.ToString());
    query.Add(addInvoiceComboBoxAddresses.SelectedItem.ToString());
    for (int i = 0; i < comboBoxesList.Length - 1; i++)
    {
        for (int j = 0; j < 3; j++)
        {
            query.Add(comboBoxesList[i][j].SelectedItem.ToString());
        }
    }
}

```



```

    }
    Services.MySQL.Queries.AddNewInvoice(query.ToString());
    MessageBox.Show("Товарно-транспортна накладна успішно створена!", "Успіх!", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

```

Лістинг 4. Модуль перегляду та пошуку наявної продукції

```

private void ClearSearchProductsForm()
{
    dataGridViewSearchProducts.DataTable = null;
    searchProductsCheckButton.Checked = false;
    searchProductsRadioButtonVaultID.Enable = false;
    searchProductsTextBoxVaultID.Text = "";
    searchProductsRadioButtonProductID.Enable = false;
    searchProductsTextBoxProductID.Text = "";
    searchProductsRadioButtonProductAndVariety.Enable = false;
    searchProductsComboBoxProduct.SelectedIndex = -1;
    searchProductsComboBoxVariety.SelectedIndex = -1;
    searchProductsRadioButtonMass.Enable = false;
    searchProductsTextBoxMassFrom.Text = "";
    searchProductsTextBoxMassTo.Text = "";
    searchProductsRadioButtonDeliveryDate.Enable = false;
    searchProductsDeliveryDateFrom = Services.Date.GetCurrentDateTime();
    searchProductsDeliveryDateTo = Services.Date.GetCurrentDateTime();
    searchProductsRadioButtonPrice.Enable = false;
    searchProductsTextBoxPriceFrom.Text = "";
    searchProductsTextBoxPriceTo.Text = "";
}

private void FillAllSearchProductsForm()
{
    FillDataGridViewSearchProducts(Services.MySQL.Queries.GetDataDataGridViewSearchProducts())
    FillSearchProductsComboBoxProduct(Services.MySQL.Queries.GetDataSearchProductsComboBoxProduct());
    FillSearchProductsComboBoxVariety(Services.MySQL.Queries.GetDataSearchProductsComboBoxVariety());
}

private void FillDataGridViewSearchProducts(string sqlQuery)
{
    MySqlCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();

```

```

        table.Load(reader);
        dataGridViewSerachProducts.DataTable = table.Data;
    }

private void FillSearchProductsComboBoxProduct(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySQLCommand(sqlQuery, mySqlConnection);
    MySQLDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] products = table["Product"];
    searchProductsComboBoxProduct.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        searchProductsComboBoxProduct.Items.Add(products[i].ToString());
}

private void FillSearchProductsComboBoxVariety(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySQLCommand(sqlQuery, mySqlConnection);
    MySQLDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] varieties = table["Variety"];
    searchProductsComboBoxVariety.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        searchProductsComboBoxVariety.Items.Add(varieties[i].ToString());
}

private void searchProductsCheckBox_CheckedChanged(object sender, EventArgs e)
{
    if (searchProductsCheckBox.Checked)
    {
        searchProductsRadioButtonVaultID.Enable = true;
        searchProductsRadioButtonVaultID.Checked = true;
        searchInvoiceTextBoxVaultID.Enabled = true;
        searchProductsRadioButtonProductID.Enable = true;
        searchProductsRadioButtonProductAndVariety.Enable = true;
        searchProductsRadioButtonMass.Enable = true;
        searchProductsRadioButtonDeliveryDate.Enable = true;
        searchProductsRadioButtonPrice.Enable = true;
    }
    else

```

```

{
    searchProductsRadioButtonVaultID.Enable = false;
    searchProductsRadioButtonVaultID.Checked = true;
    searchProductsRadioButtonProductID.Enable = false;
    searchProductsRadioButtonProductAndVariety.Enable = false;
    searchProductsRadioButtonMass.Enable = false;
    searchProductsRadioButtonDeliveryDate.Enable = false;
    searchProductsRadioButtonPrice.Enable = false;
    searchInvoiceTextBoxVaultID.Enable = false;
    searchInvoiceTextBoxProductID.Enable = false;
    searchInvoiceComboBoxProduct.Enable = false;
    searchInvoiceComboBoxVariety.Enable = false;
    searchInvoiceTextBoxMassFrom.Enable = false;
    searchInvoiceTextBoxMassTo.Enable = false;
    searchDeliveryDateFrom.Enable = false;
    searchDeliveryDateTo.Enable = false;
    searchInvoiceTextBoxPriceFrom.Enable = false;
    searchInvoiceTextBoxPriceTo.Enable = false;
}
}

private void searchProductsTextBoxVaultID_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsTextBoxVaultID.Text.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByVaultID(query.ToString()));
}

private void searchProductsTextBoxProductID_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsTextBoxProductID.Text.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByProductID(query.ToString()));
}

private void searchProductsComboBoxProduct_SelectedValueChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsComboBoxProduct.SelectedValue.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByProduct(query.ToString()));
}

```

```

private void searchProductsComboBoxVariety_SelectedValueChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsComboBoxVariety.SelectedValue.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByVariety(query.ToString()));
}

private void searchProductsTextBoxMassFrom_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsTextBoxMassFrom.Text.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByMassFrom(query.ToString()));
}

private void searchProductsTextBoxMassTo_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsTextBoxMassTo.Text.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByMassTo(query.ToString()));
}

private void searchProductsDeliveryDateFrom_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsDeliveryDateFrom.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByDateFrom(query.ToString()));
}

private void searchProductsDeliveryDateTo_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsDeliveryDateTo.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByDateTo(query.ToString()));
}

private void searchProductsTextBoxPriceFrom_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsTextBoxPriceFrom.Text.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByPriceFrom(query.ToString()));
}

```

```

private void searchProductsTextBoxPriceTo_TextChanged(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder(searchProductsTextBoxPriceTo.Text.ToString());
    FillDataGridViewSearchProducts(Services.MySQL.Queries.SearchAndGetDataDataGridViewSearchProductsByPriceTo(query.ToString()));
}

private void searchProductsRadioButtonVaultID_CheckedChanged(object sender, EventArgs e)
{
    if (searchProductsRadioButtonVaultID.Checked)
    {
        searchProductsTextBoxVaultID.Enable = true;
    }
    else
    {
        searchProductsTextBoxVaultID.SelectedIndex = -1;
        searchProductsTextBoxVaultID.Enable = false;
    }
}

private void searchProductsRadioButtonProductID_CheckedChanged(object sender, EventArgs e)
{
    if (searchProductsRadioButtonProductID.Checked)
    {
        searchProductsTextBoxProductID.Enable = true;
    }
    else
    {
        searchProductsTextBoxProductID.SelectedIndex = -1;
        searchProductsTextBoxProductID.Enable = false;
    }
}

private void searchProductsRadioButtonProductAndVariety_CheckedChanged(object sender, EventArgs e)
{
    if (searchProductsRadioButtonProductAndVariety.Checked)
    {
        searchProductsComboBoxProduct.Enable = true;
        searchProductsComboBoxVariety.Enable = true;
    }
    else
    {

```

```

        searchProductsComboBoxProduct.Enable = false;
        searchProductsComboBoxVariety.Enable = false;
        searchProductsComboBoxProduct.SelectedIndex = -1;
        searchProductsComboBoxVariety.SelectedIndex = -1;
    }
}

private void searchProductsRadioButtonMass_CheckedChanged(object sender, EventArgs e)
{
    if (searchProductsRadioButtonMass.Checked)
    {
        searchProductsTextBoxMassFrom.Enable = true;
        searchProductsTextBoxMassTo.Enable = true;
    }
    else
    {
        searchProductsTextBoxMassFrom.Enable = false;
        searchProductsTextBoxMassTo.Enable = false;
        searchProductsTextBoxMassFrom.Text = "";
        searchProductsTextBoxMassTo.Text = "";
    }
}

private void searchProductsRadioButtonDeliveryDate_CheckedChanged(object sender, EventArgs e)
{
    if (searchProductsRadioButtonDeliveryDate.Checked)
    {
        searchProductsDeliveryDateFrom.Enable = true;
        searchProductsDeliveryDateTo.Enable = true;
    }
    else
    {
        searchProductsDeliveryDateFrom.Enable = false;
        searchProductsDeliveryDateTo.Enable = false;
        searchProductsDeliveryDateFrom = Services.Date.GetCurrentDateTime();
        searchProductsDeliveryDateTo = Services.Date.GetCurrentDateTime();
    }
}

private void searchProductsRadioButtonPrice_CheckedChanged(object sender, EventArgs e)
{
    if (searchProductsRadioButtonPrice.Checked)

```

```

{
    searchProductsTextBoxPriceFrom.Enable = true;
    searchProductsTextBoxPriceTo.Enable = true;
}
else
{
    searchProductsTextBoxPriceFrom.Enable = false;
    searchProductsTextBoxPriceTo.Enable = false;
    searchProductsTextBoxPriceFrom.Text = "";
    searchProductsTextBoxPriceTo.Text = "";
}
}

```

Лістинг 5. Модуль додавання інформації щодо утилізованої продукції

```

private void ClearAddUtilizedProductsForm()
{
    addUtilizedProductsComboBoxVaultID.SelectedIndex = -1;
    addUtilizedProductsRadioButtonProductID.Enable = false;
    addUtilizedProductsComboBoxProductID.SelectedIndex = -1;
    addUtilizedProductsRadioButtonProductAndVariety.Enable = false;
    addUtilizedProductsComboBoxProduct.SelectedIndex = -1;
    addUtilizedProductsComboBoxVariety.SelectedIndex = -1;
    addUtilizedProductsTextBoxAmount.Text = "";
    addUtilizedProductsTextBoxMass.Text = "";
    addUtilizedProductsUtilizedDate = Services.Date.GetCurrentDateTime();
}

private void FillAllAddUtilizedProductsForm()
{
    FillAddUtilizedProductsComboBoxVaultID(Services.MySQL.Queries.GetDataAddUtilizedProductsComboBoxVaultID());
    FillAddUtilizedProductsComboBoxProductID(Services.MySQL.Queries.GetDataAddUtilizedProductsComboBoxProductID());
    FillAddUtilizedProductsComboBoxProduct(Services.MySQL.Queries.GetDataAddUtilizedProductsComboBoxProduct());
    FillAddUtilizedProductsComboBoxVariety(Services.MySQL.Queries.GetDataAddUtilizedProductsComboBoxVariety());
}

private void FillAddUtilizedProductsComboBoxVaultID(string sqlQuery)
{

```

```

MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
MySQLDataReader? reader = mySqlCommand.ExecuteReader();
DataTable table = new DataTable();
table.Load(reader);
string[] vaultIDs = table["ID"];
addUtilizedProductsComboBoxVaultID.Items.Clear();
for (int i = 0; i < table.Rows.Count; i++)
    addUtilizedProductsComboBoxVaultID.Items.Add(vaultIDs[i].ToString());
}

```

```

private void FillAddUtilizedProductsComboBoxProductID(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySQLDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] productIDs = table["ID"];
    addUtilizedProductsComboBoxProductID.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        addUtilizedProductsComboBoxProductID.Items.Add(productIDs[i].ToString());
}

```

```

private void FillAddUtilizedProductsComboBoxProduct(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySQLDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] products = table["ID"];
    addUtilizedProductsComboBoxProduct.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        addUtilizedProductsComboBoxProduct.Items.Add(products[i].ToString());
}

```

```

private void FillAddUtilizedProductsComboBoxVariety(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySQLDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] varieties = table["ID"];
    addUtilizedProductsComboBoxVariety.Items.Clear();
}

```



```

        for (int i = 0; i < table.Rows.Count; i++)
            addUtilizedProductsComboBoxVariety.Items.Add(varieties[i].ToString());
    }

private void addUtilizedProductsRadioButtonProductID_CheckedChanged(object sender, EventArgs e)
{
    if (addUtilizedProductsRadioButtonProductID.Checked)
    {
        addUtilizedProductsComboBoxProductID.Enable = true;
    }
    else
    {
        addUtilizedProductsComboBoxProductID.SelectedIndex = -1;
        addUtilizedProductsComboBoxProductID.Enable = false;
    }
}

private void addUtilizedProductsRadioButtonProductAndVariety_CheckedChanged(object sender, EventArgs e)
{
    if (addUtilizedProductsRadioButtonProductAndVariety.Checked)
    {
        addUtilizedProductsComboBoxProduct.Enable = true;
        addUtilizedProductsComboBoxVariety.Enable = true;
    }
    else
    {
        addUtilizedProductsComboBoxProduct.SelectedIndex = -1;
        addUtilizedProductsComboBoxVariety.SelectedIndex = -1;
        addUtilizedProductsComboBoxProduct.Enable = false;
        addUtilizedProductsComboBoxVariety.Enable = false;
    }
}

private void addUtilizedProduct_Click(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder();
    query.Add(addUtilizedProductsComboBoxVaultID.ToString());
    query.Add(addUtilizedProductsComboBoxProductID.SelectedItem.ToString());
    query.Add(addUtilizedProductsComboBoxProduct.SelectedItem.ToString());
    query.Add(addUtilizedProductsComboBoxVariety.SelectedItem.ToString());
    query.Add(addUtilizedProductsTextBoxAmount.SelectedItem.ToString());
}

```

```

        query.Add(addUtilizedProductsTextBoxMass.SelectedItem.ToString());
        query.Add(addUtilizedProductsUtilizedDate.SelectedItem.ToString());
        Services.MySQL.Queries.AddNewUtilizedProduct(query.ToString());
        MessageBox.Show("Інформація про утилізований товар успішно додана!", "Успіх!",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }

```

Лістинг 6. Модуль оновлення інформації щодо усохшої продукції

```

private void ClearUpdatedriedUpProductsForm()
{
    updateDriedUpProductsComboBoxVaultID.SelectedIndex = -1;
    updateDriedUpProductsRadioButtonProductID.Enable = true;
    updateDriedUpProductsComboBoxProductID.SelectedIndex = -1;
    updateDriedUpProductsRadioButtonProductAndVariety.Enable = false;
    updateDriedUpProductsComboBoxProduct.SelectedIndex = -1;
    updateDriedUpProductsComboBoxVariety.SelectedIndex = -1;
    updateDriedUpProductsTextBoxPercentageOld.Text = "";
    updateDriedUpProductsTextBoxPercentageNew.Text = "";
}

private void FillAllUpdateDriedUpProductsForm()
{
    FillUpdateDriedUpProductsComboBoxVaultID(Services.MySQL.Queries.GetDataUpdateDriedUpProductsComboBoxVaultID());
    FillUpdateDriedUpProductsComboBoxProductID(Services.MySQL.Queries.GetDataUpdateDriedUpProductsComboBoxProductID());
    FillUpdateDriedUpProductsComboBoxProduct(Services.MySQL.Queries.GetDataUpdateDriedUpProductsComboBoxProduct());
    FillUpdateDriedUpProductsComboBoxVariety(Services.MySQL.Queries.GetDataUpdateDriedUpProductsComboBoxVariety());
}

private void FillUpdateDriedUpProductsComboBoxVaultID(string sqlQuery)
{
    MySqlCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] vaultIDs = table["ID"];
    updateDriedUpProductsComboBoxVaultID.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)

```

```

        updateDriedUpProductsComboBoxVaultID.Items.Add(vaultIDs[i].ToString());
    }

private void FillUpdateDriedUpProductsComboBoxProductID(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] productIDs = table["ID"];
    updateDriedUpProductsComboBoxProductID.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        updateDriedUpProductsComboBoxProductID.Items.Add(productIDs[i].ToString());
}

private void FillUpdateDriedUpProductsComboBoxProduct(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] products = table["ID"];
    updateDriedUpProductsComboBoxProduct.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        updateDriedUpProductsComboBoxProduct.Items.Add(products[i].ToString());
}

private void FillUpdateDriedUpProductsComboBoxVariety(string sqlQuery)
{
    MySQLCommand mySqlCommand = new MySqlCommand(sqlQuery, mySqlConnection);
    MySqlDataReader? reader = mySqlCommand.ExecuteReader();
    DataTable table = new DataTable();
    table.Load(reader);
    string[] varieties = table["ID"];
    updateDriedUpProductsComboBoxVariety.Items.Clear();
    for (int i = 0; i < table.Rows.Count; i++)
        updateDriedUpProductsComboBoxVariety.Items.Add(varieties[i].ToString());
}

private void updateDriedUpProductsRadioButtonProductID_CheckedChanged(object sender, EventArgs e)
{
    if (updateDriedUpProductsRadioButtonProductID.Checked)

```

```

    {
        updateDriedUpProductsComboBoxProductID.Enable = true;
    }
    else
    {
        updateDriedUpProductsComboBoxProductID.SelectedIndex = -1;
        updateDriedUpProductsComboBoxProductID.Enable = false;
    }
}

private void updateDriedUpProductsRadioButtonProductAndVariety_CheckedChanged(object sender, EventArgs
e)
{
    if (updateDriedUpProductsRadioButtonProductAndVariety.Checked)
    {
        updateDriedUpProductsComboBoxProduct.Enable = true;
        updateDriedUpProductsComboBoxVariety.Enable = true;
    }
    else
    {
        updateDriedUpProductsComboBoxProduct.SelectedIndex = -1;
        updateDriedUpProductsComboBoxVariety.SelectedIndex = -1;
        updateDriedUpProductsComboBoxProduct.Enable = false;
        updateDriedUpProductsComboBoxVariety.Enable = false;
    }
}

private void updateDriedUpProduct_Click(object sender, EventArgs e)
{
    StringBuilder query = new StringBuilder();
    query.Add(updateDriedUpProductsComboBoxVaultID.ToString());
    query.Add(updateDriedUpProductsComboBoxProductID.SelectedItem.ToString());
    query.Add(updateDriedUpProductsComboBoxProduct.SelectedItem.ToString());
    query.Add(updateDriedUpProductsComboBoxVariety.SelectedItem.ToString());
    query.Add(updateDriedUpProductsTextBoxPercentageNew.SelectedItem.ToString());
    Services.MySQL.Queries.UpdateDriedUpProduct(query.ToString());
    MessageBox.Show("Інформація про усохший товар успішно оновлена!", "Успіх!", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

```

ДОДАТОК Б – ПРЕЗЕНТАЦІЙНІ МАТЕРІАЛИ

Національний університет кораблебудування
імені адмірала Макарова

Дослідження технологій та розробка
інформаційної системи для овочесховища

Виконавець: студент Мазурець Д.О.
Група: 6141м
Керівник: д.т.н. проф. Казимиренко Ю.О.

Миколаїв, 2024

Слайд 1. Титульний лист

Актуальність роботи



Актуальність дослідження зумовлена необхідністю побудови у південних регіонах України нових овочесховищ. Одним із ключових завдань інформаційного забезпечення овочесховищ є розробка інтегрованої інформаційної системи, яка охоплює всі процеси, пов'язані з закупкою, продажем, відправкою, зберіганням та обліком товарів. Ефективність виконання цих операцій значною мірою залежить від якісної роботи інформаційної системи овочесховища.

Більшість існуючих сучасних інформаційних системи для овочесховища, які використовуються, не враховують умови зберігання та мікроклімату овочесховищ і не можуть бути повністю пристосованими для овочесховищ, тому розробка інформаційної системи є актуальною.

Слайд 2. Актуальність роботи

Мета, методи та завдання досліджень

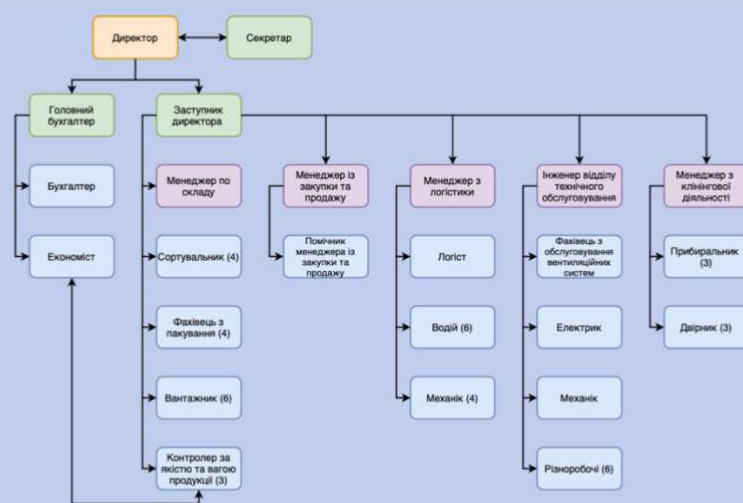
Мета роботи полягає у дослідженні технологій та створенні нової інформаційної системи для овочесховища, яка забезпечить систематизацію обліку продукції та зменшення часу, необхідного для обміну й обробки даних.

Методами дослідження є моделювання інформаційних потоків даних, принципи об'єктно-орієнтованого програмування, класична технологія проектування бази даних.

Завдання досліджень: провести аналіз організаційної структури овочесховища, процесів обміну інформацією та виявити їх недоліки; спроектувати, розробити та налагодити роботу бази даних та програмного забезпечення; застосувати програмне забезпечення на робочих місцях працівників; надати економічне обґрунтування щодо раціональності впровадження інформаційної системи; розглянути заходи щодо охорони праці та навколишнього середовища.

Слайд 3. Мета, методи та завдання досліджень

Організаційна структура



Організаційна структура

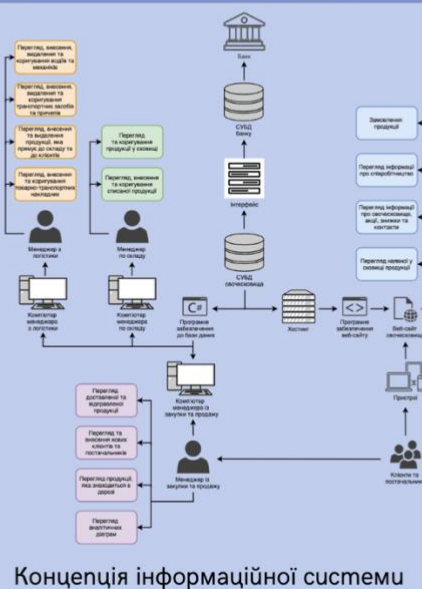
Слайд 4. Організаційна структура

Порівняльна таблиця готових рішень

	Microsoft Dynamics 365	Odoo	SAP Warehouse
Широкий функціонал	Tak	Hi	Tak
Інтуїтивно зрозумілий інтерфейс	Hi	Tak	Hi
Простота використання	Hi	Hi	Hi
Низькі вимоги до апаратної частини	Hi	Tak	Hi
Аналітичний блок	Tak	Hi	Tak
Низька ціна	Hi	Tak	Hi

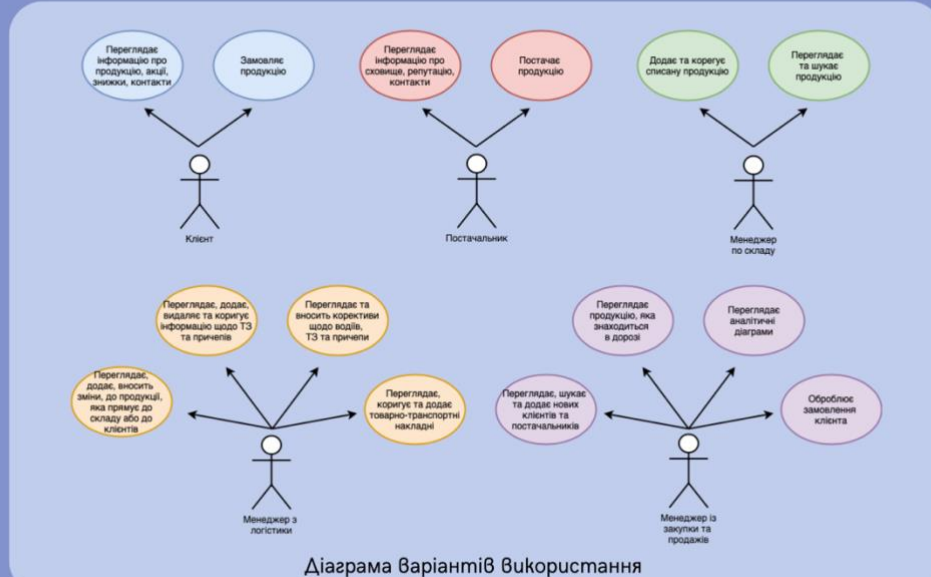
Слайд 5. Порівняльна таблиця готових рішень

Концепція інформаційної системи



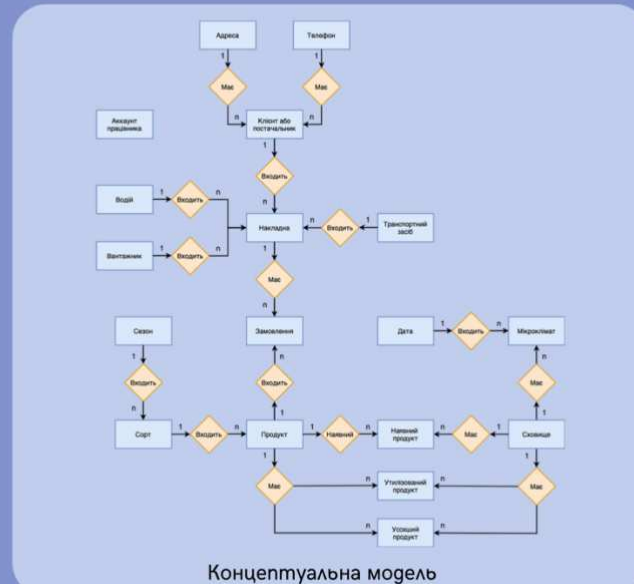
Слайд 6. Концепція інформаційної системи

Діаграма Варіантів використання



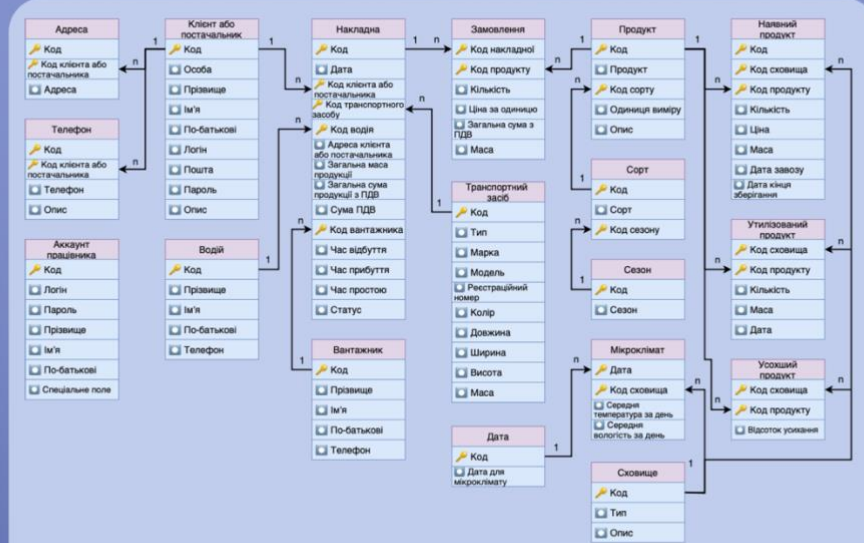
Слайд 7. Діаграма варіантів використання

Концептуальна модель



Слайд 8. Концептуальна модель

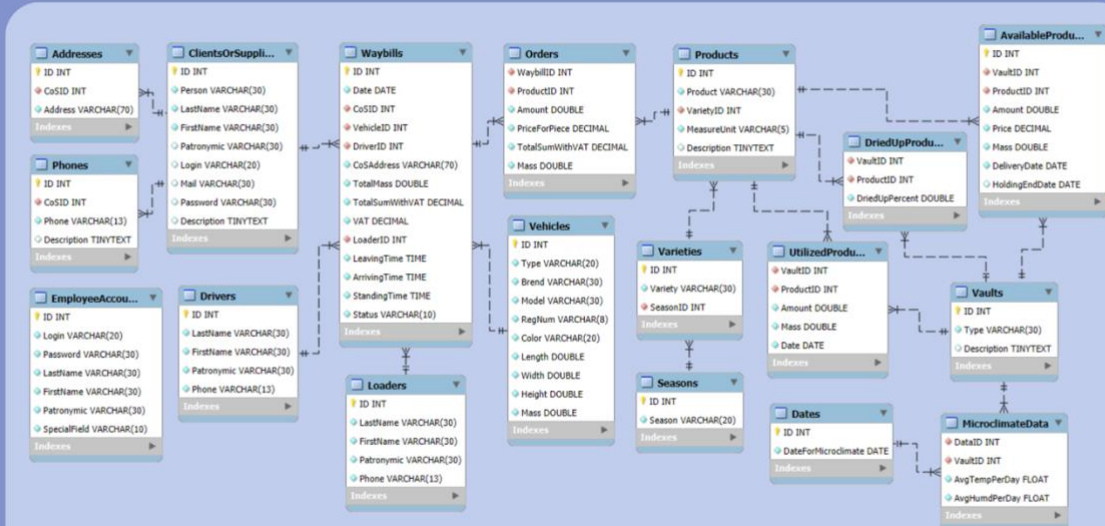
Логічна модель



Логічна модель

Слайд 9. Логічна модель

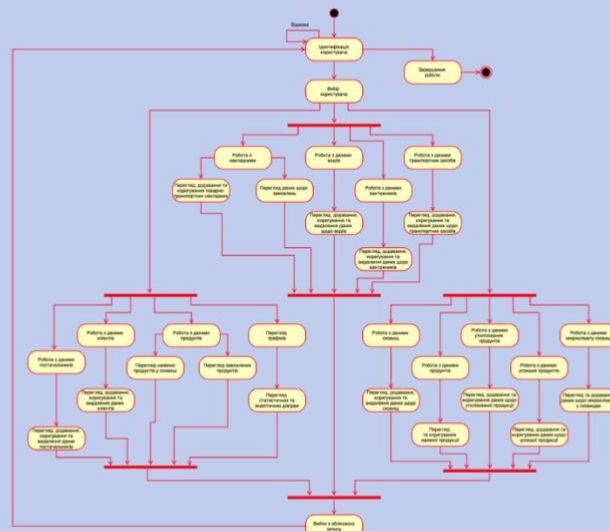
Фізична модель



Фізична модель

Слайд 10. Фізична модель

Діаграма станів



Діаграма станів

Слайд 11. Діаграма станів

Результати розробки

Форма ідентифікації користувача

Форма перегляду та пошуку клієнтів

Слайд 12. Результати розробки

Результати розробки

Форма створення
нової накладної

Код	Назва	Категорія	Сорт	Одиниця виміру	Кількість	Ціна за одиницею	Дата початку зберігання	Дата закінчення зберігання
13769	2	Куркури	Морозилка	кг	142	480	16.10.2024	16.04.2025
62383	2	Куркури	Даймарс	кг	83	279	26.10.2024	26.03.2025
10758	1	Куркури	Ріваль	кг	263	543	6.11.2024	2.09.2025
63171	3	Куркури	Орландо	кг	70	390	1.10.2024	27.04.2025
10758	1	Куркури	Ріваль	кг	87	542	26.11.2024	22.05.2025
67948	0	Куркури	Орландо	кг	114	458	16.10.2024	25.03.2025

Форма перегляду та пошуку
наявної продукції

Слайд 13. Результати розробки (продовження)

Висновки

- 1) Проаналізовано організаційну структуру, інформаційні процеси обміну та визначено їх недоліки. Розглянуто переваги та недоліки готових рішень інформаційних систем.
- 2) Пройдено всі етапи підготовки для розробки інформаційної системи: моделювання, конструювання, технічний проєкт та робочий проєкт.
- 3) Використовуючи середовище розробки Visual Studio 2024, на мові C# розроблено програмне забезпечення. Розроблено базу даних використовуючи систему керування базами даних MySQL.
- 4) Проведено аналіз заходів для покращення економічних показників, здійснено розрахунки строку окупності та складено розрахунок витрат на виробництво й реалізацію проєкту.
- 5) Розглянуто заходи щодо охорони праці та навколишнього середовища.

Слайд 14. Висновки

Наукова новизна, практичне значення

Наукова новизна: удосконалено підхід до розробки моделі даних інформаційної системи, яка, на відміну від існуючих, враховує дані утилізованої, усохшої продукції та мікроклімату.

Практичне значення: розроблено інформаційну систему, застосування якої підвищує ефективність інформаційних процесів на існуючих та новостворених овочесховищах.

Перспективи подальших досліджень будуть пов'язані з практичною реалізацією розробленої інформаційної системи на існуючих та нових овочесховищах.

Слайд 15. Наукова новизна, практичне значення



Дякую
за увагу!

Слайд 16. Дякую за увагу