

КВАЛІФІКАЦІЙНА РОБОТА

Інформаційна система управління малим бізнесом з урахуванням динаміки
замовлень та ресурсів
Назва теми

Рівень вищої освіти перший (бакалаврський)

Галузь знань 12 «Інформаційні технології»

Шифр, назва

Спеціальність 126 «Інформаційні системи та технології»

Шифр, назва

Освітня програма «Інформаційні системи та технології»

Назва

Шифр КвРІСТ 220180.22.01.03 ПЗ

Виконав здобувач IV курсу, група ІСТ-22-1


Підпис

АНТОН РАБЧЕНЮК

Ініціали, прізвище


Керівник доктор техн. наук, професор
Науковий ступінь, учене звання


Підпис

ЄЛИЗАВЕТА ГНАТЧУК

Ініціали, прізвище

Нормоконтролер канд. фіз.-мат. наук, доц.
Науковий ступінь, учене звання


Підпис

ТЕТЯНА КИСІЛЬ

Ініціали, прізвище

До захисту допускаю:
завідувач кафедри КІС


Підпис

ОЛЬГА ПАВЛОВА

Ініціали, прізвище

«01» червня 2026 р.

дата

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

Рівень вищої освіти ПЕРШИЙ (БАКАЛАВРСЬКИЙ)

Галузь знань 12 ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Спеціальність 126 ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Освітня програма «ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ»

ЗАТВЕРДЖУЮ

Завідувачка кафедри КІС



Ольга ПАВЛОВА

“ 10 ” 01 2026 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Рабченко Антон Ігорович

Прізвище, ім'я, по батькові студента

1. Тема проекту (роботи) Інформаційна система управління малим бізнесом з урахуванням динаміки замовлень та ресурсів

Керівник проекту (роботи) Гнатчук Єлизавета Геннадіївна, д.т.н., проф.

Прізвище, ім'я, по батькові, науковий ступінь, вчене звання

Затверджена наказом ректора університету від 01.03.2026 р. № 5

2. Термін подання здобувачем роботи на кафедру 01.06.2026 р.

3. Вихідні дані до роботи Завдання на кваліфікаційну роботу

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

Аналіз предметної області управління малим бізнесом та постановка задачі розроблення інформаційної системи

Проектування інформаційної системи управління малим бізнесом

Програмна реалізація та тестування інформаційної системи управління малим бізнесом

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслень) _____

Результати роботи системи

Алгоритм функціонування

Схема взаємодії модулів

6. Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання « 10 » 01 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів (розділів) дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітка
1	Вибір напрямку дослідження та узгодження тематики кваліфікаційної роботи з керівником	10.01.2026	виконано
2	Ознайомлення з предметною областю; формулювання мети та задач дослідження; визначення об'єкта та предмета дослідження	01.02.2026	виконано
3	Робота над розділом 1 – дослідження предметної області та постановка задачі	01.03.2026	виконано
4	Робота над розділом 2 – вибір компонентів для проектування інформаційної системи управління малим бізнесом	01.04.2026	виконано
5	Робота над розділом 3 – проектування програмної реалізації та тестування інформаційної системи малим бізнесом	29.04.2026	виконано
6	Оформлення пояснювальної записки згідно вимог	25.05.2026	виконано
7	Попередній захист ВКР	26.05.2025	виконано
8	Захист ВКР на засіданні ЕК	Червень 2026 року	

Здобувач  Антон РАБЧЕНЮК
Підпис Імя, ПРІЗВИЩЕ

Керівник кваліфікаційної роботи  Єлизавета ГНАТЧУК
Підпис Імя, ПРІЗВИЩЕ

АНОТАЦІЯ

Тема кваліфікаційної роботи: «Інформаційна система управління малим бізнесом з урахуванням динаміки замовлень та ресурсів».

Автор роботи:

Керівник роботи:

Пояснювальна записка: 72 с., рис., табл., 3 дод., 45 джерел.

Графічна частина: 3 креслення.

АЛГОРИТМИ, БАЗА ДАНИХ, ЗАМОВЛЕННЯ, ІНФОРМАЦІЙНА СИСТЕМА, КЛІЄНТ-СЕРВЕРНА АРХІТЕКТУРА, МАЛИЙ БІЗНЕС, РЕСУРСИ, УПРАВЛІННЯ.

Кваліфікаційна робота бакалавра присвячена розробленню інформаційної системи управління малим бізнесом з урахуванням динаміки замовлень та ресурсів. Актуальність теми зумовлена потребою малих підприємств в автоматизації обліку клієнтів, замовлень, ресурсів, оплат і звітної інформації. У роботі розглянуто проблему ручного ведення даних, несвоєчасного оновлення залишків, складності контролю активних замовлень і ризику прийняття замовлень без фактичної наявності необхідних ресурсів.

Метою роботи є проектування, реалізація та тестування інформаційної системи, яка забезпечує автоматизацію основних процесів малого бізнесу, зокрема творення замовлень, перевірку доступності ресурсів, резервування, списання, облік оплат і формування звітів. Для досягнення поставленої мети було проаналізовано предметну область, визначено функціональні та нефункціональні вимоги, спроектовано архітектуру системи, модель бази даних і алгоритми оброблення замовлень та ресурсів.

У межах роботи реалізовано web-додаток із клієнт-серверною архітектурою. Проведене тестування підтвердило працездатність системи та коректність виконання основних сценаріїв роботи.







Підпис здобувача

30.05.2026

Дата

ЗМІСТ

Вступ.....	3
1 Аналіз предметної області управління малим бізнесом та постановка задачі розроблення інформаційної системи	5
1.1 Особливості управління малим бізнесом в умовах змінного попиту	5
1.2 Аналіз процесів оброблення замовлень і використання ресурсів.....	9
1.3 Огляд існуючих інформаційних систем для малого бізнесу	14
1.4 Постановка задачі розроблення інформаційної системи.....	19
1.5 Висновки до першого розділу.....	20
2 Проктування інформаційної системи управління малим бізнесом	22
2.1 Визначення функціональних і нефункціональних вимог до системи ...	22
2.2 Проктування архітектури інформаційної системи	27
2.3 Проктування бази даних системи	34
2.4 Проктування алгоритмів оброблення замовлень і ресурсів	40
2.5 Висновки до другого розділу.....	45
3 Програмна реалізація та тестування інформаційної системи управління малим бізнесом	47
3.1 Обґрунтування вибору програмних засобів реалізації.....	47
3.2 Реалізація серверної частини інформаційної системи	51
3.3 Реалізація бази даних	55
3.4 Реалізація клієнтської частини системи.....	60
3.5 Реалізація модулів управління замовленнями та ресурсами	63
3.7 Висновки до третього розділу	71
Висновки.....	74
Перелік джерел посилань	76
Додаток А Результати роботи системи.....	80
Додаток Б Алгоритм функціонування	81
Додаток В Схема взаємодії модулів.....	82

КвРІСТ.220180.22.01.03 ПЗ				
Зм.	Арк.	№ док.ум.	Підпис	Дата
Виконав		Рибченко Антон		
Перевір.		Гнатюк Єлизавета		
Н.контр.		Гетяна КИСІЦЬ		
Затвер.		Ольга ПАВЛЮВА		
Інформаційна система управління малим бізнесом з урахуванням динаміки замовлень та ресурсів			Літера	Аркуш
			у	2
			ХНУ ІСТ-22-1	

ВСТУП

Сучасний малий бізнес працює в умовах постійної зміни попиту, обмежених ресурсів та високої конкуренції. Для невеликих підприємств важливо швидко приймати замовлення, контролювати їх виконання, стежити за залишками товарів або матеріалів, а також своєчасно реагувати на нестачу ресурсів. Якщо ці процеси виконуються вручну або за допомогою окремих таблиць, виникають помилки, дублювання інформації, затримки в обробленні замовлень і складність у прийнятті управлінських рішень.

Інформаційні системи дають змогу автоматизувати основні процеси малого підприємства та об'єднати дані про клієнтів, замовлення, ресурси, платежі й звіти в одному середовищі. Завдяки цьому керівник або адміністратор може швидше отримувати актуальну інформацію, контролювати стан виконання замовлень і бачити реальну ситуацію з ресурсами підприємства. Це особливо важливо тоді, коли кількість замовлень змінюється нерівномірно: у певні періоди виникає підвищене навантаження, а в інші - зменшується потреба в частині ресурсів.

Проблема управління замовленнями та ресурсами полягає в тому, що ці процеси тісно пов'язані між собою. Нове замовлення потребує перевірки наявності товарів, матеріалів, працівників або часу для виконання. Якщо ресурсів недостатньо, підприємство може не виконати замовлення вчасно або втратити клієнта. З іншого боку, надлишкові запаси також створюють проблему, оскільки вони збільшують витрати та заморожують кошти підприємства.

У зв'язку з цим актуальним є розроблення інформаційної системи управління малим бізнесом, яка враховує динаміку замовлень і поточний стан ресурсів. Така система повинна забезпечувати облік клієнтів, створення та супровід замовлень, контроль ресурсів, перегляд залишків, формування звітів і підтримку прийняття управлінських рішень.

					КвРІСТ.220180.22.01.03 ПЗ	Арк.
						4
Зм.	Арк.	№ докум.	Підпис	Дата		

Метою кваліфікаційної роботи є аналіз проблеми, проєктування та реалізація інформаційної системи управління малим бізнесом з урахуванням динаміки замовлень та ресурсів для автоматизації обліку замовлень, контролю ресурсів і підвищення ефективності управління підприємством шляхом візуалізації та структуризації процесів бізнесу.

Об'єктом дослідження є процеси управління замовленнями та ресурсами малого бізнесу.

Предметом дослідження є інформаційна система управління малим бізнесом з урахуванням динаміки замовлень та ресурсів.

Для досягнення поставленої мети необхідно виконати такі завдання:

1. Проаналізувати особливості управління малим бізнесом в умовах змінного попиту.
2. Дослідити процеси оброблення замовлень і використання ресурсів.
3. Визначити основні вимоги до інформаційної системи.
4. Спроекувати архітектуру інформаційної системи.
5. Розробити структуру бази даних, реалізувати основні модулі системи та провести тестування її роботи.

Практичне значення роботи полягає в тому, що розроблена інформаційна система може бути використана малим підприємством для автоматизації щоденних управлінських процесів. Її застосування дозволяє зменшити кількість ручних операцій, скоротити час оброблення замовлень, покращити контроль ресурсів і підвищити зручність формування звітної інформації.

Кваліфікаційна робота складається зі вступу, трьох розділів, висновків, переліку джерел посилань і додатків. У першому розділі розглядається предметна область та виконується постановка задачі. У другому розділі здійснюється проєктування інформаційної системи. У третьому розділі описується програмна реалізація та тестування розробленої системи.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 5
Зм.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ УПРАВЛІННЯ МАЛИМ БІЗНЕСОМ ТА ПОСТАНОВКА ЗАДАЧІ РОЗРОБЛЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

1.1 Особливості управління малим бізнесом в умовах змінного попиту

Малий бізнес є важливою складовою економічної системи, оскільки забезпечує зайнятість населення, створює товари та послуги для локального ринку, швидко реагує на потреби споживачів і часто працює ближче до клієнта, ніж великі підприємства. Водночас саме малі підприємства найбільш чутливі до змін попиту, коливань цін, нестачі ресурсів, сезонності продажів і помилок в організації внутрішніх процесів. На відміну від великих компаній, малий бізнес зазвичай не має складної управлінської структури, великого штату працівників або значних фінансових резервів, тому кожне управлінське рішення прямо впливає на стабільність його роботи [1].

Управління малим бізнесом має свої особливості. Власник або керівник часто одночасно виконує декілька функцій: контролює замовлення, спілкується з клієнтами, організовує закупівлі, розподіляє завдання між працівниками, перевіряє оплату та аналізує фінансовий результат. Через це значна частина управлінської інформації може зберігатися не в єдиній системі, а в окремих таблицях, повідомленнях, блокнотах або усних домовленостях. На початковому етапі діяльності такий підхід може здаватися достатнім, але зі збільшенням кількості клієнтів і замовлень він поступово створює проблеми.

Однією з головних проблем малого бізнесу є змінний попит. Кількість замовлень може суттєво відрізнятись залежно від сезону, дня тижня, рекламної активності, економічної ситуації, поведінки клієнтів або наявності конкурентів. Наприклад, підприємство у сфері торгівлі може мати підвищене навантаження перед святами, а сервісна компанія - у періоди активного використання певних послуг. Якщо підприємство не має інструментів для відстеження такої динаміки,

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 6
Зм.	Арк.	№ докум.	Підпис	Дата		

воно не завжди може правильно підготуватися до збільшення або зменшення кількості замовлень [2].

Змінний попит безпосередньо впливає на використання ресурсів. Якщо кількість замовлень різко зростає, підприємству може не вистачити товарів, матеріалів, обладнання, працівників або часу для якісного виконання всіх заявок. У такому випадку виникають затримки, знижується якість обслуговування, збільшується кількість помилок і зростає ризик втрати клієнтів. Якщо ж попит зменшується, підприємство може мати надлишкові запаси, невикористані робочі години або зайві витрати на зберігання ресурсів. Тому управління попитом і ресурсами має розглядатися як єдиний взаємопов'язаний процес.

Для малого бізнесу важливо не лише зафіксувати факт отримання замовлення, а й одразу оцінити можливість його виконання. Потрібно знати, чи є в наявності необхідні товари або матеріали, чи доступні працівники, який строк виконання можна запропонувати клієнту, чи не вплине нове замовлення на вже прийняті зобов'язання. Якщо ці дані відсутні або розміщені в різних джерелах, керівник змушений приймати рішення на основі приблизної оцінки. Це збільшує ризик помилок і знижує якість управління [4].

Основні труднощі управління малим бізнесом в умовах змінного попиту можна подати у вигляді схеми.

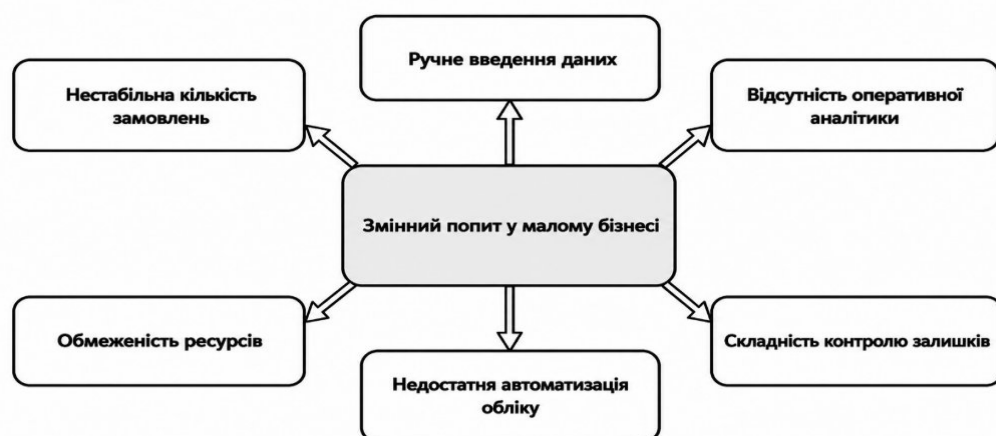


Рисунок 1.1 – Основні труднощі управління малим бізнесом в умовах змінного попиту

До таких труднощів належать: нестабільна кількість замовлень, обмеженість ресурсів, недостатня автоматизація обліку, складність контролю залишків, відсутність оперативної аналітики та залежність від ручного введення даних. Кожен із цих чинників окремо може погіршувати роботу підприємства, але найбільша проблема виникає тоді, коли вони проявляються одночасно. Наприклад, якщо кількість замовлень зростає, а залишки ресурсів не оновлюються вчасно, підприємство може прийняти більше замовлень, ніж реально здатне виконати.

Особливістю малого бізнесу є те, що управлінські процеси часто є менш формалізованими, ніж у великих компаніях. Це означає, що правила оброблення замовлень, списання ресурсів, контролю оплати або формування звітності можуть не бути чітко описаними. У багатьох випадках працівники виконують ці дії за звичкою або за усними інструкціями. Такий підхід робить підприємство гнучким, але одночасно створює ризики. Якщо працівник помиляється, забуває оновити дані або неправильно передає інформацію іншому працівнику, це може призвести до порушення строків виконання замовлення.

Інформаційна система в таких умовах може виконувати роль інструмента впорядкування діяльності підприємства. Вона дозволяє зберігати основні дані в одному середовищі, фіксувати зміни в замовленнях, контролювати залишки ресурсів, відстежувати статуси виконання та формувати звіти. При цьому система не замінює керівника, а допомагає йому швидше отримувати потрібну інформацію для прийняття рішень. Особливо корисною така система є тоді, коли підприємство має декілька напрямів діяльності, декількох працівників або значну кількість повторюваних операцій.

Для ефективного управління малим бізнесом інформація повинна бути актуальною, структурованою та доступною для користувачів відповідно до їхніх ролей. Наприклад, адміністратор повинен бачити нові замовлення та контакти клієнтів, працівник - свої завдання, керівник - загальну статистику, а відповідальна особа за склад - залишки ресурсів. Якщо всі ці дані поєднані в

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 8
Зм.	Арк.	№ докум.	Підпис	Дата		

межах однієї системи, підприємство може швидше реагувати на зміни та краще планувати свою роботу.

Управління замовленнями є одним із центральних процесів малого підприємства. Замовлення відображає реальну потребу клієнта і водночас створює навантаження на ресурси підприємства. Тому під час роботи із замовленням важливо контролювати його статус: нове, підтверджене, у роботі, виконане, скасоване або оплачене. Такий підхід дає змогу уникнути плутанини та швидко визначити, на якому етапі перебуває кожне звернення клієнта.

Не менш важливим є контроль ресурсів. Ресурси малого бізнесу можуть бути матеріальними, трудовими, фінансовими, часовими та інформаційними. Матеріальні ресурси включають товари, сировину, комплектуючі або обладнання. Трудові ресурси пов'язані з працівниками та їхнім робочим часом. Фінансові ресурси визначають можливість закупівель і покриття витрат. Часові ресурси впливають на строки виконання замовлень, а інформаційні ресурси включають дані про клієнтів, постачальників, ціни, залишки[3].

Узагальнено взаємозв'язок між замовленнями та ресурсами можна описати так: кожне нове замовлення створює потребу в певному обсязі ресурсів, а стан ресурсів визначає можливість виконання цього замовлення. Якщо система здатна автоматично або напівавтоматично показувати доступність ресурсів, це значно спрощує роботу керівника або адміністратора. У такому випадку рішення про прийняття замовлення приймається не інтуїтивно, а на основі фактичних даних.

Управління малим бізнесом в умовах змінного попиту потребує оперативного обліку, контролю та аналізу інформації. Основними проблемами є нестабільність замовлень, обмеженість ресурсів, ручне ведення даних і недостатня прозорість внутрішніх процесів.

					КвРІСТ.220180.22.01.03 ПЗ	Арк.
						9
Зм.	Арк.	№ докум.	Підпис	Дата		

1.2 Аналіз процесів оброблення замовлень і використання ресурсів

Процес оброблення замовлень є одним із основних процесів у діяльності малого бізнесу. Саме через замовлення підприємство отримує дохід, планує завантаження працівників, використовує матеріальні ресурси та підтримує взаємодію з клієнтами. Якщо цей процес організований неправильно, виникають затримки, помилки в обліку, нестача ресурсів або непорозуміння з клієнтами. Тому для малого підприємства важливо мати чітку послідовність дій від моменту отримання замовлення до його повного виконання.

Зазвичай оброблення замовлення починається з отримання заявки від клієнта. Замовлення може надходити через телефонний дзвінок, повідомлення в месенджері, форму на сайті, соціальні мережі або безпосередньо в точці продажу. На цьому етапі фіксуються основні дані: ім'я або назва клієнта, контактна інформація, перелік товарів чи послуг, кількість, бажаний строк виконання, спосіб оплати та додаткові побажання. Якщо ці дані не заносяться до єдиної системи, частина інформації може бути втрачена або неправильно передана іншим працівникам [4].

Після отримання заявки підприємство повинно перевірити можливість її виконання. Для цього необхідно визначити, чи є в наявності потрібні товари, матеріали, обладнання або працівники. Наприклад, якщо мале підприємство займається продажем товарів, система повинна перевірити залишки на складі. Якщо підприємство надає послуги, важливо оцінити завантаження працівників і наявність вільного часу. На практиці саме цей етап часто є проблемним, оскільки інформація про ресурси може бути неактуальною або розміщеною в різних документах. Життєвий цикл замовлення в інформаційній системі можна подати як послідовність взаємопов'язаних етапів.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 10
Зм.	Арк.	№ докум.	Підпис	Дата		



Рисунок 1.2 – Життєвий цикл замовлення в малому бізнесі

До основних етапів життєвого циклу замовлення належать: створення замовлення, перевірка ресурсів, підтвердження замовлення, резервування ресурсів, виконання, оплата та закриття замовлення. Така структура дозволяє краще контролювати кожне замовлення та швидко визначати його поточний стан. Для інформаційної системи це особливо важливо, оскільки кожен статус замовлення може бути пов'язаний із певними діями користувача або автоматичними змінами в базі даних.

Після перевірки доступності ресурсів замовлення може бути підтвержене або відхилене. Якщо ресурсів достатньо, система повинна дозволити перевести замовлення в активний статус і зарезервувати потрібні ресурси. Резервування означає, що певна кількість товару, матеріалу або часу працівника тимчасово закріплюється за конкретним замовленням. Це дозволяє уникнути ситуації, коли один і той самий ресурс помилково використовується для кількох замовлень одночасно.

Якщо ресурсів недостатньо, інформаційна система повинна повідомити користувача про проблему. Наприклад, система може показати, яких саме матеріалів не вистачає, скільки одиниць потрібно дозамовити або який строк виконання є більш реалістичним. У такому випадку керівник або адміністратор може прийняти одне з кількох рішень: відкласти виконання замовлення, змінити

його склад, погодити інший строк із клієнтом або створити заявку на поповнення ресурсів [5].

Виконання замовлення є наступним етапом. На цьому етапі працівники виконують необхідні дії: комплектують товар, виготовляють продукцію, надають послугу або готують замовлення до передачі клієнту. Інформаційна система повинна відображати зміну статусу замовлення, щоб користувачі бачили, на якому етапі воно перебуває. Наприклад, замовлення може мати статуси «нове», «підтверджене», «у роботі», «очікує оплати», «виконане» або «скасоване».

Окремим важливим елементом є облік оплати. Для малого бізнесу важливо розуміти, які замовлення вже оплачені, які очікують оплати, а які мають часткову оплату. Якщо інформація про оплату не пов'язана із замовленням, підприємство може мати труднощі з контролем фінансового стану. Тому в інформаційній системі доцільно передбачити зв'язок між замовленнями, платежами та статусами виконання.

Після виконання всіх дій замовлення закривається. Закрите замовлення не повинно зникати з системи, оскільки воно є частиною історії роботи підприємства. Такі дані можуть використовуватися для аналізу попиту, визначення популярних товарів або послуг, оцінювання активності клієнтів і планування ресурсів. Наприклад, якщо за певний період система показує зростання кількості замовлень на конкретний товар, підприємство може заздалегідь збільшити його запас [6].

Ресурси малого бізнесу можна поділити на кілька основних груп. До матеріальних ресурсів належать товари, сировина, комплектуючі, пакувальні матеріали та обладнання. До трудових ресурсів належать працівники, їхній робочий час і професійні можливості.

Таблиця 1.1 – Основні ресурси малого бізнесу та їх роль у виконанні замовлень

Вид ресурсу	Приклад	Роль у виконанні замовлення
Матеріальні ресурси	Товари, сировина, комплектуючі, обладнання	Забезпечують фізичне виконання замовлення
Трудові ресурси	Працівники, виконавці, адміністратори	Виконують операції, пов'язані із замовленням
Фінансові ресурси	Кошти, оплати, витрати	Забезпечують закупівлі та покриття витрат
Часові ресурси	Строки, графік роботи, робочий час	Визначають можливість виконання замовлення вчасно
Інформаційні ресурси	Дані про клієнтів, залишки, постачальників	Допомагають приймати управлінські рішення

Зв'язок між замовленнями та ресурсами є двостороннім. З одного боку, кожне нове замовлення створює потребу в певних ресурсах. З іншого боку, фактична наявність ресурсів визначає, чи може підприємство прийняти це замовлення та виконати його у потрібний строк. Саме тому інформаційна система повинна не лише зберігати інформацію про замовлення, а й автоматично враховувати їхній вплив на ресурси підприємства.

Наприклад, якщо клієнт замовляє товар у кількості 10 одиниць, а на складі є 15 одиниць, система може дозволити підтвердження замовлення та зарезервувати потрібну кількість. Після цього доступний залишок для інших замовлень становитиме вже не 15, а 5 одиниць. Якщо інший клієнт спробує замовити 8 одиниць цього самого товару, система повинна показати, що доступного залишку недостатньо. Такий підхід дозволяє уникнути помилок і підвищує надійність управління ресурсами.

У разі ручного обліку така перевірка часто виконується із запізненням. Працівник може прийняти замовлення, не знаючи, що потрібний товар уже

зарезервований для іншого клієнта. Також можуть виникати помилки через несвоєчасне оновлення таблиць або відсутність єдиних правил ведення обліку. Для малого бізнесу навіть одна така помилка може мати негативні наслідки, оскільки клієнт очікує швидкого й точного виконання замовлення.

Інформаційна система повинна підтримувати не тільки поточний облік, а й аналіз динаміки замовлень. Під динамікою замовлень можна розуміти зміну їх кількості, вартості, складу та строків виконання протягом певного періоду. Такий аналіз дозволяє визначати періоди найбільшого навантаження, популярні товари або послуги, середню кількість замовлень за день, тиждень чи місяць. Ці дані можуть бути використані для планування закупівель, графіка роботи персоналу та оцінювання ефективності діяльності підприємства.

Важливим результатом автоматизації процесів оброблення замовлень і використання ресурсів є підвищення прозорості управління. Керівник отримує можливість бачити не окремі фрагменти інформації, а повну картину: які замовлення активні, які ресурси зайняті, які товари закінчуються, які клієнти найчастіше звертаються, які періоди є найбільш завантаженими. Це дозволяє приймати більш обґрунтовані рішення та краще планувати діяльність підприємства.

Отже, процеси оброблення замовлень і використання ресурсів у малому бізнесі тісно пов'язані між собою. Ефективна інформаційна система повинна забезпечувати реєстрацію замовлень, перевірку доступності ресурсів, резервування, контроль виконання, облік оплати, закриття замовлення та подальший аналіз накопичених даних. Саме така логіка дозволяє зменшити кількість помилок, прискорити роботу підприємства та підвищити якість управління в умовах змінного попиту.

Автоматизація цих процесів дає можливість своєчасно виявляти дефіцит ресурсів та прогнозувати навантаження. У результаті підприємство отримує інструмент для ефективного управління своєю діяльністю, що позитивно впливає на його продуктивність, прибутковість і конкурентоспроможність.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 14
Зм.	Арк.	№ докум.	Підпис	Дата		

1.3 Огляд існуючих інформаційних систем для малого бізнесу

Для автоматизації діяльності малого бізнесу можуть використовуватися різні типи інформаційних систем. Найчастіше це CRM-системи, ERP-системи, складські програми, POS-системи, сервіси для обліку продажів, бухгалтерські програми та електронні таблиці. Кожен із цих інструментів має своє призначення, переваги й обмеження. Тому перед розробленням власної інформаційної системи доцільно проаналізувати існуючі підходи та визначити, які функції є найбільш важливими для малого підприємства.

Одним із найбільш поширених видів програмного забезпечення для малого бізнесу є CRM-системи. CRM-система призначена для управління взаємодією з клієнтами. Вона дозволяє зберігати контактні дані клієнтів, історію звернень, інформацію про угоди, задачі менеджерів і результати продажів. Для малого бізнесу така система є корисною, оскільки допомагає не втрачати клієнтів, контролювати етапи роботи з ними та аналізувати ефективність продажів [7].

Разом із тим CRM-системи не завжди повністю вирішують задачу управління ресурсами. Наприклад, вони можуть добре показувати, на якому етапі перебуває замовлення або угода, але не завжди мають зручний механізм перевірки залишків матеріалів, резервування ресурсів або оцінювання завантаження працівників. Через це для підприємств, де важливо одночасно контролювати і замовлення, і ресурси, однієї CRM-системи може бути недостатньо.

Іншим типом програмних рішень є ERP-системи. ERP-система призначена для комплексного управління ресурсами підприємства. Вона може охоплювати облік товарів, склад, закупівлі, фінанси, виробництво, персонал, документообіг та аналітику. Такі системи мають широкую функціональність і дозволяють об'єднати багато бізнес-процесів в одному програмному середовищі [8].

Однак для малого бізнесу ERP-системи часто є занадто складними. Їх впровадження потребує часу, налаштування, навчання персоналу та фінансових витрат. Якщо підприємство має невелику кількість працівників і просту

структуру процесів, використання повноцінної ERP-системи може бути недоцільним. У такому випадку більш практичним є створення або використання спрощеної інформаційної системи, яка охоплює тільки необхідні функції: облік замовлень, контроль ресурсів, роботу з клієнтами та формування звітів.

Окрему групу становлять складські системи. Вони орієнтовані на облік товарів, матеріалів, залишків, надходжень і списань. Такі системи добре підходять для підприємств, які займаються торгівлею або мають постійний рух товарів. Складська система дозволяє бачити залишки, контролювати дефіцит, формувати документи на надходження та витрати, а також аналізувати рух товарів за певний період.

Недоліком складських систем є те, що вони не завжди охоплюють повний цикл роботи із замовленням. Наприклад, система може показувати наявність товару на складі, але не враховувати статус замовлення, історію взаємодії з клієнтом, оплату або завантаження працівників.

Для підприємств торгівлі також часто використовуються POS-системи. Вони застосовуються у магазинах, кафе, сервісних точках та інших місцях, де потрібно швидко оформити продаж. POS-система може працювати з касовим обладнанням, фіксувати продажі, формувати чеки, враховувати оплату та передавати дані до облікової системи. Для малого бізнесу це зручно, якщо основний процес пов'язаний із прямими продажами.

Проте POS-системи зазвичай орієнтовані саме на момент продажу, а не на повне управління підприємством. Вони можуть бути менш зручними для тих випадків, коли замовлення має довгий життєвий цикл: створення заявки, погодження, резервування ресурсів, виконання, часткова оплата, завершення та подальший аналіз. Тому POS-система може бути лише одним із компонентів загального інформаційного середовища, але не завжди замінює повноцінну систему управління замовленнями.

На практиці малий бізнес часто використовує електронні таблиці, наприклад Microsoft Excel або Google Sheets. Їх популярність пояснюється

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 16
Зм.	Арк.	№ докум.	Підпис	Дата		

простотою, доступністю та зрозумілим інтерфейсом. У таблицях можна вести список клієнтів, замовлень, товарів, оплат і залишків. На початковому етапі це може бути зручним рішенням, особливо якщо підприємство має невелику кількість операцій.

Однак електронні таблиці мають суттєві обмеження. По-перше, у них складно забезпечити цілісність даних. Один користувач може випадково змінити формулу, видалити рядок або внести дані в неправильному форматі. По-друге, таблиці не завжди зручно використовувати одночасно кільком працівникам. По-третє, у таблицях складно реалізувати повноцінну систему ролей, автоматичну перевірку залишків, резервування ресурсів і контроль статусів замовлень. Через це зі збільшенням обсягу даних електронні таблиці поступово стають джерелом помилок.

Порівняння основних типів інформаційних систем для малого бізнесу наведено в таблиці 1.2.

Таблиця 1.2 – Порівняння існуючих інформаційних систем для малого бізнесу

Тип системи	Основне призначення	Переваги	Недоліки
CRM-система	Управління клієнтами та продажами	Зручна робота з клієнтами, історія звернень, контроль угод	Не завжди забезпечує повноцінний контроль ресурсів
ERP-система	Комплексне управління підприємством	Широка функціональність, єдина база даних, аналітика	Складність впровадження, висока вартість, надмірність для малого бізнесу
Складська система	Облік товарів і матеріалів	Контроль залишків, рух ресурсів, облік надходжень і списань	Обмежена робота з клієнтами та статусами замовлень
POS-система	Оформлення продажів	Швидка фіксація продажу, робота з оплатою, чеки	Орієнтація на продаж, а не на повний цикл замовлення
Електронні таблиці	Простий ручний облік	Доступність, простота, швидкий початок роботи	Ризик помилок, складність масштабування, відсутність автоматизації

З аналізу таблиці видно, що кожен тип системи вирішує лише частину задач малого бізнесу. CRM-системи краще підходять для роботи з клієнтами, ERP-системи - для комплексного управління, складські системи - для контролю залишків, POS-системи - для оформлення продажів, а електронні таблиці - для найпростішого обліку. Проте для теми цієї роботи важливо поєднати декілька напрямів: управління замовленнями, контроль ресурсів, аналіз динаміки попиту та підтримку прийняття рішень.

Існуючі системи часто мають або занадто вузьку, або занадто широку спеціалізацію. Якщо система занадто проста, вона не дозволяє автоматизувати всі потрібні процеси. Якщо система занадто складна, її використання може бути незручним для малого підприємства. Тому доцільним є проектування інформаційної системи, яка буде орієнтована саме на базові потреби малого бізнесу та матиме зрозумілу структуру. Місце інформаційної системи в управлінні малим бізнесом можна подати у вигляді схеми.

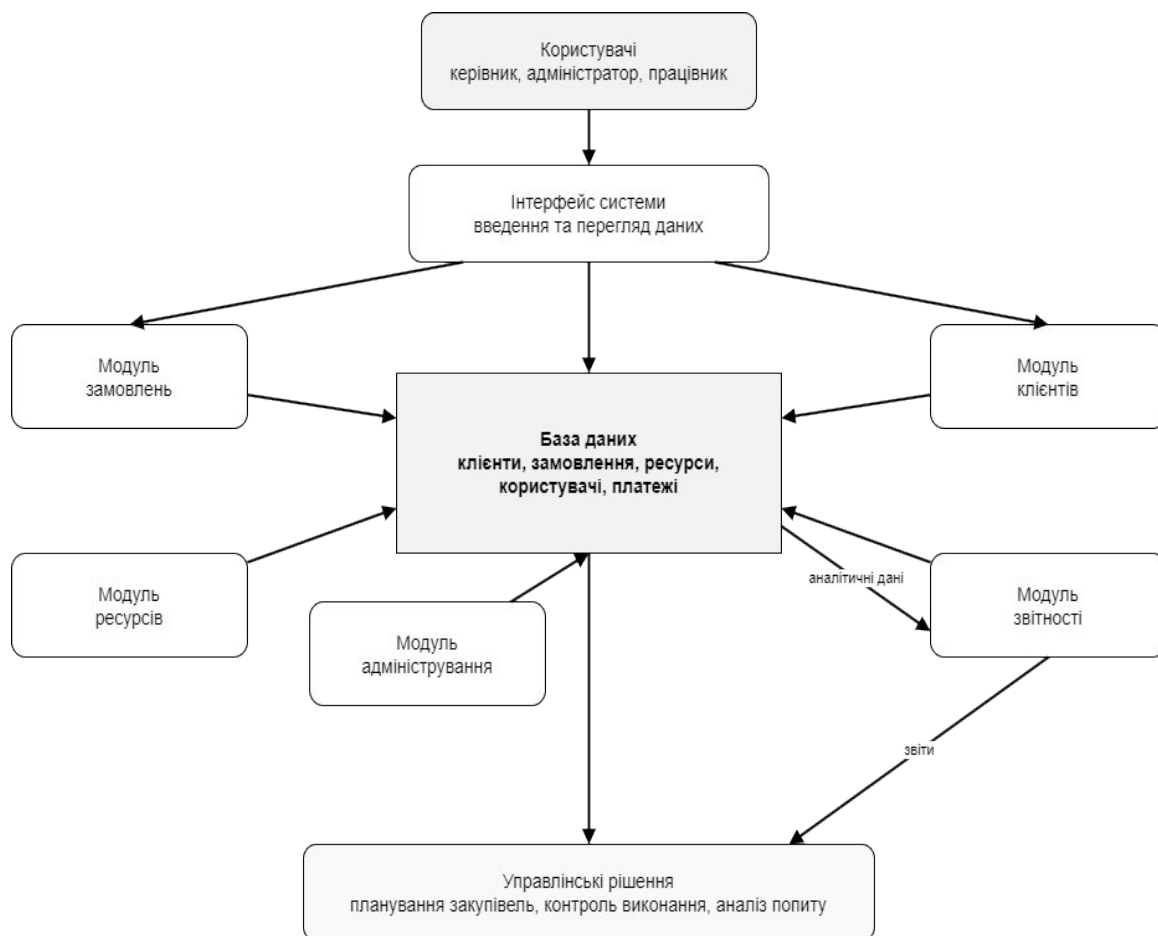


Рисунок 1.3 – Місце інформаційної системи в управлінні малим бізнесом

У центрі такої системи розміщується база даних, у якій зберігаються відомості про клієнтів, замовлення, ресурси, користувачів і платежі. Навколо неї працюють функціональні модулі: модуль замовлень, модуль ресурсів, модуль клієнтів, модуль звітності та модуль адміністрування. Користувачі взаємодіють із системою через інтерфейс, а система забезпечує збереження, оброблення та відображення даних.

Для розроблюваної інформаційної системи важливо передбачити такі можливості: створення та редагування замовлень, перевірку наявності ресурсів, зміну статусів замовлень, облік клієнтів, перегляд залишків, формування звітів і аналіз кількості замовлень за період. Саме ці функції є найбільш важливими для підприємства, яке працює в умовах змінного попиту та повинно постійно контролювати свої ресурси.

Також необхідно враховувати зручність використання системи. Малий бізнес не завжди має окремого системного адміністратора або спеціаліста з інформаційних технологій, тому програмне рішення повинно бути простим у впровадженні та зрозумілим для користувачів. Інтерфейс має бути логічним, а основні дії - доступними без складного навчання. Це стосується створення нового замовлення, пошуку клієнта, перегляду залишків і формування звіту.

Отже, огляд існуючих інформаційних систем показав, що на ринку є багато інструментів для автоматизації малого бізнесу, але кожен із них має певні обмеження. Для задачі управління малим бізнесом з урахуванням динаміки замовлень та ресурсів доцільно розробити систему, яка поєднує функції обліку замовлень, контролю ресурсів, роботи з клієнтами та формування аналітичної інформації. Такий підхід дозволить створити більш гнучке та практичне рішення для невеликого підприємства.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 19
Зм.	Арк.	№ докум.	Підпис	Дата		

1.4 Постановка задачі розроблення інформаційної системи

За результатами аналізу предметної області встановлено, що в малому бізнесі облік замовлень і контроль ресурсів мають розглядатися як взаємопов'язані процеси. Замовлення створює потребу в певних товарах, матеріалах, робочому часі або інших ресурсах, а фактичний стан ресурсів визначає, чи може підприємство виконати це замовлення у встановлений строк. Тому інформаційна система повинна не лише зберігати дані про замовлення, а й показувати, як кожне нове замовлення впливає на доступність ресурсів підприємства.

У практичній діяльності малого підприємства ці процеси часто виконуються окремо. Наприклад, замовлення можуть фіксуватися в електронних таблицях, залишки ресурсів - у складських документах, а інформація про оплати - в окремих файлах або повідомленнях. У такій ситуації керівник або адміністратор не завжди має повну й актуальну картину роботи підприємства. Це ускладнює прийняття рішень, особливо тоді, коли кількість замовлень швидко змінюється, а ресурси є обмеженими.

Основна проблема полягає у відсутності єдиного інформаційного середовища, яке поєднує роботу з клієнтами, замовленнями, ресурсами, оплатами та звітною інформацією. Якщо дані зберігаються в різних місцях, збільшується ймовірність помилок, виникають затримки під час пошуку інформації, а перевірка доступності ресурсів виконується вручну. У результаті підприємство може прийняти замовлення, для якого фактично не вистачає необхідних ресурсів.

З урахуванням цього задача розроблення інформаційної системи полягає у створенні програмного засобу, який забезпечує централізоване зберігання даних і підтримує основні процеси управління малим бізнесом. Система повинна дозволяти вести клієнтську базу, створювати замовлення, додавати до них потрібні ресурси, перевіряти їх доступність, резервувати ресурси для активних замовлень, фіксувати оплати та формувати звіти.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 20
Зм.	Арк.	№ докум.	Підпис	Дата		

Окремою вимогою є врахування динаміки замовлень. У межах цієї роботи під динамікою замовлень розуміється зміна кількості, складу та обсягу замовлень за певний період. Зростання кількості замовлень збільшує навантаження на ресурси, а скасування або завершення замовлень змінює їх доступний залишок. Тому система повинна працювати з актуальними даними та відображати реальний стан ресурсів з урахуванням уже створених, підтверджених і виконаних замовлень.

Розроблювана інформаційна система повинна виконувати такі основні задачі:

1. Забезпечувати авторизацію користувачів і розмежування прав доступу.
2. Зберігати інформацію про клієнтів малого підприємства.
3. Дозволяти створювати, редагувати та переглядати замовлення.
4. Відображати поточний статус кожного замовлення.
5. Зберігати інформацію про ресурси підприємства.
6. Перевіряти доступність ресурсів під час підтвердження замовлення.
7. Виконувати резервування ресурсів для активних замовлень.
8. Списувати ресурси після завершення замовлення.
9. Фіксувати оплати та визначати статус розрахунків.
10. Формувати звіти про замовлення, ресурси й оплати.

Розроблювана система має забезпечити зв'язок між замовленнями та ресурсами підприємства. Її використання дозволить зменшити кількість ручних операцій, уникнути дублювання даних, швидше перевіряти можливість виконання замовлень і надавати керівнику актуальну інформацію для прийняття управлінських рішень.

1.5 Висновки до першого розділу

У першому розділі було розглянуто особливості управління малим бізнесом в умовах змінного попиту. Визначено, що малі підприємства працюють в умовах обмежених ресурсів, нестабільної кількості замовлень і високої

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 21
Зм.	Арк.	№ докум.	Підпис	Дата		

залежності від оперативних управлінських рішень. Через це для них важливими є точний облік, швидкий доступ до інформації та можливість контролювати стан замовлень і ресурсів.

Було проаналізовано процеси оброблення замовлень і використання ресурсів. Встановлено, що кожне замовлення проходить кілька основних етапів: створення, перевірку ресурсів, підтвердження, резервування, виконання, оплати та закриття. Також визначено, що ресурси малого бізнесу можуть бути матеріальними, трудовими, фінансовими, часовими та інформаційними. Вони безпосередньо впливають на можливість виконання замовлень у встановлені строки.

У межах розділу було виконано огляд існуючих інформаційних систем для малого бізнесу. Розглянуто CRM-системи, ERP-системи, складські системи, POS-системи та електронні таблиці. Визначено, що кожен із цих інструментів має певні переваги, але не завжди повністю відповідає задачі одночасного управління замовленнями, ресурсами та аналітичною інформацією.

Також було сформульовано постановку задачі розроблення інформаційної системи. Розроблювана система повинна забезпечувати ведення клієнтської бази, створення та супровід замовлень, контроль доступності ресурсів, облік оплат, формування звітів і аналіз динаміки замовлень. Система має бути простою у використанні, надійною та придатною для подальшого розширення.

За результатами першого розділу обґрунтовано доцільність розроблення інформаційної системи управління малим бізнесом з урахуванням динаміки замовлень та ресурсів. Така система дозволить підвищити впорядкованість бізнес-процесів, зменшити кількість ручних операцій, покращити контроль ресурсів і створити основу для більш обґрунтованого прийняття управлінських рішень.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 22
Зм.	Арк.	№ докум.	Підпис	Дата		

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ МАЛИМ БІЗНЕСОМ

2.1 Визначення функціональних і нефункціональних вимог до системи

Після аналізу предметної області переходимо до визначення вимог до інформаційної системи управління малим бізнесом. На цьому етапі потрібно чітко встановити, які задачі повинна виконувати система, які дані вона має обробляти та які умови повинні бути забезпечені для її стабільної роботи. Вимоги до системи доцільно поділити на функціональні та нефункціональні. Функціональні вимоги описують конкретні можливості системи, а нефункціональні - характеристики її роботи, зручності, безпеки [11].

Розроблювана інформаційна система повинна бути орієнтована на практичне використання в малому бізнесі. Це означає, що її функції мають відповідати реальним щоденним процесам підприємства: облік клієнтів, створення замовлень, контроль статусів, перевірка ресурсів, фіксація оплат і формування звітів. При цьому система не повинна бути надмірно складною, оскільки для малого підприємства важлива не кількість модулів, а зручність і швидкість виконання основних операцій.

Однією з базових функцій системи є робота з користувачами. Система повинна забезпечувати реєстрацію користувачів, авторизацію та розмежування прав доступу. Це потрібно для того, щоб різні працівники мали доступ тільки до тих функцій, які відповідають їхнім обов'язкам. Наприклад, адміністратор може керувати всіма даними системи, менеджер - створювати та редагувати замовлення, працівник складу - працювати з ресурсами, а керівник - переглядати звіти й аналітичну інформацію.

Важливою функціональною вимогою є ведення бази клієнтів. У системі повинна зберігатися інформація про клієнта: ім'я або назва організації, контактний номер телефону, електронна пошта, адреса, історія замовлень і додаткові примітки. Це дозволяє швидше обробляти повторні звернення,

зменшити кількість помилок при введенні даних і забезпечити зручний пошук інформації.

Основним модулем інформаційної системи є модуль управління замовленнями. Він повинен дозволяти створювати нові замовлення, редагувати їх, додавати товари або послуги, змінювати статуси, переглядати історію та закривати виконані замовлення. Для малого бізнесу важливо, щоб користувач міг швидко зрозуміти, на якому етапі перебуває кожне замовлення: нове, підтвержене, у роботі, очікує оплати, виконане або скасоване.

Окрему роль відіграє модуль управління ресурсами. Він повинен забезпечувати облік товарів, матеріалів, обладнання або інших ресурсів, які використовуються під час виконання замовлень. Для кожного ресурсу доцільно зберігати назву, одиницю вимірювання, кількість, мінімальний допустимий залишок, вартість і статус доступності. Якщо кількість ресурсу зменшується до критичного рівня, система повинна повідомляти користувача про необхідність поповнення.

Вимоги до обліку ресурсів безпосередньо пов'язані з процесом оброблення замовлень. Під час створення або підтвердження замовлення система повинна перевіряти, чи достатньо ресурсів для його виконання. Якщо ресурсів вистачає, система дозволяє підтвердити замовлення та зарезервувати відповідну кількість. Якщо ресурсів недостатньо, користувач повинен отримати повідомлення про проблему. Такий підхід дозволяє зменшити ризик прийняття замовлень, які підприємство фактично не може виконати.

Також система повинна забезпечувати облік оплат. Для кожного замовлення необхідно фіксувати суму, спосіб оплати, дату платежу та статус оплати. Замовлення може бути неоплаченим, частково оплаченим або повністю оплаченим. Це допомагає керівнику контролювати фінансові надходження та швидко бачити, які замовлення потребують додаткової уваги.

Необхідною функцією є формування звітів. Система повинна дозволяти переглядати кількість замовлень за певний період, суму виконаних замовлень,

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 24
Зм.	Арк.	№ докум.	Підпис	Дата		

найбільш популярні товари або послуги, залишки ресурсів і замовлення, які ще не виконані. Такі звіти не тільки спрощують щоденний контроль, а й допомагають аналізувати динаміку діяльності підприємства. Функціональні вимоги до інформаційної системи наведено в таблиці 2.1.

Таблиця 2.1 – Функціональні вимоги до інформаційної системи

№	Вимога	Опис
1	Авторизація користувачів	Система повинна забезпечувати вхід користувачів за логіном і паролем
2	Розмежування ролей	Користувачі повинні мати різні права доступу відповідно до ролі
3	Ведення бази клієнтів	Система повинна зберігати контактні дані та історію замовлень клієнтів
4	Створення замовлень	Користувач повинен мати можливість створювати нові замовлення
5	Редагування замовлень	Система повинна дозволяти змінювати склад і параметри замовлення
6	Контроль статусів	Для кожного замовлення має відображатися поточний статус виконання
7	Облік ресурсів	Система повинна зберігати інформацію про наявні ресурси
8	Перевірка доступності ресурсів	Під час створення замовлення система повинна перевіряти наявність потрібних ресурсів
9	Резервування ресурсів	Ресурси можуть тимчасово закріплюватися за активним замовленням
10	Облік оплат	Система повинна фіксувати платежі та статус оплати
11	Формування звітів	Користувач повинен мати можливість формувати звіти за період
12	Пошук і фільтрація	Система повинна підтримувати пошук клієнтів, замовлень і ресурсів

Крім функціональних вимог, необхідно визначити нефункціональні вимоги. Вони описують не конкретні дії системи, а якість її роботи. Для малого бізнесу особливо важливими є простота використання, стабільність, швидкодія, захищеність даних і можливість подальшого розвитку системи.

Першою нефункціональною вимогою є зручність інтерфейсу. Користувачі системи не обов'язково мають спеціальну технічну підготовку, тому інтерфейс повинен бути простим, логічним і зрозумілим. Основні дії, такі як створення замовлення, пошук клієнта, перегляд залишків або формування звіту, повинні виконуватися без зайвих кроків.

Другою важливою вимогою є надійність зберігання даних. Інформація про клієнтів, замовлення, ресурси та платежі є важливою для роботи підприємства, тому система повинна забезпечувати коректне збереження, оновлення й отримання даних. Помилки в базі даних можуть призвести до неправильного виконання замовлень або втрати важливої інформації.

Швидкодія системи також має значення. Малий бізнес часто працює в режимі оперативного обслуговування клієнтів, тому користувач не повинен довго чекати завантаження сторінок, пошуку записів або формування базових звітів. Для більшості операцій система повинна відповідати швидко, особливо під час роботи із замовленнями та ресурсами.

Окремою вимогою є безпека. Система повинна захищати дані від несанкціонованого доступу. Для цього необхідно використовувати авторизацію, обмеження прав користувачів і перевірку введених даних. Наприклад, звичайний працівник не повинен мати можливості видаляти критично важливі записи або змінювати налаштування системи.

Також важливою є масштабованість. На початковому етапі система може використовуватися невеликою кількістю працівників і зберігати обмежений обсяг даних. Однак у майбутньому підприємство може розширитися, збільшити кількість клієнтів, замовлень або ресурсів. Тому система повинна мати таку структуру, яка дозволяє додавати нові модулі без повної переробки всієї архітектури.

Нефункціональні вимоги до інформаційної системи наведено в таблиці 2.2.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 26
Зм.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.2 – Нефункціональні вимоги до інформаційної системи

№	Вимога	Опис
1	Зручність використання	Інтерфейс повинен бути простим і зрозумілим для користувача
2	Надійність	Система повинна коректно зберігати та обробляти дані
3	Швидкодія	Основні операції повинні виконуватися без значних затримок
4	Безпека	Доступ до даних повинен контролюватися відповідно до ролі користувача
5	Масштабованість	Система повинна мати можливість подальшого розширення
6	Супроводжуваність	Код і структура системи повинні бути зрозумілими для подальшої підтримки
7	Цілісність даних	Дані повинні зберігатися без дублювання та логічних суперечностей
8	Доступність	Система повинна бути доступною користувачам у робочий час підприємства

Визначені функціональні та нефункціональні вимоги формують основу для подальшого проєктування інформаційної системи. Функціональні вимоги показують, які задачі повинна виконувати система, а нефункціональні - якою вона повинна бути з погляду якості, безпеки та зручності використання. У подальших підрозділах на основі цих вимог буде спроектовано архітектуру системи, структуру бази даних, алгоритми роботи з замовленнями та ресурсами, а також користувацький інтерфейс.

Важливою особливістю розроблюваної інформаційної системи є врахування динаміки замовлень та ресурсів. У діяльності малого бізнесу

кількість замовлень постійно змінюється залежно від попиту, сезону, рекламних кампаній та інших факторів. Тому система повинна не лише зберігати інформацію про замовлення, а й забезпечувати аналіз їх змін у часі.

Динаміка замовлень безпосередньо пов'язана з використанням ресурсів підприємства. Кожне нове замовлення впливає на запаси товарів, матеріалів або обладнання, необхідних для його виконання. У свою чергу, нестача ресурсів може призвести до затримок або неможливості виконання замовлень. Саме тому система повинна автоматично контролювати наявність ресурсів, резервувати необхідні запаси та повідомляти користувачів про критичне зменшення їх кількості.

Для ефективного управління підприємством система повинна накопичувати історичні дані щодо замовлень і ресурсів та надавати можливість формування аналітичних звітів. Це дозволить оцінювати тенденції зміни попиту, контролювати використання ресурсів і приймати обґрунтовані управлінські рішення. Таким чином, врахування динаміки замовлень та ресурсів є важливою вимогою під час проєктування інформаційної системи управління малим бізнесом і забезпечує підвищення ефективності її роботи.

2.2 Проєктування архітектури інформаційної системи

Архітектура інформаційної системи визначає склад основних компонентів програмного засобу, порядок їхньої взаємодії та спосіб передавання даних між ними. Для інформаційної системи управління малим бізнесом доцільно використати клієнт-серверну архітектуру, оскільки вона дозволяє розділити інтерфейс користувача, бізнес-логіку та рівень зберігання даних. Такий підхід спрощує розроблення, тестування, супровід системи та створює можливість для подальшого розширення функціональності [13].

У межах розроблюваної системи клієнтська частина відповідає за взаємодію користувача з програмним засобом. Через неї користувач може

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 28
Зм.	Арк.	№ докум.	Підпис	Дата		

авторизуватися, переглядати клієнтів, створювати замовлення, контролювати ресурси, додавати оплати та формувати звіти. Клієнтська частина не виконує основні правила оброблення даних самостійно, а передає запити на серверну частину.

Серверна частина є центральним елементом оброблення запитів. Вона приймає дані від клієнтської частини, перевіряє їх коректність, визначає права користувача, виконує операції із замовленнями, ресурсами й оплатами та звертається до бази даних. Саме на серверному рівні реалізуються такі дії, як перевірка доступності ресурсів, резервування, списання, скасування резерву та формування звітної інформації.

База даних використовується для зберігання структурованої інформації про користувачів, клієнтів, замовлення, ресурси, оплати та рух ресурсів. Вона є спільним джерелом даних для всіх модулів системи. Це дозволяє уникнути дублювання інформації та забезпечити узгоджену роботу системи.

Загальну архітектуру розроблюваної інформаційної системи наведено на рисунку 2.1.

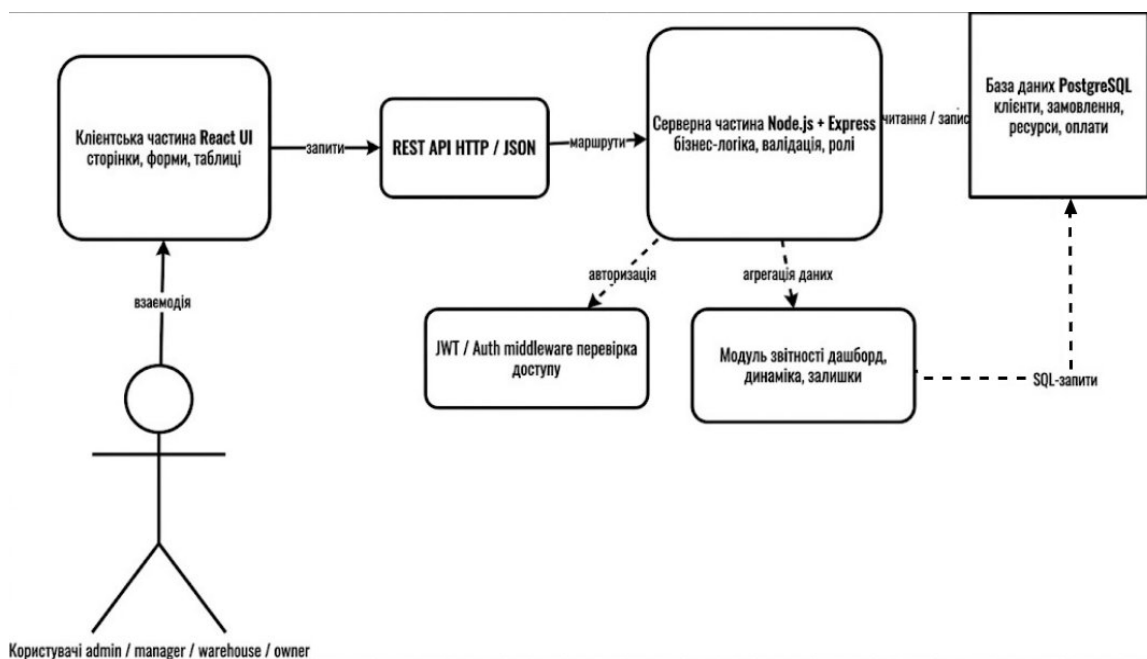


Рисунок 2.1 – Компонентна архітектура інформаційної системи управління малим бізнесом

На рисунку 2.1, система складається з трьох основних рівнів: рівня представлення, рівня бізнес-логіки та рівня даних. Рівень представлення реалізується клієнтською частиною та забезпечує роботу користувача з інтерфейсом. Рівень бізнес-логіки реалізується серверною частиною та відповідає за оброблення запитів. Рівень даних представлений базою даних, у якій зберігається основна інформація системи.

Для обміну даними між клієнтською та серверною частинами використовується REST API. Такий підхід дозволяє організувати взаємодію між компонентами системи через окремі HTTP-запити до серверних маршрутів. Наприклад, для отримання списку клієнтів використовується запит до маршруту /api/customers, для створення замовлення - /api/orders, для перегляду ресурсів - /api/resources, а для формування звітів - /api/reports. Відповідь сервера передається у форматі JSON, що зручно для подальшого відображення даних у клієнтській частині [14].

Розроблювана система складається з кількох функціональних модулів. Кожен модуль відповідає за окрему групу задач, але всі модулі використовують спільну базу даних. До основних модулів належать: модуль авторизації, модуль клієнтів, модуль замовлень, модуль ресурсів, модуль оплат, модуль звітності та модуль адміністрування.

Модуль авторизації забезпечує вхід користувачів до системи та перевірку їхніх прав доступу. Під час входу користувач передає логін і пароль, після чого система перевіряє обліковий запис і визначає роль користувача. Залежно від ролі користувач отримує доступ до відповідних функцій. Наприклад, адміністратор має доступ до всіх модулів, менеджер працює із клієнтами та замовленнями, працівник складу - з ресурсами, а керівник - зі звітами й аналітичною інформацією [15].

Модуль клієнтів призначений для ведення клієнтської бази. Він забезпечує додавання нових клієнтів, редагування контактних даних, пошук клієнтів і

перегляд історії їхніх замовлень. Цей модуль передає до бази даних пакет даних клієнта, який може містити ім'я, телефон, електронну пошту, адресу та примітки.

Модуль замовлень є центральним у системі. Він забезпечує створення нового замовлення, вибір клієнта, додавання позицій, розрахунок суми, зміну статусу та перегляд деталей. Під час створення або підтвердження замовлення цей модуль обмінюється даними з модулем ресурсів, оскільки для кожної позиції потрібно перевірити доступний залишок.

Модуль ресурсів відповідає за облік товарів, матеріалів або інших ресурсів підприємства. Він зберігає загальну кількість ресурсу, зарезервовану кількість, доступний залишок і мінімальний допустимий залишок. Під час підтвердження замовлення модуль ресурсів отримує від модуля замовлень пакет даних про потрібні ресурси та повертає результат перевірки: ресурсів достатньо або ресурсів недостатньо.

Модуль оплат забезпечує фіксацію платежів за замовленнями. Він отримує дані про суму платежу, спосіб оплати, дату та пов'язане замовлення. Після внесення платежу система оновлює статус оплати: «не оплачено», «частково оплачено» або «оплачено».

Модуль звітності використовується для формування узагальненої інформації про роботу підприємства. Він отримує з бази даних відомості про кількість замовлень, суми оплат, активні та виконані замовлення, а також ресурси з критичним залишком. На основі цих даних формуються звіти для керівника.

Модуль адміністрування призначений для керування користувачами, ролями та основними налаштуваннями системи. Він дозволяє додавати користувачів, змінювати їхні ролі та обмежувати доступ до окремих функцій.

Модульну структуру та інформаційні потоки між модулями системи наведено на рисунку 2.2.

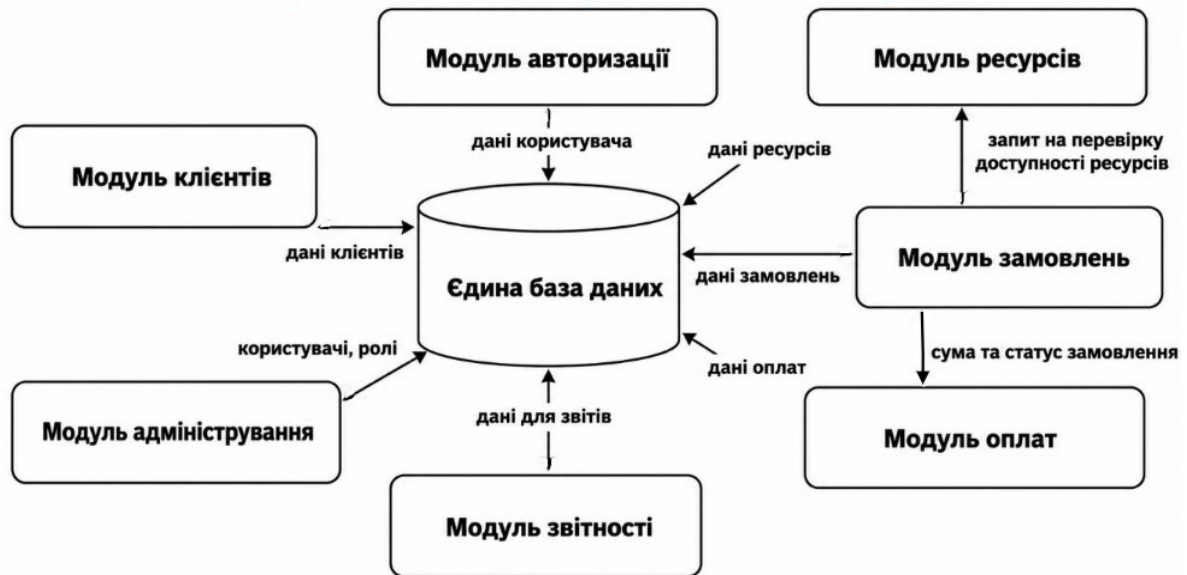


Рисунок 2.2 – Модульна структура та інформаційні потоки системи

На рисунку 2.2 доцільно показати не тільки основні модулі, а й пакети даних, якими вони обмінюються. Наприклад, між модулем клієнтів і базою даних передаються дані клієнта; між модулем замовлень і базою даних - дані замовлення та його статус; між модулем замовлень і модулем ресурсів - запит на перевірку доступності ресурсів; між модулем ресурсів і базою даних - дані про залишки, резервування та списання; між модулем оплат і базою даних - дані платежу; між модулем звітності і базою даних - агреговані дані для звітів.

Для більш точного опису інформаційної взаємодії в системі можна виділити такі основні пакети даних:

1. Дані користувача: логін, пароль, роль, токен доступу.
2. Дані клієнта: ім'я, телефон, електронна пошта, адреса, примітки.
3. Дані замовлення: клієнт, позиції, кількість, сума, строк виконання, статус.
4. Дані ресурсів: назва ресурсу, кількість, резерв, доступний залишок, мінімальний залишок.
5. Дані оплати: сума, спосіб оплати, дата платежу, статус оплати.
6. Дані звітності: кількість замовлень, активні замовлення, виконані замовлення, критичні ресурси, сума оплат.

7. Дані адміністрування: користувачі, ролі, права доступу.

Взаємодія компонентів системи відбувається послідовно. Наприклад, під час створення нового замовлення користувач заповнює форму в клієнтській частині. Далі клієнтська частина формує JSON-пакет із даними замовлення та передає його на сервер через REST API. Сервер перевіряє дані, зберігає замовлення в базі даних і повертає відповідь про успішне створення запису.

Під час підтвердження замовлення взаємодія є складнішою. Сервер отримує дані замовлення, визначає перелік потрібних ресурсів і перевіряє їх доступність. Якщо ресурсів достатньо, система резервує потрібну кількість і змінює статус замовлення. Якщо ресурсів недостатньо, сервер повертає повідомлення про помилку, а замовлення не підтверджується. Така логіка дозволяє уникнути прийняття замовлень, які підприємство фактично не може виконати.

Загальні задачі, які повинна виконувати інформаційна система, наведено на рисунку 2.3.



Рисунок 2.3 – Узагальнена схема задач інформаційної системи

Як видно з рисунка 2.3, система повинна підтримувати облік клієнтів, облік замовлень, контроль ресурсів, облік оплат, формування звітів, аналіз динаміки замовлень та адміністрування користувачів. Усі ці задачі пов'язані між собою,

оскільки замовлення створюється для конкретного клієнта, потребує певних ресурсів, може мати оплату та впливає на звітні показники підприємства.

Для реалізації поставленої задачі інформаційна система повинна виконувати такі функції:

1. Реєстрація та авторизація користувачів.
2. Розмежування прав доступу залежно від ролі користувача.
3. Ведення бази клієнтів.
4. Створення, редагування та видалення замовлень.
5. Перегляд історії замовлень клієнта.
6. Зміна статусів замовлень.
7. Облік товарів, матеріалів або інших ресурсів.
8. Перевірка доступності ресурсів для виконання замовлення.
9. Резервування ресурсів під активні замовлення.
10. Фіксація оплат і статусу розрахунків.
11. Формування звітів за вибраний період.
12. Аналіз кількості замовлень і використання ресурсів.

Окремо потрібно врахувати роботу з ролями користувачів. Адміністратор системи повинен мати доступ до всіх функцій, керівник - до замовлень, ресурсів, оплат і звітів, менеджер - до клієнтів і замовлень, а працівник складу - до залишків і руху ресурсів. Такий підхід дозволяє зробити систему більш безпечною та зручною для практичного використання.

З технічної точки зору система має бути реалізована як програмний засіб, що складається з клієнтської частини, серверної частини та бази даних. Клієнтська частина відповідає за введення та відображення даних, серверна частина обробляє запити й виконує бізнес-логіку, а база даних забезпечує зберігання структурованої інформації про користувачів, клієнтів, замовлення, ресурси та платежі [10].

До основних вимог до розроблюваної системи належать зручність використання, надійність зберігання даних, швидкий доступ до інформації,

можливість пошуку та фільтрації записів, коректна робота зі статусами замовлень і захист даних від несанкціонованого доступу. Також система повинна мати можливість подальшого розширення, наприклад додавання модуля постачальників, закупівель або розширеної фінансової аналітики.

Важливо, щоб інформаційна система не була надмірно складною. Для малого бізнесу практичну цінність має не максимальна кількість функцій, а відповідність реальним потребам підприємства. Тому під час проєктування необхідно зосередитися на тих процесах, які безпосередньо впливають на якість управління: оброблення замовлень, контроль ресурсів, облік клієнтів, облік оплат і формування звітної інформації.

2.3 Проєктування бази даних системи

База даних є одним із ключових елементів інформаційної системи управління малим бізнесом. Вона забезпечує зберігання, оновлення, пошук і зв'язування даних, які використовуються в роботі підприємства. Для розроблюваної системи база даних повинна містити інформацію про користувачів, клієнтів, замовлення, ресурси, платежі та рух ресурсів. Саме ці дані потрібні для автоматизації основних процесів малого бізнесу.

Для такої системи доцільно використовувати реляційну модель даних, оскільки вона добре підходить для зберігання структурованої інформації та дозволяє встановлювати зв'язки між окремими сутностями. Наприклад, одне замовлення пов'язане з конкретним клієнтом, замовлення може містити декілька позицій, кожна позиція пов'язана з певним ресурсом, а оплати належать до конкретного замовлення. Такий підхід дозволяє зберігати дані без значного дублювання та підтримувати їхню цілісність [17].

Основними сутностями бази даних є: користувачі, клієнти, замовлення, позиції замовлення, ресурси, платежі та рух ресурсів. Кожна сутність відповідає окремому об'єкту предметної області. Наприклад, сутність «Клієнт» зберігає контактну інформацію про покупця або замовника, сутність «Замовлення» - дані

про заявку клієнта, а сутність «Ресурс» - інформацію про товари, матеріали або інші засоби, необхідні для виконання замовлень.

Таблиця користувачів призначена для зберігання даних про осіб, які працюють із системою. До таких даних належать ім'я користувача, електронна пошта, пароль у захищеному вигляді, роль і дата створення облікового запису. Роль користувача визначає, які функції системи йому доступні. Наприклад, адміністратор має повний доступ, менеджер працює з клієнтами та замовленнями, працівник складу - з ресурсами, а керівник переглядає звіти.

Таблиця клієнтів використовується для зберігання інформації про фізичних або юридичних осіб, які роблять замовлення. У ній доцільно передбачити такі поля: ідентифікатор клієнта, повне ім'я або назва організації, телефон, електронна пошта, адреса, примітки та дата створення запису. Наявність такої таблиці дозволяє швидко знаходити клієнтів, переглядати історію їхніх замовлень і зменшувати кількість повторного введення даних.

Таблиця замовлень є центральною в базі даних. Вона зберігає інформацію про кожне замовлення: клієнта, статус, загальну суму, статус оплати, строк виконання, коментар, дату створення та дату оновлення. Кожне замовлення повинно бути пов'язане з одним клієнтом. При цьому один клієнт може мати багато замовлень, тому між таблицями клієнтів і замовлень встановлюється зв'язок «один до багатьох».

Позиції замовлення зберігаються в окремій таблиці. Це потрібно тому, що одне замовлення може містити декілька товарів, послуг або ресурсів. Наприклад, клієнт може замовити кілька різних позицій, кожна з яких має свою кількість, ціну та проміжну суму. Таблиця позицій замовлення пов'язує замовлення з ресурсами та дозволяє розрахувати загальну суму замовлення.

Таблиця ресурсів призначена для обліку товарів, матеріалів, обладнання або інших елементів, які використовуються у процесі виконання замовлень. Для кожного ресурсу потрібно зберігати назву, категорію, одиницю вимірювання, загальну кількість, зарезервовану кількість, мінімальний залишок, ціну, статус і

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 36
Зм.	Арк.	№ докум.	Підпис	Дата		

дату оновлення. Важливими є поля загальної та зарезервованої кількості, оскільки вони дозволяють визначати доступний залишок ресурсу.

Таблиця платежів використовується для фіксації оплат за замовленнями. Вона повинна містити ідентифікатор платежу, ідентифікатор замовлення, суму, спосіб оплати, статус платежу, дату оплати та дату створення запису. Одне замовлення може мати кілька платежів, наприклад у випадку часткової оплати. Тому між таблицями замовлень і платежів встановлюється зв'язок «один до багатьох».

Таблиця руху ресурсів потрібна для контролю змін кількості ресурсів. У ній фіксуються операції резервування, списання, скасування резерву або поповнення запасів. Така таблиця дозволяє зберігати історію змін і пояснювати, чому кількість певного ресурсу збільшилася або зменшилася. Це важливо для прозорості обліку та подальшого аналізу. Основні сутності бази даних наведено в таблиці 2.3.

Таблиця 2.3 – Основні сутності бази даних інформаційної системи

Назва сутності	Призначення	Основні поля
User	Зберігання користувачів системи	id, name, email, passwordHash, role, createdAt
Customer	Зберігання даних про клієнтів	id, fullName, phone, email, address, notes, createdAt
Order	Зберігання замовлень	id, customerId, status, totalAmount, paymentStatus, deadline, comment
OrderItem	Зберігання позицій замовлення	id, orderId, resourceId, quantity, price, subtotal
Resource	Облік ресурсів підприємства	id, name, category, unit, quantity, reservedQuantity, minQuantity, price
Payment	Облік оплат за замовленнями	id, orderId, amount, method, status, paidAt
ResourceMovement	Фіксація руху ресурсів	id, resourceId, orderId, type, quantity, comment, createdAt

Модель бази даних інформаційної системи відображає основні таблиці, їхні поля та зв'язки між ними. Вона поєднує логічне представлення предметної області та фізичну структуру зберігання даних. У межах цієї моделі визначено таблиці користувачів, клієнтів, замовлень, позицій замовлень, ресурсів, оплат і руху ресурсів.

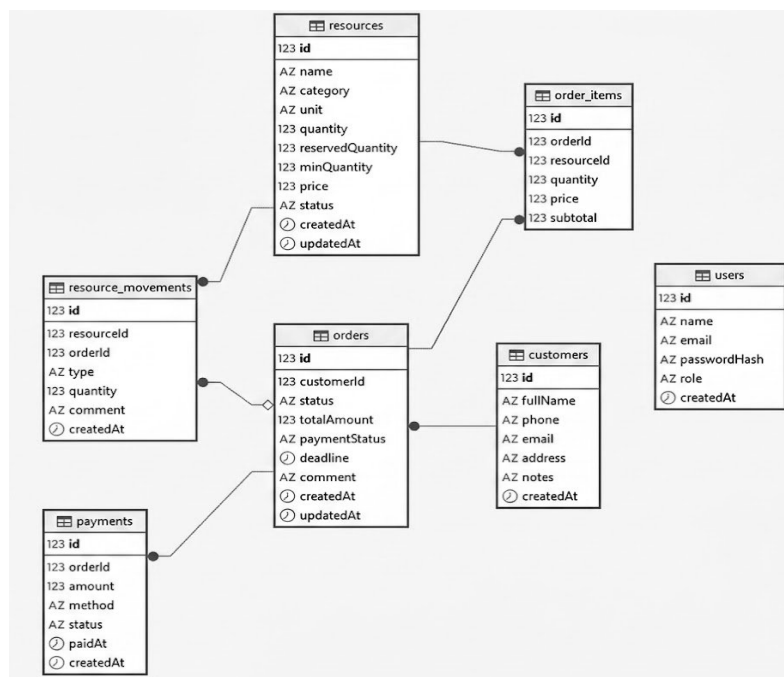


Рисунок 2.4 – Модель бази даних інформаційної системи

Основними таблицями бази даних є User, Customer, Order, OrderItem, Resource, Payment та ResourceMovement. Таблиця User призначена для зберігання даних про користувачів системи, їхні ролі та облікові записи. Таблиця Customer містить інформацію про клієнтів підприємства. Таблиця Order використовується для зберігання замовлень, їхніх статусів, строків виконання, загальної суми та статусу оплати. Таблиця OrderItem деталізує склад кожного замовлення, оскільки одне замовлення може містити кілька товарів, послуг або ресурсів. Таблиця Resource відповідає за облік ресурсів підприємства, а таблиця Payment зберігає інформацію про оплати. Таблиця ResourceMovement фіксує історію змін кількості ресурсів.

Зв'язки між таблицями мають важливе значення для правильної роботи системи. Якщо замовлення створюється для конкретного клієнта, у таблиці Order зберігається поле customerId, яке посилається на таблицю Customer. Це дозволяє швидко отримати всі замовлення певного клієнта та переглянути історію його взаємодії з підприємством. Якщо замовлення містить декілька позицій, кожна з них зберігається в таблиці OrderItem, де вказується ідентифікатор замовлення, ресурс, кількість, ціна та проміжна сума.

Таблиця OrderItem також пов'язана з таблицею Resource, оскільки кожна позиція замовлення використовує певний ресурс або товар. Завдяки цьому система може визначити, які саме ресурси потрібні для виконання замовлення, перевірити їх наявність і виконати резервування. Таблиця Payment пов'язана з таблицею Order, тому до одного замовлення може бути додано одну або кілька оплат. Це дозволяє враховувати як повну, так і часткову оплату.

Окрему роль у моделі відіграє таблиця ResourceMovement. Вона пов'язана з таблицями Resource та Order і використовується для фіксації операцій із ресурсами. Наприклад, під час підтвердження замовлення створюється запис про резервування ресурсу, після виконання замовлення - запис про списання, а після скасування - запис про зняття резерву. Така структура дозволяє не тільки бачити поточну кількість ресурсів, а й аналізувати історію їхнього використання.

Під час проєктування бази даних враховано цілісність даних. Для цього використовуються первинні та зовнішні ключі. Первинний ключ унікально ідентифікує кожен запис у таблиці, а зовнішній ключ забезпечує зв'язок між таблицями. Наприклад, поле customerId у таблиці Order є зовнішнім ключем, який посилається на таблицю Customer. Це не дозволяє створити замовлення для клієнта, якого немає в базі даних.

Також важливо уникати дублювання даних. Наприклад, контактні дані клієнта не потрібно зберігати в кожному замовленні. Достатньо зберегти їх один раз у таблиці Customer, а в замовленні вказати лише ідентифікатор клієнта. Такий

підхід спрощує оновлення інформації: якщо клієнт змінює номер телефону або адресу електронної пошти, ці дані потрібно змінити тільки в одному місці.

У моделі також відображено фізичні параметри зберігання даних. Для кожної таблиці визначаються поля, типи даних та обмеження. Наприклад, поле `id` може мати тип `integer` або `uuid`, поле `email` - тип `varchar`, поле `totalAmount` - тип `decimal`, а поле `createdAt` - тип `timestamp`. Для статусів замовлення й оплати можуть використовуватися текстові значення або перелік допустимих значень. Це дозволяє обмежити введення некоректних даних і зробити роботу системи більш надійною. У фізичній структурі бази даних доцільно передбачити такі основні зв'язки:

1. Customer - Order: один клієнт може мати багато замовлень.
2. Order - OrderItem: одне замовлення може містити багато позицій.
3. Resource - OrderItem: один ресурс може використовуватися в багатьох позиціях замовлень.
4. Order - Payment: одне замовлення може мати багато платежів.
5. Resource - ResourceMovement: один ресурс може мати багато записів руху.
6. Order - ResourceMovement: одне замовлення може бути пов'язане з кількома операціями руху ресурсів.

Для підвищення швидкодії системи доцільно використовувати індекси. Наприклад, індекси можуть бути створені для полів `customerId`, `orderId`, `resourceId`, `status` і `createdAt`. Це дозволить швидше виконувати пошук замовлень клієнта, фільтрацію замовлень за статусом, перегляд руху ресурсів і формування звітів за вибраний період.

Окрему увагу потрібно приділити правилам зміни даних. Під час підтвердження замовлення система не повинна одразу списувати ресурси, а лише резервувати їх. Списання має виконуватися після завершення замовлення. Якщо замовлення скасоване, зарезервовані ресурси повинні бути повернуті в доступний залишок. Для цього в базі даних використовується поле

reservedQuantity у таблиці Resource і таблиця ResourceMovement, яка фіксує всі операції з ресурсами.

З точки зору подальшого розвитку системи запропонована модель бази даних є достатньо гнучкою. У майбутньому до неї можна додати таблиці постачальників, закупівель, категорій товарів, знижок, повідомлень або розширених фінансових звітів. При цьому основна структура, пов'язана з клієнтами, замовленнями та ресурсами, залишиться незмінною.

Отже, спроектована модель бази даних забезпечує зберігання основної інформації, необхідної для роботи інформаційної системи управління малим бізнесом. Вона охоплює користувачів, клієнтів, замовлення, позиції замовлень, ресурси, платежі та рух ресурсів. Використання реляційної структури дозволяє забезпечити зв'язність даних, уникнути дублювання та створити основу для реалізації функціональних модулів системи.

2.4 Проектування алгоритмів оброблення замовлень і ресурсів

Проектування алгоритмів роботи інформаційної системи є важливим етапом, оскільки саме алгоритми визначають послідовність дій користувача та системи під час виконання основних бізнес-процесів. Для інформаційної системи управління малим бізнесом ключовими є алгоритм оброблення замовлення, алгоритм перевірки доступності ресурсів, алгоритм резервування ресурсів, алгоритм завершення замовлення та алгоритм скасування замовлення. Вони забезпечують зв'язок між замовленнями, ресурсами, платежами та статусами виконання.

Основним процесом у системі є оброблення замовлення. Він починається з того, що користувач створює нове замовлення та вибирає клієнта з бази даних. Якщо клієнта ще немає в системі, його дані спочатку вносяться до клієнтської бази. Після цього користувач додає до замовлення потрібні товари, послуги або ресурси, вказує кількість, строк виконання та додатковий коментар. Система

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 41
Зм.	Арк.	№ докум.	Підпис	Дата		

автоматично розраховує загальну суму замовлення на основі ціни й кількості кожної позиції.

Після створення замовлення система повинна перевірити, чи достатньо ресурсів для його виконання. Це потрібно для того, щоб підприємство не підтвердило замовлення, яке фактично не може виконати. Перевірка виконується для кожної позиції замовлення. Якщо для всіх позицій доступної кількості ресурсів достатньо, замовлення може бути підтверджене. Якщо хоча б одного ресурсу не вистачає, система повинна повідомити користувача про проблему та не дозволити підтвердження замовлення. Загальний алгоритм оброблення замовлення наведено на рисунку 2.5.

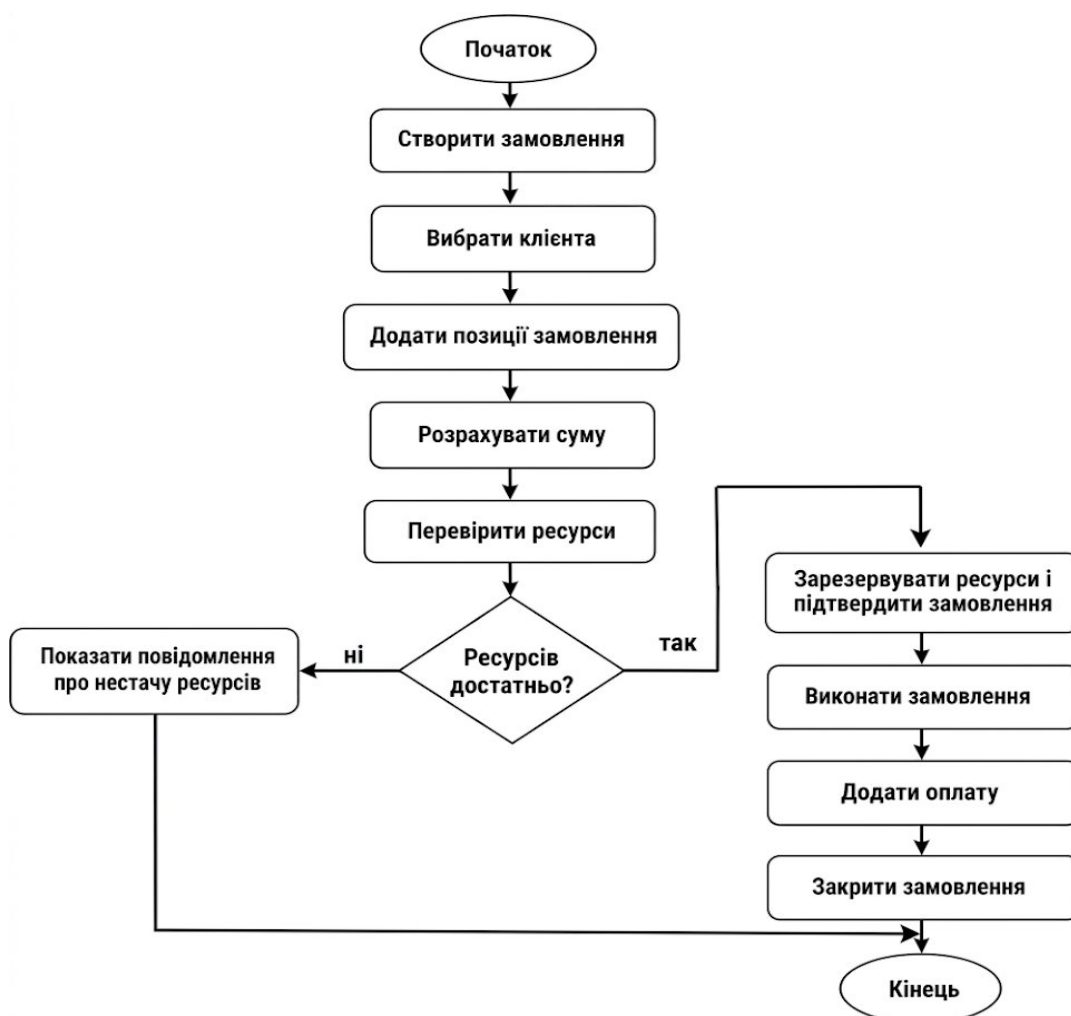


Рисунок 2.5 – Алгоритм оброблення замовлення

Для перевірки доступності ресурсів використовується показник доступної кількості. Він визначається як різниця між загальною кількістю ресурсу та вже зарезервованою кількістю. Якщо доступна кількість більша або дорівнює кількості, яка потрібна для нового замовлення, ресурс можна використати. Якщо доступна кількість менша, система повинна показати повідомлення про нестачу.

Формула визначення доступної кількості ресурсу має вигляд:

$$A = Q - R \quad (2.1)$$

де A – доступна кількість ресурсу;

Q – загальна кількість ресурсу на складі або в обліку;

R – зарезервована кількість ресурсу під активні замовлення.

Ця формула є базовою для логіки перевірки ресурсів. Наприклад, якщо на складі є 20 одиниць ресурсу, а 8 одиниць уже зарезервовано під інші замовлення, то доступна кількість становить 12 одиниць. Якщо нове замовлення потребує 10 одиниць, його можна підтвердити. Якщо потрібно 15 одиниць, система повинна заборонити підтвердження або запропонувати змінити кількість.

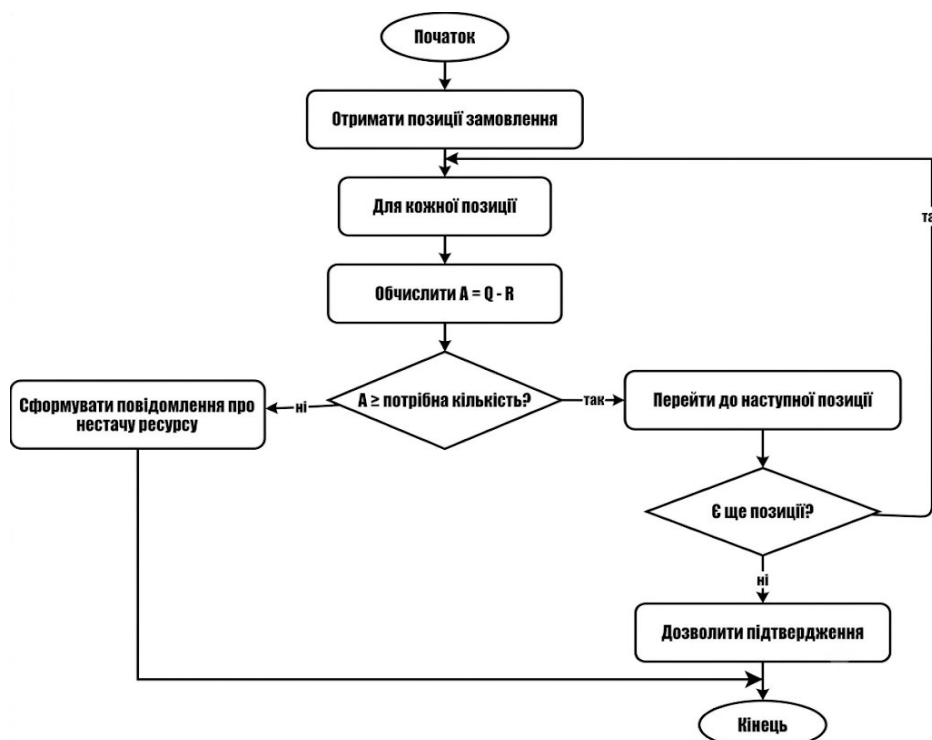


Рисунок 2.5 – Перевірка доступності ресурсів

Після успішної перевірки ресурсів виконується резервування. Резервування означає, що певна кількість ресурсу тимчасово закріплюється за конкретним замовленням. При цьому загальна кількість ресурсу ще не зменшується, оскільки замовлення ще не виконане. Змінюється тільки зарезервована кількість. Такий підхід є зручним, оскільки дозволяє бачити різницю між загальною кількістю ресурсу та реально доступною кількістю для нових замовлень.

Алгоритм послідовності резервування ресурсів:

1. Користувач підтверджує замовлення.
2. Система повторно перевіряє доступність ресурсів.
3. Якщо ресурсів достатньо, для кожної позиції збільшується зарезервована кількість.
4. Статус замовлення змінюється на «підтвержене».
5. У таблиці руху ресурсів створюється запис із типом «резервування».
6. Користувач отримує повідомлення про успішне підтвердження замовлення.

Повторна перевірка перед резервуванням потрібна для уникнення помилок. Наприклад, якщо два користувачі одночасно працюють із системою, доступна кількість ресурсу могла змінитися після відкриття форми замовлення. Тому остаточне рішення про підтвердження повинно прийматися тільки після перевірки актуальних даних у базі.

Після виконання замовлення ресурси повинні бути списані. На цьому етапі система зменшує загальну кількість ресурсу та одночасно зменшує зарезервовану кількість. Це означає, що ресурс фактично використано для виконання замовлення. Після списання замовлення може бути переведене в статус «виконане». Якщо оплата ще не внесена повністю, замовлення може мати окремий статус оплати, наприклад «частково оплачено» або «не оплачено».

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 44
Зм.	Арк.	№ докум.	Підпис	Дата		

Окремо потрібно передбачити алгоритм скасування замовлення. Скасування може відбуватися з різних причин: клієнт відмовився від замовлення, ресурсів не вистачає, строк виконання не підходить або замовлення було створене помилково. Якщо замовлення ще не було підтверджене, його можна просто перевести в статус «скасоване». Якщо ресурси вже були зарезервовані, система повинна зняти резерв.

Алгоритм скасування замовлення складається з таких кроків:

1. Користувач обирає дію «Скасувати замовлення».
2. Система перевіряє поточний статус замовлення.
3. Якщо ресурси були зарезервовані, зарезервована кількість зменшується.
4. У таблиці руху ресурсів створюється запис із типом «скасування резерву».
5. Статус замовлення змінюється на «скасоване».
6. Користувач отримує повідомлення про успішне скасування.

Для оплати замовлення доцільно передбачити окремий алгоритм. Користувач додає платіж, вказує суму, спосіб оплати та дату. Після збереження платежу система підраховує загальну суму всіх оплат за цим замовленням і порівнює її із загальною сумою замовлення. Якщо оплат немає, статус оплати залишається «не оплачено». Якщо внесена частина суми, статус змінюється на «частково оплачено». Якщо сума оплат дорівнює або перевищує суму замовлення, статус змінюється на «оплачено».

Для оцінювання стану ресурсу доцільно використовувати мінімальний залишок. Якщо загальна кількість ресурсу стає меншою або дорівнює мінімальному залишку, система повинна позначати такий ресурс як критичний. Це дозволяє користувачу швидко бачити, які товари або матеріали потрібно поповнити найближчим часом.

У загальному вигляді цю умову можна записати так:

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 45
Зм.	Арк.	№ докум.	Підпис	Дата		

$$Q \leq M$$

(2.2)

де Q – загальна кількість ресурсу;

M – мінімальний допустимий залишок ресурсу.

Якщо ця умова виконується, ресурс потрапляє до списку критичних. У звітах такі ресурси можуть відображатися окремо, щоб керівник або працівник складу міг швидко прийняти рішення про закупівлю або поповнення.

Запропоновані алгоритми повинні працювати узгоджено. Створення замовлення пов'язане з перевіркою ресурсів, підтвердження - з резервуванням, завершення - зі списанням, скасування - зі зняттям резерву, а оплата - зі зміною фінансового статусу замовлення. Такий підхід дозволяє забезпечити цілісність роботи системи та уникнути суперечностей у даних.

У межах підрозділу було спроектовано основні алгоритми роботи інформаційної системи управління малим бізнесом. Вони охоплюють оброблення замовлення, перевірку доступності ресурсів, резервування, завершення, скасування замовлення, облік оплат і визначення критичних залишків. Запропонована логіка дозволяє автоматизувати ключові процеси підприємства та забезпечити більш точне управління замовленнями і ресурсами.

2.5 Висновки до другого розділу

У другому розділі було виконано проектування інформаційної системи управління малим бізнесом з урахуванням динаміки замовлень та ресурсів. На основі аналізу предметної області визначено основні функціональні та нефункціональні вимоги до системи. До функціональних вимог віднесено авторизацію користувачів, ведення бази клієнтів, створення та супровід замовлень, контроль статусів, облік ресурсів, перевірку їх доступності, резервування, облік оплат і формування звітів. Нефункціональні вимоги охоплюють зручність використання, надійність, швидкодію, безпеку, масштабованість і цілісність даних.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 46
Зм.	Арк.	№ докум.	Підпис	Дата		

Було спроектовано архітектуру інформаційної системи. Для розроблюваного програмного засобу обрано клієнт-серверну структуру, у якій клієнтська частина відповідає за взаємодію з користувачем, серверна частина реалізує бізнес-логіку, а база даних забезпечує зберігання інформації. Такий підхід дозволяє розділити основні компоненти системи, спростити її подальшу підтримку та забезпечити можливість розширення функціональності.

Також було визначено основні модулі інформаційної системи: модуль авторизації, модуль клієнтів, модуль замовлень, модуль ресурсів, модуль оплат, модуль звітності та модуль адміністрування. Кожен із цих модулів виконує окрему групу задач, але всі вони пов'язані між собою через єдину базу даних. Це дозволяє уникнути дублювання інформації та забезпечити узгоджену роботу системи.

У межах розділу було спроектовано базу даних системи. Визначено основні сутності: користувачі, клієнти, замовлення, позиції замовлення, ресурси, платежі та рух ресурсів. Для цих сутностей встановлено логічні зв'язки, що забезпечують коректне зберігання й оброблення даних. Запропонована структура бази даних дозволяє фіксувати весь цикл роботи із замовленням: від його створення до виконання, оплати та списання ресурсів.

Окремо було розглянуто алгоритми оброблення замовлень і ресурсів. Спроектовано послідовність створення замовлення, перевірки доступності ресурсів, резервування, завершення та скасування замовлення. Також визначено логіку обліку оплат і контролю критичних залишків ресурсів. Запропоновані алгоритми дозволяють зменшити кількість помилок під час ручного обліку та забезпечити більш точне управління ресурсами підприємства.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ МАЛИМ БІЗНЕСОМ

3.1 Обґрунтування вибору програмних засобів реалізації

Для програмної реалізації інформаційної системи управління малим бізнесом необхідно обрати такі засоби розроблення, які забезпечують зручне створення web-додатку, роботу з базою даних, оброблення запитів користувача та можливість подальшого розширення системи. Оскільки розроблювана система повинна підтримувати роботу з клієнтами, замовленнями, ресурсами, платежами та звітами, доцільно використати клієнт-серверний підхід, у якому окремо реалізуються клієнтська частина, серверна частина та база даних.

Як основну платформу для серверної частини доцільно використати Node.js. Це середовище виконання JavaScript, яке дозволяє створювати серверні додатки та обробляти запити від клієнтської частини. Його перевагою є те, що одна мова програмування може використовуватися як на клієнтському, так і на серверному рівні. Для навчального проєкту це зручно, оскільки зменшується складність розроблення та спрощується підтримка програмного коду [18].

Для побудови серверного API обрано Express.js. Це web-фреймворк для Node.js, який дає змогу створювати маршрути, обробляти HTTP-запити, підключати проміжне програмне забезпечення та організувати взаємодію між клієнтською частиною й базою даних. У межах розроблюваної системи Express.js використовується для реалізації маршрутів авторизації, роботи з клієнтами, замовленнями, ресурсами, платежами та звітами [19].

Клієнтську частину інформаційної реалізуємо з використанням React. React є бібліотекою JavaScript для створення користувацьких інтерфейсів. Його використання дозволяє розділити інтерфейс на окремі компоненти: таблиці, форми, кнопки, картки статистики, панель навігації та сторінки системи. Такий підхід спрощує розроблення, оскільки кожен компонент можна створювати й перевіряти окремо [20].

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 48
Зм.	Арк.	№ докум.	Підпис	Дата		

Для оформлення інтерфейсу може використовуватися CSS або Tailwind CSS. У роботі доцільно застосувати простий адаптивний дизайн, який не перевантажує користувача зайвими елементами. Інтерфейс повинен бути зрозумілим для працівників малого підприємства, тому основна увага приділяється зручному розміщенню таблиць, форм і навігації [21].

Для зберігання даних обрано реляційну базу даних PostgreSQL або SQLite. PostgreSQL є більш потужним варіантом і добре підходить для систем, у яких потрібно зберігати структуровані дані, встановлювати зв'язки між таблицями та забезпечувати цілісність інформації. SQLite може використовуватися як простіший варіант для навчального або демонстраційного запуску, оскільки не потребує складного налаштування окремого сервера бази даних [22, 23].

У межах кваліфікаційної роботи доцільно обрати PostgreSQL як основну систему керування базами даних, оскільки вона підтримує реляційну модель, зовнішні ключі, транзакції, індекси та складні запити. Це важливо для інформаційної системи, у якій замовлення пов'язані з клієнтами, ресурси - з позиціями замовлення, а платежі - з конкретними замовленнями.

Для взаємодії серверної частини з базою даних може використовуватися Prisma ORM. ORM-засоби дозволяють працювати з таблицями бази даних через програмні моделі, а не писати всі SQL-запити вручну. Це спрощує створення, читання, оновлення та видалення записів. Крім того, Prisma дозволяє описати структуру бази даних у вигляді схем, що робить її більш зрозумілою для подальшої підтримки [24].

Для обміну даними між клієнтською та серверною частинами використовується REST API. Такий підхід передбачає, що клієнтська частина надсилає HTTP-запити до серверних маршрутів, а сервер повертає відповідь у форматі JSON. Наприклад, для отримання списку клієнтів використовується запит GET /api/customers, для створення замовлення - POST /api/orders, для отримання списку ресурсів - GET /api/resources, а для формування звітів - GET /api/reports/dashboard.

Для авторизації користувачів може бути використано JWT. JSON Web Token дозволяє після успішного входу користувача передавати клієнтській частині спеціальний токен, який надалі використовується для підтвердження доступу до захищених маршрутів [26]. Такий підхід є поширеним у web-додатках і дозволяє реалізувати розмежування прав доступу залежно від ролі користувача.

У процесі розроблення також використовується Visual Studio Code. Це середовище розробки підтримує JavaScript, React, Node.js, роботу з Git, форматування коду та встановлення додаткових розширень. Для навчального проєкту воно є зручним, оскільки дозволяє працювати з клієнтською та серверною частинами в одному робочому просторі [27].

Для перевірки роботи API доцільно використовувати Postman або вбудовані засоби браузера [28]. Postman дозволяє надсилати запити до серверної частини, перевіряти відповіді, тестувати створення клієнтів, замовлень, ресурсів і оплат. Це особливо корисно на етапі тестування серверної логіки, коли клієнтський інтерфейс ще не повністю реалізований.

Для контролю версій використовується Git [29]. Він дозволяє зберігати історію змін проєкту, повертатися до попередніх версій, створювати окремі гілки для розроблення нових функцій і зменшувати ризик втрати коду. Для кваліфікаційної роботи використання Git також є корисним, оскільки демонструє організований підхід до розроблення програмного забезпечення. Основні програмні засоби реалізації системи наведено в таблиці 3.1.

Таблиця 3.1 – Програмні засоби реалізації інформаційної системи

Програмний засіб	Призначення	Обґрунтування вибору
Node.js	Реалізація серверної частини	Дозволяє створювати серверні додатки мовою JavaScript
Express.js	Побудова REST API	Спрощує створення маршрутів і оброблення HTTP-запитів

Кінець таблиці 3.1

React	Реалізація клієнтської частини	Дає змогу створювати інтерфейс із багаторазових компонентів
CSS / Tailwind CSS	Оформлення інтерфейсу	Забезпечує просте й адаптивне оформлення сторінок
PostgreSQL	Зберігання даних	Підтримує реляційну модель, зв'язки, індекси та транзакції
Prisma ORM	Робота з базою даних	Спрощує взаємодію серверної частини з таблицями БД
REST API	Обмін даними між клієнтом і сервером	Забезпечує зрозумілу структуру взаємодії компонентів
JWT	Авторизація користувачів	Дозволяє реалізувати захищений доступ і ролі користувачів
Visual Studio Code	Середовище розробки	Підтримує роботу з JavaScript, React, Node.js і Git
Postman	Тестування API	Дозволяє перевіряти запити до серверної частини
Git	Контроль версій	Забезпечує збереження історії змін програмного коду

Обраний набір технологій відповідає задачам розроблюваної інформаційної системи. Він дозволяє реалізувати web-додаток із поділом на клієнтську та серверну частини, забезпечити роботу з базою даних, реалізувати авторизацію користувачів, оброблення замовлень, контроль ресурсів і формування звітів. При цьому технології є достатньо поширеними, мають велику кількість документації та підходять для навчального проєкту.

Таким чином, для реалізації інформаційної системи управління малим бізнесом обрано сучасний і практичний стек програмних засобів: React для клієнтської частини, Node.js та Express.js для серверної частини, PostgreSQL для бази даних, Prisma ORM для роботи з даними, REST API для взаємодії компонентів і JWT для авторизації. Такий вибір дозволяє реалізувати систему, яка відповідає поставленим функціональним і нефункціональним вимогам та може бути використана як основа для подальшого розвитку. Крім того, використання сучасних інструментів розробки сприяє підвищенню продуктивності системи, її надійності та безпеки, що є важливими аспектами для ефективного функціонування малого бізнесу.

3.2 Реалізація серверної частини інформаційної системи

Серверна частина інформаційної системи управління малим бізнесом призначена для оброблення запитів користувачів, реалізації основної бізнес-логіки, перевірки введених даних, взаємодії з базою даних і повернення результатів клієнтській частині. Вона є проміжним рівнем між інтерфейсом користувача та базою даних, тому саме на серверному рівні виконуються основні правила роботи системи: створення замовлення, перевірка доступності ресурсів, резервування, завершення замовлення, облік оплат і формування звітів.

Серверну частину доцільно реалізувати з використанням Node.js та Express.js. Node.js забезпечує виконання JavaScript-коду на сервері, а Express.js дозволяє створювати маршрути API, обробляти HTTP-запити та організовувати структуру web-додатку. У розроблюваній системі серверна частина приймає запити від клієнтської частини, перевіряє права користувача, виконує необхідні операції з даними та повертає відповідь у форматі JSON [30].

Структура серверної частини повинна бути поділена на окремі логічні блоки. Такий підхід спрощує розроблення та подальшу підтримку системи. Наприклад, маршрути, контролери, моделі, проміжне програмне забезпечення та налаштування бази даних доцільно зберігати в окремих папках. Це дозволяє швидко знаходити потрібні файли й уникати змішування різних частин програмної логіки.

У такій структурі папка controllers містить файли, у яких реалізовано логіку оброблення запитів. Папка routes містить маршрути API. Папка middleware використовується для перевірки авторизації, ролей користувачів і оброблення помилок. Папка database містить налаштування підключення до бази даних. Окреме розділення функціональних компонентів забезпечує зрозумілу організацію проєкту та спрощує його подальшу підтримку. Така архітектура дозволяє легко вносити зміни до окремих модулів без впливу на роботу інших частин системи, що підвищує гнучкість розробки та полегшує масштабування застосунку.

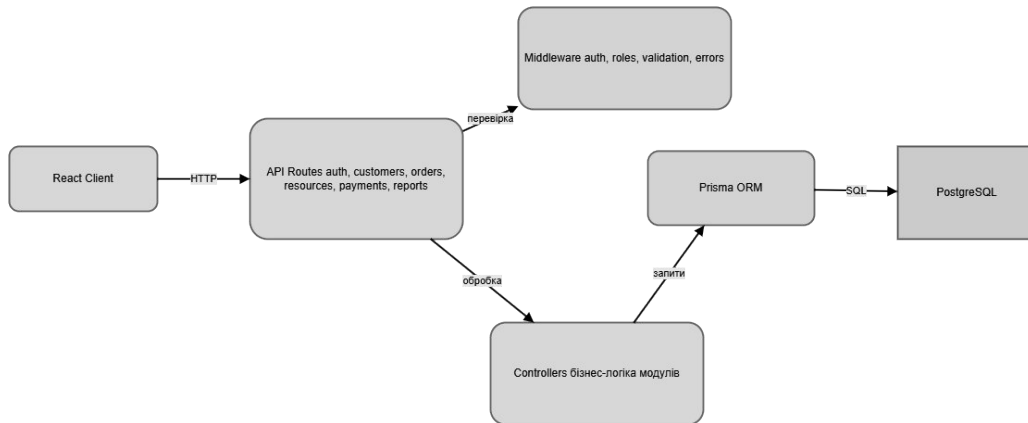


Рисунок 3.1 – Структура серверної частини інформаційної системи

Основним способом взаємодії між клієнтом і сервером є REST API. Кожен ресурс системи має власну групу маршрутів. Наприклад, для клієнтів використовуються маршрути `/api/customers`, для замовлень - `/api/orders`, для ресурсів - `/api/resources`, для оплат - `/api/payments`, а для звітів - `/api/reports`. Такий підхід робить структуру API зрозумілою та логічною. Основні API-запити інформаційної системи наведено в таблиці 3.2.

Таблиця 3.2 – Основні API-запити інформаційної системи

Метод	Маршрут	Призначення
POST	<code>/api/auth/login</code>	Авторизація користувача
GET	<code>/api/auth/me</code>	Отримання даних поточного користувача
GET	<code>/api/customers</code>	Отримання списку клієнтів
POST	<code>/api/customers</code>	Створення нового клієнта
PUT	<code>/api/customers/:id</code>	Редагування клієнта
DELETE	<code>/api/customers/:id</code>	Видалення клієнта
GET	<code>/api/orders</code>	Отримання списку замовлень
POST	<code>/api/orders</code>	Створення нового замовлення
GET	<code>/api/orders/:id</code>	Отримання деталей замовлення
POST	<code>/api/orders/:id/confirm</code>	Підтвердження замовлення
POST	<code>/api/orders/:id/complete</code>	Завершення замовлення
POST	<code>/api/orders/:id/cancel</code>	Скасування замовлення
GET	<code>/api/resources</code>	Отримання списку ресурсів
POST	<code>/api/resources</code>	Додавання ресурсу
PUT	<code>/api/resources/:id</code>	Редагування ресурсу

Кінець таблиці 3.2

GET	/api/resources/low-stock	Отримання ресурсів із критичним залишком
POST	/api/payments	Додавання оплати
GET	/api/reports/dashboard	Отримання даних для головної панелі
GET	/api/reports/orders-dynamics	Отримання динаміки замовлень

Одним із перших реалізується модуль авторизації. Користувач вводить логін і пароль, після чого сервер перевіряє ці дані в базі даних. Якщо користувач існує, а пароль правильний, сервер формує токен доступу та повертає його клієнтській частині. Надалі цей токен передається в заголовках запитів і використовується для перевірки прав доступу.

Для захисту маршрутів використовується middleware авторизації. Він перевіряє, чи є в запиті дійсний токен. Якщо токен відсутній або неправильний, сервер повертає помилку доступу. Якщо токен правильний, middleware додає інформацію про користувача до запиту, і система дозволяє виконати потрібну операцію. Для окремих маршрутів може також використовуватися перевірка ролі користувача.

Наприклад, маршрути для перегляду звітів можуть бути доступні керівнику та адміністратору, а маршрути для редагування ресурсів - адміністратору та працівнику складу. Це дозволяє реалізувати розмежування прав доступу відповідно до функціональних обов'язків користувачів.

Модуль клієнтів забезпечує створення, перегляд, редагування та видалення клієнтів. Під час створення клієнта сервер перевіряє, чи заповнені основні поля, наприклад ім'я та телефон. Якщо дані коректні, запис зберігається в базі даних. Якщо дані некоректні, користувачу повертається повідомлення про помилку.

Модуль замовлень є найбільш складним, оскільки він пов'язаний із клієнтами, ресурсами, платежами та статусами. Під час створення замовлення

сервер отримує дані про клієнта, список позицій, кількість, строк виконання та коментар. Після цього система розраховує загальну суму замовлення та створює записи в таблицях Order і OrderItem.

Підтвердження замовлення виконується окремим серверним маршрутом. Це важливо, оскільки саме на цьому етапі потрібно перевірити доступність ресурсів і виконати резервування. Сервер отримує замовлення з усіма позиціями, для кожного ресурсу визначає доступну кількість і порівнює її з необхідною кількістю. Якщо ресурсів достатньо, система збільшує reservedQuantity, змінює статус замовлення на confirmed і створює записи в таблиці руху ресурсів.

Якщо ресурсів недостатньо, сервер не виконує резервування й повертає клієнтській частині повідомлення про помилку. У повідомленні доцільно вказати, якого ресурсу не вистачає та яку кількість потрібно мати для підтвердження замовлення. Це робить роботу користувача більш зрозумілою.

Завершення замовлення також виконується окремим маршрутом. Після завершення система списує ресурси: зменшує загальну кількість і зарезервовану кількість. Також створюються записи руху ресурсів із типом write_off, а статус замовлення змінюється на completed. Такий підхід дозволяє зберігати історію змін і контролювати використання ресурсів.

Скасування замовлення залежить від його поточного статусу. Якщо замовлення ще не було підтвержене, достатньо змінити його статус на cancelled. Якщо ресурси вже були зарезервовані, система повинна зменшити reservedQuantity для кожного ресурсу та створити записи руху з типом cancel_reserve. Це потрібно для того, щоб ресурси знову стали доступними для інших замовлень.

Модуль оплат відповідає за додавання платежів і оновлення статусу оплати замовлення. Після створення платежу сервер підраховує загальну суму оплат за конкретним замовленням. Якщо оплат немає, статус залишається unpaid. Якщо оплачено частину суми, встановлюється статус partial. Якщо сума оплат дорівнює або перевищує загальну суму замовлення, статус змінюється на paid.

Модуль звітності формує узагальнені дані для керівника. Сервер може повертати кількість замовлень за період, суму виконаних замовлень, кількість активних замовлень, список ресурсів із критичним залишком і динаміку замовлень за днями. Такі дані використовуються на головній панелі та сторінці звітів.

Важливою частиною серверної реалізації є оброблення помилок. Сервер повинен повертати зрозумілі повідомлення у випадках, коли користувач ввів неправильні дані, не має прав доступу, намагається підтвердити замовлення без ресурсів або звертається до неіснуючого запису. Це підвищує стабільність роботи системи та полегшує тестування.

Також потрібно передбачити валідацію даних. Наприклад, кількість ресурсу не може бути від'ємною, сума оплати повинна бути більшою за нуль, email повинен мати правильний формат, а замовлення не може бути створене без клієнта. Частина перевірок може виконуватися на клієнтській частині, але основна валідація обов'язково повинна бути на сервері, оскільки саме сервер відповідає за коректність даних у базі.

Серверна частина інформаційної системи реалізує основну логіку роботи програмного засобу. Вона забезпечує авторизацію користувачів, оброблення клієнтів, замовлень, ресурсів, оплат і звітів, а також виконує перевірку доступності ресурсів, резервування, списання та скасування замовлень. Запропонована структура серверної частини є зрозумілою, модульною та придатною для подальшого розширення системи.

3.3 Реалізація бази даних

Реалізація бази даних є одним із основних етапів створення інформаційної системи управління малим бізнесом, оскільки саме база даних забезпечує зберігання всієї інформації, необхідної для роботи системи. У межах розроблюваного програмного засобу в базі даних зберігаються користувачі, клієнти, замовлення, позиції замовлень, ресурси, оплати та записи про рух

ресурсів. Ці дані пов'язані між собою, тому для їх зберігання доцільно використовувати реляційну структуру.

Для реалізації бази даних було обрано PostgreSQL. Ця система керування базами даних підтримує таблиці, первинні та зовнішні ключі, індекси, транзакції та обмеження цілісності. Такі можливості є важливими для розроблюваної системи, оскільки в ній необхідно забезпечити правильні зв'язки між клієнтами, замовленнями, ресурсами й платежами [32].

У базі даних інформаційної системи передбачено кілька основних таблиць. Таблиця users використовується для зберігання облікових записів користувачів. У ній містяться дані про ім'я користувача, електронну пошту, захищений пароль, роль і дату створення запису. Роль користувача визначає його права доступу до функцій системи [33].

Таблиця customers призначена для зберігання інформації про клієнтів малого підприємства. До неї заносяться ім'я або назва клієнта, телефон, електронна пошта, адреса, примітки та дата створення запису. Ця таблиця пов'язана з таблицею замовлень, оскільки один клієнт може мати декілька замовлень.

Таблиця orders використовується для зберігання основної інформації про замовлення. Вона містить ідентифікатор клієнта, статус замовлення, загальну суму, статус оплати, строк виконання, коментар, дату створення та дату оновлення. Кожне замовлення обов'язково пов'язується з конкретним клієнтом.

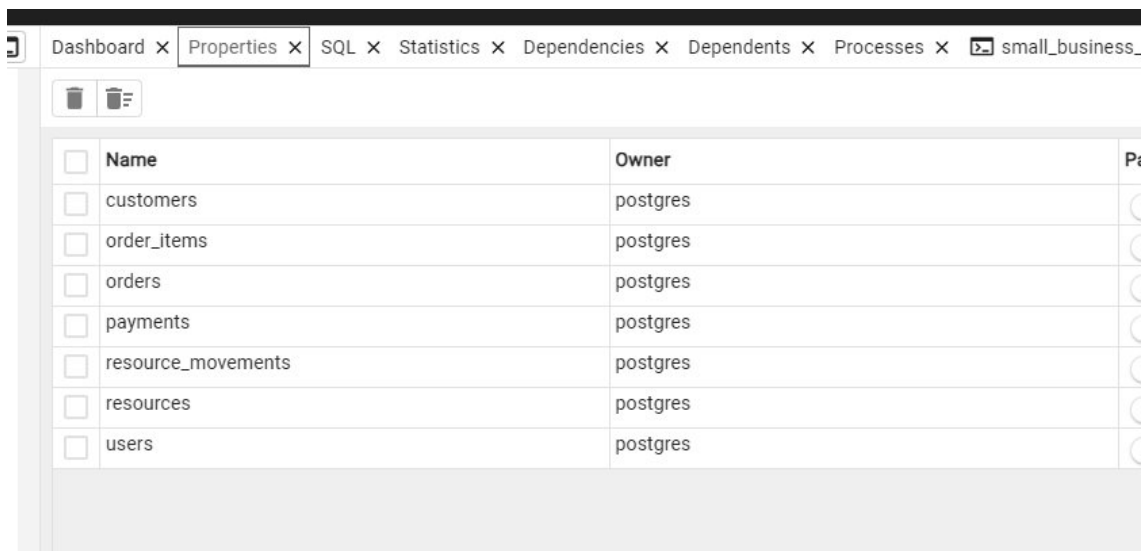
Для зберігання складу замовлення використовується таблиця order_items. Вона містить перелік ресурсів або товарів, які входять до конкретного замовлення. У цій таблиці зберігаються ідентифікатор замовлення, ідентифікатор ресурсу, кількість, ціна та проміжна сума. Такий підхід дозволяє створювати замовлення з кількома позиціями.

Таблиця resources призначена для обліку ресурсів підприємства. У ній зберігаються назва ресурсу, категорія, одиниця вимірювання, загальна кількість, зарезервована кількість, мінімальний залишок, ціна та статус. Саме на основі цієї

таблиці система визначає, чи достатньо ресурсів для підтвердження нового замовлення.

Таблиця `payments` використовується для обліку оплат. Вона містить ідентифікатор замовлення, суму платежу, спосіб оплати, статус, дату оплати та дату створення запису. Одне замовлення може мати декілька оплат, що дозволяє враховувати часткову оплату.

Для фіксації змін у кількості ресурсів використовується таблиця `resource_movements`. У ній зберігаються записи про резервування, списання, скасування резерву або поповнення ресурсу. Це дозволяє переглядати історію руху ресурсів і пояснювати зміни залишків. Загальну реалізовану структуру бази даних можна подати на рисунку 3.2.



<input type="checkbox"/>	Name	Owner	Pr
<input type="checkbox"/>	customers	postgres	(
<input type="checkbox"/>	order_items	postgres	(
<input type="checkbox"/>	orders	postgres	(
<input type="checkbox"/>	payments	postgres	(
<input type="checkbox"/>	resource_movements	postgres	(
<input type="checkbox"/>	resources	postgres	(
<input type="checkbox"/>	users	postgres	(

Рисунок 3.2 – Реалізована структура бази даних

У процесі реалізації бази даних також важливо правильно визначити типи даних. Для ідентифікаторів можуть використовуватися числові типи або `UUID`. Для текстових полів застосовуються типи `varchar` або `text`. Для сум і цін доцільно використовувати тип `decimal`, оскільки він краще підходить для фінансових значень. Для дат створення та оновлення записів використовується тип `timestamp`.

Приклад SQL-структури таблиці ресурсів може мати такий вигляд:

```

CREATE TABLE resources (
  id SERIAL PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  category VARCHAR(100),
  unit VARCHAR(30) NOT NULL,
  quantity DECIMAL(10,2) NOT NULL DEFAULT 0,
  reserved_quantity DECIMAL(10,2) NOT NULL DEFAULT 0,
  min_quantity DECIMAL(10,2) NOT NULL DEFAULT 0,
  price DECIMAL(10,2) NOT NULL DEFAULT 0,
  status VARCHAR(30) DEFAULT 'active',
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

У цій таблиці поле quantity відповідає за загальну кількість ресурсу, а поле reserved_quantity - за кількість, зарезервовану під активні замовлення. Доступна кількість ресурсу визначається як різниця між цими двома значеннями. Поле min_quantity використовується для виявлення критичного залишку.

Приклад створення таблиці замовлень:

```

CREATE TABLE orders (
  id SERIAL PRIMARY KEY,
  customer_id INTEGER NOT NULL REFERENCES customers(id),
  status VARCHAR(30) NOT NULL DEFAULT 'new',
  total_amount DECIMAL(10,2) NOT NULL DEFAULT 0,
  payment_status VARCHAR(30) NOT NULL DEFAULT 'unpaid',
  deadline DATE,
  comment TEXT,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
  updated_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

```

У таблиці orders поле status визначає стан замовлення. Воно може набувати значень new, confirmed, in_progress, waiting_payment, completed або cancelled. Поле payment_status показує стан оплати: unpaid, partial або paid.

Приклад таблиці позицій замовлення:

```

CREATE TABLE order_items (
  id SERIAL PRIMARY KEY,
  order_id INTEGER NOT NULL REFERENCES orders(id) ON DELETE
  CASCADE,
  resource_id INTEGER NOT NULL REFERENCES resources(id),
  quantity DECIMAL(10,2) NOT NULL,
  price DECIMAL(10,2) NOT NULL,
  subtotal DECIMAL(10,2) NOT NULL
);

```

Ця таблиця дозволяє додавати до одного замовлення кілька позицій. Якщо замовлення видаляється, пов'язані з ним позиції також можуть бути видалені за допомогою правила ON DELETE CASCADE.

Для прискорення пошуку даних доцільно створити індекси. Наприклад, індекс для поля customer_id у таблиці orders дозволяє швидше отримувати історію замовлень конкретного клієнта. Індекс для поля status дозволяє швидше фільтрувати замовлення за поточним станом. Індекс для поля resource_id у таблиці order_items спрощує пошук використання певного ресурсу.

Приклади індексів:

```
CREATE INDEX idx_orders_customer_id ON orders(customer_id);
CREATE INDEX idx_orders_status ON orders(status);
CREATE INDEX idx_order_items_resource_id ON
order_items(resource_id);
CREATE INDEX idx_payments_order_id ON payments(order_id);
```

Під час реалізації бази даних також потрібно передбачити тестові дані. Вони потрібні для перевірки роботи системи без необхідності вручну створювати всі записи. Наприклад, можна додати кількох клієнтів, декілька ресурсів, кілька замовлень із різними статусами та кілька оплат. Це дає змогу перевірити роботу сторінок клієнтів, замовлень, ресурсів і звітів.

До прикладів тестових клієнтів можна віднести: Іван Петренко, Олена Коваль, ТОВ «Майстер Плюс», ФОП Бондаренко. До ресурсів можна додати пакувальні коробки, каву мелену, папір офісний, деталь А-100, послугу доставки та робочий час майстра. Такий набір даних дозволяє показати роботу системи для різних типів малого бізнесу.

Окрему увагу в базі даних потрібно приділити операціям із ресурсами. Під час підтвердження замовлення система збільшує зарезервовану кількість ресурсу. Під час завершення замовлення вона зменшує і загальну кількість, і зарезервовану кількість. Під час скасування замовлення система зменшує тільки зарезервовану кількість. Такі правила дозволяють коректно контролювати залишки.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 60
Зм.	Арк.	№ докум.	Підпис	Дата		

Для збереження історії змін використовується таблиця `resource_movements`. Наприклад, якщо ресурс було зарезервовано для замовлення, у таблиці створюється запис із типом `reserve`. Якщо ресурс списано після виконання замовлення, створюється запис із типом `write_off`. Якщо резерв було скасовано, створюється запис із типом `cancel_reserve`. Це дозволяє аналізувати рух ресурсів і відновлювати логіку змін у разі потреби.

У межах реалізації бази даних було визначено основні таблиці, їхні поля, зв'язки та правила оброблення даних. Запропонована структура забезпечує зберігання інформації про користувачів, клієнтів, замовлення, ресурси, оплати та рух ресурсів. Використання реляційної бази даних дозволяє забезпечити цілісність інформації, уникнути дублювання та створити надійну основу для роботи всієї інформаційної системи.

3.4 Реалізація клієнтської частини системи

Клієнтська частина інформаційної системи призначена для взаємодії користувача з основними функціями програмного засобу. Через інтерфейс користувач може авторизуватися, переглядати клієнтів, створювати замовлення, контролювати ресурси, додавати оплати та формувати звіти.

Для реалізації клієнтської частини використано `React`. Ця бібліотека дозволяє будувати `web`-інтерфейс із окремих компонентів: таблиць, форм, кнопок, інформаційних карток, модальних вікон і сторінок. Такий підхід спрощує розроблення та подальше редагування інтерфейсу [18].

Клієнтська частина взаємодіє із сервером через `REST API`. Для отримання або зміни даних вона надсилає `HTTP`-запити до відповідних маршрутів.

Структура клієнтської частини включає сторінки авторизації, головної панелі, клієнтів, замовлень, ресурсів, оплат і звітів. Також у системі передбачено спільні компоненти, які повторно використовуються на різних сторінках: меню навігації, таблиці, форми введення, кнопки дій і повідомлення користувачу [34, 35].

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 61
Зм.	Арк.	№ докум.	Підпис	Дата		

Першою сторінкою є сторінка авторизації. Користувач вводить електронну пошту та пароль. Якщо дані правильні, система відкриває головну панель. Якщо дані неправильні, відображається повідомлення про помилку.

КВАЛІФІКАЦІЙНА РОБОТА

Інформаційна система управління малим бізнесом

Клієнти, замовлення, ресурси, оплати і звітність в одному локальному застосунку.

Email
admin@business.local

Пароль

Увійти

Тестовий доступ: admin@business.local / admin123

Рисунок 3.3 – Форма авторизації користувача

Головна панель відображає основні показники роботи підприємства: кількість активних замовлень, виконані замовлення, суму оплат, кількість клієнтів і ресурси з критичним залишком. Це дозволяє користувачу швидко оцінити поточний стан системи.

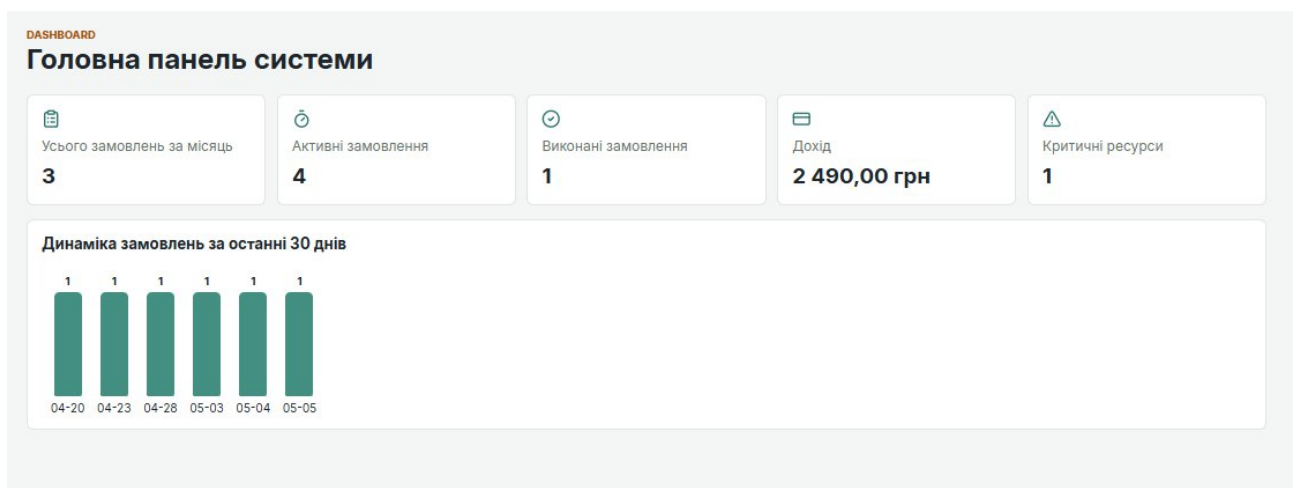


Рисунок 3.4 – Головна панель інформаційної системи

Сторінка клієнтів дає змогу переглядати список клієнтів, додавати нові записи, редагувати дані та виконувати пошук. Дані клієнтів відображаються у вигляді таблиці, що спрощує роботу з клієнтською базою.

Сторінка замовлень є однією з основних у системі. Вона дозволяє створювати нові замовлення, переглядати їх список, фільтрувати записи за статусом і відкривати деталі конкретного замовлення.

ID	КЛІЄНТ	СТАТУС	ОПЛАТА	СУМА	ДЕДЛАЙН	
#6	ТОВ "Майстер Плюс"	Ново	Не оплачено	510,00 грн	03.05.2026	Ред.
#1	Іван Петренко	Ново	Не оплачено	366,00 грн	10.05.2026	Ред.
#2	Олена Коваль	Очікує оплати	Частково	1 184,00 грн	08.05.2026	
#3	ТОВ "Майстер Плюс"	У роботі	Частково	4 030,00 грн	06.05.2026	
#4	ФОП Бондаренко	Виконане	Оплачено	690,00 грн	07.05.2026	
#5	Іван Петренко	Скасоване	Не оплачено	1 660,00 грн	13.05.2026	

Рисунок 3.5 – Сторінка керування замовленнями

Сторінка ресурсів використовується для контролю товарів, матеріалів або інших ресурсів підприємства. У таблиці відображаються загальна кількість, зарезервована кількість, доступний залишок, мінімальний залишок і ціна. Якщо ресурс має критичний залишок, система візуально позначає його.

РЕСУРС	К-СТЬ	РЕЗЕРВ	ДОСТУПНО	ЦІНА	
Папір офісний Канцелярія	6 пачка	0	6	180,00 грн	
Деталь А-100 Комплектуючі	75 шт	14	61	95,00 грн	
Пакувальні коробки Пакування	120 шт	8	112	18,00 грн	
Послуга доставки Послуги	998 послуга	0	998	150,00 грн	
Робочий час майстра Послуги	160 год	6	154	450,00 грн	
Кава мелена Товари	45 кг	4	41	260,00 грн	

Рисунок 3.6 – Сторінка керування ресурсами

Сторінка оплат дозволяє додавати платежі до замовлень і контролювати статус оплати. Якщо внесена частина суми, замовлення отримує статус «частково оплачено». Якщо сума оплачена повністю, статус змінюється на «оплачено».

Сторінка звітів відображає узагальнену інформацію про роботу підприємства: кількість замовлень за період, активні та виконані замовлення, суму оплат і ресурси з критичним залишком. Такі дані допомагають керівнику швидко оцінити стан малого бізнесу.

Також у клієнтській частині враховано ролі користувачів. Залежно від ролі користувач може мати доступ до різних розділів і дій. Наприклад, адміністратор має повний доступ, менеджер працює із клієнтами та замовленнями, працівник складу - з ресурсами, а керівник - зі звітами.

Клієнтська частина інформаційної системи забезпечує зручну роботу з основними модулями програмного засобу. Використання React дозволяє створити структурований інтерфейс, який взаємодіє із серверною частиною через REST API та забезпечує доступ до функцій управління клієнтами, замовленнями, ресурсами, платежами й звітами.

3.5 Реалізація модулів управління замовленнями та ресурсами

Модулі управління замовленнями та ресурсами є основними в розроблюваній інформаційній системі, оскільки саме вони забезпечують зв'язок між потребами клієнтів і можливостями підприємства. Модуль замовлень відповідає за створення, редагування, підтвердження, виконання та скасування замовлень. Модуль ресурсів забезпечує облік наявних товарів, матеріалів або послуг, які використовуються для виконання цих замовлень.

Під час створення нового замовлення користувач вибирає клієнта, додає потрібні позиції, зазначає кількість, строк виконання та коментар. Після цього

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 64
Зм.	Арк.	№ докум.	Підпис	Дата		

система автоматично розраховує загальну суму замовлення. На цьому етапі замовлення отримує статус «нове» і ще не впливає на залишки ресурсів.

Основна перевірка виконується під час підтвердження замовлення. Система аналізує кожну позицію та визначає, чи достатньо доступної кількості ресурсу. Доступна кількість обчислюється як різниця між загальною кількістю ресурсу та зарезервованою кількістю. Якщо ресурсів достатньо, система підтверджує замовлення та резервує необхідну кількість. Якщо хоча б одного ресурсу не вистачає, підтвердження не виконується, а користувач отримує повідомлення про помилку [37].

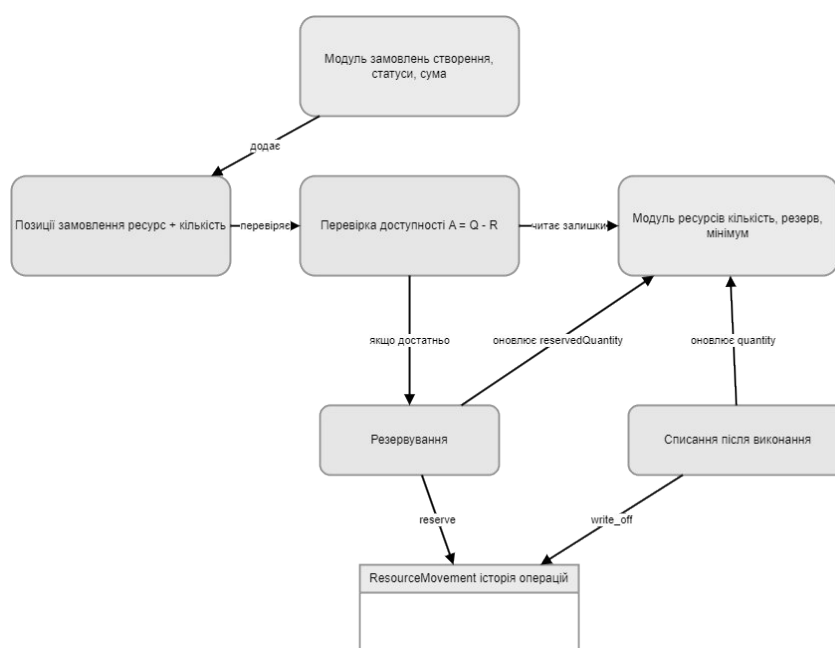


Рисунок 3.7 – Схема взаємодії модуля замовлень і модуля ресурсів

Резервування ресурсів дозволяє уникнути ситуації, коли один і той самий ресурс використовується для кількох активних замовлень одночасно. Наприклад, якщо на складі є 30 одиниць товару, а 10 одиниць уже зарезервовано, то для нових замовлень доступними є лише 20 одиниць. Така логіка підвищує точність обліку та допомагає підприємству реалістично оцінювати свої можливості.

Після фактичного виконання замовлення користувач змінює його статус на «виконане». У цей момент система списує використані ресурси: зменшує

Зм.	Арк.	№ докум.	Підпис	Дата

загальну кількість і одночасно зменшує зарезервовану кількість. Також створюється запис у таблиці руху ресурсів, щоб у системі залишалася історія операцій.

Якщо замовлення скасовується, система перевіряє, чи були ресурси зарезервовані. Якщо резерв був створений, він знімається, а зарезервована кількість ресурсу зменшується. Після цього замовлення отримує статус «скасоване». Такий механізм дозволяє повернути ресурси в доступний залишок і використати їх для інших замовлень [38].

Модуль ресурсів також контролює критичні залишки. Для кожного ресурсу задається мінімальна допустима кількість. Якщо фактична кількість ресурсу стає меншою або дорівнює цьому значенню, система позначає ресурс як проблемний. Це допомагає працівнику або керівнику швидко визначити, які запаси потрібно поповнити.

Модуль замовлень пов'язаний також із модулем оплат. Для кожного замовлення система зберігає загальну суму та статус оплати. Якщо оплата відсутня, замовлення має статус «не оплачено». Якщо внесена частина суми, статус змінюється на «частково оплачено». Після повного розрахунку статус змінюється на «оплачено».

У результаті реалізовані модулі забезпечують повний цикл роботи із замовленням: створення, перевірку ресурсів, підтвердження, резервування, виконання, оплату та закриття. Це дозволяє зменшити кількість ручних операцій, підвищити точність обліку та зробити роботу малого підприємства більш керованою.

3.6 Демонстрація роботи та тестування інформаційної системи

Після реалізації основних модулів інформаційної системи було проведено демонстрацію її роботи та тестування основних функціональних можливостей. Метою цього етапу є перевірка того, чи коректно система виконує основні дії

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 66
Зм.	Арк.	№ докум.	Підпис	Дата		

користувача: авторизацію, роботу з клієнтами, створення замовлення, перевірку ресурсів, резервування, завершення замовлення, додавання оплати та формування звітів.

Тестування виконувалося за сценаріями, які відповідають типовій роботі малого підприємства. Такий підхід дозволяє перевірити не окремі елементи інтерфейсу, а повний ланцюг роботи системи: від входу користувача до перегляду результатів у звітах. Для кожного сценарію фіксувався очікуваний результат і фактичний результат роботи системи [39].

Першим етапом було перевірено авторизацію користувача. На сторінці входу користувач вводить електронну пошту та пароль. Якщо дані введені правильно, система відкриває головну панель. Якщо логін або пароль неправильні, користувач отримує повідомлення про помилку. Це підтверджує, що доступ до системи мають лише зареєстровані користувачі.

КВАЛІФІКАЦІЙНА РОБОТА
**Інформаційна система
управління малим
бізнесом**
Клієнти, замовлення, ресурси, оплати і звітність в
одному локальному застосунку.

Failed to fetch

Email
admin@business.loc

Пароль
.....

Увійти

Тестовий доступ: admin@business.local / admin123

Рисунок 3.8 – Сторінка авторизації користувача з помилкою

Після успішної авторизації користувач потрапляє на головну панель системи. На ній відображаються основні показники: кількість активних замовлень, кількість виконаних замовлень, сума оплат, кількість клієнтів і ресурси з критичним залишком. Така сторінка дозволяє швидко оцінити поточний стан малого підприємства [40].

Наступним етапом було перевірено роботу з клієнтами. Користувач відкриває сторінку клієнтів, натискає кнопку додавання нового клієнта та заповнює форму. Після збереження запис з'являється в загальному списку клієнтів. Також перевірено пошук клієнта за іменем або телефоном.

The screenshot shows a search bar with the text 'Нов' and a 'Знайти' button. Below it is a table with columns 'КЛІЄНТ', 'ТЕЛЕФОН', and 'EMAIL'. The table contains one entry: 'Новий Новик', '0961153175', and 'teodonil2@gmail.com'. There are also edit and delete icons for this entry.

КЛІЄНТ	ТЕЛЕФОН	EMAIL
Новий Новик	0961153175	teodonil2@gmail.com

Рисунок 3.9 – Додавання нового клієнта до системи

Після створення клієнта було перевірено сценарій створення нового замовлення. Користувач переходить на сторінку замовлень, вибирає клієнта, додає позиції замовлення, зазначає кількість ресурсів і строк виконання. Система автоматично розраховує загальну суму замовлення. На цьому етапі замовлення отримує початковий статус «нове».

The screenshot shows a form titled 'Нове замовлення' (New Order) with a 'Назад' button. The form includes fields for 'Клієнт' (Client) set to 'Новий Новик' and 'Дедлайн' (Deadline) set to '20.05.2026'. There is a text area for 'Нове замовлення'. Below is a section for 'Ресурси замовлення' (Order Resources) with two items: 'Пакувальні коробки · доступно 112 шт' (Packaging boxes · available 112 pcs) with quantity 124, price 18, and total 2 232,00 грн; and 'Деталь А-100 · доступно 61 шт' (Part A-100 · available 61 pcs) with quantity 1, price 95, and total 95,00 грн. A '+ Додати ресурс' button and 'Разом: 2 327,00 грн' are also visible. At the bottom is a 'Зберегти замовлення' button.

Рисунок 3.10 – Створення нового замовлення

Одним із найважливіших сценаріїв є перевірка доступності ресурсів. Після додавання позицій користувач намагається підтвердити замовлення. Система перевіряє доступну кількість кожного ресурсу. Якщо ресурсів достатньо, замовлення підтверджується, а потрібна кількість ресурсів резервується. Якщо ресурсів недостатньо, система виводить повідомлення про помилку та не дозволяє підтвердити замовлення.

У разі достатньої кількості ресурсів система змінює статус замовлення на «підтверджене». Також у таблиці ресурсів збільшується зарезервована кількість. Це означає, що ресурс тимчасово закріплений за конкретним замовленням і не може бути використаний для інших активних замовлень.

Далі було перевірено роботу сторінки ресурсів. У списку ресурсів відображаються загальна кількість, зарезервована кількість, доступна кількість і мінімальний залишок. Якщо кількість ресурсу менша або дорівнює мінімальному значенню, система позначає його як ресурс із критичним залишком. Це дозволяє користувачу швидко визначити, які ресурси потрібно поповнити.

РЕСУРС	К-СТЬ	РЕЗЕРВ	ДОСТУПНО	ЦІНА
Папір офісний Канцелярія	6 пачка	0	6	180,00 грн

Рисунок 3.11 – Відображення ресурсів із критичним залишком

Після виконання замовлення користувач змінює його статус на «виконане». У цей момент система списує зарезервовані ресурси: загальна кількість ресурсу зменшується, а зарезервована кількість повертається до попереднього рівня. Такий сценарій підтверджує правильність роботи механізму списання ресурсів.







РЕСУРС	К-СТЬ	РЕЗЕРВ	ДОСТУПНО	ЦІНА	
Папір офісний Канцелярія	12 пачка	0	12	180,00 грн	 
Деталь А-100 Комплектуючі	75 шт	14	61	95,00 грн	 
Пакувальні коробки Пакування	108 шт	8	100	18,00 грн	 

Рисунок 3.12 – Завершення замовлення та списання ресурсів

Окремо було протестовано скасування замовлення. Якщо замовлення було підтвержене, але потім скасоване, система знімає резерв із відповідних ресурсів і змінює статус замовлення на «скасоване». Це потрібно для того, щоб зарезервовані ресурси знову стали доступними для інших замовлень.

Наступним етапом було перевірено додавання оплати. Користувач відкриває замовлення, натискає кнопку додавання платежу, вводить суму та спосіб оплати. Якщо внесена сума менша за загальну суму замовлення, система встановлює статус «частково оплачено». Якщо сума оплат дорівнює сумі замовлення, статус змінюється на «оплачено».

ФІНАНСИ
Оплати

Додати оплату

Виберіть замовлення

Сума

Готівка

+ Додати оплату

Журнал оплат

ЗАМОВЛЕННЯ	КЛІЄНТ	СУМА	МЕТОД	ДАТА
#8	Новий Новик	180,00 грн	cash	05.05.2026
#2	Олена Коваль	600,00 грн	card	04.05.2026
#3	ТОВ "Майстер Плюс"	1 200,00 грн	bank_transfer	01.05.2026
#4	ФОП Бондаренко	690,00 грн	cash	25.04.2026

Рисунок 3.13 – Додавання оплати до замовлення

Також було перевірено роботу сторінки звітів. Система формує зведену інформацію про кількість замовлень за період, суму оплат, активні замовлення, виконані замовлення та ресурси з критичним залишком. Такі дані можуть

використовуватися керівником малого підприємства для аналізу поточного стану бізнесу.

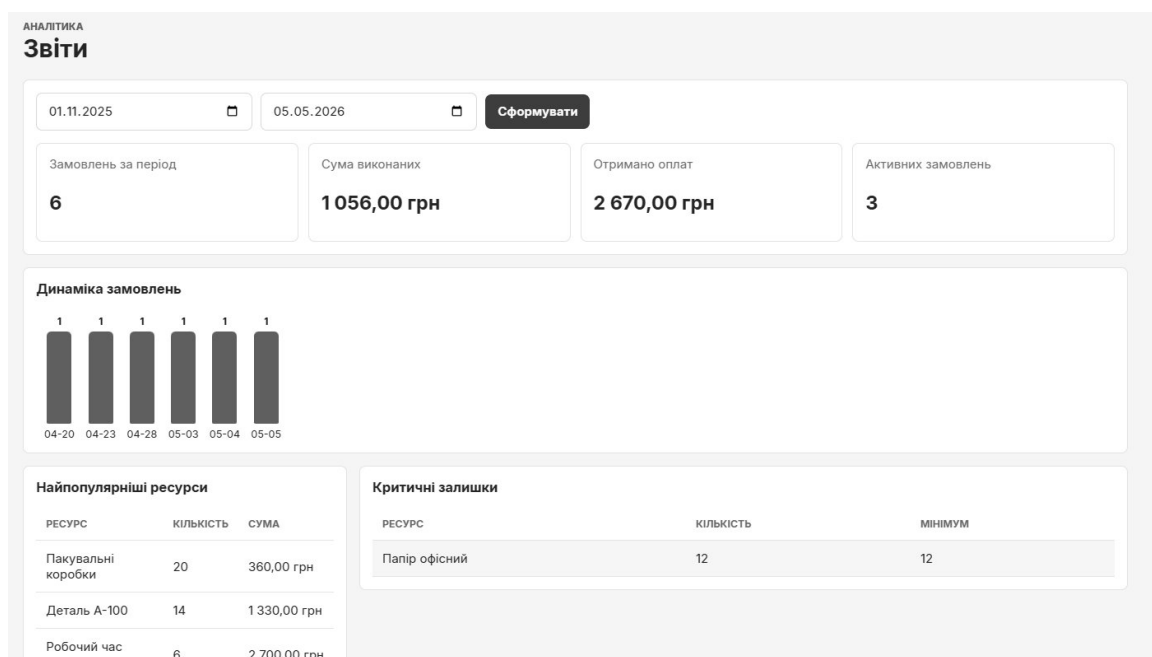


Рисунок 3.14 – Формування звіту за вибраний період

Результати тестування основних сценаріїв наведено в таблиці 3.5.

Таблиця 3.5 – Результати тестування інформаційної системи

№	Тестовий сценарій	Очікуваний результат	Фактичний результат	Статус
1	Авторизація користувача	Користувач входить у систему	Відкрито головну панель	Виконано
2	Введення неправильного пароля	Система показує помилку	Повідомлення про помилку відображено	Виконано
3	Додавання клієнта	Клієнт зберігається в базі даних	Клієнт з'явився у списку	Виконано
4	Створення замовлення	Замовлення створюється зі статусом «нове»	Замовлення додано до таблиці	Виконано
5	Автоматичний розрахунок суми	Система розраховує загальну суму	Сума розрахована правильно	Виконано
6	Підтвердження за достатніх ресурсів	Ресурси резервуються	Статус змінено на «підтвержене»	Виконано
7	Підтвердження за нестачі ресурсів	Система забороняє підтвердження	Виведено повідомлення про нестачу	Виконано
8	Перегляд ресурсів	Відображаються кількість, резерв і доступний залишок	Дані відображено коректно	Виконано
9	Критичний залишок ресурсу	Система позначає проблемний ресурс	Ресурс відображено як критичний	Виконано

Зм.	Арк.	№ докум.	Підпис	Дата
-----	------	----------	--------	------

КвРІСТ.220180.22.01.03 ПЗ

Арк.
71

Кінець таблиці 3.5

10	Завершення замовлення	Ресурси списуються	Замовлення виконано, ресурси оновлено	Виконано
11	Скасування замовлення	Резерв ресурсів знімається	Ресурси повернено в доступний залишок	Виконано
12	Додавання часткової оплати	Статус стає «частково оплачено»	Статус оновлено правильно	Виконано
13	Додавання повної оплати	Статус стає «оплачено»	Статус оновлено правильно	Виконано
14	Формування звіту	Система показує статистику за період	Звіт сформовано	Виконано
15	Фільтрація замовлень	Відображаються замовлення з вибраним статусом	Фільтр працює коректно	Виконано

За результатами тестування встановлено, що інформаційна система коректно виконує основні функції. Авторизація обмежує доступ до системи, модуль клієнтів дозволяє вести клієнтську базу, модуль замовлень забезпечує створення та зміну статусів замовлень, а модуль ресурсів правильно виконує перевірку доступності, резервування, списання та зняття резерву.

Особливо важливим результатом є правильна робота сценарію з нестачею ресурсів. Система не дозволяє підтвердити замовлення, якщо доступної кількості ресурсу недостатньо. Це відповідає основній ідеї роботи, оскільки інформаційна система повинна враховувати не тільки кількість замовлень, а й реальний стан ресурсів підприємства.

Проведене тестування підтвердило працездатність розробленої інформаційної системи. Результати перевірки показали, що система може використовуватися для автоматизації основних процесів малого бізнесу: роботи з клієнтами, оброблення замовлень, контролю ресурсів, обліку оплат і формування звітів.

3.7 Висновки до третього розділу

У третьому розділі було розглянуто програмну реалізацію та тестування інформаційної системи управління малим бізнесом з урахуванням динаміки замовлень та ресурсів. На початку розділу було обґрунтовано вибір програмних

засобів реалізації. Для створення клієнтської частини обрано React, для серверної частини - Node.js та Express.js, для зберігання даних - реляційну базу даних PostgreSQL. Такий набір технологій дозволяє реалізувати web-додаток із поділом на клієнтську частину, серверну логіку та рівень зберігання даних.

Було описано реалізацію серверної частини інформаційної системи. Серверна частина забезпечує оброблення запитів користувачів, авторизацію, роботу з клієнтами, замовленнями, ресурсами, платежами та звітами. Окрему увагу приділено логіці підтвердження, завершення та скасування замовлень, оскільки саме ці операції пов'язані з перевіркою доступності ресурсів, резервуванням, списанням і поверненням ресурсів у доступний залишок.

Також було розглянуто реалізацію бази даних. У системі передбачено таблиці користувачів, клієнтів, замовлень, позицій замовлень, ресурсів, оплат і руху ресурсів. Така структура дозволяє зберігати основні дані малого підприємства, підтримувати зв'язки між об'єктами та забезпечувати цілісність інформації. Особливе значення має таблиця руху ресурсів, оскільки вона фіксує операції резервування, списання та скасування резерву.

У межах розділу було описано клієнтську частину системи. Вона забезпечує користувачу доступ до основних функцій: авторизації, перегляду клієнтів, створення замовлень, контролю ресурсів, додавання оплат і формування звітів. Інтерфейс системи побудовано так, щоб користувач міг швидко виконувати типові операції малого бізнесу без зайвих дій.

Окремо було розглянуто реалізацію модулів управління замовленнями та ресурсами. Модуль замовлень дозволяє створювати, підтверджувати, завершувати та скасовувати замовлення. Модуль ресурсів забезпечує облік загальної, зарезервованої та доступної кількості ресурсів. Їхня взаємодія дає змогу враховувати фактичний стан ресурсів під час прийняття нового замовлення.

У підрозділі демонстрації роботи та тестування було перевірено основні сценарії використання системи. Зокрема, протестовано авторизацію

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 73
Зм.	Арк.	№ докум.	Підпис	Дата		

користувача, додавання клієнта, створення замовлення, перевірку ресурсів, підтвердження замовлення, завершення, скасування, додавання оплати та формування звітів. Результати тестування показали, що система коректно виконує основні функції та реагує на ситуації, коли ресурсів недостатньо для підтвердження замовлення.

					КвРІСТ.220180.22.01.03 ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		74

ВИСНОВКИ

У кваліфікаційній роботі було розроблено інформаційну систему управління малим бізнесом з урахуванням динаміки замовлень та ресурсів. Основною метою роботи було створення програмного засобу, який забезпечує автоматизацію процесів ведення клієнтської бази, оброблення замовлень, контролю ресурсів, обліку оплат і формування звітної інформації.

У першому розділі було проведено аналіз предметної області управління малим бізнесом. Встановлено, що діяльність малих підприємств характеризується змінним попитом, обмеженими ресурсами та необхідністю швидкого прийняття управлінських рішень. Визначено основні етапи роботи із замовленнями, які включають створення замовлення, перевірку наявності ресурсів, підтвердження, резервування, виконання, оплату та закриття. Результати аналізу показали, що більшість таких рішень не забезпечує комплексного управління всіма процесами малого підприємства, що підтвердило доцільність розроблення власної інформаційної системи.

У другому розділі виконано проектування інформаційної системи. Було сформовано функціональні та нефункціональні вимоги до програмного забезпечення, які визначають основні можливості системи та вимоги до її надійності, безпеки й продуктивності. Для реалізації обрано клієнт-серверну архітектуру, що передбачає розподіл системи на клієнтську частину, сервер бізнес-логіки та базу даних. Такий підхід забезпечує зручність супроводу, масштабованість та можливість подальшого розвитку програмного продукту.

Також було спроектовано структуру бази даних, яка включає таблиці користувачів, клієнтів, замовлень, позицій замовлень, ресурсів, оплат та руху ресурсів. Особливу увагу приділено механізму резервування ресурсів, який дозволяє уникнути ситуацій, коли одні й ті самі ресурси використовуються для кількох замовлень одночасно. Було розроблено алгоритми перевірки доступності

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 75
Зм.	Арк.	№ докум.	Підпис	Дата		

ресурсів, підтвердження замовлень, виконання операцій резервування та списання ресурсів після завершення замовлення.

У третьому розділі описано програмну реалізацію інформаційної системи. Для розроблення клієнтської частини використано React, серверної частини - Node.js та Express.js, а для зберігання даних - PostgreSQL. Реалізовано функціональні модулі авторизації користувачів, управління клієнтами, створення та супроводу замовлень, контролю ресурсів, обліку оплат і формування звітності. Серверна логіка забезпечує коректну обробку даних, перевірку доступності ресурсів, їх резервування та списання відповідно до стану замовлення.

Під час тестування було перевірено основні сценарії роботи системи: авторизацію користувачів, додавання клієнтів, створення та підтвердження замовлень, резервування ресурсів, проведення оплат, завершення та скасування замовлень. Результати тестування підтвердили коректність роботи програмного забезпечення та правильність реалізованих алгоритмів. Система успішно обробляє ситуації недостатності ресурсів і запобігає підтвердженню замовлень, які неможливо виконати.

Практична цінність розробленої інформаційної системи полягає у зменшенні кількості ручних операцій, скороченні часу оброблення замовлень, підвищенні точності обліку ресурсів і забезпеченні керівництва актуальною інформацією для прийняття управлінських рішень. Система може бути використана на малих підприємствах у сфері торгівлі, послуг або невеликого виробництва та слугувати основою для подальшого розвитку функціональності.

Отже, у процесі виконання кваліфікаційної роботи було досягнуто поставленої мети. Проведено аналіз предметної області, визначено вимоги до системи, спроектовано архітектуру та базу даних, реалізовано програмні модулі й виконано тестування. Розроблена інформаційна система забезпечує ефективне управління замовленнями та ресурсами малого бізнесу й сприяє підвищенню ефективності його діяльності.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 76
Зм.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. European Commission. Annual Report on European SMEs 2020/2021. Luxembourg : Publications Office of the European Union, 2021. URL: <https://data.europa.eu/doi/10.2826/515420> (дата звернення: 05.05.2026).

2. OECD. The Digital Transformation of SMEs. Paris : OECD Publishing, 2021. 274 p. URL: https://www.oecd.org/content/dam/oecd/en/publications/reports/2021/02/the-digital-transformation-of-smes_ec3163f5/bdb9256a-en.pdf (дата звернення: 05.05.2026).

3. OECD. SME Digitalisation to Build Back Better. OECD SME and Entrepreneurship Papers. Paris : OECD Publishing, 2021. URL: https://www.oecd.org/content/dam/oecd/en/publications/reports/2021/12/sme-digitalisation-to-build-back-better_c52afe21/50193089-en.pdf (дата звернення: 05.05.2026).

4. World Bank. Small and Medium Enterprises Finance. Washington : World Bank, 2024. URL: <https://www.worldbank.org/ext/en/topic/competitiveness/small-and-medium-enterprises-smes-finance> (дата звернення: 05.05.2026).

5. World Bank Group. Finance for Growth: Micro, Small, and Medium Enterprises. Washington : World Bank, 2019. URL: <https://openknowledge.worldbank.org/entities/publication/1bfaabd6-f323-524a-b60d-0490e5430917> (дата звернення: 05.05.2026).

6. World Bank. Typology of Small and Medium Enterprise Needs and Interventions. Washington : World Bank, 2019. URL: <https://openknowledge.worldbank.org/entities/publication/866ac6ba-2cb8-5cbd-a2c5-31ec3da370cf> (дата звернення: 05.05.2026).

7. Laudon K. C., Laudon J. P. Management Information Systems: Managing the Digital Firm. 16th ed. Harlow : Pearson, 2020. 648 p.

8. Stair R., Reynolds G. Principles of Information Systems. 13th ed. Boston : Cengage Learning, 2018. 752 p.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 77
Зм.	Арк.	№ докум.	Підпис	Дата		

9. Baltzan P. Business Driven Information Systems. 8th ed. New York : McGraw-Hill Education, 2021. 528 p.
10. Turban E., Volonino L., Wood G. Information Technology for Management: Digital Strategies for Insight, Action, and Sustainable Performance. 10th ed. Hoboken : Wiley, 2015. 480 p.
11. Rainer R. K., Prince B. Introduction to Information Systems. 7th ed. Hoboken : Wiley, 2017. 528 p.
12. Robertson S., Robertson J. Mastering the Requirements Process. 3rd ed. Boston: Addison-Wesley, 2015. 672p.
13. Sommerville I. Software Engineering. 10th ed. Harlow : Pearson, 2016. 816 p.
14. Pressman R. S., Maxim B. R. Software Engineering: A Practitioner's Approach. 9th ed. New York : McGraw-Hill Education, 2019. 705 p.
15. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Boston : Prentice Hall, 2017. 432 p.
16. Freeman S., Robson E. Growing Object-Oriented Software, Guided by Tests. 2nd ed. Boston : Addison-Wesley, 2020. 480 p.
17. Freeman E., Robson E., Bates B., Sierra K. Head First Design Patterns. 2nd ed. Boston : O'Reilly Media, 2020. 694 p.
18. OpenJS Foundation. Node.js Documentation. URL: <https://nodejs.org/docs/latest/api/> (дата звернення: 05.05.2026).
19. Express.js. Express Documentation. URL: <https://expressjs.com/> (дата звернення: 10.04.2026).
20. React. React Documentation. URL: <https://react.dev/> (дата звернення: 12.04.2026).
21. Tailwind Labs. Tailwind CSS Documentation. URL: <https://tailwindcss.com/docs> (дата звернення: 12.04.2026).
22. PostgreSQL Global Development Group. PostgreSQL 12 Documentation. URL: <https://www.postgresql.org/docs/12/> (дата звернення: 05.05.2026).

23. SQLite Consortium. SQLite Documentation. URL: <https://sqlite.org/docs.html> (дата звернення: 15.04.2026).
24. Prisma. Prisma ORM Documentation. URL: <https://www.prisma.io/docs> (дата звернення: 15.04.2026).
25. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. 2nd ed. Sebastopol : O'Reilly Media, 2020. 432 p.
26. Jones M., Bradley J., Sakimura N. JSON Web Token (JWT) : RFC 7519. Internet Engineering Task Force, 2015. URL: <https://datatracker.ietf.org/doc/html/rfc7519> (дата звернення: 05.05.2026).
27. Microsoft. Visual Studio Code Documentation. URL: <https://code.visualstudio.com/docs> (дата звернення: 05.05.2026).
28. Postman. Postman Documentation. URL: <https://learning.postman.com/docs/introduction/overview/> (дата звернення: 05.05.2026).
29. Chacon S., Straub B. Pro Git. 2nd ed. New York : Apress, 2014. URL: <https://git-scm.com/book/en/v2> (дата звернення: 20.04.2026).
30. Bray T. The JavaScript Object Notation (JSON) Data Interchange Format : RFC 8259. Internet Engineering Task Force, 2017. URL: <https://datatracker.ietf.org/doc/html/rfc8259> (дата звернення: 25.04.2026).
31. Richards M. Software Architecture Patterns. Sebastopol : O'Reilly Media, 2020. 192 p.
32. Elmasri R., Navathe S. B. Fundamentals of Database Systems. 7th ed. Boston : Pearson, 2016. 1280 p.
33. Silberschatz A., Korth H. F., Sudarshan S. Database System Concepts. 7th ed. New York : McGraw-Hill Education, 2019. 1376 p.
34. MDN Web Docs. HTTP. URL: <https://developer.mozilla.org/en-US/docs/Web/HTTP> (дата звернення: 28.04.2026).
35. Ammann P., Offutt J. Introduction to Software Testing. 3rd ed. Cambridge : Cambridge University Press, 2024. 446 p.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 79
Зм.	Арк.	№ докум.	Підпис	Дата		

36. Cormen T. H., Leiserson C. E., Rivest R. L., Stein C. Introduction to Algorithms. 4th ed. Cambridge : MIT Press, 2022. 1312 p.
37. Wengrow J. A Common-Sense Guide to Data Structures and Algorithms. 2nd ed. Raleigh : The Pragmatic Bookshelf, 2020. 506 p.
38. Roughgarden T. Algorithms Illuminated: The Basics. New York : Soundlikeyourself Publishing, 2017. 232 p.
39. Lewis W. E. Software Testing and Continuous Quality Improvement. 3rd ed. Boca Raton : CRC Press, 2016. 688 p.
40. IEEE Computer Society. SWEBOK: Guide to the Software Engineering Body of Knowledge. Version 3.0. Los Alamitos : IEEE Computer Society, 2014. URL: <https://www.computer.org/education/bodies-of-knowledge/software-engineering> (дата звернення: 01.05.2026).
41. Dick J., Hull E., Jackson K. Requirements Engineering. – 4th ed. Cham: Springer, 2017. – 616 p.
42. Laplante P. A. Requirements Engineering for Software and Systems. Boca Raton: CRC Press, 2017. – 400 p.
43. Tilley S., Rosenblatt H. Systems Analysis and Design. – 12th ed. Boston: Cengage Learning, 2019. 576p.
44. Beynon-Davies P. Business Information Systems. – 3rd ed. London: Palgrave Macmillan, 2020. 475p.
45. Wiegers K., Beatty J. Software Requirements. – 3rd ed. Redmond: Microsoft Press, 2017. 672p.

					КвРІСТ.220180.22.01.03 ПЗ	Арк. 80
Зм.	Арк.	№ докум.	Підпис	Дата		

ДОДАТОК А (обов'язковий)

Результат роботи системи

Інформаційна система управління малим бізнесом

Клієнти, замовлення, ресурси, оплати і звітність в одному локальному застосунку.

Email:

Пароль:

Увійти

Тестовий доступ: admin@business.local / admin123

Інформаційна система управління малим бізнесом

Клієнти, замовлення, ресурси, оплати і звітність в одному локальному застосунку.

Failed to fetch

Email:

Пароль:

Увійти

Тестовий доступ: admin@business.local / admin123

Звіт. Акт.	Від. доступ.	Підпис. Дата	Підпис.	Мака	Машин. Об.
Розроб.	Розробник	управління малим бізнесом з урахуванням динаміки замовлень та ресурсів			
Н. Іванко	Іванко		Архив. 1	Архив. 1	Архив. 3
Т. Козуб	Козуб				ХНУ ІСТ 22-1
С.В.В.	С.В.В.				

Операції

Замовлення

[+ Створити](#)

Усі створені	Статус	Сума	Дедлайн	Розд.	В
#0	Нові	510,00 грн	03.05.2026	Розд.	В
#1	Нові	366,00 грн	10.05.2026	Розд.	В
#2	Оплати	1184,00 грн	08.05.2026		
#3	Ураховані	4 050,00 грн	06.05.2026		
#4	Виконані	690,00 грн	07.05.2026		
#5	Скороплатити	1 660,00 грн	13.05.2026		

Нові замовлення

Клієнт: Дедлайн:

Нові замовлення

Ресурси замовлення

Пакувальні коробки - доступно 112 шт	124	18	2 232,00 грн
Надставляючі - доступно 112 шт			
Деталі А-100 - доступно 61 шт	1	95	95,00 грн

+ Додати ресурс Разом: 2 327,00 грн

Звернути замовлення

Головна панель системи

Усього замовлень за місяць

3

Активні замовлення

4

Виконані замовлення

1

Критичні ресурси

1

Дохід

2 490,00 грн

Динаміка замовлень за останні 30 днів

04-20 04-23 04-28 05-03 05-04 05-05

КВЕРСИТ: 220180.22.01.03

Зав. кафедри КПС
д-р. філософії Ользі ПАВЛОВІЙ

Антон РАБЧЕНЮК

ПІБ здобувача вищої освіти

ФІТ, 4 курсу, групи ІСТ-22-1

ЗАЯВА

З правилами чинного Положення про систему забезпечення академічної доброчесності у Хмельницькому національному університеті, згідно з яким виявлення академічного плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту і застосування заходів академічної відповідальності, ознайомлений (а). Про використання спеціалізованих програмних засобів (СПЗ) StrikePlagiarism та Anti-Plagiarism для перевірки кваліфікаційних робіт здобувачів вищої освіти на наявність академічного плагіату оповіщений (а). Надаю університету право на передачу моєї роботи для обробки та збереження в базах даних СПЗ і використання роботи для виявлення академічного плагіату в інших роботах, які перевіряються СПЗ.

Також надаю свою згоду на обробку й збереження університетом моєї роботи в Інституційному репозитарії Хмельницького національного університету.

Робота надається для перевірки в електронному варіанті. Електронна версія моєї роботи збігається (ідентична) з друкованою.

1 травня 2026 року



Протокол аналізу звіту подібності експертом

Заявляю, що я ознайомився (-лась) з Повним звітом подібності, який був згенерований Системою виявлення і запобігання плагіату щодо роботи:

Автор: Антон Рабченко

Співавтор:

Назва: 7Рабченко Антон Диплом_виправл копія плагіат

Експерт: Єлизавета Гнатчук

Підрозділ: Кафедра комп'ютерної інженерії та інформаційних систем

Коефіцієнт подібності 1: 8.81%

Коефіцієнт подібності 2: 3.9%

Мікропробіли: 3

Заміна букв: 0

Інтервали: 0

Білі знаки: 0

Дата створення звіту: 2026-06-05 07:34:51.0

Після аналізу Звіту подібності констатую наступне:

Запозичення, виявлені в роботі є законними і не є плагіатом. Рівень подібності не перевищує допустимої межі. Таким чином робота незалежна і приймається.

Запозичення не є плагіатом, але перевищено граничне значення рівня подібностей. Таким чином робота повертається на доопрацювання.

Виявлено запозичення і плагіат або навмисні текстові спотворення (маніпуляції), як передбачувані спроби укриття плагіату, які роблять роботу невідповідною вимогам законодавства (Ст. 32. ЗУ Про вищу освіту, пункт 3.1, Ст. 42. ЗУ Про освіту) та вимог НАЗЯВО (Критерій 5), а також кодексу етики і процедурам. Таким чином робота не приймається.

Обґрунтування:

2026-06-05

Дата



Доцент Андрій Нічепорук

експерт

Anti-Plagiarism (<http://ap.km.ua>) v-15.701

Максимальне співпадіння з одним документом 4.0%

Словники перевірки: en_US, ru_RU, ua_UA. **Помилки в документах: 5%**

ID: 273693 Назва: БКР Інформаційна система управління малим бізнесом з урахуванням динаміки замовлень та ресурсів Додано в БД: 2026-06-04 Автора: Антон Рабченко Керівники: Єлизавета Гнатчук Консультанти: Опоненти:	Документ		Сумарний збіг по Базі Даних	
	Символи	Лексеми	Символи	Лексеми
	118943	1141	10249 (9%)	110 (10%)

Джерело плагіату

ID	Опис	Наявність плагіату в документі	
		Символи	Лексеми

РІШЕННЯ ЕКСПЕРТНОЇ КОМІСІЇ

КАФЕДРИ КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ СИСТЕМ ПРО ДОПУСК КВАЛІФІКАЦІЙНОЇ РОБОТИ ДО ЗАХИСТУ

Назва кваліфікаційної роботи Інформаційна система управління малим бізнесом з урахуванням динаміки замовлень та ресурсів

Автор Антон РАБЧЕНЮК

Освітня програма Інформаційні системи та технології

Рівень вищої освіти перший (бакалаврський)

Спеціальність 126 Інформаційні системи та технології

Науковий керівник: Д.т.н., професор Єлизавета ГНАТЧУК

На основі аналізу кваліфікаційної роботи на дотримання вимог академічної доброчесності (у т.ч. відсутності ознак академічного плагіату) з урахуванням результатів перевірки роботи спеціалізованим програмним засобом(ами) комісія зробила такий висновок:

№	Висновок	Позначка про відповідність
1	Ознаки академічного плагіату	
1.1	Запозичення, виявлені в роботі, є законними і не є академічним плагіатом (далі – зазначаються підстави віднесення запозичень до правомірних, якщо потрібно). Робота приймається до захисту.	відповідає
1.2	Виявлені запозичення не є академічним плагіатом, розміщені в розділах, які не описують безпосередньо авторське дослідження, але кількість цитат перевищує обсяг, виправданий поставленою метою роботи; (далі – зазначаються детальні та аргументовані підстави віднесення запозичень до правомірних). Робота приймається до захисту, але має бути відкоригована.	
1.3	Виявлені запозичення не є академічним плагіатом, але частково розміщені в розділах, які описують безпосередньо авторське дослідження, а кількість цитат перевищує обсяг, виправданий поставленою метою роботи. Робота може бути допущена до захисту після того як буде відкоригована та доопрацьована і успішно пройде повторну перевірку на академічний плагіат.	
1.4	Робота містить навмисні текстові спотворення, передбачувані спроби укріття текстових запозичень або інші прояви академічного плагіату. Робота містить фабрикацію або фальсифікацію даних. Робота не допускається до захисту.	
2	Інші види порушень академічної доброчесності	

Підтвердження:

Запозичення, виявлені в роботі, є законними і не є плагіатом, оскільки:

- 1) усі запозичення фрагментарні, або мають належним чином оформленні посилання;
- 2) окремі виявлені збіги є загальноживаними фразами або виразами, про що свідчить посилання системи на збіг з джерелами на один фрагмент речення;
- 3) всі зафіксовані системою ознаки модифікації тексту відносяться до комбінування латинських символів зі україномовними скороченнями індексів в формулах, що не є модифікацією тексту.
- 4) значна частина знайденого плагіату відноситься до списку використаних джерел

Сумарний обсяг всіх запозичень, визначений системою виявлення збігів/ ідентичності/схожості StrikePlagiarism, складає 8.81%; та системою Anti-Plagiarism складає 4.0%, що, з урахуванням наведених обґрунтувань, відповідає характеру наукового дослідження і свідчить на користь кваліфікаційної роботи.

15.12.2025

Завідувач кафедри

Гарант освітньої програми

Керівник кваліфікаційної роботи


 Підпис
 Підпис
 Підпис

Ольга ПАВЛОВА
Ім'я, ПРІЗВИЩЕ

Єлизавета ГНАТЧУК
Ім'я, ПРІЗВИЩЕ

Єлизавета ГНАТЧУК
Ім'я, ПРІЗВИЩЕ

РЕЦЕНЗІЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Дипломник: Рабченюк Антон Ігорович

Тема: Інформаційна система управління малим бізнесом з урахуванням динаміки замовлень та ресурсів

Спеціальність: 126 «Інформаційні системи та технології»

Обсяг кваліфікаційної роботи:

Кількість листів креслень 3 Кількість сторінок записки 72

1. Короткий зміст роботи та прийнятих рішень: Метою роботи є розробка інформаційної системи управління малим бізнесом для автоматизації замовлень і ресурсів та підвищення ефективності управління

2. Висновок про відповідність роботи дипломному завданню: Робота повністю відповідає поставленому завданню.

3. Характеристика виконання кожного розділу, ступінь використання останніх досягнень науки і техніки і передових методів роботи: В першому розділі дослідження підтвердило, що специфіка малого бізнесу вимагає автоматизації через дефіцит ресурсів та нерівномірне навантаження. Оскільки готові ІТ-рішення не повністю покривають потреби підприємства, створення власної системи для супроводу замовлень та обліку активів є дуже важливим для ефективного управління. В другому розділі розроблено технічний проєкт системи: визначено вимоги, обрано клієнт-серверну архітектуру та виділено ключові взаємопов'язані модулі. Спроектовано структуру бази даних, яка охоплює весь цикл замовлень, та створено алгоритми для автоматичної перевірки й контролю залишків ресурсів. В третьому розділі виконано практичну реалізацію системи на базі сучасного стеку технологій. Описано створення серверної логіки, структури бази даних та клієнтського інтерфейсу. Особливу увагу надано модулям управління замовленнями та ресурсами, що забезпечують автоматичне резервування та списання активів. Проведене тестування підтвердило стабільність роботи системи у типових сценаріях та коректність обробки помилок при дефіциті ресурсів.

4. Позитивні сторони роботи: висока практична цінність роботи.

5. Негативні сторони роботи: недостатня увага аналізу предметної області; недостатньо чітко описано процес складання програмно-технічного засобу.

6. Оцінка графічного оформлення та пояснювальної записки роботи: Пояснювальна записка оформлена коректно, згідно діючих стандартів оформлення документації.

7. Відгук про роботу в цілому: Робота виконана на достатньому технічному рівні.

8. Інші зауваження: _____

9. Оцінка дипломної роботи: задовільно (D / 70)

Рецензент (прізвище, ім'я, по батькові, посада, місце роботи) _____

Гетлик Наталія Сергіївна, доцент кафедри КБ

"11" червня 2026 р.

 (підпис)