

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»
Факультет автоматизації виробництва та цифрових технологій
Кафедра цифрових технологій та проектно-аналітичних рішень

«Допущено до захисту»
Гарант ОПП

Олександр КОСТИКОВ

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

за підсумками виконання
освітньо-професійної програми
«Комп'ютерні науки»
за спеціальністю 122 Комп'ютерні науки

**на тему «Програмний комплекс для інформаційної підтримки
діяльності продавця-консультанта»**

Керівник роботи

Павло САГАЙДА

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають
посилання на відповідне джерело*

Здобувач

Олег НЕФЕДЧЕНКО

Підсумкова оцінка за атестацію			
--------------------------------	--	--	--

Голова ЕК

Олена ПАВЛЕНКО

ЗАПОРІЖЖЯ 2025

ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА»	
Факультет	автоматизації виробництва та цифрових технологій
Кафедра	цифрових технологій та проектно-аналітичних рішень
Ступінь вищої освіти	бакалавр
Спеціальність	122 Комп'ютерні науки
ОПП	Комп'ютерні науки

ЗАТВЕРДЖУЮ
Гарант ОПП

Олександр КОСТИКОВ

«05» червня 2025 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Нефедченко Олегу Олександровичу
(прізвище, ім'я, по батькові здобувача)

1. Тема роботи «Програмний комплекс для інформаційної підтримки діяльності продавця-консультанта»
керівник роботи Сагайда Павло Іванович, доцент, доктор техн. наук,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом Університету від 31.03.2025 р. №81/31.03.2025
2. Термін подання роботи 01.07.2025 р.
3. Вихідні дані до роботи Навчальна література, державні стандарти та методичні рекомендації, наукові та науково-технічні публікації з тематики інформаційної підтримки процесів торгівлі електронною технікою, дослідження в галузі побудови інформаційних систем, результати власного проектування, тестування, експериментальних розрахунків та економічного обґрунтування впровадження програмного комплексу.
4. Зміст пояснювальної записки (перелік питань) Реферат. Зміст. Вступ. Розділ 1. Аналіз стану автоматизації обробки даних про роботу продавців-консультантів в магазинах побутової техніки. Розділ 2. Математичне моделювання процесу обслуговування покупців в магазинах побутової техніки. Розділ 3. Проектування та реалізація програмного комплексу для інформаційної підтримки роботи продавців-консультантів в магазинах побутової техніки. Розділ 4. Аналіз економічної доцільності розробки програмного комплексу для інформаційної підтримки діяльності продавців-консультантів в магазинах побутової техніки. Висновки. Перелік використаних джерел. Додатки.
5. Перелік графічного (демонстраційного) матеріалу (з точним зазначенням обов'язкових креслень): Актуальність, мета, об'єкт, предмет та завдання дослідження; результати аналізу предметної області диджиталізації та їх формалізації за допомогою діаграмних методик; структура бази даних; результати логічного проектування у вигляді діаграм UML (прецедентів, класів, послідовностей, або інших, за необхідності); форми користувацького інтерфейсу; приклади реалізованих запитів до бази даних та звітів; результати розрахунків витрат та терміну окупності системи; висновки до роботи; (за наявності) – копії публікацій або розробленої документації.

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх.

Розділ	Прізвище, ініціали та посада консультанта
1	Сагайда П.І., проф. каф. ЦТПАР
2	Сагайда П.І., проф. каф. ЦТПАР
3	Сагайда П.І., проф. каф. ЦТПАР
4	Сагайда П.І., проф. каф. ЦТПАР

7. Дата видачі завдання 05.07.2025

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів роботи	Термін виконання етапів роботи
1	Аналіз стану автоматизації обробки даних про роботу продавців-консультантів в магазинах побутової техніки	05.06.2025 – 10.06.2025
2	Математичне моделювання процесу обслуговування покупців в магазинах побутової техніки	11.06.2025 – 15.06.2025
3	Проектування та реалізація програмного комплексу для інформаційної підтримки роботи продавців-консультантів в магазинах побутової техніки	16.06.2025 – 22.06.2025
5	Аналіз економічної доцільності розробки програмного комплексу для інформаційної підтримки роботи продавців-консультантів в магазинах побутової техніки	23.06.2025 – 25.06.2025
6	Висновки, перелік посилань, вступ, зміст, реферат	26.06.2025 – 27.06.2025
7	Остаточне оформлення та подання завершеної роботи.	28.06.2025 – 30.06.2025
8	Підготовка презентаційного матеріалу, рецензування завершеної роботи. Захист	01.07.2025 – 04.07.2025

Здобувач

(Олег НЕФЕДЧЕНКО)

Керівник роботи

(Павло САГАЙДА)

РЕФЕРАТ

Нефедченко О.О. Програмний комплекс для інформаційної підтримки діяльності продавця-консультанта.

Кваліфікаційна робота на здобуття освітнього ступеня бакалавра комп'ютерних наук за спеціальністю 122 «Комп'ютерні науки». – ТОВ «ТЕХНІЧНИЙ УНІВЕРСИТЕТ «МЕТІНВЕСТ ПОЛІТЕХНІКА» МОН України, м. Запоріжжя, 2025.

Метою роботи було створення програмного забезпечення, яке дозволить автоматизувати основні процеси роботи продавця-консультанта в магазині побутової техніки. Система охоплює облік продукції, управління історією продажів, управління клієнтською базою, генерацію чеків та звітність.

Об'єктом дослідження є робота продавців-консультантів у магазині роздрібної торгівлі побутовою технікою.

Предметом дослідження є процес автоматизації функціональних обов'язків продавця-консультанта за допомогою спеціалізованого програмного забезпечення.

Методологія – структурний аналіз, SADT, моделювання інформаційних потоків, об'єктно-орієнтоване програмування, проектування баз даних, економічна оцінка розробки програмного забезпечення.

У цій роботі проаналізовано функціональні вимоги роздрібного магазину, розроблено технічні умови та побудовано логічну модель інформаційної системи. Програмний продукт реалізовано на мові програмування C# у середовищі Visual Studio та використовує базу даних SQLite. Реалізовано зручний інтерфейс користувача, обліку товарів, продажів, клієнтів, а також механізми формування звітів. Було проведено техніко-економічне обґрунтування впровадження розробленої системи.

Розроблене програмне забезпечення може підвищити продуктивність роботи продавців-консультантів, зменшити кількість помилок у процесі продажу, забезпечити швидкий доступ до даних, покращити обслуговування клієнтів та оптимізувати внутрішні процеси магазину. Систему можна адаптувати для інших роздрібних підприємств з подібною структурою.

КЛЮЧОВІ СЛОВА: АВТОМАТИЗАЦІЯ, ПРОДАВЕЦЬ-КОНСУЛЬТАНТ, ОБЛІК ТОВАРІВ, ПРОДАЖІ, КЛІЄНТСЬКА БАЗА, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, VISUAL STUDIO, C#, SQLITE, ІНФОРМАЦІЙНІ СИСТЕМИ, ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ

ABSTRACT

Nefedchenko O.O. Software complex for information support of the activities of a sales consultant.

Qualification work for obtaining the degree of Bachelor of Computer Science in the specialty 122 Computer Science. – LLC «TECHNICAL UNIVERSITY «METINVEST POLYTECHNIC» MES of Ukraine, Kryvyi Rih, 2025.

The purpose of the work was to create software that will automate the main processes of the work of a sales consultant in a home appliance store. The system covers product accounting, sales history management, customer base management, receipt generation and reporting.

The object of the study is the work of sales consultants in a home appliance retail store.

The subject of the study is the process of automating the functional responsibilities of a sales consultant using specialized software.

Methodology - structural analysis, SADT method, information flow modeling, object-oriented programming, database design, economic assessment of development.

In this work, the functional requirements of a retail store are analyzed, technical conditions are developed and a logical model of the information system is built. The software product is implemented in the C# programming language in the Visual Studio environment and uses the SQLite database. A user-friendly interface, accounting of goods, sales, customers, as well as reporting mechanisms have been implemented. A feasibility study for the implementation of the developed system has been conducted.

The developed software can increase the productivity of sales consultants, reduce the number of errors in the sales process, provide quick access to data, improve customer service and optimize internal store processes. The system can be adapted for other retail enterprises with a similar structure.

KEYWORDS: AUTOMATION, SALES ASSISTANT, INVENTORY MANAGEMENT, SALES TRACKING, CUSTOMER DATABASE, SOFTWARE, VISUAL STUDIO, C#, SQLITE, INFORMATION SYSTEM, ECONOMIC FEASIBILITY

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ БІЗНЕС-ПРОЦЕСУ РОБОТИ ПРОДАВЦІВ-КОНСУЛЬТАНТІВ В МАГАЗИНАХ З ПРОДАЖУ ПОБУТОВОЇ ТЕХНІКИ	10
1.1 Аналіз предметної області та обґрунтування актуальності дослідження в сфері продажу побутової техніки	10
1.2 Аналіз сучасних аналогів програмного забезпечення та методів інформаційної підтримки роботи продавців	15
1.2.1 Детальний огляд аналогів програмного забезпечення та виявлення їх переваг та недоліків	15
1.2.2 Методи оптимізації програмного забезпечення	19
РОЗДІЛ 2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ОБСЛУГОВУВАННЯ ПОКУПЦІВ В МАГАЗИНАХ ПОБУТОВОЇ ТЕХНІКИ	23
2.1 Моделювання поведінки покупців у магазині електронної техніки	28
2.2 Розробка математичної моделі системи масового обслуговування клієнтів в магазинах побутової техніки.	29
2.3 Сучасні технології збору даних для аналізу звичок клієнтів онлайн	32
2.3 Результати розробки логічних моделей застосунку та їх відображення у вигляді діаграм	33
2.4 Технічні вимоги до програмного забезпечення, що буде розроблено для подальшої оптимізації роботи продавців консультантів	38
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ РОБОТИ ПРОДАВЦІВ-КОНСУЛЬТАНТІВ В МАГАЗИНАХ ПОБУТОВОЇ ТЕХНІКИ	40
3.1 Вибір та обґрунтування засобів розробки	40
3.2 Представлення та опис коду розробленого програмного забезпечення	54
3.3 Графічний інтерфейс застосунку	70

РОЗДІЛ 4. ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ РОЗРОБКИ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ РОБОТИ ПРОДАВЦІВ-КОНСУЛЬТАНТІВ В МАГАЗИНАХ ПОБУТОВОЇ ТЕХНІКИ	77
4.1 Вартісний аналіз.....	77
4.1.1 Розрахунок повної собівартості програмного продукту	77
4.1.2 Калькуляція собівартості програмного продукту	78
4.2 Визначення ціни програмного продукту на основі вартості його розробки	87
4.3 Оцінка конкурентоспроможності програмного продукту	88
4.4 Моделювання конкурентоспроможності розробленого ПП (ІС, КМ) та його аналогу.....	89
ВИСНОВКИ	91
ПЕРЕЛІК ПОСИЛАНЬ	93
ДОДАТОК А.....	98
ДОДАТОК Б.....	99
ДОДАТОК В.....	100
ДОДАТОК Г	106

ВСТУП

На сучасному етапі розвитку суспільства значного поширення набули інформаційні технології, які охоплюють майже всі сфери людської діяльності – від освіти та науки до промисловості, торгівлі й обслуговування. Особливу роль цифрові технології відіграють у сфері роздрібної торгівлі, де з кожним роком зростає потреба у підвищенні ефективності обслуговування клієнтів, оперативному аналізі продажів, контролі товарних залишків та автоматизації щоденних бізнес-процесів.

Одним із ключових учасників торговельного процесу є продавець-консультант – фахівець, який не лише здійснює безпосередній продаж товарів, але й виконує низку супутніх завдань: облік товарів, взаємодія з клієнтами, оформлення замовлень, контроль за залишками, формування звітності. У більшості невеликих торгових точок ці завдання все ще виконуються вручну або за допомогою застарілих засобів, таких як блокноти чи таблиці Excel. Це призводить до зниження продуктивності праці, зростання кількості помилок, складнощів в аналізі даних та втрати актуальної інформації.

В умовах сучасного конкурентного ринку ефективність роботи продавця-консультанта має безпосередній вплив на фінансові показники торгового закладу. Саме тому актуальним є питання розробки спеціалізованого програмного забезпечення, яке б дозволило автоматизувати основні операції, з якими щодня працює продавець. Йдеться про створення простої, функціональної, стабільної програми, яка забезпечить облік товарів, постачальників, продажів, бронювань, формування звітів та інші ключові процеси – без потреби в підключенні до Інтернету, сторонніх сервісах чи складному адмініструванні.

Метою даної роботи є розробка програмного забезпечення, яке дозволить оптимізувати роботу продавця-консультанта, зменшити

кількість рутинних операцій, знизити ризики помилок та підвищити загальну ефективність управління торговою точкою.

У межах проєкту передбачено:

- аналіз предметної області та вимог до системи;
- дослідження аналогічних програмних рішень;
- проєктування архітектури застосунку та структури бази даних;
- реалізація функціональних модулів мовою програмування C# із використанням бази даних SQLite;
- тестування працездатності системи;
- оцінка можливостей подальшого розвитку програмного продукту.

Таким чином, запропоноване програмне забезпечення має стати ефективним інструментом у щоденній роботі продавця-консультанта, а також прикладом практичного застосування знань і навичок у сфері розробки інформаційних систем.

РОЗДІЛ 1. АНАЛІЗ БІЗНЕС-ПРОЦЕСУ РОБОТИ ПРОДАВЦІВ-КОНСУЛЬТАНТІВ В МАГАЗИНАХ З ПРОДАЖУ ПОБУТОВОЇ ТЕХНІКИ

1.1 Аналіз предметної області та обґрунтування актуальності дослідження в сфері продажу побутової техніки

Сучасний ринок роздрібної торгівлі перебуває у стані постійної еволюції, що зумовлено зростанням конкуренції, змінами у поведінці споживачів та стрімким розвитком технологій. Клієнти сьогодні очікують не лише якісного товару, а й високого рівня обслуговування, яке включає швидке оформлення покупок, прозорість у системі повернення товарів, можливість бронювання та індивідуальний підхід. У таких умовах продавець-консультант перестає бути просто посередником між товаром і покупцем – він стає ключовою ланкою комунікації між бізнесом та клієнтом.

Однак ефективне виконання цих функцій ускладнюється через відсутність відповідного програмного забезпечення. На практиці багато малих і середніх магазинів продовжують використовувати застарілі методи обліку, такі як паперові журнали, електронні таблиці (наприклад, Excel) або локальні системи, які не забезпечують належного рівня автоматизації. Це призводить до низки проблем:

- Помилки у веденні обліку: ручне введення даних часто супроводжується помилками, особливо при великих обсягах інформації.
- Втрата часу: виконання рутинних операцій, таких як перевірка залишків або формування звітів, забирає значну кількість часу.
- Дублювання інформації: дані часто вносяться в кілька джерел одночасно, що збільшує ризик неузгодженості.

- Зниження якості обслуговування: клієнти можуть отримувати недостовірну інформацію про наявність товару або його характеристики.

Ці фактори безпосередньо впливають на прибуток підприємства. Наприклад, помилки в обліку можуть призвести до надлишкових закупівель або, навпаки, до дефіциту товарів, що вплине на продажі. Тому автоматизація процесів, пов'язаних із роботою продавця-консультанта, є критично необхідною.

Розробка спеціалізованого програмного забезпечення, яке автоматизує облік товарів, контроль залишків, обробку замовлень, продажі та бронювання, дозволить значно підвищити ефективність роботи магазину. Такий застосунок має бути інтуїтивно зрозумілим, щоб його могли використовувати працівники без спеціальної ІТ-підготовки, та автономним, щоб забезпечити стабільну роботу навіть за відсутності інтернет-з'єднання.

Мета та завдання роботи

Метою роботи є створення програмного забезпечення для автоматизації процесів, які виконує продавець-консультант у роздрібній торгівлі. Основним фокусом є оптимізація таких функцій:

- Облік товарів: ведення бази даних товарів, їх характеристик, цін та залишків.
- Контроль залишків: автоматичне оновлення даних після продажів або надходжень.
- Обробка замовлень: формування заявок на постачання, відстеження їх статусу.
- Продажі: оформлення чеків, розрахунок сум, інтеграція з фіскальним обладнанням.
- Бронювання: резервування товарів для клієнтів на певний термін.
- Звітність: генерація звітів з продажів, залишків, руху товарів.

Для досягнення мети необхідно вирішити наступні завдання:

1. Дослідити специфіку діяльності продавця-консультанта:
 - Аналіз типових операцій, які виконує продавець (консультація клієнтів, оформлення продажів, ведення обліку).
 - Визначення проблем, пов'язаних із ручним веденням обліку.
2. Проаналізувати типові проблеми ручного обліку:
 - Неможливість контролю залишків у реальному часі.
 - Високий ризик помилок при розрахунках.
 - Складність формування документів та звітів.
3. Розглянути існуючі програмні рішення:
 - Огляд таких систем, як Poster POS, BAS Облік торгівлі, RemOnline.
 - Визначення їх переваг та недоліків.
4. Сформулювати вимоги до програмного продукту:
 - Вимоги до функціоналу (наприклад, підтримка офлайн-режиму, простота інтерфейсу).
 - Вимоги до технічної реалізації (вибір мови програмування, бази даних).
5. Розробити структуру бази даних та архітектуру ПЗ:
 - Визначення таблиць бази даних (товари, постачальники, замовлення тощо).
 - Проектування взаємозв'язків між модулями програми.
6. Реалізувати інтерфейс та функціонал:
 - Створення зручного графічного інтерфейсу для користувачів.
 - Налаштування основних функцій (додавання товарів, оформлення продажів).
7. Провести тестування:
 - Перевірка стабільності роботи програми.
 - Виявлення та усунення помилок.

8. Підготувати документацію:

- Інструкція для користувачів.
- Технічний опис архітектури програми.

Об'єктом автоматизації є інформаційна підтримка діяльності продавця-консультанта в умовах роздрібної торгівлі. Ця діяльність включає такі основні процеси:

1. Облік товарів у залі:

- Фіксація надходження нових товарів. При отриманні партії товару необхідно фіксувати дані про його кількість, постачальника, дату постачання та інші супровідні реквізити. Це дозволяє своєчасно оновлювати складські залишки та забезпечувати точність даних.

- Оновлення інформації про залишки. Своєчасне внесення змін до облікової системи після реалізації товарів або їх списання є необхідною умовою підтримання актуальної інформації щодо наявності продукції в торговому залі.

- Ведення каталогу товарів (назва, артикул, ціна, категорія). Каталог має містити основні характеристики кожної одиниці товару, зокрема: назву, артикул, штрихкод, ціну, категорію, одиницю виміру та інші важливі параметри. Наявність упорядкованого каталогу спрощує процес пошуку та обліку продукції.

2. Оформлення замовлень і продажів:

- Створення заявок на постачання товарів. У разі нестачі товарів працівник формує заявку, яка містить інформацію про необхідну кількість, найменування, постачальника та бажану дату поставки. Це сприяє підтриманню стабільного товарного запасу.

- Оформлення чеків для клієнтів. Кожна операція з продажу супроводжується формуванням фіскального або нефіскального чеку. В чек вносяться назви товарів, ціни, кількість та сума до сплати. Наявність

системи автоматичного формування чеків пришвидшує обслуговування клієнтів.

- Відстеження історії продажів. Збереження інформації про всі реалізовані товари дозволяє аналізувати попит, виявляти популярні позиції, а також розглядати динаміку продажів за певний період.

3. Консультація клієнтів:

- Надання інформації про товари. Продавець-консультант повинен мати швидкий доступ до даних про технічні характеристики, наявність, ціну та акційні пропозиції, щоб оперативно відповідати на запити клієнтів.

- Допомога у виборі. На основі потреб клієнта продавець пропонує відповідні товари, порівнює характеристики та радить найкращий варіант, що підвищує ймовірність покупки.

4. Формування звітності:

- Генерація звітів з продажів. Система повинна надавати можливість створення щоденних, щотижневих або щомісячних звітів, які відображають обсяги реалізації, виторг, кількість транзакцій тощо.

- Аналіз руху товарів. Аналітична інформація щодо залишків, списань, повернень та реалізації дозволяє оптимізувати складські запаси та своєчасно реагувати на зміни в попиті.

Проблеми ручного ведення обліку:

1. Відсутність реального часу. Неможливість точно визначити наявність товару без фізичної перевірки. Дані про наявність товару оновлюються із затримкою, що ускладнює обслуговування клієнтів та призводить до помилок при замовленні або продажу.

2. Помилки при розрахунках. Ризик неправильного підрахунку сум через людський фактор. Ручне обчислення сум, знижок або податків часто супроводжується арифметичними помилками, що може призвести до фінансових втрат або конфліктів із клієнтами.

3. Складність документообігу. Ручне заповнення форм забирає багато часу. Заповнення паперових форм або неструктурованих електронних документів займає багато часу, підвищує навантаження на персонал і ускладнює пошук необхідної інформації.

4. Відсутність історії операцій. Немає можливості проаналізувати минулі продажі або замовлення. При відсутності централізованого обліку неможливо швидко відстежити минулі транзакції, проаналізувати статистику продажів або сформувати звіт за період.

5. Низька ефективність. Витрати часу на рутинні операції зменшують продуктивність. Значна частина робочого часу витрачається на виконання рутинних операцій, що знижує загальну продуктивність персоналу та ефективність управління.

Автоматизація цих процесів дозволить усунути ці проблеми та створить прозору систему управління товарообігом.

1.2 Аналіз сучасних аналогів програмного забезпечення та методів інформаційної підтримки роботи продавців

1.2.1 Детальний огляд аналогів програмного забезпечення та виявлення їх переваг та недоліків

Для порівняння було обрано три популярні системи: Poster POS, BAS Облік торгівлі та RemOnline.

Аналіз функціональних можливостей Poster POS

На рисунку 1 наведено діалогове вікно системи Poster POS.

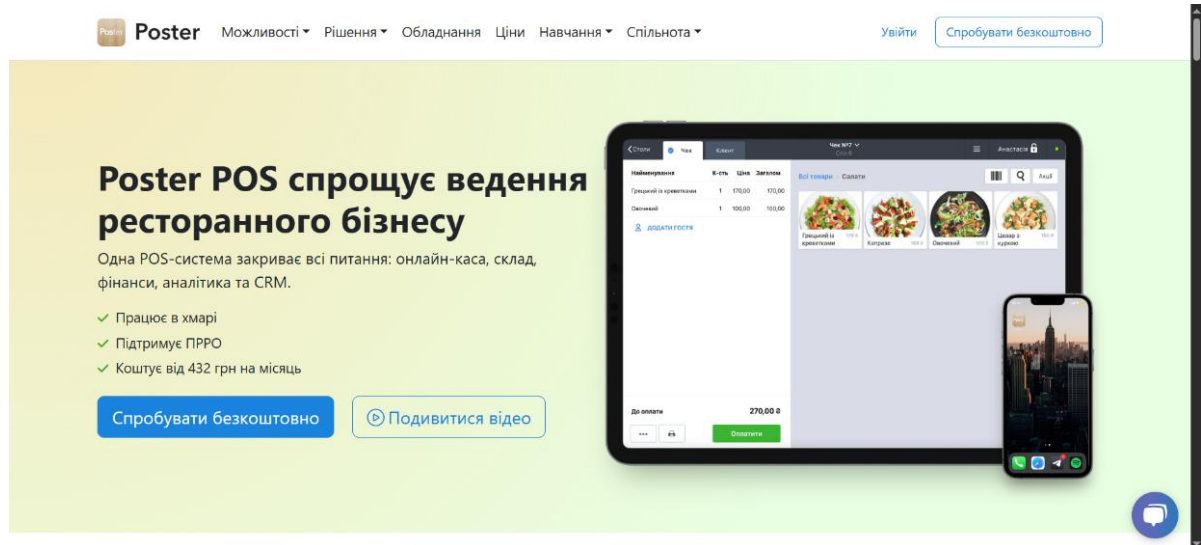


Рисунок 1 – Діалогове вікно системи Poster Pos

Poster POS – це хмарна POS-система для автоматизації кафе, ресторанів, барів, кав'ярень і магазинів [7]. Вона дозволяє керувати продажами, складом, персоналом і аналітикою з будь-якого пристрою – планшета, смартфона чи комп'ютера.

Функції:

1. Облік товарів.
2. Оформлення чеків.
3. Інтеграція з фіскальними реєстраторами.
4. Звітність.

Переваги та недоліки системи Poster Pos наведено у таблиці 1.

Таблиця 1 – Переваги та недоліки системи Poster Pos

Переваги	Недоліки
Простий інтерфейс	Потрібен постійний інтернет
Підтримка мобільних пристроїв	Абонентська плата

Аналіз функціональних можливостей BAS Облік торгівлі

На рисунку 2 наведено діалогове вікно системи BAS Облік торгівлі

Рисунок 2 – Діалогове вікно системи BAS Облік торгівлі

BAS Облік торгівлі – це програмне забезпечення, розроблене на платформі BAS (Business Automation Software) для автоматизації обліку та управління торговельною діяльністю підприємств [8]. Програма підходить для малого та середнього бізнесу, який займається оптовою, роздрібною чи змішаною торгівлею.

Функції:

1. Складський облік.
2. Управління логістикою.

Переваги та недоліки системи BAS Облік торгівлі наведено у таблиці 2.

Таблиця 2 – Переваги та недоліки системи BAS Облік торгівлі

Переваги	Недоліки
Гнучкі налаштування	Висока вартість

RemOnline

На рисунку 3 наведено діалогове вікно системи RemOnline.

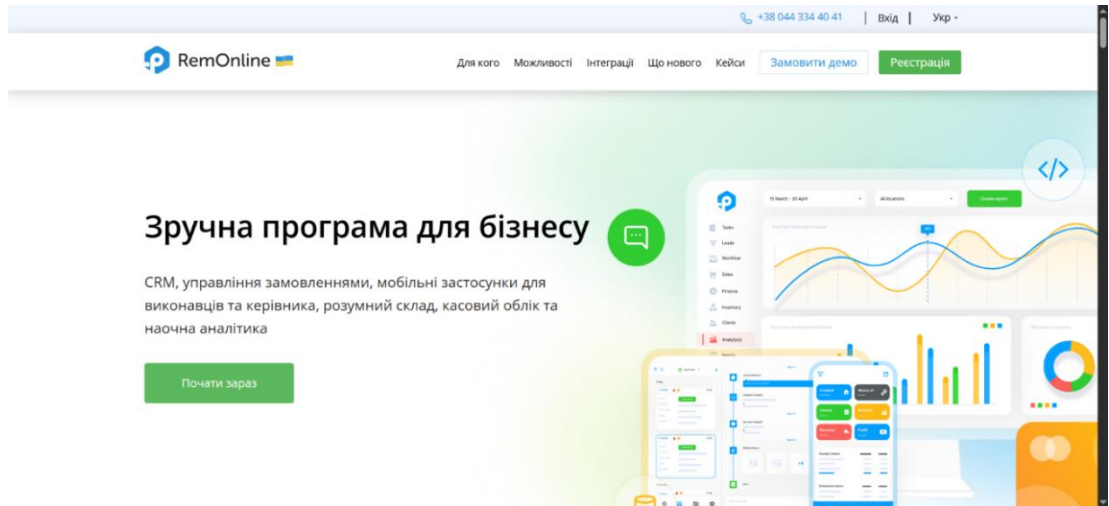


Рисунок 3 – Діалогове вікно системи RemOnline

RemOnline – це хмарна система обліку для малого та середнього бізнесу, зокрема для сервісних центрів, роздрібної торгівлі, майстерень, салонів послуг тощо [9]. Вона автоматизує щоденні бізнес-процеси, підвищує контроль і допомагає ефективніше керувати персоналом, складом, фінансами та взаємодією з клієнтами.

Функції:

1. Контроль закупівель.
2. Управління складом.

Переваги та недоліки системи RemOnline наведено у таблиці 3.

Таблиця 3 – Переваги та недоліки системи RemOnline

Переваги	Недоліки
----------	----------

Простота використання	Залежність від інтернету
-----------------------	--------------------------

Висновок. Проведений аналіз існуючих програмних рішень, зокрема RemOnline, BAS Облік торгівлі та Poster POS, дозволяє дійти висновку, що жодне з них не забезпечує повного задоволення потреб малих торговельних точок. Кожна із зазначених систем має власну специфіку, спрямовану на обслуговування різних сфер бізнесу – сервісних центрів, закладів громадського харчування або підприємств з розгалуженою структурою, що не завжди відповідає умовам і особливостям роботи малого магазину.

Окремі продукти характеризуються надмірною функціональністю, яка ускладнює процес освоєння системи та не використовується у повному обсязі, інші – потребують значних фінансових витрат на обслуговування або передбачають постійне підключення до мережі Інтернет, що не завжди є прийнятним для малого бізнесу. Також виявлено обмеження щодо адаптації інтерфейсу та бізнес-логіки під конкретні потреби окремого підприємства.

Ураховуючи вищезазначене, можна стверджувати про доцільність та обґрунтованість розробки власного програмного забезпечення, яке максимально відповідатиме індивідуальним вимогам малого магазину, забезпечуватиме ефективне управління торговельною діяльністю, спрощений облік товарів і фінансів, а також сприятиме оптимізації щоденних операцій без залучення надлишкових ресурсів.

1.2.2 Методи оптимізації програмного забезпечення

Для оптимізації програмного забезпечення, яке використовують консультанти з продажу, важливо зосередитися на підвищенні

продуктивності, зручності використання та ефективності робочого процесу.

Забезпечення високої продуктивності є ключовим для підтримки консультантів у динамічному середовищі. Для цього ми використали підхід «вимірювання-покращення-тестування»: ми оцінили вузькі місця (блокування бази даних, запити, мережа), оптимізували (використання кешів, індексів, лінивого завантаження), а потім знову протестували результати [14]. Функції автоматичного масштабування та балансування навантаження дозволяють підтримувати стабільний рівень продуктивності навіть у години пік [15].

Оптимізація користувацького інтерфейсу (UX) може значно пришвидшити ваш робочий процес. Спрощені форми, зрозумілі процеси навігації, швидке реагування на сторінки та негайний зворотний зв'язок під час операцій можуть підвищити задоволеність консультантів [16], [18]. Дослідження показали, що включення таких функцій, як скасування (операції переривання), індикатори прогресу та налаштування для користувача, у програми може значно підвищити ефективність та зручність використання [17].

Аналіз та підтримка рішень є важливими компонентами консалтингових послуг. Система повинна аналізувати купівельну поведінку клієнтів, генерувати рекомендації та вказувати популярні товари й оптимальну упаковку, тобто застосовувати керований підхід до продажу. Це допомагає консультантам коригувати котирування та уникати непотрібних проблем і затримок.

Автоматизація рутинних операцій (введення даних, обробка транзакцій, імпорт/експорт даних CRM) знімає навантаження на співробітників та зменшує кількість помилок у процесі ручного введення. Впровадження управління взаємовідносинами з клієнтами (використання історії покупок для надання персоналізованого обслуговування кожному клієнту) може збільшити середній дохід.

Оптимізація для мобільних пристроїв – якщо співробітники користуються планшетами або телефонами, інтерфейс має бути адаптований та масштабований для підтримки швидкості та зручності – значно покращить швидкість обслуговування та рівень продажів [19, 15].

Інтеграція з CRM та аналітичними платформами дозволяє консультантам отримувати контекстні дані про своїх клієнтів у реальних транзакціях: історію, сегментацію, моделі поведінки – щоб консультанти могли «фіксувати» відповідну інформацію під час взаємодії з клієнтами [20, 21].

Оптимізація бекенд-технологій – індексування баз даних, пакетна обробка транзакцій (масові вставки), кешування, реплікація – усі ці технічні рішення можуть зменшити затримку та забезпечити масштабованість і безперервність сервісу [22, 14].

Ітеративне тестування з використанням методів юзабіліті (коридорне тестування, A/B-тестування) та використання таких фреймворків, як OptiFlow, дозволило нам виявити найскладніші частини взаємодії консультантів з програмою та своєчасно їх покращити [18, 19, 23].

Систематичне вимірювання ключових показників ефективності (KPI): кількості оброблених клієнтів, середнього розміру рахунку, швидкості обслуговування – дозволяє нам відстежувати ефективність оптимізації та коригувати стратегії розвитку [14, 16, 17].

Використання машинного навчання для налаштування інтерфейсів та прогнозування потреб (рекомендації продуктів, рішення-бестселери) дозволяє нам значно покращити коефіцієнти конверсії та продуктивність [17, 24].

Загалом, поєднання технічної продуктивності, зручного інтерфейсу, автоматизації рутинних процесів та аналітичних можливостей дозволяє нам створювати ефективне програмне забезпечення, яке значно спрощує роботу продавців-консультантів та

підвищує швидкість обслуговування та рівень продажів. Це резонує із сучасними підходами до оптимізації роздрібної торгівлі [15, 16].

РОЗДІЛ 2. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСУ ОБСЛУГОВУВАННЯ ПОКУПЦІВ В МАГАЗИНАХ ПОБУТОВОЇ ТЕХНІКИ

Моделювання обслуговування покупців у магазині електронної техніки може допомогти значно покращити ефективність роботи магазину та підвищити задоволеність клієнтів. Таке моделювання дозволяє зрозуміти, як різні фактори впливають на процес обслуговування, і знайти оптимальні рішення для його організації.

По суті, моделювання обслуговування покупців передбачає створення спрощеного представлення реальної системи магазину, щоб аналізувати її поведінку. Основна мета – вивчити, як формуються черги, скільки часу покупці витрачають на очікування та обслуговування, і як можна оптимізувати ці процеси.

Для побудови такої моделі зазвичай розглядають декілька ключових елементів. По-перше, це потік покупців – як часто вони приходять до магазину, чи є якісь пікові години або дні. По-друге, це самі точки обслуговування: каси, консультанти в залі, стійки видачі онлайн-замовлень, сервісні центри тощо. Для кожної такої точки важливо розуміти, скільки часу в середньому займає обслуговування одного покупця. По-третє, це організація черг – чи є одна загальна черга, чи декілька окремих до кожної точки обслуговування, і які правила формування цих черг. Нарешті, важливим елементом є ресурси – кількість працівників на кожній точці обслуговування.

Існує декілька підходів до моделювання таких систем. Класичним є використання теорії масового обслуговування, яка пропонує математичні формули для розрахунку характеристик черг (наприклад, середній час очікування) за певних умов. Більш гнучким та потужним методом є імітаційне моделювання, зокрема дискретно-подійне моделювання. Цей підхід дозволяє створити комп'ютерну модель, яка

відтворює поведінку системи крок за кроком, враховуючи випадкові фактори, такі як час прибуття покупців або тривалість обслуговування. Також може використовуватися агентне моделювання, де кожен покупець та працівник розглядається як окремий "агент" зі своєю поведінкою, що дозволяє аналізувати складні взаємодії та вплив, наприклад, планування магазину.

Завдяки моделюванню можна досягти багатьох цілей. Наприклад, визначити оптимальну кількість касирів чи консультантів у різні години роботи, щоб мінімізувати черги, але й не мати зайвого персоналу. Можна оцінити, як зміниться час очікування, якщо додати ще одну касу або запровадити нову технологію, наприклад, термінали самообслуговування. Моделі допомагають виявити "вузькі місця" в системі обслуговування та знайти шляхи їх усунення, що в кінцевому підсумку призводить до підвищення лояльності клієнтів та збільшення прибутку магазину.

Для побудови точної моделі потрібні дані: статистика прибуття покупців, час обслуговування на різних етапах, кількість персоналу. Ці дані можна зібрати шляхом спостережень, аналізу чеків або систем відеоспостереження.

Результати моделювання можуть бути представлені у вигляді звітів, графіків та візуалізацій, які наочно показують, як працює система за різних сценаріїв. Це дозволяє керівництву магазину приймати обґрунтовані рішення щодо оптимізації процесів обслуговування.

Для моделювання обслуговування покупців у магазині електронної техніки можна застосувати цілий арсенал математичних моделей та методів. Вибір конкретних інструментів залежить від складності системи, доступності даних та цілей моделювання.

Найбільш фундаментальною та широко використовуваною є теорія масового обслуговування (СМО), яку також називають теорією черг. Ця теорія дозволяє аналізувати системи, де надходять заявки на

обслуговування (покупці), існують канали обслуговування (касири, консультанти) та можуть утворюватися черги. В рамках СМО розглядають такі характеристики, як інтенсивність потоку заявок (як часто приходять покупці), час обслуговування, кількість каналів обслуговування та дисципліна черги (наприклад, "перший прийшов – перший обслужений", FIFO).

Приклад моделі M/M/1: Це найпростіша модель, де потік покупців описується розподілом Пуассона (M), час обслуговування – експоненційним розподілом (M), і є один канал обслуговування (1). Таку модель можна використати для аналізу роботи однієї каси або одного консультаційного пункту, куди покупці підходять випадковим чином, а час консультації чи розрахунку також має випадковий характер, але з відомим середнім значенням. Ця модель дозволяє розрахувати середню довжину черги, середній час очікування, завантаженість касира.

Приклад моделі M/M/c: Ця модель схожа на попередню, але тут розглядається 'c' паралельних каналів обслуговування. Наприклад, у магазині є 5 кас, що працюють одночасно, і покупці стають у загальну чергу до вільної каси. Модель допоможе визначити, як зміниться час очікування, якщо кількість кас зменшиться до 3 або збільшиться до 6 у пікові години.

Приклад моделі M/G/1: Тут потік заявок все ще пуассонівський, канал обслуговування один, але час обслуговування має загальний (General) розподіл, не обов'язково експоненційний. Це актуально, коли час обслуговування може сильно варіюватися і не підпорядковується простому експоненційному закону – наприклад, консультації щодо складної техніки, де одна може тривати 5 хвилин, а інша – 30.

Для опису випадкових процесів у цих моделях використовуються ймовірнісні розподіли.

Розподіл Пуассона часто застосовується для моделювання кількості покупців, що прибувають до магазину за певний проміжок часу (наприклад, за годину). Якщо в середньому до магазину приходить 20 покупців на годину, розподіл Пуассона допоможе оцінити ймовірність того, що за конкретну годину прийде рівно 15, або більше 25 покупців.

Експоненційний розподіл часто використовується для моделювання часу між послідовними прибуттями покупців або для моделювання тривалості обслуговування, якщо процес "не має пам'яті" (тобто, ймовірність завершення обслуговування найближчим часом не залежить від того, скільки воно вже триває). Наприклад, якщо середній час обслуговування на касі становить 3 хвилини.

Інші розподіли, такі як нормальний, гамма-розподіл, або Вейбулла, можуть використовуватися для більш точного моделювання часу обслуговування, якщо експоненційний розподіл не відповідає реальним даним. Наприклад, час налаштування складної техніки консультантом може краще описуватися гамма-розподілом.

Коли аналітичні рішення теорії масового обслуговування стають занадто складними або неможливими через специфіку системи, застосовують методи імітаційного моделювання.

Метод Монте-Карло є одним з підходів, де випадкові події (прибуття покупців, тривалість обслуговування) генеруються на комп'ютері відповідно до заданих ймовірнісних розподілів. Проводячи велику кількість таких "прогонів" моделі, можна отримати статистичні оцінки характеристик системи. Наприклад, можна симулювати роботу магазину протягом місяця, генеруючи щоденний потік клієнтів та час їх обслуговування, щоб побачити, як змінюватиметься завантаженість персоналу.

Дискретно-подійне моделювання (DES) є потужнішим методом. Система моделюється як послідовність подій: "покупець прибув", "покупець став у чергу", "касир звільнився", "покупець почав

обслуговуватися", "покупець пішов". Модель відстежує стан системи (довжини черг, зайнятість серверів) та просуває час від однієї події до іншої. Спеціалізоване програмне забезпечення (наприклад, Arena, SimPy, AnyLogic) дозволяє створювати детальні моделі магазину з різними типами покупців, різними зонами обслуговування та складними правилами взаємодії. Наприклад, можна змодельовати рух покупців по торговому залу, їх звернення до консультантів, а потім проходження через касову зону.

Для знаходження оптимальних рішень, наприклад, оптимальної кількості персоналу, можуть застосовуватися методи оптимізації.

Лінійне або цілочисельне програмування може бути використане, якщо потрібно мінімізувати витрати на персонал при дотриманні певних вимог до рівня обслуговування (наприклад, середній час очікування не повинен перевищувати 2 хвилини). Наприклад, задача може полягати у визначенні, скільки касирів та консультантів має працювати у кожную годину дня, щоб загальні витрати на зарплату були мінімальними, а рівень задоволеності клієнтів залишався високим.

Також важливими є статистичні методи аналізу даних та прогнозування.

Аналіз часових рядів (наприклад, моделі ARIMA або експоненційне згладжування) використовується для аналізу історичних даних про потік покупців та прогнозування їх кількості на майбутні періоди. Це дозволяє, наприклад, передбачити збільшення напливу клієнтів у передсвяткові дні та завчасно підготувати ресурси.

Регресійний аналіз може допомогти виявити залежності між різними факторами, наприклад, як кількість консультантів у залі впливає на середній чек або на час, проведений покупцем у магазині.

2.1 Моделювання поведінки покупців у магазині електронної техніки

Для ефективного моделювання поведінки покупців з урахуванням їхніх звичок та переваг використовують комбінацію методів:

1. Аналіз даних про споживачів

Збір даних через CRM-системи, онлайн-трекінг, анкети або Wi-Fi-аналітику в магазині дозволяє визначити демографічні характеристики, частоту відвідувань, улюблені товарні категорії.

Метод градієнтного бустінгу використовують для прогнозування ймовірності покупки на основі історичних даних про поведінку клієнтів.

2. Симуляційні моделі

Агентні моделі (ABM) імітують індивідуальні дії покупців, враховуючи їхні переваги, бюджет та вплив соціальних факторів. Наприклад, можна спрогнозувати, як зміна розташування товарів вплине на маршрути руху.

Дискретне моделювання подій допомагає аналізувати черги біля кас, ефективність роботи консультантів або вплив акцій на навантаження магазину.

3. Психологічні та соціальні фактори

Моделі враховують емоційні реакції на дизайн магазину, освітлення або музику, використовуючи дані з сенсорів або відеоспостереження.

Соціальні норми (наприклад, вплив оточення на вибір бренду) моделюються через аналіз групової динаміки в агентних системах.

4. Оптимізаційні інструменти

Системи на базі штучного інтелекту (наприклад, нейромережі) аналізують відеозаписи руху клієнтів, визначаючи "зони застою" чи популярні товарні зони.

A/B-тестування віртуальних макетів магазину дозволяє порівнювати різні сценарії розміщення електроніки до їх фізичної реалізації.

Приклад застосування:

Використання програмного забезпечення SIMUL8 для моделювання роботи магазину під час сезонних розпродажів. Модель враховує час обслуговування, кількість кас, вплив акційних пропозицій на потік клієнтів та оптимізує розклад роботи персоналу.

Для успішного впровадження моделей рекомендовано комбінувати мікроекономічні підходи (аналіз цінової еластичності) з психологічними аспектами (емоційна реакція на бренд).

2.2 Розробка математичної моделі системи масового обслуговування клієнтів в магазинах побутової техніки.

Розглядається задача моделювання роботи магазину електронної техніки, де обслуговування клієнтів здійснюється продавцями-консультантами. Цю ситуацію можна описати за допомогою системи масового обслуговування (СМО) [2] типу M/M/c, де:

- надходження клієнтів відбувається за пуасонівським потоком [3];
- час обслуговування має експоненційний розподіл [3];
- обслуговування здійснюється c незалежними каналами (продавцями).

Вхідні параметри моделі

Нехай:

λ – інтенсивність надходження клієнтів (заявок) [заявок/год];

μ – інтенсивність обслуговування одним продавцем [заявок/год];

c – кількість продавців-консультантів (каналів обслуговування);

$\rho = \lambda c \mu$ – коефіцієнт завантаження системи;

P_0 – ймовірність відсутності клієнтів у системі;

L_q – середня кількість клієнтів у черзі;

L – середня кількість клієнтів у системі;

W_q – середній час очікування в черзі;

W – середній час перебування клієнта в системі.

Далі наведемо формули для вирішення математичного моделювання:

Ймовірність відсутності клієнтів (усі продавці вільні):

$$P_0 = \left[\sum_{n=0}^{c-1} \frac{\left(\frac{\lambda}{\mu}\right)^n}{n!} + \frac{\left(\frac{\lambda}{\mu}\right)^c}{c!} \cdot \frac{1}{1-\rho} \right]^{-1}.$$

Середня кількість клієнтів у черзі:

$$L_q = \frac{\left(\frac{\lambda}{\mu}\right)^c \cdot \rho}{c! \cdot (1-\rho)^2} \cdot P_0.$$

Середня кількість клієнтів у системі:

$$L = L_q + \frac{\lambda}{\mu}.$$

Середній час очікування в черзі:

$$W_q = \frac{L_q}{\lambda}.$$

Середній час перебування в системі:

$$W = \frac{L}{\lambda}.$$

Тепер наведемо приклад розрахунку моделі:

Припустимо:

$\lambda=12$ клієнтів/год;

$\mu=5$ клієнтів/год (один клієнт обслуговується за 12 хв);

$c=3$ продавці-консультанти.

Розрахуємо:

$$\rho = \frac{12}{3 \cdot 5} = 0.8;$$

$$\frac{\lambda}{\mu} = \frac{12}{5} = 2.4;$$

$$P_0 = \left[1 + 2.4 + \frac{2.4^2}{2!} + \frac{2.4^3}{3!} \cdot \frac{1}{1-0.8} \right]^{-1};$$

$$P_0 \approx \left[1 + 2.4 + 2.88 + \frac{13.824}{6 \cdot 0.2} \right]^{-1} = [17.8]^{-1} \approx 0.056;$$

$$L_q = \frac{2.4^3 \cdot 0.8}{3! \cdot (1-0.8)^2} \cdot P_0 = \frac{13.824 \cdot 0.8}{6 \cdot 0.04} \cdot 0.056 \approx 3.08;$$

$$L = 3.08 + \frac{12}{5} = 5.48;$$

$$W_q = \frac{3.08}{12} \approx 0.257 \text{ год} \approx 15.4 \text{ хв};$$

$$W = 0.257 + 0.2 = 0.457 \text{ год} \approx 27.4 \text{ хв}.$$

У висновку ми можемо зазначити що:

- Середній час очікування клієнта в черзі – близько 15.4 хв;
- Загальний час перебування клієнта в магазині – близько 27.4 хв;
- Середня кількість клієнтів у системі – 5.48 осіб;
- Завантаження системи $\rho=0.8$, що свідчить про стабільну роботу.

2.3 Сучасні технології збору даних для аналізу звичок клієнтів онлайн

Для глибокого аналізу звичок і переваг клієнтів в онлайн-магазинах електронної техніки використовуються такі технології та інструменти:

- Customer Data Platform (CDP) – платформи на кшталт eSputnik автоматизують збір, уніфікацію та аналіз даних про клієнтів із різних джерел (сайт, мобільний застосунок, email, офлайн) [13]. Вся інформація потрапляє у єдиний профіль клієнта й оновлюється в режимі реального часу. Це дозволяє відстежувати поведінку користувачів, сегментувати аудиторію, будувати персоналізовані пропозиції та реалізовувати омніканальний маркетинг.
- Google Analytics – один із найпопулярніших безкоштовних інструментів для аналізу поведінки користувачів на сайті. Дозволяє відстежувати джерела трафіку, сторінки, що переглядаються, тривалість сесій, демографічні характеристики, інтереси та конверсії. Можливе налаштування цілей і аналіз ефективності рекламних кампаній у реальному часі [10].
- Mixpanel – орієнтується на відстеження конкретних дій користувачів (подій) на сайті чи в додатку, що дозволяє глибше аналізувати сценарії поведінки, сегментувати аудиторію за активністю, налаштовувати A/B-тестування та автоматизувати email-розсилки.
- Hotjar – інструмент для візуального аналізу поведінки користувачів: теплові карти кліків, скролінгу, записи сесій, опитування. Дозволяє зрозуміти, які елементи сайту викликають найбільший інтерес або труднощі у відвідувачів, і оптимізувати юзабіліті [11].

- Woopra – платформа для аналізу повного життєвого циклу клієнта: від першого відвідування до повторних покупок. Відстежує дії користувача на сайті, в додатку, email-розсилках і техпідтримці, формує детальні профілі і воронки продажів у реальному часі [12].

- SimilarWeb – дозволяє аналізувати трафік, джерела відвідувань, інтереси аудиторії, популярні сторінки як на власному сайті, так і у конкурентів. Корисний для визначення ринкових тенденцій та каналів залучення клієнтів.

- Adobe Analytics – преміум-рішення для великих компаній з розширеними можливостями сегментації, аналізу великих масивів даних і побудови складних звітів.

- Інтеграція з CRM, DWH, BigQuery, Power BI – сучасні CDP та аналітичні системи дозволяють інтегрувати дані з різних платформ, створюючи єдиний центр аналітики для глибокого аналізу звичок клієнтів і прийняття стратегічних рішень.

Ці інструменти дозволяють автоматизувати збір даних, мінімізувати людський фактор і підвищити точність аналізу поведінки клієнтів, що є основою для персоналізації пропозицій та підвищення ефективності онлайн-продажів.

2.3 Результати розробки логічних моделей застосунку та їх відображення у вигляді діаграм

На рисунку 4 представлена BPMN діаграма для робочого процесу з використанням застосунку.

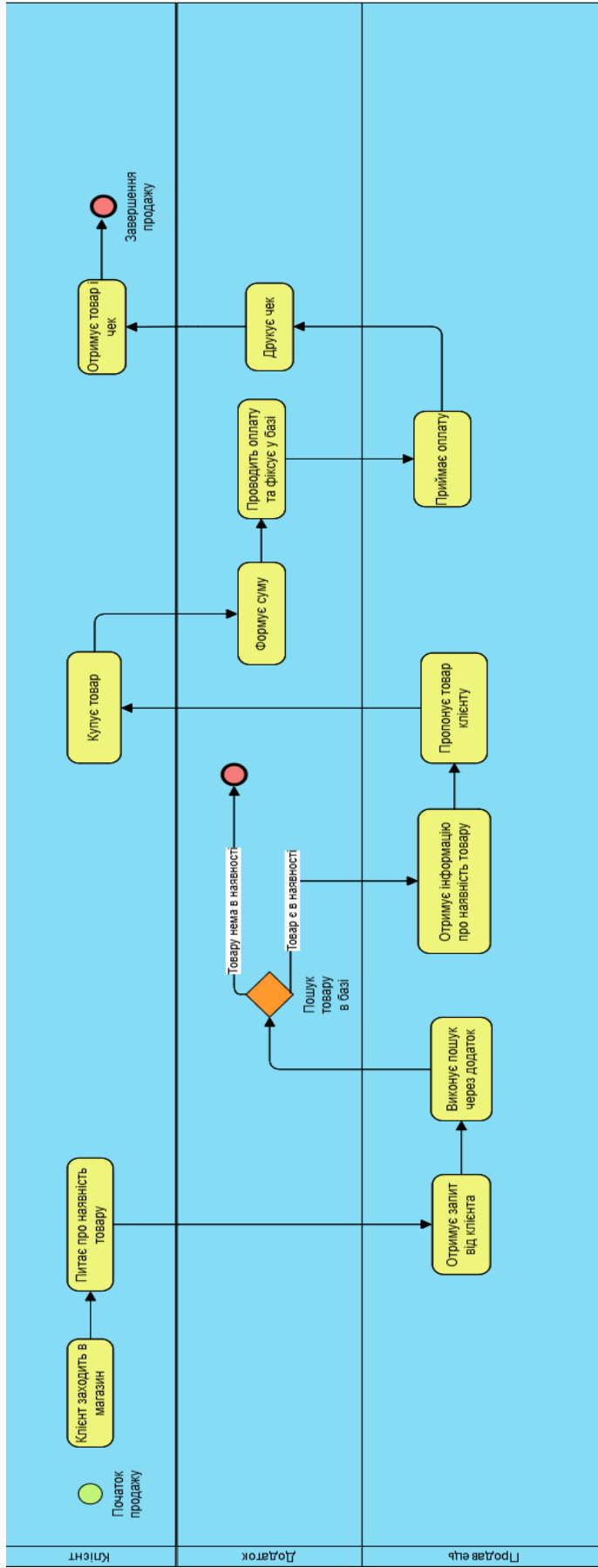


Рисунок 4 – BPMN діаграма робочого процесу з використанням застосунку

На діаграмі ми можемо побачити 3-х учасників процесу, це: Клієнт, Продавець та Додаток(Застосунок). Початок обумовлений тим, що наш клієнт зайшов до магазину за певним товаром, після чого продавець отримує цей запит та вирішує виконати пошук через додаток. Тепер починається розгалуження: або товар присутній на магазині, або ні. Якщо він відсутній продавець повідомляє про це клієнта та процес рахується завершеним. У разі наявності товару продавець отримує інформацію та починає пропонувати товар замовнику. В свою чергу він купує потрібний товар, оплачує його та дія переходить знов до продавця, якому потрібно провести цю продаж в додатку. Застосунок формує суму, фіксує продаж у базі та після друкує чек і дія переходить до продавця який передає чек клієнту. На цьому процес можна вважати завершеним.

На рисунку 5 наведено діаграму варіантів використання застосунку.

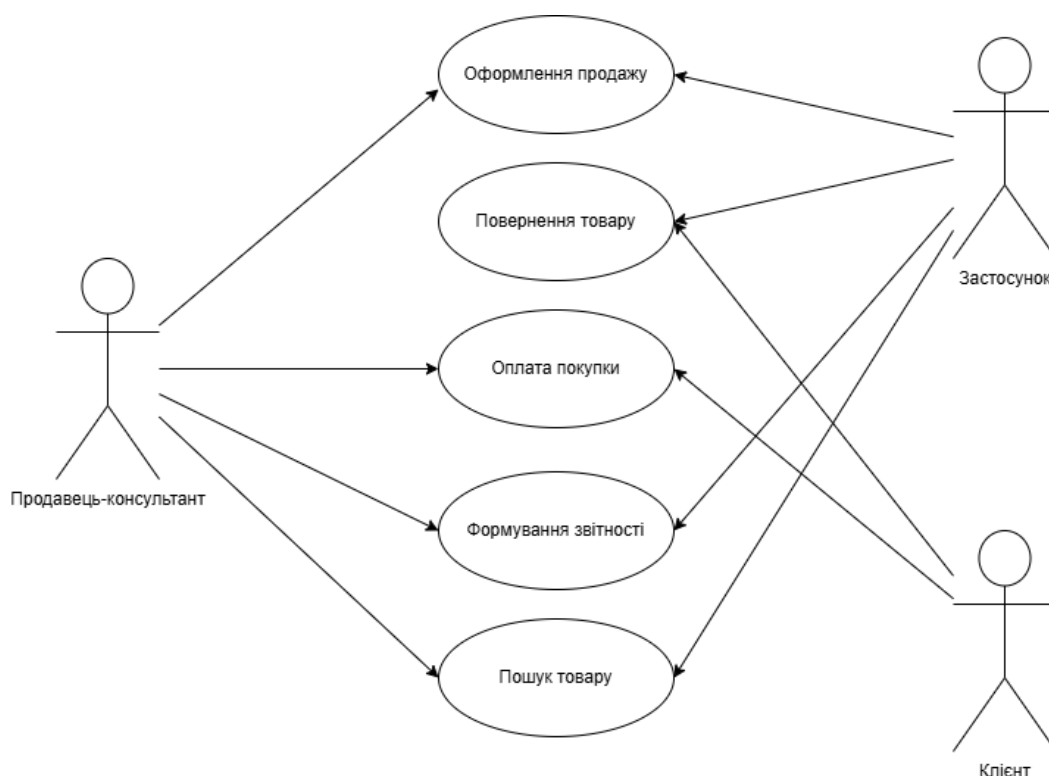


Рисунок 5 – Діаграма варіантів використання застосунку

На діаграмі ми можемо побачити трьох учасників процесу: Продавець-консультант, Клієнт та Платіжна система (Застосунок). Основна взаємодія починається з оформлення продажу (UC-1), яке робиться продавцем і включає в себе декілька підпроцесів, такі як оплату покупки, що виконується через платіжну систему. Паралельно клієнт може ініціювати процес повернення товару (UC-2), який також пов'язаний із процесом оплати, оскільки повернення передбачає грошове відшкодування. Окремо був реалізований процес формування звіту (UC-3), який автоматично генерується системою на основі продажів, повернень та оплат. Цей процес є частиною звітності або контролю діяльності у нашому застосунку. Окрім того, перед оформленням продажу може бути виконано пошук товару (UC-4), що є важливою функцією в процесі обслуговування клієнта – продавець повинен знайти потрібний товар, перш ніж його продати чи запропонувати альтернативу.

Також під час проєктування програмного комплексу було розроблено діаграму активності, результати наведено на рисунку 6.

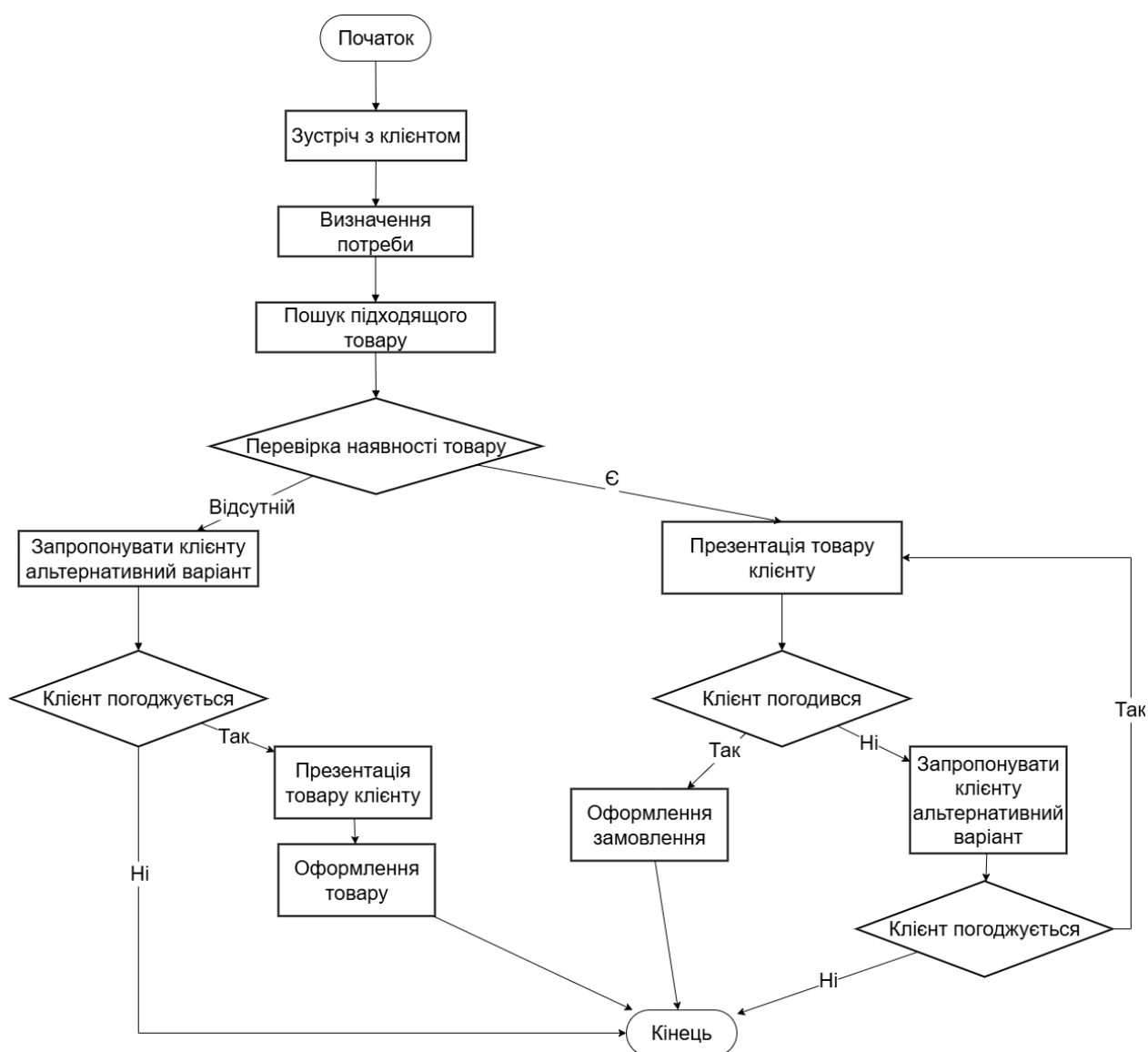


Рисунок 6 – Діаграма активності

На діаграмі наведено двох основних учасників процесу: це Клієнт та Продавець. Початок процесу розпочинається із зустрічі з клієнтом, під час якої продавець виявляє потребу покупця. Після цього він переходить до пошуку підходящого товару.

Далі відбувається перевірка наявності товару. Тут процес розгалужується: якщо товар є в наявності, продавець одразу презентує його клієнту. Після презентації клієнт приймає рішення: погоджується або ні. У разі згоди продавець оформлює замовлення і процес вважається завершеним. Якщо ж клієнт не погоджується, йому пропонують альтернативний варіант. Якщо нова пропозиція його влаштовує – товар знову презентується, оформлюється, і на цьому процес завершується. У разі відмови – процес також завершується, але без оформлення покупки. У ситуації, коли товар відсутній, продавець одразу пропонує клієнту альтернативу. Подальші дії аналогічні: якщо клієнт погоджується, то товар презентується та оформлюється, якщо ні – процес припиняється.

2.4 Технічні вимоги до програмного забезпечення, що буде розроблено для подальшої оптимізації роботи продавців консультантів

Для забезпечення стабільної та ефективної роботи програмного забезпечення необхідно дотримуватись певних технічних вимог, які стосуються як апаратного забезпечення, так і програмного середовища. Ці вимоги визначають, на яких пристроях система буде працювати, які компоненти повинні бути встановлені, а також які зовнішні ресурси можуть знадобитися під час її експлуатації.

Передусім, програмне забезпечення повинно функціонувати на комп'ютерах або ноутбуках із встановленою операційною системою Windows 10 або новішою. Рекомендується мати не менше 4 ГБ оперативної пам'яті для комфортної роботи, хоча оптимальним є 8 ГБ і більше, особливо якщо паралельно використовуються інші програми. Щодо процесора – достатньо моделі з двома або більше ядрами (наприклад, Intel Core i3 або AMD Ryzen 3), але для швидшої обробки даних бажано використовувати продуктивніші моделі.

Дисковий простір, необхідний для встановлення програми та збереження даних, становить приблизно 500 МБ, але із запасом варто передбачити не менше 2–3 ГБ на жорсткому диску або SSD. Також потрібно, щоб пристрій мав стабільне з'єднання з Інтернетом, якщо передбачається обмін даними із зовнішніми сервісами, надсилання електронних чеків або резервне копіювання в хмару.

Програмне забезпечення розробляється на мові C#, що вимагає встановленого .NET середовища (версія не нижче .NET 6). Для зручності обслуговування і тестування програма повинна бути сумісна з Visual Studio, оскільки саме в цьому середовищі здійснюється розробка.

У якості системи зберігання даних використовується база даних SQLite, яка має низькі вимоги до ресурсів і не потребує встановлення окремого серверного програмного забезпечення. Це дозволяє використовувати програму на окремих автономних пристроях, що є зручним для торгових точок без складної IT-інфраструктури.

Крім того, програмне забезпечення має бути розраховане на роботу з базами даних невеликого та середнього обсягу, із можливістю резервного копіювання, шифрування даних та авторизації користувачів. Інтерфейс повинен бути адаптований під екрани різних розмірів, з мінімальним навантаженням на систему.

Усі ці технічні вимоги забезпечують надійність, безпеку, швидкодію та масштабованість програми, що дозволяє її ефективно використовувати в реальних умовах роздрібної торгівлі.

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ РОБОТИ ПРОДАВЦІВ-КОНСУЛЬТАНТІВ В МАГАЗИНАХ ПОБУТОВОЇ ТЕХНІКИ

Розроблюване програмне забезпечення являє собою десктопний застосунок для ОС Windows, реалізований мовою програмування C# із використанням технології Windows Forms. Основною метою створення системи є інформаційна підтримка щоденної роботи продавця-консультанта, зменшення кількості рутинних операцій та підвищення точності обліку.

3.1 Вибір та обґрунтування засобів розробки

Мова програмування C# була обрана для розробки програмного забезпечення через свої численні переваги, які роблять її ідеальним інструментом для створення високоякісних додатків. C# є потужною об'єктно-орієнтованою мовою, розробленою компанією Microsoft для платформи .NET. Однією з головних причин її вибору є зручність у розробці, висока продуктивність та підтримка сучасних технологій, що робить її підходящою як для простих, так і для складних програм [1], [5].

C# має чітко структурований синтаксис, який дозволяє розробникам швидко освоїти мову та розпочати ефективну роботу. Це

особливо важливо для командної розробки, де важлива зрозумілість і стандартизація коду. Завдяки своїй об'єктно-орієнтованій природі, C# дозволяє створювати модульний, підтримуваний і розширюваний код, що є основою для побудови масштабованих додатків.

Мова програмування C# тісно інтегрується з платформою .NET, що надає величезну кількість вбудованих бібліотек для роботи з різними типами даних, файловими системами, мережею, базами даних, графікою тощо. Це значно спрощує розробку, оскільки багато стандартних завдань вже мають готові рішення у вигляді бібліотек і фреймворків, що дозволяє зосередитися на бізнес-логіці програми.

Однією з головних переваг C# є її кросплатформність. З виходом .NET Core та .NET 5 (та новіших версій), C# став доступним не лише для Windows, але й для Linux та macOS. Це дозволяє розробляти додатки для різних операційних систем, зберігаючи високу продуктивність і стабільність програми на кожній з платформ.

C# підтримує синтаксис, який є зрозумілим для розробників, знайомих з іншими популярними мовами програмування, такими як Java або C++. Це дозволяє швидко переходити між мовами, якщо необхідно, або використовувати досвід з інших проєктів при роботі з C#. Крім того, мова підтримує сучасні концепції програмування, такі як лямбда-вирази, асинхронне програмування, LINQ (Language Integrated Query) для роботи з даними, що робить код більш компактним та зручним для розробників.

C# має потужну систему управління пам'яттю, що автоматично виконується за допомогою збору сміття (Garbage Collection). Це дозволяє уникати багатьох проблем, пов'язаних з управлінням пам'яттю в інших мовах, таких як C++ або C. Оскільки система автоматично звільняє непотрібну пам'ять, розробник може зосередитися на вирішенні задач, замість того щоб витрачати час на ручне управління пам'яттю.

Ще однією важливою перевагою є підтримка багатозадачності та асинхронного програмування, що дає змогу ефективно працювати з великими обсягами даних та високими навантаженнями. Завдяки конструкціям `async/await`, розробники можуть писати асинхронний код так, як якщо б він був синхронним, що значно спрощує розробку додатків з високими вимогами до швидкості виконання та реактивності.

C# також активно підтримує найновіші інновації в області програмування, зокрема, роботи з базами даних, хмарними технологіями та інтерфейсами користувача. Мова має вбудовані можливості для роботи з різними базами даних, зокрема за допомогою Entity Framework, що дозволяє створювати об'єктно-реляційне відображення для баз даних і легко взаємодіяти з ними. Це дозволяє зменшити час на розробку та підвищити ефективність роботи з даними.

Завдяки цьому набору можливостей, C# є надзвичайно ефективним інструментом для створення різноманітних додатків – від простих десктопних програм до складних серверних рішень і мобільних застосунків. Завдяки своїй гнучкості, зручності та підтримці широкого спектру технологій, C# стає вибором номер один для багатьох розробників по всьому світу.

Середовище розробки Visual Studio було обране для створення програмного забезпечення через свою зручність, багатофункціональність та широкі можливості, які воно надає розробникам. Visual Studio є однією з найбільш популярних та широко використовуваних середовищ для розробки програмного забезпечення. Воно підтримує різноманітні мови програмування, зокрема C#, C++, Python, JavaScript та інші, що дозволяє створювати програми для різних платформ і задач. Завдяки цьому, середовище підходить для розробки різного роду програм – від простих додатків до складних корпоративних рішень.

Однією з головних причин вибору Visual Studio є її інтеграція з .NET Framework та .NET Core [6]. Це дозволяє розробляти додатки не лише для Windows, а й для інших платформ, таких як веб-сайти, мобільні програми для iOS та Android, а також розподілені сервіси. Ідеальне поєднання C# та Visual Studio дозволяє створювати швидкі, надійні та ефективні програми з мінімумом часу на розробку.

Важливою особливістю Visual Studio є набір інструментів для налагодження та тестування програм, що робить цей процес максимально зручним. Завдяки вбудованим засобам відладки можна швидко знаходити помилки в коді та виправляти їх на ранніх етапах розробки, що суттєво економить час. Для більш складних проєктів, де важливо відстежувати помилки, має сенс використовувати багатофункціональні засоби профілювання та аналізу продуктивності, які дозволяють зрозуміти, як програма працює на різних етапах виконання та де можуть виникати вузькі місця.

Автодоповнення коду та інтелектуальний аналіз забезпечують високу точність у написанні програм. Вони допомагають швидше знаходити помилки, уникати граматичних помилок і швидко знаходити відповідні методи, функції або змінні в рамках великого проєкту. Це значно підвищує продуктивність розробників і дозволяє їм зосередитися на вирішенні складних задач, а не на рутинному написанні коду.

Також важливою перевагою Visual Studio є підтримка системи контролю версій, зокрема Git, що дає змогу працювати з репозиторіями, здійснювати ефективну командну розробку та забезпечувати постійну синхронізацію коду. Завдяки цьому, розробники можуть відстежувати всі зміни, що відбуваються в коді, повернутися до попередніх версій програми та впевнено інтегрувати нові функції без ризику порушити роботу всього проєкту.

В Visual Studio також є потужні інструменти для роботи з базами даних. Вона підтримує роботу з різними типами баз даних, зокрема SQL

Server, SQLite та інші. Це дає змогу швидко інтегрувати програми з базами даних і забезпечувати ефективне зберігання та обробку даних.

Для покращення роботи з програмою Visual Studio надає можливість встановлення різноманітних плагінів та розширень, що дозволяє налаштувати середовище відповідно до вимог конкретного проєкту чи індивідуальних потреб розробника. Це може бути корисно для автоматизації певних процесів або додавання нових функціональних можливостей, що підвищує гнучкість і зручність роботи з IDE.

Завдяки зручному інтерфейсу, широким можливостям налаштування та інтеграції з різними сервісами, Visual Studio є ідеальним інструментом для розробки програмного забезпечення, особливо коли мова йде про створення складних додатків, які вимагають високої продуктивності та надійності. Середовище надає всі необхідні інструменти для ефективної та швидкої розробки програмного забезпечення, що робить його одним із найкращих виборів для розробників у різних сферах.

SQLite була вибрана як система управління базами даних для цього проєкту через свої численні переваги, що роблять її ідеальним вибором для малих та середніх підприємств, де важлива ефективність та простота використання. SQLite [4] є вбудованою, реляційною базою даних, яка не потребує додаткового серверного програмного забезпечення для роботи. Це значно знижує витрати на налаштування та обслуговування бази даних, оскільки не потрібно розгортати окремі сервери або управляти складною інфраструктурою.

Однією з основних переваг SQLite є її компактність. Всі дані зберігаються в одному файлі, що значно спрощує резервне копіювання та переносимість даних. Це ідеально підходить для додатків, які не потребують складних багатокористувацьких або розподілених рішень, а

також для середовищ, де обмежені ресурси або де потрібно зберігати дані локально без взаємодії з центральним сервером.

SQLite підтримує стандарт SQL, що дозволяє розробникам використовувати добре знайомі запити для взаємодії з базою даних, забезпечуючи сумісність з іншими реляційними СУБД. Це дозволяє спрощувати розробку, оскільки немає потреби вивчати новий синтаксис або специфічні інтерфейси для роботи з базою даних. За допомогою стандартних SQL-запитів можна виконувати всі необхідні операції, такі як вибірка даних, вставка, оновлення та видалення, а також створення складних запитів для аналізу та обробки інформації.

Ще однією суттєвою перевагою SQLite є її висока швидкість роботи, особливо при роботі з невеликими та середніми обсягами даних. Завдяки тому, що всі дані зберігаються локально і немає необхідності у взаємодії з сервером, операції на базі даних виконуються дуже швидко. Це дозволяє досягти високої продуктивності навіть на слабших пристроях, що важливо для невеликих магазинів, де можливості серверного обладнання можуть бути обмежені.

SQLite також відома своєю надійністю. Оскільки ця база даних працює в межах одного файлу, це знижує ймовірність помилок при доступі до даних, а також спрощує відновлення після збоїв. Крім того, SQLite підтримує транзакційність, що гарантує цілісність даних навіть у випадку аварійних зупинок програми або системи. Ця особливість важлива для додатків, де забезпечення коректного збереження даних є пріоритетом.

Ще одна важлива перевага SQLite полягає в її кросплатформності. SQLite працює на різних операційних системах, включаючи Windows, macOS, Linux, а також мобільні платформи, такі як iOS і Android. Це дає змогу використовувати її в різних середовищах без необхідності вносити зміни до структури бази даних або використовувати різні інструменти для кожної платформи.

Завдяки своїй простоті, ефективності та надійності, SQLite є ідеальним рішенням для малих і середніх підприємств, де важлива швидка розробка, низькі витрати на підтримку та стабільна робота з даними без необхідності в складних серверних налаштуваннях. Вона забезпечує зручний інтерфейс для роботи з базами даних і дозволяє зосередитися на розробці функціоналу програми, а не на технічних аспектах управління даними.

.NET 6 – це кросплатформна відкрита платформа для розробки програмного забезпечення, створене компанією Microsoft. Вона є частиною об'єднаної лінійки .NET, яка з версії 5 стала єдиною спадкоємицею попередніх технологій – .NET Framework, .Net Core та Xamarin[25].

Вийшла в листопаді 2021 року як LTS реліз. Це означає підтримку від компанії Microsoft протягом трьох років[26]. Саме через цей факт .NET 6 вважається стабільною та надійною платформою для створення як комерційних, та і навчальних програмних рішень.

Також ключовою його перевагою є уніфікована платформа, яка дозволяє розробляти додатки для різних середовищ, наприклад: Windows, Linux, Android, iOS, MacOS, веб, хмарні платформи та десктопні програми[27].

Саме тому для виконання кваліфікаційної роботи було обрано .NET 6 для реалізації десктопного застосунку на операційній системі Windows, що працює через технологію Windows Form. Це є перевірене рішення для побудови інтерфейсу з яким буде взаємодіяти користувач.

.NET 6 підтримує сучасні мови програмування C# 10 та F#, що дозволило нам створити лаконічний, безпечний та читабельний код. Треба зазначити що обрана нами платформа має потужну систему управління залежностями, підтримує асинхронне програмування та має зручні можливості роботи з базами даних, розвинену систему

тестування, журналювання, локалізації а також гнучкі засоби налагодження й профілювання застосунків.

Крім того .NET 6 забезпечує високу продуктивність. Якщо брати в увагу оцінки самої компанії Microsoft, в порівнянні з попередніми версіями .NET, у багатьох сценаріях продуктивність виросла на десятки разів. Це є важливим фактором для обробки великої кількості даних, виконанні обчислень або роботі з базами даних.

Однією з причин, чому я обрав для виконання кваліфікаційної роботи саме .NET 6 – це стабільність, довгострокова підтримка та активна екосистема. На момент розробки застосунку ця платформа має численні навчальні ресурси, підтримку з боку спільноти, документацію українською та багатьма іншими мовами. Вона сумісна з багатьма інструментами розробки, зокрема Visual Studio 2022 [28], яке і було використано для написання застосунку, що значно полегшило його створення, налагодження й розгортання.

Розроблений мною застосунок потребував реалізації багатьох функцій: робота з базою даних SQLite для збереження даних, створення графічного інтерфейсу користувача, обробку багатьох подій, формування звітності по касі в довільних періодах, що було чудово реалізовано за допомогою WinForms, який вбудовано в .NET 6. Окрім того, саме ця версія дає змогу легко розгортати автономні застосунку без необхідності встановлення .NET на цільовий комп'ютер майбутнього користувач, що є великою перевагою для офлайн-застосунків у сфері торгівлі.

Таким чином, вибір .NET 6 для створення десктопного програмного забезпечення в межах моєї кваліфікаційної роботи був обґрунтованим і стратегічно вдалим рішенням, яке забезпечило надійність, масштабність і перспективність подальшого розвитку розробленого застосунку.

ADO.NET – набір бібліотек у платформі .NET, який призначений для доступу до даних з різних джерел, таких як реляційні бази даних (SQL Server, SQLite, Oracle), XML-файли або інші джерела, що підтримують підключення через драйвери. Це одна з ключових технологій, яка забезпечує роботу з базами даних у середовищі .NET, зокрема у WinForms, ASP.NET та інших типах додатків. Вона надає інструменти для виконання SQL-запитів, збереження результатів, оновлення даних та керування транзакціями[29].

Основною перевагою ADO.NET є висока швидкодія, стабільність та контроль над базою з даними на низькому рівні.

ADO.NET застосовується на двох головних підходах: підключений і відключений. Підключений режим передбачає прямий і постійний зв'язок з базою даних під час виконання запитів. Відключений режим використовує такі об'єкти як DataSet і DataTable, які дозволяють зберігати дані локально, працювати з ними в оперативній пам'яті без постійного з'єднання з базою даних, а потім зберігати зміни назад. Така гнучкість дозволяє ефективно керувати ресурсами й оптимізувати продуктивність у додатках з великою кількістю запитів [30].

Основні функціональні можливості ADO.NET включають в себе [31]:

- Створення та керування з'єднанням з базою даних (SqlConnection, SQLiteConnection, OleDbConnection).
- Виконання SQL-запитів та збережених процедур (Command, ExecuteReader, ExecuteNonQuery, ExecuteScalar).
- Отримання даних за допомогою DataReader або DataAdapter.
- Робота з відключеними наборами даних (DataSet, DataTable, DataView).
- Підтримка транзакцій (SqlTransaction) для гарантування цілісності операцій.

- Обробка параметризованих запитів, що покращує безпеку.
- Інтеграція з XML для обміну даними між різними джерелами або форматами.

Застосовується ADO.NET у будь-яких типах програм, де потрібен доступ до даних – від простих настільних додатків до великих розподілених інформаційних систем. Зокрема ця технологія є фундаментом для більш сучасних підходів, таких як Entity Framework, які будуються поверх ADO.NET і автоматизують багато рутинних процесів. Проте, ADO.NET залишається популярним вибором, коли потрібна максимальна продуктивність, контроль над SQL-запитами або обмежена складність програми.

Вибір ADO.NET у межах моєї кваліфікаційної роботи зумовлений кількома практичними причинами. По-перше, мій застосунок – це настільна програма на Windows Forms, яка має працювати зі вбудованою базою даних SQLite. ADO.NET надає прямий доступ до цієї бази через драйвер System.Data.SQLite [32], дозволяючи мені реалізувати ефективну взаємодію з локальними даними без складної конфігурації. По-друге, використання ADO.NET дозволило реалізувати зручний механізм додавання, редагування та видалення товарів, клієнтів, історії продажів і генерації звітів через стандартні SQL-запити, які я можу контролювати й змінювати вручну. Це дало змогу глибше зрозуміти механізми взаємодії додатку з базою даних, що є важливим з навчальної точки зору.

Ще одним важливим фактором є простота інтеграції ADO.NET з WinForms. Оскільки технологія підтримується на рідному рівні .NET 6, немає потреби підключати сторонні ORM або складні бібліотеки – усе необхідне вже доступне в у стандартній бібліотеці .NET. Це суттєво спростило процес розробки застосунку що дозволило зосередитись на бізнес-логіці та забезпечило стабільну роботу програми без зайвих

накладних витрат. Також ADO.NET дозволяє працювати з транзакціями та параметрами, що забезпечує безпечне та надійне оновлення даних у реальному часі.

Висновок з вище переліченого виходить такий: ADO.NET – це перевірене, гнучке й потужне рішення для доступу до даних у середовищі .NET, яке дозволило мені в процесі розробки застосунку створити якісне, надійне і масштабне програмне рішення для автоматизації роботи продавця-консультанта. Простота використання, висока продуктивність та повна інтеграція з .NET 6 зробило цю платформу оптимальним вибором для моєї кваліфікаційної роботи.

Насамперед потрібно визначитися з тим, хто буде користуватися застосунком. Це продавці консультанти яким найважливіше це простота та швидкість знаходження інформації. Тому потрібно не перезавантажувати систему різними непотрібними вкладками. Це дозволить:

- Збільшити швидкість обробки замовлення на торговій точці.
- Нові продавці швидше адаптуватимуться до застосунку

Програма, яку буде розроблено, повинна мати наступні основні функції:

1. Ведення обліку товару на торговій точці(«Склад»)

Функція відстеження товарів у точці продажу реалізована як окрема підсистема та дозволяє зберігати, контролювати та оновлювати інформацію про асортимент товарів, доступних у вашому магазині побутової техніки. Ці функції забезпечують точне відображення поточного стану, що дозволяє уникнути дефіциту або надлишку продукції, оперативно реагувати на зміни попиту та ефективно планувати закупівлі.

Система фіксує повну інформацію про шкіряні вироби, включаючи назву, унікальний код або артикульний номер, кількість одиниць, ціну,

категорію товару та інші додаткові характеристики. Це дозволяє швидко шукати, сортувати та аналізувати категорії на основі повних параметрів.

Користувачі можуть додавати нові продукти до бази даних, редагувати існуючі записи та оновлювати кількість продуктів або їх вартість на основі поточних змін. Крім того, кількість товарів автоматично оновлюється після кожної операції продажу, що забезпечує точне відображення залишків на складі.

Ця функція також дозволяє відстежувати товари, які ось-ось будуть розпродані, визначаючи найпопулярніші або, навпаки, закріплені товари, що допомагає оптимізувати ваші запаси. Використання цих управлінських рішень забезпечує прозорість складського обліку та ефективність загальної транзакційної діяльності.

Отже, система обліку товарів має не лише довідкову функцію, а й аналітичну, щоб ви могли контролювати обіг товарів з новим якісним рівнем, що є запорукою успішної роботи будь-якої торговельної мережі.

2. Продаж товару («Продажі»)

Функція продажу товарів у застосунку відіграє центральну роль в обліку продажів у магазинах побутової техніки. Він забезпечує повний цикл обліку кожного продажу: від вибору товару та клієнта до автоматичного розрахунку загальної суми та збереження інформації про транзакцію в базі даних. Це забезпечує точний облік та прозорість усіх фінансових операцій.

Під час здійснення продажу користувач вибирає один або декілька товарів, вказує кількість кожного з них та записує дані про покупця. Вся інформація обробляється автоматично – система розраховує загальну вартість, враховує кількість одиниць та формує повну структуру продажів, яка включає список придбаних товарів, дату та час угоди, загальну суму та відповідного клієнта.

Кожна транзакція продажу записується в історію, що дозволяє переглядати минулі покупки в будь-який час, повторно аналізувати або

створювати звіти. Це також дозволяє контролювати наявність товарів на складі, оскільки кількість відповідного товару автоматично зменшується після завершення продажу. Це забезпечує безперервний зв'язок між підсистемами обліку товарів, обліку клієнтів та касового апарату.

Ця функція дозволяє користувачам ефективно реєструвати транзакції, мінімізувати ризик помилок та підвищити точність бухгалтерського обліку. Швидкий, структурований та прозорий облік продажів допомагає автоматизувати всю торговельну діяльність та створює надійну інформаційну основу для управлінських рішень.

3. Ведення каси

Функція ведення каси в додатку реалізована як засіб швидкого та гнучкого контролю фінансових показників магазину побутової техніки. Користувачі можуть вибрати будь-який період часу для створення звіту про готівку: це може бути певний день, тиждень, місяць або будь-який діапазон дат. Такий підхід дозволяє аналізувати фінансову діяльність вашого магазину у зручному для користувача форматі.

Протягом вибраного періоду часу програма обробляє дані про продажі та генерує зведену статистику, що відображає загальний дохід, кількість продажів, середній чек та інші показники. Це дозволяє вам швидко оцінити, наскільки ефективно працює ваш магазин протягом певного періоду часу.

Крім того, аналітичні можливості реалізовані у вигляді рейтингів, які дозволяють глибше зрозуміти динаміку продажів. Зокрема, він формує список днів з найвищим рівнем прибутку, що дозволяє визначити пікові періоди торгівлі. Також створюється список найпопулярніших товарів за обсягом продажів, щоб ви могли проаналізувати попит. Крім того, створюється список найактивніших клієнтів, який можна використовувати для персоналізації, орієнтованої на постійних клієнтів.

Загалом, функція каси дозволяє не лише отримувати основні фінансові показники, але й проводити більш поглиблений статистичний аналіз, що допомагає приймати розумні управлінські рішення та покращувати діяльність магазину.

4. Ведення клієнтської бази («Клієнти»)

Функція обслуговування клієнтів у застосунку використовується для централізованого зберігання та обліку інформації про покупців магазинів побутової техніки. Впроваджена система дозволяє легко додавати, переглядати, редагувати та видаляти записи клієнтів, а також швидко отримувати доступ до повної історії взаємодій з кожним клієнтом.

Бази даних клієнтів зберігають основну інформацію про покупців, включаючи ім'я, прізвище, по батькові, контактні дані та іншу додаткову інформацію, яка може допомогти в ідентифікації або подальшому спілкуванні. Це створює умови для персоналізованого підходу до обслуговування, а також забезпечує високоякісну підтримку постійним клієнтам.

Система також дозволяє відстежувати покупки конкретних клієнтів, що особливо важливо для створення звітів, програм лояльності або статистичного аналізу. Маючи повну клієнтську базу під рукою, керівництво магазину може легко виявляти активних покупців, аналізувати повторні продажі та покращувати якість обслуговування загалом.

Тому підтримка клієнтської бази служить не лише орієнтиром, але й важливим інструментом для підвищення рівня клієнтоорієнтованості, покращення взаємодії з клієнтами та підтримки довгострокових ділових відносин.

3.2 Представлення та опис коду розробленого програмного забезпечення

Наш застосунок складається з багатошарової структури яка дозволяє відокремлювати інтерфейс від логіки обробки даних і взаємодією бази даних [24]. Це дозволило нам змінювати якійсь один шар без необхідності торкатися інших що значно полегшить його майбутню підтримку та дасть змогу модифікувати і доповнювати наш застосунок.

Додаток складається з таких шарів як:

Presentation Layer (Forms) – користувацький інтерфейс, те с чим буде взаємодіяти наш користувач.

Businnes Logic Layer (Services) – бізнес логіка нашого застосунку. Він відповідає за всі розрахунки які ми будемо проводити під час роботи (оплата, підрахунок відсотка знижки, вираховування щоденної каси тощо.)

Data Access Layer (Models, DatabaseService) – робота з нашими даними, можливість їх редагування прямо в застосунку.

Database Layer (SQLite) – зберігання даних в програмному забезпеченні. Дає можливість зберегти всі зміни, які були внесені користувачами.

Як і було розписано вище багатошарова структура дозволила нам розподілити наш додаток по певним шарам які виконують окремі функції.

Далі можна переглянути структуру застосунку і яка його частина за що відповідає. На рисунку 7 наведено структуру застосунку, тепер перейдемо до його детального аналізу.

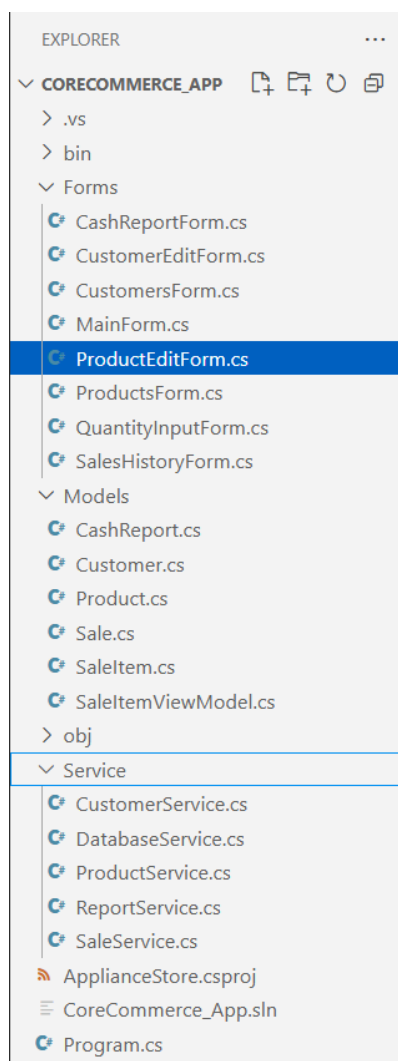


Рисунок 7 – Структура застосунку

Program.cs – точка входу в наш застосунок. Містить в собі метод Main(), який і дає змогу запустити застосунок та ініціалізує головну форму (MainForm)

Models – папка, в якій знаходяться всі наші моделі даних.

Product.cs – містить в собі визначення класу Product, який представляє собою товар у магазині побутової техніки (телевізор, телефон, пральна машина тощо.). Цей клас ми використовуємо для передачі інформації між базою даних, сервісами та форматами.

Customer.cs – містить в собі інформацію про клієнта магазину. Його ім'я, прізвище, номер телефону та його е-майл. Цей файл

важливий для побудування клієнтської бази для майбутніх розсилок з акціями та пропозиціями.

Sale.cs – представляє собою продаж. Фактично цей файл дає нам змогу проводити продажі та архівувати їх в нашій системі.

SaleItem.cs – містить в собі одиницю товару на момент продажу. Тобто його назву, вартість, кількість проданих в один момент товарів на момент самої продажі і його проведення в системі.

CashReport.cs – модель для звітності каси. Містить можливість підрахувати кількість зароблених грошей за певний період часу для звірки та аналізу ефективності продаж на магазині.

Services – серце нашого застосунку. Тут відбуваються всі основні процеси опрацювання даних – розрахунки, взаємодія з базою даних, перевірки, агрегації тощо.

Форми UI викликають сервіси, а не безпосередньо працюють з моделями або БД.

DatabaseService.cs – універсальний інтерфейс для роботи з базою даних SQLite. Реалізує відкриття та закриття з'єднання, виконання SQL-запитів. Інші сервіси використовують DatabaseService.cs для запиту або читання.

Реалізація створення бази даних у вигляді коду наведено на рисунку 8.

```

public static class DatabaseService
{
    private static readonly string dbPath = Path.Combine(Application.StartupPath, "store.db");
    private static readonly string connectionString = $"Data Source={dbPath};Version=3;";

    public static void InitializeDatabase()
    {
        try
        {
            if (!File.Exists(dbPath))
            {
                SQLiteConnection.CreateFile(dbPath);
                CreateTable();
                SeedData();
                MessageBox.Show("База даних створена успішно!", "Успіх",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            else
            {
                CreateMissingTables();
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show($"Помилка створення БД: {ex.Message}", "Помилка",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

Рисунок 8 – Створення бази даних

ProductService.cs – сервіс пов'язаний з товарами на магазині. Дає можливість редагувати, додавати та видаляти товар з бази даних. Може виконувати фільтрацію товару, тобто знаходження його знаходження по певному критерію. Також завдяки цьому сервісу ми можемо перевірити наявність товару на складі та зберігати всі зміни в базі даних.

Також цей сервіс реалізує пошук товару за його штрих-кодом. На рисунку 9 наведено реалізація цієї можливості в коді застосунку.

```

public static Product GetProductByBarcode(string barcode)
{
    try
    {
        using (var connection = new SQLiteConnection(connectionString))
        {
            connection.Open();
            string query = "SELECT * FROM Products WHERE Barcode = @Barcode AND IsActive = 1";

            using (var command = new SQLiteCommand(query, connection))
            {
                command.Parameters.AddWithValue("@Barcode", barcode);
                using (var reader = command.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        return new Product
                        {
                            Id = Convert.ToInt32(reader["Id"]),
                            Name = reader["Name"]?.ToString() ?? string.Empty,
                            Description = reader["Description"]?.ToString() ?? string.Empty,
                            Price = Convert.ToDecimal(reader["Price"]),
                            Quantity = Convert.ToInt32(reader["Quantity"]),
                            Category = reader["Category"]?.ToString() ?? string.Empty,
                            Brand = reader["Brand"]?.ToString() ?? string.Empty,
                            Model = reader["Model"]?.ToString() ?? string.Empty,
                            Barcode = reader["Barcode"]?.ToString() ?? string.Empty,
                            CreatedDate = DateTime.TryParse(reader["CreatedDate"]?.ToString(), out var date)
                                ? date : DateTime.MinValue,
                            IsActive = Convert.ToInt32(reader["IsActive"]) == 1
                        };
                    }
                }
            }
        }
    }
    catch (Exception ex)
    {
        throw new Exception($"Помилка пошуку товару за штрих-кодом: {ex.Message}");
    }

    return null;
}

```

Рисунок 9 – Реалізація пошуку товару по штрих-коду

CustomerService.cs – сервіс, який обробляє всі запити пов'язані з клієнтами магазину. Дає змогу додавати нових клієнтів, шукати за номером телефону та редагувати данні про клієнта.

SaleService.cs – реалізує в собі можливість продажів. Створює продажі, вираховує загальну суму з урахуванням кількості товару і можливості знижок на магазині. Також оновлює залишки товару після продажу та зберігає в базі даних продажі виконані на торговій точці. Реалізація цієї можливості у вигляді коду наведено на рисунку 10.

```

public static int SaveSale(Sale sale, List<SaleItem> saleItems)
{
    try
    {
        using (var connection = new SQLiteConnection(connectionString))
        {
            connection.Open();

            using (var pragmaCommand = new SQLiteCommand("PRAGMA journal_mode=WAL",
                                                         connection))
            {
                pragmaCommand.ExecuteNonQuery();
            }

            using (var transaction = connection.BeginTransaction())
            {
                try
                {
                    // Реалізація збереження продажу
                    string saleQuery = @"INSERT INTO Sales (CustomerId, SaleDate,
                                                         TotalAmount, Discount, FinalAmount, PaymentMethod, Notes)
                                                         VALUES (@CustomerId, @SaleDate, @TotalAmount,
                                                         @Discount, @FinalAmount, @PaymentMethod, @Notes);
                                                         SELECT last_insert_rowid();";

                    int saleId;
                    using (var command = new SQLiteCommand(saleQuery, connection,
                                                           transaction))
                    {
                        command.Parameters.AddWithValue("@CustomerId",
sale.CustomerId.HasValue ? (object)sale.CustomerId.Value : DBNull.Value);
                        command.Parameters.AddWithValue("@SaleDate",
                                                         sale.SaleDate.ToString("yyyy-MM-dd HH:mm:ss"));
                        command.Parameters.AddWithValue("@TotalAmount",
                                                         sale.TotalAmount);
                        command.Parameters.AddWithValue("@Discount", sale.Discount);
                        command.Parameters.AddWithValue("@FinalAmount",
                                                         sale.FinalAmount);
                        command.Parameters.AddWithValue("@PaymentMethod",
                                                         sale.PaymentMethod ?? "");
                        command.Parameters.AddWithValue("@Notes", sale.Notes ?? "");

                        saleId = Convert.ToInt32(command.ExecuteScalar());
                    }

                    // Реалізація збереження позицій товару
                    // та оновлення залишків товарів
                    foreach (var item in saleItems)
                    {
                        // Зберігаємо товар в продажі
                        string itemQuery = @"INSERT INTO SaleItems (SaleId, ProductId,
                                                                    Quantity, UnitPrice, TotalPrice)
                                                                    VALUES (@SaleId, @ProductId, @Quantity,
                                                                    @UnitPrice, @TotalPrice)";

```

Рисунок 10 – Реалізація продажі у вигляді коду


```

using (var command = new SQLiteCommand(itemQuery, connection,
                                         transaction))
{
    command.Parameters.AddWithValue("@SaleId", saleId);
    command.Parameters.AddWithValue("@ProductId",
                                   item.ProductId);
    command.Parameters.AddWithValue("@Quantity",
                                   item.Quantity);
    command.Parameters.AddWithValue("@UnitPrice",
                                   item.UnitPrice);
    command.Parameters.AddWithValue("@TotalPrice",
                                   item.TotalPrice);
    command.ExecuteNonQuery();
}

// Оновлюємо залишки
string updateQuantityQuery = @"UPDATE Products
                              SET Quantity = Quantity -
                              @SoldQuantity
                              WHERE Id = @ProductId";

using (var updateCommand = new SQLiteCommand(
    updateQuantityQuery, connection, transaction))
{
    updateCommand.Parameters.AddWithValue("@SoldQuantity",
                                         item.Quantity);
    updateCommand.Parameters.AddWithValue("@ProductId",
                                         item.ProductId);
    updateCommand.ExecuteNonQuery();
}

// Підтверджуємо транзакцію
transaction.Commit();
return saleId;
}
}
}
}
}
}
}

```

Продовження рисунку 10

ReportService.cs – сервіс для формування звітності по касі на основі даних, збережених в SaleService.cs. Витягує продажі за певний період та агрегує їх у звітність (підсумкова сума, кількість товарів, тощо.). Після цього готує звітні об'єкти для CashReport.cs.

Реалізація можливості виведення звітності, аналітики по способам оплати та топам товарів у вигляді коду застосунку наведено на рисунку 11.

```

public static CashReport GetCashReport(DateTime fromDate, DateTime toDate)
{
    var report = new CashReport
    {
        FromDate = fromDate,
        ToDate = toDate
    };

    try
    {
        using (var connection = new SQLiteConnection(connectionString))
        {
            connection.Open();

            // Загальна статистика
            string salesQuery = @"
                SELECT
                    COUNT(*) as TotalSales,
                    COALESCE(SUM(TotalAmount), 0) as TotalAmount,
                    COALESCE(SUM(Discount), 0) as TotalDiscount,
                    COALESCE(SUM(FinalAmount), 0) as FinalAmount
                FROM Sales
                WHERE SaleDate >= @FromDate AND SaleDate <= @ToDate";

            using (var command = new SQLiteCommand(salesQuery, connection))
            {
                command.Parameters.AddWithValue("@FromDate", fromDate.ToString("yyyy-MM-dd 00:00:00"));
                command.Parameters.AddWithValue("@ToDate", toDate.ToString("yyyy-MM-dd 23:59:59"));

                using (var reader = command.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        report.TotalSales = Convert.ToInt32(reader["TotalSales"]);
                        report.TotalAmount =
                            Convert.ToDecimal(reader["TotalAmount"]);
                        report.TotalDiscount =
                            Convert.ToDecimal(reader["TotalDiscount"]);
                        report.FinalAmount =
                            Convert.ToDecimal(reader["FinalAmount"]);
                    }
                }
            }

            // Статистика по способам оплаты
            string paymentQuery = @"
                SELECT
                    PaymentMethod,
                    COUNT(*) as Count,
                    COALESCE(SUM(FinalAmount), 0) as Amount
                FROM Sales
                WHERE SaleDate >= @FromDate AND SaleDate <= @ToDate
                GROUP BY PaymentMethod";

```


Продовження рисунку 11

Forms – папка в якій знаходиться інтерфейс який бачить користувач. Форми взаємодіють з користувачем та взаємодіють з сервісами. Вони не містять бізнес логіки та виконують лише візуалізацію та обробку подій.

MainForm.cs – центральна форма. З неї користувач зможе здійснити перехід до інших форм.

ProductsForm.cs – форма яка включає в себе всі товари, які знаходяться на магазині. Має можливість редагування даних про товар та додавання нового. Через прописану в коді подію викликає ProductService.cs що і дозволяє візуально відображати зміни в товарі.

CustomersForm.cs – потрібен для відображення даних про клієнтів: номер телефону, ім'я та прізвище, е-майл та примітки про них. Також дає можливість візуально відобразити редагування та додавання нового клієнта. Викликає CustomerService.cs для внесення змін в базу даних.

CashReportForm.cs – форма для відображення звітності по касі в магазині. Дає можливість виводити суму за певний період часу та відображає топ товарів, днів, та клієнтів по продажу для аналізу роботи торгової точки та статистики. Важливим фактором є довільність вибору періоду. Касу можна вивести за день, тиждень, місяць та певний період що полегшує збір статистики для подальшого аналізу.

По завершенню опису структури розробленого програмного забезпечення наведемо UML-діаграми створенної бази даних і класів з їх описом.

На рисунку 12 наведена реалізована нами база даних яка створена для нашого застосунку. Перш ніж почати її опис, пропоную ознайомитися з самим поняттям що таке ER-діаграма бази даних.

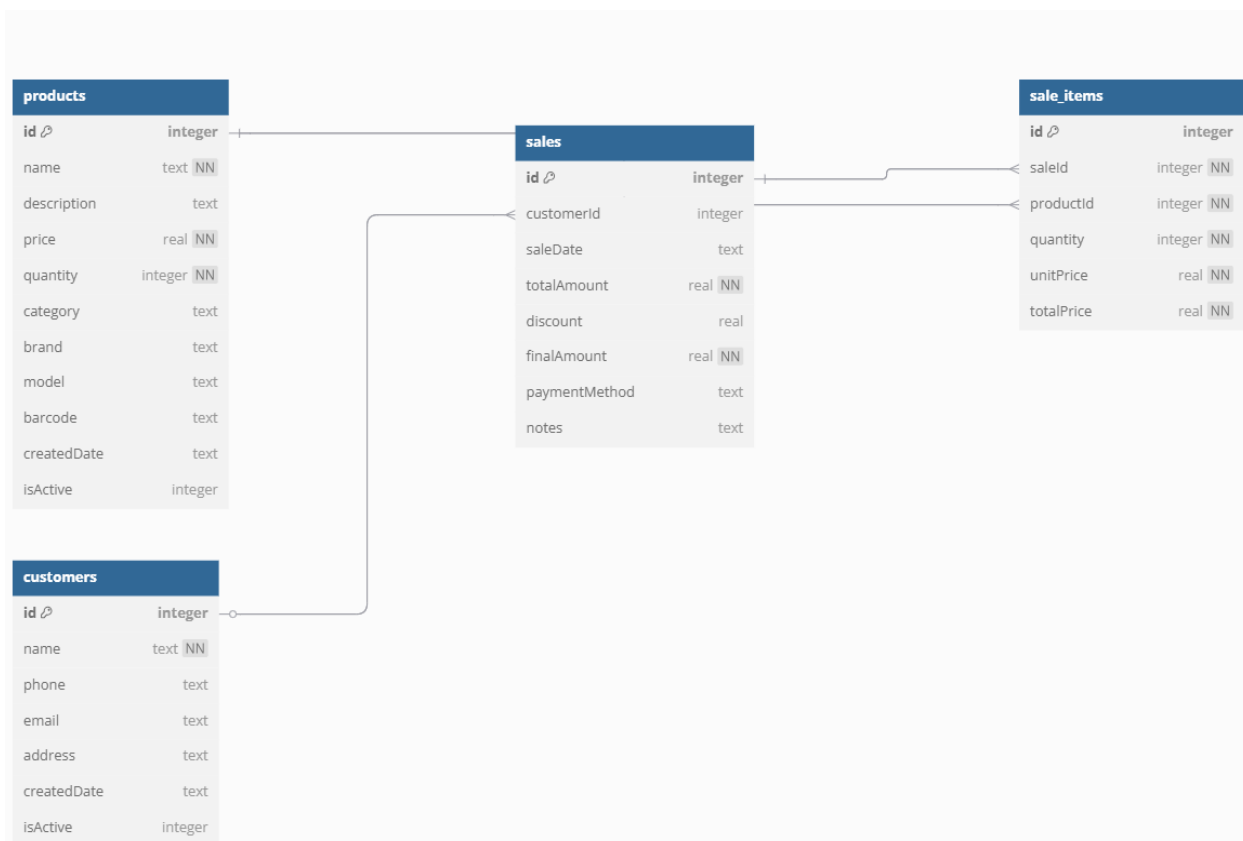


Рисунок 12 – ER-діаграма бази даних

Діаграми «сутність-зв'язок» (ER) є ключовим інструментом у проектуванні реляційних баз даних, оскільки вони дозволяють легко відображати структури даних, зв'язки між сутностями та правила взаємодії між ними. Ця інформаційна система базується на простій, але ефективній моделі ER, яка зосереджена на автоматизованих транзакціях у магазині побутової техніки. Модель визначає чотири основні сутності: Продукт, Клієнт, Продажі та Товар продажу, кожна з яких відповідає аспекту бізнес-логіки системи.

Сутність «Продукт» містить детальну інформацію про кожен продукт, включаючи такі атрибути, як назва, опис, ціна, кількість, категорія, бренд, модель, штрих-код, дата створення та статус активності. Ці поля дозволяють ефективно відстежувати продукти та реалізовувати основні функції інвентаризації та фільтрації [33]. Для

оптимізації пошуку продуктів система реалізує індекс штрих-кодів, який пришвидшує доступ до певних записів [34].

Сутність «Клієнт» зберігає дані про клієнта: прізвище, телефон, електронну пошту, адресу, дату створення та статус активності. Номер телефону дозволяє створити унікальний індекс для швидкої ідентифікації клієнта під час процесу продажу, тим самим скорочуючи час обслуговування [35]. Це також є необхідною умовою для впровадження майбутніх функцій CRM [36].

Сутність «Продажі» використовується для реалізації облікової моделі для кожної транзакції продажу. Вона містить цінову пропозицію для клієнта (представлену CustomerId), дату продажу, загальну суму, знижки, кінцеву суму, спосіб оплати та іншу інформацію. Ця таблиця є центральною для бізнес-логіки програми, оскільки вона надає історію та аналіз усіх транзакцій [37, 38].

Сутність «SaleItems» діє як проміжна таблиця у зв'язку між Sales та Products. Вона записує, які товари були придбані в певному продажу, в якій кількості та за якою ціною. Це важливо для звітності та аналізу попиту [39].

Зв'язки між сутностями базуються на зовнішніх ефектах. Зокрема, Sales.CustomerId та Customers.Id пов'язані, тоді як SaleItems.SaleId та SaleItems.ProductId пов'язані з Sales.Id та Products.Id відповідно. Такий підхід до моделювання забезпечує цілісність даних (цілісність цінової пропозиції), автоматично запобігаючи видаленню записів, що містять пов'язані елементи [40].

Дизайн цієї ER-діаграми відповідає принципу нормалізації бази даних, що дозволяє уникнути надмірного дублювання даних та забезпечує узгодженість записів [41]. Структура містить усі атрибути, необхідні для виконання типових операцій роздрібної торгівлі. Структура є гнучкою та розширюваною – її можна розширити, додавши інші сутності за потреби, наприклад, для зберігання інформації про

постачальників, логістику або програми винагород [42].

Варто зазначити, що модель повністю сумісна з технологіями .NET 6 та ADO.NET, які були обрані як частина технологічного стеку, що використовується для реалізації клієнта та бази даних. Чітка структура бази даних, відображена на ER-діаграмі, дозволяє ефективно виконувати CRUD-операції, звітність та агрегацію даних на сервісному рівні програми [43, 44].

На рисунку 13 наведена UML-діаграма класів нашого проєкту. По аналогічному з діаграмою бази даних наведемо коротке визначення та повноцінний її опис.

Діаграма класів UML у структурі програмного забезпечення для обліку товарів та продажів є основним інструментом для опису об'єктів (класів), їх атрибутів, методів та зв'язків між ними. Він є частиною стандарту Уніфікованої мови моделювання та використовується як засіб візуалізації проєктування об'єктно-орієнтованих систем [25].

Модель даних (модельний шар) представляє класи, що відображають бізнес-сутності системи. Клас `Product` містить інформацію про продукт, включаючи назву, опис, ціну, кількість, категорію, бренд, модель, штрих-код, дату створення та дію. Також є обчислювані властивості, такі як `DisplayName` та `InStock`, які забезпечують зручне представлення в інтерфейсі. Клас `Customer` зберігає дані про клієнта, включаючи ім'я, номер телефону, електронну пошту, адресу та дату створення. «М'яке видалення» тут досягається за допомогою поля `IsActive`. Клас `Sale` відповідає за зберігання інформації про кожен продаж: дата, клієнт, сума до знижки, сума знижки, кінцева сума, спосіб оплати, примітки та колекція об'єктів типу `SaleItem`.

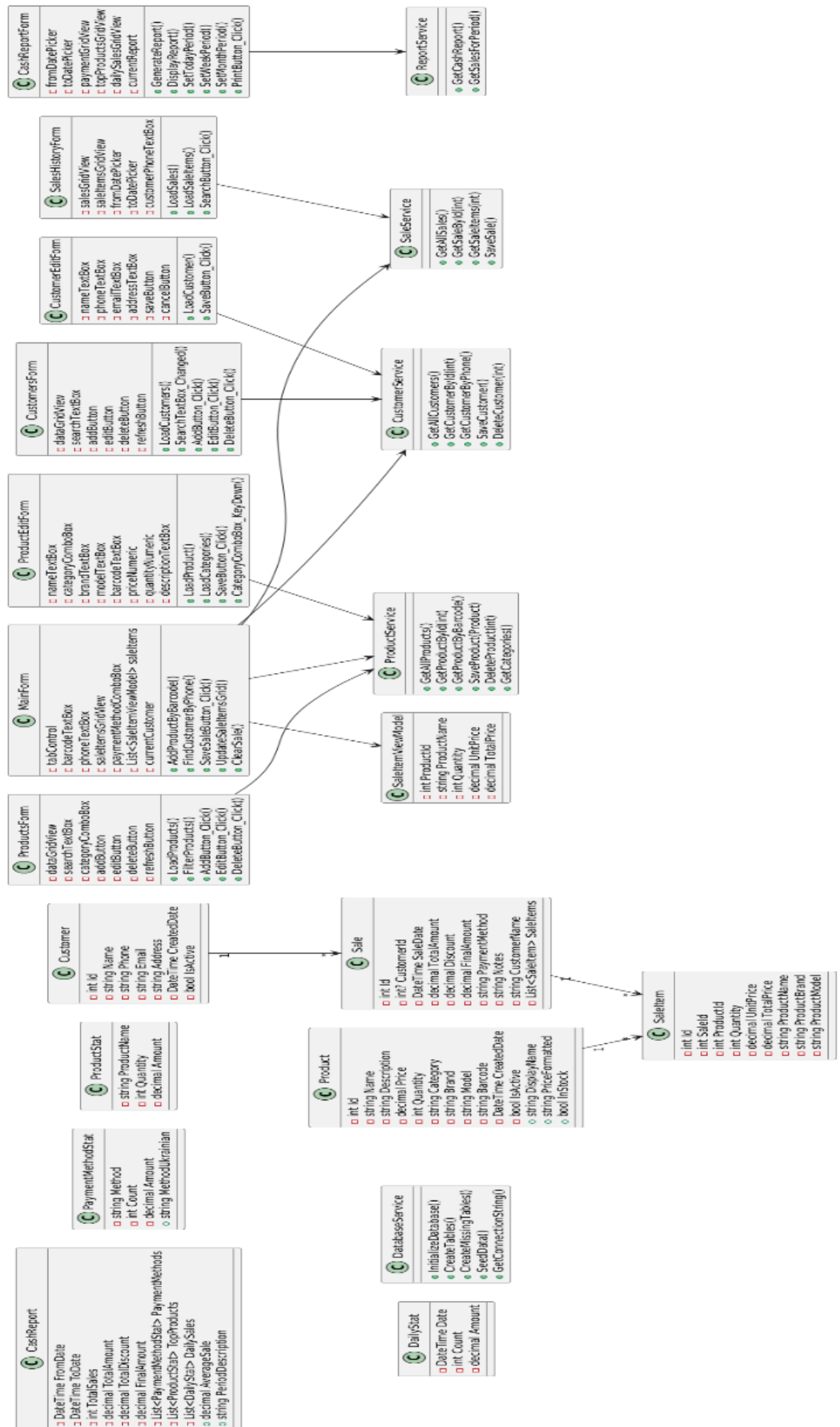


Рисунок 13 – UML-діаграма класів

Ці об'єкти зберігаються в окремому класі `SaleItem`, який записує кількість товару, його ціну під час розпродажу та розраховану загальну вартість. Для зручності перегляду в інтерфейсі використовується окремий клас `SaleItemViewModel`. Класи `CashReport`, `PaymentMethodStat`, `ProductStat` та `DailyStat` використовуються для створення звітів, які поєднують аналітичну інформацію, таку як обсяг продажів, частка кожного способу оплати, найпопулярніші продукти та динаміка продажів з плином часу [26].

На рівні бізнес-логіки (рівні сервісів) реалізовані сервіси, що забезпечують роботу для кожної моделі. `DatabaseService` відповідає за ініціалізацію бази даних, створення таблиць та підключення до SQLite. Це низькорівневий компонент, через який працюють інші служби. `ProductService`, `CustomerService`, `SaleService` та `ReportService` інкапсулюють логіку для отримання, збереження, редагування та видалення об'єктів. Вони реалізують принцип розподілу обов'язків та значно спрощують обслуговування та розширення програми [27].

Рівень презентації містить форми `WinForms`, кожна з яких орієнтована на використання певної функціональності. `MainForm` – це центральний елемент інтерфейсу, де користувач продає товари, шукає штрих-коди або номери телефонів клієнтів і зберігає результати як продажі. Форми `ProductsForm` та `CustomersForm` відповідають за перегляд та керування відповідними сутностями. Редагування даних здійснюється за допомогою `ProductEditForm` та `CustomerEditForm`, а аналіз результатів продажів – за допомогою `SalesHistoryForm` та `CashReportForm`, які використовують аналітичні сервіси для створення статистичних звітів.

Зв'язки між класами чітко структуровані. Клас `Sale` має зв'язок 1:N з класом `SaleItem`, що означає, що розпродаж може містити кілька товарів. Кожен елемент розпродажу пов'язаний з певним продуктом і

належить до певного розпродажу. Між клієнтами та відділом продажів існує зв'язок 1:N. Форми використовують сервіси для взаємодії з даними, і всі сервіси покладаються на DatabaseService як центральну точку доступу до даних.

Загалом, діаграми класів UML дозволяють візуально зрозуміти структуру системи, зрозуміти її архітектуру, а також підтримувати та розширювати систему в майбутньому. Поділ логічних рівнів забезпечує чистоту коду, гнучкість реалізації, можливість тестувати компоненти та зручність розширення функціональності. Такий підхід відповідає сучасним стандартам розробки програмного забезпечення та рекомендується як для освітніх, так і для виробничих проєктів [25], [28].

3.3 Графічний інтерфейс застосунку

Після повноцінного опису «бекенд» сторони розробленого програмного забезпечення можна перейти до перегляду його графічної частини – інтерфейсу.

Насамперед він створювався для абсолютного розуміння будь-якими користувачами, як новачками так і досвідченим користувачам. Таким чином було прийнято рішення створити його з доданням кольорів на активних елементах (кнопках) та візуальних маркерів вкладок, наприклад мішечок з монетами на вкладці продажів. Це дозволить спростити процес навчання нових продавців та просто візуально допомагати скоріше орієнтуватися в програмі.

Далі буде представлений опис елементів інтерфейсу з наданням рисунків для візуального збагачення кваліфікаційної роботи.

Перше що буде розглянуто – це вкладка «Продажі». Основна частина де і будуть відбуватися усі продажі на торговій точці. Її зовнішній вигляд представлено на рисунку 14.

Рисунок 14 – Графічний інтерфейс вкладки «Продаж»

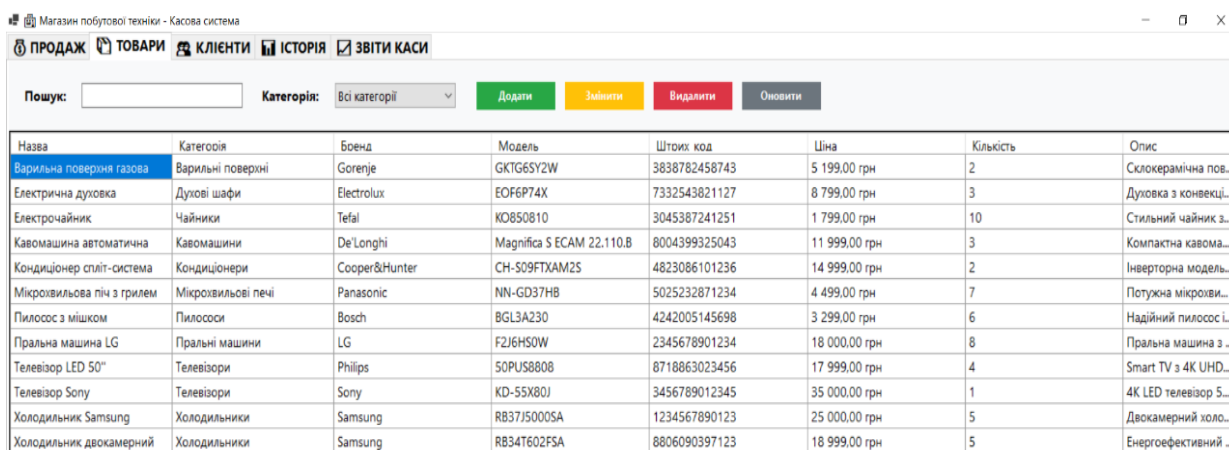
Як можна побачити тут реалізована можливість додавати товар за штрих-кодом, який наприклад просканував продавець з упаковки товару. Сам товар буде додано в поле «Товар в чеку» з повним його найменуванням.

Колонка «Оплата» надає змогу візуально побачити остаточну суму з урахуванням всього що було додано в минуле поле. Також, додатково можна виставити відсоток знижки і система автоматично її підрахує. Варто не забувати що для повноцінного обліку продажів потрібно знати яким чином товар був оплачений. Для цього є можливість у випадяючому вікні обрати спосіб оплати: карта, готівка або переказ коштів.

Поле телефон бере на себе функцію обрання клієнта магазину. Наприклад, постійний клієнт просить надати йому знижку. Продавець-консультант може перевірити: чи зареєстрований цей клієнт в базі по номеру? Якщо так, то можна надати певну знижку.

Після оформлення продажі потрібно завершити продаж. За це відповідає кнопка «Провести» яка підтвердить та закриє її і збереже у вкладку «Історія».

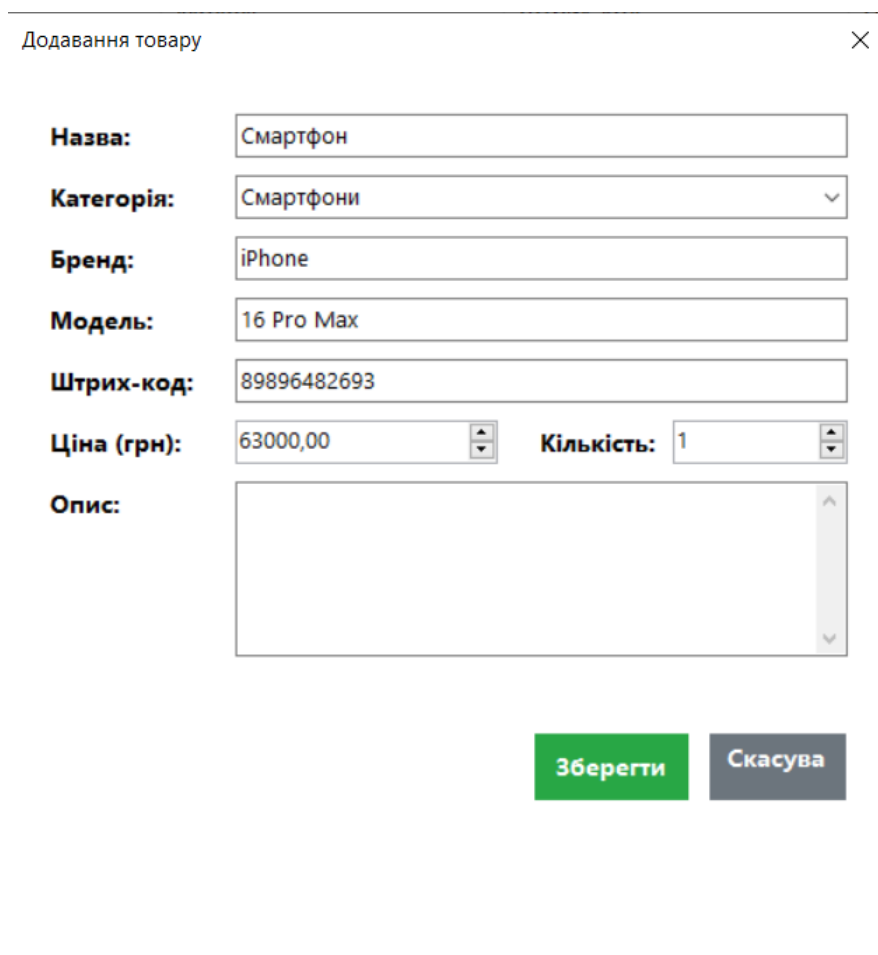
Наступним що потрібно розглянути – це вкладка «Товари», інтерфейс якої наведений на рисунку 15.



Назва	Категорія	Бренд	Модель	Штрих код	Ціна	Кількість	Опис
Варильна поверхня газова	Варильні поверхні	Gorenje	GKT6SY2W	3838782458743	5 199,00 грн	2	Склокерамічна пов...
Електрична духовка	Духові шафи	Electrolux	EOF6P74X	7332543821127	8 799,00 грн	3	Духовка з конвекці...
Електрочайник	Чайники	Tefal	KO850810	3045387241251	1 799,00 грн	10	Стильний чайник з...
Кавомашинка автоматична	Кавомашини	De'Longhi	Magnifica S ECAM 22.110.B	8004399325043	11 999,00 грн	3	Компактна кавома...
Кондиціонер спліт-система	Кондиціонери	Cooper&Hunter	CH-S09FTXAM2S	4823086101236	14 999,00 грн	2	Інверторна модель...
Мікрохвильова піч з грилем	Мікрохвильові печі	Panasonic	NN-GD37HB	5025232871234	4 499,00 грн	7	Потужна мікрохви...
Пилосос з мішком	Пилосои	Bosch	BGL3A230	4242005145698	3 299,00 грн	6	Надійний пилосос...
Пральна машина LG	Пральні машини	LG	F2J6HSOW	2345678901234	18 000,00 грн	8	Пральна машина з ...
Телевізор LED 50"	Телевізори	Philips	50PUS8808	8718863023456	17 999,00 грн	4	Smart TV з 4K UHD...
Телевізор Sony	Телевізори	Sony	KD-55X80J	3456789012345	35 000,00 грн	1	4K LED телевізор 5...
Холодильник Samsung	Холодильники	Samsung	RB37J5000SA	1234567890123	25 000,00 грн	5	Двокамерний холо...
Холодильник двокамерний	Холодильники	Samsung	RB34T602FSA	8806090397123	18 999,00 грн	5	Енергоефективний ...

Рисунок 15 – Графічний інтерфейс вкладки «Товари»

Вона представляє собою список із всіх товарів, які знаходяться на торговій точці. Він поділений на колонки: назва, категорія, бренд, модель, штрих код, ціна, кількість та опис товару. Реалізована можливість виводу по категоріям та пошук самого товару. Також є можливість його додання, редагування чи видалення. За це відповідають активні кнопки. При натисканні на кнопку «Додати» перед користувачем з'явиться вікно в якому він має змогу ввести всі дані про товар і зберегти їх. Приклад цього вікна наведено на рисунку 16.



The screenshot shows a window titled "Додавання товару" (Add Item) with a close button (X) in the top right corner. The form contains the following fields and controls:

- Назва:** Text input field containing "Смартфон".
- Категорія:** Dropdown menu showing "Смартфони".
- Бренд:** Text input field containing "iPhone".
- Модель:** Text input field containing "16 Pro Max".
- Штрих-код:** Text input field containing "89896482693".
- Ціна (грн):** Text input field containing "63000,00" with up/down arrow controls.
- Кількість:** Text input field containing "1" with up/down arrow controls.
- Опис:** Large text area for description, currently empty.

At the bottom right of the form are two buttons: a green "Зберегти" (Save) button and a grey "Скасува" (Cancel) button.

Рисунок 16 – Вікно «Додати товар»

Тепер перейдемо до вкладки «Клієнти». Тут представлена база даних у вигляді списку з переліком усіх зареєстрованих клієнтів магазину. Він складається з стовпців: Ім'я (ПІБ клієнта), номер телефону, email, адреса та дата його реєстрації. Цю вкладку можна повноцінно використовувати для таргет-розсилок з акціями та пропозиціями магазину. Графічна візуалізація представлена на рисунку 17.

Магазин побутової техніки - Касова система

ПРОДАЖ ТОВАРИ КЛІЄНТИ ІСТОРІЯ ЗВІТИ КАСИ

Пошук клієнтів: Додати Змінити Видалити Оновити

Ім'я	Телефон	Email	Адреса	Дата реєстрації
Іван Петров	+380501234567	ivan@email.com	вул. Леніна, 1	24.06.2025
Іваненко Олег Сергійович	+380673451278	o.ivanenko@example.com	м. Київ, вул. Хрещатик, 22	27.06.2025
Бондар Тетяна Валеріївна	+380932457812	t.bondar@example.com	м. Запоріжжя, вул. Перемоги, 3...	27.06.2025
Гончаренко Ірина Іванівна	+380662233445	i.goncharenko@example.com	м. Ужгород, вул. Капушанська, ...	27.06.2025
Дяченко Вікторія Андріївна	+380731112233	v.dyachenko@example.com	м. Івано-Франківськ, вул. Неза...	27.06.2025
Зінченко Денис Євгенович	+380993456789	d.zinchenko@example.com	м. Хмельницький, вул. Проскурі...	27.06.2025
Коваль Андрій Ігорович	+380661112233	a.koval@example.com	м. Львів, вул. Зеленська, 10	27.06.2025
Кравець Наталія Богданівна	+380633345567	n.kravets@example.com	м. Полтава, вул. Шевченка, 89	27.06.2025
Левченко Артем Валентинович	+380981111222	a.levchenko@example.com	м. Рівне, вул. Соборна, 56	27.06.2025
Литвин Анна Юріївна	+380503212345	a.lytvyn@example.com	м. Черкаси, вул. Чорновола, 6	27.06.2025
Мазур Оксана Михайлівна	+380681234345	o.mazur@example.com	м. Миколаїв, вул. Адміральська...	27.06.2025
Марія Сидорова	+380502345678	maria@email.com	вул. Пушкіна, 25	24.06.2025
Мельник Сергій Ігорович	+380674567123	s.melnik@example.com	м. Вінниця, вул. Келецька, 17	27.06.2025
Нефедченко Олег Олександрович	+380686441840			24.06.2025
Олійник Аліна Олексіївна	+380937451236	a.oliyk@example.com	м. Кривий Ріг, вул. Космонавт...	27.06.2025
Панченко Єлизавета Вадимівна	+380631987654	e.panchenko@example.com	м. Тернопіль, вул. Гоголя, 3	27.06.2025
Петренко Владислав Миколайович	+380671234901	v.petrenko@example.com	м. Житомир, вул. Київська, 42	27.06.2025
Романюк Дмитро Олександрович	+380502345678	d.romanyuk@example.com	м. Одеса, вул. Французький бул...	27.06.2025
Савченко Ігор Віталійович	+380935554433	i.savchenko@example.com	м. Суми, вул. Харківська, 21	27.06.2025
Соловей Микита Анатолійович	+380951234556	m.solovey@example.com	м. Луцьк, просп. Волі, 23	27.06.2025
Ткачук Ольга Василівна	+380991234890	o.tkachenko@example.com	м. Дніпро, вул. Грушевського, ...	27.06.2025
Чорний Павло Степанович	+380681234666	p.chornyi@example.com	м. Чернівці, вул. Руська, 88	27.06.2025
Шевченко Марина Петрівна	+380631234567	m.shevchenko@example.com	м. Харків, вул. Академіка Павл...	27.06.2025

Рисунок 17 – Вкладка «Клієнти»

Наступним розглянемо вкладку «Історія». Вона тісно пов'язана з «Продаж» бо зберігає в собі всі продажі, які були проведені на торговій точці. Представляє собою список та має наступні стовпці: id або номер продажі, клієнт, дата, сума, знижка, до сплати (загальна сума), спосіб оплати та примітки. Нижче, після вибору продажі за списку, можна побачити що було продано. Також реалізована можливість довільного вибору періоду продаж та пошук по клієнту. Графічна реалізація представлена на рисунку 18.

Магазин побутової техніки - Касова система

ПРОДАЖТОВАРИКЛІЄНТИІСТОРІЯЗВІТИ КАСИ

С: 27 мая 2021По: 27 июня 2021Телефон клієнта: +380501234567ПошукОновити

Id	Клієнт	Дата	Сума	Знижка	До сплати	Спосіб оплати	Примітки
4		26.06.2025 21:42	35 000,00 грн	0,00 грн	35 000,00 грн	Готівка	
3	Нефедченко Олег Оле...	25.06.2025 18:51	18 000,00 грн	360,00 грн	17 640,00 грн	Готівка	
2	Марія Сидорова	24.06.2025 19:15	18 000,00 грн	0,00 грн	18 000,00 грн	Готівка	
1		24.06.2025 19:13	18 000,00 грн	0,00 грн	18 000,00 грн	Готівка	

Товари в продажі:

Товар	Бренд	Модель	Кількість	Ціна за одиницю	Сума
Пральна машина LG	LG	F2J6HS0W	1	18 000,00 грн	18 000,00 грн

Всього продажів: 4Загальна сума: 88 640,00

Рисунок 18 – Вкладка «Історія»

Остання вкладка представляє собою звітність по касі або «Звіти каси». Представлена на рисунку 19.

Магазин побутової техніки - Касова система

ПРОДАЖТОВАРИКЛІЄНТИІСТОРІЯЗВІТИ КАСИ

ЗВІТ КАСИ ЗА ПЕРІОД

Оберіть 23 июня 2021Д 27 июня 2021СьогодніТижденьМісяцьДовільнийВВ ОновитиДрук

ЗАГАЛЬНА СТАТИСТИКА

Кількість продажів: 4Загальна сума: 88 640,00 грнСередній чек: 22 160,00 грнЗнижки: 360,00 грн

Способи оплати

Спосіб	Кількість	Сума	Відсоток
Готівка	4	88 640,00 грн	100,0%

Топ товарів

Товар	Кількість	Сума
LG F2J6HS0W - П...	3	54 000,00 грн
Sony KD-55XB0J -...	1	35 000,00 грн

Продажі по днях

Дата	Поодажів	Сума
24.06.2025	2	36 000,00 грн
25.06.2025	1	17 640,00 грн
26.06.2025	1	35 000,00 грн

Рисунок 19 – Вкладка «Звіти каси»

Тут користувач зможе побачити детальну звітність по касах за певний період. Також додатково реалізовані інші показники, такі як: перелік способів оплати за кількістю, топ товарів та продажі по дням. Вони дозволять зібрати статистичні дані, що буде доречно для аналізу роботи магазину та подальшого його розвитку.

РОЗДІЛ 4. ЕКОНОМІЧНА ДОЦІЛЬНІСТЬ РОЗРОБКИ ПРОГРАМНОГО КОМПЛЕКСУ ДЛЯ ІНФОРМАЦІЙНОЇ ПІДТРИМКИ РОБОТИ ПРОДАВЦІВ-КОНСУЛЬТАНТІВ В МАГАЗИНАХ ПОБУТОВОЇ ТЕХНІКИ

4.1. Вартісний аналіз

4.1.1 Розрахунок повної собівартості програмного продукту

Фактична або повна собівартість ПП визначається в процесі проведення калькуляції собівартості та є сумою виробничої собівартості, адміністративних витрат та витрат на збут.

Виробнича собівартість розраховується за допомогою визначення поточних витрат на його розробку (або функціонально–необхідних витрат на створення ПП – $C_{\text{вир.}}$) і визначається за формулою:

$$C_{\text{вир}} = C_{\text{ЗПр}} + C_{\text{ЕСВ}} + C_{\text{М.ч.}} + C_{\text{З-в.Н.}} + C_{\text{М}}, \quad (4.1)$$

де $C_{\text{ЗПр}}$ – заробітна плата розробників ПП, грн;

$C_{\text{ЕСВ}}$ – відрахування на соціальне страхування, грн;

$C_{\text{М.ч.}}$ – вартість машинного часу, необхідного для розробки та налаштування ПП, грн;

$C_{\text{З-в.Н.}}$ – загальновиробничі (накладні) витрати (витрати на оплату праці управлінського персоналу, оплату службових відряджень, консультаційно–інформаційні витрати, ремонт і технічне обслуговування інших основних фондів, окрім ПК, оренда приміщення тощо);

$C_{\text{М}}$ – вартість матеріалів, комплектуючих, грн.

Отже, розрахуємо повну собівартості програмного продукту «Кольоровий квест» для з'ясування раціональності розробки.

4.1.2 Калькуляція собівартості програмного продукту

До заробітної плати розробників ПП ($C_{зПр}$) належать витрати на виплату основної та додаткової зарплати виконавців, обчислені згідно із системою оплати праці, прийнятими в організації, включаючи будь-які види матеріальних та грошових доплат. Визначається за формулою:

$$C_{зПр} = C_{зПосн.} + C_{зПдод.} \quad (4.2)$$

Основна заробітна плата розробників ПП:

$$C_{зПосн.} = C_{зПден.} \times T_{заг.} , \quad (4.3)$$

де $C_{зПден.}$ – денна зарплата програміста, грн;

$T_{заг.}$ – загальна трудомісткість розробки ПП (комп'ютерної системи), людино-дні.

Денну заробітну плату визначають, виходячи з місячних окладів:

$$C_{зПден.} = \frac{C_{ок.} \times 12}{\Phi_{р.ч.}} , \quad (4.4)$$

де $C_{ок.}$ – місячний оклад розробника ПП, грн;

$\Phi_{р.ч.}$ – річний фонд робочого часу, днів; (трудовим законодавством встановлено на 2024 р.: $\Phi_{р.ч.} = 252$).

По Україні в середньому місячний оклад розробника (С#) з меншим ніж рік досвідом складає 500\$. На травень 2025 року це приблизно 21000 грн.

Розрахуємо денну заробітну плату:

$$C_{3Пден.} = \frac{21000 * 12}{252} = \frac{252000}{252} = 1000 \text{ грн}$$

Отримали розмір денної заробітної плати у розмірі 961,8 грн.

Загальну трудомісткість можна визначити за формулою:

$$T_{заг.} = N_{час} \times k_{скл.} \times k_{м.} \times k_{станд.} \times k_{станд.ПП}, \quad (4.5)$$

де $T_{заг.}$ – загальна трудомісткість, людино-дні;

$N_{час}$ – норма часу, людино-дні (у середньому: 61 – 83 людино-дні);

$k_{скл.}$ – поправковий коефіцієнт складності контролю вхідної та вихідної інформації (складає 1,08);

$k_{м.}$ – поправковий коефіцієнт використання мови певного рівня складності (мова високого рівня дорівнює 1);

$k_{станд.}$ – коефіцієнт використання стандартних програм (приблизно складає 0,7);

$k_{станд.ПП}$ – коефіцієнт розробки стандартного ПП (1,2 – 1,6).

Розробляючи ПП, використовуються стандартні модулі і/або пакети прикладних програм, чи типові програми, тому норму часу коригують за допомогою коефіцієнта $k_{станд.} = 0.6 - 0.8$.

Норму часу візьмемо 70 людино-днів, коефіцієнт розробки стандартного ПП – 1.2 та проведемо розрахунки:

$$T_{заг.} = 70 * 1,08 * 1 * 0,7 * 1,2 = 63,5 \text{ л/д}$$

Після отримання загальної трудомісткості розробки та денної заробітної плати можемо розрахувати основну заробітну плату за формулою 5.7:

$$C_{ЗПосн.} = 1000 * 63,5 = 63500 \text{ грн}$$

Додаткова заробітна плата (премії, одноразові заохочення тощо) розраховується згідно з нормативом, який установлює підприємство і який складає 10 – 40 % від основної зарплати. Витрати на додаткову заробітну плату визначаються за формулою:

$$C_{ЗПдод.} = k_{ЗПдод.} \times C_{ЗПосн.}, \quad (4.6)$$

де $k_{ЗПдод.}$ – нормативний коефіцієнт додаткової заробітної плати ($k_{ЗПдод.} = 0.1 \dots 0.4$);

$C_{ЗПосн.}$ – витрати на основну заробітну плату, грн.

Розрахуємо додаткову заробітну плату при нормативному коефіцієнті 20%:

$$C_{ЗПдод.} = 0,2 * 63500 = 12700 \text{ грн}$$

До витрат на сплату єдиного соціального внеску ($C_{ЄСВ}$) належать витрати, що здійснюються у порядку та розмірах, передбачених чинним законодавством України (тобто це нарахування від суми основної та додаткової зарплати, які беруться з підприємства; згідно з нормативами, діючими на 01.01.2012 р., і які складають 36,76%, на 2025 рік цей внесок становить 22%).

Витрати на сплату єдиного соціального внеску визначаються за формулою:

$$C_{\text{ЄСВ}} = k_{\text{ЄСВ}} \times (C_{\text{ЗПосн.}} + C_{\text{ЗПдод.}}), \quad (4.7)$$

де $k_{\text{ЄСВ}}$ – коефіцієнт витрат на сплату ЄСВ ($k_{\text{ЄСВ}} = 0.22$);

$$C_{\text{ЄСВ}} = 0,22 * (63500 + 12700) = 16764 \text{ грн}$$

Вартість машинного часу, необхідного для розробки та налаштування ПП ($C_{\text{М.ч.}}$) визначається:

$$C_{\text{М.ч.}} = C_{\text{М.-Г.}} \times t_{\text{М.}}, \quad (4.8)$$

де $t_{\text{М.}}$ – тривалість машинного часу (сума часу машинних і машинно–ручних операцій), необхідного для розробки ПП, грн.

$$t_{\text{М.}} = P_{\text{комп.}} \times 0,15 + T_{\text{заг.}} \times C_{\text{електроен}} \quad (4.9)$$

$$t_{\text{М.}} = 22000 * 0,15 + 63,5 * 4,32 = 3300 + 274,3 = 3574,3 \text{ грн}$$

Собівартість однієї машино–години роботи ПК:

$$C_{\text{М.-Г.}} = \frac{C_{\text{р.екс.}}}{T_{\text{еф.}}}, \quad (4.10)$$

де $C_{\text{р.екс.}}$ – річні експлуатаційні поточні витрати на обслуговування ПК (грн), які охоплюють:

- основну та додаткову ЗП спеціаліста (інженера–електронника), який обслуговує машину з урахуванням його зайнятості на обслуговування ПК, грн;
- витрати на сплату ЄСВ ($C_{\text{ЄСВ}} = 36,76 \%$ від фонду оплати праці, 22% на 2025 рік), грн;
- амортизаційні відрахування (A_m), які розраховуються із залишкової вартості ПК і норми амортизаційних відрахувань (згідно з нормами складає 15 % від балансової вартості ПК, $a=0.15$), грн;
- витрати на ремонт та профілактику ПК ($C_{\text{рем.}}$), грн;
- витрати на оплату електроенергії, які визначаються як добуток тарифу за 1 кВт–г електроенергії на обсяг потужності, що споживається ПК і на ефективний годинний фонд часу роботи ПК за рік (тариф за 1 кВт–г електроенергії в березні 2025 р.: 4.32 грн), грн.

Основна ЗП інженера–електронника:

$$C_{\text{ЗПосн.ел}} = (C_{\text{ок.}} * n_i) / T_{\text{заг.}}, \quad (4.11)$$

де $C_{\text{ок.}}$ – місячний оклад інженера–електронника, грн;

n_i – кількість одиниць ПК, що використовуються для розробки ПП.

$$C_{\text{ЗПосн.ел}} = \frac{13500}{63,5} = 212,6$$

$$C_{\text{ЗПдод.ел}} = 0,2 * 212,6 = 42,5$$

$$C_{\text{ЄСВ ел.}} = 0,22 * (212,6 + 42,5) = 56,1$$

$$A_m = 22000 * 0,15 = 3300 \text{ грн}$$

Витрати на ремонт та профілактику:

$$C_{\text{рем.}} = (k_{\text{рем.}} \times P_{\text{ком.}}) \times n_i, \quad (4.12)$$

де $k_{\text{рем.}}$ – коефіцієнт поточного ремонту та обслуговування ПК (залежить від $a_{\text{рем.}}$);

$a_{\text{рем.}}$ – середньостатистичний норматив витрат на поточний ремонт і обслуговування ПК ($a_{\text{рем.}} = 4\%$);

$$k_{\text{рем.}} = 0.04;$$

$P_{\text{ком.}}$ – ціна ПК, грн.

Тоді:

$$C_{\text{рем.}} = (0,04 \times 22000) \times 1 = 880 \text{ грн}$$

Ефективний годинний фонд часу роботи ПК за рік $T_{\text{еф.}}$ визначається, виходячи з календарного річного фонду часу, зменшеного з урахуванням вихідних, святкових днів і добового режиму роботи ПК з урахуванням поточного ремонту, годин:

$$T_{\text{еф.}} = T_{\text{рік.}} \times (t_{\text{ном.}} - t_{\text{рем.}}), \quad (4.13)$$

де $T_{\text{рік.}}$ – кількість робочих днів за рік,

$t_{\text{ном.}}$ – номінальна кількість годин цілодобової роботи обладнання, ($t_{\text{ном.}} = 8 \text{ год.}$);

$t_{\text{рем.}}$ – число годин на добу для поточного ремонту та обслуговування ПК:

$$t_{\text{рем.}} = 0,15 \times t_{\text{ном.}} = 1.2 \text{ год.}$$

$$T_{\text{еф.}} = 252 \times (8 - 1.2) = 1713,6$$

$$C_{\text{електр.}} = 4,32 * 0,5 * 1713,6 = 3701,4 \text{ грн.}$$

$$C_{\text{м.-г.}} = \frac{212,6 + 42,5 + 56,1 + 3300 + 880 + 3701,4}{1713,6} = 4,8 \text{ грн}$$

$$C_{\text{м.ч.}} = C_{\text{м.-г.}} \times t_{\text{м.}} = 4,8 * 3574,3 = 17156,6 \text{ грн}$$

Експлуатаційні витрати поділяють на виробничі й адміністративні витрати.

Загальновиробничі (накладні) витрати ($C_{\text{з-в.н.}}$): диски, картриджі, папір для роздрукування тощо.

Ураховуючи, що собівартість ПП визначається в умовах обмеженої інформації щодо технології розробки, до загальновиробничих витрат можуть належати також витрати на:

- освоєння нової розробки;
- відшкодування зносу спеціальних інструментів і пристроїв цільового призначення;
- утримання та експлуатацію устаткування.

При такому комплексному складі загальновиробничих витрат їх норматив досягає 50 – 150 % від основної З/П. Загальновиробничі (накладні) витрати визначаються за формулою:

$$C_{\text{з-в.н.}} = k_{\text{з-в.}} \times C_{\text{ЗПосн.}}, \quad (4.14)$$

де $k_{\text{з-в.}}$ – коефіцієнт загальновиробничих (накладних) витрат ($k_{\text{з-в.}} = 0,5 \div 1,5$). Обрано $k_{\text{з-в.}} = 0,5$:

$$C_{\text{з-в.н.}} = 0,5 * 63500 = 31750 \text{ грн}$$

Вартість витратних матеріалів, комплектуючих ($C_{M.}$) рекомендовано взяти в розмірі 4 % від фонду заробітної плати ($k_{M.} = 0,04$).

$$C_{M.} = 0,04 * 63500 = 2540 \text{ грн}$$

Окрім вказаних поточних витрат на розробку ПП, собівартість розробки та реалізації ПП передбачає визначення:

– адміністративних витрат (організаційні витрати, витрати на службові відрядження, страхування, амортизацію, опалення, освітлення, водопостачання, охорону; винагорода за професійні послуги: юридичні, аудиторські; витрати на зв'язок; витрати за послуги банку);

– витрат на збут (на рекламу та дослідження ринку: маркетинг; витрати на гарантійний ремонт і гарантійне сервісне обслуговування; комісійні витрати; витрати, пов'язані з безпосереднім постачанням: страхування, амортизація, охорона);

– повної (фактичної) собівартості (у грошовому виразі індивідуальні витрати певного розробника ПП у даних умовах).

Повна собівартість формується в процесі щоденного оперативно-технічного та бухгалтерського обліку витрат на розробку ПП, виконання робіт і забезпечення всіма матеріально-технічними, трудовими ресурсами.

Адміністративні витрати. Ці витрати належать до собівартості ПП пропорційно основній З/П і складають 100 – 200 % від основної З/П. Адміністративні витрати визначаються за формулою:

$$C_{\text{адмін.}} = k_{\text{адмін.}} \times C_{\text{ЗПосн.}}, \quad (4.15)$$

де $k_{\text{адмін.}}$ – коефіцієнт адміністративних витрат ($k_{\text{адмін.}} = 1 \div 2$).

Обрано $k_{\text{адмін.}} = 1$:

$$C_{\text{адмін.}} = 1 * 63500 = 63500 \text{ грн}$$

Витрати на збут. Витрати за цією статтею складають 2,5 – 5 % від виробничої собівартості. Витрати на збут визначаються за формулою:

$$C_{\text{збут.}} = k_{\text{збут.}} \times C_{\text{вир.}}, \quad (4.16)$$

де $k_{\text{збут.}}$ – коефіцієнт витрат на збут ($k_{\text{збут.}} = 0.025 \div 0,05$).

$C_{\text{вир.}}$ – виробнича собівартість. Згідно формулі 5.5 вона дорівнює:

$$\begin{aligned} C_{\text{вир.}} &= 63500 + 12700 + 16764 + 17156,6 + 31750 + 2540 \\ &= 144410,6 \text{ грн} \end{aligned}$$

Тоді витрати на збут:

$$C_{\text{збут.}} = 0,025 * 144410,6 = 3610,2 \text{ грн}$$

Повна (фактична) собівартість:

$$C_{\text{СП}} = C_{\text{вир.}} + C_{\text{збут.}} + C_{\text{адмін.}} = 144410,6 + 3610,2 + 63500 = 211520,8 \text{ грн}$$

Результати виконаних розрахунків занесено до таблиці 4.

Таблиця 4 – Калькуляція собівартості ПП

№ статті	Статті витрат	Сума, грн	Питома вага, %
----------	---------------	-----------	----------------

1	Основна заробітна плата	63500	30
2	Додаткова заробітна плата	12750	6
3	Витрати на сплату єдиного соціального внеску	16764	7,9
4	Вартість машинного часу, необхідного для розробки та налаштування ПП	17156,6	8,1
5	Загальновиробничі (накладні) витрати	31750	15
6	Вартість витратних матеріалів, комплектуючих	2540	1,2

Продовження таблиці 4

№ статті	Статті витрат	Сума, грн	Питома вага, %
7	Виробнича собівартість ($C_{C_{вир.}}$)	144410,6	68,3
8	Адміністративні витрати	63500	30
9	Витрати на збут	3610,2	1,7
10	Повна (фактична) собівартість ($C_{C_{п}}$)	211520,8	100,00

4.2 Визначення ціни програмного продукту на основі вартості його розробки

Ціна ПП ($P_{дог.}$) формується на основі економічно обґрунтованої собівартості його розробки, норми рентабельності, прибутку (певного відсотку торговельної надбавки) та податку на додану вартість (ПДВ):

$$P_{дог.} = (C_{C_{п}} + P_{рентаб.Н} + m) + ПДВ, \quad (4.17)$$

або

$$P_{дог.} = \left(C_{C_{п}} + \left(1 + \frac{P_{рентаб.Н} + m}{100} \right) \right) \times ПДВ \quad (4.18)$$

де $C_{C_{п}}$ – собівартість або поточні витрати на розробку ПП (системи, мережі), грн;

$P_{\text{рентаб.Н}}$ – нормативний рівень рентабельності, (дорівнює 15 %, коефіцієнт складає 0,15), грн;

m – торговельна надбавка ($m = 5-12\%$), грн.

$$P_{\text{дог.}} = 211520,8 * 1,2 = 253825 \text{ грн}$$

4.3 Оцінка конкурентоспроможності програмного продукту

У процесі дослідження виділимо основні технічні та економічні параметри базового та нового варіантів ПП. Також наводяться додаткові функції нового ПП.

Таблиця 5 – Характеристика основних техніко–економічних параметрів базового та нового варіантів ПП

№ ПП	Назва параметру	Одиниця виміру	Варіант		Характеристика параметра нового варіанта відносно базового (↑, ↓ чи =)
			базовий (аналог)	новий	
1.	Вартість ПП	грн	294035 (7)	253825 (10)	↑
2.	Простота та зручність інтерфейсу	бали	2(4)	5(10)	↑
3.	Кількість функцій ПП	шт.	4(4)	7(8)	↑
4.	Вага ПП	мбайт	17(8)	19 (7)	↓
5.	Час виконання запиту	мкс	17 (7)	13 (10)	↑

6.	Можливість нарощування функціональних характеристик	бали	4(8)	3(6)	↓
7.	Час відновлення системи після збою	сек	40(7)	40(7)	=
8.	Кількість людей, необхідних для обслуговування	чол.	1	1	=

Виходячи з отриманих результатів, розробка є кращою відносно базового варіанта за п'ятьма наступними параметрами:

- 1 – вартість ПП;
- 2 – простота та зручність інтерфейсу;
- 3 – кількість функцій ПП;
- 5 – час виконання запиту;

За двома параметрами новий і базовий варіанти програмного забезпечення є ідентичними:

- 7 – Час відновлення системи після збою;
- 8 – кількість людей, необхідних для обслуговування;

За двома параметрами новий варіант продукту поступається базовому:

- 4 – вага ПП;
- 6 – можливість нарощування функціональних характеристик;

4.4 Моделювання конкурентоспроможності розробленого ПП (ІС, КМ) та його аналогу

Моделювання оцінки конкурентоспроможності нового виробу (ПП, ІС, КМ) базується на побудові номограми для порівняння параметрів конкурентоспроможності. Відповідно, кожній із восьми осей з використанням певного масштабу виміру відмічаються точки, що

відповідають певним значенням параметрів, наведені в таблиці 5. Лінія, що проходить через них, утворює багатокутник, таким чином отримуємо номограму для порівняння параметрів конкурентоспроможності базового та нового варіантів ПП (рис. 20).

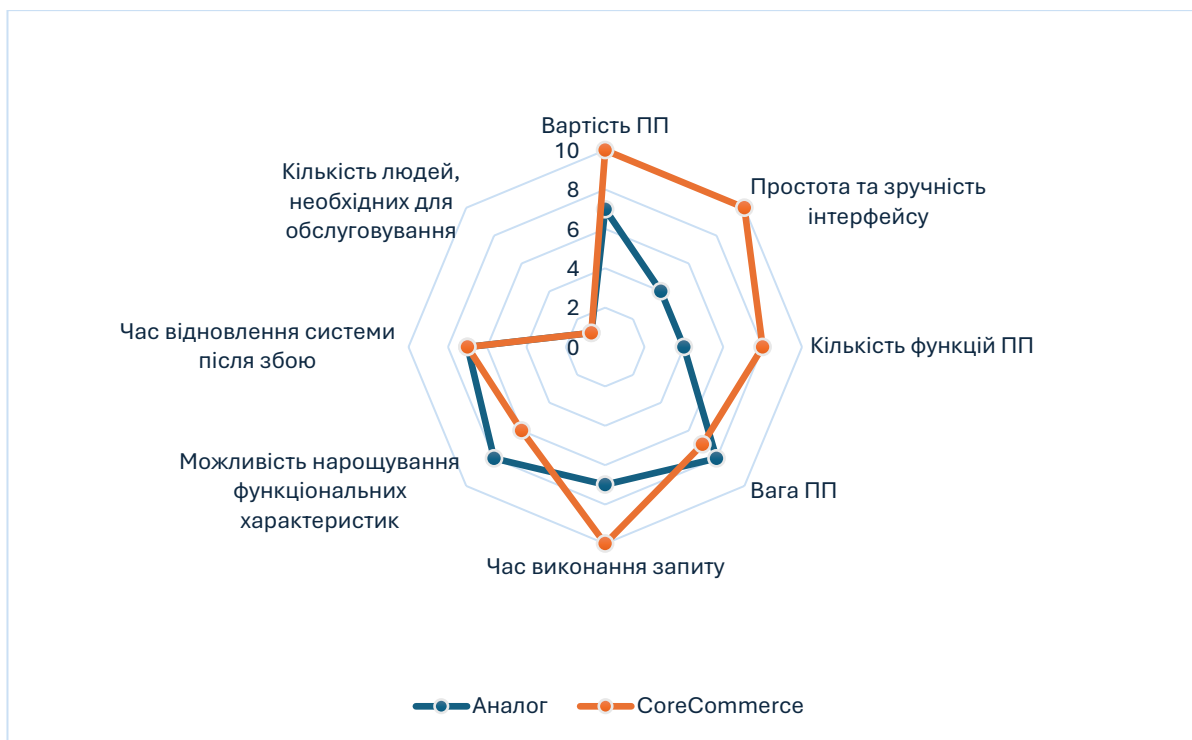


Рисунок 20 – Номограма для порівняння параметрів конкурентоспроможності базового та нового варіантів ПП «CoreCommerce»

ВИСНОВКИ

В ході виконання кваліфікаційної роботи було проаналізовано специфіку роботи продавців-консультантів в магазинах побутової техніки та обґрунтовано актуальність теми даної роботи. Також було розглянуто сучасне програмне забезпечення, що використовується в сфері продажів, для того, щоб визначитись з концептуальними рішеннями для покращення інформаційної підтримки діяльності, спираючись на їх переваги та недоліки.

Для більшого заглиблення у тему, що розглядається в кваліфікаційній роботі, було проведено математичне моделювання роботи магазину з продажу побутової техніки. Було розраховано середній час обслуговування клієнта на торговій точці, рівень завантаженості продавців в процесі роботи з ними та інші показники ефективності роботи магазину для продажу побутової техніки.

Результати моделювання можуть бути використані для прогнозування завантаженості продавців та підвищення якості обслуговування за рахунок організаційних рішень.

На основі результатів критичного аналізу існуючих аналогів програмного забезпечення, їх можливостей та недоліків, а також з урахуванням особливостей бізнес-процесу та відповідних функціональних вимог, а також додатково зібраних відомостей та даних, було розроблено концепцію продуктивного програмного забезпечення для локального розгортання на комп'ютері продавця-консультанта. Воно дозволить автоматизувати такі основні процеси, як: ведення переліку товарів на торговій точці, облік продажів, формування щоденної звітності та створення клієнтської бази магазину. Для розробки буда використана мова програмування C# з використанням платформи .NET 6, що дозволило розробити

багатошаровий застосунок, який у майбутньому може бути доповнений новими функціями без потреби якимось чином змінювати вже розгорнуті шари. Середовищем розробки обрали Visual Studio та базу даних SQLite з використанням ADO.NET для ефективного доступу до неї.

За результатами проєктування програмного комплексу, з використанням діаграмних методик UML, проведено фізичну реалізацію програмного забезпечення, як backend-частини застосунку для інформаційної підтримки діяльності продавця-консультанта, так і frontend-частини, яка реалізує простий та інтуїтивно зрозумілий користувацький інтерфейс.

Також були проведені розрахунки, що довели економічну доцільність розробки нового програмного забезпечення та наведені порівняння з застосунком, який використовувався на робочому місці продавця раніше.

ПЕРЕЛІК ПОСИЛАНЬ

1. Чекмарьов В. М. Основи програмування мовою C# : навч. посіб. / В. М. Чекмарьов. Київ : КНТ, 2022. 328 с.
2. Кузнецов А. М. Теорія масового обслуговування : підручник / А. М. Кузнецов. Київ : КНЕУ, 2021. 272 с.
3. Шамановська Ю. М. Математичне моделювання : навч. посіб. / Ю. М. Шамановська. Харків : ХНУРЕ, 2020. 164 с.
4. SQLite. Документація : веб-сайт. URL: <https://sqlite.org/docs.html> (дата звернення: 24.06.2025).
5. MSDN Documentation. C# Language Reference : веб-сайт. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 24.06.2025).
6. Visual Studio. Офіційна документація : веб-сайт. URL: <https://visualstudio.microsoft.com> (дата звернення: 24.06.2025).
7. Poster POS. Офіційний сайт. URL : <https://joinposter.com> (дата звернення: 24.06.2025).
8. BAS Облік Торгівлі. Офіційний сайт. URL : <https://bas-soft.eu> (дата звернення: 24.06.2025).
9. RemOnline. Офіційний сайт. URL : <https://remonline.app>
10. Google Analytics. Офіційний сайт. URL : <https://analytics.google.com> (дата звернення: 24.06.2025).
11. Hotjar. Офіційний сайт. URL : <https://www.hotjar.com> (дата звернення: 24.06.2025).
12. Woopra. Офіційний сайт. URL : <https://www.woopra.com> (дата звернення: 24.06.2025).
13. eSputnik. CDP-платформа автоматизації маркетингу. URL : <https://esputnik.com> (дата звернення: 24.06.2025).

14. Performance tuning. *Wikipedia*. URL : https://en.wikipedia.org/wiki/Performance_tuning (дата звернення: 24.06.2025).

15. Performance Optimization Strategies for Retail SaaS Providers – MobiDev. URL : <https://mobidev.biz/blog/performance-optimization-strategies-retail-saas-providers> (дата звернення: 24.06.2025).

16. Optimization Method of Usability Test Process Based on User Requirements – DiVA Portal. URL : <https://www.diva-portal.org/smash/get/diva2%3A1474217/FULLTEXT01.pdf> (дата звернення: 24.06.2025).

17. Impact of Usability Mechanisms. *ArXiv*. URL : <https://arxiv.org/abs/2408.12736> (дата звернення: 24.06.2025).

18. Iqbal M.W., Shinan K., Shahid Rafique S.R., Alourani A., Ashraf M.U., Rahim N.Z.A. Usability and optimization of online apps in user's context. *PeerJ Computer Science*. 10:e2561. 2024. DOI : <https://doi.org/10.7717/peerj-cs.2561>

19. Performance Optimization for SaaS APIs. *Zuplo*. URL : <https://zuplo.com/blog/2025/03/31/performance-optimization-for-saas-apis> (дата звернення: 24.06.2025).

20. Aesthetic–usability effect. *Wikipedia*. URL : https://en.wikipedia.org/wiki/Aesthetic%E2%80%93usability_effect (дата звернення: 24.06.2025).

21. User-Centric IT Security – How to Design Usable Security Mechanisms. *ArXiv*. URL : <https://arxiv.org/abs/1506.07167> (дата звернення: 24.06.2025).

22. A Comprehensive Model of Usability. *ArXiv*. URL : <https://arxiv.org/abs/1612.04598> (дата звернення: 24.06.2025).

23. Microsoft Learn. Design your application architecture. URL : <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps->

[azure/common-web-application-architectures](https://azure.com/common-web-application-architectures) (дата звернення: 24.06.2025).

24. NET 6 – Key Features & .NET 6 vs .NET 5 Comparison. *Brainhub*. URL : <https://brainhub.eu/library/net-6-upgrade> (дата звернення: 24.06.2025).

25. What's new in .NET 6 : веб-сайт. URL : <https://learn.microsoft.com/en-us/dotnet/core/whats-new/dotnet-6> (дата звернення: 24.06.2025).

26. .NET Support Policy : веб-сайт. URL : <https://dotnet.microsoft.com/en-us/platform/support/policy> (дата звернення: 24.06.2025).

27. What is New in .NET 6.0 / C# Corner : веб-сайт. URL : <https://www.c-sharpcorner.com/article/what-is-new-in-net-6-0/> (дата звернення: 24.06.2025).

28. Microsoft Learn. Overview of ADO.NET : веб-сайт. URL : <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview> (дата звернення: 24.06.2025).

29. ADO.NET Tutorial. *TutorialsTeacher*. URL : <https://www.tutorialsteacher.com/ado.net> (дата звернення: 24.06.2025).

30. C# Corner. Introduction to ADO.NET : веб-сайт. URL : <https://www.c-sharpcorner.com/article/introduction-to-ado-net> (дата звернення: 24.06.2025).

31. SQLite & ADO.NET (System.Data.SQLite) : веб-сайт. URL : <https://system.data.sqlite.org> (дата звернення: 24.06.2025).

32. Entity–relationship model. *Wikipedia*. URL : https://en.wikipedia.org/wiki/Entity–relationship_model (дата звернення: 24.06.2025).

33. Microsoft. Create and use an index (SQLite) : веб-сайт. URL : <https://learn.microsoft.com/en-us/sqlite/index> (дата звернення: 24.06.2025).

34. SQL UNIQUE Constraint. *W3Schools*. URL : https://www.w3schools.com/sql/sql_unique.asp (дата звернення: 24.06.2025).

35. Salesforce Blog. What is CRM? : веб-сайт. URL : <https://www.salesforce.com/crm/> (дата звернення: 24.06.2025).

36. Database Design for POS Systems. *Towards Data Science*. URL : <https://towardsdatascience.com/database-design-for-pos-systems> (дата звернення: 24.06.2025).

37. IBM Cloud. Transactional data model : веб-сайт. URL : <https://www.ibm.com/cloud/learn/transactional-data> (дата звернення: 24.06.2025).

38. Oracle. Many-to-Many Relationships : веб-сайт. URL : <https://docs.oracle.com> (дата звернення: 24.06.2025).

39. Microsoft. Referential Integrity Constraints : веб-сайт. URL : <https://learn.microsoft.com> (дата звернення: 24.06.2025).

40. University of Texas. Database Normalization : веб-сайт. URL : <https://www.cs.utexas.edu> (дата звернення: 24.06.2025).

41. Extending a Retail Database Model. *Vertabelo*. URL : <https://vertabelo.com> (дата звернення: 24.06.2025).

42. Microsoft Docs. ADO.NET Overview : веб-сайт. URL : <https://learn.microsoft.com/en-us/dotnet/framework/data/adonet/ado-net-overview> (дата звернення: 24.06.2025).

43. How to design a normalized database for .NET applications : веб-сайт. URL : <https://dev.to> (дата звернення: 24.06.2025).

44. Object Management Group (OMG). UML Class Diagram Specification : веб-сайт. URL : <https://www.omg.org/spec/UML> (дата звернення: 24.06.2025).

45. Alhir S. S. UML in a Nutshell. O'Reilly Media, 2002. 238 p.

46. Oracle. DAO Design Pattern : веб-сайт. URL : <https://docs.oracle.com/javaee/6/tutorial/doc/bncij.html> (дата звернення: 24.06.2025).

47. Microsoft Learn. Business Layer in .NET Apps : веб-сайт. URL : <https://learn.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/common-web-application-architectures#business-layer> (дата звернення: 24.06.2025).

ДОДАТОК А ВІДОМОСТІ РОБОТИ

Таблиця А.1 – Відомості роботи

Формат	№ п/п	Назва документу	Найменування об'єкту або вибору	Кількість сторінок
A4	1	Пояснювальна записка	КЦТПАР.122-22- 1п.01.00.КР.ПЗ	110
Графічна частина				
A4	2	Аналіз предметної області та основні положення	КЦТПАР.122-22- 1п.01.01.КР.ПЗ	1
A4	3	Аналіз предметної області та основні положення	КЦТПАР.122-22- 1п.01.02.КР.ПЗ	1
A4	4	Аналіз діючих аналогів програмного забезпечення	КЦТПАР.122-22- 1п.01.03.КР.ПЗ	1
A4	5	Аналіз діючих аналогів програмного забезпечення	КЦТПАР.122-22- 1п.01.04.КР.ПЗ	1
A4	6	Аналіз діючих аналогів програмного забезпечення	КЦТПАР.122-22- 1п.01.05.КР.ПЗ	1
A4	7	Математичне моделювання роботи магазину з продажу побутової техніки	КЦТПАР.122-22- 1п.01.06.КР.ПЗ	1
A4	8	ВРМН діаграма робочого процесу	КЦТПАР.122-22- 1п.01.07.КР.ПЗ	1
A4	9	Діаграма варіантів використання	КЦТПАР.122-22- 1п.01.08.КР.ПЗ	1
A4	10	Діаграма активності	КЦТПАР.122-22- 1п.01.09.КР.ПЗ	1
A4	11	ER-діаграма бази даних	КЦТПАР.122-22- 1п.01.10.КР.ПЗ	1
A4	12	UML-Діаграма класів	КЦТПАР.122-22- 1п.01.11.КР.ПЗ	1
A4	13	Наведення прикладу користувачького інтерфейсу	КЦТПАР.122-22- 1п.01.12.КР.ПЗ	1
A4	14	Номограма порівняння параметрів	КЦТПАР.122-22- 1п.01.12.КР.ПЗ	1

ДОДАТОК Б
ГЛОСАРІЙ ПРЕДМЕТНОЇ ОБЛАСТІ ДЛЯ АВТОМАТИЗАЦІЇ
РОБОТИ ПРОДАВЦІВ-КОНСУЛЬТАНТІВ В МАГАЗИНАХ ПОБУТОВОЇ
ТЕХНІКИ

Таблиця Б.1 – Глосарій предметної області для автоматизації роботи продавців-консультантів

Термін	Визначення
POS (Point of Sale)	Система управління торгівельними операціями
SQLite	Легка вбудована СУБД для десктопних застосунків
ADO.NET	Технологія доступу до даних у .NET
C#	Об'єктно-орієнтована мова програмування яка підтримує сучасні парадигми програмування та широко використовується для розробки десктопних, веб- та мобільних застосунків.
.NET 6	Кросплатформна програмна платформа що дозволяє створювати сучасні застосунки
Продаж	Процес реалізації товару клієнту
Облік товару	Функція фіксації наявності, залишків або руху товару
Чек	Документ, що підтверджує факт покупки
Продавець-консультант	Працівник магазину, який здійснює продажі та обслуговує клієнтів
Знижка	Зменшення вартості товару, що застосовується продавцем
Інтерфейс користувача	Візуальне середовище взаємодії користувача з програмою

ДОДАТОК В ТЕХНІЧНЕ ЗАВДАННЯ

В.1 Введення

Найменування: Програмний комплекс для автоматизації роботи продавця-консультанта в магазині побутової техніки.

Програмний продукт (ПП) призначений для автоматизації обов'язків продавця-консультанта: ведення обліку товарів, здійснення продажів, формування чеків, контроль залишків, ведення обліку клієнтів та формування звітів. Система повинна мати зручний інтерфейс, працювати локально без постійного підключення до Інтернету, забезпечувати безпеку даних та бути стійкою до збоїв.

В.2 Підстава для розробки

Основою для розробки стали завдання кваліфікаційної роботи бакалавра, затверджені завідувачем кафедри цифрових технологій, проєктування та аналітичних рішень, в рамках спеціальності 122 «Комп'ютерні науки».

В.3 Призначення розробки

Програмний комплекс має оптимізувати роботу продавців-консультантів, автоматизуючи ключові операції: підрахунок товарів, реєстрацію продажів, друк квитанцій, формування заявок та резервування товарів. Система також повинна зменшити кількість помилок, підвищити швидкість обслуговування клієнтів та забезпечити актуальність даних.

В.4 Вимоги до програмного виробу

В.4.1 Вимоги до функціональних характеристик:

Програмне забезпечення має бути реалізоване у вигляді робочого столу з графічним інтерфейсом користувача операційної системи Windows. Функція консультанта, зручне підключення до Інтернету в будь-який час. Програмування на C# та база даних SQLite.

Система повинна підтримувати авторизацію користувачів, включаючи вхід у систему за допомогою імені користувача та пароля, а також надання обмеженого доступу до функціональності на основі прав доступу.

Застосунок повинен реалізувати функціональність для додавання нових продуктів до бази даних, редагування існуючих записів та видалення застарілих або помилкових елементів. Облік залишків має вестися в режимі реального часу, а кількість продукції має автоматично оновлюватися після кожної транзакції. Продаж слід здійснити, вибравши товар, розрахувавши загальну суму з урахуванням знижки, а потім створивши електронний чек та роздрукувавши його. Додаток повинен підтримувати базу даних клієнтів, яка зберігає інформацію про клієнтів та історію їхніх замовлень.

Також необхідно реалізувати можливість створення звітів на основі різних критеріїв: щоденні продажі, щомісячні продажі, список постійних клієнтів та залишки на складі. Вся історія транзакцій повинна зберігатися у базі даних з датою, часом, сумою продажів, переліком товарів та відповідальним співробітником.

В.4.2 Вимоги до надійності

Система повинна перевіряти правильність вхідних даних, щоб уникнути логічних та технічних помилок. Підтримуються визначені

часові інтервали та резервне копіювання бази даних. Після завершення робочого процесу дані захищаються в найактуальнішому стані.

В.4.3 Вимоги до технічного забезпечення

Мінімальні вимоги до технічного забезпечення для додатка:

Операційна система: Windows 10/11.

Процесор: Intel Core i3 або аналог від AMD і вище.

RAM: від 4 ГБ (Рекомендовано 8 ГБ)

Місце на диску: не менше 2 ГБ.

База даних: локальна SQLite.

Монітор: роздільна здатність не нижче 1280×720

В.4.4 Умови експлуатації

Програму можна запускати в офісному приміщенні з температурою повітря від 10 до 30 градусів за Цельсієм та відносною вологістю від 30% до 60%. Усі функції працюють в автономному режимі, тому постійне підключення до Інтернету не потрібне. Користувач повинен мати базові навички користування комп'ютером для роботи з застосунком. Програма має підтримувати можливість резервного копіювання даних на зовнішні носії у вигляді архівного файлу.

В.4.5 Вимоги до програмної сумісності

Програмне забезпечення має бути сумісним із сучасними версіями операційних систем Windows 10 та 11. Система повинна підтримувати експорт звітів у найпоширеніші формати, включаючи PDF, CSV та XLSX, щоб сторонні програми могли обробляти дані. Передбачається підтримка друку чеків в магазині на спеціальному POS-принтері. За

потреби повинна бути можливість оновити програму, встановивши локальний інсталяційний файл або через мережу Інтернет.

В.5 Вимоги до програмної документації роботи програмного комплексу

Документаційними матеріалами до програмного продукту є:

- технічне завдання;
- керівництво адміністратора (інсталяція, резервне копіювання та оновлення);
- керівництво користувача (робота з товарами, продажами, звітами);
- Структура архітектурної системи
- Модель бази даних з описом таблиць і зв'язків

В.6 Техніко-економічні показники

У роботі використовувалися безкоштовні інструменти розробки: середовище Visual Studio Community, мова програмування C# та локальна база даних SQLite, що зменшило витрати на впровадження. Для використання системи не потрібне потужне обладнання чи серверна інфраструктура; достатньо офісного комп'ютера з мінімальними ресурсами. Розроблені програмні продукти підвищують ефективність роботи продавців-консультантів, скорочують час обробки замовлень, зменшують кількість помилок у продажах, покращують облік товарів та клієнтів. Очікуваний термін окупності впровадження програмного забезпечення становить менше одного року.

В.7 Стадії та етапи розробки

Стадії та етапи розробки програмного продукту зазначені у таблиці

Б.1.

Таблиця В.1 – Стадії та етапи розробки

№	Етап роботи	Опис виконуваної роботи
1	Аналіз вимог	Вивчення процесів роботи продавця-консультанта в магазині побутової техніки. Визначення типових дій працівника, аналіз подібних програмних рішень, виявлення ключових потреб, формулювання функціональних вимог до системи.
2	Формування технічного завдання	Систематизація зібраної інформації, оформлення вимог у вигляді технічного завдання з описом функцій. Середовища роботи, технічних параметрів, вимог до інтерфейсу, безпеки, обсягів даних і способів резервного копіювання.
3	Розробка інтерфейсу	Створення ескізів і реалізація графічного інтерфейсу користувача для обліку товарів, продажів, клієнтів, перегляду історії продажів та звітності по касі. Врахування зручності роботи продавця-консультанта, доступності всіх функцій і швидкої навігації.
4	Програмування	Реалізація структури бази даних, розробка логіки взаємодії модулів та створення основного функціоналу: реєстрація продажів, обліку товарів, збереження історії, генерація звітів.
5	Тестування	Проведення тестування кожного модуля та всієї системи в цілому. Перевірка працездатності у звичайних умовах, збереження даних й їх обробка, стійкість до помилок, аналіз швидкодії та виправлення виявлених помилок або дефектів.
6	Написання супровідної документації	Створення технічного опису програми, підготовка керівництва користувача та адміністратора, опис базових сценаріїв, інструкції з інсталяції, оновлення, резервного копіювання, відновлення даних та експлуатації програми.
7	Впровадження	Інсталяція програмного забезпечення на робочому місці, перевірка функціоналу та реальних даних, презентація системи, тестова експлуатація, збір відгуків, проведення навчання персоналу з використання програми.

В.8 Порядок контролю і приймання

Контроль та приймання програмного забезпечення здійснюються відповідно до вимог технічного завдання. Перевірка відповідності програмного продукту заданим характеристикам проводилась шляхом тестування функціональності, оцінки зручності використання інтерфейсу, надійності зберігання даних, коректності обробки запитів та стабільності роботи системи у звичайному режимі. Результати тестування підтвердили правильність реалізації основних функціональних можливостей: облік продукції, продажі, генерація чеків, управління базою даних клієнтів та генерація звітів. Програмний продукт успішно пройшов етап тестування та був визнаний повноцінним та життєздатним рішенням.

ДОДАТОК Г

ПЕРЕЛІК ТА ВІДБИТКИ НАУКОВИХ ПУБЛІКАЦІЙ ЗА ТЕМОЮ РОБОТИ



DOI <https://doi.org/10.30525/978-9934-26-506-8-113>

USING ORANGE3 TO AUTOMATE DATA MINING PROCESSES IN THE MODE OF VISUAL CONSTRUCTION OF DATA TRANSFORMATION STAGES

ВИКОРИСТАННЯ ORANGE3 ДЛЯ АВТОМАТИЗАЦІЇ ПРОЦЕСІВ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ В РЕЖИМІ ВІЗУАЛЬНОЇ ПОБУДОВИ ЕТАПІВ ПЕРЕТВОРЕННЯ ДАНИХ

Nefedchenko O.O.,
Student (group 122-22-1p),
LLC "Technical university
"Metinvest polytechnic",
Zaporizhzhia, Ukraine

Нефедченко О.О.,
студент гр. 122-22-1п,
ТОВ «Технічний університет
«Метінвест політехніка»,
м. Запоріжжя, Україна

Sahaida P.I.,
DSc (Engineering),
Associate Professor, LLC "Technical
university "Metinvest polytechnic",
Zaporizhzhia, Ukraine

Сагайда П.І.,
д.т.н., доцент,
ТОВ «Технічний університет
«Метінвест політехніка»,
м. Запоріжжя, Україна

На сучасних підприємствах та в організаціях накопичені корпоративні дані мають великі розміри та стають все більш доступними для фахівців з їх обробки та витягу з них моделей. Відповідно, зростають вимоги до рівня підготовки спеціалістів з Data Science. Якісний розвідувальний аналіз даних (exploratory data analysis) потребує володіння декількома мовами або засобами програмування для розв'язання задач статистичної обробки даних, візуалізації результатів, підготовки звітів. Між тим коло працівників-аналітиків розширюється,

СЕРТИФІКАТ
підтверджує, що

Олег Нефедченко

приймала участь у IX Всеукраїнській науково-практичній конференції
«Сучасні інформаційні технології, засоби автоматизації та електропривод»,
м. Краматорськ – Тернопіль, 17 – 19 квітня 2025 р.
(6 годин – 0,2 кредити ECTS)



Голова
організації,
д.т.н., професор
О.Ф. Тарасов



<http://cit.dgma.donetsk.ua/>

Міністерство освіти і науки України

Донбасська державна
машинобудівна
академія



Донецький
фізико-технічний
інститут
ім. А. А. Галкіна
НАН України

ПАТ
«Новокраматорський
машинобудівний
завод»



Інститут економіки
промисловості
НАН України»

Громадська спілка «ІТ кластер
Донеччини» (IT Cluster Donbass)

ТОВ «Інформаційні
технології САПР»



за підтримкою
концерну
Siemens AG

SIEMENS

ЦЕНТР **САПР**

**СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ,
ЗАСОБИ АВТОМАТИЗАЦІЇ ТА ЕЛЕКТРОПРИВОД**

МАТЕРІАЛИ

IX Всеукраїнської науково-практичної конференції

(17–19 квітня 2025 року)

Краматорськ – Тернопіль

ДДМА

2025

Людино-машинна командна робота: сучасний стан та напрямки розвитку

Плуталов Я. А., Нефедченко О. О., Сагайда П. І.
ТОВ «Технічний університет «Метінвест Політехніка»

Згідно з проаналізованими літературними джерелами [1-3], ключовими концепціями та компонентами людської та artificial intelligence (AI) agentic взаємодії є ті, що стосуються швидкого розвитку штучного інтелекту (ШІ) та його інтеграції в людино-машинні команди. Джерела підкреслюють еволюцію ШІ-агентів від пасивних інструментів до активних співпрацівників у цих командах.

Основними поняттями та компонентами сучасного стану та напрямки розвитку предметної області є наступні.

ШІ-агенти розглядаються як сутності, що сприймають своє оточення та діють раціонально, але сучасні ШІ-агенти еволюціонували до автономних, ітеративних та інтелектуальних членів команди. Вони здатні навчатися, розмірковувати та адаптуватися у складних середовищах. Характеристики агентних ШІ-систем включають планування та виконання багатоетапних