

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ПОГОДЖЕНО
Декан факультету

ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ
Завідувач кафедри інформаційних
систем і технологій

(підпис) **Ігор БОЛБОТ**

(підпис) **Михайло ШВИДЕНКО**

« ____ » _____ 2026 р.

« ____ » _____ 2026 р.

БАКАЛАВРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

**«Програмний застосунок для аналізу ефективності управління
малим бізнесом»**

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Гарант освітньої програми
д.е.н., професор

(підпис) **Роман РУДЕНСЬКИЙ**

**Керівник бакалаврської
кваліфікаційної роботи**
д.т.н., професор

(підпис) **Вікторія СМОЛІЙ**

Виконав

(підпис) **Юрій ШАГОВИЙ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри інформаційних систем і
технологій

к.е.н., доцент _____ Михайло ШВИДЕНКО
(підпис)

«___» _____ 2026 р.

ЗАВДАННЯ
ДО ВИКОНАННЯ БАКАЛАВРСЬКОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
ЗДОБУВАЧУ
Шаговому Юрію Віталійовичу
(прізвище, ім'я, по батькові)

Спеціальність «122 Комп'ютерні науки»

Освітньо-професійна програма «Комп'ютерні науки»

Тема бакалаврської кваліфікаційної роботи

«Програмний застосунок для аналізу ефективності управління малим бізнесом»

затверджена наказом від «6» січня 2026 р. № 46 «С»

Термін подання завершеної роботи на кафедру «22» травня 2026 р.

Вихідні дані до бакалаврської кваліфікаційної роботи (проєкту):
методи аналізу та проєктування програмних систем, об'єктно-орієнтований підхід,
Python, SQLite, вимоги до застосунку для аналізу ефективності управління малим
бізнесом.

Перелік питань, що підлягають дослідженню:

аналіз предметної області; визначення вимог; проєктування системи та бази даних;
вибір технологій; розробка застосунку на Python; тестування та впровадження. Дата
видачі завдання «6» січня 2026 р.

Керівник бакалаврської
кваліфікаційної роботи

(підпис)

Вікторія СМОЛІЙ

Завдання взяв до
виконання

(підпис)

Юрій ШАГОВИЙ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОГО ЗАСТОСУНКУ.....	10
1.1 Опис предметної області.....	10
1.2 Аналіз існуючих програмних рішень.....	12
1.4 Аналіз вимог до програмного застосунку.....	20
1.5 Постановка завдання.....	21
1.6 Висновки до першого розділу.....	23
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	25
2.1 Логічна модель даних програмного застосунку.....	25
2.2 Фізична модель бази даних.....	28
2.3 Діаграма пакетів програмного забезпечення.....	33
2.4 Діаграма розгортання програмного застосунку.....	35
2.5 Проєктування програмних модулів системи.....	36
2.6 Висновки до другого розділу.....	38
3 РОЗРОБКА ПРОГРАМНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ...39	39
3.1 Обґрунтування вибору технологічних засобів реалізації.....	39
3.2 Практична реалізація інформаційної бази системи.....	40
3.3 Алгоритмізація програмних модулів системи.....	42
3.4 Реалізація модуля фінансового аналізу.....	46
3.5 Реалізація модуля KPI та рекомендацій.....	47
3.6 Реалізація модуля звітності та інтерфейсу користувача.....	48
3.7 Висновки до третього розділу.....	50
4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ.....	51
4.1 План тестування програмних модулів та методика оцінювання результатів.....	51

4.2 Тестування програмного застосунку для аналізу ефективності управління малим бізнесом.....	53
4.3 Оцінювання ефективності роботи системи.....	61
4.4 Висновки до четвертого розділу.....	63
ВИСНОВКИ.....	64
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67
ДОДАТОК А.....	70
ДОДАТОК Б.....	72
ДОДАТОК В.....	74
ДОДАТОК Г.....	77

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- API – прикладний програмний інтерфейс
- БД – база даних
- БКР – бакалаврська кваліфікаційна робота
- ІС – інформаційна система
- ІТ – інформаційні технології
- КЗ – коефіцієнт завантаження або контрольне значення показника
- ККД – коефіцієнт корисної дії
- КПЕ – ключовий показник ефективності
- ПЗ – програмне забезпечення
- СУБД – система управління базами даних
- ФОП – фізична особа-підприємець
- ERP – система планування ресурсів підприємства
- CRM – система управління взаємовідносинами з клієнтами
- KPI – ключовий показник ефективності
- CAC – вартість залучення одного клієнта або однієї успішної угоди
- ROI – показник повернення інвестицій
- CSV – текстовий формат для збереження табличних даних
- TXT – текстовий формат збереження звітної інформації
- SQL – мова структурованих запитів до бази даних
- SQLite – вбудована локальна система управління базами даних
- Python – мова програмування, використана для реалізації програмного застосунку
- PyQt6 – бібліотека Python для створення графічного інтерфейсу користувача
- GUI – графічний інтерфейс користувача
- UI – інтерфейс користувача
- UX – досвід користувача під час роботи з програмним продуктом
- RBAC – рольова модель керування доступом

SHA-256 – криптографічна хеш-функція, використана для перетворення паролів

ВСТУП

Сучасні умови ведення малого бізнесу характеризуються високою конкуренцією, швидкою зміною попиту, зростанням витрат, необхідністю оперативного контролю фінансових результатів і потребою в обґрунтованих управлінських рішеннях. Для малого підприємства важливо не лише фіксувати доходи та витрати, а й регулярно аналізувати ефективність роботи, виявляти проблемні напрями, контролювати виконання планових показників і своєчасно реагувати на негативні зміни.

Актуальність теми зумовлена тим, що значна частина малих підприємств веде фінансовий та операційний облік у розрізнених електронних таблицях, окремих CRM-системах, бухгалтерських сервісах або вручну. Такий підхід ускладнює комплексний аналіз діяльності, оскільки дані про продажі, витрати, клієнтів, замовлення, ліди, повернення та маркетингові витрати часто не об'єднані в єдиному інформаційному середовищі. У результаті власник або керівник не завжди може швидко оцінити прибутковість, маржинальність, конверсію, середній чек, вартість залучення клієнта та інші показники, які безпосередньо впливають на якість управління.

Існуючі програмні рішення, такі як QuickBooks, Xero, Zoho Books, Odoo та HubSpot CRM, мають широкі функціональні можливості для бухгалтерського обліку, фінансової звітності, управління клієнтами, продажами та бізнес-процесами. QuickBooks орієнтований на облік доходів, витрат, інвойсів, cash flow та фінансових звітів; Xero підтримує фінансову панель для контролю cash flow, рахунків і ключових бізнес-показників; Zoho Books забезпечує бухгалтерські функції, інвойси, звіти й управління запасами; Odoo поєднує CRM, бухгалтерію, склад, POS і проєктне управління; HubSpot CRM надає інструменти dashboard-звітності й аналізу продажів. Однак для навчальної та прикладної задачі доцільним є створення окремого програмного застосунку, який має простішу структуру, локальну базу даних і фокус саме на аналізі KPI малого бізнесу.

Метою кваліфікаційної роботи є розроблення програмного застосунку для аналізу ефективності управління малим бізнесом, який забезпечує облік фінансових операцій, збереження операційних показників, розрахунок ключових показників ефективності, формування звітів і підготовку управлінських рекомендацій.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

- проаналізувати предметну область управління малим бізнесом;
- розглянути існуючі програмні рішення для фінансового, операційного та управлінського аналізу;
- визначити функціональні, нефункціональні, технічні та безпекові вимоги до програмного застосунку;
- розробити логічну та фізичну модель бази даних;
- спроектувати структуру програмних модулів і алгоритми роботи системи;
- реалізувати програмний застосунок з графічним інтерфейсом користувача;
- провести тестування основних функцій системи;
- оцінити ефективність розробленого програмного забезпечення та визначити напрями його подальшого розвитку.

Об'єктом дослідження є процес управління діяльністю малого бізнесу на основі фінансових та операційних показників.

Предметом дослідження є методи, моделі та програмні засоби аналізу ефективності управління малим бізнесом, розрахунку KPI, формування звітів і управлінських рекомендацій.

Методи дослідження включають системний аналіз предметної області, порівняльний аналіз існуючих програмних рішень, UML-моделювання, логічне та фізичне проектування бази даних, алгоритмізацію основних процесів, об'єктно-орієнтоване програмування, функціональне тестування та оцінювання результатів роботи системи.

Практичне значення роботи полягає у створенні програмного прототипу, який може використовуватися для оперативного аналізу діяльності малого підприємства. Застосунок дозволяє вносити фінансові операції, обліковувати операційні показники, розраховувати дохід, витрати, прибуток, маржинальність, конверсію, середній чек, САС і рівень повернень, а також формувати короткий управлінський висновок і рекомендації.

Програмне забезпечення реалізовано з використанням мови Python, бібліотеки PyQt6 для створення графічного інтерфейсу та SQLite для локального збереження даних. Python має вбудований модуль sqlite3 для роботи з SQLite через DB-API 2.0, а PyQt6 є набором Python-прив'язок до Qt для створення графічних застосунків. Такий підхід дозволяє створити автономний настільний застосунок, який може працювати на персональному комп'ютері користувача без складного серверного розгортання.

Структура кваліфікаційної роботи складається зі вступу, чотирьох розділів, загальних висновків, списку використаних джерел і додатків. У першому розділі виконано аналіз предметної області та існуючих рішень. У другому розділі здійснено проєктування інформаційного та програмного забезпечення. У третьому розділі описано реалізацію програмного застосунку, фізичну модель бази даних і алгоритми роботи модулів. У четвертому розділі проведено тестування програмного забезпечення та оцінювання його ефективності.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ ПРОГРАМНОГО ЗАСТОСУНКУ

1.1 Опис предметної області

Малий бізнес є однією з найбільш гнучких форм підприємницької діяльності, оскільки швидко реагує на зміни попиту, сезонність, поведінку клієнтів, рівень витрат і стан грошових потоків. До малого бізнесу можуть належати торговельні точки, сервісні компанії, кав'ярні, майстерні, інтернет-магазини, невеликі виробничі підрозділи та локальні підприємства сфери послуг. Для таких суб'єктів господарювання важливим є не лише ведення обліку операцій, а й постійний аналіз ефективності управління.

Ефективність управління малим бізнесом визначається здатністю керівника або менеджера своєчасно отримувати достовірні показники діяльності, оцінювати фінансовий стан, контролювати витрати, аналізувати продажі, виявляти слабкі місця та приймати обґрунтовані управлінські рішення. У практичній діяльності невеликих підприємств часто використовуються окремі таблиці, ручні записи, банківські виписки, касові звіти або дані з бухгалтерських програм. Такий підхід ускладнює узагальнення інформації та не забезпечує швидкого формування аналітичної картини.

Предметна область програмного застосунку охоплює процеси накопичення операційних даних, розрахунку ключових показників ефективності, побудови аналітичної панелі, формування рекомендацій і підготовки звітів. Основними інформаційними об'єктами є користувачі, фінансові операції, продажі, витрати, клієнти, товари або послуги, періоди аналізу, показники KPI, рекомендації та звіти. У межах роботи розглядається програмний застосунок, який допомагає власнику або менеджеру малого бізнесу оцінити результативність управління підприємством на основі зібраних даних.

У загальному вигляді процес аналізу ефективності управління складається з кількох етапів. Спочатку користувач авторизується в системі, після чого вводить або імпортує облікові дані. Далі система узгоджує дані, зберігає їх у базі, розраховує показники, формує аналітичну панель і готує рекомендації. Якщо потрібно, користувач може сформувати звіт за вибраний період. Така логіка відповідає практичним потребам малого бізнесу, де важливо швидко оцінити стан справ без складних бухгалтерських процедур.

Таблиця 1.1

Основні показники для аналізу ефективності управління малим бізнесом

Показник	Призначення	Управлінське значення
Дохід	Відображає суму отриманих надходжень за період	Дає змогу оцінити динаміку продажів і загальний масштаб діяльності
Витрати	Показують суму операційних, закупівельних, адміністративних та інших витрат	Дозволяють контролювати перевитрати та визначати напрями оптимізації
Прибуток	Розраховується як різниця між доходами й витратами	Характеризує фінансовий результат діяльності
Рентабельність	Визначає співвідношення прибутку до доходу або витрат	Показує ефективність використання ресурсів
Середній чек	Характеризує середній обсяг однієї покупки або замовлення	Допомагає оцінити поведінку клієнтів і ефективність продажів
Кількість клієнтів	Відображає обсяг клієнтської бази	Дозволяє оцінити стабільність попиту
КРІ управління	Узагальнений набір ключових показників	Використовується для прийняття управлінських рішень

Наведені показники є базою для побудови аналітичної частини програмного застосунку. Вони дозволяють не лише відобразити поточний стан бізнесу, а й виявити проблемні зони: зростання витрат, падіння прибутковості, зменшення кількості клієнтів або недостатню ефективність продажів. Саме тому програмний застосунок має поєднувати функції введення даних, імпорту, розрахунку, візуалізації та звітності.

1.2 Аналіз існуючих програмних рішень

На ринку програмного забезпечення існує значна кількість систем, які можуть використовуватися малим бізнесом для обліку, аналітики, контролю продажів і фінансового управління. Найбільш поширеними є бухгалтерські сервіси, CRM-системи, ERP-платформи та аналітичні панелі. Для порівняння було розглянуто рішення QuickBooks, Xero, Zoho Books, Odoo та HubSpot CRM.

QuickBooks орієнтований на бухгалтерський облік малого й середнього бізнесу, роботу з доходами, витратами, рахунками та фінансовими звітами. Xero забезпечує хмарну бухгалтерію, контроль cash flow, рахунків, оплат і фінансових показників. Zoho Books поєднує облік фінансів, рахунків, витрат, податкових зобов'язань і грошових потоків. Odoo є модульною ERP/CRM-платформою, яка охоплює продажі, клієнтів, склад, бухгалтерію та проєктне управління. HubSpot CRM використовується для контролю продажів, угод, клієнтів, воронки продажів і звітності.

Таблиця 1.2

Порівняльна характеристика існуючих програмних рішень

Рішення	Основне призначення	Переваги	Обмеження для теми роботи
QuickBooks	Фінансовий і бухгалтерський облік малого бізнесу	Зручна фінансова панель, облік витрат, рахунків, звіти, cash flow	Орієнтація переважно на бухгалтерський облік і залежність від

			країни використання
--	--	--	---------------------

Продовження таблиці 1.2

Xero	Хмарна бухгалтерія та фінансове управління	Контроль рахунків, оплат, банківських операцій і грошових потоків	Не всі функції адаптовані для локальних особливостей малого бізнесу
Zoho Books	Облік доходів, витрат, рахунків, платежів і звітів	Зручна панель фінансового стану, мобільна робота, автоматизація операцій	Функціонал може бути надлишковим для навчального програмного прототипу
Odoo	ERP/CRM-платформа для комплексного управління бізнесом	Модульність, CRM, продажі, склад, облік, інтеграції	Складність налаштування та значний обсяг функцій
HubSpot CRM	Управління клієнтами, продажами та аналітикою взаємодії	Зручна воронка продажів, панелі звітності, автоматизація CRM-процесів	Фокус на продажах і клієнтах, а не на повному аналізі ефективності управління

За результатами порівняння можна зробити висновок, що існуючі програмні продукти мають потужний функціонал, однак часто орієнтовані або на бухгалтерський облік, або на CRM, або на комплексну ERP-автоматизацію. Для невеликого бізнесу й навчального програмного прототипу доцільним є створення простішого застосунку, який зосереджується саме на аналізі ефективності управління: розрахунку KPI, візуалізації даних, формуванні рекомендацій і підготовці звітів.

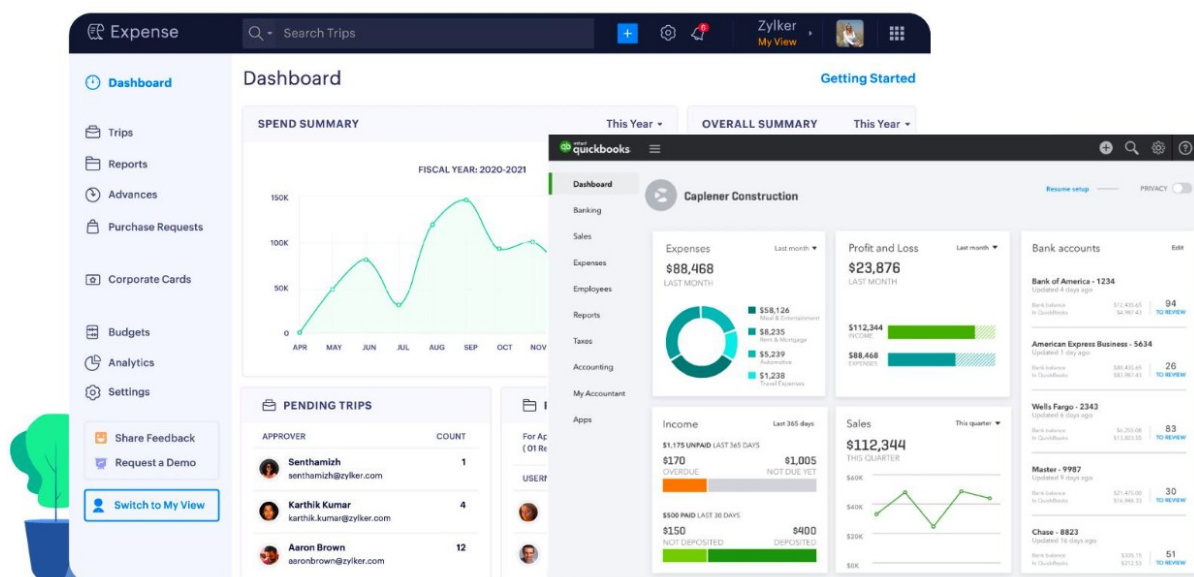


Рис. 1.1. Приклад аналітичної панелі для фінансового контролю малого бізнесу

На рис. 1.1 показано приклад інтерфейсу фінансово-аналітичної системи, у якій відображаються витрати, прибуток, банківські рахунки, продажі та інші показники. Такий підхід є корисним для формування вимог до власного програмного застосунку, оскільки користувач має бачити ключові показники одразу після входу до системи.

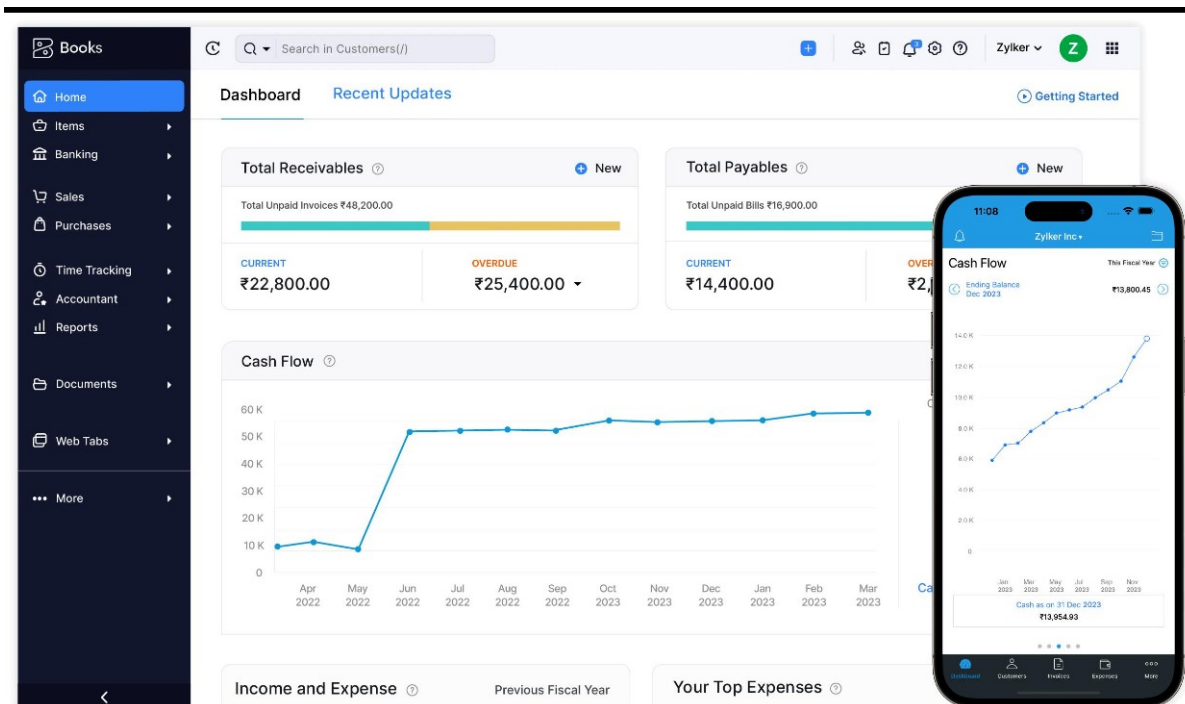


Рис. 1.2. Приклад інтерфейсу Zoho Books для аналізу грошових потоків і заборгованостей

На рис. 1.2 наведено приклад панелі Zoho Books, яка демонструє фінансові показники, грошові потоки, дебіторську та кредиторську заборгованість. Для розроблюваного застосунку важливим є не повне копіювання бухгалтерського функціоналу, а використання ідеї компактної аналітичної панелі з ключовими індикаторами.

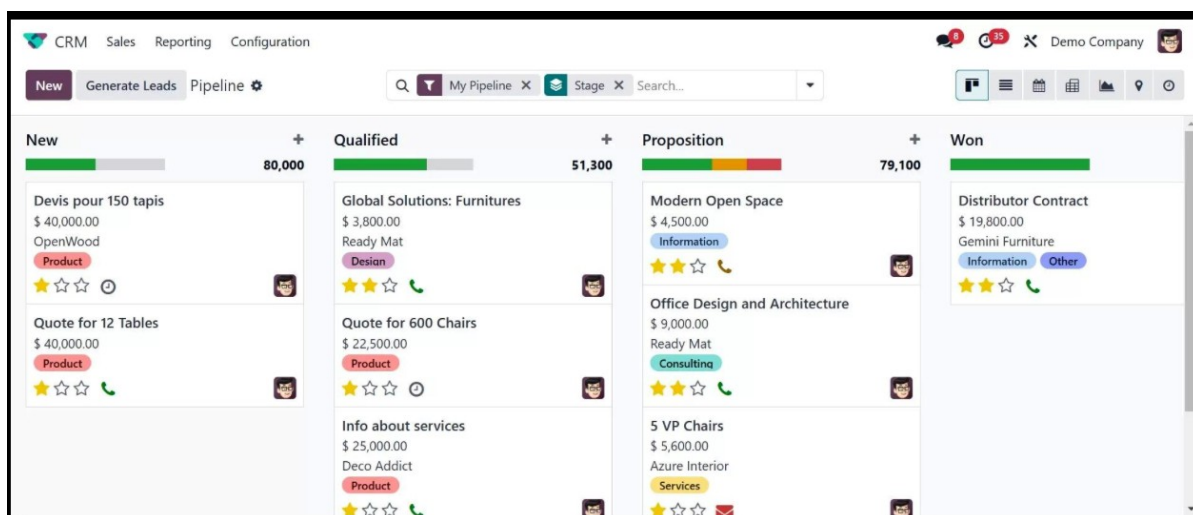


Рис. 1.3. Приклад інтерфейсу Odoo CRM для аналізу продажів і воронки угод

На рис. 1.3 показано CRM-підхід до аналізу продажів через стадії угод. Для малого бізнесу така логіка може бути використана для оцінювання результативності роботи з клієнтами, контролю етапів продажу та визначення очікуваного доходу.

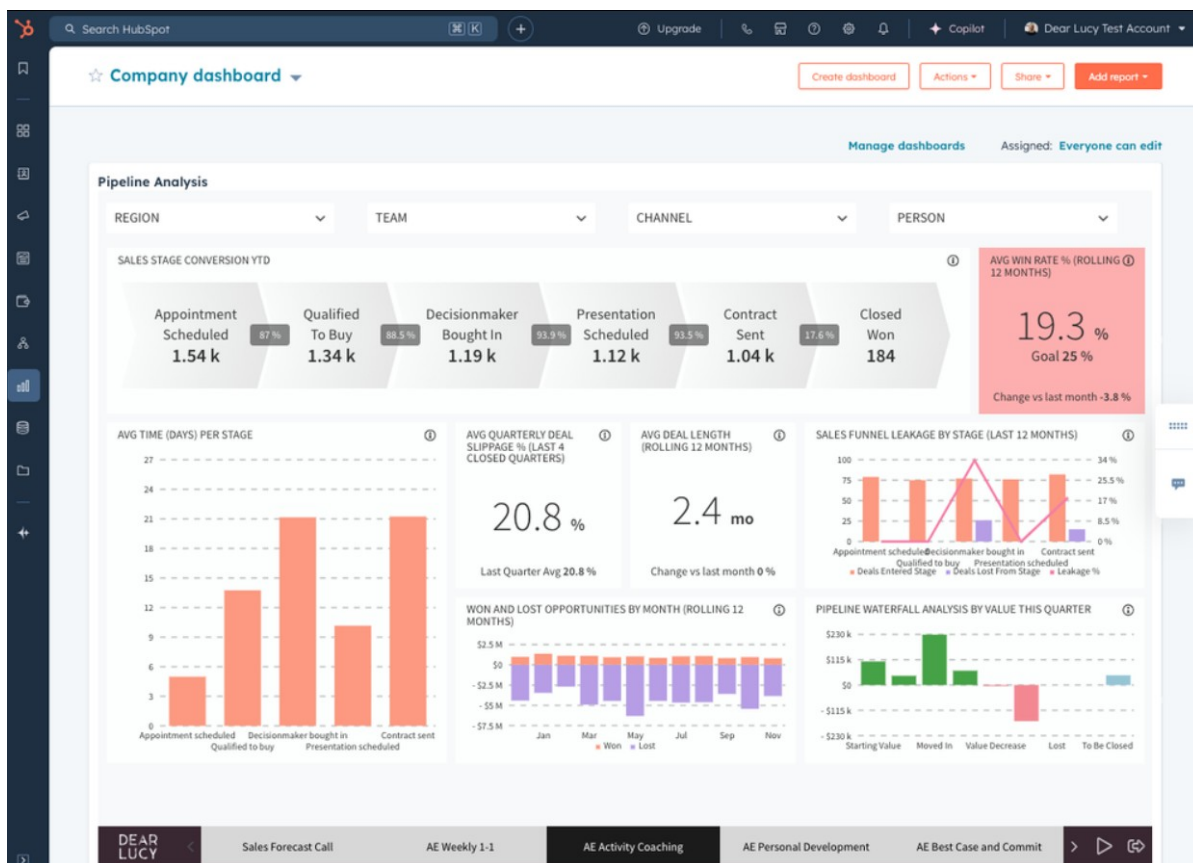


Рис. 1.4. Приклад аналітичної панелі HubSpot CRM для контролю продажів

На рис. 1.4 наведено приклад аналітичної панелі HubSpot CRM, де відображаються показники продажів, конверсії, динаміка угод і результативність команди. Для розроблюваного застосунку доцільно передбачити подібний принцип візуалізації, але з акцентом на показники управління малим бізнесом.

1.3 Моделювання предметної області

Для уточнення функціональної структури програмного застосунку виконано моделювання предметної області з використанням UML-діаграм. Моделювання дозволяє визначити основних акторів системи, їхню взаємодію з програмним забезпеченням, послідовність виконання операцій і загальний сценарій роботи користувача.

Основними акторами є власник або керівник малого бізнесу, менеджер і зовнішня облікова система. Власник або керівник використовує систему для перегляду аналітичної панелі, контролю KPI, отримання рекомендацій і формування звітів. Менеджер відповідає за введення та оновлення операційних

даних, перегляд показників і підготовку управлінської інформації. Зовнішня облікова система є джерелом імпортованих даних, які можуть використовуватися для аналізу.

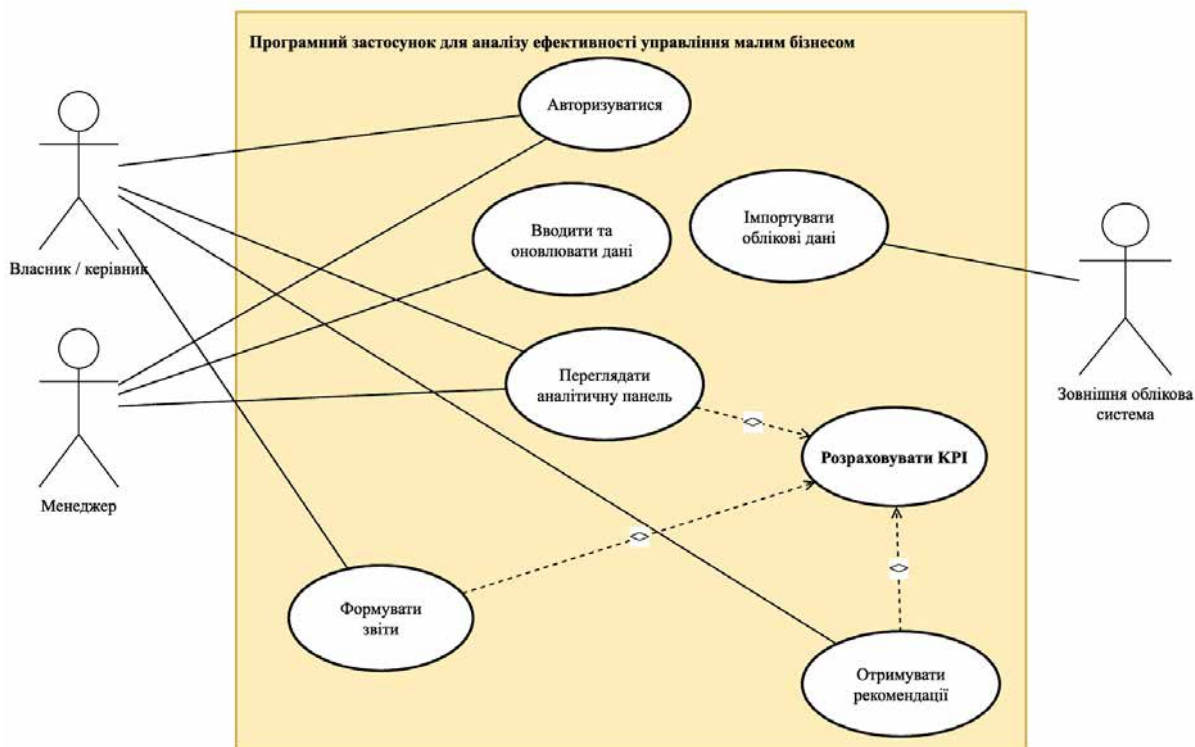


Рис. 1.5. Діаграма прецедентів програмного застосунку для аналізу ефективності управління малим бізнесом

На діаграмі прецедентів, зображеній на рис. 1.5, відображено основні функціональні можливості системи: авторизація, введення та оновлення даних, імпорт облікових даних, перегляд аналітичної панелі, розрахунок KPI, отримання рекомендацій і формування звітів. Центральним процесом є розрахунок KPI, оскільки саме він забезпечує перехід від первинних облікових даних до управлінських висновків.

Діаграма послідовності використовується для деталізації взаємодії між користувачами, програмним застосунком, базою даних і зовнішньою обліковою системою. Вона демонструє, як відбувається авторизація, введення або імпорт даних, збереження інформації, отримання консолідованих показників, розрахунок KPI, формування рекомендацій і підготовка звіту.

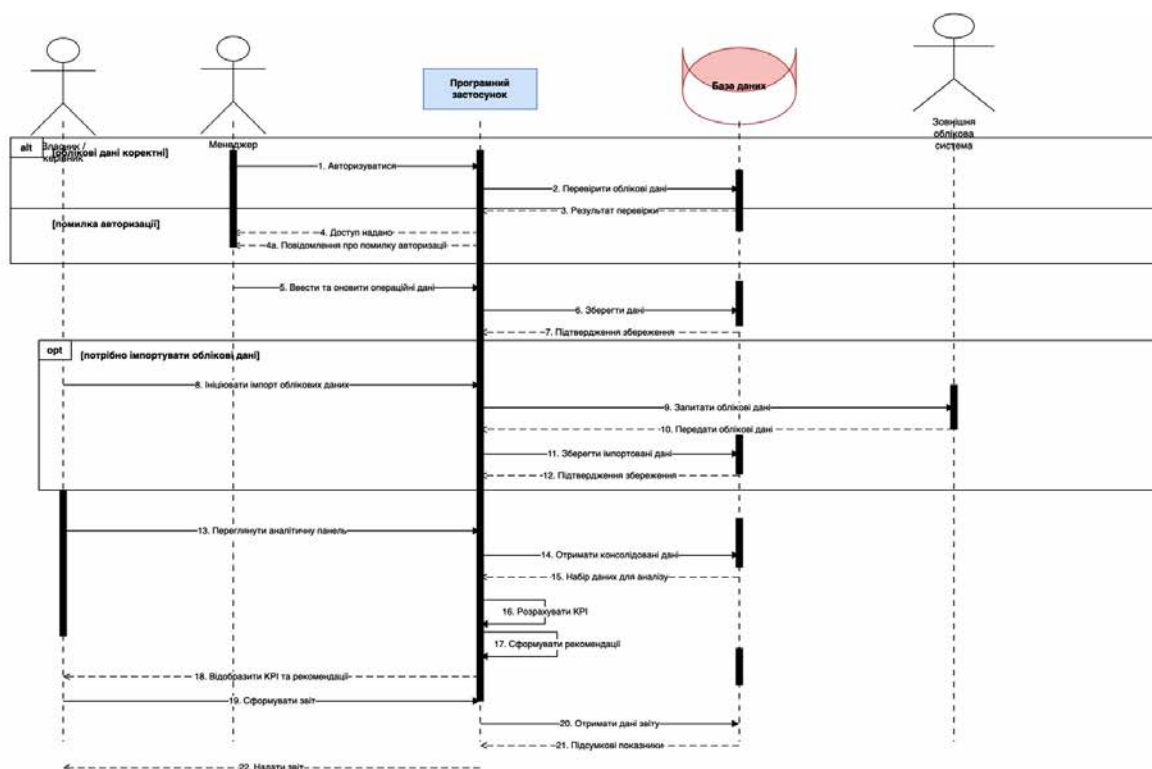


Рис. 1.6. Діаграма послідовності процесу аналізу ефективності управління малим бізнесом

Як видно з рис. 1.6, взаємодія починається з авторизації користувача. Якщо облікові дані коректні, система надає доступ до функцій. Далі користувач може вручну ввести операційні дані або ініціювати імпорт із зовнішньої облікової системи. Після збереження даних програма формує набір показників для аналізу, розраховує KPI, генерує рекомендації та повертає користувачу звіт.

Діаграма активності дає змогу описати загальний бізнес-процес роботи із системою. Вона містить окремі області відповідальності користувача, програмного застосунку, бази даних і зовнішньої облікової системи. Такий поділ дозволяє наочно показати, які дії виконує користувач, а які автоматично виконує програмний застосунок.

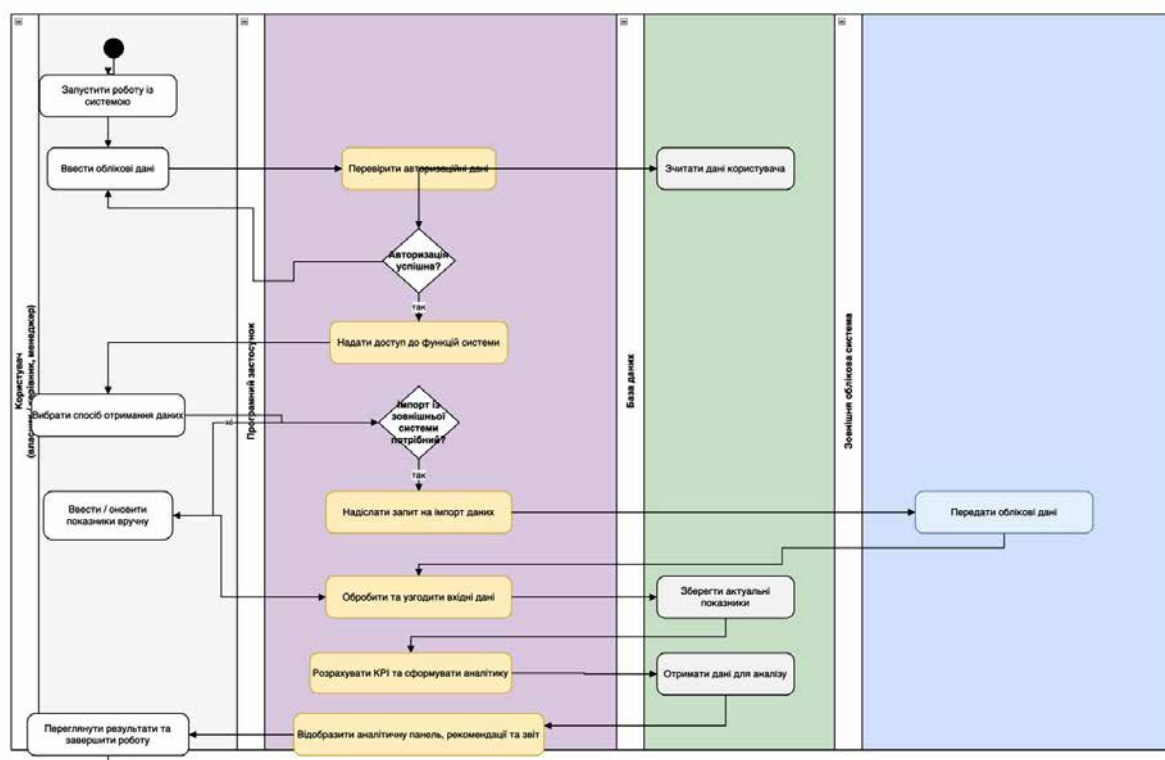


Рис. 1.7. Діаграма активності процесу роботи з програмним застосунком

На рис. 1.7 показано, що робота із системою починається із запуску застосунку та введення облікових даних. Після успішної авторизації користувач вибирає спосіб отримання даних: ручне введення або імпорт із зовнішньої системи. Далі інформація обробляється, узгоджується, зберігається в базі даних і використовується для розрахунку KPI. На завершальному етапі система відображає аналітичну панель, рекомендації та звіт.

Таблиця 1.3

Основні актори та їх функції в системі

Актор	Функції у системі	Очікуваний результат
Власник / керівник	Перегляд аналітичної панелі, KPI, рекомендацій і звітів	Отримання інформації для управлінських рішень
Менеджер	Введення та оновлення операційних даних, перегляд показників	Підготовка актуальних даних для аналізу
Зовнішня облікова система	Передавання облікових даних для імпорту	Автоматизоване поповнення бази даних

Отже, виконане моделювання підтверджує необхідність розроблення застосунку з модульною структурою, у якій окремо реалізуються авторизація, введення даних, імпорт, база даних, аналітика, рекомендації та звітність.

1.4 Аналіз вимог до програмного застосунку

На основі аналізу предметної області, існуючих рішень і побудованих моделей сформовано вимоги до програмного застосунку. Вимоги поділено на функціональні, нефункціональні, інформаційні, інтерфейсні та вимоги безпеки. Такий поділ дозволяє комплексно описати очікувану поведінку системи та критерії її подальшої реалізації.

Функціональні вимоги визначають перелік дій, які має підтримувати програмний застосунок. Вони безпосередньо пов'язані з бізнес-процесами малого підприємства: авторизацією користувача, введенням даних, імпортом, розрахунком КРІ, формуванням аналітичної панелі, рекомендацій і звітів.

Таблиця 1.4

Функціональні вимоги до програмного застосунку

№	Вимога	Опис
1	Авторизація користувача	Перевірка логіна та пароля перед наданням доступу до функцій системи
2	Введення операційних даних	Додавання та оновлення інформації про доходи, витрати, продажі, клієнтів і періоди аналізу
3	Імпорт облікових даних	Отримання даних із зовнішньої облікової системи або файлу
4	Розрахунок КРІ	Обчислення показників доходу, витрат, прибутку, рентабельності та динаміки
5	Аналітична панель	Відображення ключових показників у вигляді карток, графіків і таблиць
6	Формування рекомендацій	Створення коротких управлінських підказок на основі виявлених проблем

Нефункціональні вимоги визначають якісні характеристики системи. Для малого бізнесу важливими є простота інтерфейсу, швидкість роботи, надійність збереження даних, можливість локального використання та зрозумілість результатів аналізу.

Таблиця 1.5

Нефункціональні вимоги до системи

Група вимог	Зміст вимоги
Продуктивність	Основні операції перегляду, збереження та розрахунку показників мають виконуватися без помітних затримок для користувача
Надійність	Дані мають зберігатися в базі даних із мінімізацією ризику втрати при помилкових діях користувача
Зручність використання	Інтерфейс має бути простим, зрозумілим і придатним для користувачів без спеціальної технічної підготовки
Масштабованість	Структура системи має дозволяти додавання нових показників, звітів і модулів аналізу
Безпека	Доступ до системи має надаватися лише авторизованим користувачам
Супровід	Програмний код і структура бази даних мають бути зрозумілими для подальшого редагування

Інформаційні вимоги визначають склад даних, які обробляються в системі. До таких даних належать облікові записи користувачів, періоди аналізу, доходи, витрати, продажі, клієнти, KPI, рекомендації та сформовані звіти. Усі ці дані мають бути структуровані так, щоб система могла виконувати вибірку за періодом, розраховувати показники та формувати аналітичні висновки.

1.5 Постановка завдання

На основі проведеного аналізу предметної області та існуючих рішень сформульовано завдання розроблення програмного застосунку для аналізу

ефективності управління малим бізнесом. Розроблюваний застосунок має забезпечити введення, збереження, оброблення й аналітичне подання даних, необхідних для оцінювання результативності діяльності малого підприємства.

Метою розроблення є створення програмного застосунку, який дає змогу власнику або менеджеру малого бізнесу оперативно аналізувати фінансові та операційні показники, визначати проблемні напрями діяльності, отримувати узагальнені KPI, формувати рекомендації та готувати звіти для прийняття управлінських рішень.

Для досягнення поставленої мети необхідно виконати такі завдання:

- проаналізувати предметну область управління малим бізнесом і визначити ключові показники ефективності;

- дослідити існуючі програмні рішення для фінансового аналізу, CRM та управління підприємством;

- визначити функціональні та нефункціональні вимоги до програмного застосунку;

- побудувати UML-моделі основних процесів роботи системи;

- спроєктувати структуру даних для збереження облікової й аналітичної інформації;

- реалізувати модулі авторизації, введення даних, імпорту, аналітики, рекомендацій та звітності;

- забезпечити відображення результатів аналізу у вигляді зручної аналітичної панелі;

- провести тестування програмного застосунку та оцінити коректність роботи основних функцій.

Вхідними даними системи є відомості про доходи, витрати, продажі, клієнтів, періоди аналізу, категорії витрат, імпортовані облікові дані та параметри, задані користувачем. Вихідними даними є розраховані KPI, аналітичні графіки, текстові рекомендації, таблиці порівняння показників і сформовані звіти.

У межах кваліфікаційної роботи необхідно розробити програмний прототип, який відображає основні можливості майбутньої системи. Прототип має бути достатнім для демонстрації логіки оброблення даних, розрахунку показників, формування аналітичної панелі та підготовки звітної інформації. Особливу увагу слід приділити простоті інтерфейсу, оскільки користувачами системи можуть бути власники малого бізнесу та менеджери без спеціальної технічної підготовки.

Таблиця 1.6

Основні результати, які має забезпечити розроблюваний застосунок

Результат	Зміст
Аналітична панель	Відображення ключових показників діяльності малого бізнесу
Розрахунок KPI	Автоматичне обчислення фінансових та операційних показників
Рекомендації	Формування управлінських підказок за результатами аналізу
Звіти	Підготовка структурованого звіту за вибраний період
Імпорт даних	Можливість отримання облікових даних із зовнішньої системи або файлу
База даних	Збереження користувачів, операцій, показників і звітів

Розроблений програмний застосунок має бути логічно узгодженим із побудованими моделями предметної області. Це означає, що всі функції, показані на діаграмах прецедентів, послідовності та активності, повинні бути враховані під час подальшого проектування й реалізації системи.

1.6 Висновки до першого розділу

У першому розділі було проведено аналіз предметної області програмного застосунку для аналізу ефективності управління малим бізнесом. Визначено, що

для малого підприємства важливим є не лише ведення обліку доходів і витрат, а й оперативне отримання аналітичних показників, які дають змогу оцінювати якість управління, прибутковість, витрати, продажі, клієнтську активність і загальну результативність діяльності.

Розглянуто існуючі програмні рішення QuickBooks, Xero, Zoho Books, Odoo та HubSpot CRM. Встановлено, що вони мають широкий функціонал для бухгалтерії, CRM, фінансового аналізу та управління бізнес-процесами. Водночас такі системи можуть бути надлишковими, складними або недостатньо адаптованими для навчального програмного прототипу, орієнтованого саме на аналіз ефективності управління малим бізнесом.

Виконано моделювання предметної області за допомогою UML-діаграм. Діаграма прецедентів дозволила визначити основні функції системи та акторів. Діаграма послідовності деталізувала процес авторизації, введення, імпорту, аналізу даних і формування звіту. Діаграма активності відобразила загальний сценарій роботи користувача із системою та взаємодію із базою даних і зовнішньою обліковою системою.

На основі аналізу сформовано функціональні, нефункціональні та інформаційні вимоги до програмного застосунку. Визначено, що система має підтримувати авторизацію, введення й оновлення даних, імпорт облікової інформації, розрахунок KPI, формування аналітичної панелі, підготовку рекомендацій і звітів. Також встановлено вимоги до простоти інтерфейсу, надійності збереження даних, швидкодії, безпеки та можливості подальшого розширення.

У результаті було сформульовано постановку завдання розроблення програмного застосунку. Подальші розділи роботи мають бути спрямовані на проєктування структури даних, розроблення архітектури програмного забезпечення, реалізацію основних модулів і тестування працездатності системи. Отримані в першому розділі результати є основою для наступного етапу розроблення програмного застосунку.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Логічна модель даних програмного застосунку

Проєктування інформаційного забезпечення програмного застосунку для аналізу ефективності управління малим бізнесом передбачає визначення основних сутностей предметної області, їхніх атрибутів, зв'язків і правил взаємодії. Оскільки система орієнтована на аналіз операційної, фінансової та управлінської діяльності малого підприємства, база даних має забезпечувати збереження відомостей про користувачів, бізнес-одиниці, фінансові операції, категорії доходів і витрат, ключові показники ефективності, результати розрахунків, рекомендації та сформовані звіти.

Основною інформаційною сутністю системи є підприємство малого бізнесу, для якого здійснюється облік і аналіз показників. Підприємство може мати декількох користувачів із різними ролями: власник, керівник, менеджер або працівник. Для цього в логічній моделі передбачено зв'язувальну сутність `User_Business`, яка дає змогу реалізувати належність одного користувача до одного або кількох бізнес-проєктів.

Логічну модель даних програмного застосунку наведено на рис. 2.1.

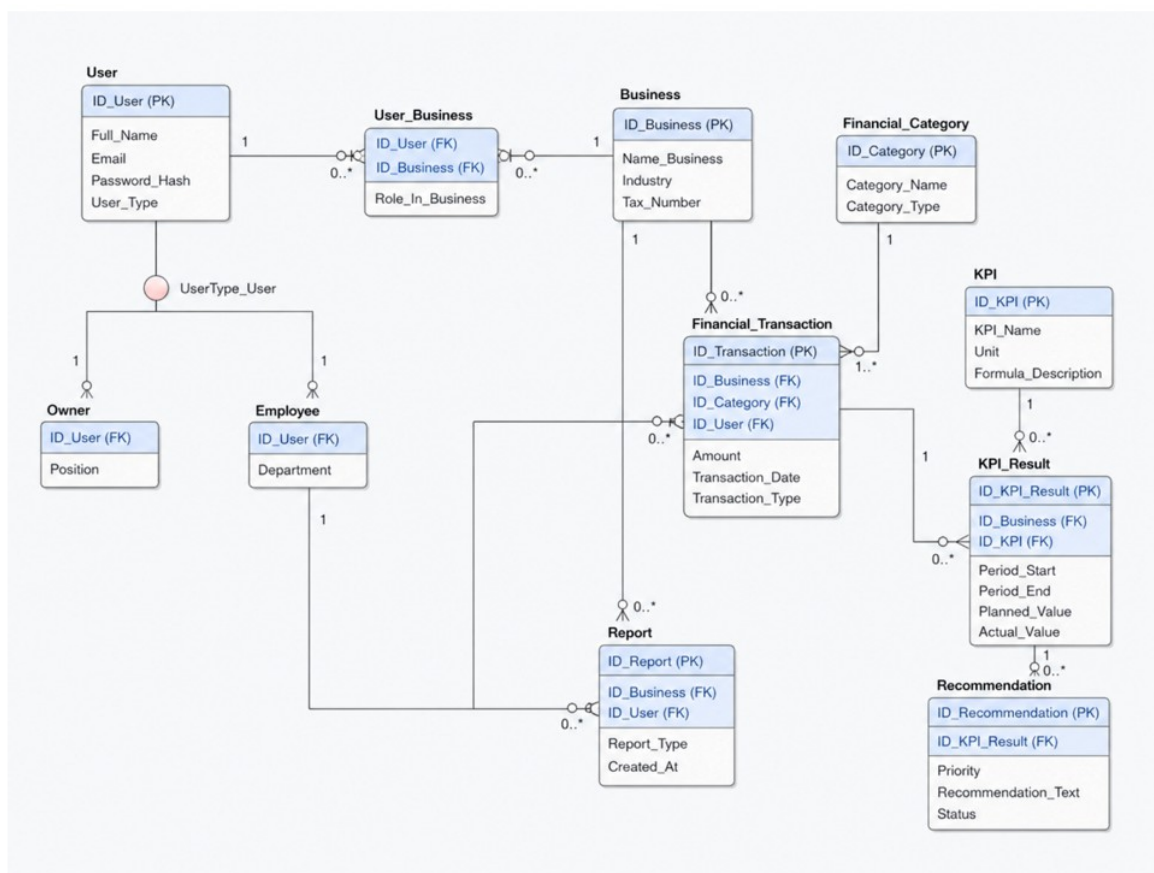


Рис. 2.1. Логічна модель даних програмного застосунку для аналізу ефективності управління малим бізнесом

У моделі визначено такі основні сутності: User, Business, User_Business, Financial_Category, Financial_Transaction, KPI, KPI_Result, Recommendation та Report. Сутність User містить дані користувача, його електронну пошту, хеш пароля та тип користувача. Сутність Business описує підприємство, для якого ведеться аналіз. Сутність Financial_Transaction зберігає інформацію про доходи, витрати та інші фінансові операції. Сутність KPI використовується для опису ключових показників ефективності, а KPI_Result - для збереження результатів їх розрахунку за певний період.

Таблиця 2.1

Основні сутності логічної моделі даних

Сутність	Призначення
User	Зберігання даних користувачів системи
Business	Зберігання відомостей про підприємство малого бізнесу

Продовження таблиці 2.1

User_Business	Встановлення зв'язку між користувачами та підприємствами
Financial_Category	Класифікація доходів і витрат за категоріями
Financial_Transaction	Облік фінансових операцій підприємства
KPI	Опис ключових показників ефективності
KPI_Result	Збереження розрахованих значень KPI за період
Recommendation	Формування управлінських рекомендацій
Report	Збереження звітів, сформованих користувачем

Зв'язки між сутностями побудовані таким чином, щоб забезпечити цілісність даних і можливість аналітичної обробки. Один бізнес може мати багато фінансових операцій, звітів і результатів KPI. Одна категорія може використовуватися у багатьох фінансових операціях. Один KPI може мати багато результатів розрахунку за різні періоди, а один результат KPI може бути підставою для формування кількох рекомендацій.

Таблиця 2.2

Зв'язки між сутностями логічної моделі

Зв'язок	Тип зв'язку	Зміст зв'язку
User – User_Business	1:N	Один користувач може бути пов'язаний із кількома підприємствами
Business – User_Business	1:N	Одне підприємство може мати кількох користувачів
Business – Financial_Transaction	1:N	Для одного підприємства може бути багато фінансових операцій
Financial_Category – Financial_Transaction	1:N	Одна категорія може використовуватися у багатьох операціях
Business – Report	1:N	Для одного підприємства може бути сформовано багато звітів
Business –	1:N	Для одного підприємства може бути багато

KPI_Result		результатів KPI
------------	--	-----------------

Продовження таблиці 2.2

KPI – KPI_Result	1:N	Один показник KPI може розраховуватися багато разів
KPI_Result – Recommendation	1:N	Один результат KPI може мати кілька рекомендацій

Запропонована логічна модель відповідає вимогам предметної області та забезпечує можливість подальшого формування аналітичної панелі, обліку фінансових операцій, розрахунку показників ефективності, створення управлінських рекомендацій і підготовки звітів.

2.2 Фізична модель бази даних

На основі логічної моделі даних розроблено фізичну модель бази даних програмного застосунку. Фізична модель визначає конкретні таблиці, поля, типи даних, первинні та зовнішні ключі, а також зв'язки між таблицями. Вона є основою для реалізації інформаційної бази системи у середовищі реляційної СУБД.

Фізичну модель бази даних наведено на рис. 2.2.

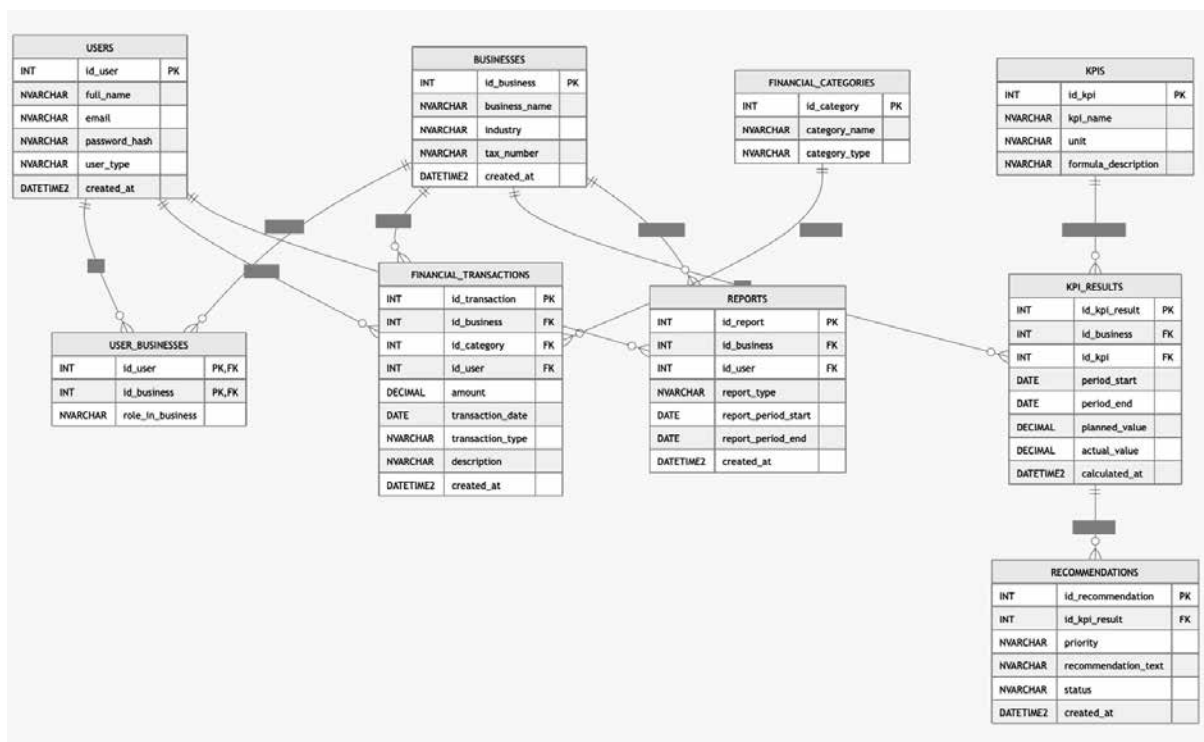


Рис. 2.2. Фізична модель бази даних програмного застосунку

До складу фізичної моделі входять таблиці users, businesses, user_businesses, financial_categories, financial_transactions, kpis, kpi_results, recommendations та reports. Для збереження текстових полів використано тип NVARCHAR, для грошових і числових значень - DECIMAL, для дат - DATE та DATETIME2. Такий підхід забезпечує коректне зберігання фінансової інформації, часових періодів аналізу та службових даних.

Таблиця 2.3

Таблиці фізичної моделі бази даних

Таблиця	Призначення
users	Облікові записи користувачів
businesses	Дані підприємств малого бізнесу
user_businesses	Зв'язок користувачів із підприємствами
financial_categories	Довідник категорій доходів і витрат
financial_transactions	Фінансові операції підприємства
kpis	Довідник ключових показників ефективності
kpi_results	Розраховані значення KPI
recommendations	Управлінські рекомендації

reports	Сформовані звіти

Фізична модель передбачає використання первинних ключів для кожної таблиці. Зовнішні ключі забезпечують цілісність даних між підприємствами, користувачами, фінансовими операціями, KPI та рекомендаціями. Наприклад, таблиця `financial_transactions` має зовнішні ключі `id_business`, `id_category` та `id_user`, що дозволяє визначити, до якого підприємства, категорії та користувача належить кожна операція.

Для реалізації бази даних може бути використаний такий SQL-код:

```
CREATE TABLE users (
    id_user INT PRIMARY KEY IDENTITY(1,1),
    full_name NVARCHAR(100) NOT NULL,
    email NVARCHAR(100) NOT NULL UNIQUE,
    password_hash NVARCHAR(255) NOT NULL,
    user_type NVARCHAR(30) NOT NULL,
    created_at DATETIME2 NOT NULL DEFAULT SYSDATETIME()
);

CREATE TABLE businesses (
    id_business INT PRIMARY KEY IDENTITY(1,1),
    business_name NVARCHAR(150) NOT NULL,
    industry NVARCHAR(100) NOT NULL,
    tax_number NVARCHAR(30),
    created_at DATETIME2 NOT NULL DEFAULT SYSDATETIME()
);

CREATE TABLE user_businesses (
    id_user INT NOT NULL,
    id_business INT NOT NULL,
    role_in_business NVARCHAR(50) NOT NULL,
    PRIMARY KEY (id_user, id_business),
    FOREIGN KEY (id_user) REFERENCES users(id_user),
    FOREIGN KEY (id_business) REFERENCES businesses(id_business)
);

CREATE TABLE financial_categories (
    id_category INT PRIMARY KEY IDENTITY(1,1),
    category_name NVARCHAR(100) NOT NULL,
    category_type NVARCHAR(30) NOT NULL
);

CREATE TABLE financial_transactions (
```

```

    id_transaction INT PRIMARY KEY IDENTITY(1,1),
    id_business INT NOT NULL,
    id_category INT NOT NULL,
    id_user INT NOT NULL,
    amount DECIMAL(12,2) NOT NULL,
    transaction_date DATE NOT NULL,
    transaction_type NVARCHAR(30) NOT NULL,
    description NVARCHAR(255),
    created_at DATETIME2 NOT NULL DEFAULT SYSDATETIME(),
    FOREIGN KEY (id_business) REFERENCES businesses(id_business),
    FOREIGN KEY (id_category) REFERENCES financial_categories(id_category),
    FOREIGN KEY (id_user) REFERENCES users(id_user)
);

CREATE TABLE kpis (
    id_kpi INT PRIMARY KEY IDENTITY(1,1),
    kpi_name NVARCHAR(100) NOT NULL,
    unit NVARCHAR(30) NOT NULL,
    formula_description NVARCHAR(255) NOT NULL
);

CREATE TABLE kpi_results (
    id_kpi_result INT PRIMARY KEY IDENTITY(1,1),
    id_business INT NOT NULL,
    id_kpi INT NOT NULL,
    period_start DATE NOT NULL,
    period_end DATE NOT NULL,
    planned_value DECIMAL(12,2),
    actual_value DECIMAL(12,2) NOT NULL,
    calculated_at DATETIME2 NOT NULL DEFAULT SYSDATETIME(),
    FOREIGN KEY (id_business) REFERENCES businesses(id_business),
    FOREIGN KEY (id_kpi) REFERENCES kpis(id_kpi)
);

CREATE TABLE recommendations (
    id_recommendation INT PRIMARY KEY IDENTITY(1,1),
    id_kpi_result INT NOT NULL,
    priority NVARCHAR(30) NOT NULL,
    recommendation_text NVARCHAR(500) NOT NULL,
    status NVARCHAR(30) NOT NULL DEFAULT 'new',
    created_at DATETIME2 NOT NULL DEFAULT SYSDATETIME(),
    FOREIGN KEY (id_kpi_result) REFERENCES kpi_results(id_kpi_result)
);

CREATE TABLE reports (
    id_report INT PRIMARY KEY IDENTITY(1,1),
    id_business INT NOT NULL,
    id_user INT NOT NULL,

```

```

report_type NVARCHAR(50) NOT NULL,
report_period_start DATE NOT NULL,
report_period_end DATE NOT NULL,
created_at DATETIME2 NOT NULL DEFAULT SYSDATETIME(),
FOREIGN KEY (id_business) REFERENCES businesses(id_business),
FOREIGN KEY (id_user) REFERENCES users(id_user)
);

```

Для підвищення швидкості вибірки даних у таблицях доцільно створити індекси за полями, які найчастіше використовуються під час пошуку, фільтрації та формування звітів. Зокрема, це поля `id_business`, `transaction_date`, `id_category`, `period_start`, `period_end` та `created_at`.

```

CREATE INDEX idx_transactions_business
ON financial_transactions(id_business);

CREATE INDEX idx_transactions_date
ON financial_transactions(transaction_date);

CREATE INDEX idx_transactions_category
ON financial_transactions(id_category);

CREATE INDEX idx_kpi_results_business
ON kpi_results(id_business);

CREATE INDEX idx_kpi_results_period
ON kpi_results(period_start, period_end);

CREATE INDEX idx_reports_business
ON reports(id_business);

CREATE INDEX idx_reports_created_at
ON reports(created_at);

```

Також доцільно передбачити внесення умовно-постійних даних, які використовуються під час першого запуску системи. До них належать типові категорії фінансових операцій і перелік базових KPI.

```

INSERT INTO financial_categories (category_name, category_type)
VALUES
('Продаж товарів', 'income'),
('Надання послуг', 'income'),
('Оренда приміщення', 'expense'),
('Заробітна плата', 'expense'),
('Маркетинг', 'expense'),
('Закупівля матеріалів', 'expense'),
('Податки', 'expense');

INSERT INTO kpis (kpi_name, unit, formula_description)
VALUES
('Дохід', 'грн', 'Сума всіх фінансових операцій типу income за період'),
('Витрати', 'грн', 'Сума всіх фінансових операцій типу expense за період'),
('Прибуток', 'грн', 'Дохід мінус витрати за період'),
('Рентабельність', '%', 'Прибуток / Дохід * 100'),
('Виконання плану', '%', 'Фактичне значення / Планове значення * 100'),
('Середній чек', 'грн', 'Дохід / Кількість продажів');

```

Розроблена фізична модель є достатньою для реалізації основних функцій програмного застосунку. Вона забезпечує збереження облікових даних, фінансових операцій, результатів KPI, рекомендацій і звітів, а також підтримує подальше розширення системи.

2.3 Діаграма пакетів програмного забезпечення

Пакетна структура програмного забезпечення відображає логічний поділ системи на окремі функціональні частини. Такий поділ спрощує розробку, супровід, тестування та подальше розширення застосунку. У межах програмного застосунку для аналізу ефективності управління малим бізнесом виділено пакети Presentation, Application, Domain Model, Data Access, Security, Integration, Analytics та Reporting.

Діаграму пакетів наведено на рис. 2.3.

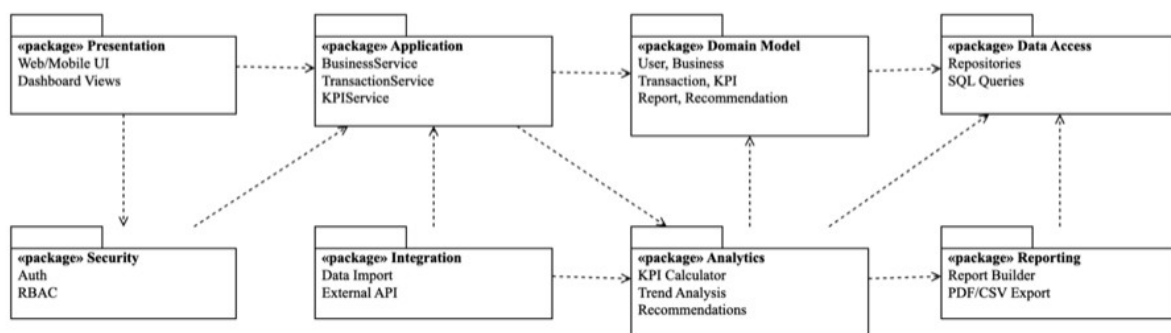


Рис. 2.3. Діаграма пакетів програмного застосунку

Пакет Presentation відповідає за інтерфейс користувача, веб- або мобільні екрани, інформаційну панель та сторінки введення даних. Пакет Application містить прикладні сервіси, які координують бізнес-процеси системи: оброблення операцій, запуск розрахунку KPI, формування рекомендацій і передавання даних між модулями. Пакет Domain Model описує основні об'єкти предметної області: користувача, підприємство, фінансову операцію, KPI, звіт і рекомендацію.

Пакет Data Access забезпечує доступ до бази даних, виконання SQL-запитів і роботу з репозиторіями. Пакет Security відповідає за автентифікацію, авторизацію та розмежування прав доступу. Пакет Integration використовується для імпорту облікових даних із зовнішніх систем. Пакет Analytics містить засоби розрахунку KPI, аналізу тенденцій і формування рекомендацій. Пакет Reporting забезпечує побудову звітів та експорт результатів у PDF або CSV.

Таблиця 2.4

Призначення пакетів програмного забезпечення

Пакет	Призначення
Presentation	Відображення інтерфейсу користувача та аналітичної панелі
Application	Координація прикладної логіки системи
Domain Model	Опис основних сутностей предметної області
Data Access	Доступ до бази даних і виконання SQL-запитів
Security	Автентифікація, авторизація та рольове розмежування
Integration	Імпорт даних із зовнішніх облікових систем
Analytics	Розрахунок KPI, аналіз тенденцій і рекомендації

Reporting	Формування звітів та експорт результатів
-----------	--

Пакетна структура забезпечує модульність програмного забезпечення. Наприклад, зміни в алгоритмі розрахунку КРІ не потребують зміни інтерфейсу користувача або структури бази даних. Аналогічно, модуль імпорту може бути замінений або розширений без зміни аналітичного ядра системи.

2.4 Діаграма розгортання програмного застосунку

Діаграма розгортання описує фізичну архітектуру програмного застосунку та показує, на яких вузлах розміщуються його основні компоненти. Для системи аналізу ефективності управління малим бізнесом передбачено клієнт-серверну архітектуру, яка включає клієнтський пристрій, сервер застосунку, сервер бази даних і вузол резервного копіювання.

Діаграму розгортання наведено на рис. 2.4.

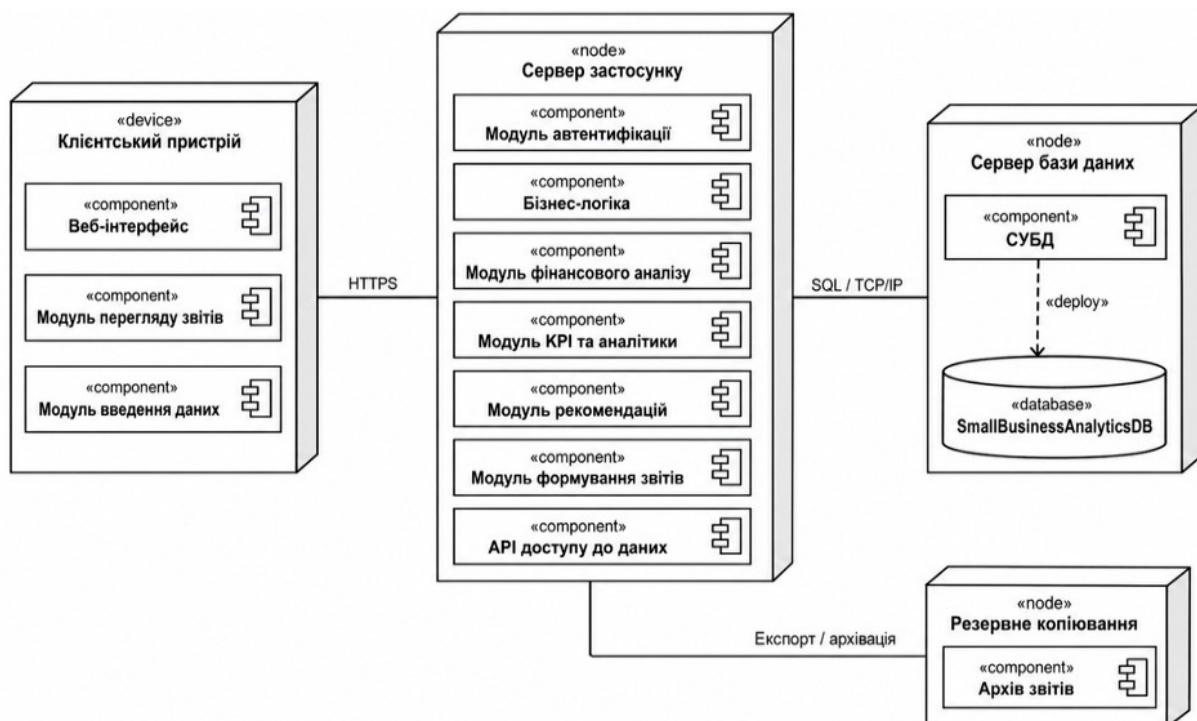


Рис. 2.4. Діаграма розгортання програмного застосунку

Клієнтський пристрій використовується власником, керівником або менеджером малого підприємства для роботи з веб-інтерфейсом, перегляду

звітів і введення операційних даних. Передавання даних між клієнтським пристроєм і сервером застосунку здійснюється через захищене з'єднання HTTPS.

Сервер застосунку містить основні програмні модулі: модуль автентифікації, бізнес-логіку, модуль фінансового аналізу, модуль KPI та аналітики, модуль рекомендацій, модуль формування звітів і API доступу до даних. Саме сервер застосунку виконує оброблення користувацьких запитів, розрахунок показників, формування рекомендацій і підготовку звітів.

Сервер бази даних забезпечує збереження всієї структурованої інформації системи. У базі SmallBusinessAnalyticsDB зберігаються користувачі, підприємства, фінансові операції, категорії, KPI, результати розрахунків, рекомендації та звіти. Окремий вузол резервного копіювання призначений для архівації сформованих звітів і збереження резервних копій даних.

Таблиця 2.5

Вузли розгортання програмного застосунку

Вузол	Призначення
Клієнтський пристрій	Робота користувача з веб-інтерфейсом
Сервер застосунку	Виконання бізнес-логіки, аналітики та звітності
Сервер бази даних	Збереження структурованих даних системи
Резервне копіювання	Архівація звітів і збереження резервних копій

Запропонована архітектура дозволяє використовувати систему як у межах локальної мережі підприємства, так і у хмарному середовищі. Це забезпечує гнучкість розгортання та можливість подальшого масштабування.

2.5 Проєктування програмних модулів системи

На основі логічної та фізичної моделей даних, пакетної структури й діаграми розгортання визначено основні програмні модулі системи. Кожний модуль відповідає за окрему частину функціоналу та взаємодіє з іншими модулями через прикладні сервіси.

До складу програмного застосунку входять такі модулі: модуль автентифікації, модуль управління бізнесом, модуль фінансових операцій, модуль KPI, модуль аналітики, модуль рекомендацій, модуль звітності, модуль імпорту даних і модуль адміністрування.

Таблиця 2.6

Основні програмні модулі системи

Модуль	Призначення
Модуль автентифікації	Перевірка облікових даних користувача
Модуль управління бізнесом	Ведення даних про підприємство та користувачів
Модуль фінансових операцій	Введення, редагування та аналіз доходів і витрат
Модуль KPI	Розрахунок ключових показників ефективності
Модуль аналітики	Виявлення тенденцій і відхилень у показниках
Модуль рекомендацій	Формування управлінських порад
Модуль звітності	Створення звітів за вибраний період
Модуль імпорту	Завантаження даних із зовнішніх облікових систем
Модуль адміністрування	Керування користувачами, ролями та довідниками

Взаємодія модулів побудована таким чином, що дані спочатку вводяться або імпортуються в систему, після чого зберігаються в базі даних. Далі аналітичний модуль отримує ці дані, виконує розрахунок KPI, порівнює планові

та фактичні значення, визначає відхилення та передає результати до модуля рекомендацій. Після цього користувач може переглянути аналітичну панель або сформувати звіт.

Такий підхід забезпечує повний цикл роботи програмного застосунку: від введення первинних даних до отримання аналітичних висновків і управлінських рекомендацій.

2.6 Висновки до другого розділу

У другому розділі виконано проектування інформаційного та програмного забезпечення програмного застосунку для аналізу ефективності управління малим бізнесом. Розроблено логічну модель даних, яка відображає основні сутності предметної області: користувачів, підприємства, фінансові операції, категорії, KPI, результати розрахунків, рекомендації та звіти.

На основі логічної моделі побудовано фізичну модель бази даних. Визначено склад таблиць, основні поля, типи даних, первинні та зовнішні ключі. Наведено SQL-код створення таблиць, індексів і внесення початкових довідкових даних. Це забезпечує практичну основу для подальшої реалізації інформаційної бази системи.

Розроблено пакетну структуру програмного забезпечення, яка передбачає поділ системи на рівні інтерфейсу, прикладної логіки, доменної моделі, доступу до даних, безпеки, інтеграції, аналітики та звітності. Такий підхід підвищує модульність системи та спрощує її подальший супровід.

Також описано діаграму розгортання, яка відображає клієнт-серверну архітектуру застосунку. Визначено основні вузли розміщення: клієнтський пристрій, сервер застосунку, сервер бази даних і вузол резервного копіювання. Запропонована архітектура дозволяє реалізувати систему як локально, так і у хмарному середовищі.

У результаті виконання другого розділу сформовано основу для подальшої реалізації програмного забезпечення. Розроблені моделі й SQL-структура

забезпечують узгоджене збереження даних, підтримку аналітичних розрахунків, формування KPI, рекомендацій і звітів для оцінювання ефективності управління малим бізнесом.

3 РОЗРОБКА ПРОГРАМНОГО ТА АЛГОРИТМІЧНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Обґрунтування вибору технологічних засобів реалізації

Розробка програмного застосунку для аналізу ефективності управління малим бізнесом потребує вибору таких технологічних засобів, які забезпечують швидке введення операційних даних, надійне збереження фінансової інформації, коректний розрахунок ключових показників ефективності та зручне подання результатів у вигляді аналітичної панелі і звітів. Для малого підприємства особливо важливо, щоб програмний продукт був простим у встановленні, не потребував складної серверної інфраструктури на етапі прототипування та дозволяв надалі масштабувати функціонал.

Для реалізації програмного прототипу обрано мову програмування Python, оскільки вона має розвинену екосистему бібліотек для роботи з базами даних, оброблення таблиць, побудови графіків, експорту звітів і швидкого створення прикладних програм. Графічний інтерфейс доцільно реалізувати з використанням PyQt6, що дає змогу створити кросплатформенний застосунок із сучасною аналітичною панеллю, таблицями, формами введення даних і вікнами звітності. Для локального збереження даних використовується SQLite, яка не потребує окремого сервера бази даних та підходить для автономної роботи малого підприємства або навчального демонстраційного прототипу.

Обрана технологічна база узгоджується з функціями, визначеними у попередніх розділах: авторизацією користувача, веденням підприємств, фінансових категорій, доходів і витрат, розрахунком KPI, формуванням рекомендацій і підготовкою звітів. Основні технології, використані під час реалізації програмного забезпечення, наведено в табл. 3.1.

Таблиця 3.1

Технологічні засоби реалізації програмного застосунку

Технологія	Призначення у системі
Python 3.x	Реалізація прикладної логіки, сервісів оброблення даних, розрахунку KPI та формування рекомендацій
PyQt6	Побудова графічного інтерфейсу користувача, форм введення, таблиць, інформаційних карток і панелі аналітики
SQLite	Локальне збереження користувачів, підприємств, фінансових операцій, KPI, звітів і рекомендацій
SQL	Створення структури бази даних, вибірка, фільтрація, групування та агрегація фінансових операцій
CSV / PDF / HTML	Експорт звітів для подальшого перегляду, друку або передавання керівнику підприємства
QSS	Стилізація інтерфейсу користувача та забезпечення єдиного візуального оформлення програми

Використання зазначених технологій дає змогу створити програмний застосунок, який може працювати на персональному комп'ютері керівника або менеджера малого підприємства. При цьому логіка роботи системи не обмежується локальним використанням: у разі потреби вона може бути перенесена на клієнт-серверну або веб-архітектуру, оскільки структура даних, модулі оброблення та алгоритми KPI мають чітко відокремлену будову.

3.2 Практична реалізація інформаційної бази системи

Інформаційна база програмного застосунку реалізується на основі фізичної моделі, розробленої у другому розділі. База даних зберігає облікові записи користувачів, дані підприємств, зв'язки між користувачами та підприємствами, фінансові категорії, операції доходів і витрат, довідник KPI, результати їх розрахунку, управлінські рекомендації та сформовані звіти.

Під час запуску програми виконується перевірка наявності файлу бази даних. Якщо файл відсутній, застосунок створює таблиці, індекси та початкові довідники. Такий підхід спрощує розгортання програмного продукту, оскільки користувачу не потрібно окремо встановлювати СУБД або виконувати SQL-сценарії вручну.

Основна логіка роботи з базою даних винесена в окремий модуль доступу до даних. Він відповідає за підключення до SQLite, виконання запитів, повернення результатів у зручному форматі та фіксацію службових дій. Фрагмент програмного коду ініціалізації бази даних наведено нижче.

```
import sqlite3
from pathlib import Path

DB_PATH = Path("small_business_analytics.db")

class Database:
    def __init__(self, path=DB_PATH):
        self.conn = sqlite3.connect(path)
        self.conn.row_factory = sqlite3.Row
        self.create_schema()

    def execute(self, query, params=()):
        cursor = self.conn.cursor()
        cursor.execute(query, params)
        self.conn.commit()
        return cursor.lastrowid

    def fetchall(self, query, params=()):
        cursor = self.conn.cursor()
        cursor.execute(query, params)
        return cursor.fetchall()

    def create_schema(self):
        self.execute("""
            CREATE TABLE IF NOT EXISTS financial_transactions (
                id_transaction INTEGER PRIMARY KEY AUTOINCREMENT,
                id_business INTEGER NOT NULL,
                id_category INTEGER NOT NULL,
                id_user INTEGER NOT NULL,
                amount REAL NOT NULL,
                transaction_date TEXT NOT NULL,
                transaction_type TEXT NOT NULL,
                description TEXT,
                created_at TEXT NOT NULL
            )
        """)
```

Для забезпечення аналітичних функцій у базі даних передбачено збереження як первинних фінансових операцій, так і розрахованих результатів. Первинні операції використовуються для побудови звітів за період, а таблиці КРІ та рекомендацій забезпечують збереження результатів аналізу й можливість повторного перегляду раніше сформованих висновків.

Таблиця 3.2

Основні програмні модулі доступу до даних

Модуль	Призначення
Database	Підключення до SQLite, виконання SQL-запитів, створення структури бази даних
UserRepository	Отримання даних користувачів, перевірка облікових записів, збереження ролей
BusinessRepository	Робота з даними підприємств і зв'язками користувачів із бізнесом
TransactionRepository	Додавання, редагування, видалення та вибірка фінансових операцій
KPIRepository	Збереження довідника KPI та результатів розрахунку показників
ReportRepository	Фіксація сформованих звітів і параметрів звітного періоду

Розподіл логіки доступу до даних за окремими репозиторіями зменшує зв'язність між інтерфейсом користувача та базою даних. Це дає змогу змінювати внутрішню структуру SQL-запитів без суттєвого переписування вікон програми або сервісів аналітики.

3.3 Алгоритмізація програмних модулів системи

Алгоритмізація є важливим етапом розроблення програмного застосунку, оскільки вона визначає послідовність оброблення вхідних даних, порядок розрахунку показників і логіку формування управлінських рекомендацій. Для системи аналізу ефективності управління малим бізнесом розроблено три ключові алгоритми: формування фінансового звіту, розрахунок KPI та формування рекомендацій за результатами відхилень.

Перший алгоритм описує формування фінансового звіту за вибраним підприємством і звітним періодом. Користувач задає підприємство та період аналізу, після чого система зчитує фінансові операції з бази даних. Якщо операції за період не знайдені, формується порожній звіт із відповідним повідомленням. Якщо дані наявні, вони групуються за типом і категоріями, після чого

обчислюються дохід, витрати та прибуток. Блок-схему алгоритму наведено на рис. 3.1.

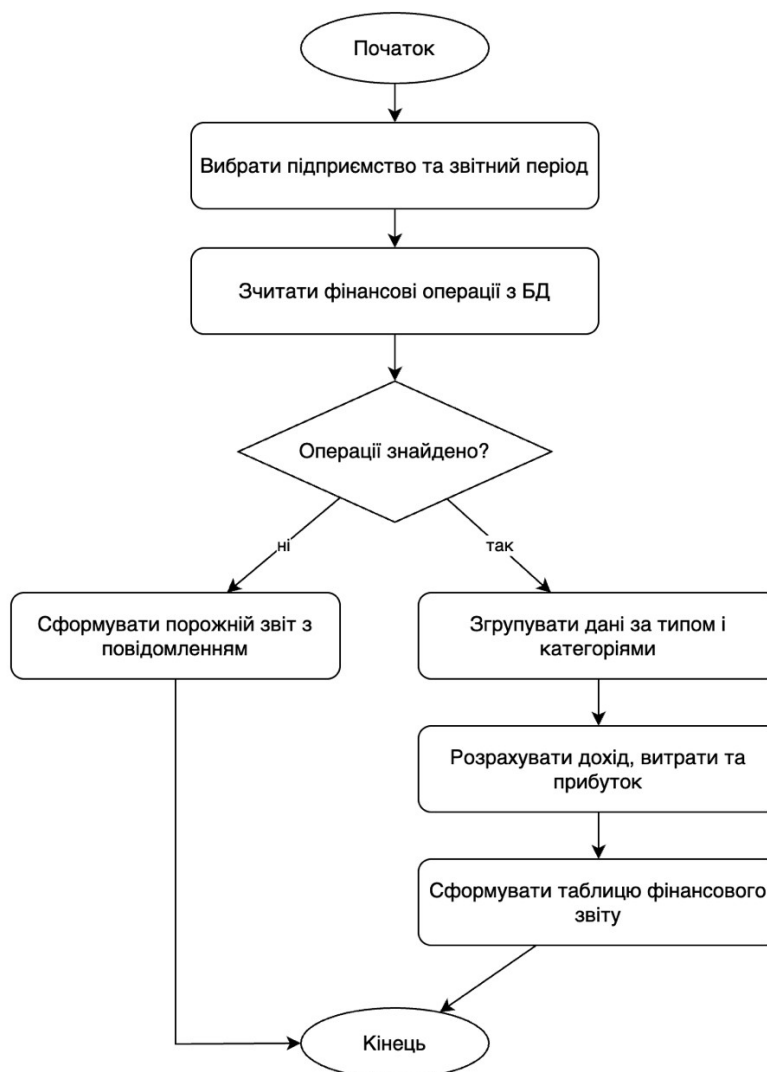


Рис. 3.1. Блок-схема алгоритму формування фінансового звіту

Запропонований алгоритм забезпечує коректну роботу навіть у випадку відсутності фінансових операцій. Це важливо для початкового етапу використання системи, коли підприємство ще не внесло достатню кількість облікових даних. У разі наявності операцій система формує підсумкову таблицю, яка надалі використовується в аналітичній панелі та під час експорту звіту.

Другий алгоритм призначений для розрахунку ключових показників ефективності. Користувач вибирає підприємство, КРІ та період аналізу. Система отримує планові значення показників, фактичні фінансові або операційні дані, обчислює КРІ за формулами та визначає наявність відхилення від плану. Якщо

відхилення відсутнє, КРІ позначається як виконаний. Якщо відхилення перевищує допустимий рівень, показник визначається як проблемний. Блок-схему алгоритму наведено на рис. 3.2.



Рис. 3.2. Блок-схема алгоритму розрахунку КРІ

Розрахунок КРІ виконується на основі фінансових операцій, планових значень і формул, збережених у довіднику показників. Для малого бізнесу

доцільно використовувати такі показники, як дохід, витрати, прибуток, рентабельність, виконання плану, середній чек, частка витрат у доході та динаміка продажів. Отримані результати зберігаються в таблиці `kpi_results` і можуть бути використані для побудови звітів за різні періоди.

Третій алгоритм описує формування управлінських рекомендацій. Система отримує результати КРІ за період, відбирає показники з негативними відхиленнями та перевіряє критичність проблеми. Якщо ризики не виявлені, формується звіт без рекомендацій. Якщо відхилення є критичними, система визначає пріоритет проблеми, формує текст управлінської рекомендації, зберігає її в базі даних і відображає у звіті. Блок-схему алгоритму наведено на рис. 3.3.

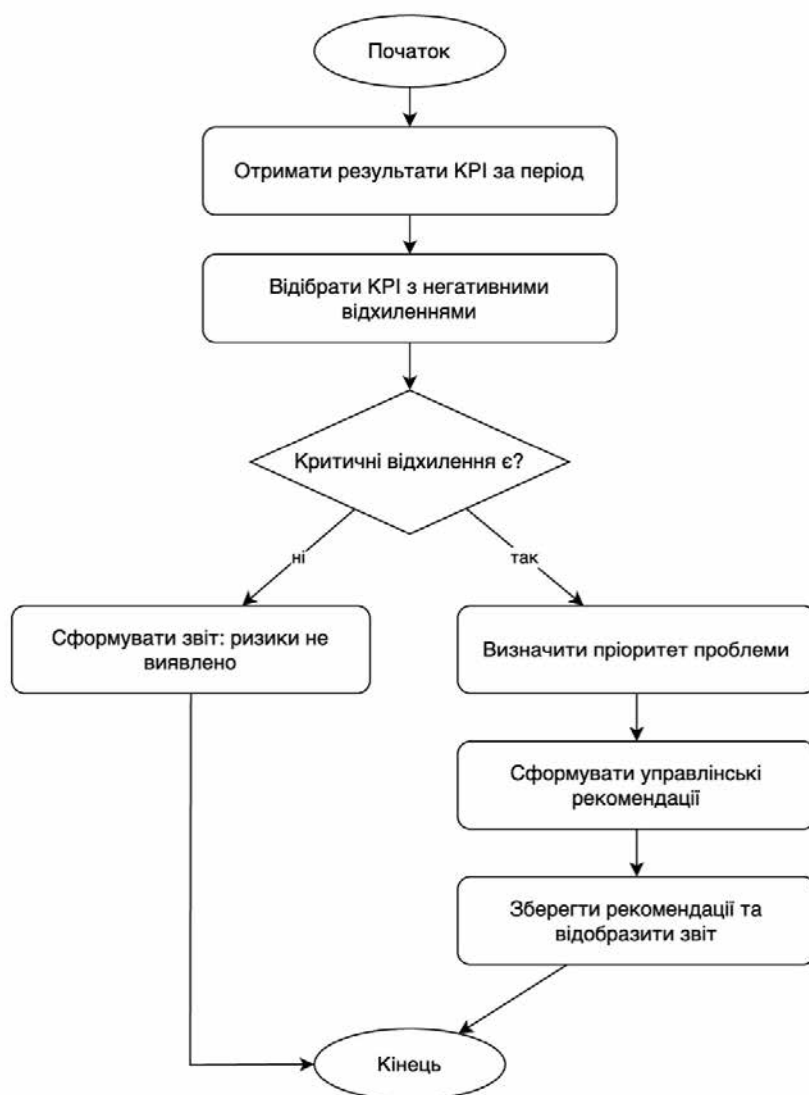


Рис. 3.3. Блок-схема алгоритму формування управлінських рекомендацій

Алгоритм формування рекомендацій дає змогу перетворити числові результати KPI на практичні управлінські висновки. Наприклад, у разі зниження прибутковості система може рекомендувати переглянути структуру витрат, посилити контроль закупівель або скоригувати ціноутворення. Якщо витрати зростають швидше за дохід, формується рекомендація щодо аналізу найбільших категорій витрат і пошуку резервів оптимізації.

Таблиця 3.3

Призначення основних алгоритмів системи

Алгоритм	Вхідні дані	Вихідні дані
Формування фінансового звіту	Підприємство, звітний період, фінансові операції	Звіт із доходами, витратами, прибутком і групуванням за категоріями
Розрахунок KPI	Планові значення, фактичні фінансові дані, формули KPI	Розраховані показники та статус виконання
Формування рекомендацій	Результати KPI, відхилення від плану, порогові значення	Пріоритети проблем і текст управлінських рекомендацій

Отже, розроблені алгоритми охоплюють повний цикл аналітичної роботи системи: отримання даних, фінансове узагальнення, розрахунок KPI, виявлення відхилень і формування рекомендацій для керівника малого підприємства.

3.4 Реалізація модуля фінансового аналізу

Модуль фінансового аналізу призначений для оброблення доходів і витрат підприємства за обраний період. Його завдання полягає в отриманні фінансових операцій із бази даних, групуванні їх за типами й категоріями, обчисленні загального доходу, загальних витрат, прибутку та частки витрат у доході.

Фінансовий аналіз є базою для подальшого розрахунку KPI, оскільки більшість показників ефективності малого бізнесу прямо залежить від доходів, витрат і прибутку. Фрагмент програмної реалізації сервісу фінансового аналізу наведено нижче.

```

class FinancialAnalysisService:
    def __init__(self, db):
        self.db = db

    def get_transactions(self, business_id, date_from, date_to):
        return self.db.fetchall(
            «»»
            SELECT t.amount, t.transaction_type, c.category_name
            FROM financial_transactions t
            JOIN financial_categories c ON c.id_category = t.id_category
            WHERE t.id_business = ?
            AND t.transaction_date BETWEEN ? AND ?
            «»»,
            (business_id, date_from, date_to)
        )

    def build_financial_summary(self, business_id, date_from, date_to):
        rows = self.get_transactions(business_id, date_from, date_to)
        income = sum(row[«amount»] for row in rows if row[«transaction_type»] ==
«income»)
        expenses = sum(row[«amount»] for row in rows if row[«transaction_type»] ==
«expense»)
        profit = income - expenses
        expense_share = (expenses / income * 100) if income > 0 else 0

        return {
            «income»: round(income, 2),
            «expenses»: round(expenses, 2),
            «profit»: round(profit, 2),
            «expense_share»: round(expense_share, 2),
            «transactions_count»: len(rows)
        }

```

Сервіс повертає не лише підсумкові значення, а й кількість опрацьованих операцій. Це дає змогу в інтерфейсі користувача попередити про недостатню кількість даних або сформувати порожній звіт, якщо за вибраний період операції відсутні. Такий підхід підвищує стійкість застосунку до неповних вхідних даних.

3.5 Реалізація модуля KPI та рекомендацій

Модуль KPI виконує розрахунок ключових показників ефективності на основі фінансових і операційних даних. Для кожного показника визначається фактичне значення, планове значення, відхилення та статус виконання. Якщо фактичний результат є нижчим за плановий або витрати перевищують допустимий рівень, система передає дані до модуля рекомендацій.

У межах програмного прототипу передбачено розрахунок прибутку, рентабельності, виконання плану доходу, частки витрат і середнього чека. Розрахунок виконується за окремими функціями, що спрощує подальше додавання нових КРІ. Фрагмент реалізації наведено нижче.

```
class KPIService:
    def calculate_profitability(self, income, expenses):
        profit = income - expenses
        if income <= 0:
            return 0
        return round(profit / income * 100, 2)

    def calculate_plan_completion(self, actual_value, planned_value):
        if planned_value <= 0:
            return 0
        return round(actual_value / planned_value * 100, 2)

    def get_kpi_status(self, actual_value, planned_value, critical_limit=10):
        if planned_value <= 0:
            return "no_plan"
        deviation = (actual_value - planned_value) / planned_value * 100
        if deviation < -critical_limit:
            return "problem"
        return "completed"

class RecommendationService:
    def build_recommendation(self, kpi_name, status, actual_value, planned_value):
        if status != "problem":
            return None

        if kpi_name == "Прибуток":
            return "Перевірити структуру витрат, переглянути ціни та проаналізувати нерентабельні напрями."
        if kpi_name == "Рентабельність":
            return "Оптимізувати постійні витрати та підвищити маржинальність основних товарів або послуг."
        if kpi_name == "Виконання плану":
            return "Посилити роботу з продажами, оновити план залучення клієнтів і перевірити ефективність реклами."
        return "Провести додатковий аналіз показника та визначити причину відхилення."
```

Збереження результатів КРІ у базі даних дозволяє відстежувати динаміку показників у часі. Керівник може порівнювати поточний період із попередніми, оцінювати вплив управлінських рішень і контролювати виконання планових значень. Рекомендації, сформовані системою, мають допоміжний характер і призначені для швидкого виявлення проблемних зон управління.

3.6 Реалізація модуля звітності та інтерфейсу користувача

Модуль звітності забезпечує підготовку фінансових і KPI-звітів у зручному для користувача вигляді. У програмному застосунку передбачено формування табличного звіту, відображення ключових показників у вигляді карток, побудову графіків динаміки та експорт результатів у зовнішній файл. Звіт може містити інформацію про дохід, витрати, прибуток, рентабельність, виконання плану й рекомендації щодо покращення управління.

Інтерфейс користувача доцільно будувати за модульним принципом. Головне вікно містить навігаційне меню, аналітичну панель, сторінку фінансових операцій, сторінку KPI, модуль рекомендацій і розділ звітів. Такий підхід дає змогу користувачу швидко переходити між основними функціями без перевантаження інтерфейсу.

Таблиця 3.4

Склад основних інтерфейсних сторінок

Сторінка	Основні елементи
Авторизація	Поля введення електронної пошти та пароля, кнопка входу, повідомлення про помилки
Аналітична панель	Картки доходу, витрат, прибутку, рентабельності, графіки та список рекомендацій
Фінансові операції	Таблиця операцій, фільтри періоду, форма додавання доходів і витрат
KPI	Планові та фактичні значення показників, статус виконання, відхилення
Рекомендації	Список управлінських порад із пріоритетом і статусом виконання
Звіти	Вибір періоду, формування підсумкового звіту, експорт у CSV або PDF

Для експорту звітів може використовуватися HTML або CSV-формат. HTML-звіт зручний для перегляду й друку, а CSV-файл може бути відкритий у табличному редакторі для додаткового аналізу. Фрагмент логіки експорту фінансового звіту наведено нижче.

```
import csv
from pathlib import Path
```

```

class ReportService:
    def export_financial_report_csv(self, path, summary, rows):
        with open(path, "w", newline="", encoding="utf-8-sig") as file:
            writer = csv.writer(file)
            writer.writerow(["Показник", "Значення"])
            writer.writerow(["Дохід", summary["income"]])
            writer.writerow(["Витрати", summary["expenses"]])
            writer.writerow(["Прибуток", summary["profit"]])
            writer.writerow(["Частка витрат, %", summary["expense_share"]])
            writer.writerow([])
            writer.writerow(["Категорія", "Тип", "Сума"])

            for row in rows:
                writer.writerow([
                    row["category_name"],
                    row["transaction_type"],
                    row["amount"]
                ])

        return Path(path).resolve()

```

Реалізація інтерфейсу та модуля звітності забезпечує практичне використання результатів аналітичної обробки. Користувач не лише вводить фінансові операції, а й отримує зрозумілу оцінку поточного стану бізнесу, проблемні KPI та рекомендації щодо підвищення ефективності управління.

3.7 Висновки до третього розділу

У третьому розділі виконано розроблення програмного та алгоритмічного забезпечення програмного застосунку для аналізу ефективності управління малим бізнесом. Обґрунтовано вибір технологічних засобів реалізації, зокрема Python, PyQt6, SQLite, SQL, QSS та форматів експорту CSV, PDF і HTML.

Описано практичну реалізацію інформаційної бази системи, яка забезпечує збереження фінансових операцій, показників KPI, рекомендацій і звітів. Визначено роль модуля доступу до даних і репозиторіїв, що дають змогу відокремити інтерфейс користувача від внутрішньої SQL-логіки.

Розроблено алгоритми формування фінансового звіту, розрахунку ключових показників ефективності та створення управлінських рекомендацій. Подані блок-схеми відображають послідовність вибору підприємства, зчитування даних, обчислення показників, перевірки відхилень і підготовки звітних результатів.

Також наведено фрагменти програмної реалізації сервісів фінансового аналізу, KPI, рекомендацій і експорту звітів. Запропонована структура програмного застосунку забезпечує повний цикл роботи: від введення або імпорту даних до формування аналітичного звіту й рекомендацій для керівника малого підприємства.

4 ТЕСТУВАННЯ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ СИСТЕМИ

4.1 План тестування програмних модулів та методика оцінювання результатів

Тестування програмного застосунку для аналізу ефективності управління малим бізнесом виконувалося з метою перевірки відповідності реалізованого функціоналу вимогам, визначеним на етапах аналізу та проєктування. Основну увагу приділено перевірці коректності авторизації користувача, роботи аналітичної панелі, введення та редагування облікових операцій, оброблення операційних показників, формування звіту, розрахунку рекомендацій і збереження цільових KPI.

Особливістю тестування є те, що система працює з фінансовими та управлінськими даними малого підприємства, тому перевірка охоплювала не лише коректність інтерфейсу, а й правильність розрахункової логіки. Під час тестування враховувалися показники доходу, витрат, прибутку, маржинальності, конверсії, середнього чека, повернень і вартості залучення клієнта. Також перевірялася узгодженість даних між вкладками: після введення операції або операційного показника відповідні значення мають відображатися в аналітичній панелі, звітах і рекомендаціях.

Тестування проводилося на демонстраційному наборі даних за період з 01.06.2026 до 04.06.2026. Для входу використовувалися тестові облікові записи admin/admin та manager/manager, що дало змогу перевірити доступ до функцій адміністратора і менеджера. Стан даних оцінювався за результатами

відображення таблиць, розрахункових карток КРІ, графіка динаміки, текстового звіту та сформованих управлінських рекомендацій.

Таблиця 4.1

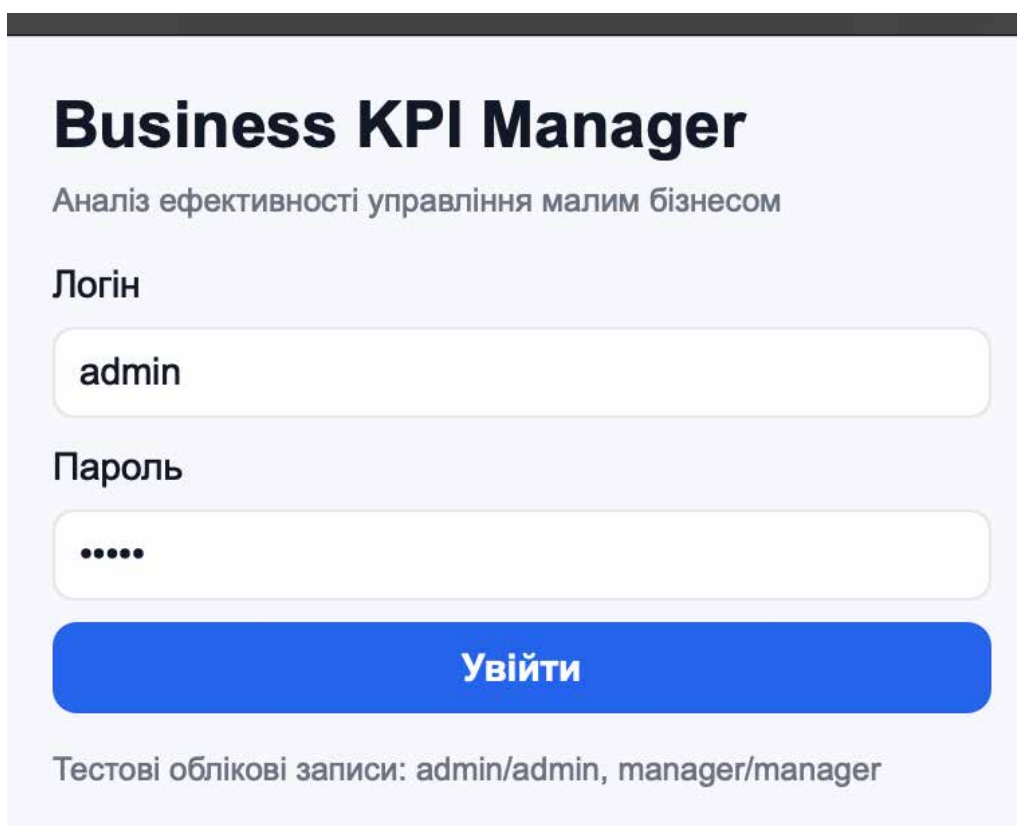
План тестування програмних модулів

Модуль	Мета перевірки	Очікуваний результат
Авторизація	Перевірити вхід за тестовими обліковими даними	Після коректного входу відкривається головне вікно системи
Аналітична панель	Перевірити розрахунок основних КРІ за період	Відображаються дохід, витрати, прибуток, маржинальність, конверсія та інші показники
Облікові дані	Перевірити створення, редагування, видалення та експорт операцій	Операції зберігаються у базі даних і коректно відображаються у таблиці
Операційні показники	Перевірити введення клієнтів, лідів, угод, повернень і рекламних витрат	Показники використовуються для розрахунку конверсії, САС і повернень
Звіти	Перевірити формування текстового звіту за вибраний період	Створюється звіт із фінансовими показниками та структурою доходів і витрат
Рекомендації	Перевірити автоматичне формування управлінських рекомендацій	Система визначає проблемні КРІ і пропонує дії для їх поліпшення
Налаштування	Перевірити збереження цільових КРІ	Цільові значення використовуються під час оцінювання відхилень

План тестування охоплює всі основні сценарії використання системи. Це дає змогу оцінити працездатність програмного застосунку як єдиного комплексу, у якому введення первинних даних, аналітичні розрахунки, звіти й рекомендації пов'язані між собою через спільну інформаційну базу.

4.2 Тестування програмного застосунку для аналізу ефективності управління малим бізнесом

На першому етапі було перевірено роботу форми авторизації. У вікні входу передбачено поля для введення логіна та пароля, а також кнопку переходу до головного вікна системи. Тестування виконувалося з використанням демонстраційних облікових записів адміністратора та менеджера. За умови правильного введення даних система відкривала робочий інтерфейс, а в разі порожніх або некоректних полів мала повідомляти користувача про неможливість входу.



The image shows a login form for a system titled "Business KPI Manager". The subtitle is "Аналіз ефективності управління малим бізнесом". The form has two input fields: "Логін" (Login) with the text "admin" and "Пароль" (Password) with masked characters ".....". Below the fields is a blue button labeled "Увійти" (Login). At the bottom, there is a line of text: "Тестові облікові записи: admin/admin, manager/manager".

Рис. 4.1. Форма авторизації користувача в системі Business KPI Manager

Результати перевірки підтвердили, що форма входу забезпечує базовий контроль доступу до функцій застосунку. Наявність тестових облікових записів спрощує перевірку роботи системи під різними ролями та дозволяє швидко перейти до тестування аналітичних модулів.

Після успішної авторизації було перевірено аналітичну панель. Вона є основним робочим екраном системи, оскільки містить узагальнені показники ефективності за вибраний період. На панелі відображаються картки доходу, витрат, прибутку, маржинальності, конверсії, середнього чека, САС і рівня повернень. Також передбачено графік динаміки показників і короткий управлінський висновок.

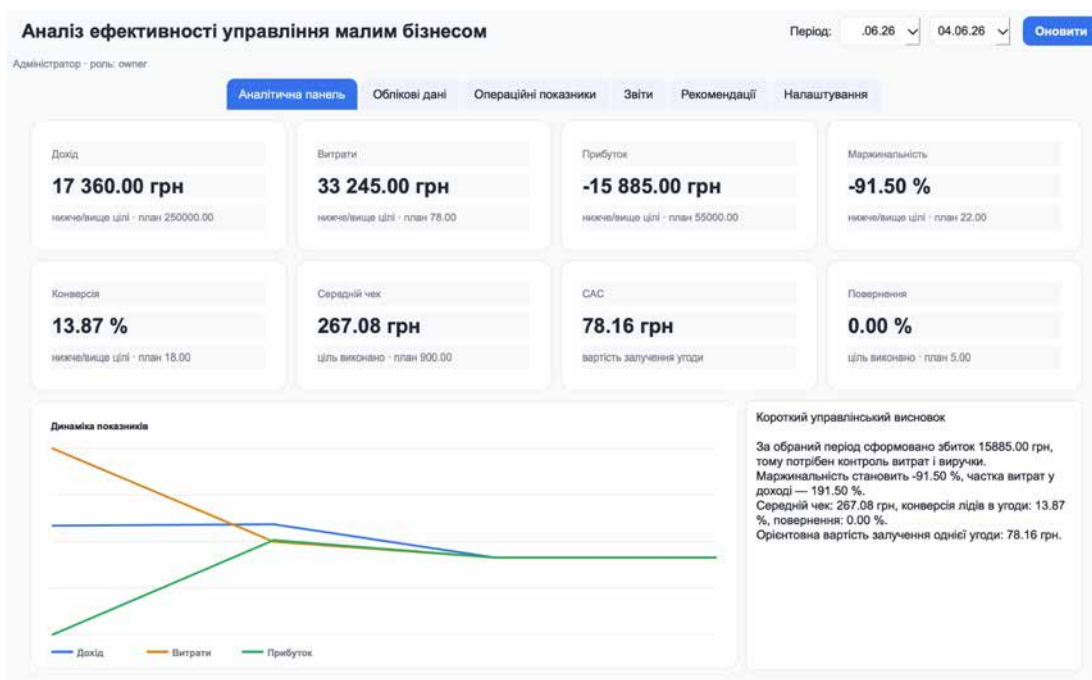


Рис. 4.2. Аналітична панель із розрахованими показниками ефективності

За тестовий період система розрахувала дохід 17360,00 грн, витрати 33245,00 грн, прибуток -15885,00 грн і маржинальність -91,50 %. Конверсія лідів в угоди становила 13,87 %, середній чек - 267,08 грн, а орієнтовна вартість залучення однієї угоди - 78,16 грн. Система коректно відобразила збитковий результат і сформуvala управлінський висновок про необхідність контролю витрат та виручки.

Наступним етапом перевірено модуль облікових даних. Він призначений для внесення фінансових операцій підприємства, зокрема доходів і витрат за категоріями. У таблиці відображаються дата операції, тип, категорія, сума, опис і джерело даних. Додатково реалізовано фільтрацію за типом операції та експорт таблиці у CSV.

Аналіз ефективності управління малим бізнесом Період: 06.26 04.06.26 [Оновити](#)

Адміністратор - роль: owner

Аналітична панель **Облікові дані** Операційні показники Звітні Рекомендації Налаштування

[Додати операцію](#) [Редагувати](#) [Видалити](#) [Експорт CSV](#) Тип: all

	ID	Дата	Тип	Категорія	Сума	Опис	Джерело
1	296	2026-06-02	expense	Маркетинг	775.00	Реклама	demo
2	295	2026-06-02	expense	Закупівлі	3440.00	Поточні закупівлі	demo
3	294	2026-06-02	revenue	Надання послуг	1810.00	Сервісні надходження	demo
4	293	2026-06-02	revenue	Продаж товарів	7040.00	Щоденний продаж	demo
5	292	2026-06-01	expense	Оренда	25000.00	Оренда приміщення	demo
6	291	2026-06-01	expense	Маркетинг	710.00	Реклама	demo
7	290	2026-06-01	expense	Закупівлі	3320.00	Поточні закупівлі	demo
8	289	2026-06-01	revenue	Надання послуг	1700.00	Сервісні надходження	demo
9	288	2026-06-01	revenue	Продаж товарів	6810.00	Щоденний продаж	demo

Рис. 4.3. Модуль перегляду та фільтрації облікових фінансових операцій

Під час перевірки встановлено, що операції за 01.06.2026 та 02.06.2026 коректно відображаються у таблиці, а доходи і витрати розділяються за типами revenue та expense. Це дає змогу системі надалі групувати операції за категоріями й використовувати їх для розрахунку фінансового звіту.

Для перевірки введення нової операції було протестовано діалогове вікно додавання запису. У формі передбачено вибір дати, типу операції, категорії, введення суми, опису та джерела. Значення суми вводиться у числовому форматі, що зменшує ризик появи помилкових текстових значень у фінансових розрахунках.

The image shows a dialog box titled "Операція" (Operation) with the following fields:

- Дата (Date): .06.26
- Тип (Type): revenue
- Категорія (Category): Надання послуг
- Сума (Sum): 1000,00
- Опис (Description): (empty)
- Джерело (Source): manual

At the bottom are "Cancel" and "OK" buttons. The background shows a table with values like 75.00, 440.00, 810.00, 040.00, 5000.00, 10.00, 320.00, 700.00, and 810.00.

Рис. 4.4. Діалогове вікно додавання фінансової операції

За результатами тестування діалогове вікно дозволяє додати нову фінансову операцію та передати її до загальної таблиці. Після збереження запису показники аналітичної панелі можуть бути перераховані для вибраного періоду, що підтверджує зв'язок між модулем введення даних і модулем розрахунку КРІ.

Окремо протестовано модуль операційних показників. Він використовується для внесення нефінансових даних, які впливають на управлінську аналітику: кількості клієнтів, замовлень, лідів, успішних угод, повернень і рекламних витрат. Ці показники застосовуються для визначення конверсії, середнього чека, повернень і вартості залучення клієнта.

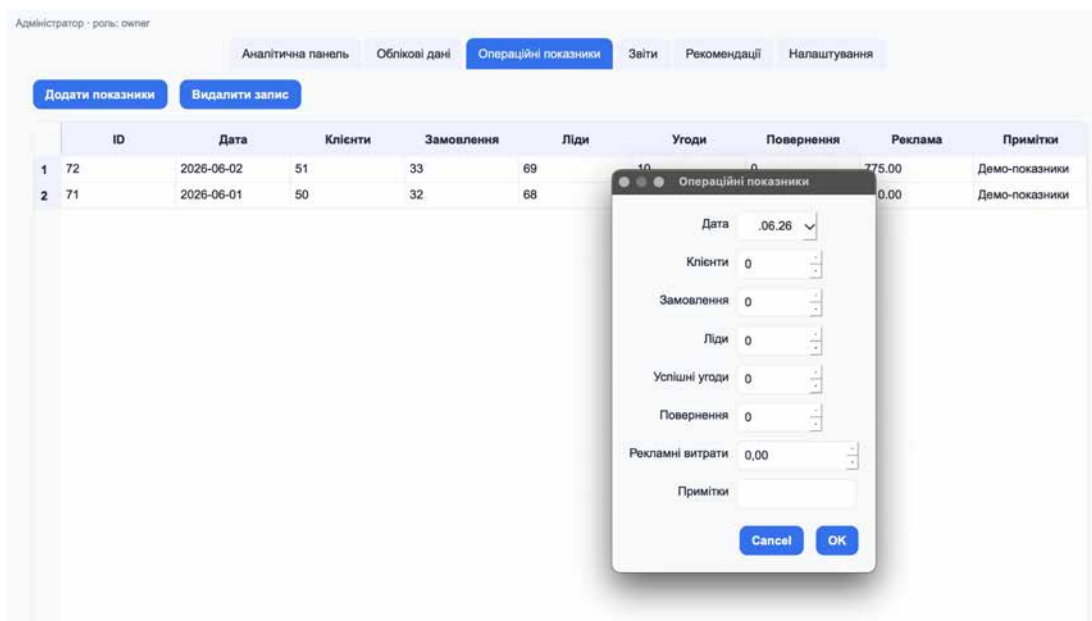


Рис. 4.5. Модуль операційних показників і діалогове вікно їх введення

Під час тестування було внесено показники за 01.06.2026 та 02.06.2026. Наприклад, для 02.06.2026 у таблиці відображено 51 клієнта, 33 замовлення, 69 лідів, 10 успішних угод, 0 повернень і 775,00 грн рекламних витрат. Дані коректно зберігаються та використовуються у розрахунках конверсії і САС.

Далі перевірено модуль формування звітів. Він узагальнює фінансові та операційні показники за вибраний період і створює текстовий звіт. У звіті наведено загальні значення доходу, витрат, прибутку, маржинальності, середнього чека, конверсії, рівня повернень і САС, а також структуру доходів і витрат за категоріями.

Сформувати звіт
Зберегти TXT

ЗВІТ ПРО ЕФЕКТИВНІСТЬ УПРАВЛІННЯ МАЛИМ БІЗНЕСОМ
 Період: 2026-06-01 — 2026-06-04

Дохід: 17360.00 грн
 Витрати: 33245.00 грн
 Прибуток: -15885.00 грн
 Маржинальність: -91.50 %
 Середній чек: 267.08 грн
 Конверсія: 13.87 %
 Рівень повернень: 0.00 %
 САС: 78.16 грн

Короткий управлінський висновок

За обраний період сформовано збиток 15885.00 грн, тому потрібен контроль витрат і виручки.
 Маржинальність становить -91.50 %, частка витрат у доході — 191.50 %.
 Середній чек: 267.08 грн, конверсія лідів в угоди: 13.87 %, повернення: 0.00 %.
 Орієнтовна вартість залучення однієї угоди: 78.16 грн.

Структура доходів:

- Продаж товарів: 13850.00 грн
- Надання послуг: 3510.00 грн

Структура витрат:

- Оренда: 25000.00 грн
- Закупівлі: 6760.00 грн
- Маркетинг: 1485.00 грн
- Зарплата: 0.00 грн
- Логістика: 0.00 грн
- Адміністративні витрати: 0.00 грн

Рис. 4.6. Сформований звіт про ефективність управління малим бізнесом

Звіт за період 01.06.2026-04.06.2026 містить короткий управлінський висновок, згідно з яким сформовано збиток 15885,00 грн, а маржинальність є від’ємною. У структурі доходів відображено продаж товарів і надання послуг, а у структурі витрат - оренду, закупівлі та маркетинг. Це підтверджує, що модуль звітності не лише виводить підсумкові значення, а й деталізує причини отриманого результату.

Наступним етапом перевірено модуль рекомендацій. Він аналізує відхилення фактичних KPI від цільових значень і формує перелік управлінських дій. У тестовому сценарії система запропонувала збільшити середній чек, підвищити конверсію продажів, оптимізувати витрати та посилити контроль прибутковості.

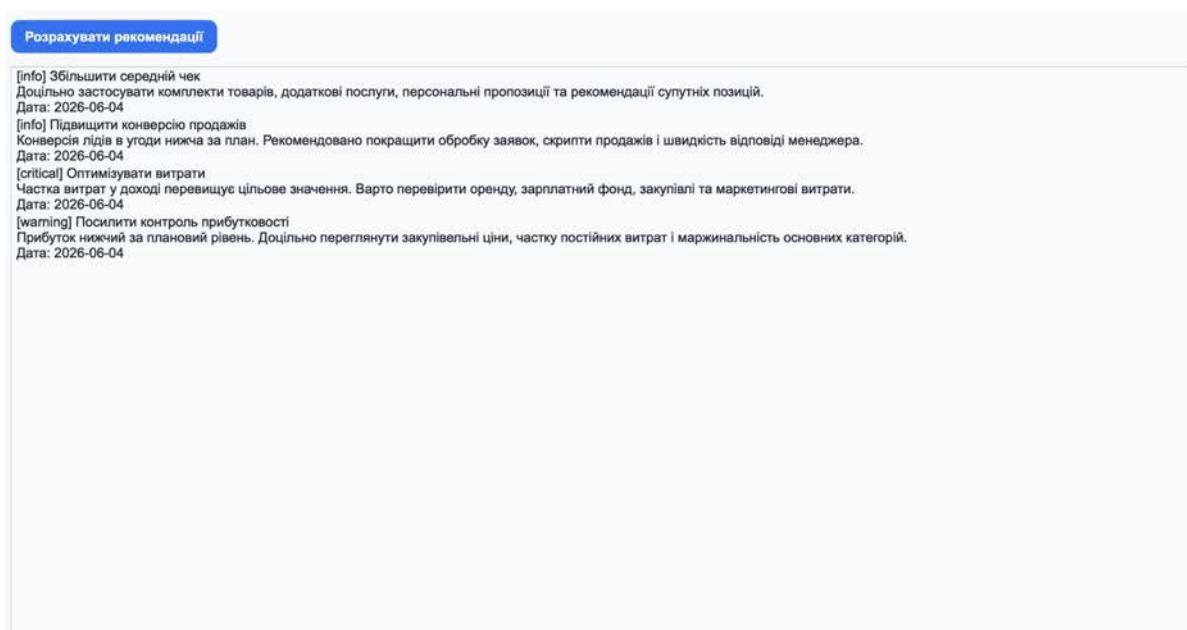


Рис. 4.7. Модуль формування управлінських рекомендацій

Рекомендації мають пріоритети info, warning та critical. Це дозволяє керівнику малого бізнесу швидко визначити, які проблеми є найбільш важливими. Наприклад, рекомендація щодо оптимізації витрат позначена як critical, оскільки частка витрат у доході перевищує цільове значення.

Завершальним етапом перевірено модуль налаштування цільових KPI. У ньому задаються планові значення доходу, прибутку, маржинальності, частки витрат, конверсії, повернень і середнього чека. Для кожного показника визначається напрям оцінювання: higher для показників, збільшення яких є позитивним, або lower для показників, зменшення яких є бажаним.

Цільові KPI

Дохід	250000,00	higher
Прибуток	55000,00	higher
Маржинальність, %	22,00	higher
Частка витрат, %	78,00	lower
Конверсія, %	18,00	higher
Повернення, %	5,00	lower
Середній чек	900,00	lower

Зберегти цілі

Рис. 4.8. Налаштування цільових KPI для оцінювання ефективності

Після збереження цілей система використовує їх під час аналізу фактичних результатів. Саме ці значення є основою для визначення відхилень, формування статусу KPI та автоматичного створення рекомендацій. Це забезпечує адаптацію системи до конкретної моделі малого бізнесу.

Таблиця 4.2

Результати тестування основних функцій системи

Функція	Перевірена дія	Результат тестування
Авторизація	Вхід за обліковим записом admin/admin	Головне вікно відкривається коректно
Аналітична панель	Розрахунок доходу, витрат, прибутку та маржинальності	Показники відповідають даним операцій
Графік динаміки	Побудова графіка доходу, витрат і прибутку	Графік відображається без помилок
Облікові дані	Додавання, редагування та видалення фінансових операцій	Дані оновлюються в таблиці
Фільтрація операцій	Вибір типу all/revenue/expense	У таблиці залишаються відповідні записи

Продовження таблиці 4.2

Експорт CSV	Збереження переліку операцій у файл	Експорт виконується коректно
Операційні показники	Додавання клієнтів, замовлень, лідів і угод	Записи зберігаються та використовуються для KPI
Звіти	Формування звіту за період	Звіт містить підсумкові показники та структуру доходів і витрат
Рекомендації	Розрахунок управлінських порад	Сформовано перелік рекомендацій з пріоритетами
Налаштування	Збереження цільових KPI	Цілі використовуються під час оцінювання відхилень

Отримані результати свідчать про те, що реалізовані модулі працюють узгоджено. Дані, введені в облікових і операційних формах, використовуються аналітичною панеллю, звітами та рекомендаціями, що підтверджує цілісність інформаційної структури системи.

4.3 Оцінювання ефективності роботи системи

Ефективність програмного застосунку оцінювалася за кількома критеріями: скорочення часу підготовки управлінської інформації, зменшення кількості ручних розрахунків, оперативність формування звітів, зручність роботи з фінансовими операціями та наявність автоматичних рекомендацій. Порівняння виконувалося між умовним ручним способом оброблення даних у таблицях і використанням розробленої системи.

Таблиця 4.3

Оцінювання ефективності використання програмного застосунку

Критерій	Ручний спосіб	Розроблена система	Ефект
Підготовка фінансового звіту	30-60 хвилин	1-3 хвилини	Суттєве скорочення часу
Розрахунок КРІ	Виконується вручну за формулами	Виконується автоматично	Зменшення ризику помилок
Порівняння плану і факту	Потребує окремих таблиць	Виконується за цільовими КРІ	Підвищення оперативності контролю
Формування рекомендацій	Залежить від досвіду керівника	Формується автоматично за відхиленнями	Підтримка управлінських рішень
Аналіз доходів і витрат	Потребує ручного групування	Групується за категоріями	Швидше виявлення проблемних статей
Експорт даних	Потребує копіювання інформації	Збереження CSV/TXT	Спрощення обміну звітами

У тестовому сценарії система виявила збитковість діяльності за вибраний період, від’ємну маржинальність, низьку конверсію та потребу в контролі витрат. Ці результати були сформовані без ручного перерахунку, що демонструє практичну цінність застосунку для власника або менеджера малого підприємства.

Важливою перевагою є те, що система об’єднує фінансові та операційні дані в одному інтерфейсі. Завдяки цьому керівник може оцінювати не лише суму доходів і витрат, а й причини зміни ефективності: кількість лідів, угод,

повернень, рекламних витрат, середній чек і вартість залучення клієнта. Такий підхід підвищує обґрунтованість управлінських рішень.

Розроблений програмний застосунок є придатним для використання у малому бізнесі, де потрібний швидкий контроль основних показників без складних ERP-рішень. Система може застосовуватися у сфері торгівлі, послуг, невеликих виробничих підприємствах або сервісних компаніях, які ведуть базовий облік операцій і потребують регулярної оцінки управлінської ефективності.

4.4 Висновки до четвертого розділу

У четвертому розділі проведено тестування програмного застосунку для аналізу ефективності управління малим бізнесом. Перевірено роботу основних модулів системи: авторизації, аналітичної панелі, облікових даних, операційних показників, звітів, рекомендацій і налаштування цільових KPI.

За результатами тестування встановлено, що система коректно обробляє фінансові та операційні дані, виконує розрахунок ключових показників ефективності, формує звіти й надає управлінські рекомендації. Введені користувачем дані узгоджено використовуються в різних модулях, що підтверджує працездатність реалізованої інформаційної структури.

Оцінювання ефективності показало, що використання програмного застосунку скорочує час підготовки управлінської інформації, зменшує обсяг ручних розрахунків і підвищує наочність аналізу. Система дозволяє швидко виявляти проблемні показники, оцінювати відхилення від цільових KPI та формувати рекомендації для покращення результатів малого підприємства.

Отже, тестування підтвердило відповідність розробленого програмного забезпечення поставленим функціональним вимогам. Програмний застосунок може використовуватися як інструмент підтримки управлінських рішень у малому бізнесі та як основа для подальшого розвитку системи з додаванням

інтеграції з банківськими сервісами, бухгалтерськими системами та хмарним сховищем даних.

ВИСНОВКИ

У кваліфікаційній роботі розроблено програмний застосунок для аналізу ефективності управління малим бізнесом. Розроблена система призначена для автоматизації обліку доходів і витрат, збереження операційних показників, розрахунку KPI, формування звітів та підготовки управлінських рекомендацій для власника або керівника малого підприємства.

У першому розділі виконано аналіз предметної області управління малим бізнесом. Встановлено, що для невеликих підприємств важливими є не лише облік фінансових операцій, а й оперативне оцінювання прибутковості, витратності, конверсії продажів, середнього чека, вартості залучення клієнта та рівня повернень. Розглянуто існуючі рішення QuickBooks, Xero, Zoho Books, Odoo та HubSpot CRM. Аналіз показав, що наявні системи мають широкі можливості для фінансового обліку, CRM, звітності та аналізу cash flow, однак часто є надлишково складними, хмарними, платними або орієнтованими на комплексне корпоративне впровадження. Тому розробка окремого навчального програмного застосунку з локальною базою даних, зрозумілим інтерфейсом та фокусом на KPI малого бізнесу є доцільною. QuickBooks позиціонується як рішення для малого бізнесу з обліком доходів, витрат, cash flow, інвойсів і фінансових звітів, Xero надає dashboard для контролю cash flow, рахунків і бізнес-показників, Zoho Books підтримує фінансові звіти й аналітичні інсайти, Odoo охоплює CRM, бухгалтерію, склад, POS і проєктне управління, а HubSpot CRM орієнтований на продажі, клієнтів і звітні панелі.

У другому розділі виконано проєктування інформаційного та програмного забезпечення системи. Розроблено логічну модель даних, яка відображає основні сутності предметної області: користувачів, підприємства, зв'язки користувачів із бізнесами, фінансові категорії, фінансові транзакції, KPI, результати KPI, рекомендації та звіти. На основі логічної моделі сформовано фізичну модель бази даних із таблицями users, businesses, user_businesses, financial_categories, financial_transactions, kpis, kpi_results, recommendations і reports. Така структура

забезпечує цілісне збереження даних і дозволяє виконувати подальші аналітичні розрахунки.

У третьому розділі описано реалізацію програмного забезпечення. Для створення застосунку використано мову Python, бібліотеку PyQt6 для побудови графічного інтерфейсу та SQLite для локального збереження даних. PyQt6 є набором Python-прив'язок до Qt і підтримує роботу на Windows, macOS та Linux, а модуль sqlite3 у Python забезпечує роботу з SQLite через DB-API 2.0. У межах реалізації створено модулі авторизації, аналітичної панелі, облікових даних, операційних показників, звітності, рекомендацій і налаштувань цільових KPI. Система підтримує введення та редагування фінансових операцій, додавання операційних показників, фільтрацію за періодом, розрахунок доходу, витрат, прибутку, маржинальності, конверсії, середнього чека, CAC і рівня повернень.

У четвертому розділі проведено тестування програмного застосунку. Перевірено авторизацію користувача, роботу аналітичної панелі, додавання та редагування фінансових операцій, внесення операційних показників, формування ТХТ-звіту, розрахунок рекомендацій і збереження цільових KPI. Результати тестування підтвердили, що система коректно обробляє дані, оновлює показники за вибраний період, формує управлінський висновок, виявляє проблемні відхилення та надає рекомендації щодо підвищення ефективності управління.

Практична цінність розробленого програмного забезпечення полягає в тому, що воно може використовуватися як навчальний або демонстраційний інструмент для аналізу діяльності малого бізнесу. Застосунок дозволяє керівнику швидко побачити поточний фінансовий стан, визначити проблемні напрями, оцінити виконання планових KPI та сформулювати звіт за вибраний період. На відміну від складних ERP або CRM-систем, розроблений прототип має просту структуру, локальну базу даних і зручний інтерфейс для швидкого аналізу.

Подальший розвиток системи може передбачати додавання імпорту даних із Excel/CSV, інтеграцію з банківськими виписками, формування PDF-звітів, розширення аналітики продажів, створення прогнозування доходів і витрат,

додавання багатокористувацького режиму, резервного копіювання та вебверсії застосунку. Також перспективним є впровадження модуля автоматичного виявлення ризиків на основі історичних даних і розширення системи рекомендацій.

Отже, мету кваліфікаційної роботи досягнуто. У результаті виконання роботи проаналізовано предметну область, досліджено існуючі рішення, визначено вимоги до програмного забезпечення, спроектовано базу даних, реалізовано програмний застосунок і проведено його тестування. Розроблена система відповідає поставленим завданням і може бути використана для підтримки управлінських рішень у малому бізнесі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. QuickBooks. Small Business Accounting Software [Electronic resource] / Intuit QuickBooks. – Mode of access: <https://quickbooks.intuit.com/global/accounting-software/> – Title from screen.
2. QuickBooks. Cash Flow Management Software [Electronic resource] / Intuit QuickBooks. – Mode of access: <https://quickbooks.intuit.com/global/cash-flow/> – Title from screen.
3. QuickBooks. Accounting Software Features [Electronic resource] / Intuit QuickBooks. – Mode of access: <https://quickbooks.intuit.com/global/accountants-software/features/> – Title from screen.
4. Xero. The Xero Dashboard: Headline Insights on Your Business [Electronic resource] / Xero. – Mode of access: <https://www.xero.com/accounting-software/dashboard/> – Title from screen.
5. Xero. About short-term cash flow [Electronic resource] / Xero Central. – Mode of access: <https://central.xero.com/s/article/STCF-Cash-Predictions-AU> – Title from screen.
6. Zoho Books. Accounting Software Features [Electronic resource] / Zoho Corporation. – Mode of access: <https://www.zoho.com/us/books/accounting-software-features/> – Title from screen.
7. Zoho Books. Reports [Electronic resource] / Zoho Corporation. – Mode of access: <https://www.zoho.com/in/books/help/reports/> – Title from screen.
8. Zoho Books. Financial reporting in accounting software [Electronic resource] / Zoho Corporation. – Mode of access: <https://www.zoho.com/us/books/accounting-software/accounting-financial-reports/> – Title from screen.
9. Odoo. Open Source ERP and CRM [Electronic resource] / Odoo S.A. – Mode of access: <https://www.odoo.com/> – Title from screen.

10. Odoo. Accounting Features [Electronic resource] / Odoo S.A. – Mode of access: <https://www.odoo.com/app/accounting-features> – Title from screen.
11. HubSpot. Reporting & Dashboard Software [Electronic resource] / HubSpot. – Mode of access: <https://www.hubspot.com/products/reporting-dashboards> – Title from screen.
12. HubSpot. CRM Software [Electronic resource] / HubSpot. – Mode of access: <https://www.hubspot.com/products/crm> – Title from screen.
13. Python Software Foundation. Python 3 Documentation [Electronic resource]. – Mode of access: <https://docs.python.org/3/> – Title from screen.
14. Python Software Foundation. sqlite3 – DB-API 2.0 interface for SQLite databases [Electronic resource]. – Mode of access: <https://docs.python.org/3/library/sqlite3.html> – Title from screen.
15. Riverbank Computing. PyQt6 Documentation [Electronic resource]. – Mode of access: <https://www.riverbankcomputing.com/static/Docs/PyQt6/> – Title from screen.
16. Riverbank Computing. What is PyQt? [Electronic resource]. – Mode of access: <https://riverbankcomputing.com/software/pyqt/intro> – Title from screen.
17. SQLite Consortium. SQLite Documentation [Electronic resource]. – Mode of access: <https://sqlite.org/docs.html> – Title from screen.
18. SQLite Consortium. Foreign Key Support [Electronic resource]. – Mode of access: <https://sqlite.org/foreignkeys.html> – Title from screen.
19. OWASP Foundation. Password Storage Cheat Sheet [Electronic resource]. – Mode of access: https://cheatsheetseries.owasp.org/cheatsheets/Password_Storage_Cheat_Sheet.html – Title from screen.
20. OWASP Foundation. Authentication Cheat Sheet [Electronic resource]. – Mode of access: https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html – Title from screen.

21. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. – Київ : ДП «УкрНДНЦ», 2016. – 16 с.
22. ISO/IEC/IEEE 12207:2017. Systems and software engineering – Software life cycle processes. – Geneva : International Organization for Standardization, 2017.
23. ISO/IEC 25010:2011. Systems and software engineering – Systems and software Quality Requirements and Evaluation – System and software quality models. – Geneva : International Organization for Standardization, 2011.
24. Sommerville I. Software Engineering / I. Sommerville. – 10th ed. – Boston : Pearson, 2016. – 816 p.
25. Pressman R. S. Software Engineering: A Practitioner's Approach / R. S. Pressman, B. R. Maxim. – 9th ed. – New York : McGraw-Hill Education, 2019. – 704 p.
26. Fowler M. UML Distilled: A Brief Guide to the Standard Object Modeling Language / M. Fowler. – 3rd ed. – Boston : Addison-Wesley, 2004. – 208 p.
27. Connolly T. Database Systems: A Practical Approach to Design, Implementation, and Management / T. Connolly, C. Begg. – 6th ed. – Harlow : Pearson, 2015. – 1440 p.
28. Laudon K. C. Management Information Systems: Managing the Digital Firm / K. C. Laudon, J. P. Laudon. – 17th ed. – Harlow : Pearson, 2022.
29. Kaplan R. S. The Balanced Scorecard: Translating Strategy into Action / R. S. Kaplan, D. P. Norton. – Boston : Harvard Business School Press, 1996. – 322 p.
30. Parmenter D. Key Performance Indicators: Developing, Implementing, and Using Winning KPIs / D. Parmenter. – 4th ed. – Hoboken : Wiley, 2020.

ДОДАТОК А

Фрагмент SQL-коду створення бази даних програмного застосування

```

CREATE TABLE users (
    id_user INTEGER PRIMARY KEY AUTOINCREMENT,
    full_name TEXT NOT NULL,
    email TEXT NOT NULL UNIQUE,
    password_hash TEXT NOT NULL,
    user_type TEXT NOT NULL,
    created_at TEXT NOT NULL
);

CREATE TABLE businesses (
    id_business INTEGER PRIMARY KEY AUTOINCREMENT,
    business_name TEXT NOT NULL,
    industry TEXT NOT NULL,
    tax_number TEXT,
    created_at TEXT NOT NULL
);

CREATE TABLE user_businesses (
    id_user INTEGER NOT NULL,
    id_business INTEGER NOT NULL,
    role_in_business TEXT NOT NULL,
    PRIMARY KEY (id_user, id_business),
    FOREIGN KEY (id_user) REFERENCES users(id_user),
    FOREIGN KEY (id_business) REFERENCES businesses(id_business)
);

CREATE TABLE financial_categories (
    id_category INTEGER PRIMARY KEY AUTOINCREMENT,
    category_name TEXT NOT NULL,
    category_type TEXT NOT NULL
);

CREATE TABLE financial_transactions (
    id_transaction INTEGER PRIMARY KEY AUTOINCREMENT,
    id_business INTEGER NOT NULL,
    id_category INTEGER NOT NULL,
    id_user INTEGER NOT NULL,
    amount REAL NOT NULL,
    transaction_date TEXT NOT NULL,
    transaction_type TEXT NOT NULL,
    description TEXT,
    source TEXT DEFAULT 'manual',
    created_at TEXT NOT NULL,
    FOREIGN KEY (id_business) REFERENCES businesses(id_business),
    FOREIGN KEY (id_category) REFERENCES financial_categories(id_category),
    FOREIGN KEY (id_user) REFERENCES users(id_user)
);

CREATE TABLE operational_metrics (
    id_metric INTEGER PRIMARY KEY AUTOINCREMENT,
    id_business INTEGER NOT NULL,
    metric_date TEXT NOT NULL,
    clients INTEGER DEFAULT 0,
    orders INTEGER DEFAULT 0,
    leads INTEGER DEFAULT 0,
    successful_deals INTEGER DEFAULT 0,
    returns_count INTEGER DEFAULT 0,

```

```

    advertising_cost REAL DEFAULT 0,
    notes TEXT,
    created_at TEXT NOT NULL,
    FOREIGN KEY (id_business) REFERENCES businesses(id_business)
);

CREATE TABLE kpi_targets (
    id_target INTEGER PRIMARY KEY AUTOINCREMENT,
    id_business INTEGER NOT NULL,
    kpi_name TEXT NOT NULL,
    target_value REAL NOT NULL,
    direction TEXT NOT NULL,
    updated_at TEXT NOT NULL,
    FOREIGN KEY (id_business) REFERENCES businesses(id_business)
);

CREATE TABLE recommendations (
    id_recommendation INTEGER PRIMARY KEY AUTOINCREMENT,
    id_business INTEGER NOT NULL,
    priority TEXT NOT NULL,
    title TEXT NOT NULL,
    recommendation_text TEXT NOT NULL,
    status TEXT DEFAULT 'new',
    created_at TEXT NOT NULL,
    FOREIGN KEY (id_business) REFERENCES businesses(id_business)
);

CREATE TABLE reports (
    id_report INTEGER PRIMARY KEY AUTOINCREMENT,
    id_business INTEGER NOT NULL,
    id_user INTEGER NOT NULL,
    report_type TEXT NOT NULL,
    period_start TEXT NOT NULL,
    period_end TEXT NOT NULL,
    file_path TEXT,
    created_at TEXT NOT NULL,
    FOREIGN KEY (id_business) REFERENCES businesses(id_business),
    FOREIGN KEY (id_user) REFERENCES users(id_user)
);

CREATE INDEX idx_transactions_business_date
ON financial_transactions(id_business, transaction_date);

CREATE INDEX idx_metrics_business_date
ON operational_metrics(id_business, metric_date);

CREATE INDEX idx_recommendations_business
ON recommendations(id_business);

CREATE INDEX idx_reports_business
ON reports(id_business);
);

```

ДОДАТОК Б

Фрагмент програмного коду розрахунку KPI

```

from dataclasses import dataclass

@dataclass
class BusinessKPI:
    revenue: float
    expenses: float
    profit: float
    margin: float
    conversion: float
    average_check: float
    cac: float
    return_rate: float

class KPIService:
    def __init__(self, db):
        self.db = db

    def calculate_kpi(self, business_id: int, date_from: str, date_to: str) ->
    BusinessKPI:
        revenue = self.db.scalar(
            """
            SELECT COALESCE(SUM(amount), 0)
            FROM financial_transactions
            WHERE id_business = ?
              AND transaction_type = 'revenue'
              AND transaction_date BETWEEN ? AND ?
            """,
            (business_id, date_from, date_to)
        )

        expenses = self.db.scalar(
            """
            SELECT COALESCE(SUM(amount), 0)
            FROM financial_transactions
            WHERE id_business = ?
              AND transaction_type = 'expense'
              AND transaction_date BETWEEN ? AND ?
            """,
            (business_id, date_from, date_to)
        )

        metrics = self.db.fetchone(
            """
            SELECT
              COALESCE(SUM(clients), 0) AS clients,
              COALESCE(SUM(orders), 0) AS orders,
              COALESCE(SUM(leads), 0) AS leads,
              COALESCE(SUM(successful_deals), 0) AS successful_deals,
              COALESCE(SUM(returns_count), 0) AS returns_count,
              COALESCE(SUM(advertising_cost), 0) AS advertising_cost
            FROM operational_metrics
            WHERE id_business = ?
              AND metric_date BETWEEN ? AND ?
            """,
            (business_id, date_from, date_to)
        )

```

```

)

profit = revenue - expenses

if revenue > 0:
    margin = profit / revenue * 100
else:
    margin = 0

if metrics["leads"] > 0:
    conversion = metrics["successful_deals"] / metrics["leads"] * 100
else:
    conversion = 0

if metrics["orders"] > 0:
    average_check = revenue / metrics["orders"]
else:
    average_check = 0

if metrics["successful_deals"] > 0:
    cac = metrics["advertising_cost"] / metrics["successful_deals"]
else:
    cac = 0

if metrics["orders"] > 0:
    return_rate = metrics["returns_count"] / metrics["orders"] * 100
else:
    return_rate = 0

return BusinessKPI(
    revenue=round(revenue, 2),
    expenses=round(expenses, 2),
    profit=round(profit, 2),
    margin=round(margin, 2),
    conversion=round(conversion, 2),
    average_check=round(average_check, 2),
    cac=round(cac, 2),
    return_rate=round(return_rate, 2)
)

```

ДОДАТОК В

Фрагмент програмного коду формування звіту та рекомендацій

```

from datetime import datetime
from pathlib import Path

class RecommendationService:
    def __init__(self, db):
        self.db = db

    def build_recommendations(self, business_id: int, kpi, targets: dict) ->
list[dict]:
        recommendations = []

        if kpi.profit < targets.get("profit", 0):
            recommendations.append({
                "priority": "warning",
                "title": "Посилити контроль прибутковості",
                "text": (
                    "Прибуток нижчий за плановий рівень. "
                    "Доцільно переглянути закупівельні ціни, частку постійних
витрат "
                    "і маржинальність основних категорій."
                )
            })

        if kpi.margin < targets.get("margin", 0):
            recommendations.append({
                "priority": "critical",
                "title": "Оптимізувати витрати",
                "text": (
                    "Маржинальність нижча за цільове значення. "
                    "Рекомендовано перевірити витрати на оренду, закупівлі,
маркетинг "
                    "та адміністративні платежі."
                )
            })

        if kpi.conversion < targets.get("conversion", 0):
            recommendations.append({
                "priority": "info",
                "title": "Підвищити конверсію продажів",
                "text": (
                    "Конверсія лідів в угоди нижча за план. "
                    "Рекомендовано покращити обробку заявок, скрипти продажів "
                    "і швидкість відповіді менеджера."
                )
            })

        if kpi.average_check < targets.get("average_check", 0):
            recommendations.append({
                "priority": "info",
                "title": "Збільшити середній чек",
                "text": (
                    "Доцільно застосовувати комплекти товарів, додаткові
послуги, "
                    "персональні пропозиції та рекомендації супутніх позицій."
                )
            })

```

```

        for item in recommendations:
            self.db.execute(
                """
                INSERT INTO recommendations(
                    id_business, priority, title, recommendation_text, status,
created_at
                )
                VALUES (?, ?, ?, ?, 'new', ?)
                """,
                (
                    business_id,
                    item["priority"],
                    item["title"],
                    item["text"],
                    datetime.now().strftime("%Y-%m-%d %H:%M:%S")
                )
            )

        return recommendations

class ReportService:
    def __init__(self, db):
        self.db = db

    def build_text_report(self, business_id: int, user_id: int, date_from: str,
date_to: str, kpi, recommendations: list[dict]) -> str:
        lines = []

        lines.append("ЗВІТ ПРО ЕФЕКТИВНІСТЬ УПРАВЛІННЯ МАЛИМ БІЗНЕСОМ")
        lines.append(f"Період: {date_from} - {date_to}")
        lines.append("")
        lines.append(f"Дохід: {kpi.revenue:.2f} грн")
        lines.append(f"Витрати: {kpi.expenses:.2f} грн")
        lines.append(f"Прибуток: {kpi.profit:.2f} грн")
        lines.append(f"Маржинальність: {kpi.margin:.2f} %")
        lines.append(f"Середній чек: {kpi.average_check:.2f} грн")
        lines.append(f"Конверсія: {kpi.conversion:.2f} %")
        lines.append(f"Рівень повернень: {kpi.return_rate:.2f} %")
        lines.append(f"CAC: {kpi.cac:.2f} грн")
        lines.append("")
        lines.append("Управлінські рекомендації:")

        if not recommendations:
            lines.append("Критичних відхилень не виявлено.")
        else:
            for item in recommendations:
                lines.append(f"[{item['priority']}] {item['title']}")
                lines.append(item["text"])

        report_text = "\n".join(lines)

        export_dir = Path("reports")
        export_dir.mkdir(exist_ok=True)

        filename =
f"business_report_{datetime.now().strftime('%Y%m%d_%H%M%S')}.txt"
        path = export_dir / filename
        path.write_text(report_text, encoding="utf-8")

        self.db.execute(
            """

```

```

        INSERT INTO reports(
            id_business, id_user, report_type, period_start, period_end,
file_path, created_at
        )
        VALUES (?, ?, 'TXT', ?, ?, ?, ?)
        """
        (
            business_id,
            user_id,
            date_from,
            date_to,
            str(path),
            datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        )
    )

    return report_text

```

ДОДАТОК Г

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Факультет інформаційних технологій

Декларація про академічну доброчесність та використання ШІ

ДЕКЛАРАЦІЯ ЗДОБУВАЧА

Я, Шаговий Юрій, здобувач(ка) НУБіП України, усвідомлюючи відповідальність за порушення академічної доброчесності, цією заявою підтверджую, що:

1. Моя бакалаврська кваліфікаційна робота (проект) на тему «Програмний застосунок для аналізу ефективності управління малим бізнесом» є результатом моєї особистої праці.
2. Усі запозичення з текстів інших авторів мають відповідні посилання згідно з чинними стандартами.
3. Щодо використання інструментів ШІ зазначаю, що (обрати необхідне):
 - ☐ інструменти генеративного ШІ не використовувалися.
 - ☐ ☒ інструменти ШІ (вказати назву: ChatGPT, Gemini, Claude, DeepL, _____ інші (вказати назву)) були використані для:
 - ☐ перекладу іншомовних джерел;
 - ☐ стилістичного редагування та перевірки граматики;
 - ☐ допомоги у написанні/відлагодженні програмного коду;
 - ☐ інше: _____.

Я розумію, що надання недостовірної інформації може призвести до скасування результатів захисту та відрахування з числа здобувачів НУБіП України.

«___» _____
2026 р.

(підпис) **Юрій ШАГОВИЙ**

