

**Міністерство освіти і науки України
Луцький національний технічний університет
Факультет комп'ютерних та інформаційних технологій
Кафедра інженерії програмного забезпечення**

**КВАЛІФІКАЦІЙНА РОБОТА
ЗА СТУПЕНЕМ ВИЩОЇ ОСВІТИ «БАКАЛАВР»**

РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ КАФЕ З ВИКОРИСТАННЯМ C# /

DEVELOPMENT OF A CAFE MANAGEMENT SYSTEM USING C#

спеціальність 121 «Інженерія програмного забезпечення»
освітня програма «Інженерія програмного забезпечення»

Виконав: здобувач вищої освіти
групи ІПЗ-41
Іванов Владислав Олександрович

Керівник:
Сичук Віктор Анатолійович

Кваліфікаційну роботу
допущено до захисту
«10» 06 2025р.
Гарант освітньої програми:
к.т.н., доцент
Ліщина Наталія Миколаївна


Луцьк – 2025 року

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних та інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

М.П.
«20» 12 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ВИЩОЇ ОСВІТИ

Іванову Владиславу Олександровичу

(прізвище, ім'я, по батькові)

1. Тема кваліфікаційної роботи «Розробка системи управління кафе з використанням C#»

Керівник роботи: к.т.н., доцент Сичук В. А.

затверджені наказом закладу вищої освіти від «19» грудня 2024 р. № 474/01-02

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи бакалавра «10» червня 2025 р.

3. Вихідні дані до роботи: C#, Visual Studio, Windows Forms, методичні вказівки до виконання кваліфікаційної роботи бакалавра.

4. Зміст розрахунково-пояснювальної записки (перелік питань, що потрібно розробити):

Аналіз сучасного стану проблеми, існуючих методів і засобів її розв'язання, аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення, опис функціонального наповнення об'єкта проектування, практична реалізація об'єкта проектування.

5. Перелік графічного матеріалу: 11 рисунків.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис	
		завдання видав	завдання прийняв
Аналіз предметної області	Сичук В. А.		
Специфікація вимог до розробленої системи	Сичук В. А.		
Розробка об'єкта проектування	Сичук В. А.		
Нормоконтроль	Вознюк А. В.		
Гарант ОП	Ліщина Н. М.		
Показник запозичень тексту		1,48 %	
Академічна доброчесність	Сичук В. А.		

7. Дата видачі завдання «20» грудня 2024 р.

№	Назва етапів кваліфікаційної роботи бакалавра	Строк виконання етапів роботи	Примітка
1	Огляд літературних джерел по темі кваліфікаційної роботи бакалавра	до 04.02.2025 р.	виконано
2	Аналіз проблеми розробки та впровадження об'єкту проектування	до 01.03.2025 р.	виконано
3	Обґрунтування вибору шляхів, технологій і засобів вирішення поставленого завдання	до 15.03.2025 р.	виконано
4	Розробка функціонально-структурної схеми роботи об'єкта проектування	до 29.03.2025 р.	виконано
5	Практична реалізація об'єкта проектування	до 26.04.2025 р.	виконано
6	Тестування та налагодження об'єкта проектування	до 03.05.2025 р.	виконано
7	Здача чистового варіанту кваліфікаційної роботи бакалавра на кафедру	до 10.06.2025 р.	виконано

Здобувач вищої освіти

(підпис)

Гванов В.О.

(прізвище, ініціали)

Керівник кваліфікаційної роботи

(підпис)

Сичук В.А.
(прізвище, ініціали)

АНОТАЦІЯ

Іванов В. О. Розробка системи управління кафе з використанням C#. Рукопис. Кваліфікаційна робота бакалавра ОП «Інженерія програмного забезпечення» спеціальності «Інженерія програмного забезпечення». Луцький національний технічний університет. Луцьк, 2025.

Дана робота присвячена розробці інформаційної системи для автоматизації процесів управління кафе з використанням мови програмування C#. У роботі розглянуто проблематику впровадження автоматизованих рішень у сфері громадського харчування, проаналізовано існуючі системи, визначено їхні переваги та недоліки. Запропоновано власне рішення, яке реалізує базові функції обліку замовлень, формування, збереження та друку чеків. У роботі описано специфікацію вимог, структуру системи, процес реалізації інтерфейсу користувача та обробки подій, а також проведено тестування програмного забезпечення. Система розроблена за допомогою Windows Forms у середовищі Visual Studio, реалізує принципи об'єктно-орієнтованого програмування і є прикладом практичного застосування засобів C# у розробці прикладного ПЗ.

Ключові слова: система управління кафе, автоматизація обліку, C#, Windows Forms, графічний інтерфейс користувача, формування чеків, Visual Studio, об'єктно-орієнтоване програмування.

ABSTRACT

Ivanov V. Development of a Cafe Management System Using C#. Manuscript. Qualification work of the bachelor's program "Software engineering" in the specialty "Software engineering". Lutsk National Technical University. Lutsk, 2025.

This work is devoted to the development of an information system for automating cafe management processes using the C# programming language. The paper examines the issues of implementing automated solutions in the catering sector, analyzes existing systems, and identifies their advantages and disadvantages. An original solution is proposed that implements the basic functions of order accounting, receipt generation, storage, and printing. The paper describes the requirements specification, system structure, user interface implementation process, and event processing, and software testing is conducted. The system is developed using Windows Forms in the Visual Studio environment, implements the principles of object-oriented programming, and is an example of the practical application of C# tools in application software development.

Keywords: cafe management system, accounting automation, C#, Windows Forms, graphical user interface, receipt generation, Visual Studio, object-oriented programming.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ	9
1.1 Аналіз сучасного стану проблеми	9
1.2 Постановка завдання на кваліфікаційну роботу бакалавра	13
РОЗДІЛ 2 СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЕНОЇ СИСТЕМИ	14
2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення	14
2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання ...	16
РОЗДІЛ 3 РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ КАФЕ	25
3.1 Практична реалізація об'єкта проектування.....	25
3.2 Тестування та налагодження системи.....	31
3.3 Розробка документації для програмного продукту	34
3.4 UX/UI-аналіз розробленої системи	36
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41

ВСТУП

Актуальність теми. У сучасному світі інформаційні технології відіграють ключову роль у розвитку практично всіх сфер людської діяльності. Зокрема, у сфері громадського харчування зростає попит на автоматизовані системи управління, які здатні оптимізувати повсякденну роботу персоналу, підвищити якість обслуговування клієнтів і скоротити витрати часу та ресурсів. Автоматизація бізнес-процесів у кафе дозволяє власникам і адміністраторам ефективно контролювати процеси продажу, управління меню, облік товарів та формування звітності. Це сприяє зменшенню кількості помилок у роботі з клієнтами, підвищенню швидкості обслуговування та задоволеності відвідувачів.

Одним з основних завдань таких систем є можливість швидко створювати чеки, здійснювати облік замовлень, формувати звіти та вести базу продуктів. Традиційне ведення таких процесів у ручному режимі вимагає значних зусиль і часу, натомість сучасні інформаційні системи надають інструменти для автоматизації цих операцій. Впровадження таких рішень у малий і середній бізнес є особливо актуальним, оскільки дозволяє суттєво зменшити витрати на адміністрування [1].

Мова програмування C# у середовищі Visual Studio є потужним інструментом для створення надійних, зручних і функціональних застосунків. Завдяки можливостям Windows Forms, можна створити інтуїтивно зрозумілий інтерфейс користувача, який забезпечує швидкий доступ до основних функцій програми, що є критично важливим у сфері громадського харчування, де швидкість і точність мають вирішальне значення.

Таким чином, розробка системи управління кафе з використанням мови програмування C# є надзвичайно актуальною, оскільки дозволяє реалізувати практичний, економічно вигідний та ефективний інструмент для вирішення щоденних задач бізнесу. Така система здатна значно поліпшити організацію

роботи кафе, що відповідає сучасним вимогам цифрової трансформації підприємств.

Мета кваліфікаційної роботи – розробити функціональну програмну систему управління кафе, яка забезпечує зручну взаємодію з користувачем, дозволяє здійснювати облік замовлень, формувати чеки, зберігати і друкувати інформацію.

Об’єкт кваліфікаційної роботи – процес автоматизації обліку замовлень та взаємодії з клієнтом у закладах громадського харчування.

Предмет роботи – програмне забезпечення, що реалізує функції обліку замовлень у кафе.

Завдання кваліфікаційної роботи:

- провести аналіз сучасних підходів до автоматизації процесів у сфері громадського харчування;
- визначити та обґрунтувати вимоги до системи управління кафе;
- вибрати оптимальні технології та засоби для розробки програмного забезпечення;
- розробити функціональне програмне забезпечення на мові програмування C# з використанням Windows Forms;
- здійснити тестування та налагодження створеного програмного продукту;
- оцінити ефективність та зручність експлуатації розробленої системи.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАНЬ НА КВАЛІФІКАЦІЙНУ РОБОТУ

1.1 Аналіз сучасного стану проблеми

У сучасних умовах цифрової трансформації бізнесу автоматизація процесів обслуговування у сфері громадського харчування є необхідною умовою для підвищення ефективності роботи та конкурентоспроможності закладу. Кафе та ресторани дедалі частіше впроваджують програмні системи управління, які дають змогу здійснювати облік замовлень, формувати чеки, керувати складом продукції, аналізувати обсяги продажів та підвищувати загальну якість обслуговування клієнтів [1-2].

Аналіз наукових джерел і публікацій у галузі інформаційних технологій засвідчує, що при створенні прикладного програмного забезпечення для сфери громадського харчування важливими є такі критерії, як зручність користування, інтуїтивно зрозумілий інтерфейс, висока швидкодія, багатофункціональність, масштабованість, безпека обробки та зберігання даних, а також адаптивність до індивідуальних потреб закладу. Ці аспекти детально описані, зокрема, у праці Прощенка А. К. «Перспективи використання CRM-систем на підприємствах громадського харчування», де автор розглядає практичні вимоги до інформаційних систем у цій сфері.

Дослідження Вінтоняка С. М., Кіся Я. П., Чируна Л. Б. в статті «Розроблення інформаційної системи для управління ресторанним бізнесом» акцентують увагу на важливості адаптивності й масштабованості програмних продуктів, що використовуються в малому та середньому бізнесі. Автори зазначають, що саме кросплатформні рішення, розроблені мовами високого рівня, забезпечують ефективне впровадження нових технологій у щоденну діяльність закладів громадського харчування.

Окремо варто згадати працю Mark Summerfield «Programming in Python 3» (2020), яка розглядає сучасні підходи до побудови інтерфейсів користувача та

обробки даних, що є релевантними і для розробки аналогічних рішень на інших мовах програмування, зокрема C#. Також у книзі Andrew Troelsen і Philip Japikse «Pro C# 8 with .NET Core 3» (Apress, 2020) детально висвітлено технічні можливості створення прикладних систем на базі C#, включаючи приклади реалізації POS-систем.

Загалом, більшість авторів доходять висновку, що використання об'єктно-орієнтованих мов програмування, таких як C#, є доцільним для розробки гнучких і надійних систем, орієнтованих на автоматизацію процесів обслуговування у кафе. Їх застосування забезпечує реалізацію необхідного функціоналу, включаючи створення замовлень, облік продуктів, формування чеків і звітності, що є критично важливими для ефективного ведення діяльності в умовах сучасного ресторанного бізнесу.

Здійснений огляд аналогічних програмних рішень дозволяє виділити декілька поширених прикладів систем управління кафе:

- Poster POS – хмарна система з широким функціоналом, проте вимагає стабільного підключення до Інтернету, що не завжди можливо у віддалених або невеликих закладах [3];
- Open Source-альтернативи (наприклад, Floreant POS) – безкоштовні рішення, які потребують значних знань для налаштування та модифікації, що ускладнює їх впровадження без участі кваліфікованого фахівця [4].

Poster POS – це сучасна хмарна система автоматизації кафе, ресторанів та барів, розроблена українською компанією Poster POS Inc., заснованою у 2013 році. Вона орієнтована на швидке та зручне обслуговування клієнтів, забезпечуючи власникам закладів доступ до системи з будь-якої точки світу через інтернет. Інтерфейс програми підтримує українську, англійську та інші мови, що дозволяє адаптувати програму до різноманітного персоналу. Система включає модулі для прийому замовлень, управління меню, контролю складу, а також потужну аналітику для керівників. Poster POS має інтуїтивно зрозумілий інтерфейс і не потребує складного налаштування, що робить її однією з найзручніших у впровадженні для малого та середнього бізнесу. Крім цього,

система має адаптивний дизайн і підтримує роботу з мобільними пристроями, що є додатковою перевагою для динамічного бізнесу (рис. 1.1) [3].



Рисунок 1.1 – Хмарна система автоматизації кафе Poster POS [3]

Florent POS є безкоштовною системою з відкритим програмним кодом, яка розроблялася в США під егідою компанії OROCUBE. Це рішення орієнтоване на підприємства ресторанного типу, які потребують базової, але стабільної та надійної системи обліку. Середовище розробки Florent POS підтримує англійську мову інтерфейсу та надає широкі можливості для модифікації, що дозволяє адаптувати систему під потреби конкретного закладу. Оскільки код програми є відкритим, розробники можуть додавати або змінювати функціонал відповідно до внутрішніх бізнес-процесів. Програма підтримує базові операції – облік замовлень, управління меню, створення звітності, а також можливість інтеграції з фіскальними пристроями. Незважаючи на дещо простіший інтерфейс, система активно використовується в багатьох країнах завдяки своїй надійності та гнучкості в налаштуванні (рис. 1.2) [4].



Рисунок 1.2 – Система Floreant POS [4]

Як Poster POS, так і Floreant POS демонструють вдале поєднання зручності, надійності та відповідності сучасним потребам закладів громадського харчування, що робить їх актуальними альтернативами для впровадження в українському бізнес-середовищі.

Важливим напрямом дослідження є також вивчення патентної документації, яка свідчить про зростаючий інтерес до створення спеціалізованих програмних продуктів для малого й середнього бізнесу в галузі харчування. Більшість запатентованих рішень акцентують увагу на оптимізації процесу формування чеків, інтеграції з мобільними додатками та підвищенні персоналізованості обслуговування.

Попри наявність різних рішень, більшість із них або надто складні в реалізації та використанні для невеликих закладів, або мають обмежений функціонал, який не враховує специфіку внутрішніх процесів кожного окремого бізнесу. Саме тому постає потреба у створенні нової інформаційної системи управління кафе, яка б відповідала таким критеріям: простота використання, зрозумілий інтерфейс, автономність, можливість локального зберігання даних, функції формування замовлень і чеків, а також підтримка друку звітної інформації.

Таким чином, створення програмного забезпечення для управління кафе на базі мови програмування C# та платформи Windows Forms є доцільним та обґрунтованим кроком, що дозволить задовольнити потреби малого бізнесу у доступній та ефективній автоматизації операційної діяльності.

1.2 Постановка завдання на кваліфікаційну роботу бакалавра

З метою вирішення завдань автоматизації обліку та обслуговування у закладах громадського харчування, а також з урахуванням потреб малого бізнесу, у межах кваліфікаційної роботи визначено такі цілі, реалізація яких дозволить досягти запланованого результату:

- провести аналіз сучасних підходів до автоматизації процесів у сфері громадського харчування;
- визначити та обґрунтувати вимоги до системи управління кафе;
- вибрати оптимальні технології та засоби для розробки програмного забезпечення;
- розробити функціональне програмне забезпечення на мові програмування C# з використанням Windows Forms;
- здійснити тестування та налагодження створеного програмного продукту;
- оцінити ефективність та зручність експлуатації розробленої системи.

РОЗДІЛ 2

СПЕЦИФІКАЦІЯ ВИМОГ ДО РОЗРОБЛЮВАНОЇ СИСТЕМИ

2.1 Аналіз, визначення вимог до розроблюваного програмного забезпечення та проектування програмного забезпечення

Процес розробки системи управління кафе починається з етапу аналізу вимог, визначення цілей, функціональних характеристик і моделювання програмної системи. У цьому розділі обґрунтовується вибір моделі проектування, визначаються вимоги до системи, описуються її функціональні можливості та структура, а також представлено моделі в нотації UML [5].

З огляду на потребу розробити десктопний застосунок для автоматизації роботи кафе, що має відносно просту логіку, обмежену кількість користувачів і не потребує масштабованої архітектури, доцільно застосувати каскадну модель життєвого циклу ПЗ. Такий підхід дозволяє послідовно реалізувати всі етапи – від аналізу вимог до тестування, а також зручно контролювати якість на кожному з етапів. Для моделювання системи обрано об'єктно-орієнтований підхід, що дозволяє ефективно реалізувати логіку бізнес-процесів через класи, об'єкти, успадкування та поліморфізм [6].

Вимоги до системи було визначено методом аналізу аналогічних рішень (Poster POS, Floreant POS) та результатів роботи персоналу кафе.

Під час формування вимог до системи особливу увагу було приділено критеріям зручності інтерфейсу, швидкодії програми та достатньої функціональності для обробки повного циклу замовлення. Ці характеристики є критично важливими для систем, що функціонують у сфері громадського харчування, де процес обслуговування має бути максимально оперативним і простим для персоналу.

Для виявлення реальних потреб користувачів було проведено інтерв'ювання персоналу кафе, включаючи барист, касирів і адміністратора. У процесі опитування з'ясовувалися такі аспекти:

- з якими труднощами вони стикаються при обробці замовлень вручну;

- які функції були б найбільш корисними в повсякденній роботі;
- які елементи інтерфейсу повинні бути максимально доступними;
- які типи звітності чи збереження чеків є необхідними;
- що саме викликає затримки у швидкості обслуговування.

На основі отриманих відповідей сформовано базовий функціональний каркас системи: передбачено чітке розділення товарів на категорії (напої, десерти), введення кількості через текстові поля, розрахунок вартості, швидке очищення форми для наступного клієнта, збереження та перегляд чеків.

Окрім інтерв'ювання, було проведено аналіз сценаріїв використання (user scenarios), які описують типові робочі ситуації, наприклад:

- сценарій 1. Клієнт робить стандартне замовлення (лате + торт), персонал вводить кількість, натискає кнопку розрахунку, зберігає чек і друкує його;
- сценарій 2. Клієнт передумав, необхідно скасувати введені дані – використовується кнопка «Очистити»;
- сценарій 3. Клієнт просить повторити попереднє замовлення – система дозволяє відкрити раніше збережений чек.

Специфікація вимог до програмного забезпечення. Загальна мета: створити програмний продукт для автоматизації ключових процесів кафе, включаючи формування замовлень, розрахунок вартості, друк чеків та управління меню.

Основні задачі системи:

- надати інтерфейс для вибору товарів (напоїв, десертів);
- виконувати автоматичний розрахунок вартості замовлення;
- зберігати інформацію про замовлення;
- створювати та зберігати електронні чеки;
- забезпечити можливість друку чеку;
- реалізувати функцію очищення форми замовлення для нового клієнта.

Програмне забезпечення реалізує такі функції:

- введення даних про кількість обраних товарів (через чекбокси і текстові поля);
- автоматичний розрахунок вартості товарів та підрахунок загальної суми (включаючи податок);
- формування текстового чека з можливістю збереження у файл або друку;
- очищення полів для нового замовлення;
- збереження та відкриття чеків;
- простий інтерфейс з інтуїтивною навігацією.

Система забезпечує:

- контроль введення даних;
- зручний візуальний інтерфейс для персоналу;
- обмежений локальний доступ (без мережевої взаємодії або ролей);
- UX/UI-дизайн і технології обслуговування

Інтерфейс реалізовано за допомогою Windows Forms, що дозволяє швидко створити класичний десктопний графічний інтерфейс. Основна форма містить:

- меню товарів;
- секції для введення кількості;
- панель для формування і перегляду чеку;
- кнопки управління (підрахунок, очищення, друк, збереження, відкриття, вихід).

UX-дизайн орієнтований на спрощення дій персоналу – уся інформація згрупована в зрозумілому порядку, кнопки мають зрозумілі позначення та логічне розташування.

2.2 Вибір засобів, методів і алгоритмів вирішення поставленого завдання

Здійснимо огляд програмних інструментів, методів та алгоритмів, які можуть бути застосовані для реалізації інформаційної системи автоматизованого

обліку та обслуговування у кафе. Вибір технологій ґрунтується на необхідності забезпечення стабільної роботи, простоти користування, достатнього функціонального наповнення та можливості подальшої модифікації системи.

У процесі аналізу можливих технологічних рішень для створення десктопної інформаційної системи управління кафе було розглянуто декілька популярних платформ і мов програмування, які активно використовуються в розробці прикладного програмного забезпечення. Кожен із варіантів має свої особливості, переваги та обмеження, які важливо враховувати при виборі оптимального середовища розробки.

Python з бібліотекою Tkinter є одним з найпоширеніших варіантів для швидкого створення GUI-додатків. Перевагами Python є простий синтаксис, висока читаємість коду, велика кількість навчальних матеріалів, активна спільнота. Tkinter, як стандартна бібліотека для побудови графічного інтерфейсу в Python, дозволяє створювати прості вікна, кнопки, поля введення та меню. Проте її функціональність обмежена, особливо при потребі реалізувати більш гнучкий та інтуїтивно зрозумілий інтерфейс користувача. Крім того, відсутність зручного візуального конструктора інтерфейсу потребує більше ручної роботи з координатами елементів, що ускладнює розробку комплексних форм.

Java з використанням Swing або JavaFX є ще одним потужним інструментом для розробки графічних застосунків. Swing – це бібліотека, яка дозволяє створювати віконні додатки з широким набором компонентів, а JavaFX – новіша платформа, що підтримує сучасні UI-елементи, анімації, стилізацію через CSS. Серед основних переваг Java – кросплатформність (один і той самий застосунок працює на Windows, Linux і macOS), багатий набір інструментів для тестування, модульності й організації коду. Однак налаштування середовища розробки (наприклад, IntelliJ IDEA або Eclipse) потребує більше часу та досвіду, а побудова інтерфейсу – більше зусиль у порівнянні з Visual Studio. Крім того, Java-додатки часто потребують додаткової конфігурації при розгортанні.

C# з Windows Forms у Visual Studio було обрано як основну платформу реалізації проєкту завдяки оптимальному балансу між функціональністю, зручністю розробки та продуктивністю. C# – сучасна об'єктно-орієнтована мова програмування, що розроблена компанією Microsoft і тісно інтегрована з .NET-платформою. Вона забезпечує доступ до широкого спектру бібліотек для побудови інтерфейсу, обробки подій, роботи з файлами, масивами, колекціями та формування звітів. Windows Forms, як технологія створення графічних додатків, дозволяє візуально проєктувати інтерфейс за допомогою вбудованого дизайнера форм, що значно спрощує та прискорює процес розробки. Крім того, Visual Studio надає зручне середовище для редагування коду, відлагодження, підсвітки синтаксису, а також можливість легко перевіряти логіку через інтерактивні засоби тестування [7].

З урахуванням потреб розроблюваної системи – зручності інтерфейсу, швидкої взаємодії з користувачем, локального збереження інформації та підтримки обробки подій – вибір на користь C# у Visual Studio з Windows Forms є обґрунтованим і оптимальним. Такий підхід забезпечує не лише ефективне вирішення поставлених задач, а й дозволяє у подальшому масштабувати систему, розширювати її функціональність та адаптувати до змін у бізнес-процесах.

У межах реалізації системи управління кафе було обрано мову програмування C#, середовище розробки Visual Studio та інструментарій побудови графічного інтерфейсу Windows Forms. Ці технології забезпечують повний цикл розробки – від проєктування інтерфейсу до налагодження та тестування програмного забезпечення. Нижче подано обґрунтування вибору кожного з інструментів і приклад їх застосування в рамках розроблюваної системи.

Мова програмування C# – це високорівнева об'єктно-орієнтована мова, яка надає засоби для ефективного створення структурованих, розширюваних і підтримуваних програм. Однією з головних переваг C# є його інтеграція з платформою .NET, що забезпечує доступ до великої кількості вбудованих

бібліотек і класів для роботи з графікою, файлами, текстом, колекціями, мережею, базами даних тощо [8-9].

Завдяки підтримці об'єктно-орієнтованого підходу, логіка проєктованої системи реалізована у вигляді взаємодії окремих класів:

- Form1 – основний клас, у якому реалізовано графічний інтерфейс користувача (GUI) та обробку подій;
- Product – клас, який описує окремий товар, включаючи його назву, тип і вартість;
- OrderItem – клас, який поєднує товар і його кількість у поточному замовленні;
- Order – клас, що зберігає повний список позицій замовлення, виконує розрахунок загальної суми, податку, сервісного збору та забезпечує формування чеку.

Середовище розробки Visual Studio – це потужний інструмент, який поєднує функції текстового редактора, дизайнера графічних форм, компілятора та налагоджувача. Однією з ключових переваг Visual Studio є вбудований дизайнер Windows Forms, що дозволяє реалізовувати інтерфейс користувача у візуальному режимі, додаючи елементи керування (кнопки, текстові поля, списки, мітки) простим перетягуванням [10].

Visual Studio також надає:

- зручне автозавершення коду (IntelliSense);
- інтерактивне виявлення помилок;
- налагодження з точками зупину (breakpoints);
- графічне представлення об'єктів у пам'яті під час виконання.

Windows Forms – це бібліотека для побудови графічних інтерфейсів у середовищі Windows, яка надає повноцінну підтримку для створення десктопних програм [11]. У межах реалізації системи управління кафе Windows Forms використано для формування зручного GUI, у якому користувач може:

- вибрати потрібні позиції меню (напої, торти) за допомогою чекбоксів;
- ввести кількість одиниць кожного товару;

- натиснути кнопку для обчислення підсумкової вартості;
- переглянути згенерований чек у текстовому полі;
- зберегти або роздрукувати сформований чек;
- очистити форму для нового замовлення.

У результаті, вибір C#, Visual Studio та Windows Forms дозволив реалізувати систему, яка відповідає вимогам щодо продуктивності, простоти використання, підтримки обробки подій, а також можливості масштабування у разі майбутнього розширення функціоналу [12].

У програмі реалізовано логіку обчислення суми замовлення шляхом обробки значень, введених користувачем, множення ціни одиниці товару на кількість і додавання сервісного збору. Загальна структура реалізації представлена у вигляді блок-схеми (рис. 2.1).



Рисунок 2.1 – Блок-схема алгоритму розрахунку загальної суми

Алгоритм містить етапи:

- зчитування значень чекбоксів та текстових полів;
- розрахунок вартості кожної позиції;
- підсумовування сум;
- обчислення податку та додавання сервісного збору;
- формування чеку у вигляді рядків тексту.

Система моделюється у вигляді:

- Use Case diagram – для опису взаємодії користувача з системою (наприклад: «Оформити замовлення», «Розрахувати суму», «Зберегти чек», «Очистити форму»);
- Class diagram – для структурування об'єктної моделі: класи Coffee, Cake, Receipt, MainForm;
- Activity diagram – для опису логіки взаємодії з елементами GUI та сценарію формування замовлення.

Діаграма прецедентів (Use Case Diagram) є одним із ключових засобів візуалізації вимог до інформаційної системи. Вона дозволяє графічно представити взаємодію користувачів (акторів) із системою через набір функціональних можливостей, які реалізовані у вигляді прецедентів (використань, сценаріїв). Така діаграма дає змогу визначити межі системи, сформувати уявлення про її зовнішні інтерфейси та забезпечує основу для подальшого проектування класів, активностей, компонентів.

У контексті розроблюваної системи управління кафе основним користувачем системи є персонал кафе (офіціант або касир), який виконує основні операції під час обслуговування клієнтів. З урахуванням результатів аналізу предметної області, інтерв'ювання користувачів та вивчення аналогічних систем, було сформовано перелік основних функцій (прецедентів), які повинна виконувати система:

- «Формувати замовлення» – користувач відмічає у графічному інтерфейсі вибрані позиції меню та вводить їх кількість;

– «Обчислити підсумок» – система на основі введених даних виконує розрахунок вартості кожної позиції, обчислює загальну суму замовлення з урахуванням податків та сервісного збору;

– «Сформувати чек» – система генерує текстовий варіант чека, який містить повну інформацію про замовлення, включно з найменуванням товарів, цінами, кількістю, підсумком та датою;

– «Зберегти чек» – користувач може зберегти сформований чек у локальний текстовий файл для подальшого використання (наприклад, архівування або повторне відкриття);

– «Друкувати чек» – передбачено можливість фізичного друку чека з програми;

– «Очистити форму» – після завершення замовлення персонал може обнулити введені дані для наступного клієнта;

– «Вийти з програми» – закриття додатку після завершення робочої зміни або у разі технічної необхідності.

Кожен із цих прецедентів є логічною одиницею функціональності, яка реалізована в межах окремих методів або обробників подій у програмі (рис. 2.2).

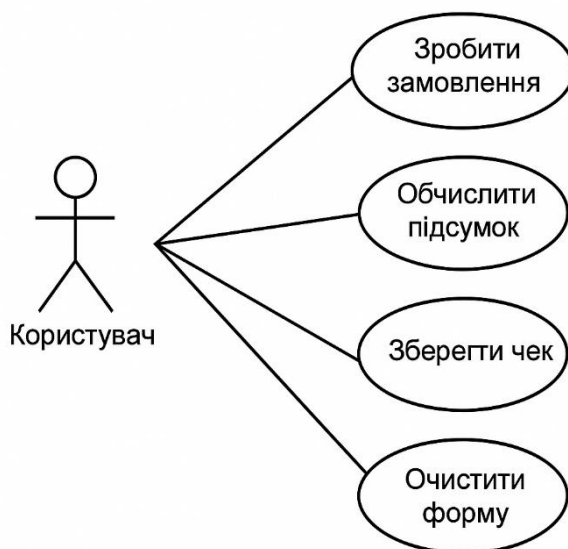


Рисунок 2.2 – Діаграма прецедентів

Діаграма класів (Class Diagram) відображає структуру системи з погляду об'єктно-орієнтованого програмування. Вона демонструє основні класи, їх атрибути, методи та зв'язки між ними. Зокрема, показано клас інтерфейсу Form1, клас замовлення Order, елемент замовлення OrderItem та товар Product. Така модель дозволяє визначити логіку взаємодії об'єктів під час виконання операцій над замовленнями та чеками (рис. 2.3).

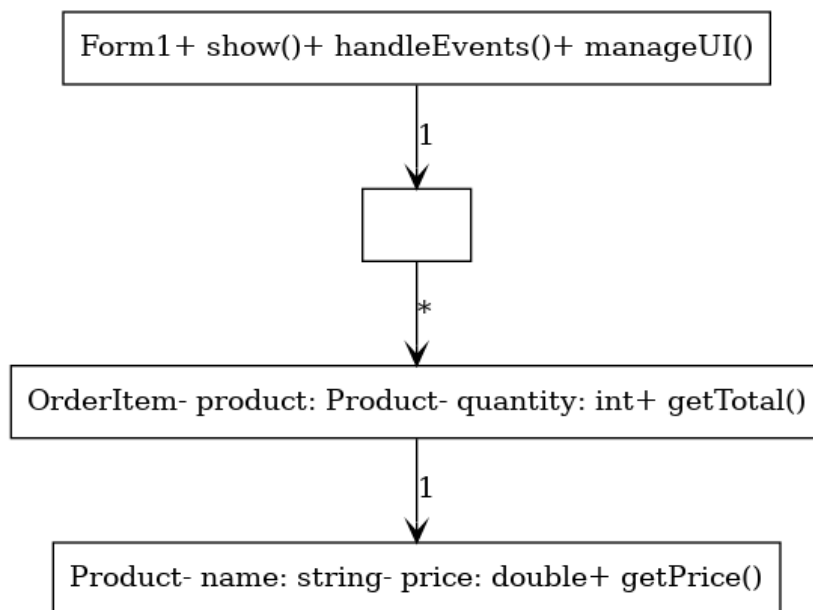


Рисунок 2.3 – Діаграма класів

Діаграма активностей (Activity Diagram) відображає послідовність дій користувача під час оформлення замовлення в системі управління кафе. Процес починається з вибору продуктів і заповнення форми замовлення. Далі користувач підтверджує замовлення, після чого система виконує розрахунок підсумкової вартості, включаючи податки. Потім формується чек, який зберігається у файл або виводиться на екран. Завершальним етапом є очищення форми для нового замовлення. На діаграмі також передбачено умовні переходи, які демонструють варіанти розвитку процесу – наприклад, при незаповнених обов'язкових полях система повертає користувача до попереднього кроку. У цілому діаграма (рис. 2.4) відображає логіку основного бізнес-процесу в кафе та можливі відхилення залежно від дій користувача.

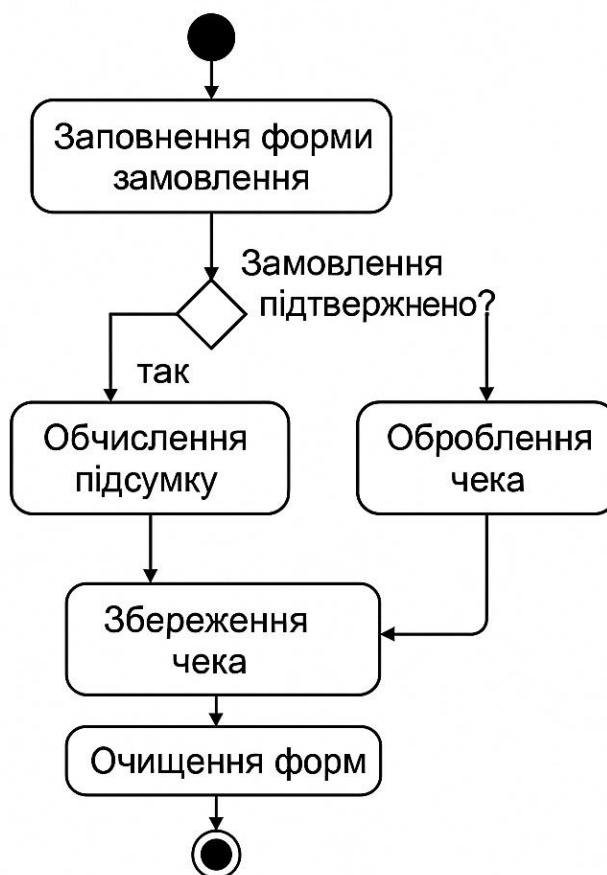


Рисунок 2.4 – Діаграма активностей

Моделі чітко відображають функціональну структуру системи та відображають ключові логічні етапи взаємодії з користувачем. Класова модель спрощує повторне використання коду і дозволяє легко розширити функціонал у майбутньому, наприклад, додати авторизацію персоналу або базу клієнтів. Використання діаграм UML допомогло систематизувати вимоги і перевірити їхню реалізованість у програмному коді.

Обрані засоби дозволяють реалізувати ефективне рішення задачі автоматизації процесів у кафе та забезпечують зручність подальшої підтримки системи.

РОЗДІЛ 3

РОЗРОБКА СИСТЕМИ УПРАВЛІННЯ КАФЕ

3.1 Практична реалізація об'єкта проектування

Опишемо практичний етап реалізації програмного забезпечення – від побудови інтерфейсу до написання ключових фрагментів коду, що забезпечують функціонування програми «Система управління кафе». Розробка здійснювалась у середовищі Microsoft Visual Studio з використанням мови програмування C# та бібліотеки Windows Forms, що дозволяє створювати класичні графічні інтерфейси користувача для платформи Windows.

На першому етапі було створено вікно додатку з базовими елементами керування: чекбоксами для вибору товарів (напоїв та десертів), текстовими полями для введення кількості обраних позицій, мітками (Label) для відображення вартості, кнопками управління – «Total», «Receipt», «Save», «Print», «Reset», «Exit» тощо. GUI формувався візуально за допомогою конструктора Windows Forms, що дало змогу швидко компоувати елементи на формі. Після формування інтерфейсу, за кожним елементом управління було закріплено відповідні обробники подій. Наприклад, кнопка «Відкрити» відкриває файл із збереженим чеком у діалоговому вікні (ліст. 3.1).

Лістинг 3.1 – Кнопка «Відкрити»

```
private void openToolStripButton_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog();
    openFileDialog.Filter = "Text Files (*.txt)|*.txt|All files (*.*)|*.*";
    if (openFileDialog.ShowDialog() == DialogResult.OK)
        rtfReceipt.LoadFile(openFileDialog.FileName,
        RichTextBoxStreamType.PlainText);
}
```

кінець лістингу 3.1

Для збереження чеку у текстовий файл застосовано компонент SaveFileDialog (ліст. 3.2).

Лістинг 3.2 – Збереження чеку у текстовий файл

```
private void saveToolStripButton_Click(object sender, EventArgs e)
{
    SaveFileDialog saveFile = new SaveFileDialog();
    saveFile.FileName = "Notepad Text";
    saveFile.Filter = "Text Files (*.txt)|*.txt|All files (*.*)|*.*";

    if (saveFile.ShowDialog() == DialogResult.OK)
    {
        using (StreamWriter sw = new
StreamWriter(saveFile.FileName))
            sw.WriteLine(rtfReceipt.Text);
    }
}
```

кінець лістингу 3.2

Функціонал друку реалізований за допомогою стандартного компонента `printPreviewDialog` (ліст. 3.3).

Лістинг 3.3 – Компонент `printPreviewDialog`

```
private void printToolStripButton_Click(object sender,
EventArgs e)
{
    printPreviewDialog1.Document = printDocument1;
    printPreviewDialog1.ShowDialog();
}
```

кінець лістингу 3.3

Функція очищення полів використовується для скидання інформації після завершення замовлення (ліст. 3.4).

Лістинг 3.4 – Скидання інформації після завершення замовлення

```
private void newToolStripButton_Click(object sender, EventArgs
e)
{
    rtfReceipt.Clear();
}
```

кінець лістингу 3.4

Опишемо реалізацію обчислення вартості замовлення. Центральним елементом логіки є обробка кнопки «Total», яка обчислює повну вартість замовлення на основі введених кількостей та фіксованих цін. Алгоритм передбачає:

- зчитування значень з полів введення;
- множення ціни одиниці товару на кількість;
- підсумовування всіх вартостей;
- додавання сервісного збору та ПДВ;
- відображення результатів у відповідних полях.

Сутність алгоритму представлена у наведеному нижче уривку (ліст. 3.5).

Лістинг 3.5 – Обчислення вартості замовлення

```
double latte = 1.20;
double qtyLatte = Convert.ToDouble(txtLate.Text);
double costLatte = latte * qtyLatte;
...
lblDrink.Text = String.Format("{0:C}", cost_of_drinks);
lblCake.Text = String.Format("{0:C}", cost_of_cakes);
lblTotal.Text = String.Format("{0:C}", (cost_of_cakes +
cost_of_drinks + tax));
```

кінець лістингу 3.5

Для кожного продукту передбачено керування станом текстового поля залежно від вибору чекбоксу (ліст. 3.6).

Лістинг 3.6 – Керування станом текстового поля

```
double latte = 1.20;
double qtyLatte = Convert.ToDouble(txtLate.Text);
double costLatte = latte * qtyLatte;
...
lblDrink.Text = String.Format("{0:C}", cost_of_drinks);
lblCake.Text = String.Format("{0:C}", cost_of_cakes);
```

```
lblTotal.Text = String.Format("{0:C}", (cost_of_cakes +  
cost_of_drinks + tax));
```

кінець лістингу 3.6

У розробленій системі управління кафе застосовано об'єктно-орієнтований підхід, що забезпечує ефективну структурування програмного забезпечення за допомогою створення окремих класів. Наприклад, реалізовано клас *Coffee*, який представляє певну одиницю продукції. Така модель дозволяє чітко розмежувати функціональні обов'язки кожного компонента системи, зберігати логіку розрахунків у відповідному класі, а також забезпечує можливість багаторазового використання коду без потреби дублювання функціональності. Подібний підхід відповідає сучасним вимогам до проектування програмних систем і є ефективним рішенням у контексті розробки бізнес-застосунків [13-14].

Створений програмний продукт наочно демонструє засвоєння ключових понять об'єктно-орієнтованого програмування. Зокрема, у проєкті застосовано інкапсуляцію, яка дозволяє приховувати внутрішні механізми реалізації класів та взаємодіяти з ними лише через визначені інтерфейси. Код програми чітко розділений за функціональними ознаками, що дозволяє зберігати незалежність компонентів і забезпечує поділ відповідальностей між ними.

Програма системи управління кафе орієнтована на автоматизацію робочих процесів, що відбуваються в кафе, і надає можливість швидко та зручно формувати замовлення. Ключова ідея системи полягає в автоматичному створенні чека при виборі клієнтом певних товарів або послуг. Система обчислює загальну суму замовлення, показує суму податку, вартість без урахування податку, а також кінцеву суму до сплати.

Головне вікно програми, зображене на рисунку 3.1, є центральним елементом користувацького інтерфейсу. У ньому реалізовано можливість вибору продуктів, введення їх кількості, додавання до замовлення, очищення форми або підтвердження покупки.

Cafe Management Systems

21 грудня 2023 р.

☐ Latte	0	☐ Coffee Cake	0
☐ Espresso	0	☐ Red Velvet Cake	0
☐ Iced Latte	0	☐ Black Forest Cake	0
☐ Vale Coffee	0	☐ Boston Cream Cake	0
☐ Cappuccino	0	☐ Lagos Chocolate Cake	0
☐ African Coffee	0	☐ Kilburn Chocolate Cake	0
☐ Iced - Cappuccino	0	☐ Carlton Hill Chocolate Cake	0

Cost of Drinks	0	Tax	
Cost of Cakes	0	SubTotal	
Service Charge	1,75	Total	

Total Receipt Reset Exit

Рисунок 3.1 – Головне вікно програми

У програмі можна дізнатися ціну продуктів вибравши їх та натиснувши кнопку «Total» і знизу відобразиться ціна на вибрані продукти. Вона також поділена на такі категорії як ціна на напої або ціна на торти (рис. 3.2).

Cafe Management Systems

21 грудня 2023 р.

☒ Latte	1	☐ Coffee Cake	0
☐ Espresso	0	☐ Red Velvet Cake	0
☐ Iced Latte	0	☐ Black Forest Cake	0
☐ Vale Coffee	0	☒ Boston Cream Cake	2
☐ Cappuccino	0	☐ Lagos Chocolate Cake	0
☐ African Coffee	0	☒ Kilburn Chocolate Cake	3
☐ Iced - Cappuccino	0	☐ Carlton Hill Chocolate Cake	0

Cost of Drinks	1,20 ₴	Tax	0,05 ₴
Cost of Cakes	9,10 ₴	SubTotal	12,05 ₴
Service Charge	1,75 ₴	Total	12,10 ₴

Total Receipt Reset Exit

Рисунок 3.2 – Ціна на продукти

Після натиснення кнопки «Receipt» справа у вікні буде показано чек зі всіма продуктами які було придбано. Це показано на рисунку 3.3.

The screenshot shows the 'Cafe Management Systems' application window. The title bar indicates 'Form1'. The interface is divided into several sections:

- Header:** Includes icons for a sun, a cocktail glass, and a plate with a fork, along with the time '1:18:50' and the date '21 грудня 2023 р.'.
- Product Selection:** Two columns of items with checkboxes and quantity input fields.

Item	Quantity
☒ Latte	1
☐ Espresso	0
☐ Iced Latte	0
☐ Vale Coffee	0
☐ Cappuccino	0
☐ African Coffee	0
☐ Iced - Cappuccino	0
☐ Coffee Cake	0
☐ Red Velvet Cake	0
☐ Black Forest Cake	0
☒ Boston Cream Cake	2
☐ Lagos Chocolate Cake	0
☒ Kilburn Chocolate Cake	3
☐ Carlton Hill Chocolate Cake	0
- Costs:**

Cost of Drinks	1,20 ₴
Cost of Cakes	9,10 ₴
Service Charge	1,75 ₴
- Tax and Totals:**

Tax	0,05 ₴
SubTotal	12,05 ₴
Total	12,10 ₴
- Receipt Window:** A separate window titled 'Just Do Cafe' showing a detailed receipt:

Item	Quantity
Latte	1
Espresso	0
Iced Latte	0
Vale Coffee	0
Cappuccino	0
African Coffee	0
Iced Cappuccino	0
Coffee Cake	0
Red Velvet Cake	0
Black Forest Cake	0
Boston Cream Cake	2
Lagos Chocolate Cake	0
Kilburn Chocolate Cake	3
Carlton Hill Chocolate Cake	0
Tax	0,05 ₴
Sub Total	12,05 ₴
Total Cost	12,10 ₴
- Buttons:** 'Total', 'Receipt' (highlighted), 'Reset', and 'Exit'.

Рисунок 3.3 – Чек з вибраними продуктами

Чеком можна проводити різноманітні операції, такі як збереження, видалення, друк, відправлення та інші дії. Розглянемо деякі з них, наприклад зберігання чеку на комп'ютері. Для цього потрібно над чеком натиснути на кнопку зберегти після цього випаде меню збереження файлу на комп'ютері.

Ще одна дія – це друк чеку. Для цього потрібно натиснути на іконку принтера яка розташована зверху чеку. Після цього випаде вікно (рис. 3.5), де можна підготувати все до друку.

The screenshot shows the 'Print preview' window. It contains a detailed receipt for 'Just Do Cafe' with the following items and quantities:

Item	Quantity
Latte	1
Espresso	0
Iced Latte	0
Vale Coffee	0
Cappuccino	0
African Coffee	0
Iced Cappuccino	0
Coffee Cake	0
Red Velvet Cake	0
Black Forest Cake	0
Boston Cream Cake	2
Lagos Chocolate Cake	0
Kilburn Chocolate Cake	3
Carlton Hill Chocolate Cake	0
Tax	0,05 ₴
Sub Total	12,05 ₴
Total Cost	12,10 ₴

The receipt also includes the date and time: '1:19:51 21 грудня 2023 р.'.

Рисунок 3.5 – Меню друку чеку

Зліва знизу є 4 кнопки які відповідають за різноманітні дії. Дві уже було розглянути, а інші це «Reset» та «Exit». Де «Reset» призначена для скидання чеку, а «Exit» для виходу з програми. Зображені вони на рисунку 3.6.

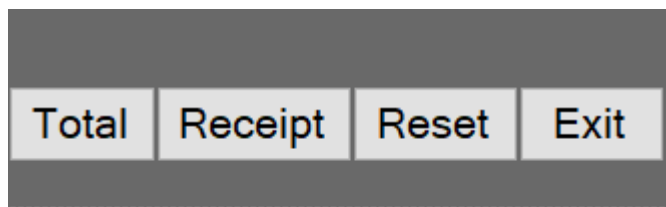


Рисунок 3.6 – Кнопки дій

3.2 Тестування та налагодження системи

Після завершення основного етапу розробки інформаційної системи управління кафе на мові C# було розпочато процес її тестування й налагодження. Цей етап мав на меті перевірити відповідність роботи системи до визначених вимог, забезпечити її стабільність та усунути виявлені помилки до моменту впровадження [15].

Метою тестування є підтвердження правильності реалізації функціональних можливостей системи, а також перевірка її поведінки при введенні некоректних або граничних значень. Окрему увагу приділено перевірці роботи обробників подій, коректності математичних розрахунків, збереження та завантаження даних, а також стабільності GUI-компонентів.

Завданнями тестування було:

- перевірити правильність обчислень при різному наборі введених значень;
- виявити помилки, пов'язані з невірним типом введених даних (символи, порожні поля тощо);
- перевірити, як поводить себе програма при повторних діях (наприклад, кілька разів підряд натискати «Очистити» або «Total»);
- перевірити роботу модулів збереження та відкриття файлів;

– виявити можливі візуальні або логічні помилки в інтерфейсі користувача.

У процесі тестування були застосовані такі методи:

– функціональне тестування – перевірка відповідності реалізованих функцій вимогам до системи. Проводилося вручну, згідно із попередньо складеним переліком функцій і сценаріїв;

– тестування на коректність введення – здійснювалася спроба ввести текстові, спеціальні або порожні значення замість чисел;

– тестування стійкості – багаторазове виконання однотипних дій (наприклад, натискання кнопок «Підрахувати» або «Очистити») з метою перевірки, чи не призводить це до збоїв;

– UX-тестування – оцінка зручності інтерфейсу, інтуїтивності розташування елементів, логіки роботи кнопок і коректності підписів [16].

Нижче наведено приклади кількох ключових сценаріїв, що використовувалися під час перевірки (табл. 3.1).

Таблиця 3.1 – Текстові сценарії

№	Сценарій	Вхідні дані	Очікуваний результат	Фактичний результат
1	Обчислення замовлення	2 кави, 1 торт	Відображення правильної суми	Успішно
2	Введення тексту в поле	«abc»	Виведення повідомлення про помилку	Успішно
3	Збереження чеку	Чек сформовано	Виведення діалогу збереження, створення .txt-файлу	Успішно
4	Завантаження файлу з	Вибрано збережений чек	Виведення вмісту в поле RichTextBox	Успішно
5	Повторне натискання «Очистити»	Всі поля порожні	Немає помилок, поля не змінюються	Успішно
6	Натискання «Друкувати» без чеку	Порожнє поле	Виведення повідомлення про відсутність вмісту	Успішно

Під час тестування було виявлено декілька недоліків, які були усунені на етапі налагодження:

- помилка перетворення типу (FormatException) при спробі ввести текст у поле кількості. Було додано перевірку `double.TryParse(...)` з відповідним повідомленням користувачеві;
- залишкові значення в полі після зняття чекбоксу. Вирішено за допомогою логіки: якщо чекбокс знято, то поле `Enabled = false`, а значення встановлюється на «0»;
- відсутність перевірки на порожнє поле чеку перед збереженням. Упроваджено перевірку `if (rtfReceipt.Text.Length == 0)` перед ініціалізацією `SaveFileDialog`;
- декілька натискань кнопки «Total» призводили до додавання значень до попереднього результату. Виправлено через попереднє обнулення змінних перед розрахунком;
- міжнародний формат чисел (крапка або кома). Забезпечено правильне перетворення чисел незалежно від регіональних налаштувань системи через `CultureInfo.InvariantCulture`.

Для коректної роботи системи необхідне дотримання таких вимог:

- операційна система: Windows 10 або вище;
- процесор: Intel Pentium або аналог з частотою не менше 1.6 ГГц;
- оперативна пам'ять: від 2 ГБ;
- жорсткий диск: не менше 100 МБ вільного місця;
- програмне забезпечення: .NET Framework 4.7 або новіша версія.

Рекомендоване середовище виконання:

- монітор: з роздільною здатністю від 1280×720 пікселів для повного відображення форми;
- наявність текстового редактора або переглядача .txt-файлів (наприклад, Notepad++);
- принтер (опційно) – для фізичного друку сформованого чеку.

У результаті проведеного тестування було підтверджено функціональну правильність реалізованих модулів системи управління кафе. Виявлені недоліки усунуто, а система показала високу стабільність під час експлуатації. Графічний інтерфейс працює без збоїв, забезпечуючи зручну взаємодію користувача з програмою. Отже, програмний продукт готовий до використання в умовах реального середовища закладу громадського харчування.

3.3 Розробка документації для програмного продукту

Для ефективного впровадження та використання розробленої системи управління кафе необхідно надати користувачеві документацію, що містить системні вимоги, інструкцію з використання, опис інтерфейсу програми та основних функцій, а також пояснення щодо збереження і відкриття чеків.

Для стабільної роботи програмного забезпечення необхідно дотримуватись таких мінімальних вимог:

- операційна система: Windows 10 / 11 (x64 або x86);
- процесор: Intel Pentium або AMD Athlon з частотою не менше 1.6 ГГц;
- оперативна пам'ять: 2 ГБ і більше;
- жорсткий диск: щонайменше 100 МБ вільного простору;
- програмна платформа: .NET Framework 4.7 або новіша;
- принтер: опційно, для друку сформованих чеків;
- текстовий редактор: для перегляду збережених чеків (наприклад, Notepad, Notepad++).

Сформулюємо інструкцію користувача: як працювати з додатком:

- запуск програми: двічі клацніть на виконуваному файлі CafeManager.exe (якщо створено інсталяційний пакет – інсталюйте через інсталятор);
- вибір позицій меню: встановіть прапорці (checkbox) біля потрібних товарів (напоїв або десертів), які бажає замовити клієнт;

- введення кількості: у відповідні текстові поля введіть кількість одиниць кожного обраного товару;
- обчислення вартості: натисніть кнопку «Total» – буде автоматично обчислено вартість кожної позиції та підсумкову суму з урахуванням сервісного збору;
- формування чеку: натисніть кнопку «Receipt» – програма згенерує текст чеку в полі RichTextBox;
- збереження чеку: натисніть кнопку «Save», виберіть місце збереження на комп'ютері та задайте ім'я файлу;
- друк чеку: натисніть «Print», і з'явиться вікно попереднього перегляду перед друком;
- очищення форми: натисніть «Reset» для скидання всіх полів та підготовки до нового замовлення;
- завершення роботи: натисніть «Exit» для закриття програми.

У таблиці 3.2 подано перелік основних кнопок користувацького інтерфейсу, що реалізовані у програмі «Система управління кафе», разом із відповідним функціональним призначенням кожної з них.

Таблиця 3.2 – Опис кнопок, інтерфейсу, можливих помилок

Назва кнопки	Призначення
Total	Обчислення вартості замовлення
Receipt	Формування текстового чеку
Save	Збереження чеку у форматі .txt
Print	Відкриття діалогу попереднього перегляду для друку
Reset	Очищення введених даних
Exit	Завершення роботи програми

Можливі помилки та поради:

- введення літер замість чисел у поле кількості – програма виведе повідомлення про помилку;
- відсутність обраних товарів або значення «0» – сума буде нульова;
- спроба зберегти або роздрукувати чек до його створення – програма не допустить виконання дії.

Якщо форма не оновлена після попереднього клієнта – потрібно використовувати кнопку «Reset».

Розглянемо як зберігати / відкривати чеки.

Збереження чеку:

- після формування чеку натисніть кнопку «Save»;
- у вікні збереження вказати назву файлу та місце на диску;
- чек збережеться у форматі .txt.

Відкриття раніше збереженого чеку:

- обрати меню File → Open або використайте кнопку Open;
- у діалоговому вікні відкрийте файл чеку;
- зміст буде завантажено у текстове поле RichTextBox для перегляду або редагування.

3.4 UX/UI-аналіз розробленої системи

Проектування користувацького інтерфейсу (UI) та забезпечення позитивного досвіду взаємодії користувача з програмою (UX) є важливими аспектами при створенні програмного забезпечення для закладів громадського харчування. Система повинна бути зрозумілою, швидкою у використанні, не перевантаженою зайвими елементами та адаптованою до специфіки повсякденної роботи персоналу кафе.

Обґрунтуємо розташування елементів. Інтерфейс програми реалізовано за допомогою бібліотеки Windows Forms, що дозволяє гнучко розміщувати елементи керування на формі. Основна форма поділена на кілька логічних блоків:

- лівий блок містить перелік напоїв, де кожна позиція представлена чекбоксом для вибору та текстовим полем для введення кількості;
- правий блок відведений для десертів і організований аналогічно;
- центральна нижня частина містить панель для підрахунку вартості, відображення підсумків (вартість напоїв, десертів, податок, загальна сума);

– праворуч розташоване текстове поле RichTextBox для виводу сформованого чеку та кнопки керування (Total, Receipt, Save, Print, Reset, Exit).

Таке компонування забезпечує логічну послідовність дій: вибір → введення → підрахунок → чек → збереження/друк → очищення. Користувачу не потрібно переміщатися між вкладками чи шукати функції — усе розташовано на одній формі у межах одного екрана, що мінімізує кількість кліків і прискорює обслуговування клієнтів.

Дизайн системи побудовано на принципах мінімалізму та функціональності. Колірна схема витримана у нейтральних тонах (світлий фон, темні кнопки), що забезпечує хорошу читабельність тексту незалежно від освітлення робочого місця.

Шрифт: стандартний Windows Sans Serif 10pt для всіх написів і полів введення.

Кнопки мають однаковий розмір (зручні для натискання як мишею, так і на сенсорному екрані), вирівняні по правій стороні.

Підписи кнопок – максимально короткі й інформативні: Total, Receipt, Save, Print, Reset, Exit (можна адаптувати на українську: «Підрахувати», «Чек», «Зберегти», «Друк», «Очистити», «Вихід»).

Чекбокси підписані назвами товарів (Latte, Espresso, Tiramisu тощо), а текстові поля мають значення за замовчуванням «0», яке скидається після вибору позиції.

Такий підхід відповідає принципам user-friendly інтерфейсу: користувач завжди бачить, що він обрав, скільки одиниць ввів і яка підсумкова сума вийшла. Всі дії передбачувані й легко відтворювані навіть без інструкції.

Для перевірки ефективності UX-дизайну було проведено спостереження за тестовим використанням програми студентами й викладачами, які діяли в ролі персоналу кафе. Було визначено, що час виконання замовлення в середньому 30-45 секунд при використанні програми проти 2-3 хвилин при ручному записі; кількість дій зменшено за рахунок автоматизації обчислення й друку.

Рівень задоволеності інтерфейсом – позитивний. Тестувальники відзначили простоту користування, зручне розміщення кнопок та відсутність необхідності прокручування чи перемикання між вікнами.

Також було проаналізовано зворотний зв'язок: усі користувачі змогли виконати повний сценарій замовлення без додаткової допомоги. Це підтвердило, що інтерфейс інтуїтивно зрозумілий і підходить для працівників без спеціальної комп'ютерної підготовки.

Таким чином, у розробці системи було реалізовано сучасний підхід до проєктування UX/UI, який забезпечує ефективну взаємодію користувача з додатком, високу продуктивність, інтуїтивну логіку та зменшення часу навчання. Це особливо важливо для програм, що використовуються в динамічному середовищі закладів обслуговування.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи на тему «Розробка системи управління кафе з використанням C#» було досягнуто поставлену мету – створено функціональну програмну систему, яка дозволяє автоматизувати процес обробки замовлень у кафе, формування чеків, їх збереження та друк.

У ході дослідження були реалізовані такі завдання:

- проведено аналіз сучасних підходів до автоматизації процесів у сфері громадського харчування. Досліджено переваги і недоліки наявних рішень, зокрема Poster POS і Floreant POS, а також вивчено наукові джерела щодо вимог до подібних інформаційних систем. Це дало змогу обґрунтувати необхідність створення нової системи, яка краще відповідала б специфіці невеликих кафе та не потребувала складного налаштування чи підключення до Інтернету;

- визначено та обґрунтовано вимоги до системи управління кафе. Здійснено інтерв'ювання персоналу кафе, проаналізовано сценарії використання та сформовано набір функціональних прецедентів. На основі цього було розроблено структуровану специфікацію вимог до майбутнього програмного продукту;

- вибрано оптимальні технології та засоби для розробки програмного забезпечення. Проаналізовано потенційні платформи – Python (Tkinter), Java (Swing/JavaFX), C# (Windows Forms). Обрано мову C# і середовище Visual Studio як найбільш відповідні вимогам проєкту щодо зручності розробки, стабільності та простоти розгортання;

- розроблено програмне забезпечення на мові C# з використанням Windows Forms. Створено повноцінний десктопний додаток із графічним інтерфейсом, що дозволяє здійснювати вибір товарів, обчислення вартості замовлення, збереження й друк чеків. Реалізовано логіку обробки подій, контролю введення даних, роботи з файлами. Код організовано згідно з принципами об'єктно-орієнтованого програмування;

– здійснено тестування та налагодження створеного програмного продукту. Проведено ручне функціональне тестування за низкою сценаріїв, виявлено типові помилки (обробка нечислових значень, помилки при повторному натисканні кнопок), які були успішно усунені. Система пройшла перевірку на стабільність і працездатність;

– оцінено ефективність та зручність експлуатації розробленої системи. Інтерфейс програми виявився інтуїтивно зрозумілим, з логічним розташуванням елементів. Результати тестування підтвердили зручність користування, коректність роботи та відповідність очікуванням потенційного користувача (персоналу кафе).

Таким чином, результати роботи повністю відповідають поставленим завданням, а створена система демонструє практичну цінність і може бути впроваджена в реальних умовах роботи кафе малого формату.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Король А. Бізнес-процеси ресторанного бізнесу: сучасні підходи до аналізу та оптимізації. Київ: Видавництво «Київський університет», 2020. 210 с.
2. Smith J. Modern Cafe Business Processes: Analysis and Improvement. New York: Coffee Books Publishing, 2021. 290 с.
3. Poster POS – Хмарна POS-система для кафе та ресторанів. URL: <https://joinposter.com/ua> (дата звернення: 10.04.2025).
4. Floreant POS – Open Source Point of Sale for Restaurants. URL: <https://floreant.org/> (дата звернення: 10.05.2025).
5. Коваленко О. С., Добровська Л. М. Проектування інформаційних систем. Загальні питання теорії проектування ІС: навч. посіб. Київ: КПІ ім. Ігоря Сікорського, 2020. 192 с.
6. Трофименко О. Г., Манаков С. Ю., Ларін Д. Г. Основи програмної інженерії: навч.-метод. Посібник. Одеса: Фенікс, 2022. 194 с.
7. Microsoft Docs – Documentation for .NET and C#. URL: <https://learn.microsoft.com/en-us/dotnet/csharp/> (дата звернення: 10.04.2025).
8. Stack Overflow. Обговорення реалізації чеків у C# Windows Forms. URL: <https://stackoverflow.com/> (дата звернення: 15.05.2025).
9. Andrew Troelsen, Phil Japikse. Pro C# 10 with .NET 6. Foundational Principles and Practices in Programming. Berkley, United States: Apress, 2022. 1705 p.
10. Коноваленко І. В., Марущак П. О. Платформа .NET та мова програмування C# 8.0: навчальний посібник. Тернопіль: ФОП Паляниця В. А., 2020. 320 с.
11. MSDN Library – Visual Studio Documentation. URL: <https://learn.microsoft.com/en-us/visualstudio/> (дата звернення: 15.05.2025).
12. Брила А. Ю., Ломага М. М., Антосяк П. П. Основи C#: навчальний посібник. Ужгород, 2023. 130 с.

13. C#. URL: <https://w3schoolsua.github.io/cs/index.html#gsc.tab=0> (дата звернення: 15.05.2025).

14. Гришанович Т. О., Глинчук Л. Я. Основи об'єктно-орієнтованого програмування: навч. посібник. Луцьк: ВНУ імені Лесі Українки, 2022. 120 с

15. ISO/IEC/IEEE 12207:2017 – Systems and software engineering – Software life cycle processes. URL: https://online.budstandart.com/ua/catalog/doc-page.html?id_doc=77957 (дата звернення: 17.05.2025).

16. Трофименко О. Г., Дика А. І. Тестування та забезпечення якості програмних систем: навч. посіб. Нац. ун-т «Одес. юрид. академія». Одеса: Фенікс, 2024. 195 с. URL: <https://hdl.handle.net/11300/27717> (дата звернення: 17.05.2025).