

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Факультет автоматизації і комп'ютерних систем
Кафедра інформаційних технологій, штучного інтелекту і кібербезпеки

«До захисту в ЕК»

Декан факультету

Андрій ФОРСЮК

(підпис)

(ім'я та прізвище)

«12» лютого 2024 р.

«До захисту допущено»

Завідувач кафедри

Сергій ГРИБКОВ

(підпис)

(ім'я та прізвище)

«12» лютого 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА

зі спеціальності 122 «Комп'ютерні науки»

(код і назва спеціальності)

освітньо-професійної програми «Інформаційні управляючі системи та технології»
на тему: «Дослідження та розробка інформаційної системи підтримки управління
замовленнями та доставки бутильованої води з використанням технології Microsoft
MAUI»

Виконав: здобувач 2 курсу, групи ІС-2-3М

Павленко Віктор Вікторович

(прізвище, ім'я, по батькові повністю)

(підпис)

Керівник: Грибков Сергій Віталійович

(прізвище, ім'я та по батькові повністю)

(підпис)

Консультанти

(ім'я та прізвище)

(підпис)

Рецензент

Арслав Смітюх

(ім'я та прізвище)

(підпис)

Я як здобувач(ка) Національного університету харчових технологій розумію і підтримую політику університету з академічної доброчесності. Я не надавав(-ла) і не одержував(-ла) незарядженої допомоги під час підготовки цієї роботи. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач

(підпис)

Київ — 2024 р.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ

Інститут (факультет) Автоматизації і комп'ютерних систем
 Кафедра Інформаційних технологій, штучного інтелекту і кібербезпеки
 Освітній ступінь магістр
 Спеціальність 122 «Комп'ютерні науки»
(код і назва)
 Освітньо-професійна програма Інформаційні управляючі системи та технології
(назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри інформаційних технологій, штучного інтелекту і кібербезпеки

Сергій ГРИБКОВ
 «19» грудня 2023 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Павленка Віктора Вікторовича

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та розробка інформаційної системи підтримки управління замовленнями та доставки бутильованої води з використанням технології Microsoft MAUI.

керівник роботи: Грибков Сергій Віталійович, професор, д.т.н.

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

затверджені наказом закладу вищої освіти від «19» грудня 2023 р. № 1006-кс

2. Строк подання здобувачем роботи: 22.01.2024

3. Вихідні дані до роботи: Інформація про водопостачальні підприємства, Положення Державної соціальної програми поліпшення стану охорони праці та виробничого середовища, міжнародні та національні стандарти оцінки якості води, технологія Microsoft MAUI

4. Зміст пояснювальної записки (перелік питань, які потрібно розробити):

Розділ 1. Дослідження предметної області та постановка задачі дослідження.

Розділ 2. Дослідження методів програмних платформ.

Розділ 3. Створення та апробація системи підтримки формування замовлення та організації постачання фасованої води.

5. Перелік графічного матеріалу:

6. Консультанти розділів роботи:

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Грибков С.В., професор, доктор технічних наук		
2	Грибков С.В., професор, доктор технічних наук		
3	Грибков С.В., професор, доктор технічних наук		

7. Дата видачі завдання:

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів виконання кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Перший розділ кваліфікаційної роботи	27.11.2023	Виконано
2	Другий розділ кваліфікаційної роботи	13.12.2023	Виконано
3	Третій розділ кваліфікаційної роботи	26.12.2023	Виконано
4	Оформлення пояснювальної записки	20.01.2024	Виконано
5	Створення автореферату та презентації	21.01.2024	Виконано

Здобувач

(підпис)

Павленко В. В.

(прізвище та ініціали)

Керівник роботи

(підпис)

Грибков С. В.

(прізвище та ініціали)

АНОТАЦІЯ

Кваліфікаційна робота на тему «Дослідження та розробка інформаційної системи підтримки управління замовленнями та доставки бутильованої води з використанням технології Microsoft MAUI» розроблена здобувачем ступеня магістр за освітньо-професійною програмою «Інформаційні управляючі системи і технології» спеціальності 122 «Комп'ютерні науки» Павленком В.В. та складається з 106 сторінок, 3 розділів, 22 рисунків, 5 таблиць, 5 додатків та 18 літературних джерел.

Метою кваліфікаційної роботи є дослідження та розробки інформаційної системи підтримки управління замовленнями та доставки бутильованої води з використанням технології Microsoft MAUI є створення ефективної, гнучкої та зручної в користуванні системи, яка поліпшить і оптимізує процеси замовлення та доставки води.

В роботі проведено дослідження інформаційної системи підтримки управління замовленнями та доставки бутильованої води методом системного аналізу предметної області, проаналізовано та синтезовано проектні рішення щодо створення прототипів інформаційних систем.

Практичне значення отриманих результатів полягає в підвищенні ефективності управління замовленнями та доставкою, зниженні витрат, підвищенні задоволеності споживачів та підвищенні загальної продуктивності бізнесу.

Інформаційна система була створена за допомогою технології Microsoft та з використанням сучасних методів та засобів розробки, а саме Microsoft Visual Studio, SQLite Maestro, DB Browser (SQLite).

Ключові слова: MICROSOFT MAUI, УПРАВЛІННЯ ЗАМОВЛЕННЯМИ, ЛОГІСТИКА ДОСТАВКИ, БУТИЛЬОВАНА ВОДА, ІНФОРМАЦІЙНА СИСТЕМА, ОПТИМІЗАЦІЯ ПРОЦЕСІВ, КРОСПЛАТФОРМЕНЕ ПРОГРАМУВАННЯ, ТЕХНОЛОГІЯ ДОСТАВКИ ВОДИ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

SUMMARY

The qualification work on the topic "Research and Development of an Information System for Order Management and Delivery of Bottled Water Using Microsoft MAUI Technology" was developed by the Master's degree candidate in the educational-professional program "Information Control Systems and Technologies," specialty 122 "Computer Science," by Pavlenko V.V. It consists of 106 pages, 3 chapters, 22 figures, 5 tables, 5 appendices, and 18 literary sources.

The aim of the qualification work is to research and develop an information system for supporting the management of orders and delivery of bottled water using Microsoft MAUI technology, to create an effective, flexible, and user-friendly system that will improve and optimize the processes of ordering and delivering water.

The work involved researching the information system for managing orders and delivering bottled water by the method of systemic analysis of the subject area, analyzing, and synthesizing design solutions for creating prototypes of information systems.

The practical significance of the obtained results lies in increasing the efficiency of order management and delivery, reducing costs, increasing customer satisfaction, and improving the overall productivity of the business.

The information system was created using Microsoft technology and modern methods and tools of development, namely Microsoft Visual Studio, SQLite Maestro, DB Browser (SQLite).

Key words: MICROSOFT MAUI, ORDER MANAGEMENT, DELIVERY LOGISTICS, BOTTLED WATER, INFORMATION SYSTEM, PROCESS OPTIMIZATION, CROSS-PLATFORM PROGRAMMING, WATER DELIVERY TECHNOLOGY, SOFTWARE.

ЗМІСТ

Вступ.....	9
РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	12
1.1 Дослідження діяльності підприємств по виробництву води та їх організаційних структур.....	12
1.2 Технологія та система очищення води.....	14
1.3 Маркетингові стратегії виробників фасованого водопостачання.....	24
1.4 Актуальність наукових питань.....	28
1.5 Сертифікація якості питної води.....	36
1.6 Розрахунок витрат на очищення питної води.....	40
1.7 Постановка задачі.....	42
1.8 Висновок до розділу № 1.....	43
РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГРАМНИХ ПЛАТФОРМ.....	44
2.1 Огляд Microsoft .NET MAUI.....	44
2.2 Можливості Microsoft MAUI.....	47
2.3 Model-View-ViewModel (MVVM).....	51
2.4 Висновок до розділу № 2.....	57
РОЗДІЛ 3. СТВОРЕННЯ ТА АПРОБАЦІЯ СИСТЕМИ ПІДТРИМКИ ФОРМУВАННЯ ЗАМОВЛЕННЯ ТА ОРГАНІЗАЦІЇ ПОСТАЧАННЯ ФАСОВАНОЇ ВОДИ.....	57
3.1 Методи виконання задачі.....	57
3.2 Створення бази даних.....	58
3.3 Обробка винятків SQLight (SqlException).....	60

3.4 Вказівки для користувачів.....	69
3.5 Висновок до розділу № 3.....	70
ВИСНОВКИ.....	74
Список використаної літератури.....	71
ДОДАТКИ.....	73

ВСТУП

Актуальність теми. Сфера доставки бутильованої води в останні роки зазнає значних змін, що вимагає від компаній більшої гнучкості та оперативності у вирішенні логістичних та управлінських завдань. Сучасний ринок висуває нові вимоги до якості обслуговування, швидкості доставки, а також до індивідуалізації замовлень. В цьому контексті, розробка ефективної інформаційної системи (ІС) для управління замовленнями та доставкою бутильованої води стає актуальною задачею.

У сучасному світі, де інформаційні технології стрімко розвиваються та глибоко проникають у різні сфери нашого життя, питання оптимізації бізнес-процесів набуває особливої ваги. Сфера логістики та управління замовленнями не є винятком, а зростаючий попит на доставку бутильованої води створює потребу в ефективних інформаційних системах. Впровадження сучасних технологій, зокрема технології Microsoft MAUI, може значно підвищити ефективність цих процесів, забезпечити швидкість та точність у виконанні замовлень, а також поліпшити якість обслуговування клієнтів.

Зв'язок роботи з науковими програмами, планами, темами кафедри, університету. Наукова робота виконувалась згідно з науково-дослідною роботою на кафедрі інформаційних технологій, штучного інтелекту і кібербезпеки «Дослідження та використання сучасних інформаційних технологій для виконання функцій та завдань виробничого і організаційного управління підприємств харчової галузі» (0120U105386 2020–2025 рр.) Національного університету харчових технологій.

Мета дослідження. Головною метою дослідження є удосконалення проєкту управління замовленнями та доставкою бутильованої води за рахунок створення та використання інформаційних технологій, що забезпечить підвищення якості обслуговування, а також збільшення прибутку.

Для досягнення цієї мети передбачено вирішення таких завдань:

- аналіз поточних проблем у сфері управління замовленнями та доставкою води;
- дослідження та обґрунтування інформаційних технологій для розробки інформаційної системи управління замовленнями та доставкою води;
- розробка прототипу інформаційної системи;
- апробація створеної інформаційної системи.

Об'єкт дослідження. Основним об'єктом дослідження є процес управління замовленнями та доставкою бутильованої води.

Предмет дослідження. Предметом дослідження є удосконалення проєкту управління замовленнями та доставкою бутильованої води за рахунок створення та використання інформаційних технологій.

Методи дослідження:

- метод системного аналізу для визначення вимог до системи та її компонентів;
- метод природного експерименту при дослідженні якості розробленого програмного продукту;
- метод функціонального програмування для розробки функцій системи;
- метод функціонального тестування для перевірки відповідності програмного продукту визначеним вимогам;
- метод тестування взаємодії з користувачем для перевірки зручності використання та інтерфейсу програми для користувача.

Наукова новизна одержаних результатів полягає у адаптації технології Microsoft MAU для оптимізації процесів управління поставок, яка може бути адаптована для різних галузей.

Практичне значення одержаних результатів відображається в підвищенні продуктивності та ефективності робочих процесів, а також у поліпшенні задоволення клієнтів.

Особистий внесок здобувача. Автор пропонує рішення для інтеграції технології Microsoft MAUI у систему управління замовленнями та доставкою, а також розробляє ключові компоненти системи.

Апробація результатів роботи. Матеріали до даної роботи опубліковані у збірниках тез:

1. X Міжнародної науково-технічної Internet-конференції «Сучасні методи, інформаційне, програмне та технічне забезпечення систем керування організаційно-технічними та технологічними комплексами», 24 листопада 2023 [Електронний ресурс]. – К: НУХТ, 2023. – 224 с.

РОЗДІЛ 1. ДОСЛІДЖЕННЯ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Дослідження діяльності підприємств по виробництву води та їх організаційних структур

Світове споживання бутильованої води значно зросло за останнє десятиліття, і дослідження показують, що воно буде зростати й надалі. Як ми всі знаємо, бутильована вода упаковується в пластикові або скляні пляшки, і подальша класифікація вказує на те, що бутильована вода, що продається, вважається негазованою або газованою водою. Обсяг продажів бутильованої води у всьому світі (у літрах) склав 181 608 млрд, вартість продукту в США – 6,61 мільярда доларів.

Індустрія бутильованої води зростає швидкими темпами — від 8% до 10% на рік, і її дохід відрізняється від інших напоїв. Промисловість бутильованої води в США була другою за обсягом торгівлі напоями, випереджаючи пиво, молоко та каву, станом на 2003 рік. Бутильована вода слідує за безалкогольними газованими напоями.

У той час як споживання бутильованої води зросло в усьому світі, Європа спостерігала найбільше зростання, тоді як інші континенти, такі як Південна Америка та Азія, спостерігали зростання швидше. Однак Північна Америка лідирує у споживанні бутильованої води, де більше американців п'ють більше бутильованої води, ніж пива чи молока, споживаючи 9 мільярдів галонів води в 2008 році, в середньому 30 галонів на людину.

У деяких регіонах Сполучених Штатів, як стверджують джерела у Каліфорнії, регіональна водопровідна вода коштує в 560 разів дорожче, ніж галон бутильованої води. Це частина стратегії виробників бутильованої води, які за

допомогою реклами роблять бутильовану воду доступнішою для споживачів, тим самим збільшуючи попит на неї. Однак це іронічно, оскільки, згідно з даними Food and Water Watch, 48% бутильованої води, виробленої у 2009 році, було з водопровідної води.

Причина, по якій споживачі все більше споживають бутильовану воду, полягає в тому, що більшість із них боїться водопровідної води, яка вважається на 90% безпечною урядами штатів і федеральними урядами, вищого рівня безпеки, ніж бутильована вода. Крім того, промисловість бутильованої води, хоча законодавство знаходиться в розробці, є більш м'яким, ніж правила водопровідної води.

Станом на 2005 рік лише 12 відсотків із 30 мільярдів пляшок бутильованої води, проданих у Сполучених Штатах, було перероблено, в результаті чого близько 25 мільярдів пляшок було викинуто на смітник, звалище чи спалено. Основна причина полягає в тому, що перероблений пластик можна повторно використовувати лише в нехарчових продуктах. Це викликає занепокоєння, особливо тому, що щороку для виробництва пляшок для води потрібно 17 мільйонів барелів нафти.

Хоча Північна Америка лідирує у світі за споживанням бутильованої води, Китай не відстає. За даними Zenith International, у 2000 році майже кожен житель Китаю випив мільярди літрів води; у 2009 році було випито близько 21 мільярда літрів. Станом на 2011 рік загальний дохід галузі бутильованої води оцінюється в 86 мільярдів доларів. Сюди входить така вода, як негазована вода, газувана вода зі смаком, дистильована вода без смаку та дистильована вода зі смаком. Також наведено організаційну структуру підприємства (див рис. 1.1).

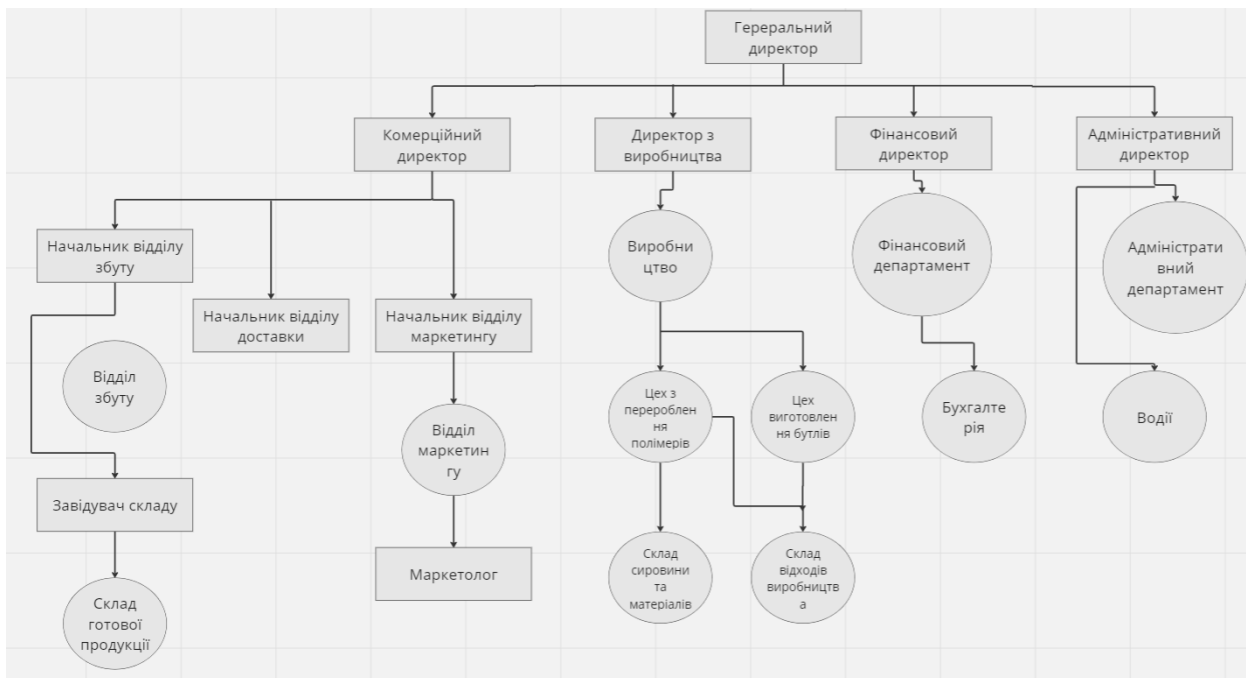


Рисунок. 1.1 Структура управління мінеральною водою на вищому рівні в компанії, що займається її реалізацією.

1.2 Технологія та система очищення води

Процес очистки води полягає у видаленні небажаних газів, хімічних елементів, біологічних забрудників та твердих частинок. Це робиться з метою забезпечення води необхідної якості для різноманітних застосувань. Найчастіше цей процес використовується для підготовки питної води для людського споживання, однак також застосовується у медичній, фармацевтичній промисловості, у виробництві хімічних продуктів та інших галузях промисловості.

Технологічні підходи до очищення води охоплюють ряд фізичних, біологічних та хімічних методів. До фізичних методів належать процеси, такі як сепарація твердих частинок через фільтрацію, осадження важких елементів з розчину та перегонка рідини для відділення компонентів. Біологічні методи включають використання мікроорганізмів у піщаних фільтрах та активованого

вугілля для поглинання забруднюючих речовин. Хімічні процеси, такі як індукування утворення флоків та додавання хлору, застосовуються для знешкодження шкідливих мікроорганізмів та забезпечення стабільності води. Додатково, застосування електромагнітного випромінювання, зокрема ультрафіолетового, є ефективним для нейтралізації мікробних забрудників.

Процес очищення води впливає на зниження рівня суспендованих та розчинених твердих речовин, включаючи металеві елементи, паразитів, бактеріальні клітини, фітопланктон, віруси та мікотичні організми. Це призводить до модифікації вмісту розчинених речовин та твердих часток у воді.

Мінімальні та максимальні концентрації забруднюючих речовин часто підпадають під дію урядових норм або міжнародних стандартів щодо якості питної води. Що стосується цільового використання води, ці правила спрямовані на забезпечення високих стандартів якості.

Візуально перевірити якість води неможливо. Стандарні побутові засоби обробки, такі як кип'ятіння або фільтри з активованим вугіллям, не можуть гарантувати видалення всіх забруднень, які можуть бути присутніми у воді з невідомих джерел. Природна джерельна вода, яка колись вважалася безпечною в 19 столітті, тепер вимагає перевірки, перш ніж приймати рішення про план лікування. Єдиний спосіб отримати необхідні дані для вибору відповідного підходу до лікування – дорогий хімічний і мікробіологічний аналіз.

Згідно з даними Всесвітньої організації охорони здоров'я (ВООЗ) у 2007 році, приблизно 1,1 мільярда людей у всьому світі не мають доступу до питної води. Небезпечна вода та неналежні санітарні умови спричиняють приголомшливі 88% із 4 мільярдів щорічних випадків діарейної хвороби, яка призводить до смерті. 1,8 мільйона людей. Зміни навколишнього середовища, наприклад безпечний доступ до води, могли запобігти 94% цих пацієнтів. Таким чином, домашні методи очищення води, такі як сонячна дезінфекція, хлорування та фільтри, у поєднанні з

безпечним зберіганням контейнерів, мають потенціал щорічно рятувати незліченну кількість життів.

У країнах, що розвиваються, основною метою охорони здоров'я є зниження смертності від захворювань, що передаються через воду.

Також доступні такі методи очищення води:

1. Завдяки вертикальному руху води та доданому шару активованого вугілля або антрацитового вугілля піщаний фільтр є помірним рівнем складності. Його верхній шар усуває органічні сполуки, що впливають на смак і запах. Однак простір між частинками піску не може вловити найдрібніші зважені частинки, що вимагає більш складного процесу фільтрації. Частинки зазвичай проникають через поверхневий шар і потрапляють у щілини та пори або прикріплюються до частинок піску. Ефективність фільтра значною мірою залежить від його товщини. Фундаментальною для роботи фільтра є його характеристика: якби початковий шар піску покривав усі частинки, фільтр легко засмітився б.

2. Покладаючись на біологічне очищення, а не на фізичну фільтрацію, повільний піщаний фільтр творить чудеса. Майстерно виготовлені, ці фільтри містять сортований пісок найбільшого розміру та три сторони на дні. На протилежному кінці знаходиться найменший пісок. Після очищення вода транспортується на дезінфекцію через дренажі, розташовані біля основи. Утворення масивного біологічного шару, відомого як *Schmutzdecke* або зооглеальна подушка, перешкоджає фільтрації в повільному піщаному фільтрі. Цей шар відповідає за ефективне очищення води під час процесу очищення, тоді як нижній шар піску діє як природне середовище існування для *Schmutzdecke*. У результаті повільні піщані фільтри можуть зберігатися протягом тривалого періоду часу, навіть місяців або років, якщо попередня обробка виконана належним чином і вода, що надходить, має низький вміст поживних речовин – результат, який майже не досягається звичайними фізичними обробками. При мінімальних необхідних

заходах дезінфекції вода може безпечно текти розподільними мережами через дефіцитні рівні поживних речовин. Це призводить до зниження витрат споживачів, оскільки не потрібна обробка залишкового хлору на поверхнях фільтрів. Самі фільтраційні компоненти не очищаються, натомість покладаються на верхній шар піску, який прилипає та блокує потік, коли зростання мікробів стає проблемою.

3. Для цілей фільтрації питної та стічної води широко використовується мембранна фільтрація. Завдяки здатності видаляти частинки розміром понад 0,2 мікрона ця техніка може видаляти бактерії та криптоспоридії, щоб забезпечити безпечну питну воду. Мембранні фільтри вважаються ефективним третинним очисником, ідеальним для тих, хто хоче повторно використовувати воду для незначних побутових цілей, у промисловості або перед випуском у нижню течію річки. Ця техніка широко використовується в промисловості, зокрема в сценаріях передачі тепла, таких як вода в вежах. Однак важливо зазначити, що розчинені речовини, включаючи іони важких металів і нітрати, неможливо відфільтрувати, попри ефективність мембранних фільтрів.

4. Відфільтрування іонів та інших розчинених речовин у рідині здійснюється за допомогою спеціальних мембран із фіксованими зарядами, утворюючи мікроосмотичні пори, які можуть ефективно видаляти крихітні частинки, не покладаючись на коагулянти. Мембранне середовище, що використовується, відіграє ключову роль у визначенні як тиску, необхідного для проходження води, так і розміру мікроорганізмів, які можуть бути відфільтровані. Інший метод відомий як іонний обмін, який передбачає використання унікальної смоли або целюлозної колонки для заміни небажаних іонів. Позбутися небажаних речовин у воді можна різними способами, але одним із найпопулярніших методів є пом'якшення води. Це передбачає позбавлення від іонів магнію та кальцію шляхом заміни їх на більш обмінні іони, такі як Na^+ або K^+ . Використовуючи переваги іонообмінних смол, можна також усунути токсичні іони, такі як ртуть, свинець,

мідь і нітрати. Жорстку воду з високим вмістом іонів кальцію та магнію також можна обробити вапном (оксидом кальцію) та/або кальцинованою содою (карбонатом натрію), щоб спричинити випадання карбонату кальцію з розчину шляхом загальної іонізації.

5. Додавання дезінфікуючих хімічних речовин і фільтрація шкідливих мікроорганізмів є двома методами, які використовуються в дезінфекції. Метою дезінфекції є усунення будь-яких мікроорганізмів, які проходять через фільтри, і забезпечення залишкової дози дезінфікуючого засобу для запобігання розвитку шкідливих організмів у системах зберігання та розподілу. Серед вірусів, бактерій і найпростіших, які можуть бути присутніми у воді, є сальмонела, холера, кампілобактер, шигелла, лямблія та інші криптоспоридії. Воду зазвичай зберігають у резервуарі для тимчасового зберігання або іншому подібному середовищі після введення хімічного дезінфікуючого засобу для забезпечення дезінфікуючого ефекту.

6. Традиційним підходом до дезінфекції є хлор та його похідні, включаючи хлорамін та діоксид хлору. Завдяки потужним окисним властивостям хлор може швидко знищувати численні шкідливі мікроорганізми. Проте, оскільки хлор може утворювати вибухонебезпечний газ або токсичні сполуки, з ним потрібно поводитися обережно. Гіпохлорит натрію, недорогий розчин, який зазвичай використовують у побутових котлах, який виділяє вільний хлор при розчиненні у воді, значно зменшує газоутворення. Крім того, електроліз розчину, що містить значну кількість солі, також може давати розчинний хлор на місці. Твердий гіпохлорит кальцію при контакті з водою виділяє хлор. Попри зручність автоматизованих газових балонів і тумбів, поводження з твердою формою вимагає більшої фізичної взаємодії через пакування та затягування. Хоча виробництво рідкого гіпохлориту натрію (дешевше) безпечніше, ніж використання

газоподібного хлору або твердого хлору. Запас безпеки для питної води підтримується з рівнями хлору до 4 міліграмів на літр (4 ppm).

7. Діоксид хлору для дезінфекції відрізняється від елементарного хлору. Попри його ефективність, це не найпоширеніший метод через можливість надмірного виробництва хлориту, який також обмежений у США.

8. Хлорамін як дезінфікуючий засіб зростає, використовуючи хлорамінування. Хоча він може не забезпечити такий самий рівень окислення, як вільний хлор, його нижчий окислювально-відновний потенціал забезпечує більш тривалий залишковий ефект. Додатковою перевагою є те, що він не легко створює побічні продукти дезінфекції, такі як ТХМ або галогенні кислоти.

У промисловості та побуті використовується безліч методів очищення води. Існує незліченна кількість систем, наприклад:

1. За допомогою напівпроникних мембран зворотний осмос є унікальною технологією очищення води. Хоча він відрізняється від традиційних методів фільтрації, зворотний осмос долає осмотичний тиск і властивості коаліції, викликані хімічним потенціалом через прикладений тиск. Цей процес ефективно усуває численні молекули та іони з розчинів, що робить його цінним інструментом як для промислових процесів, так і для виробництва питної води. В результаті тиску, який утримується з одного боку напівпроникної мембрани, чистий розчинник - у формі води - проходить на інший бік, залишаючи розчинену речовину. Система зворотного осмосу функціонує як механізм очищення води, реалізуючи «селективність» через свою мембрану. Його пори повинні дозволяти рухливість дрібніших компонентів (наприклад, розчинника), одночасно блокуючи більші молекули та іони. Успішне функціонування мембрани залежить від потужності фільтрів, які захищають від шкоди сильними окислювачами (наприклад, хлором) і механічним сміттям. Цікаво, що низькопотенційні фільтри корелюють зі зменшенням присутності таких захисних комплексів.

2. Система очищення води eSpring — це необхідне рішення для води що відсвіжує, на яке можна з упевненістю покластися. Це перший власний очисний пристрій, який використовує технологію ультрафіолетового випромінювання, виготовлений у США, з вугільним блоковим фільтром і передовою електронною системою моніторингу. За допомогою вугільного фільтра з води видаляється понад 140 забруднюючих речовин і мікрочастинок. Крім того, ультрафіолетові промені здатні знищити понад 99,99% водних мікроорганізмів. Не хвилюйтеся, електронна система моніторингу ефективно працює та обслуговує пристрій, повідомляючи вас про необхідність заміни картриджа. Поєднання різних технологій - це те, що відрізняє нашу систему. eSpring знищує понад 99,99% небезпечних бактерій і вірусів, які містяться у воді, а також покращує її смак і прозорість. Не кажучи вже про те, що eSpring також збагачує воду, захищаючи її від накопичення осаду. Використання вугільного фільтра дозволяє видалити з води мікрочастинок та понад 140 шкідливих домішок.

3. Використання тканинного фільтра – один із найпростіших способів очищення води. Вони не потребують спеціального налаштування та дуже прості у використанні. Ви навіть можете взяти їх із собою в подорож, оскільки вони портативні. Крім того, вони часто доступні. Фільтр працює, коли вода проходить через спеціальні іонообмінні або фільтруючі матеріали, такі як кварцовий пісок або полімери. Найефективнішим матеріалом для цього є активоване вугілля. Крім того, срібло можна додавати в картриджі, щоб допомогти знезаразити воду. У картриджі міститься особлива суміш речовин, яка підібрана вручну, щоб забезпечити воду необхідними факторами для отримання смачної, першокласної та безпечної питної води. Принаймні, так обіцяють виробники фільтрів.

4. Використання наповнювачів для хімічної зміни складу води є формою фільтрації, яка покращує якість води та змінює її хімічний склад. Хімічна фільтрація використовує різні матеріали, такі як:

- Метиленовий синій і деякі забруднюючі речовини можна видалити за допомогою вугілля або активованого вугілля, яке також називають активованим вугіллям. Крім того, їх використанням можна видалити дубильні речовини, такі як торф і легкі ліки.
- Буферуйте рН за допомогою коралів (подрібнених коралів або коралового піску), щоб спорадично підвищити його.
- Для зміни або стабілізації рівня рН часто використовують доломіт або вапнякову крихту.
- Зниження рН можна досягти за допомогою торфу.
- Щоб позбутися аміаку, нанесіть цеоліт, який є різновидом природної іонообмінної смоли, у довільні проміжки часу на уражену ділянку.
- Використання цеоліту має бути лише тимчасовим виправленням для певних обставин, наприклад, щоб врівноважити несподіваний сплеск рівня аміаку. Однак пам'ятайте, що цеоліт не є адекватним заміником правильного колообігу азоту. Його вплив швидко зникає, залишаючи рибу вразливою до небезпечно високого рівня аміаку. Деревне вугілля також має обмежений термін служби та потребує періодичної заміни. Вливаючи свій крутий момент в акваріум і видаляючи забруднюючі речовини з ґрунту, акваріумний фільтр позбавляє від необхідності постійного використання деревного вугілля. Після лікування він може навіть виділяти лікарські речовини, але слід бути обережним, щоб уникнути надмірного використання деревного вугілля. Майте на увазі, що торф вимагає регулярного оновлення, тоді як корал і вапняк зберігають ефективність протягом тривалого часу.

У таблиці 1.1 наведено критерії, що впливають на якість води, які ми будемо використовувати для порівняння для визначення оптимального методу очищення води після отримання інформації.

Таблиця 1.1. Порівняння параметрів очищення води

	Поліпшує смак, прозорість та запах	Знижує частку мікроорганізмів	Знижує рівень твердих часток, органічних і неорганічних забруднювачів і хлору.	Зберігайте корисні мінерали у воді та забезпечуйте велику кількість води для очищення за допомогою належної системи моніторингу
eSpring. Шматочок пресованого активованого вугілля ефективно видаляє з потоків 140 хвороботворних мікроорганізмів, а УФ (ультрафіолет) знищує мікроорганізми.	Ефективно	Ефективно	Не повністю ефективно	Ефективно
Вода в пішоходах У річках від виробників якості води може	Ефективно	Невідомо	Невідомо	Невідомо

змінюватися в залежності від способу очищення				
Знятий з нього фільтр-глечик видаляє лише невелику кількість низьких рівнів.	Ефективно	Неефективно	Не повністю ефективно	Не повністю ефективно
Фільтр-глечик може очистити лише обмежену кількість води. Фільтри необхідно часто міняти Зворотний осмос (RO) – це повільний і недорогий метод очищення води, який видаляє 80%-90% води. Це потребує великого тиску. З води можна видалити корисні мінерали.	Ефективно	Неефективно	Не повністю ефективно	Неефективно

Фасування, зберігання та транспортування бутильованої води мають свої унікальні особливості, які важливо враховувати для забезпечення якості продукту та його безпечності для споживачів.

Матеріали пакування: бутильована вода часто фасується у пластикові пляшки, які виготовлені з поліетерэфталату (ПЕТ). Важливо використовувати матеріали, що не взаємодіють з водою та не впливають на її якість.

Герметичність: пакування повинне бути герметичним, щоб запобігти контамінації та зберегти свіжість води.

Маркування: на пакуванні обов'язково повинна бути інформація про виробника, термін придатності, склад води та інші важливі відомості.

Температурний режим: важливо зберігати бутильовану воду в прохолодному місці, подалі від прямих сонячних променів та джерел тепла.

Захист від забруднення: зберігати воду в чистому, сухому місці, подалі від забруднюючих речовин та сильних запахів.

Умови перевезення: під час транспортування важливо забезпечити захист від зовнішніх впливів, ударів, перепадів температур.

Гігієнічні умови: транспортні засоби повинні бути чистими та гігієнічними, щоб уникнути можливості забруднення продукту.

Стабільність вантажу: воду необхідно належним чином закріпити, щоб запобігти її розливу або пошкодженню під час перевезення.

1.3 Маркетингові стратегії виробників водопостачання

У сучасну епоху на бізнес сильно впливають постійні зміни в технологіях, і розширювати або розвивати бізнес стає дуже важко. Підприємство з доставки води також є однією з компаній, які стикаються з такою ж проблемою. Після

проведеного аналізу одним з найпоширеніших способів, за допомогою яких ви можете легко розширити свій бізнес водопостачання. Останні п'ять років були непростими для водної індустрії: IBIS World повідомляє про зростання на 1% на рік, у тому числі величезне зростання на 1,6% лише у 2019 році. Багато факторів сприяють ранньому ранковому сплеску посухи водопостачання серії. Найпростішим є підвищення обізнаності про здоров'я та збільшення попиту на ароматизовану, газовану та бутильовану воду.

Згідно з галузевими прогнозами, до 2024 року галузь водопостачання буде стабільно зростати на 1,2%. Крім того, очікується, що до 2024 року кількість компаній зменшиться. Це означає меншу конкуренцію з боку водопровідних компаній.

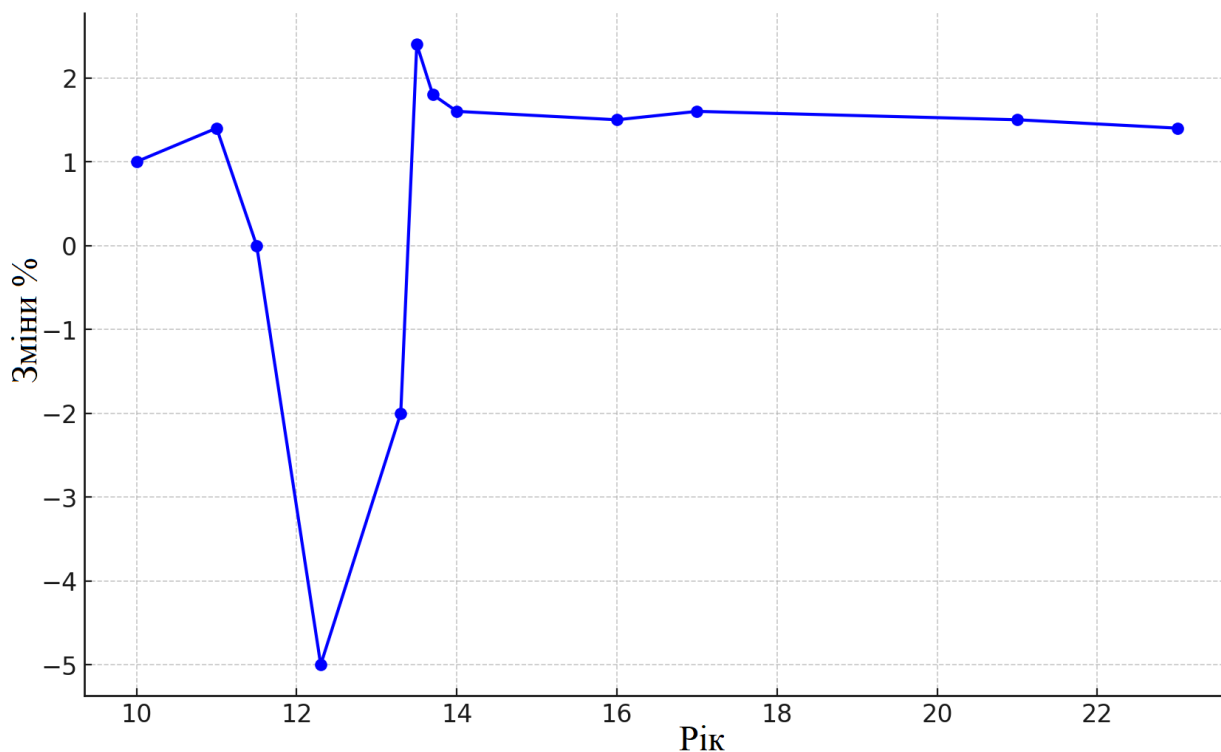


Рисунок 1.1 – Графік прогнозу доходу на основі змін у політиці

Проте галузь має свої проблеми (див. рис. 1.2), такі як підвищення вимог до фільтрації та відсутність високошвидкісних систем водопостачання. Ось чому важливо врахувати всі ці фактори перед початком власного бізнесу з водопостачання.

Після розуміння ринкового сценарію наступним кроком є визначення ринкової історії клієнта. Важливо розуміти крани-штабелери в кожному сегменті ринку. Подивімося один за одним.

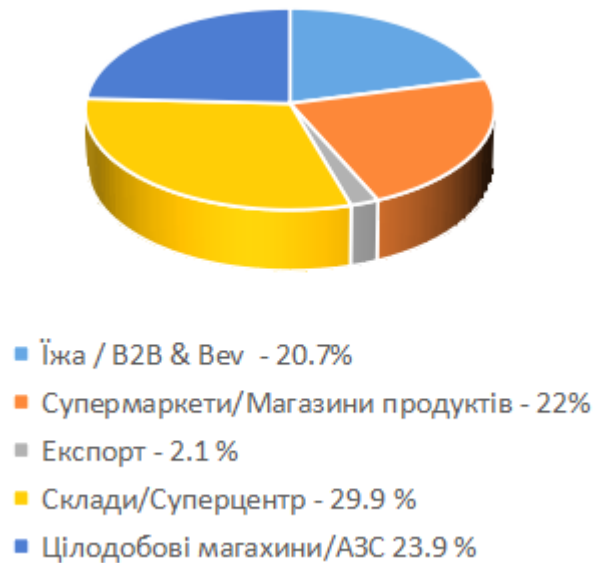


Рисунок 1.2 – Основні сегменти ринку

Корпоративна електронна комерція зростає і, як очікується, досягне 1 трильйона доларів. Це дає вам перевагу запуску системи інтернет-маркетингу, яка створює ілюзію зростання. Говорячи про зростання, очікується, що до 2020 року ринок електронної комерції B2B у США продовжить зростати на рівні приблизно 1 трильйона доларів США.

Крім того, дослідження показують, що приблизно 55% бізнес-споживачів купують їжу та напої онлайн на робочому місці. Маркетинг вашого бізнесу серед покупців B2B безмежний, оскільки для більшості інших продуктів потрібна вода.

Крім того, можна надавати послуги малому бізнесу. Ви можете допомогти малому бізнесу розвиватися, і ваші доходи зростуть одночасно.

По-друге, знайдіть оптовиків для оптового продажу ваших товарів. Можливості оптової торгівлі працюють дуже добре, оскільки більша географічна територія та більший обсяг продажів забезпечують високу рентабельність інвестицій. Прибуток оптовиків настільки високий, що багато постачальників почали заробляти на загальних стартових витратах, продаючи свою продукцію кільком оптовикам.

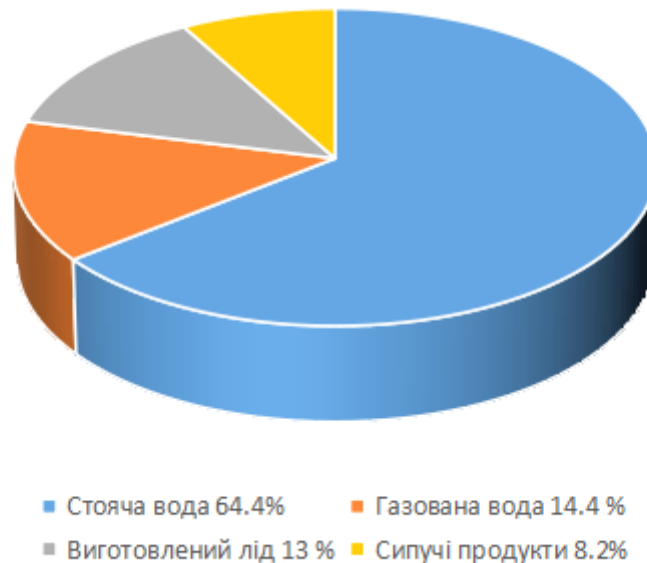


Рисунок 1.3 – Сегментація процесу продукту

Збільшення виробництва продукції є одним із способів розвитку вашого бізнесу водопостачання. У Сполучених Штатах люди продовжують купувати

різноманітні товари, пов'язані з водою. Це хороший показник того, що вам також слід розглянути можливість розширення асортименту продукції та продуктів, включаючи газовані напої, системи фільтрації води, 5-галонні охолоджувачі води, охолоджувачі без пляшок тощо.

Американці почали переходити від бутильованої води до лишайника та кулерів із системами фільтрації. Не тільки це, деякі інші нещодавні продукти, такі як збагачена вітамінами вода та інші водні продукти преміумкласу, також мають величезний потенціал. Пропонуючи більше продуктів, ви залучите більше типів клієнтів. Це також дозволить вам отримати вигоду від зростаючого попиту.

Через зростаючу загрозу глобального потепління в усіх галузях зростає прагнення до екологічно чистих компаній, і галузь водопостачання не є винятком. Сьогодні багато відомих брендів використовують більш міцні синтетичні матеріали для упаковки. Тут міцні матеріали означають використання пластмас рослинного походження, а не пластмас на основі нафти. Екологія може бути ефективним способом завоювати лояльність клієнтів і підтримку, що зрештою сприятиме розвитку вашого бізнесу.

Бренд є одним з найважливіших факторів, що веде до доходу будь-якої компанії. Він визначає, чи будуть клієнти купувати ваш продукт. Брендуння ще важливіше в продуктах харчування та напоях, де продукт продається сам.

Тому важливо розробити гарну стратегію бренду, щоб зв'язатися зі своїми клієнтами та отримати імпульс для свого бренду.

1.4 Контроль якості води на вході та виході

Якість питної води повинна відповідати вимогам чинних санітарних правил і норм, затверджених нормативними документами.

Виробничий контроль якості питної води організовується та/або здійснюється експлуатаційним підрозділом системи водопостачання, а за показниками якості проводяться необхідні дослідження.

Організація роботи з виробничого контролю повинна забезпечувати можливість отримання перевіреної та дієвої інформації про якість питної води в об'єктах розміром, визначеним ГОСТом 8.417, визначеним пожежною службою, що не перевищує нормативів, визначених ГОСТів 27384, з використанням обладнання, що входить до складу. у переліку типів затверджених державою засобів вимірювальної техніки та засобів вимірювальної техніки, що пройшли перевірку. Методи визначення показників якості води повинні бути затверджені або адаптовані відповідно до вимог ГОСТ Р 8.563, біологічні показники можуть визначатися за методиками, затвердженими МОЗ.

Лабораторії підлягають оцінці стану вимірювань згідно з сертифікації ДСТУ ISO / IEC 17025, ГОСТ Р 51500.4.

Контроль за наявністю патогенних мікроорганізмів у воді проводиться в лабораторіях і вимагає наявності ліцензії на поводження зі збудниками відповідної групи патогенності та ліцензії на виконання такої роботи. Контроль якості виробництва питної води включає:

Перед надходженням води джерела водопостачання та питної води у водопровідну та розподільну мережу в місці водозабору вимірюють її склад і властивості;

Попередній контроль за наявністю супровідної документації (технічних умов, сертифікатів відповідності або гігієнічних сертифікатів (гігієнічних звітів)) на реактиви, матеріали та іншу продукцію, що використовується в процесах водопідготовки;

Проводити попередній контроль селективності продуктів, що використовуються в процесах водоочищення, щоб відповідати належній документації щодо управління продуктом;

Терміново повідомляти центр санітарного нагляду про всі ситуації, коли результати контролю якості питної води не відповідають санітарним нормам насамперед коли мікробіологічні та токсикологічні показники перевищують нормативи;

Про результати виробничого контролю повідомити в САН.

При прийнятті адміністративного рішення, яке перевищує оцінку контрольного індексу гігієни питної води міста, враховуються результати експертизи міста. Контролюють значення показника незалежно від значення характеристики похибки. При цьому виявлені неточності повинні відповідати встановленим нормам.

З метою визначення якості питної води до контролю якості виробництва питної води на договірних засадах можуть залучатися лабораторії, акредитовані відповідно до встановлених процедур технічної компетентності; при проведенні арбітражного та сертифікованого виробництва - з урахуванням технічної компетентності та незалежності підприємства. .

Лабораторії повинні відповідати вимогам техніки безпеки, пожежної безпеки та промислової гігієни. Якість 2/3 побутової води не відповідає нормативним вимогам. Через вживання неякісної води захворюваність людей зростає в 4-5 разів. Норми водопостачання на 1 т готової продукції складають мЗ: чавун - 160-200; сталь - 150; відсутність - 10-15; нікель - 4000; маріель - 400-800; пластмаса - 500-1000; масло. - 20.

Теплові та атомні електростанції споживають велику кількість води. На 1 млн. кіловатів електроенергії теплові електростанції споживають 1,2-1,6 кубічних кілометрів води на рік, атомні – в 1,5-2 рази більше.

Охорона вод – це система заходів, спрямованих на запобігання та ліквідацію наслідків забруднення, засмічення та виснаження вод. Охорона води передбачає встановлення високих показників використання води та якості води в процесі водопостачання. Він передбачає розробку методів і заходів очищення стічних вод, контролю якості води та стічних вод.

У природі існує постійний колообіг води, який забезпечується випаровуванням, транспірацією рослин і опадами. Щільність водообміну характеризується такими даними, років: світовий океан - 2500 (оборотний - 63), підземні води - 400, озерні - 17, болотні - 5. У річках водообмін відбувається протягом доби, а в організмі людини - протягом годин

Під час колообіг води вона транспортує тепло, розчиняє і переносить природні елементи, руйнує і трансформує літосферу, бере участь у метеорологічних і гідрологічних процесах, є середовищем проживання водних організмів і рослин, забезпечує виробництво неважливих частин. кисень. Кількість і якість води відповідно важливі, якщо забезпечуються необхідні умови. Але розвиток промисловості, транспорту, сільського господарства та урбанізація призвели до того, що природні водойми перестали очищатися, тому для очищення води потрібні штучні споруди.

Вода характеризується складом і властивостями, які визначають її придатність для конкретних видів водокористування. Оцінка якості води базується на характеристиках, вибраних і стандартизованих на основі типу водокористування. Об'єкт із найнижчою концентрацією речовини у воді вибирається як змодельований об'єкт. Узагальнена чисельна оцінка якості води базується на індексі SOM, який являє собою набір основних показників, розподілених за видами водокористування. Якість, склад і властивості води у водоймах регламентуються санітарними вимогами та санітарними нормами. Для оцінки гігієни води використовуються такі показники:

- Кількість зважених речовин;
- Кількість зварювального матеріалу;
- температура;
- Вологий індикатор рН;
- мінерали;
- Розчинений кисень;
- Біологічне повне окислення киснем (БПОК повне);
- Хімічне споживання кисню (ХОК);
- наявність збудників;
- Кількість молочно-позитивних E. coli (LCC);
- Кількість гельмінтозних і протозойних кишкових зондів;
- Кількість хімічної речовини.

Показники, які використовуються для оцінки гігієни води: гранично допустима концентрація (ДК) речовин у воді; орієнтовно допустимі рівні (ОДР) речовин у воді; граничні симптоми отруєння слизом (гігієнічна токсикологія, загальна гігієна, органи чуття з розширеними вимогами: запах, смак, наявність коропа, утворення гною і глибина, присмак);

Хімічні речовини у воді поділяються на категорії безпеки:

- І клас - надзвичайно небезпечний;
- Рівень II - дуже небезпечний;

Відрахування за категоріями хімічної безпеки залежать від їх токсичності, кумулятивного характеру, здатності викликати віддалену дію та впливу граничних показників якості.

Окрім нагляду за якістю води, що здійснюється на державному рівні, суб'єкти господарювання, які використовують воду та відводять стічні води до водних об'єктів, також зобов'язані контролювати цей процес. Для цього, на території підприємств або в спеціалізованих лабораторіях встановлюються контрольні-

аналітичні пункти, обладнані відповідними аналітичними приладами. Вони застосовують різноманітні методики, включаючи фізичні, хімічні, біологічні, та методи, чутливі до органічних речовин, для моніторингу стану води та стічних вод.

Фізичні аналітичні техніки застосовують для оцінки таких параметрів води та стічних вод, як їхня прозорість, ступінь каламутності, наявність суспендованих часток та ефективність очищення.

Кількість завислих часток визначали за допомогою мембранно-паперового фільтра, через який пропускали 100-500 мл проби.

Визначають прозорість, каламутність за допомогою сенсорного порівняння обладнання або зразків.

У хімічному аналізі фокусують на виявленні рівня рН, вмісту металів, солей, органічних та синтетичних сполук, що можуть бути присутніми у воді, та впливати на її каламутність.

Бактеріальний аналіз, наведений у таблиці 1.4, проводився в лабораторії санепідемстанції за спеціальними методиками.

Заслуговує на увагу контроль контамінації за допомогою біотест-бактерій. Деякі бактерії починають світитися за нормального рівня забруднення. Чим більше токсичних речовин у воді, тим яскравіше світло випромінюють бактерії.

Таблиця 2.2 Склад і властивості води

Показники властивостей і складу води	Категорії водокористування	
	Для господарсько-питного водопостачання	Для відпочинку, купання, та водоспорту населення
Зависла речовина	Збільшення вмісту завислих речовин не повинно перевищувати:	

	0,25 мг/лм ³ ; 0,75 мг/лм ³	
Плаваючі домішки	На поверхні водойми не повинно бути плаваючих плівок, мінеральних масел.	
Аромат	Вода не повинна набувати не властивих їй запахів інтенсивністю більш як 1 бал.	
Колір	Не повинен виявлятися в стовпчику 20 см 10 см	
Температура	Підвищення температури води від скидання стічних вод у літній період не повинно перевищувати 3°C порівняно із середньомісячною.	
Водневий індекс (рН)	6.5	8.5
Мінеральний склад	Сухі залишки не повинні перевищувати 1000 мг/лм ³ , в тому числі хлоридів - 350 мг/лм ³ , сульфатів - 500 мг/лм ³ .	
Розчинений кисень	Зразки, зібрані о 12 годині дня, не повинні бути меншими за 4 мг/лм ⁴ у будь-який час року.	
БПОК завершено	Не має перевищувати 3,0 мг О ₂ /лм ³ при 20°C; 6,0 мг О ₂ /люмен ³	
ХСК	Не має перевищувати 15,0 мг О ₂ /лм ³ ; 30,0 мг О ₂ /люмен ³	
Збудник захворювань	Ні в якому разі у воді не повинно бути збудників хвороб.	
Лактозопозитивні кишкові палички (ЛКП)	Не більше 10 000 в 1 дм ³	Не більше 5000 захворювань в 1 дм ³
Каліформи у білковочних одиницях	Не більше 100 в 1 дм ³	Не більше 100 в 1 дм ³

Таблиця 2.3 ГДК рідких речовин у водоймах для господарсько-питного та культурно-побутового життя.

Таблиця 2.3 Гранично допустимі концентрації шкідливих речовин

Назва речовини	Рівні небезпеки	Допустима концентрація, мг/м
Аміак (за азотом)	III	2
Амоній сульфат	III	1
Активний хлор	III	Відсутньо
Ацетон	III	2.2
Бензол	II	0.5
Дихлоретан	IV	ОДР 0.02
Залізо	III	0.3
Кадмій	II	1
Карбонати	IV	1
Кобальт	III	0,1
Кремній	II	1
Марганець	III	0,1
Мідь	II	1
Натрій	III	200,0
Нафтопродукти	IV	0,1
Нікель	III	0.1
Нітрати (NO)	III	45

Нітрати (NO)	II	3,00
Ртуть	III	5
Свинець	II	0,03
Селен	II	0,01
Скiпiдар	IV	0,20
Цинк	III	1,00

1.5 Сертифікація якості питної води.

Сертифікація — це документальне підтвердження того, що продукція відповідає певним вимогам, концепціям, стандартам або технічним умовам. Потрібне підтвердження якості та стану води, що продається та використовується клієнтами. Є також сертифікати, які дозволяють використовувати та продавати воду для харчових цілей.

Сертифікація виробництва води - це комплекс заходів (паспорт), що здійснюються шляхом проведення національних перевірок (вимірів), які (документація) забезпечують відповідність продукції певним стандартам іншої НТД. Багато іноземних компаній заробили багато грошей і часу, щоб показати споживачам, що їхня продукція має високу якість. Тому, за даними іноземних джерел, ціна масштабних проєктів становить близько 1-2% від загальних витрат виробника.

У деяких ситуаціях витрати на сертифікацію можуть бути порівнянні з витратами на досягнення необхідного рівня якості продукції. Це обумовлено значенням сертифікації як ключового інструменту для розвитку міжнародних торговельно-економічних відносин, впровадження продукції підприємства на

внутрішній та зовнішній ринки та забезпечення його довготривалого розвитку. Широке застосування сертифікації підтверджує її значущість у цих процесах.

Сертифікація виникає для захисту внутрішнього ринку від використання незадовільної продукції. контрольне питання,

Захист здоров'я громадськості та збереження довкілля зобов'язують законодавця встановлювати відповідальність для суб'єктів господарювання (таких як виробники та продавці) за впровадження продукції на ринок, а також визначати мінімальні стандарти якості для товарів, що надходять у продаж.

До цього контексту входять законодавчі акти, як-от Закон України про захист прав споживачів, Європейський закон про відповідальність за продукцію, а також закони, що встановлюють мінімальні вимоги до якості товарів, як це може бути з Законом про іграшки, Законом про електромагнітну сумісність, і, зокрема, Законом про сертифікацію якості води.

Закон про чисту воду визначає сертифікацію якості води як важливий інструмент у забезпеченні якості водних ресурсів, дозволяючи вирішувати проблеми водопостачання відповідно до федеральних регуляцій, особливо у гірських районах. Це підкреслює значення і роль сертифікації у досягненні стандартів якості води.

Федеральні установи не мають права видачі дозволів або ліцензій, проведення сертифікації або відмови в ній для діяльності, яка може спричинити скидання стічних вод із території Сполучених Штатів у штат, де таке скидання планується. Важливим аспектом є право штату на надання, адаптацію, відкликання чи скасування сертифікації. Будь-яка відмова в сертифікації блокує можливість видачі федерального дозволу або ліцензії, незалежно від наявності повноважень на їх видачу. Відмова у видачі дозволу розуміється як видача дозволу або ліцензії без застережень від сторони штату або племені. Рішення штату про відмову, сертифікацію або умови дозволу чи ліцензії базуються на тому, наскільки

запропонований проект відповідає стандартам якості води, затвердженим штатом та племенем, включаючи обмеження на стічні води, нові стандарти продуктивності для інжекторів, обмеження токсичних забруднювачів та інші відповідні державні чи місцеві вимоги.

Візьмемо на прикладі українську компанію «CLEARWATER», щоб зрозуміти, як проходити сертифікацію.

Відома негазована мінеральна вітамінно-питна вода торгової марки «CLEARWATER» «CLEAR WATER» випускається ТОВ «Власні колодязі II». Виробник «Чиста вода» є учасником міжнародної програми «Вода для людства» і для виробництва продукції використовує воду з власних артезіанських свердловин. Виробництво здійснюється відповідно до OEM 08.002.03.025:2010. Bureau Veritas сертифікована відповідно до сертифіката якості ISO 9001:2008 та екологічно сертифікована відповідно до ISO 14024. Екологічність, якість та безпека продукції підтверджена сертифікатом відповідності № UA.08.002.432 та правом на використання екологічного маркування відповідно до вимог Всесвітнього фонду дикої природи (WWF).

Для оцінки відповідності продукції затверджені вимоги та стандарти, міжнародні стандарти якості продукції. Виробники використовують чинні стандарти екологічної стійкості продукції та безпеки мінеральної води у виробництві та покращують екологічні характеристики виробництва відповідно до планів безпеки.

На даному підприємстві функціонує інтегрована система управління якістю, що відповідає міжнародним стандартам, зокрема ДСТУ ISO 9001:2009 (який ідентичний ISO 9001:2008) та ДСТУ ISO 22000:2007 (еквівалентний ISO 22000:2005).

Корпоративне керівництво розробляє та реалізує екологічну політику для забезпечення: водозбереження, охорони води від забруднення, зменшення негативного впливу, охорони та раціонального використання водних ресурсів.

Підтримуйте високі екологічні стандарти та підтримуйте посилений відповідальний контроль для забезпечення якості. Підприємства регулярно проводять аудит якості. Має власну вимірювальну лабораторію для постійного контролю безпеки та якості на всіх етапах очищення, мінералізації та розливу води.

Виробництво продукції, включно з добуванням води, здійснюється з дотриманням законодавства про охорону навколишнього середовища. Екстракція води проводилася з артезіанських колодязів (з родовищ «чистої води»), розташованих на глибинах 126 та 90 метрів у селі Мила, що належить до Київської області. Процедури очистки та розливу води повністю автоматизовані. Воду очищають за допомогою технології, яка є безпечною для здоров'я людини. Для надання воді мінерального складу використовують мінерали, імпортовані з країн Європейського Союзу.

Вода, вироблена компанією «Чиста Вода» із свердловини П, не містить харчових добавок, які заборонені законодавством, пестицидів чи нафтопродуктів. Рівень вмісту важких металів у цій воді є нижчим за гранично допустимі показники, встановлені нормативними актами, та відповідає екологічним стандартам. Якість води регулярно перевіряється за допомогою власної незалежної лабораторії компанії.

Пляшки з полікарбонату використовуються для розливу та доставки води, що сприяє розширенню її використання. Непридатні пляшки будуть здані на утилізацію. Полікарбонат є матеріалом, який можна переробити.

Бізнес також бере участь у різноманітних ініціативах, спрямованих на відповідальне ставлення до природних ресурсів. Такі ініціативи спрямовані як на

наших власних співробітників, так і на клієнтів і природне виробництво електроенергії.

Для розливу та доставки води компанія "Чиста Вода" використовує пляшки з полікарбонату, що сприяє збільшенню обсягу її споживання. Пляшки, що вийшли з ладу, направляються на утилізацію, адже полікарбонат є матеріалом, придатним для переробки.

Компанія також активно залучена в ініціативи з відповідального використання природних ресурсів, що орієнтовані як на її співробітників, так і на клієнтів, а також включають аспекти сталого виробництва енергії.

Серед інших заходів, "Чиста Вода" пропонує своїм клієнтам спеціальний прилад - таймер PowerDown, який дозволяє автоматично вимикати та вмикати побутову техніку, таку як кулери для води, водонагрівачі, лампи гібридного дизайну, за заданим графіком.

1.6 Розрахунок витрат на очищення питної води

При використанні системи зворотного осмосу для отримання 1 літра питної води витрачається 8 літрів води (7 літрів відходять у каналізацію та 1 літр стає чистою водою). Таким чином, для виробництва 5 літрів фільтрованої води використовується 40 літрів води щодня. Це становить 14 600 літрів, або 14,6 м³ на рік, що при тарифі 12 копійок за 1 м³ виливається у вартість 175 гривень на рік.

У розрахунках враховуються всі потенційні витрати при використанні системи зворотного осмосу, включаючи важливі зауваження, що можуть надійти в ході обговорень і конференцій, для максимально точного визначення вартості 1 літра чистої питної води.

Таблиця 2.4 Обчислення ціни за один літр води.

Назва	Кількість, шт	Ціна, шт/грн	Разом, грн
Фільтр зворотного осмосу	1	1500	1500
Картридж спінений поліпропілен 5 мікронів	24	10	240
Картридж спінений поліпропілен 1 мікрон	24	10	240
Картридж гранульованого вугілля	24	40	960
Мембранна зворотного осмосу	4	350	1400
Вугільний пост фільтр	12	50	600
Оплата майстру за зміну картриджів	24	50	1200
Плата за міську воду	175	8	1400
Непередбачений ремонт одного з вузлів	1	350	350
Разом			7890
Вартість одного літра $7890 / (12 \text{ років} * 365 \text{ днів}) = / 5\text{л} = 36 \text{ коп/л}$			

Висновок: Використання фільтру зворотного осмосу для води економічно вигідно. Вартість води знижується, якщо заміну фільтруючого елементу проводити самостійно. Якщо споживання перевищує 5 літрів на день, вартість 1 літра води зменшується. Крім того, довговічність системи зворотного осмосу, виготовленої з поліпропілену і маючої мінімум механічних частин, може перевищувати 12 років, що додатково знижує вартість 1 літра очищеної води.

1.7 Постановка задачі

Дослідити та розробити інформаційну систему, яка підтримує процеси управління замовленнями та логістики доставки бутильованої води, з використанням кросплатформного фреймворку Microsoft MAUI, з метою підвищення ефективності, точності та зручності цих процесів.

Основні завдання дослідження:

- Аналіз існуючих систем управління замовленнями та доставки бутильованої води: Вивчити поточні методи та інструменти, використовувані у галузі, для ідентифікації ключових проблем та потреб оптимізації.
- Оцінка можливостей Microsoft MAUI: Дослідити функціональність та переваги використання Microsoft MAUI як інструменту для розробки кросплатформних додатків, з акцентом на його придатність для створення системи управління замовленнями та доставки.
- Розробка прототипу системи: Створити прототип інформаційної системи, що включає основні функціональні можливості для управління замовленнями, логістики доставки та взаємодії з клієнтами.
- Аналіз отриманих даних та внесення удосконалень: Проаналізувати зібрані під час тестування дані для виявлення потенційних проблем та можливостей для удосконалення системи.
- Розробка рекомендацій для майбутнього впровадження: На основі результатів дослідження та розробки системи сформулювати рекомендації щодо її впровадження та подальшої експлуатації.

Ця наукова задача спрямована на забезпечення підприємств, що займаються доставкою бутильованої води, надійним, ефективним і зручним у використанні

інструментом, який відповідає сучасним технологічним тенденціям та вимогам ринку.

1.8 Висновок до розділу № 1

В цьому розділі було проведено всебічне дослідження предметної області, що охоплює виробництво та постачання бутильованої води, включаючи аналіз організаційних структур підприємств, технології очищення води, маркетингові стратегії, наукову актуальність проблематики, сертифікацію якості та розрахунок витрат на очищення води. Основні висновки з кожного з пунктів дозволили сформулювати комплексне розуміння галузі, її викликів та можливостей.

- Аналіз діяльності підприємств у сфері виробництва води підкреслив значення їх організаційних структур, виявляючи тісний взаємозв'язок між структурною організацією підприємств та їх продуктивністю та ефективністю.
- Розгляд технологій та систем очищення води виявив прогресивні підходи та інноваційні технології, які можуть бути ключовими для підвищення якості кінцевого продукту.
- Маркетингові стратегії виробників фасованої води відіграють критичну роль у забезпеченні конкурентоспроможності на ринку, де акцент робиться на розробку гнучких та інноваційних підходів до маркетингу.
- Аналіз актуальності наукових питань підкреслює необхідність подальшого дослідження у галузі, особливо з погляду сталого розвитку та впровадження екологічно чистих технологій.

- Процес сертифікації якості питної води демонструє суттєву роль стандартів безпеки та якості у цій галузі, а також важливість їх дотримання для забезпечення довіри споживачів.
- Розрахунок витрат на очищення води відкриває перспективи для оптимізації виробничих процесів та виявлення можливостей для зниження витрат, що є ключовим для підвищення рентабельності бізнесу.
- Постановка задачі дослідження дає чітке розуміння напрямків подальшої роботи та визначає основні дослідницькі пріоритети, спрямовані на вдосконалення процесів у сфері виробництва та постачання бутильованої води.

Таким чином, цей розділ закладає фундамент для всебічного аналізу та розуміння предметної області, створюючи платформу для подальших досліджень та розробки ефективних рішень у цій галузі.

РОЗДІЛ 2. ДОСЛІДЖЕННЯ МЕТОДІВ ПРОГРАМНИХ ПЛАТФОРМ

2.1 Огляд Microsoft .NET MAUI

.NET Multi-Platform App UI (.NET MAUI) – це крос-платформна платформа для створення власних мобільних та класичних додатків за допомогою C# та XAML.

За допомогою .NET MAUI можна розробляти програми, які можуть працювати на Android, iOS, macOS та Windows з однієї загальної бази коду.

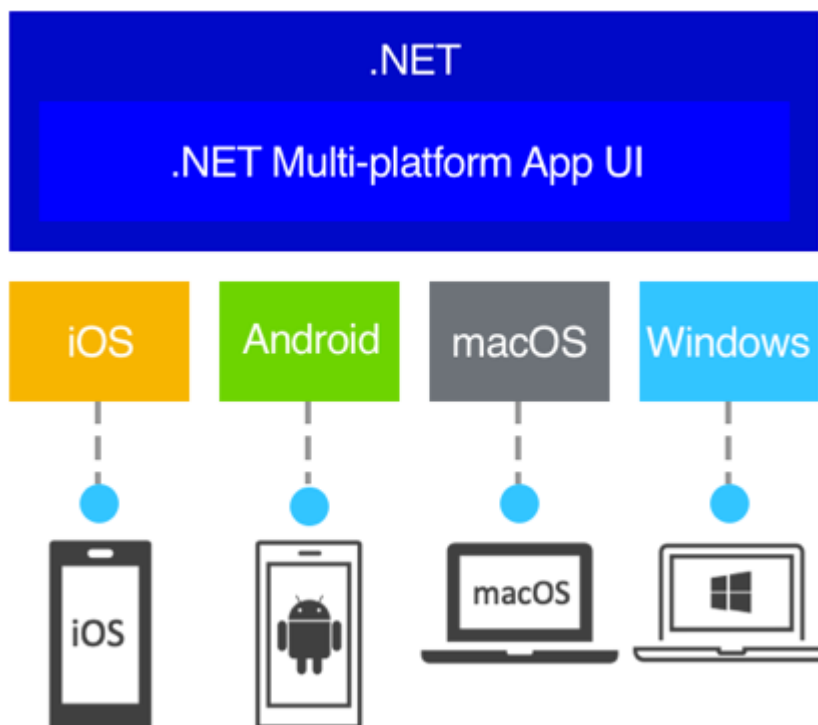


Рисунок 2.1. Схема .NET MAUI

.NET MAUI є відкритим вихідним кодом і є еволюцією Xamarin.Forms, розширеною від мобільних до класичних сценаріїв, з елементами керування інтерфейсом користувача, перебудованими з нуля для підвищення продуктивності

й розширюваності. Якщо ви раніше використовували Xamarin.Forms для створення кросплатформових інтерфейсів, ви помітите безліч подібностей з .NET MAUI. Проте з-поміж них також є деякі відмінності. За допомогою .NET MAUI можна створювати програми для кількох платформ в одному проєкті та за необхідності додавати вихідний код та ресурси для конкретної платформи. Однією з основних цілей .NET MAUI є можливість реалізувати в загальній базі коду якомога більше логіки та макета інтерфейсу для додатка.

.NET MAUI призначений для розробників, які хочуть:

- Написання кросплатформових додатків у XAML і C# з однієї загальної бази коду Visual Studio.
- Спільне використання макета інтерфейсу користувача та розробка на різних платформах.
- Спільне використання коду, тестів та бізнес-логіки на різних платформах.

Як працює .NET MAUI

.NET MAUI об'єднує API-інтерфейси Android, iOS, macOS та Windows у єдиний API, який надає розробникам можливість виконувати одноразовий запис у будь-якому місці, забезпечуючи додатковий доступ до кожного аспекту кожної власної платформи.

.NET 6 або пізнішої версії надає ряд платформ для створення програм: .NET Android, .NET iOS, .NET macOS і Windows UI 3 (WinUI 3). Всі ці платформи мають доступ до однієї бібліотеки базових класів .NET (BCL). Ця бібліотека абстрагує інформацію про базову платформу від коду. Бібліотека BCL має залежність від середовища виконання .NET, у якому виконується код програми. Для Android, iOS та macOS середовище реалізується Mono, реалізація середовища виконання .NET. Windows .NET CoreCLR надає середовище виконання.

Хоча BCL дозволяє додаткам, що працюють на різних платформах, спільно використовувати спільну бізнес-логіку, різні платформи мають різні способи визначення інтерфейсу користувача для програми, а також надають різні моделі для вказівки того, як елементи інтерфейсу користувача взаємодіють і взаємодіють. Ви можете створити інтерфейс для кожної платформи окремо за допомогою відповідної платформи (.NET Android, .NET iOS, .NET macOS або WinUI 3), але цей підхід вимагає підтримки бази коду для кожного окремого сімейства пристроїв.

.NET MAUI надає єдину платформу для створення інтерфейсів для мобільних і класичних додатків. На наступній схемі показано високорівневе представлення архітектури програми .NET MAUI:

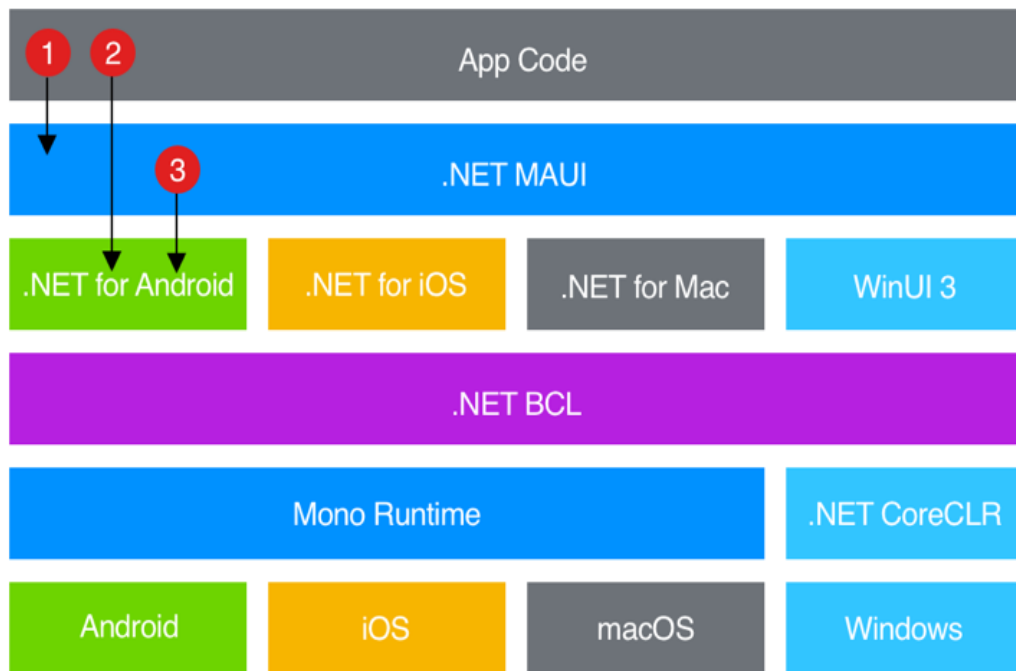


Рисунок.2.2. Високорівневе представлення архітектури програми .NET MAUI.

У програмі .NET MAUI ви пишете код, який в основному взаємодіє з API MAUI .NET . Потім .NET MAUI використовує власні платформи API. Крім того, код програми може безпосередньо виконувати API платформи, якщо це необхідно.

Програми .NET MAUI можна записувати на ПК або Mac і компілювати у власні пакети програм:

- Програми Android, створені за допомогою .NET MAUI, компілюються з C# на проміжну мову (IL), яка потім JIT-код компілюється у власну збірку при запуску програми.
- Програми iOS, створені за допомогою .NET MAUI, повністю заздалегідь компілюються із C# у власний код збірки ARM.
- Програми macOS, створені за допомогою .NET MAUI, використовують Mac Catalyst, рішення від Apple, яке приносить вашу програму iOS, створену за допомогою UIKit на робочий стіл, і розширює його за допомогою додаткових API AppKit і платформи за потреби.
- Програми Windows, створені за допомогою .NET MAUI, використовують бібліотеку інтерфейсу користувача Windows 3 (WinUI 3) для створення власних програм, призначених для робочого столу Windows. Докладніше про WinUI 3 см. у бібліотеці інтерфейсу користувача Windows.

2.2 Переваги Microsoft MAUI.

Погляд з висоти пташиного польоту Microsoft .NET MAUI (інтерфейс мультиплатформенної програми) пропонує широкий спектр можливостей для розробки кросплатформних програм. Ось деякі з ключових функцій:

- Кросплатформна розробка: розробляйте програми для iOS, Android, Windows і macOS одночасно на основі одного коду. Це економить багато часу та ресурсів, оскільки вам не потрібно писати окремий код для кожної платформи.
- Уніфікований інтерфейс користувача: .NET MAUI дозволяє створювати уніфікований і узгоджений інтерфейс користувача на всіх платформах за допомогою загальних елементів керування та макетів.
- Розширена підтримка MVVM: підтримка шаблону Model-View-ViewModel (MVVM) спрощує керування даними та взаємодію з користувачем, що робить його ідеальним для кросплатформних програм.
- Спеціальні API для платформи: попри спільну кодову базу, розробники можуть легко отримати доступ до специфічних для платформи API, щоб скористатися перевагами унікальних можливостей кожної платформи.
- Гнучкість дизайну: .NET MAUI надає інструменти для створення гнучких і адаптивних інтерфейсів, які добре виглядатимуть на різних пристроях і екранах.
- Інтеграція з сучасними технологіями: тісно інтегрована з іншими технологіями .NET, такими як Entity Framework для роботи з даними та Blazor для створення вебкомпонентів.
- Покращена продуктивність: .NET MAUI покращує продуктивність завдяки оптимізованому рендерингу й анімації, покращуючи загальну якість вашої програми.
- Інструменти розробки та тестування: широкий спектр інструментів для розробки, тестування та налагодження програм, які допомагають створювати надійні та ефективні програми.

- Відкритий вихідний код: як частина відкритої екосистеми .NET MAUI дозволяє спільноті співпрацювати та вносити власні вдосконалення та розширення.
- Підтримка спільноти: потужна підтримка спільноти розробників і постійні оновлення гарантують, що технологія є актуальною та актуальною.

Ці функції роблять Microsoft .NET MAUI потужним інструментом для створення кросплатформних мобільних і настільних програм, що забезпечує гнучкість, продуктивність і високий рівень взаємодії з користувачем.

Кросплатформні API для функцій пристроїв

.NET MAUI надає кросплатформні API для власних функцій пристроїв. Приклади функціональних можливостей, що надаються .NET MAUI для доступу до функцій пристрою, включають:

- Доступ до датчиків, таких як акселерометр, компас та гіро область на пристроях.
- Можливість перевірити стан підключення пристрою та виявляти зміни.
- Вкажіть інформацію про пристрій, на якому запущено програму.
- Скопіюйте та вставте текст у системний буфер обміну між програмами.
- Виберіть один або кілька файлів із пристрою.
- Безпечно зберігайте дані у вигляді пар "ключ-значення".
- Використовуйте вбудовані підсистеми перетворення тексту на мову для читання тексту з пристрою.
- Ініціюйте потоки автентифікації на основі браузера, які прослуховують зворотний виклик на певну зареєстровану URL-адресу програми.

Один проєкт

Єдиний проєкт .NET MAUI приймає можливості розробки для конкретної платформи, які зазвичай виникають при розробці додатків і абстрагує їх в один

спільний проєкт, який може бути орієнтований на Android, iOS, macOS і Windows. Єдиний проєкт .NET MAUI забезпечує спрощену та узгоджену кроссплатформенну розробку незалежно від цільових платформ. Єдиний проєкт .NET MAUI надає такі можливості:

- Один спільний проєкт, призначений для Android, iOS, MacOS та Windows.
- Спрощений вибір цільового об'єкта налагодження для запуску програм .NET MAUI.
- Загальні файли ресурсів у одному проєкті.
- Один маніфест програми, що вказує назву, ідентифікатор та версію програми.
- Доступ до API-інтерфейсів та засобів для конкретної платформи за потреби.
- Одна кроссплатформова точка входу програми.
- Єдиний проєкт .NET MAUI включений за допомогою багатоцільової та використання проєктів у стилі SDK. Додаткові відомості про проєкт .NET MAUI див. у статті .NET MAUI з одним проєктом.

Гаряче перезавантаження

.NET MAUI включає підтримку гарячого перезавантаження .NET, яке дозволяє змінювати керований вихідний код під час роботи програми без необхідності вручну призупинити або потрапити до точки зупинки. Потім зміни коду можна застосувати до працюючої програми без перекомпіляції.

.NET MAUI також включає підтримку гарячого перезавантаження XAML, яка дозволяє зберігати файли XAML і переглядати зміни, відображені у працюючій програмі без повторної компіляції. Крім того, стан навігації та дані будуть підтримуватися, що дозволяє швидко виконувати ітерацію в інтерфейсі користувача без втрати місця в додатку.

2.3 Model-View-ViewModel (MVVM)

MVVM (Model-View-ViewModel) є архітектурним шаблоном, який використовується у розробці програмного забезпечення, зокрема, для створення додатків з графічним інтерфейсом користувача. Цей шаблон включає три основні компоненти: модель, відображення (вид) та модель представлення (ViewModel). Кожен з цих компонентів виконує свої специфічні функції. На діаграмі нижче показано взаємозв'язки між цими трьома компонентами.

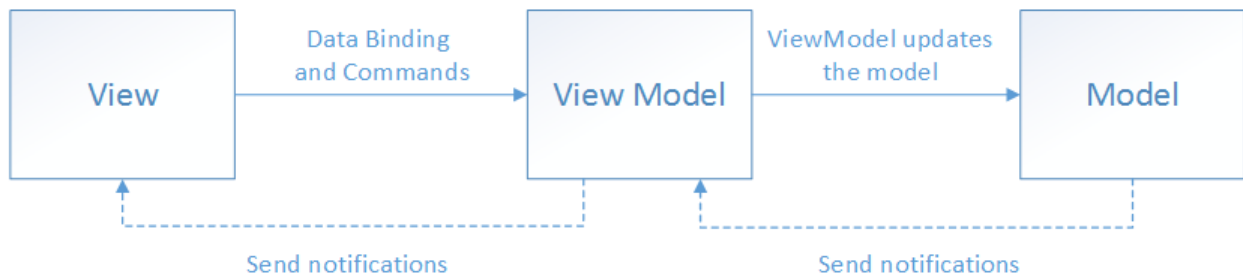


Рисунок 2.3 Діаграма взаємодії між компонентами MVVM

Окрім розуміння відповідальності кожного компонента, також важливо зрозуміти, як вони взаємодіють. На високому рівні вид "знає" про модель представлення, а модель представлення "знає" про модель, але модель не знає про модель представлення, і модель представлення не знає про вид. Таким чином, модель представлення ізолює вид від моделі та дозволяє моделі розвиватися незалежно від відображення.

Переваги використання шаблону MVVM:

- Якщо існуюча реалізація моделі містить бізнес-логіку, змінити її може бути складно або ризиковано. У цьому випадку модель представлення

виступає як адаптер для класів моделі та запобігає необхідності вносити великі зміни у код моделі.

- Розробники можуть створювати модульні тести для моделі представлення та моделі без використання виду. Тести моделі представлення можуть перевіряти саме ту ж функціональність, яку використовує вид.
- Інтерфейс користувача програми може бути перепроєктований без змін у коді моделі представлення та моделі, за умови, що вид реалізований повністю на XAML або C#. Таким чином, нова версія виду повинна працювати з існуючою моделлю представлення.
- Дизайнери та розробники можуть незалежно та одночасно працювати над своїми компонентами під час розробки. Дизайнери можуть зосередитися на виді, в той час, як розробники можуть працювати над компонентами моделі представлення та моделі.

Ключ до ефективного використання MVVM полягає у розумінні того, як правильно розподілити код програми між відповідними класами та як ці класи взаємодіють. Наступні розділи обговорюють обов'язки кожного з класу.

View

Перегляд (View) відповідає за визначення структури, розкладки та зовнішнього вигляду того, що користувач бачить на екрані. Ідеально, кожен перегляд визначається в XAML, з обмеженою кількістю коду, який не містить бізнес-логіки. Однак у деяких випадках код може містити логіку інтерфейсу користувача, яка реалізує візуальну поведінку, яку важко виразити в XAML, наприклад, анімації.

У додатку .NET MAUI перегляд зазвичай є класом, похідним від `ContentPage` або `ContentView`. Однак перегляди також можуть бути представлені шаблоном

даних, який визначає елементи інтерфейсу користувача, що використовуються для візуального представлення об'єкта під час його відображення. Шаблон даних як перегляд не має коду і призначений для прив'язки до конкретного типу моделі представлення.

Забезпечте, щоб моделі представлення відповідали за визначення логічних змін стану, які впливають на деякі аспекти відображення перегляду, наприклад, чи доступна команда або, чи є ознака виконання операції. Таким чином, включайте та вимикайте елементи інтерфейсу користувача, прив'язуючись до властивостей моделі представлення, а не включаючи та вимикаючи їх у коді.

Існує кілька варіантів виконання коду на моделі представлення у відповідь на взаємодії на перегляді, такі як натискання кнопки або вибір елемента. Якщо елемент керування підтримує команди, його властивість `Command` може бути прив'язана до властивості `ICommand` на моделі представлення. Коли команда елемента керування викликається, код у моделі представлення буде виконаний. Крім команд, до об'єкта у виді можуть бути прикріплені поведінки, які можуть слухати або виклик команди, або підняття події. У відповідь поведінка може викликати `ICommand` на моделі представлення або метод на моделі представлення.

ViewModel

Модель представлення (`ViewModel`) реалізує властивості та команди, до яких може бути прив'язаний перегляд (`View`), і повідомляє перегляд про будь-які зміни стану через події повідомлень про зміни. Властивості та команди, які надає модель представлення, визначають функціональність, яку має пропонувати інтерфейс користувача, але сам перегляд визначає, як ця функціональність буде відображена.

У багатоплатформних додатках слід тримати основний потік інтерфейсу користувача незаблокованим, щоб покращити сприйняття продуктивності користувачем. Тому в моделі представлення використовуйте асинхронні методи

для операцій вводу/виводу та піднімайте події, щоб асинхронно повідомляти перегляди про зміни властивостей.

Модель представлення також відповідає за координацію взаємодії перегляду з будь-якими необхідними класами моделі. Зазвичай існує відношення один-до-багатьох між моделлю представлення та класами моделі. Модель представлення може обрати пряме використання класів моделі у перегляді, таким чином елементи керування у виді можуть бути прив'язані безпосередньо до них. У цьому випадку класи моделі повинні бути розроблені так, щоб підтримувати прив'язку даних та події повідомлень про зміни.

Кожна модель представлення надає дані з моделі у формі, яку легко може спожити перегляд. Для цього модель представлення іноді виконує конвертацію даних. Розміщення цієї конвертації даних у моделі представлення є гарною ідеєю, оскільки це забезпечує властивості, до яких може бути прив'язаний перегляд. Наприклад, модель представлення може комбінувати значення двох властивостей, щоб їх було легше відображати у виді.

Також можливо використовувати конвертери як окремий шар конвертації даних, який знаходиться між моделлю представлення та переглядом. Це може бути необхідним, наприклад, коли дані потребують особливого форматування, яке модель представлення не надає.

Для того, щоб модель представлення могла брати участь у двосторонній прив'язці даних з переглядом, її властивості повинні підіймати подію `PropertyChanged`. Моделі представлення задовольняють цю вимогу, реалізуючи інтерфейс `INotifyPropertyChanged` і піднімаючи подію `PropertyChanged` при зміні властивості.

Для колекцій надається дружній до перегляду `ObservableCollection<T>`. Ця колекція реалізує повідомлення про зміни в колекції, звільняючи розробника від необхідності реалізовувати інтерфейс `INotifyCollectionChanged` на колекціях.

Model

Класи моделі є не візуальними класами, які інкапсулюють дані додатку. Таким чином, модель можна вважати представленням доменної моделі додатку, яка зазвичай містить модель даних, а також бізнес-логіку та логіку валідації. Прикладами об'єктів моделі є об'єкти передачі даних (DTOs), звичайні старі CLR-об'єкти (POCOs) та генеровані об'єкти сутностей та проксі-об'єкти.

Класи моделі зазвичай використовуються разом із сервісами або репозиторіями, які інкапсулюють доступ до даних та кешування. Вони відіграють ключову роль у роботі з даними, забезпечуючи міст між базою даних або іншим зовнішнім джерелом даних і бізнес-логікою додатку. Це містить отримання, зберігання та обробку даних, а також забезпечення відповідних механізмів валідації та перевірки.

Така архітектура дозволяє розробникам зосередитися на бізнес-логіці, не турбуючись про деталі низькорівневого доступу до даних. Це спрощує розробку, покращує можливості тестування та підтримки коду, і допомагає уникнути дублювання коду та знизити залежність від конкретних технологій зберігання даних.

Шаблон Model-View-ViewModel (MVVM) допомагає чітко відокремити бізнес-логіку та логіку представлення додатку від його користувацького інтерфейсу (UI). Підтримання чіткого розділення між логікою додатку та інтерфейсом користувача допомагає вирішити багато проблем розробки та робить додаток легшим для тестування, підтримки та розвитку. Це також може значно покращити можливості повторного використання коду та дозволяє розробникам та дизайнерам інтерфейсу користувача більш легко співпрацювати під час розробки своїх відповідних частин додатку.

Використовуючи шаблон MVVM, інтерфейс користувача додатка та базова логіка представлення та бізнес-логіка розділені на три окремі класи: перегляд (view), який інкапсулює інтерфейс користувача та логіку інтерфейсу; модель представлення (view model), яка інкапсулює логіку представлення та стан; та модель (model), яка інкапсулює бізнес-логіку додатка та дані.

2.4 Висновок до розділу № 2

У другому розділі роботи здійснено глибоке дослідження методів програмних платформ, зокрема фокусуючись на Microsoft .NET MAUI та архітектурному шаблоні Model-View-ViewModel (MVVM). Основні висновки цього розділу можна сформулювати наступним чином:

- Огляд Microsoft .NET MAUI: Аналіз платформи Microsoft .NET MAUI показав, що ця технологія пропонує широкі можливості для створення крос-платформних додатків. Її гнучкість та ефективність роблять її ідеальною для розробки сучасних застосунків, які вимагають високого рівня адаптивності та інтеграції.
- Можливості Microsoft MAUI: Подальше дослідження підкреслило різноманітність можливостей, які надає Microsoft MAUI, включаючи поліпшення продуктивності, спрощення розробки, та забезпечення кращої користувацької інтеракції. Ці можливості відкривають нові горизонти для розробки більш ефективних та інтуїтивно зрозумілих додатків.
- Model-View-ViewModel (MVVM): Аналіз архітектурного шаблону MVVM виявив його значення у структуруванні та організації коду застосунків, забезпечуючи високий рівень масштабованості та підтримки. Цей шаблон дозволяє відокремлювати бізнес-логіку від

інтерфейсу користувача, сприяючи більш ефективній розробці та тестуванню додатків.

Загалом, цей розділ демонструє значний потенціал технології Microsoft MAUI та шаблону MVVM для розробки програмних продуктів. Розуміння та використання цих інструментів та методологій може значно поліпшити якість та ефективність розробки програмного забезпечення, а також забезпечити високу адаптивність до різних умов використання.

РОЗДІЛ 3. СТВОРЕННЯ ТА АПРОБАЦІЯ СИСТЕМИ ПІДТРИМКИ ФОРМУВАННЯ ЗАМОВЛЕННЯ ТА ОРГАНІЗАЦІЇ ПОСТАЧАННЯ ФАСОВАНОЇ ВОДИ

3.1 Методи виконання задачі

Необхідно побудувати модель даних за допомогою інструментів SQLite Maestro. Ми створюємо порожню базу даних. На основі створеної моделі (див. додаток модель даних (SQLite Maestro)) ми згенерували базу даних за допомогою SQLite Maestro. Натиснувши кнопку New Database буде згенерована на основі згенерованої моделі. З'явиться діалогове вікно з базою даних і виконанням коду SQL, натиснувши кнопку «Виконати». У серверній частині SQL ми отримуємо готову базу даних.

Розробка інтерфейсу для взаємодії з базою даних:

- Створіть новий проект у Microsoft Visual Studio, вибравши "Створити проект".
- У діалоговому вікні "Створити проект" оберіть опцію ".NET MAUI App" і натисніть "ОК".
- Для підключення до бази даних, перейдіть у меню "Файл" та виберіть "Додати нове джерело даних".
- У майстрі налаштування джерела даних виберіть "База даних", натисніть "Далі" та виберіть "Нове підключення".
- Введіть назву сервера, використовуючи кнопку "Огляд" для вибору потрібної бази даних.
- На сторінці "Вибір об'єктів бази даних" розгорніть вузол "Таблиці".

- Встановіть прапорець біля потрібної таблиці та завершіть процес, натиснувши "Готово".

Тепер у середовищі Microsoft Visual Studio ми можемо переглядати структуру бази даних за допомогою внутрішнього інструменту Dataset Designer.(див. додаток).

3.2 Створення бази даних

Представлення структури бази даних.

Orders - сутність, що містить інформацію про замовлення, з полями для унікального ідентифікатора замовлення (Order ID), ідентифікатора клієнта (Customer ID), ідентифікатора типу замовлення (Order Type ID), дати замовлення (Order Date) та інших.

Customers - сутність, яка зберігає дані про клієнтів, включаючи ідентифікатор клієнта (Customer ID), ім'я (Name), електронну пошту (Email) та контактний телефон (Contact Phone).

OrderType - сутність, що описує тип замовлення, з полями для ідентифікатора типу замовлення (Order Type ID), назви типу (Type Name) та опису (Description).

CustomerAddresses - сутність, призначена для зберігання адрес клієнтів, містить ідентифікатор адреси (Address ID), ідентифікатор клієнта (Customer ID), а також деталі адреси, як-от вулиця, місто, штат, країна та поштовий індекс.

DeliveryInfo - сутність, що включає інформацію про доставку, з полями для ідентифікатора доставки (Delivery ID), ідентифікатора замовлення (Order ID), адреси доставки (Delivery Address ID), ідентифікатора клієнта (Customer ID) та електронної пошти співробітника (Employee Email).

Employees - сутність для зберігання інформації про співробітників, має поля для ідентифікатора співробітника (Employee ID), ім'я (Name) та електронної пошти (Email).

Між цими сутностями існують зв'язки, які показують, як дані в одній таблиці пов'язані з даними в інших таблицях. Наприклад, замовлення (Orders) пов'язані з клієнтами (Customers) через ідентифікатор клієнта, а інформація про доставку (DeliveryInfo) пов'язана із замовленнями та адресами клієнтів.

Така структура БД корисна з наступних причин:

- Інтегроване Управління Замовленнями: Сутність "Orders" дозволяє централізовано зберігати всю інформацію про замовлення, спрощуючи процеси обробки, відстеження та аналізу замовлень.
- Клієнтська Інформація: "Customers" зберігає важливу інформацію про клієнтів, що допомагає управляти відносинами з клієнтами (CRM), персоналізувати обслуговування та маркетингові зусилля.
- Різноманітність Замовлень: "OrderType" забезпечує гнучкість у класифікації замовлень, що дозволяє компанії адаптувати свої послуги до різних типів запитів клієнтів.
- Адресна Інформація: "CustomerAddresses" дає змогу зберігати декілька адрес для одного клієнта, що полегшує управління доставками та підвищує точність адресації.
- Управління Доставкою: "DeliveryInfo" включає ключові деталі для координації процесів доставки, що сприяє ефективності логістичних операцій.
- Дані Співробітників: "Employees" дозволяє відслідковувати, які співробітники обробляють конкретні замовлення чи доставки, що важливо для внутрішнього аудиту та відповідальності.

Використання такої структури бази даних може поліпшити продуктивність робочих процесів, забезпечити високий рівень обслуговування клієнтів та

покращити прийняття рішень на основі даних. Крім того, вона може сприяти підвищенню загальної ефективності управління ресурсами компанії та оптимізації взаємодії між різними відділами.

ПЗ DB Browser (SQLite) використовуємо для роботи з базою даних.

3.3 Обробка винятків SQLite (SQLException)

SQLException виникає в таких ситуаціях:

- Помилка підключення до бази даних;
- Порушення правил перевірки;
- Порушення цілісності даних тощо.

Навіть якщо це станеться, це призведе до збою програми та потребує повторного підключення до бази даних MS SQL Server. Щоб гарантувати, що програма продовжує працювати, винятки потрібно перехоплювати та обробляти програмно. Це скасовує погане з'єднання та повертає проєкт до попереднього стану.

Для таблиці «Clients» передбачено, що введені дані зберігаються в таблиці, а винятки обробляються наступним чином.

```

/// <summary>
/// Gets data client data <see cref="ClientData"/> from clients table
/// </summary>
/// </summary>
/// <returns>List of clients <see cref="ClientData"/></returns>
3 references
public Task<List<ClientData>> GetData()
{
    try
    {
        using IDbConnection cnn = new SQLiteConnection(LoadConnectionString());
        return Task.FromResult(cnn.Query<ClientData>(sql: "SELECT * FROM Clients", param: new DynamicParameters()).ToList());
    }
    catch (Exception ex)
    {
        Debug.WriteLine(message: $"Class: {nameof(ClientDataDataAccess)}. Method: {nameof(GetData)}. Message: {ex.Message}");
        throw;
    }
}

```

Рисунок 3.1. Обробка винятків SQLite.

На рисунку 3.1 наведений приклад методу для запиту з таблиці клієнтів який вибирає дані клієнтів з бази даних `SELECT * FROM Clients`. Дані фільтруються за допомогою LINQ технології.

```

1 reference
public Task<List<ClientData>> GetById(int id)
{
    return Task.FromResult(_dataAccess.GetData().Result.Where(x:ClientData => x.Id == id).ToList());
}

```

Рисунок 3.2. Фільтрація даних.

На рисунку 3.2 наведений приклад методу `Task<List<ClientData>> GetById(int id)` який використовується для фільтрації даних клієнта по `id`.

Підтримка Мови Запитів SQL та LINQ: Microsoft Visual Studio забезпечує підтримку мови запитів SQL та Language Integrated Query (LINQ), що дозволяє ефективно працювати з даними під час створення звітів.

Було додано такі звіти:

- по даті та часу доставки клієнту - який відображає інформацію по клієнту та дату й час доставки (див. рис 3.11);
- по кількості замовлень - який відображає назву продукції та кількість замовлених одиниць (див. рис 3.12);

Можливо створити різноманітні SQL запити для виконання специфічних завдань. Ось декілька прикладів:

- Отримати повний список замовлень:

```
SELECT * FROM Orders;
```

- Отримати інформацію про клієнтів та їх замовлення:

```

SELECT      Customers.Name,      Customers.Email,      Orders.OrderID,
Orders.OrderDate
FROM Customers

```

JOIN Orders ON Customers.CustomerID = Orders.CustomerID;

3.4 Функції користувача

Після запуску програми з'являється головне меню програми (рисунок 3.3 та 3.4). Доступ до даних можна отримати, натиснувши кнопку меню. На кожній формі є таблиця з даними та відповідні кнопки керування: кнопки додавання даних, видалення, оновлення, пошуку записів, входу в головне меню програми. "

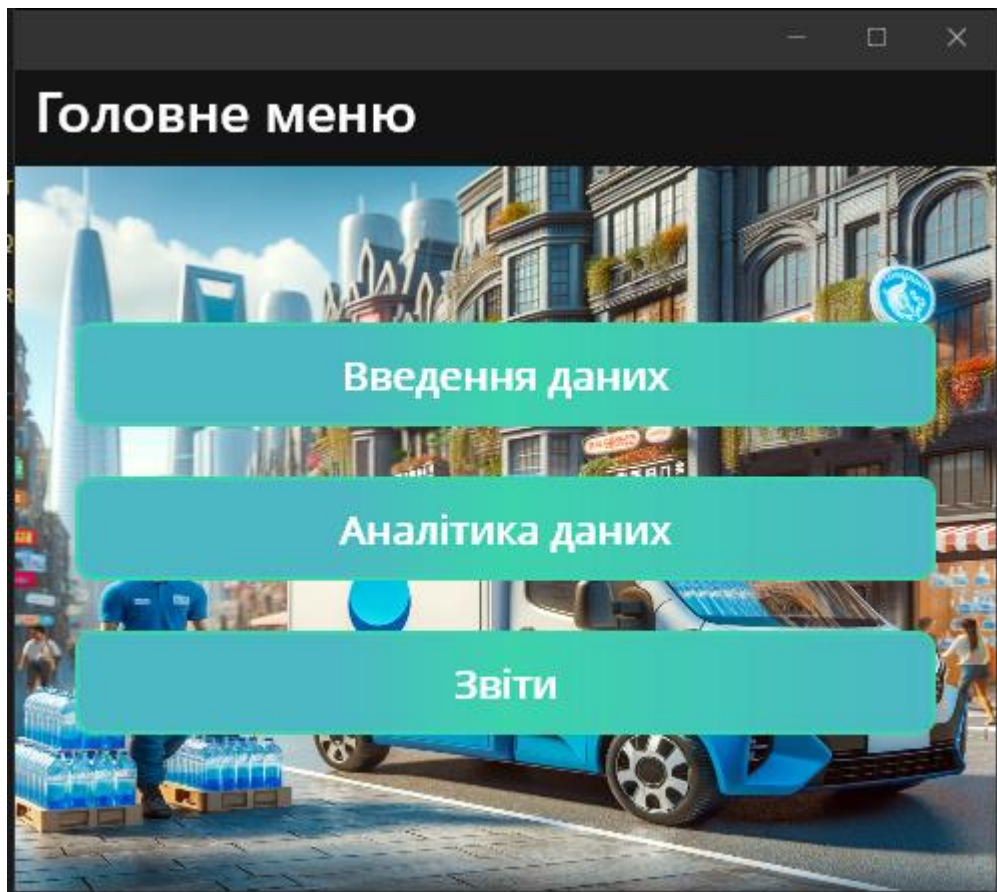


Рисунок 3.3. Головне меню програми ОС Windows.

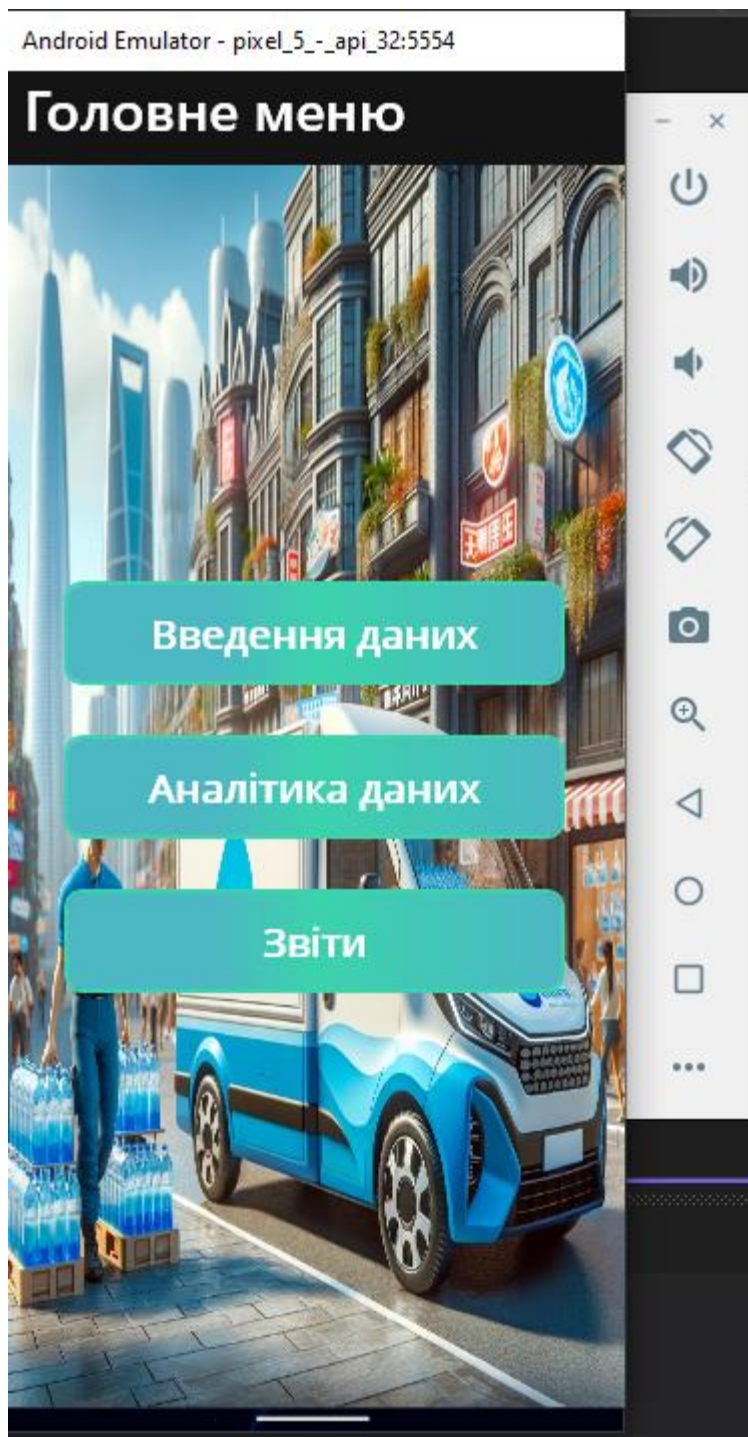


Рисунок 3.4. Головне меню програми ОС Android.

Елемент Введення даних містить такі вікна для запису даних у базу даних: Джерела, Квитанції, Клієнти, Замовлення, Знижки (рис. 3.5)

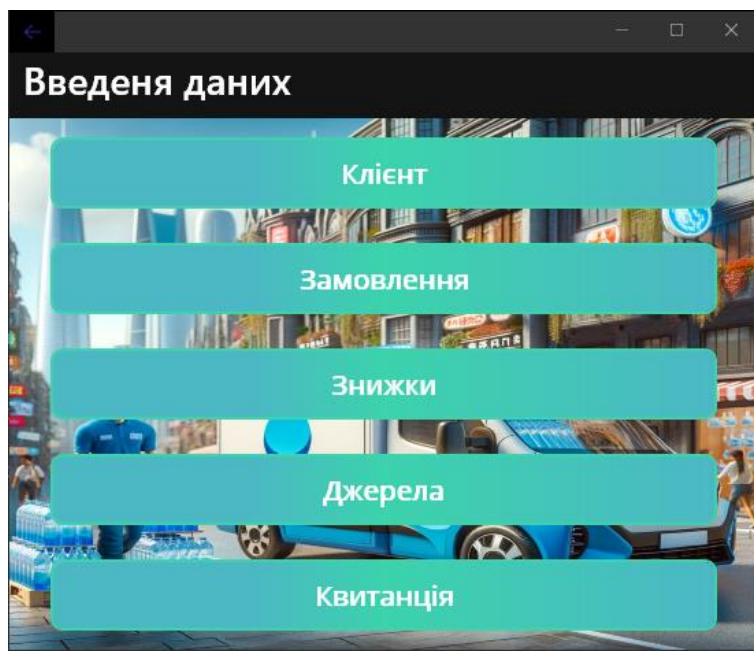
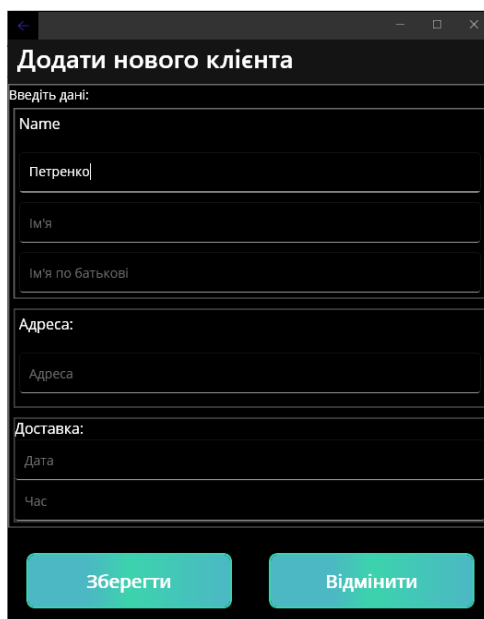


Рисунок 3.5. Опції для запису даних у базу даних.

Розглянемо на прикладі вікна “Клієнти”, заповнивши форму. Після клацання на вибраному елементі відкриється вікно форми (Рисунок. 3.6). Далі ми почнемо нового клієнта вводячи необхідні дані (Рисунок. 3.7).

Рисунок 3.6. Вікно вводу даних меню клієнт.



Додати нового клієнта

Введіть дані:

Name
Петренко

Ім'я

Ім'я по батькові

Адреса:
Адреса

Доставка:
Дата
Час

Зберегти Відмінити

Рисунок 3.7. Введення даних вікно нового клієнта.

Для збереження даних необхідно натиснути кнопку зберегти, а для скасування - відмінити. Створено відповідні форми для полегшення перегляду/додавання/редагування даних про співробітників, обладнання, звітів, додатків, підрозділів та табелів обліку робочого часу.

Меню аналітика даних має такі вікна (Рисунок 3.8).

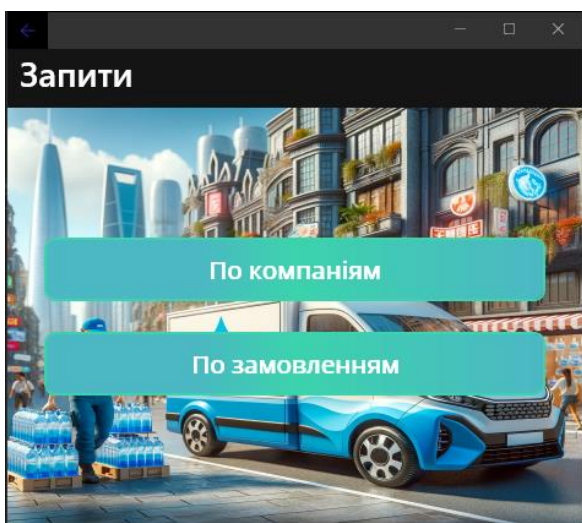


Рисунок 3.8. Виведення даних вікно аналітика даних.

Для перегляду компаній необхідно вибрати запит по компаніях та натиснути на кнопку (Рисунок 3.9)



Рисунок 3.9. Результат виконання запиту по компаніях.

Вікно звіти містить зразки звітів для створеної бази даних (рисунок 3.10). Важливу роль в аналізі роботи компанії відіграють звіти. Звіти дозволяють обробляти та аналізувати дані, що містяться в них.

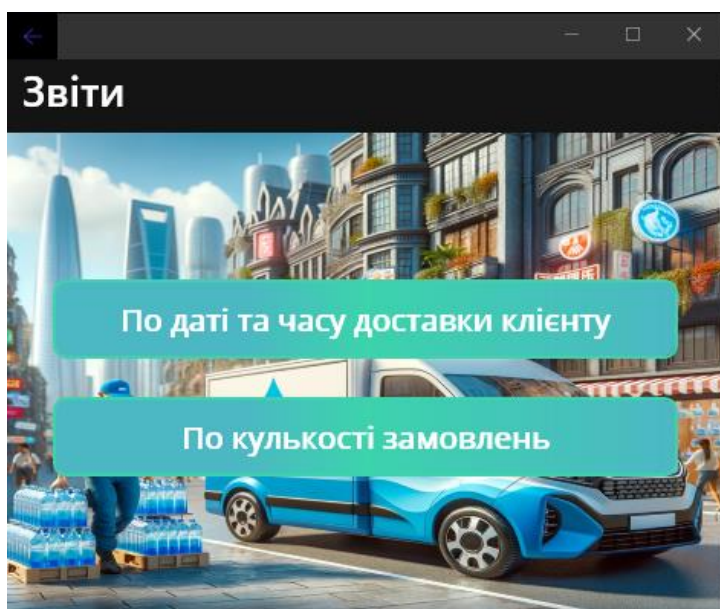


Рисунок 3.10. Вікно звіти.

На рисунку 3.11 користувач має змогу переглянути звіт по клієнтах. Рисунок 3.12 - звіт за кількістю замовлень та їх видам.

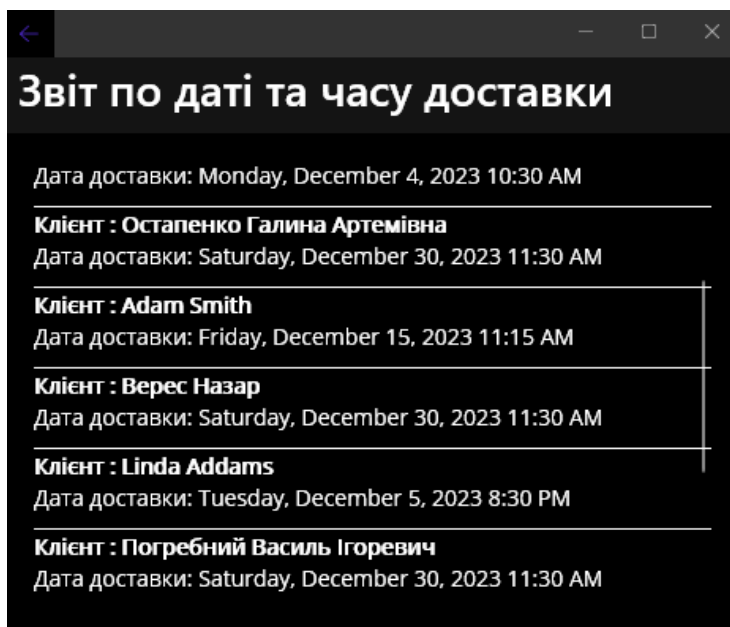


Рисунок 3.11. Звіт по клієнтах.

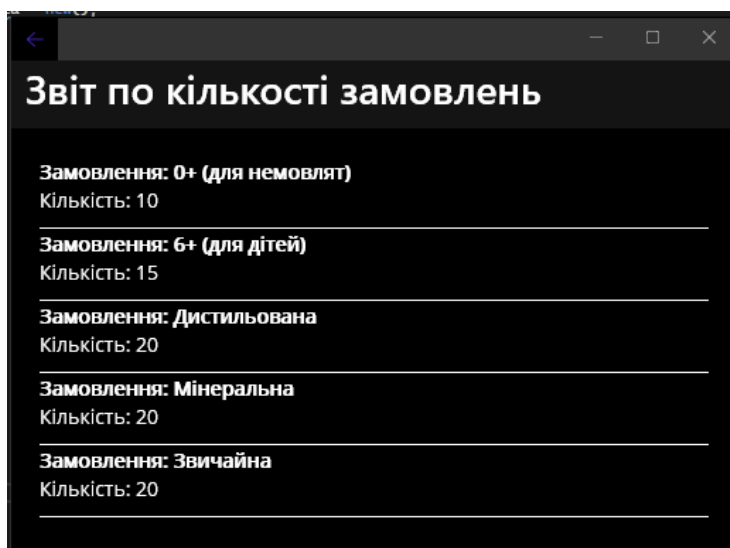


Рисунок 3.12. Звіт за кількістю замовлень та їх види.

3.5 Висновок до розділу № 3

У цьому розділі роботи зосереджено увагу на створенні та апробації системи підтримки формування замовлень та організації постачання фасованої води. Розділ включає огляд методів виконання задачі, процес створення бази даних, обробку винятків SQLite, та розробку вказівок для користувачів. Основні висновки з цього розділу можна сформулювати наступним чином:

- **Методи виконання задачі:** Реалізація системи була здійснена з використанням передових методів програмування та розробки, що забезпечили ефективність та надійність управління замовленнями і доставкою. Адаптація цих методів під конкретні потреби проекту дозволила досягти високого рівня функціональності та користувацької зручності.

- **Створення бази даних:** Розробка бази даних була ключовим елементом системи, що дозволила забезпечити ефективне зберігання, обробку та аналіз великих обсягів інформації. Структура бази даних була оптимізована для забезпечення швидкого доступу та високої надійності.

- **Обробка винятків SQLite (SQLException):** Ефективна обробка винятків в SQLite є важливою для запобігання перебоям у роботі системи. Розробка механізмів обробки винятків допомогла підвищити стабільність та надійність системи, забезпечуючи безперервну роботу навіть у випадку помилок або нештатних ситуацій.

- **Вказівки для користувачів:** Розробка детальних вказівок для користувачів є важливою для забезпечення легкого та інтуїтивно зрозумілого використання системи. Це сприяє підвищенню ефективності використання системи та зниженню можливості помилок з боку користувачів.

У сукупності, розробка та апробація системи демонструє глибокий аналіз та розуміння потреб управління замовленнями та доставкою фасованої води, а також ефективного впровадження передових технологій та методів програмування для створення надійної та зручної у використанні системи.

ВИСНОВКИ

У рамках виконаної наукової роботи було здійснено всебічне дослідження процесів управління замовленнями та логістики доставки бутильованої води, що стали об'єктом нашого аналізу. Основну увагу було приділено вивченню та розробці алгоритмів та програмних рішень для оптимізації та автоматизації цих процесів, які виступали предметом дослідження. Основна мета полягала у підвищенні ефективності роботи підприємств, зайнятих у сфері доставки бутильованої води, через впровадження сучасних технологічних рішень.

Наукова новизна одержаних у ході дослідження результатів полягає у створенні комплексної системи, яка інтегрується з різними базами даних та дозволяє ефективно використовувати можливості мобільних та десктопних платформ завдяки єдиній кодовій базі Microsoft MAUI. Це створює основу для гнучкості, масштабованості та доступності системи управління замовленнями та доставки, що є важливим кроком у розвитку бізнес-процесів сучасних підприємств.

Подальший розвиток системи може включати інтеграцію з іншими інформаційними системами, розробку додаткових модулів для розширення аналітичних можливостей та вдосконалення інтерфейсу користувача. Це не лише збільшить функціональні можливості системи, але й поліпшить взаємодію з кінцевими користувачами, сприяючи зростанню задоволеності клієнтів та ефективності роботи персоналу.

Практична значимість отриманих результатів виявляється у зменшенні часу обробки замовлень, підвищенні точності планування доставок та поліпшенні загальної задоволеності клієнтів. Така система дозволяє підприємствам бути більш

конкурентоспроможними на ринку, ефективно реагувати на змінювані потреби ринку та масштабувати свою діяльність без значних витрат на розширення та модернізацію інфраструктури.

В цілому, дослідження та розробка інформаційної системи з використанням технології Microsoft MAUI для управління замовленнями та доставки бутильованої води є важливим кроком у напрямку технологічної трансформації та цифровізації процесів у цій сфері. Це дозволяє підприємствам не тільки оптимізувати свої поточні процеси, але й надає їм інструменти для адаптації до майбутніх викликів у динамічному бізнес-середовищі.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Бажин И. И. Разработка управленческих решений: компакт-учебник. / И. И. Бажин. – Харків: Консум, 2006. – 316 с..
2. ДСТУ 3918-1999 (ISO/IEC 12207-99) Інформаційні технології. Процеси життєвого циклу програмного забезпечення, 1999. – 57 с.
3. ДСТУ ISO/IEC TR 15504 Інформаційні технології. Оцінювання процесів життєвого циклу програмних засобів, 2002. – 315 с.
4. ДСТУ 3008-2015. Документація. Звіти у сфері науки і техніки. Структура та правила оформлювання – Київ, 2016. – 31 с.
5. Matthew M. Windows Presentation Foundation in .NET 4.5 with examples in C# 5.0 for professionals / MacDonald Matthew., 2012. – 1111 с.
6. Troelsen A. C# 7 programming language and .NET and .NET Core platforms / A. Troelsen, P. Jepiks., 2017. – 1437 с.
7. Adam N. Windows Presentation Foundation Unleashed / Nathan Adam., 2007. – 638 с.
8. Mazloumi N. Building Xamarin.Forms Mobile Apps Using XAML Mobile Cross-Platform XAML and Xamarin.Forms Fundamentals / N. Mazloumi, D. Hermes., 2019. – 454 с.
9. Бідюк П. І. Проектування комп'ютерних інформаційних систем підтримки прийняття рішень: Навчальний посібник / П. І. Бідюк, Л. О. Корченко. – Київ, 2010. – 340 с.
10. SQLite [Електронний ресурс] – Режим доступу до ресурсу: <https://www.sqlite.org/about.html>. Інтернет ресурс: <https://allreferat.com.ua>.

11. NET Multi-platform App UI (.NET MAUI) Community Toolkit documentation [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/dotnet/communitytoolkit/maui/>.
12. Ткачук К. Н. Охорона прав та промислова безпека. Навчальний посібник. / К. Н. Ткачук, В. В. Заічарній., 2010. – 559 с.
13. ДСанПІН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин (57320) – Київ, 1998. – 19 с.
14. Пономаренко Л. А. Комп'ютерні технології управління інноваційними проєктами / Л. А. Пономаренко., 2001. – 423 с.
15. Загоровська Л. Г. Програма виробничої практики студентів 4-го курсу з напряму підготовки 6.050101 – «Комп'ютерні науки» денної форми навчання / Л. Г. Загоровська, О. М. М'якішко, 2012. – 14 с.
16. Довгань Л. Управління проєктами / Л. Довгань, Г. Мохонько, І. Малик., 2017. – 429 с.

ДОДАТКИ

ДОДАТОК А

МОДЕЛЬ ДАНИХ (SQLITE MAESTRO).

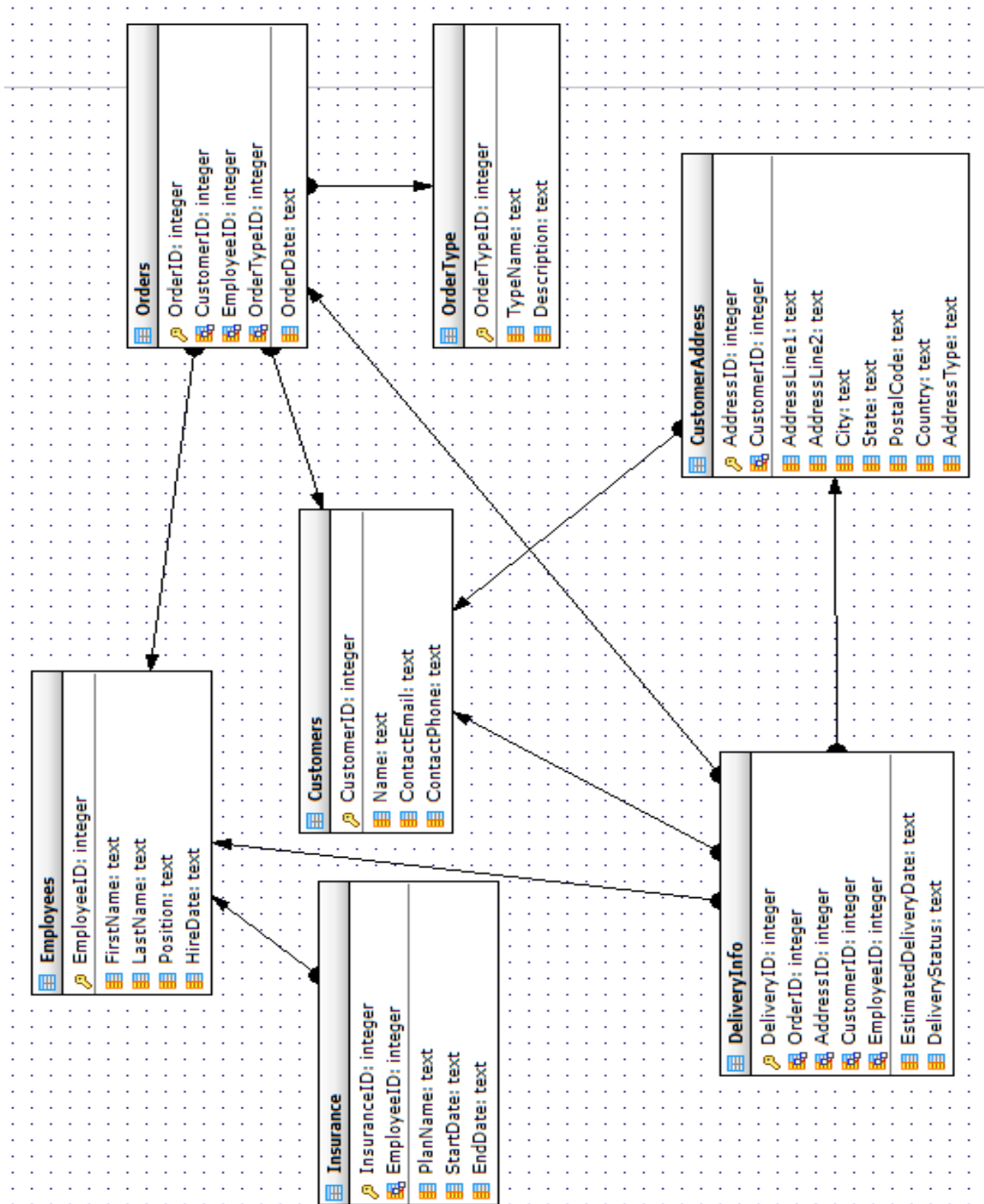


Рисунок А.1. Модель даних

ДОДАТОК Б

СТВОРЕННЯ БАЗИ ДАНИХ SQLITE

Створення бази даних.

sqlite3 MasterProjectWater.db

Таблиця співробітника.

```
CREATE TABLE Employees (  
    EmployeeID INTEGER PRIMARY KEY AUTOINCREMENT,  
    FirstName TEXT NOT NULL,  
    LastName TEXT NOT NULL,  
    Position TEXT,  
    HireDate TEXT  
);
```

Таблиця клієнта.

```
CREATE TABLE Customers (  
    CustomerID INTEGER PRIMARY KEY AUTOINCREMENT,  
    Name TEXT NOT NULL,  
    ContactEmail TEXT,  
    ContactPhone TEXT  
);
```

Таблиця адреса клієнта.

```
CREATE TABLE CustomerAddress (  
    AddressID INTEGER PRIMARY KEY AUTOINCREMENT,  
    CustomerID INTEGER,  
    AddressLine1 TEXT NOT NULL,  
    AddressLine2 TEXT,  
    City TEXT NOT NULL,  
    State TEXT NOT NULL,  
    PostalCode TEXT NOT NULL,  
    Country TEXT NOT NULL,  
    AddressType TEXT,  
    FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID) ON  
DELETE CASCADE  
);
```

Таблиця замовлень.

```
CREATE TABLE Orders (  
    OrderID INTEGER PRIMARY KEY AUTOINCREMENT,  
    CustomerID INTEGER,  
    EmployeeID INTEGER,  
    OrderDate TEXT NOT NULL,  
    OrderTypeID INTEGER,  
    FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID),  
    FOREIGN KEY (EmployeeID) REFERENCES Employees (EmployeeID),  
    FOREIGN KEY (OrderTypeID) REFERENCES OrderType (OrderTypeID)  
);
```

Таблиця типів замовлень.

```
CREATE TABLE OrderType (  
    OrderTypeID INTEGER PRIMARY KEY AUTOINCREMENT,  
    TypeName TEXT NOT NULL,  
    Description TEXT  
);
```

Таблиця страхування співробітників.

```
CREATE TABLE Insurance (  
    InsuranceID INTEGER PRIMARY KEY AUTOINCREMENT,  
    EmployeeID INTEGER,  
    PlanName TEXT NOT NULL,  
    StartDate TEXT,  
    EndDate TEXT,  
    FOREIGN KEY (EmployeeID) REFERENCES Employees (EmployeeID)  
);
```

Таблиця з інформацією про доставку.

```
CREATE TABLE DeliveryInfo (  
    DeliveryID INTEGER PRIMARY KEY AUTOINCREMENT,  
    OrderID INTEGER,  
    AddressID INTEGER,  
    CustomerID INTEGER,
```

```
EmployeeID INTEGER,  
EstimatedDeliveryDate TEXT,  
DeliveryStatus TEXT,  
FOREIGN KEY (OrderID) REFERENCES Orders (OrderID) ON DELETE  
CASCADE,  
FOREIGN KEY (AddressID) REFERENCES CustomerAddress (AddressID) ON  
DELETE SET NULL,  
FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID) ON  
DELETE SET NULL,  
FOREIGN KEY (EmployeeID) REFERENCES Employees (EmployeeID) ON  
DELETE SET NULL  
);
```

ДОДАТОК В

РІШЕННЯ ПРОЄКТУ(Project Source)

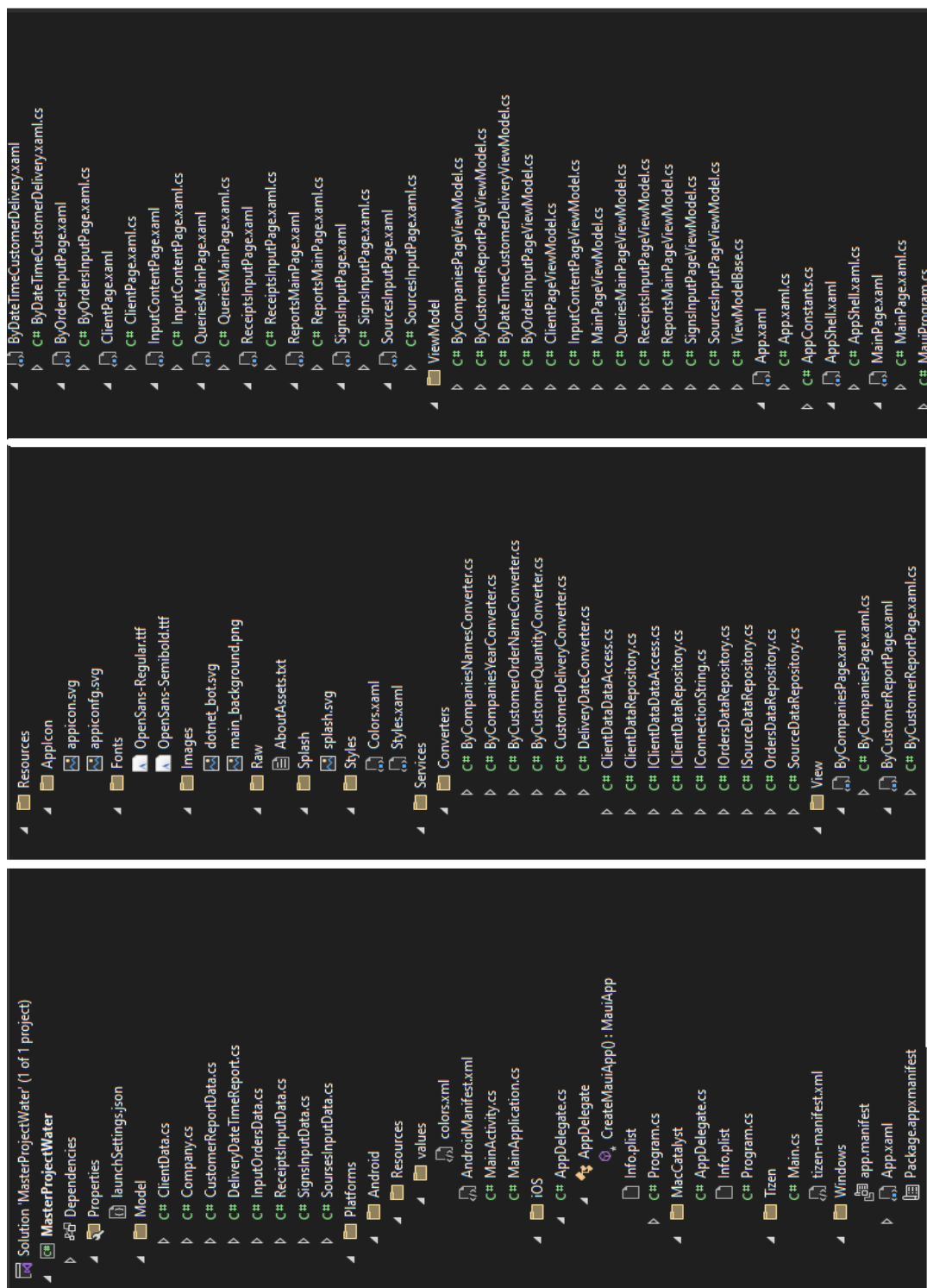


Рисунок В.1. Рішення проєкту.

ДОДАТОК Г

ДІАГРАМА ЗАЛЕЖНОСТЕЙ (Dependency diagram)



Рисунок Г.1. Діаграма залежностей частина 1.

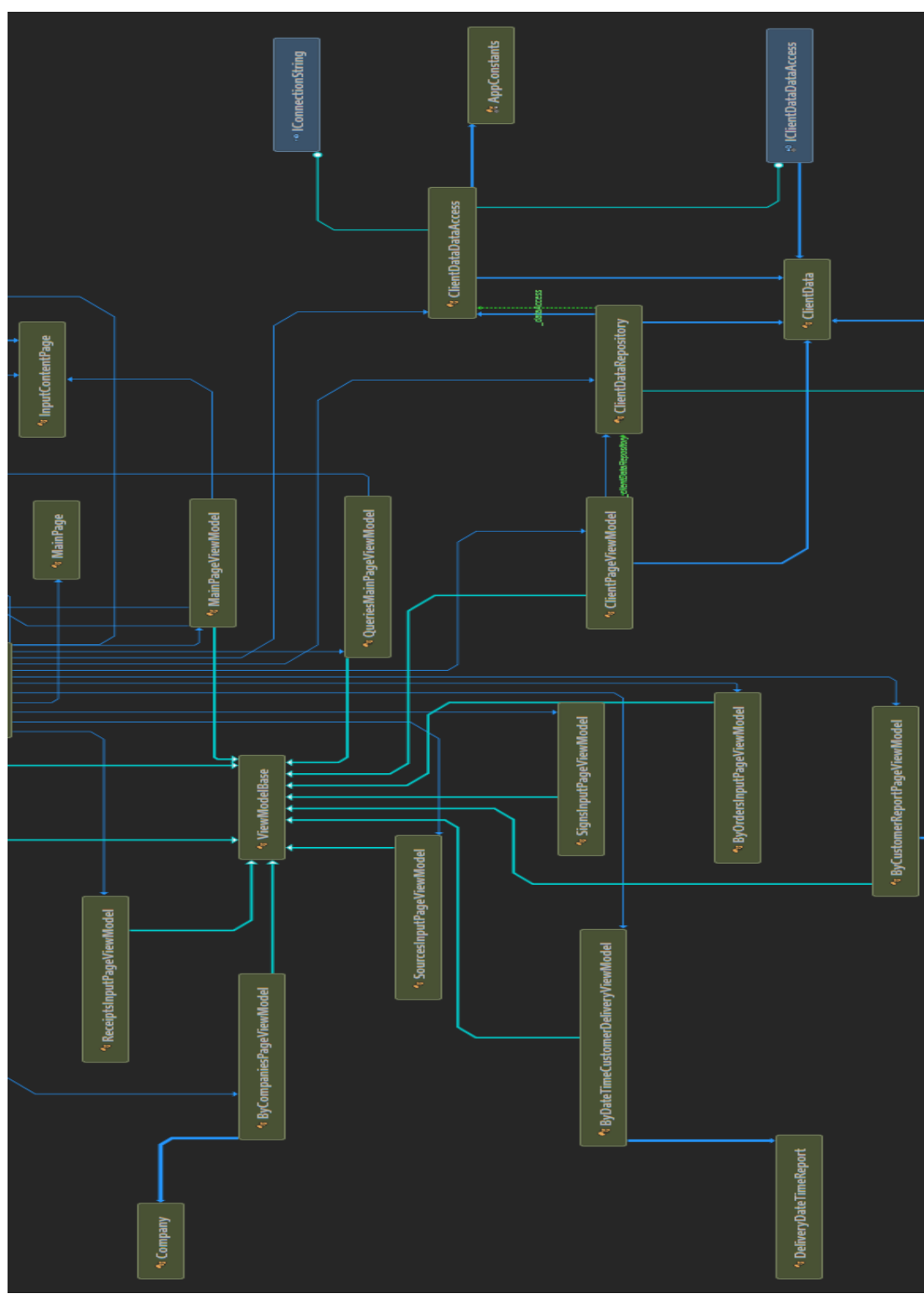


Рисунок Г.1. Діаграма залежностей частина 2.

ДОДАТОК Д КОД ПРОГРАМИ

Головне меню View

```
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:viewModel="clr-namespace:MasterProjectWater.ViewModel"
  x:DataType="viewModel:MainPageViewModel"
  x:Class="MasterProjectWater.MainPage"
  Title="Головне меню">
  <Grid>
    <Image Source="main_background.png" Aspect="Fill"/>
    <ScrollView ZIndex="99">
      <VerticalStackLayout
        Spacing="25"
        Padding="30,0"
        VerticalOptions="Center">
        <Button Text="Введення даних"
          Command="{Binding GoToInputPageCommand}"/>
        <Button Text="Запити"
          Command="{Binding GoQueriesPageCommand}"/>
        <Button Text="Звіти"
          Command="{Binding GoReportsPageCommand}"/>
      </VerticalStackLayout>
    </ScrollView>
  </Grid>
</ContentPage>
```

Головне меню ViewModel

```
namespace MasterProjectWater.ViewModel
```

```
{  
    public partial class MainPageViewModel : ViewModelBase  
    {  
        [ICommand]  
        private async Task GoToInputPageAsync()  
        {  
            await Shell.Current.GoToAsync(nameof(InputContentPage),true);  
        }  
  
        [ICommand]  
        private async Task GoQueriesPageAsync()  
        {  
            await Shell.Current.GoToAsync(nameof(QueriesMainPage),true);  
        }  
  
        [ICommand]  
        private async Task GoReportsPageAsync()  
        {  
            }  
            await Shell.Current.GoToAsync(nameof(ReportsMainPage), true);  
        }  
    }  
}
```

Введеня даних View

```

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
  xmlns:viewModel="clr-namespace:MasterProjectWater.ViewModel"
  x:Class="MasterProjectWater.View.InputContentPage"
  x:DataType="viewModel.InputContentPageViewModel"
  Title="Введення даних">
  <Grid>
    <Image Source="main_background.png" Aspect="Fill"/>
    <ScrollView ZIndex="99">
      <VerticalStackLayout
        VerticalOptions="Center"
        Spacing="25"
        Padding="30,0">
        <Button Text="Клієнт" Command="{Binding
GoToClientPageCommand}"/>
        <Button Text="Замовлення" Command="{Binding
GoToClientPageCommand}"/>
        <Button Text="Знижки" Command="{Binding
GoToClientPageCommand}"/>
        <Button Text="Джерела" Command="{Binding
GoToClientPageCommand}"/>
        <Button Text="Квитанція" Command="{Binding
GoToClientPageCommand}"/>
      </VerticalStackLayout>
    </ScrollView>
  </Grid>
</ContentPage>

```

Введеня даних ViewModel

```

namespace MasterProjectWater.ViewModel
{
    public partial class InputContentPageViewModel : ViewModelBase
    {
        [ICommand]
        private async Task GoToClientPageAsync()
        {
            await Shell.Current.GoToAsync(nameof(ClientPage), true);
        }
    }
}

```

Запити View

```

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewModel="using:MasterProjectWater.ViewModel"
    x:DataType="viewModel:QueriesMainPageViewModel"
    x:Class="MasterProjectWater.View.QueriesMainPage"
    Title="Запити">
<ContentPage.Content>
    <Grid>
        <Image Source="main_background.png" Aspect="Fill"/>
        <ScrollView ZIndex="99">
            <VerticalStackLayout
                VerticalOptions="Center"

```

```

        Spacing="25"
        Padding="30,0">
        <Button Text="По компаніям" Command="{Binding
GoToByCompaniesPageCommand}"/>
        <Button Text="По замовленням" />
    </VerticalStackLayout>
</ScrollView>
</Grid>
</ContentPage.Content>
</ContentPage>

```

Запити ViewModel

```
namespace MasterProjectWater.ViewModel
```

```

{
    public partial class QueriesMainPageViewModel : ViewModelBase
    {
        [ICommand]
        private async Task GoToByCompaniesPageAsync()
        {
            await Shell.Current.GoToAsync(nameof(ByCompaniesPage), true);
        }
    }
}

```

Звіти View

```

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"

```

```

xmlns:viewModel="clr-namespace:MasterProjectWater.ViewModel"
x:DataType="viewModel:ReportsMainPageViewModel"
x:Class="MasterProjectWater.View.ReportsMainPage"
Title="Звіти">
<ContentPage.Content>
<Grid>
<Image Source="main_background.png" Aspect="Fill"/>
<ScrollView ZIndex="99">
<VerticalStackLayout
VerticalOptions="Center"
Spacing="25"
Padding="30,0">
<Button Text="По даті та часу доставки клієнту" Command="{Binding
GoToByDateTimeCustomerDeliveryCommand}"/>
<Button Text="По кількості замовлень" Command="{Binding
GoToByCustomerReportPageCommand}"/>
</VerticalStackLayout>
</ScrollView>
</Grid>
</ContentPage.Content>
</ContentPage>

```

Звіти ViewModel

namespace MasterProjectWater.ViewModel

{

public partial class ReportsMainPageViewModel : ViewModelBase

```

{
    [ICommand]
    private async Task GoToByDateTimeCustomerDeliveryAsync()
    {
        await Shell.Current.GoToAsync(nameof(ByDateTimeCustomerDelivery), true);
    }

    [ICommand]
    private async Task GoToByCustomerReportPageAsync()
    {
        await Shell.Current.GoToAsync(nameof(ByCustomerReportPage), true);
    }
}
}

```

Додати нового клієнта View

```

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewModel="clr-namespace:MasterProjectWater.ViewModel"
    x:Class="MasterProjectWater.View.ClientPage"
    x:DataType="viewModel:ClientPageViewModel"
    Title="Додати нового клієнта">
<ContentPage.Content>
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition ></RowDefinition>
            <RowDefinition Height="100"></RowDefinition>

```

```

</Grid.RowDefinitions>
<ScrollView ZIndex="99">
  <VerticalStackLayout>
    <Border StrokeThickness="1" >
      <VerticalStackLayout>
        <Label Text="Введіть дані: "/>
        <Border StrokeThickness="1" Margin="5">
          <VerticalStackLayout>
            <Label Text="Name" FontSize="16" Margin="5"/>
            <Line StrokeThickness="1"
              Margin="5"/>
            <Entry
              Margin="5"
              Placeholder="Прізвище"
              Text="{Binding SecondName}" />
            <Entry
              Margin="5"
              Placeholder="Ім'я"
              Text="{Binding FirstName}" />
            <Entry
              Margin="5"
              Placeholder="Ім'я по батькові"
              Text="{Binding FatherName}" />
            <Line StrokeThickness="1"/>
          </VerticalStackLayout>
        </Border>
      <Border StrokeThickness="1" Margin="5">

```

```
<VerticalStackLayout>
  <Label Text="Адреса:"
    FontSize="16"
    Margin="5"/>
  <Line StrokeThickness="1"
    Margin="5"/>
  <Entry
    Margin="5"
    Placeholder="Адреса"
    Text="{Binding Address}" />
  <Line StrokeThickness="1"
    Margin="5"/>
</VerticalStackLayout>
</Border>
<Border StrokeThickness="1" Margin="5">
  <VerticalStackLayout>
    <Label Text="Доставка:" FontSize="16"/>
    <Line StrokeThickness="1"/>
    <Entry
      Placeholder="Дата"
      Text="{Binding DeliveryDate}" />
    <Entry
      Placeholder="Час"
      Text="{Binding DeliveryTime}" />
    <Line StrokeThickness="1"/>
  </VerticalStackLayout>
</Border>
```

```

        </VerticalStackLayout>
    </Border>
</VerticalStackLayout>
</ScrollView>
<Grid Grid.Row="1">
    <Grid.ColumnDefinitions>
        <ColumnDefinition/>
        <ColumnDefinition/>
    </Grid.ColumnDefinitions>
    <Button Text="Зберегти" Command="{Binding SaveButtonCommand}"
Margin="20"/>
    <Button Grid.Column="1" Text="Відмінити" Command="{Binding
CancelButtonCommand}" Margin="20"/>
</Grid>
</Grid>
</ContentPage.Content>
</ContentPage>

```

Додати нового клієнта ViewModel

```
namespace MasterProjectWater.ViewModel
```

```
{
```

```
    public partial class ClientPageViewModel : ViewModelBase
```

```
    {
```

```
        private readonly ClientDataRepository _clientDataRepository;
```

```
        [ObservableProperty] private string firstName;
```

```
        [ObservableProperty] private string secondName;
```

```
[ObservableProperty] private string fatherName;
[ObservableProperty] private string address;
[ObservableProperty] private string deliveryTime;
[ObservableProperty] private string deliveryDate;

public ClientPageViewModel(ClientDataRepository clientDataRepository)
{
    _clientDataRepository = clientDataRepository;
}

[ICommand]
private async Task SaveButtonAsync()
{
    await _clientDataRepository.Add(new ClientData
    {
        Address = Address,
        FirstName = FirstName,
        SecondName = SecondName,
        FatherName = FatherName,
        DeliveryDate = DeliveryDate,
        DeliveryTime = DeliveryTime
    });
    await CleanData();
    await Shell.Current.Navigation.PopAsync();
}

[ICommand]
```

```

private async Task CancelButtonAsync()
{
    await CleanData();
    await Shell.Current.Navigation.PopAsync();
}

private Task CleanData()
{
    return Task.Run(() =>
    {
        return Address = FatherName = FirstName = SecondName = DeliveryDate =
DeliveryTime = string.Empty;
    });
}
}
}

```

Запит по компаніям View

```

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewModel="clr-namespace:MasterProjectWater.ViewModel"
    x:Class="MasterProjectWater.View.ByCompaniesPage"
    x:DataType="viewModel.ByCompaniesPageViewModel"
    Title="Запит по компаніям">
<VerticalStackLayout>
    <CollectionView Margin="20" ItemsSource="{Binding Companies}">
        <CollectionView.ItemTemplate>

```

```

<DataTemplate>
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="50"/>
      <RowDefinition Height="0.5"/>
    </Grid.RowDefinitions>
    <StackLayout Orientation="Vertical">
      <Label FontAttributes="Bold" Text="{Binding Name,
Converter={StaticResource ByCompaniesNamesConverter}}" TextColor="White"/>
      <Label Text="{Binding Year, Converter={StaticResource
ByCompaniesYearConverter}}" />
    </StackLayout>
    <Grid Grid.Row="1" Background="white"/>
  </Grid>
</DataTemplate>
</CollectionView.ItemTemplate>
</CollectionView>
</VerticalStackLayout>
</ContentPage>

```

Запит по компаніям ViewModel

```
namespace MasterProjectWater.ViewModel
```

```
{
```

```
    public partial class ByCompaniesPageViewModel : ViewModelBase
```

```
    {
```

```
        private readonly IClientDataDataAccess _dataAccess;
```

```

[ObservableProperty] private ObservableCollection<Company> companies = new
();
[ObservableProperty] private string name;
[ObservableProperty] private string year;

public ByCompaniesPageViewModel(IClientDataDataAccess dataAccess)
{
    _dataAccess = dataAccess;
}

public void GetDataByName(string name)
{
    companies = _dataAccess.GetData().Result.Where(x => x.Name.Equals(name));
}
}
}
}

```

Звіт по даті та часу доставки View

```

<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:viewModel="clr-namespace:MasterProjectWater.ViewModel"
    x:DataType="viewModel.ByDateTimeCustomerDeliveryViewModel"
    x:Class="MasterProjectWater.View.ByDateTimeCustomerDelivery"
    Title="Звіт по даті та часу доставки">
<ContentPage.Content>
    <CollectionView Margin="20" ItemsSource="{Binding ReportData}">
        <CollectionView.ItemTemplate>

```

```

<DataTemplate>
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition Height="50"/>
      <RowDefinition Height="0.5"/>
    </Grid.RowDefinitions>
    <StackLayout Orientation="Vertical">
      <Label FontAttributes="Bold" Text="{Binding CustomerFullName,
Converter={StaticResource CustomerDeliveryConverter}}" TextColor="White"/>
      <Label Text="{Binding DeliveryDate, Converter={StaticResource
DeliveryDateConverter}}" />
    </StackLayout>
    <Grid Grid.Row="1" Background="white"/>
  </Grid>
</DataTemplate>
</CollectionView.ItemTemplate>
</CollectionView>
</ContentPage.Content>
</ContentPage>

```

Звіт по даті та часу доставки ViewModel

namespace MasterProjectWater.ViewModel

{

public partial class ByDateTimeCustomerDeliveryViewModel : ViewModelBase

{

private IOrderDataDataAccess _dataAccess;

```

    [ObservableProperty] private ObservableCollection<DeliveryDateTimeReport>
reportData = new();
    [ObservableProperty] private string customerFullName;
    [ObservableProperty] private DateTime deliveryDate;

    public ByDateTimeCustomerDeliveryViewModel(IOrderDataDataAccess
dataAccess)
    {
        _dataAccess = dataAccess;
    }
    public void GetOerDataById(int id)
    {
        reportData = _dataAccess.GetData().Result.Where(x => x.Id.Equals(id));
    }
}
}

```

Приклад конвертора що конвертую назву компанії в необхідну строку для відображення в view

```

namespace MasterProjectWater.Services.Converters
{
    internal class ByCompaniesNamesConverter : IValueConverter
    {
        public object Convert(object value, Type targetType, object parameter, CultureInfo
culture)
        {
            if (value is string name)

```

```

    {
        return $"Назва: {name}";
    }
    return value;
}

```

```

public object ConvertBack(object value, Type targetType, object parameter,
CultureInfo culture)
{
    return value;
}
}
}

```

Приклад сервісу отримання даних клієнта з бази даних

```

namespace MasterProjectWater.Services
{
    public interface IClientDataDataAccess
    {
        Task<List<ClientData>> GetData();
        Task Delete();
        Task InsertData(ClientData clientData);
    }
}

namespace MasterProjectWater.Services
{

```

```

public class ClientDataDataAccess : IClientDataDataAccess, IConnectionString
{
    /// <summary>
    /// Gets data client data <see cref="ClientData"/> from clients table
    /// </summary>
    /// <returns>List of clients <see cref="ClientData"/></returns>
    public Task<List<ClientData>> GetData()
    {
        try
        {
            using IDbConnection cnn = new SQLiteConnection(LoadConnectionString());
            return Task.FromResult(cnn.Query<ClientData>("SELECT * FROM
Clients", new DynamicParameters()).ToList());
        }
        catch (Exception ex)
        {
            Debug.WriteLine($"Class: {nameof(ClientDataDataAccess)}. Method:
{nameof(GetData)}. Message: {ex.Message}");
            throw;
        }
    }

    public async Task InsertData(ClientData clientData)
    {
        await Task.Run(() =>
        {
            using IDbConnection cnn = new SQLiteConnection(LoadConnectionString());

```

```

        cnn.Execute("INSERT INTO Clients (firstName, secondName, fatherName,
address, deliveryTime, deliveryDate) VALUES (@FirstName, @SecondName,
@FatherName, @Address, @DeliveryTime, @DeliveryDate)", clientData);
    });
}

public async Task Delete()
{

    await Task.Run(() =>
    {
        using IDbConnection cnn = new SQLiteConnection(LoadConnectionString());
        var output = cnn.Query<SqliteConnection>("DELETE FROM Clients", new
DynamicParameters());
    });
}

public string LoadConnectionString() => $"Data
Source={AppConstants.DbPath};Version=3;";
}
}

```

Базовий клас для view models

```
namespace MasterProjectWater.ViewModel
```

```
{
    public partial class ViewModelBase : ObservableObject
    {

```

```

[CommunityToolkit.Mvvm.ComponentModel.ObservableProperty]
[AlsoNotifyChangeFor(nameof(IsNotBusy))]
    public bool isBusy;

[CommunityToolkit.Mvvm.ComponentModel.ObservableProperty]
    public string title;

    public bool IsNotBusy => !IsBusy;
}
}

```

Приклад стилів

```

<LinearGradientBrush x:Key="GradientButtonBackground"
    StartPoint="0,0" EndPoint="1,0">
    <GradientStop Color="#4CB8C4" Offset="0.3"/>
    <GradientStop Color="#3CD3AD" Offset="0.5"/>
    <GradientStop Color="#4CB8C4" Offset="1.0"/>
</LinearGradientBrush>

<Style TargetType="Button">
    <Setter Property="TextColor" Value="White" />
    <Setter Property="Background" Value="{StaticResource
GradientButtonBackground}" />
    <Setter Property="FontFamily" Value="OpenSansRegular"/>
    <Setter Property="FontAttributes" Value="Bold"/>
    <Setter Property="FontSize" Value="20"/>
    <Setter Property="CornerRadius" Value="8"/>
    <Setter Property="Padding" Value="14,10"/>

```

```

<Setter Property="MinimumHeightRequest" Value="44"/>
<Setter Property="MinimumWidthRequest" Value="44"/>
<Setter Property="BorderColor" Value="#3CD3AD"/>
<Setter Property="BorderWidth" Value="2"/>
<Setter Property="VisualStateManager.VisualStateGroups">
  <VisualStateGroupList>
    <VisualStateGroup x:Name="CommonStates">
      <VisualState x:Name="Normal" />
      <VisualState x:Name="Disabled">
        <VisualState.Setters>
          <Setter Property="TextColor" Value="{AppThemeBinding
Light={StaticResource Gray950}, Dark={StaticResource Gray200}}" />
          <Setter Property="BackgroundColor" Value="{AppThemeBinding
Light={StaticResource Gray200}, Dark={StaticResource Gray600}}" />
        </VisualState.Setters>
      </VisualState>
    </VisualStateGroup>
  </VisualStateGroupList>
</Setter>
</Style>

```

Реєстрація сервісів

```

namespace MasterProjectWater
{
  public static class MauiProgram
  {
    public static MauiApp CreateMauiApp()

```

```
{
    var builder = MauiApp.CreateBuilder();
    builder
        .UseMauiApp<App>()
        .ConfigureFonts(fonts =>
        {
            fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
            fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
        });

    #if DEBUG
        builder.Logging.AddDebug();
    #endif

    // View abd view models
    builder.Services.AddSingleton<MainPage>();
    builder.Services.AddSingleton<MainPageViewModel>();
    builder.Services.AddSingleton<InputContentPage>();
    builder.Services.AddSingleton<InputContentPageViewModel>();
    builder.Services.AddSingleton<ClientPage>();
    builder.Services.AddSingleton<ClientPageViewModel>();
    builder.Services.AddSingleton<QueriesMainPage>();
    builder.Services.AddSingleton<QueriesMainPageViewModel>();
    builder.Services.AddSingleton<ByCompaniesPage>();
    builder.Services.AddSingleton<ByCompaniesPageViewModel>();
    builder.Services.AddSingleton<ReportsMainPage>();
    builder.Services.AddSingleton<ReportsMainPageViewModel>();
    builder.Services.AddSingleton<ByDate Time Customer Delivery>();
```

```

builder.Services.AddSingleton<ByDateTimeCustomerDeliveryViewModel>();
builder.Services.AddSingleton<ByCustomerReportPage>();
builder.Services.AddSingleton<ByCustomerReportPageViewModel>();
builder.Services.AddSingleton<ByOrdersInputPage>();
builder.Services.AddSingleton<ByOrdersInputPageViewModel>();
builder.Services.AddSingleton<ReceiptsInputPage>();
builder.Services.AddSingleton<ReceiptsInputPageViewModel>();
builder.Services.AddSingleton<SourcesInputPage>();
builder.Services.AddSingleton<SourcesInputPageViewModel>();
builder.Services.AddSingleton<SignsInputPage>();
builder.Services.AddSingleton<SignsInputPageViewModel>();

// services
builder.Services.AddSingleton<ClientDataRepository>();
builder.Services.AddSingleton<ClientDataDataAccess>();

return builder.Build();
}
}
}

```

Реєстрація маршрутів

```

namespace MasterProjectWater
{
    public partial class AppShell : Shell
    {
        public AppShell()
    }
}

```

```
{
    InitializeComponent();

    //Register route page in shell
    Routing.RegisterRoute(nameof(InputContentPage), typeof(InputContentPage));
    Routing.RegisterRoute(nameof(ClientPage), typeof(ClientPage));
    Routing.RegisterRoute(nameof(QueriesMainPage), typeof(QueriesMainPage));
    Routing.RegisterRoute(nameof(ByCompaniesPage),
typeof(ByCompaniesPage));
    Routing.RegisterRoute(nameof(ReportsMainPage), typeof(ReportsMainPage));
    Routing.RegisterRoute(nameof(ByDateTimeCustomerDelivery),
typeof(ByDateTimeCustomerDelivery));
    Routing.RegisterRoute(nameof(ByCustomerReportPage),
typeof(ByCustomerReportPage));
    Routing.RegisterRoute(nameof(ByOrdersInputPage),
typeof(ByOrdersInputPage));
    Routing.RegisterRoute(nameof(ReceiptsInputPage),
typeof(ReceiptsInputPage));
    Routing.RegisterRoute(nameof(SignsInputPage), typeof(SignsInputPage));
    Routing.RegisterRoute(nameof(SourcesInputPage), typeof(SourcesInputPage));
}
}
```